



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**título del TFG
Documentación Técnica**



Presentado por nombre alumno
en Universidad de Burgos — 15 de mayo
de 2023

Tutor: nombre tutor

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	iv
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	9
Apéndice B Especificación de Requisitos	17
B.1. Introducción	17
B.2. Objetivos generales	17
B.3. Catalogo de requisitos	17
B.4. Especificación de requisitos	17
Apéndice C Especificación de diseño	19
C.1. Introducción	19
C.2. Diseño de datos	19
C.3. Diseño procedimental	19
C.4. Diseño arquitectónico	19
Apéndice D Documentación técnica de programación	21
D.1. Introducción	21
D.2. Estructura de directorios	21
D.3. Manual del programador	21

D.4. Compilación, instalación y ejecución del proyecto	21
D.5. Pruebas del sistema	21
Apéndice E Documentación de usuario	23
E.1. Introducción	23
E.2. Requisitos de usuarios	23
E.3. Instalación	23
E.4. Manual del usuario	23

Índice de figuras

A.1. Sprint 1.	3
A.2. Sprint 2.	4
A.3. Sprint 3.	5
A.4. Sprint 4.	6
A.5. Sprint 5.	7
A.6. Sprint 6.	8
A.7. Sprint 7.	9

Índice de tablas

A.1. Costes de personal	10
A.2. Costes de Hardware	10
A.3. Costes de Software	10
A.4. Costes adicionales	11
A.5. Costes totales	11
A.6. Dependencias del proyecto Android	14
B.1. CU-1 Nombre del caso de uso.	18

Apéndice A

Plan de Proyecto Software

A.1. Introducción

A.2. Planificación temporal

Para la planificación temporal, cabe destacar que se ha seguido la metodología SCRUM, la cual no se ha podido seguir al 100 %, puesto que es una metodología planeada para equipos de trabajo, siendo este un proyecto personal de final de carrera.

- se programó el *sprint* de dos semanas, donde, se podrían haber hecho de una semana, pero la metodología SCRUM recomienda de dos a cuatro.
- Durante, el *sprint* se planean unas tareas a terminar en el periodo de tiempo.
- Cada tarea tiene un objetivo y una estimación; se organizan en un tablero canvas con las siguientes columnas consideradas:
 - "tareas a realizar": Son las tareas planeadas para el sprint actual o próximos.
 - "tareas en proceso": Son aquellas tareas que se están realizando en el momento.
 - "tareas terminadas": Son las tareas que ya se han terminado, las cuales se comprueban al final del sprint si se han terminado correctamente, en tal caso se cierran.

- "tareas cerradas": Son aquellas tareas que se han cerrado en *sprints* anteriores, las cuales se han implementado correctamente en el producto final.
- Con los gráficos *burndown*, se puede observar el proceso del proyecto de cada *sprint*
- Como se ha comentado, al terminar el sprint, se realizaba una reunión personal, donde se comprobaba si se habían completado correctamente las tareas terminadas. En caso de no ser así, se añadía para el siguiente *sprint*, además, se añadían las nuevas tareas a realizar en el siguiente.

Esta metodología se ha implementado usando la herramienta ZenHub, la cual permite hacer los apartados anteriores con facilidad. Para asignar los pesos de la tarea, se utiliza una aproximación de la secuencia de fibonacci. Donde los valores dependen del tiempo y dificultad que llevará la tarea, esta estimación, a lo largo del proyecto va siendo más precisa, puesto que al principio, no se controlaban los tiempos que conllevarían las dificultades del proyecto.

Los *sprints* que se realizaron se comentan a continuación.

Sprint 0 - Anterior al 27/12/2022

En este sprint inicial, del que no se poseen fechas, se realizaron diversas reuniones, en las cuales se explicaron en que iba a consistir el proyecto.

Durante este sprint, se estuvo investigando sobre aplicaciones Android con el mismo objetivo, sobre las que basar el proyecto. Encontrando de esta forma la aplicación Ret-iN CaM. La cual sirve para realizar estudios sobre los pacientes, de forma que se pueda ver la evolución. Además, se realizaron búsquedas sobre redes neuronales ya creadas las cuales añadir a la aplicación.

