

Aplicación para la gestión y mejora de *pipelines* a partir de heurísticas en el sector sanitario

Memoria Trabajo Fin de Grado Ingeniería Informática

Julio 2022



VNiVERSiDAD
D SALAMANCA

Autora:

Esther Andrés Fraile

Tutores:

Andrea Vázquez Ingelmo

Alicia García Holgado

Francisco José García Peñalvo

Certificado de los tutores

Dña. Andrea Vázquez Ingelmo, Dña. Alicia García Holgado y D. Francisco José García Peñalvo, profesorado del Departamento de Informática y Automática de la Universidad de Salamanca.

CERTIFICAN:

Que el trabajo titulado “Aplicación para la gestión y mejora de pipelines a partir de heurísticas en el sector sanitario” ha sido realizado por D. Esther Andrés Fraile, con DNI 70918564-L, para la asignatura “Trabajo de Fin de Grado” de la titulación “Grado en Ingeniería Informática de la Universidad de Salamanca”.

Y para que así conste a todos los efectos oportunos.


En Salamanca, a 7 de Julio de 2022



Dña. Andrea Vázquez Ingelmo
Dpto. Informática y Automática
Universidad de Salamanca



Dña. Alicia García Holgado
Dpto. Informática y Automática
Universidad de Salamanca



D. Francisco José García Peñalvo
Dpto. Informática y Automática
Universidad de Salamanca

Resumen

El *Machine Learning* (ML) o aprendizaje automático es una disciplina que se encuentra presente en muchos ámbitos de nuestra vida, incluido el de la medicina. Se ha convertido en un poderoso enfoque para poder solucionar tareas complejas con gran cantidad de datos sin consumir gran cantidad de tiempo o recursos. En el campo sanitario, la aplicación ML ha demostrado tener una gran utilidad en tareas como el diagnóstico o la detección de enfermedades.

En la actualidad ya existen algunas herramientas de ayuda y aprendizaje de ML que asisten a usuarios no expertos en la definición de *pipelines* de ML. Sin embargo, es difícil adaptarlas a contextos específicos con necesidades y requerimientos particulares.

Este fue el punto de partida de la aplicación web *Koopa ML*, cuyo objetivo es permitir a los usuarios el entrenamiento de modelos propios y el análisis de sus datos sin necesidad de tener conocimientos de programación o *Machine Learning*. Esta nueva herramienta permite al contexto de la salud beneficiarse en tareas como la identificación de patrones o la realización de predicciones.

Sin embargo, dicha aplicación no cumple del todo su objetivo puesto que cuenta con el problema de que muchos de sus usuarios no tienen el conocimiento necesario para comprender los contenidos o las configuraciones que deben utilizar. Centrándonos en el dominio de la salud, este desconocimiento puede resultar altamente peligroso. Una mala interpretación de un resultado podría llevar a conclusiones erróneas e incluso a negligencias médicas.

Es aquí donde se trabajará en el sistema a desarrollar. No solo se permitirá la aplicación de funciones predefinidas de ML si no que se agregará una funcionalidad que ayude a los beneficiarios del proyecto a comprender todos los pasos necesarios para crear un correcto modelo ML, equilibrando así el dominio de datos de los profesionales sanitarios con la experiencia de los expertos en aprendizaje automático.

Se llevará a cabo una interfaz de usuario intuitiva que permitirá analizar y comprender tanto los procesos como los resultados obtenidos de cada *pipeline* mediante un sistema personalizado de recomendaciones, educando así al usuario en el correcto uso de modelos ML a partir de su comprensión del dominio de los datos.

Dichas recomendaciones se llevarán a cabo mediante la utilización de heurísticas cuyo comportamiento podrá modificarse dinámicamente. Además, a lo largo de toda la aplicación se incorporarán ayudas teóricas en la que se dará una explicación detallada de aquello que se consulta.

Así mismo, el diseño se modificará por completo hacia uno mucho más intuitivo y usable, y se incorporarán nuevas características de distinta índole a la aplicación.

Palabras clave: *Machine Learning*, diagrama de flujo, *pipeline*, algoritmo, heurística, recomendación

Abstract

Machine Learning (ML) is a discipline that is present in many areas of our lives, including medicine. It has become a powerful approach to solve complex, data-intensive tasks without consuming large amounts of time or resources. In the healthcare field, the ML application has proven to be very useful in tasks such as diagnosis or disease detection.

Some ML help and learning tools already exist today to assist non-expert users in defining ML pipelines. However, it is difficult to adapt them to specific contexts with particular needs and requirements.

This was the starting point for the *Koopa ML* web application, which aims to allow users to train their own models and analyze their data without the need for programming or machine learning skills. This new tool allows the healthcare context to benefit from tasks such as identifying patterns or making predictions.

However, such an application does not fully meet its objective as it has the problem that many of its users do not have the necessary knowledge to understand the contents or the settings to be used. Focusing on the health domain, this lack of knowledge can be highly dangerous. A misinterpretation of a result could lead to erroneous conclusions and even medical negligence.

This is where work will be done on the system to be developed. It will not only allow the application of predefined ML functions but will also add functionality that will help the beneficiaries of the project to understand all the steps needed to create a correct ML model, thus balancing the data domain of healthcare professionals with the expertise of machine learning experts.

An intuitive user interface will be implemented that will allow the user to analyze and understand both the processes and the results obtained from each pipeline through a personalized system of recommendations, thus educating the user in the correct use of ML models based on their understanding of the data domain.

These recommendations will be carried out through the use of heuristics whose behaviour can be dynamically modified. In addition, throughout the application, theoretical aids will be incorporated in which a detailed explanation of what is being consulted will be given.

The design will also be completely redesigned to be much more intuitive and usable, and new features of various kinds will be added to the application.

Keywords: Machine Learning, flowchart, pipeline, algorithm, heuristic, recommendation, heuristic, recommendation

Tabla de contenido

Certificado de los tutores.....	2
Resumen.....	I
Abstract	II
Lista de figuras	V
Lista de tablas.....	VII
1. Introducción	1
2. Objetivos del proyecto	4
Objetivos del sistema	4
Objetivos personales.....	4
3. Técnicas y herramientas.....	5
Diseño de la solución.....	5
▪ Adobe XD.....	5
▪ Paletas de color	5
▪ Diagrams.net – Draw.io.....	5
Código de programación	6
▪ Python	6
▪ Django	6
▪ HTML, CSS y JavaScript.....	6
▪ Bootstrap.....	6
▪ FlowChart	6
▪ FontAwesome	7
Entorno de desarrollo	7
▪ PyCharm	7
▪ GitLab	7
▪ Postgres.....	7
▪ Slack.....	7
▪ Google Drive.....	8
4. Aspectos relevantes del desarrollo	9
Metodología.....	9
Planificación temporal	11
Especificación de requisitos	12
Proceso de diseño	16
Fase de exploración.....	17

Fase de conceptualización	22
Fase de prototipos y pruebas de usuario	27
Heurísticas.....	37
Campo de texto.....	37
Diagrama	39
Comportamiento dinámico	41
Seguridad.....	46
Código.....	46
Información teórica.....	51
6. Conclusiones y líneas futuras de trabajo	54
Conclusiones	54
Líneas futuras de trabajo	55
Referencias.....	56

Lista de figuras

Figura 1: Ciclo de desarrollo	9
Figura 2: Fases del Proceso Unificado	10
Figura 3: Planificación temporal.....	11
Figura 4: Diagrama de actores	14
Figura 5: Diagrama gestión de recomendaciones.....	15
Figura 6: Diagrama gestión de heurísticas	15
Figura 7: Ciclo de diseño	17
Figura 8: Tarea recomendaciones	23
Figura 9: Tarea Información teórica.....	23
Figura 10: Tarea heurísticas	24
Figura 11: Mapa de sitio del sistema.....	25
Figura 12: Primera versión prototipo digital: Inicio de sesión	28
Figura 13: Primera versión prototipo digital: Restablecimiento de la contraseña	28
Figura 14: Primera versión prototipo digital: Registro.....	28
Figura 15: Primera versión prototipo digital: Listado de proyectos	29
Figura 16: Primera versión prototipo digital: Creación de proyecto	29
Figura 17: Primera versión prototipo digital: Listado de instituciones.....	29
Figura 18: Primera versión prototipo digital: Listado de usuarios.....	30
Figura 19: Primera versión prototipo digital: Cuenta	30
Figura 20: Primera versión prototipo digital: Área de trabajo.....	30
Figura 21: Primera versión prototipo digital: Información	31
Figura 22: Primera versión prototipo digital: Recomendaciones	31
Figura 23: Primera versión prototipo digital: Recomendación completa.....	31
Figura 24: Segunda versión prototipo digital: Recomendaciones	33
Figura 25: Segunda versión prototipo digital: Tooltips	34
Figura 26: Segunda versión prototipo digital: Guardado.....	34
Figura 27: Segunda versión prototipo digital: Área de trabajo.....	34
Figura 28: Segunda versión prototipo digital: Construcción de flujo.....	34
Figura 29: Segunda versión prototipo digital: Heurísticas	35
Figura 30: Diagrama heurística de ejemplo	38
Figura 31: Campo de texto heurísticas.....	38
Figura 32: Diagrama de heurística	39

Figura 33: Interfaz modificación de heurísticas	40
Figura 34: Interfaz listado de heurísticas	40
Figura 35: Interfaz creación de una heurística	41
Figura 36: Interfaz edición de heurística	41
Figura 37: Ejemplo de recomendación.....	41
Figura 38: Toggle activación/desactivación recomendaciones.....	42
Figura 39: Área de trabajo con las recomendaciones activadas.....	42
Figura 40: Área de trabajo con las recomendaciones desactivadas	42
Figura 41: Ejemplo comportamiento de la heurística 1	43
Figura 42: Ejemplo comportamiento de la heurística 2	43
Figura 43: Ejemplo comportamiento de la heurística 3	44
Figura 44: Paso 2: modificación de heurística.....	44
Figura 45: Paso 3: modificación de heurística.....	45
Figura 46: Paso 4: modificación de heurística.....	45
Figura 47: Resultado modificación de heurística	46
Figura 48: Checkbox elección de heurística	46
Figura 49: Icono información	51
Figura 50: Ejemplo de apartados con ayuda	52
Figura 51: Ejemplo ayuda apartado Nodos	52
Figura 52: Ejemplo ayuda apartado Opciones	53
Figura 53: Ejemplo ayuda apartado Nodo Rellenar Valores Perdidos	53

Lista de tablas

Tabla 1: Fechas estimadas según planificación temporal	12
Tabla 2: Ejemplo participante del proyecto	12
Tabla 3: Definición de la audiencia, usuario 1.....	20
Tabla 4: Definición de la audiencia, usuario 2.....	20
Tabla 5: Definición de la audiencia, usuario 3.....	20
Tabla 6: Definición de la audiencia, usuario 4.....	21
Tabla 7: Definición de la audiencia, usuario 5.....	21
Tabla 8: Definición de la audiencia, usuario 6.....	21
Tabla 9: Journey Map.....	26

1. Introducción

El *Machine Learning* (ML) o aprendizaje automático¹ se ha convertido en un enfoque poderoso para poder abordar tareas complejas que implican el análisis de una cantidad significativa de datos. Esta rama de la Inteligencia Artificial² permite a contextos, como el de la salud, beneficiarse en torno al tratamiento de sus datos, a tareas tales como la identificación de patrones o a la realización de predicciones. Sin la ayuda del *Machine Learning* (ML) o aprendizaje automático, estas actividades consumirían mucho más tiempo y recursos. Específicamente, la aplicación de ML³ a los datos sanitarios ha demostrado su utilidad en tareas complejas como el diagnóstico, la detección de enfermedades, la segmentación, la evaluación de las funciones de los órganos, entre otras .

Sin embargo, la aplicación de ML no es sencilla. Si nos centramos en el dominio de la salud, relacionado con el proyecto a desarrollar, nos daremos cuenta de que es un campo muy sensible. Resultaría peligroso que los profesionales de la medicina no comprendieran completamente como utilizar un modelo o como analizar sus resultados.

El aprendizaje automático no consiste únicamente en aplicar un conjunto de funciones predefinidas por parte de bibliotecas externas⁴. Su aplicación necesita de un conocimiento profundo de los datos de entrada, las transformaciones que deben realizarse para ajustar los modelos, la selección de modelos adecuados al dominio del problema y la comprensión de las métricas de calidad antes de utilizar los modelos entrenados en producción⁵. De lo contrario, los resultados podrían llevar a conclusiones erróneas, pérdidas, discriminación e incluso negligencia⁶.

Es en este punto donde se trabajará en el sistema a desarrollar. No solo se permitirá la aplicación de funciones predefinidas de ML si no que se ayudará a los beneficiarios del proyecto a comprender todos los pasos necesarios para construir modelos de ML.

Se equilibrará el conocimiento del dominio datos con la experiencia ML, uniendo así la experiencia de los expertos en ML con la comprensión del dominio de los datos por parte de los profesionales de la salud.

En este sentido, se proporcionará a los profesionales sanitarios herramientas para suplir esa ausencia de conocimiento en ML permitiéndoles implementar *pipelines* mientras se les facilita información teórica sobre lo que están trabajando o se les sugiere métodos de continuación.

Para ello, se presenta una plataforma gráfica llamada *Koopa ML*, una aplicación web enmarcada en el Departamento de Cardiología de Hospital Universitario de Salamanca y que fue desarrollada en un TFG anterior⁷.

Esta plataforma se creó con el objetivo de permitir a los usuarios entrenar sus propios modelos y analizar e interpretar los resultados obtenidos sin tener conocimiento experto en la materia mediante la creación y ejecución de *pipelines* de ML.

Sin embargo, se detectó que los usuarios no expertos en programación y ML no eran capaces de entender del todo cómo funcionaba la herramienta gráfica debido a su falta de conocimiento en

los contenidos o en las configuraciones a utilizar en los flujos de ML. Aunque *Koopa ML* le quita complejidad a la tarea de crear los flujos se siguen necesitando funcionalidades para asistir explícitamente su creación y desarrollo.

Este es el punto de origen del proyecto que se va a llevar a cabo como ampliación al existente. Tiene como objetivo ayudar a los usuarios no expertos en la definición y aplicación de algoritmos de *Machine Learning* (ML) a través de una interfaz de usuario intuitiva que permitirá analizar y comprender tanto los procesos como los resultados obtenidos de cada *pipeline* mediante un sistema personalizado de recomendaciones, educando así al usuario en el correcto uso de modelos ML a partir de su comprensión del dominio de los datos.

Aunque en la actualidad ya existen algunas herramientas de ayuda y aprendizaje de ML que asisten a usuarios no expertos en la definición de *pipelines* de ML, es difícil adaptarlas a contextos específicos con necesidades y requerimientos particulares. Por estas razones, se opta por desarrollar una herramienta personalizada enfocada en la experiencia educativa de los profesionales sanitarios.

Gracias a su implementación, los usuarios pueden entrenar y crear modelos de aprendizaje automático sin necesidad de tener conocimientos previos de programación.

A lo largo del presente documento se mostrará el proceso de desarrollo de esta nueva funcionalidad, así como el proceso de rediseño y mejora de la aplicación ya existente. Para su elaboración se seguirá la guía de la plataforma Diaweb⁸ y se detallarán:

- Objetivos del proyecto: se explicarán cuáles son los objetivos que se persiguen con la realización del proyecto
- Técnicas y herramientas: se presentarán las técnicas metodológicas y las herramientas de trabajo que han sido utilizadas para llevar a cabo el proyecto
- Aspectos relevantes: se recogerán los aspectos más interesantes surgidos a lo largo del desarrollo del sistema
- Conclusiones y líneas de trabajo futuro: se presentarán las conclusiones del proyecto y se indicará como se puede continuar trabajando en la línea del sistema realizado

Los contenidos de esta memoria se complementan con:

- Anexo I - Plan del proyecto software: se recoge la planificación temporal del proyecto observando su viabilidad
- Anexo II - Especificación de requisitos software: detalla todos los requisitos software que plantea el sistema
- Anexo III - Proceso de diseño centrado en el usuario: recogerá las necesidades, limitaciones, comportamientos y características de los usuarios
- Anexo IV - Manual del programador: presentará los aspectos relativos al código fuente de la aplicación

- Anexo V - Manual del usuario: recoge un tutorial de usuario necesario para el correcto manejo de la aplicación

2. Objetivos del proyecto

En este apartado se explicarán los objetivos que se persiguen con la realización del proyecto. Se distinguirán entre los objetivos marcados por los requisitos del software y los objetivos personales propios.

Objetivos del sistema

El objetivo principal del sistema a desarrollar es la definición y el desarrollo de un sistema de gestión y configuración avanzada de *pipelines* de *Machine Learning* (ML) dentro del contexto médico que permita facilitar la selección, configuración y entrenamiento de modelos de aprendizaje a partir de heurísticas.

Dicho objetivo se completará mediante la consecución de los siguientes subobjetivos:

- Dotar a la aplicación de un carácter educativo mediante la creación de un sistema de información teórico que proporcione información formativa sobre cada opción del área de trabajo.
- Garantizar al usuario un correcto uso de la aplicación mediante la creación de un sistema de recomendaciones práctico que sea capaz de guiarle en las acciones que debe de ir realizando.
- Añadir a la aplicación un comportamiento funcional dinámico mediante la modificación de heurísticas.
- Mejorar la usabilidad y rendimiento de la aplicación mediante la implementación de nuevas funcionalidades y el rediseño de la interfaz hacia uno más intuitivo y familiar que seguirá el proceso de diseño centrado en el usuario.

