



DECIMOSEXTO WORKSHOP

Avances en Informática y
automática

Avances en Informática y Automática

Decimosexto Workshop

Avances en Informática y Automática

Decimosexto Workshop

Editor

Sergio García González

Publicado en España por:

Departamento de Informática y Automática Facultad de Ciencias
Universidad de Salamanca
Plaza de los Caídos s/n
37008, Salamanca, España
Tel.: + 34 923294653
Fax: + 34 923294514
Web: <http://mastersi.usal.es>
Email: mastersi@usal.es

ISBN 978-84-09-58892-3

Editores:

Sergio García González

Prólogo

El Máster Oficial en Sistemas Inteligentes de la Universidad de Salamanca tiene como principal objetivo promover la iniciación de los estudiantes en el ámbito de la investigación. El congreso organizado por el Departamento de Informática y Automática que se celebra dentro del Máster en Sistemas Inteligentes de la Universidad de Salamanca proporciona la oportunidad ideal para que sus estudiantes presenten los principales resultados de sus Trabajos de Fin de Máster y obtengan una realimentación del interés de los mismos.

La novena edición del *workshop* “Avances en Informática y Automática”, correspondiente al curso 2022 - 2023, ha sido un encuentro interdisciplinar donde se han presentado trabajos pertenecientes a un amplio abanico de líneas de investigación, desde los sistemas multiagente y la visualización de la información hasta la minería de datos pasando por otros campos relacionados. Todos los trabajos han sido supervisados por investigadores de reconocido prestigio pertenecientes a la Universidad de Salamanca, proporcionando el marco idóneo para sentar las bases de una futura tesis doctoral. Entre los principales objetivos del congreso se encuentran:

- Ofrecer a los estudiantes un marco donde exponer sus primeros trabajos de investigación.
- Proporcionar a los participantes un foro donde discutir ideas y encontrar nuevas sugerencias de compañeros, investigadores y otros asistentes a la reunión.
- Permitir a cada estudiante una realimentación de los participantes sobre su trabajo y una orientación sobre las futuras direcciones de investigación.
- Contribuir al desarrollo del espíritu de colaboración en la investigación.

Organización

El *workshop* “Avances en Informática y Automática” está organizado por el Departamento de Informática y Automática de la Universidad de Salamanca.

Comité Organizativo

María Gracia Manzano Arjona
María N. Moreno García
Juan M. Corchado Rodríguez
Carlos García-Figuerola Paniagua
Angélica González Arrieta
Francisco J. García Peñalvo
José L. Alonso Berrocal

Índice general

Sistema de recomendación de listas de reproducción de música	1
<i>Álvaro Gómez Gutiérrez, María N. Moreno García, María Dolores Muñoz Vicente</i>	
Aplicación de un COBOT para pick-and-place inteligente	26
<i>David Cruz García, Vidal Moreno Rodilla y Belén Curto Diego</i>	
Benchmarking de predictores inteligentes con métodos de aprendizaje profundo	47
<i>Geovanny Satama Bermeo, Pastora Vega, Mario Francisco y Óscar Aponte</i>	
Inferencia dinámica de estados contextuales en sistemas de recomendación sensibles al contexto	81
<i>Henar López Huertas, María N. Moreno-García, Álvaro Lozano Murciego</i>	
Sistema Inteligente para el Etiquetado de Texto	98
<i>Iria Ares Torres, Belén Pérez Lancho y María Navarro Cáceres</i>	
Optimización de la movilidad a partir del manejo de datos de flujo de tráfico	128
<i>Juan Manuel Otero Otero, Vivian Félix López Batista y Juan Francisco de Paz Santana</i>	
Evaluating the Effectiveness of Human-Centered AI Systems in Education	155
<i>Nastaran Shoeibi, Roberto Therón Sánchez y Francisco José García-Peñalvo</i>	
Sistema de auditoría para rastrear, descubrir y reducir el sesgo de género en sistemas de Procesamiento del Lenguaje Natural	186
<i>Samuel Adrados González, Vivian Félix López Batista, Gabriel Villarrubia González</i>	
Sistema de reconocimiento de patrones en imágenes satelitales para la detección de objetos	218
<i>Sergio Fernández Marcos, Vivian Félix López Batista and Gabriel Villarrubia González</i>	
Integración de un robot móvil en el entorno ROS (Robot Operating System). Análisis de la implementación en diferentes versiones	240
<i>Sergio García González, Vidal Moreno Rodilla y Francisco Javier Blanco Rodríguez</i>	

Sistema de recomendación de listas de reproducción de música

Álvaro Gómez Gutiérrez, María N. Moreno García, María Dolores Muñoz
Vicente

Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los
Caídos s/n. 37008, Salamanca, España
{alvarogomez,mmg,mariado}@usal.es

Resumen Este estudio aborda la creciente necesidad de sistemas de recomendación personalizados en las plataformas de *streaming* de música, en respuesta a la preferencia de los usuarios por las listas de reproducción personalizadas en lugar de los álbumes tradicionales. Se realiza un análisis exhaustivo de los sistemas de recomendación existentes, centrándose en la incorporación del comportamiento del usuario, el etiquetado social y la diversidad de canciones para mejorar la personalización y eficacia de las recomendaciones. Se destaca la insuficiencia de las técnicas convencionales de recomendación que se basan principalmente en las valoraciones de los usuarios y se hace hincapié en enfoques que consideran los patrones de comportamiento individuales. Además, se explora la importancia de la diversidad en las listas de reproducción para la satisfacción del usuario y se investigan métricas para cuantificar esta diversidad con el fin de potenciarla en futuros sistemas de recomendación.

Keywords: sistemas de recomendación, sistemas de recomendación de *playlists*, revisión sistemática de la literatura

1. Introducción

Desde las recomendaciones de productos cuando se realiza una compra en Internet hasta sugerencias de películas y música en las plataformas de *streaming*, pasando por los artículos de noticias personalizados y las publicaciones en redes sociales, los sistemas de recomendación se encuentran actualmente en prácticamente todas las aplicaciones que se utilizan diariamente.

Un sistema de recomendación se puede definir como un tipo de sistema de filtrado de información cuyo objetivo es predecir las preferencias o intereses de un usuario y sugerirle artículos (productos, servicios o contenidos) que probablemente le gusten. En definitiva, un sistema de recomendación tiene como objetivo ayudar a los usuarios a elegir sugiriéndoles productos o ítems basados en sus acciones o comportamientos anteriores, así como en las acciones y comportamientos de otros usuarios con preferencias similares.

Los sistemas de recomendación han revolucionado la forma de escuchar y descubrir música. Servicios de música en *streaming* como Spotify o Apple Music

utilizan estos sistemas para analizar el comportamiento de los usuarios y sugerir canciones, álbumes y listas de reproducción (*playlists*) a sus oyentes para, de este modo, mejorar su experiencia de usuario en la plataforma.

En la recomendación de listas de reproducción, identificar la similitud de las canciones en sus estructuras es una tarea fundamental para cualquier servicio o plataforma de *streaming* de música, ya que desempeña un papel importante en la creación de listas de reproducción personalizadas para los usuarios.

El método más sencillo para determinar si dos canciones son similares utiliza metadatos relacionados con las canciones, como el género, el artista, el álbum o el año de publicación. Por ejemplo, dos canciones del mismo artista o del mismo género suelen considerarse similares, o bien, canciones que se publicaron en la misma época pueden compartir aspectos que sugieran similitud [12]. Sin embargo, estos datos por sí solos pueden no ser suficientes para definir la similitud y puede ser necesario un análisis más profundo que implica evaluar las características acústicas de las canciones y otros factores adicionales.

La recomendación musical presenta retos únicos como la brevedad del tiempo de consumo de las canciones, la variabilidad de contextos en que se escucha música, y la falta de un sistema estandarizado para describir la música. El tiempo de consumo breve permite a los usuarios formar opiniones rápidas y decidir repetir o descartar una canción rápidamente. Además, la música se consume en diferentes situaciones, requiriendo que las recomendaciones se adapten a estos contextos. A su vez, los criterios subjetivos para describir música pueden provocar desajustes entre las recomendaciones y las preferencias del usuario. Para superar estos desafíos, se utilizan técnicas basadas en aprendizaje automático y procesamiento de lenguaje natural que analizan el historial de escucha del usuario, las reseñas musicales y la actividad en las redes sociales. Sin embargo, incorporar información contextual de manera efectiva aún es un desafío.

El objetivo principal de esta tesis de máster es desarrollar un sistema de recomendación personalizado de listas de reproducción, el cual tenga en cuenta las preferencias individuales de los usuarios y ofrezca una experiencia musical coherente y agradable. Se propone mantener un equilibrio entre la diversidad musical y la similitud de las canciones, adaptándose al grado de diversidad deseado por el usuario. Los objetivos específicos son: 1) Incrementar la diversidad en las recomendaciones musicales, ya que los sistemas actuales suelen sugerir selecciones muy homogéneas que no reflejan la diversidad de intereses de los usuarios, y 2) Adecuar las recomendaciones a las preferencias individuales, lo que implica que el sistema aprenda de las interacciones de los usuarios con las listas recomendadas y ajuste futuras recomendaciones en función de ello. Con ello, se busca superar las limitaciones de los sistemas de recomendación actuales, mejorando la calidad de las listas de reproducción a través de un enfoque centrado en la diversidad musical y la personalización.

Esta investigación se divide en seis secciones fundamentales. La introducción se centra en la explicación general de los sistemas de recomendación en el contexto de la música, con énfasis en las listas de reproducción. Posteriormente, la segunda sección aborda los fundamentos y la clasificación de los métodos de recomendación, incluyendo una revisión bibliográfica de las técnicas y evaluacio-

nes. En la tercera sección se analiza el estado del arte en la recomendación de listas de reproducción de música. La cuarta sección detalla y valida un sistema de recomendación propuesto que adapta el grado de diversidad en las listas de reproducción. En la quinta sección, se ilustra un caso de estudio que muestra la efectividad del sistema propuesto. La última sección presenta los resultados del estudio, una discusión y las conclusiones correspondientes.

2. Fundamentos y clasificación de los sistemas de recomendación

2.1 Técnicas generales de recomendación

2.1.1 Filtrado colaborativo El filtrado colaborativo es una técnica muy utilizada en sistemas de recomendación que tiene en cuenta las preferencias de otros usuarios para ofrecer recomendaciones a un usuario concreto. Consiste en recomendar ítems a los usuarios basándose en las preferencias de otros que son similares a ellos, teniendo en cuenta la premisa de que si han demostrado intereses similares en el pasado probablemente tendrán preferencias similares en el futuro. Su funcionamiento consiste en recopilar datos sobre las preferencias de los usuarios y utilizarlos después para recomendar ítems a un usuario concreto en función de sus similitudes con otros usuarios.

Existen dos tipos de técnicas de filtrado colaborativo, conocidas como filtrado colaborativo basado en el usuario y basado en ítems. El filtrado colaborativo basado en el usuario recomienda elementos a un usuario basándose en las preferencias de usuarios similares, mientras que el basado en ítems, en lugar de centrarse en las similitudes entre usuarios, este recomienda ítems similares a un usuario basándose en sus interacciones anteriores con ellos.

Por otra parte, el filtrado colaborativo tiene dos enfoques principales: basado en la memoria y basado en modelos. Los sistemas basados en la memoria utilizan los datos de valoración de los usuarios para calcular la similitud entre usuarios o elementos, mientras que los enfoques basados en modelos implican el uso de algoritmos de aprendizaje automático para predecir las valoraciones de los usuarios sobre elementos no valorados. Los sistemas híbridos combinan ambos enfoques para crear un motor de recomendación más completo.

2.1.2 Filtrado basado en contenido El filtrado basado en el contenido es un enfoque de recomendación que se basa en utilizar las características y atributos de los ítems para generar recomendaciones personalizadas a los usuarios. La idea principal consiste en recomendar ítems a un usuario similares a otros ítems con los que ha interactuado o por los que ha mostrado preferencia en el pasado, basándose en la información sobre el contenido de los ítems. Para lograrlo, se crean dos tipos de perfiles que permiten almacenar información por una parte de los ítems, denominados perfiles de ítems y por otra de los usuarios, conocidos como perfiles de usuario.

2.1.3 Sistemas de recomendación híbridos Los sistemas de recomendación híbridos son una clase de sistemas de recomendación que combinan dos o más tipos diferentes de enfoques de recomendación cuyo objetivo es mejorar la precisión y la cobertura del sistema de recomendación aprovechando los puntos fuertes de múltiples técnicas de recomendación.

2.2 Otros tipos de sistemas de recomendación

2.2.1 Sistemas de recomendación sensibles al contexto Los sistemas de recomendación sensibles al contexto mejoran la experiencia del usuario al tener en cuenta el contexto en el que éste interactúa con el sistema, proporcionando recomendaciones más precisas y relevantes [1].

El objetivo principal de un sistema de recomendación sensible al contexto es no sólo recomendar artículos basándose en las acciones o preferencias del usuario, sino también tener en cuenta el contexto en el que está interactuando con el sistema. Un sistema de recomendación para una plataforma de *streaming* de música tendría la capacidad de recomendar música diferente en función del estado de ánimo del usuario o de la franja horaria en que se escucha. Sin embargo, es importante destacar que estas recomendaciones no se basan en asumir directamente qué tipo de música querría escuchar el usuario en un determinado contexto, sino que se derivan del análisis de patrones de escucha de los usuarios en contextos similares. [17].

2.2.2 Sistemas de recomendación de grupos Los sistemas de recomendación de grupos son una extensión de los sistemas de recomendación que tiene en cuenta las preferencias y el comportamiento de varios usuarios para identificar y recomendar artículos que probablemente sean de interés para el grupo en su conjunto, en lugar de sólo para cada usuario individualmente. En general, son especialmente relevantes en contextos en los que los individuos interactúan o consumen artículos de forma colectiva, como el *streaming* de películas o música, la planificación de viajes o las plataformas digitales compartidas.

2.2.3 Sistemas de recomendación *Multi-Stakeholder* Los sistemas de recomendación Multi-Stakeholder son un tipo especial de sistemas de recomendación que tienen en cuenta las necesidades, preferencias e intereses de múltiples partes interesadas como pueden ser usuarios, proveedores de contenidos o anunciantes, y su objetivo es ofrecer recomendaciones personalizadas que sean justas y equilibradas para todas las partes interesadas [13].

2.2.4 Sistemas de recomendación multitarea Los sistemas de recomendación tradicionales suelen adaptarse a dominios o tareas específicas, como recomendar películas, libros o música, sin embargo, los usuarios tienen a menudo intereses diversos que se encuentran relacionados y que se pueden complementar. Es por ello que se utilizan sistemas de recomendación multitarea, que aprovechando una variedad de fuentes de datos, tienen como objetivo ofrecer una

experiencia de usuario mejorada al abordar simultáneamente múltiples tareas del usuario o al hacer recomendaciones a través de diferentes dominios [14]. Por ejemplo, un sistema de recomendación multitarea podría recomendar un estilo de música a un usuario y, a continuación, sugerirle una actividad que combinaría bien con la experiencia de escucha.

2.2.5 Sistemas de recomendación multiobjetivo Un sistema de recomendación multiobjetivo pretende optimizar varios objetivos al mismo tiempo, entre los que se incluyen la relevancia, la diversidad, la novedad y la serendipia [15].

3. Estudio del estado del arte de recomendación de listas de reproducción de música

Los sistemas de recomendación de listas de reproducción se han vuelto esenciales en el mundo de la música digital, al ayudar a los usuarios a navegar por bibliotecas de música extensas y proporcionar selecciones acordes a sus preferencias individuales y contextos específicos. Sin embargo, su diseño e implementación presentan desafíos, como entender y modelar las preferencias de los usuarios, manejar la diversidad y novedad de las recomendaciones, adaptarse a cambios en las preferencias a lo largo del tiempo, y considerar la coherencia y fluidez de las canciones en una lista de reproducción. Los sistemas de recomendación deben ser capaces de manejar la diversidad inherente a las listas de reproducción y proporcionar recomendaciones coherentes y atractivas para el usuario.

3.1 Métodos de recomendación para listas de reproducción de música

Los métodos utilizados para recomendar listas de reproducción de música son diversos y dependen de múltiples factores, como los datos disponibles, la capacidad de procesamiento y los requerimientos específicos de la aplicación. A continuación, se presentan algunas de las categorías más comunes de métodos de recomendación:

Métodos basados en contenido Crean listas de canciones con características similares a las que un usuario ha escuchado o le gustan, usando metadatos de las canciones y considerando la coherencia entre todas las canciones recomendadas. Sin embargo, estos dependen de la calidad y cantidad de los metadatos y pueden tener dificultades para introducir diversidad. [20].

Métodos de filtrado colaborativo Se basan en patrones de comportamiento musical de los usuarios, tanto de otros usuarios con gustos similares como de canciones similares que el usuario ha valorado positivamente. Aunque pueden ofrecer recomendaciones diversas y personalizadas, requieren muchos datos de usuarios y pueden ser menos transparentes [23].

Co-ocurrencia de canciones Analiza la frecuencia con la que canciones aparecen juntas en listas de usuarios para hacer recomendaciones. Esto es útil para descubrir nuevas canciones que pueden encajar con los gustos existentes de los usuarios [3].

Métodos híbridos Combinan características de los métodos basados en contenido y colaborativos, proporcionando recomendaciones más precisas. Sin embargo, son más complejos de implementar y mantener [22].

Sistemas de Recomendación Secuenciales (SR Secuenciales) Tienen en cuenta el orden de las canciones escuchadas, utilizando modelos de aprendizaje automático como las redes neuronales recurrentes (RNN - *Recurrent Neural Network*) o las redes de memoria a largo plazo (LSTM - *Long Short-Term Memory*). Aunque son capaces de proporcionar recomendaciones coherentes, son más complejos y requieren técnicas de modelado más avanzadas. [18][19].

En el caso de las listas de reproducción, una RNN o LSTM toman como entrada la secuencia de canciones que un usuario ha escuchado y como salida dar una probabilidad a cada canción disponible de ser la siguiente en la lista.

3.2 Métricas de evaluación para los sistemas de recomendación de listas de reproducción de música

La evaluación del rendimiento de los sistemas de recomendación es una tarea crucial que permite comparar diferentes algoritmos, identificar fortalezas y debilidades, y tomar decisiones informadas para la selección o desarrollo de algoritmos de recomendación [16]. A continuación, se muestran las métricas más utilizadas para evaluar las recomendaciones en listas de reproducción.

- **Precision:** Refiriéndose a qué porcentaje de las canciones recomendadas fueron realmente relevantes para el usuario, en el caso de una lista de reproducción, esto podría traducirse en cuántas de las canciones sugeridas por el algoritmo el usuario realmente disfrutó y decidió guardar o reproducir repetidamente.
- **Recall:** La recuperación mide la proporción de recomendaciones correctas en relación con todas las canciones que al usuario le hubiera gustado recibir como recomendación.
- **F-measure:** Representa un equilibrio entre precisión y recall.
- **Mean Average Precision (MAP):** Considera la precisión promedio en cada punto de una lista donde aparece una canción relevante, dándole más peso a los aciertos en las primeras posiciones.
- **Normalized Discounted Cumulative Gain (nDCG):** Al igual que MAP, nDCG tiene en cuenta el orden de las recomendaciones, dando más peso a las recomendaciones que aparecen más arriba en la lista. Sin embargo, nDCG también puede considerar diferentes grados de relevancia.
- **Mean Reciprocal Rank (MRR):** Útil para valorar la importancia de la primera canción recomendada. Para las listas de reproducción de música, esta métrica podría ser útil si se está particularmente interesado en acertar

con la primera canción recomendada, por ejemplo, para enganchar al usuario desde el principio.

- **Diversidad:** La diversidad en las listas de reproducción de música mide hasta qué punto las recomendaciones de canciones son diferentes entre sí [2]. Se han propuesto diversas métricas para medir la diversidad en este ámbito:
 - **Entropía de Shannon:** Esta métrica mide la incertidumbre o el desorden dentro de un conjunto de datos. En el contexto de las recomendaciones de música, una alta entropía de Shannon indica una mayor diversidad en las recomendaciones de música, ya que las canciones recomendadas están dispersas a lo largo de diferentes categorías o géneros.

$$H(X) = - \sum [p(x) \log_2 p(x)] \quad (1)$$

donde:

- X es el conjunto de posibles canciones recomendadas
- $p(x)$ es la probabilidad de que una canción sea recomendada
- **Diversidad dentro de la lista (ILD):** Calcula la disimilitud media entre cada par de canciones de una lista de reproducción [6].
- **Diversidad dentro de la lista esperada (EILD):** Variante de ILD que tiene en cuenta la probabilidad de que se escuche cada canción, pues se basa en el hecho de que los usuarios pueden no escuchar todas las canciones de una lista de reproducción, especialmente si la lista es larga [9].
- **Gini Index:** Mide la desigualdad en la distribución de las recomendaciones de elementos entre diferentes categorías, como géneros o artistas [10].
- **Coverage (Cobertura):** La cobertura en las listas de reproducción de música se refiere a la proporción de canciones o usuarios para los que el sistema puede hacer recomendaciones.

3.3 Diversidad como factor importante en la recomendación de listas de reproducción

Es importante destacar que la diversidad a la hora de generar de listas de reproducción no consiste sólo en la variedad por sí misma, pues es un método para abordar la naturaleza compleja y multidimensional de las preferencias musicales. La idea es que los oyentes pueden apreciar ciertos géneros o artistas la mayor parte del tiempo, pero también pueden apreciar la novedad o la sorpresa de una pista inesperada.

Además, el gusto musical no es estático, sino que evoluciona con el tiempo, y los oyentes suelen tratar de descubrir música nueva. Por lo tanto, una buena lista de reproducción no solo se ajusta a los gustos actuales del usuario, sino que también incorpora un cierto nivel de diversidad para adaptarse a sus preferencias cambiantes y a su deseo de novedad.

Varios estudios destacan la importancia de la diversidad en las listas de reproducción, equilibrando elementos familiares y novedosos. *G. Bonnin y D. Jannach*

[3] indican que las listas contienen canciones similares pero con cierta diversidad, ya que los oyentes incluyen tanto canciones conocidas como nuevas. *A. Gatzoura et al.*[22] proponen un sistema híbrido que genera listas manteniendo coherencia y diversidad. *T. Zhou et al.* [7] resaltan la importancia de la diversidad, novedad y sorpresa para la satisfacción del usuario. *Pichl et al.* [8] muestran que el contexto de escucha influye en las preferencias de diversidad. *V. Kumar* [11] examina la necesidad de equilibrio entre novedad y familiaridad, subrayando la importancia de estrategias de recomendación personalizadas.

3.4 Tendencias actuales y futuras en los sistemas de recomendación de listas de reproducción

Los sistemas de recomendación de listas de reproducción han experimentado cambios notables en los últimos años, y seguirán evolucionando para mejorar la precisión de sus predicciones y proporcionar experiencias de escucha personalizadas.

- *Personalización profunda*: La personalización ha sido un tema clave en la música durante mucho tiempo y continúa siendo una tendencia actual. Los servicios de streaming buscan proporcionar recomendaciones musicales adaptadas a las preferencias individuales de cada usuario [5].
- *Integración de contextos*: La inclusión del contexto, como la hora del día, la ubicación o el estado de ánimo, en los sistemas de recomendación de música es otra tendencia actual [3].
- *Aprendizaje federado*: El aprendizaje federado es un nuevo enfoque que permite entrenar modelos de aprendizaje automático en dispositivos de usuarios finales [21], lo que puede mejorar la personalización y la privacidad.
- *Recomendaciones basadas en texto*: A medida que se vuelven más sofisticados, los sistemas de recomendación de música también pueden comenzar a utilizar análisis de texto para hacer recomendaciones basadas en las letras de las canciones o las reseñas de álbumes y artistas.

3.5 Problemas abiertos

A pesar de estas innovaciones, siguen existiendo desafíos en el campo de la recomendación de listas de reproducción de música. Algunos de los problemas abiertos en esta área incluyen:

- *Long tail problem*: Se refiere a la dificultad de recomendar elementos menos populares o desconocidos que podrían ser de interés para el usuario [4].
- *Sesgo algorítmico*: Los sistemas de recomendación pueden reproducir y amplificar los sesgos presentes en los datos de entrenamiento.
- *Evaluación de sistemas de recomendación*: Actualmente, evaluar la efectividad de un sistema de recomendación es un desafío, ya que depende en gran medida de las métricas utilizadas y de la subjetividad inherente a la música.

4. Sistema propuesto

El sistema propuesto es un sistema de recomendación de música personalizado que se ha desarrollado con el objetivo de generar listas de reproducción para los usuarios que abarcan un grado de diversidad ajustado a sus preferencias. Este objetivo se logra a través de la combinación de técnicas de filtrado basado en contenido y aprendizaje por refuerzo.

4.1 Metodología de recomendación

El sistema de recomendación propuesto opera en dos etapas clave, diseñadas para ser aplicables a cualquier conjunto de datos que cumplan ciertas condiciones. En particular, el método utiliza tanto datos de interacción de los usuarios como metadatos y características de las canciones.

En la primera etapa, se recolectan y combinan datos que contienen información sobre las interacciones de los usuarios con diferentes canciones, y datos que comprenden un conjunto de metadatos y características de audio de las canciones.

Estos dos tipos de datos se combinan para obtener una representación rica y detallada del comportamiento del usuario y las características de las canciones. El objetivo de esta etapa es comprender tanto las preferencias individuales del usuario como la diversidad de la música a la que ha sido expuesto.

En la segunda etapa del sistema de recomendación musical propuesto, la diversidad de las preferencias musicales del usuario se calcula basándose en las características de audio de las canciones en las listas de reproducción del usuario. Este cálculo se realiza mediante la entropía de Shannon.

La elección de la entropía de Shannon se justifica por varias razones. En este contexto, se utiliza para cuantificar la diversidad en las preferencias musicales del usuario en términos de las características de audio de las canciones. Cuanto mayor sea la entropía, mayor será la diversidad en las preferencias musicales del usuario.

Otro punto a favor de la entropía de Shannon es que proporciona una medida de diversidad que es independiente del tamaño del conjunto de datos, lo que facilita la comparación de la diversidad entre diferentes listas de reproducción del usuario.

Una vez calculada la diversidad, el sistema calcula las similitudes entre las canciones basándose en sus características de contenido, a través de un enfoque basado en contenido. Este enfoque fue seleccionado por varias razones. En primer lugar, ofrece una mayor personalización que los métodos de filtrado colaborativo, ya que genera recomendaciones basadas en los intereses y comportamientos individuales del usuario. En segundo lugar, puede superar ciertas limitaciones del filtrado colaborativo, como el problema del *cold start*, en el que los nuevos usuarios o las nuevas canciones pueden no tener suficiente información de interacción para generar recomendaciones precisas. Además, proporciona una mayor transparencia y explicabilidad ya que las recomendaciones se basan en las características del contenido que ha interesado al usuario en el pasado.

Posteriormente, la diversidad calculada previamente se aplica a los resultados de las similitudes obtenidas mediante el sistema basado en contenido (Figura 1). Así, el sistema genera una lista de reproducción recomendada que refleja la diversidad de las preferencias musicales del usuario.

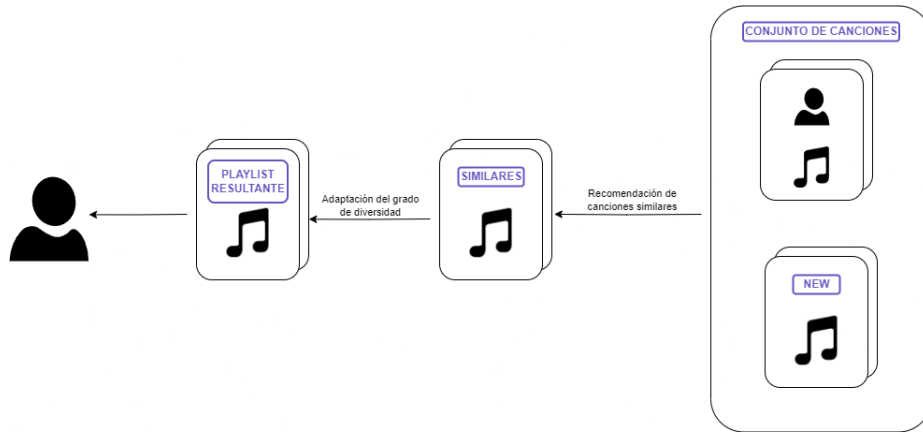


Figura 1: Modelo del sistema de recomendación

Tras la generación inicial de la lista de reproducción, se introduce un componente de aprendizaje por refuerzo para refinar aún más las recomendaciones. Este componente se basa en las interacciones simuladas del usuario con las canciones recomendadas, que se reflejan en el sistema como “me gusta” o “no me gusta”, y se utiliza para ajustar las recomendaciones futuras. De esta forma, el sistema puede adaptarse de manera más rápida y precisa a las preferencias cambiantes del usuario, lo que permite una mayor adaptabilidad y personalización.

4.1.1 Metodología de recomendación por aprendizaje por refuerzo El aprendizaje por refuerzo tiene un papel importante en el diseño de este sistema de recomendación, permitiendo mejorar continuamente las sugerencias en base a las interacciones de los usuarios (Figura 2). Este modelo de aprendizaje se adapta a las preferencias de los usuarios a medida que interactúan con las recomendaciones.

Una de las principales estrategias en el sistema es recomendar canciones individualmente, utilizando la entropía de Shannon para diversificar la lista de reproducción. Cada recomendación se realiza teniendo en cuenta las interacciones previas del usuario con las canciones sugeridas, considerando sus preferencias.

El registro de las interacciones con las recomendaciones permite al sistema actualizar su conocimiento sobre las preferencias del usuario. Así, con cada interacción, el sistema aprende y se adapta, ajustando futuras recomendaciones según las preferencias individuales de cada usuario.

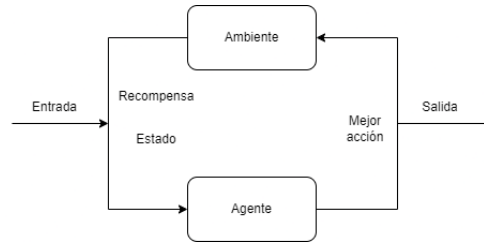


Figura 2: Aprendizaje por refuerzo

Existen varios algoritmos de aprendizaje por refuerzo, como PPO (*Proximal Policy Optimization*), A2C (*Advantage Actor-Critic*), DDPG (*Deep Deterministic Policy Gradient*), DQN (*Deep Q-Network*), SAC (*Soft Actor-Critic*), TD3 (*Twin Delayed DDPG*) y HER (*Hindsight Experience Replay*). Cada algoritmo tiene sus propias fortalezas y debilidades, y su selección depende de factores como la naturaleza del problema, la cantidad de muestras disponibles, la complejidad de implementación y los requisitos de rendimiento.

Dada la amplia variedad de canciones y la diversidad de la música, se ha seleccionado el algoritmo PPO para este sistema de recomendación. PPO es especialmente eficiente en entornos con amplios espacios de acciones y puede manejar situaciones con recompensas escasas, lo que es especialmente adecuado para este sistema, donde se otorga una recompensa solo cuando una canción diversa es bien recibida por el usuario.

En el aprendizaje por refuerzo, las políticas son estrategias que guían cómo un agente toma decisiones. El algoritmo PPO implementa sus políticas usando diversos tipos de redes neuronales, las cuales se eligen basándose en la naturaleza de las observaciones. Para este sistema, que utiliza vectores numéricos como observaciones, se ha seleccionado la *MlpPolicy*.

La actualización continua del modelo de aprendizaje por refuerzo permite al sistema adaptarse a las preferencias cambiantes de los usuarios. Si un usuario muestra una predilección por un tipo de características musicales en particular, el sistema aprenderá a recomendar más canciones de esas características. Por otro lado, si un usuario rechaza o apenas interactúa ciertas canciones, el sistema aprenderá a evitar sugerir canciones similares en el futuro. En resumen, el aprendizaje por refuerzo permite una personalización de las recomendaciones que mejora con cada interacción, adaptándose a las preferencias individuales de cada usuario.

4.2 Arquitectura del sistema

El sistema de recomendación propuesto se implementa utilizando Python, un lenguaje de programación versátil y ampliamente utilizado en el ámbito del análisis de datos y aprendizaje automático. Se ha seleccionado Python por su fácil legibilidad, flexibilidad y las numerosas bibliotecas que proporciona para la ciencia de datos y el aprendizaje automático.

En particular, se utilizan las bibliotecas Pandas y NumPy para la manipulación de los datos y cálculos numéricos, respectivamente. Para implementar el aprendizaje por refuerzo, se hace uso de una combinación de técnicas de inteligencia artificial que también se implementan con la ayuda de las bibliotecas de Python.

La implementación del sistema consta de varios componentes y se organiza en una arquitectura de microservicios. Cada componente del sistema se encapsula en su propio servicio para facilitar la reutilización, la escalabilidad y el mantenimiento.

Este sistema se integra de manera fluida con Spotify para generar y administrar listas de reproducción personalizadas para los usuarios.

4.2.1 Integración con Spotify Esta integración se realiza mediante el uso de la API de Spotify (Figura (3)), una interfaz de programación que proporciona al sistema la capacidad de interactuar con los servicios de Spotify de forma programática.

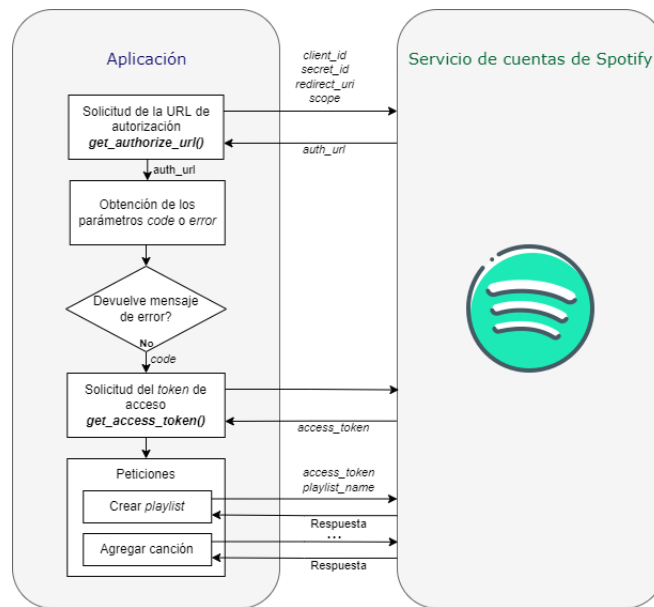


Figura 3: Funcionamiento de integración con la API de Spotify

Para acceder a los endpoints de la API de Spotify que requieren autorización, es necesario, en primer lugar, obtener un token de autenticación, también conocido como token de acceso.

El token de autenticación es un tipo de credencial que se utiliza para autenticar una aplicación frente a la API de Spotify. Este token de acceso, que es una

cadena de caracteres alfanuméricos, se debe incluir en cada solicitud HTTP que se envía a la API de Spotify.

Una vez realizado el flujo de autenticación, para generar una lista de reproducción en Spotify, el sistema utiliza las funcionalidades de la API de Spotify para crear una nueva lista de reproducción bajo la cuenta del usuario a través del endpoint `user_playlist_create(username, playlist_name)`. Después de crear la lista de reproducción, el sistema comienza a seleccionar las canciones para incluir en esta lista.

Además de la creación de listas de reproducción, el sistema también utiliza la API de Spotify para continuar recomendando canciones basadas en las interacciones de los usuarios. Al utilizar el aprendizaje por refuerzo, el sistema puede ajustar sus recomendaciones basándose en las canciones que el usuario ha marcado como “*liked*”. En cada iteración de este proceso de recomendación, el sistema genera una nueva recomendación, verifica si el usuario ha indicado que le gusta la canción, y si es así, la añade a la lista de reproducción en Spotify.

Esta integración entre el sistema y Spotify permite a los usuarios beneficiarse de las recomendaciones personalizadas y diversas del sistema directamente en esta plataforma de streaming de música.

Cabe mencionar que la integración con la API de Spotify se realiza a través del paquete `Spotipy`, que proporciona una interfaz Python para la API de Spotify Web. Este paquete permite al sistema interactuar con Spotify para recuperar listas de reproducción y añadir canciones a ellas.

4.2.2 Servicio web Para exponer el sistema como un servicio web, se utiliza FastAPI, un marco de trabajo web moderno y rápido para construir APIs con Python basado en tipos de anotación estándar de Python y `async` y `await` de Python.

En términos de ejecución, el sistema se puede iniciar mediante una serie de comandos de Python que inician el servidor FastAPI y todos los microservicios asociados. Para usar el sistema, los usuarios pueden enviar solicitudes HTTP a los endpoints de la API expuestos por FastAPI.

El código está organizado de manera modular, con funciones y clases claramente definidas para cada tarea específica. Además, el código está bien comentado y documentado para facilitar su mantenimiento y extensibilidad.

A continuación se muestra el diagrama de clases para el sistema (Figura (4)):

- **SpotifyAPI:** Se encarga de la interacción con la API de Spotify. Con credenciales de autenticación, esta clase recupera detalles de canciones, maneja listas de reproducción y gestiona las canciones del usuario.
- **PlaylistRecommender:** Es la clase principal que coordina todas las actividades de recomendación. Utiliza la clase `SpotifyAPI` para obtener datos de las canciones y un modelo de aprendizaje por refuerzo para generar recomendaciones personalizadas.
- **PlaylistRecommenderEnv:** Esta clase representa el entorno de aprendizaje por refuerzo personalizado, heredando de la clase `gym.Env` del paquete

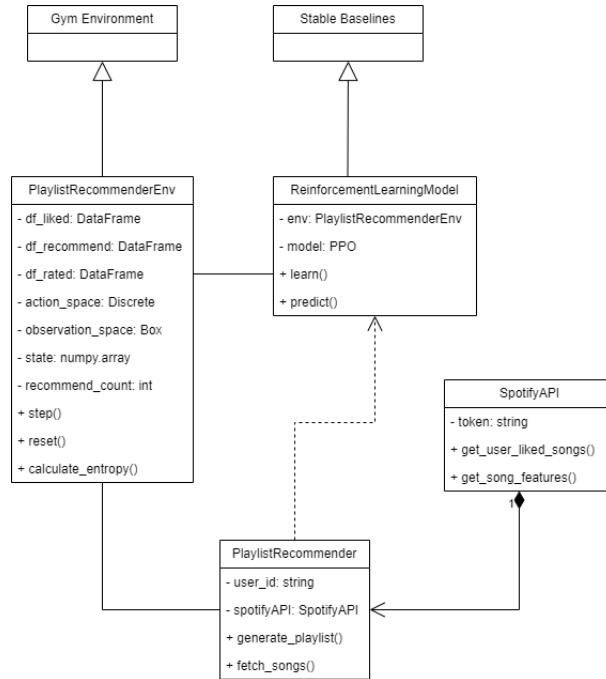


Figura 4: Diagrama de clases

Gym. Proporciona información sobre el estado del entorno y recompensa o penaliza al agente basado en sus acciones.

- ReinforcementLearningModel:** Se encarga de la implementación del modelo de aprendizaje por refuerzo, utilizando la librería *Stable Baselines* y el algoritmo *PPO*. Predice la próxima acción del agente y actualiza el modelo basado en las recompensas recibidas y el estado posterior a la acción.

En conclusión, la implementación del sistema de recomendación combina técnicas modernas de análisis de datos y aprendizaje automático en una arquitectura modular y escalable que se puede adaptar fácilmente para abordar una amplia gama de tareas de recomendación.

5. Caso de estudio

5.1 Descripción y preprocesamiento de datos

La etapa de preprocesamiento de datos es fundamental para el correcto funcionamiento del sistema de recomendación, ya que los datos procesados forman la base para generar recomendaciones personalizadas. Este proceso se realiza en varios pasos, entre los cuales se incluye: recolección de datos, limpieza y etiquetado de datos, extracción de características y fusión de conjuntos de datos.

Para la recolección de datos, se utilizan dos datasets obtenidos de Kaggle ¹. El primer dataset, “Spotify Playlists”, proporciona información sobre las interacciones de los usuarios con diversas canciones en Spotify ², representadas por las listas de reproducción que crean. Las columnas de este conjunto de datos incluyen el `'user_id'`, `'trackname'`, `'playlistname'` y `'artistname'`. El segundo dataset, “Spotify Dataset”, es un conjunto de características de audio y metadatos de canciones disponibles en Spotify, incluyendo la `'valence'`, `'year'`, `'acousticness'`, `'artists'`, `'danceability'`, `'duration_ms'`, `'energy'`, `'explicit'`, `'id'`, `'instrumentalness'`, `'key'`, `'liveness'`, `'loudness'`, `'mode'`, `'name'`, `'popularity'`, `'release_date'`, `'speechiness'`, y `'tempo'`.

Al realizar un análisis sobre el conjunto de datos *Spotify Playlists*, se puede ver la tendencia del número de canciones diferentes escuchadas por los usuarios (Figura (5)). Como se puede observar, existen unos pocos usuarios que escuchan una gran cantidad de canciones y bastantes que escuchan apenas 10 pistas, por lo que si se visualiza un rango específico se puede ver que la media está entre 100 y 999 canciones escuchadas (Figura (7)).



Figura 5: Todos los usuarios



Figura 6: Agrupación entre la media de usuarios

Por lo tanto, para limpiar los datos y garantizar su coherencia, en primer lugar, se seleccionan los registros del primer conjunto de datos que contienen el `'user_id'` específico de un usuario objetivo dentro del rango establecido para tener suficientes datos para poder realizar las recomendaciones.

Luego, utilizando la API de Spotify, en primer lugar se buscan las canciones a través de una *query* en base al nombre y artista de la pista musical. Si no existe o no se encuentra en Spotify, muestra un mensaje de error y continúa con la siguiente canción, mientras que si la encuentra, obtiene su ID de Spotify y lo agrega a una lista de “ids” que a continuación se utilizarán para obtener las características de audio (*audio_features*) para todas las canciones en las listas de

¹ Kaggle: <https://www.kaggle.com/>

² Spotify: <https://developer.spotify.com/>

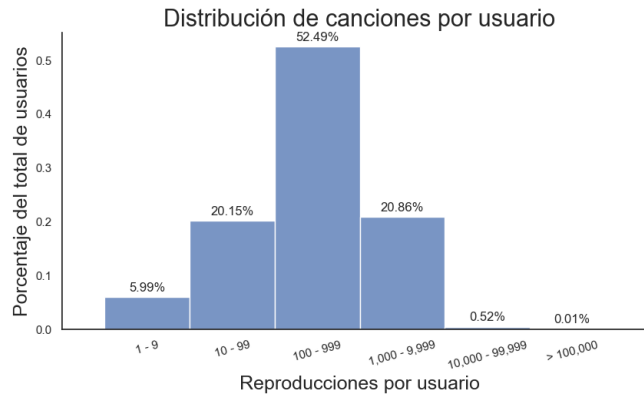


Figura 7: Número de canciones por usuario

reproducción de este usuario mediante otra consulta a la API. El resultado es un subconjunto de datos que contiene información detallada sobre las preferencias musicales del usuario.

Posteriormente, se procede a la etapa de etiquetado de datos. En este paso, una vez se han agregado las características musicales a cada canción del conjunto de datos que representa las canciones del usuario, se genera una columna “*count*” que indica la frecuencia relativa con la que una canción específica aparece en las diferentes listas de reproducción del mismo usuario. Esta frecuencia relativa se calcula como la proporción entre el número de veces que un usuario escucha una canción en particular y el número total de veces que el usuario ha escuchado cualquier canción. Esto proporciona una medida de cuánto le gusta al usuario esa canción en particular en relación con todas las demás canciones que ha escuchado.

Por otra parte, en el segundo conjunto de datos (“*Spotify dataset*”), se genera la columna “*count*” con todos los registros con el valor 0. Esta asignación se debe a que estas canciones, que forman parte de ese conjunto de datos, no han sido escuchadas por el usuario en estudio aún. Por lo tanto, se considera que el usuario no ha interactuado con ninguna de estas canciones hasta ahora.

5.2 Modelo de recomendación

Como se ha mencionado en la propuesta, el modelo de recomendación empleado en este sistema tiene como objetivo fundamental proporcionar a los usuarios listas de reproducción personalizadas basadas en sus preferencias musicales y patrones de comportamiento. Para lograr esto, el sistema se apoya en el cálculo de similitudes entre canciones y la aplicación de la entropía de Shannon para estimar la diversidad de la música.

El cálculo de la similitud entre las canciones es un componente esencial del sistema de recomendación. Para ello se hace uso de la similitud del coseno (Ecuación (2)), medida que determina la similitud entre dos vectores al calcular el coseno

del ángulo entre ellos. En este caso, los vectores corresponden a las características de audio de las canciones, donde se incluyen las siguientes: *'acousticness'*, *'danceability'*, *'energy'*, *'valence'*, *'speechiness'*, *'key'*, *'loudness'*, *'tempo'*, *'instrumentalness'*, *'liveness'* y *'mode'*.

$$s(x, y) = \frac{\sum_{i \in I_{xy}} r_{x,i} r_{y,i}}{\sqrt[2]{\sum_{i \in I_{xy}} r_{x,i}^2} \sqrt[2]{\sum_{i \in I_{xy}} r_{y,i}^2}} \quad (2)$$

Antes de proceder al cálculo de la similitud, si bien la mayoría de esos atributos presentan una medida entre 0 y 1, es necesario tener en cuenta que *'key'*, *'loudness'* y *'tempo'* deben llevar a cabo una normalización para poder trabajar con ellos en la misma escala.

Posteriormente, al calcular la similitud del coseno para todas las parejas posibles de canciones, se obtiene una matriz de similitud que refleja lo similares que son las canciones entre sí según sus características de audio.

Al calcular la entropía de Shannon para las canciones en las listas de reproducción del usuario basándose en las características de audio mencionadas, se obtiene una medida cuantitativa de la diversidad de las preferencias musicales del usuario. Esto permite personalizar el grado de diversidad de las listas de reproducción recomendadas para cada usuario. En otras palabras, si un usuario muestra una alta diversidad en sus preferencias musicales (es decir, una alta entropía de Shannon), el sistema generará listas de reproducción con una diversidad de canciones similar (Figura (8)).

Por lo tanto, este enfoque combinado de cálculo de similitud y entropía de Shannon permite al sistema proporcionar listas de reproducción personalizadas que no sólo se ajustan a las preferencias musicales actuales del usuario, sino que también tienen en cuenta la diversidad de estas preferencias.

5.3 Modelo de aprendizaje por refuerzo

Para implementar este proceso (Figura (9)), se crea un entorno que representará el conjunto de canciones disponibles. El agente, que representa el sistema de recomendación en sí, aprende a tomar decisiones (recomendar canciones) basándose en la retroalimentación que recibe del entorno. Y la recompensa que en este caso se representa por la interacción del usuario a la recomendación de una canción.

Para realizar la implementación, se simula interacción del usuario añadiendo al conjunto de datos de la lista de reproducción una columna llamada *"liked"*, que es 0 si la canción no ha gustado al usuario y 1 si le ha gustado. Con el objetivo de agilizar el proceso, se generan estos valores aleatoriamente para simular las interacciones del usuario con las canciones recomendadas, aunque en un caso real, es posible crear un método para recuperar esta información de la propia plataforma de Spotify.

Para llevar a cabo este proceso, se crea un entorno de aprendizaje por refuerzo personalizado *PlaylistRecommenderEnv* que proporciona al agente (sistema de

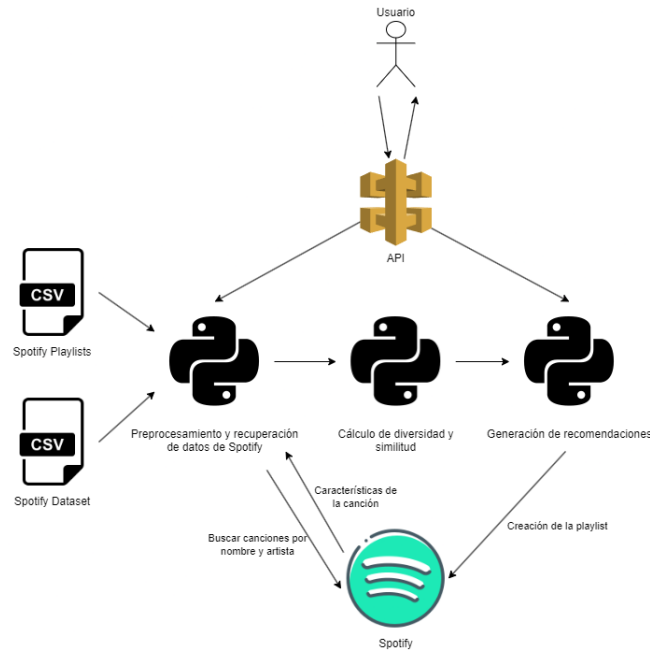


Figura 8: Diseño del sistema de recomendación

recomendación) la capacidad de interactuar con el entorno (las preferencias del usuario).

El entorno es básicamente una clase personalizada que hereda de `gym.Env`, la clase de entorno base proporcionada por *OpenAI Gym*, una biblioteca popular para el desarrollo de algoritmos de aprendizaje por refuerzo. A continuación se explica en detalle cada uno de los métodos de esta clase.

El método `__init__` se utiliza para inicializar el entorno. Este método toma tres conjuntos de datos: `df_liked`, `df_recommend` y `df_rated`. El dataframe `df_liked` contiene las canciones que le gustan al usuario, `df_recommend` contiene las canciones que se pueden recomendar y `df_rated` contiene las canciones de la playlist recomendada en el proceso anterior. Además, este método inicializa el rango de recompensas posibles a (0,1), define el espacio de acción y observación, y establece el estado inicial del entorno.

El método `step` se utiliza para realizar una acción en el entorno, que en este caso es recomendar una canción. Este método toma como entrada la acción que el agente decide tomar, que es el índice de una canción en el dataframe `df_recommend`. Después de ejecutar la acción, el método `step` calcula la recompensa basada en si la canción fue apreciada por el usuario y la entropía de las características de la canción en comparación con las canciones que el usuario ya le gusta. Finalmente devuelve el nuevo estado del entorno, la recompensa obtenida y una señal de si el episodio ha terminado.

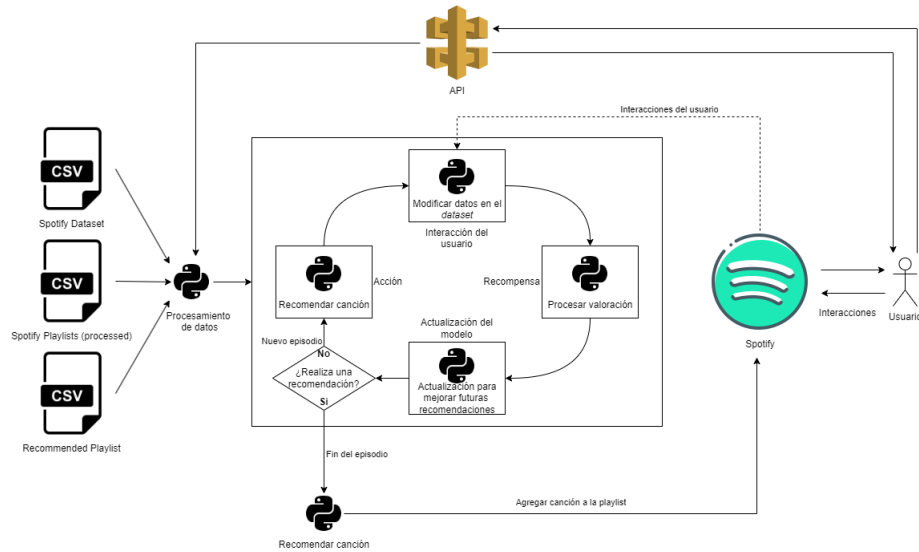


Figura 9: Diseño del sistema de aprendizaje por refuerzo

El método `reset` se utiliza para restablecer el entorno al inicio de un nuevo episodio. Selecciona una nueva canción que le gusta al usuario y la establece como el estado inicial del entorno para el próximo episodio.

El método `calculate_entropy`, muy similar al utilizado anteriormente, se utiliza para calcular la entropía de las características de una canción en relación con las canciones que le gustan al usuario.

Posteriormente, para entrenar el modelo de aprendizaje por refuerzo se utiliza la biblioteca *Stable Baselines3* que proporciona implementaciones de alta calidad y fácil de usar para algoritmos de aprendizaje por refuerzo. La biblioteca incluye implementaciones de una variedad de algoritmos de aprendizaje por refuerzo modernos, incluyendo *PPO (Proximal Policy Optimization)* que es algoritmo elegido para este propósito.

Para llevar a cabo el entrenamiento del modelo de aprendizaje por refuerzo, se utiliza el entorno creado previamente `PlaylistRecommenderEnv`. A continuación, tras comprobar que todo está correctamente configurado mediante `check_env()`, se inicializa el modelo *PPO* con una política basada en perceptrones multicapa `MlpPolicy`.

A continuación se entrena el modelo. Durante el entrenamiento, el modelo interactúa con el entorno, selecciona acciones basadas en su política actual, observa las recompensas y las siguientes observaciones, y luego actualiza su política.

Una vez finalizado el entrenamiento, se restablece el entorno a su estado inicial y se obtiene la primera observación utilizando `env.reset()`. Por último, se utiliza el modelo para predecir la acción que debe tomar, en este caso, elegir una canción para recomendar.

5.4 Resumen de los resultados

Los resultados obtenidos por el sistema de recomendación se resumen en este apartado.

En primer lugar, se aplicarán las métricas tradicionales al sistema de recomendación sin ajustar el grado de diversidad. El propósito de esta etapa es verificar que el sistema sea capaz de recomendar canciones similares con precisión, lo que es una característica fundamental de cualquier sistema basado en contenido. Al no introducir ningún ajuste de diversidad en este punto, se puede aislar y observar la eficacia del sistema en la recomendación de canciones similares basándose únicamente en sus características.

Las métricas de rendimiento aplicadas fueron:

```
Precision@20: 0.80
Recall@20: 0.80
F-beta@20: 0.82
Mean Average Precision: 0.80
Mean Reciprocal Rank: 1.00
nDCG@20: 0.69
```

- **Precision@20 (0.80)**: Esta métrica indica que el 80 % de las primeras 20 recomendaciones son relevantes para el usuario, demostrando la utilidad de las recomendaciones proporcionadas.
- **Recall@20 (0.80)**: Revela que el 80 % de las canciones que le gustan a un usuario están incluidas en las primeras 20 recomendaciones, mostrando la eficacia del sistema para capturar las preferencias del usuario.
- **F-beta@20 (0.82)**: Combina la precisión y el recall en una sola cifra. El valor de 0.82 indica un buen equilibrio entre ambas, asumiendo que la ponderación entre precisión y recall es igual ($\beta=1$).
- **Mean Average Precision (0.80)**: Esta métrica muestra que, en promedio, el 80 % de todas las recomendaciones son relevantes para el usuario.
- **Mean Reciprocal Rank (1.00)**: Demuestra que el sistema siempre coloca la canción más relevante en la parte superior de las recomendaciones.
- **nDCG@20 (0.69)**: Indica que el sistema es eficaz al proporcionar recomendaciones relevantes y colocarlas en las posiciones más altas, aunque existe margen de mejora, ya que el valor óptimo sería 1.0.

En general, estos resultados sugieren que el sistema de recomendación está realizando un buen trabajo al recomendar ítems que son relevantes para el usuario, y también al ordenar estas recomendaciones de manera eficaz.

A continuación, se utilizará la métrica de diversidad de la entropía de Shannon para evaluar el grado de variación dentro de las listas de reproducción recomendadas después de haber adaptado el grado de diversidad a las recomendaciones del sistema.:

```
Diversidad de las recomendaciones: 3.94
```


- **Diversidad de las recomendaciones (3.94):** Muestra un alto grado de diversidad en las sugerencias realizadas por el sistema. Esta medida, basada en la entropía de Shannon, indica una amplia variedad en las recomendaciones, potenciando el descubrimiento de nueva música y manteniendo el interés de los usuarios en las propuestas del sistema.

Para observar más detalladamente la diversidad en las canciones recomendadas, se ha realizado un análisis comparativo de las distribuciones de los atributos en las canciones que había escuchado previamente el usuario (Figura (10)), con respecto a las que se han recomendado (11)). Para poder visualizar los datos con detalle, se amplian las escalas para algunas características.

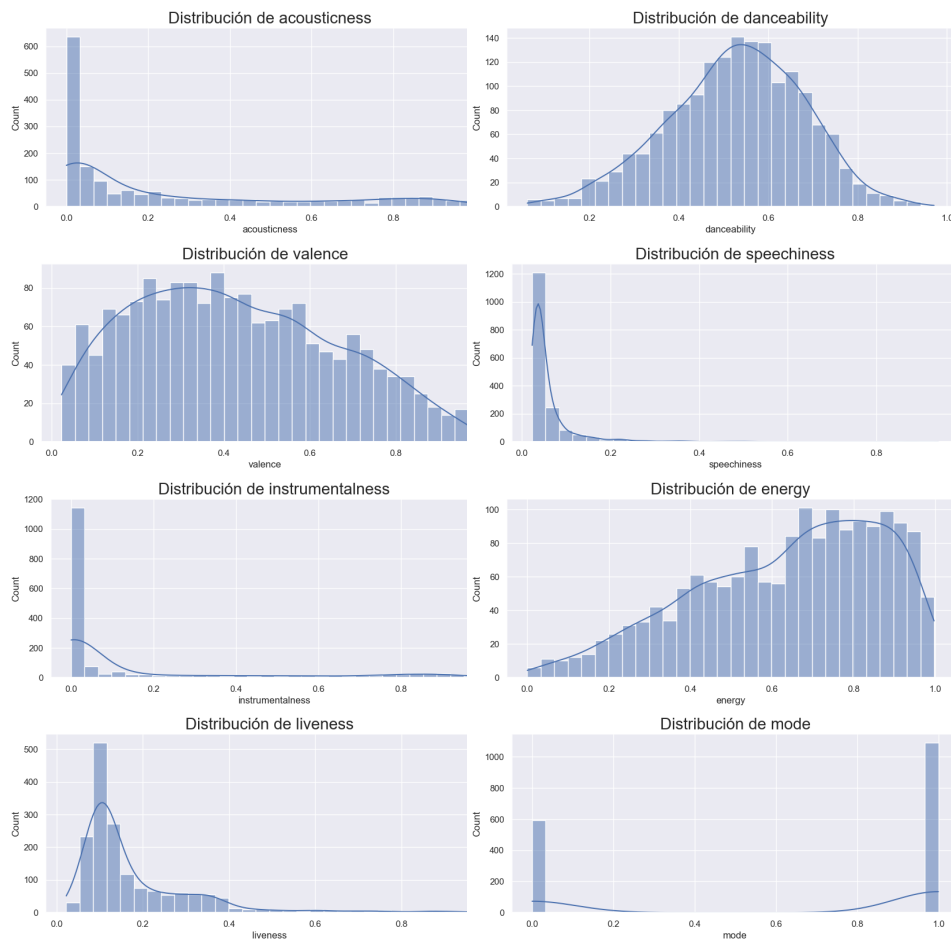


Figura 10: Distribuciones de las características de las canciones con las que el usuario había interactuado

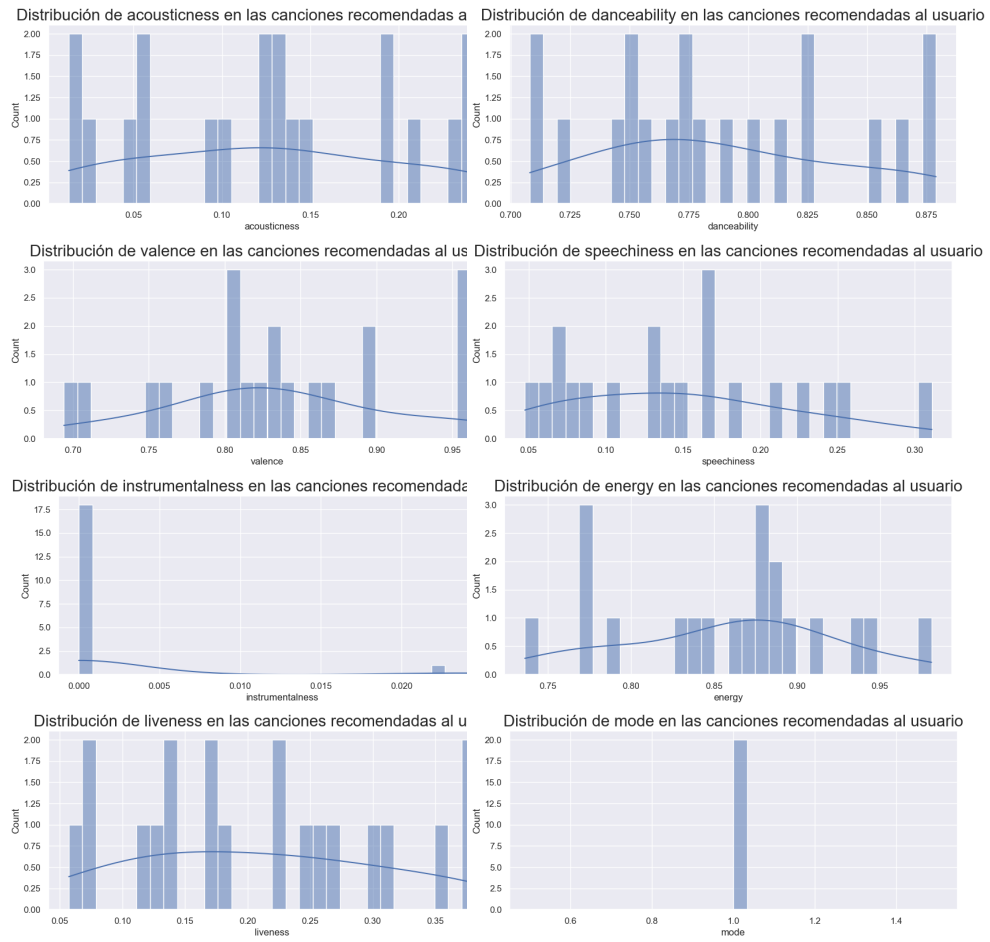


Figura 11: Distribuciones de las características de las canciones recomendadas al usuario

A través de este análisis, se puede obtener una comprensión más profunda de cómo las recomendaciones se alinean con las preferencias existentes de los usuarios y cuánta diversidad y novedad aportan.

Los atributos de las canciones recomendadas no se alejan excesivamente de los atributos de las canciones que los usuarios han escuchado previamente, esto indicaría que el sistema de recomendación se está adaptando a las preferencias individuales de los usuarios aportando el grado de diversidad correspondiente.

6. Conclusiones

Este Trabajo de Fin de Máster se enfocó en personalizar y diversificar las recomendaciones musicales, abordando estos desafíos mediante un sistema innovador basado en aprendizaje por refuerzo y la entropía de Shannon. La eficacia del sistema se demostró a través de un caso de estudio, destacando su capacidad de adaptarse a las preferencias individuales del usuario, mantener la diversidad y enriquecer la experiencia del usuario. Con una precisión y recall de 0.8, y una puntuación de nDCG de 0.69, el sistema demostró su habilidad para sugerir canciones relevantes y colocar las más relevantes en posiciones superiores. La intervención para incrementar la diversidad resultó en una entropía de Shannon de 3.94, indicando una buena diversidad en las recomendaciones, y la comparación de las distribuciones de los atributos de las canciones validó el equilibrio del sistema entre adaptación a las preferencias de los usuarios y oferta de nuevas experiencias musicales.

6.1 Limitaciones del estudio

Aunque este estudio representa un avance significativo y ofrece perspectivas prometedoras, es importante destacar algunas limitaciones inherentes a la investigación realizada. La primera limitación es la eficiencia en términos de tiempo del sistema de aprendizaje por refuerzo. Si bien el sistema se adapta y aprende de las interacciones del usuario, este proceso puede requerir un tiempo considerable, lo que podría resultar en recomendaciones más lentas.

Otra limitación es la falta de interacción en tiempo real con los datos de usuario de Spotify. Actualmente, el sistema simula las interacciones del usuario en lugar de usar datos de comportamiento en tiempo real, lo que podría limitar la efectividad y la personalización de las recomendaciones.

6.2 Líneas de trabajo futuro

Las futuras direcciones para mejorar el sistema de recomendación propuesto pueden incluir varios aspectos.

Primero, trabajar en la eficiencia del sistema de aprendizaje por refuerzo sería de gran interés. Esto podría involucrar la optimización de los parámetros del sistema o la exploración de diferentes técnicas de aprendizaje por refuerzo para acelerar el tiempo de aprendizaje y recomendación.

Además, sería útil desarrollar una interfaz de usuario más robusta que permita a los usuarios interactuar con el sistema y crear sus listas de reproducción de manera más ágil y sencilla.

Otra línea de trabajo sería la integración de datos de comportamiento en tiempo real de los usuarios de Spotify en el sistema de aprendizaje por refuerzo, lo que podría resultar en recomendaciones más personalizadas y efectivas.

Finalmente, sería beneficioso evaluar la eficacia y la eficiencia del sistema en otras plataformas de transmisión de música para asegurar su generalizabilidad y rendimiento.

Referencias

1. Adomavicius, G., Mobasher, B., Ricci, F. & Tuzhilin, A. Context-Aware Recommender Systems. *AI Magazine*. **32** pp. 67-80 (2011,9)
2. Kaminskas, M. & Bridge, D. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Trans. Interact. Intell. Syst.*. **7** (2016,12), <https://doi.org/10.1145/2926720>
3. Bonnin, G. & Jannach, D. Automated Generation of Music Playlists: Survey and Experiments. *ACM Computing Surveys*. **47** pp. 1-35 (2014,1)
4. Celma, O. Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space. *Music Recommendation And Discovery: The Long Tail, Long Fail, And Long Play In The Digital Music Space*. (2010,1)
5. Ricci, F., Rokach, L. & Shapira, B. Recommender Systems Handbook. *Recommender Systems Handbook*. **1-35** pp. 1-35 (2010,10)
6. Ziegler, C., Cai-Nicolas, McNee, S., M, S., Konstan, A, J., Lausen & Georg Improving recommendation lists through topic diversification. (2005,1)
7. Zhou, T., Kuscsik, Z., Liu, J., Medo, M., Wakeling, J. & Zhang, Y. Solving the apparent diversity-accuracy dilemma of recommender systems. *Proceedings Of The National Academy Of Sciences Of The United States Of America*. **107** pp. 4511-5 (2010,2)
8. Pichl, M., Zangerle, E. & Specht, G. Towards a Context-Aware Music Recommendation Approach: What is Hidden in the Playlist Name?. *2015 IEEE International Conference On Data Mining Workshop (ICDMW)*. pp. 1360-1365 (2015)
9. Vargas, S. & Castells, P. Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems. *Proceedings Of The Fifth ACM Conference On Recommender Systems*. pp. 109-116 (2011), <https://doi.org/10.1145/2043932.2043955>
10. Fleder, D. & Hosanagar, K. Blockbuster Culture's Next Rise or Fall: The Impact of Recommender Systems on Sales Diversity. *Management Science*. **55** (2007,10)
11. Kumar, V. Exploring the Balance Between Novelty and Familiarity in Recommendation Systems. (2018)
12. Bogdanov, D. & Herrera, P. How Much Metadata Do We Need in Music Recommendation? A Subjective Evaluation Using Preference Sets.. (2011,1)
13. Burke, R., Abdollahpouri, H., Mobasher, B. & Gupta, T. Towards Multi-Stakeholder Utility Evaluation of Recommender Systems. *User Modeling, Adaptation, And Personalization*. (2016)
14. Cantador, I. & Cremonesi, P. Tutorial on cross-domain recommender systems. *ACM Conference On Recommender Systems*. (2014)
15. Abdollahpouri, H. & Jannach, D. A survey on multi-objective recommender systems. (2023)
16. Shani, G. & Gunawardana, A. Evaluating Recommendation Systems. *Recommender Systems Handbook*. pp. 257-297 (2011), https://doi.org/10.1007/978-0-387-85820-3_8
17. Lozano Murciego, Á., Jiménez-Bravo, D., Valera Román, A., De Paz Santana, J. & Moreno-García, M. Context-Aware Recommender Systems in the Music Domain: A Systematic Literature Review. *Electronics*. **10** (2021), <https://www.mdpi.com/2079-9292/10/13/1555>
18. Quadrana, M., Jannach, D. & Cremonesi, P. Tutorial: Sequence-Aware Recommender Systems. (2019,5)
19. Kang, W. & McAuley, J. Self-Attentive Sequential Recommendation. *CoRR*. **abs/1808.09781** (2018), <http://arxiv.org/abs/1808.09781>

20. Knees, P. & Schedl, M. Music Similarity and Retrieval. (2016,1)
21. Konečný, J., McMahan, H., Yu, F., Richtárik, P., Suresh, A. & Bacon, D. Federated Learning: Strategies for Improving Communication Efficiency. *CoRR*. **abs/1610.05492** (2016), <http://arxiv.org/abs/1610.05492>
22. Gatzourou, A., Vinagre, J., Jorge, A. & Sánchez-Marrè, M. A Hybrid Recommender System for Improving Automatic Playlist Continuation. *IEEE Transactions On Knowledge And Data Engineering*. **33**, 1819-1830 (2021)
23. Hauver, D. & French, J. Flycasting: Using collaborative filtering to generate a playlist for online radio. *Proceedings Of WEDELMUSIC*. pp. 123- 130 (2001,12)

Aplicación de un COBOT para pick-and-place inteligente

David Cruz García, Vidal Moreno Rodilla y Belén Curto Diego

Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los
Caídos s/n. 37008, Salamanca, España
{david.cruz.garciaa, vmoreno, bcurto}@usal.es

Resumen En la actualidad, la colaboración entre humanos y robots se ha convertido en una realidad gracias a los grandes avances en la robótica y la inteligencia artificial. Un tipo de robot que ha ganado popularidad en la industria es el cobot, también conocido como robot colaborativo. Estos robots están diseñados para trabajar codo a codo con los humanos, brindando asistencia en tareas repetitivas, pesadas o peligrosas, y optimizando los procesos productivos. En este trabajo, nos centraremos en el cobot UR3e (Universal Robots 3e), un modelo de robot colaborativo desarrollado por Universal Robots. Además de los frameworks de Robot Operating System (ROS) en sus versiones ROS 1 y ROS 2, el software asociado al cobot UR3e incluye una herramienta fundamental: Polyscope. Permite la interacción y programación del cobot de manera intuitiva y eficiente, dirigida para todo tipo de público y facilitando su uso sin necesidad de conocimientos avanzados de programación. En conjunto, el uso de Polyscope como software de la tablet del cobot UR3e, junto con los frameworks ROS 1 y ROS 2, proporciona una solución completa y potente para la programación y control del cobot. En resumen, en este trabajo exploraremos el cobot UR3e y su programación utilizando los frameworks ROS 1 y ROS 2, junto con el software de la tablet del robot, Polyscope. Estudiaremos las capacidades de programación y control que ofrecen estas herramientas, y analizaremos su integración en entornos colaborativos.

Keywords: COBOTS, Universal Robots, Pick-And-Place, Inteligente, Software, ROS, Polyscope

1. Introducción

La industria 5.0 ha surgido como una nueva etapa de la revolución industrial, donde la integración de la tecnología y la colaboración hombre-máquina se convierten en elementos clave para impulsar la productividad en los procesos industriales. En este contexto, los cobots (robots colaborativos) están emergiendo como una solución innovadora y accesible para las pequeñas empresas que buscan mejorar su eficiencia y generar trabajo continuo ininterrumpido, acelerando las ventas y obteniendo mayores beneficios.

Los robots industriales se han convertido en una parte integral de muchas grandes empresas en los últimos años, y su integración continúa creciendo de manera significativa. Según la Federación Internacional de Robótica (IFR) [1], en 2022 se instalaron más de medio millón de nuevos robots industriales en todo el mundo, superando en un 22% el anterior record marcado en 2018 antes de la pandemia.

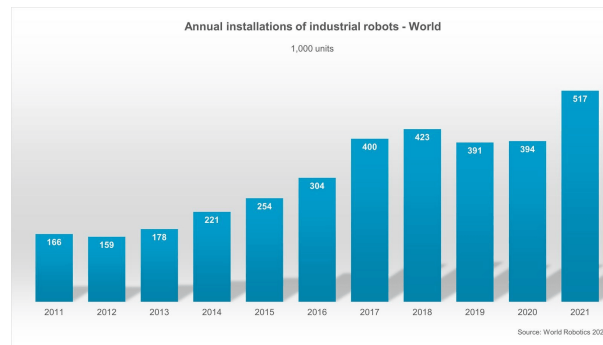


Figura 1: Crecimiento global en la instalación de robots industriales

Los robots colaborativos se están convirtiendo en grandes armas de producción, utilizándose en multitud de sectores y áreas como la sanidad, la agricultura o la hostelería fuera del ámbito industrial al que están destinados la mayoría de robots desarrollados. Se ha conseguido fomentar un tipo de robot que permite, de una forma muy sencilla y rápida, adaptarse a una tarea cualquiera (ya sea pick-and-place de objetos, cortado, fresado, etc) y trabajar en conjunto con el humano, multiplicando exponencialmente el rendimiento y la producción en muchos ámbitos.

Gracias a la aparición de estos robots, las empresas buscan facilidades para mejorar los procesos de producción a partir de robots que ofrezcan una programación sencilla y flexible. Con esto, las empresas consiguen que trabajadores con menos cualidades sean capaces de trabajar con junto a estos robots codo con codo. Esto incrementa la producción y la velocidad y agilidad a la hora de servir los productos a sus clientes. Por todo esto, las pequeñas y medianas empresas que invierten más en este tipo de robot consiguen ser mucho más competitivas dentro de sus sectores. Esto se puede ver en la figura 2 donde se muestra la evolución de los mercados por país.

En resumen, en este trabajo exploraremos los dos enfoques para el software de control de robots, código abierto y propietario, y se realizará, a ser posible, una tarea inteligente de pick-and-place como el paletizado de objetos para comparar ambos enfoques y las dificultades que surgen en la integración del robot.

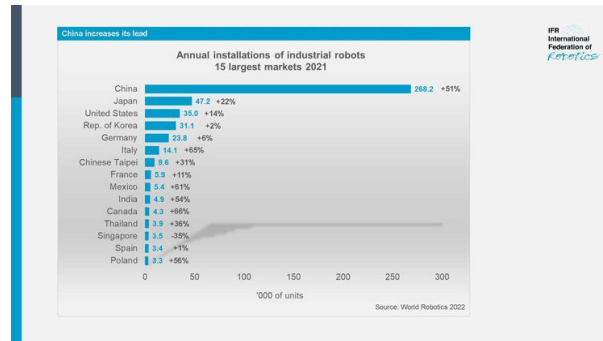


Figura 2: Crecimiento de los mercados por la instalación de robots industriales

2. Estado del arte

En este estudio del estado del arte, se llevará a cabo una recopilación general de los avances más relevantes en el campo de la robótica. Se centrará en cuatro aspectos clave: los robots convencionales y los COBOTs, los brazos robóticos, las manos o pinzas robóticas, y el software de control utilizado para manejarlos. Estos elementos representan componentes fundamentales en el diseño y funcionamiento de sistemas robóticos avanzados.

Desde hace años, los robots convencionales o industriales han demostrado su eficacia en procesos de fabricación y manipulación de materiales, mientras que por otro lado, los COBOTs o robots colaborativos han surgido hace algunos años como una solución que permite la interacción segura entre humanos y robots en entornos de trabajo compartidos.

En este artículo se revisará el estudio de los cobots y los robots convencionales, los brazos articulados y pinzas en la actualidad, y los softwares de control que se utilizan para controlarlos.

Un software de control de robots es una herramienta fundamental para programar, supervisar y controlar las operaciones de los robots industriales. Este tipo de software proporciona interfaces intuitivas y funciones avanzadas que permiten a los usuarios interactuar con los robots y definir sus movimientos y comportamientos, permitiéndoles realizar casi cualquier tarea de manipulación de objetos.

Los primeros softwares de control de robots convencionales y/o COBOTs fueron desarrollados hace varias décadas. En los inicios de la robótica, no existían softwares específicos para controlar robots industriales. En su lugar, se utilizaban sistemas de control basados en cables y levas, como en el caso del Unimate, considerado el primer robot industrial, desarrollado por George Devol y Joseph Engelberger en 1961 [2]. A medida que la tecnología robótica avanzaba, surgieron lenguajes de programación como el VAL (Versatile Assembly Language) en la década de 1970, que se utilizaba para controlar robots manipuladores [3].

Los software de control antiguos para robots convencionales solían tener interfaces de usuario más limitadas y complejas, requiriendo programación deta-

llada, compleja y manual mediante comandos específicos. En la actualidad, los software modernos ofrecen interfaces gráficas intuitivas y amigables, simplificando la programación a través de interfaces visuales y lenguajes de programación más intuitivos. Además, los software de control de robots modernos cuentan con potentes capacidades de simulación y planificación, permitiendo a los usuarios probar y optimizar programas de control en entornos virtuales antes de su implementación, algo que en el pasado era inviable por las limitaciones de potencia de los dispositivos empleados.

En la actualidad, se dispone de una amplia gama de softwares para controlar robots convencionales y cobots, diseñados para satisfacer las necesidades específicas de cada aplicación y fabricante. Algunos ejemplos de softwares modernos son:

1. ROS (Robot Operating System): ROS es una plataforma de software de código abierto ampliamente utilizada en la comunidad robótica. Fue desarrollada inicialmente en 2007 por el Laboratorio de Inteligencia Artificial de Stanford (Stanford AI Lab) y actualmente cuenta con una comunidad activa de desarrolladores que contribuyen con módulos y herramientas [4].
2. ABB RobotStudio: RobotStudio es un software de programación y simulación utilizado para los robots de ABB. Permite la creación y simulación de programas de control, así como la optimización y visualización de las tareas de los robots en un entorno virtual [5].
3. Universal Robots Polyscope: Universal Robots es conocido por sus cobots colaborativos, y el software de control utilizado para ellos es Polyscope. Según un artículo publicado en la revista *IEEE Robotics & Automation Magazine*, el software Polyscope proporciona una interfaz de programación intuitiva y basada en iconos que permite a los usuarios configurar tareas, movimientos y lógica de control de forma sencilla [6].

3. Caso de estudio

En este apartado de caso de estudio, se analizará la implementación de un brazo robótico y una pinza en un entorno de control utilizando ROS (Robot Operating System) y el software Polyscope. El brazo robótico seleccionado es el modelo UR3e de Universal Robots, junto con una pinza de agarre RG2 para manipulación de objetos. Para la programación y control del sistema, se emplearán varias versiones de ROS comparando ROS1 y ROS2 y los problemas y resultados que presentan con la implementación del driver del robot para ROS, aprovechando los paquetes y librerías disponibles para planificación de movimientos y comunicación entre componentes como lo es el paquete Moveit. Por otro lado, el software Polyscope, desarrollado por Universal Robots, será utilizado como interfaz de control del brazo robótico, proporcionando la funcionalidad necesaria para desarrollar un programa pick and place o incluso servir como puente entre el robot y ROS para su control.

3.1 Hardware

Para el desarrollo de esta investigación, se ha utilizado el brazo robótico UR3e de Universal Robots. [7]

Se trata de un robot manipulador colaborativo diseñado para realizar tareas precisas y delicadas en entornos de trabajo colaborativos con humanos. Tiene un diseño compacto y ligero y es altamente versátil, y se adapta fácilmente a espacios de trabajo reducidos por su alcance reducido. Dispone de una carga útil de 3 kg y puede manejar una gran variedad de objetos, lo que lo hace ideal para aplicaciones de ensamblaje, manipulación de materiales y tareas de precisión en industrias como la electrónica o la automotriz.

El UR3e cuenta con seis grados de libertad, lo que le permite realizar movimientos flexibles y precisos en un amplio rango de trabajo. Como cualquier robot colaborativo de Universal Robots, incorpora el software Polyscope, lo que permite programar tareas de forma sencilla e intuitiva, e integrar el robot en otro tipo de sistemas como puede ser ROS. Se explicará Polyscope en apartados posteriores. En la figura 3 se puede observar el cobot UR3e empleado.

Además se ha utilizado la pinza OnRobot RG2 de Universal Robots [8]. La pinza RG2 es un accesorio diseñado para trabajar en conjunto con los brazos robóticos de Universal Robots, como el UR3e. Es una pinza eléctrica flexible utilizada en aplicaciones industriales y de ensamblaje para agarrar y manipular una amplia variedad de objetos.

Una de las características destacadas de la pinza RG2 es su rango de apertura de 0 a 110mm, lo que le permite adaptarse a diferentes tamaños y formas de objetos, haciéndola versátil y adecuada para manipular desde objetos pequeños y delicados hasta objetos más grandes y robustos. La pinza RG2 ofrece control de agarre flexible con detección de fuerza. Puede ajustar la fuerza de agarre según las necesidades de la aplicación y también detectar si el objeto ha sido agarrado correctamente. Esto brinda precisión y adaptabilidad en la manipulación de objetos, proporcionando gran cantidad de información como feedback al usuario.

La pinza RG2 se puede programar y controlar directamente desde el software UR, lo que facilita su integración con los brazos robóticos UR. La programación se realiza a través de una interfaz gráfica intuitiva, lo que simplifica la configuración y personalización de la pinza. Esto se hace a través de un subprograma que se añade al software Polyscope del brazo robótico, permitiendo controlar la pinza desde el mismo software.

3.2 Software

3.2.1 ROS El primer software que se decidió utilizar fue ROS. ROS es un meta sistema operativo de código abierto que permite a los programadores de robots implementar funcionalidades de control de robots abstrayendo la parte hardware del usuario final.

ROS trabaja mediante paso de mensajes entre procesos, denominados nodos, que pueden ubicarse en la misma o diferentes máquinas. Se trata de un red peer-to-peer que permite una comunicación síncrona entre servicios o asíncrona a través de *topics*.



Figura 3: Brazo colaborativo UR3e utilizado



Figura 4: Pinza utilizada para el brazo robótico

Este framework se diferencia de otras plataformas robóticas por su enfoque en la reutilización de código. Se basa en una estructura distribuida de nodos que se pueden diseñar de manera individualizada pero acoplarse fácilmente entre sí. Los procesos se agrupan en paquetes que pueden intercambiarse, compartirse y distribuirse fácilmente, ofreciendo funcionalidades reutilizables para otros desarrolladores o investigadores. Además, ROS ofrece un sistema federado de repositorios de código que facilita la colaboración. Además, ROS es liviano, compatible con múltiples lenguajes de programación y cuenta con un marco de prueba integrado. Es escalable y adecuado para sistemas y procesos de desarrollo de gran envergadura.

En ROS1 [9] se desarrolla un sistema centralizado en un nodo denominado nodo maestro o “rosmaster”. Este nodo maestro sirve como coordinador, registrando todos los nodos, *topics* y servicios disponibles. Para entender este funcionamiento, hay que tener claros los siguientes conceptos, mostrados en la figura 5:

1. Nodos: Son procesos que realizan alguna tarea. Un sistema de control de robots completo puede constar de uno o varios nodos distribuidos en una o varias máquinas. Los nodos se organizan siguiendo una estructura de grafo de nodos, donde cada uno de ellos se comunica o no con otros nodos.
2. Rosmaster: Es el nodo central encargado de coordinar al resto de nodos, por lo que este nodo debe estar siempre arrancado y todos los nodos del sistema deben saber de su existencia (en qué máquina se encuentra).
3. Mensajes: Es una estructura de datos que permite a los nodos comunicarse entre ellos. Pueden ser números enteros o decimales, cadenas de texto u otros tipos de datos.
4. *Topics*: Son canales de comunicación entre nodos que actúan de forma similar a un buzón siguiendo el patrón publicador-suscriptor. Cuando un nodo

(el publicador) quiere enviar información a otros nodos (los suscriptores), publicará mensajes del tipo que corresponda a un tópic del que leerán los nodos suscriptores. Permiten una comunicación muchos a muchos de forma asíncrona y simultánea entre varios nodos.

5. Servicios: Al igual que los *topics*, los servicios también son canales de comunicación entre nodos pero del tipo uno a uno, siguiendo el patrón request-reply. Con los servicios, los nodos pueden pedir a otros nodos realizar una acción y recibir una respuesta a esa acción, es decir, un nodo solicitará y dará servicios a otros nodos para realizar acciones especificar, diferenciándose así de los *topics* cuyo objetivo es el de difundir información.

A partir de aquí surge el concepto de paquete de ROS. Una vez definida la forma de comunicación entre diferentes nodos se puede comenzar a crear una pieza software capaz de resolver un problema utilizando este sistema de paso de mensajes entre nodos. Estas unidades de software se organizan mediante los paquetes de ROS. Estos paquetes contienen los nodos, librerías, información, tipos de datos y configuraciones de forma que se compone una unidad atómica reutilizable de software.

Gracias a la creación de paquetes surgen conceptos más avanzados de ROS, como los paquetes “primitivos” que contienen funcionalidad que se podría reutilizar en todos o casi todos los proyectos. Algunos de estos paquetes son “tf” para transformaciones, “actionlib” para definir acciones, “common_msgs” contiene los mensajes utilizados comúnmente o “rviz” para la visualización. También hay otros paquetes como “urdf” que permite trabajar con ficheros URDF (Unified Robot Description Format) que son ficheros en formato XML y sirven para crear una descripción del robot que se empleará en otras herramientas para conocer su estructura física, permitiendo a las diferentes herramientas conocer la forma del robot. Otro paquete importante para este proyecto es “moveit”, que permite cargar un robot y trabajar con el de forma visual, permitiendo planificar trayectorias y ejecutarlas.

Otro concepto interesante es el espacio de trabajo o workspace. Un espacio de trabajo en ROS es un entorno estructurado para organizar, desarrollar, compilar y ejecutar paquetes de ROS con el fin de desarrollar una aplicación completa.

Para trabajar con estos espacios de trabajo existe la herramienta “catkin” [10]. Esta herramienta permite construir y gestionar los paquetes dentro de un espacio de trabajo de forma automática. Se basa en CMake como sistema de compilación, siguiendo una estructura ordenada de directorios donde se almacenarán los paquetes, los ejecutables compilados y los ficheros de configuración.

Al igual que cualquier pieza software, ROS1 tenía sus limitaciones y problemas como la seguridad o la fiabilidad en algunos entornos que limitaban o entorpecían el desarrollo de algunas funcionalidades o dificultaban la evolución de sistemas complejos. Es por todo esto que surge en 2014 ROS2 [11]. La nueva versión de ROS continua con todos los puntos fuertes de ROS1 pero trata de resolver todas estas limitaciones que presenta.

La principal diferencia entre ROS2 y ROS1 es su arquitectura: ROS1 seguía una arquitectura centralizada en el nodo maestro el cual era necesario ejecutarlo

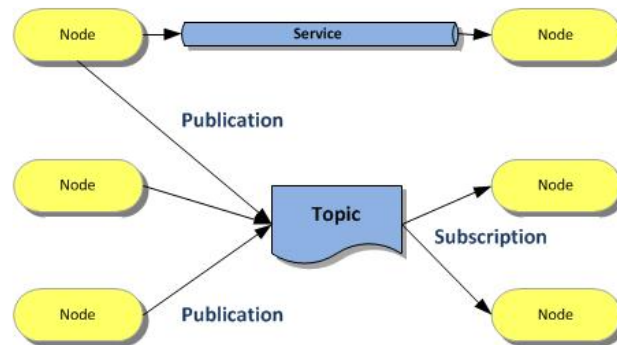


Figura 5: Comunicación en ROS 1

en alguna máquina para que el sistema funcionara y se seguían comunicaciones punto a punto; en cambio, en ROS2 se sigue una arquitectura distribuida, donde cada nodo funciona de forma independiente en cualquier entorno y se comunica con el resto de nodos a través de los llamados IDs de dominio que permiten identificar qué nodos forman parte de un ecosistema para poder comunicarse. Cuando se arranca un nodo en ROS2, se le establece un “ROS_DOMAIN_ID” por el que se anunciará y al que el resto de nodos del mismo dominio responderán. Con esto todos los nodos sabrán de la existencia de otros dentro de un dominio, aunque periódicamente se anunciarán de nuevo para seguir avisando de su correcto funcionamiento. Una vez un nodo cae, ya sea por error o por finalización correcta, avisa al resto de nodos del mismo dominio. Esta arquitectura se denomina DDS (Data Distribution Service) y permite comunicaciones más eficientes y confiables.

Además, ROS2 ofrece una compatibilidad multiplataforma mientras que ROS1 solo tenía soporte para Linux. También incluye grandes mejoras en seguridad para autenticación, cifrado o control de acceso, permitiendo comunicaciones entre nodos mucho más seguras. Otra de las mejoras que incluye es el soporte para sistemas de tiempo real con el middleware RTPS (Real-Time Publish-Subscribe) de DDS. Finalmente, otra de las mejoras que presenta ROS2 es un mayor soporte para otros lenguajes de programación o incluso versiones más actuales de los ya utilizados en ROS1.

ROS2 es una evolución significativa de ROS1, pero sigue la misma filosofía. ROS2 trabaja al igual que ROS1 con nodos, mensajes, servicios, acciones y con el mismo tipo de ficheros como los ficheros “msg” para los mensajes o los ficheros “URDF” para los modelos de los sistemas robóticos.

3.2.2 Polyscope Polyscope es una interfaz de programación y visualización para robots colaborativos (cobots) fabricados por Universal Robots. Es una herramienta que permite programar, controlar y supervisar los robots UR de forma

intuitiva y sencilla. Proporciona una interfaz gráfica fácil de usar que permite a los usuarios interactuar con el robot a través de una pantalla táctil. Esta pantalla permite controlar todo el robot e incluso dispone de mecanismos de seguridad para liberar el control del movimiento del robot y poder moverlo a mano o incluso detenerlo por completo mediante una seta de emergencia.