Como este sprint fue más un periodo de búsqueda de información, sin realizar ningún trabajo real, no se contabilizó el trabajo realizado.

Sprint 1 - 27/12/2022 - 10/01/2023

En este primer Sprint, se estuvo pensando con que herramienta realizar el proyecto. Teniendo como objetivo crear tanto el proyecto, como el LaTeX.

Para la aplicación móvil, se pensaba utilizar o Android Studio o Unity, de Android Studio ya se tenían conocimientos de uso, pero de Unity no, por

ello, se realizó un proyecto "Hola Mundo" en Unity, para ver la complejidad; al ver que la herramienta estaba más destinada a aplicaciones gráficas, se decidió realizar la aplicación en Android Studio, creando el proyecto.

Para la realización del documento, se pensó hacer de forma local, utilizando MiKTeX, pero por facilidad de uso de los tutores, quienes preferían el uso de OverLeaf, que les permitía corregir de forma concurrente el documento, se utilizó esta.

Como este primer sprint se estaba empezando, y además era Navidad, se olvidó mover las tareas de ZenHub a terminadas, y la gráfica que proporciona ZenHub es incorrecta.

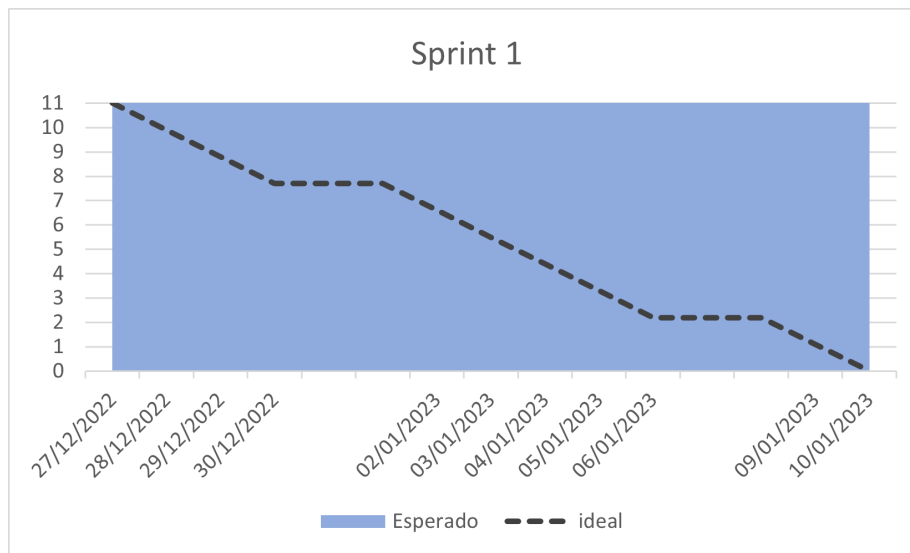


Figura A.1: Sprint 1.

Sprint 2 - 10/01/2023 - 24/01/2023

En el segundo Sprint se planeó hacer el menú principal de la aplicación, para ir haciendo la interfaz de usuario y a su vez, se hizo una investigación sobre que colores relaciona el ser humano con la retinopatía diabética. Para hacer el menú, primero se hizo un boceto a mano, por ese motivo, se puso un valor alto de la estimación. Por otro lado, se investigó sobre como implementar las redes neuronales en Android Studio, e ir añadiendo apartados al documento LaTeX.

De estas tareas, no se pudieron terminar ni la de investigar como implementar una red neuronal en Android Studio, ni añadir el apartado en

concreto del LaTeX. Aunque si que se avanzó, SCRUM no permite dividir la tarea en parte terminada y parte que no, por lo que se va a considerar que no se han realizado.

