



VNiVERSIDAD
D SALAMANCA

TRABAJO FIN DE GRADO DE INGENIERÍA INFORMÁTICA

APLICACIÓN PARA LA RECEPCIÓN,
ALMACENAMIENTO Y GESTIÓN DE
IMÁGENES DICOM EN EL SECTOR
SANITARIO

Autor:

Rubén Fraile Sanchón

Tutores:

Andrea Vázquez Ingelmo

Alicia García Holgado

Francisco José García Peñalvo

Enero 2023

CERTIFICADO DE LOS TUTORES

Dña. Andrea Vázquez Ingelmo, personal investigador de la Universidad de Salamanca, y Dña. Alicia García Holgado y D. Francisco José García Peñalvo, profesores/as del Departamento de informática y automática de la Universidad de Salamanca

CERTIFICAN:

Que el trabajo titulado “Aplicación para la recepción, almacenamiento y gestión de imágenes DICOM en el sector sanitario” ha sido realizado por D. Rubén Fraile Sanchón, con DNI 70904423T y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado de la Titulación Ingeniería Informática de esta Universidad.

Y para que así conste a todos los efectos oportunos.

En Salamanca, a 10 de Enero de 2023

Dña. Andrea Vázquez Ingelmo

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

AGRADECIMIENTOS

Estas palabras nacen de muchas horas de pensamiento y no ha sido tarea fácil ya que con ellas doy por finalizada una etapa muy dura a la par que bonita y especial en la que he aprendido mucho y he crecido como persona tanto en el ámbito personal como en el ámbito profesional, pero todo esto no hubiese sido posible sin la ayuda de todas aquellas personas que durante estos cuatro años y medio me han ido acompañando tanto dentro como fuera de la universidad.

Me gustaría agradecer en primer lugar a todos y cada uno de los profesores que he tenido el placer de conocer durante toda mi etapa educativa desde educación infantil hasta estos días ya que gracias a ellos ha sido posible llegar a escribir estas líneas.

A mi familia, por confiar en mí y transmitirme la alegría constante que me ha impulsado a seguir creyendo y a seguir creciendo. Cuando hablo de familia no hablo únicamente de hermanos, abuelos, padres sino también de esos amigos que nunca me han soltado de la mano aun sabiendo que hacía las cosas mal, esos amigos a los que no les importa darte todo lo que tienen y quedarse ellos sin nada y, por suerte, tengo a muchos de ellos que los siento como mi familia.

A mis compañeros de clase, por ser apoyos constantes y hacer que las clases, prácticas y exámenes se enfrentaran con una mayor tranquilidad y que estos cuatro años y medio hayan sido, si no la mejor, una de las mejores etapas de mi vida.

A mis padres por confiar tanto en mí, por tener la capacidad de aguantarme en momentos en los que ni yo mismo me aguantaba, por darme tanto cariño en los que más lo necesitaba y por ser siempre los primeros en estar ahí para darme un empujón para seguir creyendo. Vuestro carácter luchador me ha llevado a estar donde estoy.

A mis tutores de TFG por apoyarme en momentos en los que todo parecía ir cuesta abajo. Me habéis dado fuerzas y ánimos, confiando siempre en que lo iba a sacar.

Por último, con estas líneas me gustaría hacer mención a todas las personas que por una razón u otra ya sea buena o mala me han aportado para ser hoy la persona que soy.

A todos y cada uno de ustedes, simplemente gracias, os estaré eternamente agradecido.

No quisiera terminar sin antes destacar que este esfuerzo que he invertido realizando este proyecto ha sido dedicado para una persona que es muy especial para mí y, que lamentablemente, ya no está. Mi gran luchador, mi maestro. Siempre seré el ingeniero informático que soñaste papá, a partir de ahora todo lo que haga va a ser por y para ti.

Rubén Fraile Sanchón

Salamanca, 2022

RESUMEN

DICOM significa “Imágenes Digitales y Comunicaciones en Medicina” y fue desarrollado conjuntamente por la Asociación Nacional de Fabricante (NEMA) y el Colegio Americano de Radiología (ACR) para permitir la interoperabilidad entre equipos de imágenes con otros dispositivos. Este estándar es responsable de gobernar tanto el formato de imagen como los diversos protocolos de red necesarios para la transmisión de información de imágenes generadas durante las muchas “modalidades” de imágenes relacionadas con la atención médica, tales como resonancia magnética, medicina nuclear, tomografía computarizada y ultrasonidos. Por lo tanto, el estándar DICOM existe de una forma u otra desde 1983 y continúa evolucionando cada año.

En el marco del manejo y gestión de imágenes DICOM, este proyecto consiste en la ampliación de una aplicación ya existente como es *KoopaML*, aplicación web enmarcada en el contexto del Departamento de Cardiología del Hospital Universitario de Salamanca, cuyo objetivo es permitir que sus usuarios puedan entrenar sus propios modelos, analizar sus datos y realizar tareas sobre ellos sin la necesidad de tener conocimientos de programación. A esta herramienta se le suma la existencia de Cartier IA, plataforma de almacenamiento y visualización de datos e imágenes médicas usada también en el Departamento de Cardiología.

Esta ampliación se basa en la recepción, almacenamiento y gestión de imágenes DICOM en *KoopaML*. La recepción y almacenamiento se realiza a través de la transmisión de estos archivos a través de otros PACS (Sistema de Archivo y Comunicación de Imágenes) como puede ser Cartier IA, de esta manera se evita que los usuarios de ambos sistemas tengan que realizar unas acciones en un sistema y otras en el otro y así ahorrar tiempo.

En cuanto a la gestión de estas imágenes, se les pueden aplicar algoritmos de Inteligencia Artificial a éstas y, modificarlas utilizando diferentes herramientas de edición para medir, anotar, recortar, acercar, desplazar y segmentar entre otras.

Palabras clave: *DICOM*, Inteligencia Artificial, *Machine Learning*, PACS, algoritmos.

ABSTRACT

DICOM stands for “Digital Imaging and Communications in Medicine” and was developed jointly by the National Electrical Manufacturers Association (NEMA) as well as the American College of Radiology (ACR) to permit interoperability between imaging equipment as well as with other devices. This standard is responsible for governing both the image format as well as generated during the many healthcare-related imaging “modalities” such as magnetic resonance, nuclear medicine, computed tomography and ultrasounds. The DICOM standard has existed in one form or the other since 1983 and continues to evolve every year.

In the context of DICOM image handling and management, this project consists of the expansion of an existing application such as KoopaML, a web application framed in the context of the Cardiology Department of the University Hospital of Salamanca, whose objective is to allow its users to train their own models, analyze their data and perform tasks about them without the need to have programming knowledge. Added to this tool is the existence of Cartier IA, a storage and visualization platform for medical data and images also used in the Department of Cardiology.

This extension is based on the reception, storage, and management of DICOM images in KoopaML. The reception and storage are done through the transmission of these files through other PACS (Picture Archiving and Communication System) such as Cartier IA, in this way it is avoided that the users of both systems must carry out certain actions in one system have to carry out certain actions in one system and others in the other, this saving time.

Regarding the management of these images, Artificial Intelligence algorithms can be applied to them and modified using different editing tools to measure, annotate, crop, zoom, move, and segment, among others.

Key words: DICOM, Machine Learning, Artificial Intelligence, PACS, Algorithms.

TABLA DE CONTENIDO

1.	INTRODUCCIÓN	12
2.	OBJETIVOS DEL TRABAJO	16
2.1.	OBJETIVOS DEL SISTEMA	16
2.2.	OBJETIVOS PERSONALES	17
3.	IMÁGENES DICOM EN KOOPAML	19
3.1.	DESCRIPCIÓN DEL SISTEMA A AMPLIAR.....	19
3.2.	DESCRIPCIÓN DE LA AMPLIACIÓN DEL SISTEMA.....	21
4.	HERRAMIENTAS UTILIZADAS	24
4.1.	DISEÑO DE LA APLICACIÓN	24
4.1.1.	ADOBE XD	24
4.1.2.	DRAW.IO.....	25
4.2.	CÓDIGO DE PROGRAMACIÓN.....	25
4.2.1.	PYTHON	26
4.2.2.	DJANGO	26
4.2.3.	HTML, CSS Y JAVASCRIPT	27
4.2.4.	BOOTSTRAP	28
4.3.	ENTORNO DE DESARROLLO	29
4.3.1.	PYCHARM	29
4.3.2.	GITLAB	29
4.3.3.	POSTGRESQL.....	31
4.3.4.	SLACK.....	31
4.3.5.	GOOGLE DRIVE	32
4.4.	INTEGRACIÓN.....	33
4.4.1.	CARTIER-IA.....	33
5.	TÉCNICAS UTILIZADAS.....	34
5.1.	METODOLOGÍA.....	34
6.	ASPECTOS RELEVANTES DEL DESARROLLO	37
6.1.	PLANIFICACIÓN TEMPORAL.....	37
6.2.	ESPECIFICACIÓN DE REQUISITOS.....	40
6.2.1.	DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA	40
6.2.2.	CATÁLOGO DE REQUISITOS DEL SISTEMA	41

6.3.	PROCESO DE DISEÑO CENTRADO EN EL USUARIO	44
6.3.1.	<i>CICLO DE DISEÑO CENTRADO EN EL USUARIO</i>	45
6.3.2.	<i>FASE DE EXPLORACIÓN</i>	47
6.3.3.	<i>FASE DE CONCEPTUALIZACIÓN</i>	53
6.3.4.	<i>FASE DE PROTOTIPOS Y PRUEBAS DE USUARIOS</i>	57
6.4.	DESARROLLO DEL SISTEMA.....	64
6.4.1.	<i>INTEGRACIÓN EN KOOPAML</i>	64
6.4.2.	<i>INTEGRACIÓN CON CARTIER-IA</i>	64
6.4.3.	<i>GESTIÓN Y MANEJO DE IMÁGENES DICOM</i>	67
6.4.4.	<i>GESTIÓN DE ALGORITMOS DE INTELIGENCIA ARTIFICIAL</i>	70
6.5.	DESPLIEGUE DE LA APLICACIÓN.....	73
7.	CONCLUSIONES	75
7.1.	CONCLUSIONES	75
7.2.	POSIBLES LÍNEAS FUTURAS DE TRABAJO	76
8.	BIBLIOGRAFÍA	78

ÍNDICE DE FIGURAS

<i>Figura 1. Ejemplo de proyecto de Machine Learning con KoopaML</i>	20
<i>Figura 2. Muestra de estadísticas resumidas de un conjunto de datos de entrada</i>	20
<i>Figura 3. Pantalla principal de la herramienta KoopaML</i>	21
<i>Figura 4. Gestión de algoritmos de Inteligencia Artificial en la plataforma</i>	22
<i>Figura 5. Ejemplo de aplicación de algoritmo de Inteligencia Artificial a una imagen DICOM</i>	22
<i>Figura 6. Barra de herramientas de edición de la plataforma</i>	23
<i>Figura 7. Diseño digital mostrado en la aplicación AdobeXD</i>	24
<i>Figura 8. Espacio de realización de diagramas con Draw.io</i>	25
<i>Figura 9. Fragmento de código implementado en python</i>	26
<i>Figura 10. Fragmento de código implementado en HTML</i>	27
<i>Figura 11. Fragmento de código implementado en CSS</i>	27
<i>Figura 12. Fragmento de código implementado en JavaScript</i>	28
<i>Figura 13. Entorno de desarrollo del proyecto KoopaML</i>	29
<i>Figura 14. Código del proyecto en la aplicación Gitlab</i>	30
<i>Figura 15. Espacio de coordinación en Slack</i>	32
<i>Figura 16. Espacio de coordinación de Google Drive</i>	32
<i>Figura 17. Ejemplo de proyecto en CartierIA</i>	33
<i>Figura 18. Ciclo de desarrollo</i>	34
<i>Figura 19. Fases del proceso unificado</i>	36
<i>Figura 20. Planificación temporal del proyecto</i>	38
<i>Figura 21. Diagrama de actores del sistema</i>	40
<i>Figura 22. Paquete de análisis Gestión de algoritmos</i>	42
<i>Figura 23. Paquete de análisis Gestión de PACS</i>	42
<i>Figura 24. Paquete de análisis Gestión de imágenes DICOM</i>	43
<i>Figura 25. Ciclo de diseño del proceso centrado en el usuario</i>	46
<i>Figura 26. Usuario modelo número 1 (Almudena)</i>	50
<i>Figura 27. Usuario modelo número 2 (Carlos)</i>	51
<i>Figura 28. Sitemap del sistema</i>	54
<i>Figura 29. Wireframe primer prototipo pantalla inicial</i>	55

Figura 30. Wireframe segundo prototipo pantalla inicial	55
Figura 31. Wireframe visualización de imágenes dicom	57
Figura 32. Prototipo en papel: primera versión pantalla inicial.....	58
Figura 33. Prototipo en papel: Imágenes resultantes de una búsqueda en un PACS..	59
Figura 34. Prototipo en papel: Configuración de algoritmos de Inteligencia Artificial en la aplicación	59
Figura 35. Prototipo digital: Aplicación de algoritmo de Inteligencia Artificial a una imagen DICOM seleccionada	61
Figura 36. Prototipo digital: Pantalla inicial	61
Figura 37. Prototipo digital: Gestión de entidades PACS en el sistema.....	61
Figura 38. Fragmento de código para realizar la conexión entre JavaScript y Python	65
Figura 39. Fragmento de código para realizar la conexión entre KoopaML y PACS configurados (CartierIA).....	67
Figura 40. Esquema de comunicación entre sistemas	67
Figura 41. Detalles del fichero de una imagen DICOM.....	68
Figura 42. Herramientas de edición de imágenes DICOM.....	69
Figura 43. Aplicación de diferentes algoritmos a una imagen DICOM	71
Figura 44. Resultado de aplicar un algoritmo de Inteligencia Artificial a una imagen DICOM.....	71
Figura 45. Fragmento de código para la aplicación de un algoritmo de Inteligencia Artificial a una imagen DICOM	72
Figura 46. Requisitos de la aplicación a la hora de instalar	73

ÍNDICE DE TABLAS

<i>Tabla 1. Fechas estimadas según la planificación temporal.....</i>	<i>39</i>
---	-----------

1. INTRODUCCIÓN

El *Machine Learning* (ML) o aprendizaje automático [1] 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 Artificial [2] permite a contextos, como la identificación de patrones o la realización de predicciones. Sin la ayuda del *Machine Learning*, estas actividades consumirían demasiado más tiempo y recursos. Especialmente, la aplicación del *Machine Learning* [3] a los datos sanitarios ha llegado a demostrar su utilidad en tareas mucho más complejas como el diagnóstico, detección de enfermedades, segmentación y evaluación de las funciones de los órganos, entre otras.

Sin embargo, la aplicación del *Machine Learning* no es nada fácil. Si nos centramos en el ámbito sanitario ya que es el sector con el que está relacionado el proyecto, se plantea como es un sector demasiado sensible puesto que puede ser demasiado peligroso que los profesionales de la medicina no comprendieran de manera completa cómo utilizar un modelo o cómo analizar sus resultados.

Este concepto no consiste únicamente en aplicar un conjunto de funciones predefinidas por parte de bibliotecas externas [4]. Su aplicación necesita de un conocimiento profundo de los datos de entrada, las transformaciones que deben realizarse para ajustar las métricas de calidad antes de utilizar los modelos entrenados en producción [5]. De lo contrario, los resultados podrían llevar a conclusiones erróneas, pérdidas, discriminación e incluso llegar a negligencia [6].

En este punto es donde se avanzó con trabajos de desarrollo software previos. Concretamente con la plataforma gráfica llamada KoopaML, una aplicación web enmarcada en el Departamento de Cardiología del Hospital Universitario de Salamanca y que fue desarrollada en un Trabajo de Fin de Grado anterior [7]. Esta aplicación no solo se permitió la aplicación de funciones predefinidas de *Machine Learning* si no que ayuda a los beneficiarios del proyecto a comprender todos los pasos necesarios para llegar a construir modelos de *Machine Learning*.

Se equilibró el conocimiento del dominio de datos con la experiencia de *Machine Learning*, uniendo así la experiencia de los expertos en el contexto sanitario con la comprensión del dominio de datos por parte de los profesionales médicos. En este sentido, se proporcionó a los profesionales sanitarios herramientas para suplir esa ausencia de conocimiento en *Machine Learning* permitiéndoles implementar *pipelines* mientras se les facilita información teórica sobre lo que están trabajando o se les sugiere métodos de continuación.

KoopaML fue creada 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 *Machine Learning*.

Sin embargo, se detectaron posibles mejoras y validaciones de esta plataforma en base a las nuevas necesidades derivadas del enfoque centrado en el usuario seguido para su desarrollo. Durante la evaluación de esta plataforma, los usuarios detectaron algunas características que podrían incluirse en la plataforma de forma que se podría extender el uso de Inteligencia Artificial a las imágenes DICOM [8].

Este es el punto de partida del proyecto que se va a llevar a cabo como ampliación al existente. En concreto, el presente proyecto tiene como objetivo poder gestionar, almacenar y recibir este tipo de imágenes a través de la conexión con otras herramientas del ámbito sanitario enfocadas en el almacenar de dichas imágenes.

Cuando se habla de DICOM se habla de un estándar de atención médica responsable de regular casi todos los aspectos de las imágenes médicas como la transmisión, la interpretación y la gestión de imágenes, la gestión de procedimientos y el almacenamiento fuera de línea, tratándose así de un concepto utilizado en casi todas las “modalidades” de imágenes relacionadas con la atención médica, como puede ser la resonancia magnética, medicina nuclear, tomografía computarizada y los ultrasonidos. Casi todos los flujos de trabajo que tratan imágenes clínicas de todo el mundo se basan en el estándar DICOM.

A lo largo de este documento se mostrará el proceso de desarrollo de esta nueva funcionalidad. Para su elaboración se seguirá la guía de la plataforma *Diaweb* [9] y se detallarán los siguientes aspectos:

- **Objetivos del proyecto.** Se explicarán cuáles son los objetivos que se persiguen con la realización del proyecto.
- **Imágenes DICOM en KoopaML.** Se explicarán los principales cambios realizados en la herramienta, así como una explicación de la aplicación ya desarrollada.
- **Herramientas utilizadas.** Se expondrán las diferentes herramientas de trabajo que se han utilizado para llevar a cabo el proyecto.
- **Técnicas utilizadas.** Se expondrán las diferentes técnicas metodológicas que se han utilizado para llevar a cabo el proyecto
- **Aspectos relevantes del sistema.** Se presentarán los aspectos más interesantes surgidos a lo largo del desarrollo del sistema.
- **Conclusiones y líneas futuras de trabajo.** Se recogerán las diferentes conclusiones tomadas del proyecto y se indicará como se puede continuar trabajando en la línea del sistema realizado.

Además de este documento, se proporciona el código implementado y la documentación técnica que se compone de:

- **Anexo I. Especificación de Requisitos Software:** En este documento se identifican se especifica la información sobre los requisitos software del sistema que se iba a desarrollar.
- **Anexo II. Plan de Proyecto Software:** En este documento se realiza una estimación del esfuerzo y de la planificación temporal para la realización del proyecto, es decir, se expone el proceso detallado de ingeniería.
- **Anexo III. Análisis del Sistema Software:** Este documento contiene la información obtenida sobre el análisis, el refinamiento y la estructuración realizada sobre el proyecto.