Proporciona es una interfaz de programación versátil para robots de Universal Robots. Utilizando un enfoque de programación por flujo, los usuarios pueden crear programas de manera visual mediante bloques de función, sin necesidad de escribir código manualmente. Esta facilidad de uso se complementa con funciones de control en tiempo real, lo que permite iniciar, detener y supervisar movimientos del robot durante la ejecución. Además, Polyscope ofrece opciones de configuración y calibración sencillas, como la definición de zonas de trabajo y ajustes de velocidad y fuerza. Para tareas más complejas, la plataforma admite programación avanzada utilizando URScript, un lenguaje de script personalizado. También cuenta con herramientas de monitoreo y diagnóstico para el seguimiento del estado del robot y la solución de problemas. La extensibilidad es otra ventaja clave de Polyscope, gracias a los URCaps, que permiten añadir funcionalidades adicionales al robot de forma modular. En general, Polyscope es una interfaz completa y flexible que facilita la programación, el control y la expansión de las capacidades de los robots de Universal Robots.

4. Implementación y resultados

En esta sección se mostrarán los pasos seguidos y los resultados alcanzados para las diferentes versiones de ROS donde se ha tratado de integrar el driver del brazo robótico UR3e y de los programas desarrollados para Polyscope.

4.0.1 ROS1 En primer lugar, y para poder trabajar con todos los paquetes necesarios, se ha creado un workspace. En este workspace se almacenarán y compilarán todos los paquetes encargados de mover el brazo robótico utilizando ROS. Estos paquetes son: “Universal Robots ROS Driver” de Universal Robots [12], que proporciona el driver adaptado para los robots de Universal Robots junto a un paquete de calibración, y el paquete “universal_robots” de ros_industrial [13], que proporciona drivers, configuraciones y herramientas para robots industriales, teniendo un paquete específico con herramientas para los cobots de Universal Robots. Una vez descargados y compilados los paquetes en el workspace, se ejecuta la calibración para obtener la información del robot como la posición o los parámetros que emplea, y, finalmente, se arranca el driver. Una vez el driver ha arrancado, puede conectarse el robot mediante el URCap de “External Control” de Polyscope.

A partir de este punto, el driver está correctamente arrancado y aparecen una serie de nodos y *topics* que permiten controlar el robot estableciendo las posiciones de cada articulación o devuelven información del estado del robot. Esto se puede visualizar con la herramienta “rqt_graph” como se puede ver en la figura 6.

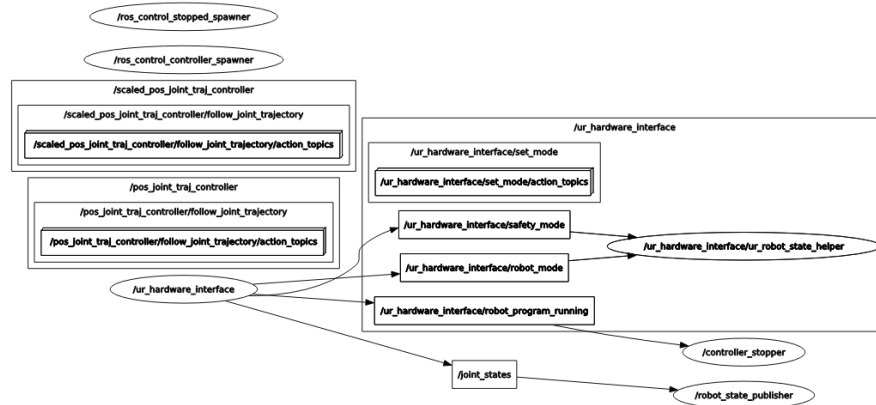


Figura 6: *Topics* y nodos creados con el driver

Como se puede ver en el grafo se inician varios *topics* referentes al robot. Los *topics* referentes al control del robot y los encargados de transmitir la información necesaria entre el robot y ROS son los siguientes:

1. Los *subtopics* dentro de “pos_joint_traj_controller”: Estos *topics* sirven para generar y realizar trayectorias de posición para las articulaciones del robot. Como se tratan de acciones a realizar, hay *topics* dentro de este conjunto para establecer el goal o recibir el feedback y el resultado, así como ver el estado de la acción. Con este tópico
2. Los *subtopics* dentro de “scaled_pos_joint_traj_controller”: Tienen como propósito escalar los valores enviados a los *topics* anteriores para adaptarlos a las necesidades de cada controlador.
3. Los *subtopics* dentro de “ur_hardware_interface”: Estos *topics* son los encargados de la comunicación entre ROS y Polyscope. Dentro de este conjunto de *topics* hay *topics* para ver el estado de la E/S del robot, el programa que se está ejecutando, el modo del robot (en caso de estar parado del todo, parado por emergencia, arrancado, etc), si está activado o no el mecanismo de seguridad o *topics* que realizan una acción para cambiar le modo del robot desde ROS.
4. Tópico “joint_states”: Sirve para comunicar información referente a la posición, velocidad y esfuerzo de cada una de las articulaciones del robot. Con este tópico los planificadores o controladores pueden obtener información en tiempo real del estado de las articulaciones del robot.
5. Otros *topics*: además de todos estos *topics*, hay algunos otros. Entre ellos estan “robot_state_publisher” que se encarga de publicar la información referente al tópico de “joint_states”; el tópico “controller_stopper” que se encarga de detener el controlador en el caso de que el programa que se ejecuta en el robot se detenga; el tópico “tf” encargado de todas las transformaciones de cada componente del robot; o otros *topics* que se ejecutan al arrancar .

Con todo esto el usuario ya puede mover el robot enviando la información necesaria a dichos *topics*. Sin embargo, para realizar planificación o facilitar estas tareas, surgen los controladores o planificadores.

Se dispone de un controlador denominado “`rq_t_joint_trajectory_controller`” que permite trabajar directamente con los *topics* de posición escalada del robot. El controlador se encarga de proporcionar una interfaz gráfica muy sencilla con unas barras donde el usuario podrá interactuar con cada articulación del robot por separado. Para poder trabajar de forma visual y poder realizar planificación de las trayectorias del robot, se ha optado por el uso de Moveit. Con esto ya podemos visualizar el robot en RVIZ y planificar movimientos de forma rápida y sencilla.

Para comprobar que el driver del cobot funciona con los planificadores de Moveit! se han diseñado 3 pruebas. Una prueba donde se comprueba el movimiento correcto del robot con la configuración más sencilla, y dos pruebas con varios escenarios para comprobar que la planificación de caminos es la correcta.

La primera prueba consta de un movimiento simple, donde se trata de colocar el robot en posición vertical. En la figura 7 se pueden observar tanto la posición inicial (en verde) como la posición final (en naranja) y la trayectoria seguida para realizar el movimiento establecido. En las figuras 8 y 9 puede observarse el inicio y final respectivamente del movimiento en el robot real.

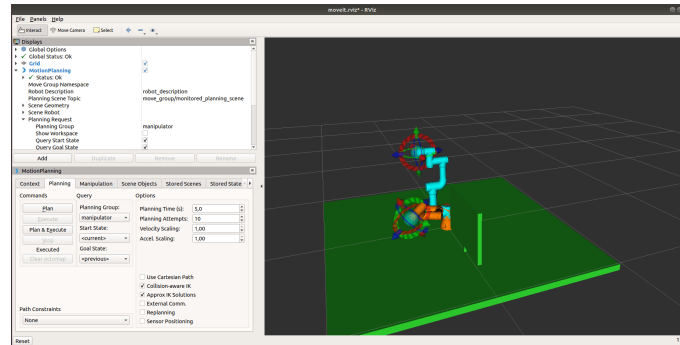


Figura 7: Escena simulada para un entorno real



Figura 8: Inicio del movimiento simple en el robot real

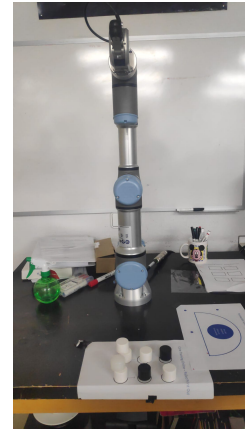


Figura 9: Fin del movimiento simple en el robot real

La segunda prueba consistió en simular un espacio de trabajo con objetos, donde hay que planificar una ruta libre de colisiones con los objetos del entorno, para que el robot se mueva desde una configuración inicial a otra final. Esta segunda prueba se puede ver en las figuras 10, 11 y 12.

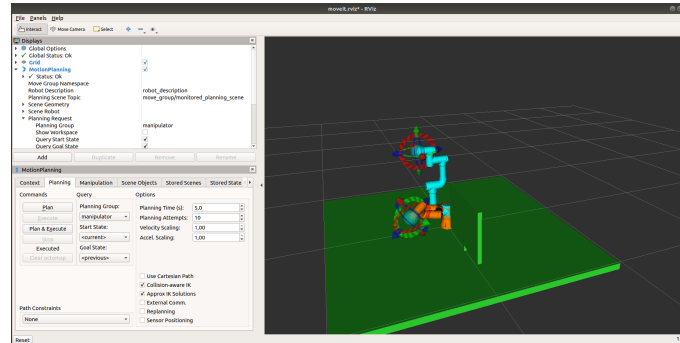


Figura 10: Escena simulada para un entorno real

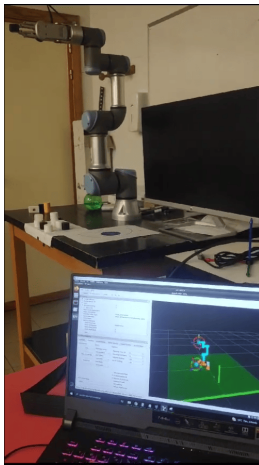


Figura 11: Movimiento intermedio del robot en un escenario real



Figura 12: Fin del movimiento del robot en un escenario real

El problema al tener un robot con un espacio de trabajo tan pequeño y colocar objetos tan grandes, el planificador no es capaz de generar algunos movimientos. Para tratar de resolver este problema, se decidió crear un tercer escenario con varias cajas flotantes simuladas (aunque no colocadas en el entorno real). Con esto se quería comprobar si, al colocar objetos más pequeños, el planificador era capaz de generar la trayectoria correcta para esquivar dichos objetos. En las figuras 13, 14, y 15 puede verse, al igual que los escenarios anteriores, tanto el

movimiento que se va a realizar, marcando el inicio en azul y el final en naranja, y el movimiento realizado por el robot real.

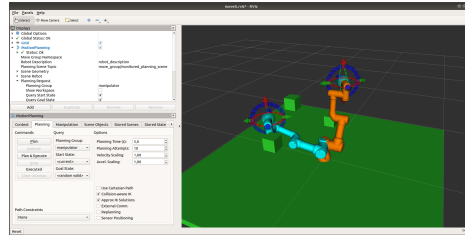


Figura 13: Escena con cajas simuladas en Moveit

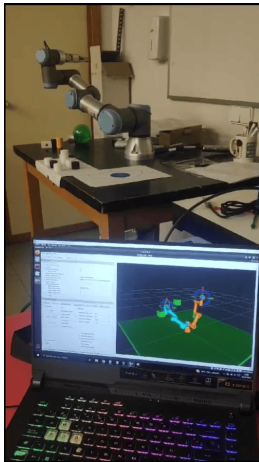


Figura 14: Inicio del movimiento del robot en un escenario con cajas simuladas

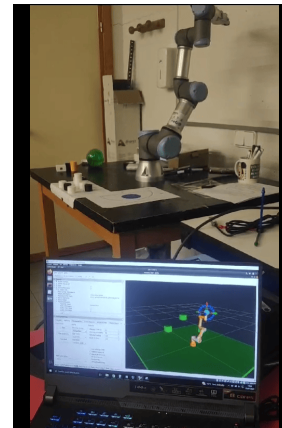


Figura 15: Fin del movimiento del robot en un escenario con cajas simuladas

Con estos ejemplos se ha comprobado que el driver del brazo funciona de forma correcta con los planificadores de Moveit.

El siguiente objetivo planteado sería añadir el funcionamiento la pinza para manipular objetos con el cobot, generando trayectorias de forma automática.

Para ello, se han investigado drivers que fueran compatibles para la pinza RG2 para ROS1 y que contengan un modelo URDF de la pinza. Solo hay disponibles de 2 drivers, ambos basados en el mismo driver de “ros_industrial” llamado “robotiq” [14].

Tras comprobar el funcionamiento de estos drivers y sabiendo que la pinza solo funciona en Polyscope añadiendo el URCap necesario, no se ha conseguido ningún resultado positivo. Este driver trabaja mediante el protocolo MODBUS, que se trata de un protocolo de comunicación para el intercambio de información con arquitectura maestro-esclavo (Polyscope sería el maestro y el driver de la pinza el esclavo). Arrancando el driver de la pinza nunca se ha conseguido que conecte con Polyscope, por lo que la pinza real del brazo nunca se abre. Además, el driver está pensado para trabajar con la pinza únicamente, por lo que sería necesario extender el driver del brazo junto al driver de la pinza, todo ello unido, al igual que los ficheros URDF para que los planificadores puedan trabajar de forma correcta. El desarrollo del driver completo para generar la funcionalidad de detección de la fuerza empleada, el movimiento de la pinza y el movimiento completo del brazo resulta una tarea larga tratándose de ROS, por lo que este sistema se ha propuesto como la principal línea de trabajo futura.

4.0.2 ROS2 En este apartado se tratará de realizar las mismas tareas que en el apartado anterior pero trabajando con el driver de ROS2 en distintas versiones: Ros Foxy Fitzroy, ROS Humble Hawksbill y ROS Rolling Rodley. El principal problema, que se verá en cada apartado, es el “poco” tiempo de desarrollo de la arquitectura de ROS2 y de este driver en ella. Esto se ve agravado por el poco interés de la comunidad en migrar sus proyectos desde ROS1, por lo que la mayoría de desarrolladores siguen trabajando en ROS1. En ROS2 las versiones cambian de forma muy rápida, por lo que el desarrollo de paquetes se ve ralentizado por las adaptaciones que necesitan para cada versión y problemas que puedan surgir. Es por esto que se ha procedido a trabajar con varias versiones de ROS2 para comprobar el funcionamiento del driver de Universal Robots entre versiones.

En cuanto a las versiones de ROS2 Foxy y Humble, no se ha conseguido inicial el driver. En estas versiones existe un bug que causa un error al cargar el modelo del robot, por lo que el driver queda en un bucle constante de carga del modelo y no permite la conexión del robot. Este error puede verse en la figura 16 Para tratar de arrancarlo se revisó el fichero URDF del robot, tratando de modificar las partes que pudieran causar error, pero seguía ocurriendo el mismo error.

Para tratar de trabajar con el software más actualizado, se trató de trabajar con la versión ROS Rolling. Esta versión sirve como campo de pruebas para el desarrollo de ROS2. En esta versión de ROS se implementan todos los nuevos

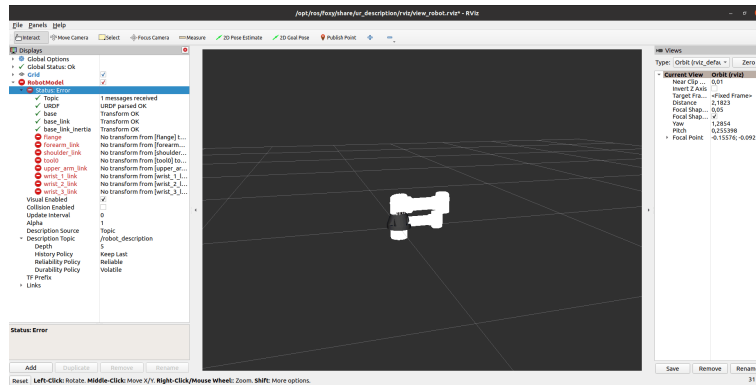


Figura 16: Error en la carga del modelo en RVIZ en ROS Foxy y ROS Humble

cambios que se implementarán en las versiones “oficiales” de ROS, por lo que no es recomendable trabajar en proyectos grandes con esta versión por posibles errores que puedan surgir en su desarrollo. Sin embargo, para este proyecto la versión con mejores actualizaciones por ser la que continuamente se mejora es esta, por lo que como última opción se trata de conseguir mover el robot desde ROS2.

A diferencia de las otras versiones de ROS2, esta vez el driver si que funciona y arranca correctamente, cargando el modelo del robot en RVIZ y permitiendo el movimiento del robot. En la figura 17 se muestra tanto la conexión correcta del robot con el driver como el modelo URDF cargado de forma correcta en RVIZ.

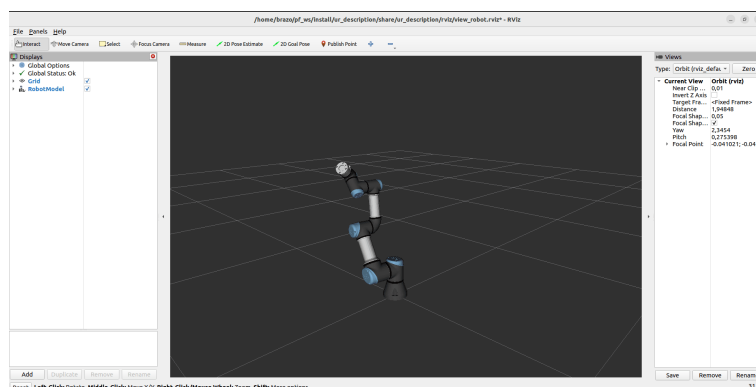


Figura 17: Modelo del robot en RVIZ en ROS2

Una vez arrancado el driver, se despliegan todos los nodos y *topics* necesarios para el control del robot. Estos nodos y *topics* se pueden ver en la figura 18.

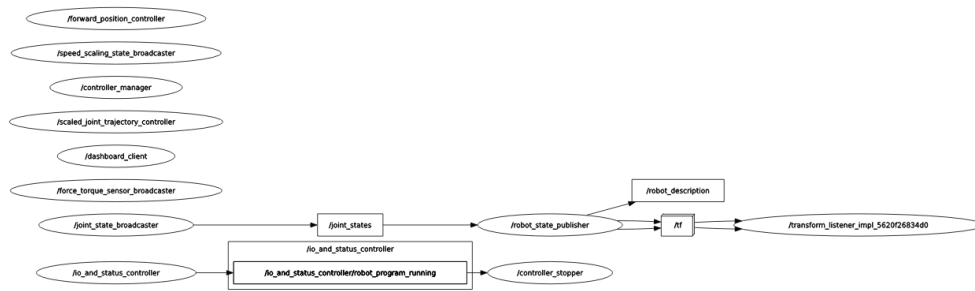


Figura 18: Grafo de nodos y *topics* en ROS2

1. Los *topics* “joint_state_broadcaster”, “joint_states”, “robot_state_publisher” y “robot_description” se asemejan a los *topics* de ROS1. Trabajan la información sobre el estado de las articulaciones y del estado del modelo del robot para coordinarlos y mostrar las diferentes posiciones en el RVIZ.
2. Los *topics* de la rama de “io_and_status_controller” se corresponden a los *topics* de la rama “ur_hardware_interface” en ROS1. Todos los *topics* correspondientes a estas ramas se encargan de la comunicación entre ROS y el Polyscope, obteniendo y enviando toda la información relativa al estado del robot, el modo activo del mismo, si hay algún programa en ejecución en el brazo robótico, etc.
3. Los *topics* ‘initial_pose’ y ‘goal_pose’ sirven para especificar en RVIZ la posición inicial del robot a la hora de cargar el modelo o indicarle la posición que se desea alcanzar.
4. Los *topics* “scaled_joint_trajectory_controller”, “speed_scaling_state_broadcaster”, “force_torque_sensor_broadcaster”, “dashboard_client” y “controller_manager” proporcionan información sobre el robot, sobre el control de trayectoria y velocidad, la fuerza que están detectando su sensores o información de control del robot.
5. Existen otros *topics* que se inician de forma automática por ros como “ros_out” y que son *topics* con los que trabaja ROS y ofrecen información de depuración.

En resumen, los *topics* generados por el driver de ROS2 son equivalentes a los *topics* generados en ROS1. La forma de trabajar de ambos framework deben proporcionar la información correcta para que Polyscope la interprete. Es por esto que los *topics* generados al lanzar el driver son equivalentes aunque con distintos nombres. Aún así, aunque el planificador no funcionara, el robot podía moverse. Para ello, utilicé el controlador del paquete “rqt” tratando de controlar el robot de forma manual y comprobar si el driver estaba funcionando correctamente, y así fue. Al igual que sucedía en ROS1, enviando la información correcta de movimiento a los *topics* a través de un controlador muy sencillo se podía controlar el robot.

4.0.3 Polyscope Polyscope es una interfaz gráfica basada en iconos y menús desplegados, diseñada para facilitar la programación y el control del cobot UR3e.

Para tratar de comparar la forma en la que Polyscope facilita el desarrollo de tareas, se han desarrollado 2 programas:

1. Un programa de paletizado, donde se le proporciona al cobot diferentes piezas en un lugar concreto y las coloca ordenadas siguiendo un patrón de paletización. Se trata de un pick and place pero con el propósito de colocar de forma ordenada objetos. Como ejemplo de aplicación industrial, podrían llegar desde una cinta transportadora y el robot se encargaría de colocarlos para el transporte.
2. Un programa de pintado, donde el robot con un rotulador sujeto con la pinza es capaz de pintar cualquier tipo de dibujo en una pizarra. Esto se realiza a través de una URCap, donde dado un trazo específico como una trayectoria en GCODE (lenguaje de programación en CNCs), el robot es capaz de reproducirlo. Como ejemplo de aplicación industrial, este mismo programa podría utilizarse a modo de CNC en el fresado y corte de pieza mecanizadas.

Para el paletizado se dispone de unas marcas a modo de rejilla y unos cilindros que el cobot tendrá que mover y colocar. Esta rejilla está hecha con marcas en un papel. Esta rejilla tiene 6 posiciones y se dispone de 8 figuras, por lo que para el paletizado se establecerán estas 6 posiciones y varias capas en altura para colocar de 6 en 6 piezas por capa (en este caso, colocará 6 piezas en la primera capa y 2 en la segunda capa). El robot coge piezas desde la posición inicial establecida, por lo que el primer paso será desplazar el robot hasta la posición de surtido de piezas. Una vez el robot coja la pieza, se encargará de realizar el movimiento hasta la siguiente posición libre de la rejilla (o de la capa correspondiente). Esto se repetirá hasta que el usuario que colabora con el robot coloque todas las piezas. Este proceso puede verse en las figuras 19 y 20, donde se puede ver el momento en el que el robot recoge una pieza desde la posición de recogida o el resultado final del paletizado.

En el proceso de pintado hay que definir una trayectoria que realice el cobot, al que se le ha colocado en la pinza un rotulador o, en una aplicación industrial, un herramienta de pintura. Estas trayectorias, en otros campos como las impresoras 3D, se realizan mediante los comandos especificados en ficheros GCODE, que indican los movimientos que debe seguir.

Los robots de Universal Robots disponen de un URCap denominado “Remote RCP & Toolpath” que permite cargar un GCODE generado para un CNC y reproducir esos movimientos.

Los pasos a seguir son sencillos. En primer lugar, se exporta utilizando un programa como Fusion 360 el GCODE de un dibujo y se carga mediante un USB a Polyscope. Esto se hace en la figura. En este caso, se ha cargado el fichero GCODE para pintar un mandala.

Una vez cargado, se genera un programa únicamente con el URCap. Este URCap permite seleccionar la trayectoria que hemos importado y un plano donde se va a realizar.



Figura 19: Cobot recogiendo la 4a pieza del paletizado

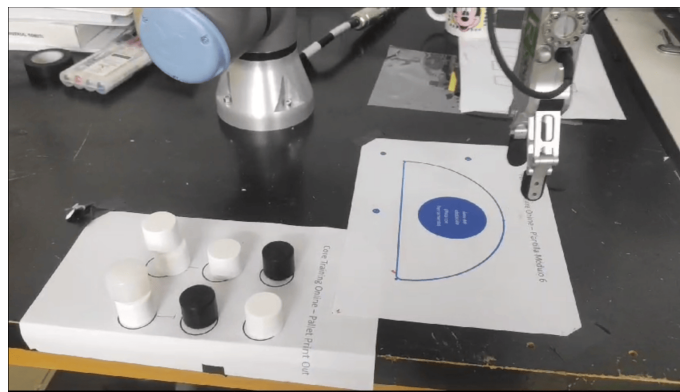


Figura 20: Resultado final del paletizado con 2 capas

Se define el plano, en este caso, vertical, ya que es una pizarra en la pared. Finalmente, se ejecuta el programa. El resultado final se puede ver en la figura 21.

Como se puede ver, los pasos son muy sencillos gracias a las herramientas proporcionadas por Polyscope que ofrecen una interfaz de alto nivel para el usuario que controla el robot, facilitando su trabajo sin necesidad de tener conocimientos avanzados de programación.

5. Conclusiones

Con el desarrollo de este trabajo de investigación se ha llevado a cabo la integración del cobot UR3e de Universal Robots en el entorno de ROS. Con esto se ha permitido conocer de primera mano la dificultad que conlleva realizar esta integración y los resultados que se pueden obtener con las versiones actuales del software. Esto ha permitido no solo aprender sobre cómo integrar un robot

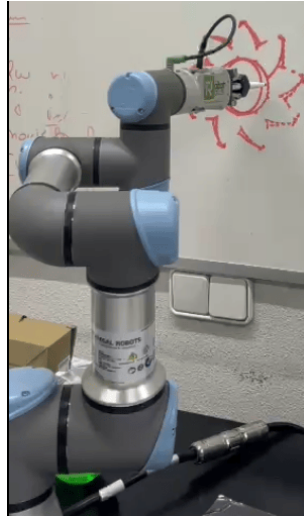


Figura 21: Resultado final del pintado del mandala

colaborativo dentro de un entorno OpenSource, sino que ha permitido investigar a fondo el funcionamiento de un framework robótico con elevada importancia en el sector industrial. También se ha conseguido adquirir los conocimientos para integrar robots de construcción propia o customizados para desarrollar otros proyectos.

Durante el desarrollo de este trabajo de investigación, se han realizado estudios en simulación e integración de un cobot real UR3e dentro del entorno ROS, tanto ROS1 como ROS2, así como la posibilidad de realizar expansiones a estos proyectos incorporando escenarios u otras funcionalidades. También se ha tratado de integrar una pinza robótica en un cobot articulado, que, aunque el resultado no ha sido exitoso, ha permitido adquirir conocimientos sobre el funcionamiento de este tipo de instrumentos de manipulación de objetos e incluso de herramientas software para su control e integración en otros robots.

También, el trabajo se ha centrado en estudiar el software Polyscope, también se ha centrado la investigación en el software Polyscope. La idea de disponer de un brazo robótico con un software tan elaborado y con tantas funcionalidades ha permitido poder comparar las funcionalidades entre un software genérico y opensource como es ROS frente a un software creado por una empresa exclusivamente para este tipo de actuadores. Con este software se han creado 2 programas que permiten realizar un pick-and-place usando el cobot UR3e junto a la pinza RG2. El propósito del primer programa es paletizar materiales, que tiene una aplicación directa en el sector industrial para el almacenaje y la logística. El segundo programa permite cargar y realizar secuencias de movimientos a partir de una trayectoria dada como GCODE, lo que da pie a multitud de aplicaciones industriales, como el pintado o el fresado de objetos.

Con este proyecto, se ha conseguido integrar el brazo UR3e en un entorno ROS y estudiar su funcionamiento. En particular, la integración en ROS 1 ha resultado muy satisfactoria, permitiendo realizar trayectorias planificadas utilizando Moveit junto al robot real. Por otro lado, la integración en ROS2 ha resultado una experiencia complicada. ROS2 ha presentado multitud de problemas asociados a las distintas versiones de ROS2, por lo que ha sido necesario investigar sobre varios sistemas operativos y varias versiones de ROS para obtener un resultado positivo que es el arranque del driver en ROS Rolling. Se ha conseguido conectar el driver de ROS2 a Polyscope, permitiendo realizar movimientos manuales con un controlador sencillo, pero rno se ha conseguido integrar en los planificadores de Moveit!. Esto asegura que, aún habiendo conseguido algunos resultados positivos, se necesita un mejor desarrollo y más tiempo para que los drivers y los controladores funciones de una forma más controlada y segura.

En cuanto a la aplicación de la pinza RG2 de OnRobot para esta integración, se ha conseguido observar que muchas de pruebas realizadas en otros artículos como el artículo [15] son sobre simulaciones sin aplicar un driver real que controle al cobot junto a la pinza.

Para finalizar este proyecto, se pueden consultar los vídeos de demostración en siguiente enlace [16].

En resumen, esta investigación ha permitido profundizar sobre el sistema operativo ROS en sus distintas versiones, adquiriendo conocimientos sobre su funcionamiento y permitiendo estudiar las diferencias y similitudes entre ROS1 y ROS2 y el funcionamiento de sus sistemas de paquetes reutilizables que permiten realizar tareas complejas de forma mucho más sencilla. También ha permitido trabajar con un software no libre como es Polyscope, estudiar su funcionamiento y poderlo comparar con ROS como entorno contrario a Polyscope. Finalmente, ha ayudado a adquirir conocimientos sobre los robots articulados y las pinzas robóticas, así como los costes necesarios para trabajar con ellos y la dificultad que presentan para integrarlos en entornos más genéricos.

6. Líneas de trabajo futuras

En base a los resultados y conclusiones obtenidos con este proyecto, se proponen algunas líneas de trabajo futuras que permitan mejorar el desarrollo y adquirir nuevos conocimientos en esta área de investigación:

1. Desarrollo del driver completo de la pinza RG2 para ROS. Se propone desarrollar un driver propio que permita trabajar con la pinza RG2 desde ROS a través de tópicos específicos para ello, permitiendo agarrar objetos o incluso actuar con control de fuerza.
2. Integrar el driver de la pinza RG2 anterior junto con el driver del cobot UR3e para Polyscope. Con esto se conseguiría realizar planificación de movimientos al igual que se ha conseguido hacer durante esta investigación pero esta vez con un actuador final que permitiera trabajar sobre escenarios reales o simulados.

3. Realizar un estudio sobre los diferentes planificadores de Moveit con el driver completo dentro de ROS1 y ROS2. Esto permitiría conocer mejor los algoritmos empleados en la planificación de movimientos y poder estudiar las diferencias entre ellos, adquiriendo conocimientos sobre su uso en diferentes aplicaciones.
4. Incorporar una cámara u otros sensores que permitan al robot mapear o conocer el entorno para poder desarrollar, en el caso de que sea necesario, aplicaciones que no requieran de colaboración humana o que faciliten la detección de colisiones para mejorar la seguridad.

Referencias

1. International Federation of Robotics (IFR). *World Robotics Report: All-time high with half a million robots installed*. (2020). Disponible en: <https://ifr.org/ifr-press-releases/news/wr-report-all-time-high-with-half-a-million-robots-installed>.
2. Engelberger, J. F. *Robotics in practice: management and applications of industrial robots*. *International Journal of Robotics Research*, 3(2), 46–56 (1980).
3. Scheinman, V. *Stanford robot arm control*. *Robotics Research*, 3, 301–328 (1977).
4. *ROS - Robot Operating System*. Disponible en: <http://www.ros.org/>.
5. *ABB RobotStudio*. Disponible en: <https://new.abb.com/products/robotics/robotstudio>.
6. Riksen, S., von Rump, S., Kortmann, R., Kober, J. *Easy Programming Interfaces for Industrial Robots: A Comparative Study*. *IEEE Robotics & Automation Magazine*, 25(3), 96–105 (2018). <https://doi.org/10.1109/MRA.2018.2828994>.
7. *UR3e - Universal Robots Collaborative Robot*. Disponible en: <https://www.universal-robots.com/products/ur3-robot/>.
8. *RG2 - Universal Robots Gripper*. Disponible en: <https://www.universal-robots.com/accessories/grippers/rg2-gripper/>.
9. ROS Wiki Contributors. *ROS Wiki*. Disponible en: <http://wiki.ros.org/>.
10. ROS Wiki Contributors. *ROS Wiki - Catkin*. Disponible en: <http://wiki.ros.org/catkin>.
11. Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W. *Robot Operating System 2: Design, architecture, and uses in the wild*. *Science Robotics*, 7(66), eabm6074 (2022). <https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>.
12. *Universal Robots ROS Driver*. Disponible en: https://github.com/UniversalRobots/Universal_Robots_ROS_Driver.
13. *ROS-Industrial Universal Robots*. Disponible en: https://github.com/ros-industrial/universal_robot.
14. *ROS-Industrial Robotiq*. Disponible en: <https://github.com/ros-industrial/robotiq>.
15. Juříček, M., Parák, R. *DESIGN AND IMPLEMENTATION OF THE ROBOTIC PLATFORM FOR AN EXPERIMENTAL LABORATORY TASK*.
16. Cruz García, D. *Resultados del proyecto*. Disponible en: <https://drive.google.com/drive/folders/1R4QSa6Do-SMsGQ9CcxYNOGECKulPewpV?usp=sharing> (2023).

Benchmarking de predictores inteligentes con métodos de aprendizaje profundo

Geovanny Satama Bermeo, Pastora Vega, Mario Francisco y Óscar Aponte

Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los
Caídos s/n. 37008, Salamanca, España
{satamageovanny,pvega,mfs,idu17344}@usal.es

Resumen En esta investigación, se implementaron las redes neuronales profundas, Multi Layer Perceptron (MLP) y Long-Short Term Memory (LSTM), para optimizar el tratamiento de aguas residuales. Debido a la importancia crucial de asegurar que las EDAR cumplan con los límites legales establecidos para los contaminantes, el modelado y la predicción precisa de estos procesos es vital. Con este propósito, se seleccionaron las redes MLP y LSTM por su capacidad para procesar y aprender de los datos en los sistemas en lazo abierto como en lazo cerrado. Este enfoque permite detectar de manera eficiente las posibles violaciones en la calidad del efluente. En cuanto a los resultados obtenidos, se encontró que las predicciones hechas por la red MLP mostraron una menor tasa de error en comparación con las realizadas por la red LSTM. Sin embargo, se identificaron desafíos durante el entrenamiento de ambas redes, tales como una tendencia al sobreajuste en las iteraciones posteriores. Esto subraya la necesidad de desarrollar estrategias de regularización y optimización de hiperparámetros para mejorar el rendimiento de los modelos de redes neuronales en el tratamiento de aguas residuales.

Keywords: LSTM, MLP, MATLAB, Redes neuronales profundas

1. Introducción

El crecimiento de la población y el aumento de industrias globales en la actualidad han sido los principales agravantes de la contaminación ambiental por las aguas residuales que generan sus actividades. Esto se ha convertido en una amenaza muy grave que, acompañada del cambio climático, genera aumentos de temperatura, fenómenos meteorológicos extremos, migración por parte del hábitat de la fauna, aumento de los mares, entre otros [1]. Las aguas residuales no solamente afectan al medio ambiente, sino también a la salud de los seres humanos, provocando enfermedades de interés público como la fiebre tifoidea o el cólera [1]. Por otro lado, la contaminación del agua podría conllevar que se depositen determinados nutrientes en los cuerpos acuáticos, promoviendo el crecimiento de ciertos organismos como las algas. Estos cuerpos de agua representan una amenaza a la fauna porque se crean zonas deficientes de oxígeno [1].

Frente a esta situación, tanto las empresas públicas como privadas como la sociedad deben emprender acciones conscientes para mitigar la contaminación del agua, lo que causaría un problema ambiental. Una de estas acciones involucraría la verificación de las concentraciones de contaminantes; conocido como tratamiento de aguas residuales (EDAR), dentro de las aguas residuales para preservar el entorno y los recursos naturales donde se riegan las aguas depuradas. En el EDAR se destinan estaciones depuradoras, para obtener un efluente de aguas residuales en el cual el ciclo de devolución del agua no tenga mayor afectación hacia el medio ambiente [1]. Sin embargo, este proceso requiere de la reducción máxima en costos operativos y la producción de lodos para ser viable económicamente [1].

El proceso de EDAR se estructura en varias etapas y procesos de tratamiento que se vinculan entre sí. Se empieza por la etapa del pretratamiento, que consiste en la eliminación de residuos de tamaño grande del agua residual bruta. Se continúa con la etapa de tratamiento primario, donde se eliminan los sólidos considerados de menor tamaño, como son los sólidos sedimentables y la materia que flota en el efluente. Dentro de la etapa del tratamiento secundario, se efectúan ciertos procesos biológicos que eliminan sólidos refractarios, es decir, el lodo que no pudo ser eliminado dentro de la etapa anterior [1]. Es preciso mencionar que dentro del tratamiento secundario es recomendable llevar a cabo un procesamiento de separación en la digestión de los lodos, reduciendo el volumen de residuos sólidos y estabilizando los materiales orgánicos. Esto último posibilita una mejor manipulación dentro del proceso y, por ende, se reducen los costos de almacenamiento y bombeo. Generalmente, se utiliza la degradación microbiana como proceso habitual para esta finalidad. Cada tratamiento se clasifica como tratamiento anóxico, aeróbico o anaeróbico [1]. Finalizadas estas etapas, se puede liberar el agua al medio ambiente ya que, como última etapa de tratamiento terciario, se encuentra dentro del efluente secundario, donde se elimina el nitrógeno y el fósforo, dando cumplimiento a la norma requerida [1].

El correcto control de los sistemas de agua se ha convertido en un reto, debido a la presencia de una gran cantidad de variables interrelacionadas y su alto grado de no linealidad. Además, la falta de datos disponibles y que estos sean precisos han dificultado la creación de controladores efectivos de aguas residuales. Por ello, la creación de controladores inteligentes, basados en aprendizaje profundo, ha llegado a ser una de las promesas que puede optimizar el control de los sistemas acuáticos. El surgimiento de controladores inteligentes forma parte de la ola de inteligencia artificial, cuyo objetivo principal es imitar varias de las funciones cognitivas de los seres humanos a través de modelos computacionales. Dentro de los procesos de control la inteligencia artificial ha aportado con varias técnicas y métodos que hoy en día son de gran utilidad para esta finalidad, donde las denominadas redes neuronales artificiales (RNA) constituyen uno de los métodos propuestos por la inteligencia artificial.

Las RNA han generado un gran potencial para la realización de modelos matemáticos precisos de procedimientos que son de mayor complejidad mejorando así el rendimiento de sus funciones [1]. En esta investigación, se consideran RNA

para modelar el comportamiento de un sistema de aguas residuales. El modelo propuesto considera los datos del influente y el efluente de una planta [1].

Dentro de los últimos años, los horizontes se han ampliado en torno a la aplicación del aprendizaje profundo para mejorar el control de aguas residuales. Particularmente, las redes neuronales artificiales profundas han demostrado ser herramientas con capacidad de modelar sistemas de mayor complejidad y que no son lineales, basados en la utilización de datos antiguos y actuales. Estas herramientas permiten obtener patrones en los que existen relaciones no lineales en los datos, facilitando la realización de predicciones para el control de procesos complejos dentro del tratamiento de aguas residuales.

Por ello, la finalidad de esta investigación de master es diseñar y poner a prueba las arquitecturas de redes perceptrón multicapa y larga memoria de corto plazo.

2. Redes neuronales artificiales

El elemento procesador es la unidad similar a la neurona del ser humano dentro de las redes neuronales artificiales [15], las entradas de un elemento procesador se pueden combinar mediante una suma básica, dicha suma puede ser modificada por una función de traslado, este valor resultante se dirige hacia la salida del elemento procesador [15].

La neurona es considerada como la unidad fundamental para el sistema nervioso y el correcto funcionamiento del cerebro. La neurona es un procesador que recibe y realiza una combinación de actividades desde y hacia otras neuronas [9]. Por lo tanto la entrada debe ser fuerte dentro de su combinación, para que la salida de la misma se active. A continuación, en la figura 1, mostraremos las partes que constituyen a una neurona.

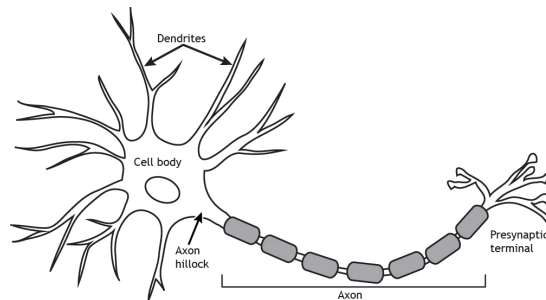


Figura 1: Elementos de una Neurona [9].

El cerebro humano se compone por billones de neuronas que se encuentran conectadas. Las Dendritas son las entradas de otras neuronas que tienen uniones llamadas sinapsis, las cuales están ramificadas hacia el axón, es decir la salida

de la neurona. [15]. Es importante saber, la eficacia que debe tener la sinapsis al ser moldeable dentro de la duración del estudio de la red.

Las salidas del elemento procesador pueden interconectarse hacia las entradas de algunos tipos de neuronas artificiales, mediante las uniones ponderadas, que corresponden al funcionamiento eficaz de la sinapsis neuronal [15]. La Figura 2 demuestra como se ve un PE de una red neuronal artificial implementada desde un ordenador.

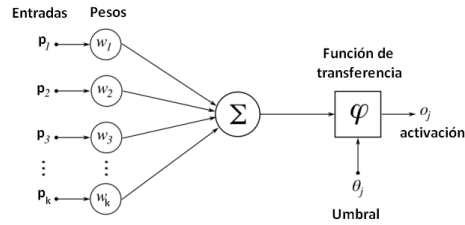


Figura 2: Esquema de una Neurona Artificial [9].

La red neuronal está compuesta por una serie de procesadores elementales que se encuentran conectados de manera concreta. Además del elemento procesador, la forma en cómo se conectan las RNA tiene gran relevancia [11]. Los PE están estructurados en conjunto, se los denomina niveles o capas. Una red neuronal tradicional contiene en un grupo de capas con uniones entre capas anexadas.

Hay un par niveles que están conectados al entorno externo. Un nivel de entrada, que funciona como un almacenamiento temporal para los datos que se introducen en la red, y otro nivel de salida, que guarda una respuesta de la red ante una entrada determinada. Los demás niveles son intermedios y son conocidos como capas ocultas [11]. La apariencia de una Red Neuronal Artificial se observa en la Figura 3.

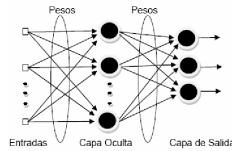


Figura 3: Estructura de una Red Neuronal Simple [9].

2.1 Red perceptrón multicapa (MLP)

La habilidad de procesamiento y rendimiento de las redes neuronales artificiales se debe a las interconexiones entre las neuronas artificiales que forman

la estructura de las RNA. La configuración más elemental consiste en un conjunto de neuronas ordenadas por una única capa, la cual se puede apreciar en la figura 4, [11]. Los uniones circulares no participan como componentes de la capa, sino que simplemente distribuyen las entradas recibidas. Mientras que una red formada de múltiples capas como se observa en la figura 5 tendrá la capacidad de resolver problemas que no son linealmente separables ya que esa sería su principal limitación.

2.2 Estructura de la red perceptrón multicapa

Las neuronas artificiales se conectan a la red neuronal por medio de un peso específico o entrada. Dentro de la práctica, existen uniones a través de las salidas o de las entradas de las RNA en una sola capa. Normalmente es claro que las redes con gran tamaño y complejidad generan un mayor rendimiento para establecer un cálculo computarizado en comparación con las redes básicas.[10].

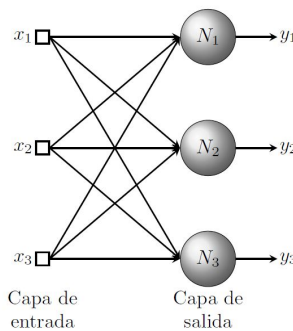


Figura 4: Estructura de una red neuronal de una capa [9].

A pesar de los cambios existentes en las redes construidas pueden presentar detalles con ciertas diferencias, por otro lado tienen similitud en la distribución de las neuronas en capas, como se aprecia en la figura 5, reproduciendo la estructura de capas localizadas en el cerebro humano [10]. Las redes neuronales artificiales multicapa están constituidas por algunas capas sencillas interconectadas en forma de cascada las cuales tienen ventajas y características superiores a las RNA de una sola capa.

Es importante resaltar que la presencia de una labor de estimulación no lineal entre varias capas, mejora con mayor efectividad el funcionamiento de las RNA multicapa.

2.3 Proceso de aprendizaje de la red perceptrón multicapa

Las redes neuronales artificiales están compuestas por un conjunto de neuronas que tienen conexión entre ellas, la relación de las conexiones neuronales

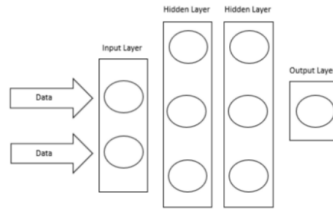


Figura 5: Red neuronal de dos Capas[9].

son valoradas basándose en su peso, lo que se relaciona por el múltiplo del valor de entrada. Dentro del proceso general se empieza por la activación neuronal, como la función sigmoïdal, tangente hiperbólica, entre otras, para determinar su salida. La función de activación es esencial ya que introduce la no linealidad durante la preparación de la red, lo cual es importante para el modelado efectivo de las redes neuronales artificiales.

Durante la preparación de una red neuronal como en la figura 6, se lleva a cabo un ciclo iterativo en el que se traspasan capa por capa de cada RNA. Durante este ciclo, satisfaciendo de esta manera las necesidades del modelo mediante el ajuste de los valores basado en los pesos y sesgos. El proceso de propagación hacia adelante, también conocido como forward-propagation, ocurre cuando la información de la entrada es transferida por medio de la red.

En este proceso, cada neurona se adapta y aprende de los datos obtenidos por cada capa anterior, luego envía esta información a la siguiente capa. Después de que toda la información haya pasado por todas capas y a su vez las neuronas siempre y cuando hayan realizado sus cálculos, llegando así a la última capa, donde se puede determinar cada entrada mediante un pronóstico de etiqueta.

Las pérdidas son utilizadas para calcular el error, por lo tanto se puede realizar una comparación cuatitativa verificando un resultado correcto, buscando como valor final igual a 0, luego, mediante una preparación del modelo se realiza un ajuste automático necesarios para las RNA y por último obtener el mejor pronóstico posible.

Una vez que se ha calculado el error en la capa de salida de la red neuronal, este error se propaga hacia atrás, retro-propagándose (back-propagation), a lo largo de la red. Dicho error es transportado desde la salida hacia las neuronas que contienen capa oculta y están interconectadas con la salida. Las capas que se encuentran ocultas obtienen una parte del valor erróneo, tomando en cuenta su aportación a la salida final. El proceso se reitera en varias ocasiones de capa en capa, asegurándose de que las neuronas reciban el fragmento de error equivalente a su participación del error total. Es decir, las capas ocultas aprenden a través de la retro-propagación del error para ajustar sus pesos y mejorar la precisión y capacidad de predicción de la red neuronal.

Luego que el error se ha propagado a través de la red neuronal, es el momento de adaptar el volumen de los valores resultantes de las conexiones con las neuronas. Para obtener un error aceptable debe ser cercano a 0, por lo que

se aplica un modelo matemático conocido como decadencia por gradiente. Este modelo matemático realiza pequeños ajustes en los pesos mediante un cálculo de la derivada en función de la pérdida. Esto va a indicar la dirección en la cual se debe avanzar para descender hacia el mínimo global de la función de pérdida. El procedimiento de la decadencia por gradiente se repite en varias ocasiones, conocidas como etapas, y se realiza utilizando bases de datos. Esto implica que se toma un conjunto de datos, se calcula el error, se ajustan los pesos y se realiza nuevamente el proceso hasta que se alcanza un nivel de convergencia satisfactorio.

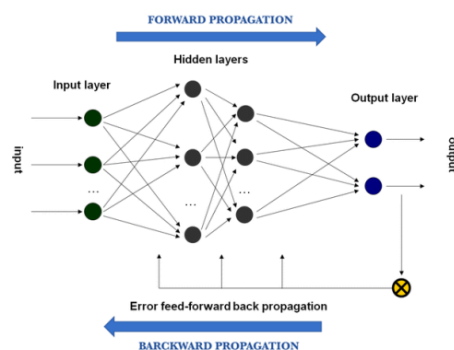


Figura 6: Procedimiento de entrenamiento de una ARN [12].

2.4 Long-short term memory (LSTM)

El modelo de memoria utilizados para corto y largo plazo esta basado ampliamente en las redes neuronales recurrentes, que por lo general, toman sus recuerdos para aprender de experiencias importantes que han sucedido hace mucho tiempo. Las LSTM permiten a la red neuronal reconocer sus entradas.

2.5 Estructura de la red LSTM

El modelo LSTM utiliza una estructura de memoria en lugar de unidades neuronales simples. Se compone de una celda de memoria, una puerta de entrada y una puerta de salida. Las puertas controlan el flujo de información, protegiendo a la celda de memoria de perturbaciones por datos de entrada irrelevantes y evitando que las perturbaciones se propaguen a otras neuronas.

En la figura 7 se presenta la configuración de una LSTM, en la cual los símbolos "x" y "+" representan multiplicaciones y sumas de elementos, respectivamente. Los bloques están representados por capas, mientras que "tanh" indica una función de tangente abultada. Además, "x" representa la entrada, "h" denota la salida y el estado en el que se encuentra [14].

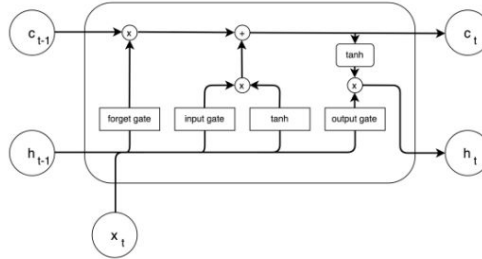


Figura 7: Estructura estándar de una red LSTM [14].

El diseño de las redes LSTM pueden recordar eventos a lo largo de períodos de tiempo arbitrarios, superando así las limitaciones de las redes neuronales recurrentes convencionales [14].

2.6 Proceso de aprendizaje de la red LSTM

Las aplicaciones que tienen las redes LSTM se utilizan comunmente para el manejo de robots, por su capacidad de predicción del tiempo, reconocimiento de voz etc [14]. Otra de las cualidades de este tipo de redes, es la habilidad de recordar valores en tiempos prolongados, lo que determina varios atributos de secuencia de entrada.

Para la función de activación del estado del modelo LSTM, su derivada será 1, por lo que al momento de la repropagación no tendrá daño alguno, y su salida tiene que ser en un rango de 0 a 1.

La ecuación de una red LSTM estándar es la que se presenta a continuación:

$$y = s\left(\sum_{i=1}^N w_i x_i\right) \quad (1)$$

En dicha ecuación s , representa la activación de la logística de la red.

El algoritmo de retropropagación a través del tiempo (BPPT) es el adecuado para el entrenamiento del modelo LSTM. La fuga de los gradientes está basada por el volumen de errores que se produzcan. El modelo recuerda sus errores mediante su función su propagación hacia atrás, reparando así cada error en cada peso, por lo que considera que la repropagación es muy efectiva para acondicionar de forma correcta una red neuronal artificial.

2.7 Funciones de activación neuronal

Las funciones de activación neuronal desempeñan un papel fundamental al propagar hacia adelante la salida de una neurona hacia las neuronas de la capa siguiente a las que está conectada. Estas funciones son esenciales para no implementar la continuidad en el proceso de preparación del modelo de las redes

neuronales artificiales, permitiendo que la red aprenda y represente relaciones complejas entre los datos. Existen varias funciones que activan las redes neuronales que se describen a continuación.

2.7.1 Activación unidad lineal rectificada (ReLU) La tarea de activación de unidad rectificada uniforme (RELU) es considerada como una herramienta de aprendizaje profundo encargada de la activación no lineal, Krizhevsky logró popularizar la función ReLU, ya que ha sido una de las funciones con mayor utilización debido a su eficacia y simplicidad [13].

La función unidad rectificada uniforme se basa en la siguiente ecuación: $f(x) = \max(0, x)$, donde x representa la neurona y la salida se representa en base a la cantidad máxima que existe entre 0 y x , es por ello que ReLU se basa en cualquier entrada x , sin importar si x es positivo o negativo siempre se devolverá 0 como se aprecia en la figura 8.

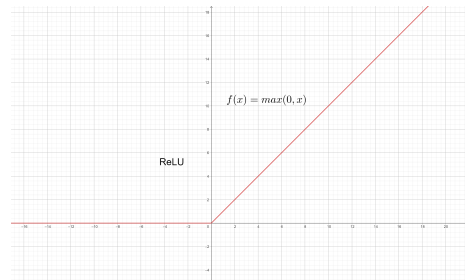


Figura 8: Función de activación ReLU.

Las ventajas que la convierten en una herramienta esencial dentro del aprendizaje profundo de las redes neuronales es, la facilidad de cálculo y el no necesitar de mucho poder para su procesamiento, tiene menos probabilidad con el problema de desvanecimiento de la gradiente, que es producida cuando la derivación de la función de activación es muy pequeña, dificultando la actualización los pesos ReLU, finalmente es utilizada dentro de las redes neuronales profundas, ya que no tiene saturación dentro de la activación lo que ocurre con otras funciones.

2.7.2 Activación sigmoide La tarea de activación sigmoide es útil para reducir la influencia de valores muy altos o diferentes, sin suprimirlos por completo. Al aplicar una función sigmoide, las variables independientes que tienen un rango casi infinito se convierten en probabilidades simples que van desde 0 hasta 1. Con esta función, los valores tienden a acercarse en su mayoría a la salida, por lo tanto estará muy cerca de los extremos de 0 o 1. La activación sigmoide produce valores de salida que se encuentran en términos ya sean altos o bajos dentro del sigmoide. Por lo tanto, cuando la inclinación de la función es alta,

es similar a una función escalón. No obstante, la activación sigmoideal es valiosa debido a que su derivación tiene resultado positivo y se acerca a cero para cantidades grandes ya sean positivas o negativas, o cuando su valor más alto resulta ser igual a 0. Esto facilita utilizar parámetros de aprendizaje diseñados para funciones de gradiente, pero con la ventaja de la derivación que esta basada en la función sigmoideal, la cual se encuentra establecida en todo el intervalo. Esta característica facilita la optimización de aprendizaje de la red neuronal, ya que la derivación es necesaria para calcular los gradientes y ajustar los pesos de manera adecuada.

2.8 Elementos del algoritmo de propagación hacia atrás

El backpropagation es un método que se utiliza para realizar cambios en cuanto al peso o al sesgo de una red neuronal en la dirección correspondiente. Empieza con el cálculo del valor de pérdida y luego encaja los factores s de la red neuronal en orden contrario, utilizando una ecuación matemática de optimización que toma en cuenta el valor erróneo. En la herramienta de software, Matlab, existen tres ecuaciones de optimización diferentes disponibles: Declinación de gradiente aleatoria con momento (SGMD), RMSProp y Evaluación adaptativa del momento (Adam). , Para algunas investigaciones se utiliza el algoritmo de optimización Adam, que es el más común para la preparación de redes neuronales.

2.9 Parametrización de modelos

Los modelos existentes pueden mejorar los procesos de un sistema, por lo tanto para llevar a cabo un correcto funcionamiento, se requiere tener el conocimiento y tener las habilidades necesaria para la práctica, tomando en cuenta los ajustes de las variables adecuadas. Sin embargo, estas mejoras en la precisión pueden aumentar el tiempo necesario para el procedimiento de aprendizaje. Es por ello que encontrar la mejor configuración de parámetros y tareas de activación exigen habilidad y destrezas debido a la cantidad de medios disponibles.

Algunos de los parámetros más utilizados se describen a continuación.

2.9.1 Número de épocas El parámetro "epochs" representa la cantidad de veces que todos los datos de entrenamiento son procesados por las redes neuronales a lo largo de su preparación.

2.9.2 Tamaño del lote Como se mencionó anteriormente, es posible agrupar la información de preparación en bloques más pequeños, conocidos como mini-lotes (mini-batch), para alimentar la red neuronal. El tamaño del lote, denominado batch-size, determina la cantidad de muestras que se utilizarán en cada repetición de la preparación para renovar el gradiente. El tamaño adecuado del bloque depende de varios parámetros, incluyendo la técnica utilizada para realizar los cálculos, denominada memoria GPU.

2.9.3 Tasa de aprendizaje El portador de gradiente tiene un sentido y un volumen. Las ecuaciones para la declinación de la gradiente es producto entre el tamaño de la gradiente por una escala, a esta ecuación se la denomina como tasa de aprendizaje (α).

$$\theta_{N+1} = \theta_n - \alpha \cdot \nabla E(\theta) \quad (2)$$

La cantidad adecuada para este factor tiene que ver con el modelo en cuestión. Generalmente, si el tamaño del lote es demasiado grande, se estarán dando movimientos de gran magnitud durante el proceso de aprendizaje. Aunque esto podría acelerar el entrenamiento, también existe el riesgo de no alcanzar el objetivo esperado de pérdidas, lo que dificultaría el procedimiento de aprendizaje podría afectar la necesidad del resultado esperado.

Por otro lado, si el learning rate es bajo, los avances serán constantes y pequeños, lo que permitirá una mejor aproximación al mínimo local. Sin embargo, esto también significa que el procedimiento de aprendizaje será más pausado. Normalmente, el learning rate óptimo es aquel que tiene un avance más lento cuando el modelo comienza a acercarse a una solución, como es el caso de la aplicación del modelo en Matlab donde se puede utilizar el parámetro "Learn-RateSchedule" con el valor "piecewise" para disminuir el learning rate a medida que avanzan los epochs [12]. Esto facilita un proceso de enseñanza veloz en las etapas iniciales mediante la utilización de tasas de aprendizaje más elevadas, conforme se avanza en el entrenamiento, se aplican ajustes cada vez más sutiles, promoviendo la convergencia hacia el valor mínimo de la función de pérdida durante el período de entrenamiento.

2.9.4 Momento Cuando se utiliza el algoritmo de descenso de gradiente estocástico, puede ocurrir cierta oscilación durante el proceso de caído hacia el mínimo local óptimo. Sin embargo, se ha encontrado que al agregar un término de impulso a la actualización de los parámetros, esta oscilación puede reducirse. Se puede definir el descenso de gradiente estocástico con actualización de momentum de la siguiente manera:

$$\theta_{n+1} = \theta_n - \alpha \cdot \nabla E(\theta) + \gamma \cdot (\theta_n - \theta_{n-1}) \quad (3)$$

El momento, que es cómo la iteración actual se ve afectada por el paso del gradiente anterior, se representa con el símbolo ∇ .

3. Caso de Estudio

Recurrir a una fuente de información y analizarla es un parte primordial de un proyecto, ya que con esto se puede tomar decisiones importantes sobre un proceso en específico. En esta sección del proyecto se presenta una síntesis general de los escenarios de control sobre los cuales se simuló la planta y las variables asociadas. Bajo estas condiciones fue posible obtener un volumen importante de datos, que posteriormente fueron almacenados en un dataset y finalmente

analizados mediante distintos métodos estadísticos, entre los cuales figuran: la t-Student, coeficiente de correlación de Pearson y coeficiente de determinación R^2 ; todos con el fin de determinar, entre otros, si las variables del dataset presentan algún tipo de relación entre ellas.

3.1 Obtención de datos de la plataforma de simulación BSM1

Con el objetivo de realizar una comparación del funcionamiento de la planta de tratamiento de aguas residuales (Waster Water Treatment Plan – WWTP) se propusieron dos escenarios de control: el primero, en donde la planta funciona a lazo abierto (es decir, no existe control sobre la misma) y el segundo, en donde se usa un modelo de control por defecto del BSM1 basado en PID's. Ambos escenarios fueron simulados a través del modelo estándar Benchmark N°1-BSM1 el cual basa su funcionamiento en el modelo de lodos activados No.1 (ASM1) desarrollado por la Asociación Internacional para la Investigación y el Control de la Contaminación del Agua [3].

La planta es un sistema multivariable (ver figura 12) con gran interacción, las variables consideradas en el entrenamiento de las redes se muestran en la tabla 1.

Tabla 1: Variables tomadas de la simulación de la planta

Descripción	Variable	Fuente
S_{NH}	Concentración de nitrógeno en forma de $NH_4^+ + NH_3$ [gN/m ³]	[2]
S_{NO}	Concentración de nitrógeno en forma de nitratos y nitritos [gN/m ³]	[2]
S_{ND}	Concentración de nitrógeno orgánico soluble biodegradable [gN/m ³]	[2]
Q	Tasa de Flujo [m ³ /d]	[2]
$S_{NH} \cdot Q$	Concentración de nitrógeno en forma de $NH_4^+ + NH_3$ por Tasa de Flujo [gN/d]	[2]
$S_{NO} \cdot Q$	Concentración de nitrógeno en forma de nitratos y nitritos por Tasa de Flujo [gN/d]	[2]
$S_{ND} \cdot Q$	Concentración de nitrógeno orgánico soluble biodegradable por Tasa de Flujo [gN/d]	[2]

Para cada escenario de control se tomó en cuenta tres tipos de influente que representan tres condiciones meteorológicas: clima seco o DRY, clima lluvioso o RAIN (combinación de clima seco y un periodo largo de lluvia) y clima tormentoso o STORM (combinación de clima seco y tormentas). Cada escenario se simuló por 14 días con un tiempo de muestreo de 15 minutos.

Puesto que la planta de tratamiento cuenta con varios componentes, con el objetivo de realizar un análisis más detallado de la misma, para cada escenario

de control y a para cada condición meteorológica se midieron las variables de la Tabla 1 en tres puntos de referencia, tal como se detalla a continuación:

- **Punto 1:** Influyente (ingreso del agua contaminada a la planta).
- **Punto 2:** Ingreso del tanque 1 (reactor biológico anóxico), en donde se lleva a cabo el proceso de desnitrificación.
- **Punto 3:** Salida del tanque 5 (reactor biológico aerobio), en donde se lleva a cabo el proceso de nitrificación.

Finalmente, con las consideraciones mencionadas, de cada simulación se extrajeron los valores de las variables de la Tabla 1 para formar una base de datos, y posteriormente, realizar un análisis estadístico.

3.2 Métodos estadísticos propuestos

De acuerdo a la base de datos obtenida, se llevaron a cabo tres tipos de análisis estadísticos: coeficiente de correlación de Pearson, coeficiente de determinación R^2 y t-Student.

El objetivo de estos análisis es determinar si los componentes químicos residuales en el líquido efluente tienen algún tipo de relación con los químicos residuales medidos en cada uno de los tres puntos definidos para la planta. Los resultados obtenidos se muestran y explican a continuación.

3.3 Análisis de los resultados usando Coeficiente de correlación de Pearson

La correlación de Pearson r_{xy} es una herramienta estadística que generalmente se usa como prueba de hipótesis y sirve para determinar si existe o no relación lineal entre dos variables que se están comparando. El coeficiente de correlación de Pearson puede tomar valores entre -1 y 1 ($-1 \leq r_{xy} \leq 1$), en donde, si el valor de r_{xy} se acerca a 1 entonces las variables tendrán una relación directa (es decir si aumenta la una, la otra también lo hace); si por el contrario, el valor de r_{xy} es cercano a -1 entonces la relación es inversa. En caso de que el valor de r_{xy} se acerque a cero, entonces las variables no estarán relacionadas linealmente [4].

Para nuestro caso de estudio, se determinó el coeficiente de correlación de Pearson de cada una de las variables de la Tabla 1 respecto la variable S_{NH} del efluente para cada condición meteorológica (Dry, Rain y Storm) y se ordenaron tal como se detalla en la Tabla 2.

A continuación se analizan los resultados determinados:

3.3.1 Correlación de Pearson – Punto 1(Influente) Al observar la Fig. 9, se puede inferir que en general, para ambos escenarios de control, el coeficiente de correlación de Pearson r_{xy} presenta valores muy cercanos a cero, lo que indica que las variables relacionadas presentan una correlación lineal muy débil o nula. Es decir, que los componentes químicos medidos en el punto influente de la

Tabla 2: Detalle de cálculo para coeficiente de correlación de Pearson

<i>DRY</i>	S_{NH}	S_{NO}	S_{ND}	Q	$S_{NH} \cdot Q$	$S_{NO} \cdot Q$	$S_{ND} \cdot Q$
	*	*	*	*	*	*	*
<i>RAIN</i>	S_{NH}	S_{NO}	S_{ND}	Q	$S_{NH} \cdot Q$	$S_{NO} \cdot Q$	$S_{ND} \cdot Q$
	*	*	*	*	*	*	*
<i>STORM</i>	S_{NH}	S_{NO}	S_{ND}	Q	$S_{NH} \cdot Q$	$S_{NO} \cdot Q$	$S_{ND} \cdot Q$
	*	*	*	*	*	*	*
	S_{NHE}	S_{NHE}	S_{NHE}	S_{NHE}	S_{NHE}	S_{NHE}	S_{NHE}
	S_{NH}	S_{NO}	S_{ND}	Q	$S_{NH} \cdot Q$	$S_{NO} \cdot Q$	$S_{ND} \cdot Q$
	S_{NH}	S_{NO}	S_{ND}	Q	$S_{NH} \cdot Q$	$S_{NO} \cdot Q$	$S_{ND} \cdot Q$

planta no se relacionan con el S_{NH} del efluente. El elemento que parece destacar es la concentración de nitrógeno soluble (S_{ND}) del punto influente, ya que al observar la Fig. 9 en la estación Dry de lazo cerrado, el SND alcanza un valor máximo de 0.35 indicando una relación moderada con el S_{NH} del efluente.

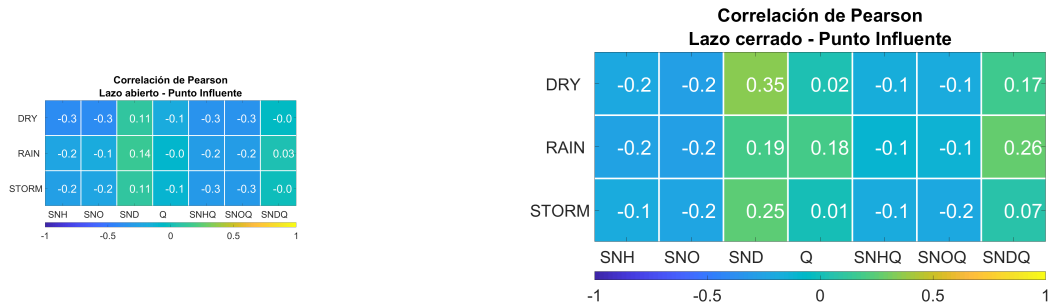


Figura 9: Coeficiente de correlación de Pearson – Punto 1(Influente).

3.3.2 Correlación de Pearson – Punto 2 (Tanque 1) Para el Punto 2 (Tanque 1), la Fig. 10 Coeficiente de correlación de Pearson – Punto 2 (Tanque 1 Fig. 10 muestra que para ambos escenarios de control (lazo abierto y control MPC) el coeficiente de correlación de Pearson presenta un nivel de asociación de variables moderado. Un ejemplo es la cantidad de nitrógeno soluble biodegradable $S_{ND} \cdot Q$ del escenario de lazo cerrado (para la estación meteorológica Rain), el cual presenta una relación lineal directa (de 0.57) con el SND del efluente.

3.3.3 Correlación de Pearson – Punto 3 (Tanque 5) En el Punto 3 (Tanque 5) de la planta se observa que, en ambos escenarios, todas las variables tienen una relación directa la concentración de nitrógeno del efluente (Ver Fig.

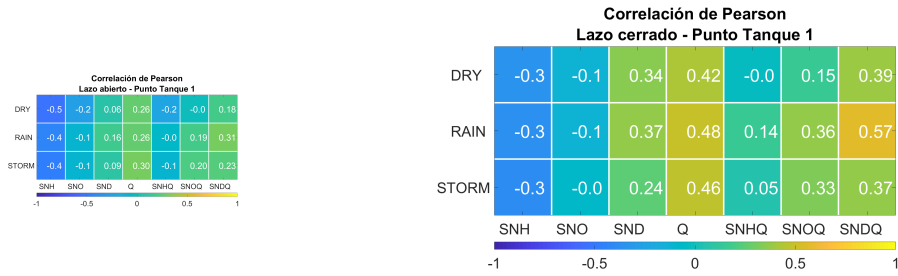


Figura 10: Coeficiente de correlación de Pearson – Punto 2 (Tanque 1).

11). En este caso el nivel de correlación lineal es directo y muy fuerte (cercano a 1). Tal es el caso para la concentración de nitrógeno orgánico biodegradable (S_{ND}) para la estación Rain en el escenario de lazo cerrado (Fig. 11 - derecha) que presenta un valor de 0.78. Estos resultados son lógicos puesto que el punto 3 (tanque 5) es el punto que está más cerca del efluente de la planta, por lo que se podría intuir que la cantidad de nitrógeno orgánico SND medido en este punto tendrá un valor similar al del punto emergente de la planta.

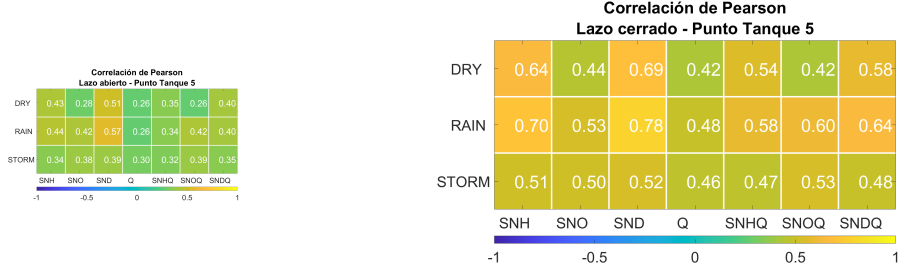


Figura 11: Coeficiente de correlación de Pearson – Punto 3 (Tanque 5).

Como comentario general, se puede mencionar que para los tres análisis estadísticos presentados hasta ahora (análisis para los puntos influentes, tanque 1 y tanque 5), la concentración de nitrógeno orgánico soluble biodegradable (S_{ND}) es la variable que tiene la relación más fuerte con el S_{NH} del efluente. Esto puede explicarse al observar la Fig. 12; en esta figura se presentan los procesos biológicos que caracterizan al modelo ASM1 y las variables asociadas cada uno. Como se observa, las variables S_{ND} y S_{NH} tienen una estrecha relación a través del proceso biológico de amonificación, en el cual, básicamente, el nitrógeno orgánico soluble se transforma en amonio NH_4^+ [5].

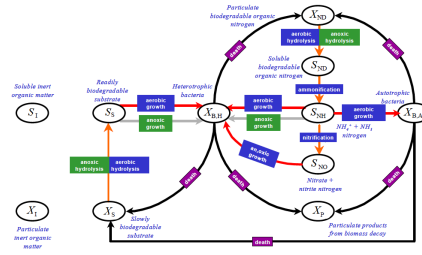


Figura 12: Procesos biológicos reflejados en el ASM1 [6].

3.4 Análisis de los resultados usando Coeficiente de determinación R^2

A diferencia del coeficiente de correlación de Pearson que mide la relación de linealidad de dos variables, el coeficiente de determinación R^2 tiene un objetivo diferente, y es medir lo bien que un modelo se ajusta a los datos reales observados. El coeficiente de determinación R^2 es una medida acotada, siendo sus límites $0 < R^2 < 1$. Un R^2 de uno (1) significa un ajuste perfecto, y un valor cero (0) indica la no representatividad del modelo, lo que significa que el modelo usado no explica nada de la variación de los datos reales observados [7].

Para esta parte de nuestro caso de estudio se consideró el coeficiente de determinación R^2 de cada una de las variables de la Tabla 1 respecto la concentración de nitrógeno en forma de $NH_4^+ + NH_3$ del efluente para cada condición climática (Dry, Rain y Storm) y se ordenaron tal como se detalla en la Tabla 3.

Tabla 3: Detalle de cálculo para coeficiente de determinación R^2 .

DRY	$R^2(S_{NH} \cdot S_{NHE})$	$R^2(S_{NO} \cdot S_{NHE})$	$R^2(S_{ND} \cdot S_{NHE})$	$R^2(Q \cdot S_{NHE})$
RAIN	$R^2(S_{NH} \cdot S_{NHE})$	$R^2(S_{NO} \cdot S_{NHE})$	$R^2(S_{ND} \cdot S_{NHE})$	$R^2(Q \cdot S_{NHE})$
STORM	$R^2(S_{NH} \cdot S_{NHE})$	$R^2(S_{NO} \cdot S_{NHE})$	$R^2(S_{ND} \cdot S_{NHE})$	$R^2(Q \cdot S_{NHE})$
	S_{NH}	S_{NO}	S_{ND}	Q

Los resultados y análisis de este cálculo estadístico se presentan a continuación:

3.4.1 Coeficiente de determinación R^2 – Punto 1(Influente) Como se puede observar en la Fig. 13, para el punto 1 (en ambos escenarios de control), ninguna de las variables (S_{NH} , S_{NO} , S_{ND} y Q) se ajusta a la tendencia dada por la S_{NH} del efluente. Los valores determinados para el coeficiente de determinación R^2 es casi nulo en todos los casos.

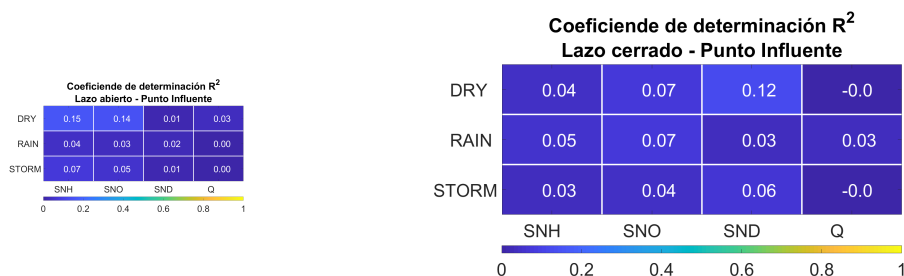


Figura 13: Coeficiente de determinación R^2 – Punto 1(Influyente).

3.4.2 Coeficiente de determinación R^2 – Punto 2 (Tanque 1) En el caso del punto 2 (tanque 1), se aprecia que en la Fig. 14, al igual que en el punto 1, el coeficiente de determinación presenta valores cercanos a cero para ambos escenarios de control (lazo abierto y lazo cerrado). El máximo valor encontrado es el 0.28 para la concentración de nitrógeno (S_{NH}) respecto al S_{NH} del efluente. Aún con este hecho, se puede afirmar que, para el punto 2, las variables S_{NH} , S_{NO} , S_{ND} y Q no se ajustan al modelo dado por el S_{NH} del efluente.

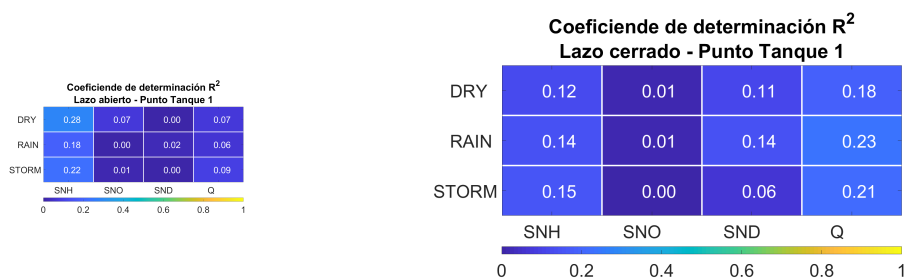


Figura 14: Coeficiente de determinación R^2 – Punto 2 (Tanque 1).

3.4.3 Coeficiente de determinación R^2 – Punto 3 (Tanque 5) Tal como se observa en la Fig. 15, el coeficiente de determinación para el escenario de lazo abierto demuestra que las variables S_{NH} , S_{NO} , S_{ND} y Q no se ajustan al comportamiento de la concentración de nitrógeno S_{NH} del efluente. Para el escenario de lazo cerrado (Fig. 15-derecha) la única variable que presenta un nivel de ajuste medio es la concentración de nitrógeno orgánico biodegradable (S_{ND}) con un valor 0.61 (61 %) para la estación Rain.

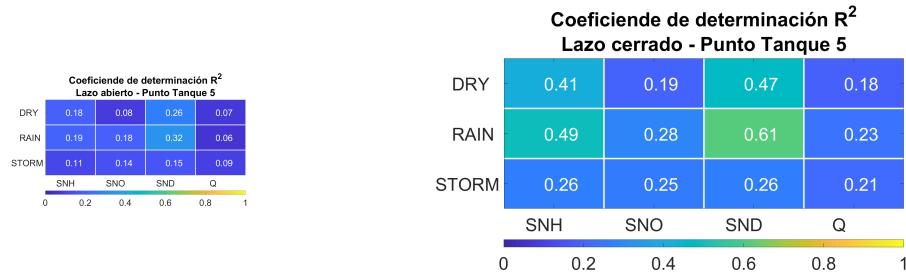


Figura 15: Coeficiente de determinación R^2 – Punto 3 (Tanque 5).

3.5 Análisis de los resultados usando la t-Student

La contribución de la prueba t-Student, específicamente, es para comparar dos grupos de datos o muestras. Al ser una prueba de hipótesis, la hipótesis nula establece que no hay diferencia entre las medias de los grupos de datos, mientras que la hipótesis alternativa establece que, de existir una diferencia, esta solo se debe al azar [8].

Para nuestro caso de estudio, se realizó la comparación de las variables S_{NH} , Q y S_{NO} con el S_{NH} del efluente para los tres puntos elegidos en la planta (en cada condición meteorológica y en escenario de lazo cerrado) usando la prueba t-Student. Estos resultados se muestran en la Tabla 4, Tabla 5 y Tabla 6.

Tabla 4: Resultados prueba t-Student en lazo cerrado para el punto 1 (Influente).

h del Punto 1 – Influyente			
DRY	1	1	1
RAIN	1	1	1
STORM	1	1	1
	S_{NH}	Q	S_{NO}

Tabla 5: Resultados prueba t-Student en lazo cerrado para el punto 2 (tanque 1).

h del Punto 2 – Influyente			
DRY	1	1	1
RAIN	1	1	1
STORM	1	1	1
	S_{NH}	Q	S_{NO}

Los valores observados en las columnas de las tres tablas son el resultado de la prueba de hipótesis t-Student (representado con la variable h). Si el valor de h es 1 indica que la prueba t-Student rechaza la hipótesis nula a un nivel de significación del 5%. Es decir, para la planta en estudio, en todos los casos, la concentración

Tabla 6: Resultados prueba t-Student en lazo cerrado para el punto 3 (Tanque 5).

h del Punto 3 – Influyente			
DRY	1	1	1
RAIN	1	1	1
STORM	1	1	1
	S_{NH}	Q	S_{NO}

de nitrógeno en forma de nitratos y nitritos (S_{NO}), la concentración de nitrógeno en forma de $NH_4^+ + NH_3$ (S_{NH}) y la concentración de nitrógeno orgánico soluble biodegradable (S_{ND}) en los puntos 1, 2 y 3, no es igual a la concentración de nitrógeno en forma de $NH_4^+ + NH_3$ (S_{NH}) medida en el efluente.

3.6 Limpieza de Datos

Del análisis realizado al dataset recolectado para este estudio, podemos observar que los valores representados en los diferentes campos muestran precisión y consistencia sin que exista algún tipo de anormalidad o datos no coherentes que afecten a este estudio. Además, se observa que el formato y la estructura expuesta no muestran errores, y que los datos obtenidos para cada variable están dentro de rangos esperados de acuerdo con el dominio del problema.

Es importante destacar que la fuente de los datos es confiable y tuvo un correcto manejo en cada simulación, por lo que se sugiere que se incorpore un control de calidad para reducir la probabilidad de error. Con esto, se puede concluir que los datos obtenidos para este estudio cumplen con los requisitos y formatos requeridos para ser aplicados en el BSM1 de simulink, es decir, no es necesario realizar una limpieza de datos adicional, por tanto, se demuestra que el dataset a utilizar dentro de esta investigación es adecuado y no requiere de modificaciones significativas.

4. Diseño de la Red Neuronal

La ciencia de datos, el machine learning y las redes de aprendizaje profundo o Deep learning en términos técnicos ha demostrado ser de gran ayuda para la toma de decisiones basadas en los datos y no en la intuición en diferentes campos como la industria, la academia y la medicina. Los datos históricos de serie cronológica desempeñan un papel importante en muchas implementaciones, lo que muestra la capacidad que tienen las redes neuronales en tratar este tipo de información.

En el marco de esta investigación de máster, se tomo la decision de explorar el uso de dos tipos principales de redes neuronales: las redes perceptrón multicapa (MLP) y las redes de gran memoria a corto plazo (LSTM). Estas dos arquitecturas han demostrado ser muy eficaces en tareas de predicción y modelado, y cada una tiene sus propias ventajas y características únicas que las hacen adecuadas para distintas clases de problemas.

La red MLP, posee una alimentación hacia adelante, es conocida por su capacidad para modelar funciones no lineales y su facilidad de uso así como de ser fácilmente escalable a grandes conjuntos de datos y ser altamente personalizable en términos del número de capas ocultas y unidades en cada capa. Mientras que la LSTM es un tipo de red recurrente que ha sido diseñada para abordar el problema de la desaparición del gradiente en el entrenamiento de las RNNs. La ventaja que la caracteriza es la capacidad para aprender dependencias a corto plazo, lo que la hace muy adecuada para tareas que involucran datos secuenciales o temporales.

Se explorarán y compararán ambas arquitecturas de redes en el contexto del control de sistemas de tratamiento de aguas residuales, con el objetivo de desarrollar y optimizar controladores inteligentes capaces de predecir y regular eficazmente el comportamiento de estos sistemas. A continuación, se detalla el diseño y el procedimiento de entrenamiento de cada uno de estos modelos.

4.1 Multi Layer Perceptron (MLP)

4.1.2 Red para datos obtenidos del sistema de control en lazo abierto

4.1.2 Diseño y optimización de la red MLP Inicia con una capa de entrada de secuencia que recibe los datos normalizados de entrada. Luego, tiene cuatro capas ocultas, cada una con 160 unidades ocultas. Después de cada capa oculta, se aplica una capa de activación ReLU para introducir no linealidades y una capa de dropout con una tasa de 0.2 para evitar el sobreajuste. La red concluye con una capa completamente conectada que se conecta a una capa de regresión, que se utiliza para predecir el valor de salida, como se puede observar en la figura 16.

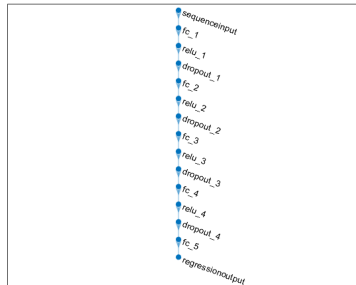


Figura 16: Estructura de la red neuronal MLP.

La optimización de esta red se lleva a cabo mediante el optimizador Adam. Este método es conocido por su eficacia en la resolución de problemas no convexos asociados con el entrenamiento de redes neuronales. La tasa de aprendizaje

se ajusta de acuerdo con un enfoque "piecewise", donde se reduce en un factor de 0.5 cada 200 épocas.

Entrenamiento y configuración de la red MLP El conjunto de datos utilizado para el entrenamiento de la red se divide en tres categorías basadas en las condiciones climáticas: seco, lluvia, y tormenta. Los datos se normalizan en el rango de 0 a 1 antes del entrenamiento para garantizar una convergencia más rápida y un mejor rendimiento de la red. Para cada categoría, los últimos 16 registros equivalentes a 4 horas, ya que cada registro representa 15 minutos, se omiten del conjunto de entrenamiento y se usan como objetivo para predecir el comportamiento futuro del sistema.

Los hiperparámetros son valores establecidos antes de iniciar el entrenamiento y no se actualizan durante el mismo. Estos incluyen aspectos como la tasa de aprendizaje, el número de capas ocultas en la red, el número de unidades en cada capa, entre otros. La tabla 7 presenta los hiperparámetros utilizados para configurar y entrenar la red neuronal MLP en MATLAB, estos valores han sido seleccionados y ajustados para proporcionar el mejor rendimiento posible con nuestros datos de entrada y salida.

Tabla 7: Hiperparámetros MLP para lazo abierto.

Hiperparámetro	Valor utilizado
Unidades de entrada	4
Unidades ocultas	160
Número de clases	1
Número de capas ocultas	3
Función de activación	ReLU
Dropout	0.2
Tamaño de lote (mini-batch)	256
Máximo número de épocas	1000
Frecuencia de validación	10
Aprendizaje programado (LearnRateSchedule)	piecewise
Factor de caída del aprendizaje (LearnRateDropFactor)	0.5
Periodo de caída del aprendizaje (LearnRateDropPeriod)	200

Además de la configuración de los hiperparámetros, otro aspecto importante es el seguimiento del progreso. La figura 17 proporciona información en tiempo real sobre la evolución del modelo a lo largo de las iteraciones de entrenamiento, incluyendo métricas como el error de entrenamiento y validación. Esto nos permite detectar problemas como el sobre ajuste y tomar decisiones sobre cuándo detener el entrenamiento.

En la figura 17 se establece un hardware de un solo CPU para realizar el entrenamiento, lo que puede afectar la velocidad a la que se procesa cada época. No obstante, un solo CPU puede ser suficiente para el entrenamiento de redes neuronales, especialmente si el conjunto de datos no es extremadamente grande.

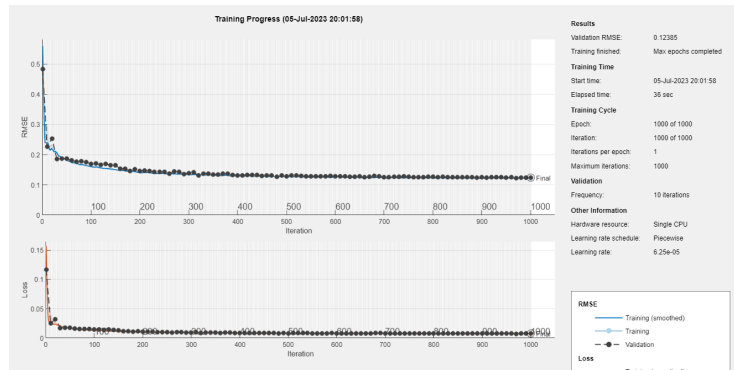


Figura 17: Progreso de entrenamiento MLP.

El aprendizaje se ajusta según la tasa de aprendizaje "piecewise", lo que significa que se reduce por un factor después de un número específico de épocas. En este caso, se estableció una tasa de aprendizaje inicial de $6.25e-0.5$, ya que es fundamental determinar la velocidad a la que el modelo aprende. Una tasa de aprendizaje demasiado alta puede hacer que el modelo oscile alrededor de la solución óptima, mientras que una tasa de aprendizaje demasiado baja puede hacer que el aprendizaje sea muy lento o que el modelo se quede atascado en mínimos locales.

Se debe prestar atención en la pérdida de entrenamiento como a la pérdida de validación. Ambas deben disminuir a lo largo del tiempo y converger a valores similares. Si la función de pérdida de validación comienza a aumentar, mientras que la pérdida de entrenamiento sigue disminuyendo, es posible que estemos ante un caso de sobre ajuste. En este escenario, el modelo está aprendiendo las características del conjunto de entrenamiento sin ningún problema, pero no es capaz de generalizar eficazmente a datos no vistos.

4.1.4 Red para datos obtenidos del sistema de control en lazo cerrado

4.1.4 Diseño y optimización de la red MLP La arquitectura de la red es de naturaleza profunda, compuesta por una capa de entrada de secuencia y cuatro capas ocultas. Cada una de las etapas ocultas está constituida por unidades completamente conectadas, seguidas de una capa de activación ReLU para incorporar no linealidades en el modelo y así permitir que la red aprenda y modele relaciones más esenciales entre los datos. Para prevenir el sobre ajuste, se incluye una capa de dropout después de cada capa ReLU. Este diseño bien estructurado, se puede observar en la figura 18, el cual busca alcanzar una capacidad de modelado, manteniendo siempre un control sobre el posible sobre ajuste y garantizando una buena capacidad de generalización.

Para la optimización se emplea el algoritmo de Adam, una modalidad de descenso de gradiente estocástico reconocida por su eficacia en el entrenamiento

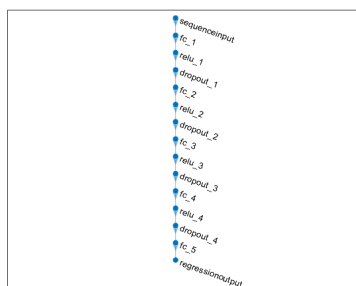


Figura 18: Estructura de la red neuronal MLP.

de redes profundas. Se utilizan estrategias avanzadas de programación de la tasa de aprendizaje, incluyendo una reducción "piecewise" que ajusta la tasa de aprendizaje en ciertos intervalos. Este enfoque proporciona un equilibrio entre convergencia inicial y la optimización de largo plazo, garantizando que el modelo resultante sea robusto y capaz de adaptarse a diversas situaciones de predicción, destacando su aplicabilidad y eficacia en el ámbito de estudio.

Entrenamiento y configuración de la red MLP Los hiperparámetros seleccionados para el entrenamiento de la red pueden observarse en la tabla 8, proporciona una visión clara de la configuración utilizada en el entrenamiento, desde la cantidad de características de entrada hasta los parámetros específicos del entrenamiento como número de épocas, tamaño del lote y frecuencia de validación. La configuración detallada de los hiperparámetros proporciona una base para realizar los experimentos o llevar a cabo ajustes futuros con el fin de mejorar la precisión del modelo.

Tabla 8: Hiperparámetros MLP para lazo cerrado.

Hiperparámetro	Valor
Número de características	Basado en los datos de entrada
Unidades ocultas	165
Número de clases	1
Número de capas ocultas	4
Dropout	0.2
Tamaño de lote	256
Máximo número de épocas	1000
Frecuencia de validación	10
Aprendizaje programado	piecewise
Factor de caída del aprendizaje	0.5
Periodo de caída del aprendizaje	200

Para el entrenamiento de la red, se utilizó una función de activación ReLU y un dropout de 0.2. Las unidades ocultas y las capas ocultas se fijaron en 165, y 4, respectivamente. Se entrenó con un tamaño de lote de 256, un máximo de 1000 épocas y una frecuencia de validación de cada 10 épocas. El aprendizaje se programó de manera 'piecewise', con un factor de caída de 0.5 y un periodo de caída de 200. En la figura que ilustra el progreso de entrenamiento de nuestro modelo. La figura 19 permite apreciar el comportamiento de la red neuronal durante las iteraciones de entrenamiento, mostrando cómo los valores de pérdida de entrenamiento y validación evolucionan con el paso de las épocas. Esta visualización es fundamental para entender cómo nuestro modelo está aprendiendo y permite identificar potenciales problemas como el sobre ajuste.