El gráfico burndown se vería incorrectamente, debido al inconveniente ocurrido en el primer sprint, arreglando este defecto, se vería de esta forma:

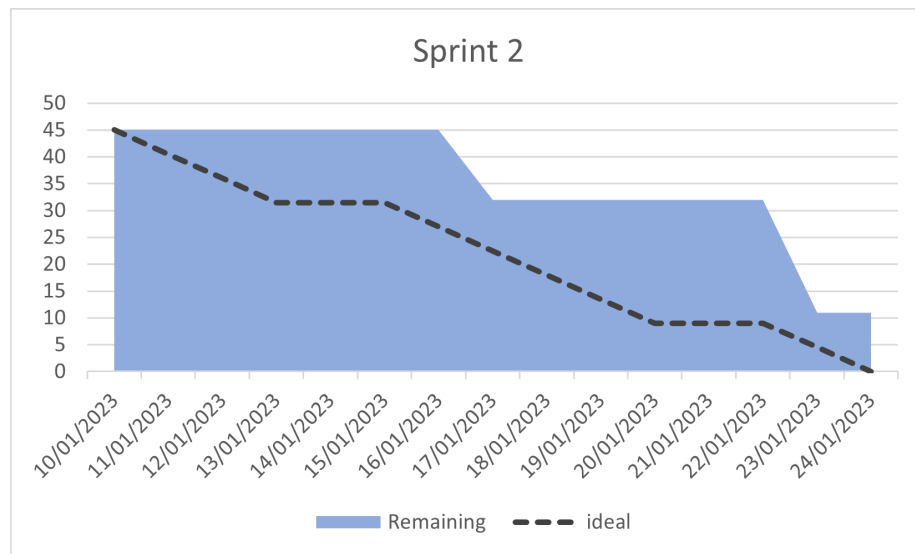


Figura A.2: Sprint 2.

Sprint 3 - 24/01/2023 - 07/02/2023

En el tercer Sprint, se añadieron las tareas no terminadas en el anterior, y se añadió también, el terminar el boceto inicial de la aplicación. Al iniciar el sprint, se pensó en que la aplicación tenía que tener nombre y un logo, pero se programó para futuros sprints porque ya se tenían varias tareas para este. Además, no se pudo avanzar mucho en la introducción del LaTeX porque era necesario aclarar una duda presencialmente, y la reunión se hizo a finales del sprint.

Como se iba avanzando en las tareas simultáneamente, tanto terminar el boceto de la aplicación y buscar información de la implementación, se terminaron el mismo día.

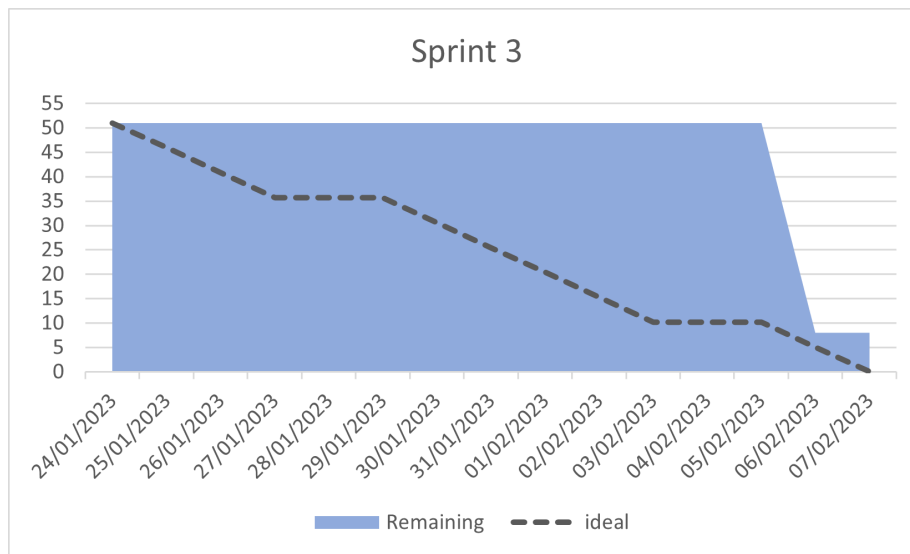


Figura A.3: Sprint 3.