Objetivos personales

El objetivo personal principal que se busca con este proyecto es la superación exitosa de la realización de un proyecto de gran tamaño sobre el que se puede plasmar cualquier tipo de idea sin limitación de ningún tipo de enunciado. Supone un reto personal que, junto con el afán de superación, implica la obtención de una mayor confianza en uno mismo.

Se busca marcar huella en el desarrollo del proyecto y una futura implantación real, no solo en el departamento de cardiología, si no en todos aquellos en los que sea necesario y resulte útil. Supone una satisfacción el poder ayudar en un contexto tan importante como es el de la salud y la medicina.

Así mismo, se espera obtener una familiarización con lenguajes y herramientas no utilizados anteriormente y que serán muy útiles de cara al mercado laboral. Se tiene también como objetivo el desarrollo de competencias orales a la hora de presentar un proyecto.

De igual modo, se intenta conseguir una mejora de la capacidad de adaptación realizando un proyecto totalmente diferente a los realizados hasta ahora.

3. Técnicas y herramientas

En este apartado se explicarán las herramientas de desarrollo, las técnicas metodológicas, los *frameworks* y las librerías que han sido utilizados a lo largo de todo el proyecto.

Diseño de la solución

Se detallarán las herramientas utilizadas para la elaboración de los prototipos digitales y de los diagramas.

- Adobe XD

Se utiliza la aplicación Adobe XD para la realización de todas las versiones de los prototipos digitales.

Fue elegida debido a que, gracias a un uso previo anterior, se tenía dominio sobre la herramienta. Además, resulta muy sencillo tanto crear como conectar las pantallas que forman la aplicación.

Además, ofrece la posibilidad de compartir el prototipo mediante la generación de un enlace a través del cual, quien realice la prueba de usuario, podrá escribir comentarios fácilmente.

- Paletas de color

Para elegir los colores que debía tener la nueva versión de la aplicación se utilizaron distintas herramientas:

- Colors: se utilizó para crear paletas de colores
- Paletton: se utilizó para obtener distintas tonalidades cromáticas de los colores. Gracias a esta herramienta también se puede ver cómo afectan los colores elegidos a los problemas de visión
- Snook.ca: se utilizó para validar el uso de los colores encima de los colores de los fondos
- PineTools: se utilizó para poder extraer colores de imágenes

- Diagrams.net – Draw.io

Para realizar los diagramas de casos de uso y actores se ha utilizado el software de dibujo de gráficos proporcionado por Diagrams.net.

Se eligió debido a que, gracias a un uso anterior, se estaba familiarizado con la herramienta. Además, está conectada a Google Drive, lo que facilita el almacenamiento de todos los diagramas.

Código de programación

- Python

El lenguaje de programación utilizado para el desarrollo de la aplicación ha sido Python debido a que la aplicación base existente estaba desarrollada en dicho lenguaje. Dispone de numerosas librerías de ciencias de datos y matemáticas, muy útiles a la hora de implementar ML.

Sin duda, se considera una gran elección puesto que la sintaxis es muy sencilla, tanto de aprender como de utilizar. Además, se trata de un lenguaje de código abierto multiplataforma.

- Django

El *framework* web utilizado para el desarrollo de la aplicación ha sido Django debido a que la aplicación base existente utilizaba dicho *framework*.

Se trata de una buena elección debido a que es un *framework* de alto nivel que permite un desarrollo rápido de sitios web seguros y con gran mantenibilidad. Además, ofrece muchas funcionalidades básicas por lo que no será necesario implementarlas.

- HTML, CSS y JavaScript

Los lenguajes utilizados para llevar a cabo el *front-end* de la aplicación han sido HTML, CSS y JavaScript.

Se utiliza HTML para estructurar la página web y sus contenidos, dándole estilo a cada uno de ellos mediante CSS. Se separa por tanto la presentación del contenido. Respecto a JavaScript, se utiliza para añadir a la aplicación características interactivas, tales como el cambio de apariencia en botones o elementos.

- Bootstrap

Se utiliza la biblioteca Bootstrap para diseñar de manera adaptativa (responsiva) la aplicación web. Utiliza para ello estructuras predefinidas y componentes reusables como tablas, filas y columnas. Sirve de base para crear tu propio diseño web a partir de elementos básicos.

- FlowChart

Se hace uso de la biblioteca FlowChart.js para dibujar los diagramas de flujo resultantes de la representación textual de las heurísticas, permitiendo así su mejor visualización.

Se eligió esta biblioteca debido a su correcta ejecución en el navegador a través de un lenguaje muy sencillo parecido al pseudocódigo.

Además, se destaca su forma de definición independiente de nodos y conexiones que permite la reutilización de elementos y conexiones y un rápido cambio en cualquiera de ellos. Como añadido, permite personalizar el diseño de los diagramas generados.

- FontAwesome

Se utiliza la biblioteca FontAwesome para la inserción y diseño de todos los iconos de la aplicación web.

Se eligió esta biblioteca debido a su gran cantidad de iconos disponibles, todos ellos personalizables mediante CSS. La selección se hacía de manera muy sencilla e intuitiva y su inserción se hacía fácilmente con tan solo una línea de código. Al ser de código abierto, se tiene libertad de uso sobre cualquiera de sus diseños.

Entorno de desarrollo

- PyCharm

Se utiliza como entorno de desarrollo PyCharm, IDE especializado en el lenguaje de programación Python.

Se elige debido a sus características de finalización de código inteligente, inspecciones de código, errores automáticos con soluciones sugeridas y funcionalidades de navegación muy completas. Se ha usado su versión profesional para estudiantes.

- GitLab

Para gestionar las distintas versiones del código, administrar, crear y conectar los repositorios con la aplicación desarrollada se ha utilizado GitLab, servicio colaborativo basado en Git. En particular, se ha utilizado una instalación del Grupo de Investigación en el que se ha elaborado el TFG.

- Postgres

Se utiliza Postgres como sistema gestor de la base de datos. Es de código libre y se considera en la actualidad uno de los más robustos y potentes.

Se escogió esta base de datos estructurada debido a su fácil instalación que permite un sistema multiplataforma. Además, es capaz de ajustarse al número de CPU y a la cantidad de memoria disponible por lo que se podrán recoger múltiples peticiones de manera simultánea. Además, cuenta con una útil herramienta gráfica.

- Slack

Como herramienta de seguimiento se ha utilizado Slack. La comunicación con las tutoras se ha desarrollado a través de esta herramienta, de tal forma que se ha realizado un seguimiento asíncrono combinado con sesiones síncronas y reuniones presenciales que han permitido trabajar en un contexto real.

- Google Drive

Para gestionar las distintas versiones de los informes y documentos generados se ha utilizado una carpeta compartida en Google Drive a la que tenían acceso todas las personas implicadas en el desarrollo de la aplicación.

4. Aspectos relevantes del desarrollo

En este apartado se detallarán los aspectos más relevantes e interesantes surgidos a lo largo de las distintas fases de desarrollo del proyecto.

Metodología

Para el desarrollo del proyecto se ha seguido el modelo de desarrollo del Proceso Unificado⁹ con un enfoque ágil ya que se han ido tomando las decisiones a medida que se avanzaba teniendo reuniones periódicas de seguimiento. Se considera la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Se decidió utilizar este modelo debido a su capacidad de adaptación a la mayoría de sistemas software y a las necesidades de cada organización.¹⁰

Algunas de las características que presenta son:

- Está basado en componentes
- Es un proceso conducido por casos de uso
- Está centrado en la arquitectura
- Es incremental e iterativo
- Utiliza UML¹¹ como lenguaje de representación visual
- Está preparado para desarrollar grandes proyectos

El Proceso Unificado se repite a lo largo de ciclos de desarrollo denominados iteraciones. Constituyen la vida de un sistema y concluyen con una versión entregable del producto.

Cada ciclo consta de 4 fases:

- Inicio: se define el alcance que tendrá el proyecto y se desarrollan los casos de negocio.
- Elaboración: se lleva a cabo la planificación, la especificación detallada y la arquitectura del sistema.
- Construcción: se elabora lo acordado en las fases anteriores.
- Transición: se obtiene una versión apta para su publicación, se corrigen fallos y se incorporan las mejoras observadas.

Las fases de inicio y elaboración pertenecen a la etapa de Ingeniería, mientras que las fases de construcción y transición a la etapa de producción. La *Figura 1* muestra el ciclo de dichas fases.



Figura 1: Ciclo de desarrollo

Cada una de las fases anteriores debe terminar en hito. Un hito es un punto de control en el que los participantes en el proyecto revisan el progreso. Así mismo, dentro de las fases se encuentran las siguientes disciplinas que agrupan tareas similares:

- Modelado de negocio
- Requisitos
- Análisis
- Diseño
- Implementación
- Pruebas

Para llevar a cabo todo lo comentado, se ha utilizado el esquema mostrado en la *Figura 2* como referencia:

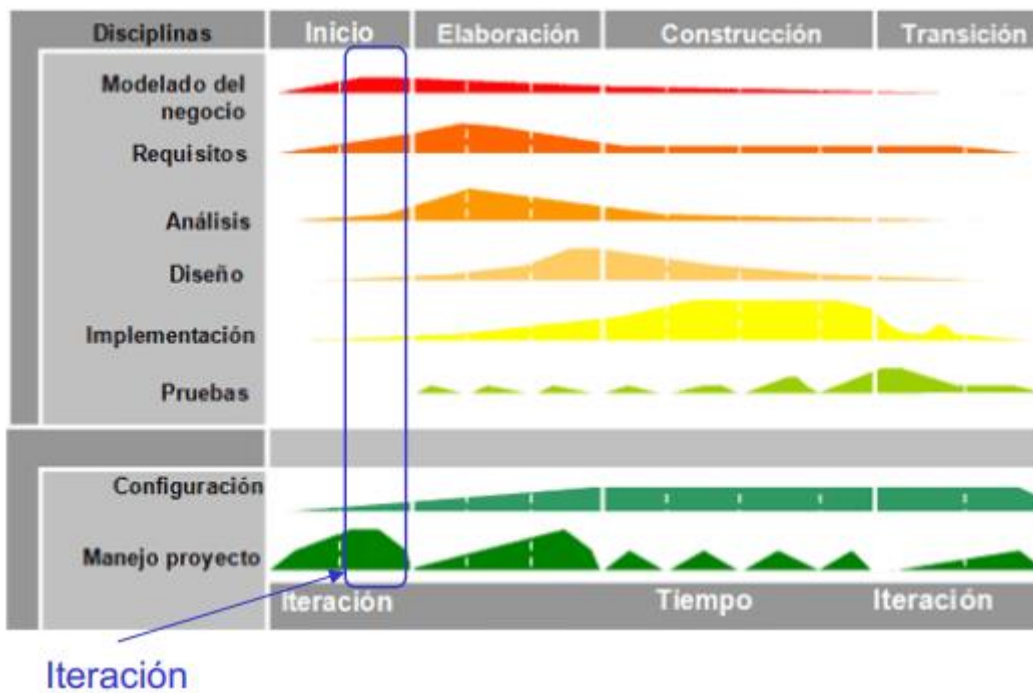


Figura 2: Fases del Proceso Unificado

Planificación temporal

En este apartado se detalla la distribución seguida en base al Proceso Unificado, que se repite a lo largo de ciclos de desarrollo buscando que el tiempo de desarrollo sea mínimo.

Se presenta, en la *Figura 3*, la planificación temporal seguida:

		<i>Semanas</i>
Fase 1 - Inicio		
1.1	Preparación de los lenguajes y herramientas	2
1.2	Familiarización con el proyecto base	1
1.3	Planificación temporal	1
1.4	Especificación de requisitos del Software	1
1.4.1	Análisis de los participantes del sistema	
1.4.2	Definición de los objetivos del proyecto	
1.4.3	Especificación de requisitos del sistema	
1.5	Proceso de diseño centrado en el usuario	2
1.5.1	Definición de audiencia	
1.5.2	Escenarios de uso	
1.5.3	Análisis de tareas	
1.5.4	Site Map	
1.5.5	Journey Map	
1.5.6	Prueba de usuario del proyecto base	
1.5.7	Primer boceto de diseño	
Fase 2 - Elaboración		
2.1	Inicio de los primeros anexos	1
2.2	Refinamiento de requisitos y actores	1
2.3	Diagramas de casos de uso	1
2.4	Elaboración Prototipo Digital	3
2.4.1	Primera versión	
2.4.2	Evaluación primera versión	
2.4.3	Segunda versión	
2.4.4	Evaluación segunda versión	
2.4.5	Refinamiento	
Fase 3 - Construcción		
3.1	Continuación de los anexos	1
3.2	Implementación de código	4
3.2.1	Nuevo diseño de la aplicación	
3.2.2	Funcionalidades extras al proyecto base	
3.2.3	Recomendaciones teóricas	
3.2.4	Recomendaciones prácticas	
3.3	Pruebas	1
3.3.1	Nuevo diseño de la aplicación	
3.3.2	Funcionalidades extras al proyecto base	
3.3.3	Recomendaciones teóricas	
3.3.4	Recomendaciones prácticas	
Fase 4 - Transición		
4.1	Revisión final requisitos	1
4.2	Revisión final diagramas	1
4.3	Revisión final funcionalidad	1
4.4	Prueba final de la aplicación	1
4.5	Mejoras aplicación	1
4.6	Elaboración de la documentación final	3
4.7	Últimas revisiones y entrega	1

Figura 3: Planificación temporal

Las fechas estimadas para cada fase siguiendo dicha planificación son:

Fase	Comienzo	Fin
Inicio	14 de febrero de 2022	20 de marzo de 2022
Elaboración	14 de marzo de 2022	17 de abril de 2022
Construcción	11 de abril de 2022	22 de mayo de 2022
Transición	23 de mayo de 2022	4 de julio de 2022

Tabla 1: Fechas estimadas según planificación temporal

Como aspecto destacable comentar el hecho de trabajar sobre un proyecto base ya existente. Al tiempo empleado en realizar cada una de las tareas de la nueva implementación hay que sumarle el usado en entender lo que ya está hecho. Como se puede observar en la planificación, se dedicó una semana solo a la familiarización con el proyecto anterior.

Para una información más detallada, como el diagrama de Gantt, se puede consultar el *Anexo I: plan de Proyecto software* adjuntado junto con esta memoria.

Especificación de requisitos

Para la especificación de requisitos se va a seguir el método Durán y Bernárdez¹² siendo los requisitos detallados mediante métodos gráficos, formales o textuales, según sea oportuno.

Se decidió usar esta metodología ya que fue utilizada en ocasiones anteriores en alguno de los trabajos realizados durante el grado.

En primer lugar, se define a los *participantes del proyecto*. Se incluye en este apartado a la desarrollada, a los tutores académicos, a la desarrolladora anterior y a todos los miembros colaboradores perteneciente al departamento de cardiología del Hospital de Salamanca.

Nombre	Esther Andrés Fraile
Organización	Universidad de Salamanca
Rol	Autor
Tipo de usuario	Desarrollador
Comentarios	No

Tabla 2: Ejemplo participante del proyecto

Para consultar el resto de tablas de participantes consultar el *Anexo II: especificación de requisitos software*.

En segundo lugar, se *describe el sistema actual*. Como ya se ha explicado anteriormente, se trabaja sobre una aplicación existente por lo que se ha dado mucha importancia a este apartado. Es necesario detallar cómo es la aplicación existente y que pasos se van a llevar a cabo para mejorarla.

La herramienta existente *Koopa ML* es una aplicación enmarcada dentro del contexto del departamento de cardiología del Hospital Universitario de Salamanca que trata de acercar a usuarios sin conocimientos de programación al aprendizaje automático mediante la definición y aplicación de *pipelines ML*.¹³

Permite la creación, modificación, unión y eliminación de nodos permitiendo ejecutar el flujo construido y analizar los resultados obtenidos.

Sin embargo, aunque nuestra herramienta elimina la complejidad a la hora de trabajar con flujos de ML se siguen necesitando funcionalidades para asistir explícitamente el desarrollo de estos flujos.

Es por ello por lo que en el proyecto del presente informe se intentará suplir esa carencia a partir de:

- Creación de un sistema de sugerencias o recomendaciones personalizado: el usuario estará guiado en todo momento por la propia aplicación. Irá recibiendo información de qué hacer y de cómo debe seguir para obtener un flujo exitoso. De igual modo, podrá desactivar y activar las sugerencias en cualquier momento.
- Incorporación de información teórica: aquellas partes de la aplicación relacionadas con ML y sus algoritmos irán acompañados de un desplegable que se podrá visualizar siempre que se desee y que devolverá una explicación detallada sobre aquello que se consulta.
- Modificación de heurísticas: se añadirá la funcionalidad de modificar las heurísticas a través de una plataforma gráfica basada en reglas. De esta manera, las recomendaciones serán dinámicas y podrán ser modificadas cuando se desee.

De igual modo, aunque no suponía un problema para la funcionalidad de la aplicación, la desarrolladora actual tomó la decisión de cambiar el diseño integral de la aplicación hacia uno mucho más vistoso e intuitivo con el objetivo de conseguir una mayor usabilidad y eficiencia.

También se han añadido pequeñas funcionalidades en cada pantalla de la aplicación que hacen a la aplicación mucho más completa y robusta. Serán explicadas con detalle en próximos apartados.

En el siguiente apartado a comentar, se definen los *actores del sistema*.

Se utiliza una relación de generalización para indicar que el Administrador desempeña el mismo papel que el Usuario, pero con funcionalidades extras. La *Figura 4* muestra el diagrama de actores.