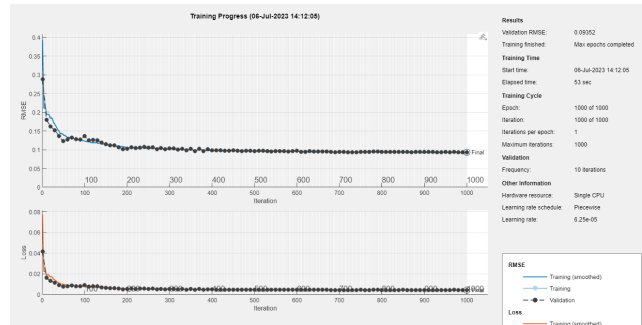


Figura 19: Progreso de entrenamiento MLP.

La figura 19 muestra una disminución constante en los valores de pérdida a medida que avanzan las iteraciones. Inicialmente, el modelo aprende a un ritmo bastante rápido, reflejado en la pronunciada caída de la pérdida durante las primeras 100 iteraciones, luego comienza a estabilizarse, mostrando un decrecimiento gradual en la pérdida. La tasa de aprendizaje utilizada es de $6.25e-05$, es apropiada, ya que permite al modelo aprender de manera efectiva sin causar oscilaciones bruscas en la pérdida. Se observa que el modelo se valida cada 10 iteraciones, permitiendo tener un seguimiento frecuente del desempeño en el conjunto de validación. La pérdida de validación sigue una tendencia similar a la pérdida de entrenamiento, lo que nos indica que el modelo no está sobre ajustando los datos de entrenamiento, es decir, está aprendiendo de manera efectiva, con buen equilibrio entre velocidad de aprendizaje y prevención del sobre ajuste.

4.2 Long-Short Term Memory (LSTM)

4.2.2 Red para datos obtenidos del sistema de control en lazo abierto

4.2.2 Diseño y optimización de la red LSTM El diseño empezó con una capa de entrada de secuencia que recibe los datos normalizados de entrada,

seguido de dos capas ocultas, cada una compuesta por 50 unidades LSTM, luego de cada una de ellas, se aplica una capa de dropout con una tasa de 0.1 para mitigar el sobre ajuste, permitiendo que ciertas neuronas se acoplen durante el proceso de entrenamiento y así evitar que la red dependa de una pequeña cantidad de neuronas, se introduce una capa completamente conectada a una función de activación ReLU, con la finalidad de introducir la no linealidad en la red, permitiendo que aprenda y modele relaciones más complejas entre las entradas y las salidas.

El diseño de la red se muestra en la figura 20, donde se puede apreciar la secuencia de capas que constituyen la arquitectura de la red.

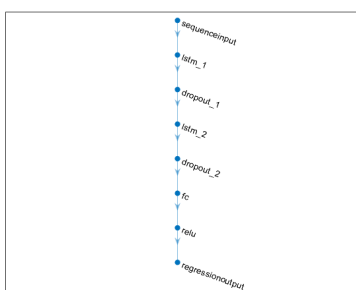


Figura 20: Estructura de la red neuronal LSTM.

Para optimizar el rendimiento, se aplicaron técnicas de ajuste de hiperparámetros. Se empleó el algoritmo Adam para el entrenamiento, configurando un número máximo de 400 épocas y un tamaño de minibatch de 64. Se usó un enfoque de validación durante el proceso de entrenamiento para evaluar el desempeño de la red, registrándose cada cinco épocas. Se optó por reorganizar los datos en cada época para evitar cualquier sesgo debido al orden de los datos. Con esto se buscó lograr un modelo que ofrezca precisión en las predicciones y resistencia al sobre ajuste.

Entrenamiento y configuración de la red LSTM Los hiperparámetros utilizados para configurar y entrenar la red neuronal LSTM en MATLAB, han sido seleccionados y ajustados para proporcionar el rendimiento óptimo, este tipo de red neuronal se destaca por su capacidad para aprender dependencias a largo plazo, lo que la hace especialmente útil para tareas de secuencia a secuencia, como es el caso de nuestro estudio. La eficiencia y precisión de la LSTM reside en su arquitectura robusta y en el ajuste de los hiperparámetros que se presentan en la tabla 9.

La red neuronal LSTM, construida y entrenada, presenta algunas diferencias clave en sus hiperparámetros con respecto a una red MLP. La cantidad de

Tabla 9: Hiperparámetros LSTM para lazo abierto.

Hiperparámetro	Valor
Épocas Máximas	400
Tamaño de Mini Lote	64
Frecuencia de Validación	5
Número de Capas Ocultas	2
Número de Unidades Ocultas	50
Tasa de Abandono (Dropout)	0.1
Tasa de Aprendizaje	Adam
Tipo de Capa de Entrada	Capa de Entrada de Secuencia
Tipo de Capa de Salida	Capa de Regresión
Función de Activación	ReLU
Método de Normalización	Escalado Min-Max

épocas y el tamaño del mini lote son dos de estos parámetros que se han ajustado de manera específica para la red. Se limitó las épocas máximas a 400, en comparación con las 1000 épocas utilizadas en la MLP. Esto se debe a que las redes LSTM son computacionalmente más costosas y se necesita encontrar un equilibrio entre el rendimiento y el tiempo de entrenamiento. Además, la red utiliza un tamaño de mini lote de 64, en lugar del tamaño de mini lote de 256 utilizados en la MLP, un tamaño más pequeño puede ayudar a aprender mejor las dependencias a largo plazo en los datos.

En cuanto al número de capas ocultas y las unidades ocultas, la red LSTM utiliza 2 capas ocultas con 50 unidades ocultas, mientras que la MLP utiliza 3 capas con 160 unidades. Aunque a primera vista la red LSTM parezca más simple, no es así, ya que cada unidad es un sistema complejo capaz de retener y olvidar información, por lo que su capacidad de modelado no se reduce. Es importante mencionar que el tiempo de entrenamiento de una red LSTM es mayor que el de una red MLP debido a su mayor complejidad. Sin embargo, la capacidad de la LSTM para modelar dependencias a largo plazo en los datos puede resultar en un mejor rendimiento en tareas de secuencia a secuencia.

La figura 21 muestra el progreso del entrenamiento de la red LSTM, el gráfico, muestra el comportamiento del error en los conjuntos de entrenamiento y validación a medida que avanza el número de épocas. La disminución del error en ambos conjuntos nos permite confirmar la efectividad del aprendizaje de la red y su capacidad para generalizar a datos no vistos.

En la figura 21 se observa una representación visual de cómo los valores de la raíz del error cuadrático medio y la función de pérdida evolucionan durante las iteraciones de entrenamiento, como era de esperarlo, existe una disminución constante en los errores tanto de entrenamiento como de validación, lo cual es un indicador de que nuestra red está aprendiendo y ajustándose adecuadamente con cada iteración del entrenamiento.

4.2.4 Red para datos obtenidos del sistema de control en lazo cerrado

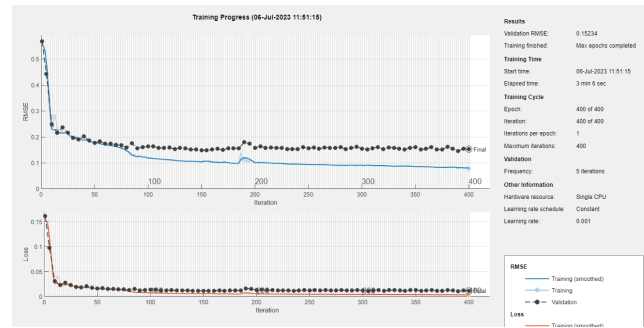


Figura 21: Progreso de entrenamiento LSTM.

4.2.4 Diseño y optimización de la red LSTM En el diseño y optimización de la red, su configuración y arquitectura es fundamental, que a menudo implica hallar un equilibrio entre la complejidad y la capacidad de generalización. En esta investigación, la red se compone de un nivel de entrada de secuencia, múltiples estratos LSTM ocultos y un nivel de regresión al final. A cada estrato LSTM le sigue un módulo de abandono con una tasa del 10% para la regularización. Se emplea la función de activación ReLU en el nivel totalmente conectado para gestionar las no linealidades del modelo, permitiendo que la red capture y modele las relaciones más complejas entre los datos como se observa en la figura 22.

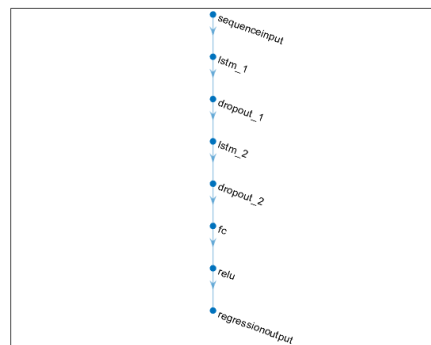


Figura 22: Estructura de la red neuronal LSTM.

En la optimización de la red, el número de unidades ocultas, el número de capas ocultas y la tasa de abandono son parámetros críticos. Con una adecuada selección y ajuste de estos parámetros. A través de la validación cruzada, la red puede ser entrenada para minimizar el error en los datos de entrenamiento mientras se mantiene una alta capacidad de generalización en los datos de validación

y prueba. La eficacia del diseño y optimización de la red puede evaluarse mediante métricas de rendimiento como el error cuadrático medio, que proporciona una medida cuantitativa de la precisión del modelo.

Entrenamiento y configuración de la red LSTM Los hiperparámetros configurados y entrenados para la red LSTM en MATLAB han sido elegidos y ajustados para obtener la mejor eficacia posible. Este tipo de red es reconocido por su habilidad para aprender dependencias a largo plazo, una característica particularmente valiosa para tareas de secuencia a secuencia como la que se aborda en nuestro estudio. El desempeño preciso y eficaz de la red se debe a su sólida arquitectura y calibración de los hiperparámetros que se observan en la tabla 10.

Tabla 10: Hiperparámetros LSTM para lazo cerrado.

Hiperparámetro	Valor
Función de activación	ReLU
Dropout	0.1
Número de características	4 (dependiendo de los datos)
Número de unidades ocultas	50
Número de clases	1
Número de capas ocultas	2
Máximas épocas	400
Tamaño de lote miniatura	64
Frecuencia de validación	10
Agitación	en cada época
Programa de aprendizaje	'piecewise'
Factor de caída de aprendizaje	0.5
Periodo de caída de aprendizaje	200

En la figura 23, se muestra el progreso de entrenamiento para nuestra red, observando el rendimiento en los datos de entrenamiento como en los de validación, demostrando que la pérdida información del modelo afecta las predicciones y los valores reales, el eje vertical se representa la pérdida del modelo y en el horizontal, el número de épocas. Como se puede observar, ambas pérdidas disminuyen inicialmente, indicando que el modelo está aprendiendo a partir de los datos de entrenamiento. Sin embargo, a partir de la época 60, vemos una tendencia hacia la desestabilización del modelo. La pérdida en el conjunto de validación comienza a aumentar, lo cual es una señal de que el modelo puede estar empezando a sobreajustarse a los datos de entrenamiento generando una perdiendo en su capacidad para generalizar a nuevos datos.

5. Resultados

Esta sección presenta las características comunes del entrenamiento en lazo abierto y cerrado de las redes neuronales MLP y LSTM y los resultados del entrenamiento y la validación en ciertos escenarios.

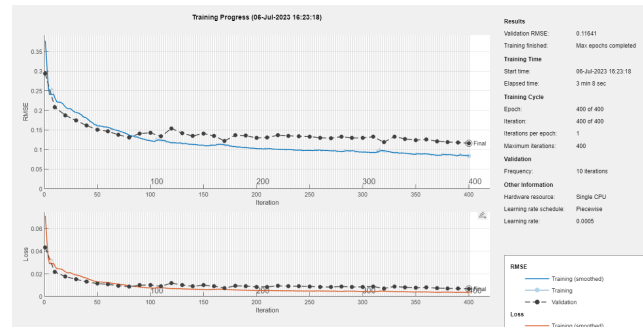


Figura 23: Progreso de entrenamiento LSTM.

5.1 Comparación de desempeño en lazo abierto

Tabla 11: Hiperparámetros de las redes MLP y LSTM en modo lazo abierto

Hiperparámetro	MLP	LSTM
Número de características	4	4
Número de unidades ocultas	160	50
Número de clases	1	1
Número de capas ocultas	4	2
Tasa de abandono	0.2	0.1
Número máximo de épocas	1000	400
Tamaño del lote	256	64
Frecuencia de validación	10	5
Factor de caída de la tasa de aprendizaje	0.5	-
Período de caída de la tasa de aprendizaje	200	-

Las redes neuronales MLP y LSTM representan dos enfoques muy diferentes para modelar relaciones temporales en los datos. En nuestras pruebas para la investigación, el modelo MLP resultó ser más rápido en términos de tiempo de entrenamiento. Esto se debe a que son simples en comparación con las LSTM, ya que no tienen la complejidad adicional de las dependencias a largo plazo en los datos, que es lo que hace que las LSTM sean más costosas computacionalmente. A pesar de su mayor rapidez, el modelo MLP logró superar a la LSTM en términos de rendimiento, lo que indica que este tipo de red es capaz de capturar las relaciones subyacentes en el conjunto de datos de manera más efectiva.

Por otro lado, aunque el modelo LSTM tomó más tiempo para entrenar, no fue capaz de superar el rendimiento de la MLP en nuestras pruebas. Sin embargo, vale la pena destacar que las redes LSTM son buenas en el manejo de secuencias temporales y datos con dependencias a largo plazo, lo cual es una ventaja clave en ciertos dominios de aplicación, como el procesamiento del lenguaje natural y el análisis de series de tiempo. En nuestro caso, a pesar de

que las LSTM son técnicamente más avanzadas y flexibles, no cumplieron con nuestras expectativas. Estos resultados subrayan la importancia de la elección del modelo correcto según la naturaleza y la complejidad del problema a tratar.

En la figura 24, se observa el desempeño del modelo MLP al predecir los datos de prueba, comparado con los resultados reales, las predicciones realizadas por nuestro modelo se representan en azul, mientras que los resultados reales se representan en rojo.

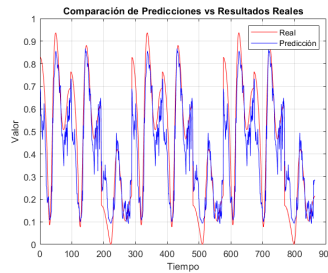


Figura 24: Resultados de comparación de predicciones vs resultados reales.

El trazado de las predicciones se alinea en gran medida con el trazado de los resultados reales, lo que indica que nuestro modelo es capaz de capturar la tendencia general de los datos. Sin embargo, existen ciertas áreas donde el modelo no logra predecir los resultados exactos, lo cual se puede esperar en cualquier problema de aprendizaje automático debido a la naturaleza probabilística y la presencia de ruido en los datos.

La medida cuantitativa del rendimiento del modelo es el error cuadrático medio, en nuestro caso, el MSE es de 0.012119, es un valor relativamente pequeño, lo que indica que las predicciones del modelo son cercanas a los resultados reales. Sin embargo, siempre es recomendable comparar este valor con una métrica de referencia o con los resultados obtenidos por otros modelos para tener un mejor contexto de qué tan bueno es este MSE.

La figura 25 compara las predicciones generadas por la red LSTM con los resultados reales, el eje Y se representa el valor de la variable objetivo, mientras que en el eje X se muestra el tiempo. Las líneas azules representan las predicciones realizadas y las líneas rojas los datos reales, lo permite visualizar la precisión de las predicciones y evaluar si es capaz de capturar los patrones subyacentes en los datos.

La red realiza un buen trabajo al predecir los picos más altos en los datos, ya que las predicciones se mantienen dentro del rango esperado, es decir, la red es capaz de capturar y predecir con precisión los cambios más significativos. Sin embargo, se observa cierta discrepancia en la predicción de los picos medios, donde las predicciones no coinciden exactamente con los valores reales, debido a la naturaleza compleja de los datos en estos puntos. El error cuadrático medio es 0.031455, que es una medida cuantitativa del error de nuestras predicciones.

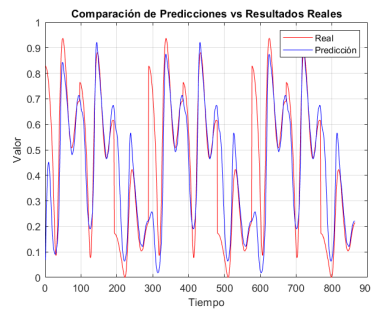


Figura 25: Resultados de comparación de predicciones vs resultados reales.

Aunque este valor no es alto, indica que hay cierto grado de error en nuestras predicciones. Si hablamos de rendimiento del modelo, un MSE bajo sería lo más deseable, ya que indica una menor discrepancia entre las predicciones y los valores reales.

A pesar de las diferencias en su configuración y la naturaleza de las tareas para las que son más adecuadas, en ambas redes tanto la MLP como la LSTM han demostrado ser capaces de aprender eficazmente a partir de los datos de entrenamiento y realizar predicciones precisas. Sin embargo, en términos de rendimiento, la MLP ha demostrado producir un error cuadrático medio más bajo, lo que indica una menor discrepancia entre las predicciones del modelo y los valores reales, por lo que se sugiere que la red MLP, es más adecuada para nuestro problema específico y conjunto de datos.

5.2 Comparación de desempeño en lazo cerrado

Se presenta una comparación detallada del desempeño de las redes neuronales MLP y LSTM operando los datos en lazo cerrado. Ambos tipos de redes se utilizan para modelar relaciones temporales en los datos, pero difieren significativamente en su estructura y funcionamiento interno. La tabla 12 muestra los hiperparámetros usados para entrenar ambas redes, lo que permite una comparación de su rendimiento. Aunque ofrecen ventajas únicas, su capacidad para un problema específico depende en gran medida de las características de los datos y de la tarea en cuestión. En nuestra investigación, observamos diferencias notables en el tiempo de entrenamiento y precisión de los modelos.

En nuestra investigación, implementamos dos tipos de redes neuronales, una MLP y LSTM, ambas configuradas en un diseño de lazo cerrado. A pesar de que su finalidad era modelar la secuencia de datos de entrada, estos modelos difieren en términos de su estructura y configuración de hiperparámetros. El modelo MLP consiste en 4 capas ocultas y utiliza 165 unidades ocultas, con una función de activación de ReLU. En cambio, el modelo LSTM contiene 2 capas ocultas con 50 unidades ocultas cada una, también utilizando una función de activación ReLU. Ambos modelos emplean un optimizador Adam con una tasa

Tabla 12: Comparación de los hiperparámetros de los modelos de redes neuronales para lazo cerrado.

Modelo	Capas	Ocul- Unidades Ocultas	Función de Activación	Tasa de Apre- ndi- zaje	Batch Si- ze
MLP	4	165	ReLU	0.001 (adam)	256
LSTM	2	50	ReLU	0.001 (adam)	64

de aprendizaje de 0.001, pero difieren en su tamaño de lote o 'batch size' durante el entrenamiento. El modelo MLP usa un tamaño de lote de 256, mientras que el modelo LSTM usa un tamaño de lote más pequeño de 64. Estas diferencias en los hiperparámetros reflejan las características únicas de cada tipo de red neuronal y su adaptabilidad para el conjunto de datos usado.

Además de estas diferencias estructurales y de configuración, también observamos variaciones en el tiempo de simulación. La red LSTM se demora más tiempo debido a su complejidad inherente y a su capacidad para manejar dependencias a largo plazo, mientras que la red MLP, con su estructura más sencilla, requiere menos tiempo. Sin embargo, a pesar de esta mayor demanda de tiempo, en nuestra investigación, la red MLP proporcionó mejores resultados en términos de rendimiento en la tarea de modelado de la secuencia.

Las diferencias en los hiperparámetros y tiempos de simulación reflejan las características únicas de cada tipo de red neuronal y su funcionalidad para el conjunto de datos utilizado.

La figura 26 representa los trazados de las predicciones generadas en nuestro modelo en contraste con los valores reales. Nos permite evaluar visualmente la precisión de nuestras predicciones y cómo estas se alinean con los datos verdaderos.

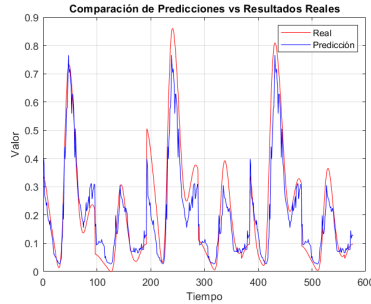


Figura 26: Resultados de comparación de predicciones vs resultados reales.

Analizando la figura 26, se observa que el modelo predice de manera efectiva los picos de los datos reales, sin embargo, en los valores pequeños y medianos, se evidencia una menor precisión, al parecer tiene dificultades para capturar

la forma de los datos en estas regiones, lo que resulta una discrepancia entre las predicciones y los datos reales. A pesar de esto, el error cuadrático medio obtenido es 0.0059678. Este valor es bajo, queriendo decir que nuestro modelo no está demasiado alejado de los valores reales. Sin embargo, debemos considerar la posibilidad de optimizar más el modelo, con especial atención a los valores pequeños y medianos, para mejorar aún más este indicador.

La figura 27 presenta una comparación entre las predicciones generadas por nuestro modelo LSTM y los resultados reales, se puede observar que nuestro modelo ha logrado capturar la dinámica general de los datos, lo que indica que aprendió patrones importantes en los datos de entrenamiento. Sin embargo, al revisar la gráfica con más detalle, se identifican algunas discrepancias. En los picos altos, por ejemplo, las predicciones tienden a sobrepasar los valores reales. Esto concuerda con el progreso de entrenamiento, en la que notamos una posible tendencia al sobre ajuste después de la época 60. Estos resultados nos indican que se debe continuar afinando nuestro modelo y explorando estrategias para evitar el sobre ajuste, como la regularización o el ajuste de los hiperparámetros, con el fin de mejorar la precisión de las predicciones en estos puntos críticos.

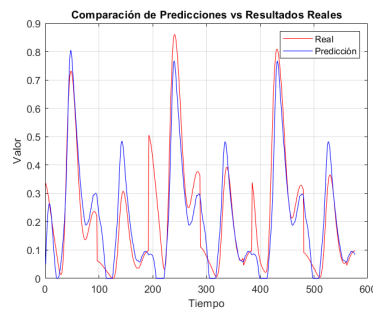


Figura 27: Resultados de comparación de predicciones vs resultados reales.

Los modelos de redes neuronales LSTM y MLP aplicados a nuestro conjunto de datos, han demostrado un comportamiento superior a lo esperado, la principal métrica utilizada para evaluar ambos modelos fue el error cuadrático medio, en la cual la red MLP arrojó un error más bajo en comparación a la LSTM, indicando una precisión en los resultados más alta.

6. Conclusiones y Recomendaciones

Las redes MLP y LSTM han demostrado ser capaces de aprender de manera efectiva y realizar predicciones precisas en los conjuntos de datos tanto en lazo abierto como en lazo cerrado. Sin embargo, en términos de rendimiento, la red MLP ha mostrado un error cuadrático medio (MSE) más bajo en comparación con la red LSTM en ambos escenarios. Esto indica que la red MLP ha logrado

una mayor precisión en sus predicciones, capturando de manera más precisa los patrones subyacentes en los datos.

En el entrenamiento de ambas redes, se observa una tendencia al sobreajuste a medida que avanzan las iteraciones. En el caso de la red LSTM en lazo cerrado, se observa un aumento en la pérdida de validación después de la época 60, lo que indica una posible pérdida de capacidad de generalización del modelo. Esto resalta la importancia de implementar estrategias como la regularización o la optimización de hiperparámetros para mitigar el sobreajuste y mejorar la capacidad de generalización del modelo.

Las redes MLP y LSTM han sido capaces de capturar la tendencia general de los datos y predecir los picos más altos en los resultados. Sin embargo, en los valores pequeños y medianos, se ha observado una menor precisión en las predicciones, especialmente en la red LSTM. Esto sugiere que hay margen para mejorar la capacidad de los modelos para capturar patrones en estas regiones de los datos.

La arquitectura de la red, incluyendo el número de capas ocultas, unidades ocultas y la selección de la función de activación, juegan un papel crucial en el rendimiento del modelo. La configuración adecuada de estos hiperparámetros es fundamental para lograr un buen desempeño. En ambos escenarios, se han utilizado diferentes configuraciones de hiperparámetros para la red MLP y LSTM, lo que destaca la importancia de adaptar la arquitectura de la red a las características específicas del problema y los datos.

A pesar de los buenos resultados obtenidos, existen oportunidades para mejorar aún más el rendimiento de los modelos. Estas mejoras podrían incluir la optimización de los hiperparámetros, la exploración de técnicas de regularización para evitar el sobreajuste, el análisis y la corrección de las discrepancias en las predicciones en los valores pequeños y medianos, así como la exploración de arquitecturas de red más complejas o enfoques de ensamblaje para mejorar la precisión y la capacidad de generalización del modelo.

Referencias

1. Hamed, M. M., Khalafallah, M. G., Hassanien, E. A. (2004). Prediction of wastewater treatment plant performance using artificial neural networks. *Environmental Modelling and Software*, 19(10), 919-928. <https://doi.org/10.1016/j.envsoft.2003.10.005>
2. Sutil, M. F. (2011). Diseño simultáneo de procesos y sistemas de control predictivo mediante índices de controlabilidad basados en normas. *Ph.D. thesis, Universidad de Salamanca*.
3. Santín, I., Pedret, C., Vilanova, R. (2014). MPC tuning in a hierarchical control structure for a wastewater treatment process. *Revista Iberoamericana de Automática e Informática Industrial*.
4. Dagnino, J. (2014). Coeficiente de correlación lineal de Pearson. *Chil Anest*, 43(1), 150-153.
5. Vilanova, R., Santín, I., Pedret, C. (2017). Control y operación de estaciones depuradoras de aguas residuales: Modelado y simulación. *Revista iberoamericana de automática e informática industrial*, 14(3), 217-233.

6. Alex, J., Benedetti, L., Copp, J. B., Gernaey, K. V., Jeppsson, U., Nopens, I., Pons, M. N., Rieger, L., Rosen, C., Steyer, J. P., et al. (2008). Benchmark simulation model no. 1 (BSM1). *Report by the IWA Taskgroup on benchmarking of control strategies for WWTPs, 1*.
7. Rodríguez, E. M. (2005). Errores frecuentes en la interpretación del coeficiente de determinación lineal. *Anuario jurídico y económico escorialense, 1*(38), 315-331.
8. Sánchez Turcios, R. A. (2015). t-Student: Usos y abusos. *Revista mexicana de cardiología, 26*(1), 59-61.
9. Pisa, I., Santín, I., Lopez Vicario, J., Morell, A., Vilanova, R. (2018). A recurrent neural network for wastewater treatment plant effluents' prediction. *Actas de las XXXIX Jornadas de Automática, Badajoz, 5-7 de Septiembre de 2018*.
10. Tarun, A., Reddy, G. P. (2022). Artificial Neural Networks for Waste-water Treatment Plant Control. *Proceedings of International Conference on Industrial Instrumentation and Control: ICI2C 2021, 409-419*.
11. Tyagi, A. K., Abraham, A. (2022). *Recurrent Neural Networks: Concepts and Applications*. CRC Press.
12. P. López Miguel *et al.*, "Detección de actividades anómalas en espacios públicos mediante redes neuronales profundas," Master's thesis, Universidad de Alcalá, Escuela Politécnica Superior, 2019.
13. A. Vilagran Solsona, "Facial Expression Detection using Convolutional Neural Networks," Bachelor's thesis, Universitat Politècnica de Catalunya, 2018.
14. J. Pérez Guerrero, Redes recurrentes," Master's thesis, Universidad de Sevilla, 2020.
15. A. Dikshit, B. Pradhan, and M. Santosh, "Artificial neural networks in drought prediction in the 21st century—A scientometric analysis," *Applied Soft Computing*, vol. 114, p. 108080, 2022. DOI: <https://doi.org/10.1016/j.asoc.2021.108080>.

Inferencia dinámica de estados contextuales en sistemas de recomendación sensibles al contexto

Henar López Huertas, María N. Moreno-García, Álvaro Lozano Murciego

Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los
Caídos s/n. 37008, Salamanca, España
{henar,mmg,loza}@usal.es

Resumen Este trabajo aborda el tema de la inferencia dinámica del contexto en sistemas de recomendación sensibles al contexto y basados en sesión. El objetivo es plantear un sistema que asigne automáticamente etiquetas contextuales a las canciones en función de sus características musicales, utilizando un modelo de clasificación multi-etiqueta. Estas etiquetas se usarán posteriormente para inferir el contexto latente del usuario en la sesión en la que escucha música y proporcionarle recomendaciones adecuadas a ese contexto mediante métodos de recomendación basados en sesión. El sistema se evalúa utilizando un caso de estudio en el que se comparan los resultados de diferentes métodos de clasificación multi-etiqueta y de métodos de recomendación basados en sesiones, además de analizar la influencia de las etiquetas contextuales y otras características musicales en la fiabilidad de las recomendaciones.

Keywords: sistemas de recomendación, contexto, música, sesión.

1. Introducción

La enorme cantidad de información y recursos a los que se tiene acceso actualmente a través de medios electrónicos resulta beneficiosa ya que permite satisfacer las necesidades concretas de información de cada individuo. Sin embargo, lograr encontrar los resultados adecuados para el usuario puede resultar complicado, y es por eso que se utilizan los sistemas de recomendación. Los sistemas de recomendación (SR) son herramientas que permiten predecir las preferencias de los usuarios y sugerirles productos o servicios de forma personalizada. Establecen un conjunto de criterios y clasificaciones de los datos de los usuarios para hacer recomendaciones sobre elementos que pueden ser útiles o valiosos. Estos sistemas hacen uso de los datos proporcionados por el usuario bien de forma directa o indirectamente a partir de las interacciones del usuario con el sistema, para luego analizar y procesar dicha información, convertir estos datos en conocimiento sobre sus preferencias y proporcionarle recomendaciones.

Los sistemas de recomendación surgieron a partir de la observación de que los individuos a menudo se basan en las opiniones y recomendaciones de otros. Los primeros SR simulaban este comportamiento aplicando algoritmos que hacían

uso de recomendaciones hechas por una comunidad de usuarios y se las proporcionaban a usuarios activos en busca de las mismas. Otro principio básico en que se basan los algoritmos de recomendación [1] es la dependencia existente entre los usuarios y las actividades centradas en ítems. Esto quiere decir que, normalmente, va a existir cierta correlación entre los distintos ítems consumidos por una persona. Encontrar esa relación permitirá que la recomendación sea más precisa.

Los datos utilizados por los sistemas de recomendación pueden clasificarse en tres clases [2]. Los ítems son los objetos recomendados al usuario y en su naturaleza dependerá del tipo de proceso. Los usuarios, por otra parte, constituyen una parte fundamental ya que son el objeto hacia el que va dirigida la recomendación y es importante tener toda la información posible acerca de los mismos. Por último, las transacciones consisten en una interacción registrada entre el usuario y el sistema de recomendación. Se dedican a almacenar la información generada durante la interacción humano-ordenador y poder utilizarla en otras operaciones.

Una vez identificado el ítem y el usuario, el sistema de recomendación debe ser capaz de predecir si realmente vale la pena recomendar el ítem a ese usuario concreto. Para ello, existen unos elementos básicos que son imprescindibles para la personalización: el histórico de datos que contiene información acerca del perfil de los usuarios, recogida a través de su comportamiento en el sistema; los datos de entrada obtenidos a través de las interacciones de los usuarios con el sistema; y el algoritmo, es decir, el método que se utiliza para realizar la recomendación.

Algunos de los resultados conseguidos con los sistemas de recomendación, aplicándolos sobretodo a la hora de vender algún producto o servicio [2], son el incremento del número de ítems vendidos, la oferta de un rango más diverso de productos, un aumento de la satisfacción y la fidelidad del usuario y un mejor entendimiento de lo que el usuario quiere. Por último, es importante destacar el uso actual que tienen estos sistemas de recomendación [2]. En primer lugar, juegan un papel importante en algunas de las mayores aplicaciones de Internet como son Amazon, Netflix, Facebook, Youtube, Spotify, entre otras. Por otro lado, también tiene aplicaciones relacionadas con la educación o la economía.

Este artículo se estructura de la siguiente forma: la sección 2 refleja brevemente los fundamentos de los sistemas de recomendación sensibles al contexto, los sistemas de recomendación basados en sesión y la inferencia del contexto en el dominio de la música; en la sección 3 se presenta el sistema propuesto para aunar la inferencia del contexto del usuario y la recomendación basada en sesión en el dominio de la música; la sección 4 presenta un caso de estudio que consiste en una serie de experimentos para evaluar el sistema propuesto; en la sección 5 se presentan y discuten los resultados obtenidos y, por último, en la sección 6 se reflejan las conclusiones obtenidas y las líneas de trabajo futuras.

El objetivo principal de este trabajo es obtener características contextuales a partir de los atributos de las canciones utilizando para ello la clasificación multi-etiqueta. Asimismo, se recurre a los sistemas de recomendación basados en sesión, con la finalidad de inferir el contexto dinámicamente en las sesiones del usuario.

2. Fundamentos

2.1 Sistemas de recomendación sensibles al contexto

Una vez advertida la importancia y utilidad de los SR, es momento de introducir los sistemas de recomendación sensibles al contexto (CARS). Estos sistemas van un paso más allá y realizan recomendaciones que, además de tener en cuenta la información habitual, tienen en cuenta la información contextual. El contexto [2] es un concepto que se aplica en numerosas disciplinas, entre las que se incluyen la informática, la filosofía y la psicología, y que se afronta desde un punto de vista determinado en cada una de ellas. De forma genérica, el contexto se puede definir como un conjunto de condiciones o circunstancias que afectan a una situación y sin las cuales no puede comprenderse correctamente. En el ámbito de los sistemas de recomendación el contexto puede clasificarse de la siguiente manera [3]:

- **Contexto individual:** se corresponde con información observada de entidades independientes que pueden tener rasgos en común. Aquí se incluyen desde la información meteorológica como las preferencias de pago del usuario.
- **Localización:** se refiere a la localización del usuario.
- **Tiempo:** contiene la información relativa al momento en que se encuentra el usuario, como el día de la semana o la hora.
- **Actividad:** información acerca de la actividad o tarea que se está llevando a cabo.
- **Contexto social:** si el usuario se encuentra solo o acompañado y, si es así, de quién.

Los sistemas de recomendación funcionan de manera que su objetivo es estimar la función de *rating* R , siendo el *rating* la valoración del usuario sobre un determinado ítem. Los sistemas tradicionales utilizan un set de *ratings*, proporcionado explícitamente por los usuarios o inferido por el sistema para estimar dicha función [4]. Esto se ilustra con la siguiente expresión:

$$R : User \times Item \rightarrow Rating \quad (1)$$

Estos sistemas son bidimensionales ya que durante el proceso solo se consideran dos dimensiones: el usuario y el ítem.

En la recomendación sensible al contexto se va un paso más allá y se incorpora o utiliza evidencia adicional, más allá de la información sobre usuarios, para estimar las preferencias de los usuarios sobre elementos no consumidos por estos. De esta manera, el contexto va a influenciar la función *Rating*, que va a modelarse ahora de la siguiente manera:

$$R : User \times Item \times Context \rightarrow Rating \quad (2)$$

En este caso se trabaja en un espacio multidimensional donde la dimensión de contexto es la variable y la condición de contexto son los valores del mismo.

Como se ha comentado anteriormente, la información contextual puede ser de diferentes tipos como, por ejemplo, tiempo, lugar, compañía, etc., lo que provoca que puedan existir varias dimensiones o variables contextuales.

Para terminar, destacar que la información contextual puede obtenerse de varias maneras [5]. Puede obtenerse explícitamente solicitándosela directamente al usuario, implícitamente a través de otras fuentes o mediante inferencia a partir del comportamiento del usuario. Algunos ejemplos serían pedir al usuario que conteste una serie de preguntas u obtener su localización a través de los datos de su teléfono. También puede conseguirse implícitamente, a partir de los datos y el entorno. En este caso no es necesario interactuar directamente con el usuario u otras fuentes de información. Por ejemplo, puede conocerse el contexto temporal a través del *timestamp* de una transacción. Por último, puede obtenerse mediante inferencia utilizando métodos estadísticos o minería de datos. Aunque la información contextual esté oculta en los datos de forma latente, se puede usar implícitamente para estimar los valores desconocidos.

2.2 Sistemas de recomendación basados en sesión

Los sistemas de recomendación mencionados hasta ahora tienden a utilizar todo el histórico de datos, que contiene las acciones directas o indirectas de los usuarios hacia los ítems, para poder comprender las preferencias de los usuarios y realizar sugerencias. Sin embargo, esto puede no ser siempre la manera más efectiva. En primer lugar, la elección de ítems por parte del usuario suele estar marcada por sus últimas interacciones con los mismos, que representan una pequeña parte de todo el histórico de datos. Además, las preferencias tienden a variar con el tiempo [6].

Para afrontar esta cuestión aparecen, entre otros, los sistemas de recomendación basados en sesión. El concepto de sesión define una lista de interacciones con un límite claro, que se refiere al par de interacción de inicio y finalización para iniciar y finalizar una sesión específica en un evento. El objetivo de estos sistemas es adaptar las sugerencias que se hacen a los usuarios de acuerdo con sus intenciones a corto plazo asumidas durante una sesión en curso. Los *inputs* principales en tales entornos consisten en la secuencia de interacciones con un usuario individual que fueron observadas por el sistema desde el inicio de la sesión así como información sobre sesiones pasadas de otros usuarios [7]. Estos sistemas aprenden las preferencias del usuario a partir de las sesiones asociadas y generadas durante el proceso de consumo. Cada sesión se compone de múltiples interacciones del usuario sobre el ítem que ocurren en un determinado periodo de tiempo que puede durar algunos minutos o varias horas. De esta manera, al tomar como unidad de entrada cada sesión, el sistema puede capturar tanto las preferencias a corto plazo como la dinámica que se refleja en el cambio de preferencias de una sesión a otra para recomendaciones más precisas y oportunas [6].

2.3 Dominio de la música

En los últimos años, la cantidad de servicios para escuchar música en streaming ha aumentado considerablemente, y con ellos la cantidad de música disponible. Esto provoca una necesidad sustancial de utilizar sistemas de recomendación en este ámbito para que el usuario no se vea abrumado por una oferta tan extensa y pueda consumir canciones que le gusten. Como ya se ha mencionado, es muy importante tener en cuenta el tipo de ítem con el que se va a trabajar a la hora de diseñar el sistema de recomendación. En el caso de la música, por ejemplo, un aspecto que lo diferencia de otros ítems, como pueden ser los libros o las películas, es el tiempo que tarda en consumirse. Mientras que un libro puede ocupar días y una película horas, una canción, normalmente, dura un par de minutos. Este factor debe tenerse muy presente ya que reduce el tiempo que tarda un usuario en formar sus gustos musicales y, a su vez, lo convierte en un ítem que puede consumirse repetidas veces. Es bastante habitual que una persona escuche varias veces una canción que le gusta, sin embargo, no es tan normal que esto ocurra con una película o un libro.

3. Sistema propuesto

Para conseguir aunar la inferencia del contexto del usuario y la recomendación basada en sesión o *session-based* se propone en esta sección un sistema de recomendación, que puede dividirse en dos partes. La primera abarca el etiquetado contextual a partir de los atributos de las canciones que escucha el usuario, empleando técnicas de clasificación multi-etiqueta. La segunda aborda la recomendación de canciones apropiadas para el contexto o contextos latentes de la sesión, lo que se infiere utilizando las etiquetas contextuales de las canciones inducidas en la parte anterior. Para ello se aplican algoritmos de recomendación basados en sesiones.

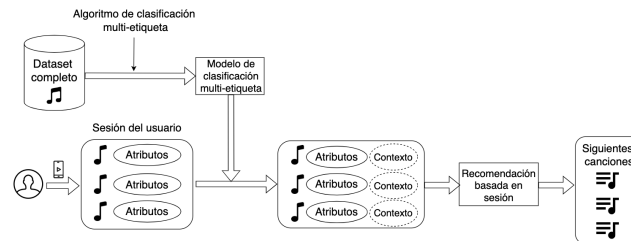


Figura 1: Diagrama general del sistema propuesto.

Para el etiquetado contextual de las canciones se empleará un sistema de clasificación multi-etiqueta que, utilizando las características de las canciones que proporciona el servicio de música *Spotify* a través de su API [8], obtendrá las etiquetas contextuales correspondientes para cada canción. Esto permitirá realizar

un proceso de *tagging* o etiquetado de las canciones escuchadas por el usuario con etiquetas vinculadas a uno varios contextos determinados, permitiendo de esta forma inferir posteriormente el contexto latente del usuario en las sesiones. En la Figura 2 se muestra un diagrama simplificado de este proceso.

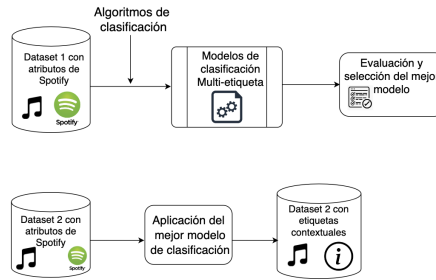


Figura 2: Diagrama simplificado del etiquetado contextual.

La secuencia de canciones escuchada por el usuario hasta el momento y las etiquetas contextuales asignadas en esta parte del sistema serán la entrada de la siguiente parte, el sistema de recomendación basado en sesiones. Una vez etiquetadas las canciones con el contexto del usuario se procederá a generar los modelos de recomendación basados en sesión para poder recomendar canciones adecuadas al contexto latente de la sesión del usuario.

En primer lugar, las canciones y sus etiquetas contextuales se utilizarán como entrada a los algoritmos de recomendación basados en sesiones, para obtener modelos de recomendación que permitan predecir la siguiente canción en la sesión que puede interesarle al usuario en el contexto de dicha sesión (conocido en la literatura este proceso como una de tarea de *Next Item Prediction* o predicción del siguiente ítem dada una secuencia de éstos). En la Figura 3 se muestra un diagrama simplificado de este proceso.

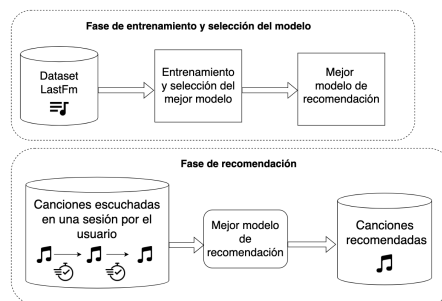


Figura 3: Diagrama simplificado del sistema de recomendación basado en sesiones.

4. Caso de estudio

Para validar el sistema propuesto y la hipótesis planteada en este trabajo se propone un caso de estudio para obtener resultados que permitan dar respuesta a las siguientes preguntas de investigación (*Research Questions*):

RQ1: ¿Qué algoritmo de clasificación multi-etiqueta de los seleccionados presenta mejores resultados al inferir el contexto?

RQ2: ¿Qué algoritmo de recomendación de los seleccionados presenta mejores resultados al realizar la recomendación basada en sesión?

RQ3: ¿La incorporación de etiquetas contextuales mejora los resultados de las recomendaciones respecto a los obtenidos empleando solo las secuencias de canciones?

RQ4: ¿Cuál es la influencia de las características de los ítems frente a la influencia de las características contextuales?

4.1 Etiquetado contextual

Para el dataset se ha tomado como referencia el trabajo realizado en Ibrahim et al. [9]. En dicho estudio se plantea un procedimiento semiautomático que utiliza los títulos de las listas de reproducción para el etiquetado contextual, como también se propuso en Pichl et al. [10]. En Ibrahim et al. se realizó un filtrado de títulos de listas de reproducción utilizando palabras claves referidas al contexto. Finalmente, se escogieron las 15 palabras claves más frecuentes en las listas de reproducción encontradas en la plataforma de *streaming* Deezer. Estos 15 contextos son: *car, chill, club, dance, gym, happy, night, party, relax, running, sad, sleep, car,summer, work* y *workout*. De esta manera se creó un dataset [11] con 49929 canciones. Cada canción tiene asociado su *song id*, que es el identificador de dicha canción en el catálogo de Deezer, y las 15 etiquetas contextuales. Si la canción se encuentra dentro de un contexto esa etiqueta tendrá un valor de 1 y, en caso contrario, tendrá un valor de 0.

En nuestro caso de estudio, el objetivo es inferir el contexto utilizando los atributos de las canciones que proporciona Spotify. Por ello, partiendo del dataset descrito en el párrafo anterior, el primer paso consiste en obtener el identificador de cada canción en Spotify a partir de su identificador en Deezer, el título y el artista. Una vez localizada cada canción en Spotify se obtienen sus correspondientes valores de los atributos. A estas características, además, se le añade la duración en milisegundos de la canción y ya queda totalmente preparado el dataset a utilizar.

Para asociar las etiquetas contextuales a las canciones se utiliza la clasificación multi-etiqueta, dentro de la cual podemos encontrar enfoques de adaptación del algoritmo y de transformación del problema. En este caso se han utilizado algunos de los algoritmos disponibles con la librería *scikit multilearn* [12] que se agrupan de la siguiente manera:

- **Enfoque de adaptación del algoritmo:** Estos métodos se concentran en adaptar los algoritmos de clasificación de una sola etiqueta al caso de múltiples etiquetas, normalmente con técnicas de aprendizaje automático. Para

este caso de estudio se han utilizado `BRkNNaClassifier`, `BRkNNbClassifier`, `MLkNN` y `MLARAM`.

- **Enfoque de transformación del problema:** Convierte los problemas de etiquetas múltiples en problemas de etiqueta única: clase única o clase múltiple. Se han utilizado *Binary Relevance*, con tres clasificadores distintos: *Random Forest (RF)*, *Support Vector Classifier (SVC)* y *XGBoost*; *Classifier Chain* y *Label Powerset*.
- **Agrupación de clasificadores:** Por último, *scikit multilearn* dispone de *ensembles* de clasificadores. De este grupo se utilizan *RAkELd*, *Label Powerset*, *LabelSpacePartitioningClassifier* y *EmbeddingClassifier*.

En cuanto a métricas se utilizan *Hamming loss*, *Precisión*, *Recal*, *AUC* y *F-measure*, también denominada *F1-score*. Para las últimas cuatro métricas se utilizan los promedios macro y micro. El promedio macro calcula la métrica de forma independiente para cada clase y luego toma el promedio, tratando a todas las clases por igual. Por su parte, el promedio micro tiene en cuenta las contribuciones de todas las clases, su número de instancias, para calcular la métrica promedio. Este promedio es preferible si se sospecha que puede haber un desequilibrio de clases.

4.2 Recomendaciones basadas en sesión

Para la recomendación se ha utilizado el dataset de LastFM pero solamente con los 100 usuarios más activos. La razón de esto es que se han hecho pruebas con más usuarios pero el tiempo de entrenamiento era excesivamente alto y la diferencia en los resultados no era muy grande.

Para la recomendación se utiliza la librería *RecBole*, que permite reproducir y desarrollar algoritmos de recomendación. Se prueban tres modelos distintos: *FDSA (Feature-level Deeper Self Attention Network)*, *GRU4RecF* y *SASRecF*.

Las métricas elegidas para evaluar los distintos modelos para la recomendación son: *Precisión*, *Recall*, *MRR (Mean Reciprocal Rank)*, *NDCG (Normalized Discounted Cumulative Gain)* y *Hit Ratio*. Estas métricas presentan variaciones $@N$ adecuadas cuando se comparan listas de recomendación de distinto tamaño, ya que miden hasta cierto punto, un N dado. En este caso de cada una de ellas se han obtenido cuatro valores correspondientes a $@5$, $@10$, $@15$ y $@20$.

Para la aplicación de los algoritmos y la obtención de las métricas mencionadas se ha empleado la biblioteca *RecBole* [13]. Esta biblioteca dispone de diferentes métodos y herramientas de evaluación. Dos características importantes a definir de la evaluación son la partición del conjunto de datos (*data splitting*) y el método de *ranking*. En este caso para el *data splitting* se ha elegido *Temporal Ordering + Leave-one-out Splitting* y para el método de *ranking* se utiliza el modo *full*, que hace un *ranking* completo de todos los ítems, es decir, se evalúa durante el proceso de entrenamiento en todos los ítems.

En el caso de la recomendación basada en sesión, objeto de este caso de estudio, es necesario hacer alguna aclaración de qué protocolo de evaluación utiliza esta biblioteca para obtener sus métricas. En este caso de estudio se ha utilizado el enfoque antes mencionado *Leave One Out*, concretamente la estrategia

LS:valid_and_test que indica que se crean conjuntos de validación y conjuntos de test. Para este caso, la biblioteca hace un *Data Augmentation* de las sesiones basándose en el tamaño máximo de lista tomado.

5. Resultados y Discusión

En el siguiente apartado se muestran y se interpretan los resultados obtenidos dando así respuesta a las *research questions* que se plantearon al proponer el sistema.

5.1 Etiquetado contextual

RQ1: ¿Qué algoritmo de clasificación multi-etiqueta de los seleccionados presenta mejores resultados al inferir el contexto?

Para responder a esta pregunta se representan gráficamente los resultados obtenidos de las métricas de evaluación. Una de las métricas que podemos considerar más relevante es *Hamming loss*, que indica las predicciones realizadas incorrectamente. Cuanto menor es el valor de esta métrica, mejor se considera el clasificador. En la `imgHenarLopez.4` aparecen los valores obtenidos para esta métrica por los distintos algoritmos. Como se observa en la figura, la mayoría han obtenido un valor que ronda el 0,5, excepto *Binary Relevance-RF* y *Binary Relevance-XGBOOST* que presentan los valores más bajos, y por tanto más adecuados, con 0,34 y 0,33, respectivamente.



Figura 4: Resultados *Hamming loss* de los algoritmos de clasificación.

Por otro lado, se hace también uso de métricas cuyo valor ideal es 1, por lo que se busca que tengan un resultado alto, al contrario que *Hamming loss*. Estos resultados se representan en la [imgHenarLopez. 5](#).

Con la gráfica se observan varios aspectos a destacar de los resultados obtenidos. Se aprecia claramente que los algoritmos *Binary Relevance-SVC*, *LabelSpacePartitioningClassifier* y *EmbeddingClassifier* son los que obtienen los resultados más bajos con diferencia y, por tanto, pueden considerarse como los algoritmos menos adecuados para este caso. Por otro lado, *Binary Relevance-RF* y *Binary Relevance-XGBOOST* consiguen los mejores resultados en la mayoría de las métricas.

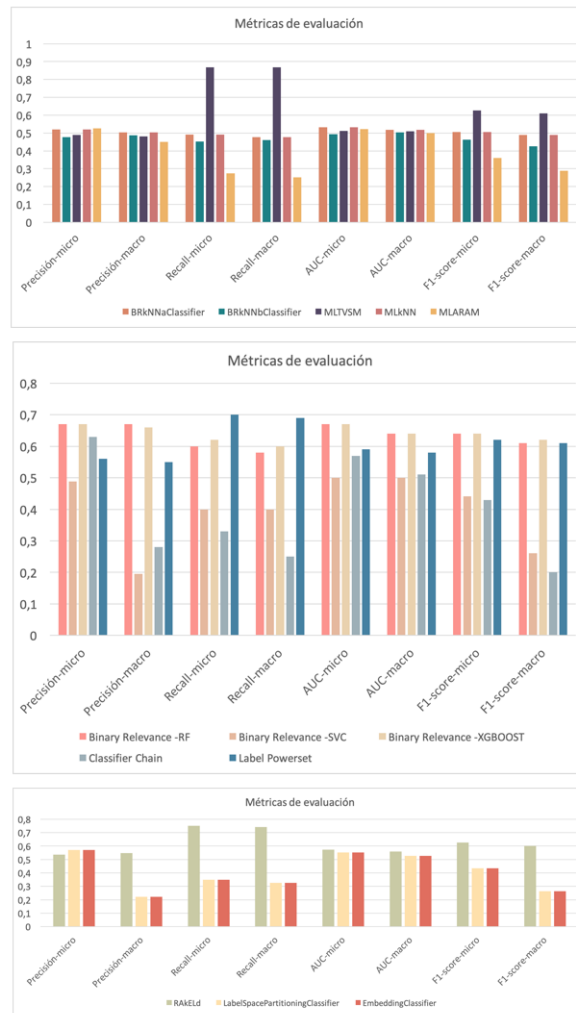


Figura 5: Resultados de las métricas de los algoritmos de clasificación.

5.2 Recomendación basada en sesión

RQ2: ¿Qué algoritmo de recomendación de los seleccionados presenta mejores resultados al realizar la recomendación basada en sesión?

Para contestar a esta pregunta, se generan modelos para cada uno de los algoritmos elegidos sin utilizar las características de los ítems. A continuación, se muestran los valores de las diferentes métricas para FDSA [14], GRU4RecF [15] y SASRecF [16] representados gráficamente.

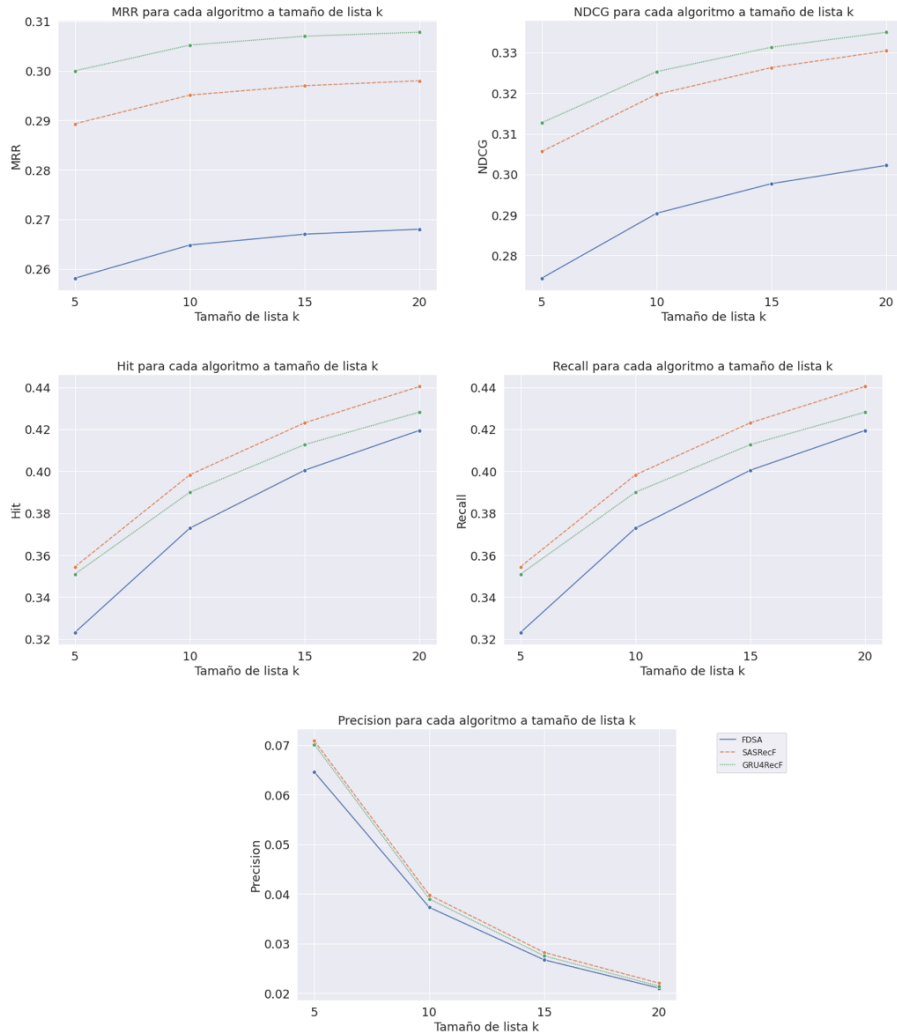


Figura 6: Resultados de las métricas para los ítems sin características.

Para entender las gráficas debe entenderse cómo utiliza la biblioteca *Recbole* las distintas métricas. Para el caso de MRR (*Mean Reciprocal Rank*) calcula el rango recíproco del primer elemento relevante encontrado por el algoritmo. En la [imgHenaarLopez. 6](#) se observa que para todos los algoritmos el valor de MRR aumenta ligeramente a medida que aumenta el tamaño de lista. Además, el mejor resultado lo consigue el algoritmo GRU4Recf con un valor de $MRR@20 = 0,31$. No es un resultado demasiado adecuado pero aquí lo importante es comparar los distintos algoritmos. El peor resultado es obtenido por FDSA.

Los valores de *Hit Ratio* también aumentan según aumenta el tamaño de lista k . En este caso se tienen valores superiores a los de MRR, rondan desde 0,32 a 0,44. El algoritmo con mejores resultados es *SASRecF* y el que tiene peores es *FDSA*, y en este caso la diferencia es mayor que para MRR. Para NDCG se tienen valores similares y también aumentan con k .

Respecto a la precisión se observa que, al contrario que con las otras métricas, el valor disminuye según aumenta el tamaño de lista. Esto se debe a cómo está definida la precisión en *Recbole*. En este caso el ítem relevante es solamente uno y es por eso también que los valores obtenidos son muy bajos.

Finalmente, de los resultados obtenidos se concluye que el algoritmo de recomendación más óptimo al realizar la recomendación basada en sesión sin utilizar las características de los ítems es SASRecF. Este obtiene mejores resultados en 3 de las 5 métricas utilizadas (Hit, Recall y Precisión). Sin embargo, los resultados no son muy diferentes a los que obtiene GRU4RecF por lo que este también es un buen algoritmo.

RQ3: ¿La incorporación de etiquetas contextuales mejora los resultados de las recomendaciones respecto a los obtenidos usando solo las secuencias de canciones?

Se sigue el mismo procedimiento anterior probando los distintos algoritmos pero esta vez se incluyen las etiquetas contextuales de las canciones. En la *imgHernarLopez*. 7 se observa, en primer lugar, que el algoritmo que peor se comporta es de nuevo FDSA, al igual que cuando se utilizan los ítems sin características. El algoritmo que obtiene mejores resultados en la mayoría de las métricas (3 de las 5) es GRU4RecF. El comportamiento de los valores se repite en cuanto a que los valores aumentan junto con el tamaño de la lista en todas las métricas salvo en precisión. Para contestar a la pregunta RQ3, deben compararse estos resultados con los obtenidos anteriormente. De esta manera se comparan las métricas obtenidas en la recomendación cuando se utilizan ítems sin características y cuando se utilizan ítems con etiquetas contextuales. Respecto a la métrica MRR, el valor más alto que se obtiene con los ítems sin características es $MRR@20 = 0,308$ y con etiquetas contextuales es $MRR@20 = 0,285$. Con respecto a *Hit Ratio* los valores más altos son $Hit@20 = 0,44$ y $Hit@20 = 0,283$, para el caso sin características y el caso con contexto respectivamente. De la misma manera se tienen $NDCG@20 = 0,334$ y $NDCG@20 = 0,319$; así como $Recall@20 = 0,44$ y $Recall@20 = 0,427$. Por último, los valores de precisión más altos para los dos casos son $Precisión@20 = 0,07$ y $Precisión@20 = 0,068$.

Lo que se observa también es que los resultados obtenidos utilizando ítems sin características son mejores que los obtenidos utilizando ítems con etiquetas contextuales.

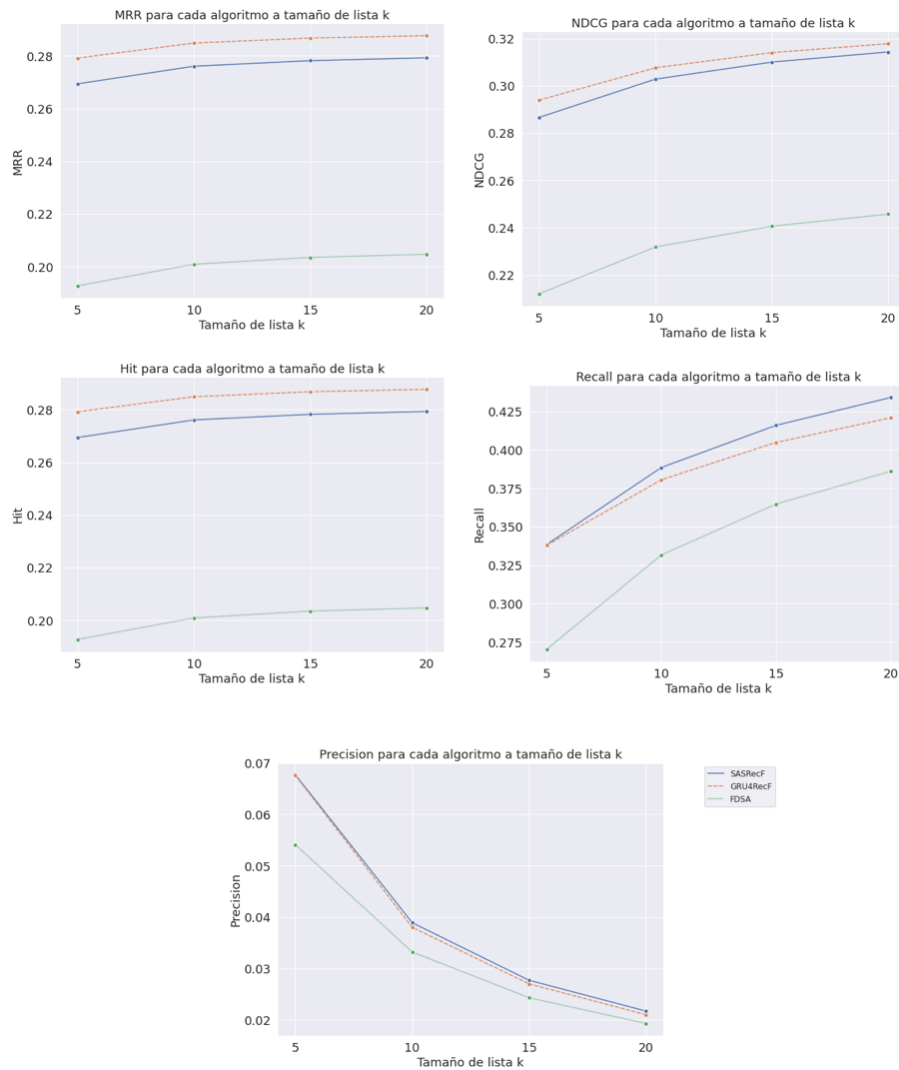


Figura 7: Resultados de las métricas para los ítems con etiquetas contextuales.

RQ4: ¿Cuál es la influencia de las características de los ítems frente a la influencia de las características contextuales?

Para responder a esta pregunta se obtienen las métricas para los tres algoritmos distintos con ítems con características musicales (atributos). Se representan en la misma gráfica que las métricas obtenidas en los dos casos anteriores para poder realizar una comparación. De esta gráfica la conclusión obtenida es que, en este caso de estudio, los modelos que mejor funcionan en la recomendación son

aquellos que no incluyen características de los ítems, seguidos por aquellos que incluyen las etiquetas contextuales y finalmente aquellos con las características de *Spotify*.

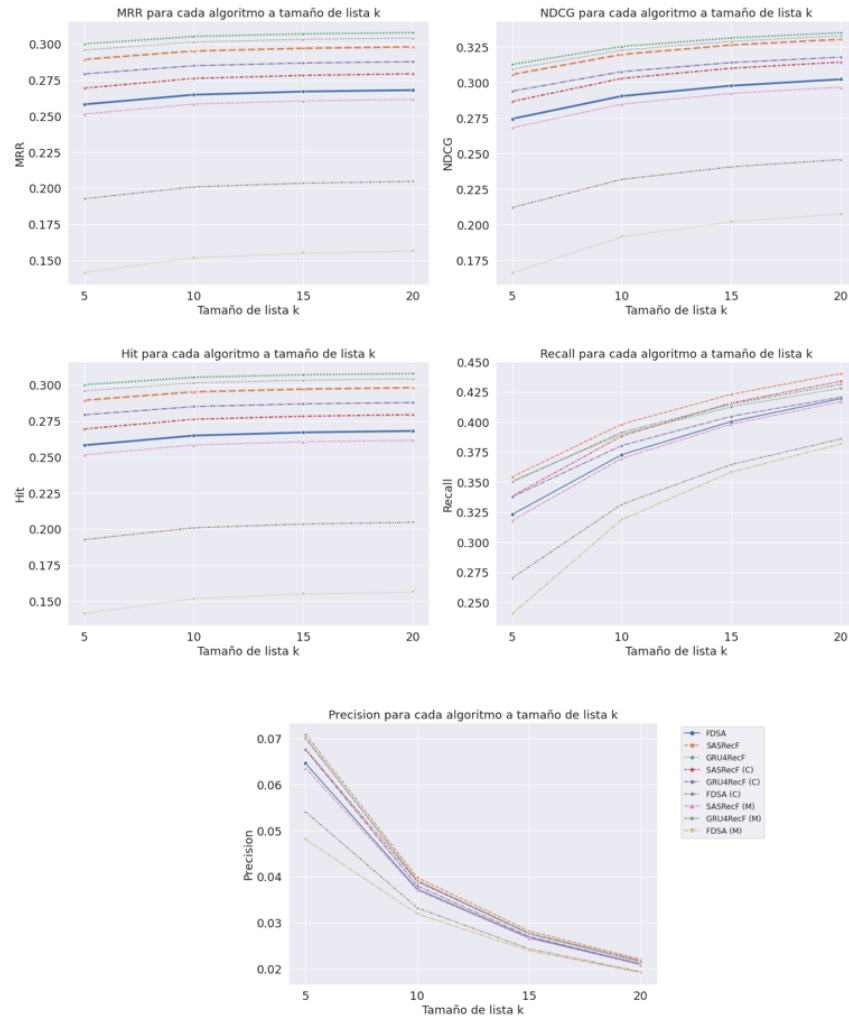


Figura 8: Resultados de las métricas en los tres casos.

6. Conclusiones

A continuación, se presentan las conclusiones a las que se han llegado tras finalizar este trabajo. Una vez descritos los fundamentos, se ha propuesto un sistema que infiere el contexto o contextos en los que se escuchan las canciones a partir de los atributos de dichas canciones mediante un modelo de clasificación multi-etiqueta. Posteriormente se utilizan las etiquetas contextuales para realizar recomendaciones secuenciales basadas en sesiones que sean adecuadas al contexto del usuario, el cual se infiere de forma latente a partir de las canciones que escucha en dicha sesión.

Finalmente, se ha planteado un caso de estudio para evaluar el sistema propuesto y dar respuesta a una serie de preguntas planteadas. De esta forma, los resultados permiten concluir en primer lugar que de los 13 algoritmos de clasificación multi-etiqueta probados, el que presenta mayor fiabilidad es *Binary Relevance-XG Boost*. En segundo lugar se ha encontrado que el mejor algoritmo de recomendación basado en sesión cuando se utilizan los ítems sin características es SASRecF frente a los otros dos algoritmos probados, FDSA y GRU4RecF, ya que obtiene mejores resultados en 3 de las 5 métricas utilizadas. Sin embargo, cabe destacar que la diferencia existente entre las métricas obtenidas por ese algoritmo y las obtenidas por GRU4RecF no presentan valores que difieran demasiado. Por tanto, ambos algoritmos podrían considerarse adecuados.

En relación a las pregunta 3, los resultados obtenidos han sido mejores en el caso de recomendar ítems sin características frente al caso de ítems con etiquetas contextuales, lo que contradice la hipótesis planteada en esta pregunta. Sin embargo, sería necesario realizar un mayor número de experimentos para afirmar este hecho de forma concluyente. Por último, los resultados referentes a la cuarta pregunta han demostrado que el uso de las características musicales de las canciones conducen a peores recomendaciones que las obtenidas utilizando etiquetas contextuales.

A la hora de interpretar los resultados deben tenerse en cuenta las limitaciones del estudio llevado a cabo. En primer lugar, para la recomendación solamente se ha utilizado un dataset con un número no muy elevado de usuarios, debido a la exigencia de recursos que plantea el experimento. Para poder obtener resultados más concluyentes sería necesario realizar un estudio con más datasets y de mayor tamaño.

6.1 Líneas futuras

En cuanto a líneas futuras, la primera propuesta, con el objetivo de mejorar, sería realizar los experimentos utilizando distintos datasets de manera que pudieran obtenerse más variedad de resultados de los que extraer conclusiones fuertes en cuanto a la inclusión del contexto. Dichos datasets, además, deberían ser más amplios y contar con un número mayor de muestras.

Además de aumentar el número y tamaño de los datasets utilizados en la validación de la propuesta, se podrían probar más algoritmos de recomendación lo cual aumentaría las posibilidades de encontrar aquel que resulta óptimo.

Por último, sería conveniente realizar experimentos empleando otras características que permitan inferir el contexto del usuario que puedan estar disponibles para datasets de canciones. Para ello sería necesario determinar que características, distintas de las que proporciona *Spotify*, podrían ser adecuadas. Una posibilidad sería utilizar otras características de fuentes distintas o incluso obtenidas de la señal en bruto de las canciones con técnicas del campo de *Music Information Retrieval*.

Referencias

- [1] Aggarwal, Charu C.: An introduction to recommender systems. Springer. (2016)
- [2] Ricci, Francesco and Rokach, Lior and Shapira, Bracha: Recommender systems handbook. Springer. (2011)
- [3] Villegas, Norha M and Müller, Hausi A.: Managing dynamic context to optimize smart interactions and services. The smart internet. Springer. (2010)
- [4] Adomavicius, Gediminas and Tuzhilin, Alexander: Context-Aware Recommender Systems. Recommender systems handbook. Springer. (2011)
- [5] Lozano Murciego, Álvaro and Jiménez-Bravo, Diego M and Valera Román, Adrián and De Paz Santana, Juan F and Moreno-García, María N: Context-aware recommender systems in the music domain: A systematic literature review. Electronics, 10. Multidisciplinary Digital Publishing Institute. (2021)
- [6] Wang, Shoujin and Cao, Longbing and Wang, Yan and Sheng, Quan Z and Orgun, Mehmet A and Lian, Defu: A survey on session-based recommender systems. ACM Computing Surveys (CSUR), 54. ACM New York, NY, USA. (2021)
- [7] Jannach, Dietmar and Quadrana, Massimo and Cremonesi, Paolo: Session-based recommender systems. Recommender Systems Handbook. Springer. (2022)
- [8] Spotify API Documentation. <https://developer.spotify.com/documentation/web-api/reference/operations/get-several-audio-features>. Accessed: 2022-06-1.
- [9] Ibrahim, Karim M and Royo-Letelier, Jimena and Epure, Elena V and Peeters, Geoffroy and Richard, Gael: Audio-based auto-tagging with contextual tags for music. ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE. (2020)
- [10] Pichl, Martin and Zangerle, Eva and Specht, Günther: Towards a context-aware music recommendation approach: What is hidden in the playlist name?. 2015 IEEE international conference on data mining workshop (ICDMW). IEEE. (2015)
- [11] Contextual Tags for music auto-tagging. <https://zenodo.org/record/3648287>. Accessed: 2022-06-1. (2022)
- [12] Szymański, P. and Kajdanowicz, T.: A scikit-based Python environment for performing multi-label classification. ArXiv e-prints. (2017)
- [13] Zhao, Wayne Xin and Mu, Shanlei and Hou, Yupeng and Lin, Zihan and Chen, Yushuo and Pan, Xingyu and Li, Kaiyuan and Lu, Yujie and Wang, Hui and Tian, Changxin and others: Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. Proceedings of the 30th ACM International Conference on Information & Knowledge Management. (2021)
- [14] Zhang, Tingting and Zhao, Pengpeng and Liu, Yanchi and Sheng, Victor S and Xu, Jiajie and Wang, Deqing and Liu, Guanfeng and Zhou, Xiaofang: Feature-level Deeper Self-Attention Network for Sequential Recommendation. IJCAI. (2019)

- [15] Hidasi, Balázs and Quadrana, Massimo and Karatzoglou, Alexandros and Tikk, Domonkos: Parallel recurrent neural network architectures for feature-rich session-based recommendations. Proceedings of the 10th ACM conference on recommender systems. (2016)
- [16] Kang, Wang-Cheng and McAuley, Julian: Self-attentive sequential recommendation. 2018 IEEE International Conference on Data Mining (ICDM). (2018)

Sistema Inteligente para el Etiquetado de Texto

Iria Ares Torres, Belén Pérez Lancho y María Navarro Cáceres

Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los
Caídos s/n. 37008, Salamanca, España
{iria.ares,lancho,maria90}@usal.es

Resumen Este estudio demuestra el valor y la utilidad del Procesamiento del Lenguaje Natural (NLP) en la detección de estereotipos de género y discriminación en las canciones infantiles tradicionales españolas. Mediante el análisis morfosintáctico, el uso de NLP permitió visualizar la prevalencia de adjetivos masculinos y la falta de representación de los femeninos en las letras. Además, el análisis del sentimiento y el topic modeling revelaron una temática basada en estereotipos arraigados en la sociedad. Estos resultados enfatizan la importancia del NLP como una herramienta efectiva para comprender y denunciar la falta de diversidad en las letras de las canciones infantiles, impulsando así la necesidad de promover una educación inclusiva y libre de estereotipos desde una edad temprana

Keywords: NLP, traditional children's songs, gender bias

1. Introducción

En este proyecto recoge la elaboración y demostración de la detección del sesgo de género presente en las canciones infantiles tradicionales en España, a través de herramientas de procesamiento de lenguaje natural (Natural Language Processing).

Las canciones tradicionales infantiles son un estilo de música vocal, y son uno de los medios de entretenimiento y enseñanza musical más antiguos. En sus letra se reflejan situaciones cotidianas y comunes, como irse a dormir, jugar, realizar tareas de casa, etc, ya que son canciones populares. Las letras están acompañadas de ritmos y rimas fáciles de recordar, ya que la finales es entretener a los niños y niñas de forma lúdica y sencilla. Estas canciones son un medio de transmisión de costumbres y valores culturales, por lo que pueden influir significativamente en la formación de la perspectiva de género de los niños y niñas, en la construcción de la identidad y en la perpetuación de desigualdades de género en la sociedad.

Se han llevado a cabo un constante y exhaustivo análisis de aspectos lingüísticos y sociales de estas obras, analizando las estructuras, temáticas y sesgos presentes en las canciones, con la complicación de la cantidad elevada de material a evaluar.

Hoy en día, gracias a la capacidad computacional de la inteligencia artificial y machine learning para la interpretación de texto natural, pueden llevar a cabo

este tipo de estudios de manera eficiente, precisa y automática, pudiendo analizar gran cantidad de datos, como se detallará en la revisión bibliográfica.

El uso de técnicas de NLP permite una comprensión automática del texto, así como descomponer y examinar las letras de las canciones, identificando patrones morfosintácticos [73] y semánticos [75].

La comprensión semántica del texto facilita la detección de posibles sesgos de género o contenido misógino. Con ese objetivo, se han propuesto modelos capaces de detectar y clasificar conceptos sexistas en redes sociales, como los desarrollados en [77], y en [74].

En la literatura de sesgo, se han aplicado algunas técnicas como análisis de sentimiento [3] o detección de temáticas [75] o [76], donde emplean la clasificación del sentimiento y detección de temáticas en textos y para visualizar los temas más relevantes en textos de carácter misógino para encontrar las relaciones semánticas entre ellos de forma precisa.

El análisis del sentimiento ayuda a comprender la carga emocional asociada con las letras, y si estas emociones refuerzan o desafían estereotipos de género. Asimismo, la detección de temáticas relevantes permite identificar los temas recurrentes en las canciones y analizar cómo se abordan desde una perspectiva de género.

En resumen, para llevar a cabo este trabajo se ha partido de la colección de canciones tradicionales infantiles y estudios acerca de la perspectiva de género que podrían proyectar en los niños y niñas, y se ha planteado cómo podrían aplicarse técnicas de inteligencia artificial, machine learning, minería de datos y texto, para extraer de forma sencilla e intuitiva información de ellas, evaluando las diferentes herramientas y librerías disponibles en Python, y valorando la calidad de los resultados así como las técnicas de visualización a emplear.

En el resto del documento figura una sección que recoge la revisión bibliográfica, una sección de metodología que recoge el diseño de cada uno de los pasos, una sección de evaluación de los resultados de cada uno de los apartados realizados, y una sección final con conclusión y trabajo a futuro.

2. Objetivos e Hipótesis

La motivación de este trabajo se basa en como el uso de herramientas propias de la Inteligencia Artificial pueden contribuir a estudios sociales y lingüísticos, como es la determinación de sesgos de género que pueden estar presentes en letras de canciones tradicionales infantiles españolas.

Por ello, el objetivo principal es analizar las letras de las canciones infantiles y detectar la presencia de sesgo de género en ellas. Para lograrlo, se plantean los siguientes objetivos específicos:

- O1: Identificar patrones de uso de categorías gramaticales como los adjetivos, y analizar su utilización en función del género gramatical de la palabra, para conocer cómo se describen los sustantivos femeninos y masculinos.
- O2: Analizar la carga emocional de las letras de las canciones y evaluar si el uso de ciertas palabras está asociado a una polaridad concreta.

- O3: Categorizar las canciones según la temática con la que están relacionadas y evaluar si fomentan estereotipos de género.
- O4: Demostrar que este tipo de análisis puede desarrollarse de forma sencilla, y pueden mejorar la metodología de trabajos de ámbito sociolingüístico.

A continuación, se plantean las siguientes hipótesis de las que parte el estudio, inspiradas por la revisión de la literatura de estudios similares:

- H1: Las letras de las canciones infantiles tradicionales presentarán patrones morfosintácticos que refuerzan estereotipos de género, como roles y características atribuidas de manera desigual a hombres y mujeres. [7]
- H2: Las canciones infantiles tradicionales transmitirán una carga emocional que estará asociada a estereotipos de género, con una mayor tendencia a destacar cualidades y características específicas según el género. [11]
- H3: Los temas recurrentes en las letras de las canciones infantiles tradicionales estarán desequilibrados en cuanto a la representación de género, con una mayor presencia de temas asociados a estereotipos de género tradicionales. [12]

Para demostrar las hipótesis planteadas, se desarrollan las etapas y técnicas aplicadas en la sección correspondiente a la metodología 8

3. Revisión Bibliográfica

En esta sección se detallarán los estudios y trabajos publicados hasta la fecha de los conceptos clave de este trabajo, como son la perspectiva de género y el uso del procesamiento de lenguaje natural.

3.1 Estudios de Perspectiva de Género

En la literatura, están presentes estudios que denuncian la presencia de sesgos de género en textos literarios tradicionales en todas las partes del mundo, enfocadas en analizar el contexto que consumen los niños y niñas, en su educación [56]. En sus estudios se refleja la baja representación de personajes femeninos en ese conjunto de libros y su atribución a personalidades más débiles que los personajes masculinos, y concluyendo que las presentes diferencias en el género son una fuerte influencia en la percepción de los niños y niñas en la forma en la que los hombres y mujeres deben actuar, pensar y comportarse. En España, el uso del género gramatical en el español ha sido motivo de estudio sobre el impacto psicolingüístico que este tiene [6], ya que en la forma de expresar y cualificar están impresos los estereotipos de géneros propios de las personas hablantes.

En España, se han llevado a cabo diversos estudios sobre el cancionero español. Como se refleja en el estudio “Música y género: análisis de una muestra de canciones populares” elaborado por E. Berrocal de Luna y J. Gutiérrez Pérez [7] donde denuncian la vaga representación de mujeres en el cancionero seleccionado, tras realizar un análisis morfosintáctico del género de palabras como

adjetivos y verbos. Los autores señalan así mismo la importancia de la propia sensibilidad de la persona que realiza el estudio frente a la discriminación por género, limitación que puede ser transcendental cuando el análisis es realizado por modelos inteligentes de lenguaje que han sido entrenados para ello.

Años más tarde, Diego Pascual y Cabo construye una continuación del análisis de [11], donde realiza una evaluación cualitativa y cuantitativa de la representación del género femenino en canciones de cuna tradicionales españolas. En su estudio, analiza el papel desempeñado por sujetos femeninos y masculinos, evaluando si están representados por de forma activa o pasiva, concluyendo, por una parte, la mínima representación de las mujeres, y segundo, la presencia pasiva de las mujeres, en actividades relacionadas con el servicio al hombre, el hogar, o la cría de hijos o hijas, frente a la mayor asignación masculina de acciones o representaciones activas.

3.2 Estudios sobre Procesamiento de Lenguaje Natural

Los estudios contemplados en la sección anterior han sido llevados a cabo por expertos en lingüística y otras ramas como sociología o psicología. En ellos, el contenido analizado, tanto textos literarios como imágenes, fue estudiado e interpretado por expertos en la materia de manera manual. Esto quiere decir que se analizaron una a una todas las letras para extraer de ellas características gramaticales o semánticas relevantes para poder conocer cómo se representan las diferencias de ellas en género.

Hoy en día, el Procesamiento del Lenguaje Natural (NLP) es una rama de la Inteligencia Artificial que se basa en un conjunto de técnicas computacionales que permiten analizar y representar textos naturales en uno o más niveles de análisis lingüístico con el fin de lograr un procesamiento del lenguaje similar al humano. Estas técnicas permiten realizar análisis morfológicos, sintácticos, semánticos y léxicos de un texto dado.[13]

Con el paso del tiempo y el exponencial aumento la capacidad computacional de los equipos, se han desarrollado una serie de modelos y herramientas que permiten desarrollar tareas basadas en el lenguaje natural, como la identificación, comprensión y generación de textos, mediante modelos basados en probabilidad como Naive Bayes o Hidden Márkov Models [14], Redes Neuronales Profundas como las Redes Neuronales Convolucionales [10], o Redes Neuronales Recurrentes [17], obteniendo con el paso del tiempo modelos cada vez más precisos en identificación, comprensión y generación de textos [14]

En los últimos años, uno de los modelos más empleados en el NLP es el modelo BERT (Bidirectional Encoder Representations from Transformers) [18] el cual emplea una arquitectura única para desarrollar diferentes tareas de NLP, basada en encoder tipo Transformador bidireccional multi-capas. Esta arquitectura será explicada en la Metodología.

A partir de la arquitectura de BERT, se han diseñado modelos paralelos como RoBERTa [19] [20], un modelo más robusto, o modelos entrenados con corpus en español, como RoBERTuito, entrenado con tweets en español [21] o BETO[25], otro modelo basado en BERT entrenado con una gran cantidad de

datos en español, de tamaño similar a los datos de entrenamiento de BERT. En este trabajo se empleará la librería Pysentimiento [69], que integra estos dos últimos modelos Transformer en español para llevar a cabo tareas de NLP en este idioma.

Dentro del NLP, se han empleado estos modelos mencionados en diferentes estudios relacionados con la comprensión del texto natural, análisis o generación. Una de las tareas más comunes es el etiquetado de texto, tanto de análisis morfológico ([52]) como reconocimiento de entidades como nombres de personas o corporaciones, localizaciones, etc ([51]). Otra tarea es la comprensión del contexto, lo cual permite desarrollar tareas como la clasificación del sentimiento, basado en determinar y clasificar las emociones o actitudes expresadas en un texto, ya sea positivas, negativas o neutras [23] y el topic modeling, que permite descubrir y extraer automáticamente temas o temáticas latentes presentes en un conjunto de documentos, facilitando así la comprensión y organización de la información contenida en los textos. ([22])

Sin embargo, varios estudios ([45], [47], [49]) han detectado como los propios modelos empleados en NLP cuentan con un sesgo interiorizado, al haber sido entrenados con conjuntos de datos generados por una sociedad donde los estereotipos y la discriminación de género están arraigados, presentes en textos sobre noticias, opiniones o contenido de redes sociales, fuentes con las que se suelen entrenar los modelos. Fuentes como [45] demuestran cómo esto está presente en las “word embeddings”, término que describe la conversión de palabras a vectores de alta dimensionalidad, y construye la relación semántica entre dos palabras a partir de la relación entre sus respectivos vectores. Bolukbasi et al. definen cómo los sesgos sociales están presentes en las asociaciones vectoriales generadas por modelos como word2vec (desarrollado en [46]) por ejemplo comparando qué ocupaciones están más relacionadas con la palabra “él” y cuales con “ella” asociando con el género masculino ocupaciones profesionales más populares y asignando al género femenino aquellas más relacionadas con la familia y el cuidado, reflejando los sesgos presentes en la sociedad. Proponen métodos para mitigar este sesgo, como la identificación de palabras de género neutro, y la igualación de las palabras neutras a ambos géneros.

Por ejemplo, en [44] se analiza esta característica propia del modelo BERT, analizando cómo realiza la asignación de género gramatical en inglés y alemán a sustantivos relacionados con profesiones. En su estudio, demuestran el sesgo propio del modelo BERT pre-entrenado, y proponen su mitigación entrenando el modelo BERT con un corpus de asociaciones pronombre-sustantivo donde se realiza un intercambio de género en los pronombres. De esta forma, la asociación a una ocupación profesional se muestre tanto con pronombre masculino como pronombre femenino, acción que reducirá el sesgo propio de BERT y generará frases con menor sesgo de género.

Un ejemplo de ello se desarrolla en [41] donde se evalúan como se asignan los género gramaticales a los sustantivos en las traducciones de inglés a español mediante Google Translate. Se demuestra la baja representación de género femenino al traducir sustantivos referentes a ocupaciones así como “I am a doctor - Soy un doctor” y “I am a nurse - Soy una enfermera”. Un estudio similar se

ha llevado a cabo en [42] donde proponen la necesidad de aplicar técnicas de eliminación de sesgo o “de-biasing” en modelos de NLP.

En [50] se realiza un estudio más desarrollado sobre la detección de sesgo de género en letras de canciones, incluyendo la mitigación del sesgo de género en el modelo de NLP empleado, demostrando como la discriminación de género está presente en las letras de canciones cada vez con mayor frecuencia, generalmente en aquellas producidas por artistas masculinos.

Otra limitación propia del NLP es la gran magnitud de conjunto de datos disponibles en inglés, en comparación con otros idiomas. De esta forma, para emplear los modelos más empleados de NLP, es necesario recurrir a la traducción al inglés del texto en idioma original. En ese proceso se pierde gran cantidad de información o contexto, según se muestra en [23] o [37], donde se demuestra cómo se pierde el tono de ironía en la traducción de español al inglés para realizar clasificación de sentimientos en textos en español.

Estudios como [9], [54] y [57] proponen el desarrollo de un léxico propio en español adaptado al corpus con el que se trabaja, aporta resultados mejores en la clasificación de sentimiento que el uso de modelos entrenados únicamente con textos en inglés.

3.3 Aplicaciones del NLP en la Extracción de Información de Letras de Canciones

Partiendo de un conjunto de letras de canciones extraídas de canciones populares infantiles, el objetivo es extraer información sobre ellas mediante el uso de herramientas de NLP para detectar patrones o características que muestren el sesgo de género presente en las letras.

Por ello, se elabora un análisis morfosintáctico de las letras de las canciones, de la misma forma que se realizó en los estudios [7] y [11] para analizar el uso de adjetivos en función del género del sujeto, y comprobar si las conclusiones son similares.

En [8] se propone uno de los métodos de análisis de letras de canciones es la clasificación de emociones o clasificación de sentimiento, ya que son factores que tienen una gran influencia sobre las emociones de las personas que escuchan las canciones, en el caso de canciones tradicionales infantiles, serán los niños y niñas. Además, en [48] se muestra que el análisis del sentimiento es de las aplicaciones de NLP que menor grado de discriminación tiene, ya que los modelos tienden a clasificar el texto en función de estereotipos del corpus con el que fueron entrenados, pero no perpetúan la denigración de un género o la menor representación del género femenino como otras herramientas de NLP como es el word embedding.

Otra información clave que se puede extraer de las canciones, es la identificación de las temáticas principales que tratan, como se propone en [55]. Un estudio similar se ha llevado a cabo en [12] donde ha analizado los estereotipos de género presentes en los temas principales de canciones tradicionales infantiles de España y México. Por ello, se pretenderá hacer el mismo procedimiento

empleando Topic Modeling al conjunto de datos de canciones infantiles con las que se trabaja.

Las librerías más empleadas en este ámbito de la Inteligencia artificial con Python son NLTK [61] y [60], y modelos enfocados al NLP de textos en español generado por instituciones nacionales, como Freeling [62] y [63], pero su implementación en Python es compleja ya que fueron diseñados con otros lenguajes de programación.

Para llevar a cabo el estudio, se siguieron las pautas recogidas en la sección siguiente.

4. Metodología

El objetivo principal de este estudio es analizar las letras de las canciones infantiles para determinar si existe un sesgo de género en ellas. Para ello, se utilizará el etiquetado morfosintáctico para identificar patrones lingüísticos y comprender la estructura de las frases y oraciones presentes en cada canción. Además, se aplicará una clasificación de sentimiento para evaluar las emociones transmitidas en las letras y se identificarán las temáticas predominantes en las canciones. Estas tareas nos permitirán evaluar el grado de sesgo de género que se presenta en las letras de las canciones infantiles y obtener una comprensión más profunda sobre la influencia que pueden tener en la percepción de roles y estereotipos de género desde una edad temprana

4.1 Conjuntos de Datos Utilizados: Letras de Canciones Tradicionales Infantiles Españolas

Se trata de un conjunto de 862 canciones con temática de canciones infantiles tradicionales, en español, italiano y portugués, de las que se recogen parte de las estrofas o estribillos más populares, redactadas en prosa, y una columna que contiene anotaciones sobre la persona que las ha dictado, baile con el que se acompaña, si forman parte de algún cancionero específico, comentarios acerca de la música, etc.