Sprint 4 - 07/02/2023 - 21/02/2023

Para el cuarto sprint, se planeó realizar como una investigación sobre como se podría incluir la aplicación en la play store, y sobre como integrar tensorflow en Android Studio.

Al investigar sobre la play store, se dio cuenta que era necesario realizar un pago de 25\$, por ese mismo motivo, se preguntó a los tutores, quienes dijeron que no hacia falta subir la aplicación a ninguna tienda.

Por otro lado, se terminaron las demás tareas, cumpliendo los objetivos del sprint. Y aunque se podría haber eliminado la tarea de investigación de la play store, el tiempo gastado en la investigación se había realizado, por lo que no se elimino.

El burndown queda de la siguiente manera:

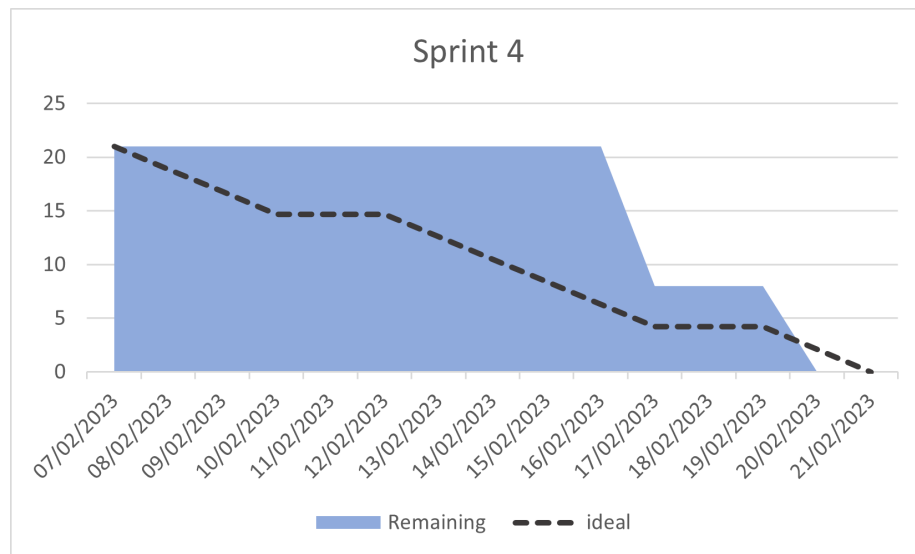


Figura A.4: Sprint 4.

Sprint 5 - 21/02/2023 - 07/03/2023

Para este sprint, como no había tareas atrasadas, se añadieron la creación del logo y el nombre de la aplicación, de la investigación anterior de como implementar una red neuronal en Android Studio, el caso concreto de una red neuronal .h5; realizar cambios en la interfaz para añadir y quitar funcionalidad. Y añadir una base de datos; para lo que se pensó realizar un conjunto de clases java, las cuales simulaban la base de datos, y con un constructor, inicializar los datos.

Debido a que no se me da muy bien en el diseño artístico, se me ocurrió la idea de utilizar la herramienta DALL · E2 para un diseño preliminar, una vez se proporcionó un resultado aceptable, se editó el icono eliminando ruido, y cambiando los colores.

Debido a las diversas actividades designadas en este sprint, no se pudo realizar la base de datos.

