Trabajo de Fin de Grado

Herramienta de etiquetado de imágenes mediante aprendizaje activo

Acitve learning-based image labeling tool

Memoria

VNiVERSiDAD D SALAMANCA

CAMPUS DE EXCELENCIA INTERNACIONAL

Alumno

José María Martín Ramos

Tutores

Juan Francisco de Paz Santana

Gabriel Villarrubia González

Septiembre, 2023

VNiVERSiDAD D SALAMANCA
CAMPUS DE EXCELENCIA INTERNACIONAL

Facvltad D Ciencias
VNiVERSiDAD D SALAMANCA

Certificado del/los tutor/es TFG

D. Gabriel Villarrubia González y D. Juan Francisco De Paz Santana profesores del Departamento de Departamento de Informática y Automática de la Universidad de Salamanca

HACEN CONSTAR:

Que el trabajo titulado "Herramientas de etiquetado de imágenes mediante aprendizaje activo", ha sido realizado por José María Martín Ramos, con DNI. 70909366K y constituye al memoria del trabajo realizado para la superación de la asignatura Trabajo Fin de Grado en Ingeniería Informática en esta Universidad.

Salamanca, 7 de septiembre de 2023

VILLARRUBIA GONZALEZ GABRIEL - 71932845T

Firmado digitalmente por VILLARRUBIA GONZALEZ GABRIEL - 71932845T
Fecha: 2023.09.07 20:45:38 +02'00'

Fdo.: _____

DE PAZ SANTANA JUAN FRANCISCO - 76125754D

Firmado digitalmente por DE PAZ SANTANA JUAN FRANCISCO - 76125754D
Nombre de reconocimiento (DN): c=ES, serialNumber=IDCES-76125754D, givenName=JUAN FRANCISCO, sn=DE PAZ SANTANA, cn=DE PAZ SANTANA JUAN FRANCISCO - 76125754D
Fecha: 2023.09.07 17:58:59 +02'00'

Fdo.: _____

Firmado digitalmente por MARTIN RAMOS JOSE MARIA - 70909366K
Fecha: 2023.09.07 20:51:04 +02'00'

Resumen

En este documento se presentan los apartados clave del plan de proyecto de software.

La inteligencia artificial es un campo que está ganando cada vez más y más popularidad. Ya está presente en nuestra vida diaria, alcanzado unos avances espectaculares en todos los campos, tales como la medicina, investigación o industria.

Una de sus ramas más conocidas es la visión artificial, que pretende que las máquinas consigan percibir y comprender las imágenes y actuar de la forma que se les indique.

En esta rama se encuadran algoritmos de clasificación de imágenes, entre otros. Uno de los métodos que mejores resultados da en estos casos de uso son las redes neuronales convolucionales, que son redes neuronales artificiales cuyos campos receptivos son muy similares a los de la corteza visual primaria (V1) de un cerebro biológico.

Estas redes neuronales convolucionales requieren de un entrenamiento que necesita una ingente cantidad de imágenes etiquetadas por un ser humano con el resultado esperado del algoritmo. Dado que la tarea de etiquetado es larga y tediosa, en este trabajo de fin de grado se propone un sistema que la facilite mediante el uso de técnicas como el aprendizaje activo.

Gracias a estas técnicas el sistema realizará el etiquetado de forma semi-automática, de manera que los trabajadores humanos sólo necesiten etiquetar un pequeño porcentaje de las imágenes. El sistema debe permitir la creación de distintos conjuntos de datos que podrán contener un número arbitrario de imágenes.

Este trabajo de fin de grado se enfoca en el desarrollo de una aplicación Web donde los usuarios podrán clasificar sus fotos de forma inteligente.

La página web está diseñada siguiendo el estilo de Material-UI, una librería Open Source que implementa Material Design de Google, lo que nos va a permitir realizar un sistema responsivo que se adapte a números dispositivos como tabletas inteligentes o móviles.

En cuanto a la funcionalidad de esta plataforma comprende desde la subida de nuevas imágenes hasta su correcta clasificación, pasando por la generación de modelos y dando uso del aprendizaje activo a la hora de determinar el conjunto de imágenes necesarias a clasificar por el usuario.

El almacenamiento de los datos de usuarios, conjuntos de imágenes y modelos se delega parcialmente a una base de datos de MongoDB, la otra parte de los datos, como su interpretación se alberga en un servidor backend.

Se ha elaborado una memoria junto con seis anexos complementarios para documentar este proyecto. Estos anexos proporcionan información detallada sobre la planificación y la ingeniería del software del proyecto, así como manuales de ayuda destinados tanto a los programadores como a los usuarios del sistema. De esta manera, se garantiza una documentación exhaustiva y completa del proyecto.

Summary

This document presents the key sections of the software project plan.

Artificial intelligence is a field that is gaining increasing popularity. It is already present in our daily lives, achieving spectacular advancements in various fields such as medicine, research, and industry.

One of its most well-known branches is computer vision, which aims to enable machines to perceive and understand images and act according to instructions.

Within this branch, there are image classification algorithms, among others. One of the methods that yields the best results in such cases is convolutional neural networks, which are artificial neural networks whose receptive fields closely resemble those of the primary visual cortex (V1) of a biological brain.

These convolutional neural networks require training that involves a substantial number of images labelled by humans with the expected algorithm outcome. Since labelling is a lengthy and tedious task, this bachelor's thesis proposes a system that facilitates it through the use of techniques such as active learning.

Thanks to these techniques, the system will perform labelling semi-automatically, reducing the need for human workers to label only a small percentage of the images. The system should allow the creation of different datasets that can contain an arbitrary number of images.

This bachelor's thesis focuses on the development of a web application where users can intelligently classify their photos.

The website is designed following the Material-UI style, an open-source library that implements Google's Material Design. This allows us to create a responsive system that adapts to various devices, such as tablets and smartphones.

Regarding the functionality of this platform, it ranges from uploading new images to their proper classification, including model generation and the use of active learning to determine the set of images that the user needs to classify.

The storage of user data, image datasets, and models is partially delegated to a MongoDB database, while the other part of the data, including its interpretation, is hosted on a backend server.

A report, along with six complementary appendices, has been prepared to document this project. These appendices provide detailed information about project planning and software engineering, as well as user and programmer manuals. This ensures comprehensive and complete documentation of the project.

Índice de contenido

Índice de tablas

Índice de figuras

1. Introducción

En la era digital actual, la Inteligencia Artificial (IA) se ha convertido en una herramienta fundamental en una amplia gama de aplicaciones y sectores, desde la medicina hasta la industria del entretenimiento. La capacidad de las máquinas para aprender y tomar decisiones basadas en datos ha revolucionado la forma en que interactuamos con la tecnología y ha abierto nuevas posibilidades para la automatización de tareas complejas. Sin embargo, la efectividad de los modelos de IA depende en gran medida de la calidad de los datos utilizados para su entrenamiento.

En este contexto, el Aprendizaje Activo ha emergido como una estrategia esencial para mejorar y optimizar los modelos de IA. A diferencia del aprendizaje pasivo convencional, donde los modelos se entrenan con conjuntos de datos estáticos y etiquetados de antemano, el Aprendizaje Activo permite a los modelos seleccionar activamente las instancias de datos que consideran más informativas para su entrenamiento, lo que resulta en modelos más precisos y eficientes.

En este artículo académico, exploraremos el emocionante mundo de la Inteligencia Artificial apoyada por el Aprendizaje Activo y su aplicación en el desarrollo de una innovadora aplicación web. A lo largo de estas páginas, desentrañaremos los conceptos clave de la Inteligencia Artificial y el Aprendizaje Activo, examinaremos cómo se integran en el proceso de desarrollo de aplicaciones web y exploraremos casos de uso concretos que ilustrarán la utilidad y el impacto de esta poderosa combinación.

Nuestro objetivo es proporcionar una visión completa de cómo la IA y el Aprendizaje Activo pueden impulsar la creación de aplicaciones web inteligentes, adaptativas y altamente efectivas. A lo largo de este viaje, desafiaremos las fronteras de la innovación tecnológica y presentaremos ejemplos concretos de cómo esta sinergia está transformando la forma en que interactuamos con el mundo digital.

Al combinar los principios del aprendizaje activo, las redes neuronales convolucionales y los conocimientos extraídos de la investigación académica, este trabajo de fin de grado tiene como objetivo contribuir al desarrollo de un sistema novedoso que simplifique el proceso de etiquetado de imágenes y mejore la eficiencia de la clasificación inteligente de fotos.

La memoria del proyecto se divide en varios apartados. En primer lugar, se exponen los objetivos del proyecto, tanto los finales como los intermedios, así como los personales del autor. A continuación, se incluye un apartado de conceptos teóricos para definir y describir brevemente los términos utilizados en el documento.

En la sección sobre las técnicas y herramientas utilizadas durante el desarrollo del proyecto, se enumeran y explican brevemente cada una de ellas, así como, la razón de su elección.

En el punto de las fases del desarrollo se explica que el trabajo se ha llevado a cabo en el marco del Proceso unificado, que cuenta con un marco de trabajo genérico ha adaptado a una importante variedad de software.

El apartado de aspectos relevantes del desarrollo recoge las diferentes fases del proyecto, desde la planificación hasta la implementación y el despliegue final.

En las conclusiones y líneas de trabajo futuras se exponen los resultados obtenidos durante el proyecto y se sugieren posibles áreas de mejora o desarrollo futuro.

Finalmente, se incluye una sección de referencias y bibliografía, donde se citan todas las fuentes consultadas o utilizadas en el proyecto.

Además del documento principal, se adjuntan seis anexos:

El Anexo 1 aborda la planificación temporal, estimación de costes y esfuerzo del proyecto.

El Anexo 2 contiene la especificación de requisitos software, detallando los componentes del sistema, objetivos, requisitos de información, requisitos de restricción de información, requisitos no funcionales, actores y casos de uso.

El Anexo 3 se centra en el análisis del sistema, presentando una primera versión de la arquitectura y desarrollando el modelo de dominio, la vista arquitectónica y la realización de los casos de uso en análisis.

El Anexo 4 se enfoca en el diseño del sistema, presentando una solución para implementar el proyecto. Se desarrolla el modelo de diseño, la vista arquitectónica, el modelo de despliegue y la realización de los casos de uso en diseño.

El Anexo 5 es el manual del programador, que describe la estructura del código y proporciona detalles sobre las diferentes clases y carpetas del sistema.

Por último, **el Anexo 6** es el manual del usuario, que describe la funcionalidad de la aplicación, las principales páginas y las diferentes funcionalidades del sistema, con el objetivo de guiar al usuario a través de la interfaz.

2. Objetivos

El desarrollo de esta aplicación pretende hacer frente a un conjunto de objetivos tanto a nivel técnico como personal.

2.1. Objetivos del sistema

Los objetivos principales que se persiguen con este proyecto son:

- **Gestión de usuarios**. En primer lugar, el sistema debe permitir el alta y baja de usuarios en el sistema, así como gestión del registro, permitiendo realizar cambios de la contraseña asociada a un determinado usuario. Además, debe controlar que únicamente el conjunto de usuarios determinados pueda acceder a los diferentes recursos, para que no se vea comprometida la seguridad de los datos.

- **Gestión de imágenes**. Es uno de los objetivos básicos y principales del sistema, los usuarios deben poder agregar nuevas imágenes en el sistema y así mismo poder acceder a ellas sin ningún tipo de restricción. Será a partir de este punto donde se desencadenen las diferentes acciones relacionadas a las mismas, como puede ser la colocación, características visuales y clasificación entre otras.

- **Gestión de conjunto de imágenes (dataset).**
  El dataset es el contenido de una tabla dentro de una base de datos que posee diferentes columnas, en donde se van almacenando registros en cada una de sus filas.
  Desde el momento que un usuario es registrado en el sistema, este puede crear un nuevo dataset donde deberá adjuntar imágenes como hemos comentado en el punto anterior. Los usuarios asignarán un nombre a cada dataset y estos deben de permanecer en el tiempo. El usuario registrado una vez haya clasificado una cantidad de imágenes podrá pasar, por tanto, a la generación del modelo asociado a este conjunto de datos. Con dicho modelo generado y almacenado en la parte del servidor, el usuario podría generar la predicción de aquellas imágenes que aún no han sido clasificadas o subir nuevas imágenes al conjunto sin necesidad de generar un nuevo modelo para ser, estas nuevas, asignadas una correspondiente etiqueta. Otro punto importante en el desarrollo del sistema es el derecho por parte del usuario a eliminar una dataset del sistema, así como todas sus imágenes asociadas.

2.2. Objetivos técnicos

Para la realización de este proyecto se hace necesario el conocimiento en tres áreas técnicas bien definidas.

Inteligencia Artificial

La Inteligencia Artificial (IA) es un campo multidisciplinario que ha experimentado un crecimiento exponencial en las últimas décadas. Su desarrollo se ha visto influenciado por avances en la informática, la neurociencia y la estadística, y ha tenido un impacto significativo en diversas áreas, desde la medicina hasta la conducción autónoma.

Será la Inteligencia Artificial la que formalice el centro neurálgico de la aplicación y a partir de la cual se desplegará la misma. Contiene la generación de modelos para cada una de las datasets, la selección inteligente de las imágenes a clasificar y la ejecución de las predicciones. Para ello, se implementará la librería Keras dentro de Python junto con TensorFlow.

Evolución Histórica de la Inteligencia Artificial

Los Primeros Pasos

El concepto de IA se remonta a la antigüedad, pero su formalización y desarrollo como disciplina moderna comenzaron en la década de 1950. En la *Figura 1. Test de Turing* se puede observar el "Test de Turing", propuesto por Alan Turing en 1950, que planteó la pregunta de si una máquina podría exhibir un comportamiento inteligente indistinguible del de un ser humano. En ese período, los investigadores se centraron en el uso de la lógica y los algoritmos para resolver problemas.

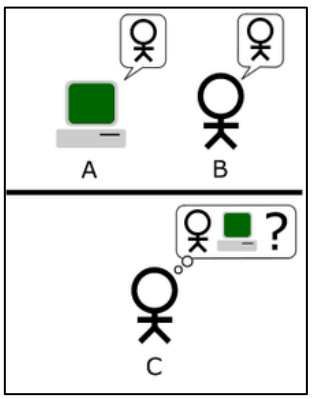

Figura 1. Test de Turing

Alan Turing, un matemático británico, es considerado uno de los padres fundadores de la IA. Su trabajo pionero en la computación y la teoría de la computabilidad sentó las bases para la creación de las primeras computadoras digitales y, eventualmente, para la IA. El "Test de Turing" sigue siendo un tema de debate en la comunidad de IA y plantea cuestiones profundas sobre la inteligencia y la simulación de la misma.

Los sistemas expertos

En las décadas de 1960 y 1970, la IA se centró en sistemas basados en reglas y conocimiento experto, cuyo esquema se observa en la *Figura 2. Diagrama Sistema Experto* Los sistemas expertos fueron pioneros en la resolución de problemas específicos utilizando una base de conocimientos codificada. Aunque son efectivos en dominios limitados, estos sistemas carecen de la capacidad de razonamiento y adaptación generalizada.

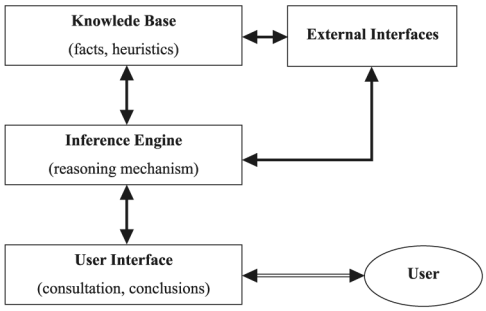

Figura 2. Diagrama Sistema Experto

Los sistemas expertos, como Dendral y MYCIN, se convirtieron en ejemplos emblemáticos de aplicaciones tempranas de la IA. Dendral, desarrollado en la década de 1960 en la Universidad de Stanford, fue diseñado para el análisis químico y demostró la capacidad de la IA para abordar problemas complejos en dominios especializados. MYCIN, desarrollado en la década de 1970 en la Universidad de Stanford, se centró en el diagnóstico médico y mostró cómo la IA podría ayudar a los médicos a tomar decisiones informadas.

El Auge del Machine Learning

A medida que avanzaba la investigación en IA, surgió una rama especializada, conocida como Machine Learning o Aprendizaje Automático. El Machine Learning se centró en el desarrollo de algoritmos y técnicas que permitieran a las computadoras aprender a partir de datos y mejorar su rendimiento en tareas específicas sin programación explícita.

Aprendizaje Supervisado y No Supervisado

El Aprendizaje Supervisado implica entrenar un modelo utilizando un conjunto de datos etiquetado, donde se conocen las respuestas correctas. El modelo aprende a realizar predicciones basadas en ejemplos de entrenamiento. En cambio, el Aprendizaje No

Supervisado se basa en datos no etiquetados y busca patrones y estructuras en los datos sin guía externa.

Aprendizaje Profundo (Deep Learning)

En la década de 1980, surgió un subcampo del Machine Learning conocido como Aprendizaje Profundo (Deep Learning). Se caracteriza por el uso de redes neuronales artificiales con múltiples capas interconectadas, conocidas como redes neuronales profundas. Estas redes se inspiran en la estructura del cerebro humano y tienen la capacidad de aprender representaciones jerárquicas de datos.

Capas de Redes Neuronales en Deep Learning

Las redes neuronales profundas constan de varias capas como se observa en la *Figura 3. Capas de Red Neuronal Profunda* :

1. Capa de Entrada. Recibe los datos de entrada y los propaga a través de la red.
2. Capas Ocultas. Son capas intermedias que realizan transformaciones no lineales de los datos. Cuantas más capas ocultas tenga una red, mayor será su capacidad para aprender representaciones complejas.
3. Capa de Salida. Genera las predicciones o resultados finales.

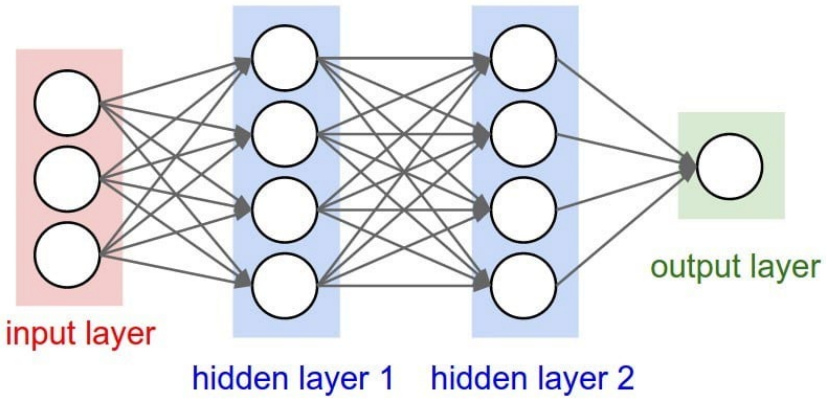

Figura 3. Capas de Red Neuronal Profunda

El Deep Learning ha demostrado ser altamente efectivo en aplicaciones como el reconocimiento de voz, la visión por computadora y el procesamiento del lenguaje natural.

Machine Learning y Deep Learning en la Actualidad

Aplicaciones de Machine Learning

El Machine Learning se ha convertido en una tecnología clave en muchas aplicaciones de la vida cotidiana. Algunos ejemplos notables incluyen:

- Motores de Búsqueda

  Los motores de búsqueda, como Google, utilizan algoritmos de Machine Learning para clasificar y mostrar resultados de búsqueda relevantes para las consultas de los usuarios.

- Recomendaciones de Contenido

  Plataformas de entretenimiento, como Netflix y Spotify, utilizan algoritmos de Machine Learning, para recomendar películas, series y música a los usuarios en función de sus preferencias.