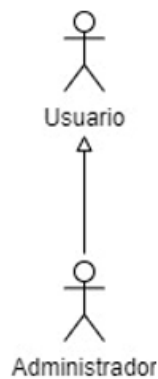


Figura 4: Diagrama de actores

Para consultar la definición detallada de cada uno consultar el *Anexo II: especificación de requisitos software*.

A continuación, se expone el *catálogo de requisitos del sistema*: los requisitos funcionales, no funcionales y de información. Cada uno de ellos se detalla en el *Anexo II: especificación de requisitos software* y se sumará a los ya existentes en la primera versión de la aplicación.

Los requisitos de información añadidos son:

- Información educativa
- Recomendaciones
- Diagrama de flujo para la gestión de heurísticas

Los requisitos funcionales añadidos son:

- Descripción teórica de las opciones
- Creación de sugerencias de uso personalizadas
- Modificación de heurísticas

Son añadidos, por tanto, los paquetes de gestión de recomendaciones y de gestión de heurísticas. Se pueden observar en la *Figura 5* y *Figura 6*.

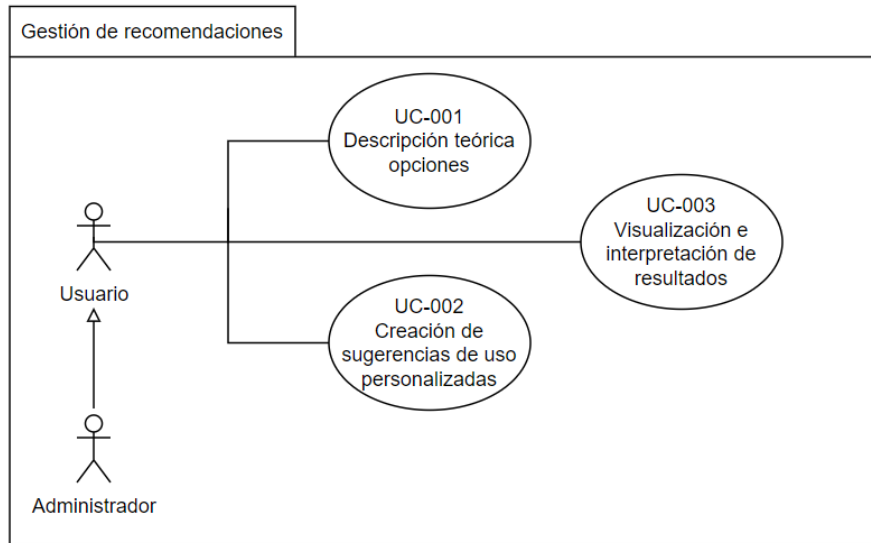


Figura 5: Diagrama gestión de recomendaciones

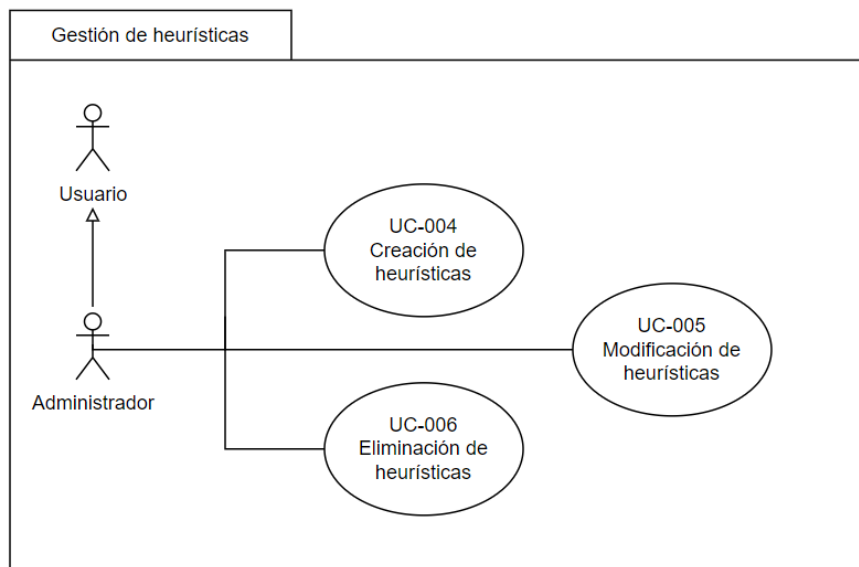


Figura 6: Diagrama gestión de heurísticas

Para la clasificación de los requisitos no funcionales se utilizó el modelo Sommerville¹⁴. Fue elegido debido a su previo uso en trabajos anteriores realizados en el grado.

Siguiendo dicha clasificación que divide los requisitos no funcionales en requerimientos de producto, requerimientos organizacionales y requerimientos externos se tendrán en cuenta los relacionados con:

- Incorporación de la nueva funcionalidad sin influir en la ya implementada en *Koopa ML*
- Usabilidad: el sistema será fácil y cómodo de utilizar para los usuarios
- Eficiencia: el sistema responderá rápidamente a las peticiones de los usuarios

- Disponibilidad: los usuarios tendrán en todo momento acceso a la aplicación
- Confiabilidad: la probabilidad de que el sistema complete un periodo de tiempo sin fallar será muy elevada
- Confidencialidad e integridad: el acceso a la información comprometida estará limitado y protegido
- Fiabilidad: se garantizará un funcionamiento adecuado en todo momento
- Mantenibilidad: la capacidad de realizar modificaciones sin afectar a la continuidad del servicio será alta
- Entorno y desarrollo; se tendrá en cuenta el mejor entorno y desarrollo para la aplicación

Proceso de diseño

En este apartado, se describe todo el proceso de creación del diseño de la aplicación. Se realizará comenzando por la creación de prototipos digitales que tendrán un proceso de evaluación con usuarios reales. Después de dichas pruebas, se elaborarán nuevas versiones de los prototipos. Cuando se obtenga una versión válida se procederá a su implementación final, pudiendo incorporarse nuevas sugerencias o mejoras.

En el proceso de creación, se incluirán pautas para cumplir con los objetivos de un intercambio de información eficiente: minimizar errores, incrementar la satisfacción, ampliar la facilidad de uso, etc. A mayores, se tendrán en cuenta elementos como una jerarquía visual clara, información de la ubicación del usuario en todo momento y diseño coherente en toda la aplicación.

Para la creación de las distintas interfaces se siguió el proceso de diseño centrado en el usuario¹⁵. Se eligió dicha filosofía de diseño debido a que gracias a ella se puede comprender cuales son las necesidades de aquellos a los que va dirigida la aplicación. Gracias a ello, se espera que los usuarios tengan una mejor experiencia y una mayor satisfacción con respecto a la aplicación.

Se ve necesario destacar que en el proyecto actual se decide hacer un rediseño de todas las pantallas de la aplicación ya existente ya que muchos de los principios del proceso de diseño no se cumplían como puede ser el tener un diseño coherente en toda la aplicación o la correcta información sobre la ubicación del usuario en todo momento.

El esquema seguido para todo este rediseño va a seguir las fases contenidas en el *ciclo de diseño centrado en el usuario*, que son:

1. Análisis de los competidores
2. Definición de la audiencia
3. Escenarios de uso
4. Contenido de la encuesta
5. Flujos de proceso
6. Mapa de sitio
7. Esquema de página
8. Diseño

9. Prototipo
10. Pruebas de usuario
11. Revisión
12. Aprobación

Se pueden observar en la *Figura 8*.

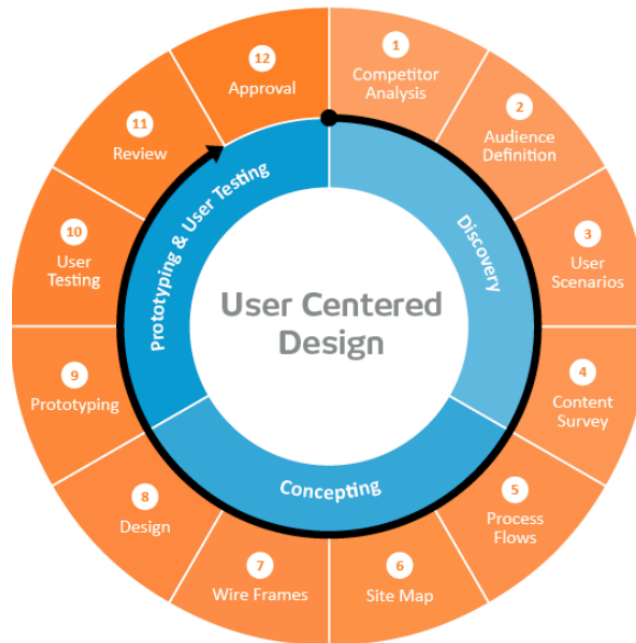


Figura 7: Ciclo de diseño

Todas las decisiones tomadas con respecto a este apartado y todos los cambios explícitos sobre la aplicación anterior se podrán consultar en el *Anexo III: proceso de diseño centrado en el usuario*.

A continuación, se exponen los aspectos más relevantes de cada una de las fases realizadas.

Fase de exploración

Esta fase constituye la primera del ciclo de diseño. Durante ella, se realiza un trabajo de investigación para conocer las necesidades e intereses de los usuarios futuros de la aplicación.

Comprende las tareas de:

- Búsqueda de necesidades
- Definición de la audiencia
- Escenarios de uso

Búsqueda de necesidades

En este apartado se ha encontrado la dificultad de no poder llevar a cabo un estudio exhaustivo de las necesidades de los usuarios puesto que al estar la aplicación ya implementada esta fase fue desarrollada con anterioridad. Tampoco se realizarán encuestas de búsqueda de estas.

De hecho, el presente proyecto surge de esas necesidades detectadas que aún no habían sido resueltas. Se citan a continuación dichas necesidades para los dos grupos de usuario:

- Grupo principal:
 - o Obtención de una herramienta en la que aplicar Inteligencia Artificial y algoritmos *Machine Learning* en un conjunto de datos médicos sin necesidad de tener experiencia técnica digital ni conocimientos de programación ML.
 - o Visualización y análisis de datos médicos.
 - o Aprendizaje sobre el análisis de datos, su utilidad y beneficios en medicina.
 - o Aprendizaje práctico sobre algoritmos ML.
 - o Capacidad de uso de una aplicación ML que permita una visualización detallada de datos y ofrezca ayuda para su interpretación.
- Grupo secundario:
 - o Facilidad de integración de algoritmos en los *pipelines* de ML.
 - o Personalización sencilla de la herramienta.
 - o Modificación de parámetros y heurísticas.
 - o Aprendizaje a partir de los resultados de las pruebas.

Definición de la audiencia

En este apartado se analiza a la audiencia que será cliente de la aplicación identificando las características de todos los tipos de usuarios.

Se resalta que se considerará un sistema usable cuando es utilizado por un grupo concreto y bien definido de personas que realizan tareas concretas en un contexto concreto.

Aun estando la audiencia ya definida en la versión anterior del sistema, se revisó en profundidad para ver si las nuevas funcionalidades afectaban a alguno de los tipos de usuario.

Se definen, de manera resumida, dos tipos de usuario junto con su participación en las nuevas funcionalidades:

- *Tipo de usuario principal: Usuario no experto en IA/ML*
 - o **Médicos/as sin conocimientos en programación** y ML pero que conocen a la perfección el dominio de los datos. No han usado la herramienta previamente (ni ninguna similar).
 - Tienen interés en aprender para poder utilizar la aplicación
 - No saben definir flujos ML ni las propiedades de los algoritmos
 - Se supone un rango de edad amplio por lo que se tiene en cuenta la posible dificultad de aprendizaje

- Departamento de cardiología del hospital de Salamanca, aunque con proyección futura de ampliación
 - Podrán crear proyectos ML y analizar sus resultados
 - Usarán la aplicación con el **modo recomendación** por defecto
- *Tipo de usuario secundario: Usuario experto en IA/ML*
- **Administrador:** para la gestión de usuarios
 - **Médicos/as con conocimientos** de programación o previo uso de la herramienta
 - Se han familiarizado con la herramienta (previamente pertenecieron al grupo principal)
 - Son expertos en el dominio de los datos
 - Tendrán acceso a las mismas funciones que los usuarios principales
 - No pueden hacer mejoras o modificaciones de heurísticas
 - Elegirán si usar o no el modo **recomendación**, pudiendo activarlo y desactivarlo cuantas veces quieran
 - **Desarrolladores de IA**
 - No son expertos en el dominio de los datos
 - Pueden hacer mejoras o modificaciones de heurísticas (interfaz gráfica privada) así como crear proyectos
 - Su objetivo es facilitar la integración de algoritmos en los *pipelines* de ML mediante un proceso guiado durante la definición, selección, aplicación y entrenamiento de algoritmos y lograr la interpretación por parte de los usuarios no expertos.
 - Usarán la aplicación con el **modo recomendación** desactivado.

Para llevar a cabo una mejor caracterización de los arquetipos de los usuarios se ha utilizado el *método Lene Nielsen*¹⁶, consistente en 10 pasos:

1. Recopilación de datos
2. Formulación de una hipótesis
3. Aceptación de la hipótesis
4. Fijación de un número de personas
5. Descripción de personas
6. Creación de situaciones para los personajes
7. Obtención de aceptación de la organización
8. Difusión del conocimiento
9. Preparación de escenarios
10. Realización de ajustes continuos

Para ver con detalle todos los pasos del método Lene Nielsen consultar el *Anexo III: proceso de diseño centrado en el usuario*.

Los usuarios modelo resultantes de esta fase fueron:

Identificación (nombre y edad)	Alfredo, 60 años
Grupo al que pertenece	Primario
Ocupación	Médico cardiólogo veterano del Hospital de Salamanca
Emociones y actitudes	Es una persona extrovertida. Pese a su edad tiene gran energía y ganas de aprender, aunque su conocimiento en tecnología es nulo. Le gustaría automatizar las tareas que tanto tiempo lleva haciendo.
Rasgos personales	Es organizado, solitario y no le gusta tener una rutina. Evita las situaciones que le lleven a estrés o discusión.

Tabla 3: Definición de la audiencia, usuario 1

Identificación (nombre y edad)	Antonia, 48 años
Grupo al que pertenece	Primario
Ocupación	Médica de familia
Emociones y actitudes	Se trata de una persona metódica con grandes habilidades para aprender cosas nuevas. Le encantan los retos y poder aprender informática es uno de ellos. Adora su trabajo, pero no tiene nada de tiempo libre.
Rasgos personales	Le gusta trabajar en equipo y optimizar las tareas para que se realicen en el menor tiempo posible.

Tabla 4: Definición de la audiencia, usuario 2

Identificación (nombre y edad)	Anais, 30 años
Grupo al que pertenece	Primario
Ocupación	Médica interna residente
Emociones y actitudes	Es sensible, introvertida y ordenada. Le gusta hacer su trabajo de la mejor manera posible y ayudar a los demás en todo lo que pueda.
Rasgos personales	Lleva poco tiempo trabajando, pero se adapta bien. Maneja gran cantidad de datos y tiene muchas ganas de aprender. En la carrera estudió algo de informática, aunque no se considera experta.

Tabla 5: Definición de la audiencia, usuario 3

Identificación (nombre y edad)	Jorge, 24 años
Grupo al que pertenece	Secundario
Ocupación	Desarrollador IA
Emociones y actitudes	Es curioso e inquieto. Tiene facilidad aprendiendo cosas nuevas y le encanta desarrollar aplicaciones que hagan la vida más fácil al resto de personas.
Rasgos personales	No le gusta el desorden ni que le lleven la contraria. Tiene sus objetivos claros y lucha por conseguirlos.

Tabla 6: Definición de la audiencia, usuario 4

Identificación (nombre y edad)	Laura, 39 años
Grupo al que pertenece	Secundario
Ocupación	Desarrolladora IA y experta en ML
Emociones y actitudes	Es curiosa, extrovertida y sensible. Le gusta observar lo que la rodea y poder replicar comportamientos humanos en máquinas de inteligencia artificial. Domina el ML.
Rasgos personales	Es una chica alegre, aunque un poco insegura. Tiene mucha memoria y está acostumbrada a retener gran cantidad de datos en su trabajo.

Tabla 7: Definición de la audiencia, usuario 5

Identificación (nombre y edad)	Pablo, 45 años
Grupo al que pertenece	Secundario
Ocupación	Médico cardiólogo
Emociones y actitudes	Le interesa la informática y estaría encantado de poder compaginarla con su trabajo como cardiólogo. Lleva meses utilizando la herramienta propuesta y aunque él consiguió aprender a manejarla considera que necesita mejorar y ser más intuitiva.
Rasgos personales	Tiene un gran sentido del humor. Lo utiliza como herramienta para hacer sentir a los demás. El bienestar de las personas es lo que más le importa.

Tabla 8: Definición de la audiencia, usuario 6

Escenarios de uso

Una vez se ha realizado la definición de la audiencia se crean los escenarios de uso¹⁷. Se trata de historias ficticias cuyo objetivo es centrarse en la experiencia que varios usuarios pueden llegar a tener con el sistema que se quiere desarrollar.

Para realizar este apartado se tuvo en cuenta:

- Representación de los personajes. La estructura seguida será:
 - o Quién es el usuario
 - o Situación que impulsa al usuario a querer utilizar el sistema
 - o Tarea u objetivo específico que tiene el usuario
 - o Información sobre los ingresos y gastos del usuario
 - o Cómo finalizará la tarea deseada gracias al uso del sistema
 - o Puntos de fricción en la vida diaria o en la experiencia del usuario
- Tipos de persona: se clasifican en grupos de usuario primario y secundario y se utilizan las personas descritas en el apartado de *Definición de la audiencia*.
- Tipo de escenario: fue elegido el método de *escenario narrativo de tareas*. Fue elegido ya que se trata de una historia completa de la interacción de la persona con el sistema y de una descripción del mundo del personaje tal y como existe ahora.