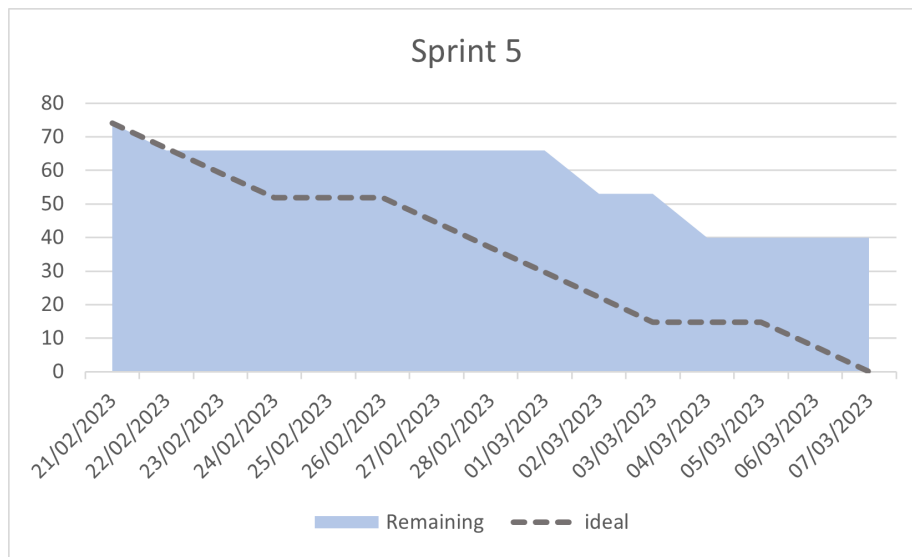


Figura A.5: Sprint 5.

Sprint 6 - 07/03/2023 - 21/03/2023

En el sexto sprint, se añadieron realizar el apartado de trabajos relacionados del documento LaTeX, y probar una integración de una red neuronal básica. Ambas tareas se terminaron, y se avanzó en la creación de la base de datos; pero no se terminó debido a que se querían añadir más datos, los cuales solo se crearon un médico, un paciente y un informe.

Sin contar que se podría haber considerado como la tarea terminada, el gráfico burndown, sería este:

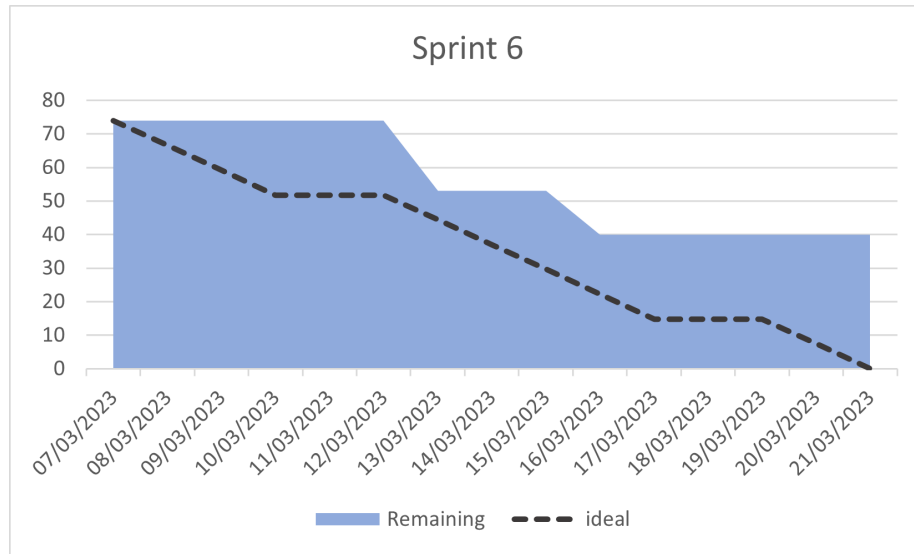


Figura A.6: Sprint 6.

Sprint 7 - 21/03/2023 - 04/04/2023

En este sprint, se termino la base de datos añadiendo otro par de pacientes y médicos, y se planearon hacer una eliminación de bugs, debido al modo oscuro, y que no guardaba bien datos entre las distintas actividades. Y también se pensó hacer el apartado "añadir técnicas y herramientas", pero se avanzó, sin poder acabar la tarea. Quedando para la siguiente actividad.