- Diagnóstico Médico

  El Machine Learning se utiliza en aplicaciones médicas para ayudar en el diagnóstico de enfermedades a través del análisis de imágenes médicas, como resonancias magnéticas y tomografías computarizadas.

Ética y Desafíos de la Inteligencia Artificial

A medida que la IA ha avanzado, también ha planteado importantes cuestiones éticas y sociales. Es fundamental abordar estos problemas para garantizar que la IA se desarrolle de manera responsable y beneficiosa para la humanidad. Algunos de los desafíos y consideraciones éticas más destacados incluyen:

- Desigualdad y Sesgo

  Los algoritmos de IA pueden heredar sesgos de los datos de entrenamiento, lo que puede llevar a resultados sesgados y discriminación. Es crucial abordar este sesgo y garantizar que los sistemas de IA sean justos y equitativos.

- Privacidad y Seguridad

  El uso de datos personales en aplicaciones de IA plantea preocupaciones sobre la privacidad y la seguridad. Las empresas y organizaciones deben tomar medidas para proteger los datos y garantizar la confidencialidad de la información del usuario.

- Empleo y Automatización

  La automatización impulsada por la IA tiene el potencial de cambiar la naturaleza del trabajo en muchas industrias. Es importante considerar cómo la IA afectará el

empleo y cómo podemos prepararnos para una economía impulsada por la tecnología.

- Responsabilidad y Ética

La cuestión de la responsabilidad de las decisiones tomadas por sistemas de IA es un tema importante. ¿Quién es responsable cuando un algoritmo toma una decisión incorrecta o perjudicial? Para abordar esta preocupación, es esencial establecer normativas y directrices éticas.

Futuro de la Inteligencia Artificial

El futuro de la Inteligencia Artificial es prometedor y lleno de posibilidades. A medida que la tecnología continúa avanzando, es probable que veamos avances en áreas como la IA general, la computación cuántica aplicada a la IA y la integración más amplia de la IA en la vida cotidiana.

- IA General

La búsqueda de la IA general, que puede razonar y aprender en múltiples dominios como los seres humanos, es un objetivo de investigación de largo plazo. Si se logra, podría tener aplicaciones revolucionarias en la resolución de problemas complejos y en la toma de decisiones en una amplia gama de industrias.

- Computación Cuántica y IA

La computación cuántica tiene el potencial de acelerar significativamente los cálculos necesarios para entrenar modelos de IA. Esto podría llevar a avances más rápidos en la investigación de IA y al desarrollo de modelos aún más poderosos.

- IA en la Vida Cotidiana

Es probable que la IA se integre cada vez más en la vida cotidiana, desde asistentes virtuales en el hogar hasta sistemas de salud personalizados. Esto podría mejorar la comodidad y la eficiencia en muchas áreas de la vida.

La evolución de la Inteligencia Artificial, desde sus raíces en la lógica hasta la creación de redes neuronales y la expansión hacia el Machine Learning y el Deep Learning, refleja un campo en constante evolución y desarrollo. La IA está cambiando la forma en que vivimos y trabajamos, y su impacto seguirá siendo profundo y duradero en los años venideros.

API REST

La REST API, o API Representational State Transfer (Transferencia de Estado Representacional, en español), es una interfaz de programación de aplicaciones (API) diseñada siguiendo los principios y restricciones de la arquitectura REST. Esta arquitectura se basa en la idea de representar recursos y permitir su manipulación a

través de un conjunto de operaciones bien definidas. Las APIs RESTful se han convertido en un estándar ampliamente utilizado para desarrollar servicios web que son eficientes, escalables y altamente interoperables.

La principal función de la REST API en el contexto de una aplicación web es actuar como el puente de comunicación entre la parte visual de la página web, que normalmente es la interfaz de usuario que los usuarios finales ven y utilizan, y el backend del sistema. Esta API permite que las solicitudes y respuestas se gestionen de manera eficiente y estructurada, facilitando la transferencia de datos y la ejecución de acciones en el servidor.

La modularidad y la reutilización son dos ventajas clave que ofrece la REST API. Al dividir las funcionalidades en endpoints claramente definidos, se crea un sistema modular en el que cada recurso o servicio puede ser accesado y utilizado de forma independiente. Esto significa que una determinada funcionalidad de la API se puede emplear de forma parcial o total en el desarrollo actual, y se puede extender o adaptar en el futuro sin afectar el funcionamiento de otras partes del sistema.

La modularidad promovida por la REST API también facilita la colaboración en el desarrollo de aplicaciones, ya que diferentes equipos o desarrolladores pueden trabajar en paralelo en diferentes aspectos de la aplicación web, cada uno interactuando con la API de manera independiente y utilizando los endpoints que son relevantes para su tarea. [1]

Página Web

Es el punto principal de la aplicación, pues será a través de la cual el usuario interactué con el sistema a desarrollar. Centraremos el desarrollo de la interfaz en los diagramas de Material-UI, para que el usuario tenga una sensación de continuidad entre las diferente ventanas y pantallas de la aplicación.

Para el desarrollo de este punto haremos uso de los lenguajes como JSX, HTML y CSS, para definir la estructura y funcionalidad de la página Web. Todas las acciones deberán conectarse con la base de datos presente en MongoDB haciendo uso de la correspondiente librería. [3]

2.3. Objetivos personales

Los principales objetivos de este proyecto son profundizar los conocimientos relacionados con el desarrollo de un proyecto en todas sus dimensiones, para poder llegar a darles uso en la carrera profesional y la creación de servicios REST para comunicar las dos partes del sistema.

Ampliando los conocimientos de la asignatura de Fundamentos en Sistemas Inteligentes, se realizará el desarrollo de una Red Neuronal Residual encargada de la clasificación de las imágenes.

3. Conceptos teóricos

Base de datos no relacional

Una base de datos no relacional es una base de datos que no utiliza el esquema tabular de filas y columnas que se encuentra en la mayoría de los sistemas de bases de datos tradicionales. En su lugar, las bases de datos no relacionales utilizan un modelo de almacenamiento optimizado para los requisitos específicos del tipo de datos que se está almacenando. [4]

Las **bases de datos no relacionales** son un sistema de almacenamiento de información que usan el lenguaje SQL como apoyo. Por ello también se les suele llamar **NoSQL** o «no solo SQL».

Tienen una gran escalabilidad y están pensadas para la **gestión de grandes volúmenes de datos**. Son más actuales que las relacionales, y su desarrollo se ha basado en la necesidad de crear sistemas de gestión capaces de trabajar con **datos no estructurados** o semi-estructurados.

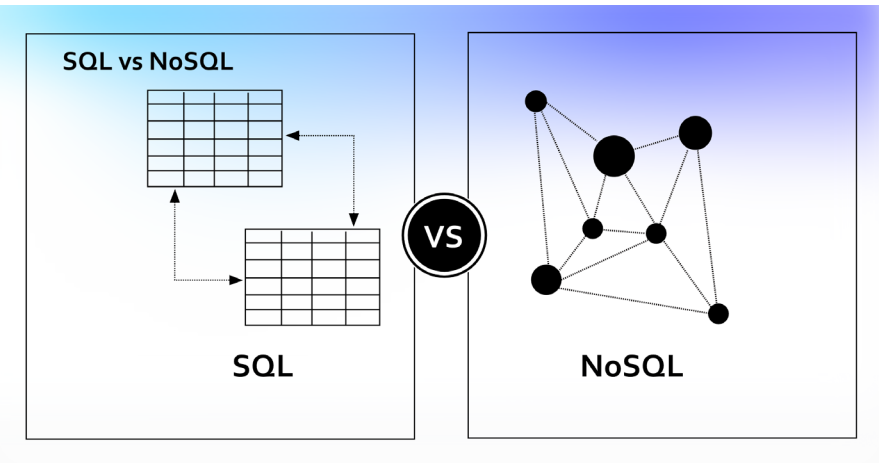

Figura 4. Diferencias BBDD SQL vs NoSQL

En la *Figura 4. Diferencias BBDD SQL vs NoSQL* se puede observar las discrepancias entre las dos tipologías de Bases de datos y ver que las NoSQL no trabajan con estructuras definidas. **Los datos no se almacenan en tablas**, ni la información se organiza en registros o campos.

Progressive Web App

Las Progressive Web Apps (PWAs) son aplicaciones web construidas y mejoradas con APIs modernas para ofrecer capacidades mejoradas, confiabilidad e instabilidad, llegando a cualquier persona, en cualquier lugar, desde cualquier dispositivo, todo con un único código base.

Las PWA reales constan de un **protocolo HTTPS** encriptado, uno o varios *service workers*, un archivo de manifiesto y un tiempo de carga rápido debido a su arquitectura central.

Frontend

El frontend es todo lo que un usuario ve e interactúa cuando hace clic en un enlace o escribe una dirección web. La dirección web también se conoce como URL (Uniform Resource Locator) y determina qué página web se debe cargar y mostrar en el navegador.
Es la parte del lado del cliente de una aplicación web.

Backend

Un backend es un sistema utilizado para ejecutar un sitio web o una empresa, como sistemas de gestión de pedidos, procesamiento de inventario y suministro. Este sistema recopila información de los usuarios u otros sistemas de procesamiento de datos.

De igual manera, es el encargado de gestionar todos los datos que introduce el usuario.

Por lo tanto, el backend es cualquier sistema que soporta aplicaciones de “back office”, que se usan como parte de la gestión social y funciona con los datos facilitados por el usuario y los aportados por otros sistemas, para dar respuestas.

Middleware

Middleware es un software y servicios en la nube que proporciona servicios y capacidades comunes a las aplicaciones, y ayuda a los desarrolladores y operadores a construir e implementar aplicaciones de manera más eficiente.

El middleware actúa como el tejido conectivo entre aplicaciones, datos y usuarios.
Se localiza encapsulando a la aplicación como se observa en la *Figura 5. Esquema de Middleware* y será a través de ella por donde pasen siempre las peticiones y las respuestas de la aplicación.

Agiliza el desarrollo de aplicaciones y acelera la comercialización, facilitando la conexión de aplicaciones que no han sido diseñadas para conectarse entre sí y proporcionando la funcionalidad para conectarlas de manera inteligente.

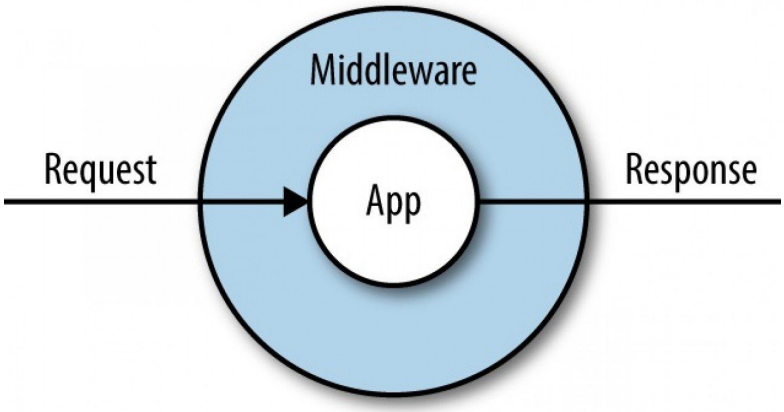

Figura 5. Esquema de Middleware

Hay muchos tipos de middleware. Algunos, como los intermediarios de mensajes se centran en un tipo de comunicación. Otros, como los servidores de aplicaciones web proporcionan el rango completo de prestaciones de conectividad y comunicación necesarias para crear un tipo determinado de aplicación.

Framework

Un framework es un conjunto de herramientas y estructuras que proporciona una base sólida para la creación de proyectos con objetivos específicos. Actúa como una plantilla inicial que establece las bases organizativas y facilita el desarrollo de software.

El framework permite agilizar los procesos de desarrollo, ya que evita tener que escribir código de forma repetitiva, asegura unas buenas prácticas y la consistencia del código.

Dataset

Un Dataset, como su nombre indica, es simplemente un conjunto de datos, ordenado bajo un sistema de almacenamiento. Son de gran utilidad para automatizar tareas relacionadas con esos datos o para analizar los mismos de una forma más ágil.

Modelo

Un modelo de IA es un programa o algoritmo que se basa en datos de entrenamiento para reconocer patrones y realizar predicciones o tomar decisiones. Cuantos más puntos de datos reciba un modelo de IA, más preciso puede ser en su análisis de datos y pronósticos.

Red neuronal

Una red neuronal es un método en inteligencia artificial que enseña a las computadoras a procesar datos de una manera inspirada en el cerebro humano. Es un tipo de proceso de aprendizaje automático, llamado aprendizaje profundo, que utiliza nodos o neuronas interconectadas en una estructura en capas que se asemeja al cerebro humano. Crea un sistema adaptativo que las computadoras utilizan para aprender de sus errores y mejorar continuamente.

En una red neuronal artificial, cada neurona realiza una operación matemática en las entradas ponderadas y luego pasa el resultado a través de una función de activación. Esta función de activación introduce no linealidad en la red, lo que es crucial para que la red sea capaz de modelar datos y relaciones complejas. Sin funciones de activación, una red neuronal sería simplemente una combinación lineal de sus entradas, lo que limitaría su capacidad.

Tipos de funciones de activación:

- **Sigmoide**: Utilizada en redes neuronales tradicionales, produce salidas entre 0 y 1. Especialmente útil en problemas de clasificación binaria, donde la red debe predecir probabilidades.
- **ReLU (Rectified Linear Unit)**: Una función lineal simple que acelera el entrenamiento y mejora el rendimiento en muchas aplicaciones. Transforma las entradas en cero si son negativas y las deja sin cambios si son positivas.
- **Tangente hiperbólica (Tanh)**: Similar a la sigmoide pero con un rango entre -1 y 1. Es útil en redes neuronales donde las salidas deben ser centradas alrededor de cero.

De esta manera, las redes neuronales artificiales intentan resolver problemas complicados, como resumir documentos o reconocer rostros, con mayor precisión.

Aplicaciones de las Redes Neuronales

Las redes neuronales han demostrado ser extremadamente versátiles y se han aplicado con éxito en una amplia variedad de campos. A continuación, se presentan algunas de las aplicaciones más destacadas de las redes neuronales:

- Visión por Computadora

  En el campo de la visión por computadora, las redes neuronales convolucionales (CNN) han revolucionado la capacidad de las computadoras para comprender y procesar imágenes y videos. Estas redes son capaces de realizar tareas como el reconocimiento facial, la detección de objetos, la segmentación de imágenes y la clasificación de escenas.

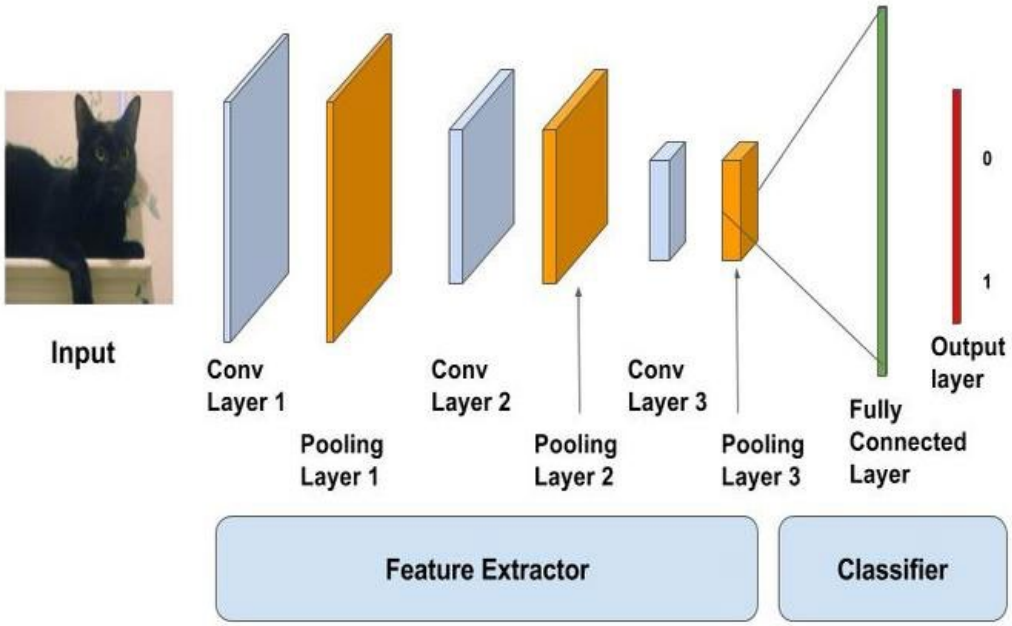

Figura 6.Esquema Red Neuronal Convolucional

En la *Figura 6.Esquema Red Neuronal Convolucional* se observa un posible esquema de CNN donde se clasifica de forma binaria una imagen gracias a la extracción, en las capas previas, de las características de la misma.

Las CNN han encontrado aplicaciones en la industria automotriz para la conducción autónoma, en la medicina para el análisis de imágenes médicas y en la seguridad para la detección de anomalías.

- Procesamiento de Lenguaje Natural

  Las redes neuronales recurrentes (RNN) y las redes neuronales de atención se utilizan en el procesamiento de lenguaje natural (NLP) para tareas como la traducción automática, el análisis de sentimientos, la generación de texto y la respuesta automática a preguntas. Estas redes son capaces de comprender y

generar texto de manera más similar al lenguaje humano que los enfoques tradicionales.

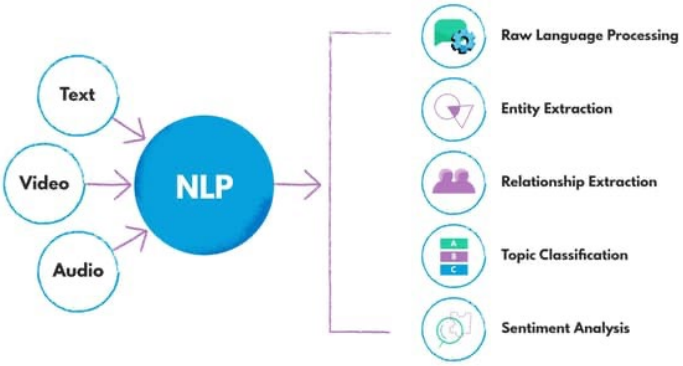

Figura 7. Procesamiento de lenguaje natural

Las aplicaciones de NLP son fundamentales en la industria de los chatbots, la traducción automática de idiomas y el análisis de opiniones en redes sociales como se puede observar en la Figura 7. Procesamiento de lenguaje natural.

- Juegos y Deportes

  Las redes neuronales también se han aplicado en el campo de los juegos y los deportes. En juegos de mesa como el ajedrez y el Go, se han desarrollado programas de IA basados en redes neuronales que han superado a los campeones mundiales humanos. Además, en el deporte, las redes neuronales se utilizan para el análisis de datos deportivos y la toma de decisiones estratégicas.

Red Neuronal Residual

La Red Neuronal Residual (ResNet) es un modelo de aprendizaje profundo utilizado para aplicaciones de visión por ordenador. Es una arquitectura de Red Neuronal Convolucional (CNN) diseñada para admitir cientos o miles de capas convolucionales.[5]

Aprendizaje Activo

El Aprendizaje Activo es una técnica de aprendizaje automático que implica la interacción activa del modelo con un conjunto de datos. En lugar de recibir datos de entrenamiento pasivamente, el modelo selecciona activamente las instancias de datos que considera más informativas para mejorar su rendimiento.

Estrategias de Aprendizaje Activo

Existen varias estrategias de selección de datos en el Aprendizaje Activo:

- **Balance exploration and exploitation**: La elección de imágenes para etiquetar se percibe como un dilema entre la exploración y la explotación en el espacio de representación de datos. Esta estrategia gestiona este compromiso al modelar el problema de aprendizaje activo como un problema de bandit contextual. Por ejemplo, Bouneffouf et al. proponen un algoritmo secuencial llamado Muestreo Activo de Thompson (ATS), que en cada ronda asigna una distribución de muestreo en el conjunto de datos, selecciona una imagen de esta distribución y consulta al oráculo para esta imagen seleccionada.