Para ver con detalle todos los escenarios definidos consultar el *Anexo III: proceso de diseño centrado en el usuario*.

Fase de conceptualización

En este apartado se tiene como objetivo final la creación de un prototipo digital que sirva como primera aproximación a la futura implementación real.

Comprende las tareas de:

- Análisis de tareas
- Mapa de sitio
- *Journey Map*
- Diseño

Análisis de tareas

En este apartado se analizará la manera en la que los futuros usuarios realizarán las distintas acciones de la aplicación. Se tendrá en cuenta lo que hace, sobre qué cosas actúa y qué necesita saber.

El modelo de análisis de tareas que se va a utilizar es el *jerárquico (HTA)*¹⁸ junto con la estrategia “divide y vencerás”, consistente en descomponer tareas complejas en otras más simples. Ha sido elegido debido a su previa utilización en trabajos del grado.

Para realizar los flujos de tareas se ha partido de los resultantes del proyecto base anterior siendo modificados para adaptarlos a la nueva versión del sistema.

Para ver con detalle todos los cambios realizados consultar el *Anexo III: proceso de diseño centrado en el usuario*. A modo resumen se tiene:

- Recomendaciones:
 - o Se añade el apartado correspondiente a las recomendaciones. Se incluye el esquema en la *Figura 8* debido a su importancia:

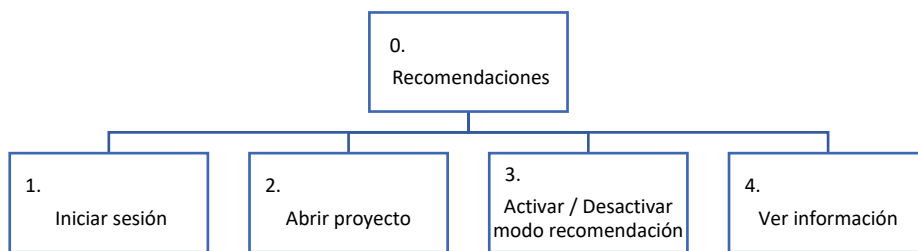


Figura 8: Tarea recomendaciones

- Información teórica:
 - o Se añade el apartado correspondiente a la información teórica de cada apartado. Se incluye el esquema en la *Figura 9* debido a su importancia:

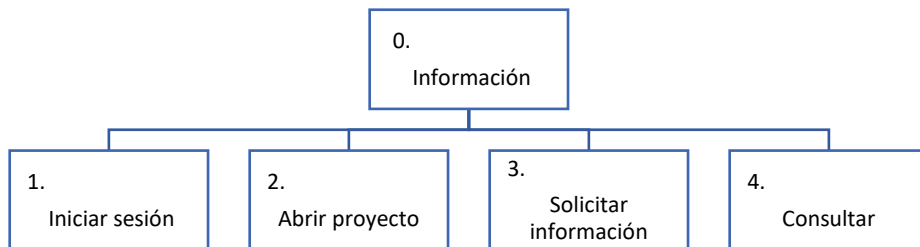


Figura 9: Tarea Información teórica

- Heurísticas:
 - o Se añade el apartado correspondiente a las heurísticas. Se incluye el esquema en la *Figura 10* debido a su importancia.

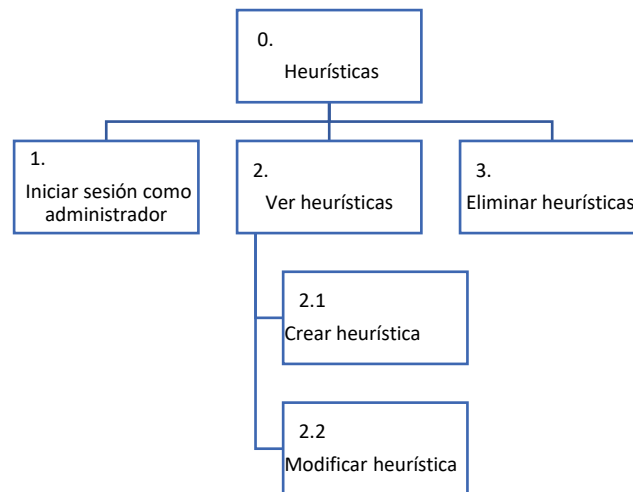


Figura 10: Tarea heurísticas

- Nuevo proyecto:
 - o Se añade una pantalla de configuración del proyecto previa a su creación
 - o Se eliminan los distintos modos de ejecución para su fusión en uno
 - o Se modifican las opciones disponibles para cada proyecto, añadiendo la de editar y duplicar
- Cuenta:
 - o Se añade la opción de modificar cuenta y contraseña de manera independiente
 - o Se añade la opción de eliminar cuenta
- Usuarios:
 - o Se unifican los apartados de creación y modificación en una misma pantalla
- Instituciones:
 - o Se unifican los apartados de creación y modificación en una misma pantalla

Mapa de sitio

En este apartado se utilizará un mapa de sitio que contendrá la información necesaria para indicar el camino correcto que se debe seguir para llevar a cabo una determinada tarea en el sistema planteado.

Ha sido modificado frente al del proyecto inicial mediante los nuevos flujos de tareas creados, previamente explicados. Se pueden consultar dichos cambios en la *Figura 11*, donde estarán coloreados.

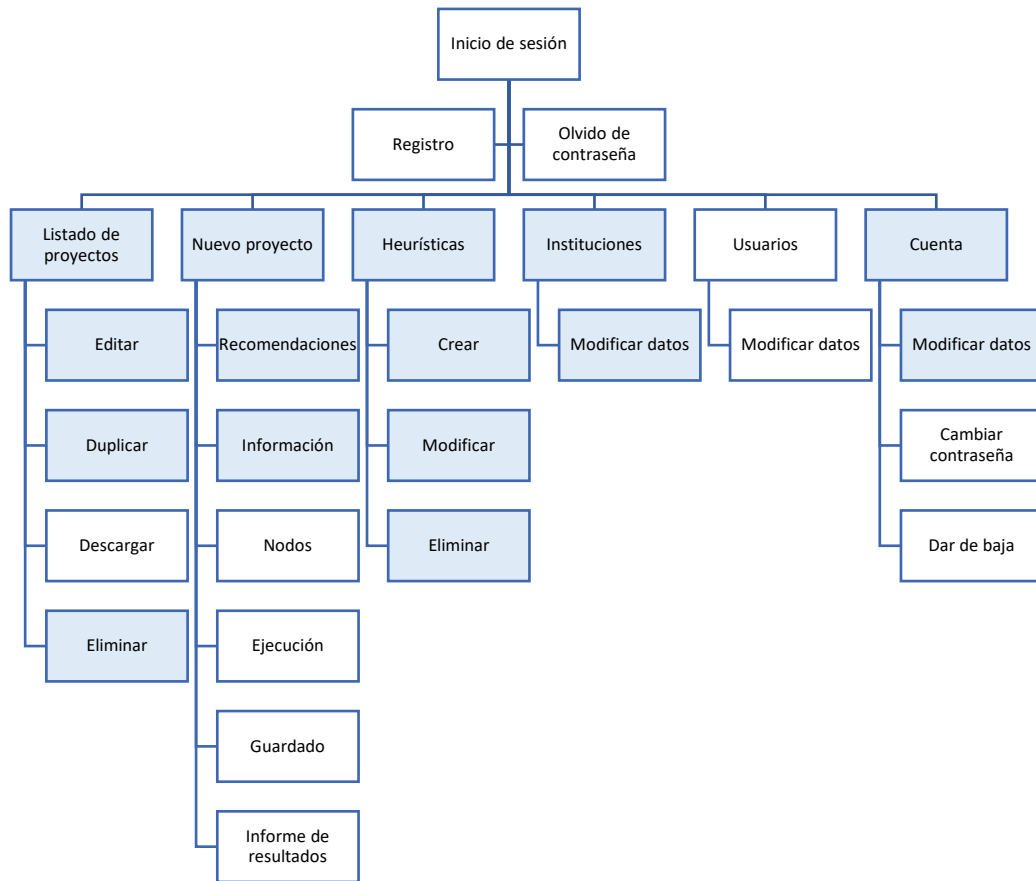


Figura 11: Mapa de sitio del sistema. Las secciones coloreadas representan las modificaciones y adiciones en el desarrollo de este proyecto.

Journey Map

En este apartado¹⁹, se comprenderán las necesidades de los usuarios con el fin de mejorar todos los procesos ya existentes y crear una experiencia de usuario más satisfactoria.

Se consideró importante realizar este apartado ya que a partir de él se identificaron²⁰ los elementos concretos del recorrido de la aplicación base existente que causaban satisfacción o problemas al usuario, teniendo en cuenta los sentimientos, dificultades y satisfacciones del usuario a lo largo de las distintas fases del sistema.

Para su realización, se utilizó la experiencia de la propia desarrolladora cuando utilizó la aplicación por primera vez.

Es el siguiente:






ETAPAS DE INTERACCIÓN DEL USUARIO	Inicio de sesión / Registro	Listado de proyectos	Modificación del perfil	Inserción de bloques y conexiones	Opciones de ejecución	Análisis de resultados
Objetivos del usuario	Acceder de manera sencilla a la página principal	Gestionar los proyectos y la cuenta	Consultar y cambiar los datos de registro	Crear un diagrama de bloques que permita el procesamiento de datos uniendo nodos	Ejecutar el bloque creado	Analizar los resultados obtenidos de la ejecución del modelo creado
Actividades	Inicio de sesión Registro Olvido de contraseña	Ver, crear, buscar, eliminar, descargar, ordenar proyectos y modificar datos	Cambiar nombre, apellidos, correo, contraseña e institución	Crear, modificar, unir y eliminar nodos	Ejecutar el flujo de nodos	Obtener resultados
Emociones						
Puntos críticos	Problemas en el registro como nombre de usuario ocupado Olvido de contraseña Validación	Algunos campos de ordenación son incorrectos, cuesta encontrar algunas opciones y faltan otras como poder dar nombre y descripción	Si accedes a modificar datos no puedes cambiar la contraseña desde ahí No puedes cambiar el nombre de usuario	No conocer el significado de cada bloque No comprender cómo debe realizarse la unión de nodos	No comprender el funcionamiento de las opciones de compilación No saber solucionar los posibles errores de los que se nos avise	Dificultad a la hora de interpretar los resultados
Posibilidades de mejora	Mensaje de "nombre en uso" Restricciones en la contraseña con mensajes informativos Explicación de alta del usuario	Poner más accesibles todas las opciones, modificar el diseño de la interfaz, añadir campos y opciones nuevos e incluir la modificación de los proyectos	Añadir la opción de cambiar la contraseña en cambio de datos y de cambiar el nombre de usuario	Añadir un sistema de recomendaciones que ofrezca ayudas teóricas y prácticas de cómo se debe actuar en cada caso	Añadir un sistema de recomendaciones que ofrezca ayuda para la ejecución y solución de errores	Añadir un sistema de recomendaciones que analice los resultados como herramienta de apoyo

Tabla 9: Journey Map

Diseño

Los aspectos técnicos del diseño como la tipografía elegida fueron determinados en la versión ya existente de la aplicación.

Sin embargo, se comentan algunos cambios importantes llevados a cabo:

- Se cambia el color de las distintas categorías de nodos. Este cambio se realizó ya que los colores previamente utilizados eran muy parecidos entre sí, con tonalidades azules todos.
En la nueva versión, se utilizarán unos mucho más diferenciados (rojo, amarillo, azul y púrpura) que, además, como se puede observar, siguen el degradado de un arcoíris por lo que implican un orden.
- En la versión base los colores de las categorías y subcategorías de nodos eran iguales. Se ha modificado el color de las subcategorías hacia uno de la misma gama de la categoría, pero en una tonalidad más suave.
- Las categorías de los nodos irán ahora acompañadas de un número que indicará el orden en el que debe ser utilizada cada una.

- Todas las pantallas que tienen una misma funcionalidad (por ejemplo, proyectos-instituciones-usuarios-heurísticas) han sido modificadas para que usen la misma plantilla de diseño.
- Todos los botones serán representados por el mismo formato y color tal y como marca el sexto principio de usabilidad de Nielsen, *reconocer mejor que recordar*.
- Los apartados comunes y recurrentes en otras páginas web (por ejemplo, inicio de sesión, registro, etc) han sido modificados para asemejarse a lo conocido y familiar usando de nuevo el sexto principio de usabilidad de Nielsen. Por ejemplo, la opción de “Entrar” se sustituyó por “Inicio de sesión” o la de “Solicitar alta de alta en el sistema” por “Registro”.

Fase de prototipos y pruebas de usuario

En este apartado se realizarán los distintos prototipos digitales a partir de los cuales se llegará al que será el posterior prototipo funcional. Para todo este proceso de creación, se llevarán a cabo pruebas de usuario.

Prototipo digital

Durante esta fase se elaborarán varias versiones de manera incremental de tal manera que la segunda será la primera con mejoras. Así mismo, se tendrá en cuenta la opinión de los usuarios interesados para realizar los cambios y mejoras entre ellos.

En concreto, se elaborarán dos, siendo un tercero la futura implementación.

Se considera importante destacar que desde este paso todos los posteriores que se realicen se desvinculan del proyecto base sobre el que se trabaja ya que, como se comentó anteriormente, el diseño se transforma por completo.

- **Primera versión**

Se realiza el primer rediseño de la aplicación y se incorporan todas las funcionalidades que podrían hacer a la aplicación más completa.

Se podrán observar de manera detallada todos los cambios realizados en el *Anexo III: proceso de diseño centrado en el usuario*. Se adjunta captura de pantalla de los más significativos para su futura comparación con la segunda versión del prototipo. Se encontrarán en *Figura 12 – Figura 23*.



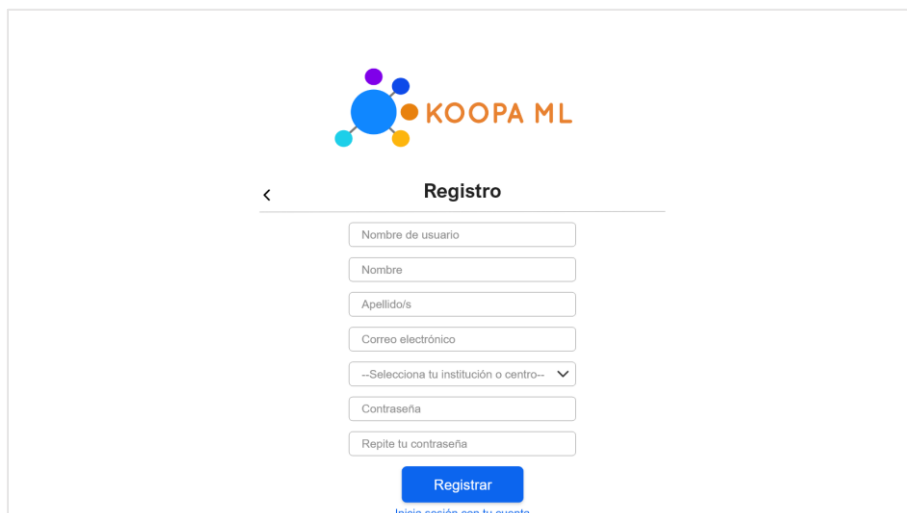
The login screen features the KOOPA ML logo at the top, which consists of a blue circle with four smaller colored circles (purple, blue, orange, yellow) around it. Below the logo is the title "Inicio de sesión". The form includes two input fields: "Nombre de usuario o correo electrónico" and "Contraseña" with a visibility toggle icon. A blue "Iniciar sesión" button is positioned below the fields. At the bottom, there are two links: "¿Has olvidado tu contraseña?" and "¿No tienes cuenta? [Regístrate](#)".

Figura 12: Primera versión prototipo digital: Inicio de sesión



The reset password screen features the KOOPA ML logo at the top. Below it is a back arrow and the title "Restablecer contraseña". The instruction "Introduce tu correo electrónico y se te enviarán las instrucciones para restablecer tu contraseña" is centered. There is a single input field for "Correo electrónico" and a blue "Enviar" button. At the bottom, there are two links: "Iniciar sesión" and "¿No tienes cuenta? [Regístrate](#)".

Figura 13: Primera versión prototipo digital: Restablecimiento de la contraseña



The registration screen features the KOOPA ML logo at the top. Below it is a back arrow and the title "Registro". The form includes seven input fields: "Nombre de usuario", "Nombre", "Apellido/s", "Correo electrónico", "--Selecciona tu institución o centro--" (a dropdown menu), "Contraseña", and "Repite tu contraseña". A blue "Registrar" button is positioned below the fields. At the bottom, there is a link: "Inicia sesión con tu cuenta".