- **Anexo IV. Proceso de Diseño Centrado en el Usuario:** Este documento expone el proceso detallado de diseño centrado en el usuario, acompañado de los recursos utilizados y requeridos para llegar a comprender el desarrollo y evolución del proceso.
- **Anexo V. Documentación del Código Fuente:** Este documento expone la documentación técnica del código que permite facilitar la comprensión del código implementado. Por otro lado, presentará los aspectos relevantes del código fuente de la herramienta.
- **Anexo VI. Manual del Usuario:** Este documento expone una guía para el usuario de la interacción con el sistema conjunto un vídeo explicativo a modo de tutorial para el correcto manejo de la aplicación para los futuros usuarios.

2. OBJETIVOS DEL TRABAJO

En este apartado se expondrán los objetivos que se persiguen con la realización del proyecto. En primer lugar, se describirán los objetivos marcados por los requisitos del *software* y, en segundo lugar, los objetivos personales propios que se han querido alcanzar con este proyecto.

2.1. OBJETIVOS DEL SISTEMA

El objetivo principal que persigue este proyecto es la creación de una aplicación web que permita la recepción, el almacenamiento y la gestión de imágenes DICOM en el sector sanitario. La aplicación ofrecerá soporte a usuarios que no poseen conocimientos técnicos o esos conocimientos son muy básicos sobre *Machine Learning*, ya que su funcionamiento se basa en almacenar las imágenes DICOM y, su posterior aplicación de diferentes algoritmos a estas.

Otro objetivo derivado del objetivo principal es la gestión de los propios proyectos DICOM creados, de forma que al usuario se le permita la creación de nuevos proyectos, descargar los ya existentes, editarlos, configurarlos y eliminarlos.

Un tercer objetivo derivado del objetivo principal es la gestión de los algoritmos de *Machine Learning* y, su posterior aplicación en imágenes DICOM. Partiendo de que este proyecto no se centra en la creación de esos algoritmos, si no que únicamente permitirá almacenarlos, así como sus diferentes datos relacionados con ellos. Estos únicamente podrán ser almacenados por los usuarios de la aplicación que tengan el rol de administrador o Experto en *Machine Learning* ya que serán las personas que tengan conocimientos sobre ello y, por otra parte, cualquier usuario podrá aplicarlo en las imágenes DICOM.

Un cuarto objetivo es la integración entre varias aplicaciones como es el caso de KoopaML y Cartier-IA ya que serán las utilizadas en este proyecto. Estas herramientas deberán conectarse y comunicarse mediante el paso de imágenes DICOM entre ellas llegando a almacenarse finalmente en KoopaML.

Además, el diseño de la aplicación ha de estar centrado en el usuario, realizando todo el proceso necesario para poder ajustarlo a las distintas necesidades de la audiencia objetivo.

Finalmente, es necesario que la aplicación sea escalable y flexible, permitiendo así añadir múltiples y diversas actividades a las ya existentes.

2.2. OBJETIVOS PERSONALES

El objetivo personal principal que se ha buscado con este proyecto ha sido la superación exitosa de la realización de un sistema con un gran tamaño sobre el que se puede llegar a plasmar cualquier tipo de idea sin limitación de ningún tipo de enunciado ni requisitos expuestos por clientes. Para el autor del proyecto supone un reto personal que, junto con el afán de superación, implica la obtención de una mayor confianza en sí mismo.

Se busca marcar huella en el desarrollo del proyecto y una futura implementación real, ya no solo en el Departamento de Cardiología del Hospital Universitario de Salamanca, si no en todos aquellos lugares en los que se determine útil la utilización de esta herramienta.

Otro objetivo derivado del objetivo principal es la familiarización con herramientas y tecnologías desconocidas hasta el momento relacionadas con la salud y el bienestar de las personas.

Así mismo, se espera obtener una familiarización con lenguajes y herramientas que no habían sido utilizados anteriormente y que serían útiles para un futuro en el mercado laboral. Otro de los objetivos personales que se tiene es el desarrollo de competencias orales a la hora de defender, presentar y vender un proyecto.

Por otra parte, se intentará poner en práctica todos los conocimientos adquiridos durante el grado tanto en las asignaturas de Ingeniería del Software como en la asignatura de Interacción Persona-Ordenador, no dejando de lado los conocimientos que se han obtenido de las demás asignaturas.

En último lugar, se pretende conseguir una capacidad mejorada de adaptación realizando el proyecto de una manera general y no a partes como se habían realizado hasta ahora todos los proyectos.

3. IMÁGENES DICOM EN KOOPAML

3.1. DESCRIPCIÓN DEL SISTEMA A AMPLIAR

En primer lugar, se describirá el sistema actual. Se está trabajando sobre una aplicación ya existente por lo que se ha tenido que dar bastante importancia a este apartado ya que se ve muy necesario detallar cómo era la aplicación existente y que pasos se han llevado a cabo para añadir las funcionalidades nuevas.

KoopAML se trata de una aplicación ya existente 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 *Machine Learning*.

Esta primera versión de la aplicación proporcionó un conjunto de funcionalidades relacionadas con la definición y ejecución de nodos de *Machine Learning*, validación de datos de entrada, visualización y la posibilidad de definir heurísticas para llegar a impulsar y facilitar el desarrollo de pipelines de *Machine Learning*.

Para crear una nueva canalización, el sistema proporciona un espacio de trabajo vacío con una barra de herramientas que contiene las tareas potenciales que se pueden incluir en el flujo de trabajo de *Machine Learning*. Estas tareas se encuentran conectadas por enchufes relacionados con sus entradas y salidas. Las conexiones deben de ser compatibles para que el pipeline llegue a funcionar, es decir, si una tarea requiere de un conjunto de datos como entrada, únicamente se podrán conectar a ella aquellas tareas que generan como salida un conjunto de datos.

La *Figura 1* muestra uno de los diferentes *pipelines* de *Machine Learning* en la que se entrena un clasificador de bosque aleatorio con un conjunto de datos de entrada con 6 variables.

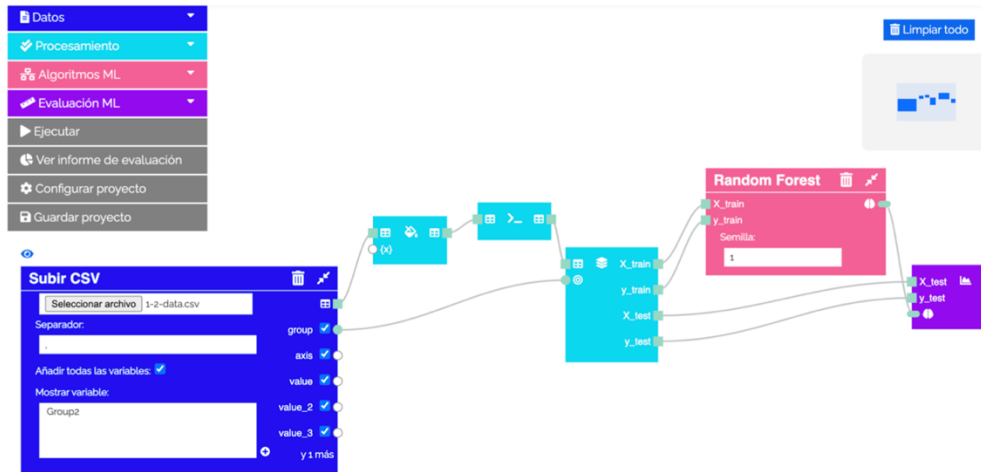


FIGURA 1. EJEMPLO DE PROYECTO DE MACHINE LEARNING CON KOOPAML

Una vez ejecutada el *pipeline*, los resultados intermedios de cada uno de los pasos se pueden consultar individualmente, lo que permite una mejor trazabilidad de los errores (en el caso de que los haya) y otros problemas relacionados con los modelos resultantes.

KoopaML también brinda la oportunidad de explorar visualmente el conjunto de datos de entrada y los resultados de la evaluación de modelos como se puede ver en la *Figura 2*. Esta función aprovecha la generación automática de paneles de información a través de ingeniería del dominio y metamodelo [10] [11] [12] [13].

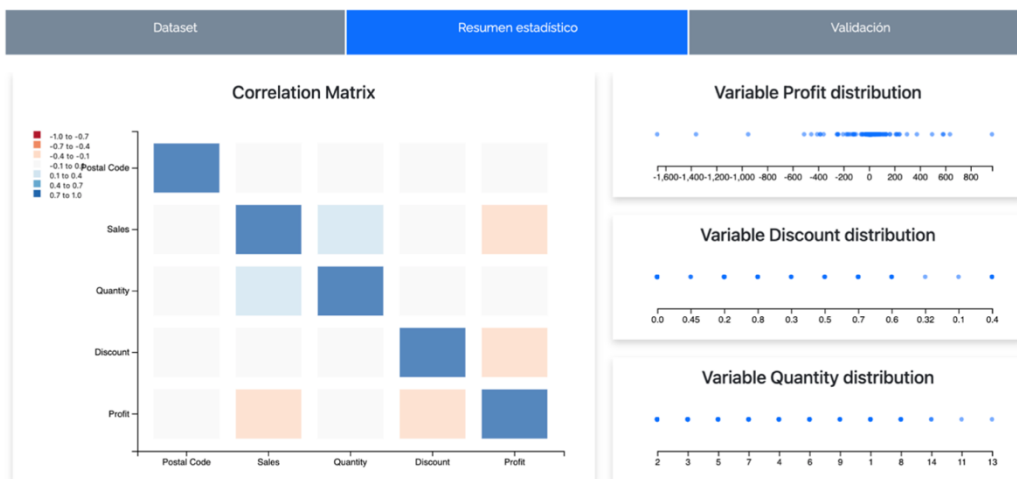


FIGURA 2. MUESTRA DE ESTADÍSTICAS RESUMIDAS DE UN CONJUNTO DE DATOS DE ENTRADA

Aunque la aplicación ya se veía bastante completa, a la hora de evaluarla por profesionales, los usuarios detectaron algunas características que podrían incluirse en la plataforma para obtener así una herramienta mucho más completa y con más funcionalidades. Una de las características que se comentaron y que se podrían añadir fue la de extender la aplicación de la Inteligencia Artificial a las imágenes DICOM [8], característica en la que se centra este Trabajo de Fin de Grado.

3.2. DESCRIPCIÓN DE LA AMPLIACIÓN DEL SISTEMA

En cuanto a la ampliación de la aplicación, el desarrollador se centró en el soporte de imágenes DICOM como datos de entrada para algoritmos de Inteligencia Artificial [14].

En este punto, se ideó un nuevo espacio de trabajo, separando el espacio de trabajo de *Machine Learning* del espacio de trabajo de proyectos DICOM y siendo un punto crucial para evitar una interfaz abarrotada. Además, las diferencias sustanciales entre la gestión de datos estructurados y los datos DICOM requerían tratamientos diferentes para estos dos tipos de entradas.

Por estos motivos, como se puede observar en la Figura, al crear un nuevo proyecto, los usuarios podrán elegir entre crear un proyecto de *Machine Learning* o un proyecto DICOM. En este caso, es necesario agregar una sección de configuración para poder gestionar los sistemas de archivo y comunicación de imágenes (PACS) para permitir que la aplicación KoopaML solicite imágenes de los servidores configurados (*Figura 3*).



FIGURA 3. PANTALLA PRINCIPAL DE LA HERRAMIENTA KOOPAML

Cuando se crea un nuevo proyecto DICOM, se les mostrará a los usuarios las imágenes que ha almacenado en ese proyecto y, a las que puede aplicar los diferentes algoritmos de *Machine Learning* que se encuentran en el sistema, estos algoritmos podrán ser almacenados por los administradores de la aplicación y, como se puede ver en la *Figura 4*, se le pide al usuario una serie de características como ficheros de pesos para poder aplicar los algoritmos a las imágenes DICOM.

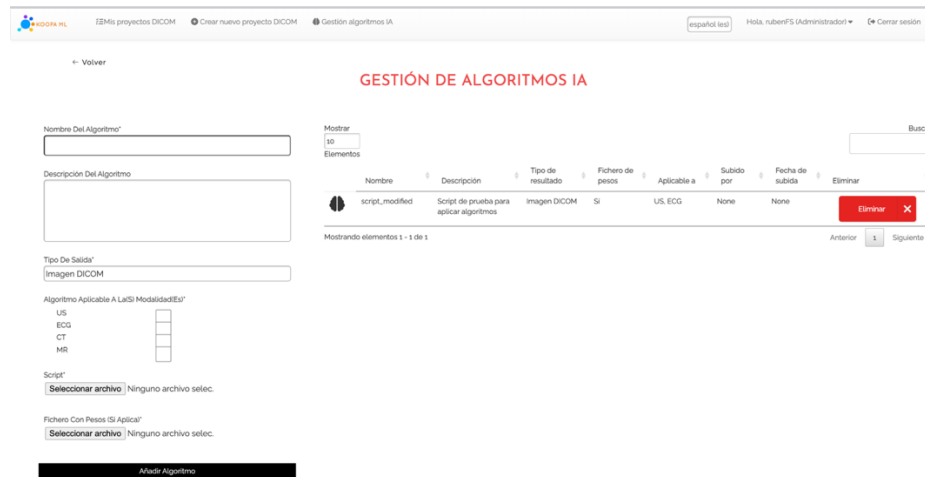


FIGURA 4. GESTIÓN DE ALGORITMOS DE INTELIGENCIA ARTIFICIAL EN LA PLATAFORMA

Estos algoritmos son cargados en la plataforma por expertos en Inteligencia Artificial autorizados, haciéndolos así accesibles a usuarios no expertos. En la *Figura 5* se puede ver la aplicación de un algoritmo a una imagen seleccionada.

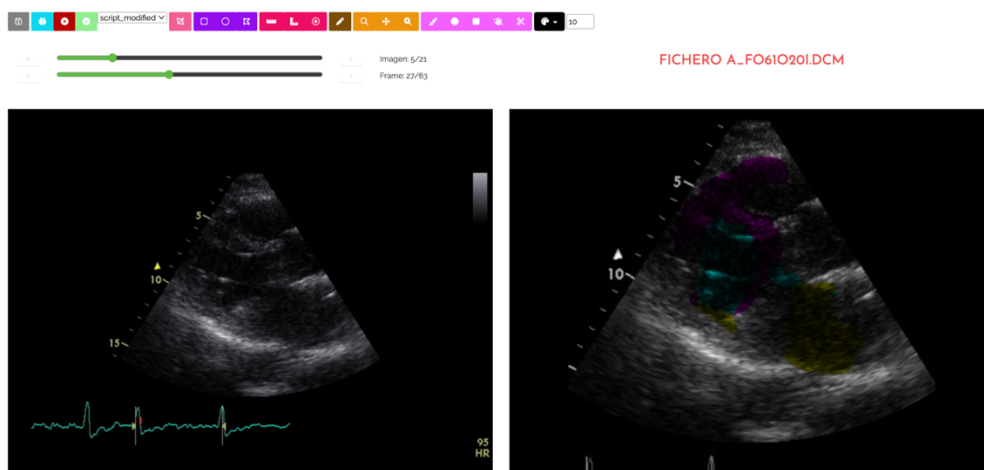


FIGURA 5. EJEMPLO DE APLICACIÓN DE ALGORITMO DE INTELIGENCIA ARTIFICIAL A UNA IMAGEN DICOM

Además, el espacio de trabajo DICOM también permite a los usuarios modificar imágenes utilizando diferentes herramientas de edición para medir, anotar, recortar, acercar, desplazar y segmentar, entre otras (*Figura 6*).



FIGURA 6. BARRA DE HERRAMIENTAS DE EDICIÓN DE LA PLATAFORMA

En el siguiente enlace se podrá observar la aplicación realizada

<https://koopam1.grial.eu/>

Por otra parte, si se desea realizar un seguimiento más a fondo de la aplicación, en el [Anexo VI. Manual del Usuario](#) adjunto con este documento se expone un tutorial grabado para que los usuarios tengan más facilidad en el manejo de la aplicación, así como un manual escrito de las diferentes acciones que se pueden realizar en la aplicación.

4. HERRAMIENTAS UTILIZADAS

En este apartado se exponen las principales técnicas y herramientas que han sido utilizadas para la creación y realización de este proyecto, así como las razones por las que han sido seleccionadas.

4.1. DISEÑO DE LA APLICACIÓN

4.1.1. ADOBE XD

Se utiliza esta aplicación, *Adobe XD*, para la realización de todas las versiones de los prototipos digitales de la aplicación. Esta aplicación fue elegida debido a que, gracias a su uso previo, se habían adquirido conocimiento en cómo manejar dicha herramienta. Además, resulta bastante sencillo tanto a nivel de diseño para crear las diferentes pantallas, cómo a la hora de conectarlas y simular la interacción entre los diferentes elementos que forman la aplicación.

Por otra parte, esta herramienta ofrece la posibilidad de compartir el prototipo mediante la generación de un enlace a través del cual, quien realice la prueba de usuario, podrá exponer sus comentarios de una manera sencilla (*Figura 7*).

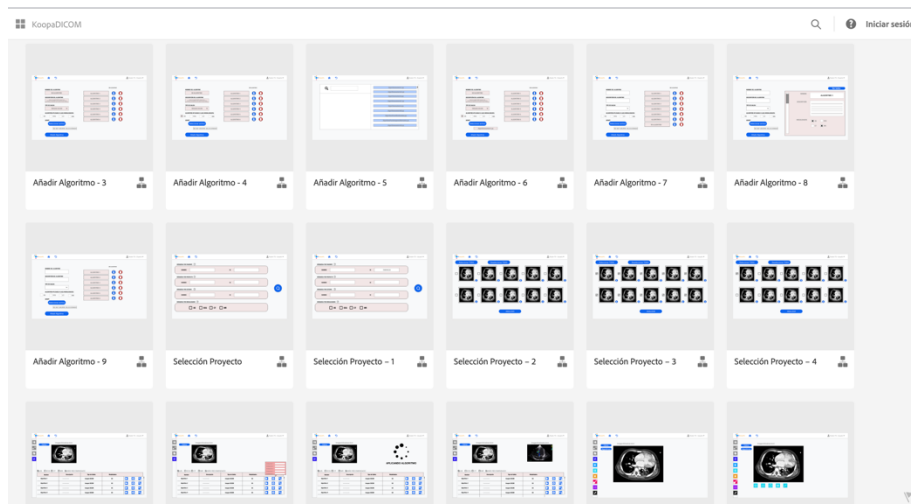


FIGURA 7. DISEÑO DIGITAL MOSTRADO EN LA APLICACIÓN ADOBEXD

4.1.2. DRAW.IO

Para realizar los diferentes diagramas del proceso de elicitación y especificación de requisitos *software*, se ha utilizado el *software* de dibujo de gráficos *Draw.io*. Fue escogida debido a que, gracias a un uso anterior, se estaba familiarizado con la herramienta (*Figura 8*).