- **Expected model change**: etiquetar aquellas imágenes que cambiarían más el modelo actual.

- **Expected error reduction**: etiquetar aquellas imágenes que reducirían más el error de generalización del modelo.

- **Exponentiated Gradient Exploration for Active Learning**: propone un algoritmo secuencial llamado gradiente exponenciado (EG)-activo que puede mejorar cualquier algoritmo de aprendizaje activo mediante una exploración aleatoria óptima.

- **Random Sampling**: se selecciona una muestra al azar.

- **Uncertainty sampling**: etiquetar aquellas imágenes para las cuales el modelo actual tiene menos certeza sobre cuál debería ser la salida correcta. Esto implica medir la incertidumbre del modelo sobre sus predicciones y seleccionar las imágenes donde esta incertidumbre es mayor. Al hacerlo, el modelo puede centrarse en áreas donde necesita mejorar su capacidad predictiva.

    - **Entropy Sampling**: Se utiliza la fórmula de entropía en cada muestra y se considera que la muestra con la entropía más alta es la menos segura.
    - **Margin Sampling**: Se considera que la muestra con la diferencia más pequeña entre las dos probabilidades de clase más altas es la más incierta. Esta estrategia se basa en la idea de que los ejemplos cerca del límite son los más informativos, ya que pueden ayudar al modelo a definir mejor las fronteras de decisión.
    - **Least Confident Sampling**: Se considera que la muestra con la probabilidad de clase más baja es la más incierta.

- **Query by committee**: Mantiene un conjunto de modelos y selecciona ejemplos que generen desacuerdo entre ellos. Esta estrategia se basa en la idea de que, si varios modelos no están de acuerdo en una predicción, ese ejemplo particular es difícil y, por lo tanto, es valioso para el proceso de aprendizaje.

- **Querying from diverse subspaces or partitions**: Cuando el modelo subyacente es un bosque de árboles, los nodos hoja pueden representar particiones (superpuestas) del espacio de características original. Esto ofrece la posibilidad de seleccionar instancias de particiones no superpuestas o mínimamente superpuestas para etiquetar.

- **Variance reduction**: etiquetar aquellas imágenes que minimizarían la varianza de la salida, que es uno de los componentes del error.

- **Conformal prediction**: predice que una imagen tendrá una etiqueta similar a las imágenes previamente clasificadas de alguna manera especificada y el grado de similitud entre las ya clasificadas se utiliza para estimar la confianza en la predicción.

- **Mismatch-first farthest-traversal**: El criterio de selección principal es la discrepancia de predicción entre el modelo actual y la predicción del vecino más cercano. Se centra en las imágenes que se predicen incorrectamente. El segundo criterio de selección es la distancia a las imágenes previamente clasificadas, el más alejado primero. Su objetivo es optimizar la diversidad de los datos seleccionados.

- **User Centered Labeling Strategies**: El aprendizaje se logra aplicando reducción de dimensionalidad a gráficos y figuras como gráficos de dispersión. Luego, se le pide al usuario que etiquete los datos compilados (categóricos, numéricos, puntuaciones de relevancia, relación entre dos imágenes).

El Aprendizaje Activo es especialmente útil cuando se enfrenta a conjuntos de datos grandes y costosos de etiquetar. Al permitir que el modelo seleccione los ejemplos más informativos, se puede lograr un aprendizaje más eficiente y preciso.

IDE

Un entorno de desarrollo integrado (IDE, por sus siglas en inglés) es una aplicación de software que ayuda a los programadores a desarrollar código de software de manera eficiente. Incrementa la productividad del desarrollador al combinar capacidades como la edición de software, la compilación, las pruebas y el empaquetado en una aplicación fácil de usar.

Un IDE consta principalmente de un editor de código fuente, herramientas de construcción automáticas y un depurador.

Suelen tener un auto-completado inteligente de código (*IntelliSense*). Además, algunos IDE contienen un compilador, un intérprete, o ambos.

Modelo Vista Controlador (MVC)

MVC (Modelo-Vista-Controlador) es un patrón de diseño de software comúnmente utilizado para implementar interfaces de usuario, datos y lógica de control. Enfatiza la separación entre la lógica empresarial del software y su visualización.

El modelo de MVC le ayuda a crear aplicaciones que separan los diferentes aspectos de la aplicación (lógica de entrada, lógica comercial y lógica de la interfaz de usuario), y proporciona un vago acoplamiento entre estos elementos.

En la *Figura 8. Esquema Modelo Vista Controlador* se puede observar las relaciones e iteraciones entre los distintos módulos de este patrón de diseño.

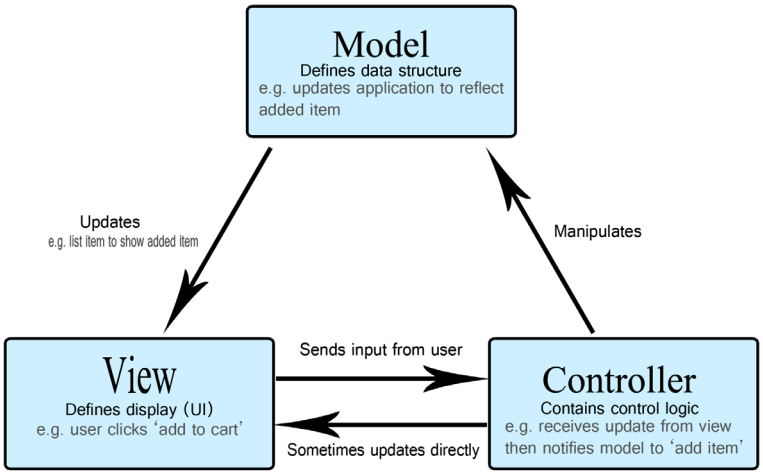

Figura 8. Esquema Modelo Vista Controlador

Sistema de control de versiones

Los sistemas de control de versiones son herramientas de software que permiten a los equipos de desarrollo gestionar los cambios en el código fuente a lo largo del tiempo.[7]

Este sistema de control de versiones sincroniza las versiones y se asegura de que los cambios no entren en conflicto con los cambios de otros usuarios.

El código almacenado en versiones anteriores se puede ver y restaurar desde el control de versiones en cualquier momento y según sea necesario. Las versiones facilitan la creación de nuevas versiones de cualquier versión del código.

Librería

Librería o biblioteca es un conjunto de utilidades, implementaciones funcionales codificadas y programas externos focalizados en aumentar la funcionalidad del sistema, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

La implementación de una biblioteca tiene como finalidad el ser utilizada por otras aplicaciones de forma independiente y simultaneas. Además, estas librerías pueden requerir de otras para funcionar y pueden vincularse a un programa o a otra librería en distintos puntos del desarrollo o la ejecución.

Las librerías, básicamente, se pueden clasificar en estáticas y dinámicas. Las primeras están constituidas por archivos de código objeto empaquetados en su interior. Por su parte las dinámicas son ficheros que contienen código objeto construido de forma independiente de su ubicación y se pueden cargar en cualquier programa.

JSDoc

JSDoc es un estándar de documentación de código utilizado en JavaScript para documentar funciones, clases y objetos en el código fuente. Proporciona una manera estructurada de describir la interfaz, el propósito y el uso de las partes del código JavaScript.

También se pueden utilizar para generar automáticamente documentación legible para humanos.

```
1  /**
2   * JSDoc what is that?
3   * @param {number} a
4   * @param {number} b
5   */
6  function sum(a, b){
7      return a + b;
8  }
```

Figura 9. Ejemplo definición en JSDoc

En la *Figura 9. Ejemplo definición en JSDoc* se puede observar un ejemplo donde se documenta y explica la función sum, indicando que dispone de dos parámetros de entrada (@param), esto serán de tipo numérico al definirlos como “{number}” y serán las variables a y b.

Diagrama de Gantt

Un diagrama de Gantt es una herramienta de gestión de proyectos que ilustra el trabajo completado durante un período de tiempo en relación con el tiempo planificado para dicho trabajo que nos ayudará a visualizar las tareas y principales hitos de una forma práctica.

Normalmente, consta de dos secciones: el lado izquierdo describe una lista de tareas, mientras que el lado derecho muestra una línea de tiempo con barras de programación que visualizan el trabajo.

JWT

JWT (JSON Web Token) es un objeto de JSON, una herramienta de estándar abierto cuyo objetivo es establecer una transmisión de información de manera segura entre dos partes, pues se firma de forma virtual. Definido en la RFC 7519. [8]

Puede funcionar desde cualquier espacio, dado que su tamaño no es demasiado extenso, y su cadena compuesta por tres partes separadas por un punto (.)

HTTPS

El protocolo de transferencia de hipertexto seguro (HTTPS) es la versión segura de HTTP, que es el principal protocolo utilizado para enviar datos entre un navegador web y un sitio web.

Este protocolo está encriptado para incrementar la seguridad de las transferencias de datos.

Swagger

Swagger es un conjunto de herramientas de código abierto que se utiliza para diseñar, documentar y probar API (Interfaces de programación de aplicaciones).

Proporciona una forma estandarizada y altamente eficiente de describir las capacidades de una API, lo que facilita su comprensión y utilización tanto para los desarrolladores como para las máquinas.

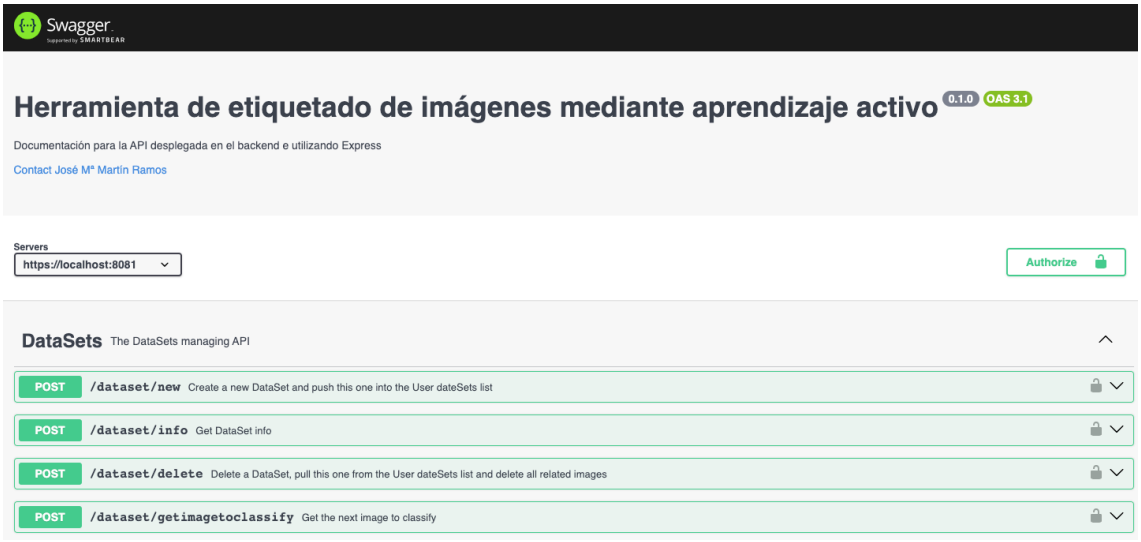

Figura 10. Web desplegada con la doc en Swagger UI

En la *Figura 10. Web desplegada con la doc en Swagger UI* se puede observar el portal generado con Swagger para poder revisar la documentación relacionada con los endpoints del backend.

4. Técnicas y herramientas utilizadas

4.1. Entorno de desarrollo

PyCharm CE

PyCharm es un entorno de desarrollo integrado, utilizado para programar en Python. Proporciona análisis de código, un depurador gráfico, un tester unitario integrado, integración con sistemas de control de versiones y admite el desarrollo web.

Como se observa en la *Figura 11. PyCharm CE*, se ha utilizado para la implementación de la Red Neural dando uso de las herramientas de debug que esta plataforma presenta.[9]

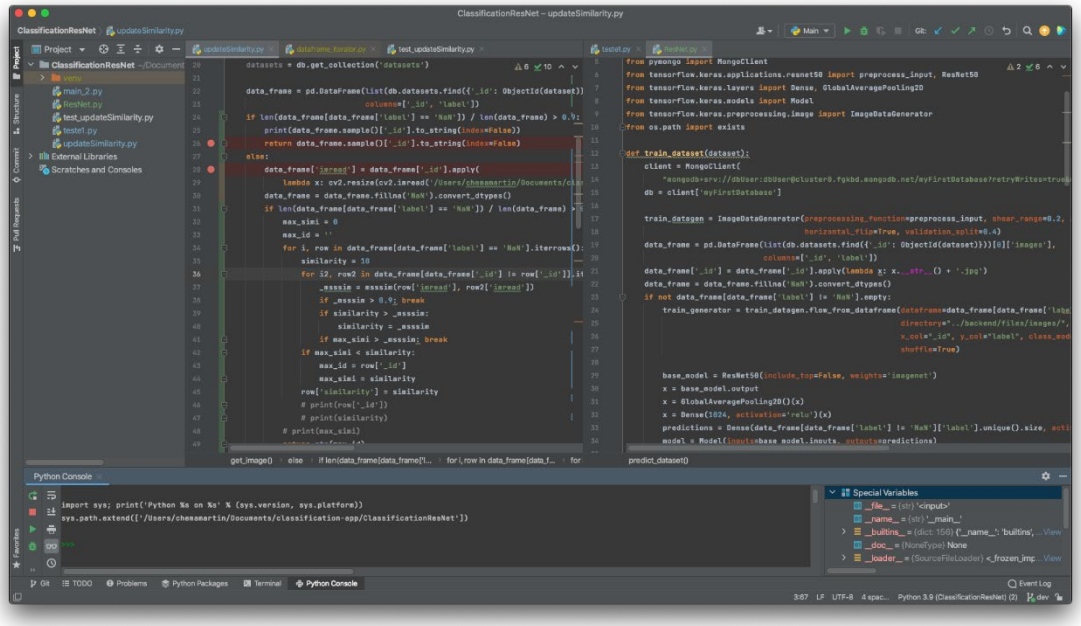

Figura 11. PyCharm CE

Tiene un editor de código inteligente que ayuda al programador a escribir códigos de alta calidad y a navegar fácilmente. También permite realizar cambios rápidos y eficaces.

WebStorm

WebStorm, igual que el caso anterior, es un entorno de desarrollo integrado (IDE) especializado en JavaScript, con un editor de código inteligente y otras herramientas

para desarrolladores. Está diseñado con características que pueden hacer que la programación sea más productiva y agradable.

WebStorm se centra en proporcionar herramientas y características avanzadas para desarrolladores que trabajan con tecnologías web, como HTML, CSS, JavaScript y otros lenguajes. *En la Figura 12. WebStorm* observamos una pantalla del mismo.

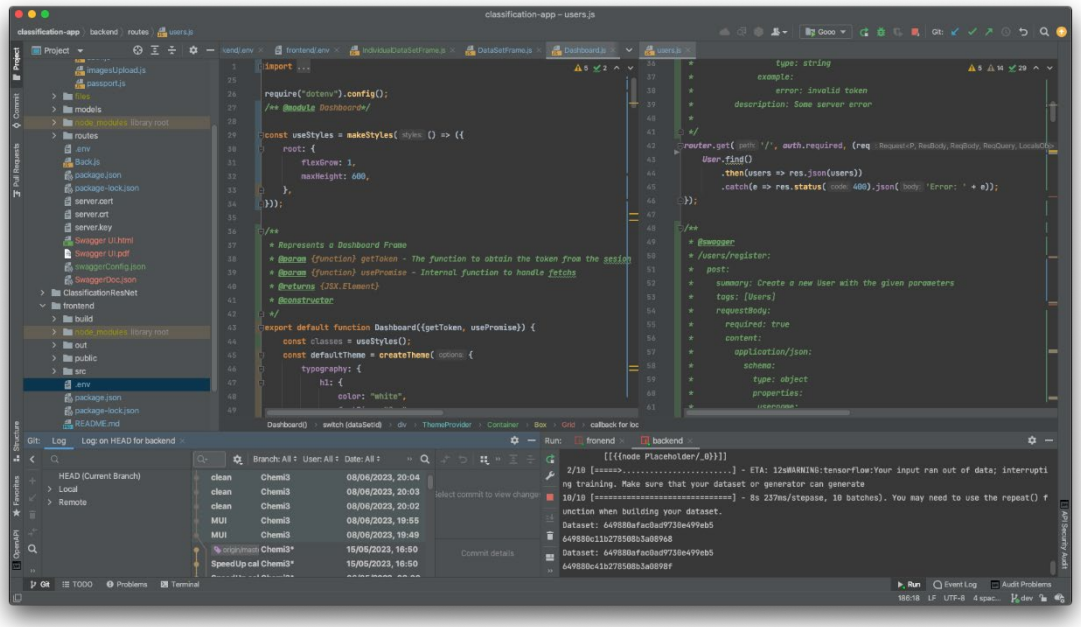

Figura 12. WebStorm

Se ha utilizado este IDE por ser completamente compatible con nuestra aplicación, dado que nos permite desarrollar con React y haciendo uso de múltiples herramientas localizadas en la productividad como son el control de versiones, autocompletado, refactorización u optimización de importaciones.[10]

4.2. Herramientas de pruebas

Google Chrome DevTools

Conjunto de herramientas para desarrolladores web incorporado directamente en el navegador Google Chrome. Estas herramientas están diseñadas para ayudar a los desarrolladores web a depurar, analizar y optimizar sus aplicaciones y sitios web. DevTools ofrece una amplia gama de características y funciones que permiten a los desarrolladores inspeccionar el código, monitorear el rendimiento, simular dispositivos móviles y mucho más.

DevTools puede ayudarte a editar páginas sobre la marcha y diagnosticar problemas rápidamente, lo que en última instancia te ayuda a construir sitios web mejores y más rápidos.[11]

Google Chrome DevTools es una herramienta esencial para los desarrolladores web, ya que ayuda a optimizar el rendimiento y la calidad de las aplicaciones y sitios web.

Insomnia

Un cliente REST es una herramienta utilizada para interactuar con la API RESTful expuesta para la comunicación. Insomnia REST Client es un cliente de API REST de código abierto y potente utilizado para almacenar, organizar y ejecutar solicitudes de API REST de manera elegante como podemos observar en la *Figura 13. Insomnia*.

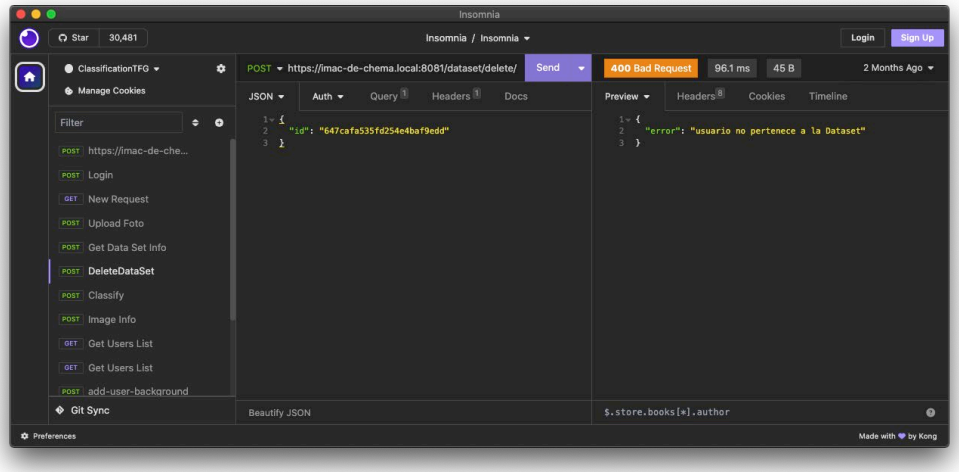

Figura 13. Insomnia

Facilita el proceso de desarrollo de aplicaciones que interactúan con los servicios web, dado que permite a los desarrolladores probar, depurar y documentar estas solicitudes.