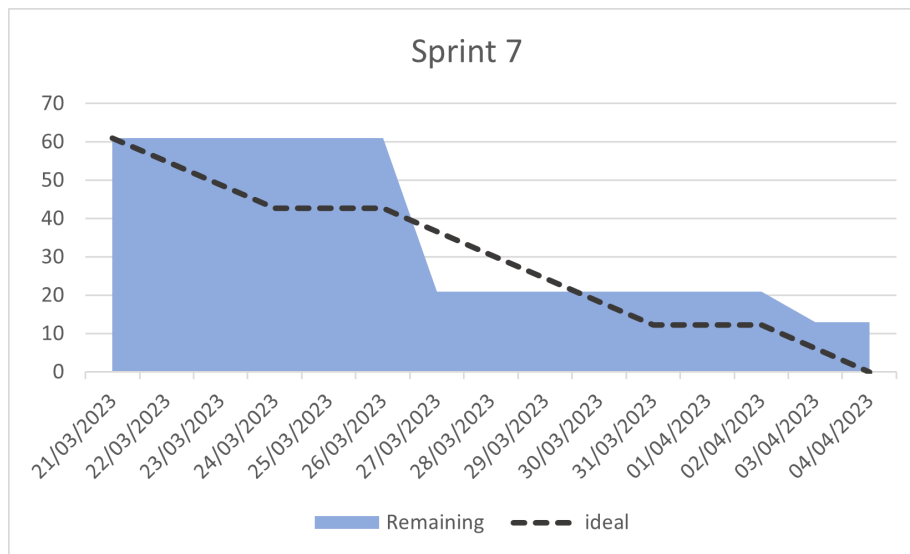


Figura A.7: Sprint 7.

A.3. Estudio de viabilidad

Este proyecto, al estar orientado a la sanidad pública, no busca obtener beneficios, sino a facilitar la labor del médico.

En caso de querer desarrollar este proyecto con una intención comercial.

Viabilidad económica

Costes

En personal

Para el desarrollo del proyecto, se ha habría contratado a un programador *junior* a tiempo parcial durante siete meses. Para la cotización de la seguridad social, se consideran los siguientes porcentajes: un 23,60 % debido a las contingencias comunes, debido al desempleo de tipo general, al ser un contrato de tiempo indefinido, es un 5,50 %; un 0,2 % para el Fondo de Garantía Salarial, y un 0,60 % para formación profesional. Sumando estos porcentajes, un 29,9 % de retribución a la seguridad social.

El coste se puede ver en la tabla: [A.1](#)

En *hardware*

Concepto	Coste
Salario mensual neto	973€
Tipo de retención del IRPF	11,69 %
Cuota a pagar del IRPF	195€
Seguridad Social (29.9 %)	498€
Salario mensual bruto	1,666€
Total 6 meses	9,996€

Tabla A.1: Costes de personal

Concepto	Coste	Amortización
Ordenador	1,200€	100€
Dispositivo móvil	300 €	25 €
Oftalmoscopio Portátil D-Eye	400€	33,33€
Total	1,900€	158,33€

Tabla A.2: Costes de Hardware

Concepto	Coste	Amortización
Windows 11 Pro	259€	129,5€
ZenHub para equipos (3 meses)	36€	18€
Total	295€	147,5€

Tabla A.3: Costes de Software

Para realizar el proyecto, se hará uso de un ordenador portátil y opcionalmente, un dispositivo móvil, para comprobar el correcto funcionamiento de la aplicación.

Suponiendo que los dispositivos hardware se amortizan en seis años, y se han utilizado durante siete meses:

El coste se puede ver en la tabla: [A.2](#)

En *software*

Se muestran aquellas licencias de *software* que no son gratuitas, y considerando que su amortización es de un año.

Concepto	Coste
Cuenta de desarrollador Play Store	25€
Internet	210€
Alquiler de oficina	800€
Total	1,035€

Tabla A.4: Costes adicionales

Concepto	Coste
En personal	9,996€
En hardware	1,900€
En software	295€
Adicionales	1,035€
Total	13,226€

Tabla A.5: Costes totales

El coste se puede ver en la tabla: [A.3](#)

Costes adicionales

Como se pretende comercializar la aplicación, la *Play Store* es una de las mejores plataformas de distribución de aplicaciones móviles. Pero, para publicar una aplicación, es necesario una cuenta de desarrollador, la cual cuesta 25€.