Por otra parte, esta herramienta está conectada a *Google Drive*, lo que facilita el almacenamiento de los diagramas realizados y se integra con la cuenta institucional de la Universidad de Salamanca.

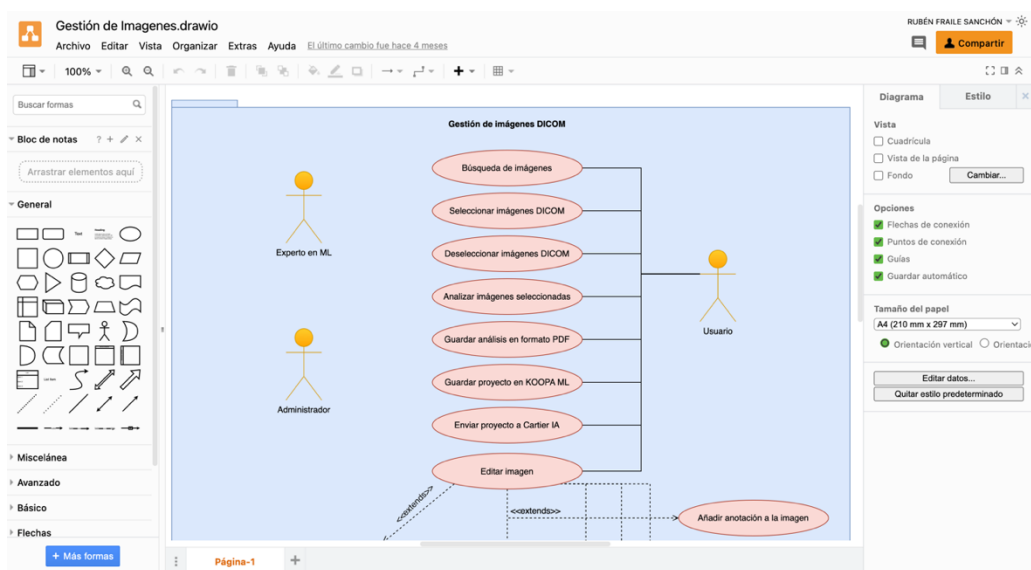


FIGURA 8. ESPACIO DE REALIZACIÓN DE DIAGRAMAS CON DRAW.IO

4.2. CÓDIGO DE PROGRAMACIÓN

Las decisiones de desarrollo, en cuanto a lenguaje de programación y *frameworks*, han venido dadas por la herramienta KoopaML en la cual se integra el presente proyecto.

4.2.1. PYTHON

Para el desarrollo de la aplicación se ha escogido el lenguaje de programación *Python*, este lenguaje tiene una sintaxis sencilla, muy similar al pseudocódigo siendo este de código abierto, multiplataforma y multiparadigma [15].

Además de las características ventajosas mencionadas, este lenguaje cuenta con numerosas librerías de matemáticas y ciencias de datos muy útiles para la funcionalidad de aprendizaje automático, como *NumPy* y *ScyPi* [16] (Figura 9).

```
@login_required(login_url='/login/')
def algorithm_registration(request):...

@login_required(login_url='/login/')
def algorithm_delete(request, algorithm_id):
    if request.user.is_authenticated:
        request.user = request.user
        user_role = request.user.user_role

        if request.method == 'POST':
            algorithm = Algorithm.objects.get(id=algorithm_id)

            try:
                os.remove(algorithm.script_path)
            except Exception as e:
                print(e)

            if algorithm.weights_path is not None:
                try:
                    os.remove(algorithm.weights_path)
                except Exception as e:
                    print(e)

            try:
                os.rmdir(os.path.join(BASE_DIR, 'algorithms_management/scripts/{0}'.format(algorithm.name)))
            except Exception as e:
                print(e)

            algorithm.delete()

        return redirect('/configuration-algorithms')
    else:
        return redirect('/configuration-algorithms')
else:
    return redirect('/configuration-algorithms')
```

FIGURA 9. FRAGMENTO DE CÓDIGO IMPLEMENTADO EN PYTHON

4.2.2. DJANGO

Se ha utilizado el *framework* web *Django*, ya que permite desarrollar de manera rápida y segura sitios web.

Además, ofrece gran parte de las funcionalidades básicas que un sitio web puede llegar a requerir, evitando así que el programador tenga que implementarlas. *Django* permite que las aplicaciones sean escalables gracias a su filosofía de bajo acoplamiento [17], pues cada una de las partes de la arquitectura de una aplicación va a ser independiente al resto, lo que llega a facilitar su modificación.

4.2.3. HTML, CSS Y JAVASCRIPT

Para el desarrollo del *front-end* se han utilizado los lenguajes de programación HTML, CSS y JavaScript.

El lenguaje de programación HTML (*Figura 10*) se ha usado para la definición de la estructura de la página, complementado con la definición de estilos que ofrece el lenguaje de programación CSS (*Figura 11*), separando de esta forma el contenido de la presentación, lo que permite obtener códigos mucho más limpios y sencillos.

```

<div class="col-md-5">
  <br>
  <div>
    <b>{% trans 'Nombre del proyecto' %}</b>
    <i class="fas fa-question-circle" data-toggle="tooltip"
      data-placement="top"
      title='{% trans "Introduzca un nombre de proyecto registrado en el PACS configurado" %}'></i>
  </div>
  <div style="...">
    <input type="hidden" name="search_param" value="all" id="search_param">
    <input type="text" class="form-control" name="search_name_project" id="search_name_project"
      placeholder='{% trans 'Nombre del proyecto' %}'>
  </div>
  <br>
  <div>
    <b>{% trans 'Código del proyecto' %}</b>
    <i class="fas fa-question-circle" data-toggle="tooltip"
      data-placement="top"
      title='{% trans "Introduzca un ID de proyecto registrado en el PACS configurado" %}'></i>
  </div>
  <div style="...">
    <input type="hidden" name="search_param" value="all" id="search_param">
    <input type="text" class="form-control" name="search_id_project" id="search_id_project"
      placeholder='{% trans 'Codigo del proyecto' %}'>
  </div>
</div>

```

FIGURA 10. FRAGMENTO DE CÓDIGO IMPLEMENTADO EN HTML

```

html,
body {
  height: 100%;
}

body {
  height: 100%;
  display: -ms-flexbox;
  display: -webkit-box;
  display: flex;
  -ms-flex-align: center;
  -ms-flex-pack: center;
  -webkit-box-align: center;
  align-items: center;
  -webkit-box-pack: center;
  justify-content: center;
  padding-top: 40px;
  padding-bottom: 40px;
}

.pacs-form {
  width: 100%;
  max-width: 330px;
  padding: 15px;
  margin: 0 auto;
}

.pacs-form .checkbox {
  font-weight: 400;
}

```

FIGURA 11. FRAGMENTO DE CÓDIGO IMPLEMENTADO EN CSS

Para añadir a la página atributos interactivos, como pueden ser la aparición y desaparición de iconos los cambios en el aspecto del menú, la creación de nodos de actividades o la aparición de los resultados de haber aplicado algoritmos a las imágenes se ha utilizado el lenguaje de programación JavaScript (*Figura 12*). JavaScript se trata de un lenguaje de programación que puede aplicarse a un documento HTML para ofrecerle el dinamismo que se ha mencionado anteriormente.

```
$.post({
  url: "/post-info-images/",
  beforeSend: function (xhr, settings) {
    xhr.setRequestHeader("X-CSRFToken", '{{ csrf_token }}')
  },
  data: jsonData,
  success: function(response) {
    console.log(response)
    if (response == 0) {
      let string = "Error, no se han almacenado imágenes en el proyecto presente." +
        "Vuelva a probar con otros datos"
      $('#myModal').toggle();
      $('#modal-message').text(string);
    } else{
      let string = "Se han almacenado un total de " + response + " imágenes en el proyecto presente"
      $('#myModal').toggle();
      $('#modal-message').text(string);
    }
  }
});
```

FIGURA 12. FRAGMENTO DE CÓDIGO IMPLEMENTADO EN JAVASCRIPT

4.2.4. BOOTSTRAP

Para lograr el desarrollo de un diseño web adaptativo (*responsive design*) se ha utilizado la herramienta *Bootstrap*, que permite realizar este tipo de diseños a través de componentes reutilizables y estructuras predefinidas como son las filas, las columnas o las tablas, además de ser una herramienta muy flexible pues permite la personalización de todas las estructuras y componentes [19].

4.3. ENTORNO DE DESARROLLO

4.3.1. PYCHARM

En cuanto al entorno de desarrollo se utiliza *PyCharm*, IDE especializado en el lenguaje de programación *Python*. Se escoge este entorno debido a sus grandes características de finalización de código inteligente, inspecciones de código, errores automáticos con diversas soluciones sugeridas y funcionalidades de navegación muy completas (*Figura 13*).

En particular, se ha utilizado la versión profesional para estudiantes.

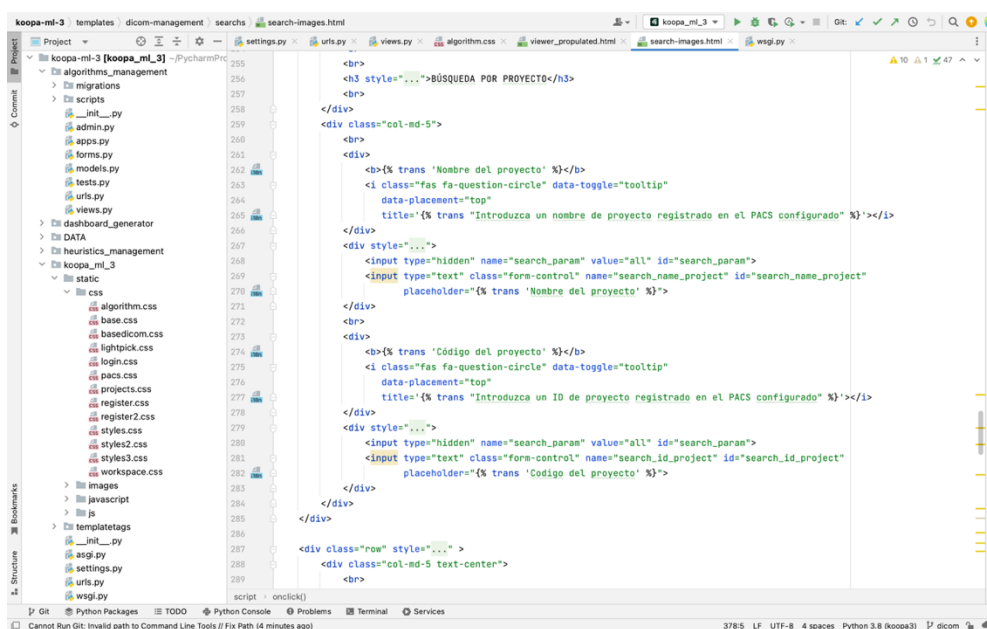


FIGURA 13. ENTORNO DE DESARROLLO DEL PROYECTO KOOPAML

4.3.2. GITLAB

Para la gestión de versiones del código de la aplicación se ha utilizado *Gitlab*, siendo este un servicio de control de versiones y desarrollo de *software* colaborativo basado en Git [20], de forma que todos los miembros del equipo tuvieran acceso al historial de las distintas versiones del código en cualquier momento del proyecto.

En particular, se ha utilizado una instalación del Grupo de Investigación en Interacción y eLearning (GRIAL) en el que se ha desarrollado el Trabajo de Fin de Grado.

Para integrar de manera eficiente este sistema en el desarrollo se realizan periódicamente las acciones “commit” y “push”, de forma que el resto de los miembros del equipo puedan revisar la versión más actualizada posible.

Otro aspecto que se tuvo en cuenta a la hora de trabajar con Git fue el desarrollo de código utilizando ramas (*branches*), de forma que, cuando se tiene una versión estable de la aplicación además de realizar las acciones “commit” y “push”, se crea una rama dedicada a la funcionalidad a desarrollar, de forma que la versión estable no sea modificada inintencionadamente. Una vez comprobado el correcto funcionamiento de esta nueva funcionalidad se fusiona con la rama de la última versión estable y se repite el proceso cada vez que haya una nueva funcionalidad.

El uso de esta herramienta también ha facilitado el despliegue de esta en el entorno de desarrollo y, posteriormente, en el entorno de producción (*Figura 14*).

Si se quisiese visualizar el código realizado, se podrá hacer en el siguiente enlace que se corresponde con el repositorio del código en Gitlab [21]:

<https://gitlab.grial.eu/AndreaVazquez/koopa-ml-dev>

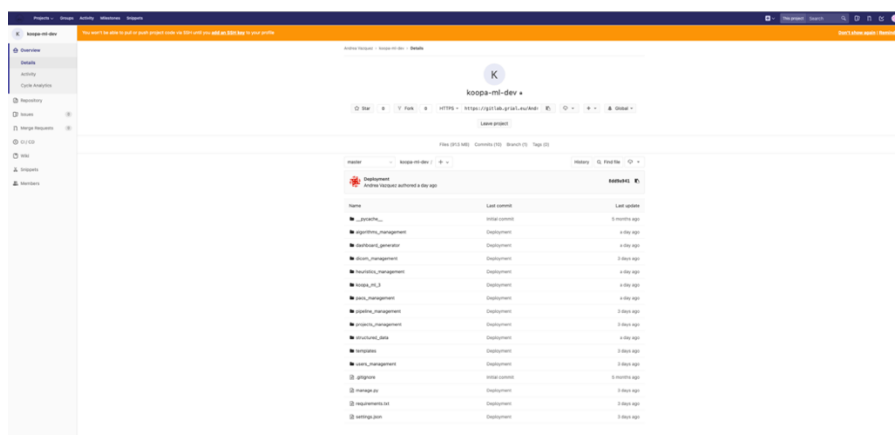


FIGURA 14. CÓDIGO DEL PROYECTO EN LA APLICACIÓN GITLAB

4.3.3. POSTGRESQL

En primer lugar, se ha utilizado *Postgres* [22] como base de datos SQL. Se ha hecho uso de una base de datos estructurada por varias razones:

- Gran variabilidad de datos presentes en la aplicación, ya que si se utilizará una base de datos estructurada existirían una gran cantidad de tablas con demasiados campos vacíos.
- La aplicación ofrece un alto grado de flexibilidad permitiendo al usuario añadir nuevas tareas de distinto carácter a la base de datos, el resultado de esto es el almacenamiento de diversas estructuras de datos de distinta naturaleza.

Por estas razones, se ve importante recalcar que se necesita hacer uso de una base de datos estructurada para enfrentar con éxito la variabilidad de tipos y estructuras de datos y, lograr así, ofrecer un alto grado de flexibilidad al usuario.

Esta herramienta se utiliza como sistema gestor de la base de datos. Se trata de una herramienta de código libre considerada en la actualidad como uno de los más robustos y potentes.

Se escogió esta base de datos estructurada debido a su fácil instalación que permite un sistema multiplataforma. Por otra parte, 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.

4.3.4. SLACK

Para realizar el seguimiento del proceso del Trabajo de Fin de Grado se ha utilizado *Slack* (Figura 15). La comunicación con los tutores se ha desarrollado a través de esta herramienta, de forma que se ha podido realizar un seguimiento asíncrono combinado con sesiones síncronas y reuniones presenciales que han permitido trabajar en un contexto real.

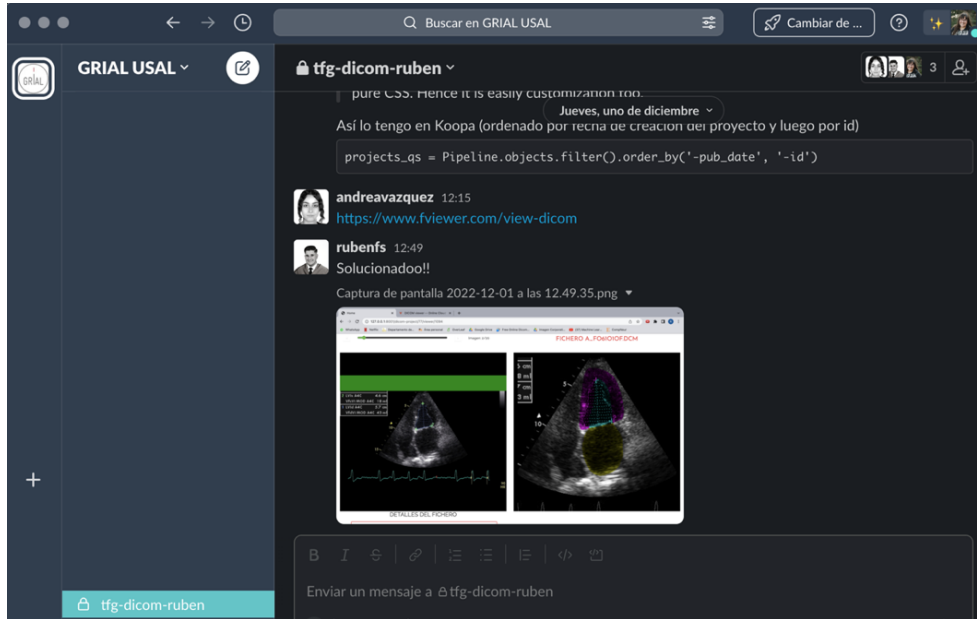


FIGURA 15. ESPACIO DE COORDINACIÓN EN SLACK

4.3.5. GOOGLE DRIVE

Para poder gestionar las diferentes versiones de los informes y documentación general se ha utilizado una carpeta compartida de esta herramienta a la que tenían acceso todas las personas implicadas en el desarrollo de esta aplicación (Figura 16).

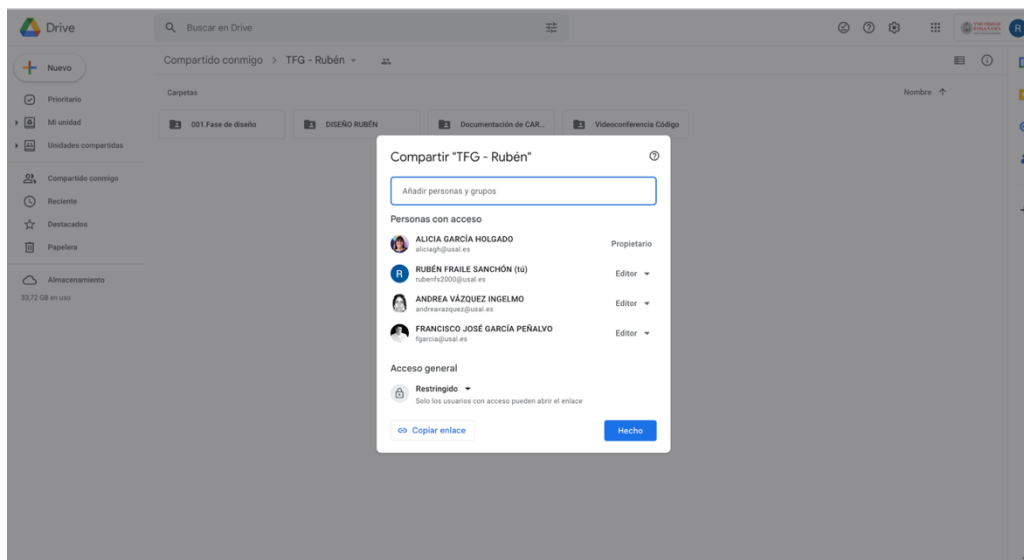


FIGURA 16. ESPACIO DE COORDINACIÓN DE GOOGLE DRIVE

4.4. INTEGRACIÓN

4.4.1. CARTIER-IA

Para poder realizar el almacenamiento de las imágenes DICOM en la aplicación KoopaML se han utilizado sistemas de archivo y comunicación de imágenes (PACS) como es el caso de Cartier-IA (*Figura 17*) [23].