4.3. Librerías

PyMongo

Contiene herramientas para trabajar con MongoDB, y es la forma recomendada de trabajar con MongoDB desde Python. Es adecuada para almacenar datos semi-estructurados y no relacionales y facilita la comunicación entre aplicaciones Python y bases de datos MongoDB, permitiendo a los desarrolladores realizar operaciones como inserción, actualización, eliminación y consulta de datos.

En nuestro proyecto se ha implementado para la comunicación entre la IA que está desarrollada con Python con nuestra base de datos. [13]

Sewar

Es un paquete de Python que se utiliza para evaluar la calidad de las imágenes utilizando diferentes métricas. Esta librería está diseñada para medir y comparar la calidad de las imágenes en el contexto de la evaluación de algoritmos de procesamiento de imágenes y visión por computadora.

En nuestro caso se utiliza para calcular el indicador de similaridad entre las diferentes imágenes, para así poder determinar la más correcta a clasificar.

Pandas

Librería encargada de trabajar con los conjuntos de datos en la parte de Python, así como volcado de nuevo a la base de datos. Proporciona estructuras de datos y herramientas de análisis de datos eficientes y flexibles.

Tensorflow

Paquete de Python que implementa la Inteligencia Artificial del proyecto. que se utiliza principalmente para la creación, entrenamiento e implementación de modelos de aprendizaje automático y redes neuronales.

En nuestro caso se encarga del entrenamiento de los diferentes modelos, así como su predicción. Dadas las características del entorno hemos optado por la parte que trabaja únicamente con la CPU. [15]

Keras

Es una biblioteca de código abierto para Python que otorga una interfaz para el trabajo con redes neuronales, se puede ejecutar sobre TensorFlow. [16]. Facilita enormemente la creación de capas para las redes neuronales o la configuración de arquitecturas complejas.

ExpressJWT

Es un Middleware para Express que realiza las validaciones de los diferentes JWTs a través del módulo jsonwebtoken. Este módulo decodifica los tokens que vienen en cada una de las peticiones hacia el backend.

Express

Es un Framework (esquema o marco de trabajo de una estructura base para elaborar un proyecto con objetivos específicos, plantilla para comenzar) construido con una API REST sobre el entorno Node.js para el desarrollo de backends, proporciona características fundamentes de la aplicación web. Forma el centro de procesado de las peticiones procedentes de la página Web. [17]

Passport

Passport es un middleware de autenticación para Node.js, que se basa en el concepto de autentificación y ofrece gran variedad de estrategias. Puede ser integrado fácilmente en cualquier aplicación web basada en Express sin ser intrusivo. Lo utilizaremos para dar acceso a los usuarios dentro del sistema.

NodeCallsPython

Es un Módulo de Express que es capaz de ejecutar llamadas a scripts de Python y a sus correspondientes módulos dando uso del entorno instalado en el sistema. Lo utilizaremos para ejecutar las acciones sobre las redes neuronales alojadas en los diferentes Pythons.

DotEnv

Es una Librería de JS que carga variables de entorno a la plataforma, facilitando el despliegue del proyecto en diferentes situaciones ajustando las distintas variables, que permite separar la configuración sensible y específica de la aplicación del código fuente, lo que es útil para mantener la seguridad y la portabilidad en diferentes entornos.

Cors

CORS es un paquete de Node.js que proporciona un middleware para Connect/Express que se puede utilizar para habilitar CORS (Cross-Origin Resource Sharing) con varias opciones. Estas opciones están orientadas al control de los orígenes desde los cuales se realizan las llamadas al backend. [19]

Mongoose

Librería de JS encargada de realizar las conexiones con la base de datos, que aporta una interfaz sencilla y está basada en esquemas para trabajar con bases de datos.

React

React es una librería de JavaScript encargada de crear una interfaz gráfica al usuario en la parte frontend de la aplicación. Actualiza y renderiza automáticamente los datos mediante la construcción eficiente de componentes reutilizables que actualizan y renderizan automáticamente cuando cambian los datos. También realiza los diferentes enrutamientos desde la página principal. [20]

Material UI

Es un componente de React que se encarga de la generación de los diferentes elementos que el usuario se puede encontrar en la página web, así como las diferentes disposiciones es estos.

React Images Uploader

Es un componente de React que nos va a facilitar el control de la carga de imágenes al sistema, así como la generación del correspondiente elemento y sus controles de errores. Este módulo nos permite tanto subir múltiples imágenes a la vez como arrastrar todas estas al elemento para que se carguen también al sistema.

4.4. Herramientas CASE

Las herramientas CASE son un conjunto de aplicaciones informáticas, utilizadas para automatizar actividades del ciclo de vida de desarrollo de sistemas (SDLC).

REM

Requirements Management es una herramienta de gestión de requisitos desarrollada en la Universidad de Sevilla que proporciona una metodología y una interfaz basada en tablas para la especificación de requisitos del sistema, siguiendo la metodología propuesta por Durán y Bernárdez. Esta herramienta ofrece funcionalidades adicionales que facilitan la elaboración, trazabilidad y control de los requisitos durante el desarrollo del proyecto.

El "Requirements Management" es un proceso esencial en el desarrollo de software y en la ingeniería de sistemas. Se refiere a la identificación, documentación, seguimiento, control y gestión de los requisitos de un proyecto a lo largo de todo su ciclo de vida. Estos requisitos pueden incluir especificaciones funcionales, no funcionales y restricciones que guían el diseño, desarrollo y prueba de un sistema.

Online Gantt

Herramienta Web, con una interfaz capturada en la *Figura 14. Online Gantt,* utilizada en la generación de la planificación temporal del proyecto esbozando el diagrama de Gantt.[22]

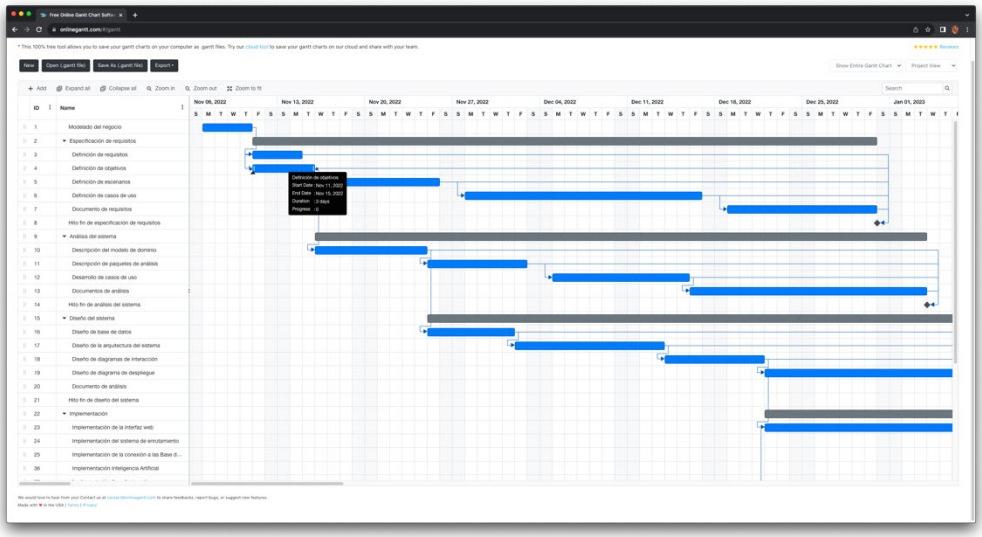

Figura 14. Online Gantt

Visual Paradigm

El software utilizado para la generación de los diagramas UML, herramienta fundamental en este proyecto como observamos en la *Figura 15. Visual Paradigm*, permitiendo la visualización y documentación efectiva de las diferentes perspectivas del sistema. Contribuyendo a la planificación y desarrollo exitoso del mismo. [21]

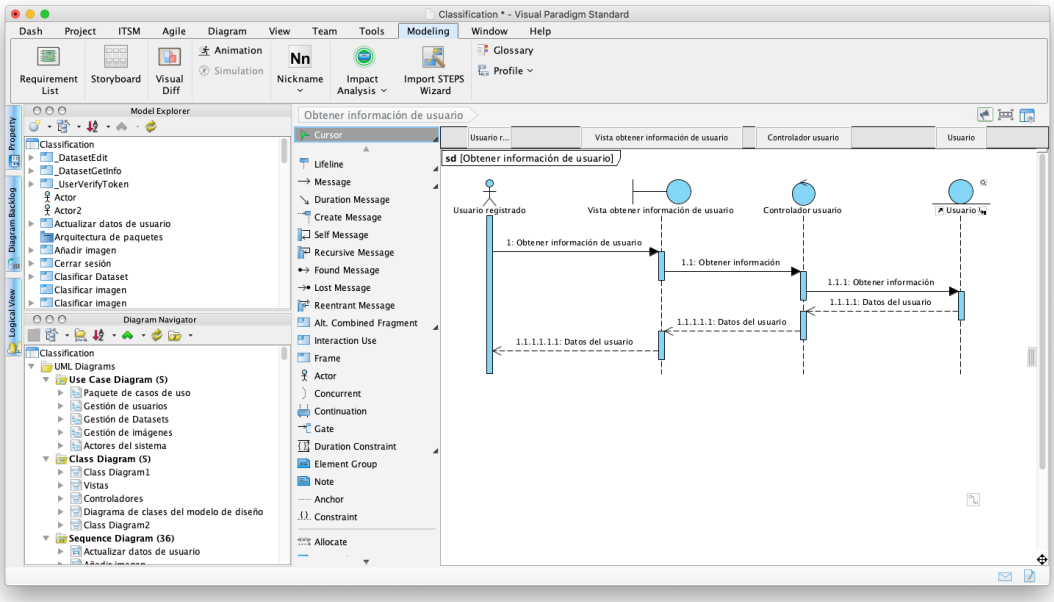

Figura 15. Visual Paradigm

También proporciona capacidades de ingeniería de código y generación de informes, incluida la generación de código.

4.5. Lenguajes

HTML

HyperText Markup Language (HTML), también conocido como lenguaje de marcas de hipertexto, es un lenguaje de marcado utilizado para el desarrollo de la estructura, el diseño y presentación de las páginas web.

El lenguaje HTML se compone de elementos que se definen utilizando etiquetas. Cada etiqueta tiene un nombre y encierra el contenido al que se aplica, “**<p>**” “**<h1>**”

En el contexto del sistema en cuestión, se ha empleado HTML para estructurar la vista de cada una de las páginas que componen el sistema. Mediante el uso de etiquetas y elementos HTML, se ha definido la organización y disposición de los contenidos en cada página.[23]

CSS

Cascading Style Sheets (CSS) es un lenguaje de hojas de estilo, utilizado para describir la presentación de un documento escrito en un lenguaje de marcado, como: HTML o XML. CSS es una tecnología fundamental de la World Wide Web, junto con HTML y JavaScript. Y describe cómo se deben representar los elementos en pantalla, en papel, en voz o en otros medios.

El CSS es uno de los lenguajes centrales de la web abierta y está estandarizado en todos los navegadores web**.**

En este proyecto se va a emplear en la configuración de la interfaz gráfica de la página web. [24]

JavaScript

JavaScript es un lenguaje de programación o scripting que te permite implementar características complejas en páginas web. Es la tercera capa del conjunto de tecnologías estándar para la web, de las cuales hemos cubierto en detalle otras dos (HTML y CSS).

Se utiliza para el desarrollo del servidor frontend, permitiendo entregar toda la funcionalidad esperada. Sus capacidades dinámicas incluyen la construcción de objetos en tiempo de ejecución, listas de parámetros variables, variables de función, creación de scripts dinámicos, introspección de objetos, y recuperación de código fuente. [25]

Python

Python es un lenguaje de programación interpretado, orientado a objetos y de alto nivel con semántica dinámica. Sus estructuras de datos incorporadas de alto nivel, combinadas con su tipado dinámico y enlace dinámico, lo hacen muy atractivo para el Desarrollo Rápido de Aplicaciones. La sintaxis simple y fácil de aprender de Python enfatiza la legibilidad y, por lo tanto, reduce el costo del mantenimiento del programa.

Python admite módulos y paquetes, lo que fomenta la modularidad del sistema y la reutilización del código.

Por todo ello, Python se ha convertido en uno de los lenguajes de programación más populares y ampliamente utilizados en una variedad de campos, desde desarrollo web y científico hasta inteligencia artificial

En nuestro proyecto se da uso para el desarrollo de las redes neuronales y el análisis de las imágenes. [26]

JSON

JSON (JavaScript Object Notation) es un formato de intercambio de datos ligero y fácil de leer y escribir tanto para humanos como para máquinas.

Se utiliza comúnmente para transmitir datos entre un servidor y una aplicación web, como una alternativa a XML. Los datos en JSON se representan mediante pares clave-valor y pueden incluir matrices, objetos, cadenas, números, booleanos y valores nulos.[27]

Un objeto JSON es un conjunto no ordenado de pares clave-valor. Se delimita con llaves **{}**. Cada clave es una cadena y está seguida por dos puntos **:**. Los valores pueden ser cadenas, números, booleanos, objetos, matrices o null.

Por su parte, una matriz JSON es una colección ordenada de valores, delimitada por corchetes **[]**. Los valores pueden ser cualquiera de los tipos mencionados anteriormente.

JSX

JSX (JavaScript XML) es una extensión de sintaxis utilizada en React, una biblioteca de JavaScript, para construir interfaces de usuario. Combina elementos de JavaScript y HTML para describir la estructura y apariencia de los elementos de la interfaz de usuario.[28]

JSX es una herramienta poderosa que hace que la construcción de interfaces de usuario en React sea más legible y eficiente al combinar la sintaxis de HTML con la potencia de JavaScript.

4.6. Herramientas de despliegue

Serve

Es un paquete de Node.js que proporciona de manera sencilla el despliegue de un servidor web estático.
A la hora de utilizar el módulo se debe indicar la carpeta en la que se encuentran todos los ficheros correspondientes al despliegue estático del servidor, se inicia por tanto un servidor web en el puerto predeterminado (3000) que servirá los archivos referidos con anterioridad. [29]

5. Fases del desarrollo

El trabajo realizado se ha llevado a cabo siguiendo el marco de trabajo proporcionado por el Proceso Unificado. Este enfoque se caracteriza por ser un marco de trabajo genérico que se adapta a una amplia variedad de sistemas de software en diferentes áreas de aplicación, tipos de organizaciones y tamaños de proyectos.

El Proceso Unificado es un enfoque de desarrollo de software que se basa en principios iterativos e incrementales para crear aplicaciones de alta calidad.

El Proceso Unificado se basa en componentes y utiliza el lenguaje UML para una definición precisa de los mismos. Está impulsado por casos de uso, pone un fuerte énfasis en la arquitectura y sigue un enfoque iterativo e incremental.[30]

La naturaleza iterativa e incremental del Proceso Unificado le permite ser altamente flexible en cuanto a los tipos de proyectos que puede abordar. Esto se debe a que las fases del proyecto se desarrollan de manera cíclica en lo que se conocen como ciclos de vida.

El Proceso Unificado consta de cuatro fases principales:

- **Fase de Inicio:** Es la primera etapa del Proceso unificado y marca el comienzo del desarrollo de la aplicación. En esta fase, se establece una base sólida para el proyecto al definir su alcance, objetivos y viabilidad. En esta fase se define el alcance del proyecto y se especifica el modelo de negocio.

  En nuestra planificación, esta fase incluye la actividad de modelado del negocio.

- **Fase de Elaboración:** En esta fase se realiza la planificación del proyecto, que coincide con la especificación de los casos de uso y el diseño de la arquitectura del sistema.

Se profundiza en los detalles del proyecto. Se definen los requisitos con mayor precisión y se establece una base arquitectónica sólida para la aplicación. Su principal objetivo es asegurar que el proyecto esté bien fundamentado.

En nuestra planificación de tareas, esta fase abarca la especificación de requisitos y el análisis del sistema.

- **Fase de Construcción**: En esta fase se lleva a cabo la implantación, el desarrollo y construcción del proyecto.

  En nuestra planificación, esta fase comprende la implementación del sistema.

- **Fase de Transición**: En esta fase se obtiene una versión beta del producto y se incorporan mejoras sugeridas en dicha versión. La aplicación está lista para ser lanzada al público o para implantarse en un entorno de producción.

  En nuestra planificación, esta fase abarca las pruebas y la retroalimentación basada en los resultados obtenidos.

Cada uno de estos ciclos se cierra con una versión entregable del producto. Además, los ciclos de vida incluyen hitos de control, generalmente hitos principales y más exhaustivos al final de cada fase, y hitos secundarios y más flexibles al final de cada iteración. Estos hitos desempeñan una función fundamental en la toma de decisiones para la siguiente fase, permitiendo un control continuo del desarrollo del proyecto y brindando retroalimentación para estimar los tiempos y recursos en futuros desarrollos.

Una iteración se define como una secuencia de actividades con un plan establecido y criterios de evaluación propios como se puede observar en la *Figura 16. Iteraciones del Proceso Unificado.* Su resultado es una versión ejecutable que no se orienta a la entrega final, sino que representa un hito secundario o interno dentro del equipo de desarrollo.

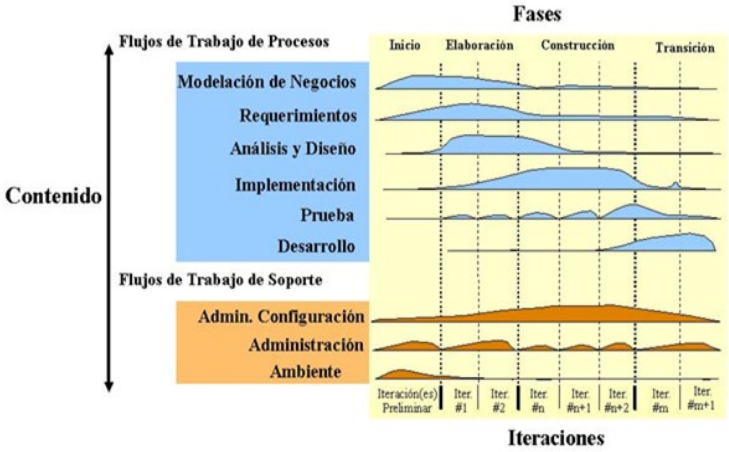

Figura 16. Iteraciones del Proceso Unificado.

5.1. Aspectos relevantes

5.1.1. Anexo I

En este anexo se presentan los apartados clave del plan del proyecto software. Recopila los cálculos que estiman la dimensión del sistema, la planificación temporal y el esfuerzo necesario para llevar a cabo su implementación. Además, se identifican las distintas tareas a realizar, proporcionando información sobre su duración y secuencia, con el fin de llevar a cabo el desarrollo del sistema dentro del tiempo estimado.

Igualmente busca brindar una comprensión detallada de los elementos fundamentales del proyecto.

5.1.2. Anexo II

Este documento abarca la especificación de los requisitos que el sistema debe cumplir. Incluye una descripción detallada de los actores que interactúan y forman parte del sistema, así como los objetivos que se pretenden alcanzar con su implementación.

El principal es poder añadir nuevos usuarios que se registren mediante un formulario de registro, así como modificar sus datos una vez registrados. Además, debe controlar la autorización de cada usuario para acceder a cada una de las páginas. También deberá seleccionar la mejor imagen a clasificar dentro del dataset, almacenar el modelo generado tras la clasificación parcial o total de las imágenes que forman el dataset, y ejecutar la predicción de la clasificación de nuevas imágenes cargadas en el dataset.