Figura 14: Primera versión prototipo digital: Registro

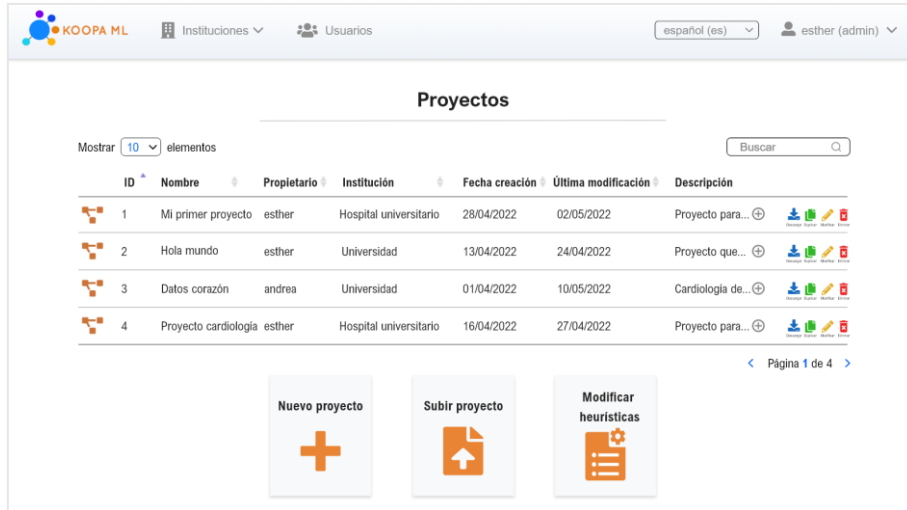


Figura 15: Primera versión prototipo digital: Listado de proyectos



Figura 16: Primera versión prototipo digital: Creación de proyecto

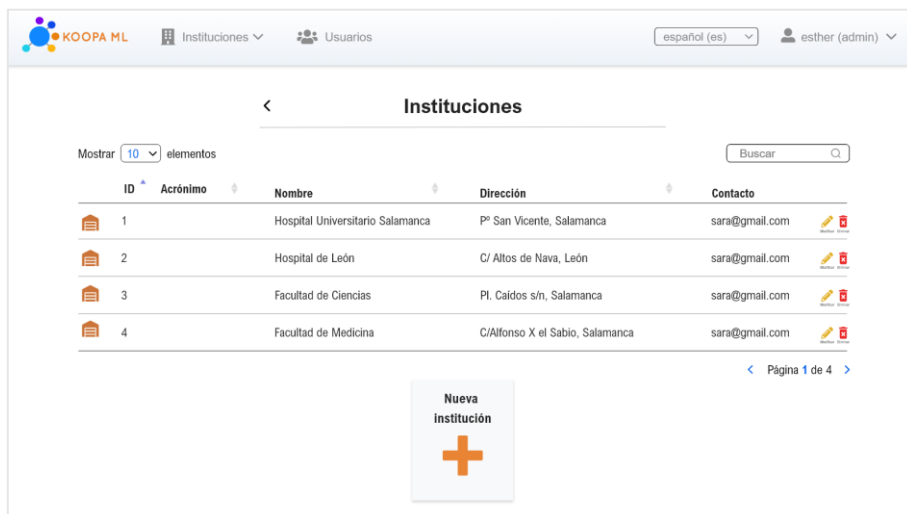


Figura 17: Primera versión prototipo digital: Listado de instituciones

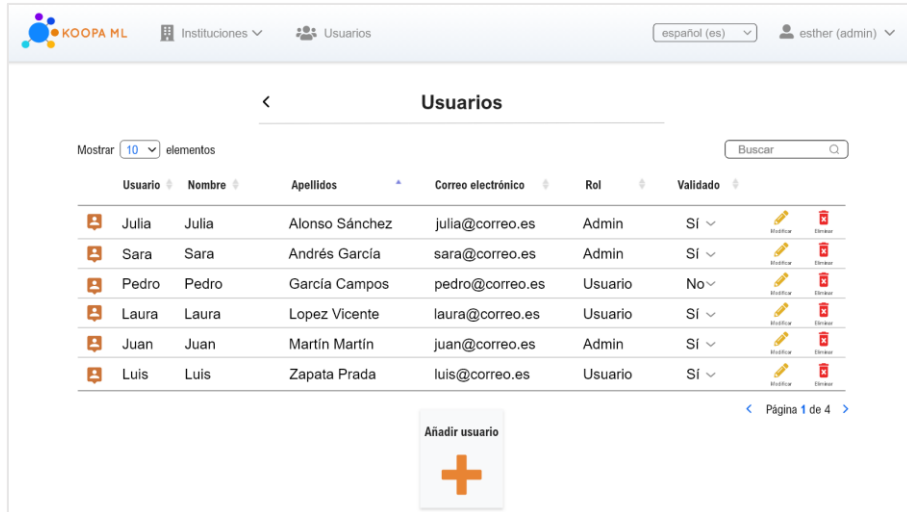


Figura 18: Primera versión prototipo digital: Listado de usuarios

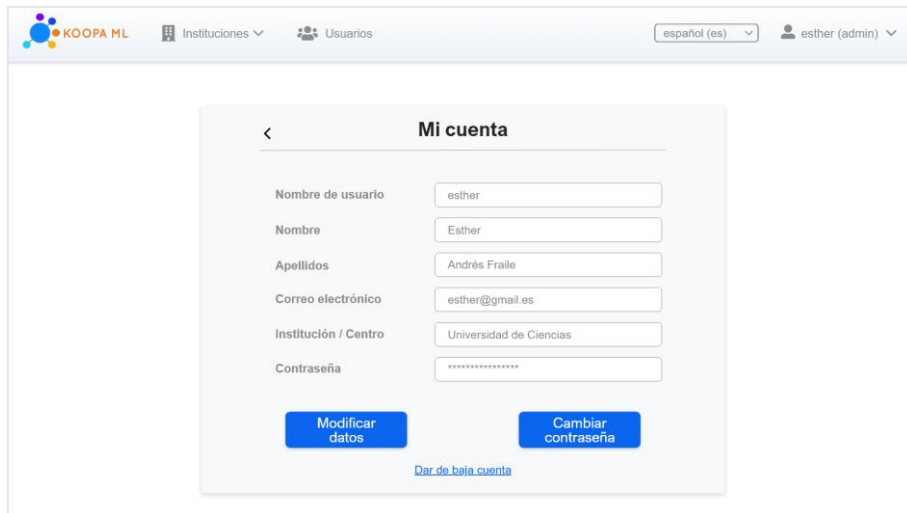


Figura 19: Primera versión prototipo digital: Cuenta

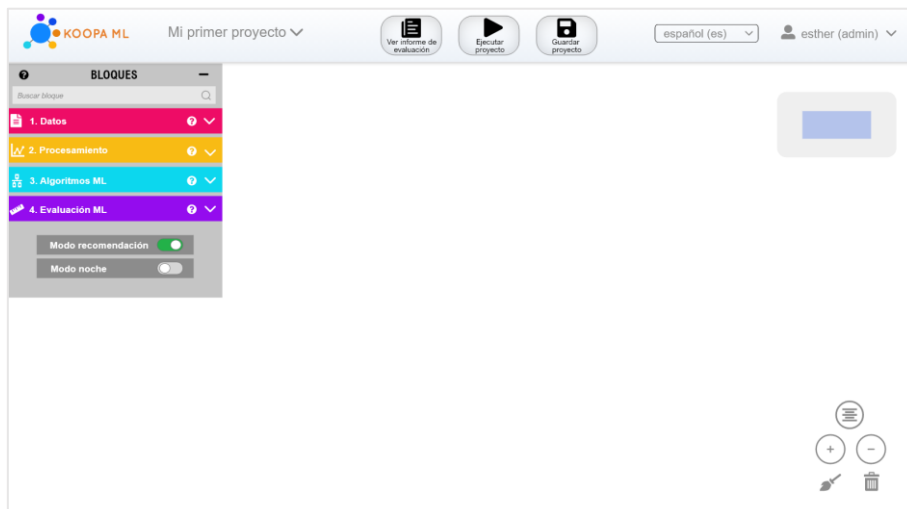


Figura 20: Primera versión prototipo digital: Área de trabajo

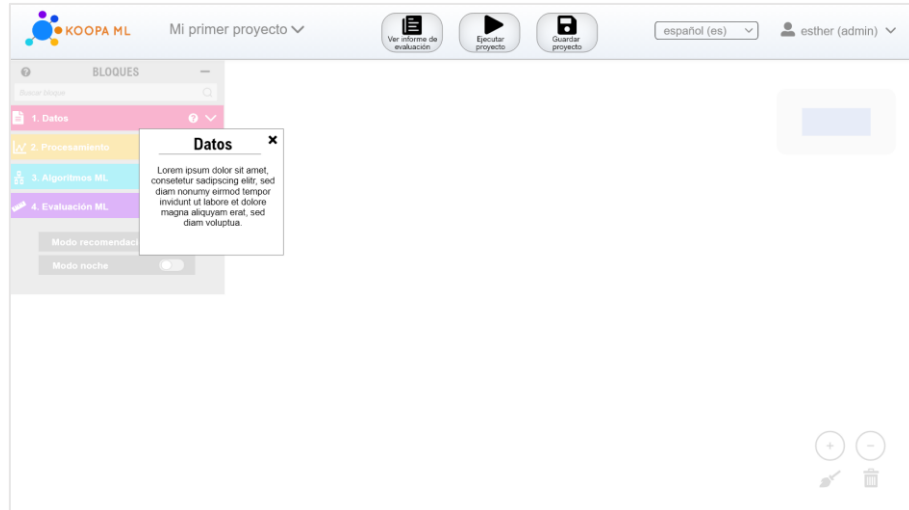


Figura 21: Primera versión prototipo digital: Información

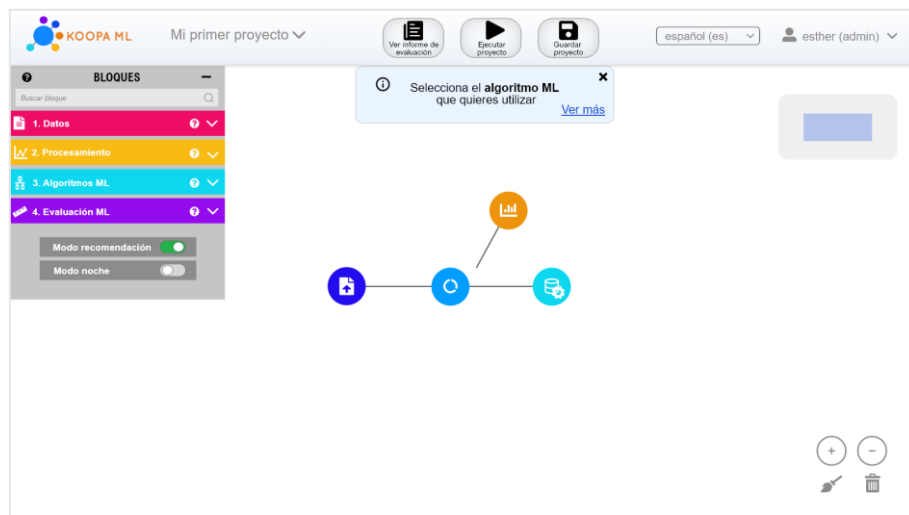


Figura 22: Primera versión prototipo digital: Recomendaciones

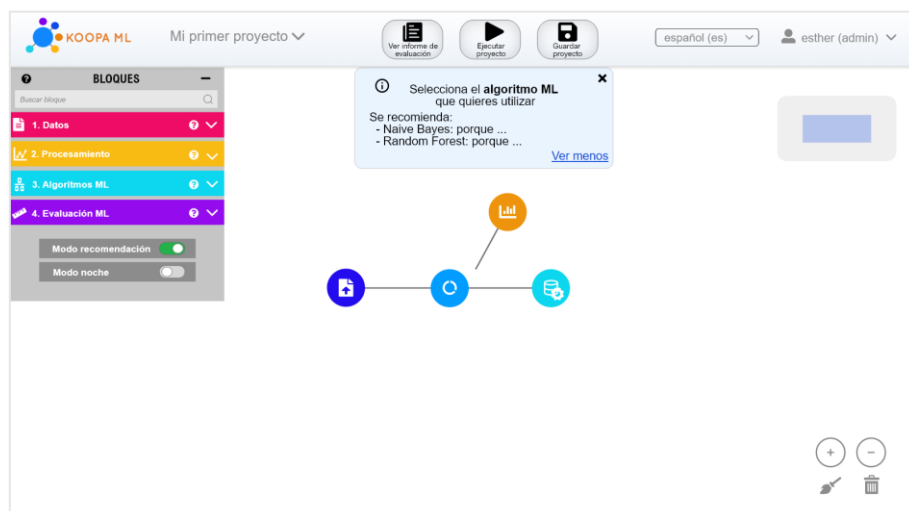


Figura 23: Primera versión prototipo digital: Recomendación completa

Se comentan los cambios más significativos:

- Se cambia la sintaxis empleada en la aplicación por otra más común y familiar a la de otras aplicaciones existentes
- Se añade la opción de poder visualizar la contraseña, si se desea, a la hora de iniciar sesión
- Se añade la opción de poder iniciar sesión con el correo electrónico y no solo con el nombre de usuario
- Se modifica el diseño de las pantallas unificando todas bajo un mismo patrón de diseño
- Se añade una pantalla de confirmación de registro exitoso
- Se añaden enlaces útiles desde unas pantallas a otras
- Se cambia el diseño de las pantallas principales de proyectos, instituciones, usuarios y heurísticas hacia uno mucho más vistoso e intuitivo.
- Las opciones de los proyectos dejan de estar escondidas en un submenú y se añaden algunas nuevas como campo de última modificación, duplicar y editar
- A la hora de crear un proyecto, se añade una pantalla de configuración en la que se podrá introducir un nombre y descripción en lugar de una creación por defecto
- La barra de navegación superior se simplifica
- Se añade la opción de poder ver tu cuenta y se completa con las opciones de modificar y cambiar contraseña
- Se añade la opción de poder modificar el nombre de usuario, así como el resto de campos
- Se añade el modo recomendación en el área de trabajo con su opción de desactivación
- Se añaden iconos de ayuda para solicitar información
- Se añade la opción de limpiar el área de trabajo
- Se modifica el panel lateral del área de trabajo, haciéndolo mucho más fácil de usar y separando las opciones de ejecución de los nodos
- Se añade la opción de poder buscar el nodo deseado
- Se crean todas las pantallas relacionadas con las heurísticas

▪ **Evaluación de la primera versión**

Para la evaluación de esta primera versión del prototipo se efectuó una prueba a un usuario perteneciente al grupo principal. Dicho usuario poseía conocimientos sobre el dominio de los datos, pero no de Machine Learning o aprendizaje automático.

Fue realizada de manera online a través de la compartición de pantalla de la aplicación *Zoom* y sin límite de tiempo. Se utilizó la herramienta digital en la que la persona en remoto puede utilizar el ratón local.

Una vez realizadas las pruebas de usuario pertinentes se extraen las siguientes conclusiones de modificación:

- Añadir la opción de volver a inicio en la barra superior
- Trasladar las opciones de ejecución a una barra nueva inferior en la que también se agregarán las opciones de zoom, centrado y papelera
- Añadir las opciones de eliminado y maximizado en los bloques minimizados mediante el deslizamiento del ratón por encima y no de forma explícita
- Se sustituirá el icono “i” de las recomendaciones por uno de una bombilla, para su mejor comprensión
- Se eliminará el desplegable de las instituciones
- Se utilizarán *tooltips* para indicar que bloque es cada uno
- El encabezado será fijo, se eliminan las sliders

Se podrán consultar la evaluación completa en el *Anexo III: proceso de diseño centrado en el usuario*.

▪ Segunda versión

Se realiza el segundo diseño de la aplicación y se incorporan todos los aspectos comentados en la evaluación de la primera versión realizada.

Se podrán observar de manera detallada todos los cambios realizados en el *Anexo III: proceso de diseño centrado en el usuario*. Se adjunta captura de pantalla de los más significativos para su comparación con la primera versión del prototipo. Se encontrarán en *Figura 24 – Figura 29*.

Además, esta segunda versión podrá consultarse al completo en el siguiente enlace:

<https://xd.adobe.com/view/f3e80a2d-b3f8-46ef-91f9-45fe91b9db62-a6bb/> .