Esta herramienta se trata de una plataforma para la gestión de datos e imágenes médicas. El objetivo de este proyecto se centra en proporcionar una interfaz sencilla para poder organizar los datos estructurados, visualizar y editar imágenes médicas, así como aplicar algoritmos de Inteligencia Artificial sobre los recursos almacenados.

The screenshot displays the CARTIER-IA web application interface. At the top, there is a navigation bar with the application name 'CARTIER-IA' and several menu items: 'Proyectos', 'Instituciones', 'Usuarios', and 'Scripts'. On the right side of the navigation bar, there is a language selector set to 'español (es)', a user greeting 'Hola, ruben (Administrador)', and a 'Cerrar sesión' button.

The main content area is titled 'Proyecto cartier'. It features a toolbar with icons for 'Plantillas', 'Identificadores', 'Datos estructurados', 'Imágenes', 'Validación', and 'PACS'. Below the toolbar, the interface is divided into several sections:

- Datos generales:** This section lists project details: 'Código del proyecto: cartier', 'Nombre: cartier', 'Descripción: cartier', and 'Investigador Principal: ruben (None)'. It also shows '1 instituciones asociadas al proyecto:' with a list containing 'Hospital de Salamanca' and '1 usuarios asociados al proyecto:' with a list containing 'ruben (None)'. There are input fields for these lists.
- Variables específicas:** This section contains buttons for 'Añadir tabla', 'Descargar esquema de variables', 'Añadir variables', and 'Gestionar'. Below these buttons, there are three columns representing different levels: 'A nivel de paciente', 'A nivel de estudio', and 'A nivel de fichero'. Each column shows 'Número de tablas: 0' and 'Número de variables: 0'.
- Filtros:** At the bottom left, there is a section for filters. It includes a checkbox for 'Mostrar únicamente pacientes con estudios de imágenes asociados', three filter categories: 'Filtros específicos de paciente', 'Filtros específicos de estudio', and 'Filtros específicos de fichero', a 'Filtrar' button, and an 'Acceder al visor' button.

FIGURA 17. EJEMPLO DE PROYECTO EN CARTIERIA

5. TÉCNICAS UTILIZADAS

5.1. METODOLOGÍA

Para el desarrollo del proyecto se ha seguido el modelo de desarrollo del Proceso Unificado [24] con un enfoque ágil ya que se han ido tomando las decisiones según se ha ido avanzando a la vez que se tenían reuniones periódicas de seguimiento. Esta metodología es considerada como la metodología estándar más utilizada para la realización del análisis, la implementación y la documentación de sistemas orientados a objetos.

Se decidió usar este método debido a su capacidad de adaptación a la mayoría de los sistemas *software* y a las necesidades de cada organización [25].

Algunas de las características que presenta esta metodología son:

- Es un proceso conducido por casos de uso, siendo un caso de uso la descripción de una acción o una actividad.
- Se trata de un proceso incremental e iterativo.
- Está basado en componentes.
- Está centrado en la arquitectura.
- Hace uso de UML [26] como lenguaje de representación visual.
- Está preparado para el desarrollo de grandes proyectos.

El Proceso Unificado se trata de un proceso repetitivo que se reproduce a lo largo de ciclos de desarrollo denominados iteraciones. Estos ciclos constituyen la vida de un sistema llegando a concluir con una versión final del producto.

La *Figura 18* muestra cada uno de los ciclos, que, cada uno de ellos, consta de cuatro fases:



FIGURA 18. CICLO DE DESARROLLO

- **Inicio.** En esta fase se desarrolla una descripción del producto final, presentándose así el análisis del negocio. En esta fase también se identifican y priorizan los riesgos más importantes teniendo como objetivo el ayudar al equipo de proyecto a decidir cuáles son los verdaderos objetivos del proyecto.
- **Elaboración.** Durante esta fase se especifican en detalle la mayoría de los casos de uso del producto y se diseña la arquitectura. Esta fase tiene como resultado la línea base de la arquitectura.
- **Construcción.** Durante esta fase se crea el producto. La línea base de la arquitectura crece hasta llegar a convertirse en el sistema completo. Al finalizar esta fase, el producto va a contener todos los casos de uso implementados, aunque puede tener algún defecto.
- **Transición.** Esta fase cubre el período durante el cual el producto se convierte en la versión beta. Durante esta fase, todas las diferentes iteraciones agregan características al sistema software.

Las fases de inicio y de elaboración pertenecen a la etapa de Ingeniería del Software, y, en cuanto a las fases de construcción y transición pertenecen a la etapa de Producción.

Cada una de las fases anteriores debe de terminar en hito siendo este un punto de control en el que los participantes del proyecto van a revisar el progreso de éste. Así mismo, dentro de cada una de las diferentes fases se encuentran disciplinas que agrupan tareas similares. Estas disciplinas son:

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

- Pruebas

Cada disciplina está asociada con un conjunto de modelos que se desarrollan. Estos modelos están compuestos por artefactos. Los artefactos más importantes son los modelos que cada disciplina realiza, estos son:

- Modelo de casos de uso
- Modelo de diseño
- Modelo de implementación
- Modelo de pruebas

Para llevar a cabo lo ya comentado, se ha utilizado el esquema mostrado en la *Figura 19*:

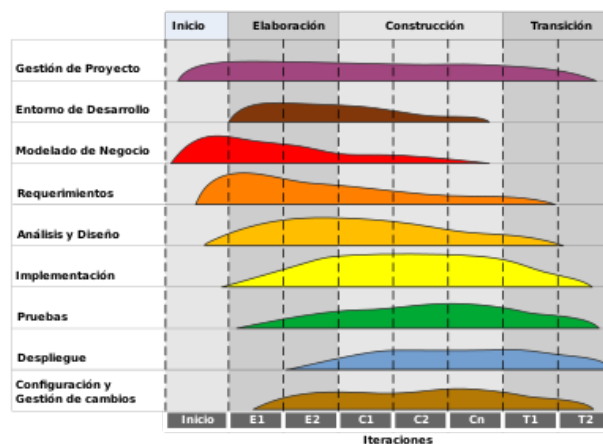


FIGURA 19. FASES DEL PROCESO UNIFICADO

Por otra parte, en cuanto a la especificación de los diferentes requisitos se ha seguido el método de Durán y Bernárdez siendo los requisitos expuestos mediante métodos gráficos, formales o textuales, según como se ha ido viendo oportuno en cada momento.

Esta metodología fue escogida debido a que en ocasiones anteriores ya fue utilizada y se poseía un dominio sobre ella mucho mayor que el de otras metodologías conocidas, pero no utilizadas.

En el apartado de aspectos relevantes, se comentarán más a fondo algunos conceptos de la especificación de los requisitos.

6. ASPECTOS RELEVANTES DEL DESARROLLO

En este apartado se expondrán los aspectos más relevantes e interesantes surgidos a lo largo de todo el proceso de desarrollo del proyecto.

6.1. PLANIFICACIÓN TEMPORAL

En este apartado se va a detallar la distribución que se ha seguido en base al Proceso Unificado, que se consigue repitiendo a lo largo de varios ciclos de desarrollo buscando que el tiempo final de desarrollo sea lo más mínimo posible.

A continuación, se muestra la planificación temporal que se ha seguido:

	<i>Semanas</i>
1. PRIMERA ITERACIÓN – ANÁLISIS Y DISEÑO	
1.1. Preparación de los lenguajes y herramientas	3
1.2. Familiarización con el proyecto base	1
1.3. Planificación temporal	1
1.4. Especificación de requisitos del Software	2
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 la audiencia	
1.5.2. Escenarios de uso	
1.5.3. Análisis de tareas	
1.5.4. <i>Sitemap</i>	
1.5.5. <i>Wireframes</i>	
1.6. Inicio de los primeros anexos	2
1.7. Refinamientos de los requisitos y actores del sistema	1
1.8. Diagramas del análisis del software	1

1.9. Elaboración del Prototipo en Papel	2
1.9.1. Diseño	
1.9.2. Evaluación	
1.10. Elaboración del Prototipo Digital	3
1.10.1. Diseño	
1.10.2. Evaluación	
2. SEGUNDA ITERACIÓN – DESARROLLO Y PRUEBAS	
2.1. Continuación de los anexos	2
2.2. Implementación de código	4
2.2.1. Gestión de Algoritmos	
2.2.2. Gestión de entidades PACS	
2.2.3. Gestión de imágenes DICOM	
2.3. Pruebas	3
2.3.1. Gestión de Algoritmos	
2.3.2. Gestión de entidades PACS	
2.3.3. Gestión de imágenes DICOM	
3. TERCERA ITERACIÓN – MEJORAS	
3.1. Revisión final requisitos	1
3.2. Revisión final diagramas del análisis	1
3.3. Revisión final funcionalidad	1
3.4. Prueba final de la aplicación	1
3.5. Últimas mejoras de la aplicación	2
3.6. Elaboración de la documentación final	3
3.7. Últimas revisiones y entrega	1

FIGURA 20. PLANIFICACIÓN TEMPORAL DEL PROYECTO

Por lo tanto, las fechas estimadas para cada una de las fases de la planificación han sido:

Iteración	Comienzo	Fin
Análisis y Diseño	22 de febrero de 2022	21 de junio de 2022
Desarrollo y Pruebas	16 de agosto de 2022	25 de octubre de 2022
Mejoras	1 de noviembre de 2022	10 de enero de 2023

TABLA 1. FECHAS ESTIMADAS SEGÚN LA PLANIFICACIÓN TEMPORAL

Uno de los aspectos que se debe destacar es el hecho de haber trabajado sobre un proyecto real con unos tiempos preestablecidos a partir de un proyecto base ya existente. Al tiempo que se le ha dedicado a cada una de las diferentes fases que conforman el Proceso Unificado se le debe añadir un tiempo extra ya que se ha invertido tiempo en entender lo que ya estaba realizado. Este aspecto se puede ver en la *Tabla 1* ya que se dedicaron dos semanas en entender y familiarizarse con los proyectos que ya estaban realizados.

En cuanto a las horas dedicadas los días que se ha realizado el proyecto fueron muy reducidas puesto que los primeros meses se tuvo que compaginar el proyecto con las prácticas externas del Grado en Ingeniería Informática y con alguna asignatura del grado, dedicándole así un menor tiempo del deseable.

En segundo punto, cuando se realizó una reestructuración del calendario, se aumentaron el número de horas dedicadas al proyecto, aunque el autor únicamente tenía la oportunidad de dedicarle horas por la mañana al proyecto.

En conclusión, se dirá que se dedicaron entre 2 y 4 horas diarias, aumentando ese número en las dos últimas iteraciones del proyecto.

Para una información más detallada sobre la planificación temporal que se ha llevado y, el porqué de los tiempos empleados en cada una de las tareas se puede consultar el [Anexo II. Plan de Proyecto Software](#) adjunto con este documento.

6.2. ESPECIFICACIÓN DE REQUISITOS

Para consultar la información que no aparece en este documento como puede ser el resto de los ejemplos de cada uno de los puntos, consultar el [Anexo I. Especificación de Requisitos Software](#) adjunto con este documento.

6.2.1. DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA

En primer lugar, cabe destacar los diferentes actores que conforman el sistema. Para esta descripción se ha utilizado una relación de generalización para indicar que el Administrador desempeña las mismas funciones que el Experto en *Machine Learning*, pero, con funcionalidades extras. Por otra parte, el actor Experto en *Machine Learning* desempeñará las mismas funciones que el actor Usuario, pero, con funcionalidades extras. Por último, el usuario anónimo será el usuario que se quiere registrar.

Las la *Figura 21* muestra el diagrama de actores del sistema.

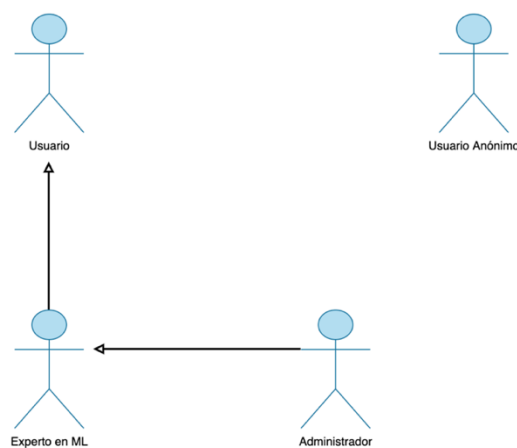


FIGURA 21. DIAGRAMA DE ACTORES DEL SISTEMA

Los cuatro usuarios son actores principales, pues tienen objetivos de usuario que se satisfacen a través de utilizar servicios del sistema [27]. Los cuatro actores son también actores humanos.

Para una descripción más detallada de los diferentes usuarios del sistema, se puede consultar el [Anexo I. Especificación de Requisitos Software](#) adjunto con este documento.

6.2.2. CATÁLOGO DE REQUISITOS DEL SISTEMA

Respecto al catálogo de requisitos del sistema, se han identificado requisitos funcionales, requisitos no funcionales y requisitos de información. Es importante destacar que los requisitos definidos se centran en aquellos asociados a la extensión del sistema KoopaML del presente proyecto, sin mencionar los requisitos previamente identificados en [7].

En cuanto a los requisitos de información que se han definido para este sistema son:

- **Información sobre Usuarios.** El sistema deberá almacenar la información correspondiente con los usuarios registrados en la aplicación. Los datos que se almacenarán de estos serán el nombre, los apellidos, el nombre de usuario, correo electrónico, etc.
- **Información sobre Imágenes DICOM.** El sistema deberá almacenar la información correspondiente con las imágenes DICOM que se encuentran en el sistema. Los datos que se almacenarán sobre estas serán el nombre y código del paciente, nombre y código del proyecto al que pertenecen, etc.
- **Información sobre entidades PACS.** El sistema deberá almacenar la información correspondiente con las entidades PACS a las que se puede conectar este. Los datos que se almacenarán sobre estas serán la dirección IP y el puerto, así como un nombre y una descripción de la entidad.
- **Información sobre Algoritmos.** El sistema deberá almacenar la información correspondiente con los algoritmos que se encuentran en el sistema. Los datos que se almacenarán sobre estos serán el nombre, la descripción, el tipo de salida, etc.

En cuanto a los requisitos funcionales, se han realizado mediante casos de uso, organizándolos en los siguientes paquetes de análisis: Gestión de Algoritmos, Gestión de entidades PACS y Gestión de Imágenes DICOM. Las siguientes figuras muestran los diagramas de casos de uso de estos paquetes de análisis (*Figura 22 - Figura 24*).

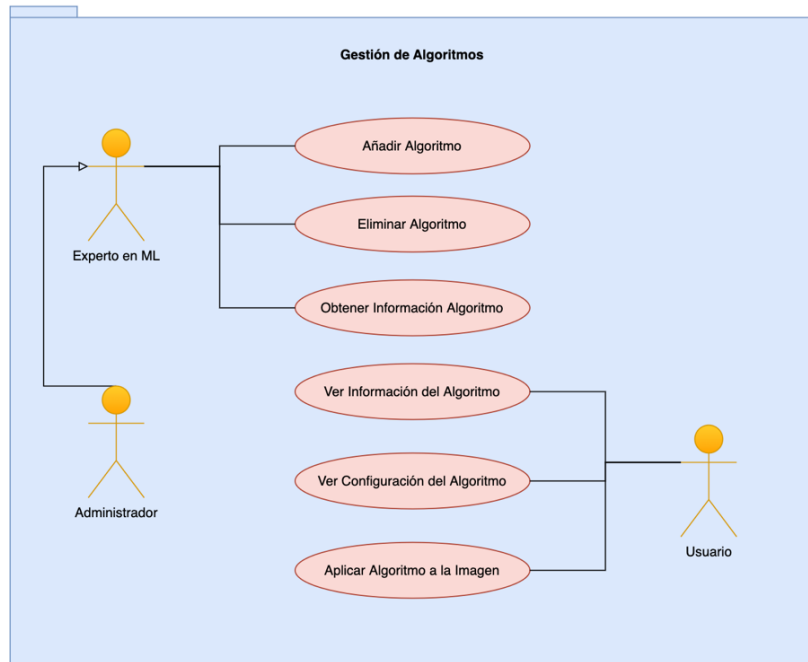


FIGURA 22. PAQUETE DE ANÁLISIS GESTIÓN DE ALGORITMOS

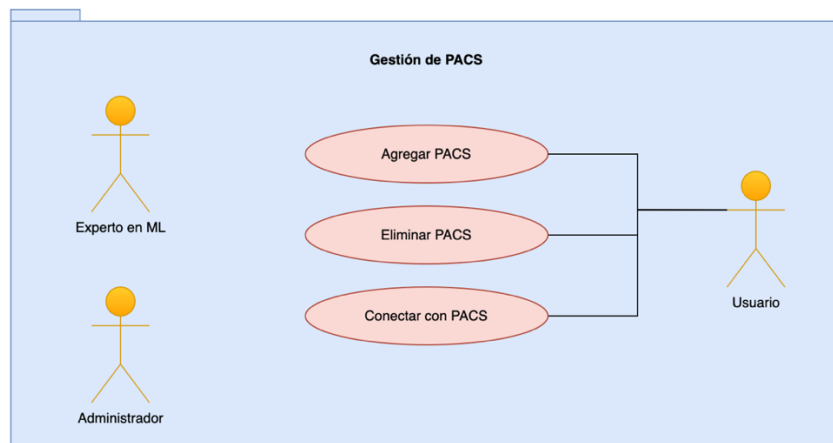


FIGURA 23. PAQUETE DE ANÁLISIS GESTIÓN DE PACS

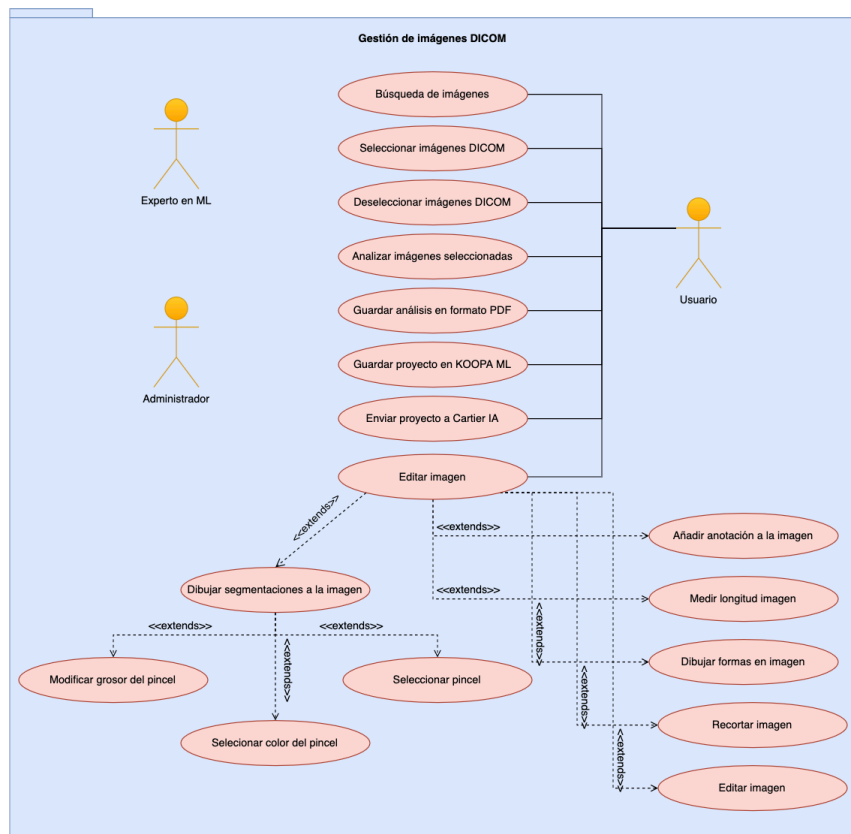


FIGURA 24. PAQUETE DE ANÁLISIS GESTIÓN DE IMÁGENES DICOM

Luego, para identificar los requisitos no funcionales se utilizó el modelo planteado por Sommerville [28]. Este modelo se escogió debido a su previo uso en trabajos realizados en el grado y a su familiaridad con él.