Finalmente deberá almacenar y gestionar el acceso a las imágenes cargadas por el usuario.

También se presentan los requisitos de restricción, de información y no funcionales que se deben considerar. Además, se detallan los casos de uso del sistema, que describen las diferentes interacciones entre los actores y el sistema, proporcionando una imagen completa de las funcionalidades que el sistema debe ofrecer.

5.1.3. Anexo III

En este documento se incluye el modelo de dominio como análisis del problema, donde se representan las entidades y relaciones principales.

El diagrama de modelo de dominio sirve como representación inicial del dominio del problema, analizando las clases principales que compondrán la solución. Este diagrama ha sufrido diferentes cambios en las sucesivas iteraciones aplicadas en el desarrollo del Proceso Unificado.

Del mismo modo, se presentan las clases de análisis organizadas en diagramas de paquetes que muestran la arquitectura de paquetes del sistema. Y se pueden observar las dependencias entre los paquetes que se representan con flechas discontinuas.

Por último, se recogen los requisitos funcionales del sistema, presentados como diagramas de secuencia, que muestran la interacción de los actores y ofrecen un primer detalle de la estructura de estos en relación al sistema.

5.1.4. Anexo IV

En este anexo se encuentra el diseño del sistema, que se describe a través del diagrama de arquitectura, los diagramas de paquetes y los diagramas de secuencia que representan la realización de los casos de uso.

En la primera parte se presenta la arquitectura del sistema global, representada mediante el patrón arquitectónico MVC, y se detalla la composición interna de los paquetes que conforman dicha arquitectura, así como las relaciones entre ellos y los archivos de código que los componen.

En el siguiente apartado se presenta el diagrama de despliegue del sistema, el cual muestra la distribución física y de software de los elementos que componen el sistema, así como sus interrelaciones.

También se incluye el diseño de la base de datos y el diseño gráfico de la interfaz del sistema. Por último, se detalla el diseño de las pruebas que se llevarán a cabo.

5.1.5. Anexo V

Contiene un manual para programadores con el objetivo de facilitar el mantenimiento de la plataforma. En este manual se explica la estructura básica del sistema de manera detallada, brindando información y guías prácticas para los programadores.

El primer apartado presenta la arquitectura del sistema global, representada mediante el patrón arquitectónico MVC. Además, se detalla la composición interna de los paquetes que conforman dicha arquitectura, así como las relaciones entre ellos y los archivos de código que los componen.

En el siguiente apartado se presenta el diagrama de despliegue del sistema, el cual muestra la distribución física y de software de los elementos que componen el sistema, así como sus interrelaciones.

A continuación, se incluyen una serie de diagramas de secuencia, que representan la interacción y los mensajes relacionados con los casos de uso expuestos en el Anexo II. Estos diagramas profundizan en el nivel de análisis descrito en el Anexo III.

Finalmente, se proporciona una explicación detallada del diagrama de la base de datos del sistema, acompañado de una leyenda que aclara los elementos que lo componen.

5.1.6. Anexo VI

Por último, se proporciona una descripción detallada de la funcionalidad de la aplicación web, con el objetivo de que los usuarios puedan orientarse a través de la interfaz utilizando la guía. Se describen las principales páginas y se detallan las diferentes funcionalidades del sistema, ofreciendo una visión completa de cómo utilizar la aplicación de manera efectiva.

5.2. Estimación del esfuerzo

La estimación del esfuerzo desempeña un papel fundamental en la determinación de los costes asociados al proyecto. Para evaluar el esfuerzo necesario, se ha empleado una medida de funcionalidad conocida como análisis de puntos de casos de uso (UCP), que implica una serie de cálculos y estimaciones detallados que se exponen a continuación.

La fórmula para calcular los puntos de casos de uso (UCP) es la siguiente:

$$UCP = UCCP * TCF * ECF$$

Aquí, UCCP representa los puntos de casos de uso no ajustados y se compone de dos elementos: UUCW, que denota el peso de los actores no ajustados, y UAW, que indica el peso de los casos de uso no ajustados.

$$UCCP = UUCW + UAW$$

Por otro lado, TCF son los factores de complejidad técnica, mientras que ECF son los factores de complejidad del entorno

Casos de Uso	Razón	Peso
Simple	El número de transacciones es 3 o menor.	5
Medio	El número de transacciones está entre 4 y 7.	10
Complejo	El número de transacciones es mayor de 7.	15

Tabla 1. Pesos de casos de uso

La asignación de pesos a los actores y casos de uso se realiza para calcular ambas contribuciones en relación a su complejidad. De esta manera, se obtiene una estimación precisa del esfuerzo requerido para el proyecto.

Es importante tener en cuenta que esta metodología de análisis de puntos de casos de uso (UCP) proporciona una base sólida para comprender y cuantificar el esfuerzo involucrado en el proyecto, permitiendo una planificación y gestión efectivas de los recursos necesarios.

Actores	Razón	Peso
Simple	Representa a otro sistema con una API.	1
Medio	Representa a otro sistema con un protocolo de interacción como TCP/IP.	2
Complejo	Representa a un actor humano interactuando con una interfaz.	3

Tabla 2. Pesos de actores.

Los actores que forman el sistema son usuarios, por lo que son complejos.

	Caso de uso	Categoría	Peso
UAW	Registrarse	Medio	11
	Iniciar sesión	Medio	11
	Cerrar sesión	Simple	6
	Modificar contraseña	Medio	11
	Obtener información de usuario	Simple	6
	Registrar nueva dataset	Simple	11
	Eliminar dataset	Medio	6
	Obtener información de dataset	Simple	6
	Añadir imagen	Simple	8
	Obtener datos de imagen	Medio	6
	Obtener imagen a clasificar	Simple	6
	Clasificar imagen	Simple	9
	Crear modelo de dataset	Medio	9
	Ejecutar predicción	Simple	6
	Consultar resultado predicción	Medio	7
Total			**119**

Tabla 3. Cálculo de UAW.

	Actor	Categoría	Peso
UUCW	Usuario anónimo	Complejo	3
	Usuario registrado	Complejo	3
Total			**6**

Tabla 4. Cálculo de UUCW.

$$UCCP = 119 + 6 = 125$$

Factor	Nombre	Peso	Valor	Factor	Descripción
E1	Familiaridad con UML	1.5	3	4.5	Modelo trabajado a lo largo de la carrera, requiere profundizar.
E2	Trabajadores a tiempo parcial	-1	5	-5	El desarrollador trabaja a jornada parcial
E3	Capacidad de los analistas	0.5	2	1	En este caso el analista es el mismo desarrollador del proyecto, su capacidad es media
E4	Experiencia en la aplicación	0.5	1	0.5	El desarrollador no ha trabajado previamente con los lenguajes requeridos por lo que supone un riesgo.
E5	Experiencia en la orientación a objetos	1	4	4	La experiencia es de unos 2 años en desarrollos orientados a objetos.
E6	Motivación	1	4	4	El equipo se encuentra motivado con el trabajo.
E7	Dificultad del lenguaje de programación	-1	4	-4	El lenguaje de programación es relativamente fácil de aprender.
E8	Estabilidad de los requisitos	2	3	6	Los requisitos del proyecto son relativamente estables
Total				11	EFactor

Tabla 5. Factores de complejidad del entorno.

$$ECF = 1{,}4 + (-0{,}03 * EFactor) = 1{,}07$$

Factor	Nombre	Peso	Valor	Factor	Descripción
T1	Sistema distribuido	2	2	4	El sistema está formado por un único servidor junto con una base de datos remota.
T2	Rendimiento	2	2	4	No es factor fundamental, se requiere una mínima fluidez para garantizar la experiencia del usuario.
T3	Eficiencia del usuario final	1	2	2	Al no ser una herramienta basada en la productividad no es una prioridad.
T4	Procesamiento interno complejo	1	5	5	El sistema debe calcular todos los modelos de IA, de ahí su índice computacional
T5	Reusabilidad	1	5	5	Se debe intentar orientar el proceso en diferentes bloques para facilitar la reusabilidad.
T6	Facilidad de instalación	0.5	0	0	Al ser una aplicación web no precisa instalación.
T7	Facilidad de uso	0.5	4	2	El usuario debe comprender de forma sencilla las diferentes acciones que puede llevar a cabo en el sistema.
T8	Portabilidad	2	1	2	Dado que es una aplicación web, se puede acceder desde las diferentes plataformas sin excesivas modificaciones.
T9	Facilidad de cambio	1	5	5	El sistema debe plantearse para facilitar la implantación de posibles evolutivos o mejoras en el mismo.
T10	Concurrencia	1	4	4	Múltiples usuarios deben poder trabajar de forma simultánea.
T11	Características especiales de seguridad	1	3	3	Debe cumplir un mínimo de seguridad al tratar datos personales.
T12	Acceso directo a terceras partes	1	0	0	El sistema no se comunica con aplicaciones de terceros.
T13	Entrenamiento especial del usuario	1	0	0	El usuario no necesita un entrenamiento especial para hacer uso de la aplicación.
Total				36	TFactor

Tabla 6. Factores de complejidad técnica.

$$TCF = 0{,}5 + (0{,}01 * TFactor) = 0{,}86$$

Una vez calculados todos los factores que componen los puntos de casos de uso, obtenemos.

$$UCP = 125 * 0{,}86 * 1{,}07 = 115{,}025$$

Si lo multiplicamos por la constante del esfuerzo **F** cuyo valor es 20 obtenemos.

$$TPH = 20 * UCP = 2300{,}5 \ horas$$

El valor TPH indica las horas totales necesarias para completar el proyecto, por lo tanto, una persona a jornada completa de 8 horas tardará en completar el proyecto 363,6 días o lo que es aproximadamente lo mismo 1 año.

5.3. Planificación temporal y de tareas

La planificación temporal desempeña un papel crucial en los proyectos de desarrollo de software, ya que actúa como una referencia antes, durante y después del proyecto. Antes de comenzar el proyecto, se utiliza para desglosar las tareas de desarrollo y estimar el tiempo necesario para completar todas las subtareas que conforman el proyecto, lo que se conoce como planificación macroscópica. Durante el proyecto, se pueden agregar detalles a los bloques de tareas que se estimaron inicialmente, lo que permite ser más concretos y precisos en cuanto a los tiempos de ejecución. Después del proyecto, la planificación temporal nos permite obtener una perspectiva del tiempo estimado al inicio y compararlo con el tiempo real invertido. Esto nos ayuda a calcular el margen de error en nuestras estimaciones y a retroalimentar futuras planificaciones de proyectos.

La visualización de la distribución temporal del proyecto se facilita mediante el uso del **diagrama de Gantt**. Este diagrama, presente en la *Figura 17. Diagrama de Gantt,* brinda una representación clara de las distintas etapas del proyecto, así como de las tareas necesarias para su conclusión. Además, se emplea para realizar la planificación de tareas, establecer su secuencia temporal y definir las dependencias entre ellas.

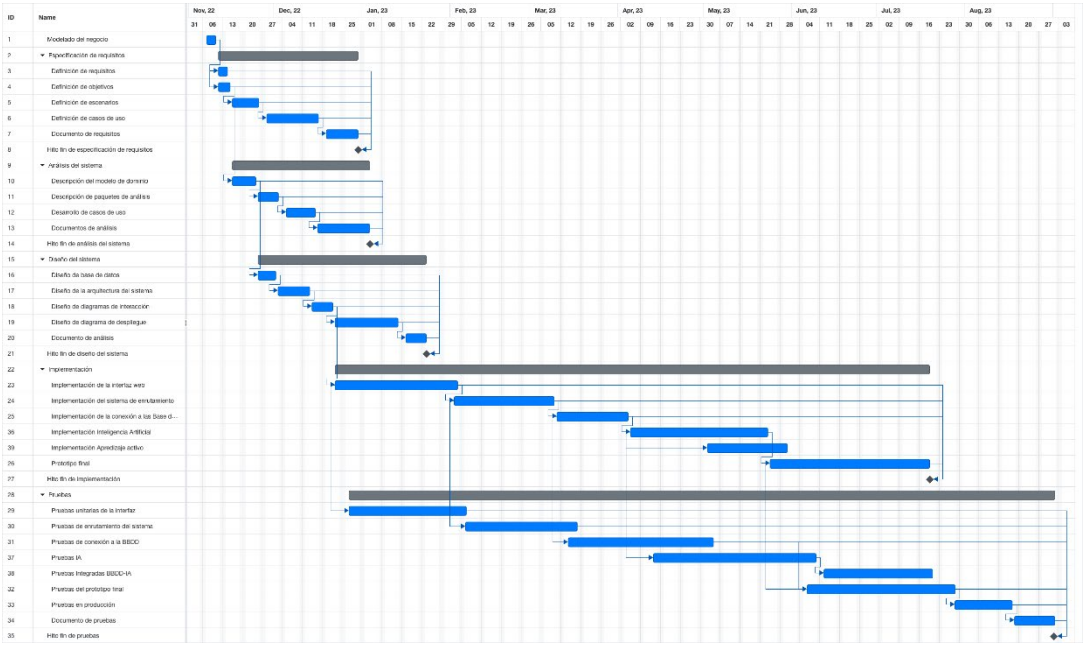

Figura 17. Diagrama de Gantt

5.4. Especificación de requisitos

A continuación, se exponen una serie de tablas que mediante la metodología de Durán y Bernárdez exponen los requisitos de los casos de uso, los requisitos no funcionales, los requisitos de información, así como de otros requisitos del sistema.
Las plantillas utilizadas para realizar la elicitación de requisitos han sido extraídas del programa REM (REqueriments Manager) creado y distribuido por la Universidad de Sevilla.

En el caso de precisar más información sobre la especificación de los requisitos del sistema puede encontrar todo el detalle en el Anexo II.

OBJ-0002	Gestión de datasets
Versión	1.0 (15/11/2022)
Autores	José María Martín Ramos
Fuentes	-
Descripción	El sistema deberá *permitir crear datasets a partir de un pequeño formulario.*
Subobjetivos	• **[OBJ-0003] Seleccionar imagen a clasificar**: El sistema deberá seleccionar la mejor imagen a clasificar dentro del dataset. • **[OBJ-0004] Almacenar modelo**: El sistema deberá guardar el modelo generado tras la clasificación parcial o total de las imágenes que forman el dataset. • **[OBJ-0005] Ejecutar la predicción de imágenes**: El sistema deberá predecir la clasificación de nuevas imágenes cargadas en el dataset.
Importancia	Vital

Tabla 7. Objetivo. Gestión de datasets.

ACT-0003	Usuario registrado
Versión	1.0 (15/11/2022)
Autores	José María Martín Ramos
Fuentes	-
Descripción	Este actor representa *a los usuarios que pueden iniciar sesión en el sistema*
Comentarios	Ninguno

Tabla 8. Actor. Registrado

NFR-0002	Precisión en la creación de modelo
Versión	1.0 (15/11/2022)
Autores	José María Martín Ramos
Fuentes	-
Dependencias	• [OBJ-0002] Gestión de datasets • [OBJ-0004] Almacenar modelo • [IRQ-0004] Modelos • [UC-0010] Crear modelo de dataset • [UC-0011] Ejecutar predicción
Descripción	El sistema deberá *ser preciso a la hora de generar el modelo del dataset para poder entregar los máximos resultados en la predicción.*
Importancia	Vital
Urgencia	Hay presión
Estado	Validado
Estabilidad	Media
Comentarios	Ninguno

Tabla 9. Requisito no funcional. Precisión en la creación de modelo.

UC-0011	Ejecutar predicción	
Versión	1.0 (12/11/2022)	
Autores	José María Martín Ramos	
Fuentes	-	
Dependencias	• [OBJ-0002] Gestión de datasets • [OBJ-0005] Ejecutar la predicción de imágenes • [OBJ-0006] Gestión de imágenes • [IRQ-0002] Datasets • [IRQ-0003] Imágenes • [IRQ-0004] Modelos • [NFR-0002] Precisión en la creación de modelo • [CRQ-0002] Datasets • [UC-0015] Obtener datos de imagen	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el usuario decida realizar la predicción del resto de imágenes presentes en la dataset y aún no han sido clasificadas.	
Precondición	El usuario debe haber iniciado sesión y debe existir un modelo generado para la dataset.	
Secuencia normal	Paso	Acción
	1	El actor Usuario registrado [ACT-0003] inicia el caso de uso.
	2	El sistema ejecuta una predicción para aquellas imágenes que pertenecen al dataset y no tienen una label asociada.
	3	El sistema asigna las correspondientes predicted_label a las imágenes procesadas.
	4	El sistema finaliza el caso de uso.
Postcondición	-	
Excepciones	Paso	Acción
	2	Si todas las imágenes del dataset tienen una predicted_label establecida, el caso de uso se queda sin efecto.
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	-	
Importancia	Importante	
Urgencia	Puede esperar	
Estado	Validado	
Estabilidad	Alta	
Comentarios	Ninguno	

Tabla 10. Caso de uso. Ejecutar predicción.

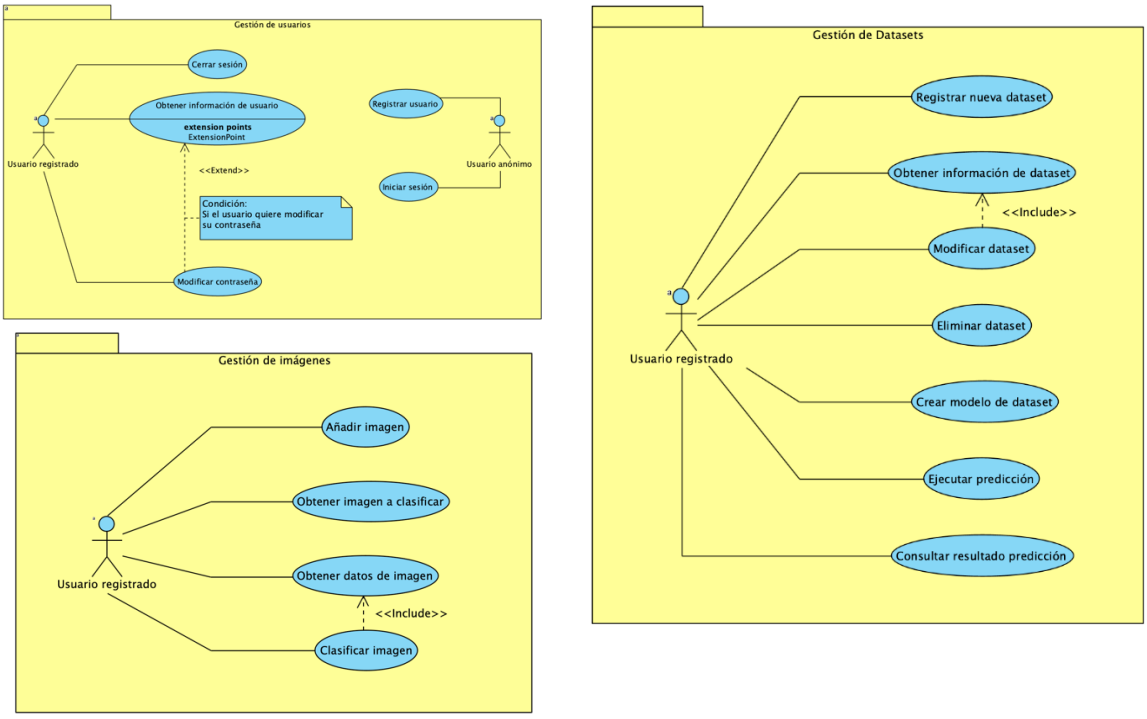

Figura 18. Paquetes de casos de uso.