Un ejemplo de canción es: *El patio de mi casa es particular cuando llueve se moja como los de más agáchate y vuélvete agachar que las agachaditas saben bailar hachei jotaka, ele eme eñe a que si usted no me quiere otro amante me querrá hachei jotaka ele eme eñe a que si usted no me quiere otro amante me querrá.*

Otro ejemplo de canción popular es: *El cocherito, leré, me dijo a noche, leré, que si quería, leré, montar en coche, leré; y yo le dije, leré, con gran salero, leré, no quiero coche, leré, que me mareo, leré.*

Los textos que forman las letras de las canciones son transcripciones literales de los versos cantados, por lo que son frases con características lírica y melódicas, como la presencia de onomatopeyas, la repetición de palabras o frases para acompañar al ritmo, así como frases con poco contexto al ser únicamente la parte más conocida de la canción.

El conjunto de todas las canciones ha sido recopilado de las siguientes fuentes, mostradas en la Tabla 1.

Título	Referencia
Cancionero Musical Popular Español (4 tomos)	[1]
Folk music and poetry of Spain and Portugal	[2]
Cancionero gallego	[4]
Cancionero musical de la lírica popular asturiana	[26]
Cancionero popular de la provincia de Santander	[27]
Cancionero popular de Burgos	[28]
Colección de cantos populares burgaleses	[29]
Las mil y una canciones populares de la región leonesa	[30]
Nuevo Cancionero Salmantino	[31]
Cancionero Salmantino	[32]
Lírica popular de la alta Extremadura	[34]
Cancionero popular de Burgos	[33]
Cancionero Musical de Córdoba	[40]
Colección de cantos populares de Murcia	[35]
Cançons Populares Catalanes	[36]
Cançoner del Calic	[38]
Obra del Cançoner Popular de Catalunya	[39]

Tabla 1: Títulos de colecciones de canciones tradicionales infantiles españolas y sus referencias

4.2 Análisis de Librerías NLP en Python

Para llevar a cabo los diversos análisis realizados, se han analizado diferentes librerías disponibles para entorno Python para análisis de texto. Una de las herramientas más empleadas, NLTK [61] NLTK no dispone de modelos entrenados con datos en español para llevar a cabo esta tarea. Sin embargo, Spacy dispone de 4 modelos diferentes pre-entrenados elaborar tareas de análisis morfosintáctico, NER (etiquetado de entidades) y lematización en español, entre otras tareas. Se compara la lematización empleando los tres modelos de Spacy.

Los resultados se comparan con otra librería diferente, llamada Stanza [53] generada por StandfordNLP, otra de las herramientas de NLP con mayor uso global. Este modelo tiene implementación de lematización en castellano, y se ha probado la lematización con las mismas palabras que Spacy. No obstante, se han obtenido resultados poco precisos, similares a los obtenidos en el resto de pruebas de lematización. Estos resultados de lematización de los cuatro modelos son evaluados en la sección de resultados.

Finalmente, tras valorar las herramientas disponibles, se empleó el modelo de Spacy “es_dep_news_trf”, ya que es el modelo entrenado en español que elabora la lematización con mayor precisión. Permite implementar una pipeline o cadena de procesos que incorpora diferentes herramientas de NLP, empleando el

modelo BETO basado en Transformer, detallado en ¹, por lo que es un modelo especialmente diseñado para el análisis y comprensión del texto.

4.3 Preprocesamiento del Texto de las Canciones

El paso previo a aplicar modelos de análisis de texto, consiste en limpiarlo y preprocesarlo, de forma que los datos de texto con los que se trabajan sean uniformes, en el mismo idioma, compuesto por caracteres alfabéticos y eliminando palabras repetidas o palabras vacías.

Se ha generado un conjunto de palabras vacías específicas del corpus que representan sílabas u onomatopeyas melódicas como “la, leré, ay, ea, pun, ro”, las cuales no portan información al texto. Además, se utiliza un conjunto de palabras vacías en español obtenidas de [67], un paquete más elaborado que el paquete de palabras vacías en español de NLTK, ya que tiene el doble de palabras, y elimina del texto palabras innecesarias para el análisis.

De esta forma, se puede obtener un conjunto simplificado y uniforme, lo cuál mejora los resultados de las etapas posteriores del desarrollo.

4.4 Etiquetado Morfológico y Gramatical

Partiendo del dataset de canciones, se procede a realizar un etiquetado morfosintáctico, el cual permite identificar a qué categoría gramatical pertenece cada una de las palabras que forman las letras de las canciones, para identificar palabras clave como los adjetivos más frecuentes. Este proceso forma parte del proceso POS Tagging (part of sentence tagging) etiquetado de parte de la oración, herramienta común del análisis de lenguaje natural. Simultáneamente, se extrae el género gramatical de cada palabra, determinado como masculino, femenino o invariable.

4.5 Análisis del Sentimiento

El análisis y clasificación del sentimiento es un enfoque del NLP que permite conocer el tono que transmite un texto, si representa emociones positivas, negativas o neutrales. [23] [15]

En la literatura, está presente la falta de modelos de lenguaje ajustados al español, ya que muchos modelos, sobre todo en Python se basan en el uso de clasificadores entrenados con corpus en inglés, y los resultados aportan una precisión mejorable, ya que se pierde mucha información y contexto semántico al traducir los textos al inglés previamente.

Se han desarrollado una serie de modelos Transformer muy populares por su alta eficiencia en tareas de como la clasificación o generación de texto, como GPT, BART, BERT, DistilBERT, pero todos ellos han sido entrenados con corpus en inglés por lo que su uso se limita a textos en ese idioma [24]. Por ello, el uso de modelos Transformer que integren clasificación del sentimiento

¹ https://huggingface.co/spacy/es_dep_newstnf

adaptados al español es limitada, y se ha escogido el uso de Pysentimiento ya que emplea los modelos RoBERTuito, (entrenado con 500 millones de tweets en español y su precisión es máxima al evaluar textos procedentes de Twitter) y BETO, entrenado con una gran cantidad de textos en español, para la tarea de clasificación del sentimiento de las canciones en español.

Para cada texto de cada canción analizado, el modelo de pysentimiento devuelve la probabilidad entre 0 y 1 de que ese texto pertenezca a cada grupo sentimental, positivo, negativo o neutral, en función de las palabras que componen el texto. Finalmente, asigna la probabilidad mayor a 0.5 al texto analizado.

Por otra parte, el modelo VADER (Valence Aware Dictionary and sEntiment Reasoner) [72] es un modelo de NLP basado en léxico y en reglas, que ha sido entrenado y ajustado para detectar los sentimientos de texto de redes sociales, en inglés, entrenado para clasificar tweets como positivos, negativos, neutrales o compuestos. Se emplea una versión del modelo que incluye la traducción al inglés del texto original, denominada VADER Sentiment Analysis Multilanguage disponible en ².

El texto de entrada del modelo son las letras de las canciones sin onomatopéyas musicales pero incluyendo las palabras vacías del español para mantener las estructuras sintácticas de las frases a analizar y palabras emocionalmente relevantes como “no” o “nunca”. El resultado de usar el modelo de clasificación del sentimiento de pysentimiento, en el conjunto de canciones determinará la polaridad de la emoción o contexto que evoca el texto, si es positivo, negativo o neutral, generando tres grupos diferenciados de canciones. Esta clasificación del sentimiento permite conocer la emoción que evocan las canciones y que se pretenden transmitir a los niños y niñas pequeños, y conocer qué emociones les puede hacer sentir el escucharlas y cantarlas.

Además, se visualizará en qué categoría se emplean más palabra clave como niño o niña, para conocer si hay alguna diferencia en el uso de palabras que definen personajes femeninos o masculinos, igual que se analiza en [7].

4.6 Topic Modeling: Detección de Temáticas

Otra herramienta de Procesamiento de Lenguaje Natural que se aplicará a las canciones tradicionales infantiles es el Topic Modeling. El estudio desarrollado en [8] proponen el “Topic Modeling” como una técnica óptima de análisis de letras de canciones, ya que generalmente se parte de conjunto de datos de letras sin etiquetas, y el topic modeling es un tipo de herramienta que emplea aprendizaje no supervisado. Por ello, es una tarea idónea para aplicar al conjunto de letras de canciones tradicionales infantiles que se proponen en este trabajo, ya que no disponen de etiqueta de temática a la que pertenecen. Los resultados serán comparados con los obtenidos en [12] sobre las temáticas más frecuentes en canciones tradicionales infantiles y el sesgo de género que presentan.

La librería gensim de Python es la más empleada en este tipo de análisis [65], ya que permite implementar tareas de procesamiento de texto para topic modeling de forma simple e intuitiva.

² <https://github.com/brunneis/vader-multi/tree/master>

La mayoría de modelos de topic modeling se basan en la frecuencia de las palabras en un conjunto de textos. Para ello, se debe transformar las palabras en un concepto numérico, a través de un diccionario que relaciona cada palabra, con un identificador.

A continuación, se genera el corpus, donde se almacena, por cada documento, en este caso por cada canción, la relación del identificador de cada palabra con el número de veces que aparece en la misma canción. De esta forma, se conoce cuáles son las palabras más frecuentes en cada canción.

Por último, se implementa un modelo de detección de temáticas o topics. Los modelos de detección de temáticas pueden centrarse a nivel de frases, o a nivel de documentos. En este caso, las letras de las canciones pueden contener frases complejas o varias dentro de una misma estrofa, por lo que cada una de las canciones se considerará un documento.

El objetivo de obtener las temáticas más frecuentes en las canciones, es el identificar cuales son los conceptos que se transmitían a los niños y niñas con estas canciones, y evaluar si forman parte de estereotipos propios de la discriminación de género, comparando los resultados con el trabajo llevado a cabo en [12] pero empleando NLP para el mismo propósito.

Para ello, se ha empleado un modelo LDA (latent dirichlet allocation). El algoritmo LDA es un modelo probabilístico de aprendizaje no supervisado [79], que a partir de las frecuencias de palabras recogidas en el corpus, segmenta los datos a tres niveles: palabra, temática y documento. Dado un número de topics fijado, el algoritmo calcula la distribución a priori de Dirichlet (un tipo de distribución de probabilidad) de cada tema y genera un vector que representa la mezcla de las probabilidades de los topics. A nivel de palabra, genera una matriz beta con la que parametriza las probabilidades de cada palabra, calculando así la distribución de temas para cada documento y la distribución de palabras para cada tema. El algoritmo itera y se actualizan las probabilidades estimadas durante el entrenamiento, donde se puede configurar el número documentos que se usan, y el número de veces que se analiza todo el corpus. Cuando finaliza el entrenamiento, se definen los diferentes topics según la probabilidad de que una palabra pertenezca a un tema en particular y la probabilidad de que un documento contenga un tema en particular, pudiendo visualizar fácilmente mediante el diccionario, qué palabras pertenecen a cada tema. El desarrollo matemático del algoritmo se puede consultar en los estudios de [79] y [58].

Un parámetro característico en la definición de un modelo de LDA es el alfa, con el que se indica la relación entre los documentos y temáticas a priori. Este parámetro puede tener una distribución simétrica, equivalente en todos los diferentes topics, o asimétrica, de forma que este valor varía en función del topic. En el estudio de [70] se analiza la importancia y diferencia de este valor, y se visualiza el impacto que tiene el variar el parámetro alfa en el modelo empleado. Para valorar qué modelo es mejor, se emplea la medida de perplejidad, que evalúa cuanto de similar es una palabra nueva a un topic dado usando parte del corpus para evaluar. El uso de esta medida para determinar la validez de un modelo se define en la documentación de gensim [65] y estudios en la literatura [43].

Tras parametrizar el modelo, se emplea para visualizar los temas de todas las canciones, se visualizará las temáticas más frecuentes en las canciones de cada clasificación del sentimiento, de las canciones positivas, negativas y neutrales, para conocer si hay distinción de temáticas entre ellas.

Están han sido las pautas seguidas en la elaboración del estudio. A continuación, en la siguiente sección, se detallarán el desarrollo y resultados de cada una de las fases, seguido de una discusión de estos.

5. Evaluación de Resultados

En este apartado se mostrarán los resultados obtenidos en los scripts de Python, y se detallarán las características con las que se aplican los modelos empleados. Se realizará discusión de los resultados obtenidos para valorar si satisfacen los objetivos propuestos en este trabajo, enfocados en detectar sesgo de género en letras de canciones tradicionales infantiles:

- A través del etiquetado morfológico, identificar patrones de uso de adjetivos según el género gramatical.
- Analizar la carga emocional de las canciones y la representación de figuras como "niño." "niña" dentro de cada categoría emocional.
- Extraer las temáticas más comunes en las letras para valorar si fomentan estereotipos de género.

A continuación, se detallarán cada una de las etapas seguidas en el trabajo.

5.1 Preprocesamiento del Texto de las Canciones

Se parte del conjunto de datos denominado "lyrics_raw" el cuál contiene 862 canciones tradicionales infantiles en español, catalán, italiano y portugués. El conjunto de datos tiene las siguientes columnas:

- "song" : contiene el identificador del número de la canción. Está compuesto por dos letras que contienen el idioma, el año de la canción, y una serie de letras y números identificativas.
- "in_song_lyrics": contiene la letra de la parte de la canción. Generalmente son simplemente las partes más conocidas de cada canción, siendo estribillo o parte de alguna estrofa
- "annotated_lyrics": incluye alguna anotación sobre la canción en concreto.

Tras eliminar las entradas vacías o con identificadores de idioma diferentes al español, se trabajará con 728 canciones.

5.1.1 Identificación y Filtrado de Idioma Aunque las canciones fueron inicialmente catalogadas por su idioma, dentro de las identificadas como español, se encuentran un cierto número de canciones en idiomas diferentes al español. Se procede a la identificación manual del idioma y a filtrar únicamente las canciones

en español, con el fin de partir de un conjunto de datos uniforme, teniendo en cuenta que los modelos que se emplearán han sido diseñados para trabajar en español. Para ello, se usa la librería langdetect [66]. Esta herramienta detecta los siguientes idiomas dentro del grupo de canciones en español:

Idioma detectado	Número de canciones
es	679
ca	20
it	10
pt	8
Otros idiomas	11
Total	728

Tabla 2: Número de canciones según el idioma identificado

Se valora si las canciones catalogadas como “otros idiomas” son realmente en español y se modifica la etiqueta. El nuevo conjunto de canciones categorizadas en español contiene 685 canciones.

5.1.2 Tokenización y Limpieza de Palabras Vacías Para realizar la tokenización del texto (identificación de cada palabra individualmente) y las siguientes tareas de procesamiento de texto, se emplea la librería de Spacy, y en concreto el modelo “es_dep_news_trf”, ya que emplea un modelo Transformer entrenado en español muy eficiente. Se tokeniza cada palabra, eliminando caracteres no alfabéticos y palabras repetidas, y convirtiendo los tokens a minúscula. El proceso de tokenización es útil en el NLP ya que permite identificar palabra a palabra de una frase y encontrar la relación entre diferentes tokens es más efectivo que procesar frase a frase. [14]

La siguiente tarea a realizar en el preprocesado de texto, es la eliminación de palabras vacías o “stop words”.

Se eliminan las palabras, sílabas u onomatopeyas cuya única función es acompañar a la melodía, por ejemplo: “la, leré, ay, ea, pun, ro” etc. Además, es conveniente eliminar las palabras vacías que carecen de significado semántico relevante en las frases, para reducir el tamaño de palabras a procesar. Para ello, se eliminan las palabras que forman parte del grupo de stopwords obtenido de [67].

Una vez se ha preprocesado el texto con Spacy, se pueden llevar a cabo tareas de análisis de forma más eficiente. Se detallarán en la siguiente sección.

5.2 Análisis Morfosintáctico y Lematización

El uso de herramientas de análisis de texto natural permite extraer información sobre el contenido del texto. Partiendo de un conjunto de datos como son las letras de canciones infantiles, y con el objetivo de detectar y analizar la existencia

de sesgo de género escondido en el texto, se propone un análisis morfosintáctico de las frases que componen las canciones. El etiquetado morfológico (Part-Of-Speech Tagging, POS Tagging) permite comprender la estructura gramatical y la función de las palabras en una frase, y su finalidad es el localizar y analizar el género gramatical de algunas palabras como los adjetivos o sustantivos empleados en la prosa de las canciones tradicionales infantiles.

Partiendo de los textos preprocesados, se extrajo la categoría gramatical (la etiqueta de Part-Of-Speech) de cada una de las palabras y se obtuvieron los lemas de las palabras que forman las frases, herramienta del procesamiento de texto denominada lematización. El objetivo de este paso es de reducir el número de palabras del conjunto de datos, y simplificar formas gramaticales como las diferentes conjugaciones verbales a un único verbo en infinitivos, o plurales a singulares, con el fin de mantener el significado semántico en la frase.

Se evaluó la eficiencia de diferentes modelos para decidir con cuál se realizaría la lematización, ya que, para lematizar correctamente las palabras de un texto, el modelo debe haber sido entrenado con un corpus lo suficientemente grande como para comprender y simular las normas y reglas gramaticales de un idioma para poder extraer correctamente el lema de formas irregulares o de palabras que forman parte de excepciones.

Para realizar la evaluación, se compararon los resultados de lematizar una serie de conjugaciones verbales con el modelo tipo Transformer de “es_dep_news_trf” con los resultados de lematizar empleando los modelos estadísticos “es_core_news_lg” y “es_core_news_sm”, entrenados con conjuntos de textos sobre noticias en español, con la diferencia de que el tamaño del corpus con el que se entrenó el primero de ellos es mayor.

A continuación, se muestran los resultados de lematizar unas conjugaciones verbales imperativas reflexivas, con los diferentes modelos evaluados de SpaCy, en la tabla 3 se compara, por cada modelo, la palabra original y el lema correspondiente.

Modelos pre-entrenados Spacy					
SpaCy - es trf		SpaCy - es lg		SpaCy - es sm	
Arréglate	Arregir él tú	Arréglate	Arréglatar	Arréglate	Arréglate
Duérmete	Duerir yo tú	Duérmete	Duérmete	Duérmete	Duérmete
Ámate	Amar tú	Ámate	Ámatir	Ámate	Ámate
Muérete	Morar tú	Muérete	Muérete	Muérete	Muérete
Dúchate	Duchir tú	Dúchate	Dúchate	Dúchate	Dúchate
Cálmate	Calmar tú	Cálmate	Cálmate	Cálmate	Cálmate
Cálzate	Calcer tú	Cálzate	Cálzate	Cálzate	Cálzate
Vístete	Visir tú tú	Vístete	Vístete	Vístete	Visir tú tú
Quiero	Querer	Quiero	Querer	Quiero	Querer
Duermo	Dormir	Duermo	Duermo	Duermo	Duermo

Tabla 3: Prueba de lematización con diferentes modelos 1) Spacy es_dep_news_trf 2) Spacy SpaCy es_core_news_lg 3) Spacy es_core_news_sm

También se contempla el uso de una librería diferente, llamada Stanza.

Los resultados de la lematización empleando el modelo en español de Stanza se recogen en la tabla 4:

Modelo Stanza	
Modelo español	
Arréglate	Arréglate
Duérmete	Duérmete
Ámate	Ámate
Muérete	Muérete
Dúchate	Dúchate
Cálmate	Cálmate

Tabla 4: Prueba de lematización con Stanza, modelo en español

Como se puede comprobar en la tabla 3l, a mayoría de las conjugaciones imperativas reflexivas no son detectadas como verbos, sino como sustantivos, por ello no se lematiza correctamente. El modelo Stanza lematiza incorrectamente la conjugación simple del verbo “dormir” (Tabla 4). El modelo que mejor reconoce las formas verbales es el modelo transformer de Spacy “es_dep_news_trf”. Los errores cometidos por este modelo fueron mínimos en comparación con el tamaño del corpus usado, por lo que se decidió el uso de este modelo para la lematización de las palabras de las frases. Se lematizaron todas las palabras salvo los adjetivos, para que mantuviesen el género gramatical original.

El corpus lematizado contiene un total de 10849 palabras.

Por último, se extrae el género gramatical de cada lema, información que genera el modelo de Spacy tras el análisis morfosintáctico. Un ejemplo de información extraída del texto inicial de una canción se muestra a continuación:

```

ID canción = ES-1948-03-03-003
Texto canción A los siete colchones, my señoras mías, que me ha dicho el madre que me dé usted la pitulá para ir a rezar a la
Virgen del Pilar.
Texto canción los colchones = colchones señoras madre de ella rezar virgen pilar
lemas = ['colchón', 'señora', 'madre', 'ella', 'rezar', 'virgen', 'pilar']
stop = ['mas', 'me', 'mea', 'mía', 'mías', 'mías', 'mías', 'mías']
diferé palabras = ['mas', 'me', 'mea', 'mía', 'mías', 'mías', 'mías', 'mías']

```

Figura 1: Ejemplo análisis morfológico canción

Una vez el texto ha sido correctamente preprocesado y etiquetado, puede utilizarse para realizar otro tipo de análisis de texto, como se detalla en las siguientes secciones.

5.3 Aplicaciones del Etiquetado Morfológico y Gramatical

Tras el preprocesamiento y lematización del texto, en la Tabla 5 se muestra un resumen de las palabras que aparecen con mayor frecuencia, así como el porcentaje sobre el número total de palabras.

Palabras lematizadas			Palabras sin lematizar		
Palabra	Frecuencia	Porcentaje	Palabra	Frecuencia	Porcentaje
niño	164	1.51 %	niño	150	1.38 %
querer	134	1.24 %	niña	85	0.78 %
venir	107	0.99 %	madre	77	0.71 %
dormir	99	0.92 %	duérmete	53	0.49 %
niña	98	0.90 %	virgen	52	0.48 %

Tabla 5: Frecuencia de palabras lematizadas y sin lematizar

Otra aplicación del etiquetado gramatical para extraer información del contenido de las canciones es el análisis de la frecuencia y uso de una categoría gramatical concreta, como los adjetivos. El objetivo del estudio de los adjetivos es el de visualizar la frecuencia del género gramatical de los adjetivos empleados, y extraer a qué sustantivos o personas hacen referencia con mayor frecuencia.

Se ha creado una variable tipo diccionario para poder almacenar asociaciones de sustantivos y su modificador, mediante un diccionario con formato (adjetivo, sustantivo): género. Un ejemplo de este diccionario es (sentado, gato): Masc, (morena, gata): Fem.

Con el fin de visualizar los adjetivos más empleados, se ha generado un diccionario de adjetivos ordenados según la frecuencia, con formato (adjetivo, género): frecuencia, el cual permite consultar, en este caso, la frecuencia de las palabras categorizadas gramaticalmente como adjetivos.

Este ejercicio podría modificarse en función de la categoría gramatical que se desee estudiar, así como otras etiquetas morfosintácticas que fuesen de interés. En este ejemplo, se ha generado un gráfico de barras para conocer la frecuencia de cada categoría de adjetivos, mostrado a continuación en la Tabla 2

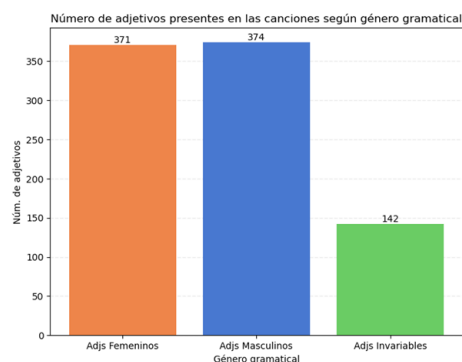


Figura 2: Frecuencia de adjetivos según género gramatical

La frecuencia de uso de adjetivos está balanceada en cuanto a género femenino y masculino, y figuran con menor frecuencia los adjetivos invariantes. A continuación, se visualizan los primeros quince adjetivos de cada categoría en la Tabla 6

Adjs. Femeninos	Frec.	Adjs. Masculinos	Frec.	Adjs. Invariantes	Frec.
bonita	27	puesto	20	pobre	20
blanca	16	blanco	15	verde	15
resalada	13	roto	11	triste	6
viva	10	querido	10	dulces	6
hermosas	8	lindo	9	natural	5
negra	8	sentado	8	azul	5
chiquita	8	fino	7	humilde	3
pequeñita	8	chiquito	7	pobres	3
rica	7	vestido	6	inmortal	2
vestida	7	bonito	6	fiel	2
alta	6	bello	6	constante	2
bella	6	dormido	6	ideal	2
pequeña	6	hermoso	6	grande	2
hermosa	6	cansado	5	alto	2
salada	5	muerto	5	rosas	2

Tabla 6: Adjetivos más frecuentes según género gramatical

Cabe destacar la diferencia semántica entre los adjetivos femeninos y los adjetivos masculinos, ya que en el conjunto de los 15 adjetivos gramaticalmente femeninos, un 66,6 % de ellos describen cualidades físicas o estéticas (como bonita, alta, hermosa) y aunque estos adjetivos también son empleados con género gramatical masculino, es en minoría, únicamente un 33.33 % ya que el resto definen otro tipo de características sobre propiedades o estados de los sujetos, como roto, querido, sentado, cansado, dormido etc.

Estos resultados corroboran las conclusiones que se describieron en, [7] donde se reflejaba que, tras realizar análisis gramatical de las palabras y comprobar el uso de los adjetivos para calificar mujeres y hombres, los empleados en mujeres tienen un contexto común de descripción física (bonita, morena, bella, linda, chiquita...) mientras que los calificativos masculinos son mucho más generales (pobre, muchachito, pícaro, chiquito...). Esto demuestra el favorable uso de herramientas inteligentes de NLP para análisis sintácticos.

En la siguiente sección se extrae información del texto, pero en forma de análisis y clasificación del sentimiento.

5.4 Análisis del Sentimiento

El análisis del sentimiento es un proceso que utiliza técnicas de NLP para identificar, extraer y comprender las emociones y opiniones expresadas en un

texto, permitiendo evaluar el tono y la polaridad emocional del mismo [15]. En este apartado, se detallará la aplicación de un modelo de clasificación de sentimiento al conjunto de letras de canciones tradicionales infantiles, que en la sección previa han sido preprocesadas. Para esta tarea de análisis se aplica un modelo clasificador de sentimientos basado en Transformer al conjunto de letras de canciones en español. Esto permite detectar si en la letra de cada canción se refleja un sentimiento positivo, negativo o neutral, a partir de las palabras que contiene, y poder obtener información acerca del contexto general.

Se emplea la librería “Pysentimiento” [69] ya que ha sido la que mejor resultados ha logrado. Se han probado otras librerías conocidas para análisis de sentimientos como TextBlob [71], pero esta no incluye análisis de sentimientos en español. Por ello, se compararán los resultados de Pysentimiento con el modelo de Vader[72].

5.4.1 Conjunto de Datos y Limpieza de Texto Se analizarán un total de 685 canciones en español, las cuales han sido previamente preprocesadas y se ha realizado un análisis y etiquetado morfológico y gramatical de las palabras.

En este caso, se genera una lista de palabras semánticamente vacías y que se repiten en las canciones como “la, olé, ea” u otras interjecciones, y se mantienen en el texto el resto de palabras vacías en español para mantener palabras con información sentimental como “no”.

5.4.2 Clasificador pysentimiento El clasificador de sentimientos de pysentimiento[69] emplea los modelos pre-entrenados de BERT en español, denominados RoBERTa y BETO.

El modelo analizador de pysentimiento devuelve la probabilidad de que un texto sea positivo, negativo o neutral, dando como output la categoría que supera un valor de 0.5 en cada texto. En la Figura 3 se muestra el número de canciones según la categoría de sentimiento.

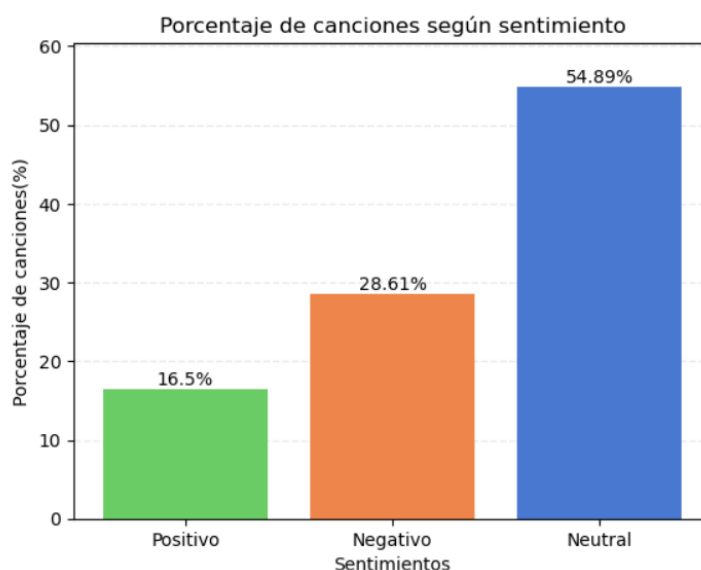


Figura 3: Número de canciones según categoría de sentimiento

Aproximadamente la mitad de las canciones han sido catalogadas como neutrales, ya que la mayoría de ellas narran situaciones cotidianas imparciales y no contienen palabras que expresan emociones. Sin embargo, han sido catalogadas como canciones con tono negativo casi el doble de canciones que se catalogaron como positivas, resultado que podría ser poco esperable al tratarse de letras de canciones tradicionales infantiles.

Se contempla en la literatura el uso de clasificadores con modelos más precisos que emplean herramientas como la lógica difusa [78] para catalogar emociones de textos, ya que presentan una precisión mucho mayor en la clasificación de sentimientos.

Una vez el conjunto de canciones se ha clasificado sentimentalmente, se guardan conjuntos de datos independientemente según la categoría sentimental, para poder estudiar posteriormente las propiedades semánticas de las letras de las canciones de cada grupo y compararlas entre sí.

5.4.3 Analizador del Sentimiento VADER multi-language Otra opción de modelo de análisis del sentimiento que se ha probado ha sido una librería basada en VADER (Valence Aware Dictionary and sEntiment Reasoner) [72] y la librería disponible en ³, que implementa el modelo de VADER (basado en conjunto de datos en inglés) junto una función traductora que auto detecta el idioma, para poder emplearla en textos en español. Se ha hecho una prueba de análisis de un número de canciones para saber si la clasificación es correcta.

³ <https://github.com/brunneis/vader-multi/tree/master>



Figura 5: WordCloud canciones positivas

En el caso de las canciones positivas, sí que tienen reiteración de palabras que manifiestan sentimientos positivos, como palabras como bonita, amor, corazón, sol. Se genera un gráfico similar visualizando las palabras con mayor frecuencia dentro del conjunto de canciones neutrales, en la Figura 6

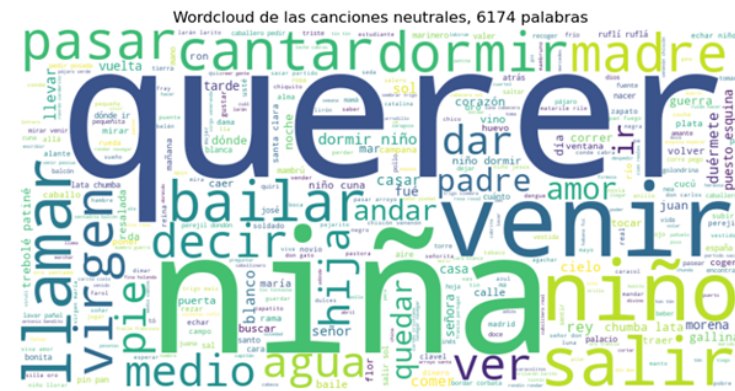


Figura 6: WordCloud canciones neutrales

A este grupo pertenecen la mitad de las canciones analizadas por lo que las palabras destacadas tienen un contexto muy general.

5.4.5 Análisis de la Frecuencia de Palabras según Fentimiento Una vez se han clasificado las canciones por el sentimiento reflejado en sus letras, se evalúa la frecuencia de palabras como “niño, niña” y “madre, padre” en función de la categoría sentimental de la canción. Para ello, se extraen el valor de palabras

totales de cada categoría, la frecuencia de aparición de las palabras mencionadas, y se calcula el porcentaje respecto al total de palabras de cada grupo. El gráfico resultante se muestra en la Figura 7

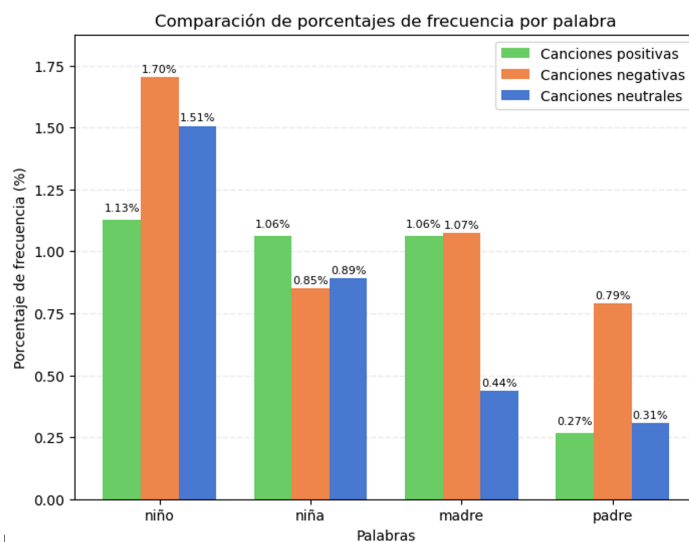


Figura 7: Frecuencia de palabras de interés en función del sentimiento

La frecuencia de reiteración de la palabra “niño” es la más frecuente en todas las categorías de canciones. En comparación, la frecuencia de la palabra “niña” es prácticamente la mitad que la frecuencia de “niño” tanto en categoría negativa como neutral, siendo mayoritarias estas dos categorías. La palabra “madre” está presente en el mismo ratio tanto en canciones positivas como negativas, pero su frecuencia es mínima en las neutrales, representando estas la mitad de las canciones totales. Por otro lado, la representación de la figura paterna con la palabra “padre” aparece mínimamente en comparación con las otras palabras, y su frecuencia se dispara en las canciones negativas, siendo casi tan alta como la frecuencia de la palabra “niña” en las canciones negativas.

5.5 Topic Modeling

El topic modeling en NLP permite identificar y extraer automáticamente los temas principales de un conjunto de documentos, facilitando la organización y comprensión de grandes cantidades de información textual. En este caso, se aplicará al conjunto de datos de letras de canciones tradicionales infantiles para conocer en qué temáticas se agrupan las canciones y cuales son las más comunes. Un estudio similar se ha realizado en [12] pero sin ayuda de herramientas de NLP automáticas.

Para implementar esta tarea en Python, se utiliza la librería `gensim` [65] ya que está especializado en `topic modeling` y permite utilizar herramientas y modelos con esa finalidad de forma sencilla e intuitiva.

Para generar `topics` o agrupaciones de palabras en temas comunes entre las letras de las canciones, se parte de texto tokenizado y sin palabras vacías y palabras lematizadas (excepto los adjetivos) para mantener el género gramatical.

Con el uso del modelo LDA con el conjunto de letras de canciones populares infantiles españoles, se pretende visualizar cuáles son las temáticas más comunes, identificando si resaltan aquellas que transmiten valores que perpetúan los estereotipos de género, objetivo que también se desarrolla en [12], pero sin el uso de NLP, y se compararán los resultados obtenidos.

A continuación se detallan cada uno de los pasos llevados a cabo para la detección de temáticas en las letras de canciones tradicionales infantiles.

5.5.1 Limpieza de Texto Dado que el modelo LDA de `topic modeling` se basa en la frecuencia de las palabras en los documentos, es necesario generar un diccionario que identifique cada palabra, y un corpus que contenga la frecuencia de las mismas.

El conjunto de datos de partida que se van a usar va a ser el dataset completo de canciones en español, . Partiendo del texto preprocesado (sin palabras vacías, tokenizado y lematizado). En el siguiente apartado se detalla la creación de un diccionario y corpus a partir de las palabras de las canciones.

5.5.2 Generación de Temáticas a partir de Diccionario y Corpus BoW

Una vez se han extraído las palabras o tokens que forman cada frase de cada canción, se debe crear un diccionario que recoja todos los tokens del conjunto de canciones. Se genera un diccionario que permite identificar cada palabra con un índice, que será empleado en la construcción del corpus.

Así mismo, se crea el corpus, un conjunto de tipo “Bag of Words” que recoge el identificador de cada palabra, y el número de veces que se repite en una frase dada. Se crea una entrada para cada canción, de forma que se obtiene la frecuencia de cada palabra dentro de la letra de cada canción. Esto permite generar los temas y valorar el grado de pertenencia de una frase a una temática, a partir de la probabilidad de que esté compuesta por palabras que pertenecen a una misma temática. El diccionario está compuesto por un total de 4514 tokens.

A continuación, se aplica el modelo LDA al corpus y se vincula al diccionario para poder identificar las palabras de cada temática. En este estudio se define la identificación de 4 temas, pero este número puede ser mayor al trabajar con conjuntos de texto de mayor tamaño. Además, se ha comprobado que al aumentar este número a 10 o a 20, se generan agrupaciones con menor relación semántica entre las palabras, por lo que con 4 se definen temáticas visualmente más aisladas entre sí.

En el uso del modelo, se indican con el parámetro `chunksize= 50` el número de documentos (palabras) usadas en cada iteración de entrenamiento. Con el parámetro `passes` se define el número de veces que se recorre el corpus durante

el entrenamiento, establecido en 10. Por último, el parámetro alfa define cómo están distribuidos los diferentes topics a lo largo de un documento, cuanto menor sea, indica que un topic concreto lo definen un número de documentos concretos. [70] Si es más alto, indica que hay una distribución mucho más general dentro del corpus. La librería gensim permite distribuir este valor entre tres modalidades:

- alfa simétrico: aplica un alfa igual a todos los topics con valor $1,0/numtopics$.
- alfa asimétrico: aplica un valor de alfa en función de cada topic, de forma $1,0/(topic_index + \sqrt{num_topics})$.
- auto: aplica de forma asimétrica el valor de alfa en función del corpus empleado.

En la Figura 8 se compara la distribución de canciones por temática en función del tipo de alfa.

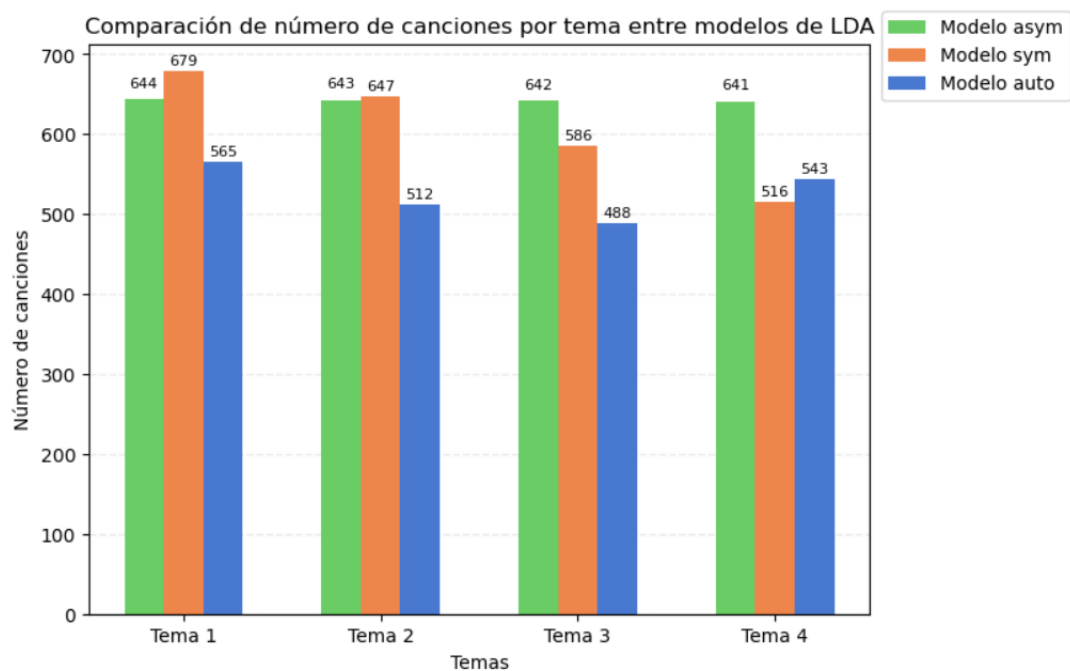


Figura 8: Distribución número de canciones por topic en función de distribución del modelo

Además, para determinar si un modelo de detección de temáticas es válido, se emplea un concepto denominado perplejidad, que evalúa cuanto de similar es una nueva palabra a un topic dado, empleando una parte del corpus como evaluación. Cuanto menor sea este valor, mejor será el modelo al definir topics donde las palabras están relacionadas entre sí, como se define en la documentación de [65]

Los valores de perplejidad de los modelos LDA con alfa simétrico, asimétrico y auto son, respectivamente, de $-7,8582$, $-7,870$ y $-7,8534$.

La diferencia entre los valores perplejidad es muy mínima y similar a los que se obtienen en el estudio de [43], en el que emplean el mismo modelo de la librería gensim, pero aplicado a un dataset de texto extraído de Twitter, obteniendo perplejidad de -8.8702 . Por ello, se valora que el modelo ha catalogado los topics favorablemente.

En [70] se define la importancia de este valor y como una definición asimétrica de alfa ayuda a encontrar puntos en común entre los documentos, en este caso canciones. Teniendo esto en cuenta, se selecciona el método “auto” ya que aplica una distribución asimétrica pero adaptada al corpus, por lo que hace el modelo más adecuado al texto de las canciones empleado.

Para cada temática, se genera el WordCloud con las 50 palabras más frecuentes, usando el corpus y diccionario de todas las canciones. Los WordClouds resultantes se muestran en la Figura 9.



Figura 9: WordCloud de las cuatro temáticas identificadas en todas las canciones

Puede que las diferencias semánticas y de contextuales entre las temáticas de las canciones sean algo generales, posiblemente debido a la similitud entre las letras de canciones, el uso de palabras muy similares al tratarse de vocabulario dirigido a niños y niñas, y el tratarse de un conjunto de datos relativamente pequeño. Comprobando los wordclouds generados por temáticas, se pueden intuir temas como cantar, bailar, la presencia de figuras como el padre o la Virgen, temas relacionados con dormir, y un conjunto más general como la temática 4.

Cabe destacar, así como se concluye en [12], la alarmante frecuencia de conceptos como “casar” “novia” y similares, al tratarse de canciones destinadas a niños y niñas pequeñas, se inculca desde que tienen corta edad, aspectos platónicos como el amor, y la necesidad de casarse, evento que solía ser decisión de los hombres, pero destino de las mujeres. De la misma forma, la presencia de la pa-

labra “madre” es mucho más frecuente en canciones con temática de cuidar a los hijos e hijas, ocupándose de que duerman, y cuidándolos, perpetuando de esta forma el estereotipo de que es la mujer la que se debe encargar exclusivamente de tareas de cuidados del hogar y de sus hijos, ya que la palabra “padre” no se relaciona con temas relacionados.

Este mismo método podría aplicarse a cada conjunto de canciones según la categoría de sentimiento, para poder visualizar los temas comunes dentro de cada clasificación.

5.5.3 Identificación de Temática dada una Canción A continuación, se propone un método con el que se puede visualizar el tema más probable de una canción en concreto, para comprobar que el modelo clasificó las temáticas correctamente, así como mostrar información morfosintáctica de la letra. La función mostrará la información del etiquetado morfosintáctico y lematización de las palabras, así como la categoría emocional a la que pertenece y la contribución a cada temática. Esta probabilidad se visualiza en un gráfico de barras, así como el tema más probable de todos y las palabras que lo definen. Por ejemplo, se escoge obtener información de una canción dado su ID, que será “ES-1948-CB-CO-003”. El resultado se muestra en la Figura 10

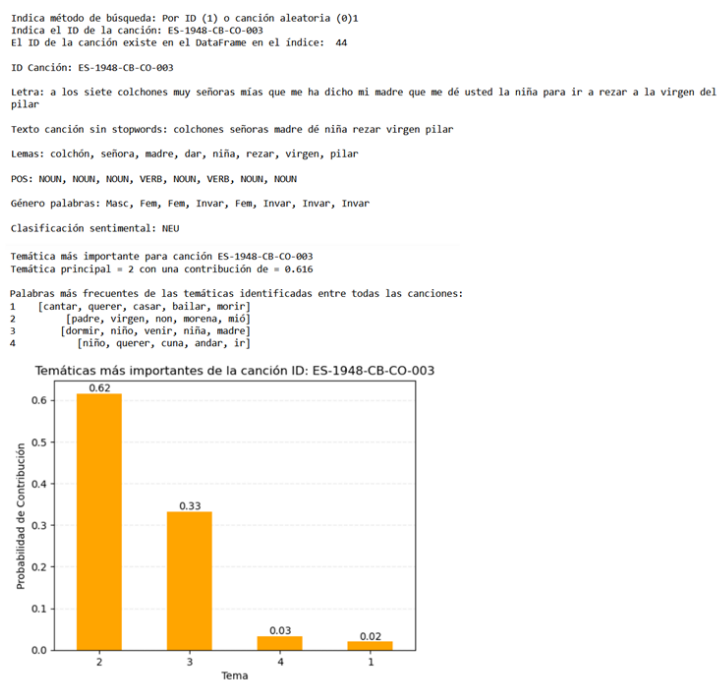


Figura 10: Ejemplo de análisis de una canción, dado un ID

Como se ha comprobado en este desarrollo, se pueden agrupar canciones según las palabras más frecuentes en ellas de forma sencilla, lo cual permite estudiar los temas comunes entre canciones. Se ha hecho un estudio similar en [12] y, disponiendo de un conjunto de datos de mejor calidad, se podría realizar un análisis de temas comunes en las canciones de forma más sencilla sin necesidad de visualizar canción a canción, empleando modelos como LDA para obtener las palabras más importantes de cada categoría.

6. Conclusiones y Trabajo a Futuro

Se ha utilizado el Procesamiento del Lenguaje Natural (NLP) para examinar estereotipos de género y discriminación en las letras de canciones. Este enfoque ha demostrado su eficacia en el análisis automatizado, revelando una mención desequilibrada de género en las canciones, donde los adjetivos femeninos se centran en cualidades estéticas y estereotipadas de la mujer ideal, mientras que los masculinos son más arbitrarios. Además, se ha encontrado una temática predominante en las canciones infantiles que perpetúa roles y estereotipos, limitando la percepción y el potencial de las niñas en la sociedad.

El uso de herramientas de NLP ha sido valioso para visualizar y analizar estereotipos de género en canciones infantiles, resaltando la importancia de promover diversidad y representación equitativa en las letras para fomentar una educación inclusiva desde temprana edad. Sin embargo, el estudio se ha visto limitado por el conjunto de datos compuesto por canciones tradicionales, que presentan características peculiares como onomatopeyas y frases con poca concordancia semántica debido a su naturaleza rítmica y pegadiza.

Como trabajo futuro, se plantea investigar la discriminación de género en textos escritos y abordar los sesgos presentes en modelos de lenguaje como GPT-3 o BERT, que pueden generar textos con sesgos de género debido a los datos utilizados para su entrenamiento. Se propone utilizar técnicas de NLP para analizar y minimizar estos sesgos, con el objetivo de avanzar hacia un estudio del lenguaje sin estereotipos de género.

Referencias

1. Pedrell, F.: *Cancionero musical popular español*. Vol. 1. E. Castells (1900)
2. Schindler, K.: *Folk music and poetry of Spain and Portugal= Música y poesía popular de España y Portugal*. (No Title) (1941)
3. Venkit, P. N., Wilson, S.: *Identification of bias against people with disabilities in sentiment analysis and toxicity detection models*. arXiv preprint arXiv:2111.13259 (2021)
4. Martínez Torner, E., Bal y Gay, J.: *Cancionero gallego*. A Coruña: Fundación Barrié de la Maza (1973)
5. Creany, A. D.: *The Appearance of Gender in Award-Winning Children's Books*. ERIC (1995)
6. Beatty-Martínez, A. L., Dussias, P. E.: *Revisiting masculine and feminine grammatical gender in Spanish: Linguistic, psycholinguistic, and neurolinguistic evidence*. *Frontiers in Psychology*, 10, 751 (2019)

7. de Luna, E. B., Pérez, J. G.: *Música y género: análisis de una muestra de canciones populares. Comunicar: Revista científica iberoamericana de comunicación y educación*, 1(18), 187-190 (2002)
8. Watanabe, K., Goto, M.: *Lyrics information processing: Analysis, generation, and applications*. In *Proceedings of the 1st Workshop on NLP for Music and Audio (NLP4MusA)*, 6-12 (2020)
9. Roca, J.: *Lexicon Based Approaches to Sentiment Analysis of Spanish Tweets: A Comparative Study. Computer Science and Computer Engineering Undergraduate Honors Theses* (2020)
10. Lopez, M. M., Kalita, J.: *Deep Learning applied to NLP*. arXiv preprint arXiv:1703.03091 (2017)
11. Pascual, D., et al.: *Construcción social de género en el cancionero infantil español. Ensayos: Revista de la Facultad de Educación de Albacete*, 1(25), 141-155 (2010)
12. Fernández Poncela, A. M.: *Género y canción infantil. Política y cultura*, 1(26), 35-68 (2006)
13. Liddy, E. D.: *Natural language processing*. (2001)
14. Khurana, D., Koli, A., Khatter, K., Singh, S.: *Natural language processing: State of the art, current trends and challenges. Multimedia tools and applications*, 82(3), 3713-3744 (2023)
15. Yi, J., Nasukawa, T., Bunescu, R., Niblack, W.: *Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques*. In *Third IEEE international conference on data mining*, 427-434 (2003)
16. Yin, W., Kann, K., Yu, M., Schütze, H.: *Comparative study of CNN and RNN for natural language processing*. arXiv preprint arXiv:1702.01923 (2017)
17. Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., Heck, L.: *Contextual lstm (clstm) models for large scale nlp tasks*. arXiv preprint arXiv:1602.06291 (2016)
18. Devlin, J., Chang, M. W., Lee, K., Toutanova, K.: *Bert: Pre-training of deep bidirectional transformers for language understanding*. arXiv preprint arXiv:1810.04805 (2018)
19. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: *Roberta: A robustly optimized bert pretraining approach*. arXiv preprint arXiv:1907.11692 (2019)
20. Staliūnaitė, L., Iacobacci, I.: *Compositional and lexical semantics in RoBERTa, BERT and DistilBERT: A case study on CoQA*. arXiv preprint arXiv:2009.08257 (2020)
21. Pérez, J. M., Furman, D. A., Alemany, L. A., Luque, F.: *Robertuito: a pre-trained language model for social media text in Spanish*. arXiv preprint arXiv:2111.09453 (2021)
22. Asgari-Chenaghlu, M., Feizi-Derakhshi, M. R., Farzinvasht, L., Balafar, M. A., Motamed, C.: *Topic detection and tracking techniques on Twitter: a systematic review. Complexity*, 2021, 1-15 (2021)
23. Osorio Angel, S., Peña Pérez Negrón, A., Espinoza-Valdez, A.: *Systematic literature review of sentiment analysis in the Spanish language. Data Technologies and Applications*, 55(4), 461-479 (2021)
24. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: *Huggingface's transformers: State-of-the-art natural language processing*. arXiv preprint arXiv:1910.03771 (2019)
25. Cañete, J., Chaperon, G., Fuentes, R., Ho, J. H., Kang, H., Pérez, J.: *Spanish pre-trained bert model and evaluation data. Pml4dc at iclr*, 2020(2020), 1-10 (2020)
26. Torner, E. M.: *Cancionero musical de la lírica popular asturiana*. Establecimiento tipográfico Nieto y compañía (1920)

27. Córdova, S.: *Cancionero popular de la provincia de Santander*. OÑA, Santander, Aldus, SA, Artes Gráficas (1948)
28. Olmeda, F.: *Cancionero popular de Burgos*. Librería Editorial de M. Auxiliadora (1903)
29. Martínez Palacios, A. J.: *Colección de cantos populares burgaleses*. (Nuevo cancionero burgalés). Madrid, Unión Musical Española (1980)
30. Blanco, V.: *Las mil y una canciones populares de la región leonesa*. Astorga: Talleres Gráficos Ángel Julián (1931)
31. Fraile, A. S.: *Nuevo cancionero salmantino*. Imprenta Provincial (1943)
32. Ledesma, D.: *Cancionero salmantino (1907)*. Salamanca, Imprenta Provincial (1972)
33. Manzano, M.: *Cancionero popular de Burgos*. Diputación Provincial de Burgos (2006)
34. García Matos, M.: *Lírica popular de la Alta Extremadura*. (No Title) (1925)
35. Verdú, J.: *Colección de cánticos populares de Murcia*. Orfeo Tracio (1906)
36. Biblioteca Popular de L'Avenç: *40 cançons populars catalanes*. L'Avenç (1909)
37. Frenda, S., Patti, V., et al.: *Computational models for irony detection in three Spanish variants*. In *CEUR Workshop Proceedings*, 2421, 297-309 (2019)
38. Vilaró, S. J.: *El Cancioner del Calic*. Ajuntament de la Vila de Bagà (1989)
39. Pujol, F., Joan, P. i C., Joan, L. i B.: *Obra del Cancionero Popular de Catalunya*. Fundació Concepció Rabell i Cibils, Vda. Romaguera (1926)
40. Terrera, G. A.: *Primer Cancionero popular de Córdoba. Investigación Científico folklorica. With musical notes..* (Undefined) (1948)
41. Lopez-Medel, M.: *Gender bias in machine translation: an analysis of Google Translate in English and Spanish*. (2021)
42. Prates, M. O., Avelar, P. H., Lamb, L. C.: *Assessing gender bias in machine translation: a case study with google translate*. *Neural Computing and Applications*, 32, 6363-6381 (2020)
43. Tijare, P., Rani, P. J.: *Exploring popular topic models*. In *Journal of Physics: Conference Series*, 1706, 012171 (2020)
44. Bartl, M., Nissim, M., Gatt, A.: *Unmasking Contextual Stereotypes: Measuring and Mitigating BERT's Gender Bias*. (2020)
45. Bolukbasi, Tolga and Chang, Kai-Wei and Zou, James Y and Saligrama, Venkatesh and Kalai, Adam T. "Man is to computer programmer as woman is to homemaker? Debiasing word embeddings." *Advances in Neural Information Processing Systems*, vol. 29, 2016.
46. Mikolov, Tomas and Chen, Kai and Corrado, Greg and Dean, Jeffrey. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*, 2013.
47. Basta, Christine and Costa-Jussa, Marta R and Casas, Noe. "Extensive study on the underlying gender bias in contextualized word embeddings." *Neural Computing and Applications*, vol. 33, no. 8, pp. 3371-3384, 2021.
48. Sun, Tony and Gaut, Andrew and Tang, Shirlyn and Huang, Yuxin and ElSherief, Mai and Zhao, Jieyu and Mirza, Diba and Belding, Elizabeth and Chang, Kai-Wei and Wang, William Yang. "Mitigating gender bias in natural language processing: Literature review." *arXiv preprint arXiv:1906.08976*, 2019.
49. May, Chandler and Wang, Alex and Bordia, Shikha and Bowman, Samuel R and Rudinger, Rachel. "On measuring social biases in sentence encoders." *arXiv preprint arXiv:1903.10561*, 2019.

50. Betti, Lorenzo and Abrate, Carlo and Kaltenbrunner, Andreas. "Large scale analysis of gender bias and sexism in song lyrics." *EPJ Data Science*, vol. 12, no. 1, p. 10, 2023.
51. Sharma, Abhishek and Amrita and Chakraborty, Sudeshna and Kumar, Shivam. "Named entity recognition in natural language processing: A systematic review." In *Proceedings of Second Doctoral Symposium on Computational Intelligence: DoSCI 2021*, pp. 817-828, 2022. Springer.
52. Kumawat, Deepika and Jain, Vinesh. "POS tagging approaches: A comparison." *International Journal of Computer Applications*, vol. 118, no. 6, 2015.
53. Qi, Peng and Zhang, Yuhao and Zhang, Yuhui and Bolton, Jason and Manning, Christopher D. "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2020.
54. Sazzed, Salim. "Improving sentiment classification in low-resource Bengali language utilizing cross-lingual self-supervised learning." In *International Conference on Applications of Natural Language to Information Systems*, pp. 218-230, 2021. Springer.
55. Lukic, Alen. "A comparison of topic modeling approaches for a comprehensive corpus of song lyrics." Tech report, Language Technologies Institute, School of Computer Science, 2015.
56. Narahara, May M. "Gender Stereotypes in Children's Picture Books." *Psychological monographs*, vol. 60, no. 3, p. i, 1998.
57. Rendón-Cardona, Paula and Gil-Gonzalez, Julian and Páez-Valdez, Julián and Rivera-Henao, Mauricio. "Self-supervised sentiment analysis in Spanish to understand the university narrative of the Colombian conflict." *Applied Sciences*, vol. 12, no. 11, p. 5472, 2022.
58. Vayansky, Ike and Kumar, Sathish AP. "A review of topic modeling methods." *Information Systems*, vol. 94, p. 101582, 2020. Elsevier.
59. Palani, Sarojadevi and Rajagopal, Prabhu and Pancholi, Sidharth. "T-BERT-Model for Sentiment Analysis of Micro-blogs Integrating Topic Model and BERT." *arXiv preprint arXiv:2106.01097*, 2021.
60. Honnibal, Matthew and Montani, Ines and Van Landeghem, Sofie and Boyd, Adriane. "spaCy: Industrial-strength Natural Language Processing in Python." 2020. doi: 10.5281/zenodo.1212303.
61. Bird, Steven and Klein, Ewan and Loper, Edward. "Natural language processing with Python: analyzing text with the natural language toolkit." 2009. O'Reilly Media, Inc.
62. Carreras, Xavier and Chao, Isaac and Padró, Lluís and Padró, Muntsa. "FreeLing: An Open-Source Suite of Language Analyzers." In *LREC*, pp. 239-242, 2004.
63. Gamallo, Pablo and Pichel, Juan Carlos and Garcia, Marcos and Abuín, José Manuel and Pena, Tomás Fernández. "Análisis morfosintáctico y clasificación de entidades nombradas en un entorno Big Data." *Procesamiento del Lenguaje Natural*, vol. 53, pp. 17-24, 2014.
64. Orrequia-Barea, Aroa and Marín-Honor, Cristian. "Building a parallel corpus of literary texts featuring onomatopoeias: ONPACOR." *Research in Corpus Linguistics*, vol. 8, no. 2, pp. 46-62, 2020.
65. Rehůrek, Radim and Sojka, Petr. "Software Framework for Topic Modelling with Large Corpora." In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 45-50, 2010.
66. Nakatani, Shuyo. *Language Detection Library for Java*. Available at: <https://github.com/shuyo/language-detection>, 2010.

67. Savand, Alireza. *Stop Words*. Available at: <https://github.com/Alir3z4/stop-words>, 2018.
68. Mueller, Andreas and Fillion-Robin, Jean-Christophe and Boidol, Raphael and Tian, Font and Nechifor, Paul and Rampin, Remi and Corvellec, Marianne and Medina, Juan and Dai, Yuchao, et al. .^mueller/Word.Cloud: Wordcloud 1.5.0." *Zenodo*, 2018.
69. Pérez, Juan Manuel and Giudici, Juan Carlos and Luque, Franco. "pysentimiento: A Python Toolkit for Sentiment Analysis and SocialNLP tasks." *arXiv preprint arXiv:2106.09462*, 2021.
70. Wallach, Hanna and Mimno, David and McCallum, Andrew. Rethinking LDA: Why priors matter." *Advances in neural information processing systems*, vol. 22, 2009.
71. Loria, Steven. *TextBlob: Simplified Text Processing*. Available at: <https://github.com/sloria/TextBlob>, 2020.
72. Hutto, Clayton and Gilbert, Eric. "Vader: A parsimonious rule-based model for sentiment analysis of social media text." In *Proceedings of the international AAAI conference on web and social media*, vol. 8, pp. 216-225, 2014.
73. Sánchez, Belém Priego. .^nálisis de la diversidad morfosintáctica en las locuciones verbales." *Res. Comput. Sci.*, vol. 97, pp. 113-125, 2015.
74. García-Díaz, José Antonio and Cánovas-García, Mar and Colomo-Palacios, Ricardo and Valencia-García, Rafael. "Detecting misogyny in Spanish tweets: An approach based on linguistics features and word embeddings." *Future Generation Computer Systems*, vol. 114, pp. 506-518, 2021. Elsevier.
75. Lucy, Li and Demszky, Dorottya and Bromley, Patricia and Jurafsky, Dan. Content analysis of textbooks via natural language processing: Findings on gender, race, and ethnicity in Texas US history textbooks." *AERA Open*, vol. 6, no. 3, p. 2332858420940312, 2020. Sage Publications Sage CA: Los Angeles, CA.
76. Melville, Sophie and Eccles, Kathryn and Yasseri, Taha. "Topic modeling of everyday sexism project entries." *Frontiers in Digital Humanities*, vol. 5, p. 28, 2019. Frontiers Media SA.
77. Jiang, Aiqi and Zubiaga, Arkaitz. "QMUL-SDS at EXIST: Leveraging Pre-trained Semantics and Lexical Features for Multilingual Sexism Detection in Social Networks." In *IberLEF@ SEPLN*, pp. 469-483, 2021.
78. Maheswari, S Uma and Dhenakaran, SS. .^spect-based fuzzy logic sentiment analysis on social media big data." In *2020 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0971-0975, 2020. IEEE.
79. Blei, David M and Ng, Andrew Y and Jordan, Michael I. "Latent dirichlet allocation." *Journal of Machine Learning Research*, vol. 3, pp. 993-1022, 2003.

Optimización de la movilidad a partir del manejo de datos de flujo de tráfico

Juan Manuel Otero Otero, Vivian Félix López Batista y Juan Francisco de Paz Santana

Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
{juanm.otero2,vivian,fcofds}@usal.es

Resumen La congestión del tráfico es un problema que afecta la economía y calidad de vida de las personas. En España, el crecimiento vehicular ha generado impactos negativos en el medio ambiente y la seguridad vial. En este contexto, este trabajo se centró en optimizar la gestión semafórica de una intersección en Salamanca utilizando simuladores de tráfico y técnicas de optimización. Se lograron configuraciones óptimas de tiempos de semáforos mediante una metodología rigurosa que incluyó la recopilación de datos, modelado y ajuste de simulaciones, y el uso del algoritmo PSO. Los resultados mostraron una distribución más equitativa del flujo vehicular, demostrando la importancia de la simulación y optimización para mejorar la eficiencia vial. A futuro, se plantea la implementación en tiempo real utilizando tecnologías de visión artificial y la exploración de otros simuladores y técnicas de optimización, buscando sistemas de control de tráfico más eficientes y sostenibles.

Keywords: Congestión, simulador de tráfico, algoritmo de optimización, SUMO, PSO.

1. Introduction

La congestión del tráfico produce un gran impacto en la economía de un país y en la salud de sus habitantes [1]. Este fenómeno global resulta de la alta densidad de población, el crecimiento de la cantidad de vehículos motorizados y la proliferación de servicios de entrega y viajes compartidos [2]. La congestión es producto de la fricción entre los vehículos en tránsito, donde cada vehículo adicional interrumpe el libre tránsito de los demás [3].

Existen varios escenarios que producen congestión vehicular, incluyendo el exceso de demanda de la vía, señalización inadecuada, incidentes o zonas congestionadas por horarios laborales o escolares. Las causas más comunes son: cuellos de botella, infraestructuras o controladores deficientes, variación del flujo de tráfico, accidentes, obras, eventos climáticos adversos y eventos especiales [4][5][6][7].

El volumen vehicular ha aumentado a nivel mundial, produciendo que en algunas ciudades la cantidad de vehículos en uno o más puntos de la vía exceda su

capacidad [8]. Este crecimiento ha causado un incremento de la contaminación ambiental, acústica, visual y atmosférica [9][10]. Además, la inadecuada planificación del tráfico ha causado altos niveles de congestión y siniestralidad vial [11]. Este incremento se puede ver reflejado utilizando los datos de Parque Nacional de Vehículos de España. En la siguiente gráfica se muestra la evolución del Parque Nacional de Vehículos desde el 2006 hasta el 2020, con un incremento del 19,6 % de vehículos matriculados durante este periodo [Figura 1].

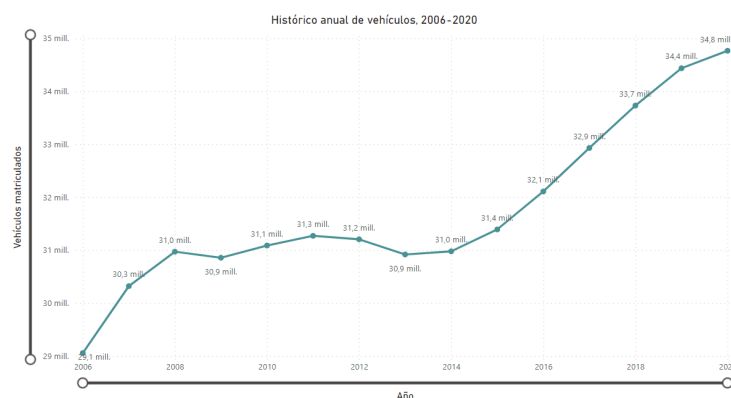


Figura 1: Evolución del Parque Nacional de Vehículos 2006-2020 [12].

Según los datos proporcionados por la Dirección General de Tráfico, en España, a 31 de diciembre de 2021, había 35.178.808 vehículos matriculados [64]. Esto se traduce en un volumen de 743,25 vehículos por cada 1000 habitantes [30]. En el caso de la provincia de Salamanca, en la misma fecha había matriculados 243.268 vehículos [64], lo que se traduce en un volumen de 745,15 vehículos por cada 1000 habitantes [30].

Este constante crecimiento vehicular, junto con el incremento de la población mundial, ha generado la necesidad de pensar en diseños urbanos más eficientes y sostenibles [57]. Por este motivo, se requiere que las ciudades inviertan en sistemas de control del flujo de tráfico, implementando soluciones como control a tiempo real de semáforos o análisis con *software* especializado [50].

Dentro de las soluciones, los simuladores de tráfico se postulan como una herramienta con alto potencial para estudiar el flujo vehicular, su optimización y el análisis de las políticas municipales implementadas [33]. Los simuladores de tráfico ofrecen una gran variedad de opciones como el control de semáforos, control de aforo o control de zonas restringidas para ciertos vehículos [74].

Este trabajo se enfoca en optimizar la gestión semafórica de una intersección específica en Salamanca. El objetivo principal es mejorar la eficiencia del flujo vehicular y reducir la congestión mediante una técnica de optimización. Además, se abordan objetivos secundarios que incluyen el análisis de simuladores de tráfi-

co, estudios de casos reales en la gestión del tráfico, investigación de técnicas de optimización para semáforos y la selección de una intersección relevante en Salamanca. Se desarrollará una metodología que incluirá la recopilación de datos relevantes, calibración del simulador de tráfico e implementación de la técnica de optimización.

1.1 Impacto esperado - Objetivos de Desarrollo Sostenible

Los avances en simuladores de tráfico y técnicas de optimización ofrecen nuevas oportunidades para mejorar la eficiencia y la sostenibilidad de la movilidad urbana como: la reducción de las emisiones [58], la planificación sostenible de las ciudades y la calidad del transporte con el fin de lograr una gestión eficiente del tráfico que permita obtener un desarrollo urbano equitativo, resiliente y respetuoso con el medio ambiente [35]. Los ODS directamente abordados en este trabajo son [65]: el ODS 3 - Salud y bienestar [58][17][56][59], ODS 8 - Trabajo decente y crecimiento económico [54], ODS 9 - Industria, innovación e infraestructura [40] y el ODS 11 - Ciudades y comunidades sostenibles [53][48][20].

2. Estado del arte

La gestión del tráfico en las ciudades es un desafío debido a su complejo sistema vial. Interrupciones en el flujo vehicular por intersecciones, avenidas y cruces peatonales causan problemas de congestión de tráfico y tiempos de espera elevados [34]. Una intersección señalizada es aquella que está regulada por un conjunto de señales, como semáforos, que regulan el flujo de tráfico [72]. Estos sistemas ayudan a evitar accidentes, reducir los tiempos de espera y disminuir tanto el consumo de combustible como las emisiones de gases contaminantes [31].

Los semáforos tradicionales son sistemas automáticos con políticas de cambio de señales fijas. Es decir, no se adaptan a las variaciones del flujo vehicular ni a las necesidades cambiantes de los usuarios de la vía, lo que puede alargar los tiempos de espera [36]. Una mejora a la política de semaforización tradicional son los semáforos inteligentes. Estos, adaptan sus decisiones en función del tráfico existente, detectando el flujo vehicular mediante sensores y ajustando los tiempos de paso o detención en consecuencia [36][27][46].

Existen varios sistemas de semaforización inteligente. Algunos usan tarjetas RFID en los vehículos para ajustar tiempos de espera según la congestión vehicular. Otros emplean redes de sensores inalámbricos que miden datos como el número de vehículos, sus velocidades y el espacio entre ellos. Por último, algunos sistemas de semaforización inteligente usan procesamiento de imágenes para recoger información a través de sensores ópticos [25]. El último método, la semaforización inteligente utilizando procesamiento de imágenes, tiene varias ventajas. Proporciona una detección más precisa y detallada de los vehículos y su comportamiento, lo que permite ajustar de manera más precisa los tiempos de paso y optimizar el flujo de tráfico. Además, puede proporcionar información adicional como la identificación de vehículos de emergencia o el comportamiento de los conductores [75].

2.1 Simuladores de tráfico

La ingeniería de transporte y la gestión del tráfico urbano son áreas fundamentales para el desarrollo sostenible de las ciudades. El modelado computacional, mediante simuladores, proporciona teorías, herramientas y tecnologías para explorar la dinámica de fenómenos complejos [49]. Los simuladores de tráfico se clasifican en tres tipos: macroscópicos, microscópicos y mesoscópicos [67]:

- **Simuladores macroscópicos:** Se enfocan en el estado general del tráfico, proporcionando atributos como la densidad de vehículos, la velocidad promedio y el flujo del tráfico [16]. No distinguen componentes individuales, como el cambio de carril de un vehículo. Son útiles por su menor necesidad de parámetros para calcular los datos requeridos [24].
- **Simuladores microscópicos:** Analizan el comportamiento en tiempo y espacio de cada vehículo de forma individual, incluyendo sus interacciones [24]. Definen interacciones y atributos de cada vehículo, como la velocidad, la aceleración y el consumo de combustible [39].
- **Simuladores mesoscópicos:** Son una aproximación intermedia entre los modelos microscópicos y macroscópicos, combinando características de ambos [24]. No rastrean vehículos individuales, sino que se centran en el comportamiento de grupos de entidades de tráfico, representando actividades e interacciones con un nivel de detalle reducido [39].

Estos tres tipos de simuladores ofrecen enfoques complementarios para el estudio y la optimización de sistemas de transporte. La elección del tipo de simulador adecuado depende de las características específicas del caso de estudio y de los objetivos planteados.

2.2 Estudio de los principales simuladores en el mercado

En el transporte, los simuladores de tráfico desempeñan un papel fundamental al permitir explorar la dinámica del tráfico y evaluar diferentes escenarios virtualmente [29]. En la Figura 2 los simuladores se clasifican según su tipo (microscópico, macroscópico y mesoscópico), tipo de licencia, modelado del tiempo en la simulación (discreto y continuo), la posibilidad de configurar los peatones, los tipos de vehículos y los modelos de visualización. A continuación se explican los principales simuladores ampliando la información particular de cada uno, que de manera general se resumen en la Figura 2.

<i>Simulators</i>	Type	Category	Discrete/ continuous	Pedestrians	Vehicle types	Visualization
<i>MITSIM</i>	Microscopic	Open source	Discrete	No	All vehicles are the same	2D
<i>VISSIM</i>	Microscopic	Partially commercial	Continuous	Yes	Buses and trams	Both
<i>SIM-R</i>	Microscopic	Open source	Continuous	No	Buses and trams	3D
<i>SUMO</i>	Microscopic	Commercial	Continuous	No	All vehicles are the same	2D
<i>Paramics</i>	Microscopic	Commercial	Discrete	Yes	Buses and trams	Both
<i>AIMSUM</i>	Micro/ mesoscopic	Commercial	Continuous	Yes	Buses and trams	Both
<i>DTALite</i>	Mesoscopic	Commercial	Discrete	No	All vehicles are the same	Both
<i>TransModeler</i>	Macroscopic and can be macro/ micro	Open source	Discrete	Yes	All vehicles are the same	Both
<i>KSUTraffic</i>	Microscopic	Open source	Discrete	Yes	Buses and trams	2D

Figura 2: Simuladores de tráfico existentes en el mercado [15].

- **Simulador VISSIM:** Se trata de un simulador multimodal y microscópico que puede modelar el comportamiento vehicular teniendo en cuenta condiciones complejas. Este simulador recrea patrones de tráfico, mostrando las iteraciones entre los usuarios de la carretera. Además, permite la toma de decisiones de enrutamiento de vehículos y el uso de matrices Origen-Destino basadas en la densidad vehicular [29]. Proporciona resultados como la efectividad, la demora total, el tiempo de espera, la longitud de cola, el consumo de combustible y la emisión de gases [69].
- **Simulador AIMSUN:** Es un sistema de simulación híbrido mesoscópico-microscópico desarrollado por la Universidad de California, que ayuda en la toma de decisiones en la gestión de tráfico en tiempo real [21]. Entre las características destacadas de AIMSUN se incluyen el uso de matrices Origen-Destino basadas en la oferta y demanda de tráfico, la capacidad de distinguir entre diferentes tipos de vehículos y manejar una variedad de tipos de vías [69][29].
- **Simulador SUMO:** Es un paquete de simulación de tráfico multimodal y microscópico, portátil y de código abierto, apto para su uso en grandes redes viales [32]. SUMO ofrece una gran variedad de características y herramientas que permiten medir el tráfico de carril perimetral, información del viaje y de las rutas de los vehículos [29]. Como herramienta de código abierto implementada en C++, SUMO permite implementar nuevos modelos y controlar la simulación empleando diferentes entornos de programación [37]. Además, SUMO ofrece un alto rendimiento en la simulación a través de la API de TraCI, permitiendo realizar simulaciones en línea. Gracias a su gran comunidad de usuarios, su flexibilidad, facilidad de uso y constante evolución que mantiene sus características actualizadas, SUMO se destaca como una herramienta importante en el campo de la simulación del tráfico [70].
- **Simulador PARAMICS:** Proporciona una representación gráfica detallada de escenarios de tráfico a nivel microscópico [45]. Su objetivo principal

es proporcionar datos y medidas estadísticas esenciales para resolver problemas como la congestión vehicular [55]. Sin embargo, una limitación de esta herramienta es la necesidad de ajustes manuales de los parámetros predeterminados [28].

- **Simulador TRANSMODELER:** Es una plataforma de simulación de tráfico orientada a la planificación y gestión del tráfico, incluyendo situaciones de emergencia. Este *software* con interfaz gráfica permite crear simulaciones que representan el flujo de vehículos, el funcionamiento de las señales de tráfico y el rendimiento general de la red de transporte, incluyendo el transporte público, el tráfico de bicicletas y de camiones [41].

Los simuladores de tráfico han sido ampliamente utilizados en diferentes lugares del mundo con el objetivo de comprender y mejorar la gestión de los sistemas viales. Algunos ejemplos destacados de la aplicación de simuladores de tráfico fueron realizados en: Quito (Ecuador) [47], Lima (Perú) [26], Valencia (España) [62], Málaga (España) [32], Jinhua (China) [73] y Kenitra (Marruecos) [37]. Estos ejemplos muestran cómo los simuladores de tráfico se utilizaron en diferentes ciudades para comprender y mejorar la gestión de los sistemas viales, proporcionando información necesaria para la toma de decisiones en materia de transporte y planificación urbana.

3. Técnicas de optimización

En la actualidad, la optimización se ha convertido en una práctica habitual en el ámbito de la investigación, con el objetivo de encontrar la mejor solución a un problema o sistema, reduciendo costos y tiempo [19]. Muchos problemas de optimización pueden ser abstraídos y modelados como problemas de optimización de funciones, los cuales sirven como referencia para probar algoritmos de optimización [60].

Uno de los ámbitos en los que se aplica la optimización es el flujo vehicular. Una vez construidos los posibles escenarios en el simulador, es necesario aplicar algoritmos de optimización para encontrar el mejor escenario posible, es decir, modificar los elementos o características del sistema para cumplir una función objetivo. La función objetivo puede cubrir un amplio abanico de posibilidades, como la reducción de los tiempos de viaje o el aumento de la durabilidad del combustible [63].

Durante el proceso de optimización, es importante que la solución definitiva constituya una mejora global. Es decir, la solución debe tener en cuenta el escenario al completo y constituir una mejora para el flujo en todo el marco de estudio [23].

En el trabajo se describen 5 algoritmos o métodos de optimización diferentes entre sí, con el fin de proporcionar información sobre sus características. Los algoritmos descritos son: algoritmos genéticos, algoritmo de optimización por enjambre de partículas (PSO, por sus siglas en inglés, *Particle Swarm Optimization*), aprendizaje por refuerzo, optimización por lógica difusa y optimización

por agentes inteligentes. La elección de un algoritmo u otro dependerá de las necesidades de la investigación.

- El **algoritmo genético (AG)** se basa en la selección natural definida por Darwin y utiliza una población de individuos representados por cromosomas para encontrar soluciones óptimas [61]. Los algoritmos genéticos evolucionan mediante operadores genéticos como el cruce y la mutación, seleccionando los mejores candidatos en cada generación [52]. Los AG han sido utilizados en el control de semáforos para reducir el tiempo de espera en las intersecciones de tráfico [14].
- El **algoritmo de optimización por enjambre de partículas (PSO)** se inspira en el comportamiento social de una población o enjambre de organismos que cooperan entre sí en busca de alimento [44]. Cada posible solución se representa como una partícula en el espacio y busca iterativamente las mejores soluciones, actualizando su posición y velocidad en función de las mejores soluciones encontradas hasta el momento [51]. El PSO se ha utilizado en el control de semáforos para determinar programas de ciclos semafóricos y ha mostrado una convergencia rápida a soluciones adecuadas [23].
- El **aprendizaje por refuerzo** es una técnica de aprendizaje automático en la que un agente interactúa con su entorno y toma decisiones para maximizar la recompensa recibida [71]. El agente recopila información a través de ciclos de interacción, utilizando estados, acciones y funciones de recompensa, y aprende qué acciones son beneficiosas o no en diferentes situaciones [36]. En el control de tráfico, el aprendizaje por refuerzo se utiliza para optimizar el control de semáforos y minimizar los tiempos de viaje [18].
- La **lógica difusa** permite manejar datos lingüísticos e inexactos, representando una amplia gama de valores de entrada y salida [68]. Se utiliza en controladores de lógica difusa basados en reglas *IF-THEN* y funciones de pertenencia para tomar decisiones óptimas en diferentes escenarios, como el control de señales de tráfico [42]. La inferencia difusa mapea una entrada a una salida utilizando funciones de pertenencia y reglas *IF-THEN* [73].
- Los **agentes inteligentes autónomos** captan información del entorno a través de sensores y toman decisiones para maximizar los resultados [22]. Utilizan conocimiento y modelos para inferir y razonar, siguiendo un ciclo de observar-pensar-actuar [38]. Los agentes inteligentes se aplican en problemas de transporte para mejorar el rendimiento de la monitorización de los sistemas de transporte [13].

En resumen, los algoritmos genéticos, el algoritmo de optimización por enjambre de partículas, el aprendizaje por refuerzo, la lógica difusa y los agentes inteligentes son métodos de optimización que pueden ser usados en el ámbito del flujo vehicular para mejorar la gestión del tráfico, reducir los tiempos de viaje y aumentar la eficiencia de los sistemas de transporte. Cada algoritmo tiene características y aplicaciones específicas, y la elección depende de las necesidades del estudio o investigación en cuestión.

4. Metodología

En este apartado se presenta la metodología utilizada para abordar el desafío de la optimización vehicular en una intersección de la ciudad de Salamanca. Se emplearon los siguientes elementos clave: coches, semáforos y cámaras, cuya aplicación se realizará a futuro.

La metodología se dividió en varias fases [Figura 3]. En primer lugar, se seleccionó la intersección y se estudiaron los agentes de tráfico involucrados. Luego, se creó y ajustó el modelo de simulación utilizando SUMO, OpenStreetMap y netedit. SUMO fue elegido debido a su capacidad para modelar el tráfico vehicular de manera precisa y su facilidad de implementación empleando la API de TraCI [32][43].

Para la optimización del flujo vehicular, se utilizó el algoritmo PSO debido a su facilidad de implementación en Python [44]. La metodología se desarrolló en varias fases, incluyendo la creación del modelo de simulación, la implementación del algoritmo PSO a través de *scripts* en Python, y la ejecución y análisis de los resultados obtenidos.

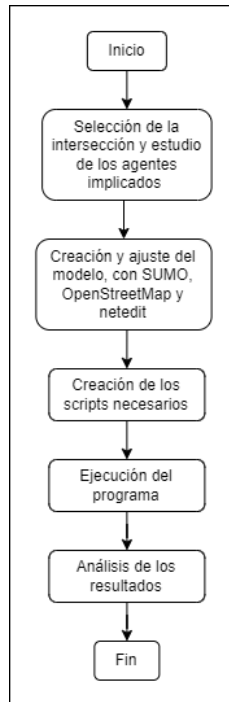


Figura 3: Metodología empleada en la realización del proyecto.

En los siguientes apartados se detallan las tres primeras fases de la metodología, mientras que la ejecución del programa y el análisis de los resultados se presentarán en el apartado de caso de estudio y resultados.

4.1 Selección de la intersección y estudio de los agentes implicados

Para lograr una representación precisa y efectiva del tráfico en el entorno simulado, es esencial seleccionar una intersección que cumpla con ciertos criterios clave. En primer lugar, se deben considerar los niveles de congestión y tránsito vehicular actuales. Además, la ubicación estratégica de intersección también debe ser tenida en cuenta. Factores como la proximidad a instituciones educativas, centros médicos o centros de eventos, son aspectos relevantes que considerar.

En este trabajo, se han considerado tres elementos clave para la optimización vehicular de una intersección: los semáforos, los coches y las cámaras de tráfico, que se plantean como una futura adicción al sistema. Para la realización de este proyecto no se han tenido en cuenta a los peatones.

Los semáforos atraviesan diferentes fases que determinan cuándo se permite el paso a los diferentes carriles. Cada fase corresponde a un conjunto específico de movimientos permitidos en la intersección, como el paso de los vehículos en una dirección u otra, el tiempo de espera o la transición a la fase de “ámbar.” antes de iniciar un nuevo ciclo. La configuración de las fases se adapta a las necesidades de la intersección, siguiendo un ciclo básico de verde-ámbar-rojo. La coordinación adecuada de los semáforos en una intersección es esencial para garantizar una mejor fluidez del tráfico. Esto implica establecer una secuencia lógica de fases que permita un movimiento continuo y seguro de los vehículos a través de la intersección. La coordinación se logra ajustando la duración de las fases de cada semáforo de manera que se minimice el tiempo de espera y se mejore la eficiencia del cruce en su conjunto. Es importante considerar tanto las condiciones actuales de tráfico como las proyecciones y tendencias históricas al coordinar los semáforos [66].

Los coches representan los vehículos que circulan por la intersección y se caracterizan por atributos como velocidad, aceleración, posición y destino. La interacción entre los coches y los semáforos permite identificar los patrones de tráfico y evaluar la eficiencia del cruce. Durante la simulación mediante SUMO, se consideran los comportamientos individuales de los coches, así como las intersecciones entre ellos y con los semáforos, para evaluar el impacto de las estrategias de optimización propuestas.

En cuanto a las cámaras de tráfico, se plantea su utilización como un elemento adicional de monitoreo y control del tráfico. Estas cámaras capturarán datos en tiempo real sobre el flujo vehicular, las condiciones de tráfico y el comportamiento de los conductores. La información recopilada por las cámaras se utiliza para ajustar y mejorar la configuración de los semáforos, así como para futuros estudios de análisis y optimización de tráfico. Se propone la utilización de YOLO (*You Only Look Once*) para la implementación de cámaras, lo cual implica el entrenamiento del modelo, la implementación de cámaras, el seguimiento y el

análisis en tiempo real. Estos resultados se integrarán con el algoritmo PSO ya implementado en el proyecto [76].

La selección de una intersección adecuada y el estudio de los agentes implicados son partes fundamentales de la metodología empleada en este proyecto de optimización vehicular.

4.2 Creación y ajuste de modelo, con SUMO, OpenStreetMap y netedit

En este apartado se aborda el proceso de generación del modelo de la intersección seleccionada utilizando el paquete SUMO y las herramientas complementarias OpenStreetMap y netedit. En primer lugar, se obtienen los datos necesarios de OpenStreetMap, una plataforma colaborativa de datos geoespaciales, seleccionando el área de interés y accediendo a su configuración. OpenStreetMap proporciona información geoespacial configurable, incluyendo la geometría de las vías, los nombres de calles, los límites de velocidad, los semáforos y las paradas de transporte público [36]. En segundo lugar, en caso de que sea necesario, se editan y refinan los datos obtenidos utilizando la herramienta netedit; es una herramienta gráfica integrada en SUMO para la edición de modelos de simulación de tráfico. Con netedit, se puede editar los elementos de la red vial, como carreteras, intersecciones, carriles, semáforos y zonas peatonales, definiendo su geometría exacta, velocidades máximas y tipos de vehículos permitidos. Por último, se configura la simulación en SUMO utilizando los archivos XML generados con OpenStreetMap y netedit. SUMO permite visualizar y ejecutar la simulación, agregando vehículos con diferentes atributos, estableciendo rutas predefinidas y simulando el transporte público o redes de trenes. La simulación resultante se almacena en archivos XML editables para su uso futuro [47].

Al combinar los datos obtenidos con OpenStreetMap con las herramientas SUMO y netedit, se obtiene una representación realista y detallada de la intersección en el entorno de simulación. Este modelo generado proporciona una buena base para evaluar y optimizar el comportamiento del tráfico, así como para implementar estrategias de control y optimización en la intersección seleccionada.

4.3 Creación de los *scripts* necesarios

En este apartado se desarrolla el método de optimización del ciclo semafórico de la intersección seleccionada. Para ello, se emplea la API de TraCI para interactuar con los mapas generados previamente, siguiendo una estructura cliente-servidor [Figura 4]. SUMO actúa como servidor, mientras que el *script* desarrollado actúa como cliente, enviando las solicitudes necesarias para la interacción [62].

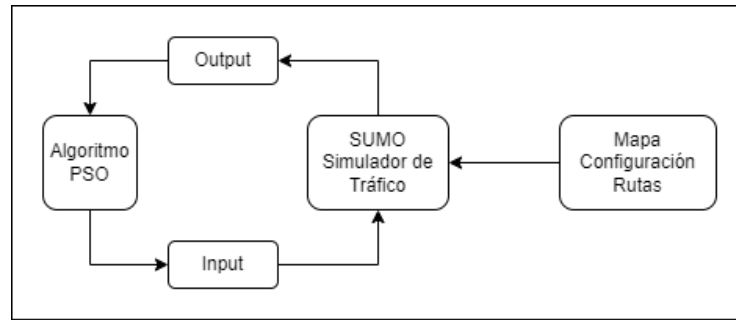


Figura 4: Flujo de trabajo con PSO + SUMO.

El objetivo principal es mejorar el rendimiento del tráfico en la intersección, reduciendo el tiempo de espera de los vehículos, el tiempo de viaje y minimizando la congestión. Se utiliza el algoritmo de optimización PSO, implementado con la biblioteca “pyswarm”, aunque también podrían emplearse otros métodos de optimización como los algoritmos genéticos. Para emplear el algoritmo PSO se emplean los siguientes parámetros [77]:

- **Tiempo mínimo de la fase:** Se establece en 5 segundos para evitar tiempos de semáforo demasiado bajos que puedan afectar el flujo vehicular.
- **Tiempo máximo de la fase:** Se fija en 60 segundos para evitar tiempos de semáforo excesivamente altos que puedan causar largas esperas a los conductores.
- **Tamaño del enjambre:** Se utilizan 50 partículas, que representan las soluciones candidatas, para realizar la optimización.
- **Número máximo de iteraciones:** Se establece en 99, lo que implica 100 iteraciones en total al ser la primera el número cero.
- **Coefficiente de inercia:** Se fija en 0.8 para equilibrar la influencia de la velocidad anterior de una partícula en su movimiento actual, favoreciendo tanto la exploración global como la explotación local.
- **Coefficiente cognitivo:** Se selecciona un valor de 1.0 para enfatizar la influencia de la mejor solución personal de una partícula en su movimiento actual.
- **Coefficiente social:** Se emplea un valor de 1.0 para resaltar la importancia de la mejor solución global, que representa la mejor posición entre todas las partículas, en el movimiento actual de una partícula.

En la “función objetivo” empleada por el algoritmo PSO se crean vehículos en todas las rutas con una velocidad inicial de 0 m/s y se realiza una simulación en 1000 pasos para obtener una visión global de la intersección. Se considera que hay congestión cuando la velocidad es inferior a 5 m/s. Una vez finalizados los 1000 pasos, se emplea una operación matemática con el objetivo de que su resultado sea lo más cercano a cero posible, es decir, el resultado óptimo entre

todas las posibilidades analizadas [63]. Esta operación matemática se expresa mediante la siguiente métrica de desempeño:

$$performance_metric = \frac{total_waiting_time + total_travel_time}{total_congestion + 1}$$

Esta función tiene en cuenta el tiempo de espera de los vehículos en la intersección (*total_waiting_time*) y lo combina con el tiempo de viaje total (*total_travel_time*). Mediante esta técnica, se obtiene el valor que cuantifica el rendimiento del tráfico. Por último, este valor se divide entre la congestión total (*total_congestion*) +1 con el fin de evitar divisiones entre 0.

Los resultados obtenidos, incluyendo los tiempos óptimos y la métrica de desempeño de los semáforos, se almacenan en un archivo TXT. También se registran todos los valores de la métrica de desempeño (*performance_metric*) calculados durante la ejecución en un archivo CSV, y se genera una gráfica que muestra la evolución de estos valores.

Para finalizar, se automatizó la ejecución del código para obtener 10 resultados que permitan comparar los resultados y validar manualmente la eficiencia del código en busca de su optimización.