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

- **Usabilidad.** El sistema deberá proporcionar una interfaz de usuario minimalista e intuitiva para el usuario final, sin llegar a sobrecargar de información al usuario.
- **Portabilidad.** El sistema deberá poder desplegarse en cualquier sistema operativo de Linux, MacOS o Windows.
- **Compatibilidad.** El sistema deberá tener una interfaz de usuario compatible con la mayoría de los navegadores actuales.

- **Rendimiento.** El sistema deberá funcionar con el menor consumo de recursos necesarios.
- **Fiabilidad.** El sistema deberá levantarse inmediatamente después de una caída del sistema, asegurando la consistencia de la información adecuadamente.
- **Implementación.** El sistema deberá proporcionar una aplicación web como interfaz de usuario para poder acceder al sistema con cualquier equipo informático a través de un navegador sin necesidad de ninguna instalación.
- **Privacidad.** El sistema deberá guardar la mínima información de los usuarios que tengan acceso al sistema.

6.3. PROCESO DE DISEÑO CENTRADO EN EL USUARIO

En este apartado, se expondrá todo el proceso de creación del diseño de la aplicación. Se realizará comenzando por los diagramas de casos de uso que han dado una visión general a cómo enfrentar las diferentes funcionalidades del proyecto. Este proceso se continuó con la creación de prototipos en papel que tendrán un proceso de evaluación con usuarios reales. Después de dichas pruebas, se creará una primera versión de prototipo digital que, al igual que los prototipos en papel, tendrán su evaluación con usuarios reales. Cada vez que se realizaban pruebas de usuario del prototipo digital, 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 cuanto al proceso de creación, se incluirán pautas para llegar a cumplir con los objetivos de un intercambio de información eficiente, algunos de estos objetivos son el reducir el número de errores a un mínimo, incrementar la satisfacción de los usuarios con el sistema, así como ampliar la facilidad de uso, entre otros. Por otra parte, se van a tener en cuenta elementos como una jerarquía visual clara,

información de la ubicación del usuario en todo momento y un diseño coherente en toda la aplicación desarrollada.

Es muy importante diseñar sistemas que sean efectivos, eficientes y sencillos a la hora de utilizarlos, dado que la sociedad disfrutará de estos avances. La dificultad viene dada por una serie de restricciones que obligan a los equipos de diseño a hacer algunos sacrificios en este.

Cabe destacar que en la ampliación de la aplicación se ha partido del diseño que existía en KoopaML, pero se ha planteado todo un proceso de rediseño por dos motivos principales. En primer lugar, la decisión de separar los proyectos de *Machine Learning* de los proyectos DICOM, planteando espacios diferentes dentro de la misma aplicación. En segundo lugar, a la hora de analizar la aplicación se detectaron problemas de usabilidad, tales como falta de información en todo momento sobre la ubicación del usuario dentro de la aplicación; y problemas relacionados con los principios de diseño, como por ejemplo, la falta de unicidad en toda la aplicación.

Para consultar la información que no aparece en este documento como puede ser el resto de los ejemplos de cada uno de los puntos, consultar el [Anexo IV. Proceso centrado en el usuario](#) adjunto con este documento.

6.3.1. CICLO DE DISEÑO CENTRADO EN EL USUARIO

En cuanto al proceso del diseño centrado en el usuario de la aplicación se han seguido las siguientes fases :

- **Fase de Exploración:**
 - Búsqueda de necesidades
 - Definición de la audiencia
 - Escenarios de uso
- **Fase de Conceptualización:**
 - *Sitemap*
 - *Wireframes*

- Aspectos del diseño
- **Diseño y Prototipado:**
 - Prototipo en Papel
 - Pruebas con Usuarios del prototipo en papel
 - Prototipo Digital
 - Pruebas con Usuarios del prototipo digital

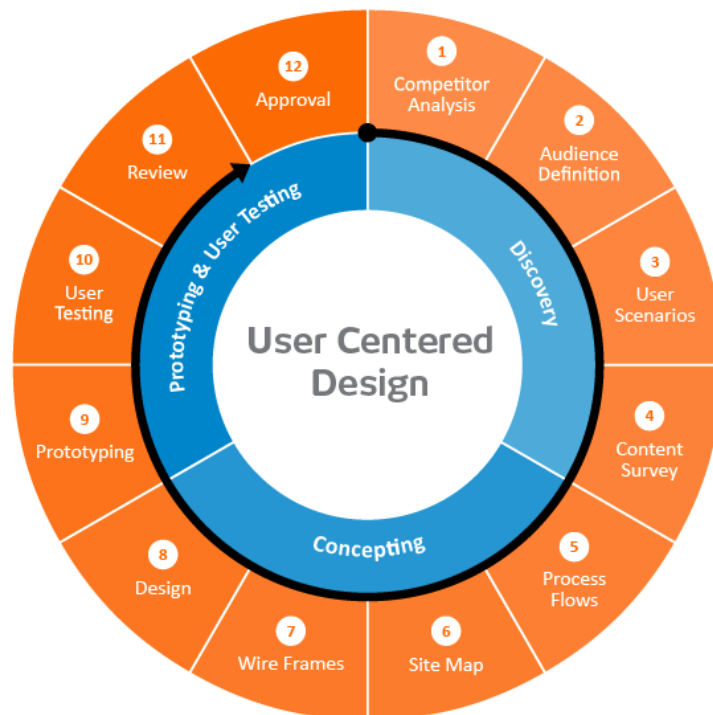


FIGURA 25. CICLO DE DISEÑO DEL PROCESO CENTRADO EN EL USUARIO

El esquema mostrado en la *Figura 25* está particularizado para diseños web, por lo que, será el ciclo que será utilizado para el proyecto. El desarrollo de un sistema se cimienta sobre criterios consistentes basados en la experiencia, evitando así juicios intuitivos que se tuvieran previamente.

A continuación, se expondrán los aspectos más relevantes de cada una de las fases realizadas, no obstante, si se desea consultar más información de la detallada en este documento, consultar el [Anexo IV. Proceso centrado en el usuario](#) adjunto con este documento.

6.3.2. FASE DE EXPLORACIÓN

Durante esta fase se realiza un trabajo de investigación para llegar a conocer las necesidades e intereses de los futuros usuarios de la aplicación. Para este proceso se ha utilizado la metodología *Engaging Persona* de Alan Cooper [29] y la representación de escenarios de uso de las situaciones y condiciones en las que los usuarios de la aplicación puedan llegar a sentir que necesitan su utilización [30].

Búsqueda de necesidades

Los usuarios, hasta el momento, utilizaban dos herramientas, *Cartier-IA* y *KoopaML*, para poder almacenar, gestionar y recibir imágenes DICOM. Se solicitó una unificación de ambas herramientas para poder llegar a entrenar algoritmos y modelos de aprendizaje automático con las imágenes DICOM que se manden a través de una entidad PACS (*Cartier-IA*) que se encuentra conectada a la aplicación desarrollada (*KoopaML*).

Definición de la audiencia

Como ya se ha comentado, para la definición de la audiencia se ha utilizado la metodología *Engaging Persona* de Alan Cooper [29], cuyo objetivo es el de definir un arquetipo de persona o un perfil objetivo de la aplicación que permita lograr una comprensión mucho más profunda de los usuarios que conformarán la audiencia.

Para aplicar esta metodología de la mejor forma posible hay que seguir una serie de 10 pasos [31] basada en los pasos que expone Lene Nielsen en su artículo *Personas* [32]. Estos 10 pasos están pensados para cubrir todo el proceso desde la recopilación de datos preliminares, hasta el uso más activo de un desarrollo continuo de la personalidad, estos son:

- **Recopilación de la información.** En este paso se recopila tanta información sobre los usuarios como sea necesaria a través de una investigación de usuarios reales, es decir, usuarios que vayan a hacer uso de la aplicación.

El método que se ha utilizado en este paso es la recolección de datos cuantitativos, obteniendo como resultado la obtención de la información a través de las siguientes preguntas:

- ¿Quiénes son los usuarios?
 - ¿Cuántos usuarios hay?
 - ¿Qué hacen los usuarios con el sistema?
- **Formulación de una hipótesis.** En este paso se va a comenzar a formar una idea general sobre los distintos usuarios que van a hacer uso de la aplicación.

Los métodos utilizados en este paso son el análisis del material, agrupación de usuarios e identificación y nombramiento de los grupos de usuario, obteniendo como resultado un borrador que va a reflejar, de forma muy genérica la distinción de los grupos de usuarios que van a hacer uso de la aplicación.

- **Aceptación de la hipótesis.** Este paso tiene como objetivo tomar la decisión de apoyar o rechazar la hipótesis planteada en el paso anterior sobre las diferencias entre los usuarios que hacen uso de la aplicación.

El método que se ha utilizado en este paso es la recolección de datos cualitativos, obteniendo como resultado la aportación de datos relevantes sobre la situación de los usuarios.

- **Establecer un número de personas.** En este paso se decide el número final de personas que se van a necesitar crear.

El método utilizado en este paso es la caracterización, obteniendo como resultado la descripción de grupos o categorías de usuarios que se van a utilizar en la aplicación.

- **Descripción de las personas.** En este paso se han realizado una descripción detallada de tipos de personas que van a hacer uso de la aplicación.

El método utilizado en este paso es la categorización, obteniendo como resultado la descripción de las diferentes personas que se van a poder encontrar en la aplicación.

- **Definición de situaciones.** En ese paso se describen una serie de situaciones específicas a las que las Personas se van a poder enfrentar y que puedan desencadenar en el uso de la aplicación.

El método utilizado en este paso es el análisis de datos para las necesidades y situaciones de los usuarios de la aplicación, obteniendo como resultado la documentación de necesidades y situaciones.

- **Aceptación por parte de la organización.** En este paso se va a realizar un agrupamiento de todos los miembros del equipo que han ido involucrando en la aplicación a lo largo del proceso.

El método utilizado en este paso es la corrección por parte de los tutores de las descripciones de usuarios.

- **Difundir el conocimiento.** En este paso se debe difundir las Personas que se han descrito y desarrollado.

El método utilizado en este paso son reuniones *online* y presenciales con los tutores del Trabajo de Fin de Grado.

- **Creación de escenarios de uso.** En este paso se realizarán descripciones de las situaciones en las que las Personas utilizan la aplicación, reflejando sus necesidades, rasgos personales, problemas y soluciones, etc.

El método utilizado en este paso es el del escenario narrativo, obteniendo como resultado varios documentos como son los escenarios de uso.

- **Realización de ajustes continuos.** En este paso se observa cómo los usuarios van a hacer uso de la aplicación en un futuro.

Como resultado de la definición de la audiencia, se van a definir, de forma resumida, los dos tipos de usuarios que se pueden encontrar en la aplicación desarrollada.

- Los médicos que van a ser el grupo de usuarios principales puesto que la aplicación va destinada al uso y análisis de las imágenes DICOM por parte de estos usuarios. Este grupo puede tener conocimiento sobre IA/ML y, en el caso de que no lo tengan, suelen mostrar un gran interés por conseguir ese conocimiento. Por el contrario, este grupo de usuarios no posee conocimientos de programación. En cuanto al rango de edad de este grupo, es bastante amplio.
- Los expertos en IA/ML van a ser el grupo de usuarios secundarios y tienen conocimiento bastante amplio sobre IA/ML. La gran diferencia que tiene este grupo de usuarios con el grupo de usuarios principales es la de que estos sí que tienen un conocimiento bastante amplio de programación. Al igual que el grupo de usuarios principales, el rango de edad de este grupo es bastante amplio.

Por otra parte, alguno de los usuarios modelo resultantes de esta fase fueron (*Figura 26 y Figura 27*):



FIGURA 26. USUARIO MODELO NÚMERO 1 (ALMUDENA)

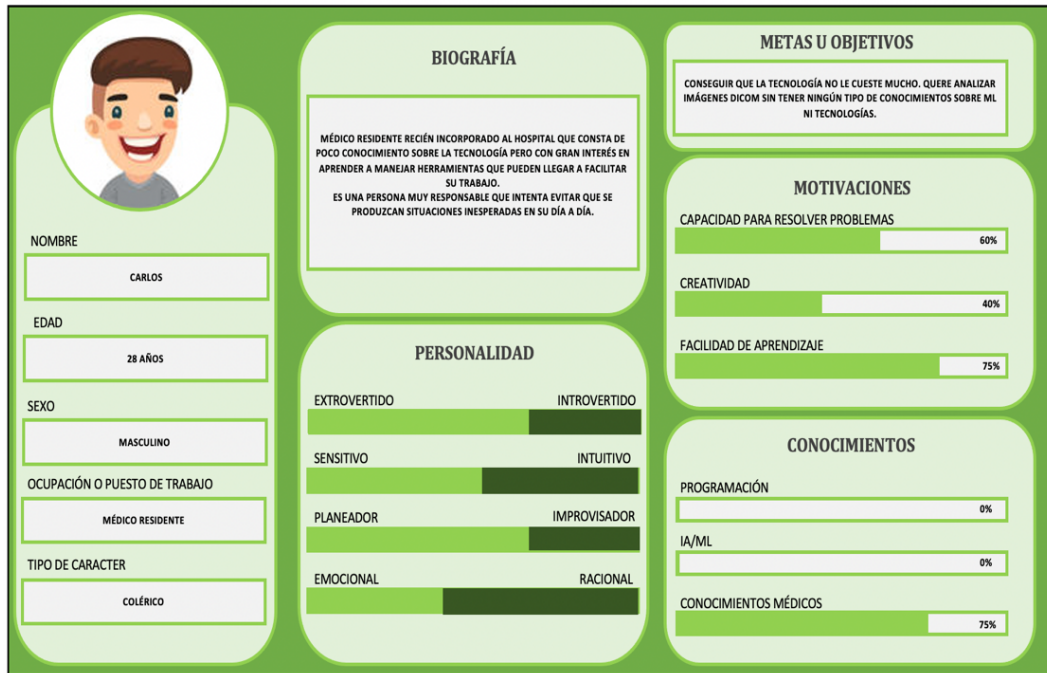


FIGURA 27. USUARIO MODELO NÚMERO 2 (CARLOS)

Escenarios de uso

Una vez que se ha realizado la definición de la audiencia, se crean los diferentes escenarios de uso. Un escenario de uso se trata de una historia ficticia cuyo objetivo es centrarse en la experiencia que varios usuarios pueden llegar a tener con el sistema que se quiere desarrollar.

Para ello, se deben responder a las siguientes preguntas [33]:

- ¿Quién es el usuario de la historia?
- ¿Por qué va a utilizar la aplicación? Es decir, la situación que va a llevar al usuario protagonista al uso de la aplicación.
- ¿Qué objetivo específico tiene el usuario?
- ¿Cómo realiza y completa ese objetivo?
- Experiencias del usuario que sean relevantes en la situación, es decir, puntos de fricción o estrés en la vida diaria o experiencias del usuario.

A continuación, se mostrará un ejemplo de escenario de uso narrativo, aunque para obtener los demás escenarios se puede consultar el [Anexo IV. Proceso centrado en el usuario](#) adjunto con este documento.

Mario Ingelmo Fraile (Dr. Ingelmo) es una persona de 45 años que dedica su vida a la medicina. Estudió la carrera de medicina en la Universidad Complutense de Madrid donde conoció al que sería el amor de su vida, una vez acabados sus estudios decidió trasladarse a Salamanca donde consiguió una plaza en el equipo de Cardiología del Hospital Clínico Universitario como residente.

Ángel Luis De cabo Fernández (Dr. De Cabo) es una persona de 45 años que se decantó por estudiar la carrera de medicina en la Universidad Complutense de Madrid, cuando acabó sus estudios decidió volver a su ciudad natal, Salamanca, donde consiguió plaza como residente en el equipo de cardiología del Hospital Clínico Universitario. Su sueño desde que tenía 22 años era ser cardiólogo puesto que, cuando se encontraba estudiando la carrera, su madre falleció esperando un trasplante de corazón y, desde ese momento, siempre quiso ayudar a personas que padezcan enfermedades relacionadas con el corazón.

Mario y Ángel Luis se conocieron el primer día de carrera y, desde entonces, son mejores amigos, trabajan juntos y son vecinos.

Al estar en el mismo equipo de trabajo, les gusta comentar las situaciones de cada uno de los pacientes para poder evitar negligencias médicas a la hora de comentarle a los pacientes sus diagnósticos. Casi siempre están de acuerdo y, cuando no lo están ambos argumentan sus posibles diagnósticos.

Un día, un paciente llegó a la consulta del Dr. Ingelmo para comentarle su situación. Tenía una enfermedad denominada enfermedad de las arterias coronarias, Mario antes de comentarle la situación a su amigo decidió hacerle todas las pruebas posibles y así ir adelantando el tiempo ya que suele ser una enfermedad bastante grave. Una vez que tenía las pruebas resueltas ambos amigos se reunieron para debatir sobre las posibles soluciones que podría tener el paciente.

El Dr. Ingelmo mantiene la idea de que la única solución que hay a esa enfermedad es realizar un trasplante de corazón al paciente mientras que el Dr. De Cabo contradice a su amigo diciendo que no es necesario un trasplante ya que con un baipás coronario se podría solucionar gran parte de la enfermedad y así ganar tiempo para un posterior trasplante.