5.5. Análisis del sistema

En esta fase se realiza un análisis a partir de todos los requisitos diligenciados en las anteriores etapas.

En el caso de requerir más información se puede encontrar en el Anexo III.

En la Figura 3 se puede observar el modelo de dominio, el cual sirve como representación inicial del dominio del problema, analizando las clases principales que compondrán la solución.

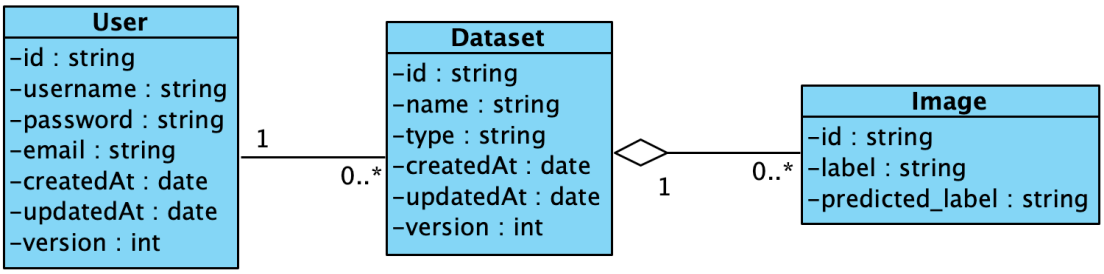

Figura 19. Diagrama de modelo de dominio

En la *Figura 20. Arquitectura de paquetes de análisis.* se representan las clases de análisis que componen cada paquete de modo que se pueda ver en detalle la interacción entre los elementos de análisis, dando así una primera vista de la arquitectura del sistema

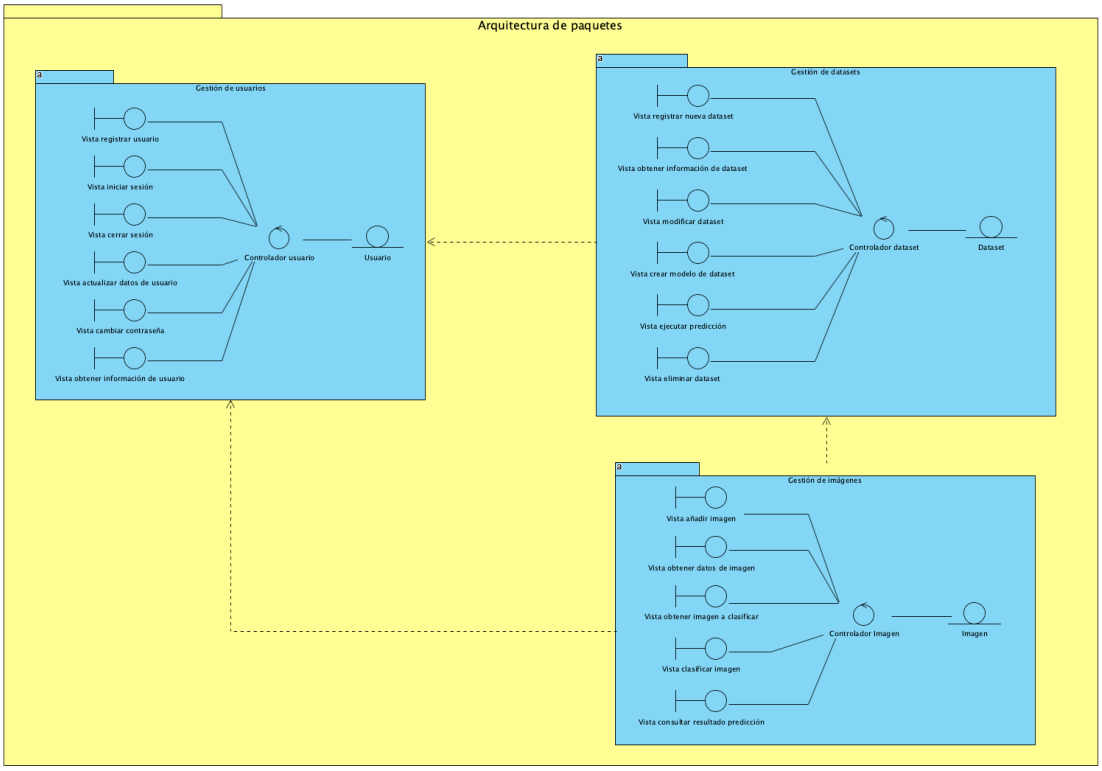

Figura 20. Arquitectura de paquetes de análisis.

Los diagramas de secuencia, como el presente en la *Figura 21. Diagrama de secuencia. Registrar nueva dataset,* permiten ver la realización de los casos de uso del sistema dando así una idea de la interacción del usuario que los realiza y de su complejidad.

Se visualiza la interacción del actor con la vista, estableciendo una comunicación entre ambos. A su vez, la vista se comunica con el controlador, el cual interactúa con el modelo. Estos diagramas proporcionan una representación de alto nivel de la interacción entre los diferentes componentes del sistema. Posteriormente, estos diagramas serán refinados en el nivel de diseño, donde se podrá observar de manera más detallada la interacción del usuario con los componentes específicos de los paquetes de diseño.

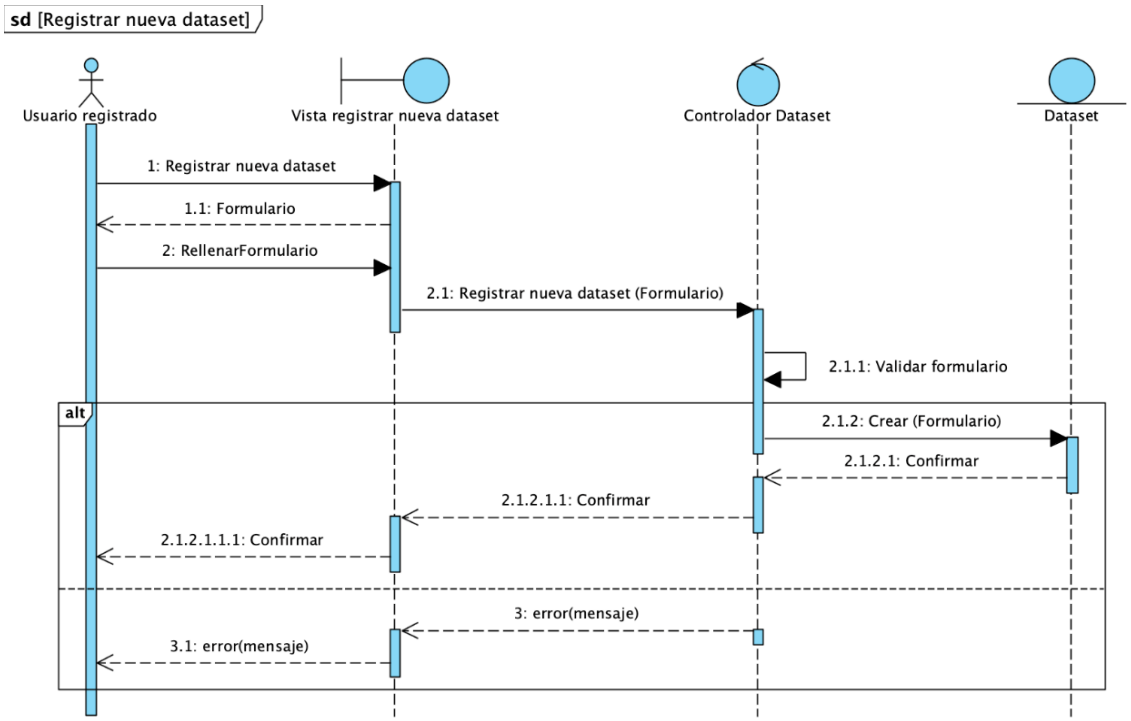

Figura 21. Diagrama de secuencia. Registrar nueva dataset.

5.6. Diseño del sistema

El primer apartado presenta la arquitectura del sistema global, representada mediante el patrón arquitectónico MVC. Además, se detalla la composición interna de los paquetes que conforman dicha arquitectura, así como las relaciones entre ellos y los archivos de código que los componen.

En el siguiente apartado se presenta el diagrama de despliegue del sistema, el cual muestra la distribución física y de software de los elementos que componen el sistema, así como sus interrelaciones.

A continuación, se incluyen una serie de diagramas de secuencia que representan la interacción y los mensajes relacionados con los casos de uso expuestos en el Anexo II. Estos diagramas profundizan en el nivel de análisis descrito en el Anexo III.

Finalmente, se proporciona una explicación detallada del diagrama de la base de datos del sistema, acompañado de una leyenda que aclara los elementos que lo componen.

5.6.1. Patrón arquitectónico

En este proyecto, se implementará el patrón arquitectónico Modelo-Vista-Controlador (MVC). Este patrón divide el sistema en tres partes claramente diferenciadas, separando la interfaz de usuario de la funcionalidad y los datos subyacentes, cuyo esquema se ve representado en la *Figura 22. Arquitectura MVC.* Estas partes son:

- Modelo: Contiene los datos que se mostrarán al usuario en la vista.
- Vista: Presenta los componentes con los que el usuario interactuará y muestra los datos del modelo.
- Controlador: Contiene la lógica del sistema y se encarga de modificar los datos en función de las interacciones del usuario.

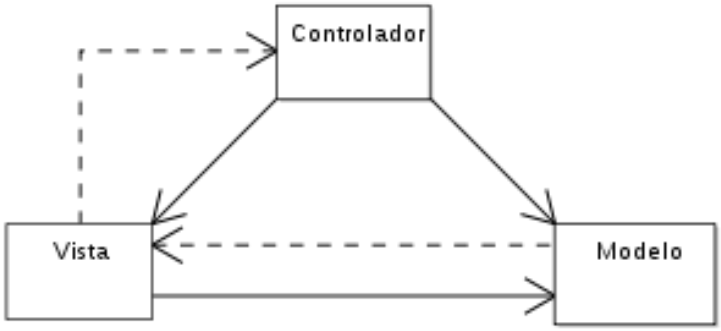

Figura 22. Arquitectura MVC.

5.6.2. Subsistemas de diseño

En estos diagramas se representa la estructura del sistema mediante la división en componentes y las relaciones entre ellos. A continuación, se muestra un ejemplo en la *Figura 23. Diagrama de subsistemas de vistas,* que ilustra el subsistema de vistas. Este diagrama proporciona una visión general de cómo se organiza y se relaciona cada componente en el sistema.

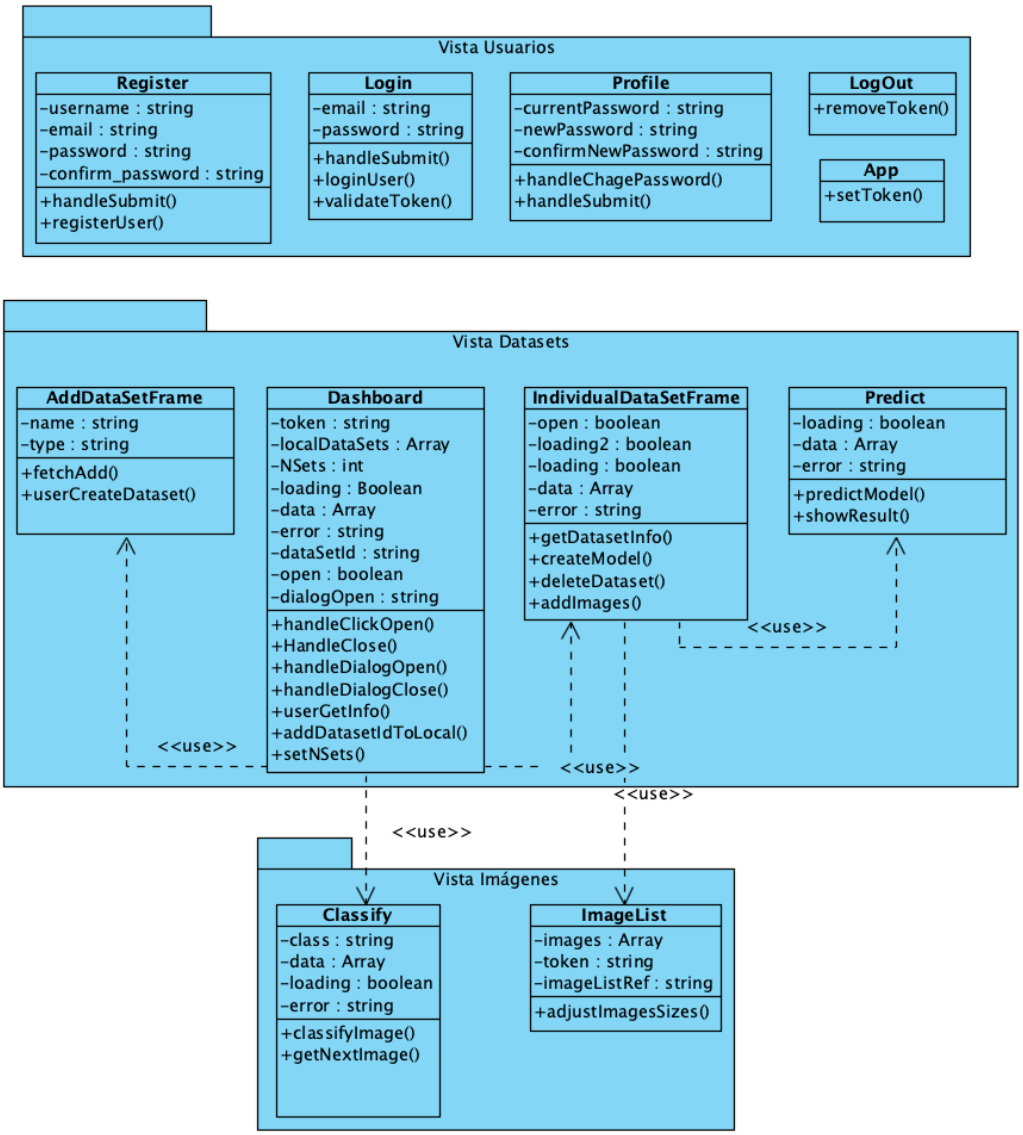

Figura 23. Diagrama de subsistemas de vistas.

5.6.3. Diagrama de despliegue

El diagrama de despliegue ilustra la distribución de los elementos del sistema en su entorno de producción. Cada nodo en el diagrama representa un componente de hardware o software que forma parte del sistema.

En el centro del diagrama, se encuentra el servidor web que aloja la aplicación web y puede ser desplegado en diferentes tipos de servidores físicos.

Los nodos laterales izquierdos representan las API REST que el sistema consulta en ciertas partes de su funcionalidad.

Los nodos conectados en la parte superior representan los diferentes dispositivos en los cuales se puede visualizar la aplicación web. En los ordenadores de sobremesa, la plataforma puede ser accedida como una página web. En dispositivos móviles, la web se puede acceder tanto como una página web convencional o como una PWA (Progressive Web App).

El último nodo del diagrama *(Figura 24. Diagrama de despliegue)* representa la base de datos MongoDB, que se utiliza para la autenticación de usuarios y el almacenamiento de datos y fotos de los usuarios.

Es importante destacar que todos los nodos se conectan a través de comunicaciones HTTPS para garantizar la seguridad de los datos.

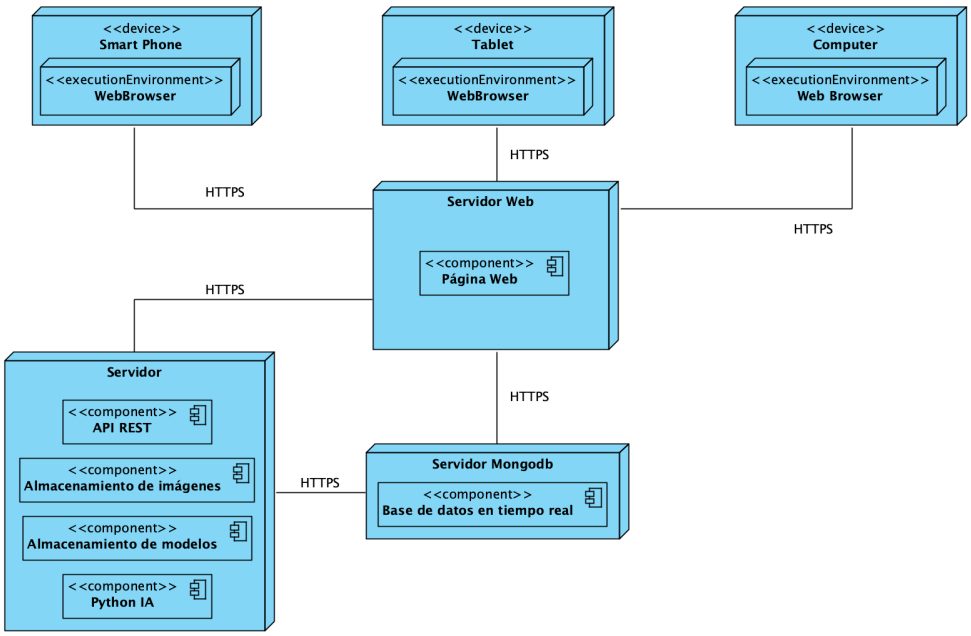

Figura 24. Diagrama de despliegue

5.6.4. Realización de casos de uso

La implementación de los casos de uso se representa a través de diagramas de secuencia que muestran los mensajes intercambiados entre los componentes del sistema.

A continuación, se presenta un ejemplo de la implementación del caso de uso de eliminación de una dataset. Ref. *Figura 25. Diagrama de secuencia. Eliminar dataset*.
En la siguiente figura, se muestran los distintos componentes implicados, así como los mensajes enviados entre ellos.

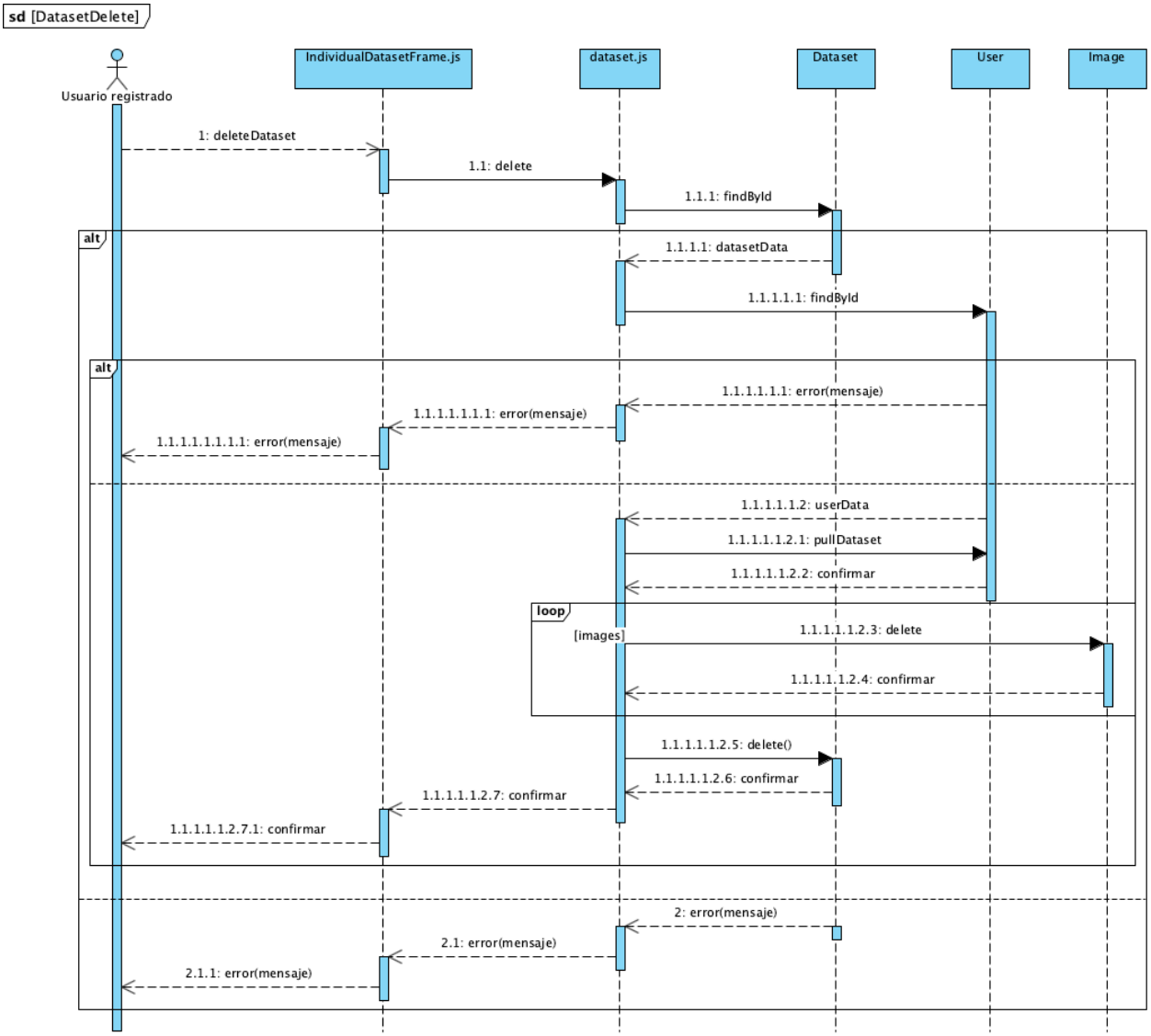

Figura 25. Diagrama de secuencia. Eliminar dataset.