5. Caso de estudio

En este caso de estudio, se aplica la metodología desarrollada a una intersección real, específicamente en la intersección entre el Paseo de San Vicente (N-630) con la Avenida de los Maristas y con la Calle Espejo, en Salamanca [Figura 5]. Esta intersección representa un desafío en términos de optimización del tráfico debido a la alta afluencia de vehículos y la necesidad de garantizar un flujo eficiente y seguro. A lo largo del caso de estudio, se implementarán las técnicas y herramientas descritas anteriormente para mejorar la eficiencia y reducir los tiempos de espera en la intersección.



Figura 5: Vista aérea de la intersección seleccionada.

5.1 Selección de la intersección y estudio de los agentes implicados

La intersección seleccionada para este estudio se encuentra en una ubicación estratégica cerca del centro histórico de la ciudad de Salamanca. Debido a su ubicación y la presencia de instituciones y áreas de alto tránsito, esta intersección experimenta un alto volumen de tráfico, especialmente durante las horas punta. La intersección está compuesta por tres calles reguladas con un límite máximo de velocidad de 30 km/h y cuenta con un total de seis semáforos que regulan el tráfico vehicular en cada vía.

Las tres calles involucradas en la intersección son [Figura 5] : el Paseo de San Vicente, la Avenida de los Maristas y la Calle Espejo. El Paseo de San Vicente tiene dos carriles que se dirigen hacia la zona de hospitales y tres hacia el Campo de San Francisco, y cuenta con un carril de espera semaforizado que permite el tránsito hacia la Calle Espejo. La Avenida de los Maristas tiene cuatro carriles que desembocan en la intersección, con sentido hacia el Paseo de San Vicente (zona de hospitales) o hacia la Calle Espejo. La Calle Espejo se compone de dos carriles de sentido único para abandonar la intersección.

Dependiendo de la posición del vehículo en la intersección, existen siete rutas posibles que determinan la trayectoria a seguir. La coordinación adecuada de los semáforos es fundamental para permitir un movimiento fluido y seguro en todas estas rutas.

5.2 Creación y ajuste del modelo, con SUMO, OpenStreet-Map y netedit

Inicialmente, se ejecutó OpenStreetMap con el fin de seleccionar y obtener el modelo de la intersección mediante su descarga. Se configuró OpenStreetMap para obtener únicamente los datos relevantes para la intersección, como semáforos, vehículos y carreteras.

Una vez obtenido el modelo inicial, se procedió a ajustarlo, empleando la herramienta netedit. Esta herramienta permitió realizar modificaciones en los semáforos existentes en la intersección. En el cruce en cuestión, se identificaron un total de seis semáforos que se distribuyeron en cuatro grupos semafóricos. Se ha considerado como grupo semafórico a uno o más semáforos que actúan de manera coordinada en la regulación del tráfico en una intersección. Para una mejor organización y coordinación se agruparon según lo expuesto en la Tabla 1. Además en la Figura 6 se observan los cuatro grupos semafóricos representados con un símbolo de semáforo.

Tabla 1: Semáforos y su división en grupos semafóricos.

Semáforo	Ubicación	Grupo semafórico
Semáforo 1	Semáforo de Paseo de San Vicente en dirección a la zona de hospitales, entrando en la intersección.	Grupo 1
Semáforo 2	Semáforo Avenida de los Maristas, antes de entrar en la intersección.	Grupo 1
Semáforo 3	Paseo de San Vicente en dirección a la zona de hospitales, saliendo de la intersección.	Grupo 2
Semáforo 4	Paseo de San Vicente en dirección al Campo de San Francisco, entrando en la intersección.	Grupo 3
Semáforo 5	Semáforo situado en el carril central de espera del Paseo de San Vicente. Regula el paso a Calle Espejo.	Grupo 3
Semáforo 6	Se encuentra a la entrada de Calle Espejo, saliendo de la intersección.	Grupo 4

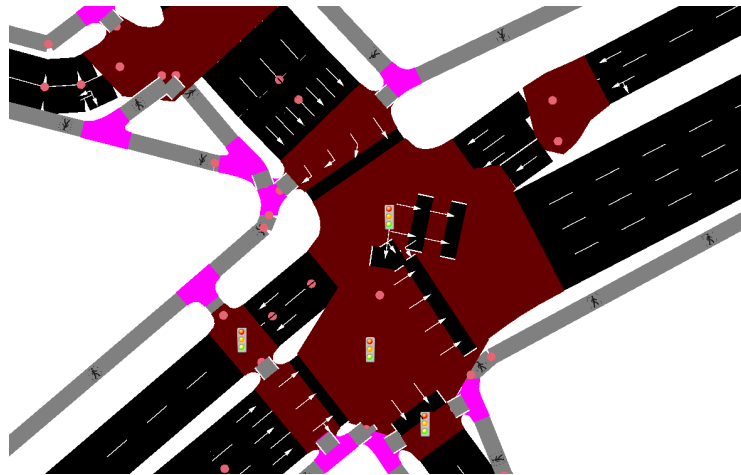


Figura 6: Diseño de los grupos semafóricos en netedit.

Un ciclo semafórico completo se divide en una secuencia ordenada de fases. Cada fase representa un estado específico de los semáforos, indicando qué carriles tienen luz verde, amarilla o roja en un momento dado. Mediante código XML se define el comportamiento inicial de los semáforos en la intersección a través de la especificación de las diferentes fases. Este código viene dado por defecto al obtener los datos de OpenStreetMap y modificarlos con netedit.

En el siguiente código XML, se definen cuatro grupos semafóricos identificados por los elementos “*tlLogic*” con sus respectivos identificadores únicos (atri-

buto “id”). Los cuatro grupos semafóricos se representan en el código ordenados según la numeración aportada en la Tabla 1.

Definición de los ciclos semafóricos en el archivo XML

```
<tlLogic id="458496572" type="static" programID="0" offset="0">
  <phase duration="42" state="GGrrrrr"/>
  <phase duration="3" state="yyrrrrr"/>
  <phase duration="42" state="rrGGGGG"/>
  <phase duration="3" state="rryyyyy"/>
</tlLogic>
<tlLogic id="1373663392" type="static" programID="0" offset="0">
  <phase duration="82" state="GG"/>
  <phase duration="3" state="yy"/>
  <phase duration="5" state="rr"/>
</tlLogic>
<tlLogic id="1373663389" type="static" programID="0" offset="0">
  <phase duration="42" state="rGGGGG"/>
  <phase duration="3" state="ryyyyy"/>
  <phase duration="42" state="GGrrrrr"/>
  <phase duration="3" state="yyrrrrr"/>
</tlLogic>
<tlLogic id="1373663387" type="actuated" programID="0" offset="0">
  <phase duration="82" state="GG"/>
  <phase duration="3" state="yy"/>
  <phase duration="5" state="rr"/>
</tlLogic>
```

Los estados de los semáforos se identifican mediante las letras “G” para luz verde, “y” para luz amarilla y “r” para luz roja. La duración de cada fase se especifica con el atributo “duration”. En el código XML anterior se representa una coordinación adecuada de los colores de los semáforos en la intersección. En el caso del primer y del tercer grupo semafórico, existen cuatro fases que coordinan cada uno dos semáforos a la vez, esto explica que el tiempo asignado para las diferentes fases sea el mismo, a excepción del ámbar que siempre son 3 segundos.

En el atributo “state” se definen los colores de los semáforos para la fase determinada. Este atributo indica que carriles tienen luz verde, amarilla o roja. En la Figura 7 se numeran los diferentes carriles según su orden establecido en el código XML anterior con el fin de facilitar el entendimiento de las diferentes combinaciones de fases.

Como se ha mencionado anteriormente, el código XML presentado aporta una configuración inicial que se modificará posteriormente durante la ejecución del *script* de Python. Esta modificación es necesaria para que el algoritmo PSO pueda probar las posibles soluciones y encontrar los tiempos óptimos. Además, en este caso de estudio se han configurado los grupos semafóricos para que ciertas fases tengan obligatoriamente la misma duración evitando bloqueos en el tráfico. Esto ocurre en la primera y tercera fase de los grupos semafóricos 1 y 3.



Figura 7: Número de carril según el grupo semafórico.

Una vez configurados los grupos semafóricos y sus fases, se procedió a definir las rutas disponibles para los vehículos en la intersección. Esto se realizó empleando la interfaz gráfica de la herramienta netedit. El resultado es un fichero XML en el que se definen las siete rutas existentes mediante los identificadores de los tramos que las componen. En la Tabla 2 se exponen todas las rutas clasificadas según su entrada y salida de la intersección.

En la Figura 8 se presentan todas las posibles rutas previamente definidas, las cuales se muestran en color amarillo. Además, se representa el diseño del mapa utilizando un código de colores específico de netedit. En este código, los tramos de la carretera se muestran en color verde, las intersecciones entre los tramos se destacan en granate y áreas peatonales, que no son relevantes en este caso, se representan en color fucsia. De esta manera, se obtiene una representación visual clara y distintiva de los diferentes elementos de las rutas.

Tabla 2: Rutas de los vehículos en la intersección.

Ruta	Entrada	Salida
Ruta 1	Avenida de los Maristas.	Paseo de San Vicente (Dirección hospitales).
Ruta 2	Avenida de los Maristas.	Calle Espejo.
Ruta 3	Avenida de los Maristas.	Paseo de San Vicente (Dirección Campo de San Francisco).
Ruta 4	Paseo de San Vicente (Dirección Campo de San Francisco).	Calle Espejo.
Ruta 5	Paseo de San Vicente (Dirección Campo de San Francisco).	Paseo de San Vicente (Dirección Campo de San Francisco).
Ruta 6	Paseo de San Vicente (Dirección zona de hospitales).	Paseo de San Vicente (Dirección hospitales).
Ruta 7	Paseo de San Vicente (Dirección zona de hospitales).	Calle Espejo.

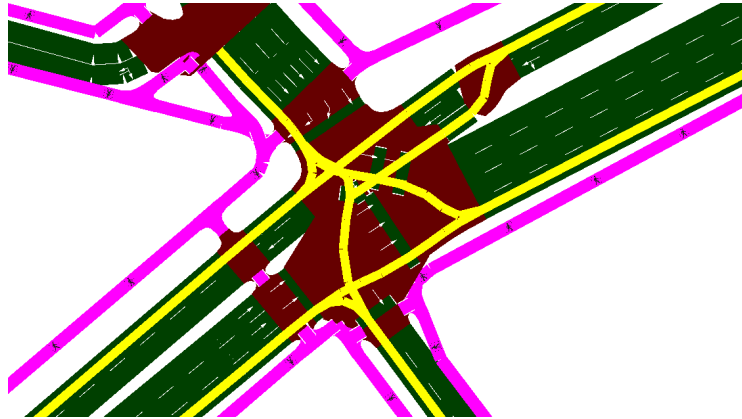


Figura 8: Configuración de las rutas en netedit.

Para validar visualmente el modelo de la intersección con la configuración de los semáforos y las rutas incorporadas, se utilizó la herramienta SUMO. En esta fase, se comparó el modelo creado con los datos reales previamente analizados para asegurar que la configuración se aplicara correctamente. La Figura 9 muestra el resultado final de la simulación gráfica de la intersección empleando SUMO.

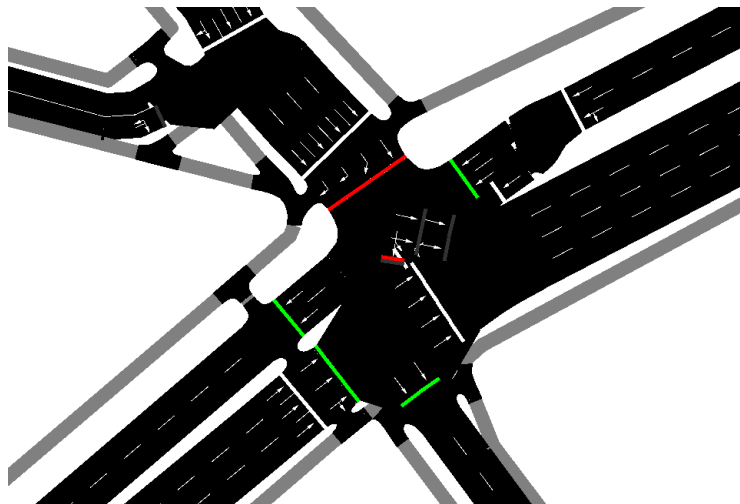


Figura 9: Resultado final de la simulación en SUMO.

Una vez obtenido el modelo de simulación final, se incorporó el archivo de configuración de la simulación y el archivo de rutas al código desarrollado. Es necesario mencionar que en el archivo de configuración están incluidas las configuraciones de los grupos semafóricos. Luego, se ejecutó el *script* que realizó 10 ejecuciones del programa de optimización, obteniendo los resultados del proyecto. De cada ejecución, se obtuvieron los mejores tiempos en segundos para las fases de todos los semáforos calculados con la métrica de desempeño (*performance_metric*), se generó un archivo CSV con todos los valores de *performance_metric* y se creó una gráfica que muestra la evolución de esta métrica a lo largo de las iteraciones [Figuras 11a, 11b, 12a, 12b, 13a, 13b, 14a, 14b, 15a, 15b].

6. Resultados y discusión

El objetivo principal de este trabajo es realizar la optimización de la gestión semafórica de la intersección Paseo de San Vicente – Avenida de los Maristas – Calle Espejo, de la ciudad de Salamanca, mediante la optimización con el algoritmo PSO, con el fin de mejorar la eficiencia del flujo vehicular y reducir la congestión de dicha intersección. Para llevar a cabo esta optimización, se establecieron límites inferiores y superiores para la duración de las fases “verde” y “rojo” en cada grupo semafórico. En este caso, se fijó un tiempo mínimo de 5 segundos y un tiempo máximo de 60 segundos. Estos límites fueron determinados empleando el simulador SUMO considerando la seguridad vial, la fluidez del tráfico y las restricciones establecidas.

El algoritmo PSO se configuró para realizar 100 iteraciones con un tamaño de enjambre de 50 partículas, lo que resultó en un total de 5.000 intentos de

búsqueda para encontrar la configuración óptima. En cada uno de estos intentos se planteó una configuración semafórica y se introdujeron 35 vehículos en la intersección, generando un total de 5.000 configuraciones diferentes y 175.000 vehículos en cada simulación. Estos vehículos representaron el flujo de tráfico en la intersección y permitieron evaluar el rendimiento de todos los posibles tiempos de los semáforos. Se realizaron 10 ejecuciones de este algoritmo con el fin de obtener una variedad significativa de resultados que permita validar los tiempos obtenidos.

La Tabla 3 muestra los resultados obtenidos mediante las 10 ejecuciones del programa. En ella se observan los tiempos de duración de las fases “verde” y “rojo” en segundos para cada grupo semafórico (GS, en la Tabla 3), junto con el valor *performance_metric*. Es importante destacar que para los grupos semafóricos 1 y 3, solo se obtiene un tiempo de duración de la fase “verde” o “rojo”, respectivamente. Esto se debe a que, en estos grupos semafóricos, durante el mismo tiempo, ciertos carriles están en verde mientras que otros están en rojo, y viceversa. Por otro lado, en los grupos semafóricos 2 y 4, se obtuvieron tiempos diferentes para las fases “verde” y “rojo”, ya que estos grupos semafóricos solo coordinan, cada uno, un único semáforo para dos carriles.

Tabla 3: Tiempos óptimos obtenidos para los diferentes grupos semafóricos.

Iteración	GS 1		GS 2		GS 3		GS 4		<i>performance_metric</i>
	Verde	Rojo	Verde	Rojo	Verde	Rojo	Verde	Rojo	
1	51	40	25	48	18	5	64,28		
2	5	60	5	51	60	5	59,47		
3	5	59	18	60	56	5	61,452		
4	7	14	5	14	57	8	59,031		
5	5	60	7	41	52	5	60,161		
6	5	43	5	57	60	10	59,834		
7	7	60	8	52	20	5	60,387		
8	5	56	21	60	28	5	61,456		
9	9	54	7	18	58	5	60,19		
10	5	40	5	48	60	5	59,552		

Para analizar los resultados de manera visual, se presenta el siguiente gráfico de dispersión [Figura 10]. Este gráfico muestra cómo se distribuyen los tiempos de las diferentes fases de los semáforos en cada ejecución y cómo se acercan a una distribución uniforme, lo que indica una buena optimización del modelo logrando siempre resultados más o menos similares.

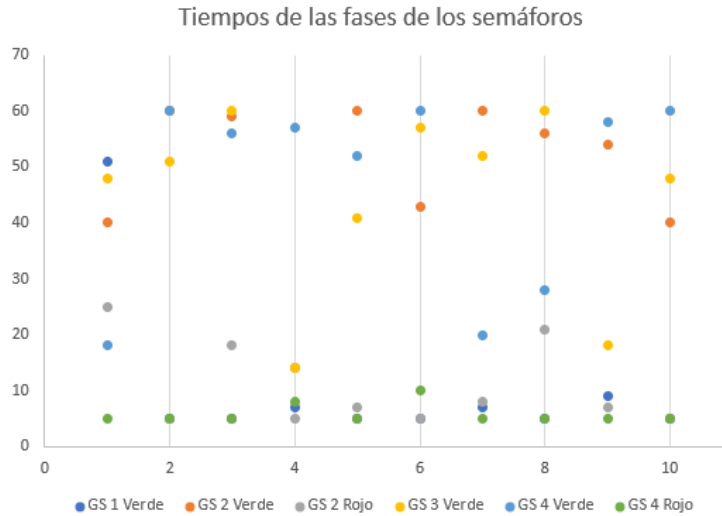


Figura 10: Gráfica de dispersión. Tiempos de las fases de los semáforos.

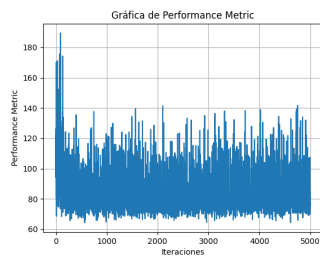
Como se observa en la gráfica de dispersión [Figura 10] los resultados para cada grupo semafórico (GS) [Tabla 3] son los siguientes:

- **GS 1:** Se observa una variabilidad de los tiempos de duración. Este tiempo corresponde a el tiempo establecido para la duración de ambas fases complementarias. Los valores son generalmente bajos, en torno a 5-9 segundos, a excepción de un valor de 51 segundos que se aleja de la tendencia general. Además, este valor coincide con la *performance_metric* más alta por lo que se considera el valor menos eficiente, aunque se aceptaría como medición válida.
- **GS 2:** Los tiempos de duración de la fase “verde” rondan entre 40 y 60 segundos, a excepción de un valor de 14 que se sale del rango. Pese a que no sigue la tónica general, el valor de *performance_metric* en esa ejecución es bueno, por lo que esa configuración de semáforos se considera válida. En lo que respecta a los tiempos de duración de la fase “rojo” se encuentran en un rango de 5-25 segundos predominando los valores por debajo de 10 segundos.
- **GS 3:** Se observa una variabilidad de los tiempos de duración. Este tiempo corresponde a el tiempo establecido para la duración de ambas fases complementarias. El rango se establece entre 14 y 60 segundos, situando el grueso de los valores por encima de 40 segundos.
- **GS 4:** Los tiempos de duración de la fase “verde”, en este caso, la distribución de los valores se encuentra entre 18 y 60 segundos distribuidos con 3 valores entorno a 20 y 7 valores entorno a 60. Los resultados de *performance_metric* validan ambos grupos por lo que todas las configuraciones serían

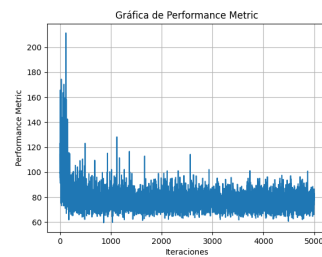
válidas. En cuanto a los valores de tiempo “rojo” el rango es más uniforme estando situado entre los 5 y los 10 segundos.

En conclusión, se observa que los resultados obtenidos tienen buenos datos de *performance_metric*. Esto queda validado al observar que en 10 ejecuciones diferentes del algoritmo los valores de esta métrica están en el rango 59,031 a 64,280. La variabilidad en los tiempos de duración de las fases “verde” y “rojo” muestra que se han explorado diferentes opciones para encontrar la configuración óptima de los semáforos. Además, como se observa en las Figuras 11a, 11b, 12a, 12b, 13a, 13b, 14a, 14b, 15a, 15b, la métrica de desempeño refleja una mejora consistente en el rendimiento del tráfico a lo largo de las iteraciones del algoritmo PSO. Estos resultados respaldan la efectividad de la optimización realizada.

Finalmente, como resultado del proyecto también se obtienen 10 gráficas que representan todos los resultados del valor *performance_metric* de cada una de las 10 ejecuciones [Figuras 11a, 11b, 12a, 12b, 13a, 13b, 14a, 14b, 15a, 15b]. Tal y como se ha desarrollado en apartados anteriores, el algoritmo PSO emplea esta métrica para hallar la mejor opción de duración de las fases de los semáforos, es decir, el valor más bajo de *performance_metric*. La mejor métrica de cada una de las iteraciones es la que se representó en la Tabla 3.

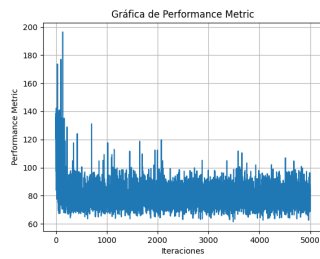


(a) Ejecución 1.

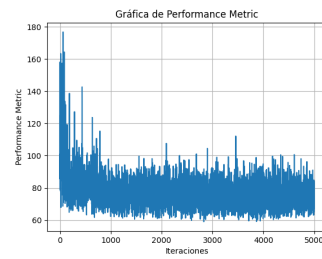


(b) Ejecución 2.

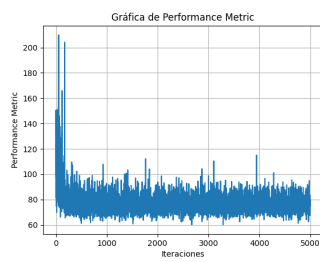
Figura 11: Valores de *performance_metric*.



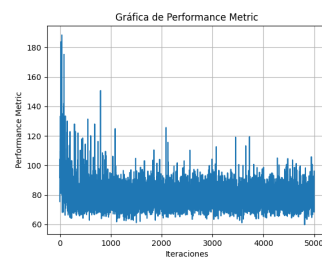
(a) Ejecución 3.



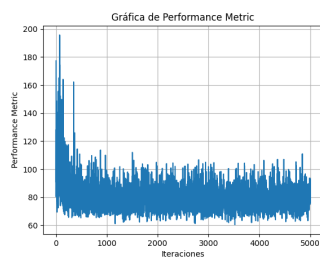
(b) Ejecución 4.

Figura 12: Valores de *performance_metric*.

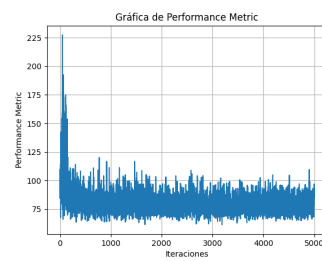
(a) Ejecución 5.



(b) Ejecución 6.

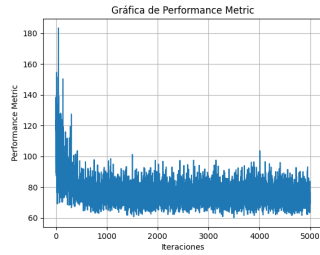
Figura 13: Valores de *performance_metric*.

(a) Ejecución 7.

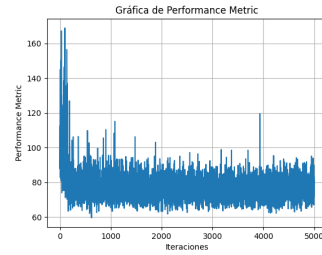


(b) Ejecución 8.

Figura 14: Valores de *performance_metric*.



(a) Ejecución 9.



(b) Ejecución 10.

Figura 15: Valores de *performance_metric*.

7. Conclusiones y líneas futuras de trabajo

En conclusión, este trabajo ha logrado alcanzar el objetivo principal planteado; optimizar la gestión semafórica de una intersección de la ciudad de Salamanca. Además, se han abordado objetivos secundarios como la revisión exhaustiva de los principales simuladores de tráfico disponibles en el mercado, el análisis de casos de estudio reales aplicados con simulador en la gestión del tráfico, la investigación sobre diferentes técnicas de optimización y la selección de un cruce específico en Salamanca como caso de estudio.

A lo largo del trabajo, se ha empleado una metodología que ha permitido recopilar datos relevantes de la intersección seleccionada y crear y ajustar el modelo de simulación definiendo los parámetros y las restricciones necesarias. Se ha empleado el algoritmo PSO como técnica de optimización para ajustar los tiempos de los semáforos y encontrar una configuración óptima para la duración de las fases de los semáforos en la simulación de la intersección.

Las optimizaciones realizadas han permitido ajustar los tiempos de los semáforos de manera efectiva reduciendo los valores de *performance_metric*. Con ello, se han adaptado los tiempos a las necesidades específicas de la intersección planteadas en la simulación y logrando una distribución más equitativa del flujo vehicular.

En resumen, este trabajo ha demostrado la importancia de utilizar simuladores de tráfico y técnicas de optimización en la gestión del tráfico para mejorar la eficiencia de la gestión vial. La aplicación de un enfoque metodológico, junto con el uso del parque de simulación SUMO y el algoritmo PSO, ha proporcionado resultados que abren las puertas para futuras investigaciones en el campo de la gestión del tráfico urbano.

Como línea de trabajo futuro, se sugiere explorar la implementación en un marco real de este trabajo, empleando tecnologías de visión artificial que permitiría obtener datos en tiempo real y lograr una optimización dinámica de los semáforos. Además, sería interesante la exploración de diferentes simuladores de tráfico, aparte de SUMO para evaluar su efectividad y comparar los resultados

obtenidos. También se podrían investigar otras técnicas de optimización, como algoritmos genéticos o basados en inteligencia artificial.

Referencias

1. Akhtar, Mahmuda and Moridpour, Sara. "A review of traffic congestion prediction using artificial intelligence." *Journal of Advanced Transportation* 2021, 1-18. ISSN: 2042-3195.
2. Reed, Trevor. "Inrix global traffic scorecard." *Journal Article* 2019.
3. Thomson, Ian and Bull, Alberto. *La congestión del tránsito urbano: causas y consecuencias económicas y sociales*. Cepal, 2001. ISBN: 9213218656.
4. Afrin, Tanzina and Yodo, Nita. "A survey of road traffic congestion measures towards a sustainable and resilient transportation system." *Sustainability* 12.11 (2020): 4660. ISSN: 2071-1050.
5. Wang, Yanli, Zhu, Xiaoyu, Li, Linbo, and Wu, Bing. "Reasons and countermeasures of traffic congestion under urban land redevelopment." *Procedia-Social and Behavioral Sciences* 96 (2013): 2164-2172. ISSN: 1877-0428.
6. Falcocchio, John C. and Levinson, Herbert S. *Road traffic congestion: a concise guide*. Springer, 2015. Volume: 7. ISBN: 3319151657.
7. Robinson, R. Michael, Collins, Andrew J., Jordan, Craig A., Foytik, Peter, and Khattak, Asad J. "Modeling the impact of traffic incidents during hurricane evacuations using a large scale microsimulation." *International journal of disaster risk reduction* 31 (2018): 1159-1165. ISSN: 2212-4209.
8. Tamakloe, Reuben, Hong, Jungyeol, and Tak, Jihoon. "Determinants of transit-oriented development efficiency focusing on an integrated subway, bus and shared-bicycle system: Application of Simar-Wilson's two-stage approach." *Cities* 108 (2021): 102988. ISSN: 0264-2751.
9. Uribe, Sandra Liliانا. "Propuesta metodológica para el diseño de intersecciones urbanas." *INVENTUM* 4.7 (2009): 46-54. ISSN: 2590-8219.
10. Pérez, Sócrates Muñoz, Reátegui, José Salcedo, and Mendoza, Arturo Sotomayor. "Contaminación ambiental producida por el tránsito vehicular y sus efectos en la salud humana: revisión de literatura." *INVENTUM* 16.30 (2021): 20-30. ISSN: 2590-8219.
11. Estupiñan, Nicolás. "Movilidad compartida: un cambio de paradigma para la equidad y la inclusión." *Transporte y desarrollo en América Latina* 1.1 (2018): 9-10.
12. Tráfico, Dirección General de. *Anuario Estadístico del Ministerio del Interior 2020*. 2020. Type: Dataset.
13. Ahmed, A. I. and El. seed, A. G. "Intelligent Traffic Light Based on Multi-agent System." *2013 International Conference on Advanced Computer Science Applications and Technologies* (2013): 89-92. DOI: 10.1109/ACSAT.2013.25. Type: Conference Proceedings.
14. Ahmed, E. M. "Continuous Genetic Algorithm for Traffic Signal Control." *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)* (2018): 1-5. DOI: 10.1109/ICCCEEE.2018.8515770. Type: Conference Proceedings.
15. Al-Nabhan, N., AlDuhaim, M., AlHussan, S., Abdullah, H., AlHaid, M., and Al-Duhaishi, R. "KSUTraffic: A Microscopic Traffic Simulator for Traffic Planning in Smart Cities." *Cmc-Computers Materials & Continua* 68.2 (2021): 1831-1845. ISSN: 1546-2218. DOI: 10.32604/cmc.2021.012231.

16. Backfrieder, Christian, Mecklenbräuker, Christoph F., and Ostermayer, Gerald. "TraffSim—A Traffic Simulator for Investigating Benefits Ensuing from Intelligent Traffic Management." *IEEE* (ISBN: 1479925780).
17. Ballesteros-Pacheco, J. C. "Detección y mejora de tiempos en el flujo vehicular con semáforo inteligente controlador de tráfico para disminuir las emisiones de CO₂." *Revista de Ingeniería* 1.1 (2017): 1-8.
18. Bouktif, Salah, Cheniki, Abderraouf, Ouni, Ali, and El-Sayed, Hesham. "Deep reinforcement learning for traffic signal control with consistent state and reward design approach." *Knowledge-Based Systems* 267 (2023): 110440. ISSN: 0950-7051. DOI: <https://doi.org/10.1016/j.knosys.2023.110440>.
19. Caballero, José A. and Grossmann, Ignacio E. "Una revisión del estado del arte en optimización." *Revista Iberoamericana de Automática e Informática Industrial RIAI* 4.1 (2007): 5-23. ISSN: 1697-7912. DOI: [https://doi.org/10.1016/S1697-7912\(07\)70188-7](https://doi.org/10.1016/S1697-7912(07)70188-7).
20. Carballo, Laura, Villagra, Norma Andrea, and Pandolfi, Daniel Raul. "Funciones de evaluación de calidad para un tránsito inteligente." *Informe Científico Técnico UNPA* 13.1 (2021): 77-94. ISSN: 1852-4516.
21. Casas, Jordi, Ferrer, Jaime L., Garcia, David, Perarnau, Josep, and Torday, Alex. "Traffic simulation with aimsun." *Fundamentals of traffic simulation* (2010): 173-232. ISSN: 1441961410.
22. Castán, José A., Ibarra, Salvador, Laria, Julio, Guzmán, Javier, and Castán, Emilio. "Control de tráfico basado en agentes inteligentes." *Polibits* 50 (2014): 61-68. ISSN: 1870-9044.
23. Celtek, Seyit Alperen, Durdu, Akif, and Ali, Muzamil Eltejani Mohammed. "Real-time traffic signal control with swarm optimization methods." *Measurement* 166 (2020): 108206. ISSN: 0263-2241. DOI: <https://doi.org/10.1016/j.measurement.2020.108206>.
24. Chaves Pinzón, César Alejandro. "Simulación de escenarios de tráfico vehicular para la ciudad de Bogotá." 2016. Type: Journal Article.
25. Chávez Fuentes, Carla Patricia. "Sistema de semaforización inteligente para el control de flujo vehicular mediante el procesamiento digital de imágenes." 2015. Type: Journal Article.
26. Cisneros, C., Morales, L., Silvera, M., and Campos, F. "Design of mini roundabouts with adaptive traffic lights to improve vehicular travel times and roundabout capacity in metropolitan cities." *2022 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONIITI)* (2022): 1-6. ISBN: 2539-4320. DOI: 10.1109/CONIITI57704.2022.9953666. Type: Conference Proceedings.
27. Delgado López, Kevin Paul. "PROTOTIPO DE SEMÁFORO INTELIGENTE PARA CONTROL DE TRÁNSITO TERRESTRE MEDIANTE SENSORES DE PROXIMIDAD PARA EL CANTÓN MONTECRISTI." 2023. Type: Journal Article.
28. Dowling, Richard, Skabardonis, Alexander, Halkias, John, McHale, Gene, and Zammit, Grant. "Guidelines for calibration of microsimulation models: framework and applications." *Transportation Research Record* 1876.1 (2004): 1-9. ISSN: 0361-1981.
29. Ejercito, P. M., Nebrija, K. G. E., Feria, R. P., and Lara-Figueroa, L. L. "Traffic simulation software review. In *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1-4. DOI: 10.1109/IISA.2017.8316415, 2017.
30. Instituto Nacional de Estadística. "Población residente en España por fecha, sexo y generación (edad a 31 de diciembre)." 2021. Type: Dataset.

31. Fernández, Carmen Socorro Lema, Andrade, Luis Pedro Pedreira, Allende, Gemayqzel Bouza, and Alonso, Sira Allende. "Estudio de la optimización del tráfico en un cruce a través del ajuste de los ciclos de los semáforos mediante recocido simulado." In *Anales de ASEPUMA*, number 19, page 23. ISSN: 2171-892X, 2011.
32. Ferrer-Urbano, Francisco Javier, Garcia-Nieto, Jose Manuel, Alba-Torres, Enrique, and Chicano, Francisco. "Validación Inteligente para la Sincronización de Semáforos Basada en Feature Models." 2013. Type: Journal Article.
33. Fonseca Guilarte, Orlando, Pujol Fariña, Gabriel Santiago, Allende Alonso, Sira, and Bouza Allende, Gemayqzel. "Una herramienta para simular y analizar el flujo de tráfico en un cruce regulado por semáforos con tiempos de ciclos variables." In *Revista Cubana de Ciencias Informáticas*, volume 8, pages 29-40. ISSN: 2227-1899, 2014.
34. Garber, Nicholas J. and Hoel, Lester A. *Traffic and highway engineering*. Cengage Learning, ISBN: 0357111788, 2019.
35. González Cabrera, Jorge. "Herramienta de modelado de carreteras para simuladores de tráfico." 2020. Type: Journal Article.
36. Guerrero Paredes, Sergio. "Control inteligente de semáforos mediante aprendizaje automático." 2016. Type: Journal Article.
37. Haddouch, S., Hachimi, H., and Hmina, N. "Modeling the flow of road traffic with the SUMO simulator." In *2018 4th International Conference on Optimization and Applications (ICOA)*, pages 1-5. DOI: 10.1109/ICOA.2018.8370580, 2018.
38. Hirankitti, V. and Krohkaew, J. "An Agent Approach for Intelligent Traffic-Light Control." In *First Asia International Conference on Modelling & Simulation (AMS'07)*, pages 496-501. DOI: 10.1109/AMS.2007.11, 2007.
39. Hoogendoorn, Serge P. and Bovy, Piet H. L. "State-of-the-art of vehicular traffic flow modelling." In *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, volume 215, number 4, pages 283-303. ISSN: 0959-6518, 2001.
40. Jacobo, Armando. "Sistema de Semáforos Inteligentes Utilizado Sensores de Presencia." In *Revista Aplicaciones de la Ingeniería*, pages 147-152, 2015.
41. Jiménez, D., Muñoz, F., Arias, S., and Hincapie, J. "Software for Calibration of Transmodeler traffic microsimulation models." In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1317-1323. ISBN: 2153-0017, DOI: 10.1109/ITSC.2016.7795727, 2016.
42. Karakuzu, Cihan and Demirci, Osman. "Fuzzy logic based smart traffic light simulator design and hardware implementation." In *Applied Soft Computing*, volume 10, number 1, pages 66-73. ISSN: 1568-4946, DOI: <https://doi.org/10.1016/j.asoc.2009.06.002>, 2010.
43. Krajzewicz, Daniel, Hartinger, Markus, Hertkorn, Georg, Mieth, Peter, Ringel, Julia, Rössel, Christian, and Wagner, Peter. "The "Simulation of Urban Mobility" package: An open source traffic simulation." ISBN: 907738104X, 2012.
44. Liu, C., Yan, C., Ma, C., and Suo, Q. "Traffic Light Timing Optimization based on Improved Particle Swarm Optimization." In *2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 635-639. ISBN: 2689-6621, DOI: 10.1109/IAEAC54830.2022.9929570, 2022.
45. Ma, T., & Abdulhai, B. (2001). Genetic algorithm-based combinatorial parametric optimization for the calibration of microscopic traffic simulation models. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)* (pp. 848-853). DOI: 10.1109/ITSC.2001.948771. *Conference Proceedings*.

46. Matsuo, T., Kaneko, Y., & Matano, M. (1999). Introduction of intelligent vehicle detection sensors. In *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)* (pp. 709-713). DOI: 10.1109/ITSC.1999.821148. *Conference Proceedings*.
47. Mena, R., Zumárraga, F., Urquiza, L., & Calderón, X. (2019). Google Maps Route Color Mapping with SUMO Simulator. In *2019 International Conference on Information Systems and Software Technologies (ICI2ST)* (pp. 92-99). DOI: 10.1109/ICI2ST.2019.00020. *Conference Proceedings*.
48. Minguillán Van Kapel, L. J. (2022). Red de semáforos inteligentes para la reducción de partículas contaminantes y mejoras de tiempos de tránsito. *Journal Article*.
49. Mubasher, M. M., & ul Qounain, J. S. W. (2016). Systematic literature review of vehicular traffic flow simulators. In *IEEE* (pp. 1-6). ISBN: 1467374652. *Conference Proceedings*.
50. Munguía Torres, I. A. (2016). Sistema de optimización de tráfico vehicular aplicado a la glorieta de Santa Fe. *Universidad de Guanajuato. Thesis*.
51. Panovski, D., & Zaharia, T. (2016). Simulation-Based Vehicular Traffic Lights Optimization. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)* (pp. 258-265). DOI: 10.1109/SITIS.2016.49. *Conference Proceedings*.
52. Pérez, J. I. H., & Rückauer, C. C. (2002). Una revisión de los algoritmos evolutivos y sus aplicaciones. *Madrid, España:[sn]. Generic*.
53. Quintero González, J. R., & Prieto Vaca, L. F. (2015). Sistemas inteligentes de transporte y nuevas tecnologías en el control y administración del transporte. *Puente. Revista Científica. Journal Article*.
54. Quintero-González, J.-R. (2017). Del concepto de ingeniería de tránsito al de movilidad urbana sostenible. *Ambiente y desarrollo, 21(40)*, 57-72. *Journal Article*.
55. Ratrou, N. T., Rahman, S. M., & Reza, I. (2015). Calibration of paramics model: Application of artificial intelligence-based approach. *Arabian Journal for Science and Engineering, 40*, 3459-3468. *Journal Article*.
56. Romero Placeres, M., Diego Olite, F., & Álvarez Toste, M. (2006). La contaminación del aire: su repercusión como problema de salud. *Revista cubana de higiene y epidemiología, 44(2)*, 0-0. *Journal Article*.
57. Ruiz, M. E., Mayorga, C. M., Aldas, D. S., & Reyes, J. P. (2019). El costo y la percepción en la sociedad por congestión vehicular causada por el transporte público urbano en la ciudad de Ambato, Ecuador. *Espacios, 40(43)*, 42. *Journal Article*.
58. Samaniego-Calle, V., Viñán-Ludeña, M. S., Jaramillo-Sangurima, W., Jácome-Galarza, L., & Sinche-Freire, J. (2019). Semáforos inteligentes y tráfico vehicular: un caso de estudio comparativo para reducir atascos y emisiones contaminantes. *Revista Ibérica de Sistemas e Tecnologías de Informação, E19*, 403-414. ISSN: 1646-9895. *Journal Article*.
59. Segura-Contreras, J. F., & Franco, J. F. (2016). Exposición de peatones a la contaminación del aire en vías con alto tráfico vehicular. *REVISTA DE SALUD PÚBLICA, 18(2)*, 179-187. *Journal Article*.
60. Shi, Y., Liu, H., Gao, L., & Zhang, G. (2011). Cellular particle swarm optimization. *Information Sciences, 181(20)*, 4460-4493. DOI: <https://doi.org/10.1016/j.ins.2010.05.025>. *Journal Article*.
61. Tan, J., Chen, X., & Du, M. (2010). A Novel P2P Identification Algorithm Based on Genetic Algorithm and Particle Swarm Optimization. In *2010 3rd International Symposium on Parallel Architectures, Algorithms and Programming* (pp. 22-29). ISBN: 2168-3042. DOI: 10.1109/PAAP.2010.69. *Conference Proceedings*.

62. Terol Lloret, M. (2021). Solución para la gestión de tráfico de Valencia en base a criterios ambientales. *Journal Article*.
63. Torres, I. A. M. (2016). SISTEMA DE OPTIMIZACIÓN DE TRÁFICO VEHICULAR APLICADO A LA GLORIETA SANTA FE. *Journal Article*.
64. Tráfico, D. G. de. (2022). Parque nacional de vehículos por comunidad autónoma, provincia, tipo de vehículo y tipo de carburante. *Dataset*.
65. Unidas, N. (2015). Objetivos de Desarrollo Sostenible. Retrieved from <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>. *Web Page*.
66. Uribe Chavert, P. (2020). Sistema de control de tráfico automático basado en dispositivos modulares heterogéneos. *Journal Article*.
67. Valladares Mejía, A. D. J. (2016). Comparación de un enfoque macroscópico y otro microscópico al estimar las demoras por la congestión urbana. *Escuela de Ingeniería Civil. Journal Article*.
68. Vogel, A., Oremović, I., Šimić, R., & Ivanjko, E. (2018). Improving Traffic Light Control by Means of Fuzzy Logic. In *2018 International Symposium ELMAR* (pp. 51-56). DOI: 10.23919/ELMAR.2018.8534692. *Conference Proceedings*.
69. Xiao, H., Ambadipudi, R., Hourdakis, J., & Michalopoulos, P. (2005). Methodology for selecting microscopic simulators: Comparative evaluation of AIMSUN and VISSIM. *Journal Article*.
70. Zambrano Martínez, J. (2016). Sistema centralizado para la optimización del tráfico mediante un servidor de rutas. *Journal Article*.
71. Zhancheng, S. (2021). Research on Application of Deep Reinforcement Learning in Traffic Signal Control. In *2021 6th International Conference on Frontiers of Signal Processing (ICFSP)* (pp. 17-21). DOI: 10.1109/ICFSP53514.2021.9646417. *Conference Proceedings*.
72. Zhang, Y., & Su, R. (2021). An optimization model and traffic light control scheme for heterogeneous traffic systems. *Transportation Research Part C: Emerging Technologies*, 124, 102911. DOI: <https://doi.org/10.1016/j.trc.2020.102911>. *Journal Article*.
73. Zhao, Y., Shi, J., Xia, S., Li, S., & Zhang, K. (2020). Study on the Evaluation of Microcirculation Traffic Organization Optimization in old Urban Area Based on Microscopic Simulation. In *2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)* (pp. 93-98). DOI: 10.1109/AEMCSE50948.2020.00027. *Conference Proceedings*.
74. Padrón, J. D., Terol, M., Zambrano-Martinez, J. L., Calafate, C. T., Cano, J.-C., & Manzoni, P. (2022). Evaluación del impacto de las restricciones de tráfico en los niveles de contaminación. *Journal Article*.
75. Chávez Fuentes, C. P. (2015). Sistema de semaforización inteligente para el control de flujo vehicular mediante el procesamiento digital de imágenes. *B.S. thesis*. Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas.
76. Martínez, D. P., Batista, V. F. L., & de Paz, J. F. (Avances en Informática y Automática, 81). Sistemas de detección y seguimiento de objetos a través de visión artificial. *Journal Article*.
77. Robles, G. R., Mastache, J. E. M., Ramírez, R. L., & Jiménez, J. N. (2022). MPP optimization in photovoltaic systems under partial shading conditions applying PSO, IPSO and AIWPSO. *Revista Aristas*, 9(17), 226-234. *Journal Article*.

Evaluating the Effectiveness of Human-Centered AI Systems in Education

Nastaran Shoeibi, Roberto Therón Sánchez y Francisco José Garcia-Peñalvo

Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los
Caídos s/n. 37008, Salamanca, España
{Nastaran, theron, fgarcia}@usal.es

Resumen This thesis examines how AI can improve human-computer interaction (HCI) and user experience in education. A systematic literature review (SLR) and LATILL case study show how AI can be used in education. The SLR examines existing literature to determine how AI affects user experience and HCI in education, highlighting personalization and adaptability of learning experiences, improved task performance, and improved user experience for teachers and students. AI implementation in education faces obstacles. Using CEFR levels and linguistic traits, the LATILL project uses a user-centered design to give students personalized guidance and support. It transforms language instruction and fosters engaging and successful learning by encouraging educator collaboration and resource sharing. This study emphasizes the importance of user experience and HCI principles in designing AI-driven educational systems. AI and user-centered design can improve learning, student engagement, and educational outcomes.

Keywords: Artificial Intelligence, Generative AI, Human-computer Interaction, User-experience, Education, Personalized Learning, SLR, Systematic literature

1. Introduction

The term “Artificial Intelligence” was coined during the Dartmouth Conference in 1956, where scientists like Warren McCulloch, Walter Pitts, and Alan Turing laid the foundation for AI. Neural networks and computational brain models formed the basis for AI. Since then, AI has been used in various fields [11][10][9][46]. The question of whether AI will surpass human intelligence and become superintelligent is ongoing. AI-based systems can use more powerful algorithms and capabilities to overcome the biological limitations of the human brain. This could lead to the creation of better AI systems, accelerating technological advancement [5]. Computer-based AI with human-like minds has many advantages compared to biological brains. Moore’s Law ensures that hardware becomes exponentially faster and more capable, allowing AI to reason and analyze information faster than humans. Additionally, AI can access vast amounts of data stored on expanding hard drives, enabling thorough research and consideration of relevant information without slowdowns [14][13]. AI’s software-based

nature makes it easy to duplicate and network. Once trained, AI can replicate itself, eliminating the need for repetitive training. Networking a group of AIs with diverse skills and compatible motivations can create a super-committee that outperforms any human committee, excelling in information sharing and problem-solving [5]. AI has various definitions and is used to solve cognitive problems, develop systems with human-like abilities, and mimic human cognition and behavior. It enables machines to approximate human reasoning, adapt to the environment, and perform tasks requiring human intelligence [24].

AI has been introduced into education, where human and AI teachers collaborate to teach subjects like English. AI tools have the potential to teach any subject. They can transform education by providing convenience, interactive and personalized learning experiences, global classrooms through virtual reality, and AI-based chatbots for personalized online learning and assessment [47][48][45].

Using computer technology in the teaching and learning of second or foreign languages is known as intelligent computer-assisted language learning (ICALL)¹. It is a branch of computer-assisted language learning (CALL) that creates intelligent tutoring programs to give language learners individualized feedback [17]. Natural language processing (NLP), spell and grammar checkers, feedback, student models, and future directions are all possible components of ICALL systems [68].

Generative AI has improved over the years with advances in generative adversarial networks (GANs) and deep neural networks. These advancements allow computer-generated media to resemble human-produced media, opening up opportunities in various industries, including education [8][6][7]. Integrating AI into educational settings requires a focus on user interface and experience. A Systematic Literature Review (SLR) examines how AI is used in education and affects user experience, seeking to improve the user experience when integrating AI technologies into education [12].

The study focuses on using AI in educational settings to improve user experience. It includes an overview of key concepts in hiddensection 10, a systematic literature review, an analysis of research questions, implications, and importance in hiddensection 10, a case study of the LATILL project's technologies in hiddensection 10, and conclusion and suggestions for future research in hiddensection 10.

2. Theoretical Background

J.C.R. Licklider's visionary concept of a symbiotic relationship between humans and computers, as outlined in his 1960 paper "Man-Computer Symbiosis," [1] has resurfaced in the fields of Augmented Intelligence and Human-Computer Interaction (HCI). Influential figures like Alan Kay[4] and Douglas Engelbart [3] have embraced Licklider's ideas, shaping the field of HCI and influencing user-centered technology design. The cooperative relationship between humans and

¹ https://en.wikipedia.org/wiki/Intelligent_computer-assisted_language_learning

computers, combining their strengths to enhance problem-solving capabilities, lies at the core of Licklider's vision. Augmented Intelligence design, focusing on human-centered approaches and ethical considerations, reinforces this symbiotic collaboration, empowering individuals and maximizing human potential in human-computer interactions [2].

HCI, a multidisciplinary field, is dedicated to designing, evaluating, and implementing interactive technologies to enhance how people interact with computers and other technology. It plays a significant role in education, contributing to designing and developing interactive computing systems that enhance the learning experience. HCI researchers collaborate with educators and instructional designers to create user interfaces for educational software and digital learning platforms that are intuitive, visually appealing, and easy to navigate. Usability studies and user-centered design principles are applied to ensure the interfaces meet learners' specific needs and preferences. Furthermore, HCI principles are crucial in software development, as they involve continuously analyzing user feedback and data to improve educational systems, resulting in more personalized and effective learning experiences. The application of HCI in education extends to leveraging emerging technologies such as augmented reality (AR) and virtual reality (VR) to create immersive and interactive learning experiences [25][23][38][26][33][32][40][22].

In the context of AI technologies, various advancements have contributed to enhancing the user experience in educational settings. Affective computing enables machines to understand and react to human emotions, allowing personalized learning experiences and emotional support[34]. AR and VR technologies offer immersive and interactive learning experiences, allowing students to explore complex concepts and receive hands-on training [35]. Computer vision enables computers to comprehend and interpret visual data, supporting tasks like image and video analysis [37]. Conversational AI facilitates natural language interactions between learners and AI systems, providing personalized guidance and tutoring [44]. Deep learning extracts complex patterns from data, allowing for personalized learning experiences and advancements in student engagement [43]. Intelligent assessment automates educational evaluation, providing individualized feedback and supporting data-driven decision-making [39]. Machine learning analyzes large datasets and automates tasks, enhancing personalized learning and adaptive learning platforms [41]. Speech recognition technology transforms spoken words into text, facilitating voice-activated interfaces and accessibility in education [39].

Intelligent Computer-Assisted Language Learning (ICALL) combines language learning with computer science, utilizing AI technologies like NLP and ITS to create interactive language learning experiences [66]. These systems offer personalized instruction, immediate feedback, and tailored exercises, enhancing language acquisition [67]. By incorporating technology, ICALL provides authentic language input, cultural exposure, and opportunities for communicative language use. NLP algorithms enable error detection and grammar analysis, while ITS employs learner modeling and adaptive learning to provide individualized instruction [64]. Platforms like Duolingo, Rosetta Stone, Babbel, Lingoda, and

Busuu leverage gamification and interactive lessons to engage learners and offer immersive language learning experiences.

Natural language processing enables computers to understand and produce human language, supporting tasks like language learning applications and text analysis [36]. In NLP, text summarization is a technique used to automatically generate concise and coherent summaries of longer text documents. There are two main approaches to text summarization: extractive and abstractive. Extractive summarization involves selecting important sentences or phrases from the original text, while abstractive summarization involves generating a summary containing words, phrases, and sentences not present in the original text. Various NLP frameworks and libraries are available for text summarization, including NLTK, Gensim, spaCy, and TextBlob. These libraries provide models and tools for tasks such as tokenization, lemmatization/stemming, stop word removal, sentence simplification, word substitution, sentence fusion/splitting, grammatical simplification, and rephrasing/paraphrasing. As explained in [18], Spacy is known for efficient tokenization and part-of-speech tagging, while NLTK is a comprehensive NLP toolkit suitable for beginners. Gensim focuses on topic modeling and document similarity analysis, and TextBlob offers a simplified API for common NLP tasks. The table highlights each library's key features, ease of use, performance, and multilingual capabilities.

Machine translation (MT) is another area within NLP that involves using algorithms to translate large amounts of text. While MT has improved with machine learning and neural networks, it still has limitations in handling complex texts, idiomatic expressions, cultural nuances, and domain-specific terminology. There are many translators, such as Translators, Deep-translator, Googletrans, and DeepL. "Translators" is well-supported and versatile but lacks customization and language detection. "Deep translator" supports multiple APIs, but language detection requires a private key. "Googletrans" is simple but unmaintained. "DeepL" has limited Python support but high-quality translations.

Generative AI, also known as GAI, is a technology that generates content within responses, surpassing human-like interactions of Conversational AI. Unlike Conversational AI, which relies on programmed responses, it independently produces fresh, original content. In generative AI, large Language Models (LLMs) and diffusion models are used to generate new text, images, and media.

LLMs, such as GPT-4, BLOOM, StableLM, LLaMA, Dolly, and Alpaca, have transformed natural language processing. They have educational applications where they can enhance learning experiences and create personalized content. LLMs have generated interactive educational materials, supported foreign language learning, improved questioning skills, and created chatbot tutors. The versatility of LLMs allows them to perform tasks across multiple languages and domains, thanks to zero-shot and few-shot learning techniques.

Diffusion models, such as Stable Diffusion, Midjourney, and DALL-E, offer an alternative approach to image generation. They gradually transform noisy images into target images by iteratively reducing the noise. Diffusion models have advantages over traditional GANs regarding stability during training and control over the image generation process. Each diffusion model has unique cha-

racteristics and training techniques, such as using VQ-VAE-2 in DALL-E to condition the model on text prompts for generating specific visual outputs.

These generative AI technologies open up new possibilities in various fields, including education, by providing enhanced content generation and personalization capabilities.

3. Systematic Literature Review (SLR)

3.1 Methodology

The study conducted a Systematic Literature Review (SLR) using the PRISMA model [31] and the SLR methodology by Kitchenham [29][27]. The SLR focused on evaluating and interpreting primary research in the field of AI technologies in educational domains, with a specific emphasis on enhancing user experience and human-computer interaction. The review followed a well-established methodology, including the mapping study guidelines by Petersen [28]. Relevant papers were selected based on specific criteria to provide a comprehensive overview of the different AI technologies used in education. The SLR protocol is described in this hiddensection, covering the review’s planning, conducting, and reporting phases.

3.2 Review Planning

The review planning phase of an SLR involves formulating research questions, creating a review protocol, and documenting important details. This phase sets the foundation for a structured and rigorous review by guiding the objectives, scope, methodology, and data extraction techniques. Documenting key information enhances transparency and replicability.

3.2.1 Research Questions Research questions were formulated to gather information on AI’s application, assessment, effects, and future potential in education, particularly in enhancing the user experience. A literature mapping was conducted to analyze the research domain. The mapping questions (MQs) and research questions (RQs) related to the search were:

- MQ1** How is the publication distribution of AI in education research studies over the few years?
- MQ2** What type of papers have been published?
- MQ3** In which domains and education levels are the distribution of research papers on AI in education being used?
- RQ1** How is AI being used to improve the quality of human-computer interaction in education?
- RQ2** What are the impacts on students and teachers regarding user experience when implementing AI in education?
- RQ3** What are the future outlooks for using AI tools to enhance the user experience of students and teachers in educational technology?

3.2.2 Inclusion and exclusion criteria Inclusion criteria (IC) and exclusion criteria (EC) are important for ensuring the selection of relevant and high-quality studies in the systematic literature review (SLR). For this research, the ICs specify that only English articles published between 2019 and 2022 are included. The studies must be full-text available, peer-reviewed, and indexed in Scopus or Web of Science. They should focus on integrating AI in education, user experience, and human-computer interaction. The ECs exclude studies unrelated to AI in education or lack explanations of user experience and human-computer interaction in education. These criteria ensure a focused and comprehensive review of the field of AI in education. The following details the criteria used:

- **IC1** Studies published in English
- **IC2** Studies published between 2019 and 2022
- **IC3** Peer-reviewed journal articles, book chapters, and conference proceedings
- **IC4** The full text is available
- **IC5** The study must use AI-based technologies
- **IC6** The study must include in Education and Learning
- **IC7** The study must include in User-experience and Human-computer interaction in education
- **EC1** Studies published in languages other than English
- **EC2** Studies published before 2019 or after 2022
- **EC3** Not primary research (e.g., review)
- **EC4** The full text is not available
- **EC5** Not mentioning user-interface but about Implementing AI-embedded technologies
- **EC6** Not related to education and learning
- **EC7** Not explaining the User-experience and Human-computer interaction in education

3.2.3 Data Sources Scopus and Web of Science (WoS) were selected as key databases for the research due to their wide research scope, reliability, flexible search capabilities, and relevance to the study's context. These databases provide valuable and comprehensive academic publications related to the research topic.

3.2.4 Query Strings As outlined below, the search strings for each selected source were constructed based on the references provided in Tables 2 and 3. Boolean operators (AND/OR) connected the terms, and the wildcard (*) accommodated singular and plural forms. The search was specifically targeted toward human-computer interactions in integrating AI in education, ensuring that the retrieved works were highly relevant and offered insights into the use of AI to enhance interactions between humans and computers in educational settings.

- **Scopus:** TITLE-ABS-KEY ((education* OR .AIED") AND (.artificial intelligence. OR .AI") AND (user experience. OR user-experience. OR "Human-computer interaction. OR User-centered design") AND (learn* OR student* OR teach*)) AND PUBYEAR >2018 AND PUBYEAR <2023
- **Web of Science:** ts=((education* OR .AIED") AND (.artificial intelligence. OR .AI") AND (user experience. OR user-experience. OR "Human-computer interaction. OR User-centered design") AND (learn* OR student* OR teach*))

3.2.5 Quality Assessment While inclusion and exclusion criteria ensure the relevance of articles in a review, they do not assess the quality of papers about the research questions. To evaluate the quality of papers in the final literature review, four criteria are employed. Three criteria are graded on a scale of 0 to 3, and one criterion is graded on a scale of 0 to 1. Only papers that score in the top quartile, representing 25% of the possible total score, are included in the review. These criteria assess the accuracy and relevance of papers in addressing the research questions, ensuring a comprehensive literature review.

3.3 Review Process

The PRISMA flow diagram (Figure 1) depicts the systematic literature review process and the data extraction actions. It indicates that no additional records were included in the review.

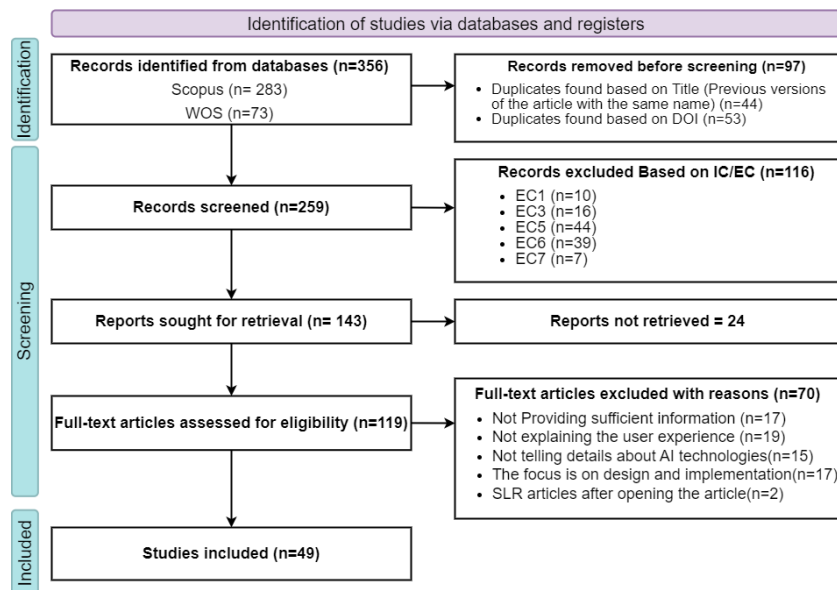


Figura 1: PRISMA 2020 Flow diagram [31].

During the review process, the following steps were taken:

- **Phase 1: Identification** - A total of 356 records were retrieved from the selected databases (283 from Scopus and 73 from WOS).
- **Phase 2: Screening** - Duplicate records (97 in total) were removed, and 259 records were screened. Among them, 116 records were excluded and did not meet the inclusion/exclusion criteria. The full-text articles of the remaining 143 records were examined for eligibility, but 24 articles were inaccessible. This left 119 articles for further review. Out of these, 70 articles were removed during the quality assessment.

- **Phase 3: Included** - After the review process, 49 articles met the criteria for inclusion and quality and were selected for further analysis.

3.4 Data Extraction

The data extraction form, available at the provided Google Drive link (<https://bit.ly/dataset-mapping>), will be utilized to analyze the 49 selected primary studies. These studies will be examined in the order of their titles. The data extraction form includes information such as article ID indicator, authors, titles, quality scores, and more for each study.

3.5 Results of the SLR

The collected data include descriptive information, approach-related information, quality-related information, the use of AI in Education, and its impact on human-computer interaction. The following analysis provides detailed insights from the 49 selected primary studies, addressing the mapping questions and presenting the results that answer the research questions.

3.5.1 Results of Mapping Questions

MQ1: Publication distribution of AI in education research studies over the years

As shown in Figure 2, which presents the interest in integrating AI technologies in education from 2019 to 2022, particularly in 2022, a noticeable trend appears, and 16 studies have been published. Out of the 49 studies, 20.4% (ten studies) were published in 2019, and also 2020 was the same; 26.5% (thirteen studies) in 2021, and 32.7% (sixteen studies) in 2022 were published. These findings show that the importance of AI in education is a challenging and trending topic, and researchers are exploring and contributing to this area, and it is expected that further research and advancements in integrating AI technologies in education.

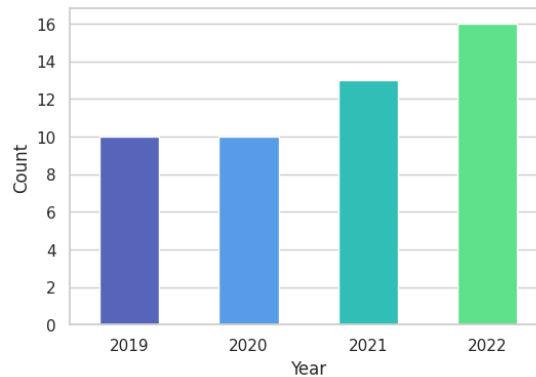


Figure 2: Distribution of publications per year

MQ2: Type of published papers

The complete list of paper types extracted from the records is shown in Table 1. Only papers that have gone through peer review in journals, conferences, or book chapters are considered after the inclusion and exclusion criteria have been considered.

Tabla 1: Papers grouped by the publication type

Type	Total	Papers
Article	12	P33, P25, P44, P16 ,P14, P18, P34, P23, P22, P35, P41, P45
Conference paper	37	P20, P27, P30, P32, P10, P36, P02, P04, P06, P09, P13, P24, P28, P29, P31, P37, P38, P42, P08, P40, P03,P21, P01, P11, P17, P43, P47, P15, P19, P46, P48, P49, P07, P05, P12, P26, P39

MQ3: Educational domains and levels

Table 2 details the effects of artificial intelligence in education across different domains and levels of education.

Tabla 2: Different Educational Domains and Levels that AI was used

Domain	Level	Articles
Academic Management	Higher Education	(P31)
	Secondary Education	(P35)
	Various Levels	(P37,P43)
Art Education	Higher Education	(P16, P19,P26)
E-Learning	Higher Education	(P22, P34, P36)
	Maritime Education	(P40)
	Various Levels	(P10)
General Education	K-12	(P2, P46)
Language Learning	Higher Education	(P6, P8, P13, P23, P32)
	Various Levels	(P9, P33, P48)
Legal Education	Higher Education	(P44)
Medical Education	Higher Education	(P24, P38, P42)
Personalized Learning Environment	Higher Education	(P5, P17)
	Preschool Education	(P29)
	Various Levels	(P4)
Physical Education	Primary Education	(P18, P27)
	Various Levels	(P21, P25)
Pilot Training	Aviation Industry	(P11)
STEM	Higher Education	(P3, P7, P12, P28, P30, P45, P47)
	Preschool Education	(P41)
	Various Levels	(P01)
Various Domains	Higher Education	(P39)
	Secondary Education	(P20)
	Various Levels	(P15, P49)

The observations from Table 2 are as follows: AI is utilized across various educational areas and levels, highlighting its versatility and broad application. It is present in domains such as Academic Management, Art Education, E-Learning, Language Learning, Medical Education, and more, including Higher Education. AI's impact extends to K-12 education, focusing on STEM, academic management, various domains, General Education, and Physical Education.

Furthermore, AI is used in diverse fields like pilot training, legal education, maritime education, and personalized learning environments. STEM education, particularly in higher education, demonstrates significant AI integration. AI also heavily influences language learning, with numerous mentions indicating its impact across different educational levels.

3.5.2 Results of Research Questions

RQ1: AI tools used to improve HCI in education

Figure 3 provides an overview of AI technologies used in education research over the past four years, highlighting the importance of research and the steady development of AI tools and technologies in the field. Deep Learning and Machine Learning techniques have shown significant growth and impact, with an increasing number of papers focusing on their application and positive effects on education. NLP continues to be a field of interest, with a consistent increase in research papers each year. Other AI techniques such as Computer Vision, Speech Recognition, AR&VR, and Intelligent Assessment have also attracted researchers' attention and shown promising contributions to education. Conversational AI, particularly in the form of chatbots, has gained increasing interest and is expected to have broad applications in content creation and user engagement. Additionally, the emerging field of affective computing has garnered attention and seen a rise in publications, indicating its growing importance and potential for further research. These AI technologies, including affective computing, conversational AI, deep learning, machine learning, NLP, computer vision, speech recognition, AR&VR, and intelligent assessment, will continue to drive developments in educational AI in the coming years.

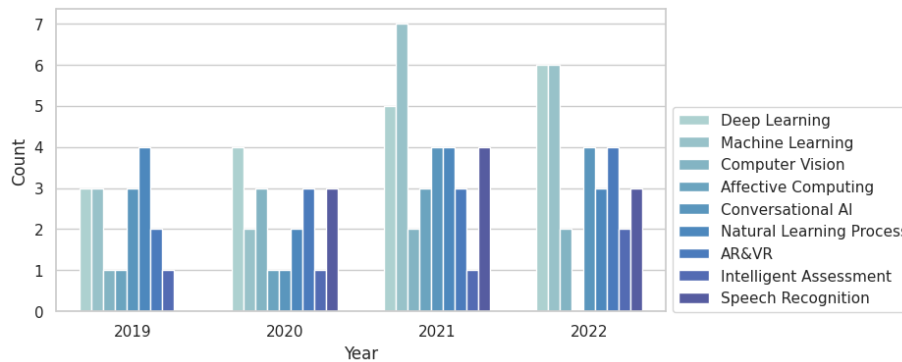


Figura 3: Number of different AI Technologies in each study per year

Table 3 details the trends and research on various AI applications in education. It highlights areas that have drawn more attention and those that may need additional research. Research papers in these fields highlight the significance of incorporating AI tools in education to improve evaluation processes, personalized learning experiences and support effective and efficient communication between users and technology. it reveals persistent interest and research focus

on AI technologies in education, consistent presence, and ongoing developments in AR&VR, computer vision, and NLP. Deep Learning and Conversational AI have seen an increase in research papers over time, indicating growing exploration of their potential in education. However, areas like Intelligent Assessment show a decline in research publications after 2020, and Affective Computing requires further recent research. These observations highlight the dynamic nature of research in AI technologies in education.

Tabla 3: Paper (IDs) published in each year across different AI Technologies

AI technologies	2022	2021	2020	2019
Affective Computing		(P06, P17, P37)	(P02)	(P04)
AR&VR	(P23, P40, P41, P45)	(P06, P09, P29)	(P21, P28, P42)	(P03, P46)
Computer Vision	(P14, P38)	(P27, P37)	(P12, P21, P28)	(P46)
Conversational AI	(P05, P16, P22, P36)	(P09, P26, P43, P47)	(P02)	(P08, P32, P39)
Deep Learning	(P18, P33, P35, P38, P49, P45)	(P10, P26, P27, P37, P44)	(P01, P12, P30, P31)	(P08, P15, P48)
Intelligent Assessment	(P05, P13)	(P06)	(P20)	(P08)
Machine Learning	(P14, P16, P22, P34, P35, P36)	(P06, P10, P11, P17, P44, P47, P49)	(P20, P30)	(P19, P24, P46)
NLP	(F05, F36, F38)	(P10, P17, P47, P49)	(P12, P20)	(P08, P24, P32, P39)
Speech Recognition	(P13, P25, P38)	(P01, P07, P12)	(P01, P07, P12)	

RQ2: The impacts of AI in education on students and teachers

The use of AI in education has significantly impacted the user experience for teachers and students. Table 4 highlights important findings about the impact of AI technologies in education from the user-experience part. AI tools have provided teachers with valuable administrative and management support, as discussed in 20.4% (10 of the papers). They facilitate the creation of personalized educational programs and help to streamline administrative tasks and resource allocation. Teachers can concentrate more on providing high-quality instruction and interacting with others by reducing administrative work.

The number of research papers focusing on how AI can improve teachers' understanding and support of students, as seen in 34.7% of the papers (17 of the papers), highlights the importance of further research in this area to explore how AI tools can help teachers gain insights into students' needs, emotions, and progress. Researchers can learn more about this subject and create innovative approaches by getting more deeply, which will help improve educational practices through effective AI integration. Furthermore, as mentioned in 14.3% of the articles (7 of the papers), AI tools offer suitable tools for evaluation and feedback, enabling teachers to give students helpful feedback and support their growth. It was mentioned in 22.45% of the articles (11 of the papers) that AI tools could improve teachers' ability to engage and inspire students and offer technical support and design teaching methods. The significant interest in studying how AI can improve the user experience is shown by 36.7% of the papers (18 of the papers) focusing on the impact of AI in education and by 30.6%

of the papers (16 of the papers) devoted to personalized learning and knowledge acquisition. It reveals a developing trend in research by highlighting the significance of looking into and creating AI-based solutions that are personalized for students' individual requirements, ultimately improving their learning experiences and outcomes [P1, P10, P16, P26]. Additionally, 32.6% (16 of the papers) of the publications support immersive, interactive learning environments using computer vision, virtual reality, and augmented reality. With the help of these technologies, education becomes more enjoyable, interactive, and engaging for students, encouraging greater participation and memory retention. By adapting resources and content to user needs and preferences, AI also supports personalized learning experiences. Enabling self-regulated learning [P20, P43], which gives students more control over their learning process and pace, has an additional effect on students. Additionally, AI tools provide real-time feedback and guidance, supporting students instantly and fostering their academic success [P21, P25, P27, P30, P33, P34, P39]. AI systems considering students' needs and emotional factors can enhance psychological well-being. AI helps students feel good and motivated by fostering a welcoming and inclusive learning environment [P8, P15, P40, P43].

Tabla 4: Impacts of using AI in education on teachers and students

Persona	Impacts	Articles
Teachers	Administrative and Management	
	• <i>Comprehensive management tool</i>	(P16, P43)
	• <i>Optimization of resource allocation and individualized learning programs</i>	(P4, P30, P36)
	• <i>Reduced workload for instructors</i>	(P39, P49)
	• <i>Teaching Quality Improvement</i>	(P31, P34, P48)
	Effective evaluation and feedback for learning outcomes	(P6, P8, P13, P15, P29, P34, P39)
	Interactive and Immersive Learning Environment	(P9, P10, P11)
	Teacher Engagement and Motivation	
	• <i>Enhancing teachers' ability to engage and motivate students</i>	(P9, P15, P20, P7, P12, P22, P28)
	• <i>Providing technical support and strategy design for teaching</i>	(P13, P30, P31, P49)
Students	Understanding and Support for Students	
	• <i>Enhanced understanding of students' emotions and instructional practices</i>	(P4, P6, P7, P26, P31, P37, P39, P43, P49)
	• <i>Monitoring student progress and identifying areas for additional help</i>	(P6, P10, P14, P19, P29, P35, P37, P39)
	Continuous improvement of education	(P1, P10, P16, P26)
	Enhanced User Experience	(P1, P3, P5, P6, P7, P9, P11, P13, P15, P16, P22, P23, P24, P28, P31, P35, P37, P43, P48)
	Facilitation of self-regulated learning	(P17, P20, P43)
	Gender and disability equality	(P3, P46)
	Immersive and Interactive Learning	(P3, P6, P7, P11, VP14, P18, P21, P26, P28, VP39, P40, P41, P42, P45)
	Personalized Learning and Knowledge Acquisition	(P1, P5, P8, P10, P12, P19, P20, P21, P25, P27, P29, P34, P36, P40, P49)
	Psychological impact	(P8, P15, P40, P43)
Real-time Feedback and Guidance	(P21, P25, P27, P30, P33, P34, P39)	

RQ3: The outcomes of using AI tools in education

The use of AI tools in education has the potential to enhance the user experience for both students and teachers greatly. Assessments with eight papers (16.3% of the articles) highlight how AI systems improve student performance by providing real-time evaluation and feedback, enabling personalized guidance and improvement opportunities. Accessibility and flexibility in education, mentioned in 16 papers (32.6% of the studies), are significantly improved by AI solutions that make educational platforms and resources accessible to more students, fostering creativity and engagement. Digital literacy, mentioned in two papers, equips students with essential digital skills to thrive in a technological society. Improved human-computer interaction, accounting for six papers (12.2% of the studies), enhances the usability and effectiveness of educational platforms and tools, leading to improved learning outcomes and user satisfaction.

According to 49% of the studies, AI enhances learning experiences by increasing students' motivation and engagement, creating immersive and interactive learning environments. Personalized guidance, AI-powered tools, and customized content delivery tailored to each student's needs contribute to improved learning outcomes and a better user experience. Fairness in education is addressed through AI's contribution to gender and disability equality and its ability to overcome geographical restrictions, as highlighted in three papers. By addressing the diverse needs of students and ensuring equal opportunity for all, AI technologies contribute to an inclusive and equitable learning environment. Pedagogical or instructional outcomes, emphasized in 28 studies (57% of the studies), demonstrate how AI adjusts education to individual needs, encourages active learning, and offers personalized, real-world learning experiences. Personalized resource allocation using AI optimizes the allocation of educational resources based on each student's needs, improving the user experience by providing the right resources at the right time and maximizing learning potential.

The use of AI tools in education has the potential to enhance the user experience for both students and teachers greatly. Here are some outcomes associated with using AI tools to enhance the user experience in education described in Table 5. One is assessments with eight papers (16.3% of the articles); teachers can better track and support student collaboration. By providing real-time evaluation and feedback, enabling personalized guidance and improvement opportunities, AI systems improve student performance. Students actively engage in collaborative activities and receive prompt feedback through dynamic and interactive assessments, which improves the user experience. AI also significantly improves accessibility and flexibility in education, with 32.6% of the studies (16 papers). AI solutions make educational platforms and resources accessible to more students. This removes financial, geographic, and physical barriers to ensure all students can access educational materials and participate in learning activities. AI technology boosts student engagement by fostering creativity and imagination. Another area mentioned in [P21, P30] is digital literacy. Students gain essential digital skills from improved AI and digital competency. This improves the user experience by helping students use digital resources for learning and thrive in a technological society. Another aspect of the user experience is improved human-computer interaction, which accounts for 12.2% of the studies

(6 papers). AI technologies enhance the usability, intuitiveness, and effectiveness of educational platforms and tools by enhancing human-computer interaction. As a result, learning outcomes are improved, and user satisfaction is raised.

Tabla 5: Outcomes of AI tools on the user experience in education

Categories	Description	Outcomes	Articles
Assessment	<i>AI positively impacts assessment outcomes by improving student collaboration and providing real-time evaluation and feedback.</i>	• Monitor and support student collaboration more effectively	(P3, P15, P35, P36)
		• Real-time evaluation and feedback	(P13, P18, P21, P34)
Accessibility and Flexibility	<i>AI enhances accessibility and flexibility in education, making it more affordable, accessible, and inclusive for learners.</i>	• Cost-effective and accessible manner	(P3, P16, P20, P24, P29, P34, P39, P43)
		• Improved interaction, creativity, and imagination compared to traditional learning methods	(P9, P27, P36, P37, P42, P45, P46, P48)
Digital Literacy	<i>AI contributes to developing digital literacy skills by providing access to advanced AI characteristics and digital competency.</i>		(P10, P30)
Enhanced Human-Computer Interaction	<i>Enhanced human-computer interaction through AI technologies improves communication and collaboration between students, teachers, and educational platforms.</i>		(P2, P4, P14, P22, P33, P32)
Enhanced Learning Experience	<i>AI enhances the overall learning experience by increasing student engagement, motivation, providing personalized guidance, and improving instruction effectiveness.</i>	• Increased Engagement	(P16, P17, P25, P28, P29, P39, P41, P43, P47, P49)
		• Learning efficiency	(P2, P8, P10, P17, P19, P21, P22, P23, P25, P26, P32, P40, P45, P49)
Fairness	<i>AI promotes fairness by addressing gender and disability equality in education and overcoming geographical limitations.</i>	• Gender and disability equality	(P3, P46)
		• Overcoming Geographical Limitations	(P13)
Pedagogical or instructional	<i>AI supports innovative teaching methods, practical learning experiences, and personalized instruction, transforming pedagogy.</i>	• Personalized Learning Experience	(P2, P3, P5, P10, P17, P19, P21, P22, P36, P40, P44)
		• Practical learning experiences	(P12, P13, P19, P21, P27, P30, P36, P38, P40, P42)
		• Offers innovative teaching methods	(P3, P13, P16, P29, P31, P45, P47)
Personalized resource allocation	<i>AI enables personalized resource allocation, optimizing the distribution of educational resources to meet individual student needs.</i>		(P4, P6, P11, P16, P17, P24, P25, P32, P44)

3.6 Discussion

A total of 49 selected articles were thoroughly analyzed to address the research questions posed in the study. The analysis reveals that integrating AI into education involves different aspects. Still, this study emphasizes the user interface and experience to enhance user interaction and educational tools. Numerous other studies have concentrated on designing, developing, and using AI tools in education in addition to this one [63][62][61][60][55][54][53][49]. The design of embedded systems and environments based on AI, such as smart campuses, has

been one of the many topics covered in these studies [51][56][57][58][59]. Some articles explore using gamification in educational settings, utilizing AI methods to produce interesting and interactive learning opportunities [50][52].

The data analysis highlights the significance and growth of AI in education, particularly in STEM and language learning domains. It is evident that AI is being applied in various areas and is expected to continue expanding. Additionally, AI exhibits notable applications in e-learning and physical education [P1, P5, P9, P12, P19, P21, P24].

The Cognitive Immersive Language Learning Environment (CILLE), described in Article P24, combines Artificial Intelligence (AI) and Extended Reality (XR) to enable naturalistic, multi-modal conversations for comprehensive foreign language acquisition, which is illustrated through a study on teaching Chinese as a foreign language. General education, as mentioned in [P2, P46], utilizes AI to teach students, and many articles [P35, P2, P46, P29, P18, P27, P41, P20] focus on using AI to teach different levels of K-12 education in various domains. While most articles concentrate on integrating AI into higher education, it is essential to consider the potential utilization of AI tools across different educational levels. Moreover, educational purposes extend beyond academic learning. For example, in P2, DL, and Speech Recognition were used to teach plant science, and AI was applied in a botanical garden to facilitate learning about plants. Additionally, platforms like STEP ² (mentioned in P45) enhance traditional maritime learning cycles by making them more engaging and interactive with AR & VR. Furthermore, P12 presents an ML-aided pilot training and education framework. AI tools, such as ML, DL, and ITS, are also utilized in Art Education for higher education students [P17, P20, P31].

The use of AI in education has significantly impacted the user experience for both teachers and students. AI tools provide teachers with valuable administrative and management support [P16, P34], enabling personalized educational programs [P4, P30, P36] and reducing administrative workload [P39, P49]. Additionally, teachers have access to better materials, improving the quality of teaching [P31, P32, P48]. Further research is needed to explore how AI can help teachers gain insights into students' needs, emotions, and progress, with 17 papers (34.7%) focusing on this area. AI tools also facilitate evaluation and feedback, helping teachers provide meaningful support and guidance to students (mentioned in 7 papers [14.3%]). Additionally, AI improves teachers' ability to engage and inspire students and offers technical support and innovative teaching methods (mentioned in 11 papers [22.45%]). There is a growing interest in exploring the impact of AI in education and personalized learning, with 18 papers (36.7%) and 16 papers (30.6%), respectively, focusing on these areas.

Furthermore, many articles [P9, P10, P11, P3, P6, P7, P11, P14, P18, P21, P26, P28, P39, P40, P41, P42, P45] highlight using computer vision, virtual reality, and augmented reality to create immersive and interactive learning environments for both students and teachers. Immersive education enhances learning outcomes by creating a highly engaging and interactive learning environment

² (<http://www.smarteducationplatform.it/>)

that promotes active participation and deep understanding of the subject matter. AI also supports personalized learning experiences, enabling self-regulated learning, providing real-time feedback and guidance, and fostering students' psychological well-being [P8, P15, P40, P43]. The impacts of AI in education are broad and multifaceted. One notable impact is the automation of administrative tasks, which frees up teachers' time, enabling them to dedicate more attention to personalized instruction and support. This automation contributes to the efficiency and organization of the educational system [P17, P20, P26].

AI benefits students by enhancing personalized learning experiences, motivation and engagement, immersive and interactive learning environments, accessibility and flexibility, and learning effectiveness [P3, P8, P14, P23]. By addressing individual needs, encouraging a sense of fairness, and offering support for diverse learners, AI technologies also help to promote inclusivity and equity in education [P6, P13, P27].

Researchers in P3 and P46 suggested AIED assist students with disabilities and mentioned the significance of equality among students from various backgrounds. This strategy aims to develop a just educational system where all students have equal learning opportunities regardless of background.

It is also crucial to recognize how interconnected and mutually reinforcing these effects of AI on education are. Although each AI tool contributes differently to the overall user experience, they are all interdependent and shape the learning environment. For instance, the comprehensive management tool and resource allocation optimization decrease teachers' workload and produce a more effective and organized system, which benefits students' learning experiences. Similarly, interactive and immersive learning environments powered by AI technologies boost students' motivation and engagement while enhancing the effectiveness of instruction and interactions between teachers and students.

Integrating AI tools in education has many potential outcomes for the user experience. One key outcome is the improvement in assessments. With the help of AI, teachers can better track and support student collaboration. AI systems offer real-time evaluation and feedback, providing timely insights into student performance. This enables individualized guidance and opportunities for improvement. By creating a more dynamic and interactive assessment process that actively involves students and provides prompt feedback, AI enhances the user experience [P3, P13, P15, P18, P21, P34, P35, P36]. Another significant outcome is the increased accessibility and flexibility of education. AI solutions make educational platforms and resources more easily accessible to a wider range of learners. By eliminating physical, financial, and geographic barriers, AI ensures that all students can interact with educational materials and participate in learning activities. This improved accessibility and flexibility not only enhances the user experience but also encourages imagination, creativity, and engagement in the classroom [P3, P9, P16, P20, P24, P27, P29, P34, P37, P39, P42, P43, P45, P46, P48]. AI also plays a crucial role in developing digital literacy skills among students. By providing access to advanced AI characteristics and digital competency, AI enhances students' abilities to navigate the digital world. This, in turn,

improves the user experience by fostering students' competence and confidence in using digital resources for learning [P10, P30].

One of the key aspects of user experience in AI-enabled education is improving human-computer interaction. AI technologies enhance the usability, intuitiveness, and effectiveness of educational platforms and tools, improving communication and collaboration between students, teachers, and educational platforms. This improved human-computer interaction improves learning outcomes and enhances user satisfaction [P2, P4, P14, P22, P32, P33]. AI significantly impacts the overall learning experience by increasing student motivation and engagement. Through personalized guidance, AI-powered tools, and customized content delivery, AI creates a more immersive and interactive learning environment. Students benefit from personalized learning experiences, adaptive content, and data-driven insights, leading to improved learning outcomes and a more satisfying user experience [P2, P8, P10, P16, P17, P19, P21, P22, P23, P25, P26, P28, P29, P32, P39, P40, P41, P43, P45, P47, P49].

Furthermore, AI contributes to fairness in education by addressing gender and disability equality and overcoming geographical limitations. By catering to the diverse needs of students and ensuring equal opportunities for all, AI technologies promote an inclusive and equitable learning environment. This fosters a sense of fairness, encourages diversity and inclusivity, and provides a supportive and accessible educational experience for all learners, enhancing the user experience [P3, P13, P46].

Pedagogical or instructional practices also see a significant impact from AI. AI supports innovative teaching methods, practical learning experiences, and personalized instruction, transforming traditional pedagogy. It enables personalized learning experiences, real-world learning opportunities, and cutting-edge teaching methods. These outcomes provide teachers with the tools to deliver effective, engaging, and interactive instruction, ultimately improving the user experience for both teachers and students [P2, P3, P5, P10, P12, P13, P16, P17, P19, P21, P22, P27, P29, P30, P31, P36, P38, P40, P42, P44, P45, P47].

Lastly, AI enables personalized resource allocation, optimizing the distribution of educational resources based on individual student needs. By analyzing data and identifying specific requirements, AI ensures that students receive the right resources at the right time. This personalized resource allocation maximizes learning potential and offers a customized educational experience, enhancing the user experience [P4, P6, P11, P16, P17, P24, P25, P32, P44].

Integrating AI approaches in education presents challenges that directly impact the user experience. Data and analysis emerge as a critical factor affecting the user experience, as emphasized by 57.1% of the articles. Building comprehensive databases and conducting accurate data analysis is essential for successful AI implementation. The availability of reliable data and the ability to interpret it correctly contribute to producing valuable insights and improving learning outcomes. Ensuring the accuracy and validity of AI-generated information is crucial to address ongoing clarification and correction needs [P5, P6, P15, P19, P21, P24, P27, P30, P35, P36, P38, P39, P42, P43, P47, P48, P49].

Ethical and privacy concerns also pose challenges that affect the user experience in AI-enabled education. Approximately 8.3% of the studies highlight these concerns, particularly related to the disclosure of personal information and the potential for machines to gain insights into human emotions. The absence of universal standards and guidelines for AI in education further exacerbates these concerns. Addressing these ethical conundrums, establishing robust frameworks to protect privacy, and promoting responsible use of AI technologies is imperative to ensure a positive user experience [P4, P18, P19, P43, P47].

Integration represents a significant obstacle to adopting AI in education, as discussed in 61% of the articles. Challenges in this category include the cost of implementation, difficulties in integrating AI technology with existing educational systems, and the need for adequate resources in digital learning environments. Overcoming these obstacles requires extensive digital transformation and efficient information management. Additionally, professionals in education must adapt to and embrace AI advancements to maximize the potential benefits for the user experience [P1, P6, P7, P9, P10, P13, P14, P17, P20, P25, P26, P28, P29, P30, P34, P36, P40, P41, P44, P45].

Learning and adaptation are critical aspects that significantly impact the user experience in AI-enabled education. Approximately 22.4% of the studies emphasize the importance of fostering self-learning abilities among students and the need for accurate emotion recognition and analysis to create personalized learning experiences. Adaptive training tailored to students' needs enhances their educational background and contributes to a more positive user experience [P4, P12, P25, P28, P32, P33, P37, P38, P43, P46].

Pedagogical and research challenges also influence the user experience, as highlighted in 22.4% of the studies. Overcoming implementation difficulties, designing effective AI experimental teaching content, and addressing inequalities and gaps in current learning systems are crucial for ensuring equitable access to AI-based education. Research and development efforts in these areas are necessary to enhance the user experience and improve educational outcomes [P9, P10, P12, P23, P27, P29, P31, P39, P44, P46, P49]. Lastly, technology and human interaction play a significant role in shaping the user experience. Building trust in human-computer collaboration and interaction is essential for successfully integrating AI in educational settings. Fostering productive and fruitful learning experiences requires establishing a positive relationship between users and AI technologies [P1, P17, P20, P31, P38].

The distribution of articles across different categories reveals varying levels of research attention. While areas like data and analysis, implementation and integration, and ethical and privacy concerns have received more research focus, categories such as learning and adaptation, technology and human interaction, and pedagogical and research challenges have relatively fewer papers, suggesting potential research gaps. Further studies in these areas are necessary to address the identified challenges, establish best practices, and ensure AI's ethical and effective application in education, ultimately enhancing the user experience.

4. Case Study: LATILL Project

According to a systematic review discussed in hiddensection 10, AI technologies have been applied in education, particularly in language learning, to enhance communication between teachers and students, improve learning outcomes, and provide personalized guidance. The LATILL Project is a case study that utilizes AI to assist German Foreign Language (GFL) teachers in responding to students' questions and offering tailored support. This project focuses on creating digital tools for GFL teachers, incorporating features like search and analysis to identify level-appropriate texts based on the Common European Framework of Reference for Languages (CEFR) [15]. The LATILL platform aims to enhance language programs by promoting a self-regulatory approach to foreign language reading and implicit lexico-grammatical learning [30].

The LATILL project is supported by the Erasmus+ programme of the European Union and involves collaboration with various institutions. Its main objectives include providing open educational resources for German language teachers, emphasizing current and authentic reading materials suitable for different proficiency levels. The project aims to promote innovative teaching methods, learner-oriented materials, and key competences in reading skills. It also focuses on ensuring teaching quality, fostering teacher cooperation, and supporting professional development. The project encompasses an educational platform, a CEFR Reference Corpus for German, didactical materials with methodological recommendations, and a teacher professional development program.

One noteworthy feature of the LATILL platform is the integration of generative AI techniques for creating and developing text bundles. These bundles consist of related texts generated by the system upon user request. AI technologies, such as translation, summarization, and text adaptation, enhance the platform's content and expand its corpus of authentic texts. Additionally, the platform offers an image-generation service to create specific sentences, which further enhances learners' comprehension. In the following hiddensection of this case study, we will go further into the details of the LATILL project, discussing subjects such as the process of developing the platform itself, the methodology employed, user stories and their significance, user testing and its outcomes, and potential future directions for the project.