Como no se pusieron de acuerdo, decidieron coger una tercera opinión, esta vez decidieron comentarle la situación al hermano de Ángel Luis que, también se dedica a la medicina, pero en este caso, desde un punto más científico. El hermano comenta la posibilidad de usar una herramienta de almacenamiento y gestión de imágenes DICOM llamada Koopa ML pudiendo aplicarles varios algoritmos a las imágenes DICOM que se le han realizado al paciente y, después analizarlos y comparar los resultados. Ambos médicos se quedan sorprendidos puesto que podrían ahorrar bastante tiempo con dicha herramienta y evitar esas largas discusiones que han tenido durante toda su profesión.

Decidieron analizar las imágenes DICOM que tenían realizadas al paciente aplicando varios algoritmos y su posterior comparación. Como resultado, obtuvieron que la enfermedad no estaba tan desarrollada como ellos se pensaban y que Ángel Luis tenía razón en aplicarle un baipás coronario y poder frenar en la mayor medida la enfermedad.

Gracias al hermano de Ángel Luis, estos dos médicos descubrieron una gran herramienta que les va a facilitar bastante sus análisis de imágenes DICOM para así poder ayudar más rápidamente a los pacientes que les pidan ayuda como profesionales. Desde ese momento, no solo comentan sus posibles diagnósticos como profesionales que son si no que se apoyan en la herramienta para evitar futuras confusiones entre ellos.

6.3.3. FASE DE CONCEPTUALIZACIÓN

Durante esta fase se van a identificar y especificar los requisitos de los usuarios de la aplicación para su posterior implementación en esta. Mediante los requisitos que tienen los usuarios, podemos adoptar nuevas funcionalidades de nuestra aplicación.

Para una descripción más detallada de esta fase del sistema, se puede consultar el [Anexo I. Especificación de Requisitos Software](#) adjunto con este documento.

Mapa del sitio o SiteMap

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

Se ha modificado con respecto al proyecto de KoopaML inicial ya que, al añadir nuevas funcionalidades, se han debido de añadir nuevas tareas relacionadas con la recepción, almacenamiento y gestión de imágenes DICOM. En la se puede ver el mapa del sitio.

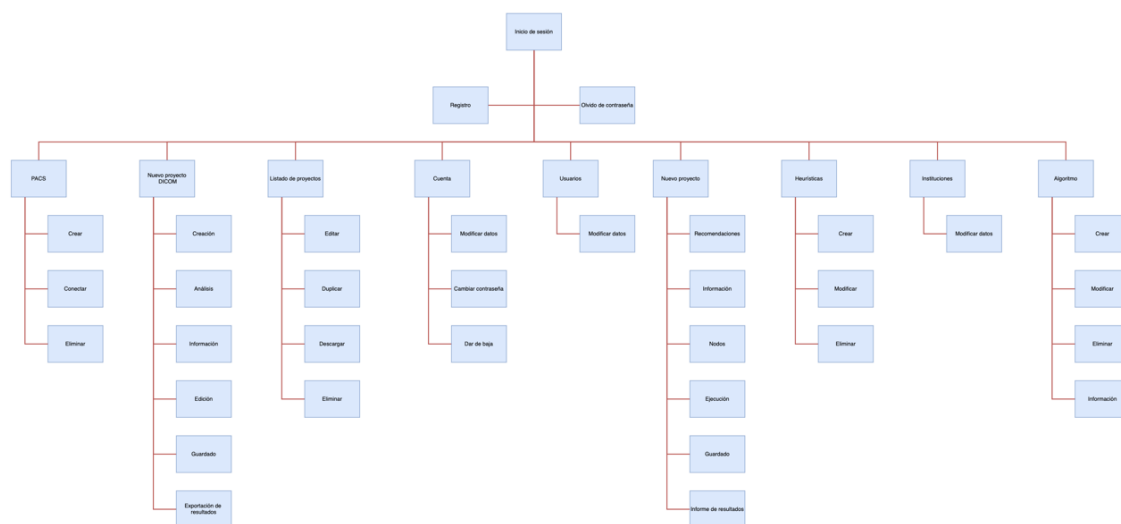


FIGURA 28. SITEMAP DEL SISTEMA

Wireframes

Un *wireframe* es un boceto donde se representa visualmente de una forma muy sencilla y esquemática la estructura de la aplicación. En esta etapa, se ha definido el contenido y la posición inicial de los diferentes elementos que va a contener la aplicación (menús de navegación, páginas de portada, bloques de contenido, etc.).

Antes de realizar el prototipo ya sea digital o en papel, se optó por esta fase para realizar una aproximación de lo que va a ser la aplicación teniendo en cuenta que es

una extensión de una aplicación existente. Únicamente se han realizado *wireframes* de las pantallas más principales de la aplicación para aclarar espacios.

A continuación, se muestran algunos de los *wireframes* realizados para este proyecto:

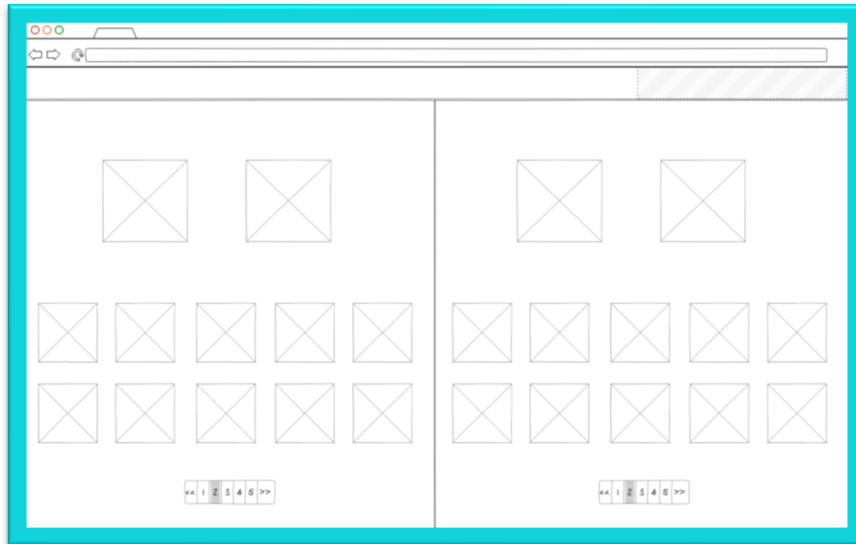


FIGURA 29. WIREFRAME PRIMER PROTOTIPO PANTALLA INICIAL

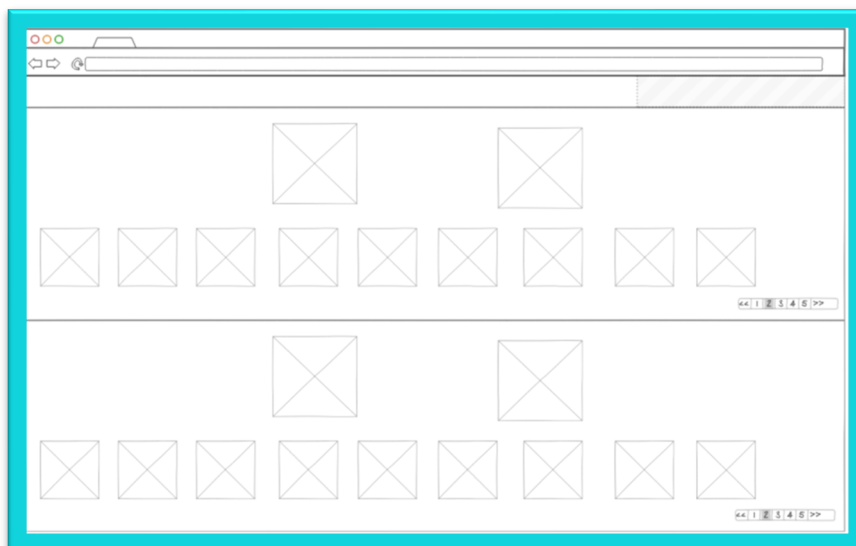


FIGURA 30. WIREFRAME SEGUNDO PROTOTIPO PANTALLA INICIAL

Estas dos primeras pantallas (*Figura 29 - Figura 30*) se corresponden con la página de inicio de la aplicación. Se han realizado dos diseños diferentes para proporcionar

varias opciones a los usuarios de cómo desean observar dicha pantalla. En esta pantalla se muestra lo siguiente:

- Por un lado (*Figura 29*), se ubican las diferentes opciones para crear un nuevo proyecto ML y los diferentes proyectos que el usuario ha ido creando y guardando en la aplicación. En función de que tipo de usuario sea, se tienen diferentes opciones:
 - Si el usuario es un usuario principal podrá tener la oportunidad de crear un nuevo proyecto y, también, subir un proyecto que tenga almacenado.
 - Si el usuario es un usuario experto podrá tener a parte de las opciones del usuario principal, la opción de modificar heurísticas.
- Por otro lado (*Figura 30*), se encuentran las diferentes opciones para crear un nuevo proyecto DICOM y los diferentes proyectos que el usuario ha ido guardando en la aplicación. Al igual que en los proyectos ML, en función del tipo de usuario, se pueden realizar diferentes acciones:
 - Si el usuario es un usuario principal podrá tener la oportunidad de crear un nuevo proyecto y, también configurar los diferentes PACS a los que se desea conectar la aplicación.
 - Si el usuario es un usuario experto podrá realizar las acciones del usuario principal y, también, la opción de añadir y modificar algoritmos a través de scripts.

Una vez que se ha entrado en la aplicación y se ha seleccionado el proyecto al que se quiere acceder, se tiene la oportunidad de editar las imágenes DICOM que se han seleccionado y, a parte, el poder seleccionar los diferentes algoritmos que se le pueden aplicar a las imágenes (*Figura 31*).

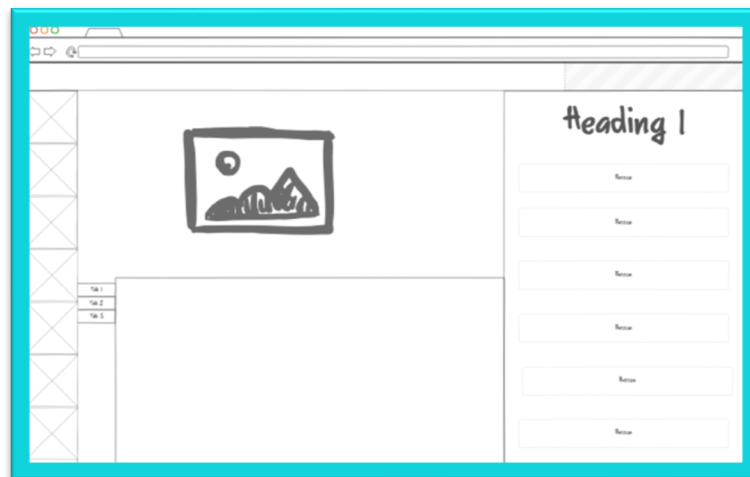


FIGURA 31. WIREFRAME VISUALIZACIÓN DE IMÁGENES DICOM

Diseño

Los aspectos técnicos del diseño como pueden ser la tipología escogida fueron determinados en la primera versión de la aplicación, es decir, en la versión ya existente.

Sin embargo, se pueden comentar algunos cambios significativos que se han llevado a cabo en la ampliación de la aplicación:

- Todas las pantallas que tienen una misma funcionalidad (gestión de algoritmos, gestión de entidades PACS, etc.) han sido creadas directamente con la misma plantilla, aspecto que no pasa con la parte de *Machine Learning* de la aplicación.
- Todos los botones de igual significado han sido representados por el mismo formato y color tal y como marca el sexto principio de usabilidad de Nielsen.

6.3.4. FASE DE PROTOTIPOS Y PRUEBAS DE USUARIOS

En este apartado se realizaron los diferentes prototipos (tanto el prototipo realizado en papel como el prototipo digital) a partir de los cuales se va a llegar al que será el prototipo funcional de la aplicación. A su vez, para todo este proceso, se llevarán a cabo pruebas de usuario con cada uno de los prototipos.

Prototipo en papel

Durante esta fase se elaboró una versión de manera incremental con respecto a los *wireframes* que se realizaron en la fase anterior siendo así esta versión una con novedades y cambios con respecto a la anterior. Así mismo, se tendrá en cuenta la opinión de los usuarios interesados para realizar las mejoras entre ellos.

Dicho diseño se someterá a distintas versiones y pruebas de usuarios mediante las cuáles se llegará a una posible versión final, la cual será transformada en futuras etapas en un prototipado digital.

El diseño se podrá visualizar en el siguiente enlace <https://marvelapp.com/prototype/1822b7ij> donde se podrá navegar a través de las distintas pantallas que se han realizado.

Se podrán observar de manera más detallada todos los cambios realizados en el [Anexo IV. Proceso centrado en el usuario](#) adjunto con este documento. Se adjunta imagen de las pantallas más significativas de la aplicación (*Figura 32 - Figura 34*).

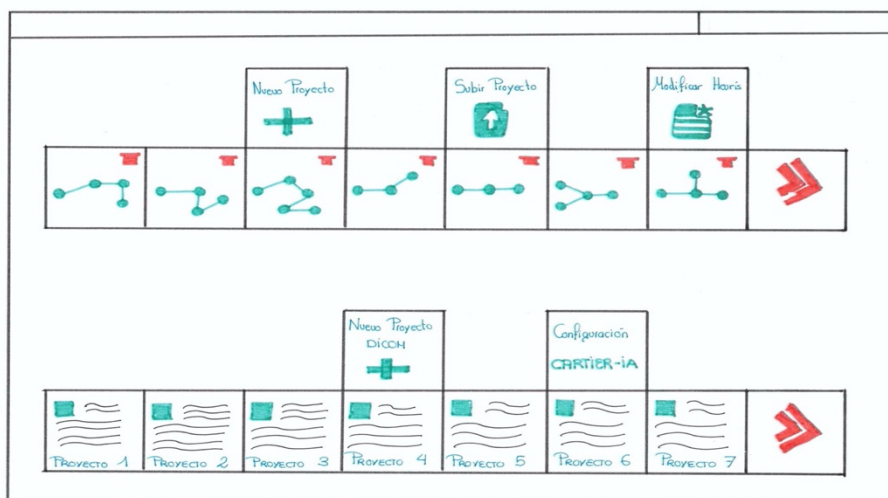


FIGURA 32. PROTOTIPO EN PAPEL: PRIMERA VERSIÓN PANTALLA INICIAL

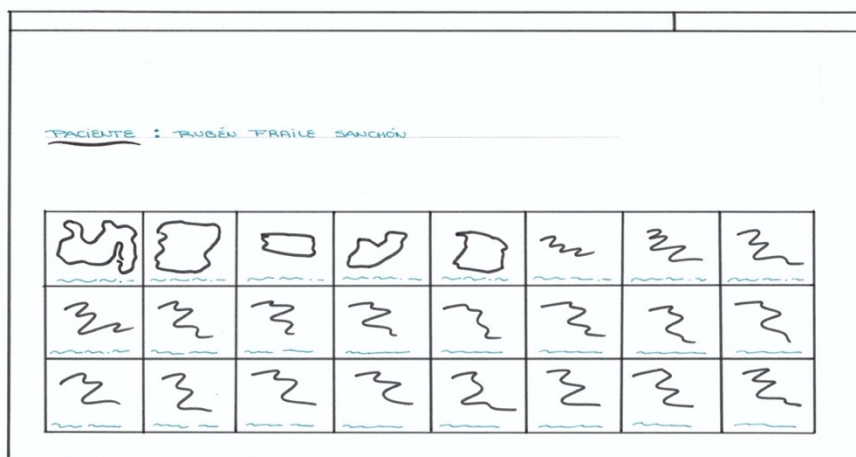


FIGURA 33. PROTOTIPO EN PAPEL: IMÁGENES RESULTANTES DE UNA BÚSQUEDA EN UN PACS

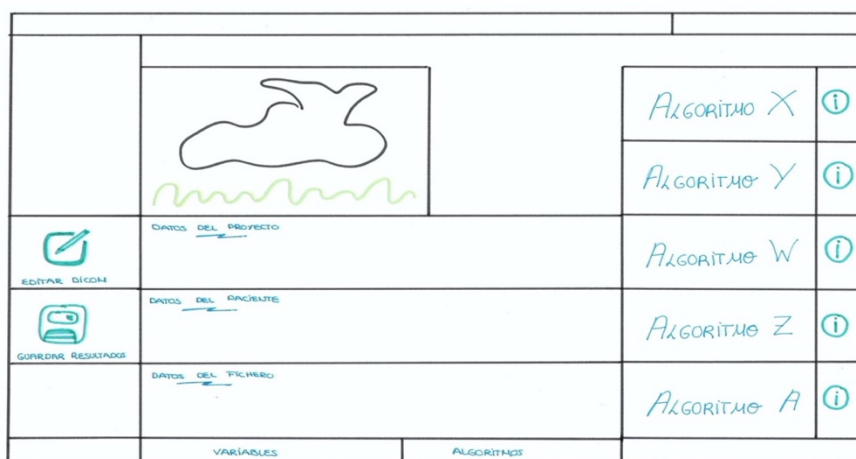


FIGURA 34. PROTOTIPO EN PAPEL: CONFIGURACIÓN DE ALGORITMOS DE INTELIGENCIA ARTIFICIAL EN LA APLICACIÓN

Los cambios más significativos con respecto al primer diseño realizado son:

- Ampliación del diseño, añadiendo así nuevas pantallas que pueden formar parte de la aplicación.
- Una mejor distribución del espacio de la pantalla.

Evaluación del prototipo en papel

Para la evaluación de este prototipo se efectuó una prueba a los tutores de este Trabajo de Fin de Grado, estos usuarios poseían conocimientos sobre el dominio de los datos, así como de aprendizaje automático y de programación, pudiendo así orientar de una mejor manera en los errores que se visualicen.

Esta evaluación fue realizada de manera online en la que se presentó el diseño y se fueron comentando los distintos cambios que se podrían realizar, al igual que la elección entre las opciones de diseño en las pantallas en las que se dudaba qué diseño escoger.

Una vez realizadas las pruebas con los usuarios, se extrajeron las siguientes conclusiones:

- Pantalla sobrecargada de información a la hora de analizar las imágenes.
- No se le deja espacio al menú de herramientas, se encuentra mal situado.
- La situación de los resultados una vez se hayan aplicado los algoritmos a las imágenes no es muy apropiada puesto que se le deja más espacio a la configuración de los algoritmos que a los resultados de la aplicación.
- En cuanto a la búsqueda de las imágenes, se propuso el realizar una búsqueda mucho más concreta pudiendo así realizar búsquedas por todas las distintas entidades que pertenecen a las imágenes.

Prototipo digital

En cuanto a este prototipo, se ha diseñado la aplicación introduciendo funcionalidad, colores e incorporando todos los aspectos comentados en la evaluación de la versión realizada en papel.