5.6.5. Diagrama de base de datos

Este diagrama, presente en la *Figura 26. Diagrama de base de datos*, proporciona una visión general de la organización de los datos en la base de datos, incluyendo las colecciones, los campos y las relaciones entre ellos. Ayuda a comprender la estructura de la base de datos y cómo se almacenan y relacionan los datos en el sistema.

Eligiendo MongoDB como base de datos en la nube, se ha diseñado un diagrama de base de datos que representa la estructura y relaciones de los datos en el servicio de MongoDB. A continuación, se muestra el diagrama de base de datos utilizado en la plataforma:

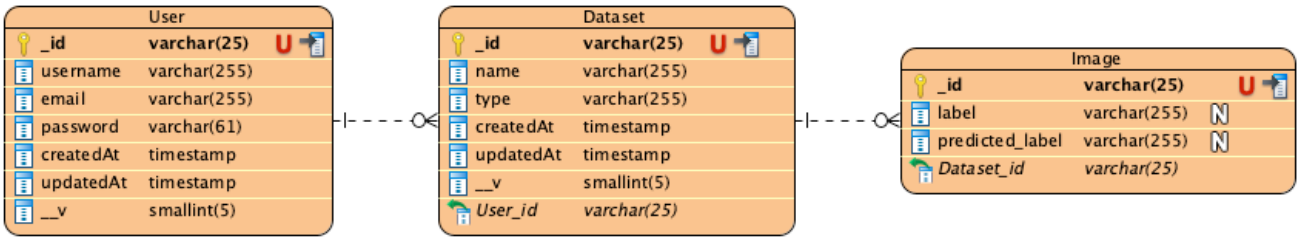

Figura 26. Diagrama de base de datos.

5.7. Implementación de la IA

Durante el proceso de desarrollo de este proyecto, he trabajado en la implementación de una Inteligencia Artificial especializada en la clasificación, generación y predicción de modelos para cada uno de los conjuntos de datos disponibles para el usuario. Este enfoque modular ha sido fundamental para proporcionar un sistema versátil que se adapta a diversas necesidades del usuario.

En cuanto a la clasificación de datos, se desarrollaron algoritmos avanzados de aprendizaje activo que permiten identificar de forma precisa la mejor de las imágenes a clasificar. Estos algoritmos se basaron en la interpretación del contenido de las diferentes imágenes contenidas en un determinado dataset.

En lo que respecta a la generación de modelos, se implementaron algoritmos de generación automática de modelos que permiten la creación de modelos predictivos personalizados a partir de las imágenes ya clasificadas. Estos algoritmos se basaron en técnicas de preprocesamiento y redes neuronales.

En el proceso de implementación, se utilizaron una variedad de librerías y herramientas de código abierto que facilitaron el desarrollo y la optimización de los algoritmos. Entre las librerías más destacadas se incluyen TensorFlow, PyMongo, OpenCV y Pandas.

Aprendizaje Activo

En un primer momento de la implantación, como se puede observar en la FIGURA se crea una conexión a una base de datos MongoDB utilizando la biblioteca **pymongo** a través de la instancia **MongoClient**. Los detalles de la conexión, como la URL de la base de datos y las credenciales de acceso, están codificados directamente en la función.

```
client = MongoClient(
    "mongodb+srv://dbUser:pssw@cluster0.xxxx.mongodb.net/myFirstDatabase?retryWrites=true&w=majority")
db = client['myFirstDatabase']
```

Figura 27. Conexión Python a MongoDB.

Una vez establecida la conexión, se utiliza la biblioteca **pandas** para crear un DataFrame a partir de los datos recuperados de la colección "datasets" en MongoDB cuyo ID sea el mismo que el ID del dataset especificado como argumento a la función. El DataFrame resultante tiene dos columnas: "_id" y "label".

```
data_frame = pd.DataFrame(list(db.datasets.find({'_id': ObjectId(dataset)}))[0]['images'], columns=['_id', 'label'])
```

Figura 28. Filtrado de documentos por ObjectID.

Para llevar a cabo este punto, he empleado una combinación de dos estrategias previamente mencionadas en anteriores apartados: Muestreo Aleatorio (Random Sampling) y Cambio Esperado en el Modelo (Expected Model Change). La elección de cuál estrategia aplicar dependerá de la proporción de imágenes clasificadas en el conjunto de datos. Cuando esta proporción es inferior al 0.1, optamos por utilizar la estrategia de Muestreo Aleatorio. En este caso, seleccionamos una imagen no clasificada de forma completamente aleatoria.

Por otro lado, cuando la proporción de imágenes clasificadas supera el umbral del 0.1, preferimos emplear la estrategia de "Cambio Esperado en el Modelo" (Expected Model Change). Esta estrategia nos permite seleccionar aquellas imágenes que se espera que tengan un mayor impacto en la mejora del modelo después de ser clasificadas.

Esta combinación de estrategias nos permite optimizar el proceso de selección de muestras en función de la disponibilidad de datos clasificados y garantizar un enfoque eficiente en la mejora de nuestro modelo.

Estrategia de Random Sampling

Para implementar esta estrategia, se realizará la selección de imágenes no clasificadas de manera aleatoria hasta que se alcance el umbral especificado. Esta fase es crucial para garantizar que se pueda explorar la totalidad de las clases o etiquetas que se trabajarán en este conjunto de imágenes en el contexto de la Inteligencia Artificial. De esta manera, se busca evitar posibles errores de clasificación en el futuro.

```
if len(data_frame[data_frame['label'] == 'NaN']) / len(data_frame) > 0.9:
    return data_frame.sample()['_id'].to_string(index=False)
```

Figura 29. Selección aleatoria de la imagen.

Estrategia de Expected Model Change con MS-SSIM

El MS-SSIM es una métrica utilizada en procesamiento de imágenes para medir la similitud estructural entre dos imágenes, pero no se relaciona directamente con la calidad de la imagen en términos de su percepción visual. El MS-SSIM evalúa cómo las estructuras y los detalles en una imagen se comparan con los de otra imagen y proporciona una medida de cuán similares son en términos de su contenido estructural.

1. Cálculo de MS-SSIM

Una vez que tenemos el conjunto de imágenes definido y cerrado, comenzamos por calcular el valor de MS-SSIM para todas las imágenes no etiquetadas en el conjunto y contra todas las imágenes clasificadas o no del conjunto. El resultado de estos cálculos serán todas las distancias lógicas desde las imágenes no clasificadas, lo cual se podría representar en un diagrama como el siguiente, ampliando la cantidad de clases a las posibles etiquetas a asignar.

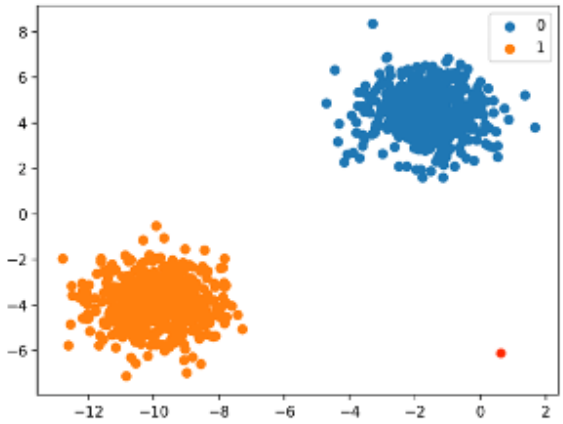

Figura 30. Diagrama de dispersión

2. Selección de Ejemplos Inciertos

Seleccionamos la imagen con el valor MS-SSIM más elevado del conjunto, como se puede observar en la *Figura 8. Diagrama de dispersión,* donde hay dos clases definidas (naranja y azul), el punto rojo sería el elegido para ser clasificado dado que se sitúa distanciado de las dos clases del conjunto, por lo que tras su clasificación el modelo actual se verá modificado para albergar esta nueva casuística.

Primero, se carga cada imagen en escala de grises (flags=cv2.IMREAD_GRAYSCALE) y se redimensionan a 500x500 píxeles utilizando interpolación de área (interpolation=cv2.INTER_AREA). Esto nos generará un conjunto de datos armónico a nivel de características para todas las imágenes a evaluar.

```
data_frame['imread'] = data_frame['_id'].apply(
    lambda x: cv2.resize(cv2.imread('/Users/chemamartin/Documents/classification-app/backend/files/images/' + x.__str__() + '.jpg',
                                    flags=cv2.IMREAD_GRAYSCALE), (500, 500), interpolation=cv2.INTER_AREA)
)
```

Figura 31.Preprocesado y armonización de imágenes.

A continuación, se calcula la similitud entre las imágenes utilizando el índice SSIM (Structural Similarity Index). Se itera a través de las imágenes no etiquetadas y se compara cada una con las demás imágenes para encontrar la imagen con la mayor similitud, la cual se devuelve como resultado.

```
if len(data_frame[data_frame['label'] == 'NaN']) / len(data_frame) > 0.8:
    max_simi = 0
    max_id = ''
    for i, row in data_frame[data_frame['label'] == 'NaN'].iterrows():
        similarity = 10
        for i2, row2 in data_frame[data_frame['_id'] != row['_id']].iterrows():
            _msssim = msssim(row['imread'], row2['imread'])
            if _msssim > 0.9: break
            if similarity > _msssim:
                similarity = _msssim
            if max_simi > _msssim: break
        if max_simi < similarity:
            max_id = row['_id']
            max_simi = similarity
        row['similarity'] = similarity
    return str(max_id)
```

Figura 32. Algoritmo de selección de imagen por máxima similitud.

En el caso de que la proporción de imágenes clasificadas supere el 0.2, se calcula la similitud de manera similar, pero se busca la imagen con la menor similitud para ser clasificada.

```
else:
    min_simi = 1
    min_id = ''
    for i, row in data_frame[data_frame['label'] == 'NaN'].iterrows():
        similarity = 0
        for i2, row2 in data_frame[data_frame['_id'] != row['_id']].iterrows():
            _msssim = msssim(row['imread'], row2['imread'])
            if _msssim > 0.9: break
            if similarity < _msssim:
                similarity = _msssim
            if min_simi < _msssim: break
        if min_simi > similarity:
            min_id = row['_id']
            min_simi = similarity
        row['similarity'] = similarity
    return str(min_id)
```

Figura 33. Algoritmo de selección de imagen por mínima similitud.

Generación De Modelos

Del mismo modo que en el punto anterior, se establece una conexión a la base de datos MongoDB utilizando la biblioteca **pymongo**.
A continuación, se configura un generador de datos de imagen utilizando **ImageDataGenerator** de TensorFlow. Este generador se encarga de realizar transformaciones en las imágenes de entrenamiento, como el preprocesamiento y la

aplicación de aumentos de datos, como el recorte, el zoom y la inversión horizontal. También se establece una división de validación del 40%.

```
train_datagen = ImageDataGenerator(preprocessing_function=preprocess_input, shear_range=0.2, zoom_range=0.2,
                                   horizontal_flip=True, validation_split=0.4)
```

Figura 34. Configuración del ImageDataGenerator.

Igual que en la FIGURA se obtiene la dataset a trabajar filtrando el conjunto por el determinado ID. Una vez extraído el correspondiente DataFrame, se añade ".jpg" a cada uno de los IDs de las imágenes, para así, generar la ruta de trabajo para la misma y se filtra dicho DataFrame para quedarnos únicamente con las imágenes que ya han sido clasificadas y por tanto disponen de una etiqueta en el campo "label".

Si existen imágenes con etiquetas asignadas, se crea un DataFrameIterator utilizando **ImageDataGenerator**. Este generador toma los datos del DataFrame filtrado para garantizar que solo se utilicen ejemplos con etiquetas válidas ('NaN' no se considera válida en este contexto). El generador de datos se configura para tomar las imágenes del directorio específico y utilizar la columna '_id' como datos de entrada, mientras que la columna 'label' se utiliza como las posibles categorías. También se habilita la opción de mezclar los datos durante el entrenamiento.

```
flow_from_dataframe(dataframe=data_frame[data_frame['label'] != 'NaN'],
                    directory="../backend/files/images/",
                    x_col="_id", y_col="label", class_mode="categorical",
                    shuffle=True)
```

Figura 35. Configuración del método flow_from_dataframe

Una vez explorado el generador desde el DataFrame, se crea un modelo de red neuronal basado en la arquitectura ResNet50, que se carga con pesos preentrenados del conjunto de datos ImageNet. Esta técnica aprovecha los conocimientos y las características aprendidas por una red neuronal en un conjunto de datos grande y diverso (ImageNet, en este caso) y los adapta a un nuevo conjunto de datos específico.

Se agrega una capa de agrupación global promediada (**GlobalAveragePooling2D**), la cual calcula la media en la salida de la última capa convolucional, lo que conserva información importante. Seguida de una capa densa con 1024 unidades y función de activación ReLU, aumentando la capacidad del modelo para aprender características complejas y patrones en las imágenes. Especialmente útil cuando se trabajan varias clases.

Finalmente, se agrega una capa densa con un número de unidades igual al número de clases únicas en el conjunto de datos, con una función de activación softmax para la clasificación multiclase.

```
base_model = ResNet50(include_top=False, weights='imagenet')
x = base_model.output
x = GlobalAveragePooling2D()(x)
x = Dense(1024, activation='relu')(x)
predictions = Dense(data_frame[data_frame['label'] != 'NaN']['label'].unique().size, activation='softmax')(x)
model = Model(inputs=base_model.inputs, outputs=predictions)
```

Figura 36. Definición de la Red Neuronal.

Una vez definida la estructura de nuestra red neuronal, se procede a congelar las capas del modelo base ResNet, dado que esto nos proporcionará estabilidad en el entrenamiento del modelo dado que los pesos asignados en esas capas han sido ya calculados con un conjunto de datos muchas veces mayor al nuestro.

```
for layer in base_model.layers:
    layer.trainable = False
```

Figura 37. Congelación de las capas base del modelo.

A continuación, se compila el modelo utilizando el optimizador 'adam' (Adaptive Moment Estimation), optimizador de gradiente que adapta la tasa de aprendizaje de forma dinámica durante el entrenamiento para evitar quedar atrapado en mínimos locales y la función de pérdida 'categorical_crossentropy', que es comúnmente utilizada para problemas de clasificación multiclase. Además se establece la métrica ('accuracy') como medida de precisión durante el entrenamiento.

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(train_generator, epochs=10)
model.save('../backend/files/h5s/' + dataset + '.h5')
```

Figura 38.Compilación y entrenamiento del modelo.

En este punto, se entrena el modelo durante 10 iteraciones utilizando el método fit.

Finalmente, se guarda el modelo entrenado en un archivo con formato HDF5 (extensión .h5) en una ubicación específica del servidor backend para que pueda seguir siendo accesible en un futuro.

Predicción De Modelos

En el momento de la predicción, partimos del archivo previamente generado en el apartado anterior, el cual se carga utilizando la librería ‘tf.keras.models’.