4.1 User stories

During the development process of the LATILL platform, 41 user stories provided by domain experts played a crucial role in guiding the identification of important platform features. The platform primarily targets teachers of German as a second or foreign language, but it can also benefit pre-service teachers. The platform allows both registered and anonymous users, ensuring inclusivity and accessibility. The prioritized features of the platform were carefully analyzed to meet the requirements mentioned in the user stories. Detailed information about each text, including copyright information and source links, is available in the results viewer. Users can save texts, sort search results, and organize texts based on their preferences. Texts can be exported in popular formats like Word and

PDF, and users can submit texts for analysis and access previously analyzed texts. The platform also provides instructional materials, methodological advice, customizable activities, and a search engine for additional resources. The platform’s user-friendly interface allows users to personalize settings and improve user comfort and accessibility. By addressing the specific needs outlined in the user stories, the LATILL platform provides valuable tools and assistance to teachers, ultimately enhancing students’ educational experiences.

4.2 Platform development

The LATILL project focused on designing and implementing an educational platform for German as a foreign language (GFL) teachers. The platform development process involved creating functional and non-functional prototypes to enhance features and the user interface. Cutting-edge AI technologies, such as NLP and generative AI, were integrated into the platform to provide advanced functionalities like text translation, summarization, and image generation. The platform includes a German text search tool with customizable filters, a comprehensive results viewer, user accounts for text management and storage, and the ability to upload texts for analysis. Methodological tips for working with authentic texts are also provided, and users can personalize the interface settings according to their preferences. The platform development followed the SCRUM methodology [16], emphasizing incremental and iterative development. The platform’s backend was built using Django, a robust Python web framework known for its effectiveness and scalability.

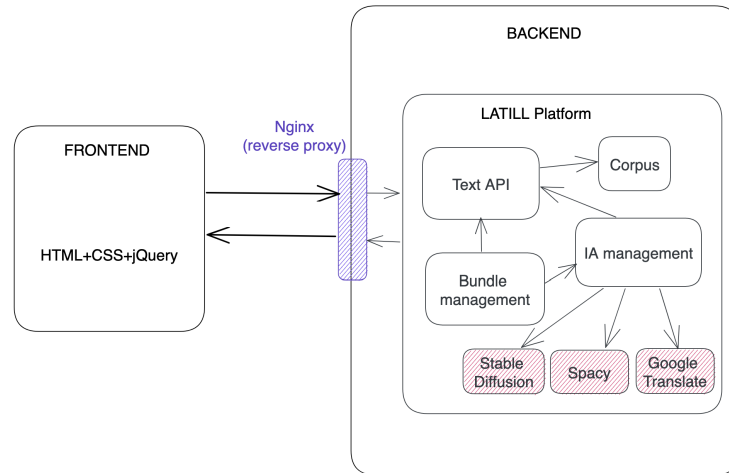


Figura 4: The architecture of the LATILL Platform

The architecture of the LATILL platform, as depicted in Figure 4, incorporates various components and technologies. The front end utilizes HTML, CSS, and jQuery to create an engaging and interactive user interface, structuring content, controlling visual presentation, and adding dynamic functionality. The backend of the platform includes several key components. The text API

enables seamless integration with external sources to retrieve relevant texts, while the corpus management module organizes and manages the CEFR Reference Corpus for German, as the source of up-to-date texts. The bundle management feature allows users to create customized sets of texts for specific learning purposes. To optimize image creation and AI-related tasks, the platform leverages cutting-edge technology, such as the RTX4090 graphics card and CUDA, for fast computations and graphics processing. The performance of the RTX4090 and the parallel computing capabilities of CUDA are utilized to ensure efficient and effective computations for image generation and AI tasks. AI management, including the stable diffusion algorithm, Spacy, and deep-Translator, facilitates image generation, simplification, and translation.

4.2.1 Prototypes

Prototypes in platform development are mockups used to validate design concepts and gather user feedback. They visualize the user interface and functionality, allowing developers to test interactions and refine the design. Prototypes facilitate user testing and help identify usability issues, ensuring the final product meets user expectations.

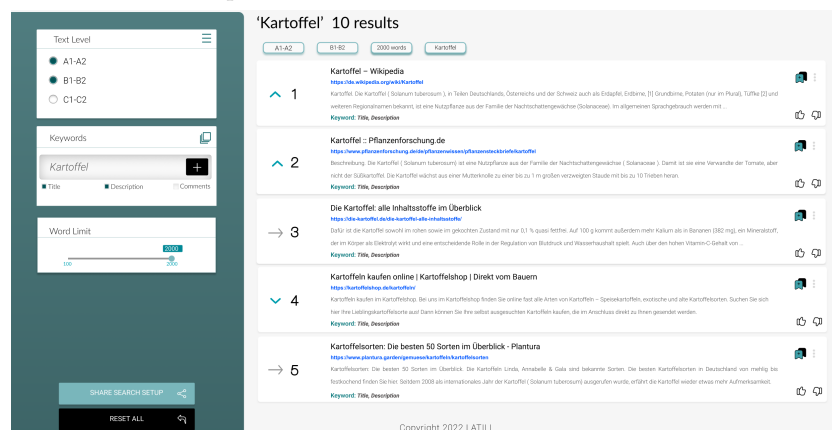


Figura 5: Search results in the nonfunctional prototype

1. Non-functional Prototype:

To shape the idea and improve the final product's vision, a non-functional prototype was created for the LATILL platform. This low-fidelity mockup emphasized the platform's structure, layout, and user experience, developed using the collaborative design tool, Figma³. Key features of the platform were implemented during the prototype development stage, as depicted in Figure 5.

2. Functional Prototype:

A functional platform prototype allows for user feedback and early validation, refinement, and optimization of design, usability, and performance. The LATILL platform utilizes generative AI to automatically generate diverse text bundles, saving time and effort in preparing different variations for educational scenarios.

³ <https://www.figma.com>

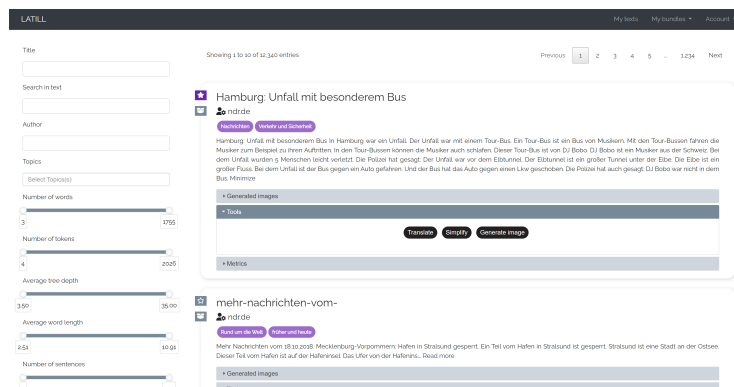


Figura 6: Main page of the first version of LATILL

4.3 User test

The LATILL platform was tested by five German teachers, who performed tasks such as filtering texts, translating them, summarizing them, and creating images from them. The participants were between 40 and 70 years old and encountered difficulties in finding certain functions, particularly related to image generation. Improvements are needed in several areas of the LATILL platform. Firstly, the filtering system must be reviewed to eliminate irrelevant filters and focus on those crucial for text simplification and language learning. The user interface will be simplified, and platform performance will be enhanced. Secondly, carefully considering the color scheme used for generated images is important to increase their educational value and engagement. Enhancing the user experience includes highlighting search terms in the results to facilitate finding relevant textual passages. Users will find it easier to adjust filters while comparing the original and simplified texts through fixed filters on the side of the interface and a reset filter button. Including translated text alongside the original text enables language learners to compare and comprehend the target language. Making the tools available whenever needed improves user accessibility and creates a more personalized and efficient learning environment. For longer texts, breaking them into manageable hidden sections and creating multiple images for each hidden section helps language learners comprehend complex information step by step. Continual improvements in the text simplification and image generation processes are necessary, focusing on preserving the main ideas while making the text simpler through enhancements in the underlying algorithms. The heatmap analysis using Hotjar⁴ revealed that participants used only a small set of filters, further supporting the need for filtering system improvements. Participants expressed satisfaction with the translation feature of the AI tools. Overall, the image generation feature showed promising results, although participants required several attempts to obtain satisfactory outcomes. However, the text simplification feature received criticism for producing summaries that were considered too brief

⁴ <https://www.hotjar.com/>

and not capturing the essence of the original text. Despite these challenges, participants acknowledged the LATILL platform's potential as a valuable tool for teaching German as a foreign language and expressed overall satisfaction with its functionality.

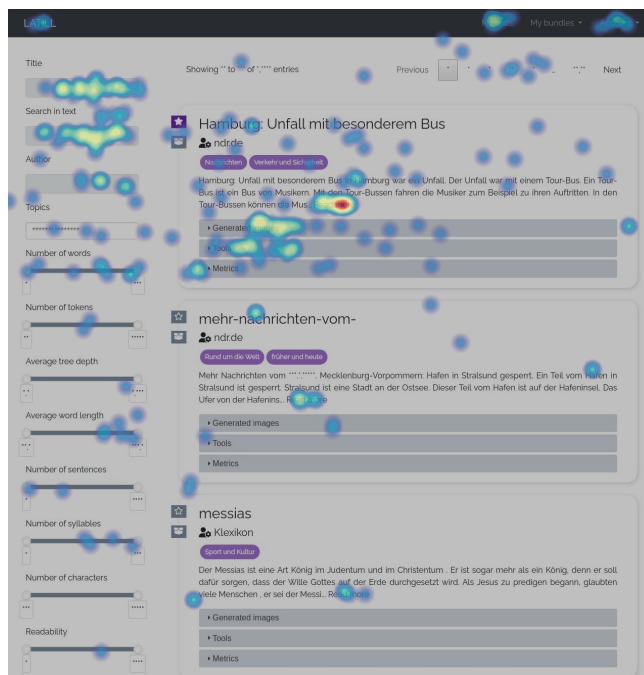


Figura 7: Heatmap of the user interaction captured with Hotjar

4.4 Current Version

The platform underwent significant upgrades based on evaluation and user testing, resulting in improved corpus quality and filter accuracy. The development team analyzed user feedback and observations to enhance functionality and user experience. The interface was redesigned to be more intuitive and user-friendly, aligning with user expectations and preferences. AI-based tools, including translation, text simplification, and image generation, were developed and continuously improved for accuracy and effectiveness in language learning and teaching. The figures below depict visual representations of the platform's diverse functionalities and features.

- Text Filters:** The upgraded filters hiddensection of the platform is presented in Figure 8. The color boxes show different topics that each document has.



Figura 8: Filters of the search

- Tools:** Some changes related to the tools hidden section were done, and the new interface is shown in the following figures:

- Text Translation:** The Translation hidden section of the tools is the same because the user tests and evaluation did not recommend further changes, Figure 9.

Figura 9: Text Translation tool

- Text Simplification:**

By analyzing the structure and content of the text, the feature generates a summary that captures the essential information while eliminating unnecessary details. However, it is important to note that the text simplification tool is currently still under development and will be available in future versions of the platform. In Figure 10, a visual representation showcases the simplification tool, where the selected text is transformed into a more straightforward and easily understandable sentence that encapsulates the text's main idea.

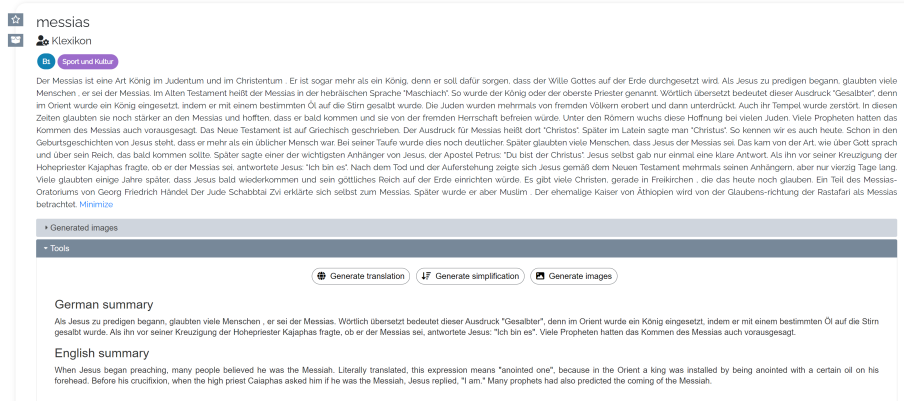


Figura 10: Text Simplification tool

3. Image generation: To make this tool more user-friendly, some changes were done regarding the interface shown in figure 11

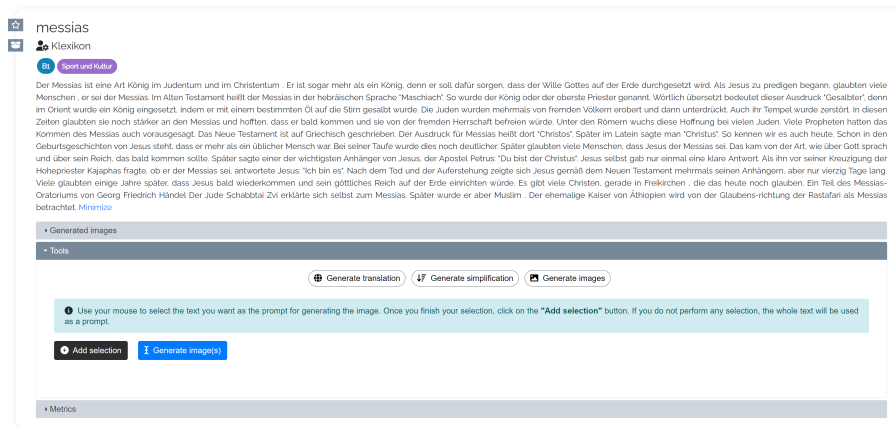


Figura 11: Image generation tool

Figures 12a, 12b, and 12c show the image generated for the selected prompts below from the document:

Prompt 1: *"Das Neue Testament ist auf Griechisch geschrieben"*

Translate: *"The New Testament is written in Greek"*

Prompt 2: *"Nach dem Tod und der Auferstehung zeigte sich Jesus gemäß dem Neuen Testament mehrmals seinen Anhängern, aber nur vierzig Tage lang. Viele glaubten einige Jahre später, dass Jesus bald wiederkommen und sein göttliches Reich auf der Erde einrichten würde. Es gibt viele Christen, gerade in Freikirchen"*

Translate: *"After the death and resurrection, according to the New Testament, Jesus showed himself to his followers several times, but only for forty days. Many believed a few years later that Jesus would soon co-*

me again and establish his divine kingdom on earth. It there are many Christians, especially in free churches”

Prompt 3: *„Auch ihr Tempel wurde zerstört. In diesen Zeiten glaubten sie noch stärker an den Messias und hofften, dass er bald kommen und sie von der fremden Herrschaft befreien würde. Unter den Römern wuchs diese Hoffnung bei vielen Juden. Viele Propheten hatten das Kommen des Messias auch vorausgesagt. Das Neue Testament ist auf Griechisch geschrieben. Der Ausdruck für Messias heißt dort Christus“.*”

Translate: “Their temple was also destroyed. During these times they believed even more strongly in the Messiah and hoped that he would come soon and free them from foreign rule. Under the Romans, this hope grew among many Jews. Many prophets also foretold the coming of the Messiah. The New Testament is written in Greek. The term for Messiah there is Christus”.”



Figure 12: Three samples of generated images

- Metrics:** Figure 13 shows that the platform provides text linguistic metrics. The metrics provide quantitative measurements related to the text, such as the number of words, number of tokens, average tree depth (a measure of sentence complexity), average word length, number of sentences, number of syllables, number of characters, readability score (a measure of how easy the text is to read), syllables per word, and average sentence length. These metrics can help users analyze and compare texts based on their linguistic properties and readability.

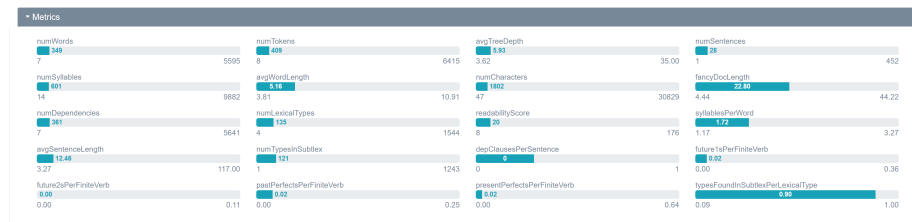


Figure 13: Metrics

- Bundles:** The platform offers bundle configuration options, as depicted in Figure 14. After conducting tests and evaluations, as explained in hidden-

section, it was determined that important changes needed for the bundles are in the user interface. These changes aim to provide a more user-friendly interface allowing easier access to the tools within the bundles.

The image shows a 'Bundle configuration' form with the following elements:

- Title:** Bundle configuration
- Bundle name:** A text input field with the placeholder text 'Give a meaningful title for your bundle'.
- Bundle description:** A larger text input field with the placeholder text 'Describe the contents of your bundle'.
- Options:** Three rounded rectangular buttons with icons and text:
 - Icon: A globe with a plus sign. Text: 'Add translation to every text'.
 - Icon: A downward arrow with a plus sign. Text: 'Add simplification to every text'.
 - Icon: A camera with a plus sign. Text: 'Generate images for every text'.
- Submit:** A prominent blue button at the bottom labeled 'Create bundle'.

Figura 14: Bundles configuration

This platform uses AI to improve language learning, as shown in the figures in this hidden section. Improved accessibility, comprehension, and user engagement result from using AI in text simplification, translation, and image generation. The platform is constantly improving its corpus, user interface, and AI tools to achieve excellence in learning. The most recent version shows improvements in filter performance, interface refinements, and ongoing work on AI-based tools. The team is committed to considering user feedback and meeting their changing needs to deliver a more thorough and user-centric experience.

5. Conclusions

This thesis has clarified the advantages of integrating AI into educational settings to improve user experience. Significant findings and implications have been discovered through a systematic literature review and a case study focusing on a language learning platform. The SLR found that integrating AI into education has numerous positive effects on teachers and students. AI tools offer invaluable administrative assistance, individualized learning plans, better lesson plans, improved evaluation and feedback capabilities, and increased student engagement and inspiration. AI makes learning more effective for students by enabling personalized learning experiences, improved motivation and engagement, immersive and interactive learning environments, and greater accessibility and flexibility. These effects work together to create a transformative learning environment that meets the various needs of students.

An example of how AI is used in practice to improve user experience is shown in the case study of the LATILL Project, a platform for language learning. The platform is constantly being improved due to user feedback and needs, demonstrating a dedication to developing an extensive and user-centric learning environment. An improved language learning experience results from improved corpus quality, filter improvement, interface enhancements, and ongoing AI-based tool development. Future work and research in the field of artificial intelligence in education will likely focus on many areas. These include examining how generative AI can be integrated while addressing ethical issues, performing long-term impact analyses, looking into how AI can be integrated with various pedagogical concepts, concentrating on user-centered design principles, and dealing with implementation difficulties.

The platform's future plans also emphasize ongoing efforts to improve the user experience. These plans now include Language Models for text simplification, allowing students to access texts that have been condensed to their level of language proficiency. Finding the most effective diffusion model to enhance image quality also aims to produce more aesthetically pleasing and educational learning materials. These upcoming improvements show the platform's attention to meeting the changing demands of language learners to deliver an improved and all-encompassing language learning experience by integrating Generative AI tools like LLMs and improved image diffusion models.

6. ACKNOWLEDGMENTS

The project was undertaken with the support of the Erasmus+ Programme of the European Union: "KA2 - Cooperation partnership in school education." Level-Adequate Texts in Language Learning (LATILL) (Reference number 2021-1-AT01-KA220-SCH-000029604). Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA).

I also acknowledge partial financial support from project Analitica visual como apoyo a la Inteligencia Artificial explicable with reference R035.

Referencias

1. Licklider, J. C. R. (1960). Man-computer symbiosis. *IRE Transactions on Human Factors in Electronics*, 1, 4-11.
2. Cerf, V. G. (2013). Augmented intelligence. *IEEE Internet Computing*, 17(5), 96-96.
3. Engelbart, D. C. (1962). *Augmenting human intellect: A conceptual framework*. Menlo Park, CA, 21.
4. Kay, A. (1998). *The computer revolution hasn't happened yet*. University Video Communications.
5. Chalmers, D. J. (2016). The singularity: A philosophical analysis. In *Science fiction and philosophy: From time travel to superintelligence*, pp. 171-224. Wiley Online Library.

6. Nguyen, A., Yosinski, J., Clune, J. (2016). Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. arXiv preprint arXiv:1602.03616.
7. Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4), 834-848.
8. Karras, T., Laine, S., Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401-4410.
9. Machinery, C. (1950). Computing machinery and intelligence-AM Turing. *Mind*, 59(236), 433.
10. McCulloch, W. S., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 115-133.
11. McCarthy, J., Minsky, M. L., Rochester, N., Shannon, C. E. (2006). A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, 27(4), 12.
12. García-Peñalvo, F. J. (2022). Developing robust state-of-the-art reports: Systematic Literature Reviews. *Education in the Knowledge Society*, 23, e28600. DOI: 10.14201/eks.28600.
13. Walter, C. (2005). Kryder's law. *Scientific American*, 293(2), 32-33.
14. Good, I. J. (1966). Speculations concerning the first ultraintelligent machine. In *Advances in computers*, Vol. 6, pp. 31-88. Elsevier.
15. Soproni, Z. (2020). Common European Framework of Reference for Languages: Learning, Teaching, Assessment, Companion Volume with New Descriptors. *Modern Nyelvoktatás*, 26(1-2), 168-170.
16. Rising, L., Janoff, N. S. (2000). The Scrum software development process for small teams. *IEEE software*, 17(4), 26-32.
17. What is iCall?, JRahm, <https://community.f5.com/t5/technical-articles/what-is-icall/ta-p/288206>
18. Spacy documentation, <https://spacy.io/usage>
19. Stable LM documentation, <https://github.com/Stability-AI/StableLM>
20. Dolly documentation, <https://github.com/databricks/dolly>
21. Stanford Alpaca documentation, https://github.com/tatsu-lab/stanford_alpaca
22. Kuo, Hsu-Chan, Tseng, Yuan-Chi, Yang, Ya-Ting Carolyn: Promoting college student's learning motivation and creativity through a STEM interdisciplinary PBL human-computer interaction system design and development course. *Thinking Skills and Creativity*, 31, 1-10 (2019), Elsevier.
23. Guzman, Andrea L, Lewis, Seth C: Artificial intelligence and communication: A human-machine communication research agenda. *New Media & Society*, 22(1), 70-86 (2020), Sage Publications Sage UK: London, England.
24. Shneiderman, Ben: Human-centered artificial intelligence: Three fresh ideas. *AIS Transactions on Human-Computer Interaction*, 12(3), 109-124 (2020).
25. Lowgren, Jonas, Carroll, John M, Hassenzahl, M, Erickson, T, Blackwell, A: The encyclopedia of human-computer interaction. *Interaction design foundation* (2019).
26. Holmes, Wayne, Bialik, Maya, Fadel, Charles: Human-centered artificial intelligence in education: Seeing and mitigating bias with ai. *International Journal of Artificial Intelligence in Education*, 29(3), 411-431 (2019), Springer.

27. Kitchenham, Barbara, Brereton, O. Pearl, Budgen, David, Turner, Mark, Bailey, John, Linkman, Stephen: Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology*, 51(1), 7-15 (2009), Elsevier.
28. Petticrew, Mark, Roberts, Helen: *Systematic Reviews in the Social Sciences: A Practical Guide*. Volume 11. ISBN: 9781405121101. Wiley (2006).
29. Kitchenham, Barbara A, Budgen, David, Brereton, O Pearl: Using mapping studies as the basis for further research—a participant-observer case study. *Information and Software Technology*, 53(6), 638–651 (2011), Elsevier.
30. Kienberger, M., García-Holgado, A., Schramm, K., Raveling, A., Meurers, D., Labinska, B., Koropatnitska, T. & Therón, R. Enhancing Adaptive Teaching of Reading Skills Using Digital Technologies: The LATILL Project. (2023,5)
31. Page, Matthew J, McKenzie, Joanne E, Bossuyt, Patrick M, Boutron, Isabelle, Hoffmann, Tammy C, Mulrow, Cynthia D, Shamseer, Larissa, Tetzlaff, Jennifer M, Akl, Elie A, Brennan, Sue E, et al.: The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *International Journal of Surgery*, 88, 105906 (2021), Elsevier.
32. Kinnula, M., Iivari, N., Kuure, L. & Molin-Juustila, T. Educational Participatory Design in the Crossroads of Histories and Practices—Aiming for Digital Transformation in Language Pedagogy. *Computer Supported Cooperative Work (CSCW)*. pp. 1-36 (2023)
33. Roldan, W., Gao, X., Hishikawa, A., Ku, T., Li, Z., Zhang, E., Froehlich, J. & Yip, J. Opportunities and challenges in involving users in project-based HCI education. *Proceedings Of The 2020 CHI Conference On Human Factors In Computing Systems*. pp. 1-15 (2020)
34. Marín-Morales, Javier, Llinares, Carmen, Guixeres, Jaime, Alcañiz, Mariano: Emotion recognition in immersive virtual reality: From statistics to affective computing. *Sensors*, 20(18), 5163 (2020), MDPI.
35. Elmquaddem, Nouredine: Augmented reality and virtual reality in education. Myth or reality? *International journal of emerging technologies in learning*, 14(3) (2019).
36. Mathew, Ann Neethu, Rohini, V, Paulose, Joy: NLP-based personal learning assistant for school education. *Int. J. Electr. Comput. Eng. (2088-8708)*, 11, 4522–4530 (2021).
37. Deniz, Sarah, Lee, Dakyung, Kurian, Grace, Altamirano, Lourdes, Yee, Darren, Ferrá, Michael, Hament, Blake, Zhan, Justin, Gewali, Laxmi, Oh, Paul: Computer vision for attendance and emotion analysis in school settings. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 0134–0139 (2019), IEEE.
38. Al Mahdi, Z., Rao Naidu, V. & Kurian, P. Analyzing the Role of Human Computer Interaction Principles for E-Learning Solution Design. *Smart Technologies And Innovation For A Sustainable Future: Proceedings Of The 1st American University In The Emirates International Research Conference—Dubai, UAE 2017*. pp. 41-44 (2019)
39. Jiang, Yuyin, Li, Xiaomei: Intelligent online education system based on speech recognition with specialized analysis on quality of service. *International Journal of Speech Technology*, 23, 489–497 (2020), Springer.
40. Wong-Villacres, M., Alvarado Garcia, A. & Tibau, J. Reflections from the classroom and beyond: Imagining a decolonized hci education. *Extended Abstracts Of The 2020 CHI Conference On Human Factors In Computing Systems*. pp. 1-14 (2020)

41. Luan, Hui, Tsai, Chin-Chung: A review of using machine learning approaches for precision education. *Educational Technology & Society*, 24(1), 250–266 (2021), JS-TOR.
42. King, Michael R, ChatGPT: A conversation on artificial intelligence, chatbots, and plagiarism in higher education. *Cellular and Molecular Bioengineering*, 16(1), 1–2 (2023), Springer.
43. Manikandan, S, Dhanalakshmi, P, Priya, S, Teena, A Mary Odilya: Intelligent and deep learning collaborative method for E-learning educational platform using TensorFlow. *Turkish Journal of Computer and Mathematics Education*, 12(10), 2669–2676 (2021), Karadeniz Technical University Distance Education Research and Application.
44. Xu, W., Dainoff, M., Ge, L. & Gao, Z. From human-computer interaction to human-AI Interaction: new challenges and opportunities for enabling human-centered AI. *ArXiv Preprint ArXiv:2105.05424*. **5** (2021)
45. Gao, Peng, Li, Jingyi, Liu, Shuai: An introduction to key technology in artificial intelligence and big data driven e-learning and e-education. *Mobile Networks and Applications*, 26(5), 2123–2126 (2021), Springer.
46. Dotter, Aaron, Chaboyer, Brian, Jevremović, Darko, Kostov, Veselin, Baron, E, Ferguson, Jason W: The Dartmouth stellar evolution database. *The Astrophysical Journal Supplement Series*, 178(1), 89 (2008), IOP Publishing.
47. Li, Jiehao, Wang, Junzheng, Peng, Hui, Zhang, Longbin, Hu, Yingbai, Su, Hang: Neural fuzzy approximation enhanced autonomous tracking control of the wheel-legged robot under uncertain physical interaction. *Neurocomputing*, 410, 342–353 (2020), Elsevier.
48. Yang, Chenguang, Chen, Chuize, He, Wei, Cui, Rongxin, Li, Zhijun: Robot learning system based on adaptive neural control and dynamic movement primitives. *IEEE transactions on neural networks and learning systems*, 30(3), 777–787 (2018), IEEE.
49. Jiang, L. (2021). *Virtual reality action interactive teaching artificial intelligence education system*. *Complexity*, 2021, 1-11.
50. Schiefelbein, J., Chounta, I. A., & Bardone, E. (2019). *To gamify or not to gamify: Towards developing design guidelines for mobile language learning applications to support user experience*. *Transforming Learning with Meaningful Technologies: 14th European Conference on Technology Enhanced Learning, EC-TEL 2019, Delft, The Netherlands, September 16–19, 2019, Proceedings*, 626-630.
51. Korozi, M., Stefanidi, E., Samaritaki, G., Prinianakis, A., Katzourakis, A., Leonidis, A., ... & Antona, M. (2019). *Shaping the intelligent classroom of the future*. *HCI International 2019–Late Breaking Posters: 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings*, 200-212.
52. Ruan, S., Jiang, L., Xu, J., Tham, B. J., Qiu, Z., Zhu, Y., ... & Landay, J. A. (2019). *Quizbot: A dialogue-based adaptive learning system for factual knowledge*. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 1-13.
53. Sales, J., Tzafilkou, K., Koumpis, A., Gees, T., Zimmermann, H., Protogeris, N., & Handschuh, S. (2020). *Let the End User in Peace: UX and Usability Aspects Related to the Design of Tutoring Systems*. *Intelligent Tutoring Systems: 16th International Conference, ITS 2020, Athens, Greece, June 8–12, 2020, Proceedings*, 23-26.
54. Meis, E., Pugh, S., Dickler, R., Tissenbaum, M., & Hirshfield, L. (2022). *HCI Strategies for Informing the Design of a Teacher Dashboard: How Might Real-Time Situational Data Determine the Potential for Technological Support in the Classroom?*. *International Conference on Human-Computer Interaction*, 161-168.

55. Kuo, J. Y., Song, X. T., Chen, C. H., & Patel, C. D. (2021). *Fostering design thinking in transdisciplinary engineering education. Transdisciplinary Engineering for Resilience: Responding to System Disruptions*; IOS Press: Amsterdam, The Netherlands, 63-70.
56. Ciolacu, M. I., & Svasta, P. (2021). *Education 4.0: AI empowers smart blended learning process with Biofeedback. 2021 IEEE Global Engineering Education Conference (EDUCON)*, 1443-1448.
57. Feng, W. (2019). *Design for Teaching Surroundings Based on Human-Computer Interaction Techniques. Design, User Experience, and Usability. Application Domains: 8th International Conference, DUXU 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings, Part III*, 503-513.
58. Wang, T., & Park, J. (2021). *Design and implementation of intelligent sports training system for college students' mental health education. Frontiers in psychology*, 12, 634978.
59. Liu, S., & Ba, L. (2021). *Construction and Implementation of Embodied Mixed-Reality Learning Environments. 2021 International Conference on Big Data Engineering and Education (BDEE)*, 126-131.
60. Sasaki, Y., Fukui, M., Hagikura, J., Moriyama, J., & Hirashima, T. (2020). *Development of an interactive educational tool to experience machine learning with image classification. 2020 IEEE 9th Global Conference on Consumer Electronics (GCCE)*, 78-80.
61. Mackare, Kristine and Jansone, Anita. Artificial Intelligence aspects in developed E-material formatting application. In Proceedings of the 2019 International Conference on Artificial Intelligence and Advanced Manufacturing, 1–6 (2019).
62. Eltahir, Asim Mohammed, Abdulla, Hussam, Platos, Jan, and Snasel, Vaclav. An Overview of Chatbot Structure and Source Algorithms. In 2022 26th International Conference on Circuits, Systems, Communications and Computers (CSCC), 91–98 (2022). IEEE.
63. Fisher, Joshua A and Samuels, Janice Tisha. A proposed curriculum for an introductory course on interactive digital narratives in virtual reality. In International Conference on Interactive Digital Storytelling, 462–477 (2021). Springer.
64. Weng, Xiaojing and Chiu, Thomas KF. Instructional design and learning outcomes of intelligent computer assisted language learning: Systematic review in the field. In Computers and Education: Artificial Intelligence, 100117 (2023). Elsevier.
65. Heift, Trude. Intelligent Computer-Assisted Language Learning. In Intelligent Computer Assisted Language Learning, edited by Trude Heift, 2012, doi:10.1002/9781405198431.wbeal0548.
66. Warschauer, Mark. Computer-assisted language learning: An introduction. In Multimedia language teaching, 320 (1996).
67. Lu, Xiaofei. "Natural Language processing and intelligent computer-assisted language learning (ICALL). In The TESOL encyclopedia of English language teaching, 1–6 (2018). Wiley Online Library.
68. Heift, Trude. Intelligent Computer Assisted Language Learning. In Research Questions in Language Education and Applied Linguistics: A Reference Guide, 655–658 (2021). Springer International Publishing. doi:10.1007/978-3-030-79143-8-114.

Sistema de auditoría para rastrear, descubrir y reducir el sesgo de género en sistemas de Procesamiento del Lenguaje Natural

Samuel Adrados González, Vivian Félix López Batista, Gabriel Villarrubia González

Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
`{i_c048842,vivian,gvlg}@usal.es`

Resumen En un mundo cada vez más dependiente del uso de la Inteligencia Artificial y de los sistemas de aprendizaje automático, es crucial asegurarse de que estos sistemas no contengan sesgos. A medida que se va confiando cada vez más en su uso, la influencia en la sociedad aumenta, por lo que se debe asegurar, de forma imprescindible, que sean justos. No se puede permitir que estén diseñados para perpetuar injusticias, sino que deben ser entrenados para ser imparciales y equilibrados en la toma de decisiones. En este trabajo se va a desarrollar una revisión de la literatura científica más relevante del sesgo en el Procesamiento del Lenguaje Natural, más en concreto, el sesgo en los *word embeddings*, poniéndose el foco sobre el sesgo en el aprendizaje automático para poder tratarlo desde un punto de vista más amplio. También se ha desarrollado un sistema de auditoría de los sesgos, en el que se pueden crear y gestionar *word embeddings* así como identificar, medir y mitigar los sesgos presentes, mediante la aplicación de los métodos más relevantes, además de la visualización de los resultados mediante gráficas y tablas. Para comprender los métodos utilizados se realizan casos de estudios sencillos, utilizando el sistema desarrollado, permitiendo explorar como funcionan y mostrando los efectos que tienen en las propiedades geométricas de las incrustaciones de palabras.

Keywords: PLN, Procesamiento del Lenguaje Natural, sesgo, mitigación del sesgo, eliminación del sesgo, método, visualización, género, incrustación de palabras, vector de palabras, NLP, Natural Language Processing, bias, debias, debiasing, method, visualization, gender, word embedding, word vector.

1. Introducción

Vivimos en una sociedad en constante evolución, donde los sistemas de aprendizaje automático y el Procesamiento del Lenguaje Natural (PLN) desempeñan un papel cada vez más importante en nuestro día a día. Estos sistemas han demostrado ser herramientas poderosas y versátiles en diversas áreas, desde la

detección de emociones en el análisis de sentimientos hasta la traducción automática de idiomas. Sin embargo, a medida que confiamos cada vez más en ellos, surge la necesidad crítica de examinar y abordar los sesgos y disparidades que pueden contener estos sistemas.

Dado que las sociedades humanas poseen sesgos y desigualdades estructurales, las herramientas de aprendizaje automático inevitablemente también los reflejan, por lo que es inaceptable que los desarrolladores de estas herramientas pasen por alto los riesgos que conllevan estos sesgos. La única forma de solucionarlo sería eliminar definitivamente tanto los prejuicios como las desigualdades existentes en la sociedad. Esto se sabe que es muy difícil de conseguir, por lo que el aprendizaje automático y el PLN tan sólo pueden ofrecer un apoyo limitado para conseguirlo. Sin embargo, es importante asegurarse de que se contrarresten estas desigualdades, en lugar de reforzarlas. El impacto de estos sesgos puede ser perjudicial tanto para los individuos, como para la sociedad. Un ejemplo de ello es el estudio de Kiritchenko et al. [1], en el que se analizaron 219 sistemas PLN de predicción de intensidad de sentimientos y emociones. Sus hallazgos revelaron que más del 75% de los sistemas tendían a asignar puntuaciones de intensidad más altas a frases que involucraban un género o raza específicos en comparación con las frases que involucraban otros géneros o razas. Esto puede tener consecuencias graves, como la censura de frases que contengan ciertas características demográficas o incluso un aumento de la intensidad de los sesgos con el tiempo.

Afortunadamente, en los últimos años ha habido un creciente interés en investigar y abordar el sesgo en el PLN. Sin embargo, aún queda mucho por explorar y comprender plenamente. Es fundamental realizar un análisis crítico de los algoritmos y modelos utilizados en estos sistemas, así como buscar soluciones que permitan identificar, medir y mitigar los sesgos. La Figura 1, muestra el número de publicaciones por año en Scopus al buscar *Bias in NLP*, evidenciando el creciente interés en este campo.

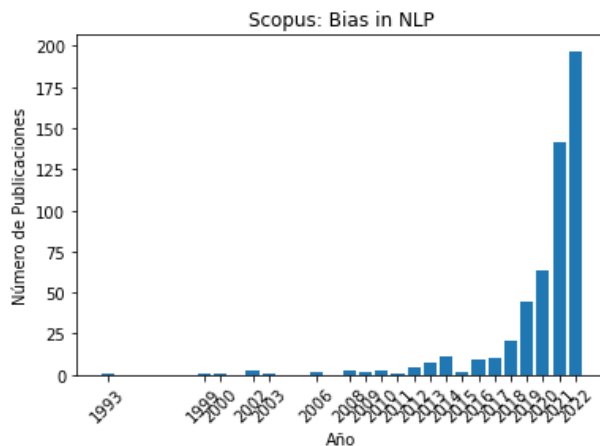


Figura 1: Búsqueda en Scopus de "Bias in NLP".

Word embedding o incrustación de palabra es el nombre de un conjunto de modelos de lenguaje y técnicas de aprendizaje en PLN en donde las palabras o frases del lenguaje natural son representadas como vectores de números reales. Conceptualmente implica el encaje o la incrustación matemática de un espacio con una dimensión por palabra, a un espacio vectorial continuo con menos dimensiones [2]. Estos modelos son actualmente un componente esencial en la mayoría de las tareas de PLN, pero se ha demostrado que contienen y amplifican los sesgos contenidos en los datos con los que son entrenados. Lo que ha dado lugar a numerosas publicaciones en las que se muestran nuevos métodos utilizados para medir y mitigar los sesgos que puedan contener estos modelos. En este trabajo nos centramos sobre todo en tratar el sesgo de género en los *word embeddings* entrenados con corpus en inglés, ya que la mayoría de las investigaciones sobre el sesgo de estos modelos se realizan con estas características, y, por lo tanto, hay más información disponible. Cada una de las dimensiones de estos modelos o la combinación de ellas puede corresponderse con una característica de la palabra asociada al modelo, como por ejemplo, el género o la raza.

En este trabajo se realiza una revisión general de la literatura relacionada con los sesgos y el daño que produce o puede producir en el aprendizaje automático. Una vez que se ha entendido de forma general el problema, nos centramos en cómo se aborda la identificación, medición y mitigación de los sesgos de géneros en los *word embeddings* así como sus limitaciones. Además, se presenta una sistema de auditoría que permite una visualización interactiva de métodos de identificación, medición y mitigación del sesgo en los *word embeddings* para que los usuarios puedan obtener una comprensión técnica de forma visualmente intuitiva de su funcionamiento mediante las propiedades geométricas de estos modelos. Para una mejor comprensión, en este trabajo se implementan casos de estudio sencillos que exploran los efectos de estas técnicas en la geometría de los vectores.

Este sistema se puede utilizar como herramienta de ayuda tanto para los profesionales de PLN que diseñan sistemas de toma de decisiones utilizando *word embeddings*, como para los investigadores que trabajan en la equidad, igualdad o ética de los sistemas de aprendizaje automático en PLN. Además, es útil para que los usuarios puedan interactuar y observar los efectos del sesgo en estos vectores de palabras. Así como para mejorar los datos que se utilizan en la mayoría de las tareas de PLN en el aprendizaje automático, garantizando una mejora en la equidad, además de servir como medio visual para el aprendizaje, ayudando a novatos en PLN a comprender y a mitigar los sesgos en las incrustaciones de palabras.

2. Revisión de la Literatura

En primer lugar, se consultó el libro *Fairness and Machine Learning* para obtener una perspectiva general de la justicia en el aprendizaje automático [3]. En el artículo [4], se proporciona una perspectiva inicial sobre el sesgo de género en el PLN y en los *word embeddings*, señalándose la falta de consenso en cuanto a la mejor forma de abordar la detección y evaluación de sesgo de género. En

otro artículo [5], se muestra un enfoque práctico de como trabajar en Python con *word embeddings* y reducir el sesgo de género utilizando el método y los conceptos propuestos en [6]. También se ha realizado una revisión de literatura [7], siendo una de las bases de referencia para la realización del trabajo, ya que se enfoca en una revisión de los estudios más relevantes anteriores al año 2019 de detección y mitigación de sesgo en el PLN.

Para complementar esta información se han visualizado conferencias recientes relacionadas con el sesgo en el PLN, para obtener un contenido más actualizado sobre el tema [8][9]. Además, se han seleccionado los artículos más relevantes en las bases de datos bibliográficas más importantes, *Scopus* y *Web Of Science* utilizando el término de búsqueda *bias in word embedding* desde el año 2020 hasta la fecha actual, para completar aún más la información recopilada.

2.1 El Sesgo en el Aprendizaje Automático

Para entender mejor como se propagan las disparidades o sesgos sociales a través del proceso del aprendizaje automático, se van a explicar las etapas de un sistema de aprendizaje automático [4].

El hecho de que la sociedad haya manifestado desigualdades demográficas y sesgos durante la historia, afecta a la hora de medir el estado del mundo real y transformarlo en datos, siendo inevitable que los datos de entrenamiento utilizados para el aprendizaje automático, no reflejen estas desigualdades.

La medición de los datos debe tener como paso previo la definición de variables de interés. Estas variables almacenarán los datos obtenidos de las observaciones del mundo real. Hay que tener en cuenta que las observaciones pueden estar sesgadas, ya que puede no haber una suficiente representación de todos los grupos que se observan. Incluso la medición puede no ser exacta, ya que pudiera darse el caso de que la observación no se pueda codificar directamente.

En el aprendizaje automático, cuando un modelo aprende de los datos, extrae los estereotipos, incluyendo aquellos que son incorrectos o dañinos, de la misma forma que extraen conocimiento.

Para poder comprender las propiedades de una predicción, se requiere comprender no solo el modelo, sino también las diferencias de población entre los grupos a los que se les aplican las predicciones. Una de las limitaciones importantes en el aprendizaje automático es que este solamente revela correlaciones, aunque a menudo se utilizan como si revelara causalidad, pero la correlación no implica causalidad.

Para entender los conceptos de correlación y causalidad, y porqué la correlación no implica causalidad, se utiliza como ejemplo uno de los primeros sistemas de aprendizaje automático en cuidados de la salud [10], el cual aprendió la regla de que los pacientes con asma tenían un menor riesgo de desarrollar neumonía. Esto era un patrón que se reflejaba en los datos afectando en los resultados del sistema, indicando que la causalidad de tener una baja probabilidad de padecer neumonía, era tener asma. Pero, en realidad, la razón más probable era que los pacientes asmáticos tendían a recibir más atención hospitalaria frente a los pacientes no asmáticos, por lo que ya no supondría una causa directa sino una

correlación, ya que estas dos variables varían conjuntamente, pero una no causa a la otra.

Muchos sistemas de aprendizaje automático reciben retroalimentación cuando realizan predicciones, pero la retroalimentación es difícil de interpretar correctamente, ya que a veces no se puede saber si la retroalimentación está sesgada, pudiendo reflejar prejuicios culturales, los cuales son difíciles de caracterizar. Además, hay muchas formas no intencionadas en la que estos sesgos pueden surgir en la retroalimentación.

2.2 Sesgos en el PLN

El sesgo se puede categorizar en sesgo de asignación y sesgo de representación. Los daños causados por el sesgo de asignación surgen cuando un sistema de aprendizaje automático asigna recursos u oportunidades injustamente a diferentes grupos sociales. El sesgo de asignación es inmediato, fácil de cuantificar, discreto y transaccional. El sesgo de representación surge cuando un sistema representa a algunos grupos sociales de una forma menos favorable, los degrada, o falla en reconocer su existencia de forma completa, teniendo consecuencias a largo plazo, siendo difícil de formalizar, difuso y cultural. En el artículo [11] se lleva a cabo una clasificación del sesgo de representación obteniendo cuatro tipos diferentes:: estereotipado, reconocimiento, denigración y subrepresentación.

El primero refuerza los estereotipos existentes en la sociedad, como asociar determinados oficios con mayor frecuencia o de forma absoluta a determinados grupos frente a otros. El sesgo de reconocimiento se produce cuando el sistema elimina o hace invisible a un grupo. Por ejemplo, el reconocimiento de una cara en una imagen. La denigración consiste en asociar términos dañinos o indeseables a ciertos grupos y la subrepresentación se presenta cuando un determinado grupo no aparece o aparece de forma muy limitada.

Hay que tener en cuenta que, dentro de los sesgos de representación, el sesgo de género y el sesgo racial son dos de los tipos de sesgo más abordados en el PLN.

En el artículo [12], se realiza una revisión sobre 146 artículos que analizan el sesgo en los sistemas PLN, encontrando que las motivaciones son a menudo vagas, inconsistentes y presentan falta de razonamiento normativo. Se ha observado discordancia entre las motivaciones y técnicas cuantificativas de los artículos propuestos para medir el sesgo. Además, se han dado cuenta de que los artículos sobre este campo no recogen literatura relevante externa al PLN. Por lo que proponen que se busque una literatura relevante externa al campo del PLN; que provean definiciones explícitas de por qué el comportamiento del sistema o de los sistemas sesgados son dañinos, de qué formas, y a quién; y que se aborden las experiencias vividas por las comunidades afectadas por los sesgos de los sistemas PLN.

2.3 Identificación y Medición del Sesgo en el PLN

Las técnicas comúnmente empleadas para la detección y medición de sesgo en el PLN se han dividido en tres subapartados: pruebas de asociación, *word embeddings* y diferencias de rendimiento entre grupos.

2.3.1 Pruebas de Asociación. Para medir el sesgo en los *word embeddings* se usa la Prueba de Asociación de Word Embeddings (*Word Embedding Association Test*, WEAT) [13], la cual mide la diferencia de asociación de los dos conjuntos de palabras objetivo (con estereotipos femeninos y masculinos) con los dos conjuntos de atributos protegidos (femenino y masculino) entre la asociación diferencial de todas las palabras objetivo con los dos conjuntos de palabras objetivos. La métrica del sesgo se corresponde con el *tamaño de efecto* y también se calcula el *valor p*.

Tras la publicación de WEAT, se desarrolló una extensión para medir los sesgos en los codificadores de frases o *contextual embeddings* denominada SEAT (*Sentence Encoder Association Test*) [14], y se muestra que las incrustaciones contextuales ELMo (*Embeddings from Language Models*) [15] y BERT (*Bidirectional Encoder Representations from Transformers*) [16] contienen menos evidencia de sesgos de asociación comparados con los *word embeddings*.

En el artículo [13], se comenta que es posible que las plantillas de las oraciones de SEAT puede que no estén “tan semánticamente blanqueadas”, pudiendo ser esta una de las razones por las que se observa menor sesgo en ELMo y BERT.

Uno de los problemas de las pruebas de asociación es que estas pruebas pueden ser usadas para identificar si los modelos codifican sesgos, pero la falta de evidencia de sesgo empleando estos métodos no demuestra la inexistencia de sesgo.

2.3.2 Word Embeddings. El artículo [6] fue el precursor de la investigación posterior en el sesgo intrínseco de los *word embeddings*. En este artículo se distingue entre las palabras con un género específico por definición (*man, woman, brother, sister...*), y las palabras restantes consideradas palabras neutrales en cuanto al género (*nurse, doctor, arts, football...*). Si existe una analogía entre una palabra neutral en cuanto al género y un par de palabras de género, se le denomina sesgo directo, proponiéndose el método de medición *Direct Bias* o Sesgo Directo. También se denomina sesgo indirecto a la asociación entre dos palabras neutrales en cuanto al género que están claramente asociadas por el género, proponiéndose el método de medición *Indirect Bias* o Sesgo Indirecto. Además, presenta por primera vez el concepto de subespacio o dirección de género, y explica cómo identificarlo.

En el artículo [17], se extiende el método anterior, pudiendo utilizarse para encontrar sesgos de géneros no binarios, o cualquier tipo de sesgo multiclase, agregando n-tuplas en vez de pares de género.

Sin embargo, en el artículo [18] se dieron cuenta de que el artículo [6] fallaba en identificar el espacio completo del sesgo de género en los subespacios de

género. Incluso después de que las proyecciones de los *word embedding* de género neutral en el subespacio de género se hayan eliminado (mitigando el sesgo de género), estas palabras con sesgos similares se agrupan entre sí, introduciendo la noción de sesgo de grupo.

Otro método de identificación del sesgo es el que se propone en el artículo [19], en el que se entrena un clasificador cuya precisión indica lo fácil o difícil que es predecir el atributo protegido.

2.3.3 Diferencias de Rendimiento entre los Grupos. En el artículo [20] se realiza una cuantificación del sesgo presente en un corpus de comentarios examinando aquellos comentarios que contienen ciertos términos y verificando los falsos positivos, como el caso de *I am a gay man*. Al analizar la palabra *gay* se comprueba que aparece en el 3% de los comentarios tóxicos, pero solo en el 0.5% de todos los comentarios en general.

En [21] se propone el método *WinoBias* que pretende medir el grado de sesgo de un sistema de aprendizaje automático mediante la diferencia de evaluación al intercambiar dos palabras en una oración. Esta puntuación o medida de rendimiento puede calcularse mediante la tasa de error de los diferentes grupos, comparando los falsos positivos, los falsos negativos, etc.

En [22], se diseñó un Conjunto de Datos de Prueba de Evaluación de Sesgos de Género (*Gender Bias Evaluation Test-sets*, GBETs) independiente basado en el estilo de oraciones utilizado en *WinoBias*. Para cada oración se considera tres tipos de pronombres (femenino, masculino o neutro). Además, se diseñaron métricas para analizar el sesgo de género, examinando como la diferencia de rendimiento entre géneros con respecto a cada ocupación laboral se correlacionaba con las estadísticas ocupacionales de género de la Oficina de Estadísticas Laborales de Estados Unidos.

Otro ejemplo de identificación de sesgo que analiza las tasas de error, es el caso del artículo [23], en el cual se examinan cinco sistemas de reconocimiento de habla automáticos (Amazon, Apple, Google, IBM y Microsoft), entre los cuales se utilizó una muestra de 42 personas angloparlantes blancas y 73 personas angloparlantes negras, obteniendo una tasa de error para las personas negras de 0.35 frente al 0.19 de las personas blancas.

Un problema que se descubrió en la evaluación del sesgo mediante las diferencias de rendimiento de grupos, es que las frases pueden estar sesgadas, ya que al introducir dos palabras en una frase para saber cuál es el más probable y medir el sesgo, estas frases pueden ser poco comunes en el mundo real, como en el caso de la frase *the straight/gay man drew his gun*, normalmente no se menciona *straight* (heterosexuales) en las oraciones porque se suele asumir que todos los hombres son heterosexuales hasta que se indica explícitamente que es homosexual [24].

2.4 Mitigación del Sesgo en el PLN

Se presentan algunos de los métodos más relevantes, aunque imperfectos, comúnmente empleados para mitigar el sesgo en los sistemas PLN. Los métodos

de mitigación del sesgo se ha dividido en: mitigación del sesgo en el corpus, mitigación de sesgo en los *word embeddings*, y mitigación de sesgo en algoritmos de ajuste.

2.4.1 Mitigación del Sesgo en el Corpus. Uno de los métodos más empleados es el aumento de datos, en el que se altera programáticamente el texto invirtiendo los géneros y después combinando los datos originales con los datos manipulados, por ejemplo *doctor run because he is late* se convierte en *doctor run because she is late* [25].

Otro método es el etiquetado de género (masculino o femenino, M o F) para cada oración del corpus, tanto en el momento del entrenamiento como en el de la traducción. Esto se realiza en [26] para mantener la concordancia morfológica al traducir del inglés al francés. Sin embargo, esta solución requiere metadatos adicionales del género de los usuarios que no siempre se pueden adquirir.

El ajuste fino (*fine-tuning*) del sesgo es un método que utiliza la transferencia del aprendizaje mediante un conjunto de datos grande sin o con poco sesgo, para asegurarse que el modelo contiene un sesgo mínimo antes de realizar el ajuste fino del modelo [27].

Una opción similar es contratar personas que modifiquen el corpus para hacer etiquetas contrafactuales aplicables para después generar un nuevo conjunto de datos sin sesgo y unirlo con el corpus original [28][29].

2.4.2 Mitigación del Sesgo en los *Word Embeddings*. En [6] se propone un método que neutraliza palabras neutrales en cuanto al género (las que se hayan escogido), eliminando la proyección de sus respectivos *word embeddings* en el subespacio de género, y transformando los *word embeddings* de los pares de palabras con un género por definición (los que se hayan escogido). De tal forma que estos pares sean equidistantes al subespacio de género. Este proceso se denomina *Hard Debiasing* o Mitigación del Sesgo Fuerte. Además, se propone otro método menos agresivo que busca preservar el producto interno de los *word embeddings* de todas las palabras mientras se minimiza la componente de género proyectando las palabras fuera de la dirección de género. Se busca mantener la similitud con los *word embeddings* originales tanto como sea posible. Este método se denomina *Soft Debiasing* o Mitigación del Sesgo Suave. El problema de estos métodos es que, considerando dos pares de términos, *grandmather*, *grandfather* y *guy*, *gal*, la palabra neutral en cuanto al género *babysit* será equidistante respecto a las palabras de cada par, pero en cuanto a la frase *Grandfather a regulation* (significa cláusula de antigüedad), se convierte en igual de probable que *Grandmother a regulation* (significa regulación de la abuela) [8].

En el método GN-GloVe (*Gender Neutral variant of GloVe*) los autores entrenan *word embeddings* aislando la información de género en dimensiones específicas y manteniendo la información de género neutral en las otras dimensiones, pudiéndose eliminar las dimensiones de género tras el entrenamiento [30].

En [31], se muestra que la proyección lineal de todos los *word embeddings* fuera de la dirección de género, es decir, la aplicación del paso de neutralización

del método Mitigación del Sesgo Fuerte en todas las incrustaciones de palabras, generan mejores resultados en cuanto a la mitigación del sesgo de género. Este método mejorado recibe el nombre de *Linear Projection* o Proyección Lineal.

Otro método de mitigación de sesgo denominado *Attract-Repel* o Atraer-Repeler utiliza dos conjuntos de palabras. Uno formado por pares de antónimos y otro por pares de sinónimos. Lo que busca este método es acercar los pares de sinónimos entre sí y repeler los pares de antónimos entre sí [32].

En [33], se demuestra que la frecuencia de palabras en el corpus interfiere con los componentes de las incrustaciones de palabras asociados con el género, y propone el método *Double-Hard Debiasing* o Doble Mitigación del Sesgo Fuerte, el cual elimina la dirección de frecuencia de todos los *word embeddings*. Después utiliza las palabras más sesgadas, en cuanto al género, para aplicar el método de Mitigación del Sesgo Fuerte, eliminando así el sesgo de género de una manera más efectiva.

En [18], descubrieron que los métodos Mitigación del Sesgo Fuerte y Mitigación del Sesgo Suave (métodos de mitigación de sesgo de posprocesamiento) y GN-GloVe (método de mitigación de sesgo en el proceso de entrenamiento), en realidad esconden el sesgo más que eliminarlo. Acaban creando agrupaciones de palabras con sesgos masculinos y palabras con sesgos femeninos, alineándose estas agrupaciones con el género en un 85 % de las veces.

Otro método que ha sido publicado recientemente es la Proyección Iterativa en el Espacio Nulo (*Iterative Null-space Projection*, INLP) [19], el cual aprende una transformación que busca que la etiqueta de una categoría sensible de un conjunto de palabras no pueda ser predicha, y se realiza este proceso de forma iterativa.

2.4.3 Mitigación del Sesgo con Algoritmos de Ajuste. En [34] se propone un algoritmo de clasificación que busca conseguir justicia de grupo, es decir, que la proporción de los miembros de un grupo protegido (género, raza, etc.) reciba la misma clasificación positiva que la proporción de la población en su conjunto; y busca conseguir justicia individual, es decir, que individuos similares sean tratados igual.

También se utiliza el entrenamiento de adversario, en el que se aprenden representaciones invariantes hacia los atributos protegidos hasta que el adversario no sea capaz de predecir los atributos protegidos [35].

Los problemas que presentan estos dos métodos anteriores es que la información demográfica puede ser recuperada incluso después del entrenamiento adversario, obteniendo una alta precisión de clasificación, incluso si se utilizan los mismos datos con los que se entrenaron [36].

En [37] se propone un Reductor de la Amplificación del Sesgo (*Reducing Bias Amplification*, RBA), siendo un método de mitigación de sesgo para la calibración de las predicciones mediante un modelo de predicción estructurado, el cual se basa en un modelo condicional restrictivo [38]. La intuición detrás de este algoritmo es inyectar restricciones en el corpus, para asegurarse que las predicciones del modelo siguen la distribución observada en el conjunto de datos.

3. Metodología

Se describirán detalladamente los tipos de *word embeddings*, los métodos para la identificación, medición y mitigación del sesgo en las incrustaciones de palabras y se definirán los casos de estudio que se van a realizar.

3.1 *Word Embeddings*

Para poder convertir un texto a datos numéricos y que puedan ser utilizados por los sistemas del aprendizaje automático surgieron distintos modelos de *word embeddings*.

3.2 Word2Vec.

Es un algoritmo de aprendizaje autosupervisado en el que palabras similares tienen *word embeddings* similares [39]. Existen dos algoritmos principales, *Skip-Gram* y CBOW (*Continuous Bag of Words*), ambos algoritmos utilizan redes neuronales con una sola capa oculta. Para entrenar a los *word embeddings* con estos algoritmos, se requiere de una palabra principal extraída de un corpus o de algún conjunto de datos, y se utilizan las palabras cercanas o de contexto a esta palabra principal. El número de palabras de contexto está definido por una ventana. Los *word embeddings* entrenados con el algoritmo *Skip-Gram* intentan predecir las palabras de contexto utilizando la palabra principal como entrada. En cambio, los *word embeddings* entrenados con el algoritmo CBOW intentan predecir la palabra principal utilizando las palabras de contexto como entrada.

3.3 GloVe

GloVe (*Global Vectors for Word Representations*) [40], es un algoritmo de aprendizaje no supervisado que se utiliza para obtener vectores de representación de las palabras. Su entrenamiento se basa en las estadísticas globales de las coocurrencias palabra a palabra de un corpus, mostrando las representaciones resultantes como subestructuras lineales interesantes del espacio vectorial de las palabras.

3.4 Identificación de la Dirección de Género en los *Word Embeddings*

Se van a explicar los diferentes métodos más relevantes en la identificación de la dirección de género en los *word embeddings* que se utilizan en los métodos de medición y mitigación de sesgo del sistema de auditoría. Los métodos que se van a exponer se puede extender a cualquier otro tipo de sesgo.

3.4.1 PCA por Pares. En [6] se utiliza PCA (Principal Component Analysis) para desarrollar el método *PCA Pairs* o PCA por Pares con el fin de identificar el subespacio o la dirección de género. A partir de pares de palabras de género que tienen un género por definición (*man, woman; she, he...*) identifican el subespacio de género. El PCA se calcula de la diferencia entre los vectores de los pares de género.

3.4.2 Dos Medias. En [32] se calcula la media de todos los *word embeddings* de las palabras de género masculino por definición y la media de todos los *word embeddings* de las palabras de género femenino por definición, se restan y se normalizan.

3.4.3 Clasificación. [19] utiliza un clasificador lineal c , que a partir de un conjunto de *word embeddings* W , es entrenado para predecir los atributos protegidos $z_i \in Z$ (género, raza, ...), de cada *word embedding* $w_i \in W$. Los parámetros del clasificador c tras el entrenamiento, corresponderían al subespacio de género.

3.5 Medición del Sesgo de Género en los *Word Embeddings*

3.5.1 Sesgo Directo. Para calcular el Sesgo Directo o *Direct Bias* [6], primero se identifican las palabras neutrales en cuanto al género, denotadas como N , después se identifica la dirección de género, denotada como g , y sea c un parámetro para indicar cuan estricto sea la métrica, se define la ecuación (1), como la fórmula del sesgo directo.

$$\text{DirectBias}_c = \frac{1}{|N|} \sum_{w \in N} |\cos(\vec{w}, g)|^c \quad (1)$$

3.5.2 Indirect Bias. El *Indirect Bias* [6] hace referencia al sesgo que existe entre las palabras que deberían ser neutrales en cuanto al género. Sean dos *word embeddings* \vec{w} y \vec{v} , la dirección de género g , la proyección del vector \vec{w} en g como \vec{w}_g y la proyección del vector \vec{w} al vector ortogonal de la dirección de género definido como $\vec{w}_\perp = \vec{w} - \vec{w}_g$, el sesgo indirecto se calcula mediante la ecuación (2).

$$\beta(\vec{w}, \vec{v}) = \left(\vec{w} \cdot \vec{v} - \frac{\vec{w}_\perp \cdot \vec{v}_\perp}{\|\vec{w}_\perp\|_2 \|\vec{v}_\perp\|_2} \right) / \vec{w} \cdot \vec{v} \quad (2)$$

3.5.3 WEAT. El método *WEAT* [13] utiliza el *valor p* o *p-value* para demostrar el sesgo, el cual utiliza la prueba de permutación, midiendo la probabilidad de que una permutación aleatoria produzca una diferencia observable o mayor que las medidas de la muestra. Y utiliza el *tamaño de efecto* o *effect size* para cuantificar el sesgo, midiendo lo separado que se encuentran dos distribuciones de asociaciones entre dos conjuntos de palabras objetivo (p. ej., *programmer, engineer, ...* y *nurse, teacher, ...*) y dos conjuntos de palabras atributo atributo (p. ej., *man, male, ...* y *woman, female, ...*).

3.5.4 Métrica de Vecindad. En [18] se propone una métrica que se basa en la proporción de palabras con el mismo estereotipo de género (masculino o femenino), que se encuentran entre las palabras vecinas más cercanas a una palabra. Para ello se utiliza el algoritmo KMeans (número de agrupaciones = 2).

3.6 Mitigación del Sesgo de Género en los *Word Embeddings*

Se explicarán en detalle los métodos de medición del sesgo de género en los *word embeddings*, que se han mencionado en la Revisión de la Literatura y que se utilizan en el sistema de auditoría.

3.6.1 Mitigación del Sesgo Fuerte. En [6] se propone este método, en el que primero se neutralizan las palabras neutrales en cuanto al género por definición eliminando el componente de la dirección género, y después se igualan las distancias de los pares de palabras con género por definición (p. ej. *he* y *she*) en direcciones opuestas respecto al subespacio de género.

3.6.2 Mitigación del Sesgo Suave. En [6] se propone también este método, en el que se realiza una mitigación en cuanto al sesgo de género menos rigurosa que el método Mitigación del Sesgo Fuerte, ya que intenta mantener lo máximo posible la información importante de los *word embeddings*, mientras se elimina el sesgo de género mediante la resolución de un problema de optimización.

3.6.3 Atraer-Repeler. En [31] se propone un método que utiliza dos conjuntos con pares de palabras, uno de sinónimos y otro de antónimos, y se busca acercar las palabras sinónimas entre sí y alejar las palabras antónimas entre sí. El conjunto de pares de palabras sinónimas debe estar compuesto por una palabra de género y una palabra con estereotipos hacia el género opuesto, para acércalas entre sí. En cambio, el conjunto de pares palabras antónimas debe estar compuesto por una palabra de género y una palabra con estereotipos del mismo género, para alejarlas entre sí.

3.6.4 Proyección Lineal. En [32] se propone un método que demuestra que aplicar el paso de neutralizar del método Mitigación del Sesgo Fuerte sobre todos los *word embeddings* de todas las palabras del vocabulario, reduce mejor el sesgo que al igualar la distancia de los pares de palabras con un género por definición respecto a la dirección de género.

3.6.5 Doble Mitigación del Sesgo Fuerte. En [33] se demuestra que la frecuencia de las palabras en el corpus interfiere con los componentes de los *word embeddings* asociados con el género, por lo que se propone un método que primero elimina el sesgo de frecuencia o la dirección de frecuencia de todos los *word embeddings* y después aplica el método de Mitigación del Sesgo Fuerte eliminando la dirección de género de todos los *word embeddings*.

3.6.6 Proyección Iterativa en el Espacio Nulo. Los métodos anteriores se centraban en identificar un solo subespacio o dirección de género, pero en [19], se demuestra que el subespacio de género puede estar presente en docenas o cientos de direcciones ortogonales en el espacio latente, las cuales no son necesariamente

interpretables con la dirección *he – she*. El objetivo de este método es proteger la información sensible, eliminando su codificación de los *word embeddings*. Para ello, de forma iterativa se entrena un clasificador para predecir el género y se proyectan los *word embeddings* en el subespacio ortogonal a la matriz de parámetros del clasificador. La proyección ortogonal se corresponde con la proyección en el espacio nulo.

3.7 Definición de los Casos de Estudio

3.7.1 Conjunto de *Word Embeddings*. El conjunto de *word embeddings* que se utiliza en todos los casos de estudio se extrae de la biblioteca de Gensim [41], la cual utiliza los vectores preentrenados de una parte del conjunto de datos de las noticias de Google (Google News), el cual contenía sobre 100 mil millones de palabras. Este modelo está formado por, aproximadamente, 3 millones de vectores de 300 dimensiones.

3.7.2 Conjuntos de Palabras.

1. Pares de palabras de género por definición (masculinas-femeninas): {*man, woman; boy, girl; he, she; father, mother; son, daughter; guy, gal; male, female; his, her; himself, herself; john, mary*}.
2. Palabras de género femenino por definición: {*woman, girl, she, mother, daughter, gal, female, her, herself, mary*}.
3. Palabras de género masculino por definición: {*man, boy, he, father, son, guy, male, his, himself, john*}.
4. Pares de nombres (masculinos-femeninos): {*john, mary; william, victoria; george, carolina; liam, maria; andrew, anne; michael, kelly; louis, marie; tony, anna; scott, sarah; jackson, jane*}.
5. Nombres femeninos: {*mary, victoria, carolina, maria, anne, kelly, marie, anna, sarah, jane*}.
6. Nombres masculinos: {*john, william, george, liam, andrew, michael, louis, tony, scott, jackson*}.
7. Palabras neutrales en cuanto al género por definición: {*caretaker, dancer, homemaker, librarian, nurse, secretary, hairdresser, housekeeper, warrior, president, boxer, bodyguard, officer, mathematician, astronaut, coach*}.
8. Palabras con estereotipos femeninos: {*caretaker, dancer, homemaker, librarian, nurse, secretary, hairdresser, housekeeper*}.
9. Palabras con estereotipos masculinos: {*warrior, president, boxer, bodyguard, officer, mathematician, astronaut, coach*}.

3.7.3 Caso de Estudio 1: Comparación PCA por Pares y Dos Medias.

Se realizarán dos comparaciones:

1. En PCA por Pares se utilizarán los pares de palabras de género por definición, mientras que en Dos Medias se utilizarán las palabras de género femenino por definición y las palabras de género masculino por definición.

2. También se hará otra comparación, utilizando en PCA por Pares los pares de nombres, mientras que en Dos Medias se utilizarán los nombres femeninos y los nombres masculinos.

3.7.4 Caso de Estudio 2: Mitigación del Sesgo Fuerte.

1. Los pares de palabras de género por definición, las palabras neutrales en cuanto al género por definición.
2. El método de identificación del subespacio de género PCA por Pares, el cual utilizará los pares de palabras de género por definición.

3.7.5 Caso de Estudio 3: Mitigación del Sesgo Suave.

1. Las palabras de género femenino por definición, las palabras de género masculino por definición y las palabras neutrales en cuanto al género por definición.
2. El método de identificación del subespacio de género PCA por Pares, el cual utilizará los pares de palabras de género por definición.
3. Parámetros de entrada: $\lambda=0.2$, épocas=100, learning rate=0.001, momentum=0.0.

3.7.6 Caso de Estudio 4: Proyección Lineal.

1. Las palabras neutrales en cuanto al género por definición.
2. El método de identificación del subespacio de género PCA por Pares, el cual utilizará los pares de palabras de género por definición.

3.7.7 Caso de Estudio 5: Doble Mitigación del Sesgo Fuerte.

1. Las palabras con estereotipos femeninos, las palabras con estereotipos masculinos y las palabras neutrales en cuanto al género por definición.
2. El método de identificación del subespacio de género PCA por Pares, el cual utilizará los pares de palabras de género por definición.

3.7.8 Caso de Estudio 6: Atraer-Repeler.

1. Las palabras de género femenino por definición, las palabras de género masculino por definición, las palabras con estereotipos femeninos, y las palabras con estereotipos masculinos.
2. Iteraciones=5, tamaño del lote=10, margen de atracción=0.6, margen de repulsión=0.0 y constante de regulación $L2=0,000000001$.

3.7.9 Caso de Estudio 7: Proyección Iterativa en el Espacio Nulo.

1. Las palabras con estereotipos femeninos, las palabras con estereotipos masculinos y los pares de palabras de género por definición.
2. Parámetros de entrada: iteraciones=10.

3.7.10 Caso de Estudio 8: Medición del Sesgo Antes y Después de la Mitigación. En este caso de estudio se aplicarán los siguientes métodos de medición del sesgo en el conjunto de *word embeddings* inicial y en los conjuntos de *word embeddings* mitigados de los Casos de Estudio 2, 3, 4, 5, 6 y 7:

1. **Sesgo Directo:** primera palabra *softball*, segunda palabra *receptionist* y el método de identificación del subespacio de género PCA por Pares, el cual utilizará los pares de palabras de género por definición.
2. **Sesgo Indirecto:** los pares de palabras neutrales en cuanto al género por definición, $c = 1$ y el método de identificación del subespacio de género PCA por Pares, el cual utilizará los pares de palabras de género por definición.
3. **WEAT:** palabras objetivo X las palabras con estereotipos femeninos; palabras objetivo Y las palabras con estereotipos masculinos; palabras atributo A las palabras de género femenino por definición; palabras atributo B las palabras de género masculino por definición.
4. **Métrica de Vecindad:** las palabras con estereotipos femeninos y las palabras con estereotipos masculinos.

4. Sistema de Auditoría Propuesto

Se presenta el sistema de auditoría, exponiendo su arquitectura y sus distintas funcionalidades. Además, se desarrollan los casos de estudio, en los que se explica, de forma simple, el funcionamiento de los métodos de identificación, mitigación y medición del sesgo implementados en el sistema de auditoría. Para acceder al código fuente original que se ha desarrollado ir a [42]. En la Figura 2 se muestra la página web del sistema de auditoría.

The screenshot shows the ASWEB interface. The 'Debiasing Options' section includes dropdowns for 'Selection of Word Embedding' (test: w=446, s=300), 'Debiasing Methods' (Hard Debiasing), and 'Gender Direction Methods' (PCA Pairs). Below these are lists for 'Gender Specific Words' and 'Gender Neutral Words'. The 'Measurements' section contains a table with the following data:

Method	Word Embedding 1	Word Embedding 2
Indirect Bias	test: (softball/receptionist)+ 0.655922340581801	glove: (softball/receptionist)- -0.026753703132271767
Direct Bias	test: 0.06952262330742967	glove: 0.1895822265472466
WEAT	test: effect size=1.5817583798562183, p-value=0	glove: effect size=1.2296223640441895, p-value=0
Neighborhood Metric	test: 0.425	glove: 0.8125

Figura 2: ASWEB: *Auditing System for Word Embeddings Bias*.

4.1 Arquitectura del Sistema de Auditoría

En la Figura 3 se muestra la arquitectura del sistema de auditoría. El *frontend* está formado por un archivo HTML [43], un archivo CSS [44] y un archivo de JavaScript [45], este último utiliza las peticiones HTTP del módulo Ajax [46] de la biblioteca jQuery [47] para comunicarse con el *backend*, que recibe las peticiones mediante la biblioteca Flask [48] de Python [49], las procesa y devuelve una respuesta HTTP al *frontend* para mantenerlo actualizado. Además, el archivo de JavaScript del *frontend* se encarga de la generación de gráficas mediante la biblioteca d3 [50] para la visualización del proceso que llevan a cabo los métodos de mitigación del sesgo, y se encarga de añadir entradas a una tabla para la visualización de las métricas calculadas con los métodos de medición del sesgo.

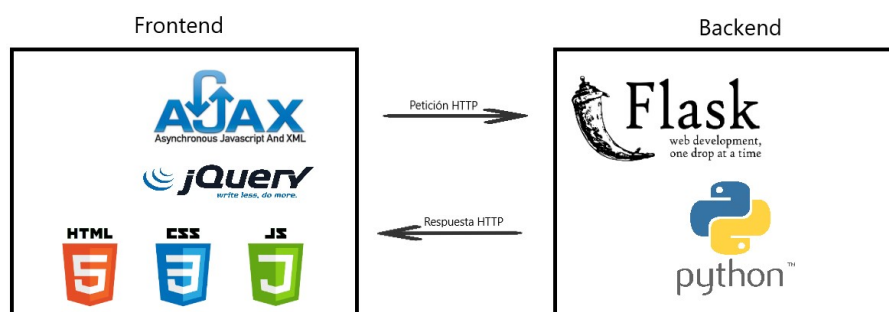


Figura 3: Arquitectura del Sistema de Auditoría.

La comunicación mediante el protocolo HTTP entre el *frontend* y el *backend* puede darse en varios casos. En la creación, subida o eliminación de un conjunto de *word embeddings* del sistema, si un usuario solicita la aplicación de cualquier método de mitigación del sesgo en un conjunto de *word embeddings* del sistema, si un usuario solicita la aplicación de cualquier método de medición del sesgo en un conjunto de *word embeddings* del sistema, o Si un usuario solicita el almacenamiento de un conjunto de *word embeddings* del sistema.

El *backend* está compuesto por varios archivos de Python: `files.py`, `word2vec.py`, `glove.py`, `word_embeddings.py`, `gender_direction_search.py`, `measurements.py`, `debias.py`, `visualization.py` y `main.py`. En `files.py` se trata todo lo relacionado con la carga, el almacenamiento y el preprocesamiento de los ficheros. En `word2vec.py` se almacenan y se entrenan conjuntos de *word embeddings* con el modelo Word2Vec. En `glove.py` se almacenan y se entrenan conjuntos de *word embeddings* con el modelo GloVe. En `word_embeddings.py` se almacena la información de los conjuntos de *word embeddings* de los que se desconoce su algoritmo de entrenamiento. En `gender_direction_search.py` se procesan los métodos de identificación de la dirección de género implementados en el sistema. En `measurements.py` se procesan los métodos de medición del sesgo implementados en el sistema. En `debias.py`

se procesan los métodos de mitigación del sesgo implementados en el sistema. En `visualization.py` se transforman los *word embeddings* de d dimensiones a 2 dimensiones para su visualización en el *frontend*. En `main.py` se lanza el servidor web y se recogen, procesan y responden todas las peticiones HTTP enviadas por el *frontend*.

4.2 Desarrollo de los Casos de Estudio

En los casos de estudio 2, 3, 4, 5, 6 y 7 se van a mostrar dos tipos de gráficas. En el primer caso, el eje x corresponde a la primera componente PCA y el eje y a la segunda componente PCA, haciendo referencia a las dos primeras componentes que se obtienen tras aplicar PCA a todos los *word embeddings* de las palabras mostradas en la gráfica, para que los *word embeddings* se puedan representar en 2 dimensiones. En el segundo caso, el eje x corresponde a la similitud coseno de los *word embeddings* de las palabras con la dirección de género o frecuencia y el eje y a la primera componente PCA. El segundo caso se utiliza para poder comprender el funcionamiento de los métodos de una forma más intuitiva y para mostrar cómo se organizan las palabras en torno a la dirección de género o frecuencia. En cuanto a la dirección de género o frecuencia también se convierte a 2 dimensiones mediante PCA, correspondiendo estas 2 dimensiones con el punto final del vector, y se le añade el punto inicial del vector en el origen de coordenadas para poder representarlo como un vector y poder observar su dirección.

4.2.1 Caso de Estudio 1: Comparación PCA por Pares y Dos Medias.

En [32] muestran que los métodos de identificación de la dirección de género PCA por Pares y Dos Medias funcionan de forma similar cuando se utilizan a los pares de palabras de género por definición que sean opuestos como *he* y *she*, pero que en los pares de nombres de géneros opuestos, por ejemplo *John* y *Mary*, como estos no son opuestos, la técnica de PCA por Pares no captura bien la dirección de género.

En la Figura 4 se muestra la visualización de los pares de palabras de género por definición (masculinas-femeninas) en PCA por Pares y en Dos Medias. Como se puede observar, no hay mucha diferencia en cuanto el vector de la dirección de género de PCA por Pares y Dos Medias utilizando pares de palabras de género por definición, de géneros opuestos, ya que los vectores son casi igual.

En la Figura 5 se muestra la visualización de los pares nombres (masculinos-femeninos) en PCA por Pares y en Dos Medias. En este caso sí se nota diferencia en cuanto el vector de la dirección de género de PCA por Pares y de Dos Medias utilizando pares nombres (masculinos-femeninos).

4.2.2 Caso de Estudio 2: Mitigación del Sesgo Fuerte.

El gráfico 1 de la Figura 6 muestra las palabras utilizadas como parámetros antes de realizar el proceso de mitigación. En la parte superior del gráfico se encuentran la mayoría de las palabras de género masculino por definición, mientras que en la parte

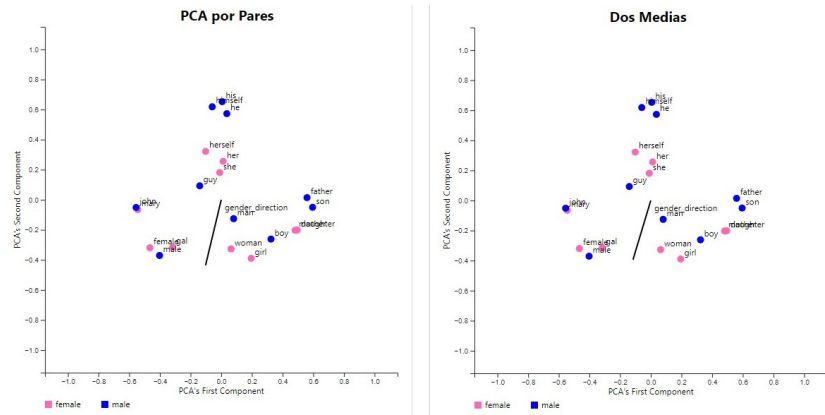


Figura 4: Caso de estudio 1: Comparación PCA por Pares y Dos Medias con pares de palabras de género por definición (masculinas-femeninas).

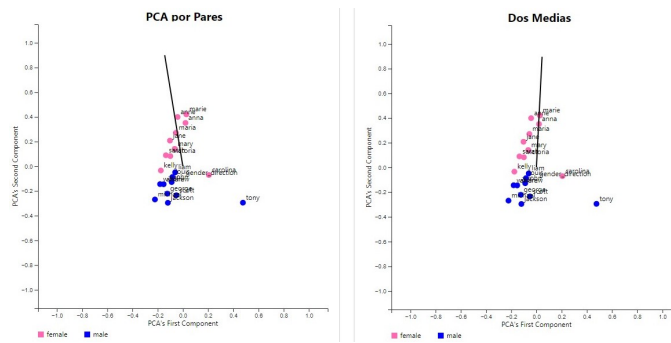


Figura 5: Caso de estudio 1: Comparación PCA por Pares y Dos Medias con pares de nombres (masculinos-femeninos).

inferior se encuentran las palabras de género femenino por definición, correspondiendo con la dirección del vector del subespacio de género.

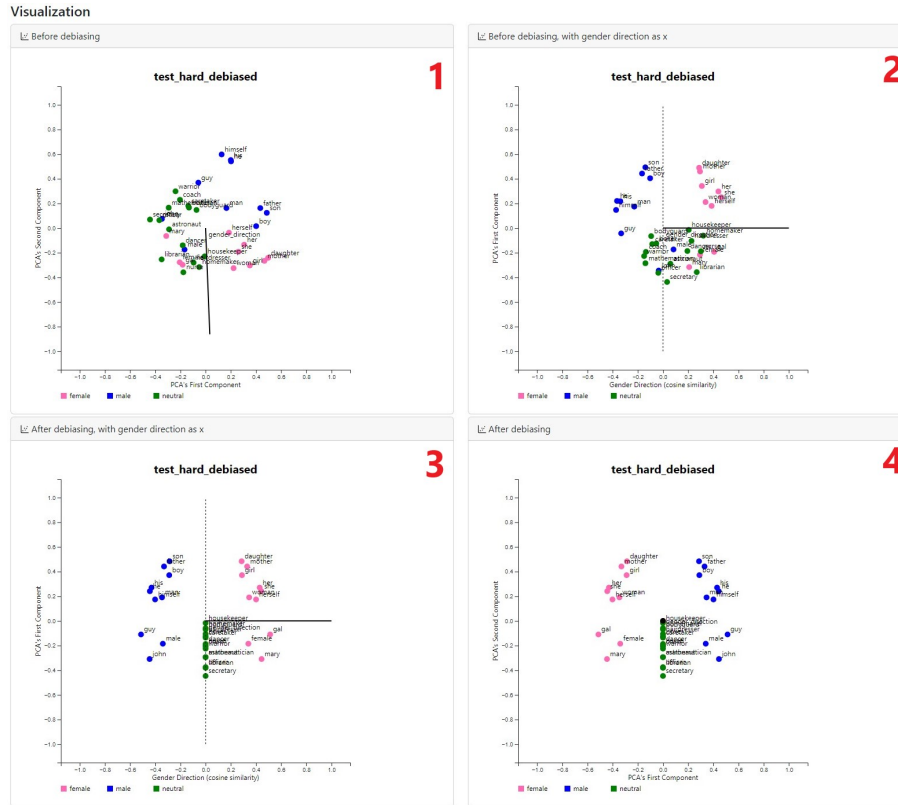


Figura 6: Visualización del Caso de Estudio 2.

El gráfico 2 muestra las palabras utilizadas como parámetros antes de realizar el proceso de mitigación, con el eje x como dirección de género. Como se puede observar, la mayoría de las palabras de género masculino por definición se encuentran en la parte izquierda de la gráfica o con una similitud coseno menor a 0 respecto al eje x , y la mayoría de las palabras de género femenino por definición se encuentran en la parte derecha de la gráfica o con una similitud coseno mayor a 0 respecto al eje x . Lo mismo ocurre con las palabras con estereotipos masculinos como *warrior*, *coach* o *mathematician* y con las palabras con estereotipos femeninos como *librarian*, *homemaker* o *nurse*.

El gráfico 3 muestra las palabras utilizadas como parámetros después de realizar el proceso de mitigación. En el método Mitigación del Sesgo Fuerte se neutralizan las palabras neutras en cuanto al género por definición proyectándolas en el espacio nulo o fuera de la dirección de género, por lo que estas palabras ya

no contienen la componente de género. En cuanto a las palabras con un género por definición se iguala la distancia de los pares en direcciones opuestas respecto a la dirección de género como se puede observar en la gráfica, ya que los pares de palabras con géneros opuestos se encuentran en la misma posición en el eje y y a la misma distancia del eje x pero cada palabra del par en una dirección opuesta.

El gráfico 4 muestra las palabras utilizadas como parámetros después de realizar el proceso de mitigación. Las palabras con estereotipos de género se encuentran a la misma distancia respecto a los pares de palabras de género por definición.

4.2.3 Caso de Estudio 3: Mitigación del Sesgo Suave. Las gráficas 1 y 2 de la Figura 7 coinciden con las mismas gráficas que en el Caso de Estudio 2, ya que se han utilizado las mismas palabras y el mismo método de identificación del subespacio de género con los mismos parámetros.

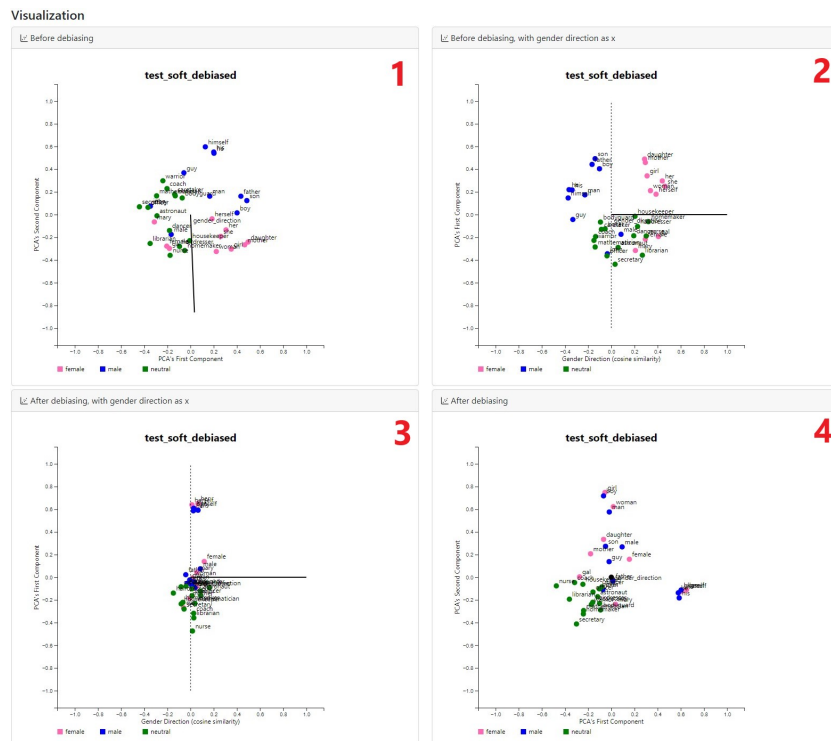


Figura 7: Visualización del Caso de Estudio 3.

La gráfica 3 muestra las palabras utilizadas como parámetros después de realizar el proceso de mitigación. En el método Mitigación del Sesgo Suave se

aplica una proyección de las palabras en el espacio nulo o fuera de la dirección de género menos rigurosa que en el método Mitigación del Sesgo Fuerte. Como se puede observar en la gráfica la similitud de todas las palabras con la dirección de género es muy cercana a 0, pero no llega a ser 0, ya que intenta preservar la información relevante de los *word embeddings* originales.

La gráfica 4 muestra las palabras utilizadas como parámetros después de realizar el proceso de mitigación. Si se compara la primera gráfica con esta, se puede observar que los pares de palabras de género por definición después de la mitigación se encuentran más cerca entre sí, por lo que las palabras neutrales en cuanto al género por definición se encuentran a una distancia más o menos similar respecto a ambas palabras de los pares de palabras de género por definición, ya que antes de la mitigación las palabras con estereotipos femeninos se encontraban más cerca de las palabras de género femenino por definición que de las palabras de género masculino por definición y viceversa. Ahora este sesgo se ha reducido o eliminado.

4.2.4 Caso de Estudio 4: Proyección Lineal. Las gráficas 1 y 2 de la Figura 8 coinciden con las mismas gráficas que en el Caso de Estudio 2 y que en el Caso de Estudio 3, ya que se han utilizado las mismas palabras y el mismo método de identificación del subespacio de género con los mismos parámetros.

La gráfica 3 muestra las palabras utilizadas como parámetros después de realizar el proceso de mitigación. El método Mitigación del Sesgo Fuerte solo neutraliza las palabras neutrales en cuanto al género por definición. En cambio, el método Proyección Lineal, el cual se utiliza en este caso de estudio, neutraliza todas las palabras del conjunto de *word embeddings* proyectándolas en el espacio nulo o fuera de la dirección de género. Como se puede observar en la gráfica, la similitud de estas palabras con la dirección de género es 0 y, además, se encuentran en la dirección ortogonal a la dirección de género, la cual corresponde al espacio nulo del subespacio de género, por lo que ya no contienen la componente de género.

La gráfica 4 muestra las palabras utilizadas como parámetros después de realizar el proceso de mitigación. En este caso la dirección de género ha sido eliminada completamente de todo el conjunto de *word embeddings*. Por lo que la dirección de género se corresponde al punto negro en el origen de coordenadas. Además, si se compara esta gráfica con la primera gráfica, se puede observar que los pares de palabras de género por definición después de la mitigación se encuentran más cerca entre sí, por lo que las palabras neutrales en cuanto al género por definición se encuentran a una distancia más o menos similar respecto a ambas palabras de los pares de palabras de género por definición, ya que antes de la mitigación las palabras con estereotipos femeninos se encontraban más cerca de las palabras de género femenino por definición que de las palabras de género masculino por definición y viceversa. Ahora este sesgo se ha reducido o eliminado.