Se podrán observar de manera más detallada todos los cambios realizados en el [Anexo IV. Proceso centrado en el usuario](#) adjunto con este documento. Se adjunta imagen de las pantallas más significativas de la aplicación ()

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

<https://xd.adobe.com/view/14857260-8c43-4fc5-9066-9f7707c23960-41d6/>

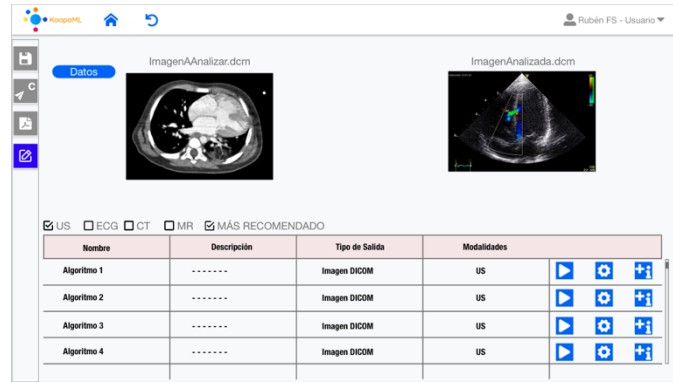


FIGURA 35. PROTOTIPO DIGITAL: APLICACIÓN DE ALGORITMO DE INTELIGENCIA ARTIFICIAL A UNA IMAGEN DICOM SELECCIONADA

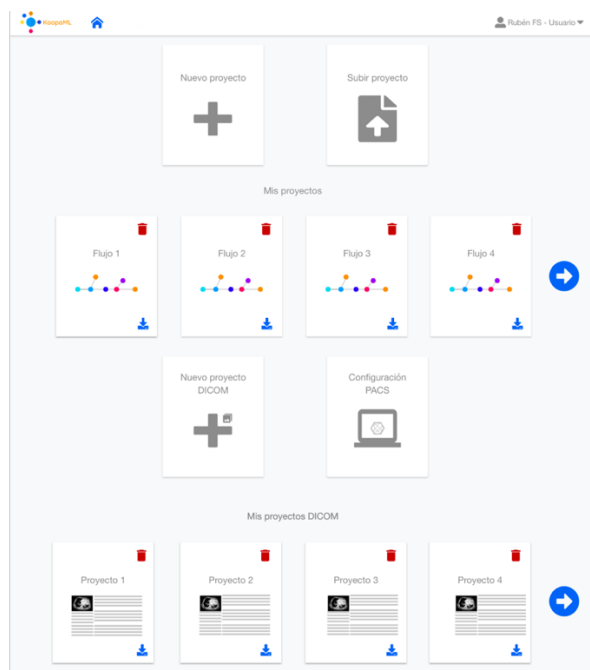


FIGURA 36. PROTOTIPO DIGITAL: PANTALLA INICIAL



FIGURA 37. PROTOTIPO DIGITAL: GESTIÓN DE ENTIDADES PACS EN EL SISTEMA

Los cambios más significativos con respecto al prototipo en papel realizado son:

- Remodelado del diseño para que las pantallas no se recargasen.
- En el caso de los algoritmos, se han situado debajo de las imágenes en una lista, poniendo en la parte superior de la pantalla las dos imágenes.
- El menú se ha cambiado de lugar situándolo en uno de los laterales y no debajo de las imágenes como se encontraba en el prototipo en papel.

Evaluación del prototipo digital

La evaluación de los usuarios con este prototipo se ha llevado a cabo a través de la técnica de grupos focales, pues el objetivo de esta evaluación es estudiar las actitudes y experiencias de los usuarios con la aplicación [34].

Debido a las condiciones socio sanitarias derivadas de la crisis del COVID-19 la evaluación se lleva a cabo a través de la plataforma de Zoom y tuvo una duración aproximada de una hora [35]. Se solicitó el consentimiento de grabación del proceso de evaluación de los usuarios con el objetivo de recopilar la máxima información posible a través de las posteriores revisiones y así poder analizar sus respuestas. Los usuarios dieron su consentimiento expreso para la grabación.

Durante la evaluación a los usuarios se les compartió la pantalla para que pudieran seguir toda la navegación en tiempo real, se les realizaron diversas preguntas de las pantallas que iban observando y la navegación entre ellas. Los usuarios participaron activamente a través de respuestas y sugerencias, así como interactivamente, pues en ocasiones se les dio la posibilidad de controlar la navegación para poder observar su interacción con la aplicación.

El guion utilizado para la evaluación contenía preguntas que buscaban fomentar el debate entre los usuarios, de forma que se pudiera recopilar una gran cantidad de información, así como comprobar si algún aspecto de la herramienta no era intuitivo o suficientemente claro. Un aspecto importante de las preguntas es que eran neutras, no presentaban juicios de valor ni afirmaciones con carácter ideológico, ni orientaban o condicionaban las respuestas de los usuarios [36].

Se obtuvo la siguiente retroalimentación para las funcionalidades del prototipo DICOM. En general, ambos expertos destacaron la sencillez e intuición de la plataforma y que estas nuevas funcionalidades les pueden resultar muy útiles.

Experto 1

- Los PACS son fáciles de configurar, y los parámetros de configuración son correctos y suficientes.
- Las imágenes DICOM son fáciles de buscar mediante ID de pacientes, estudios o existen proyectos de construcción.
- Las herramientas de edición DICOM parecen buenas. Tal vez la herramienta de segmentación no sea empleada por expertos porque utilizan otras herramientas.
- La interfaz para ejecutar modelos de Inteligencia Artificial es simple e intuitiva.

Experto 2

- Plataforma intuitiva y fácil de usar, incluso por usuarios no expertos.
- Sería útil tener alguna información y recomendaciones relacionadas con los algoritmos de Inteligencia Artificial disponibles antes de aplicarlos a las imágenes DICOM.
- Es probable que no se utilice la herramienta de segmentación. Quizá sería interesante subir imágenes ya segmentadas.

Por lo tanto, la versión que se intentará implementar será el prototipo digital que se ha realizado y explicado anteriormente.

Se podrán asumir nuevos cambios en el sistema siempre y cuando durante el desarrollo se vea necesario para así mejorar la aplicación y que los usuarios estén contentos con su uso.

6.4. DESARROLLO DEL SISTEMA

En este punto se expondrán los aspectos del desarrollo que se han implementado, así como los detalles que más esfuerzo le han supuesto al desarrollador a la hora de la realización del código de la aplicación.

Para complementar este apartado, se podrá consultar:

- El [Anexo V. Documentación del código fuente](#) adjunto con este documento se podrá ver una documentación del código fuente desarrollado.
- El [Anexo VI. Manual del usuario](#) adjunto con este documento se podrá observar una guía del usuario, así como un manual para que los usuarios puedan manejarse con fluidez en la aplicación. Este manual se puede encontrar en el siguiente enlace:

<https://www.youtube.com/watch?v=jShNxv23aXM>

6.4.1. INTEGRACIÓN EN KOOPAML

En un primer lugar, el desarrollador se centró en comprender el código de la aplicación ya implementado por los desarrolladores anteriores. Para poder integrar la nueva funcionalidad en KoopaML se ideó un nuevo espacio de trabajo, separando así el espacio de trabajo de *Machine Learning* del espacio de trabajo de DICOM ya que fue crucial para evitar una interfaz abarrotada. Además, las diferencias sustanciales entre la gestión de datos estructurados y los datos DICOM requieren tratamientos diferentes dependiendo del tipo de entrada.

6.4.2. INTEGRACIÓN CON CARTIER-IA

Cuando se crea un proyecto DICOM, se les pide a los usuarios qué PACS (Sistema para Archivos y Comunicación de Imágenes) quieren configurar en la herramienta, pudiendo acceder de esta forma a las diferentes imágenes que tiene el sistema que está configurado.

Entre los PACS configurados en las pruebas del desarrollador, hay que destacar Cartier-IA, una herramienta que puede verse como un ecosistema tecnológico que soporta todas las tareas relacionadas con la gestión de datos (incluyendo la recopilación de datos e imágenes médicas) y también permiten tanto la asistencia de sanitarios profesionales y científicos de datos para aplicar modelos de Inteligencia Artificial a las imágenes almacenadas [37].

En cuanto a la conexión entre KoopaML y cualquiera de los PACS que se encontrarían configurados en la herramienta, se ha seguido la siguiente estructura:

En primer lugar, se ha realizado una conexión entre la plantilla HTML y el fichero *Python* de la plataforma KoopaML con el siguiente código. En este fragmento de código se puede observar cómo se ha realizado una petición a través del *framework Django en JavaScript*.

```
$. post({
  url: "/post-info-images/",
  beforeSend: function (xhr, settings) {
    xhr.setRequestHeader("X-CSRFToken", '{{ csrf_token }}')
  },
  data: jsonData,
  success: function (response) {
    console.log(response)
    if (response == 0) {
      let string = "Error, no se han almacenado imágenes en el proyecto presente."
+ "Vuelva a probar con otros datos"
      $('#myModal').toggle();
      $('#modal-message').text(string);
    } else {
      let string = "Se han almacenado un total de " + response + " imágenes en el
proyecto presente"
      $('#myModal').toggle();
      $('#modal-message').text(string);
    }
  }
});
```

FIGURA 38. FRAGMENTO DE CÓDIGO PARA REALIZAR LA CONEXIÓN ENTRE JAVASCRIPT Y PYTHON

En segundo lugar, se realizan los diferentes response para terminar la conexión entre KoopaML y Cartier-IA. La primera de las herramientas pedirá los diferentes datos a la segunda y, este, le responderá con los datos necesarios. En el siguiente fragmento de código se puede observar cómo los datos recogidos del formulario del código anterior se desglosan y, posteriormente, se envían a Cartier-IA para que así pueda recibir las imágenes DICOM correspondientes y, devolviendo a la plantilla

HTML inicial el número de imágenes que ha recibido esta plataforma con los datos recogidos del formulario.

```
@login_required(login_url='/login/')
def post_info_images(request):

    if request.user.is_authenticated and request.user.user_role >= 3:
        request_user = request.user
        user_role = request_user.user_role
        id_patient = request.POST.get('id_patient')
        name_project = request.POST.get('name_project')
        id_project = request.POST.get('id_project')
        name_study = request.POST.get('name_study')

        ip_to_connect = request.POST.get('ip_to_connect')
        port_to_connect = request.POST.get('port_to_connect')
        dicom_code = request.POST.get('dicom_code')

        num_imgs = 0

        prefix = "http://"
        port_joiner = ":"
        if port_to_connect == "443":
            port_to_connect = ""
            port_joiner = ""
            prefix = "https://"

        url_post = prefix + ip_to_connect + port_joiner + port_to_connect +
        "/postKoopaa/"
        print(port_to_connect)
        print(url_post)
        response = requests.post(
            url_post,
            data={'id_patient': id_patient,
                  'name_project': name_project,
                  'id_project': id_project,
                  'name_study': name_study},
            json=None,
        )
        responseJson = response.json()
        y = json.loads(responseJson)
        for id in y["files"]:
            num_imgs = num_imgs + 1
            responseFilename = requests.post(
                prefix + ip_to_connect + port_joiner + port_to_connect +
                "/postFilename/",
                data={'index': id},
                json=None,
            )
            filename = responseFilename.text
            print("Filename -> ", filename)

            project = Project.objects.get(id=dicom_code)
            new_file = File(uploaded_by=request.user, name=filename, project=project)
            new_file.save()

            file_path = Path('DATA/' + dicom_code + '/' + filename)
            new_file.file = file_path
            try:
                os.mkdir(os.path.join(BASE_DIR, 'DATA/{0}'.format(dicom_code)))
            except:
                pass
            file = open(file_path, 'w')
            file.close()

            #POST PARA PEDIR EL CONTENIDO DEL FICHERO
            responseContent = requests.post(
                prefix + ip_to_connect + port_joiner + port_to_connect + "/postImage/",
                data={'filename': filename},
                json=None,
```

```

)
content = responseContent.content
try:
    os.mkdir(os.path.join(BASE_DIR, 'DATA/{}'.format(dicom_code)))
except:
    pass
file_path = Path('DATA/' + dicom_code + '/' + filename)
with open(file_path, 'wb+') as f:
    f.write(content)

return HttpResponse(num_imgs)

```

FIGURA 39. FRAGMENTO DE CÓDIGO PARA REALIZAR LA CONEXIÓN ENTRE KOOPAML Y PACS CONFIGURADOS (CARTIERIA)

Llegando así a una conexión satisfactoria entre KoopaML y los diferentes PACS que los usuarios configuren en los diferentes proyectos y, un posterior almacenamiento de las imágenes DICOM que el usuario ha querido seleccionar para una posterior gestión de estas.

En resumen, en la *Figura 40* se puede observar la comunicación entre los sistemas de una forma más esquematizada.

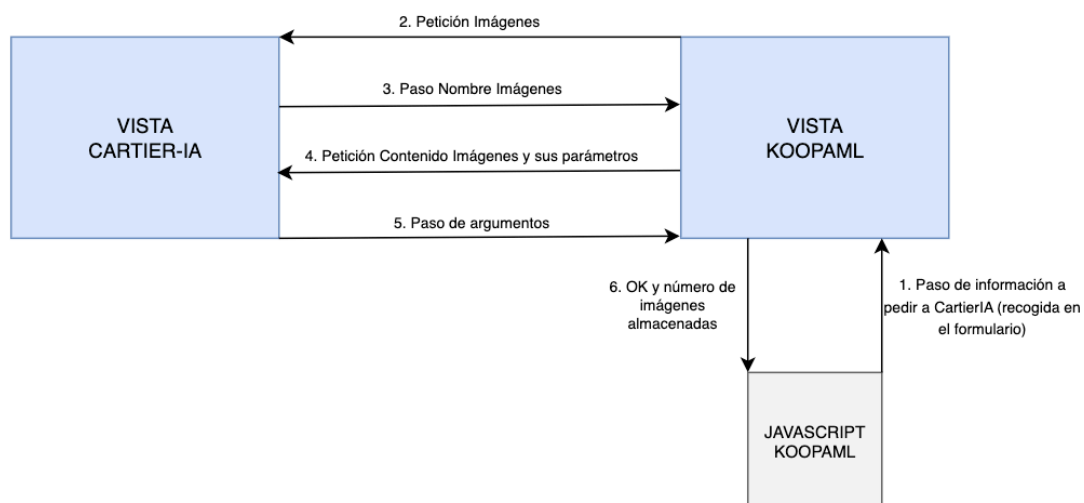


FIGURA 40. ESQUEMA DE COMUNICACIÓN ENTRE SISTEMAS

6.4.3. GESTIÓN Y MANEJO DE IMÁGENES DICOM

Como ya se ha comentado, una de las motivaciones que llevo a aplicar esta funcionalidad a la herramienta fue unificar datos de diferentes fuentes y organizarlos en una estructura más amigable.

Una vez almacenadas las imágenes DICOM recogidas de los diferentes PACS configurados en cada uno de los proyectos del usuario se podrán visualizar en un editor conjunto a sus características. Como la plataforma se organiza principalmente a través de proyectos, que contendrán datos de diferentes pacientes, que a su vez contendrán diferentes estudios a los que finalmente pertenecerán las imágenes DICOM. Estos pacientes o estudios son datos obtenidos de los diferentes PACS y, que, posteriormente se han almacenado en la herramienta actual, es decir, en KoopaML, obteniendo así las características y argumentos de una imagen DICOM como se puede observar en la *Figura 41*.



FIGURA 41. DETALLES DEL FICHERO DE UNA IMAGEN DICOM

Otra funcionalidad principal de la plataforma es su editor de imágenes. Cuando se realiza la carga de imágenes DICOM, los usuarios pueden explorarlas más de cerca a través de esta herramienta.

Uno de los beneficios de este editor de imágenes es que está totalmente integrado dentro de la plataforma, por lo que no es necesario utilizar herramientas externas para realizar modificaciones de imagen. Esto es posible porque la edición de imágenes tiene lugar a través de la biblioteca *Cornerstone.js* [38], una biblioteca de código abierto para analizar y renderizar imágenes DICOM.

Gracias a este enfoque, los usuarios pueden editar las imágenes que están actualmente almacenadas y decidir más adelante si se pueden hacer anotaciones y modificaciones persistentes. Estas modificaciones no se almacenan a lo largo de la imagen en sí, si no como objetos JSON que contienen todos los metadatos necesarios con respecto a las modificaciones o anotaciones realizadas.

La mayoría de las herramientas y funcionalidades de edición de imágenes son compatibles por otro marco de código abierto, que proporciona una extensible solución para crear herramientas sobre *Cornerstone.js* [38]. Esta biblioteca se trata no de una aplicación completa en sí misma, si no un componente que se puede usar como parte de aplicaciones más grandes y complejas. El objetivo es proporcionar flexibilidad con respecto al contenedor y el transporte, pero, no resulta nada sencilla su implementación debido a que hay que tener en cuenta varios aspectos de las imágenes DICOM para que la biblioteca llegue a funcionar correctamente como son la codificación o la modalidad de la imagen. Por otra parte, *cornerstone Core* es independiente del paradigma de interacción exacto que se usa, es decir, no incluye enlaces del ratón, táctiles o de teclado para manipular las diversas propiedades de la imagen, como la escala o la traducción.

En concreto, las herramientas que se encuentran disponibles a través del editor de imágenes y en las que se ha hecho uso de la biblioteca comentada son (*Figura 42*):



FIGURA 42. HERRAMIENTAS DE EDICIÓN DE IMÁGENES DICOM

- Herramientas de pincel y recorte para la segmentación de imágenes.
- Capas de segmentación y selectores de tamaño de pincel para facilitar el proceso de segmentación de las imágenes.
- Herramientas de longitud y área para medir fragmentos de imágenes.
- Herramientas de anotación.
- Herramientas de Zoom.
- Una herramienta de recorte (calculada en el *back-end*).
- Una herramienta para aplicar algoritmos de Inteligencia Artificial y validarlos (computado en el *back-end*).

6.4.4. GESTIÓN DE ALGORITMOS DE INTELIGENCIA ARTIFICIAL

La principal característica en la que se centra este punto es la integración de la Inteligencia Artificial dentro de la plataforma. Esta característica tiene dos motivaciones principales:

- Ofrecer los beneficios de los algoritmos de Inteligencia Artificial sin necesidad de salir de la plataforma para aplicarlos.
- Proporcionar una interfaz amigable para aplicar algoritmos de Inteligencia Artificial y abrir sus usos para usuarios no expertos.

Esta función permite a los investigadores cargar sus scripts de Inteligencia Artificial en la plataforma y ponerlos a disposición de otros usuarios. Únicamente los investigadores con privilegios podrán agregar nuevos algoritmos, que deben ser minuciosamente probados por el investigador que los cargue en la plataforma antes de integrarlos para llegar a asegurar de una funcionalidad confiable ya que dejar la funcionalidad de cargar algoritmos de Inteligencia Artificial a cualquiera de los usuarios puede llegar a ser un agujero de seguridad, por ello, solo una cantidad mínima de usuarios se encuentran con esa capacidad, además deben ser Expertos en *Machine Learning*. Por otra parte, la plataforma también proporciona un cargador para definir el algoritmo de metadatos. En este caso, los metadatos del algoritmo son muy importantes para integrar adecuadamente los algoritmos dentro de la plataforma debido a que proporcionan información sobre la salida del algoritmo (una imagen modificada, un conjunto de medidas, una máscara de segmentación, etc.), su aplicación (ya que su aplicación puede estar limitada a modalidades DICOM específicas) u otros parámetros dependiendo de la salida.