```
model = tf.keras.models.load_model('../backend/files/h5s/' + dataset + '.h5')
pred = model.predict_generator(predict_generator, steps=10, verbose=1)
predicted_class_indices = np.argmax(pred, axis=1)
labels = dict((v, k) for k, v in train_generator.class_indices.items())
predictions = [labels[k] for k in predicted_class_indices]
data_frame.loc[data_frame['label'] == 'NaN', 'predicted_label'] = predictions
data_frame['_id'] = data_frame['_id'].apply(lambda x: ObjectId(x.__str__().replace('.jpg', '')))
db.datasets.find_one_and_update({'_id': ObjectId(dataset)}, {"$set": {"images": data_frame.to_dict('records')}}, upsert=True)
```

Figura 39. Implementación de la predicción para los modelos.

A continuación, se utiliza (**predict_generator**) para obtener probabilidades de clases a partir del modelo cargado con un número de pasos establecido en 10. Luego, se determina la clase predicha para cada imagen mediante **np.argmax**, utilizando un diccionario que relaciona los índices de clase con las etiquetas del modelo entrenado, obtenido de **train_generator.class_indices**. Estas predicciones se almacenan en la columna 'predicted_label' del DataFrame. Posteriormente, se actualiza la base de datos MongoDB con las etiquetas predichas, empleando **db.datasets.find_one_and_update** para buscar y actualizar el documento correspondiente al dataset. Antes de la actualización, el DataFrame se convierte a un formato de diccionario de registros mediante **data_frame.to_dict('records')**.

5.8. Pruebas

Durante el desarrollo del sistema se han llevado a cabo pruebas incrementales para abordar problemas que surgieron en diferentes fases del desarrollo. Estas pruebas se centraron tanto en la funcionalidad interna del sistema como en su funcionamiento en diversos escenarios que un usuario puede encontrarse.

Estas pruebas fueron fundamentales para asegurar el correcto funcionamiento del sistema a lo largo de su desarrollo, detectando y corrigiendo problemas a medida que surgían, y garantizando que el producto final cumpliera con los requisitos y expectativas establecidos.

5.8.1. Prueba registrar usuario

Prueba realizada para comprobar que un usuario anónimo es capaz de registrarse en el sistema de forma correcta.

Primero el usuario rellena el formulario representado en la *Figura 40. Registrar nuevo usuario,* informando todos los campos que son obligatorios. Cuando ha finalizado de rellenarlos y pulsa sobre el botón “Sign Up” el sistema valida los datos introducidos y en el caso de cumplir los requisitos el usuario es redirigido a la página principal de la aplicación web.

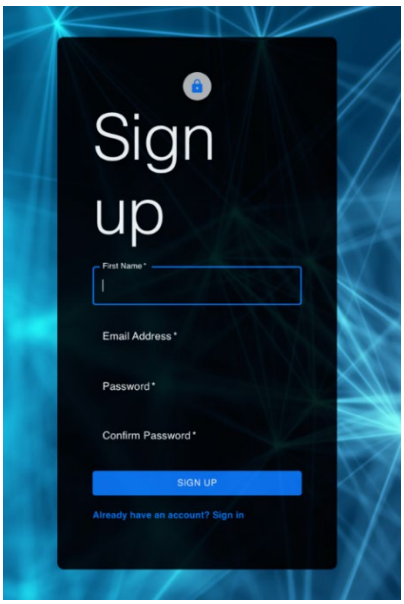

Figura 40. Registrar nuevo usuario.

5.8.2. Prueba modificar contraseña

Prueba realizada para validar que los usuarios una vez acceden al sistema pueden realizar el cambio de la contraseña.

En un primer momento solo se hacía necesaria la nueva contraseña, pero según ha avanzado el proyecto se ha podido observar que la contraseña actual también era necesaria junto con una contraseña de confirmación del cambio.

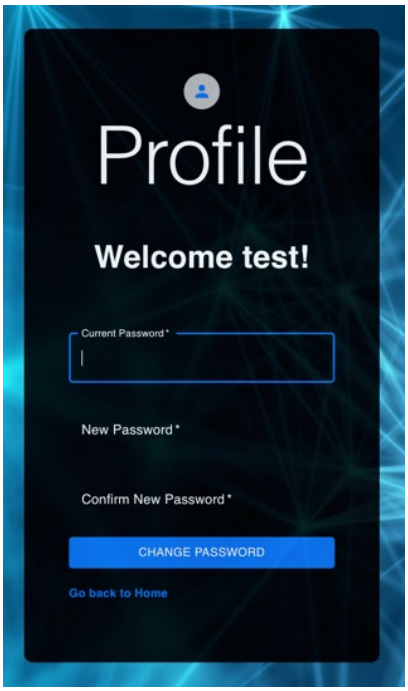

Figura 41. Cambiar contraseña.

El usuario debe completar correctamente los datos en el formulario de la figura anterior, cuando pulsa sobre el botón “Change Password” la aplicación solicita confirmación antes de efectuar el cambio a través de una ventana emergente, como observamos en la *Figura 42. Confirmar cambio de contraseña,* la misma que servirá para notificar en caso de fallo al realizar el cambio de contraseña.

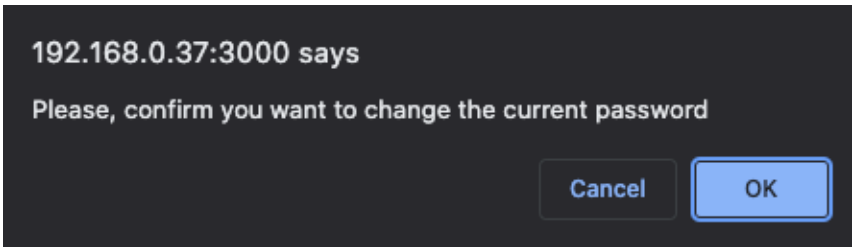

Figura 42. Confirmar cambio de contraseña.

5.8.3. Prueba registrar nueva dataset

Prueba realizada para comprobar que al registrar una nueva dataset en el sistema se realizaba de forma consistente.

En un primer momento se había planteado introducir esta nueva dataset en el objeto del propio usuario; pero se pudo observar que si se realizaba de esta manera se perdía manejabilidad en los datos asociados.

La primera funcionalidad que se implantó fue a través de un formulario donde el usuario tenía que informar tanto el título de este nuevo dataset como el conjunto de etiquetas separadas por un delimitador para que en el momento de la clasificación le aparecieran las diferentes opciones.

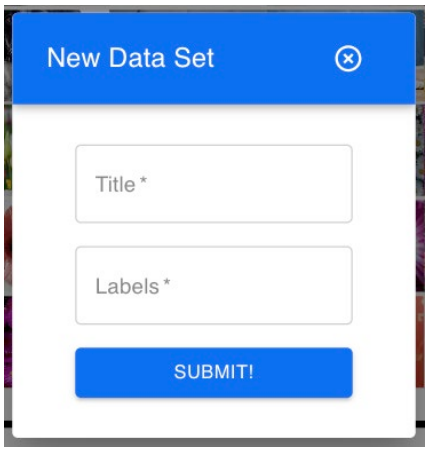

Figura 43. Diálogo añadir dataset.

Dado que el desarrollo del sistema como se ha comentado en el párrafo anterior tenía implícito limitar el alcance de la funcionalidad de la aplicación, se ha optado por remover esta limitación para que el dataset sea incremental a nivel de posibles etiquetas asignadas a las imágenes.

5.8.4. Prueba eliminar dataset

Prueba realizada para validar el correcto funcionamiento en el momento que un usuario registrado decide eliminar un dataset haciendo uso del botón “Delete”.

Al principio esta funcionalidad estaba planteada para eliminar únicamente los registros de la dataset almacenados en la base de datos, cuya implementación fue completamente directa haciendo uso de los correspondientes métodos de la librería. Después se modificó esta funcionalidad para que eliminara también todas las imágenes físicas asociadas a la dataset y así limpiar todo rastro de la aplicación.

5.8.5. Prueba crear modelo de dataset

Se ha creado esta prueba para comprobar que los usuarios pueden generar los modelos de las diferentes datasets.

Cuando el usuario pulsa sobre la opción “Create Model” o cuando intenta hacer una predicción sin haber generado antes un modelo, el sistema lo genera de forma interna.

Esta es una prueba bastante sencilla, pues las interacciones con el usuario no son excesivas; pero es un paso crucial para poder darle un sentido al sistema.

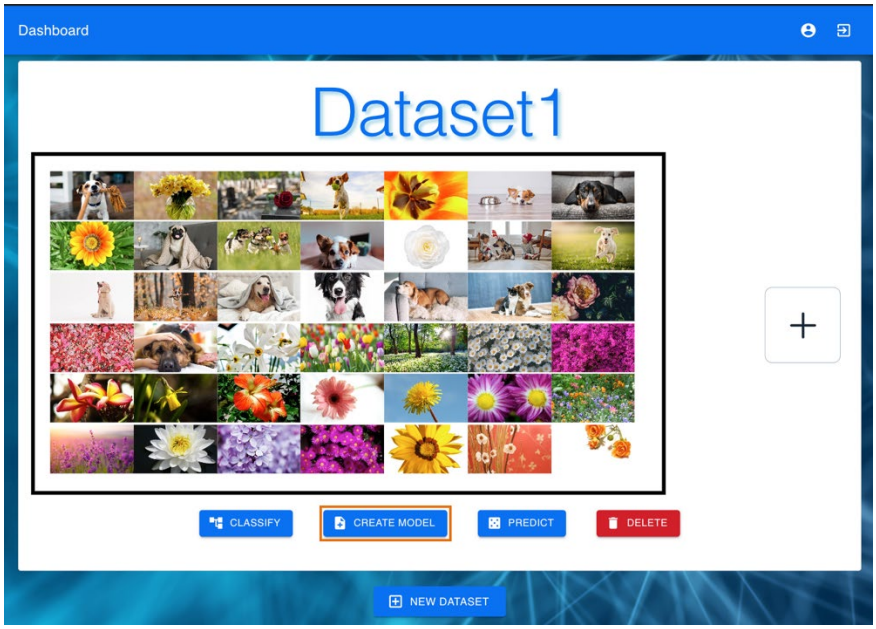

Figura 44. Crear modelo de dataset.

En la *Figura 45. Acciones deshabilitadas* se observa que desde que se inicia la acción hasta que finaliza con la creación del modelo, el resto de acciones permanecen deshabilitadas para que no interfieran.

Figura 45. Acciones deshabilitadas.

5.8.6. Prueba añadir imagen

Prueba realizada para comprobar que un usuario puede subir imágenes y estas se ven correctamente reflejadas en el almacenamiento del sistema.

En un primer momento se comenzó a desarrollar la subida de las imágenes a servicios en nube para no tener que almacenar de manera local todas las imágenes. Pero tras unas etapas iniciales del desarrollo se tuvo que desestimar este despliegue dado que producía problemas en el procesado de la inteligencia artificial.

Finalmente se ha optado por un módulo de React que nos permitía arrastrar y seleccionar múltiples imágenes de una sola carga y a la vez nos muestra las previsualizaciones mientras se está realizando la subida como se observar en la *Figura 46. Previsualizaciones al subir imágenes*.

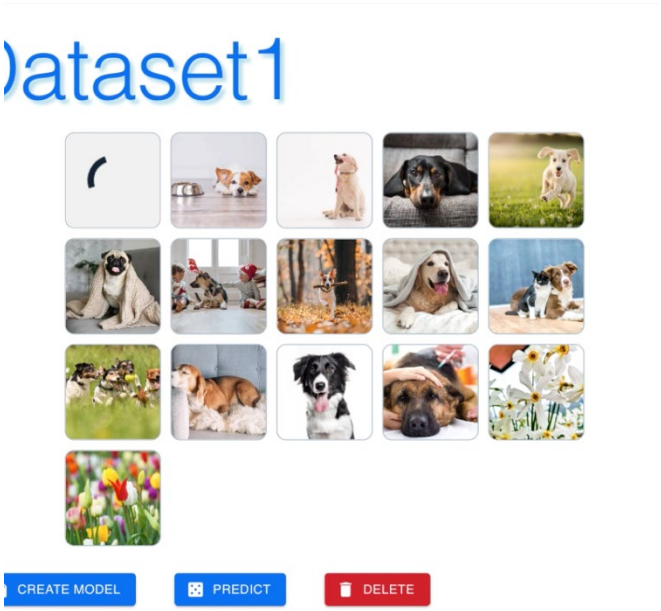

Figura 46. Previsualizaciones al subir imágenes.

5.8.7. Prueba clasificar imagen

Prueba focalizada en validar el funcionamiento correcto a la hora de clasificar imágenes por porte de los usuarios.

En un principio se planteó darle al usuario la opción de elegir la imagen a clasificar para facilitar las pruebas y el desarrollo de este módulo, finalmente no se llevó a cabo y se optó por seleccionar la imagen de forma aleatoria por parte del sistema y más adelante respetar una fórmula de selección.

Desde la pantalla inicial, el usuario puede hacer uso del botón “Classify”, momento en el cual se le presenta una ventana emergente, como la presente en la *Figura 47. Diálogo para clasificar imagen.* Mientras el sistema analiza el dataset e identifica la mejor imagen a clasificar le aparecerá un icono de cargando.

Una vez finalizada la selección de la imagen, esta se le muestra al usuario junto con un formulario que debe radicar con la correspondiente etiqueta.

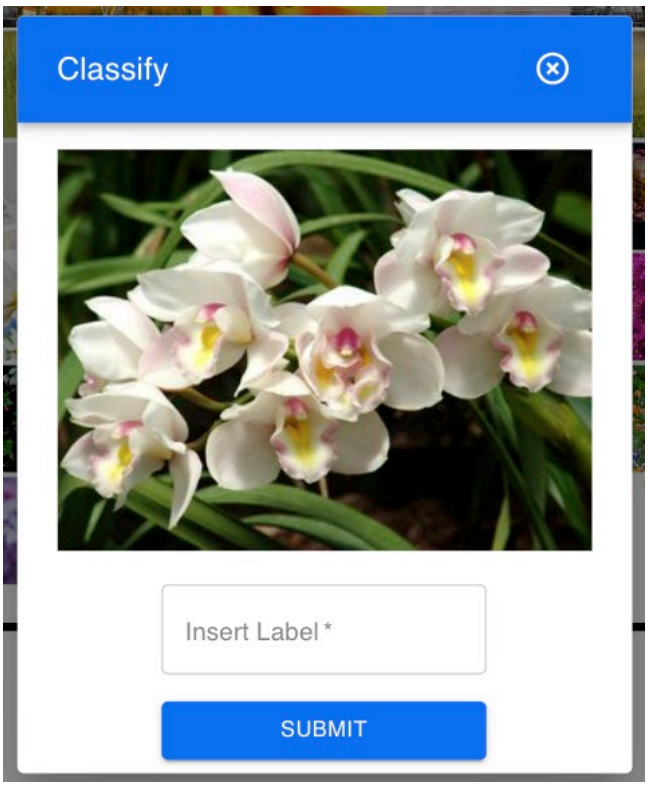

Figura 47. Diálogo para clasificar imagen.

6. Conclusiones

A través de este trabajo de fin de grado, he adquirido valiosos conocimientos y experimentado el desarrollo de un proyecto real, complejo y extenso. Cada uno de los objetivos establecidos ha sido cumplido exitosamente, y a continuación expondré las conclusiones obtenidas de cada uno.

En primer lugar, he tenido la oportunidad de profundizar en estructura y despliegue de una aplicación basada en una API REST que comunique dos partes de la misma. Entendiendo de una mejor manera lo aprendido en la asignatura de Sistemas Distribuidos. He podido aprender la importancia de la sincronización desde que se realiza la llamada hasta que se le da una respuesta, también he podido conocer todas las opciones que esto nos otorga en el día a día.

Durante el desarrollo de esta parte anterior, se ha observado la sencillez que una API REST proporciona a la hora de aislar y analizar los fallos en la aplicación web. Pues se podía analizar la traza generada con un capturador de paquetes o también se podía replicar estas llamadas desde aplicaciones externas, sin tener que realizar todas las navegaciones intermedias a través de la aplicación web.

En un segundo lugar y muy relacionado con el anterior, tenemos el desarrollo de la página web. Este es el punto que más tiempo de me ha llevado dado que no había trabajado nunca con el lenguaje de JavaScript por lo que he tenido que buscar mucha información para poder desplegarlo satisfactoriamente. En un primer momento focalicé el trabajo en desarrollar todas las funcionalidades del sistema, para más adelante aplicarle la capa de estilos con la ayuda de Material-UI. Uno de los puntos que más esfuerzo me ha llevado ha sido llegar a entender los distintos estados en React, para que no se generen bucles infinitos en la carga de las diferentes ventanas.

Por otro lado, tenemos la integración de una base de datos en la nube, MongoDB. Este punto ha sido uno de los más gratificantes en el desarrollo del proyecto dado que he podido observar cómo de útil puede llegar a ser una base de datos en tiempo real, no localizada y no relacional.

La elección de esta estructura de base de datos fue algo comprometida, dado que mi experiencia con bases de datos relacionales es mucho más amplia y esta nueva estructura tendría un lenguaje de programación asociado completamente diferente. Finalmente opté por aprender esta nueva tecnología, para también aprovechar que se encontraba en la nube.

Finalmente, como punto principal de este proyecto, tenemos el desarrollo de la Inteligencia Artificial. Este punto era el que más motivación me generaba de forma interna, dado que en las asignaturas impartidas en la carrera no alcanzaban el despliegue o implementación de una red neuronal. Durante el desarrollo de este punto he tenido que comprender todos los conceptos necesarios para poder llevarlo a cabo como podía ser las funciones de activación, el tamaño del batch o la densidad de cada una de las capas que conforman esta red neuronal.

Como el sistema debe de seleccionar la mejor imagen a clasificar, también aprendí las diferentes opciones que la Visión Artificial nos ofrece a la hora de analizar la estructura y contenido de las imágenes, generando un análisis del conjunto de imágenes.

Durante el desarrollo de este proyecto, adquirí importantes aprendizajes. Descubrí la importancia de realizar no solo una planificación exhaustiva, sino también los diagramas de secuencia y los dibujos preliminares. Estas etapas previas a la programación facilitaron en gran medida la tarea, brindando una visión clara del objetivo a alcanzar y fomentando la reutilización y modularidad de los componentes.

Dado que el sistema se basaba en páginas web, aprendí a diseñar interfaces adaptables a cualquier dispositivo. Esta habilidad me permitió crear una experiencia de usuario consistente y accesible, sin importar el tipo de dispositivo utilizado.

Además, este proyecto me brindó la valiosa oportunidad de aplicar la ingeniería del software en un entorno real, en contraste con ejercicios teóricos basados en enunciados ficticios. Esto me permitió enfrentarme a desafíos reales y adquirir una comprensión más profunda de la disciplina.

Durante el despliegue del proyecto en un servidor, adquirí conocimientos en la configuración de un servidor Express, la implementación de certificados para garantizar conexiones seguras y cifradas. Estas habilidades fueron fundamentales para asegurar la confiabilidad y la privacidad de la aplicación en su entorno de producción.

7. Líneas de trabajo futuras

Este proyecto presenta varias oportunidades de expansión y potencial comercialización. A continuación, se destacan algunos puntos que podrían permitir ampliar la funcionalidad del sistema y explorar oportunidades de mercado:

- Llevar la generación del modelo a servidores de computación en la nube para que pueda entregar un mejor rendimiento en la aplicación
- La funcionalidad de seleccionar la imagen a clasificar se podría migrar de la misma manera.
- Ampliar la funcionalidad de la aplicación, dando uso a APIs externas como puede ser Google Vision AI. Generando por tanto un programa de suscripciones.
- Permitir al usuario clasificar una imagen del conjunto seleccionada manualmente.
- Permitir al usuario personalizar la interfaz, generando una experiencia adaptada al usuario.
- Generar aplicaciones en las principales plataformas móviles como son Android y iOS.
- Ampliar las posibilidades de inicio de sesión a Google o Facebook.
- Implementar un código OTP generado de forma única para iniciar sesión en la aplicación.
- Desarrollar un sistema de suscripciones con limitaciones en las posibles acciones a realizar en la aplicación.

8. Referencias

[1] «¿Qué es una API REST?,» [En línea]. Available: https://www.ibm.com/es-es/topics/rest-apis.

[2] «Material UI,» [En línea]. Available: https://mui.com/.

[3] «Mongo DB,» [En línea]. Available: https://www.mongodb.com/.

[4] «Base de datos no relacional» [En línea]. Available:

https://www.mongodb.com/nosql-explained.

[5] «Deep Residual Learning for Image Recognition» [En línea]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf.

[6] «Multi-Scale Structural Similarity» [En línea].

Available: https://vicuesoft.com/glossary/term/ssim-ms-ssim/.

[7] «Git,» [En línea]. Available: https://git-scm.com/.

[8] «RFC 7519,» [En línea]. Available: https://www.rfc-editor.org/rfc/rfc7519

[9] «PyCharm,» [En línea]. Available: https://www.jetbrains.com/pycharm/.

[10] «WebStorm,» [En línea]. Available: https://www.jetbrains.com/webstorm/.

[11] « Google Chrome DevTools,» [En línea]. Available:

https://developer.chrome.com/docs/devtools/.

[12] «Insomnia,» [En línea]. Available: https://insomnia.rest/.

[13] «PyMongo,» [En línea]. Available: https://github.com/mongodb/mongo-python-driver.

[14] «Sewar,» [En línea]. Available: https://github.com/andrewekhalel/sewar.

[15] «TensorFlow,» [En línea]. Available: https://www.tensorflow.org/.

[16] «Keras,» [En línea]. Available: https://keras.io/.

[17] «Express,» [En línea]. Available: https://expressjs.com/.

[18] «Passport,» [En línea]. Available: https://www.passportjs.org/.

[19] «CORS,» [En línea]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS.

[20] «React,» [En línea]. Available: https://react.dev/.

[21] «Visual Paradigm,» [En línea]. Available: https://www.visual-paradigm.com/.

[22] «Online Gantt,» [En línea]. Available: https://www.onlinegantt.com/#/gantt.

[23] «HTML,» [En línea]. Available: https://developer.mozilla.org/en-US/docs/Web/HTML.

[24] «CSS,» [En línea]. Available: https://developer.mozilla.org/en-US/docs/Web/CSS.

[25] «JavaScript,» [En línea]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript.

[26] «Python,» [En línea]. Available: https://www.python.org/.

[27] «JSON,» [En línea]. Available: https://www.json.org/json-en.html.

[28] «JSX,» [En línea]. Available: https://legacy.reactjs.org/docs/introducing-jsx.html.

[29] «Serve,» [En línea]. Available: https://github.com/vercel/serve.

[30] «UML,» [En línea]. Available: http://www.uml.org/.