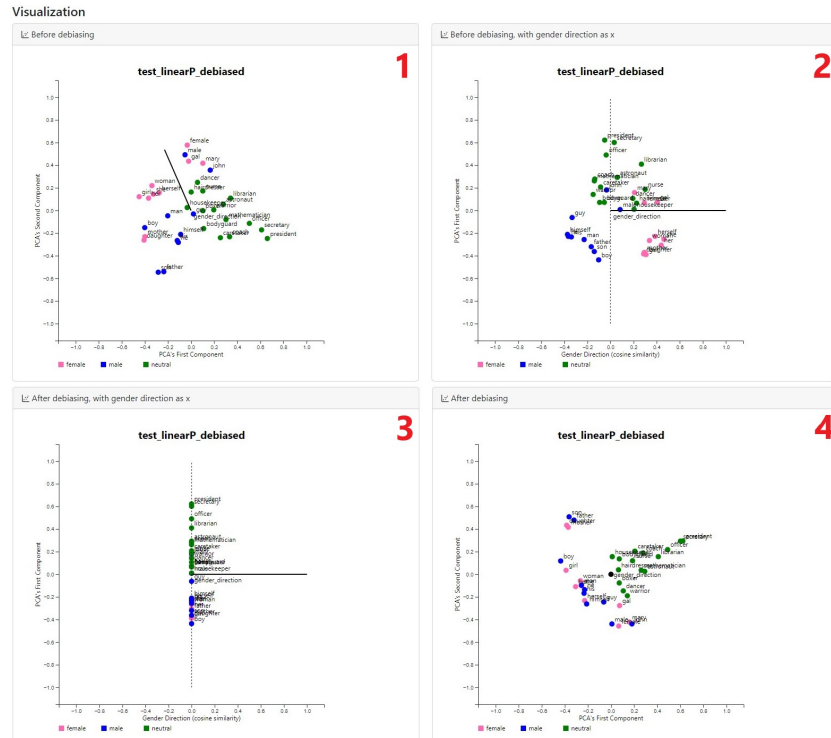


Figura 8: Visualización del Caso de Estudio 4.

4.2.5 Caso de Estudio 5: Doble Mitigación del Sesgo Fuerte. La gráfica 1 de la Figura 9 muestra las palabras utilizadas como parámetros antes de realizar el proceso de mitigación de la dirección de frecuencia.

La gráfica 2 muestra las palabras utilizadas como parámetros antes de realizar el proceso de mitigación de la dirección de frecuencia. Las palabras con mayor frecuencia son aquellas que tienen una mayor similitud con el subespacio de frecuencia, como es el caso de las palabras *father*, *mother*, *daughter* y *son*, lo cual indica que estas palabras son las que más se han repetido dentro del corpus con el que fueron entrenado los *word embeddings* de las palabras representadas en la gráfica. En cuanto a las palabras con menor frecuencia serían los nombres de personas *john* y *mary*, lo cual tiene sentido porque los nombres propios suelen aparecer con poca frecuencia en los textos.

La gráfica 3 muestra las palabras utilizadas como parámetros después de realizar el proceso de mitigación de la dirección de frecuencia. En el método Doble Mitigación del Sesgo Fuerte, primero se neutralizan todas las palabras del conjunto de *word embeddings* proyectándolas en el espacio nulo o fuera de la dirección de frecuencia. Como se puede observar en la gráfica, la similitud de estas palabras con la dirección de frecuencia es 0 y, además, se encuentran en la dirección ortogonal a la dirección de género, la cual corresponde al espacio

Visualization

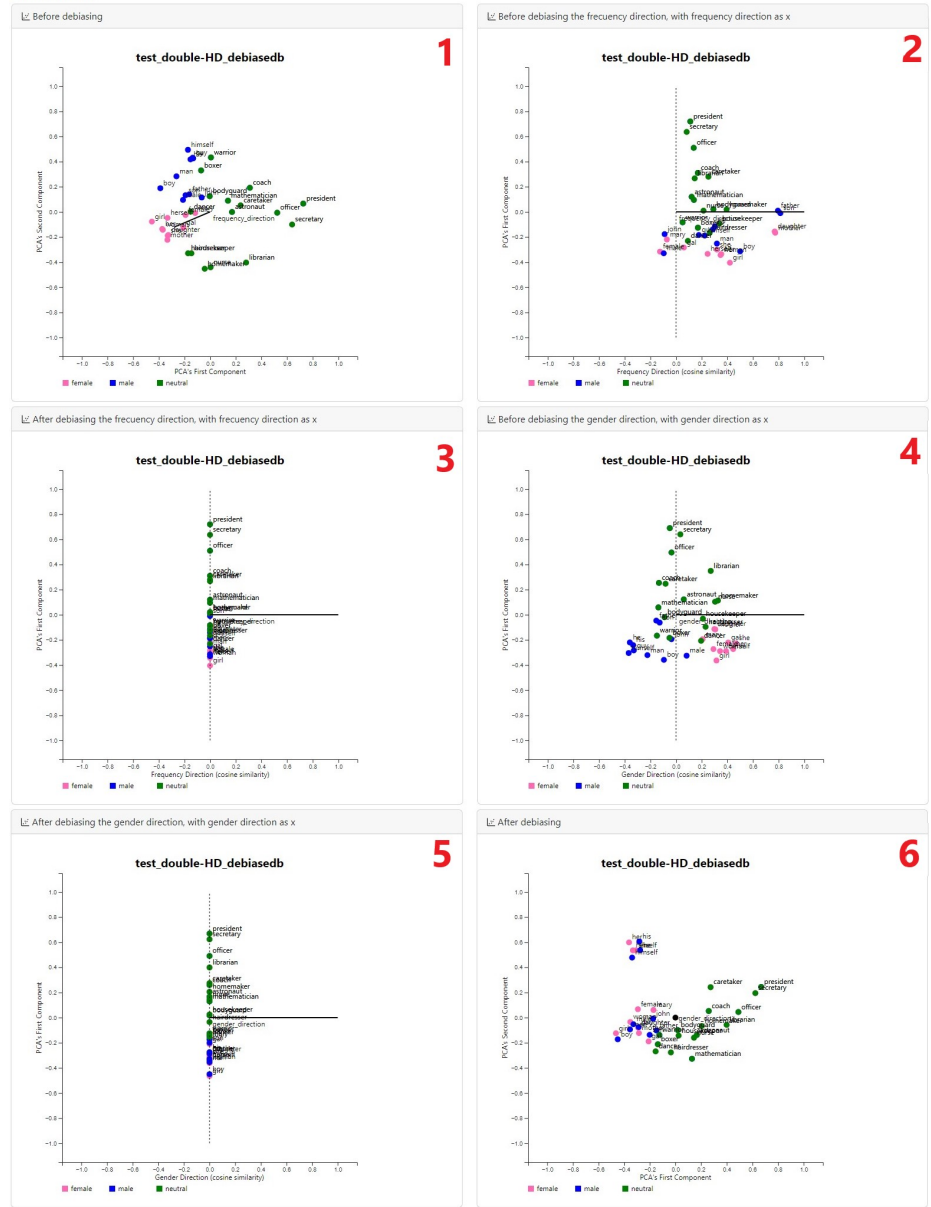


Figura 9: Visualización del Caso de Estudio 5.

nulo del subespacio de género, por lo que ya no contienen la componente de frecuencia.

La gráfica 4 muestra las palabras utilizadas como parámetros antes de realizar el proceso de mitigación de la dirección de género. La mayoría de las palabras de género masculino por definición se encuentran en la parte izquierda de la gráfica o con una similitud coseno menor a 0 respecto el eje x , y la mayoría de las palabras de género femenino por definición se encuentran en la parte derecha de la gráfica o con una similitud coseno mayor a 0 respecto el eje x . También se puede observar como las palabras con estereotipos masculinos como *officer*, *president* o *boxer* se encuentran en la parte izquierda de la gráfica como las palabras de género masculino por definición mientras que las palabras con estereotipos femeninos como *dancer*, *hairdresser* o *housekeeper* se encuentran en la parte de la derecha de la gráfica como las palabras de género femenino por definición.

La gráfica 5 muestra las palabras utilizadas como parámetros después de realizar el proceso de mitigación de la dirección de género. En esta gráfica se muestra como ahora se neutralizan todas las palabras del conjunto de *word embeddings* proyectándolas en el espacio nulo o fuera de la dirección de género. Como se puede observar en la gráfica, la similitud de estas palabras con la dirección de género es 0 y, además, se encuentran en la dirección ortogonal a la dirección de género, la cual corresponde al espacio nulo del subespacio de género, por lo que ya no contienen la componente de género.

La gráfica 6 muestra las palabras utilizadas como parámetros después de realizar el proceso de mitigación. En este caso la dirección de frecuencia y la dirección de género han sido eliminadas completamente de todo el conjunto de *word embeddings*, por lo que estas direcciones corresponden al punto negro en el origen de coordenadas. Además, si se compara esta gráfica con la primera gráfica, se puede observar que los pares de palabras de género por definición después de la mitigación se encuentran más cerca entre sí, por lo que las palabras neutrales en cuanto al género por definición se encuentran a una distancia más o menos similar respecto a ambas palabras de los pares de palabras de género por definición, ya que antes de la mitigación las palabras con estereotipos femeninos se encontraban más cerca de las palabras de género femenino por definición que de las palabras de género masculino por definición y viceversa. Ahora este sesgo se ha reducido o eliminado.

4.2.6 Caso de Estudio 6: Atraer-Repeler. La gráfica 1 de la Figura 10 muestra las palabras utilizadas como parámetros antes de realizar el proceso de mitigación. Las palabras con estereotipos masculinos están más cerca de las palabras de género masculino por definición y las palabras con estereotipos femeninos están más cerca de las palabras de género femenino por definición, como existen estas relaciones, son agrupados en parejas como palabras antónimas, porque se busca que se separen entre sí para que se pierdan los estereotipos. Como se puede observar, las palabras con estereotipos masculinos están más lejos de las palabras de género femenino por definición y las palabras con estereotipos femeninos están más cerca de las palabras de género masculino por definición,

siendo agrupadas por parejas como palabras sinónimas, porque se busca que se acerquen entre sí para que se pierdan los estereotipos.

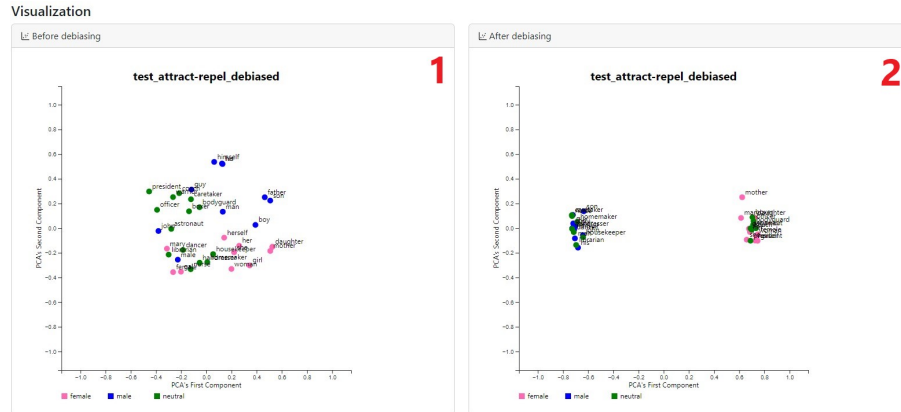


Figura 10: Visualización del Caso de Estudio 6.

La gráfica 2 muestra las palabras utilizadas como parámetros después de realizar el proceso de mitigación. Se puede observar que ahora las palabras con estereotipos femeninos se agrupan con las palabras de género masculino por definición, y que las palabras con estereotipos masculinos se agrupan con las palabras de género femenino. En este caso se han separado demasiado ambos grupos, lo que se busca es que las palabras con estereotipos estén en una posición intermedia entre las palabras de género femenino por definición y las palabras de género masculino por definición.

4.2.7 Caso de Estudio 7: Proyección Iterativa en el Espacio Nulo. La gráfica 1 de la Figura 11 muestra las palabras utilizadas como parámetros antes de realizar el proceso de mitigación. Las palabras con estereotipos masculinos están más cerca de las palabras de género masculino por definición y las palabras con estereotipos femeninos están más cerca de las palabras de género femenino por definición.

La gráfica 2 muestra las palabras utilizadas como parámetros después de realizar el proceso de mitigación. Las palabras con estereotipos se agrupan cerca del origen de coordenadas, y esto ocurre porque se han eliminado 10 subespacios de género ya que se ha ejecutado el método 10 veces, por lo que los *word embeddings* ahora tienen 10 dimensiones menos. Este método es el más agresivo comparado con los anteriores, ya que elimina tantas direcciones de género como número de iteraciones se hayan introducido.

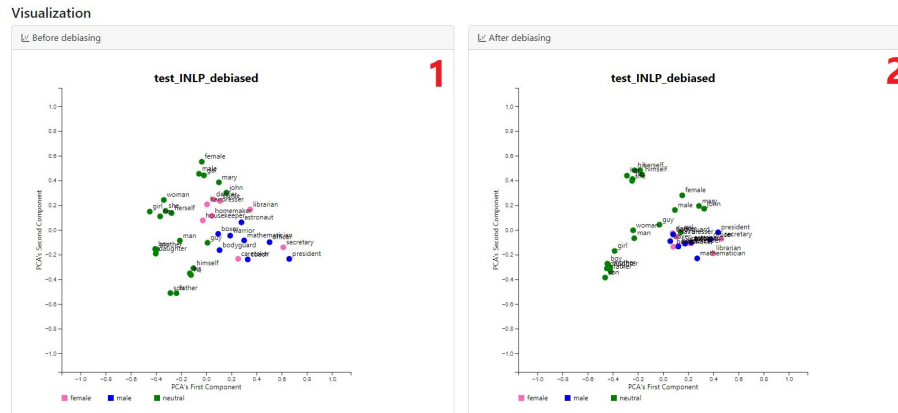


Figura 11: Visualización del Caso de Estudio 7.

4.2.8 Caso de Estudio 8: Medición del Sesgo Antes y Después de la Mitigación.

En la Tabla 1 se muestran los resultados de la medición, de los conjuntos de *word embeddings*.

El Sesgo Indirecto que existe en el par de palabras *softball/receptionist* en los conjuntos *test* y *test-hard-debiased* es igual, esto se puede deber a lo que se comentaba en [18], que después eliminar el sesgo del conjunto *test* con el método Mitigación del Sesgo Fuerte dando lugar al conjunto *test-hard-debiased*, las palabras con estereotipos de género siguen agrupándose entre sí por género en el conjunto mitigado *test-hard-debiased*, aunque estas palabras en el conjunto mitigado estén a la misma distancia de los pares de palabras de género por definición, manteniendo un sesgo de grupo. En el resto de conjuntos se puede observar que el Sesgo Indirecto del par de palabras *softball/receptionist* ha disminuido.

En cuanto al Sesgo Directo, en el Caso de Estudio 6, la aplicación del método de mitigación Atar-Repeler al conjunto *test*, dando como resultado el conjunto mitigado *test-attract-repel-debiased* que ha acabado agrupando las palabras de género masculino por definición con las palabras con estereotipos femeninos y las palabras de género femenino por definición, con las palabras con estereotipos masculinos de una forma mayor que la que había entre las palabras de género masculino por definición con las palabras con estereotipos masculinos y entre las palabras de género femenino por definición con las palabras con estereotipos femeninos. Por lo que el Sesgo Directo del conjunto mitigado *test-attract-repel-debiased* es mayor al resto. En el resto de conjuntos se puede observar que tras la mitigación del conjunto *test* el Sesgo Directo se ha reducido respecto a este.

Como se puede observar en la Tabla 1 las métricas WEAT de los conjuntos *test-attract-repel-debiased* y *test-INLP-debiased* no pueden ser tomados en cuenta porque el *valor p* es 1. El resto de conjuntos mitigados tienen un *tamaño de efecto* menor que el conjunto inicial *test*, teniendo un *tamaño de efecto* medio los conjuntos *test-hard-debiased* y *test-linearP-debiased*, indicando que el método Mitigación del Sesgo Fuerte y el método de mitigación Proyección Lineal son los

Tabla 1: Métricas del Caso de Estudio 8.

Conjunto de <i>word embeddings</i>	Sesgo Indirecto (softball / receptionist)	Sesgo Directo	WEAT	Métrica de Vecindad
test (conjunto inicial)	0.65592	0.135416	effect size=1.58, 0.625 p-value=0	
test-hard-debiased (Caso de Estudio 2)	0.65592	0.00333	effect size=0.52, 0.625 p-value=0	
test-soft-debiased (Caso de Estudio 3)	0.20722	0.0808	effect size=1.13, 0.625 p-value=0	
test-linearP-debiased (Caso de Estudio 4)	-0.0136	0.05075	effect size=0.47, 0.625 p-value=0	
test-double-HD-debiasedb (Caso de Estudio 5)	0.24193	0.06971	effect size=0.99, 0.5625 p-value=0	
test-attract-repel-debiased (Caso de Estudio 6)	-0.0703	0.63325	effect size=-1.8, 0.875 p-value=1	
test-INLP-debiased (Caso de Estudio 7)	0.09055	0.07858	effect size=0.0, 0.5 p-value=1	

que mejor funcionan según la métrica WEAT; y teniendo un *tamaño de efecto* grande los conjuntos *test-soft-debiased* y *test-double-HD-debiasedb*.

Los conjuntos mitigados *test-hard-debiased*, *test-soft-debiased* y *test-linearP-debiased* tienen la misma Métrica de Vecindad que el conjunto inicial *test*. Esto se puede deber a lo expuesto en [18], que al eliminar el sesgo del conjunto *test*, los métodos de mitigación se centran en igualar las distancias entre las palabras con estereotipos de género y las palabras con un género por definición, mientras que las palabras con estereotipos de género se agrupan entre sí por género. El conjunto mitigado *test-attract-repel-debiased* tiene un Métrica de Vecindad incluso mayor que la del conjunto inicial *test*, ya que como se comentaba en el Caso de Estudio 6, la aplicación del método de mitigación Attract-Repel al conjunto inicial *test* ha acabado agrupando las palabras con un género por definición con las palabras con estereotipos del género opuesto a las palabras con un género por definición, de mayor forma que la agrupación que había en el conjunto *test*. Los otros dos conjuntos tienen una menor Métrica de Vecindad que el conjunto inicial.

5. Discusión

Los métodos utilizados para la identificación de la dirección de género de los *word embeddings* implementados en el sistema de auditoría, son los métodos PCA por Pares y Dos Medias. Los cuales actúan de forma muy similar cuando se utilizan pares de palabras con un género por definición, pero en el caso de

utilizar nombres propios con un género, el método PCA por Pares funciona peor a la hora de identificar la dirección de género que el método Dos medias. Esto se ha demostrado en el Caso de Estudio 1.

Referente a los métodos de mitigación del sesgo, el método Mitigación del Sesgo Fuerte es el método inicial sobre el que se basan los demás métodos utilizados en este trabajo, ya que en este algoritmo se neutraliza o se elimina la dirección de género de los *word embeddings* de las palabras neutrales en cuanto al género, y se iguala la distancia de los *word embeddings* de los pares de palabras con un género por definición respecto a la dirección de género.

El método Mitigación del Sesgo Suave utiliza una versión menos rigurosa del método Mitigación del Sesgo Fuerte, ya que no elimina por completo la dirección de género de los *word embeddings*, sino que reduce la componente de género. Este método intenta mantener, lo máximo posible, la información relevante de los *word embeddings*.

El método de mitigación Proyección Lineal utiliza una parte del método Mitigación del Sesgo Fuerte, la fase de neutralización, pero en este caso se elimina la componente de género o el subespacio de género de todos los vectores del conjunto de *word embeddings*, independientemente del tipo de palabras que sean.

El método de mitigación Doble Mitigación del Sesgo Fuerte utiliza la idea de la neutralización utilizada en Mitigación del Sesgo Fuerte, pero en este caso la utiliza dos veces, ya que primero elimina la componente de frecuencia o el subespacio de frecuencia de todos los *word embeddings* de todas las palabras, y después hace lo mismo con la componente de género o el subespacio de género de todos los *word embeddings* de todas las palabras.

El método de mitigación Proyección Iterativa en el Espacio Nulo utiliza la idea de la eliminación de la componente de género o del subespacio de género utilizado en el método Mitigación del Sesgo Fuerte, la cual corresponde a la proyección ortogonal o a la proyección en el espacio nulo del subespacio de género. En este método se realiza la proyección de los *word embeddings* en el espacio nulo del subespacio de género de forma iterativa.

En cuanto al método de mitigación Atraer-Repeler es el único método implementado que no se basa en el método Mitigación del Sesgo Fuerte como tal, sino en el artículo [18]. El cual mostraba que el método Mitigación del Sesgo Fuerte no elimina el sesgo sino que lo enmascara, ya que, tras aplicarlo, las palabras con estereotipos femeninos se siguen agrupando juntas, y lo mismo ocurre con las palabras con estereotipos masculinos. Por lo que el método de mitigación Atraer-Repeler busca agrupar las palabras con estereotipos femeninos junto con las palabras de género masculino por definición y las palabras con estereotipos masculinos junto con las palabras de género femenino por definición. Mientras aleja las palabras con estereotipos femeninos de las palabras de género femenino por definición y las palabras con estereotipos masculinos de las palabras de género masculino por definición. De esta forma consigue eliminar o reducir los estereotipos o la componente de género.

Las métricas obtenidas en el Caso de Estudio 8 mediante los métodos de medición del sesgo en los *word embeddings* implementados en el sistema de auditoría muestran que tras aplicar los métodos de mitigación del sesgo a un conjunto de

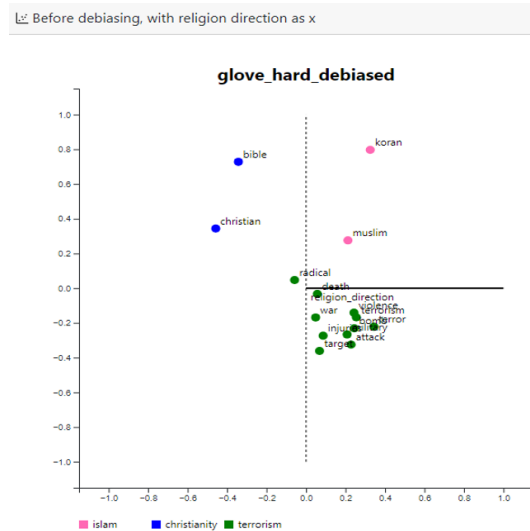


Figura 12: Visualización de antes de la mitigación de sesgo con el método Mitigación del Sesgo Fuerte, con la dirección de religión identificada con el método Dos Medias representada como eje x .

word embeddings, los conjuntos resultantes contienen menos sesgos que el conjunto inicial, salvo con el método Atraer-Repeler, pero esto se debe a que en el Caso de Estudio 6 donde se aplicó éste método, las palabras de género y las palabras con estereotipos se habían agrupado aún más que en el conjunto de *word embeddings* inicial debido a los hiperparámetros utilizados.

Además, el sistema de auditoría desarrollado permite comprender, de una forma más sencilla, algunos de los algoritmos más relevantes en cuanto a la identificación, medición y mitigación del sesgo en los *word embeddings*. Permitiendo visualizar el antes y el después de la mitigación de sesgo de los *word embeddings*, así como comparar la efectividad de los métodos empleados. Este sistema de auditoría no solo permite visualizar la mitigación del sesgo de género en los *word embeddings*, sino que puede ser extendido hacia otros tipos de sesgo, como la raza, religión, etc. En la Figura 12 se muestran los *word embeddings* de las palabras relacionadas con el terrorismo (*radical, death, war, violence, terrorism, attack, target, injuries, bomb, terror, military*), de la religión islámica (*koran, muslim*) y de la religión cristiana (*bible, christian*). Donde se identifica la dirección de religión con el método Dos Medias, cuya similitud coseno con los *word embeddings* de las palabras de la gráfica, corresponde al eje x , y el eje y a la primera componente PCA de todas las palabras de la gráfica. La imagen muestra los datos de antes de aplicar el método Mitigación del Sesgo Fuerte, pudiéndose observar como los *word embeddings* de las palabras relacionadas con el terrorismo están más cerca del islam que del cristianismo.

6. Conclusiones

En este trabajo se ha realizado una revisión general de la literatura existente sobre los sesgos y el daño que pueden causar, o que han causado, en el aprendizaje automático. Después se ha ido centrando en la literatura referente a la identificación, medición y mitigación del sesgo en el PLN, enfocándose sobre todo en revisar exhaustivamente la literatura más relevante relativa al sesgo en los *word embeddings*.

Posteriormente se ha desarrollado un sistema de auditoría del sesgo que permite su verificación mediante la creación de conjuntos de *word embeddings* de tipo Word2Vec o GloVe dado un corpus. Así como, la carga al sistema de un fichero que contenga un conjunto de *word embeddings* ya preentrenados. Estos conjuntos cargados, creados o generados en el sistema de auditoría, pueden ser eliminados del sistema o almacenados en un fichero de texto plano.

Además, el sistema de auditoría permite aplicar distintos métodos de mitigación de sesgo. Siendo estos: *Hard Debiasing* o Mitigación del Sesgo Fuerte, *Soft Debiasing* o Mitigación del Sesgo Suave, *Attract-Repel* o Atraer-Repeler, *Linear Projection* o Proyección Lineal, *Double-Hard Debiasing* o Doble Mitigación del Sesgo Fuerte, e *Iterative Nullspace Projection* o Proyección Iterativa en el Espacio Nulo. En el sistema de auditoría se puede elegir, en algunos de estos métodos de mitigación, el método de identificación de la dirección de género que se quiera utilizar. Siendo estos: PCA por Pares, Dos Medias y Clasificación. El proceso que se lleva a cabo en cada uno de los métodos de mitigación es representado mediante gráficas, para que los usuarios puedan comprender su funcionamiento de una forma más sencilla.

También se pueden aplicar distintos métodos de medición del sesgo, siendo estos *Indirect Bias* o Sesgo Indirecto, *Direct Bias* o Sesgo Directo, WEAT y *Neighborhood Metric* o Métrica de Vecindad. Los resultados de las mediciones se mostrarán en una tabla en el sistema de auditoría y se pueden realizar en dos conjuntos de *word embeddings* simultáneamente para poder compararlos de forma más sencilla.

Se han descrito y desarrollado casos de estudio sencillos que explican los distintos métodos implementados en el sistema de auditoría, respecto al proceso que siguen para mitigar el sesgo de género en los *word embeddings*. Comparando los métodos de identificación y mitigación del sesgo de género en las incrustaciones de palabras. Además, se han analizado las diferencias entre los métodos de mitigación del sesgo de género en los *word embeddings* implementados en el sistema.

En este trabajo se ha llegado a la conclusión a través de la revisión de la literatura relacionada con los sesgos en los *word embeddings*, que existe poco consenso en cuanto a los métodos de identificación, medición y mitigación del sesgo en los *word embeddings*. Por lo que aún queda mucho por investigar en relación a este tema, ya que cada cierto tiempo aparece una nueva investigación que demuestra que los métodos que supuestamente funcionaban no son tan efectivos como parecían.

6.1 Líneas Futuras

En cuanto a algunas de las líneas futuras que se consideran para este trabajo, la primera propuesta, la cual es muy recomendable, es la investigación e implementación de los *contextual embeddings*. Ya que mientras que los *word embeddings* son representaciones estáticas de las palabras, los *contextual embeddings* son representaciones de una palabra según su contexto. Esto evita que las palabras polisémicas sean representadas de la misma forma, suponiendo una mejor representación de las palabras. Debido a esto, los *contextual embeddings* están siendo cada vez más utilizados frente a los *word embeddings*.

Otras propuestas más relacionadas con la implementación: el almacenamiento de los conjuntos de *word embeddings* en una ruta especificada, la posibilidad de eliminar elementos de la tabla de mediciones y poder colocarlos manualmente con el ratón, la implementación de más métodos de identificación, medición y mitigación de sesgo de los *word embeddings* y que se puedan modificar los nombres de los conjuntos de *word embeddings* del sistema de auditoría.

Por último, sería muy recomendable implementar que las etiquetas de los gráficos se muestren al hacer clic en los puntos correspondientes, ya que cuando hay muchas cerca unas de otras no se pueden visualizar con claridad.

Referencias

1. Kiritchenko, S., & Mohammad, S. M. (2018). Examining gender and race bias in two hundred sentiment analysis systems. arXiv preprint arXiv:1805.04508.
2. Word embedding. (24 de abril de 2023). En Wikipedia. https://es.wikipedia.org/w/index.php?title=Word_embedding&oldid=150760075.
3. Barocas, S., Hardt, M., & Narayanan, A. (2019). Fairness and Machine Learning. Obtenido de: <http://www.fairmlbook.org>.
4. Costa-jussà, M.R. (2019). An analysis of gender bias studies in natural language processing. Nature Machine Intelligence, 1-2.
5. Peixeiro, M. (2019). Introduction to Natural Language Processing (NLP) and Bias in AI. towards data science. <https://towardsdatascience.com/>.
6. Bolukbasi, T., Chang, K. W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. Advances in neural information processing systems, 29, 4349-4357.
7. Sun, T., Gaut, A., Tang, S., Huang, Y., ElSherief, M., Zhao, J., Mirza, D., Belding-Royer, E.M., Chang, K., & Wang, W.Y. (2019). Mitigating Gender Bias in Natural Language Processing: Literature Review. ACL.
8. Kaushik, D. [graham Neubig] (22 de abril de 2021). CMU Neural Nets para NLP 2021 (22): Sesgo en NLP [Archivo de Vídeo]. Youtube. <https://www.youtube.com/watch?v=f01EKESnDv4>.
9. Subramonian, A. [uclanlp] (11 de julio de 2022). COM SCI 263: Bias in Natural Language Processing [Archivo de Vídeo]. Youtube. <https://www.youtube.com/watch?v=okU4meBeATo>.
10. Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

11. Crawford, K. (2017). The Trouble With Bias. Keynote at Neural Information Processing Systems (NIPS'17).
12. Blodgett, S.L., Barocas, S., Daumé H., & Wallach, H.M. (2020). Language (Technology) is Power: A Critical Survey of “Bias” in NLP. ACL.
13. Caliskan, A., Bryson, J.J., & Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356, 183 - 186.
14. May, C., Wang, A., Bordia, S., Bowman, S. R., & Rudinger, R. (2019). On measuring social biases in sentence encoders. arXiv preprint arXiv:1903.10561.
15. Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations.
16. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
17. Manzini, T., Lim, Y.C., Tsvetkov, Y., & Black, A.W. (2019). Black is to Criminal as Caucasian is to Police: Detecting and Removing Multiclass Bias in Word Embeddings. NAACL.
18. Gonen, H., & Goldberg, Y. (2019). Lipstick on a Pig: Debiasing Methods Cover up Systematic Gender Biases in Word Embeddings But do not Remove Them. NAACL-HLT.
19. Ravfogel, S., Elazar, Y., Gonen, H., Twiton, M., & Goldberg, Y. (2020). Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection. ACL.
20. Dixon, L., Li, J., Sorensen, J., Thain, N., & Vasserman, L. (2018). Measuring and Mitigating Unintended Bias in Text Classification.
21. Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K. W. (2018). Gender bias in coreference resolution: Evaluation and debiasing methods. arXiv preprint arXiv:1804.06876.
22. Rudinger, R., Naradowsky, J., Leonard, B., & Durme, B.V. (2018). Gender Bias in Coreference Resolution. NAACL.
23. Koenecke, A., Nam, A., Lake, E., Nudell, J., Quartey, M., Mengesha, Z., Toups, C., Rickford, J., Jurafsky, D., & Goel, S. (2020). Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences*, 117, 201915768.
24. Blodgett, S., Lopez, G., Olteanu, A., Sim, R., & Wallach, H. (2021). Stereotyping Norwegian Salmon: An Inventory of Pitfalls in Fairness Benchmark Datasets. 1004–1015. doi:10.18653/v1/2021.acl-long.81.
25. Lu, K., Mardziel, P., Wu, F., Amancharla, P., & Datta, A. (2020). Gender Bias in Neural Natural Language Processing. ArXiv, abs/1807.11714.
26. Vanmassenhove, E., Hardmeier, C., & Way, A. (2018). Getting Gender Right in Neural Machine Translation. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3003–3008. <https://doi.org/10.18653/v1/D18-1334>.
27. Park, J.H., Shin, J., & Fung, P. (2018). Reducing Gender Bias in Abusive Language Detection. EMNLP.
28. Kaushik, D., Hovy, E.H., & Lipton, Z.C. (2020). Learning the Difference that Makes a Difference with Counterfactually-Augmented Data. ArXiv, abs/1909.12434.
29. Kaushik, D., Kiela, D., Lipton, Z.C., & Yih, W. (2021). On the Efficacy of Adversarial Data Collection for Question Answering: Results from a Large-Scale Randomized Study. ACL/IJCNLP.
30. Zhao, J., Zhou, Y., Li, Z., Wang, W., & Chang, K. (2018). Learning Gender-Neutral Word Embeddings. EMNLP.
31. Mrkšić, N., Vulić, I., Séaghdha, D., Leviant, I., Reichart, R., Gašić, M., Korhonen, A., & Young, S. (2017). Semantic Specialization of Distributional Word Vector

- Spaces using Monolingual and Cross-Lingual Constraints. *Transactions of the Association for Computational Linguistics*, 5, 309–324. https://doi.org/10.1162/tacl_a_00063.
32. Dev, S., & Phillips, J. M. (2019). Attenuating Bias in Word Vectors. CoRR, abs/1901.07656. Recuperado de <http://arxiv.org/abs/1901.07656>.
 33. Wang, T., Lin, X. V., Rajani, N. F., McCann, B., Ordonez, V., & Xiong, C. (2020). Double-Hard Debias: Tailoring Word Embeddings for Gender Bias Mitigation. *Association for Computational Linguistics (ACL)*.
 34. Zemel, Richard., Wu, Y., Swersky, K., Pitassi, T., & Dwork, C. (2013). Learning fair representations. *30th International Conference on Machine Learning, ICML 2013*. 1362-1370.
 35. Ganin, Y., & Lempitsky, V.S. (2015). Unsupervised Domain Adaptation by Back-propagation. ArXiv, abs/1409.7495.P.
 36. Elazar, Y., & Goldberg, Y. (2018). Adversarial Removal of Demographic Attributes from Text Data. CoRR, abs/1808.06640.
 37. Zhao, J., Wang, T., Yatskar, M., Ordonez, V., & Chang, K. (2017). Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. EMNLP.
 38. Roth, Dan & Yih, Wen-tau. (2004). A Linear Programming Formulation for Global Inference in Natural Language Tasks.
 39. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
 40. Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
 41. Řehůřek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA.
 42. Adrados, S. (2023). ASWEB [Software]. En [samueladrados/ASWEB](https://github.com/samueladrados/asweb). <https://github.com/samueladrados/asweb>.
 43. Patel, K. (2013). Incremental journey for World Wide Web: introduced with Web 1.0 to recent Web 5.0—a survey paper. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(10).
 44. Duckett, J. (2011). *HTML & CSS: design and build websites (Vol. 15)*. Wiley Indianapolis, IN.
 45. Flanagan, D. (2006). *JavaScript: the definitive guide*. “O’Reilly Media, Inc.”.
 46. Bruno, E.J.. (2006). *Ajax: Asynchronous JavaScript and XML*. *Doctor Dobbs Journal*. 31. 32-35.
 47. Resig J. (2006). jQuery. <http://jquery.com/>.
 48. Pallets. (2018). Flask [Source Code]. <https://github.com/pallets/flask>.
 49. van Rossum, G. (1995). Python tutorial. Department of Computer Science [CS]. CWI.
 50. Bostock, M. (2012). D3.js - Data-Driven Documents.

Sistema de reconocimiento de patrones en imágenes satelitales para la detección de objetos

Sergio Fernández Marcos, Vivian Félix López Batista and Gabriel Villarrubia González

Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los Caídos s/n. 37008, Salamanca, España
{sergio.fernandez, vivian, gvg}@usal.es

Resumen A lo largo de los últimos años, la inteligencia artificial (IA) ha experimentado un avance significativo en la detección de objetos en imágenes. La aplicación de técnicas de aprendizaje profundo y de *machine learning* ha revolucionado la capacidad de los algoritmos para identificar y localizar objetos en una variedad de entornos visuales. En este artículo, se presenta un estudio detallado sobre la detección de objetos en imágenes satelitales utilizando la arquitectura YOLO (*You Only Look Once*). Se realiza el proceso de entrenamiento de varios modelos YOLO utilizando un conjunto de datos específico, con el objetivo de comparar y analizar los resultados obtenidos. La investigación comienza con un marco teórico que aborda los conceptos clave de la IA, el *deep learning* y el *machine learning*, y se enfoca en las arquitecturas más destacadas para la detección de objetos en imágenes. Se destacan las características, las ventajas y la evolución de los modelos YOLO en particular, y se establece una base sólida para comprender el entrenamiento y evaluación de los algoritmos en el caso de estudio. A continuación, se presenta un caso de estudio en el cual se detalla el proceso de entrenamiento de cada algoritmo YOLO paso a paso. Se describen las configuraciones utilizadas, el conjunto de datos empleado y los métodos de evaluación para garantizar la precisión y la eficiencia en la detección de objetos. Finalmente, se presentan y analizan los resultados obtenidos al comparar los diferentes algoritmos YOLO entrenados. A pesar de la probada eficacia de diversos métodos de detección en múltiples contextos, seleccionar el más adecuado para cada implementación concreta representa un reto. La comparación y evaluación de distintas técnicas, centrada en métricas de rendimiento y conjuntos de datos representativos, son fundamentales para determinar la mejor opción para una detección de objetos precisa y confiable en imágenes.

Keywords: Reconocimiento de patrones, detección de objetos, imágenes satelitales, algoritmos de aprendizaje profundo, técnicas aprendizaje automático.

1. Introducción

La detección de objetos en imágenes es un factor indispensable en el campo de la visión por ordenador o visión artificial, que consiste en identificar y localizar

la presencia de objetos específicos en una imagen[35]. El objetivo es entrenar un modelo o algoritmo que pueda reconocer patrones visuales característicos de los objetos de interés y luego aplicar este modelo a nuevas imágenes para detectar y delimitar la ubicación de dichos objetos. En las últimas décadas, ha experimentado una gran evolución gracias al desarrollo de técnicas de aprendizaje automático y algoritmos de visión por ordenador[36]. Estos avances han permitido abordar con mayor eficacia y precisión la tarea de identificar y localizar objetos en imágenes. La detección basada en redes neuronales convolucionales (CNN, *Convolutional Neural Networks*) es eficaz para detectar objetos en imágenes al aprender características discriminativas automáticamente. Algunas arquitecturas comunes son YOLO y (R-CNN, *Region-Based Convolutional Neural Networks*). La detección basada en aprendizaje automático implica entrenar modelos como máquinas de vectores de soporte (SVM, *Support Vector Machines*), los bosques aleatorios (*Random Forests*) o clasificadores basados en CNN con conjuntos de datos etiquetados para discriminar objetos y fondos y detectar objetos en imágenes nuevas. La detección basada en características consiste en extraer características visuales relevantes y utilizar algoritmos de clasificación para identificar y localizar objetos. Estos métodos son eficaces en diversas aplicaciones, pero encontrar el enfoque más adecuado para cada implementación sigue siendo un desafío. La evaluación comparativa basada en métricas de rendimiento y el uso de conjuntos de datos representativos ayudan a identificar el método más apropiado para una detección precisa y confiable de objetos en imágenes. Este trabajo tiene como objetivo principal la investigación y evaluación de diferentes métodos de detección de objetos para seleccionar el enfoque más adecuado a una aplicación específica. Las principales tareas para lograr este objetivo son: realizar un estudio del estado del arte de los métodos existentes, crear y preparar un conjunto de entrenamiento con imágenes satelitales, entrenar modelos de detección de objetos con el conjunto de datos, evaluar el rendimiento de los algoritmos implementados, y contrastar y evaluar los diferentes métodos de detección de objetos para seleccionar el enfoque más adecuado a una aplicación específica.

2. Estado del arte

El objetivo principal de este apartado es explorar diferentes enfoques y técnicas para la detección de algoritmos y patrones de objetos en imágenes satelitales. Se explorarán algoritmos basados en aprendizaje automático, así como métodos de procesamiento de imágenes y segmentación para mejorar la precisión y eficiencia de la detección.

2.1 Inteligencia artificial

La inteligencia artificial (IA) es un campo de estudio y desarrollo de tecnología que se centra en la creación de sistemas y programas capaces de realizar tareas que normalmente requieren de la inteligencia humana. Estos sistemas son capaces de aprender, razonar, percibir, planificar y tomar decisiones, entre otras capacidades[37].

La IA se basa en la idea de que las máquinas pueden procesar y analizar grandes cantidades de datos para identificar patrones y aprender de ellos. Esto se logra a través de algoritmos y modelos matemáticos que permiten a las máquinas reconocer y comprender información, tomar decisiones y resolver problemas.

En el contexto de la IA, una de las ramas más relevantes y utilizadas es el aprendizaje automático o *machine learning*.

2.2 Aprendizaje automático

El aprendizaje automático, o *machine learning*, se centra en el desarrollo de algoritmos y modelos que permiten a los ordenadores aprender y encontrar información útil automáticamente[38]. Utiliza técnicas estadísticas y computacionales para reconocer patrones y tomar decisiones sin una programación explícita. Hay diferentes enfoques en el aprendizaje automático, como el aprendizaje supervisado, donde se proporcionan datos etiquetados para entrenar el modelo, el aprendizaje no supervisado, que busca patrones en datos no etiquetados, y el aprendizaje por refuerzo, que se basa en la interacción con un entorno para tomar decisiones secuenciales y maximizar las recompensas a lo largo del tiempo [39].

2.3 Aprendizaje profundo

El aprendizaje profundo, también conocido como *deep learning*, se enfoca en el entrenamiento de redes neuronales profundas para extraer representaciones de alto nivel de los datos[40]. Estas redes están compuestas por múltiples capas de nodos interconectados que procesan y transforman gradualmente los datos. Ha revolucionado áreas como el procesamiento de imágenes, el procesamiento del lenguaje natural y la visión por ordenador[41]. Las redes neuronales convolucionales son utilizadas para el reconocimiento visual, mientras que las redes neuronales recurrentes son eficaces en el procesamiento del lenguaje. El aprendizaje por transferencia permite aprovechar el conocimiento previo de una tarea para mejorar el rendimiento en otra. Además del aprendizaje supervisado, el aprendizaje profundo también aborda el aprendizaje no supervisado mediante técnicas como *autoencoders*, redes generativas adversarias y agrupamiento (*clustering*)[40] para descubrir patrones ocultos y estructuras en los datos sin etiquetas previas.

2.4 Detección de objetos

La detección de objetos en imágenes consiste en identificar y localizar objetos de interés en una imagen o secuencia de video. A diferencia de la clasificación de imágenes, la detección de objetos también determina la ubicación precisa del objeto y dibuja un cuadro delimitador alrededor del mismo. Tiene aplicaciones en seguridad, vigilancia, reconocimiento facial, conducción autónoma, inspección y control de calidad en la fabricación, entre otros. Para esta tarea, existen métodos y algoritmos que van desde técnicas tradicionales hasta enfoques basados en el

aprendizaje profundo. Las CNN son especialmente efectivas, ya que aprenden automáticamente características de las imágenes a diferentes niveles, lo que las hace robustas a variaciones en iluminación, orientación y escala. Además, las CNN pueden ser entrenadas con grandes cantidades de datos, permitiéndoles reconocer una amplia variedad de objetos.

2.5 Elaboración de un detector de objetos

En este apartado se va a realizar una descripción detallada de las partes que conforman la elaboración de un detector de objetos eficiente y preciso.

2.5.1 Preprocesamiento de datos Para entrenar un detector de objetos, es importante seleccionar un conjunto de datos representativo y variado. Esto implica filtrar imágenes borrosas o incorrectas y garantizar la representatividad de cada clase de objeto. Además, redimensionar las imágenes a un tamaño común mejora el rendimiento del modelo al estandarizar la escala. El etiquetado preciso de las imágenes mediante cajas delimitadoras es esencial para el entrenamiento efectivo del modelo. Estos pasos aseguran la calidad y consistencia de los datos utilizados en el proceso de detección de objetos.

2.5.2 Entrenamiento del modelo En la etapa de entrenamiento del modelo, se divide el conjunto de datos en conjuntos de entrenamiento y prueba mediante una división conocida como *train/test split*. Se aplican técnicas de aumento de datos, como el volteo y la rotación, para aumentar la variabilidad de los datos y mejorar la capacidad de generalización del modelo. Además, se ajustan parámetros clave como la arquitectura de red, la tasa de aprendizaje, el número de capas, el tamaño del lote y el número de épocas para obtener un modelo óptimo en términos de precisión y eficiencia.

2.5.3 Detección de los objetos Finalmente, llegamos al proceso de detección de objetos utilizando la red que ha sido entrenada específicamente para esta tarea. Una vez que hemos ajustado los pesos de la red durante el entrenamiento, proporcionamos a la red datos de entrada, que en este contexto son imágenes que pueden contener el objeto que deseamos detectar.

En este punto, el sistema realiza la detección de los objetos presentes en las imágenes de entrenamiento, utilizando el proceso que hemos descrito anteriormente. Los cuadros delimitadores se dibujan alrededor de los objetos detectados, indicando su presencia. Además, en las arquitecturas que vamos a detallar más adelante, obtenemos información precisa sobre la ubicación de los objetos en la imagen, así como la clase a la que pertenecen.

2.5.4 Estudio de la literatura sistemática Se ha llevado a cabo un estudio de la literatura sistemática, donde se han identificado varios artículos relevantes que abordan la detección de objetos en imágenes satelitales utilizando dos enfoques populares: YOLO [4,8,9,2] y R-CNN [13,6,11,19,20,25,23,10,17].

Las redes neuronales convolucionales que también se han utilizado en varios artículos [5,7,22,21,16] son la base fundamental tanto de YOLO como de R-CNN. Estas arquitecturas de red están diseñadas específicamente para el procesamiento y análisis de imágenes. Las CNN son capaces de aprender y extraer características relevantes de las imágenes a través de capas de convolución y *pooling*, lo que las hace especialmente efectivas en tareas de visión por ordenador como la detección de objetos.

Por otro lado, R-CNN adopta un enfoque basado en regiones, donde se generan propuestas de regiones de interés en la imagen utilizando técnicas como *Selective Search*. Luego, cada región propuesta se procesa individualmente utilizando una red neuronal convolucional para clasificar y localizar los objetos presentes.

A su vez hay artículos donde suelen utilizar varios algoritmos juntos como Yolo, R-CNN y SSD (*Single Shot MultiBox Detector*) y compararlos [14,15,12] o artículos que utilizan modelos que emplean diferentes versiones de los algoritmos como Fast-R-CNN o RestNet, que se destacan por su capacidad para entrenar redes muy profundas, evitando el problema de la degradación del rendimiento que se encuentra en las redes más profundas convencionales [24,18,3,1].

3. Métodos y algoritmos de detección de objetos

En esta sección se destacan diversas estructuras arquitectónicas utilizadas en la detección de objetos en la actualidad. Es indispensable comprender su funcionamiento, ya que permite obtener una visión global de los diversos enfoques empleados para llevar a cabo un detector.

3.1 Sliding Window Systems

Este método es el que primero apareció y se empezó a utilizar, según Viola y Jones (2001) [26], implica deslizar una ventana rectangular de tamaño fijo sobre la imagen de entrada, examinando cada posición de la ventana para determinar la presencia de objetos de interés. Además, Dalal y Triggs (2005) [27] mencionaron que el sistema de ventana deslizante se basa en la extracción de características relevantes de la región cubierta por la ventana y la clasificación utilizando un clasificador previamente entrenado. En el método de ventana deslizante, se elige el tamaño de la ventana según el objeto a detectar y se desliza sobre la imagen para extraer características relevantes de cada región. Luego, se utiliza un clasificador previamente entrenado y se establece un umbral de detección para determinar si la ventana contiene el objeto o no. El sistema de ventana deslizante se repite en diferentes escalas y aspectos de la imagen para abordar la variabilidad en el tamaño y la orientación de los objetos.

3.2 Region-Based Convolutional Neural Networks

El R-CNN (*Region-based Convolutional Neural Network*) es un enfoque pionero en la detección de objetos que fue presentado por Girshick et al. en 2014

[28]. Este método introdujo una nueva forma de abordar la detección de objetos mediante el uso de redes neuronales convolucionales.

En el método R-CNN, se utilizan algoritmos de generación de propuestas para identificar regiones de la imagen con objetos, que luego se redimensionan y pasan por una red neuronal convolucional preentrenada para extraer características. Estas características se utilizan en clasificadores individuales para determinar la clase de cada región y se emplea un regresor para mejorar la precisión de la localización del objeto dentro de la región.

El enfoque R-CNN introdujo el uso de redes neuronales convolucionales para la detección de objetos y ha demostrado un rendimiento favorable en términos de precisión. Sin embargo, este método tenía la desventaja de ser computacionalmente costoso y lento debido a la necesidad de procesar cada región propuesta de forma independiente.

Posteriormente, han surgido mejoras al método R-CNN para abordar sus limitaciones. Estas incluyen el Fast R-CNN[29], el Faster R-CNN[30] y el Mask R-CNN [31].

3.3 Single Shot Detector

El *Single Shot Detector* (SSD) es un método rápido y preciso para la detección de objetos introducido por Liu et al. en 2016. Se destaca por su diseño de red neuronal en una sola pasada, el uso de anclas o *anchors*, clasificación por niveles, características multi-escala, su eficiencia y precisión. El SSD ha demostrado excelentes resultados en la detección de objetos en tiempo real y es ampliamente utilizado en aplicaciones de visión por ordenador[33].

3.4 RetinaNet

El RetinaNet es un método de detección de objetos introducido por Lin et al. en 2017 [34]. El RetinaNet es un método de detección de objetos que aborda el desafío de detectar objetos en diferentes escalas y tamaños con alta precisión. Utiliza una arquitectura basada en redes neuronales convolucionales que combina características de alto y bajo nivel para obtener detecciones precisas. El RetinaNet se compone de un extractor de características y una cabeza de detección, que utilizan anclas y clasificadores por niveles para detectar objetos en diferentes escalas.

3.5 You Only Look Once

YOLO *You Only Look Once* es un enfoque popular introducido por Redmon et al. en 2016[32]. Una de las características distintivas de YOLO es su enfoque en la detección de objetos en tiempo real de forma directa y eficiente en una sola pasada de la red neuronal sobre la imagen completa de la red neuronal convolucional.

La red neuronal se divide en dos partes principales: la parte de extracción de características y la parte de detección, como se observa en la Figura 1.

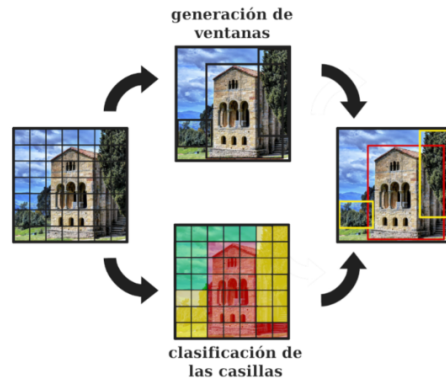


Figura 1: Detección de imágenes Yolo.[46]

La extracción de características se realiza mediante una red convolucional profunda, como Darknet, entrenada previamente en grandes conjuntos de datos. YOLO realiza detección en una sola pasada, dividiendo la imagen en celdas y prediciendo cuadros delimitadores y probabilidades de clase. Aunque es eficiente y rápido, YOLO puede tener dificultades en objetos pequeños y detalles finos, y presenta limitaciones en la precisión de la localización en comparación con enfoques basados en regiones.

3.5.1 Evolución de Yolo desarrollado en 2015 por Joseph Redmon et al., revolucionó la detección de objetos al proponer un enfoque de detección en un solo paso. Dividiendo la imagen en una cuadrícula, cada celda realizaba predicciones de cuadros delimitadores y probabilidades de clase. Aunque YOLO era rápido, tenía problemas de falsas detecciones y dificultades con objetos pequeños.

3.5.2 YOLOv2 (también conocido como Yolo 9000) La versión YOLOv2 2.7.1[44] mejora la precisión y velocidad de detección en comparación con las versiones anteriores. Se incorpora normalización por lotes en las capas de convolución para normalizar y centrar los resultados. El clasificador se entrena y detecta utilizando imágenes de 448x448, lo que mejora la precisión media. Se introducen cajas de anclaje para detectar múltiples objetos por celda. Las coordenadas de ubicación se predicen en función de las celdas de la cuadrícula. La arquitectura de red utiliza el modelo Darknet19 con 19 capas convolucionales y 5 capas de máxima agrupación, logrando una precisión notablemente superior a la versión anterior.

3.5.3 YOLOv3 [43] es una versión actualizada y mejorada de YOLO que ofrece mejoras significativas en varios aspectos. Es más rápido y preciso que sus predecesores, alcanzando una precisión similar a SSD pero siendo tres veces más rápido

a una resolución de 320×320 . Predice cuadros delimitadores utilizando cajas de anclaje y asigna un cuadro delimitador previo para cada objeto de fondo. En lugar de *softmax*, utiliza clasificadores logísticos independientes para predecir las clases. Realiza predicciones en 3 escalas y utiliza una red de extracción de características llamada "Darknet-53". Se entrena en imágenes completas, empleando entrenamiento a múltiples escalas, ampliación de datos y normalización de lotes.

3.5.4 Yolov4 [42] es una arquitectura avanzada de detección de objetos que utiliza la red troncal CSPDarknet53. Incorpora técnicas como el módulo de atención espacial (SAM, *Spatial Attention Module*) y la red de agregación de rutas (PAM, *Path Aggregation Network*) para mejorar la precisión y velocidad. Utiliza la estrategia (CIOU, *Complete Intersection over Union*) para la localización precisa de objetos y la eliminación de mosaico para detectar objetos pequeños. Aunque es más grande, mantiene una alta velocidad de procesamiento. El aumento de datos y el entrenamiento previo en ImageNet son parte del proceso de entrenamiento de YOLOv4.

3.5.5 Yolov5 es un modelo de detección de objetos utilizado en visión por computadora. Viene en diferentes versiones (pequeña, mediana, grande y extra grande) con tasas de precisión crecientes. Se basa en la arquitectura de YOLOv4 y utiliza los procedimientos de entrenamiento de PyTorch para mejorar su rendimiento. Se ha demostrado que todas las variantes de YOLOv5 son más rápidas en el entrenamiento que EfficientDet, y el modelo más preciso (YOLOv5x) procesa imágenes más rápido con una precisión similar. YOLOv5 utiliza aumentación de datos y cálculos de pérdida durante el entrenamiento. Su mayor contribución es la traducción de Darknet al marco PyTorch, lo que facilita la implementación de avances en investigación. La aumentación de datos en YOLOv5 incluye escalado, ajustes de espacio de color y aumentación de mosaico, que ayuda a abordar el problema de los objetos pequeños en el popular conjunto de datos COCO. La configuración del modelo se realiza en archivos .yaml, y los modelos se entrenan de manera rápida y eficiente, lo que reduce los costos de experimentación.

3.5.6 Yolov6 también conocido como MT-YOLOv6, es un modelo de detección de objetos basado en la arquitectura YOLO. Desarrollado por investigadores en Meituan, YOLOv6 supera en rendimiento a YOLOv5 en el conjunto de datos MS COCO. El modelo realiza mejoras en la parte posterior y el cuello de YOLO, introduciendo el *EfficientRep Backbone* y el *Rep-PAN Neck*, optimizados para el hardware. En YOLOv6, las cabezas de clasificación y regresión de cajas están desacopladas, lo que aumenta el rendimiento del modelo. Además, se implementan mejoras en el proceso de entrenamiento, como el entrenamiento libre de anclajes, la asignación de etiquetas *SimOTA* y la pérdida de regresión de cajas *SIoU*. YOLOv6 logra un mayor mAP en COCO en diferentes tasas de FPS en comparación con YOLOv5, YOLOX y PPE-YOLOE. Sin embargo, es importante tener en cuenta que el conjunto de datos COCO es una aproximación y no refleja perfectamente el rendimiento en otros conjuntos de datos.

3.5.7 YOLOv7 [45] es una arquitectura de detección de objetos diseñada para mejorar la precisión de las cajas delimitadoras a velocidades de inferencia similares. Los autores implementaron varias innovaciones, incluyendo la Agregación Extendida de Capas Eficientes (E-ELAN) para una inferencia más eficiente, técnicas de escalado del modelo para adaptarse a diferentes necesidades y la planificación de re-parametrización para aumentar la robustez. También introdujeron una cabeza auxiliar en el medio de la red para mejorar las predicciones, aunque su entrenamiento no es tan eficiente como la cabeza final, su supervisión mejora el rendimiento global del modelo. Estas mejoras buscan lograr una detección de objetos más precisa y eficiente en YOLOv7.

3.5.8 YOLOv8 es la última versión del modelo YOLO y presenta varias mejoras y cambios en su arquitectura. Entre las novedades se incluye la detección libre de anclas, lo que elimina la necesidad de utilizar cajas de anclas para hacer predicciones y mejora la velocidad de la supresión no máxima (NMS). Además, se han introducido nuevas convoluciones y bloques de construcción en la red para reducir el número de parámetros y el tamaño de los tensores. En cuanto a la aumentación de mosaicos, se ha observado que puede degradar el rendimiento si se utiliza durante todo el entrenamiento, por lo que se desactiva en las últimas etapas. YOLOv8 ha demostrado una alta precisión en la evaluación COCO y ha superado a versiones anteriores en términos de mAP@.50 en la evaluación RF100 de Roboflow[48]. La investigación se ha centrado en experimentos empíricos y el código está disponible en un repositorio público para su uso y mejora por parte de la comunidad.

3.6 Evaluación de los métodos

En la evaluación de un modelo de visión por ordenador, se buscan maximizar los verdaderos positivos (TP) y verdaderos negativos (TN), mientras se intenta minimizar los falsos positivos (FP) y falsos negativos (FN). Un TP ocurre cuando el modelo detecta y clasifica correctamente un objeto, mientras que un FP ocurre cuando el modelo detecta incorrectamente un objeto que no debería estar presente. Por otro lado, un verdadero negativo ocurre cuando el modelo reconoce correctamente la ausencia de un objeto, y un FN ocurre cuando el modelo no logra detectar un objeto que sí está presente. Mejorar el etiquetado de objetos similares y abordar casos atípicos puede ayudar a reducir los FPs y FNs en las predicciones del modelo.

Los cuatro grupos forman la denominada matriz de confusión que se puede observar en la siguiente Figura 2.

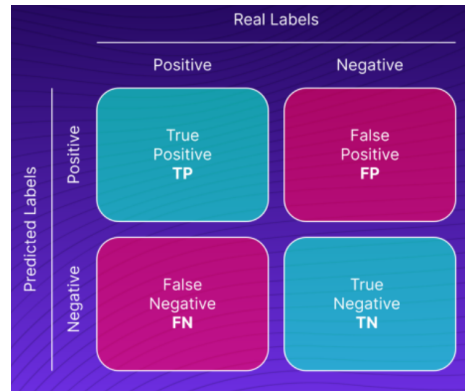


Figura 2: Matriz de confusión.[47]

Una vez explicados los posibles resultados de la previsión, las métricas que se utilizan para determinar el comportamiento de los modelos entrenados son los siguientes:

- **Precisión:** La fórmula de precisión es Verdadero Positivo dividido por la suma de Verdadero Positivo y Falso Positivo.

$$P = \frac{TP}{TP + FP}$$

- **Recall:** La fórmula para recordar es Verdadero Positivo dividido por la suma de Verdadero Positivo y Falso Negativo.

$$P = \frac{TP}{TP + FN}$$

- **Curva de Precisión-Recall:** Gráfico que muestra la relación entre la precisión y el *recall* al variar el umbral de clasificación del modelo.
- **Boss loss:** Métrica de pérdida que evalúa la calidad de los cuadros delimitadores predichos en relación con los objetos reales del conjunto de datos.
- **Pérdida de clasificación:** Métrica de pérdida que evalúa la corrección de la clasificación de los cuadros delimitadores predichos.
- **mAP:** Precisión media utilizada para medir el rendimiento de modelos de visión artificial, se obtiene como promedio de la métrica de precisión promedio para todas las clases.
- **F-score:** Medida de la precisión y exhaustividad de un modelo en un problema de clasificación, se calcula a partir de la combinación armónica de la precisión y el *recall*.

$$F1 = \frac{2 \cdot (precision \cdot recall)}{precision + recall}$$

- La **precisión** es la proporción de casos positivos correctamente clasificados sobre el total de casos clasificados como positivos.
- El **recall** es la proporción de casos positivos correctamente clasificados sobre el total de casos positivos en el conjunto de datos.

4. Caso de estudio

El conjunto de datos utilizado ha consistido en 3510 imágenes satelitales clasificadas en cinco categorías principales: piscinas, paneles solares, aviones, barcos y pistas de tenis. Estas categorías se han seleccionado debido a su presencia común en mapas satelitales y su relevancia en diversas aplicaciones. Se ha llevado a cabo un proceso de preprocesamiento para fusionar proyectos y filtrar imágenes borrosas o mal seleccionadas. Además, se ha ajustado el nivel de *zoom* y se han redimensionado las imágenes a un tamaño común de 600x600 píxeles.

Se ha realizado el etiquetado de las imágenes, asignando etiquetas que indican la presencia y ubicación de los objetos de interés. Se han aplicado técnicas de aumentación, como volteo y rotación, para enriquecer el conjunto de datos y mejorar el rendimiento del modelo de detección. Estas técnicas han permitido al modelo aprender desde diferentes perspectivas y condiciones, mejorando su capacidad de generalización. En total, se ha obtenido un conjunto de datos de 7,610 imágenes, lo que proporciona una amplia variedad de escenarios para el aprendizaje del modelo de detección de objetos.

4.1 Entrenamiento de los modelos

Se ha llevado a cabo un proceso general para la configuración y entrenamiento de los modelos de YOLO. Esto ha implicado clonar el repositorio de GitHub correspondiente a la versión de YOLO, preparar el entorno de Google Colaboratory e instalar las dependencias necesarias. El modelo se ha configurado para el entrenamiento, especificando parámetros como el número de épocas y la tasa de aprendizaje.

Se ha realizado el entrenamiento del modelo utilizando el conjunto de datos importado de Roboflow y se ha guardado el modelo entrenado para su posterior evaluación. Se ha llevado a cabo un primer entrenamiento con 200 épocas y un segundo entrenamiento con 50 épocas. Este proceso ha permitido reproducir los resultados originales de cada versión de YOLO y adaptar los modelos según las necesidades del estudio.

4.1.1 Yolov3 El primer paso en el proceso de configuración para el entrenamiento del modelo es asegurarse de que Google Colab ha asignado una GPU para el uso durante el cuaderno. Posteriormente, se descomprime el archivo .ZIP que contiene el conjunto de datos de imágenes previamente preparado para entrenar y probar el modelo YOLOv3. Finalmente, cuando se establece el entorno, se realiza el entrenamiento del modelo YOLOv3 con los siguientes parámetros:

Tamaño de imagen: 640x640 píxeles (*-img 640*), Tamaño del lote: 16 (*-batch 16*), Épocas: 200 (*-epochs 200*), Configuración del conjunto de datos: archivo *data.yaml* (*-data dataset.location/data.yaml*), Pesos preentrenados: archivo *yolov3.pt* (*-weights /content/yolov3/yolov3.pt*), Caché de imágenes y etiquetas (*-cache*). Se utilizarán estos pasos para entrenar el modelo YOLOv3 con el conjunto de datos disponible en Google Colab. Tras completar el proceso de entrenamiento de 200 épocas, se obtuvieron los siguientes resultados

200 epochs completed in 22.183 hours.

Optimizer stripped from runs/train/exp/weights/last.pt, 123.6MB

Optimizer stripped from runs/train/exp/weights/best.pt, 123.6MB

El entrenamiento de 200 épocas tardó aproximadamente 22.183 horas, generando los archivos de pesos *last.pt* y *best.pt*. *Last.pt* contiene los pesos del modelo al final de la última época, mientras que *best.pt* contiene los pesos en la época con la mejor precisión en el conjunto de validación. La optimización de estos archivos reduce su tamaño y permite su uso en inferencia sin cargar el estado del optimizador. Además, se crea un archivo *results.csv* para analizar los resultados del entrenamiento y obtener conclusiones. En la Figura 3 se puede observar la implementación del modelo propuesto. Es evidente que el modelo identifica de manera efectiva las clases dentro de las imágenes satelitales proporcionadas. Las clases están asignadas con números que van desde 0 hasta 4, donde cada uno de ellos corresponde a una clase específica: 0 para los barcos, 1 para los aviones, 2 para las piscinas, 3 para las placas solares, y finalmente 4 para las pistas de tenis.

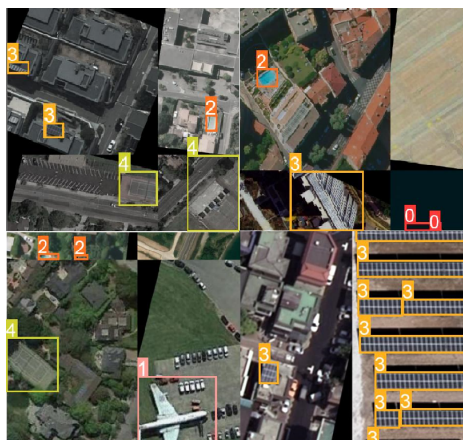


Figura 3: Validación YoloV3.

Al finalizar el entrenamiento del modelo se realizó un segundo entrenamiento del modelo YoloV3 con los mismos parámetros, exceptuando el número de épocas, el cual se ha modificado de 200 a 50 épocas.

4.1.2 YOLOv5 Se copia el repositorio de YOLOv5 de Ultralytics y se instalan las dependencias. Una vez creado y exportado el conjunto de datos en formato YOLOv5, se procede al entrenamiento del modelo seleccionado. Se utiliza el script *train.py* de YOLOv5 con argumentos específicos, como el tamaño de imagen, el tamaño del lote, el número de épocas, la ubicación del archivo YAML y los pesos iniciales del modelo. Después de completar el entrenamiento, el modelo está listo para su uso.

```
!python train.py --img 640 --batch 16 --epochs 200 \
--data {dataset.location}/data.yaml --weights yolov5s.pt --cache
```

Durante el entrenamiento de YOLOv5, se utilizan argumentos claves para ajustar el proceso, como el tamaño de las imágenes de entrada (`-img 640`) y el tamaño del lote (`-batch 16`). Además, se define el número de épocas (`-epochs 200`) y se especifica la ubicación del archivo YAML que contiene la información del conjunto de datos (`-data dataset.location/data.yaml`).

Un aspecto importante es la inicialización de los pesos del modelo (`-weights yolov5s.pt`), utilizando los pesos del modelo YOLOv5s preentrenado, lo que proporciona una base sólida para el entrenamiento. También se activa el almacenamiento en caché de las imágenes (`-cache`) para acelerar el proceso. La matriz de confusión de la Figura 4 muestra una clasificación precisa de las clases, con un rendimiento destacado en las clases de piscinas y pistas de tenis. Sin embargo, la clase de barcos presenta la menor precisión en la clasificación.

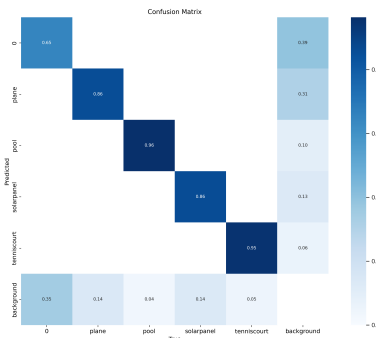


Figura 4: Matriz de confusión YOLOv5.

Al finalizar el entrenamiento del modelo se realizó un segundo entrenamiento del modelo YOLOv5 con los mismos parámetros, exceptuando el número de épocas, el cual se ha modificado de 200 a 50 épocas.

4.1.3 YOLOv7 Para utilizar YOLOv7 en la detección de objetos, primero se instalan las dependencias requeridas y se clona el repositorio. Luego, se carga el conjunto de datos y se descargan los pesos preentrenados. A continuación,

se utiliza el script de entrenamiento con parámetros como el tamaño del lote, el número de épocas, la ubicación de los datos y los pesos preentrenados. El entrenamiento se realiza en el dispositivo especificado.

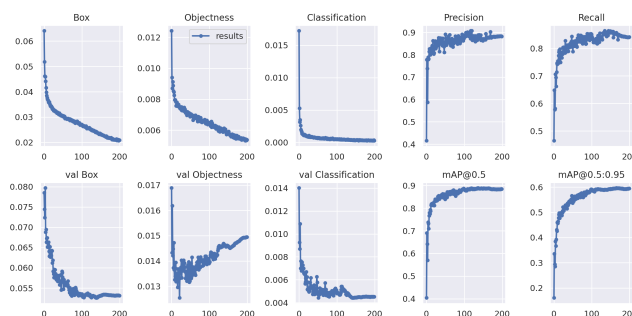


Figura 5: Resultados YoloV7.

En la Figura 5 se muestran todos los resultados del entrenamiento del modelo YoloV7. El modelo fue entrenado durante 200 épocas, y se obtuvieron los siguientes resultados: pérdida de caja de entrenamiento (0.027656), pérdida de objeto de entrenamiento (0.006507), pérdida de clasificación de entrenamiento (0.000729) y pérdida total (0.034892). Además, se lograron métricas sólidas, como una precisión del 86.2 %, un *recall* del 82.2 %, un mAP a 0.5 de 0.861656 y un mAP a 0.5:0.95 de 0.559435. También se evaluaron las pérdidas y métricas correspondientes al conjunto de validación.

4.1.4 YoloV8 El entrenamiento de YOLOv8 se realiza a lo largo de 200 épocas, es decir, un total de 200 pasadas a través de todo el conjunto de datos de entrenamiento. En la Tabla 1 se muestran los resultados tras finalizar el entrenamiento. En cada época, el modelo está tratando de minimizar tres tipos de pérdida: la pérdida de caja (*box_loss*), la pérdida de clase (*cls_loss*), y la pérdida de deformación de filtro (*dfl_loss*).

Tabla 1: Resultados de entrenamiento tras 200 épocas con YOLOv8

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95
all	700	2334	0.872	0.819	0.871	0.592
0	700	579	0.791	0.549	0.628	0.356
plane	700	499	0.853	0.832	0.883	0.553
pool	700	335	0.92	0.946	0.968	0.578
solarpanel	700	780	0.933	0.819	0.927	0.704
tenniscourt	700	141	0.863	0.95	0.949	0.768

Por último, el modelo ha sido validado con los mejores pesos guardados, obteniendo resultados similares a los del entrenamiento. También se almacenan métricas como la precisión o el *recall* proporcionando información sobre la eficiencia del modelo en términos de tiempo de ejecución.

En la Figura 6, se observa la matriz de confusión de la clasificación de las 5 clases usando YoloV8. En el eje vertical se muestran las clases predichas, mientras que en el eje horizontal están las que son verdaderas. Las clases que mejor han sido clasificadas son las piscinas (*pool*) y las pistas de tenis (*tenniscourt*) aunque en general todas se han clasificado correctamente, exceptuando la clase más baja, la de los barcos (θ). El resultado es mejor que en el modelo YoloV5 para todas las clases, pero peor que el modelo YoloV7.

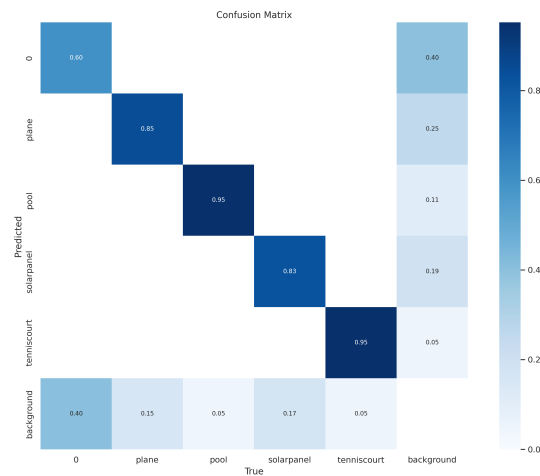


Figura 6: Matriz de confusión YoloV8.

5. Resultados

En esta sección se ha realizado una comparación exhaustiva entre los resultados de cada modelo entrenado. Se comparan los modelos entre sí y por último se realiza una comparación entre los valores máximos que ha obtenido cada algoritmo en las métricas más relevantes.

Tabla 2: Resultados de entrenamiento para todos los modelos con 200 épocas.

Métricas	Yolov3	Yolov5	Yolov7	Yolov8
train/box_loss	0.024511	0.030749	0.027656	1.042255
train/obj_loss	0.014552	0.018133	0.006507	0.621270
train/cls_loss	0.000514	0.000588	0.000729	1.093480
metrics/precision	0.874514	0.862278	0.862050	0.847831
metrics/recall	0.814735	0.812028	0.822230	0.806700
metrics/mAP_0.5	0.853922	0.856920	0.861656	0.849348
metrics/mAP_0.5:0.95	0.554961	0.539615	0.559435	0.563308
val/box_loss	0.033257	0.034163	0.055734	1.297333
val/obj_loss	0.015781	0.013572	0.014189	0.687272
val/cls_loss	0.000457	0.000330	0.005099	1.196873
x/lr0	0.005575	0.005575		0.005575
x/lr1	0.005073	0.005073		0.005073
x/lr2	0.005073	0.005073		0.005073
hours	22.183	4.194	21.018	5.309

En la Tabla 2 se realiza una comparación entre todos los modelos entrenados con 200 épocas. La comparación de los modelos entrenados con 200 épocas revela que YOLOv3, YOLOv5 y YOLOv7 tienen pérdidas de entrenamiento similares, con YOLOv3 mostrando la menor pérdida para 'box' y 'obj', y YOLOv5 la menor para 'cls'. Por otro lado, YOLOv8 muestra pérdidas considerablemente más altas en todas las categorías. En términos de precisión, YOLOv3 tiene la mayor precisión, seguido de cerca por YOLOv5 y YOLOv7. A pesar de su alta pérdida, YOLOv8 también muestra una buena precisión. En cuanto al *recall*, YOLOv7 tiene el *recall* más alto, seguido de cerca por YOLOv3 y YOLOv5. Nuevamente, YOLOv8 muestra un *recall* competitivo a pesar de las altas pérdidas.

En cuanto a las métricas mAP, todos los modelos son comparables en mAP@0.5, con YOLOv7 liderando ligeramente. Sin embargo, en mAP@0.5:0.95, YOLOv8 supera al resto. Al observar la pérdida de validación, YOLOv8 tiene una pérdida mucho mayor que los otros modelos. Entre los otros modelos, YOLOv7 tiene la mayor pérdida de 'box', mientras que YOLOv3 tiene las menores pérdidas de 'obj' y 'cls'. Es notable como la pérdida de 'cls' en YOLOv8 es mucho mayor en comparación con los otros modelos, lo que podría indicar problemas en la clasificación durante el entrenamiento.

En cuanto al desempeño de validación versus entrenamiento, YOLOv7 muestra una diferencia considerable entre la pérdida de entrenamiento y la de validación para 'box' y 'cls', sugiriendo un posible sobre ajuste del modelo a los datos de entrenamiento. Por otro lado, YOLOv3 presenta una consistencia más alta entre los resultados de entrenamiento y validación, lo que podría indicar una mejor generalización del modelo.

La relación entre precisión y *recall* también varía entre los modelos. Aunque YOLOv3 presenta la mayor precisión, su *recall* no es el más alto, siendo superado por YOLOv7. Esto resalta la importancia de considerar el equilibrio entre precisión y *recall* al elegir el modelo adecuado.

En términos de mAP, YOLOv8 tiene la puntuación más alta en mAP@0.5:0.95, pero también tiene pérdidas de entrenamiento y validación mucho más altas. Esto podría indicar que, a pesar de su rendimiento en esta métrica, el modelo podría no ser el más robusto o el más generalizable.

Tabla 3: Resultados de entrenamiento para todos los modelos con 50 épocas.

Métricas	Yolov3	Yolov5	Yolov7	Yolov8
train/box_loss	0.033568	0.038457	0.033356	1.287606
train/obj_loss	0.018916	0.021636	0.007599	0.817990
train/cls_loss	0.001204	0.001441	0.001399	1.219344
metrics/precision	0.850224	0.842288	0.821928	0.821393
metrics/recall	0.795184	0.775106	0.776222	0.771372
metrics/mAP_0.5	0.837811	0.820441	0.811752	0.817963
metrics/mAP_0.5:0.95	0.515674	0.494896	0.497808	0.522049
val/box_loss	0.035766	0.036847	0.058317	1.380988
val/obj_loss	0.014280	0.014038	0.014244	0.808646
val/cls_loss	0.000730	0.000701	0.005372	1.231450
x/lr0	0.007137	0.007137		0.007137
x/lr1	0.005132	0.005132		0.005132
x/lr2	0.00513	0.005132		0.005132
hours	6.911	1.475	5.445	2.801

En la Tabla 3 se realiza una comparación entre todos los modelos entrenados con 50 épocas.

La comparación de los modelos revela que YOLOv3, YOLOv5 y YOLOv7 tienen pérdidas de entrenamiento similares, con YOLOv3 mostrando la menor pérdida para 'box' y 'obj', YOLOv5 la menor para 'cls', y YOLOv7 con pérdidas intermedias. YOLOv8 muestra pérdidas considerablemente más altas en todas las categorías. En cuanto a la precisión, YOLOv3 tiene la mayor precisión, seguido de cerca por YOLOv5 y YOLOv7. A pesar de su alta pérdida, YOLOv8 también muestra una buena precisión. En términos de *recall*, YOLOv7 tiene el *recall* más alto, con YOLOv3 y YOLOv5 muy cerca. YOLOv8 no se queda muy atrás a pesar de las altas pérdidas.

En cuanto a las métricas mAP, todos los modelos son comparables en mAP@0.5, con YOLOv7 liderando ligeramente. Sin embargo, en mAP@0.5:0.95, YOLOv8

supera al resto a pesar de su alta pérdida. YOLOv8 tiene una pérdida de validación mucho mayor que los otros modelos, y entre los otros modelos, YOLOv7 tiene la mayor pérdida de 'box', mientras que YOLOv3 tiene las menores pérdidas de 'obj' y 'cls'.

Resulta notable como la pérdida de 'cls' en YOLOv8 es órdenes de magnitud mayor en comparación con los otros modelos, lo que podría indicar problemas en la clasificación durante el entrenamiento. YOLOv7 muestra una diferencia considerable entre la pérdida de entrenamiento y la de validación para 'box' y 'cls', sugiriendo un posible sobre ajuste del modelo a los datos de entrenamiento. Por otro lado, YOLOv3 presenta una consistencia más alta entre los resultados de entrenamiento y validación, lo que podría indicar una mejor generalización del modelo.

Tabla 4: Resultados de 200 épocas

	Yolov3	Yolov5	Yolov7	Yolov8
Precisión	0.90174	0.89157	0.9076	0.87427
Recall	0.84385	0.84305	0.8643	0.87427
mAP	0.87601	0.87968	0.8897	0.87133

En la Tabla 4 se realiza una comparación entre los valores máximos que han alcanzado los modelos con un entrenamiento con 200 épocas.

YOLOv7 muestra la mayor precisión, seguido por YOLOv3 y YOLOv5 con resultados cercanos. YOLOv8 tiene la menor precisión. En términos de *recall*, YOLOv7 lidera, seguido por YOLOv3 y YOLOv8, mientras que YOLOv5 tiene el valor más bajo. En cuanto a *mAP*, YOLOv7 lidera, seguido de cerca por YOLOv5 y YOLOv3, y YOLOv8 obtiene la menor puntuación.

Tabla 5: Resultados de 50 épocas

	Yolov3	Yolov5	Yolov7	Yolov8
Precisión	0.90564	0.89176	0.8996	0.86331
Recall	0.84214	0.82183	0.8534	0.82692
mAP	0.88078	0.86555	0.8854	0.86786

En la Tabla 5 se realiza una comparación entre los valores máximos que han alcanzado los modelos con un entrenamiento con 50 épocas.

En cuanto a precisión, YOLOv3 tiene la mayor, seguido de cerca por YOLOv7. YOLOv5 y YOLOv8 presentan resultados similares y menor precisión. En términos de *recall*, YOLOv7 tiene el valor más alto, seguido por YOLOv3, YOLOv8 y finalmente YOLOv5. Con relación a *mAP*, YOLOv7 obtiene la mayor puntuación, seguido de YOLOv3, YOLOv5 y finalmente YOLOv8.

6. Conclusiones

En este trabajo se ha realizado un estudio detallado de los métodos de detección de objetos en imágenes satelitales, centrandolo en los algoritmos más empleados hasta la fecha. Se ha llevado a cabo una exhaustiva revisión de la literatura, incluyendo una amplia gama de técnicas y arquitecturas. Por otro lado, se ha investigado la evolución del algoritmo YOLO, analizando las distintas versiones y mejoras a lo largo de su desarrollo. Esta revisión del estado del arte ha proporcionado una base sólida para el desarrollo y la evaluación posterior de los métodos de detección implementados.

A su vez se ha creado un conjunto de datos personalizado, para el caso de estudio, compuesto por imágenes satelitales. Este conjunto fue cuidadosamente seleccionado y adaptado, con el objetivo de ser utilizado en la detección de objetos en los métodos implementados. Se ha realizado el preprocesamiento de las imágenes, el cual implicó la normalización, el redimensionado y la eliminación de imágenes para asegurar las condiciones adecuadas para el entrenamiento de los modelos. Además, se llevaron a cabo anotaciones para etiquetar los objetos de interés en las imágenes, permitiendo así el entrenamiento y validación de los modelos con una elevada precisión. La creación de este conjunto de datos personalizado demostró ser un pilar fundamental en el éxito del proyecto. Se ha realizado el entrenamiento de los modelos de detección de objetos con el conjunto de datos. Hemos logrado entrenar con éxito varios modelos de detección de objetos utilizando nuestro conjunto de datos. Posteriormente, se ha realizado la evaluación del rendimiento de los algoritmos y métodos de detección implementados: Hemos evaluado de manera exhaustiva los algoritmos y métodos de detección que implementamos, usando una variedad de métricas para obtener una comprensión clara de su rendimiento. Por último, se ha realizado una comparación y evaluación de los diferentes métodos de detección de objetos con el objetivo de poder seleccionar el enfoque que mejor se adapte a una aplicación concreta. En general, el entrenamiento de los modelos durante 200 épocas produjo mejores resultados en términos de pérdidas y métricas de rendimiento en comparación con 50 épocas. Este resultado es consistente en todos los modelos y sugiere que un mayor número de épocas permite al modelo aprender y ajustarse mejor a los datos de entrenamiento.

Sin embargo, en términos de eficiencia (tiempo y recursos de cómputo), entrenar durante 50 épocas puede ser una mejor opción. Aunque los modelos entrenados durante 50 épocas pueden no rendir tan bien como sus contrapartes de 200 épocas, aún proporcionan resultados razonablemente buenos y lo hacen en un tiempo más corto.

Al comparar todos los modelos entre sí, YOLOv7 y YOLOv3 tienden a proporcionar los mejores resultados en general. Estos dos modelos muestran menores pérdidas y métricas de rendimiento más altas en comparación con YOLOv5 y YOLOv8.

YOLOv8 se destacó por tener valores de pérdida significativamente más altos, lo que podría indicar un sobre ajuste del modelo. Aunque las métricas de rendimiento de YOLOv8 no son las más bajas, la alta pérdida sugiere que el modelo puede tener dificultades para generalizar a datos nuevos. En términos de las métricas máximas alcanzadas, YOLOv7 nuevamente tiende a liderar en la mayoría de las métricas para ambas duraciones de entrenamiento. YOLOv3 y YOLOv5 a menudo le siguen de cerca.

Estos resultados sugieren que YOLOv7 puede ser la mejor opción si se busca el máximo rendimiento, independientemente de la duración del entrenamiento. Sin embargo, la elección del modelo también puede depender de otros factores, como las necesidades específicas de la aplicación, la disponibilidad de recursos de cómputo y la cantidad de tiempo disponible para el entrenamiento.

Si la prioridad es obtener el mejor rendimiento posible, parece que un entrenamiento más largo (200 épocas) con YOLOv7 sería la mejor opción. Sin embargo, si la eficiencia y la economía de recursos son una consideración importante, entonces un entrenamiento más corto (50 épocas) con YOLOv3 podría ser una buena elección.

6.1 Líneas futuras

Como líneas de trabajo futuras planteamos algunas alternativas:

1. Ampliación de la comparación a otros modelos de detección de objetos: Existen otros modelos de detección de objetos más allá de los basados en YOLO, como SSD, Faster R-CNN, o EfficientDet. La inclusión de estos modelos en la comparativa proporcionaría una visión más completa del panorama de la detección de objetos.
2. Prueba con diferentes configuraciones de los modelos: Cada uno de los modelos de YOLO tiene diversas configuraciones que se pueden ajustar. Analizar cómo la modificación de ciertos parámetros afecta al rendimiento de los modelos podría aportar una mayor comprensión de su funcionamiento.
3. Investigar la resistencia de los modelos a las adversidades en el mundo real: Muchas aplicaciones de detección de objetos tienen que lidiar con condiciones desafiantes en el mundo real, como la poca luz, la obstrucción de objetos, las condiciones climáticas adversas, etc. Analizar cómo cada modelo se comporta frente a estos desafíos podría ser de interés.
4. Detección de objetos en vídeos: Investigar la detección de objetos en vídeos, donde se busca identificar y rastrear objetos en movimiento a lo largo del tiempo y entrenar y comparar los modelos con el nuevo conjunto de datos.

Referencias

1. Navnath, N.N, Chandrasekaran, K., Stateczny, A., Sundaram, V.M. & Panneer, P. *Spatiotemporal Assessment of Satellite Image Time Series for Land Cover Classification Using Deep Learning Techniques: A Case Study of Reunion Island, France*, *Remote Sensing*, vol. 14, no. 20, pp. 5232, (2022).
2. Wang, Y., Cui, L., Zhang, C., Chen, W., Xu, Y. & Q. Zhang. *A Two-Stage Seismic Damage Assessment Method for Small, Dense, and Imbalanced Buildings in Remote Sensing Images*, *Remote Sensing*, vol. 14, no. 4, pp. 1012, (2022).
3. Wiam, S., Khouloud, T., Bouchra, H., Nabil, S.M. & K. Adil. *Hybrid Deep Learning Architecture for Land Use: Land Cover Images Classification with a Comparative and Experimental Study*, *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 12, pp. 899-910, (2022).
4. Shi, P., Jiang, Q., Shi, C., Xi, J., Tao, G., Zhang, S., Zhang, Z., Liu, B., Gao, X. & Wu, Q. *Oilwell detection via large-scale and high-resolution remote sensing images based on improved YOLO v4*, *Remote Sensing*, vol. 13, no. 16, pp. 3243, (2021).
5. Krinitskiy, M., Verezemskaya, P., Grashchenkov, K., Tilinina, N., Gulev, S. & Lazara M. *Deep convolutional neural networks capabilities for binary classification of polar mesocyclones in satellite mosaics*, *Atmosphere*, vol. 9, no. 11, pp. 426, (2018).
6. Sakeena, M., Stumpe, E., Despotovic, M., Koch, D. & Zeppelzauer M. *On the Robustness and Generalization Ability of Building Footprint Extraction on the Example of SegNet and Mask R-CNN*, *Remote Sensing*, vol. 15, no. 8, pp. 2135, (2023).
7. Yeşilmen, S. & Tatar, B. *Efficiency of convolutional neural networks (CNN) based image classification for monitoring construction related activities: A case study on aggregate mining for concrete production*, *Case Studies in Construction Materials*, vol. 17, pp. e01372, (2022).
8. La Grassa, R., Cremonese, G., Gallo, I., Re, C. & Martellato, E. *YOLOLens: A Deep Learning Model Based on Super-Resolution to Enhance the Crater Detection of the Planetary Surfaces*, *Remote Sensing*, vol. 15, no. 5, pp. 1171, (2023).
9. Patel, K., Bhatt, C. & Mazzeo, P.L. *Improved Ship Detection Algorithm from Satellite Images Using YOLOv7 and Graph Neural Network*, *Algorithms*, vol. 15, no. 12, pp. 473, (2022).
10. Doğan, F. & Turkoğlu, I. *Comparison of deep learning models in terms of multiple object detection on satellite images*, *Journal of Engineering Research (Kuwait)*, vol. 10, no. 3, pp. 89-108, (2022).
11. Yan, D., Zhang, H., Li, G., Li, X., Lei, H., Lu, K., Zhang, L. & Zhu, F. *Improved method to detect the tailings ponds from multispectral remote sensing images based on faster r-cnn and transfer learning*, *Remote Sensing*, vol. 14, no. 1, pp. 103, (2022).
12. Syed Muhammad Arsalan Bashir & Wang, Yi. *Small object detection in remote sensing images with residual feature aggregation-based super-resolution and object detector network*, *Remote Sensing*, vol. 13, no. 9, pp. 1854, (2021).
13. de Carvalho, O.L.F., de Carvalho, O.A., Albuquerque, A.O., Bem, P.P., Silva, C.R., Ferreira, P.H.G., de Moura, R.S., Gomes, R.A.T., Guimarães, R.F. & Borges, D.L. *Instance segmentation for large, multi-channel remote sensing imagery using mask-RCNN and a mosaicking approach*, *Remote Sensing*, vol. 13, no. 1, pp. 1-24, (2021).
14. Stuparu, D.-G., Ciobanu, R.-I. & Dobre, C. *Vehicle detection in overhead satellite images using a one-stage object detection model*, *Sensors (Switzerland)*, vol. 20, no. 22, pp. 1-18, (2020).