Es importante aclarar que los algoritmos necesitan ser entrenados previamente (acción que el desarrollador actual no ha realizado debido a que no formaban parte de sus objetivos del proyecto, se han obtenido algoritmos que ya estaban entrenados con anterioridad).

En resumen, para integrar un algoritmo en la plataforma es necesario proporcionar el modelo entrenado anteriormente y la secuencia de comandos que hace uso de este modelo con el objetivo de permitir su invocación por parte del algoritmo de Inteligencia Artificial de la plataforma.

Una vez que se ha integrado un algoritmo en la plataforma, estará disponible un botón en el editor de las imágenes para así aplicar un algoritmo seleccionado entre los disponibles para la imagen actual. Cuando el usuario confirma la solicitud, la plataforma producirá el resultado que, dependiendo del tipo de salida del algoritmo, podría resultar en la visualización de una nueva imagen.



FIGURA 43. APLICACIÓN DE DIFERENTES ALGORITMOS A UNA IMAGEN DICOM



FIGURA 44. RESULTADO DE APLICAR UN ALGORITMO DE INTELIGENCIA ARTIFICIAL A UNA IMAGEN DICOM

En el siguiente fragmento de código, se puede observar la función utilizada para la aplicación de los algoritmos de Inteligencia Artificial, obteniendo así el algoritmo seleccionado por el usuario y la imagen a la que aplicarlo y, devolviendo, la imagen resultante de la aplicación, esta imagen es guardada como una nueva imagen en el proyecto bajo el nombre “IDAlgoritmo-segmented-nombreimagen.dcm”.

```
@login_required(login_url='/login/')
def apply_algorithm(request, ai_id, dicom_code, file_id):

    if request.user.is_authenticated:
        request_user = request.user

        user_role = request_user.user_role

        if user_role != 4:
            user_projects = request.user.d_projects.all()
            if Project.objects.get(id=dicom_code) not in user_projects:
                return JsonResponse(
                    _({'Código': '204', _('Mensaje'): _("No tienes permiso para
acceder a este proyecto")}))

            algorithm = Algorithm.objects.get(id=ai_id)
            print(algorithm.name)
            file = File.objects.get(id=file_id)

            print(file.name)
            print("Start")

            if algorithm.output_type == 1:
                file_ai_name = ai_id + "-segmented-" + file.name.lower()
                output_path = os.path.join(BASE_DIR, 'DATA/{0}/{1}'.format(dicom_code,
file_ai_name))
                if not File.objects.filter(name=file_ai_name.lower()).exists():
                    path =
'algorithms_management.scripts.{0}.{0}.py'.format(algorithm.name)
                    module =
importlib.import_module('algorithms_management.scripts.{0}.{0}'.format(algorithm.name))
                    file_path = Path('DATA/' + dicom_code + '/' + file.name)
                    module.main([algorithm.script_path, str(file_path), output_path,
algorithm.weights_path])

                    f = file
                    f.pk = None
                    f.name = file_ai_name
                    f.save()

                    url = '/dicom-project/{0}/viewer/get-file/{1}'.format(dicom_code,
f.id)
                else:
                    f = File.objects.get(name=file_ai_name.lower())
                    url = '/dicom-project/{0}/viewer/get-file/{1}'.format(dicom_code,
f.id)

                response = JsonResponse({'url': url, 'output_type': 'image'},
content_type='application/json')

                print("Ok.")
                return response
            else:
                return redirect('/')
```

FIGURA 45. FRAGMENTO DE CÓDIGO PARA LA APLICACIÓN DE UN ALGORITMO DE INTELIGENCIA ARTIFICIAL A UNA IMAGEN DICOM

Por último, si se quisiese aprender a manejar la aplicación desarrollada, el desarrollador ha creado un manual de usuario que, en el [Anexo VI. Manual del usuario](#) adjunto con este documento, se podrá observar dicho manual.

6.5. DESPLIEGUE DE LA APLICACIÓN

Para el despliegue de la aplicación en un servidor se ha hecho uso del servidor *Gunicorn* [39], el servidor proxy *Nginx* [40], un entorno virtual y una base de datos.

Al tratarse de una aplicación que ya se encontraba desplegada, no se ha tenido que realizar una configuración inicial de los servidores. Por lo tanto, para desplegar la aplicación en el servidor creado se han debido de seguir los siguientes pasos:

1. Se crea una carpeta en el servidor llamada “*KoopaML*” junto con un entorno virtual vacío, el cual se activa. También se crea un archivo llamado “*requirements.txt*” con los requisitos para el correcto funcionamiento de la aplicación. Estos requisitos se obtienen de la salida del comando

\$ pip freeze

Algunos de los requisitos de esta aplicación son los mostrados en la *Figura 46*

```
1  asgiref==3.4.1
2  boto3==1.19.3
3  botocore==1.22.3
4  certifi==2021.10.8
5  charset-normalizer==2.0.10
6  Django==3.2.7
7  django-crispy-forms==1.13.0
8  docutils==0.17.1
9  EditorConfig==0.12.3
10 idna==3.3
11 Jinja2==3.0.3
12 jmespath==0.10.0
13 joblib==1.1.0
14 #jsbeautifier==1.14.6
15 jsonfield==3.1.0
16 lockfile==0.12.2
17 luigi==2.8.0
18 MarkupSafe==2.0.1
19 numpy==1.19.5
20 pandas==1.1.5
21 pycopg2==2.9.1
22 python-daemon==2.1.2
23 python-dateutil==2.8.2
24 pytz==2021.1
25 requests==2.27.1
26 s3transfer==0.5.0
27 scikit-learn==0.24.2
28 sciluigi==0.9.7
29 scipy==1.5.4
30 six==1.16.0
31 sklearn==0.0
32 sqlparse==0.4.2
33 tenacity==6.3.1
34 threadpoolctl==3.0.0
35 tornado==4.5.3
36 typing-extensions==3.10.0.2
```

FIGURA 46. REQUISITOS DE LA APLICACIÓN A LA HORA DE INSTALAR

2. Una vez generados, se instalan los requisitos con el siguiente comando:

\$ pip install -r requirements.txt

3. En este paso es necesario el Código Fuente de la aplicación, gracias a la utilización del sistema de controles Git es posible el acceso y la descarga de cualquier versión del código subida por cualquier miembro del equipo. Por tanto, a través del enlace que referencia al código fuente a utilizar se utiliza el comando “git clone” y se descarga el código fuente.

Es necesario realizar algunas modificaciones a este código para su correcto despliegue en el servidor como, por ejemplo, el nombre de la base de datos para que coincida con el nombre de la ya creada.

4. Por último, se debe crear un nuevo super usuario de la aplicación con el comando

create superuser

7. CONCLUSIONES

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

7.1. CONCLUSIONES

Tal y como se ha descrito a lo largo del trabajo, se ha incluido un enfoque de diseño centrado en el usuario, en el que, tras varias iteraciones, definiciones de la audiencia y creación de prototipos digitales, se ha obtenido como resultado una interfaz clara, sencilla y descrita por los usuarios de la audiencia objetivo como intuitiva. De esta forma se ha logrado crear una aplicación que ofrece soporte a usuarios a la hora de juntar herramientas en un mismo entorno, proceso que el desarrollar le ha tenido que dedicar más tiempo debido a no ser nada sencillo el tratar ficheros médicos y, comunicar dos herramientas para un posterior almacenamiento.

La oportunidad de haber podido trabajar en un proyecto relacionado con el sector sanitario ha resultado totalmente satisfactoria puesto que este campo es el elegido por el desarrollador para continuar creciendo como profesional.

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

Por otra parte, durante el proceso de creación del sistema 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 no seguir el diseño creado por la autora inicial si no, al dividir por un lado los proyectos de *Machine Learning* y, por otro, los proyectos DICOM, se realizó un diseño diferente, pudiendo así aportar los conocimientos adquiridos durante el grado en cuanto a la parte de diseño de interfaces.

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

La búsqueda de información teórica sobre las imágenes DICOM, los algoritmos de Inteligencia Artificial y las entidades PACS sirvió para ampliar el conocimiento propio del desarrollador sobre estos conceptos, es decir, antes de intentar enseñar algo debes de conocerlo y saberlo tú.

La realización de este gran proyecto ha permitido aplicar los conocimientos que se han adquirido durante el grado. Los más útiles y que han servido como base del desarrollo han sido las relaciones con Ingeniería de Software, Interacción Persona-Ordenador, Gestión de Proyectos y las diferentes asignaturas asociadas a Programación. Sin embargo, todas las demás asignaturas han servido en algún modo a ayudar al desarrollador a pensar o tomar nuevos conceptos.

Se agradece el poder haber podido aprender un nuevo lenguaje de programación que para el desarrollador era totalmente desconocido como es Python y, a su vez, haber obtenido una mayor soltura en los lenguajes dedicados al diseño de interfaces web, es decir, HTML, CSS y JavaScript.

7.2. POSIBLES 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 se 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 llegar a trabajar en un futuro:

- **Automatización de envío y recepción de imágenes.** El proceso de envío y recepción de las imágenes se puede automatizar, de esta forma, cada vez que se añada una imagen en el PACS configurado que tenga las mismas propiedades que las buscadas y almacenadas en uno de los proyectos de KoopaML, esta imagen DICOM se va a añadir automáticamente al proyecto.
- **Mejoras en la interfaz a la hora de búsquedas de imágenes.** Es decir, se podrían realizar modificaciones en la interfaz y seleccionar las imágenes DICOM que el usuario quiera y no añadiendo a KoopaML todas las imágenes con los parámetros buscados.
- **Recomendaciones de Algoritmos.** El sistema sugerirá que algoritmos de Inteligencia Artificial aplicar a las diferentes imágenes pudiendo aceptar la recomendación de tal manera que el sistema muestre y aplique el mejor algoritmo que se encuentra almacenado en el sistema para dicha imagen.
- **Incorporación de más idiomas a la aplicación.** De esta forma, aunque el idioma internacional sea el inglés, se podría comercializar de una mejor manera la aplicación a otros países.
- Permitir a los usuarios compartir entre ellos los diferentes proyectos que desee desde la propia aplicación.

8. Bibliografía

- [1] G. Litjens, «A survey on Deep Learning in medical image analysis,» de *Med Image Anal*, 2017, pp. 60-88.
- [2] S. González Izard, R. Sánchez Torres, Ó. Alonso Plaza, J. A. J. Méndez y F. J. García Peñalvo, «Nextmed: Automatic Imaging Segmentation, 3D Reconstruction, and 3D Model Visualization Platform Using Augmented and Virtual Reality,» de *Sensors (Basel)*, vol. 20, n^o10, p. 50.
- [3] S. González Izard, J. A. Juanes, F. J. García Peñalvo, J. M. Gonsalves Estella, M. J. Sánchez Ledesma y P. Ruisoto, «Virtual Reality as an Educational and Training Tool for Medicine,» de *Journal of Medical Systems*, p. 50.
- [4] X. Ferrer, T. van Nuenen, J. M. Such , N. Criado y M. Coté, «Bias and Discrimination in AI: a cross-disciplinary perspective,» de *IEEE Technology and Society Magazine*, pp. 72-80.
- [5] S. Hoffman, «Emerging Hazard of AI-Related Health Care Discrimination,» de *Hasting Center Report*, vol 51, no 1, 2021, pp. 8-9.
- [6] S. Wachter, B. Mittelstadt y C. Russell, «Why fairness cannot be automated: Bridging the gap between EU non-discrimination law and AI,» de *Computer Law & Security Review*, 2021, pp. 105-567.
- [7] J. Alonso Sánchez, Trabajo Fin de Grado: Plataforma para dar soporte a la aplicación de Machine Learning en el conexto médico, Salamanca: Universidad de Salamanca, 2021.
- [8] P. Mildemberger, M. Eichelberg y E. Martin, «Introduction of DICOM standard,» de *Introduction of DICOM standard*, European Radiology 12, 2012, pp. 920-927.
- [9] F.J. García Peñalvo, J.M. Maudes Radeo, M.G. Piattini Velthius, J.R. García-Bermejo Giner y M.N. Moreno García , *Proyecto Final De Carrera en la Ingeniería Técnica en Informática: Guía de Realización y Documentación*, Salamanca, 2000.

- [10] A. Vázquez-Ingelmo, A. García-Holgado, F.J. García-Peñalvo y R. Therón, *Proof-of-concept of an information visualization classification approach based on their fine-grained features*. *Expert Systems*, 2021.
- [11] F.J. García-Peñalvo, A. Vázquez-Ingelmo y R. Therón, *Towards a Technological Ecosystem to Provide Information Dashboards as a Service: A Dynamic Proposal for Supplying Dashboards Adapted to Specific Scenarios*, *Applied Sciences* 11, 2021.
- [12] A. Vázquez-Ingelmo, F.J. García Peñalvo, D. Amo Filva, D. Fonseca Escudero y R. Theron, «Connecting domain-specific features to source code: towards the automatization of dashboard generation,» *Cluster Computing* 23, pp. 1803-1816, 2020.
- [13] A. Vazquez Ingelmo, F.J. Garca Penalvo y R. Theron, Taking advantage of the software product line paradigm to generate customized user interfaces for decision-making processes: a case study on university employability., *PeerJ Computer Science* 5, 2019.
- [14] A. Vazquez Ingelmo, A. Garca Holgado, F. J. Garca Penalvo, R. Fraile Sanchon, E. Andres Fraile, P. Perez Sanchez, P. Antunez-Muinos, A. Sanchez Puente, V. Vicente Palacios y P. I. Dorado Dıaz, «KoopamL tests and improvements: A platform to facilitate the development of Machine Learning projects in the healthcare sector,» 2022.
- [15] J. C. Urquiaga, «¿Por que aprender Python? Retrieved por DevCode,» n.d.. [En lınea]. Available: <https://devcode.la/blog/por-que-aprender-python/>.
- [16] GRAPH, «GRAPH,» [En lınea]. Available: <https://www.grapheverywhere.com/machine-learning-en-python/>.
- [17] Mdn Web Docs, «Django introduction. Retrieved from developer mozilla,» [En lınea]. Available: <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>. [Ultimo acceso: 1 Diciembre 2022].
- [18] M. Atlas, «MongoDB Atlas. Retrieved from MongoDB Atlas,» [En lınea]. Available:

https://www.mongodb.com/cloud/atlas/lp/try2?utm_source=google&utm_campaign=gs_emea_spain_search_core_brand_atlas_desktop&utm_term=mongo%20db&utm_medium=cpc_paid_search&utm_ad=e&utm_ad_campaign_id=12212624563&gclid=CjwKCAjwoNuGBhA8EiwAFxomAyJbn_ERAbRVjS

[Último acceso: 1 Diciembre 2022].

- [19] T. Bacinger, «What is Bootstrap? A Short Bootstrap Tutorial on the What, Why, and How. Retrieved from Developers,» [En línea]. Available: <https://www.toptal.com/front-end/what-is-bootstrap-a-short-tutorial-on-the-what-why-and-how>. [Último acceso: 1 Diciembre 2022].
- [20] Wikipedia, «GitLab. Retrieved from Wikipedia,» 29 Julio 2019. [En línea]. Available: <https://es.wikipedia.org/wiki/GitLab>.
- [21] A. Vázquez Ingelmo y R. Fraile Sanchón, «Código en Gitlab de Koopa-ML-Dev,» [En línea]. Available: <https://gitlab.grial.eu/AndreaVazquez/koopa-ml-dev>.
- [22] «PostgreSQL,» [En línea]. Available: <https://www.postgresql.org/>.
- [23] A. Vázquez Ingelmo, «Cartier-IA,» Marzo 2021. [En línea]. Available: <https://cartier2.grial.eu/login/?next=/>.
- [24] F.J. García Peñalvo , A. García Holgado y A. Vázquez Ingelmo, *Transparencias Ingeniería del Software I, Tema 7: Proceso Unificado*, Universidad de Salamanca, 2020.
- [25] E.V.A. UCI, «Conferencia 1 "Introducción a la Ingeniería del Software,» Ecured, [En línea]. Available: https://www.ecured.cu/Proceso_unificado_de_desarrollo.
- [26] F.J. García Peñalvo, A. García Holgado y A. Vázquez Ingelmo, *Transparencias de Ingeniería del Software I. Tema 8: UML. Fundamentos de la vista de casos de uso*, Departamento de Informática y Automática, Universidad de Salamanca, 2020.
- [27] F.J. García Peñalvo , *Fundamentos de la vista de casos de uso*, Salamanca, Castilla y León, España, 2017.

- [28] F.J. García Peñalvo, A. Garcia Holgado y A. Vázquez Ingelmo, *Transparencias de Ingeniería del Software I. Tema 4, Ingeniería de Requisitos*, Departamento de Ingormática y Automática, Universidad de Salamanca, 2020.
- [29] S. Qiany, «A Personas Guideline, From What They Are to How To US,» 23 Septiembre 2020. [En línea]. Available: <https://uxdesign.cc/while-we-are-talking-about-personas-what-exactly-are-we-talking-525a645eb61a>.
- [30] Interaction Design Foundation, «User Scenarios,» [En línea]. Available: <https://www.interaction-design.org/literature/topics/user-scenarios>.
- [31] R. Friis Dam y T. Yu Siang, «Personas – A Simple Introduction,» [En línea]. Available: <https://www.interaction-design.org/literature/article/personas-why-and-how-you-should-use-them>.
- [32] L. Nielsen, «The Encyclopedia of Human- Computer Interaction, 2nd,» Interaction Design Foundation, 2010, p. Chapter 30. Personas.
- [33] Justinmind, «How to design user scenarios: best practices and examples,» 5 Marzo 2020. [En línea]. Available: <https://www.justinmind.com/blog/how-to-design-user-scenarios/>.
- [34] M. Varela Ruiz y A. Hamui-Sutton, «La técnica de grupos focales. Investigación en Educación Médica,» [En línea]. Available: <https://www.elsevier.es/es-revista-investigacion-educacion-medica-343-articulo-la-tecnica-grupos-focales-S2007505713726838>.
- [35] M. Amezcu, «12 Golden Rules for conducting a Focus Group. Granada, Andalucía, España,» Octubre 2016. [En línea]. Available: https://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1132-12962016000300016.
- [36] X. Guiteras Vila, «Consejos de redactar cuestionario,» 2012. [En línea]. Available: <https://www.investigacionmercados.es/consejos-para-redactar-cuestionarios/>.

- [37] F.J. García Peñalvo ,A. Vázquez Ingelmo, A. García Holgado, J. Sampedro Gómez, A. Sánchez Puente, V. Vicente Palacios Víctor, P.I. Dorado Díaz y P.L. Sánchez , «Application of Artificial Intelligence Algorithms Within the Medical Context for Non-Specialized Users: the CARTIER-IA Platform,» 2021.
- [38] I. Octaviano, «CornerstoneJS,» 19 Noviembre 2021. [En línea]. Available: <https://github.com/cornerstonejs/cornerstone>.
- [39] Gunicorn, «Gunicorn,» [En línea]. Available: <https://gunicorn.org/>.
- [40] Nginx, «Nginx,» [En línea]. Available: <https://www.nginx.com/>.