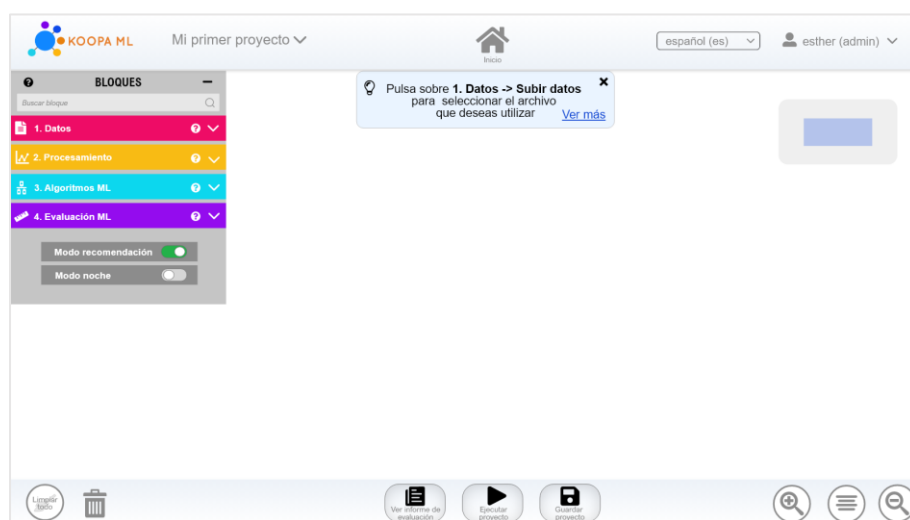


Figura 24: Segunda versión prototipo digital: Recomendaciones



Figura 25: Segunda versión prototipo digital: Tooltips

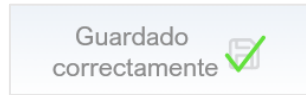


Figura 26: Segunda versión prototipo digital: Guardado



Figura 27: Segunda versión prototipo digital: Área de trabajo



Figura 28: Segunda versión prototipo digital: Construcción de flujo

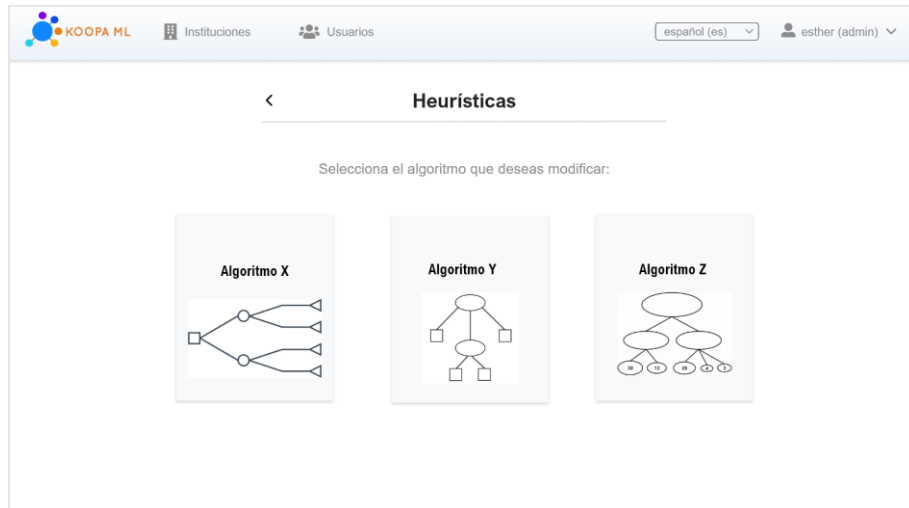


Figura 29: Segunda versión prototipo digital: Heurísticas

Se comentan los cambios más significativos:

- Se agregan *tooltips* a aquellas opciones cuya funcionalidad no queda clara
- Se añaden las opciones de maximización y eliminación en el bloque minimizado
- Se agrega la opción de volver a inicio desde cualquier parte de la aplicación
- Se optimizará aún más la barra de navegación
- Se eliminan los *scrolls* laterales
- Se agrega la funcionalidad de Guardar proyecto a la barra de navegación

▪ **Evaluación segunda versión**

Para la evaluación de esta primera versión del prototipo se efectuaron dos pruebas de usuario. Ambos participantes son pertenecientes al grupo principal y poseen conocimientos sobre el dominio de los datos, pero no de Machine Learning o aprendizaje automático. Además, se destaca que están familiarizados con la aplicación existente de *Koopa ML*. Gracias a esto podrán, no solo valorar la nueva versión, si no determinar si realmente los cambios y funcionalidades añadidos suponen una mejora.

Fue realizada a partir de la compartición del link generado por la aplicación *Adobe XD* en la cual fue realizado el prototipo. Ambos lo probaron sin recibir ninguna indicación sobre su uso y anotaron sus opiniones en un informe.

Las conclusiones extraídas de dicho informe han sido:

- Se destaca la disposición intuitiva y de fácil uso de los botones, tanto a nivel del perfil de usuario, como de la disposición de los proyectos o del proyecto en sí
- Resulta realmente valorada la nueva funcionalidad de recomendaciones e informaciones teóricas
- Valoran la unificación de diseños en pantallas similares como las de Inicio de sesión – Registro – Panel de usuarios – Panel de administrador

- Se considera mejor el nuevo diseño realizado en las pantallas de Proyectos, Instituciones y Usuarios
- Se prefiere la separación de las opciones de creación en lugar de su inclusión en la barra de navegación
- Se agradece la incorporación de una pantalla de configuración del proyecto
- Se sugieren distintos formatos de texto para la subida de archivos
- No se comprende la funcionalidad de algunos bloques de datos como el de procesamiento o codificación de datos
- Se sugiere la opción de poder añadir conjuntos de datos nuevos al modelo una vez haya sido completado y evaluado
- La interfaz de modificación de heurísticas no es del todo entendible

▪ **Conclusiones**

Los cambios significativos de diseño, así como las nuevas funcionalidades incorporadas han sido evaluadas muy positivamente. Es por ello por lo que no se considera necesaria la realización de una tercera versión del prototipo digital.

Se implementará, por tanto, la segunda versión del prototipo digital junto con las mejoras que vayan surgiendo en el proceso de creación.

Heurísticas

Para la representación de las heurísticas en forma de diagrama de flujo se ha utilizado FlowChart.js. A partir de un campo de texto y el correspondiente pseudocódigo que se introduce en él, esta biblioteca es capaz de dibujar un diagrama de flujo que contenga la información proporcionada.

Campo de texto

Se utilizará un lenguaje parecido al pseudocódigo y que el administrador deberá conocer. En primer lugar, se declararán las variables, que representarán los nodos del diagrama de flujo. Su sintaxis es la siguiente:

nombre de la variable => tipo de la variable : Texto para el diagrama

Por ejemplo:

st=>start: Inicio

e=>end: Fin

cond1=>condition: SubirDatos?

cond2=>condition: Rellenar?

op1=>operation: Inserta el nodo 1. Datos

En segundo lugar, se indicarán las conexiones entre las variables previamente declaradas. Representarán las flechas del diagrama de flujo. Su sintaxis es la siguiente:

nombre de la variable(opciones) -> variable a la que se une

Por ejemplo:

st->cond1

cond1(no)->op1->e

cond1(yes)->cond2

El ejemplo propuesto mostraría el resultado observado en la *Figura 30* .

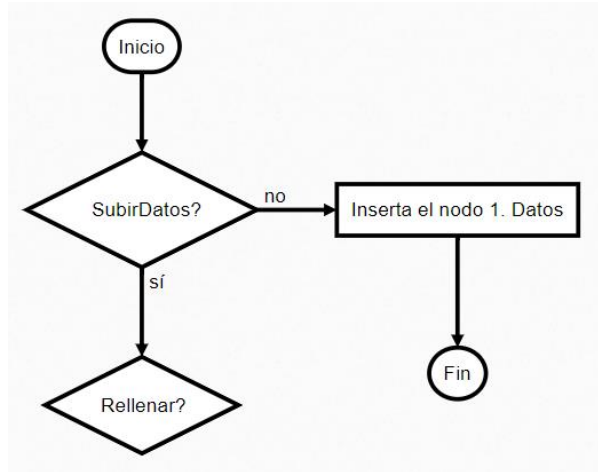


Figura 30: Diagrama heurística de ejemplo

El aspecto del campo de texto en la aplicación se puede consultar en la *Figura 31*.

La imagen muestra una interfaz de usuario con un campo de texto que contiene el siguiente código de heurísticas:

```
st->start: Inicio  
e->end: Fin  
cond1->condition: SubirDatos?  
cond2->condition: Rellenar?  
cond3->condition: Codificar?  
cond4->condition: Separar/Validar?  
cond5->condition: Algoritmo?  
cond6->condition: Evaluación?  
op1->operation: Inserta el nodo 1. Datos  
op2->operation: Inserta el nodo 2.1 Rellenar valores
```

Debajo del campo de texto hay dos botones azules: 'Guardar' (con un icono de disco) y 'Gráfico' (con un icono de gráfico). Debajo de los botones hay una casilla de verificación marcada con 'Usar esta heurística'.

Figura 31: Campo de texto heurísticas

Como se puede observar, incluye a parte de la zona de escritura algunas opciones:

- Guardar: guardará todos los cambios realizados en el código desde el último guardado.

- Actualizar: todos los cambios realizados en el código se verán reflejados automáticamente en el diagrama
- Usar esta heurística: este *checkbox* nos permitirá seleccionar la heurística que se desea utilizar para el comportamiento de la aplicación

Diagrama

Dibujará el diagrama de flujo resultante del pseudocódigo del campo de texto. Se actualizará cada vez que se quiera, obteniendo un resultado automático.

El aspecto del diagrama en la aplicación puede consultarse en la *Figura 32*.

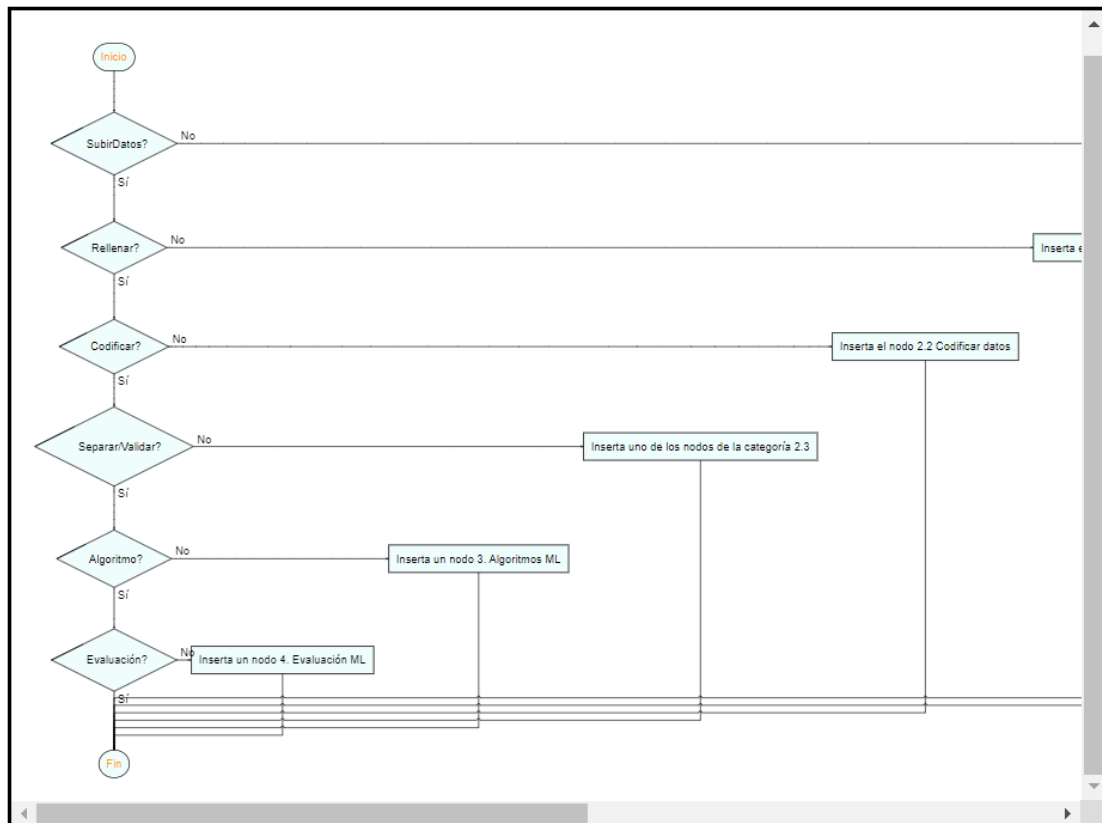


Figura 32: Diagrama de heurística

La página de modificación de heurísticas tiene, por tanto, la interfaz mostrada en la *Figura 33*.

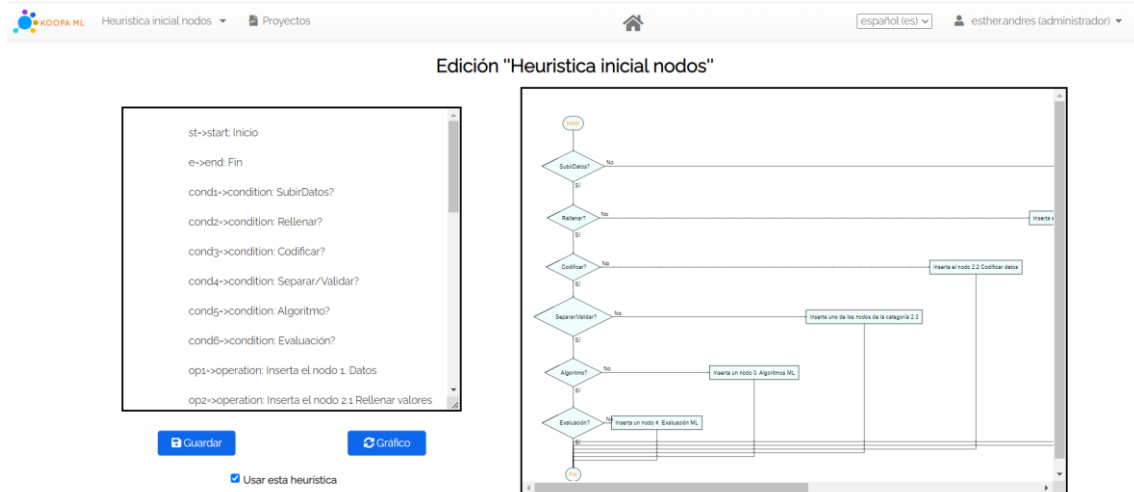


Figura 33: Interfaz modificación de heurísticas

La interfaz para llegar a cada heurística es mostrada en la Figura 34.



Figura 34: Interfaz listado de heurísticas

Y la de creación puede consultarse en la Figura 35.

Figura 35: Interfaz creación de una heurística

Se permitirá la modificación de nombre y descripción, como se puede observar en la Figura 36.

ID	Nombre	Descripción	Fecha de creación	Última modificación	
93	Heurística conexiones	Segunda	03/07/2022	05/07/2022	✓ ✗ 🗑️
91	Heurística inicial nodos	Primera	01/07/2022	05/07/2022	✎ 🗑️
94	Pruebas		04/07/2022	05/07/2022	✎ 🗑️

Figura 36: Interfaz edición de heurística

Comportamiento dinámico

Uno de los mayores desafíos presentados en el proyecto fue la realización de un sistema de recomendaciones *dinámico*.

Las heurísticas no solo se pueden crear, consultar y eliminar si no que pueden modificarse de manera que el comportamiento de la aplicación cambie automáticamente.

Esto nos permitirá no solo seleccionar una u otra heurística en función de las necesidades si no editar la funcionalidad que implica cada una de ellas.

El resultado será mostrado en el área de trabajo y tendrá la apariencia mostrada en la Figura 37.

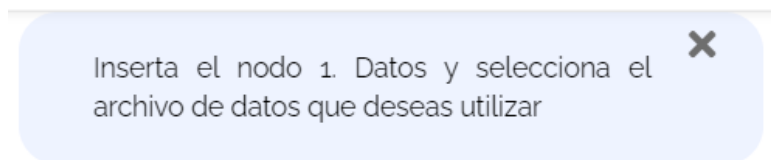


Figura 37: Ejemplo de recomendación

Aparecerá en el área de trabajo de los proyectos de la siguiente manera, pudiendo ser activado o desactivado. Ver Figuras 38, 39 y 40.



Figura 38: Toggle activación/desactivación recomendaciones

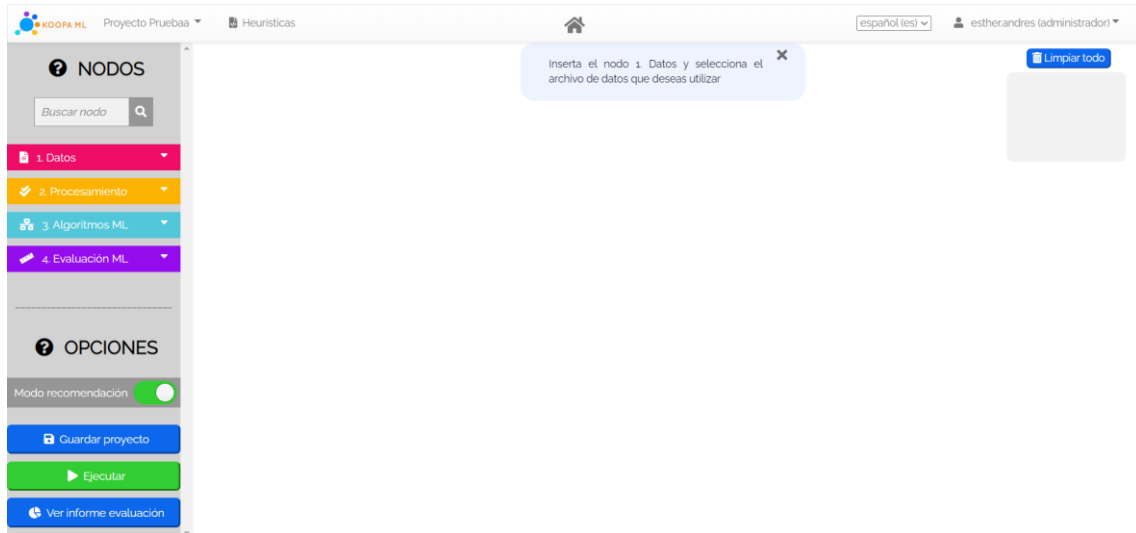


Figura 39: Área de trabajo con las recomendaciones activadas

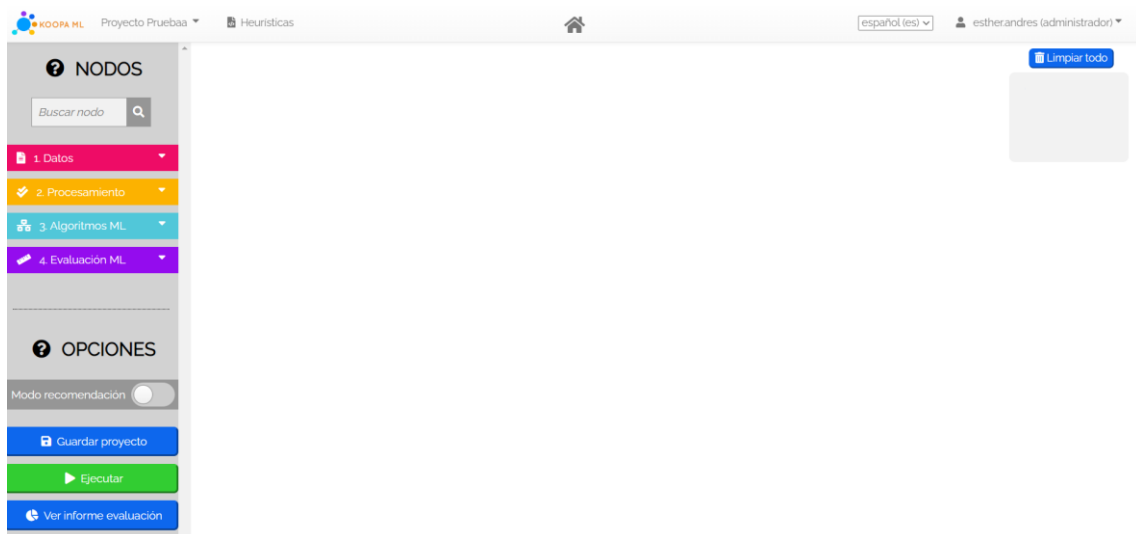


Figura 40: Área de trabajo con las recomendaciones desactivadas

Según se vaya construyendo el flujo de nodos el texto de recomendación irá variando en función del comportamiento definido en la heurística. Se puede observar esta funcionalidad en las Figuras 41, 42 y 43.