El coste se puede ver en la tabla: [A.4](#)

Costes totales

Siendo los costes totales, los vistos en: [A.4](#)

Beneficios

Como se ha comentado anteriormente, el proyecto tiene un objetivo altruista de ayudar en la labor del médico, dejando a un lado el objetivo de lucrarse con el proyecto. Hay varios focos en los que se podrían obtener beneficios:

- Vender la aplicación al servicio sanitario, con un beneficio del 30 %.
Dando la opción de comprar servicios de mantenimiento.

- Ofrecer licencias de la aplicación, cobrando cada año por mantener la suscripción.
- Introducir un anuncios en la aplicación.
- Distintas licencias de la aplicación, cada una con unas ventajas, con diferentes precios.

Entre las opciones anteriores, se ha descartado el uso de anuncios, principalmente, porque elimina profesionalidad a la aplicación; y también la opción de las distintas licencias, pero al ofrecer un único servicio, no se podía plantear esta opción.

Como las otras dos opciones se han considerado viables, se desarrollaran a continuación, además, es importante recalcar que los valores pueden cambiar debido a la gran cantidad de centro sanitarios públicos que hay, proporcionando ofertas según esa cantidad.

Vender la aplicación

Vendiendo la aplicación al servicio sanitario, solo se realizaría un pago por la compra, de forma que la empresa obtenga un 30 % de beneficios. De esta forma, 13,226€ sería el 70 % del precio y la aplicación se vendería por 18,894,28€; dando un beneficio de 5,668,28€.

También se podría ofrecer al servicio sanitario una opción de compra, donde la empresa se encarga de hacer el mantenimiento y actualizaciones de la aplicación. Siendo un servicio que depende del numero de centros sanitarios que haya en la comunidad autónoma. Un posible precio razonable, sería que por cada centro sanitario, anualmente se pague una tarifa de 10€.

Poniendo el caso de que el SACYL comprase la aplicación para el 25 % de los centros sanitarios, como tiene 10.218, debería pagar por $(10,218 \times 0,25 \times 10)$, siendo de esta forma anualmente se alcanzaría una cifra de 25,545€ al año. Como una persona cuesta cada 6 meses, 9,996€, al año serian 19,992€, teniendo beneficios anuales.

Licencias de la aplicación

Como el caso anterior, se va a hacer una oferta por cada centro sanitario, donde anualmente, se cobraría 100€.

Suponiendo que el SACYL pidiese una licencia para todos los centros, se haría el calculo de $(10,218 \times 0,25 \times 100)$ dando una cifra anual de 255,450€ en este precio ya estaría incluido el precio del mantenimiento.

En este caso, la empresa se encargaría de ofrecer los servicios al SACYL, se debería contratar un servidor con una base de datos, como el servidor guarda imágenes, se necesitaría bastante espacio, pongamos que se contrata un servidor Microsoft SQL Server de 1 GB, con un precio de 350€ al mes. Como 1GB pueden ser poco para Castilla y León, se considera que se compran 50 servidores. Añadiendo al coste anual, 4200×50 siendo así, 210,000€

Por tanto, con 255,450€ al año; habría que restar el coste de los servidores y el del personal donde se podría aumentar el número a 2 personas, $255,450 - 210,000 - 39,984$, haciendo que anualmente, se tengan 5,466€ de beneficio, como el coste de la aplicación era de 13,226€, harían falta 3 años para obtener un beneficio real sobre el proyecto.

Otros casos

Hay un caso que es importante destacar, y es el caso concreto de que el SACYL fuese la empresa encargada.

En este caso, no hay ningún beneficio monetario para la empresa, puesto que es sanidad pública. Pero, realizando el proyecto, a futuro se pueden evitar costes adicionales, como por ejemplo:

- Coste de transporte de la ambulancia.
- Costes de citas innecesarias y el tiempo que conllevan estas.
- Costes de material utilizado.

Viabilidad legal

Software

Esta sección aborda el contexto legal del proyecto, junto con las licencias *software* utilizadas.