15. Courtrai, L., Pham, M.-T. & Lefèvre, S. *Small object detection in remote sensing images based on super-resolution with auxiliary generative adversarial networks*, *Remote Sensing*, vol. 12, no. 19, pp. 1-19, (2020).
16. Jiang, S., Yao, W., Wong, M.S., Li, G., Hong, Z., Kuc, T.-Y. & Tong, X. *An Optimized Deep Neural Network Detecting Small and Narrow Rectangular Objects in Google Earth Images*, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1068-1081, (2020).
17. Chen, Z., Lu, K., Gao, L., Li, B., Gao, J., Yang, X., Yao, M. & Zhang, B. *Automatic detection of track and fields in China from high-resolution satellite images using multi-scale-fused single shot multibox detector*, *Remote Sensing*, vol. 11, no. 11, pp. 1-19, (2019).
18. Yu, Y., Ai, H., He, X., Yu, S., Zhong, X. & Lu, M. *Ship Detection in Optical Satellite Images Using Haar-like Features and Periphery-Cropped Neural Networks*, *IEEE Access*, vol. 6, pp. 71122-71131, (2018).
19. Amo-Boateng, M., Ekow Nkwa Sey, N., Ampah Amproche, A. & Kyereh Domfeh, M. *Instance segmentation scheme for roofs in rural areas based on Mask R-CNN*, *Egyptian Journal of Remote Sensing and Space Science*, vol. 25, no. 2, pp. 569-577, (2022).
20. Wu, Qifan, Feng, Daqiang, Cao, Changqing, Zeng, Xiaodong, Feng, Zhejun, Wu, Jin & Huang, Ziqiang. *Improved Mask R-CNN for Aircraft Detection in Remote Sensing Images*, *SENSORS*, vol. 21, no. 8, pp. 2618, (2021).
21. Rodriguez-Garlito, E. C., Paz-Gallardo, A & Plaza, A. *Automatic Detection of Aquatic Weeds: A Case Study in the Guadiana River, Spain*, *IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING*, vol. 15, pp. 8567-8585, (2022).
22. Bandyopadhyay, M. *Multi-stack hybrid CNN with non-monotonic activation functions for hyperspectral satellite image classification*, *NEURAL COMPUTING & APPLICATIONS*, vol. 33, no. 21, pp. 14809-14822, (2021).
23. Rubanenko, Lior, Pérez-López, Sebastian, Schull, Joseph & Lapôtre, Mathieu G. A. *Automatic Detection and Segmentation of Barchan Dunes on Mars and Earth Using a Convolutional Neural Network*, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 9364-9371, (2021).
24. Song, X., Hua, Z., & Li, J. *PSTNet: Progressive Sampling Transformer Network for Remote Sensing Image Change Detection*. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 8442-8455, (2022).
25. Maggiori, E., Charpiat, G., Tarabalka, Y., & Alliez, P. *Recurrent Neural Networks to Correct Satellite Image Classification Maps*. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 4962-4971, (2017).
26. Viola, P., & Jones, M. *Rapid object detection using a boosted cascade of simple features*. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. I-511, (2001).
27. Dalal, N., & Triggs, B. *Histograms of oriented gradients for human detection*. in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, (2005).
28. Girshick, R., Donahue, J., Darrell, T., & Malik, J. *Rich feature hierarchies for accurate object detection and semantic segmentation*. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580-587, (2014).
29. Girshick, R. *Fast R-CNN*. in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440-1448, (2015).

30. Ren, S., He, K., Girshick, R., & Sun, J. *Faster R-CNN: Towards real-time object detection with region proposal networks*. in *Advances in neural information processing systems*, pp. 91–99, (2015).
31. He, K., Gkioxari, G., Dollár, P., & Girshick, R. *Mask R-CNN*. in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, (2017).
32. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. *You only look once: Unified, real-time object detection*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, (2016).
33. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. *SSD: Single shot multibox detector*. in *European Conference on Computer Vision*, pp. 21–37, (2016).
34. Lin, T., Goyal, P., Girshick, R., He, K. & Dollár, P. *Focal loss for dense object detection*, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, (2017).
35. J. Smith, *Object detection in computer vision*, *International Journal of Computer Vision*, vol. 75, no. 1, pp. 119–130, (2008).
36. Gonzalez, R. C., Woods, R. E., & S. L. Eddins *Digital image processing using MATLAB*, Gatesmark Publishing, (2004).
37. Russell, S. J., & Norvig, P *Artificial Intelligence: A Modern Approach*, Pearson, (2016).
38. Bishop, C. M. *Pattern Recognition and Machine Learning*, Springer, (2006).
39. Hastie, T., Tibshirani, R. & Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, (2009).
40. Goodfellow, I., Bengio, Y.& Courville, A. *Deep Learning*, MIT Press, (2016).
41. LeCun, Yann, Bengio, Y. & Hinton, G. *Deep learning*, *Nature*, vol. 521, no. 7553, pp. 436–444, (2015).
42. Bochkovskiy, A., Wang, C. Y. & Liao, H. Y. M. *YOLOv4: Optimal speed and accuracy of object detection*, *arXiv preprint arXiv:2004.10934*, (2020).
43. Redmon, J.& Farhadi, A. *YOLOv3: An incremental improvement*, *arXiv preprint arXiv:1804.02767*, (2018).
44. Redmon, J. & Farhadi, A. *YOLO9000: Better, faster, stronger*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, (2017).
45. Chien-Yao Wang, Alexey Bochkovskiy & Hong-Yuan Mark Liao. *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. arXiv:2207.02696. cs.CV. (2022).
46. <https://lamaquinaoraculo.com/computacion/deteccion-de-objetos/>
47. <https://blog.roboflow.com/what-is-a-confusion-matrix/>
48. Roboflow. <https://roboflow.com/>

Integración de un robot móvil en el entorno ROS (Robot Operating System). Análisis de la implementación en diferentes versiones

Sergio García González, Vidal Moreno Rodilla y Francisco Javier Blanco Rodríguez

Departamento de Informática y Automática, Universidad de Salamanca. Plaza de los
Caídos s/n. 37008, Salamanca, España
{sergio.gg,vmoreno,fjblanco}@usal.es

Resumen En los últimos años, el campo de la robótica ha experimentado una evolución exponencial, y el uso de robots móviles se ha vuelto cada vez más común en diversos sectores de la sociedad, como la industria, la medicina, la agricultura y la logística. Por ello, cada vez más los desarrolladores optan por el uso de herramientas o recursos que les permitan un desarrollo de software ágil. Para poder desarrollar software en la robótica, se requiere de herramientas y plataformas de desarrollo software especializadas. Una de las más destacadas y ampliamente utilizadas es el Robot Operating System (ROS).

En esta memoria, se realizará un análisis detallado de las diferentes versiones de ROS, desde las primeras versiones hasta las más recientes. Se examinarán las características y mejoras introducidas en cada una de ellas, así como las posibles limitaciones y desafíos asociados durante la prueba de las mismas.

Además, en este proyecto se abordará el proceso de integración de un robot móvil en las versiones de ROS 1 y ROS 2. Indicando el proceso seguido desde la selección de componentes principales, como la percepción, el control, los recursos software y la planificación de movimiento. En este proceso se explorarán diferentes paquetes, repositorios y herramientas disponibles en ROS que facilitan esta integración.

Keywords: ROS, Robotics, Turtlebot, Mapping, Navigation

1. Introducción

Es una realidad que la robótica se desarrolla actualmente a una velocidad exponencial, y cada vez más, se encuentra presente en nuestra realidad. Dentro de la robótica existen multitud de recursos para generar software, algunos de ellos creados por empresas y que son exclusivamente privados y otros libres. En concreto para el software libre una de las opciones principales usadas por los desarrolladores es el uso del marco de trabajo ROS, que facilita el desarrollo de todo tipo de robótica.

ROS o **Robot Operative System** no es considerado un sistema operativo como tal, pero ejerce en muchos aspectos de forma similar a como lo podría

hacer un sistema operativo convencional. ROS no realiza una gestión de los recursos hardware, ni administra la memoria, ni se encarga de la gestión de interrupciones o el control de periféricos, sin embargo se encarga de facilitar las tareas más esenciales de la robótica como la comunicación y coordinación entre diferentes componentes de un sistema robótico, proporciona un conjunto de herramientas, bibliotecas y convenciones que facilitan la programación y el desarrollo de software para robots y permite la modularización y reutilización del código.

Es por ello que *ROS no es considerado un sistema operativo* en sí mismo, pero realiza acciones de comunicación y coordinación similares a las que realiza un sistema operativo tradicional.

En concreto este marco de trabajo ha tenido dos versiones principales, ROS 1 con la que se ha estado trabajando más de una década tanto a nivel educativo como a nivel empresarial y ROS 2 que es una nueva versión que intenta mejorar aspectos como la seguridad, el rendimiento y la escalabilidad de la robótica con el objetivo de hacer un "sistema operativo" mucho más robusto.

La integración de ROS ha sido fundamental en el desarrollo de la robótica ya que ha permitido que se hayan producido avances significativos en los últimos años. Con este proyecto, se quieren abordar los aspectos claves de la integración de ROS en la robótica, como es el uso de diferentes versiones, la construcción de prototipos, el uso de herramientas o recursos comerciales y la investigación actual en torno a ello.

Además de incidir en la integración de ROS 1 y ROS 2, en este proyecto se va a insistir en el uso de las técnicas de navegación como el **SLAM 2D** (Simultaneous Localization and Mapping) y **3D**, así como la navegación autónoma y la localización precisa en entornos robóticos. Al igual que las técnicas son muy importantes, *el hardware desempeña una tarea crucial* a la hora de la elaboración de un prototipo robótico por lo que es importante resaltar las tareas de elección de recursos hardware del mismo modo.

El SLAM 2D y 3D permite a los robots construir mapas del entorno a medida que se mueven, al mismo tiempo que estiman su propia posición dentro de ese mapa. Esto es esencial para la navegación autónoma, ya que proporciona al robot una comprensión espacial y la capacidad de planificar rutas eficientes para alcanzar destinos específicos. Estas técnicas se basan en el uso de sensores como cámaras, LIDAR (Light Detection and Ranging) y sistemas de odometría para recopilar información del entorno y determinar la posición relativa del robot en tiempo real.

La **construcción de prototipos funcionales** desempeña un papel crucial en la integración de ROS en la robótica. Los prototipos permiten a los desarrolladores probar y validar las funcionalidades implementadas, así como identificar posibles limitaciones o mejoras requeridas. Mediante la construcción de prototipos, se pueden realizar pruebas exhaustivas de la integración de los diferentes componentes de un sistema robótico, como la percepción, la planificación de movimientos, la navegación y la interacción con el entorno.

En conclusión, la integración de ROS 1 y ROS 2 en la robótica impulsa la necesidad de utilizar técnicas avanzadas de navegación, como el SLAM 2D y

3D, y aborda el desafío de la localización precisa. Estos aspectos son esenciales para lograr la navegación autónoma de los robots en entornos reales. Además, la construcción de prototipos funcionales permite a los desarrolladores probar y mejorar las funcionalidades implementadas, garantizando un desarrollo robusto y confiable de los sistemas robóticos. En conjunto, estos avances contribuyen al rompecabezas de implementar una integración exitosa de ROS en la robótica.

2. Estado del arte

Para el desarrollo del estado del arte, se presentará el estado de las investigaciones actuales en cuanto a la integración de ROS en robótica móvil, así como de su implantación en el sector servicios como parte de la continuación del trabajo de fin de grado. Se ha recurrido a uno de los trabajos desarrollados en este máster en Sistemas Inteligentes, en el que se realiza una revisión de la literatura mediante un Mapping en el que se recogen de manera rigurosa y transparente las investigaciones de un campo de forma más concreta y que ayudarán a la elaboración de un estado del arte más completo.

2.1 Revisión sistemática de la literatura sobre la robótica móvil asociada al sector servicios

La robótica móvil se ha convertido en un campo de investigación y de rápido crecimiento en los últimos años. Con avances tecnológicos cada vez más sorprendentes, los robots móviles se están convirtiendo en una presencia común en diversas industrias, especialmente en el sector servicios. Pero antes de sumergirse en la teoría y los conceptos fundamentales de la robótica móvil para poder integrar un robot, es crucial comprender por qué es importante conocer ejemplos de investigaciones en este ámbito, específicamente en el sector servicios que es el campo que ya he tratado.

El sector servicios abarca una amplia gama de áreas, como la atención médica, la logística, la hostelería, la seguridad y muchas más. En cada una de estas áreas, los robots móviles están desempeñando un papel cada vez más importante, ya sea realizando tareas repetitivas, ayudando en la atención al cliente o incluso tomando decisiones complejas basadas en algoritmos avanzados. Estos avances en la robótica móvil están transformando la forma en que se brindan los servicios y tienen un impacto significativo en la eficiencia, la calidad y la experiencia del cliente.

Al conocer ejemplos de investigaciones en robótica móvil aplicada al sector servicios, se pueden comprender mejor los desafíos y las posibilidades al integrar un robot en un marco de trabajo. Estos ejemplos prácticos permiten visualizar cómo los robots móviles están siendo implementados con éxito en situaciones del mundo real, superando obstáculos y mejorando los procesos existentes.

Para ello se ha realizado una revisión del arte inicial cercana al trabajo de integración ya realizado durante el trabajo de fin de grado.

2.1.1 Análisis de la integración robótica en el sector servicios En el ámbito de la robótica aplicada al sector hostelero y sanitario, se han realizado diversos estudios y proyectos relacionados con la implementación y diseño de robots, a menudo basados en plataformas como Turtlebot. Estos artículos exploran la integración de la robótica en entornos como restaurantes, hoteles, hospitales y centros de atención médica, con el objetivo de mejorar la eficiencia, la calidad de los servicios y la interacción con los usuarios. Mediante el análisis de estos artículos, se puede obtener una visión detallada de las tecnologías utilizadas, los desafíos enfrentados y los resultados obtenidos en la aplicación de robots en estos sectores, lo que contribuye a la comprensión y el avance de la robótica en contextos hostelería y sanitarios.

En el artículo *Inferring user intent to interact with a public service robot using bimodal information analysis* [23] se presenta un método novedoso y práctico para predecir las intenciones de interacción de las personas con un robot de servicio público en entornos poblados. A diferencia de las investigaciones tradicionales, que utilizan solo señales visuales para analizar la atención del usuario, este método combina la información de una cámara RGB-D y un láser para percibir al usuario, logrando una percepción de 360 grados y compensando la falta de perspectiva de la cámara RGB-D.

Se extrajeron siete tipos de características de intención interactiva y se entrenó un modelo de regresión de bosques aleatorios para puntuar las intenciones de interacción de las personas en el campo de visión. Además, se diseñó una regla de prioridad para la inferencia de intenciones considerando el orden de dos sensores diferentes. El algoritmo se implementó en un sistema operativo de robot (ROS) y se evaluó en un robot de servicio público. Los resultados experimentales extensos demuestran que el método propuesto permite que los robots de servicio público logren un nivel más alto de cortesía que el enfoque tradicional e interactivo pasivo en el que los robots esperan comandos de los usuarios. Aunque este primer artículo estaba más relacionado con la interacción de los humanos con el robot es importante tener esto en cuenta ya que el sector hostelero o de la restauración es un sector que está en alto contacto con personas.

Es por ello por lo que la investigación de Ruchik Mishra [26] presenta un **robot hostelero** inteligente basado en ROS que simplifica el proceso de registro, uno de los problemas más relevantes en este sector en el que tienen que recibir a multitud de personas. Además, el robot puede planificar rutas con precisión y llegar a las habitaciones en diferentes pisos. También, se agregó un sistema de voz inteligente que brinda asistencia a los clientes. De nuevo se muestra la importancia de la interacción humano máquina que es tan necesaria de potenciar últimamente en la robótica.

Finalmente decidí que uno de los artículos [28] que trataba sobre la aplicación de algoritmos de localización, era relevante ya que utilizaba el mismo robot que había usado para el desarrollo de mi TFG [17]. En el muestra resultados de la implementación de un sistema de localización en el que gracias a un escaneo realizado con una cámara RGB-D que utiliza el algoritmo Monte Carlo, consigue ofrecer unos resultados de localización excelentes.

En el caso de las investigaciones relacionadas con las **robótica y el sector sanitario** parece ser que he encontrado algunas más que para el sector hostelero que está todavía creciendo en este aspecto. Por lo general la mayoría de soluciones robóticas planteadas en los estudios han sido los AGVS. [18]. Los AGVS son la solución más usada ya que la mayoría de empresas que se dedican a la robótica orientada al transporte de productos se enfocan en el desarrollo de estos.

Tras leer algunos de los artículos de forma breve se puede concluir que los robots se utilizan en el transporte sanitario debido a varias razones. En primer lugar, el creciente número de personas mayores [24] y su tendencia a sufrir enfermedades crónicas requieren una atención médica adecuada. Los robots pueden desempeñar un papel fundamental en la entrega de medicamentos a tiempo y en el monitoreo de signos vitales en hogares de ancianos, lo que ayuda a garantizar el cuidado adecuado de los adultos mayores. Además, los robots pueden actuar como intermediarios en la entrega de alimentos, medicamentos y suministros médicos en entornos hospitalarios [29], lo que facilita la gestión logística y libera a los profesionales de la salud para que se centren en tareas más especializadas. Los robots también pueden proporcionar una navegación autónoma basada en inteligencia artificial, lo que les permite moverse de manera eficiente en entornos hospitalarios y cumplir con sus tareas asignadas de manera segura [29] [32] [30] [19], como se ha mencionado en otros artículos.

Para el caso de **robots orientados al sector doméstico**, pero basados en el marco de trabajo ROS y en relación con turtlebot también se han encontrado varios resultados. Encontrando diversos estudios que llevan a cabo una implantación similar a las anteriores en cuanto a hardware y software, pero con un punto de vista distinto [31].

Sin irse más lejos, el uso de robots domésticos como robots de servicio es cada vez más común debido a varias razones. En primer lugar, el envejecimiento de la población y la necesidad de asistencia para personas discapacitadas [20] [21] han generado problemas sociales importantes. Los robots de servicio en el hogar pueden ayudar a mitigar estos problemas al proporcionar asistencia en tareas diarias y cuidado personal.

Además, los robots domésticos están diseñados para interactuar de manera natural con los usuarios humanos, lo que incluye la comprensión y respuesta a comandos de voz, así como la capacidad de interpretar el contexto social y moverse de manera segura y socialmente apropiada en entornos domésticos. Estos robots también pueden brindar asistencia en la realización de tareas cotidianas y ofrecer seguridad, como la detección de fugas de gas [22] o alarmas de detección de movimiento. Aunque todavía existen desafíos y áreas de mejora en el desarrollo de robots domésticos, como la autonomía completa y la realización de inspecciones y servicios en el hogar de manera independiente, los avances en tecnología y la implementación de sistemas basados en ROS han permitido el diseño de robots domésticos de bajo costo con funciones completas y buen rendimiento. [25] [27]

En conclusión, la implementación y el diseño de robots orientados al sector servicios y sanitario han experimentado avances significativos en los últimos años. Los estudios y proyectos llevados a cabo han demostrado el potencial de

la robótica para mejorar la eficiencia operativa, la calidad de los servicios y la interacción con los usuarios en entornos como la hostelería y la atención médica. La utilización de plataformas como Turtlebot ha facilitado el desarrollo de aplicaciones robóticas versátiles y personalizables.

Sin embargo, a pesar de los avances realizados, aún existen desafíos y complicaciones a la hora de encontrar estudios relacionados. La implementación de robots en estos sectores conlleva desafíos técnicos, éticos y regulatorios. La adaptación de los robots a entornos complejos y cambiantes, la garantía de la seguridad y la privacidad de los datos, así como la aceptación y confianza de los usuarios son aspectos críticos a abordar.

Con el tiempo, se espera que la implementación de robots en estos sectores se convierta en una práctica más común, mejorando la calidad de vida de las personas y optimizando los procesos en la industria de servicios y sanitaria.

Por último cabe destacar la falta de artículos relacionados con un tema tan concreto como es la robótica orientada a este tipo de sectores, en los cuales la investigación todavía es breve. También es importante resaltar las palabras claves que se han utilizado, ya que se han usado con el objetivo de acercarse tanto a la robótica desarrollada con el framework ROS como al sector servicios de forma muy concreta por lo que es posible mejorar estos resultados separando ambas búsquedas con una temática concreta y ampliando el rango de búsqueda con otros conceptos pertenecientes a dicha temática. Ya que aunque el diseño e implementación de robots de servicios sea pequeño, aunque en auge, el uso de ROS esta cada vez más extendido en el ámbito de la robótica.

2.2 Revisión sistemática de la literatura sobre la robótica móvil y su integración en ROS

Sin embargo, después de toda la búsqueda de investigación relacionada en torno a ROS, el sector servicios y turtlebot, se quería indagar algo más en la integración de la robótica móvil en el marco de trabajo de ROS, es por ello que el mapping fue de ayuda en los primeros pasos del estado del arte pero la investigación va más allá de este mapping inicial. Para ello se quería sentar unas bases en cuanto a las tecnologías de escaneo utilizadas, las técnicas y algoritmos de navegación autónoma y la localización interior de un robot móvil.

2.3 Técnicas de escaneo

Las **técnicas de escaneo** en ROS, como SLAM [1], escaneo láser [2], escaneo RGB-D [3] y fusionado de nubes de puntos, son fundamentales en la robótica y la percepción del entorno. Estas técnicas permiten a los robots construir mapas tanto bidimensionales como tridimensionales precisos de los interiores, lo que es crucial para la navegación autónoma, la planificación de trayectorias y la interacción segura con el entorno. Los datos capturados por los sensores son procesados y fusionados para crear una representación espacial del entorno en tiempo real. Esto proporciona información vital para la toma de decisiones y la ejecución de tareas complejas en entornos dinámicos.

Las técnicas de escaneo en ROS son importantes porque permiten a los robots comprender su entorno, evitar obstáculos, planificar rutas óptimas y realizar tareas de manera autónoma y eficiente. Entre las técnicas más relevantes de escaneo encontramos las siguientes:

- **SLAM (Simultaneous Localization and Mapping):** SLAM es una técnica esencial en robótica y que permite a cualquier robot móvil construir un mapa del entorno a la vez que se localiza mediante los puntos que está escaneando. En ROS se puede realizar un escaneo 2D o 3D, todo dependerá de los recursos hardware que se utilicen para ello. Además, encontramos diferentes paquetes disponibles para SLAM como son GMapping, Hector SLAM, Cartographer, SLAM Toolbox y otros más. Estos paquetes utilizan sensores como láseres (LiDAR), cámaras y odometría para estimar la posición del robot y construir un mapa.
- **Escaneo 2D:** A diferencia de la técnica de escaneo anterior, podemos encontrar técnicas que nos permitan construir un mapa bidimensional sin necesidad de que el robot sea capaz de localizarse. Para el escaneo 2D de interiores es muy común el uso de sensores infrarrojos o láser como los populares LiDAR A1 o Hokuyo.
- **Escaneo 3D:** Los sensores de escaneo 3D también conocidos como lidars 3D, pueden proporcionar información tridimensional del entorno. Utilizando sensores como láseres 3D, se registran múltiples puntos en el espacio para obtener una nube de puntos tridimensional que representa la forma y la geometría del objeto o entorno escaneado. Los sensores láser 3D emiten haces de luz y miden el tiempo que tarda en reflejarse en las superficies, para luego procesar esos datos y poder crear modelos tridimensionales detallados y precisos.
- **Escaneo RGB-D:** Las técnicas de escaneo que se basan en sensores RGB-D, como la Kinect de Microsoft, el Intel RealSense o la Orbbec Astra [4] proporcionan información de color y profundidad. En ROS, existen paquetes como RGBDSLAM y RTABMap [5] que utilizan sensores para realizar el escaneo 3D del entorno. El escaneo RGB-D se basa en el uso de sensores que combinan una cámara RGB (que captura información de color) y un sensor de profundidad (generalmente un sensor infrarrojo). La información de profundidad capturada permite estimar la distancia de los objetos con respecto al sensor y construir un mapa tridimensional del entorno. Estas técnicas de escaneo no son iguales que las de escaneo 3D aunque puedan parecerse, la diferencia principal es que éstas, utilizan la profundidad recogida por la cámara y las técnicas de escaneo 3D miden el tiempo que tarda la luz en reflejarse.

2.4 Navegación autónoma

La navegación autónoma en ROS es la capacidad fundamental para permitir que cualquier tipo de robot móvil se desplace de forma libre, es decir sin ser teleoperado, en un entorno desconocido. La navegación no es una función de

un robot móvil por si solo si no que se basa en la combinación de sensores como los nombrados anteriormente así como de los algoritmos de percepción y planificación de rutas, y control de movimiento para permitir que los robots naveguen de manera segura y eficiente.

Es importante destacar que la navegación autónoma es un concepto general para cualquier robot móvil, sin embargo si se habla de ROS, tenemos que tener en cuenta que las dos versiones presentan diferencias clave en cuanto a la navegación autónoma:

La primera diferencia, es la más clara cuando se trabaja con las diferentes versiones y es que ROS 1 y ROS 2 tiene distinta arquitectura de comunicación. ROS 1 utiliza un sistema de comunicación basado en mensajes llamado ROS Topic que utiliza un patrón de publicación/suscripción. ROS 2, por otro lado, introduce una arquitectura de comunicación más flexible y escalable llamada ROS 2 Middleware [9] que incluye tanto el patrón de publicación/suscripción como el de servicio/respuesta. Esto lo podemos ver más claramente en la siguiente figura:

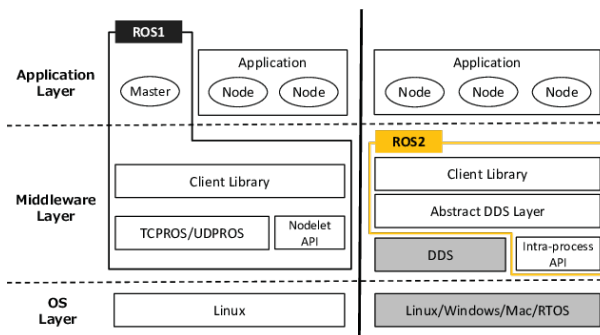


Figura 1: Arquitecturas de comunicación en las diferentes versiones de ROS

No solo se diferencian en cuanto a comunicación estas dos versiones sino también en cuanto a lenguajes de codificación, en ROS 1 se admite principalmente la programación en C++ y Python, mientras que ROS 2, se centra más en Python y otros lenguajes como Java. [10]

Al igual que ya he mencionado anteriormente ROS 2 está más orientado a la escalabilidad por lo que permite que el rendimiento sea mayor y la gestión de recursos sea menor.

En resumen, se puede concluir que como se hace referencia en algunas investigaciones [6] [7], que ROS 2 ofrece una mejor comunicación, una mejor escalabilidad, un mayor rendimiento y una mejora de la confiabilidad que no puede ofrecer ROS 1.

2.5 Tipos de navegación

En la robótica y no solo en ROS, existen varios tipos de navegación que los robots móviles pueden utilizar para desplazarse de un punto a otro dependiendo de diversos factores, es por ello que pueden haber varios tipos.

La navegación de un robot móvil estima su complejidad en función de la cantidad de información de su entorno. Normalmente el entorno en el que se desplaza un robot móvil no es estático ya que siempre se produce algún cambio, por lo que consideramos que su entorno es dinámico, de esta forma podemos considerar dos tipos de navegación:

Navegación reactiva: La navegación reactiva como indica su nombre se basa en las respuestas directas de lo que percibe un robot en su entorno, sin tener un conocimiento previo como es un mapa. Este tipo de navegación consiste en la toma de decisiones de movimiento en función de la información que captan en tiempo real los sensores [11], por lo que permiten respuestas rápidas a cambios producidos en el entorno. Por tanto son robots que tienen una gran autonomía en un entorno pero que carecen de un plan establecido, se centran en lograr objetivos específicos, como alcanzar un punto de interés o explorar un área determinada. Su misión se enfoca en tomar decisiones inmediatas y adaptativas para lograr esos objetivos en función de las condiciones presentes.

Navegación deliberativa: La navegación deliberativa a diferencia de la navegación reactiva se basa en la planificación de alto nivel y en la toma de decisiones anticipadas para lograr un objetivo de navegación. Esta se centra en una planificación mucho mayor a la anterior, una visión más amplia teniendo en cuenta restricciones, objetivos y preferencias.

Uno de los factores que hacen la navegación deliberativa totalmente distinta de la reactiva, es que está incorpora lo que se denomina planificación de trayectorias [12], que permiten la generación de una serie de acciones que permiten desplazarse al robot desde un punto de origen hasta su destino. Estos algoritmos pueden tener en cuenta factores como la eficiencia de la ruta, la evitación de obstáculos, la optimización de recursos o cualquier otra restricción específica. La navegación deliberativa es un tipo de navegación que puede ser muy útil cuando la navegación reactiva no es suficiente en un entorno dinámico.

Si bien la navegación reactiva o deliberativa son unos tipos muy generales que pueden englobar tipos de navegación más específicos en el ámbito de ROS y más concretamente en la librería de ROS *navigation* encontramos otros dos tipos de navegación muy usados en todos los desarrollos y referente a el coste de la navegación en un mapa. Estos dos tipos de navegación son la navegación global y la local [13] que están asociadas con algoritmos de planificación como los mapas de coste.

La navegación global, es también conocida con el nombre de planificación global, esta navegación se encarga de trazar una ruta desde el origen del robot hasta el destino que se haya seleccionado y tiene en cuenta todo su entorno, por ello normalmente es un tipo de navegación que se utiliza cuando tenemos información de un entorno como puede ser un mapa. Para esta navegación se

utilizan algoritmos de planificación de trayectorias como se comentará posteriormente como son el A* (A estrella) o Dijkstra, para calcular una ruta óptima en función de la información que tienen.

La navegación local en cambio no tiene en cuenta el todo sino una porción pequeña de ello, se centra en el control del movimiento y la evitación de obstáculos en un rango de conocimiento pequeño o cercano, es decir local.

2.6 Planificación de trayectorias

La planificación de trayectorias tiene que ver con la capacidad que tiene un robot móvil de trazar una trayectoria desde un origen a un destino, evitando los obstáculos que interfieren la llegada a ese destino y en la que se dispone de una información adicional del entorno y de donde se encuentran algunos de los obstáculos estáticos [14].

En el contexto de ROS (Robot Operating System), la planificación de trayectorias se refiere a la generación de una secuencia de movimientos que permiten al robot desplazarse desde su posición actual hacia un destino deseado, evitando obstáculos en su entorno.

Al generar una trayectoria, los algoritmos de planificación de trayectorias en ROS pueden utilizar diferentes enfoques, como algoritmos basados en métodos de búsqueda (como Dijkstra y A*), grafos (como RRT y PRM) u optimización (como CHOMP). A continuación, en el siguiente apartado se detallarán los métodos o algoritmos de planificación de trayectorias mencionados.

2.7 Localización y escaneo (SLAM)

Anteriormente se ha mencionado algunas técnicas de mapeado que además integran métodos de localización que permiten al robot móvil en todo momento conocer su posición. SLAM [8] es un de los métodos más utilizados y es por ello que se ha usado para el desarrollo de este trabajo fin de máster porque soluciona uno de los problemas más complicados en la robótica móvil que es la localización, problema que ha sido un rompecabezas desde los principios de la robótica.

Básicamente si se toma como ejemplo el movimiento de un roomba, si funcionará sin SLAM se movería de forma descontrolada y sin ningún objetivo, lo que haría ineficiente el objetivo del robot móvil que es limpiar. La tecnología de SLAM en este caso permite al robot móvil conocer información de su entorno como donde se encuentra, por donde ha pasado, las revoluciones de sus ruedas, que no dejan de ser la propia odometría del robot, los obstáculos que hay haciendo que por ejemplo no pase dos veces por una zona por donde ya ha pasado haciendo que su objetivo de limpieza ahora si que sea eficiente. Este ejemplo podemos verlo representado en la siguiente figura:

Sin embargo, este tipo de tecnología de localización simultanea y mapeado no es perfecta y presenta algunos inconvenientes. El primero de ellos es que por lo general los errores de localización se acumulan y es algo inevitable que produce una desviación de los valores reales, es decir que los puntos recogidos por el robot

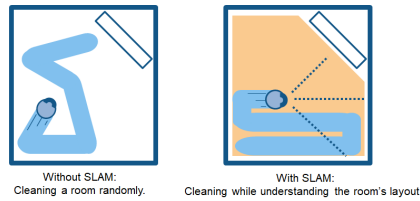


Figura 2: Ejemplo de uso de la tecnología SLAM

no coinciden. Este tipo de errores se pueden mitigar utilizando soluciones que optimicen los datos y minimicen los errores.

El segundo problema que nos encontramos cuando realizamos un escaneo que utiliza una técnica de localización simultánea es que puede fallar momentáneamente la localización ya que el mapeo puede no tener en cuenta las características de movimiento de un robot móvil dando lugar a estimaciones de posición discontinuas.

Para poder realizar un escaneo ROS tiene varios paquetes o herramientas para poder implementar este tipo de métodos. Algunos de los paquetes más usados son Gmapping que es el más usual, Hector SLAM, SLAM ToolBox [15] y Google Cartographer.

SLAM ToolBox es una herramienta de código abierto desarrollada en ROS que proporciona una amplia gama de algoritmos y funciones para la implementación de SLAM en robots. Permite la construcción de mapas 2D y 3D, la estimación de la posición del robot y la generación de trayectorias. Por otro lado, Gmapping es una librería popular en ROS que se enfoca en el mapeo 2D utilizando el filtro de partículas para la localización y actualización del mapa. Hector SLAM es otra opción para el mapeo 2D que utiliza una representación basada en rejillas para la generación de mapas. Finalmente, Google Cartographer es una herramienta de SLAM desarrollada por Google que se destaca por su capacidad para crear mapas 2D y 3D de alta calidad [16], además de ofrecer soporte para localización y navegación simultáneas. Estas herramientas son ampliamente utilizadas en la comunidad de robótica y proporcionan opciones flexibles y potentes para implementar SLAM en una variedad de aplicaciones robóticas.

3. Casos de estudio

Tras la investigación realizada anteriormente en el estado del arte, se procede a definir los diferentes sistemas que se van a ser objeto de estudio. Para esta parte de la experimentación es importante resaltar los recursos hardware que se ha utilizado para la investigación. Durante la misma se han utilizado tanto recursos físicos como recursos preparados para simulaciones y que son los siguientes:

- Robot móvil Turtlebot 2, formado por una plataforma Kobuki y una cámara de profundidad Orbbec astra.
- Robot prototipo creado a partir de un hoverboard

- Robot móvil Turtlebot 3 (burger) simulado
- Roomba iCreate 3 con lidar A1

3.1 Análisis del hardware utilizado

Al usar ciertos componentes o recursos hardware en este proyecto, se ha considerado importante realizar un análisis del hardware utilizado. Esta sección se centra en proporcionar una descripción concisa de los componentes de hardware utilizados en el proyecto, destacando sus especificaciones técnicas y características relevantes. El conocimiento detallado de estos componentes es fundamental para establecer las bases necesarias que permitirán una comprensión profunda de las etapas posteriores del estudio. A través de este análisis, se sentarán los cimientos para una evaluación rigurosa y una interpretación adecuada de los resultados obtenidos en el contexto del sistema de hardware implementado.

Plataforma Kobuki: La plataforma robótica Kobuki es un sistema de hardware diseñado para facilitar el desarrollo de robots móviles de forma rápida y sencilla. Esta plataforma robótica está equipada con ruedas omnidireccionales de tal forma que es considerada como robot holonómico, esto quiere que le permiten un movimiento en todas las direcciones sin girar previamente. También al igual que otras plataformas comerciales similares incluye múltiples sensores como infrarrojos y sensores de caída que permiten que este tipo de plataformas eviten los obstáculos y caídas.

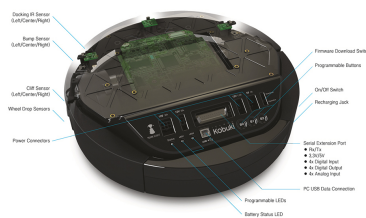


Figura 3: Composición de la plataforma Kobuki

Hoverboard: Para este proyecto también se ha explorado el uso de un Hoverboard como una opción de bajo costo para crear un robot autónomo integrado en ROS. a elección de utilizar un Hoverboard como plataforma se basa en varias ventajas y características clave que ofrece. En la siguiente figura se puede ver la plataforma construida con la placa y los dos motores:

La utilización de la placa controladora y los motores del Hoverboard ha sido posible gracias a un proceso de ingeniería inversa que ha permitido desbloquear y acceder a su funcionalidad. Esto ha brindado la oportunidad de aprovechar la tecnología existente en el Hoverboard y utilizarla para la creación de un robot autónomo. La placa controladora del Hoverboard puede proporcionar la



Figura 4: Prototipo de robot móvil con placa de Hoverboard



Figura 5: Prototipo de robot móvil con placa de Hoverboard (LIDAR A1)

capacidad necesaria para realizar tareas como odometría, control de motores y comunicación con otros dispositivos, permitiendo la integración sin problemas en el entorno de ROS.

Roomba iCreate 3: En esta sección de análisis de hardware, se explora la plataforma robótica Roomba iCreate 3 como una opción destacada para el desarrollo de robots autónomos. La Roomba iCreate 3 es una plataforma diseñada y fabricada por iRobot, reconocida por su línea de robots de limpieza doméstica.

La Roomba iCreate 3 se ha convertido en una elección popular debido a sus características y capacidades versátiles. Además, la Roomba iCreate 3 viene equipada con una serie de sensores integrados, como sensores de infrarrojos y de choque, que permiten la detección de obstáculos y la navegación inteligente en diferentes entornos.



Figura 6: Composición de la Roomba iCreate 3

4. Resultados

4.1 Integración en ROS 1

Para este proyecto se ha integrado dos robots móviles distintos en ROS 1. El primero de ellos ha sido la plataforma Kobuki que ha sido integrada en la

versión Kinectic de ROS 1 y el segundo una placa controladora de Hoveboard que ha servido de controlador de motores en la versión Melodic de ROS 1.

El objetivo con estos dos robots móviles era el mapeo y navegación de un entorno en 2D y 3D. Para poder realizarlo se necesitan una serie de herramientas para el escaneo, como es el SLAM que nos permiten realizar ese tipo de mapeado con diferentes sensores.

Para el primer caso de estudio se ha realizado un escaneo 2D y 3D de una habitación con la cámara de profundidad Orbbec Astra y la herramienta de escaneo Gmapping.

Gmapping es un algoritmo de SLAM para construir un mapa bidimensional. Es por ello que se ha utilizado este algoritmo / herramienta ya que permite generar el mapa del robot móvil utilizando los datos de odometría de las ruedas y la información recogida por la cámara Orbbec Astra. Estos datos son utilizados para alimentar un filtro de partículas denominado *Rao-Blackwellized* para generar el mapa de cuadrículas de RVIZ. Es importante recordar que aunque la cámara escanea una nube de puntos solo utiliza los datos de profundidad al nivel de la cámara para generar el mapa bidimensional. En la siguiente figura 7 se puede ver como se realiza la generación del mapa:

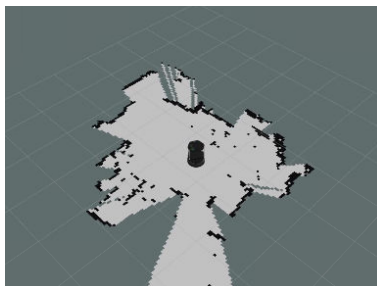


Figura 7: Escaneo 2D con cámara Orbbec Astra

Todo el proceso de escaneo descrito anteriormente se puede visualizar y comprender mejor mediante un grafo que muestra la interconexión entre los diferentes nodos en ROS. En este grafo, se puede observar cómo los nodos de la plataforma, la cámara y las herramientas de mapeo y navegación se comunican entre sí a través de los tópicos y servicios correspondientes. Este grafo de la Figura 8 proporciona una representación visual clara de cómo se establecen las conexiones y fluye la información entre los componentes del sistema, permitiendo una comprensión más completa de la comunicación en ROS.

La interconexión entre los nodos de los grafos es fundamental para el funcionamiento de varias funcionalidades en ROS. En este caso, podemos observar tres conexiones importantes:

Entre los nodos **/camera** y **/scan**: El nodo **/camera** proporciona información visual, mientras que el nodo **/scan** representa la información del sensor láser. Estos nodos se comunican para integrar datos visuales y de escaneo láser,

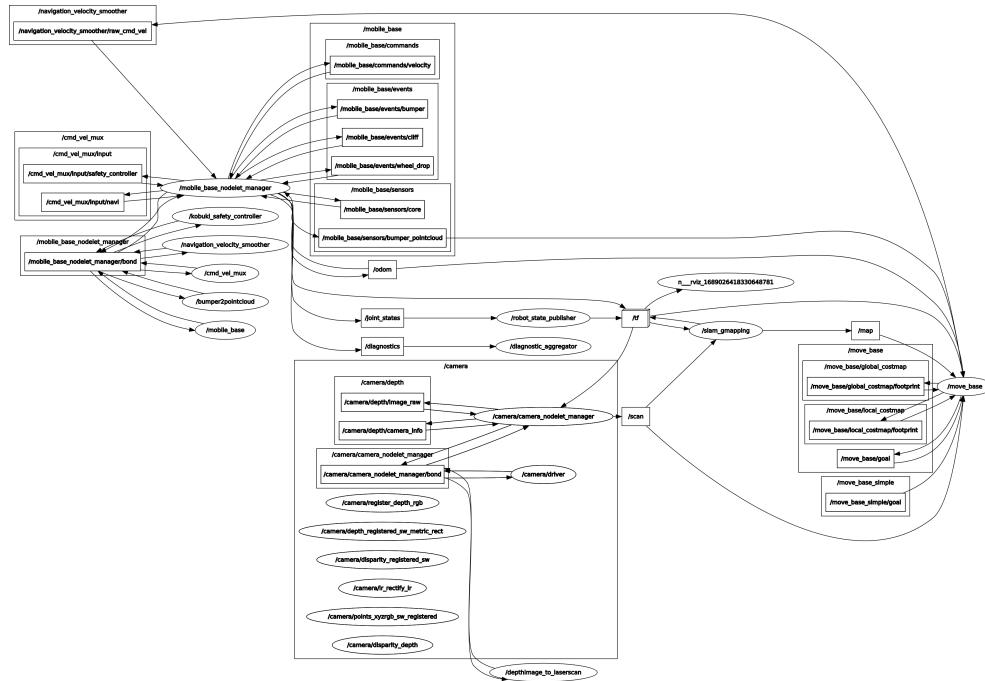


Figura 8: Grafo RQT_GRAPH de los nodos y tópicos ejecutados

lo que permite una mejor percepción del entorno y una generación de mapa más precisa en el nodo **Gmapping**.

Entre el nodo `/odom` y el nodo `/move_base`: El nodo `/odom` recopila información de la odometría del robot, como la posición y la orientación. Esta información es crucial para la planificación y navegación del robot. El nodo `/move_base` es responsable de la planificación de rutas y el control del movimiento del robot. La conexión entre estos nodos permite que la odometría se utilice en el proceso de navegación para estimar la posición y el movimiento del robot.

Entre el nodo `/tf` y el nodo **Gmapping**: El nodo `/tf` es responsable de gestionar las transformaciones entre los diferentes marcos de referencia (como el marco base del robot, el marco de la cámara, etc.). En este caso, se conecta con el nodo `Gmapping` para proporcionar las transformaciones necesarias que permiten al algoritmo de mapeo `Gmapping` entender y ajustar correctamente las posiciones relativas de los objetos en el mapa.

Estas conexiones entre los nodos permiten una comunicación efectiva y una colaboración entre los diferentes componentes del sistema en ROS, lo que facilita el mapeo, la navegación y la percepción del entorno para el robot. Cada nodo desempeña un papel específico y la interconexión entre ellos garantiza un funcionamiento fluido y coordinado del sistema robótico.

En cambio para la generación del mapa en tres dimensiones, se ha utilizado RTAB-Map. Es por ello que tras realizar el escaneo podemos obtener un mapa de tres dimensiones en color, ya que se fusiona la información del color y de la profundidad capturada por la cámara.

En el mapeado 2D, la información se guarda en dos archivos distintos: uno en formato .pgm, que contiene el mapa de ocupación en dos dimensiones, y otro en formato .yaml, que almacena los metadatos relacionados con el mapa, como su resolución y tamaño. Por otro lado, RTAB-Map utiliza una base de datos para almacenar los datos generados durante el proceso de mapeo, lo que permite un almacenamiento más eficiente y la posibilidad de realizar consultas y análisis más complejos de los datos capturados por los sensores del robot durante la exploración. Además RTAB-Map permite la visualización posterior en cualquier momento de los datos capturados, como se puede ver en la siguiente imagen:

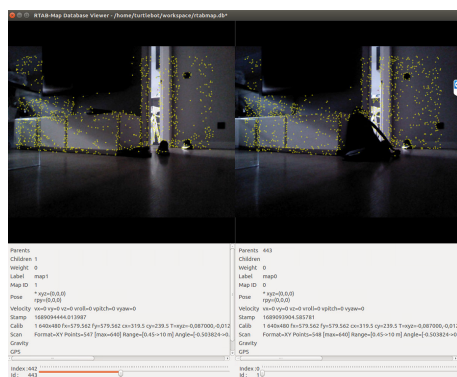


Figura 9: Base de datos de RTAB-Map

En la integración del robot móvil controlado por los motores con una placa de Hoverboard el hecho más destacable es el cambio de la herramienta Gmapping por Google Cartographer. La principal diferencia entre Cartographer y Gmapping radica en los algoritmos y enfoques utilizados. Mientras que Gmapping se basa en un filtro de partículas para generar mapas 2D, Cartographer utiliza un enfoque más avanzado llamado SLAM (Simultaneous Localization and Mapping) basado en optimización de grafo. Esta técnica permite una mejor estimación de la posición del robot y una mayor precisión en el mapeo del entorno. Además, Cartographer es capaz de generar mapas en 3D, lo que brinda una representación más detallada del entorno tridimensional, aunque en este caso no hayamos usado el escaneo 3D.

4.2 Integración en ROS 2

En este otro apartado, se aborda la transición de ROS 1 a ROS 2 y se exploran las mejoras que esta nueva versión ofrece en comparación con su predecesora.

A diferencia de las integraciones en ROS 2, éstas si que han presentado un mayor grado de dificultad debido a la poca experiencia con el sistema y la escasa documentación y comunidad que hay detrás de estas versión. Para este caso también se proponía integrar dos plataformas móviles diferentes, la primera la Kobuki que ya se había conseguido integrar en ROS 1, una simulación y la roomba iCreate 3.

Antes de entrar en el proceso de integración mas manual de este proyecto, se consigue mediante unos paquetes prácticamente olvidados lograr la configuración de la cámara y se consigue realizar una pequeña demostración del típico comportamiento de un robot que centra su objetivo y lo sigue mientras no se salga de su foco principal.

Podemos definir la integración de el robot móvil que usa la plataforma Kobuki en tres iteraciones:

En la primera iteración se utilizan los paquetes de Turtlebot 2 que utilizan la plataforma Kobuki para agilizar el proceso, ya que realizando las configuraciones adecuadas y cuidando que los nodos se comuniquen de forma adecuada, se daba un gran paso para poder utilizar esa información para realizar SLAM con Google Cartographer. Sin embargo, aparece el primer problema y es que el árbol de transformaciones no se completa de forma correcta y por lo tanto el escaneo no puede llegar a hacerse si cada frame no comunica su sistema de coordenadas al mapa.

En la segunda iteración, en vez de usar esos paquetes se inicia una solución más radical y es realizar la construcción de los paquetes de manera manual, eso implica instalar y configurar cada uno de los paquetes necesarios. No solo eso sino que se opta a cambiar la cámara Orbbec Astra por el LIDAR A1 y Google Cartographer por SLAM ToolBox, ya que SLAM ToolBox parecía una herramienta más sencilla para trabajar con el LIDAR. Sin embargo, el problema de las transformaciones sigue persistiendo.

En la tercera y ultima iteración, sin perder el animo, a partir de realizar las transformaciones de manera manual, si que se consigue una solución casi exitosa donde el mapa es escaneado pero solo parcialmente porque de manera discontinua se queda congelado y los nodos dejan de publicar información. Esto *permitted establecer un **TF tree adecuado***. En la figura 10 se puede ver como es un árbol de transformaciones correcto:

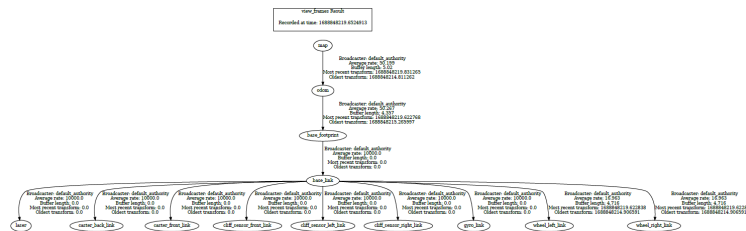


Figura 10: Grafo rqt de ROS 2 del árbol de transformaciones

En esta figura se puede ver como las transformaciones manuales, para el paso de información, que hice para poder interpretarlo en RVIZ se ven de forma correcta. Realmente lo que tenemos que observar es que las transformaciones estén correctamente ordenadas, siendo el padre siempre un frame como map o world, seguido de los datos de odometría y a su vez seguido de el /los frames encargados de los sensores externos.

Los círculos o elipses que se pueden ver en la figura 10 son los nombres de los frames, que son el sistema de coordenadas correspondientes de los componentes del robot y las transformaciones son cada una de las flechas que relacionan cada frame.

Para llevar a cabo la simulación del Turtlebot 3 Burger, se utilizó el entorno proporcionado por Gazebo. Este entorno de simulación, representado como un hexágono virtual, permitió teleoperar y controlar el Turtlebot 3 Burger de forma remota. Gracias al uso del sensor LIDAR A1, se pudo realizar un mapeo completo de este entorno simulado. A continuación en la siguiente figura 11 se puede ver el entorno de simulación de Turtlebot 3:

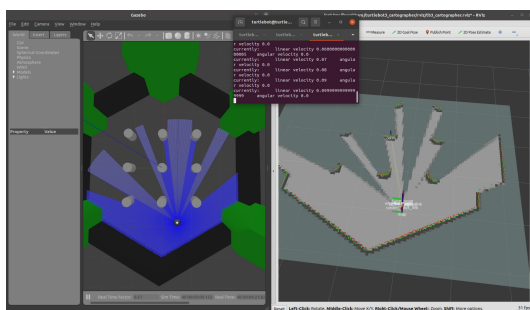


Figura 11: Mapeado del entorno de simulación usando Google Cartographer

Durante la simulación, el LIDAR A1 escaneó el entorno y recopiló datos precisos de distancia y ubicación. Estos datos fueron procesados y utilizados para generar un mapa detallado del entorno en el que se encontraba el Turtlebot 3 Burger. Este mapa proporcionó una representación visual clara de todas las características y obstáculos presentes en el entorno simulado. Al igual que cuando se trata de un robot real, en una simulación obtenemos los dos tipos de ficheros, tanto el fichero con extensión pgm que es el propio como si fichero de configuración con extensión yaml.

Una vez generado el mapa, se pudo utilizar para realizar la navegación autónoma del robot. Con el mapa como referencia, el Turtlebot 3 Burger fue capaz de planificar rutas y evitar obstáculos de manera inteligente mientras se desplazaba por el entorno simulado. Se puede ver en la figura 12 tomada de las simulaciones como el robot traza una ruta hasta el objetivo seleccionado y trata de recorrerla:

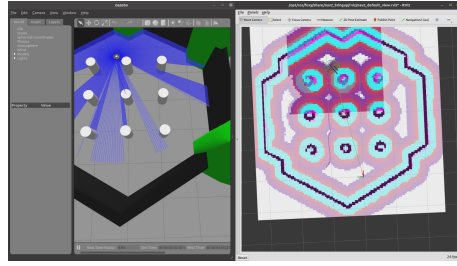


Figura 12: Navegación autónoma del Turtlebot 3 Burger

En resumen, mediante el uso del entorno de simulación proporcionado por Gazebo, se pudo teleoperar el Turtlebot 3 Burger y utilizar el LIDAR A1 para realizar el mapeo del entorno. El resultado fue la generación de un mapa detallado que permitió la navegación autónoma del robot en el entorno simulado. Esta simulación proporcionó una valiosa plataforma de pruebas y desarrollo si se desarrollará el robot real.

La comparación entre la simulación y el mundo real en ROS 2 es crucial para evaluar el rendimiento del sistema. La simulación ofrece un entorno controlado para probar y depurar, pero no captura completamente las complejidades del mundo real. El mundo real presenta desafíos reales, como la incertidumbre del sensor y las limitaciones físicas del robot. Ambos entornos son importantes y se complementan entre sí en la evaluación del sistema en ROS 2.

Por último, la integración con el Roomba iCreate 3 ha sido exitosa en términos de teleoperación y la integración de un Lidar A1 para realizar un escaneo en dos dimensiones del entorno. Estas funcionalidades permitieron obtener datos de distancia y ubicación precisos, lo cual fue un logro importante en el desarrollo del proyecto.

Sin embargo, lamentablemente, se encontraron dificultades al intentar implementar técnicas de SLAM (Simultaneous Localization and Mapping) y navegación en esta plataforma. Uno de los principales desafíos fue la falta de una documentación clara sobre el estado actual de los paquetes utilizados, los cuales eran bastante recientes. Además, se encontraron algunos bugs en los paquetes, lo que dificultó su implementación exitosa. Este tipo de información se puede encontrar en el repositorio de github de Turtlebot 4, ya que hay se muestran la cantidad de Issues relacionados con el mapeo y navegación al utilizar estos paquetes.

5. Conclusiones

A lo largo de este proyecto, se ha llevado a cabo la investigación en cuanto a la integración de un robot móvil en el entorno de ROS (Robot Operating System) lo que ha permitido que se desarrollen pequeñas soluciones que permiten un rápido uso de varias plataformas para el posterior desarrollo de otros sistemas

o arquitecturas. Esta investigación ha permitido aumentar de gran manera el conocimiento sobre este marco de trabajo y allanar el terreno sobre la investigación relacionada sobre todo con las plataformas comerciales como Turtlebot.

Durante este proyecto se han llevado a cabo diversos estudios y se han realizado integraciones de ROS en diferentes plataformas robóticas. Se ha investigado y evaluado técnicas de escaneo o mapeo, así como técnicas de navegación, lo que ha permitido adquirir un mayor conocimiento sobre estos temas.

Se ha logrado integrar tanto ROS 1 como ROS 2 en la plataforma Kobuki (aunque no en su totalidad), así como en un hoverboard se ha conseguido integrar ROS 1, lo que ha ampliado el abanico de posibilidades para realizar experimentos y pruebas. Sin embargo, se ha encontrado que la integración de ROS 2 en el Roomba iCreate 3 ha sido más desafiante debido a la falta de documentación y a la relativa novedad de la versión Humble de ROS 2. Aunque en la versión de ROS 2 Foxy, se ha conseguido grandes avances en ROS 2 todavía queda mucho trabajo en estas nuevas versiones, ya que su alto grado de cambios y de pocas pruebas por parte de la comunidad hacen que sea mucho más difícil conseguir integrar los diferentes componentes hardware.

En términos de estabilidad y documentación, se ha observado que ROS 1 cuenta con una amplia documentación y versiones más estables, mientras que ROS 2 aparenta mejoras no muy notables en los tiempos de navegación, por otra parte si que está maás destina a algoritmos de navegación más precisos. Sin embargo, todavía existe una brecha significativa en cuanto a la documentación y el apoyo de la comunidad, lo que hace que la mayoría de los usuarios continúen trabajando con ROS 1.

Con este trabajo se ha logrado un avance significativo en la comprensión del entorno y el funcionamiento de ROS (Robot Operating System), así como en el dominio de diversas herramientas de SLAM (Simultaneous Localization and Mapping), como gmapping, Google Cartographer y SLAM Toolbox. Esta comprensión más profunda ha permitido que los desarrollos futuros en el campo de la robótica sean mucho más rápidos y sencillos. Ahora, los investigadores y desarrolladores tienen a su disposición una amplia gama de herramientas y bibliotecas que les permiten crear sistemas de navegación y mapeo más eficientes y precisos. Gracias a esto, se ha reducido significativamente el tiempo y los recursos necesarios para implementar soluciones robóticas avanzadas, abriendo las puertas a un nuevo mundo de posibilidades en la automatización y la interacción hombre-máquina.

En conclusión, este proyecto ha sido enriquecedor y ha permitido adquirir conocimientos valiosos sobre ROS, explorar diferentes plataformas robóticas y evaluar las capacidades y limitaciones de las versiones de ROS. Se ha evidenciado el potencial de ROS 2, aunque también se ha destacado la importancia de contar con una comunidad sólida y una documentación exhaustiva para aprovechar al máximo las capacidades de esta versión.

6. Líneas futuras

En cuanto a las líneas futuras de desarrollo de este proyecto, se plantean diversas áreas de enfoque que permitirán seguir avanzando en la mejora de las capacidades y funcionalidades de los sistemas robóticos integrados con ROS.

Una de las líneas de investigación es la aplicación de algoritmos para **determinar qué algoritmo de navegación y mapeo es más adecuado** para cada entorno específico. Esto implica evaluar y comparar diferentes algoritmos, teniendo en cuenta factores como el tamaño del entorno, la presencia de obstáculos y las restricciones de movimiento. Se busca seleccionar y adaptar los algoritmos más óptimos para garantizar una navegación y mapeo eficientes en una amplia gama de escenarios.

Otro aspecto importante es la exploración de la **integración de ROS 2 en la placa controladora del hoverboard**. Esto permitiría aprovechar las capacidades de ROS 2 en esta plataforma, mejorando el rendimiento y la estabilidad de los sistemas implementados en ella.

Además, se propone realizar **benchmarks y pruebas de rendimiento sobre los algoritmos DWA Planners** y otros algoritmos de planificación integrados en ROS. Estas pruebas permitirán evaluar y comparar el rendimiento y la eficacia de los algoritmos en diferentes escenarios y condiciones, proporcionando información valiosa para la selección y optimización de algoritmos de navegación.

Una línea importante de investigación es lograr la implementación e **integración exitosa de la Roomba iCreate 3 en ROS 2**. Esto implica superar los desafíos identificados anteriormente, como la falta de documentación y el soporte limitado de la comunidad para esta plataforma en ROS 2. Una vez lograda la integración, se podrá aprovechar las mejoras y funcionalidades de ROS 2 en la Roomba iCreate 3, ampliando sus capacidades y posibilidades de uso.

Por último, se propone explorar el uso de un planificador de trayectorias como **OMPL (Open Motion Planning Library)** para comparar y evaluar diferentes algoritmos de planificación en entornos complejos. Esto permitirá mejorar la planificación de trayectorias en tiempo real, garantizando movimientos eficientes y seguros para los robots en diferentes contextos y escenarios.

Estos enfoques permitirán continuar mejorando los sistemas robóticos integrados con ROS y explorar nuevas capacidades y aplicaciones en el campo de la robótica.

Referencias

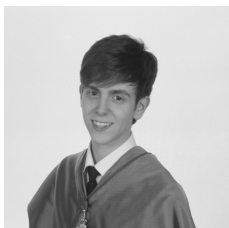
1. Mathieu Labbe, Francois Michaud, *Appearance-based loop closure detection for online large-scale and long-term operation*, *IEEE Transactions on Robotics*, vol. 29, no. 3, 2013, pp. 734–745.
2. Rui Wang, Xiaoge Li, Shuo Wang, *A laser scanning data acquisition and display system based on ROS*, *Proceedings of the 33rd Chinese Control Conference*, 2014, pp. 8433–8437. <https://doi.org/10.1109/ChiCC.2014.6896415>

3. Karl Pauwels, Vladimir Ivan, Eduardo Ros, Sethu Vijayakumar, *Real-time object pose recognition and tracking with an imprecisely calibrated moving RGB-D camera*, 2014 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4271-4278. <https://doi.org/10.1109/IRoS.2014.6942936>
4. Basavanna, M., Shivakumar, M., Prakash, K. R. *Navigation of mobile robot through mapping using Orbbec Astra camera and ROS in an indoor environment*, *Recent Advances in Manufacturing, Automation, Design and Energy Technologies: Proceedings from ICoFT 2020*, Springer, 2022, pp. 465-474.
5. Cerpa, Francisco Javier Bernal *Proyecto Fin de Grado Grado en Ingeniería de las Tecnologías de Telecomunicación Intensificación: Sistemas de Telecomunicación*.
6. Zutell, Joshua M., Conner, David C., Schillinger, Philipp *Motor de comportamiento flexible basado en ROS 2 para navegación flexible*, 2022 *IEEE/SoutheastCon*, 2022, pp. 674-681. <https://doi.org/10.1109/SudesteCon48659.2022.9764047>
7. DiLuoffo, Vincenzo, Michalson, William R., Sunar, Berk *The Transition from ROS 1 to ROS 2*, 1917.
8. Xuexi, Zhang, Guokun, Lu, Genping, Fu, Dongliang, Xu, Shiliu, Liang *Análisis de algoritmo SLAM de robot móvil basado en Lidar*, *Conferencia de control chino (CCC) de 2019*, 2019, pp. 4739-4745. <https://doi.org/10.23919/ChiCC.2019.8866200>
9. Yang, Yuqing, Azumi, Takuya *Exploring real-time executor on ros 2*, 2020 *IEEE International Conference on Embedded Software and Systems (ICESS)*, IEEE, 2020, pp. 1-8.
10. Stavrinou, George *ROS2 for ROS1 users*, *Robot Operating System (ROS) The Complete Reference (Volume 5)*, Springer, 2021, pp. 31-42.
11. Acosta, Gustavo, Gallardo, José, Pérez, Ricardo *Arquitectura de control reactiva para la navegación autónoma de robots móviles*, *Ingeniare. Revista chilena de ingeniería*, 2016, vol. 24, no. 1, pp. 173-181.
12. Goñi, Oscar Enrique, Fernández León, José A, Acosta, Nelson *Arquitectura de un controlador híbrido para navegación robótica en ambientes parcialmente conocidos*, *VIII Workshop de Investigadores en Ciencias de la Computación*, 2006.
13. Pérez Rodríguez, Antonio Javier, et al. *Desarrollo de sistema de navegación para vehículos autónomos terrestres utilizando ROS*, 2017.
14. Padilla Ericksen, A. Implementación y comparativa de algoritmos de control y planificación local para robots móviles utilizando ROS. (Industriales,2017)
15. Macenski, S. & Jambrecic, I. SLAM Toolbox: SLAM for the dynamic world. *Journal Of Open Source Software*. **6**, 2783 (2021)
16. Krinkin, K., Filatov, A., Filatov, A., Huletski, A. & Kartashov, D. Evaluation of modern laser based indoor slam algorithms. *2018 22nd Conference Of Open Innovations Association (FRUCT)*. pp. 101-106 (2018)
17. García González, S. and Others *Desarrollo de una plataforma robótica para la gestión de pedidos en la restauración*.(2022)
18. Palacios Marín, C. *ESTUDIO DE LA APLICABILIDAD DEL TRANSPORTE AUTOMATIZADO PARA EL TRANSPORTE DE MATERIAL EN HOSPITALES*. (Universitat Politècnica de València,2021)
19. Nambiappan, H., Kodur, K., Kyrarini, M., Makedon, F. and Gans, N. MINA: A Multitasking Intelligent Nurse Aid Robot. *ACM International Conference Proceeding Series*. pp. 266 - 267 (2021), <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85109322599&doi=10.1145>
20. Zhi-Jie, L., Ming-Yu, C., Zhi-Ming, Z. and You-Ling, Y. Design and Implementation of Home Service Robot. *Proceedings*

- 2020 Chinese Automation Congress, CAC 2020. pp. 3541 - 3546 (2020), <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85100926603&doi=10.1109>
21. Sanjeeva, E., Herath, K., Madhusanka, B., Priyankara, H. and Herath, H. Understanding the hand gesture command to visual attention model for mobile robot navigation: Service robots in domestic environment. *Cognitive Computing For Human-Robot Interaction: Principles And Practices*. pp. 287 - 310 (2021), <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85129121728&doi=10.1016>
 22. Peng, J., Ye, H., He, Q., Qin, Y., Wan, Z. and Lu, J. Design of Smart Home Service Robot Based on ROS. *Mobile Information Systems*. **2021** (2021), <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85112254344&doi=10.1155>
 23. Li, K., Sun, S., Zhao, X., Wu, J. & Tan, M. Inferring user intent to interact with a public service robot using bimodal information analysis. *Advanced Robotics*. **33**, 369 - 387 (2019), <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85063883693&doi=10.1080>
 24. Salinas-Avila, J., Gonzalez-Hernandez, H., Martinez-Chan, N., Bielma-Avendano, C., Salinas-Molar, X. & Garcia-Garcia, M. Assistant Delivery Robot for Nursing Home using ROS: Robotic prototype for medicine delivery and vital signs registration. *ACM International Conference Proceeding Series*. pp. 141 - 148 (2022), <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85134535422&doi=10.1145>
 25. Xu, C., Li, W., Tan, J., Chen, Z., Zhang, H. & Duan, F. Developing an identity recognition low-cost home service robot based on turtlebot and ROS. *2017 29th Chinese Control And Decision Conference (CCDC)*. pp. 4043-4048 (2017)
 26. Zhang, Y., Wang, X., Wu, X., Zhang, W., Jiang, M. & Al-Khassawneh, M. Intelligent Hotel ROS-based Service Robot. *2019 IEEE International Conference On Electro Information Technology (EIT)*. pp. 399-403 (2019)
 27. Li, K. & Tu, H. Design and Implementation of Autonomous Mobility Algorithm for Home Service Robot Based on Turtlebot. *2021 IEEE 5th Information Technology,Networking,Electronic And Automation Control Conference (ITNEC)*. **5** pp. 1095-1099 (2021)
 28. Mishra, R. & Javed, A. ROS based service robot platform. *2018 4th International Conference On Control, Automation And Robotics (ICCAR)*. pp. 55-59 (2018)
 29. Opha, A., Titaviriya, S. & Piyawattanametha, W. Fully Autonomous Drug and Food Delivery Robot for Hospital and Patient Care Logistic Management based on Artificial Intelligence. *Multidisciplinary Technologies For Industrial Applications*. (2020), <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85124334205&partnerID=40&md5=e309c113d513802d30656bae8c3848f6>, Cited by: 0
 - bibitemKoubâa2016270Koubâa, A., Sriti, M., Javed, Y., Alajlan, M., Qureshi, B., Ellouze, F. & Mahmoud, A. Turtlebot at Office: A Service-Oriented Software Architecture for Personal Assistant Robots Using ROS. *Proceedings - 2016 International Conference On Autonomous Robot Systems And Competitions, ICARSC 2016*. pp. 270 - 276 (2016), <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85010423718&doi=10.1109>
 30. Ribot Lacosta, P. Robot móvil manipulador TiaGo en un entorno hospitalario. (2019)
 31. Pinta, J. Integration of service robots in the smart home. (2017)

32. Koubâa, A., Sriti, M., Javed, Y., Alajlan, M., Qureshi, B., Ellouze, F. & Mahmoud, A. Turtlebot at Office: A Service-Oriented Software Architecture for Personal Assistant Robots Using ROS. *Proceedings - 2016 International Conference On Autonomous Robot Systems And Competitions, ICARSC 2016*. pp. 270 - 276 (2016).

Autores



Álvaro Gómez Gutierrez, graduado en Ingeniería Informática por la Universidad de Salamanca, ha continuado su formación especializándose en áreas clave de la tecnología moderna, completando un Máster en Seguridad en Internet y un Máster Universitario en Sistemas Inteligentes, en la misma institución. Actualmente, está finalizando un Máster Universitario en Dirección en Proyectos Informáticos y Servicios Tecnológicos en la Universidad

Pontificia de Salamanca. En el ámbito profesional, ha contribuido al desarrollo de soluciones tecnológicas en reconocidas empresas como Viewnext, Mecalux e Irium, destacándose por su enfoque en la calidad y la innovación. En su posición actual como Software & Data Engineer en Hiberus, aplica sus habilidades avanzadas en el diseño y desarrollo de soluciones de software, reafirmando su compromiso con la excelencia tecnológica.

Álvaro Lozano Murciego, es actualmente Profesor Contratado Doctor en el área de Lenguajes y Sistemas Informáticos. En cuanto su formación académica, es Ingeniero Técnico en Informática de Sistemas, Graduado en Ingeniería Informática, ha realizado el Máster de Sistemas Inteligentes y es Doctor en Ingeniería Informática. Ha sido beneficiario de contratos predoctorales y postdoctorales en su etapa formativa y ha participado en diferentes proyectos Europeos Nacionales y regionales a lo largo de los últimos años. Es miembro del grupo de investigación Expert Systems and Application Lab (ESALAB) y colaborador en el Grupo de Minería de Datos (MIDA). Sus líneas de investigación incluyen el machine learning aplicado a diferentes áreas, sistemas de recomendación, sistemas de optimización, desarrollo de sensores inalámbricos en IoT y edge computing.



Belén Curto Diego, Licenciada en Ciencias Físicas (Electrónica y Automática) y Doctora en Ciencias por la Universidad de Salamanca. Tras un periodo de tres años en el sector privado en la empresa ENUSA Industrias Avanzadas, se incorpora a la Universidad de Salamanca, por su interés en el ámbito docente e investigador. Pertenece al área de Ingeniería de Sistemas y Automática, donde imparte docencia en titulaciones de Grado y Máster en materias

relacionadas con los Sistemas Inteligentes, Robots Autónomos, Automática y

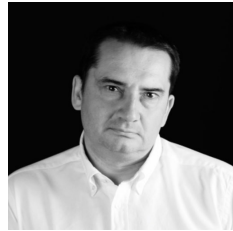
Control y Computadores. Ha ocupado cargos de dirección académica de Postgrado, Máster y Doctorado. Respecto a su labor investigadora, es codirectora del GIR: Robótica y Sociedad”, donde ha dirigido más de 7 Tesis Doctorales y TFMs sobre Robótica y Sistemas Inteligentes. Ha sido directora y ha participado en multitud de proyectos de investigación de ámbito nacional y autonómico financiado por entidades públicas. Además, ha sido directora y componente del equipo investigador de más 40 proyectos de transferencia tecnológica e innovación industrial con empresas privadas del sector cárnico, nuclear, médico, seguridad, medioambiental, etc. Cuenta con tres (3) sexenios de investigación reconocidos y uno (1) de Transferencia. Es autora de más de 100 publicaciones en revistas, comunicaciones a congresos, presentaciones en jornadas, etc. así como revisora de diversas revistas y editora de 2 números especiales relacionados con la Robótica Colaborativa. También, es autora de tres (3) patentes en explotación.

Belén Pérez Lancho, es Licenciada y Doctora en Ciencias Físicas por la Universidad de Salamanca. Realizó su formación postdoctoral en la Universidad Paris VI (curso 1996-97) y desde 1998 es Profesora Titular en el Área de Ingeniería de Sistemas y Automática de la Universidad de Salamanca. Imparte o ha impartido docencia en las titulaciones de Ingeniería Informática, Física, Estadística y en el Máster en Sistemas Inteligentes. Perteneció al grupo de investigación BISITE y ha participado en más de 20 proyectos de investigación, principalmente en aplicaciones de sistemas de control y sistemas multiagente, colaborando en varias publicaciones científicas, en la dirección de tres tesis y de numerosos proyectos de fin de carrera. Ha ocupado cargos de gestión en la Facultad de Ciencias como Secretaria Académica (años 2000 a 2004) y como Vicedecana (años 2008 a 2016).



David Cruz García, actualmente, es Personal Investigador en Formación por la Universidad de Salamanca, siendo beneficiario de un contrato predoctoral Baco Santander en la convocatoria 2022-2023. Actualmente doctorando en el Doctorado de Ingeniería Informática y, graduado en el Grado de Ingeniería Informática y Máster Universitario en Sistemas Inteligentes en la Universidad de Salamanca. Durante su formación del grado, máster y doctorado, ha colaborado como Personal Investigador con el Laboratorio de Sistemas Expertos y Aplicaciones (ESALab) de la misma universidad, donde ha participado en varios proyectos de investigación relacionados principalmente con la robótica.

Francisco Javier Blanco Rodríguez, es Licenciado en Ciencias Físicas y Doctor por la Universidad de Salamanca desde el año 2003. En la actualidad es Profesor Contratado Doctor en el Área de Ingeniería de Sistemas y Automática en el Departamento de Informática y Automática de la Universidad de Salamanca. Imparte varias asignaturas de robótica tanto en titulación de Grado como de Máster. En cuanto a la investigación está interesado en temas de Robots autónomos, robótica educativa y nuevas tecnologías aplicadas en la educación.



Francisco José Peñalvo, es Catedrático de Universidad del Departamento de Informática y Automática en la Universidad de Salamanca (USAL), con 4 sexenios de investigación, 1 sexenio de transferencia y 5 quinquenios docentes reconocidos. Recibió el premio Gloria Begué a la excelencia docente en 2019 y el premio María de Maeztu a la excelencia investigadora en 2023. Además, fue Profesor Distinguido de la Escuela de Humanidades y Educación del Tecnológico de Monterrey, México entre 2016 y 2018 y es Investigador de Impacto Internacional de la Universidad Nacional San Agustín, Arequipa, Perú. Desde 2006 es el director del Grupo de Investigación Reconocido por la USAL GRIAL (GRupo de investigación en InterAcción y eLearning), grupo que es Unidad de Investigación Consolidada de la Junta de Castilla y León (UIC 81). Ha dirigido 30 tesis doctorales. Ha sido Vicedecano de Innovación y Nuevas Tecnologías de la Facultad de Ciencias de la USAL entre 2004 y 2007 y Vicerrector de Innovación Tecnológica de esta Universidad entre 2007 y 2009. Actualmente es el Subdirector del Instituto Universitario de Ciencias de la Educación (IUCE), el Delegado del Rector para la Docencia Digital y el Coordinador del Programa de Doctorado en Formación en la Sociedad del Conocimiento de la USAL. Es Editor Jefe de la revista Education in the Knowledge Society. Editor Asociado en Computers in Human Behavior, Computers in Human Behavior Reports, IEEE Access, entre otras, habiendo sido editor asociado de las revistas IEEE RITA e IEEE Transactions on Learning Technologies. Ha publicado más de 100 artículos en revistas indexadas en el JCR SCIE/SSCI (61 Q1).

Gabriel Villarrubia González, Doctor de la Universidad de Salamanca en el Departamento de Informática y Automática. Ingeniero Informático por la Universidad Pontificia de Salamanca (2011), habiendo estudiado el Máster en Sistemas Inteligentes en la Universidad de Salamanca (2012). Además, ha realizado el Máster Universitario en Seguridad e Internet (2014) y el Máster en Dirección de Sistemas de Información (2015). A lo largo de su formación ha seguido una línea de investigación bien definida, centrada en la aplicación de los sistemas multi-agente a los entornos de inteligencia ambiental, con especial atención a la definición de arquitecturas inteligentes y a la fusión de información. Ha participado en más de 30 proyectos de investigación, colaborando en la obtención de 15 propiedades intelectuales, habiendo realizado estancias internacionales. En el ámbito científico, es autor de más de 40 publicaciones científicas y codirector de 50 proyectos de fin de grado. Es miembro del comité científico de diferentes congresos, siendo miembro del comité organizador de ISAMI, PACBB, MIS4TEL, PAAMS y DCAI. A nivel profesional fue nombrado Microsoft Student Partner en el año 2010. Ha ganado diferentes premios en el desarrollo de aplicativos móviles. Actualmente centra sus esfuerzos en diseñar una arquitectura multi-agente especialmente diseñada para la fusión dinámica de información en entornos inteligentes caracterizados por estar dotados de sensores heterogéneos que capturan información del entorno. Actualmente es personal docente investigador en el Departamento de Informática.



Geovanny Satama Bermeo, natural de Ecuador, ha obtenido su título de Graduado en Ingeniería Eléctrica y Electrónica (2021) por la Universidad de San Francisco de Quito. Enfocando su trayectoria académica hacia el ámbito de la informática contemporánea, actualmente curso el Máster en Sistemas Inteligentes (2023) en la Universidad de Salamanca. Actualmente, se encuentra trabajando como ingeniero QA para la empresa HP SCDS.

Henar López Huertas, oriunda de Toledo, 1998. Tiene un background en el ámbito de la Ingeniería Industrial y Electrónica, habiendo obtenido su grado en esta materia por la Universidad de Castilla La Mancha (2021). Complementándolo posteriormente, profundizando sus conocimientos en Inteligencia Artificial, habiendo cursado el Máster en Sistemas Inteligentes de la Universidad de Salamanca (2022), donde desempeña su labor investigadora, enfocando sus esfuerzos en los sistemas de recomendación, siendo el culmen de



esto su TFM, donde expuso mejoras en el sistema de recomendación de Spotify empleando múltiples algoritmos de Machine Learning. Actualmente, desempeña su carrera profesional en INDRA, trabajando en proyectos de ámbito internacional de aviónica e ingeniería electrónica, mientras complementa su formación con curso de diversas disciplinas, como son las metodologías ágiles, desarrollo de radares, etc.



Iria Ares Torres, natural de Reboiras, Galicia, es Graduada en Ingeniería Electrónica y Automática (2022) por la Universidade da Coruña. Ha dirigido su formación al campo de la informática actual, a través del Máster en Sistemas Inteligentes (2023) en la Universidad de Salamanca. Actualmente, a través de una beca de formación, cursa un Máster en Productividad y Desarrollo Personal por la Universidad Camilo José Cela mientras realiza prácticas en el

departamento de Network Delivery de Endesa Medios y Sistemas, entorno en el que inicia su etapa laboral.

Juan Francisco de Paz Santana, Doctor en Informática y Automática por la Universidad de Salamanca en 2010. Ingeniero técnico en Informática y de Sistemas (2003), Ingeniero Informático (2005) y Diplomado en Estadística (2007) por la Universidad de Salamanca. Actualmente es Catedrático de Universidad, director de departamento en el Departamento de Informática y Automática de la Universidad de Salamanca y coordinador del programa de



doctorado en Ingeniería Informática por la Universidad de Salamanca. Además, ha realizado el Máster Universitario en el Desarrollo de Sistemas para el Comercio Electrónico y el Máster Universitario en Animación Digital. Ha obtenido diferentes premios a nivel académico como el Premio de Grado, Premio de doctorado y en otros ámbitos como un premio de SUN Microsystems y otro de la Junta de Castilla y León. En la actualidad trabaja en el campo de los sistemas distribuidos basados en sistemas multi-agente, la Inteligencia Artificial, la bioinformática, la inteligencia ambiental, internet de las cosas, entre otros. Ha participado en 48 proyectos de investigación a nivel regional, nacional e internacional. Es coautor de más de 180 artículos publicados en diversas revistas de ámbito nacional e internacional, workshops y simposios, 50 de estas publicaciones en revistas con índice de impacto. Ha sido organizador de numerosas conferencias y además ha sido revisor en diferentes conferencias y revistas de ámbito internacional en el área de las Ciencias de la Computación e Inteligencia Artificial. Es actualmente el director del Departamento de Informática y Automática, director del Grupo de Investigación Reconocido ESAL, miembro del grupo IBSAL y socio de la AEPIA. Evaluador de la ANECA desde 2010 y FONCyT.



Juan Manuel Otero Otero, natural de Salamanca, Castilla y León, ha obtenido su título de Graduado en Ingeniería Informática (2022) por la Universidad de Salamanca. Enfocando su trayectoria académica hacia el ámbito de la informática contemporánea, actualmente se encuentra inmerso en el Máster en Sistemas Inteligentes (2023) en la misma universidad. Actualmente, se encuentra trabajando como programador backend para la empresa Axpe

Consulting.

María Dolores Muñoz Vicente, es Contratada Doctor en el Departamento de Informática y Automática de la Universidad de Salamanca y miembro del Grupo de Investigación en Minería de Datos (mida.usal.es). Su investigación está centrada en dos líneas de investigación: los sistemas de información geográficos y la Minería de datos en diferentes escenarios.



Maria N. Moreno García, es Catedrática de Universidad en el Departamento de Informática y Automática de la Universidad de Salamanca. Actualmente es directora del Grupo de Investigación en Minería de Datos (mida.usal.es) y de la Cátedra Extraordinaria Viewnext- Universidad de Salamanca. Ha sido coordinadora del Programa de Doctorado en Ingeniería Informática de la misma Universidad desde 2013 hasta 2020. Sus intereses de investigación se

centran en el área del aprendizaje automático y su aplicación en varios dominios, especialmente en medicina, medios sociales y fundamentalmente en el campo de los sistemas de recomendación. En estos ámbitos, ha colaborado con grupos de investigación punteros, habiendo realizado estancias de investigación en el Intelligent System Lab de la Universidad de Bristol (Reino Unido) y en el College of Computing and Digital Media de la Universidad DePaul de Chicago, Estados Unidos. Ha organizado diferentes ediciones de sesiones especiales sobre "Web mining and recommender systems" sobre "Web and Social Media Mining.^{en} conferencias internacionales. Es miembro del consejo editorial de revistas internacionales, así como editora invitada de varios números especiales y revisora de revistas indexadas en posiciones relevantes en WoS Journal Citation reports. También es autora de numerosos artículos publicados en revistas y conferencias de reconocido prestigio.

María Navarro Cáceres, Doctora en Informática y Musicología por la Universidad de Salamanca, comenzó a trabajar en la universidad en 2012 con una beca de colaboración. Posteriormente, pudo obtener su doctorado a través de un contrato predoctoral dentro del programa de formación del personal universitario (FPU) promovido por el Ministerio de Educación y Cultura (MECD). Actualmente, está vinculada a la Universidad de Salamanca como



Profesora Titular de Universidad. La mayor parte de la carrera investigadora de la solicitante, con más de 50 publicaciones de revista y congreso ha estado orientada al estudio y análisis, creación e innovación de algoritmos evolutivos y técnicas de aprendizaje aplicadas al campo artístico, en concreto, la música. Ha participado en proyectos educativos, proyectos IoT y proyectos sobre energía (DREAM-GO, SURF, etc.), entre otros. Ha sido investigadora principal en el Proyecto “Co-POEM: Platform for the Collaborative Generation of Popular Music” financiado por la comisión europea bajo el programa “Erasmus+: KA2-Asociaciones estratégicas para la Educación Escolar”. Además, también ha sido investigadora principal del proyecto “FolkAI: Preservation and Dissemination of the European Folk Music through Artificial Intelligence” del programa “Europa Investigación, financiado por el Ministerio de Ciencias e Innovación. Actualmente es Investigadora Principal del proyecto “DIGIFOLK: An European and Ibero-American Approach for the Digital Collection, analysis and dissemination of folk music”; financiado por el programa Marie-Curie Staff Exchanges con el ánimo de hacer intercambios internacionales de personal para mejorar las competencias investigadores en música tradicional y nuevas tecnologías.



Mario Francisco Sutil, es licenciado en Ciencias Físicas, especialidad Electrónica, por la Universidad de Salamanca (España) en 1997, y Doctor en Ciencias en 2011. Trabaja en el Departamento de Informática y Automática de la Universidad de Salamanca desde 2000, y actualmente es Profesor Contratado Doctor. Sus áreas de investigación de interés son el Control Avanzado de Procesos, en particular Control Predictivo Basado en Modelos, Control

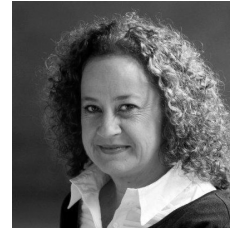
Distribuido basado en sistemas multiagente, y Control Inteligente. Las aplicaciones son 263 principalmente sistemas integrados de aguas residuales y otros procesos de gran escala. Ha trabajado en más de 20 proyectos de investigación, con numerosas publicaciones en revistas indexadas SCI.

Nastaran Shoeibi, natural de Irán, ha obtenido su título de Graduado en Ingeniería Informática (2018) por la Universidad de BNUT (Babol Noshirvani University of Technology). Enfocando su trayectoria académica hacia el ámbito de la informática contemporánea, ha cursado el Máster en Sistemas Inteligentes (2023) en la Universidad de Salamanca. Actualmente es investigadora de IA en el Grupo de Investigación GRIAL, y su objetivo es avanzar en la tecnología educativa.



Oscar Aponte, es graduado en Electrónica Industrial y Automática por la Universidad de Valladolid. Ha realizado el Máster en Sistemas Inteligentes por la Universidad de Salamanca en la cual se encuentra ahora mismo realizando el doctorado en Ingeniería informática siguiendo como línea de investigación el Control Inteligente donde destaca la aplicación del Aprendizaje por Refuerzo en sistemas de control.

Pastora Isabel Vega Cruz, es Catedrática de Universidad del área de Ingeniería de Sistemas y Automática y lidera el Grupo de Investigación Reconocido de la USAL denominado "Supervisión y Control de Procesos". La actividad del mismo se centra en proponer soluciones eficientes para la optimización y el control de procesos industriales, integradas en los niveles más altos de la jerarquía del control y con el objeto de incrementar la seguridad, la fiabilidad y la calidad en la operación y en el control en las industrias de procesos. Concretamente, se investigan soluciones teóricas y prácticas, para la optimización global de la operación, el control avanzado de los procesos y la supervisión inteligente de los mismos. Entre las líneas de investigación más destacadas se encuentran: Control Predictivo Distribuido basado en sistemas multiagente, Optimización, supervisión y control de EDARs y Síntesis y diseño integrado de procesos. Las aplicaciones industriales más importantes incluyen los Sistemas Integrados de Aguas y los Sistemas de Poligeneración.

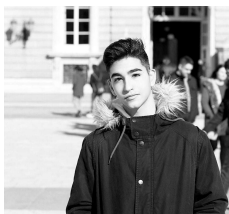
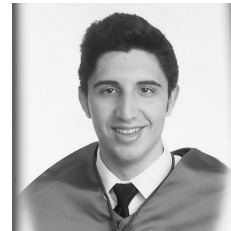




Roberto Therón Sánchez, cursó sus estudios de Informática en la Universidad de Salamanca (Diplomatura) y la Universidad de la Coruña (Licenciatura). Tras entrar a formar parte del Grupo de Investigación en Robótica de la Universidad de Salamanca, presentó su trabajo de Tesis, "Cálculo paralelo del espacio de las configuraciones para robots redundantes", recibiendo el Premio Extraordinario 265 de Doctorado. Posteriormente ha obtenido los títulos

de Licenciado en Comunicación Audiovisual (Universidad de Salamanca) y Licenciado en Humanidades (Universidad de Salamanca). En la misma Universidad de Salamanca continúa realizando su trabajo de investigador, como encargado del grupo VisUsal (dentro del Grupo de Investigación GRIAL) que se centra en la combinación de enfoques procedentes de la Informática, Estadística, Diseño Gráfico y Visualización de Información, para obtener una adecuada comprensión de conjuntos de datos complejos. En los últimos años, se ha dedicado al desarrollo de herramientas de visualización avanzada para datos multidimensionales, como por ejemplo datos genéticos o paleoclimáticos. En el área de Analítica Visual desarrolla productivas colaboraciones con grupos e instituciones de reconocido prestigio internacional, como el Laboratorio de Ciencias del Clima y del Medio Ambiente (París), la Real Academia Española o la Academia de Ciencias de Austria. Ha sido coordinador del Programa de Doctorado en Ingeniería Informática de la USAL (2012-2017) y actualmente es el Director del Máster Universitario en Sistemas Inteligentes (desde 2012).

Samuel Adrados González, natural de Salamanca, Castilla y León, ha obtenido su título de Graduado en Ingeniería Informática (2021) por la Universidad de Salamanca. Enfocando su trayectoria académica hacia el ámbito de la informática contemporánea, actualmente se encuentra inmerso en el Máster en Sistemas Inteligentes (2023) en la misma universidad. Actualmente, se encuentra trabajando para el grupo de investigación BISITE en las instalaciones del Air Institute.



Sergio Fernández Marcos, natural de Salamanca, Castilla y León, ha obtenido su título de Graduado en Ingeniería Informática (2022) por la Universidad de Salamanca. Enfocando su trayectoria académica hacia el ámbito de la informática contemporánea, actualmente se encuentra inmerso en el Máster en Sistemas Inteligentes (2023) en la misma universidad. Actualmente, se encuentra tra-

bajando en el desarrollo de software con la empresa CarttecLab orientada al sector retail.

Sergio García González, actualmente estudiante del programa de doctorado de Ingeniería Informática en la Universidad de Salamanca. Sergio curso sus estudios de Ingeniería informática en la Universidad de Salamanca, donde también ha cursado el máster oficial en Sistemas Inteligentes. Durante su período de formación como estudiante de grado y de postgrado ha colaborado intensamente, como Personal Investigador, con los proyectos relacionados del grupo de investigación ESALAB (Laboratorio de Sistemas Expertos y Aplicaciones) de la universidad de Salamanca. En la misma Universidad de Salamanca continúa realizando su trabajo de investigador (dentro del Grupo de Investigación ESALAB) donde se centra en los campos de la robótica, dispositivos IoT y IA.



Vidal Moreno Rodilla, es Licenciado en Ciencias Físicas en la rama de Electrónica y Doctor en Ciencias en 1996. Actualmente es Profesor Titular de Universidad del área de Ingeniería de Sistemas y Automática de la Universidad de Salamanca. Ha impartido docencia sobre Robótica, Inteligencia Artificial y Control en niveles de Grado, Máster y Doctorado tanto en Salamanca como en el extranjero. Miembro fundador del Grupo de Investigación Reconocido “Robótica y Sociedad”, es autor de varias decenas de publicaciones en revistas internacionales fruto de la dirección de una centena trabajos de investigación que incluye tesis doctorales, Trabajos de Fin de Master, etc.

Vivian López Batista, es profesora titular de la Universidad de Salamanca en el área de Ciencias de la Computación e Inteligencia Artificial. Doctorada en Informática por la Universidad de Valladolid en 1996. Miembro del Grupo de Minería de Datos. Ha realizado investigación en diferentes campos como procesamiento del lenguaje natural, redes neuronales y minería de datos. Tiene 80 artículos publicados en revistas de reconocido prestigio, talleres y actas de conferencias, 20 libros y capítulos de libros y 20 informes técnicos, la mayoría de ellos en estos temas. Miembro del comité organizador y científico de varios simposios internacionales. Fue directora del Máster en Sistema Inteligente y del Programa de Doctorado en Informática y Automática de la Universidad de Salamanca desde junio de 2010 hasta octubre de 2012.