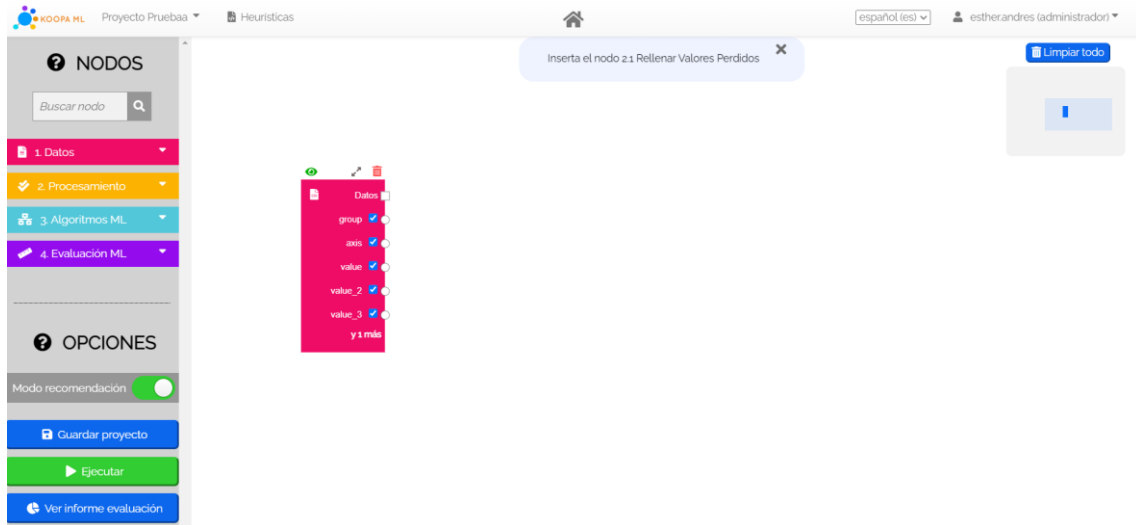


Figura 41: Ejemplo comportamiento de la heurística 1

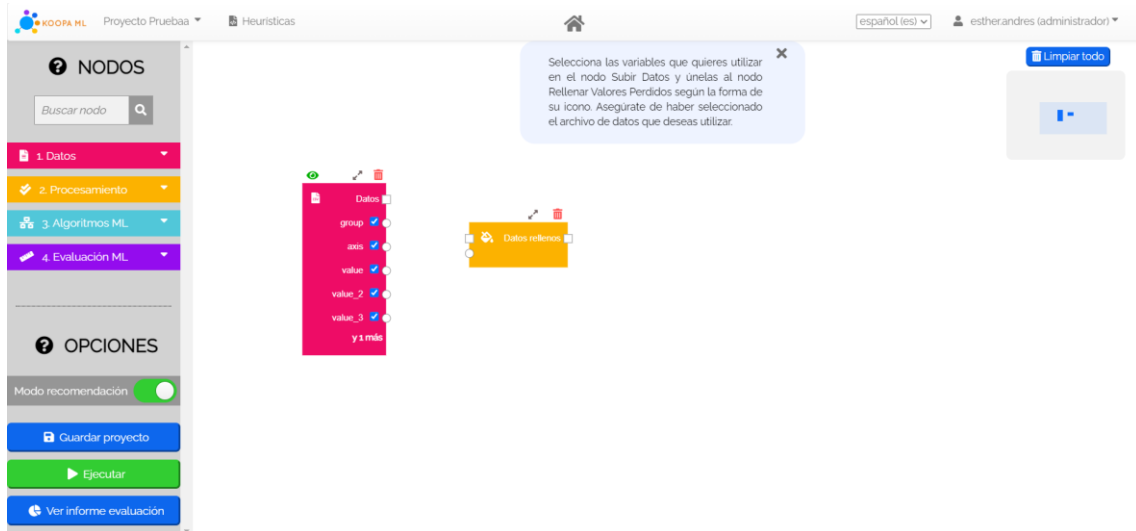


Figura 42: Ejemplo comportamiento de la heurística 2

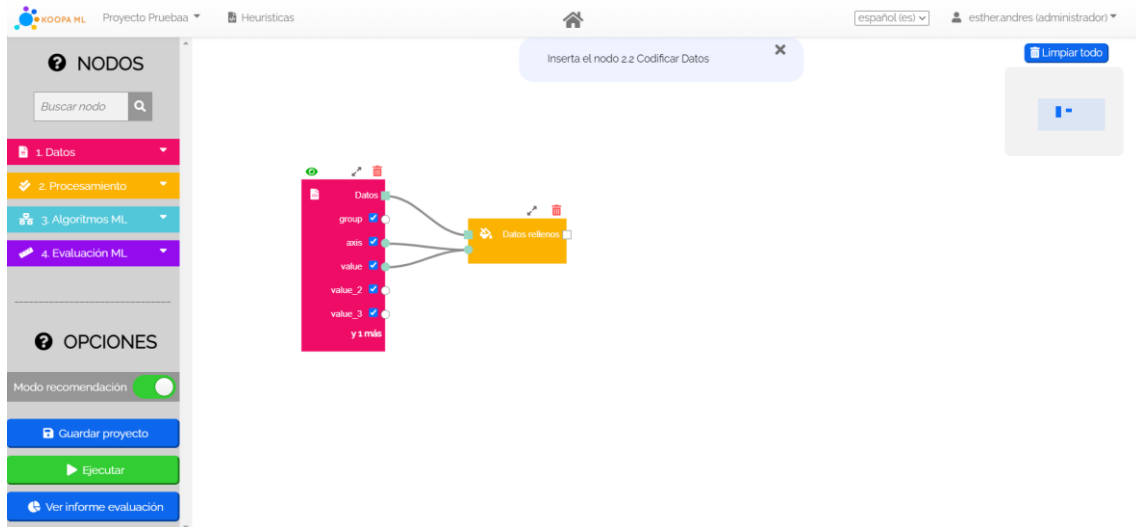


Figura 43: Ejemplo comportamiento de la heurística 3

Si por ejemplo se quisiera modificar la recomendación de la *Figura 23* sería tan fácil como:

1. Acceder a la heurística correspondiente
2. Localizar la variable que implica ese comportamiento

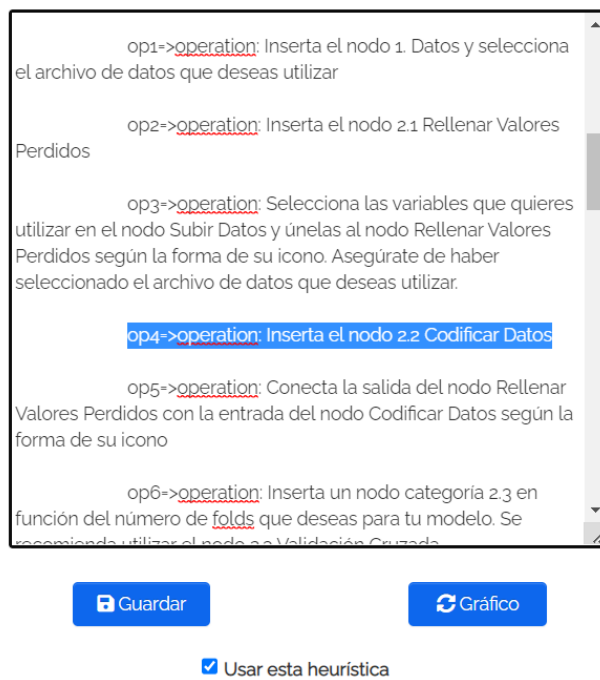


Figura 44: Paso 2: modificación de heurística

3. Modificar la variable

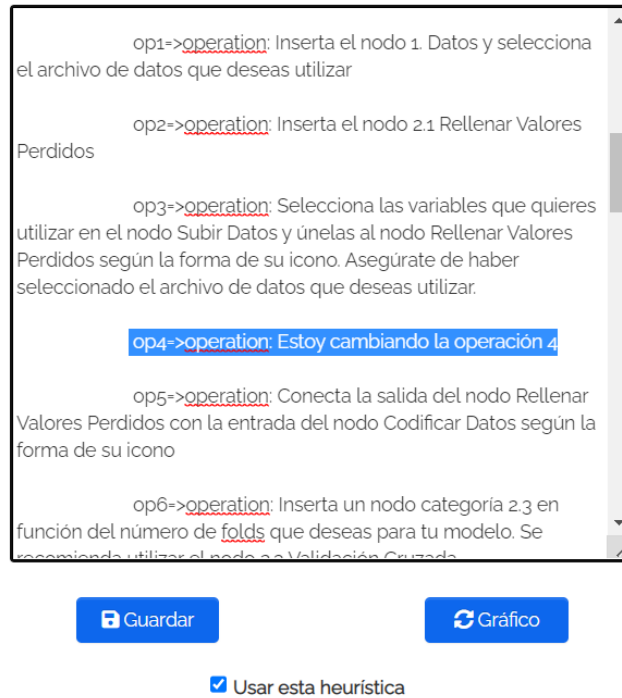
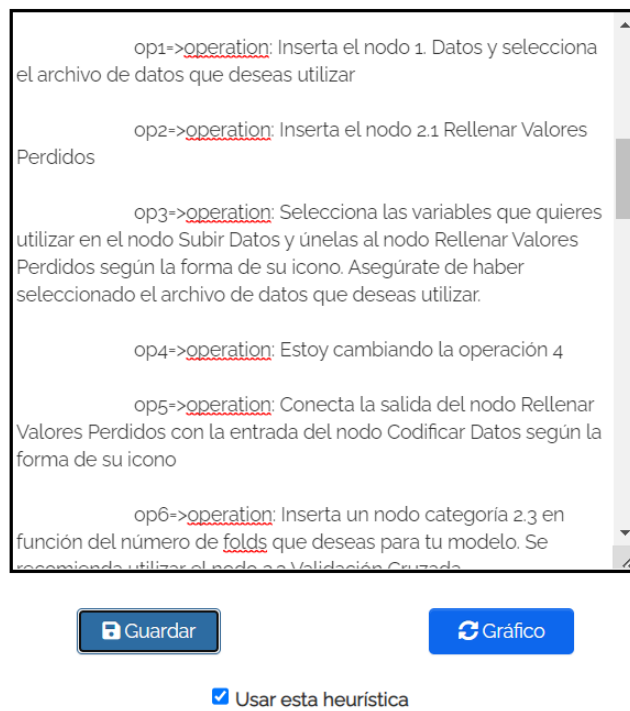


Figura 45: Paso 3: modificación de heurística

4. Guardar los cambios



Guardado! Prueba a actualizar para ver los cambios en el diagrama

Figura 46: Paso 4: modificación de heurística

Si se accede de nuevo al proyecto de ejemplo se verá cómo, en efecto, la heurística ha cambiado. Observar este comportamiento en la *Figura 47*.

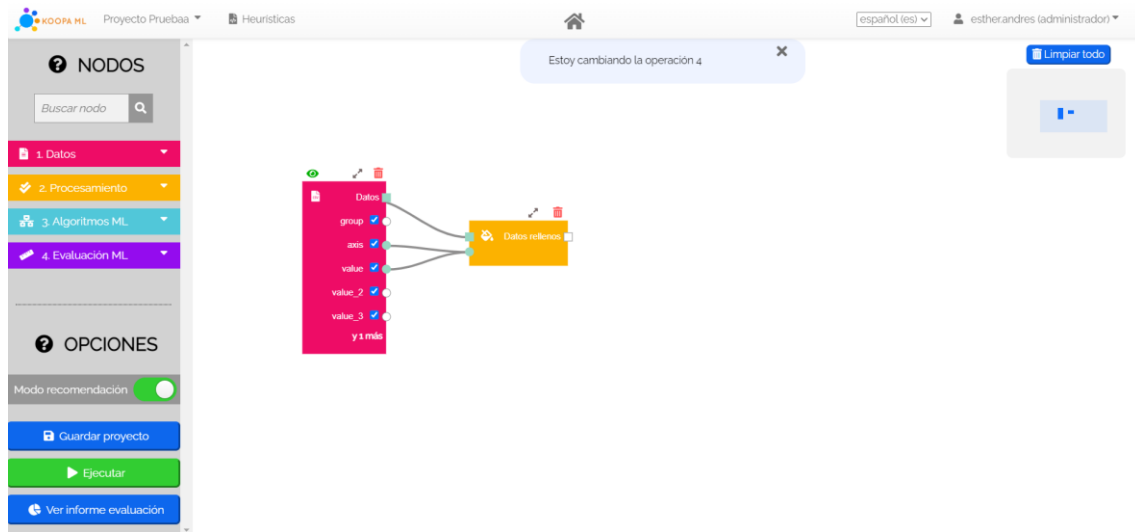


Figura 47: Resultado modificación de heurística

De igual manera se procedería al cambio de heurística. Solo sería necesario marcar el *checkbox* de la que se desee utilizar y guardar los cambios.

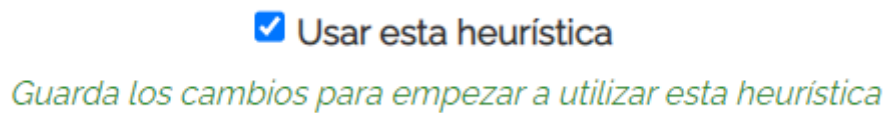


Figura 48: Checkbox elección de heurística

Seguridad

Se ve necesario comentar que para dotar a la aplicación de seguridad las heurísticas solo podrán ser consultadas y modificadas por los administradores.

De igual modo, el código de las heurísticas que implica condiciones está limitado. No cualquier condición podrá ser ejecutada.

Todo ello se controlará mediante el uso de un diccionario que posteriormente será explicado.

Código

Para llevar a cabo la implementación de este sistema de recomendación dinámico se hizo uso de dos funciones.

Función obtener recomendación

El código es el siguiente:

```
@login_required(login_url='/login/')
def get_pipeline_recommendation(request, pipeline_id):
    heuristic = Heuristic.objects.get(default=True)
    pipeline = Pipeline.objects.get(id=pipeline_id)
    nodes = Node.objects.filter(pipeline_id=pipeline_id)
    code_pipeline = heuristic.code
    parsed_pipeline = {
        '00': [], '11': [], '12': [], '10': [], '14': [], '2_': [],
'3_': []
    }
    translate_dictionary = {
        'SubirDatos?': "len(parsed_pipeline.get('00')) == 0",
        'Rellenar?': "len(parsed_pipeline.get('11')) == 0",
        'Codificar?': "len(parsed_pipeline.get('12')) == 0",
        'Separar/Validar?': "(len(parsed_pipeline.get('10')) == 0) and
(len(parsed_pipeline.get('14')) == 0)",
        'Algoritmo?': "len(parsed_pipeline.get('2_')) == 0",
        'Evaluación?': "len(parsed_pipeline.get('3_')) == 0",
        'SalidaSubirDatos?':
"parsed_pipeline.get('00')[1]['outputs']==False",
        'SalidaRellenar?':
"parsed_pipeline.get('11')[1]['outputs']==False",
        'SalidaCodificar?':
"parsed_pipeline.get('12')[1]['outputs']==False",
        'SalidaSeparar/Validar?': "(len(parsed_pipeline.get('10'))!=0
and parsed_pipeline.get('10')[1]['outputs']==False) or
(len(parsed_pipeline.get('14'))!=0 and
parsed_pipeline.get('14')[1]['outputs']==False)",
        'SalidaAlgoritmo?':
"parsed_pipeline.get('2_')[1]['outputs']==False",
    }

    for node in nodes:
        if node.node_type not in parsed_pipeline.keys():
            parsed_pipeline[node.node_type] = []
            parsed_pipeline[node.node_type].append({'inputs':
node.inputs})
            parsed_pipeline[node.node_type].append({'outputs':
node.outputs})
        if node.node_type.startswith('2'):
            parsed_pipeline['2_'].append({'inputs': node.inputs})
            parsed_pipeline['2_'].append({'outputs': node.outputs})
        if node.node_type.startswith('3'):
            parsed_pipeline['3_'].append({'inputs': node.inputs})
            parsed_pipeline['3_'].append({'outputs': node.outputs})
```



```

# Se eliminan los espacios en blanco del código
cleaned_code = code_pipeline
valueToBeRemoved = ''
cleaned_code = [value for value in cleaned_code if value !=
valueToBeRemoved]

# Se divide cada elemento de la lista en antes de => (keys) y
despues (values)
key_list = []
value = []
for par in cleaned_code:
    parts = par.split("=>")
    key_list.append(parts[0])
    value.append(parts[-1])

# Ajustamos el formato de la lista de valores
value_list = []
for par in value:
    parts = par.split(": ")
    value_list.append(parts[-1])

# Se crea el diccionario de pares key-value
dictionary_code = {k: v for k, v in zip(key_list, value_list)}

# Creamos un nuevo diccionario solo con los campos necesarios
validos = []
for clave in dictionary_code:
    if (clave.startswith('op')) or (clave.startswith('cond')):
        if "->" not in clave:
            validos.append(clave)
dictionary_valid = {x: dictionary_code[x] for x in validos}

# Dividimos el diccionario en uno de condiciones y otro de
operaciones
conditions = []
operations = []
for clave in dictionary_valid:
    if clave.startswith('cond'):
        conditions.append(clave)
    else:
        operations.append(clave)
dictionary_cond = {x: dictionary_valid[x] for x in conditions}
dictionary_op = {x: dictionary_valid[x] for x in operations}

# En funcion de como son los nodos del flow usamos un valor del
diccionario
result = recorrerDiccionario(dictionary_cond, dictionary_op,
parsed_pipeline, translate_dictionary)

return JsonResponse({"text": result}, status=200)

```

La lógica que se utiliza es la siguiente:

En primer lugar:

- Se indica la heurística que se va a utilizar: será aquella que tenga el *checkbox* de uso activado
- Se recupera el código de dicha heurística seleccionada
- Se indica el proyecto en el cual se van a reflejar las recomendaciones correspondientes
- Se construye un diccionario que traduce el lenguaje natural del diagrama de heurística en lenguaje máquina

En segundo lugar:

- Se crea un diccionario con los nodos contenidos en el proyecto mediante su previa inicialización y la lectura de su flujo de datos
- Se le da formato al código recuperado de la heurística: se eliminan los espacios en blanco, se dividen los campos entre claves o valores y se crea finalmente el diccionario de la heurística
- Del diccionario de la heurística se construirán otros dos, el de condiciones y el de operaciones

Una vez tenemos todo inicializado y construido se llamará a una función recursiva que recorrerá los diccionarios de condición y operación obteniendo el texto de recomendación que deberá devolverse al proyecto en función de sus características.

Para mostrar el resultado en el área de trabajo se utiliza el siguiente código de JavaScript:

```
$.ajax({
  url: url,
  type: 'POST',
  processData: false,
  contentType: false,
  data: formData,
  beforeSend: function (xhr, settings) {
    xhr.setRequestHeader("X-CSRFToken", csrf_token);
  },
  success: function (result) {
    console.log(result)

    $.ajax({
      url: '/get_pipeline_recommendation/' + pipeline_id,
      type: 'GET',
      success: function (result) {
        let recommendation_text = result['text']
        console.log(recommendation_text)
        $('#recommendation_text').html(recommendation_text)
      },
    });
  }
});
```

Función recursiva

El código es el siguiente:

```
def recorrerDiccionario(dictionary_cond, dictionary_op,
    parsed_pipeline, translate_dictionary):
    for u, v in dictionary_cond.items():
        for x, w in dictionary_op.items():
            if v in translate_dictionary:
                question = translate_dictionary[v]
            else:
                question = v

            if (eval(question)):
                resultado = w
            else:
                conditions = []
                for clavel in dictionary_cond:
                    if clavel != u:
                        conditions.append(clavel)
                dictionary_cond = {u: dictionary_cond[u] for u in
conditions}

                operations = []
                for clave in dictionary_op:
                    if clave != x:
                        operations.append(clave)
                dictionary_op = {x: dictionary_op[x] for x in
operations}

                resultado = recorrerDiccionario(dictionary_cond,
dictionary_op, parsed_pipeline,
                                                    translate_dictionary)

            if (resultado == None):
                resultado = "Prueba a ejecutar y ver el informe de
evaluación del flujo construido"

        return resultado
```

La lógica que se utiliza es la siguiente:

La función recursiva recibe por parámetros el diccionario de condiciones, el de operaciones, el de traducciones y el flujo de datos del proyecto.

En primer lugar, se recorren los diccionarios de condiciones y de operaciones. De este recorrido se obtendrá la condición que se tiene que evaluar y su correspondiente operación sucesoria. La condición será traducida gracias al diccionario traductor.

En segundo lugar, y una vez obtenida la condición en lenguaje máquina, se evalúa:

- Si la condición se cumple: el resultado será la operación previamente obtenida. Se devuelve y se pone fin a la función recursiva.
- Si la condición no se cumple: se deberá pasar a evaluar la siguiente condición y operación.
Para ello se elimina la condición y operación ya evaluadas de sus respectivos diccionarios y se procede a una llamada recursiva con los nuevos resultantes.

Cuando los diccionarios se vacían por haber evaluado todas sus condiciones y operaciones correspondientes el resultado será una frase predefinida de finalización.

Información teórica

Se desea destacar la incorporación de ayudas a lo largo de toda el área de trabajo de los proyectos.

Explicarán, brevemente, qué hace cada uno de los apartados de la zona de trabajo y qué características posee cada nodo ML.

Se busca con ella la familiarización de los usuarios con la materia *Machine Learning*.

Se usará el icono mostrado en la *Figura 49*.



Figura 49: Icono información

Cada vez que se pinche sobre él se obtendrá la información buscada.

Algunos ejemplos:



Figura 50: Ejemplo de apartados con ayuda

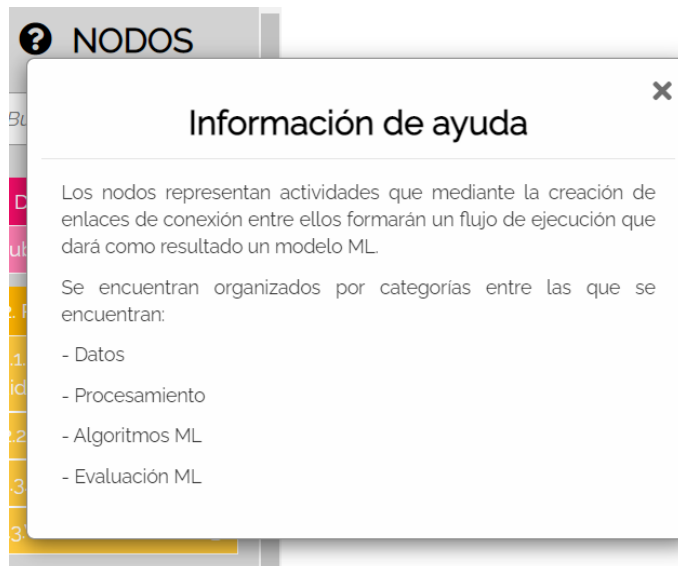


Figura 51: Ejemplo ayuda apartado Nodos

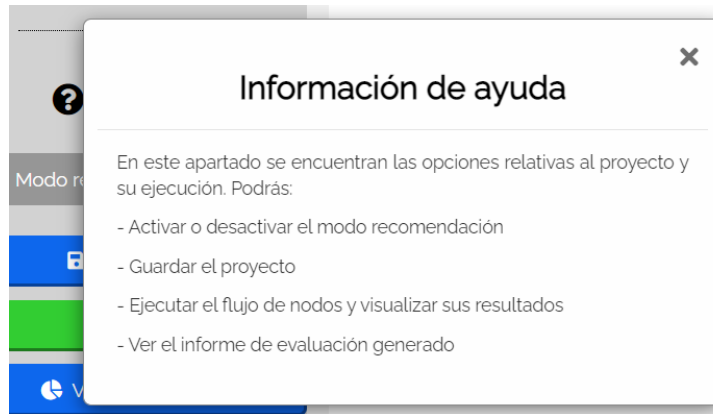


Figura 52: Ejemplo ayuda apartado Opciones

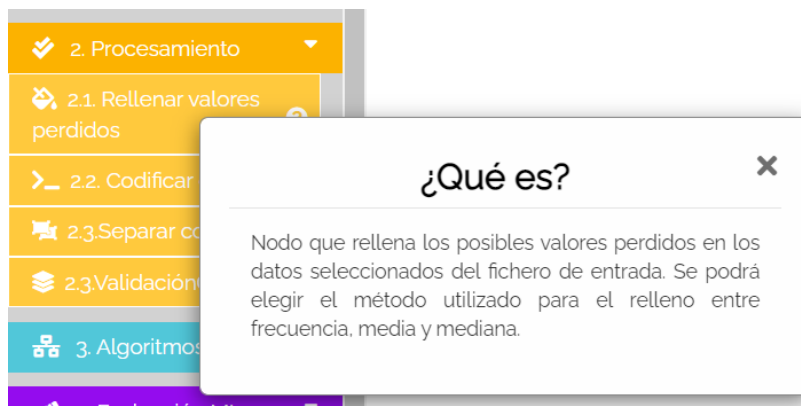


Figura 53: Ejemplo ayuda apartado Nodo Rellenar Valores Perdidos

6. Conclusiones y líneas futuras de trabajo

En este apartado se presentarán las conclusiones derivadas del desarrollo del sistema implementado. Se detallarán también posibles futuras líneas de trabajo que dotarán al sistema de una mayor funcionalidad y robustez.

Conclusiones

Tal y como se comentó al inicio del informe, el *Machine Learning* o aprendizaje automático es una disciplina muy útil que cada vez está presente en más ámbitos de nuestra vida diaria, incluido el de la medicina.

La oportunidad de haber podido trabajar en un proyecto relacionado con ello ha sido totalmente satisfactoria.

Al principio supuso un reto, ya que no solo había que crear funcionalidades nuevas si no que había que entender un gran proyecto ya realizado. Sin embargo, poco a poco y tras mucho esfuerzo se consiguió una familiarización total con él.

Además, se aprovechó esta situación para localizar aquellos puntos en los que más hacía falta una reestructuración ya que antes de ser desarrolladora la autora fue usuaria.

Poder detectar fallos o cosas mejorables y tener la oportunidad de modificarlas ha sido realmente gratificante. Durante todo el proceso de creación se ha seguido el principio de “*el primer paso para saber utilizar algo es entenderlo*”. Es por ello por lo que se tomó la decisión de realizar un rediseño total de la aplicación, pudiendo aportar un granito de arena a su mejora. Se aprovecha para agradecer a las tutoras su permiso para poder hacer tantos cambios como fuera necesario sin oponer nunca ninguna limitación.

De igual modo, este inicio de la relación entre desarrolladora y código a través del diseño sirvió para comprender mucho mejor cómo funciona un lenguaje como Python y como se había llevado a cabo la estructura de la aplicación existente. A partir de ahí, el desarrollo de nuevas funcionalidades para la aplicación fue mucho más sencillo.

La búsqueda de los apartados de información teórica en forma de ayuda sirvió para ampliar el conocimiento propio de la desarrolladora sobre *Machine Learning*. Antes de intentar enseñar algo a alguien debes conocerlo tú.

Sin duda, la parte más interesante ha sido la relacionada con las heurísticas.

Dicho apartado, dotará de un valor añadido a la aplicación ya que gracias a su dinamismo la aplicación admitirá cualquier cambio repentino o a largo plazo y de manera muy sencilla. Con la agregación de más heurísticas se podrá controlar y guiar cualquier apartado de la aplicación, incluso los que aún no existan.

Además, de nada sirve una aplicación si los usuarios tienen algún tipo de dificultad en su uso y no pueden aprovecharla al completo. Ahora esa usabilidad ha aumentado y se espera con ello

un mayor número de usuarios que puedan aprovechar las grandes posibilidades que la herramienta brinda.

Por último, comentar lo que a nivel personal ha supuesto este trabajo.

La realización de este gran proyecto ha permitido aplicar los conocimientos que durante 4 años se han obtenido en el grado. Los más útiles y que se ha agradecido tener han sido los relacionados con Ingeniería del Software, Interacción Persona Ordenador y Administración de Sistemas. Sin embargo, todos los obtenidos han ayudado no solo a la implementación si no a la búsqueda de soluciones cuando surgían inconvenientes.

Se agradece al proyecto el poder haber podido aprender un lenguaje de programación tan valorado actualmente como es Python, así como el haber ampliado los conocimientos de *front-end* en los lenguajes HTML, CSS y JavaScript.

Llevar a cabo este proyecto me ha hecho sentir dentro de un proyecto real por lo que también se considera a la desarrolladora mucho más preparada para el mundo laboral.

Líneas futuras de trabajo

Como se ha podido observar a lo largo de todo el proyecto, la aplicación desarrollada es sin duda una herramienta necesaria en la actualidad. Es por ello por lo que no puede quedar aquí si no que sería recomendable seguir ampliando sus funcionalidades.

Durante el desarrollo se han observado ciertos puntos en los que se podría trabajar en un futuro:

- Recomendaciones activas: el sistema no solo te sugerirá que debes hacer si no que podrás *aceptar la recomendación* de tal manera que el sistema implemente el flujo automáticamente en lugar de que sea el usuario quien lo haga
- Modos de ejecución: se podrían llevar a cabo diferentes modos de ejecución como los existentes en cualquier IDE de programación como podría ser la ejecución paso a paso
- Modo noche: funcionalidad muy reclamada en la actualidad por los usuarios
- Opciones para el espacio de trabajo como el zoom, el centrado o deshacer y rehacer una acción
- Incorporación de más idiomas
- Incorporación de más algoritmos y métodos de evaluación ML
- Poder exportar el flujo de datos en distintos formatos
- Permitir a los usuarios compartir entre ellos los proyectos que deseen desde la propia aplicación

Referencias

- ¹ G. Litjens et al., "A survey on deep learning in medical image analysis," (in eng), *Med Image Anal*, vol. 42, pp. 60-88, Dec 2017, doi: 10.1016/j.media.2017.07.005.
- ² S. González Izard, R. Sánchez Torres, Ó. Alonso Plaza, J. A. Juanes Méndez, and F. J. García-Peñalvo, "Nextmed: Automatic Imaging Segmentation, 3D Reconstruction, and 3D Model Visualization Platform Using Augmented and Virtual Reality," (in eng), *Sensors (Basel)*, vol. 20, no. 10, p. 2962, 2020, doi: 10.3390/s20102962.
- ³ S. G. Izard, J. A. Juanes, F. J. García Peñalvo, J. M. G. Estella, M. J. S. Ledesma, and P. Ruisoto, "Virtual Reality as an Educational and Training Tool for Medicine," *Journal of Medical Systems*, vol. 42, no. 3, p. 50, 2018/02/01 2018, doi: 10.1007/s10916-018-0900-2.
- ⁴ X. Ferrer, T. van Nuenen, J. M. Such, M. Coté, and N. Criado, "Bias and Discrimination in AI: a cross-disciplinary perspective," *IEEE Technology and Society Magazine*, vol. 40, no. 2, pp. 72-80, 2021, doi: 10.1109/MTS.2021.3056293.
- ⁵ S. Hoffman, "The Emerging Hazard of AI-Related Health Care Discrimination," *Hastings Center Report*, vol. 51, no. 1, pp. 8-9, 2021, doi: 10.1002/hast.1203.
- ⁶ S. Wachter, B. Mittelstadt, and C. Russell, "Why fairness cannot be automated: Bridging the gap between EU non-discrimination law and AI," *Computer Law & Security Review*, vol. 41, p. 105567, 2021, doi: 10.2139/ssrn.3547922.
- ⁷ Alonso Sánchez, Julia. Trabajo Fin de Grado: Plataforma para dar soporte a la aplicación de Machine Learning en el contexto médico. Universidad de Salamanca. 2021.
- ⁸ "Guía de realización y documentación Proyecto Fin de Carrera en la Ingeniería Técnica Informática," Diaweb, Universidad de Salamanca.
- ⁹ F. J. García Peñalvo, A. García Holgado, and A. Vázquez Ingelmo, "Transparencias de Ingeniería del Software I," tema 7, Proceso Unificado, Universidad de Salamanca.
- ¹⁰ E.V.A. UCI, I. D. S. Conferencia #1, "Introducción a la Ingeniería de Software, ISW 1," Ecured s. f, https://www.ecured.cu/Proceso_unificado_de_desarrollo .
- ¹¹ F. J. García Peñalvo, A. García Holgado, and A. Vázquez Ingelmo, "Transparencias de Ingeniería del Software I," tema UML, Fundamentos de la vista de casos de uso, Universidad de Salamanca.
- ¹² A. Durán Toro, and B. Bernárdez Jiménez, "Metodología para la Elicitación de Requisitos de Sistemas Software".
- ¹³ J. Alonso Sánchez, "Plataforma para dar soporte a la aplicación de Machine Learning en el contexto médico," Trabajo Fin de Grado, Universidad de Salamanca, 2021.
- ¹⁴ F. J. García Peñalvo, A. García Holgado, and A. Vázquez Ingelmo, "Transparencias de Ingeniería del Software I," tema 5, Ingeniería de Requisitos, Universidad de Salamanca.

¹⁵ R. Therón, "Transparencias de Interacción Persona Ordenador," tema 4, Diseño centrado en el usuario, Universidad de Salamanca.

¹⁶ Interaction Design Foundation, "Course: The Practical Guide to Usability," <https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them> .

¹⁷ Justinmind, "How to design user scenarios: best practices and examples," reference: <https://www.justinmind.com/blog/how-to-design-user-scenarios/> .

¹⁸ R. Therón, "Transparencias de Interacción Persona Ordenador," tema 4, Diseño centrado en el usuario, Universidad de Salamanca.

¹⁹ R. Therón, "Transparencias de Interacción Persona Ordenador," tema 4.2, User Journey Maps, Universidad de Salamanca.

²⁰ DIGITAL NATIVES Ltd., "A Beginner's Guide To User Journey vs User Flow" reference: <https://www.digitalnatives.hu/blog/user-journey-vs-user-flow/> .