Dependencia	Versión	Descripción	Licencia
Appcompat	1.6.1	Permite a versiones anteriores de Android una compatibilidad mejorada	Apache License 2.0
continúa en la página siguiente			

continúa desde la página anterior			
Dependencia	Versión	Descripción	Licencia
Material	1.8.0	Proporciona componentes de Material Design para Android.	Apache License 2.0
Constraint Layout	2.1.4	Permite crear diseños complejos y flexibles	Apache License 2.0
JUnit	4.13.2	Framework de pruebas unitarias en Java	Eclipse Public License
Ext JUnit	1.1.5	Biblioteca de pruebas unitarias para Android que extiende JUnit	Apache License 2.0
Espresso-core	3.5.1	Biblioteca de pruebas de interfaz de usuario	Apache License 2.0
TensorFlow Lite	2.7.0	Biblioteca de aprendizaje automático, más ligera que TensorFlow	Apache License 2.0
SQLite	2.2.0	Biblioteca que utiliza bases de datos SQLite, proporcionando una API para interactuar con la base de datos	No requiere

Tabla A.6: Dependencias del proyecto Android

Como el proyecto también ha tenido una parte realizada en python, se incluyen las dependencias utilizadas, donde solo se usa la biblioteca de tensorflow, ya comentada anteriormente.

Imágenes

Hay varios tipos de imágenes utilizadas en el proyecto, donde el logo, se ha utilizado usando DALLE2, la cual en sus políticas y términos se explica que el usuario tiene todos los derechos, tanto comerciales y no comerciales.

Por otro lado, para los iconos de la aplicación, se usaron tanto iconos por defecto de Android Studio, como iconos de Google Fonts, donde todas las imágenes son de código abierto y no tienen costo. Con una licencia Apache 2.0.

Y por ultimo, las imágenes de la retina,

Licencia del proyecto

Marco legal del proyecto

Es importante recalcar que al estar trabajando sobre una aplicación que usa datos delicados de las personas, siendo estos, además, datos relacionados con la salud.

RGPD

Según viene descrito en el artículo 9 del Reglamento General de Protección de Datos, esta prohibido hacer un tratamiento de categorías especiales de datos personales a menos que ocurra:

- El interesado ha dado el consentimiento explícito para el tratamiento de datos.
- Resulte necesario para proteger los intereses vitales del interesado u otra persona física cuando no este capacitado para dar su consentimiento.
- Es necesario para fines de medicina preventiva o laboral.
- Es necesario con fines de archivo en interés público, fines de investigación científica o histórica o fines estadísticos.

En el **artículo 9**, vienen más puntos, pero solo se explican los referentes al uso de la aplicación.

En el **capítulo 4** del Reglamento General de Protección de Datos, se trata sobre el responsable de tratar los datos y el encargado del tratamiento.

Por ello, con el fin de evitar que los datos sensibles se confundan entre los distintos pacientes, se ha implementado una interfaz, en la cual el médico introduce el DNI del paciente a ser tratado en ese momento, mostrando el nombre del paciente del DNI, en caso de haberse confundido a la hora de poner el DNI, se podrá dar cuenta al ver el nombre del paciente.

Apéndice B

Especificación de Requisitos

B.1. Introducción

Una muestra de cómo podría ser una tabla de casos de uso:

B.2. Objetivos generales

B.3. Catalogo de requisitos

B.4. Especificación de requisitos

CU-1	Ejemplo de caso de uso
Versión	1.0
Autor	Alumno
Requisitos asociados	RF-xx, RF-xx
Descripción	La descripción del CU
Precondición	Precondiciones (podría haber más de una)
Acciones	<ol style="list-style-type: none"> 1. Pasos del CU 2. Pasos del CU (añadir tantos como sean necesarios)
Postcondición	Postcondiciones (podría haber más de una)
Excepciones	Excepciones
Importancia	Alta o Media o Baja...

Tabla B.1: CU-1 Nombre del caso de uso.

Apéndice C

Especificación de diseño

- C.1. Introducción
- C.2. Diseño de datos
- C.3. Diseño procedimental
- C.4. Diseño arquitectónico

Apéndice D

Documentación técnica de programación

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución
del proyecto
- D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario