



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**
título del TFG



Presentado por Nombre del alumno
en Universidad de Burgos — 29 de mayo
de 2023

Tutor: nombre tutor



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. nombre tutor, profesor del departamento de nombre departamento, área de nombre área.

Expone:

Que el alumno D. Nombre del alumno, con DNI dni, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado título de TFG.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 29 de mayo de 2023

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor

Resumen

La retinopatía diabética es una de las mayores causas de ceguera en los países desarrollados. Para diagnosticar el grado de la anomalía, se realiza una imagen de la retina del paciente, captada por un retinógrafo. Esta prueba tiene un coste elevado para la sanidad pública, e implica a su vez que el paciente se desplace a un centro médico especializado.

Por ello, se han desarrollado unos dispositivos, para las cámaras de los móviles, que permiten obtener las fotografías de las retinas; de esta forma, el paciente no tendría que desplazarse a un hospital que disponga de un retinógrafo y el médico de familia le realizaría la foto al paciente.

En este trabajo, se propone facilitar la labor del médico, haciendo una aplicación Android a través de la cual se realizaría un primer diagnóstico del paciente a partir de una imagen que se enviará esta foto a modelos de aprendizaje computacional existentes; haciendo que los pacientes que tengan algún grado de retinopatía diabética, se realice un estudio exhaustivo.

Como resultado, se ha creado la aplicación RetinAI.

Descriptores

Salud, retinopatía diabética, redes neuronales, aplicación Android.

Abstract

Diabetic retinopathy is a leading cause of blindness in developed countries. To determine the degree of abnormality, the retinographer takes a image of patient's retina, but this procedure is costly for spanish public health system, also requires the patient to travel to a medical center.

To address this, there are some lenses which have been developed for mobile phone cameras to take retina photos. This way, the patient would not have to travel to a hospital that has a retinographer, and the family doctor would take the patient's photo.

This work proposes to make the doctor's labor easier by creating an Android app that would make a preliminary diagnosis of the patient based on the photo taken, this photo will be sent to existing computational learning models. This will allow patients with any degree of diabetic retinopathy to undergo a comprehensive study.

As a result, the app RetinAI has been created.

Keywords

Health, diabetic retinopathy, neural networks, Android app.

Índice general

Índice general	iii
Índice de figuras	v
Índice de tablas	vi
Introducción	1
1.1. Estructura	3
Objetivos del proyecto	5
2.1. Objetivos generales	5
2.2. Objetivos técnicos	5
2.3. Objetivos personales	6
Conceptos teóricos	7
3.1. Definiciones	7
3.2. Preprocesado	8
3.3. Formato de la red neuronal	9
Técnicas y herramientas	11
4.1. Metodologías	11
4.2. Herramientas	12
4.3. Patrones de diseño	14
Aspectos relevantes del desarrollo del proyecto	17
5.1. Inicio	17
5.2. Metodologías	17

Trabajos relacionados	19
6.1. Proyectos	19
6.2. Comparativa del proyecto	20
Conclusiones y Líneas de trabajo futuras	23
Bibliografía	25

Índice de figuras

Índice de tablas

6.1. Comparativa de las características de los proyectos.	20
---	----

Introducción

La discapacidad visual afecta a más de 2200 millones de personas en el mundo, siendo las principales causas de pérdida de visión: degeneración macular relacionada con la edad, cataratas, retinopatía diabética, glaucoma y errores de refracción no corregidos.[1]

Entre las personas con discapacidad visual, hay al menos 1000 millones de personas que tienen un deterioro moderado o grave de la visión.[1]

En Estados Unidos, cada año la retinopatía diabética suma un 12 % de nuevos casos de ceguera.

La retinopatía diabética es una complicación de la diabetes, que afecta al sistema ocular del ser humano. Es causada por la inflamación, escape o cierre de los vasos sanguíneos de la retina, pudiendo desarrollarse nuevos vasos sanguíneos a lo largo del tiempo.[12] Esta anomalía, encabeza las causas de ceguera en los países desarrollados. Según la Organización Mundial de la Salud, hasta 1 millón de personas tienen ceguera debido a la diabetes. [2] Hay varios grados de la patología de la retinopatía diabética entre los que se encontrarían por orden de menor a mayor peligrosidad:

- NPDR (Non-proliferative diabetic retinopathy), este grado se caracteriza por la ausencia de retinopatía;
- NPDR leve, donde los vasos sanguíneos empiezan a debilitarse, creando protuberancias llamadas micro-aneurismas;
- NPDR moderada, donde los vasos sanguíneos se siguen debilitando, se producen más hemorragias, pudiendo provocar visión borrosa;

- NPDR severa, en esta etapa, los vasos sanguíneos están dañados, causando falta de oxígeno en la retina y en la formación de nuevos vasos;
- PDR(proloferative diabetic retinopathy) o retinopatía diabética proliferativa, donde los vasos sanguíneos anormales que crecen en la retina y en el vítreo. Estos vasos pueden sangrar y provocar desprendimiento de retina, provocando la perdida de visión.

[8] En la actualidad, cuando un paciente acude a su médico de familia por perdida de visión, el médico determinaría si hacerle una cita para que el especialista le haga una imagen de la retina con el retinógrafo, para descartar la posibilidad de retinopatía diabética.

Este proceso, conlleva elevados costes para la sanidad publica, ya que implica un gran número de pruebas, junto con el número de tramites administrativos, en algunos casos, solo para descartar las personas que no tienen la enfermedad. Para el paciente supone un gran número de desplazamientos, que en algunos casos no se pueden permitir. Además de las horas de trabajo o lectivas que le supone al paciente.

Como solución a este problema, siguiendo con la tendencia de aplicar los avances informáticos a la medicina, se propone realizar una aplicación móvil, que permita al médico de familia hacer un estudio, haciendo una foto de la retina del paciente, utilizando una lente para el dispositivo móvil.

Además, para facilitar la labor del médico, esta aplicación móvil, proporciona redes neuronales convolucionales ya entrenadas, con las que se obtendría un primer análisis, enviando al especialista a aquellas personas que en cuyo resultado haya algún grado de retinopatía diabética.

De esta forma, al instalar la aplicación RetinAI creada con el objetivo de ser útil para el sistema sanitario, usando una interfaz sencilla para médicos sin experiencia previa en aplicaciones Android. Además, la aplicación facilita la creación de un informe médico ya que interpreta los resultados obtenidos de las imágenes, dando prioridad a aquellos casos que tengan más gravedad; y descartando aquellos casos que no tengan retinopatía diabética. Evitando una lentitud en el sistema sanitario debido a pacientes que no tienen esta complicación.

Permitiendo a los especialistas centrar su atención en aquellos pacientes que realmente tienen retinopatía diabética.

1.1. Estructura

La memoria se puede organizar en los siguientes apartados:

- **Introducción:** Apartado que dispone el tema y resume el contenido del trabajo.
- **Objetivos del proyecto:** Apartado donde se explican los objetivos que se buscan conseguir.
- **Conceptos teóricos:** Apartado del documento donde se exponen los conceptos claves del proyecto.
- **Técnicas y herramientas:** Apartado donde se explican las metodologías y herramientas utilizadas durante la realización del proyecto.
- **Aspectos relevantes:** Apartado donde se recogen los datos más importantes del desarrollo.
- **Trabajos relacionados:** Apartado donde se compara el trabajo realizado, con otros anteriores.
- **Conclusiones y líneas de trabajo futuras:** Apartado donde se expone de forma crítica el trabajo realizado, indicando posibles mejoras a realizar en un futuro.

Además, se adjunta como anexo los siguientes apartados:

- **Plan de proyecto:** Apartado donde se explica como se ha organizado el proyecto, y estudios sobre la rentabilidad tanto económica como legal tiene el proyecto.
- **Requisitos:** Apartado donde se explican los requisitos funcionales y no funcionales que tiene el proyecto.
- **Diseño:** Apartado donde se describe la fase de diseño del proyecto.
- **Manual del programador:** Apartado donde se recogen los aspectos que necesitaría un programador, tanto como para seguir el trabajo, como para poder entender el funcionamiento.
- **Manual del usuario:** Apartado donde se realiza una guía de usuario, con los pasos a seguir para un correcto funcionamiento de la aplicación.

Objetivos del proyecto

Los objetivos del proyecto se pueden dividir en 3 apartados, los cuales se verán a continuación.

2.1. Objetivos generales

- Desarrollar una aplicación Android, que permita realizar un estudio de la retinopatía diabética sobre los pacientes.
- Agilizar el sistema sanitario, haciendo que los médicos de familia puedan dar un diagnóstico preliminar a partir de los datos proporcionados.

2.2. Objetivos técnicos

- Desarrollar una aplicación Android con soporte API 21, siendo compatible con dispositivos Android 5.0 (Lollipop) y superiores.
- Hacer uso de Material 3 como fuente para que la aplicación sea más accesible al usuario.
- Hacer uso de Gradle para la automatización de la construcción de software.
- Hacer uso de GitHub como herramienta para alojar proyectos.
- Hacer uso de Git como herramienta de control de versiones.
- Hacer uso de la metodología SCRUM haciendo uso a su vez de la herramienta ZenHub.

- Hacer uso de GitKraken para hacer uso de una interfaz que facilite las acciones con el repositorio en GitHub.
- Hacer uso de pruebas, automáticas y manuales de forma que no se produzcan errores de ejecución.
- Hacer uso de Python para convertir un modelo keras con formato “.h5” a un modelo tensorflow Lite con formato “.tflite”.
- Hacer uso de keras, en Python, para realizar una red neuronal convolucional.
- Hacer uso de TensorFlow Lite, para poder proporcionar los resultados en una aplicación Android.

2.3. Objetivos personales

- Hacer uso de los conocimientos vistos durante la carrera.
- Aprender cómo desarrollar aplicaciones, en las que hay que implementar una red neuronal.
- Realizar una aplicación para dispositivos Android, que se pueda ejecutar en la mayoría de estos teléfonos, facilitando de esta forma la integración en el sistema sanitario.

Conceptos teóricos

En este proyecto, la mayor completitud del trabajo se encuentra en el preprocesado de las imágenes, el cual es necesario uno distinto según la red con la que se trabaje.

También el como se ha escogido la red neuronal para saber si una imagen tiene una buena calidad.

3.1. Definiciones

- Una red neuronal convolucional (CNN) es aquella que es una red neuronal en la que al menos hay una capa convolucional, normalmente consiste en la combinación de capas convolucionales, capas de agrupación y capas densas. [5]
- La capa convolucional es una capa de una red neuronal que consiste en realizar varias series de operaciones convolucionales, actuando cada una sobre un espacio de la matriz de entrada. [3]
- Una operación convolucional es aquella que a partir de una submatriz de la matriz de entrada, donde a veces es necesario realizar un filtrado, para ponderar los datos, calculando la suma de los valores de la submatriz, o en caso de que se haya realizado un filtro, del resultado de este, asignando el valor de la suma a una nueva matriz resultado con las dimensiones de la matriz de entrada.[4]
- Una operación convolucional es aquella que a partir de una submatriz de la matriz de entrada, donde a veces es necesario realizar un filtrado, para ponderar los datos, calculando la suma de los valores de la

submatriz, o en caso de que se haya realizado un filtro, del resultado de este, asignando el valor de la suma a una nueva matriz resultado con las dimensiones de la matriz de entrada.[4]

- La capa de agrupación es una capa de una red neuronal, es una capa que reduce el tamaño de la matriz de entrada, mediante operaciones como el máximo o la media de los valores de una submatriz.[7]
- La capa densa también llamada capa totalmente conectada, es una capa oculta, donde cada nodo esta conectado a todos los nodos de la siguiente capa oculta.[6]

3.2. Preprocesado

En el proyecto, se han realizado varios preprocesados, puesto que hay varias redes neuronales convolucionales.

Red neuronal convolucional VGG16, para la calidad de la imagen

En una red neuronal convolucional VGG16, el preprocesado necesario es que la imagen en vez de estar en formato RGB (Rojo verde y azul), tiene que ser formato BGR con los valores centralizados en 0. Además de este cambio, la imagen tiene que tener de 224 x 224 píxeles. [11] Para realizar este preprocesado en Python, se puede llamar a la función `tf.keras.applications.vgg16.preprocess_input`, para la aplicación de Android Studio, se debe realizar este preprocesamiento a mano.

Red neuronal convolucional ResNet50V2, para la detección de retinopatía

En la red neuronal convolucional ResNet50V2, el procesamiento es distinto al de VGG16, la gama de color es RGB, también se tiene que normalizar los datos, pero en este caso, el intervalo es $[-1,1]$. [10]

Esta red neuronal ya estaba entrenada, por tanto, no se tuvo que hacer ningún preprocesado en Python; pero al implementar la red en Android Studio, al igual que en la red VGG16, se debía realizar el preprocesamiento a mano.

Donde la formula para cambiar este valor vendría dada por:

$$ColorPreprocesado = (ColorSinPreprocesar - 0)/255,0 * 2 - 1$$

- Donde 0 representa el valor mínimo que puede tomar el color en concreto.
- Donde 255 representa el valor máximo que puede tomar el color en concreto.
- Y donde $*2 - 1$ es la operación para normalizar el valor en formato $[-1, 1]$

3.3. Formato de la red neuronal

El formato de las redes neuronales convolucionales suele venir dado en formato `<.h5>` o `keras`, para facilitar la implementación en Android Studio, se ha tenido que transformar este archivo en formato TensorFlow Lite, el cual requiere menos recursos, permitiendo su integración en dispositivos móviles.

Para realizar esta conversión, se ha utilizado un fichero Python, el cual se ha buscado en la documentación de TensorFlow Lite, y posteriormente, guardar el fichero. [9]

Técnicas y herramientas

4.1. Metodologías

SCRUM

SCRUM es una metodología ágil basada en una estrategia continua e incremental, cuyo objetivo es proporcionar un producto funcional al final de cada periodo de trabajo planeado (*sprint*), haciendo reuniones diarias y antes y después de cada *sprint* otras reuniones donde se explican los problemas que se han tenido y como se va a planear el siguiente.

El impedimento más notorio de esta metodología, es que hay un equipo de personas entre las que se encuentra el *product owner*, el *SCRUM master* y el equipo de desarrollo. Además se recomienda que el equipo este formado por 3 a un máximo de 9 personas para un buen desarrollo, por tanto, en este trabajo ha sido complicado ir haciendo todas las acciones que se piden en la metodología SCRUM.

GitFlow

GitFlow es un flujo de trabajo, en el cual se ramifica el proyecto en *branches* donde cada una, contiene una parte del proyecto; de esta forma, se puede dejar una parte funcional sin modificar que sería la rama principal, y otras ramas, donde se van realizando los cambios, cuando en estas ramas, se termina la tarea que se esta realizando, haciendo un producto funcional, se realiza una operación de *pull request* para combinar ambas ramas. Esta operación es aceptada o denegada por el equipo de desarrollo que no haya participado en la realización de esta rama.

En mi proyecto, se han creado tres *branches*, que son:

- *main*: es la rama principal, donde se alberga la versión estable del proyecto.
- *boceto*: es una rama secundaria, donde se realizan los cambios que se están realizando en la aplicación móvil.
- *latex*: es la rama secundaria donde se realizan los cambios que se están realizando en el documento LaTeX.

Método del pato de goma

El método del pato de goma, o *rubber duck debugging*, es un método informal para la revisión de código.

Este método es utilizado por muchos programadores y surgió porque normalmente los programadores han tenido experiencias donde han tenido que explicar el problema a otra persona que no entiende sobre programación, y mientras se esta explicando el código, encontrar posibles soluciones.

Por tanto, este método, consiste en vez de explicar el código a otra persona, en explicárselo a un pato de goma.

4.2. Herramientas

Repositorio

Entre las opciones para realizar el repositorio, se pensó en GitLab y en GitHub, tomando esta última por conocimiento de uso de esta.

GitHub es una plataforma web de alojamiento de repositorios que utiliza Git como sistema de control de versiones.

Por tanto, como GitHub utiliza Git de forma nativa, no fue necesario la elección de un sistema de control de versiones porque ya estaba incluido.

Como para el repositorio se necesitan archivos de mas de 100 MB, se ha usado la extensión Git LFS, estos ficheros son los las redes neuronales, tanto formato keras, como formato TensorFlow Lite.

Gestión del proyecto

Entre las opciones para gestionar el proyecto, se pensó en poder realizarlo con GitHub Projects, con Jira, con ZenHub y Trello; tomando la opción de ZenHub, por ya estar familiarizado con la herramienta.

ZenHub es una herramienta de gestión de proyectos que se integra con GitHub. Proporciona una tabla Kanban, donde se ponen las actividades a realizar durante el *sprint*. Permite poner una prioridad a las tareas, siguiendo una estimación de poker, y además, ofrece la posibilidad de ver gráficos *burndown* donde se representa las actividades que faltan por hacer en el *sprint*, también ofrece otros gráficos como *cumulative flow* y *velocity tracking*.

Guía de diseño

Para la realización de la aplicación, es necesario una guía de diseño que facilite al usuario la interacción con la aplicación. Por ello, se utilizó la última guía ofrecida por Google, llamada Material 3.

Material 3 es la última versión del sistema de diseño del *open source* de Google. En el documento, viene información de recomendación de componentes ante el mismo problema.

Herramienta de iconos

Para la adición de iconos en la aplicación, es necesario que las imágenes y/o iconos sean *open source* para ello, se estuvieron mirando páginas que ofrecían estos iconos, entre las páginas, al final se seleccionaron pixabay para la obtención de imágenes, puesto que ofrece licencia de uso para proyectos comerciales y no comerciales.

Por otro lado, para los iconos se utilizó Google Fonts y la galería que ofrece Android por defecto, se utilizó Google Fonts, porque en la guía de desarrollo se recomendaba y a su vez, es de código abierto y gratuito.

Para el logotipo de la aplicación, se ha utilizado DALL-E, una inteligencia artificial que convierte texto a imágenes, tras varios intentos, se logró conseguir un logotipo bastante bueno, y con ello, se hicieron unos retoques con el uso del programa de edición de imágenes GIMP, para la eliminación de ruido y para proporcionarle una gamma de colores; como resultado se obtuvo el icono actual.

Entorno de desarrollo integrado (IDE)

Para el desarrollo de la aplicación móvil, se pensó en Eclipse, Android Studio y Unity.

Al final, se descartó Eclipse por no ofrecer una experiencia de desarrollo específica para aplicaciones Android. Y también se descartó Unity, aunque

Unity si que esta especializado en aplicaciones móviles, por otro lado, se especializa en el desarrollo de videojuegos y en aplicaciones con realidad virtual. Por lo tanto, se selecciono Android Studio, es el IDE oficial de Android y está desarrollado por Google, basado en IntelliJ IDEA. Ofrece un emulador donde se compila la aplicación y poder comprobar el correcto funcionamiento de esta. Además, para la compilación hace uso de Gradle, separando el código de la aplicación, de la compilación.

El lenguaje utilizado ha sido Java, aunque Android Studio ofrece la opción de Kotlin, con Java no hacia falta aprender un nuevo lenguaje, puesto que se ha cursado durante la carrera.

LaTeX

Para la realización del documento, se pensó utilizar MiKTeX, TeX Live u Overleaf.

Al final, se escogió Overleaf, puesto que las otras 2 herramientas eran locales, y Overleaf ofrece acceso desde la nube, además tiene integración con proyectos GitHub, facilitando la exportación al repositorio.

Comunicación

Para la comunicación con los tutores, se uso tanto email, como tutorías presenciales, de esta forma, las cuestiones y los avances realizados se hacen por email, y en caso de mostrar el funcionamiento de la aplicación o alguna duda más importante, se realiza presencialmente para un mejor entendimiento.

4.3. Patrones de diseño

Modelo-Vista-Presentador

Es una derivación del patrón modelo-vista-controlador, la diferencia es que se cambia el controlador por un presentador, el cual sirve como intermediario entre el modelo y la interfaz.

- El modelo, define los datos que se utilizan en la interfaz.
- El presentador, funciona como intermediario, recuperando los datos del modelo, y cambiando la vista de la interfaz.

- La vista, es la interfaz de usuario.

Puesto que Android Studio, ya proporciona separaciones para hacer uso de un modelo-vista-presentador, su implementación ha sido sencilla, donde las *activities* son las diferentes interfaces, los archivos java del directorio `com.example.retinopatia` son los presentadores de las vistas, y en el directorio `DataBase` se encuentra el modelo de datos.

Aspectos relevantes del desarrollo del proyecto

En este apartado se recogen los aspectos más importantes. Explicando las decisiones tomadas del proyecto, y las consecuencias que suponen, comentando los errores y como se solucionaron.

5.1. Inicio

Una vez se me explicó la idea que se buscaba con este proyecto, me gustó la idea de poder hacer una aplicación que pudiera servir al sistema sanitario público.

En este inicio, empecé a imaginarme como podría ser la aplicación haciendo bocetos mentales de sus distintas pantallas, como era muy efímero, se hizo una búsqueda de aplicaciones similares, para comprobar alguna interfaz a añadir, donde se encontró la aplicación RetinCam. Con una idea más formada, se hicieron los bocetos iniciales de la aplicación.

5.2. Metodologías

Como ya se ha comentado anteriormente, se decidió utilizar una metodología SCRUM, no siguiéndose al completo, puesto que los equipos de desarrollo en esta metodología están compuesto de 3 a 9 personas, a su vez, hay reuniones que no se han podido realizar, entre otras cosas. Pero, con esta metodología se ha buscado que el proyecto tuviese una metodología ágil.

Un fallo cometido con respecto esta metodología ha sido que los martes cambiaba de sprint a las 8 de la mañana, y la revisión para finalizar el sprint, se realizaba los martes por la mañana, haciendo que algunas veces las tareas cambiasen de sprint cuando no debían, y por tanto, la reunión para comenzar el sprint se realizaba con el sprint comenzado.

Las características ágiles de este proyecto son:

- Los sprints planeados han tenido una duración de 2 semanas, entregando un incremento al final de cada uno.
- Se han realizado reuniones, tanto para finalizar el sprint, como para el inicio del siguiente, teniendo en consideración el fallo comentado anteriormente.
- Las tareas que se planeaban para un sprint, se estimaban y priorizaban en un tablero canvas, tanto físicamente como con la herramienta ZenHub. Aunque con el paso de los sprints, se dejó de hacer físicamente.
- Para comprobar el progreso del proyecto, se ha utilizado los gráficos burndown, que aunque los ofrece ZenHub, se han realizado a mano, por el fallo comentado.

Al principio del proyecto se planeaba utilizar otras metodologías como *Test-Driven-Development* o como *Data-driven testing*, junto con pruebas automáticas, vistas durante el año académico en la asignatura Validación y pruebas, para comprobar como las implementaciones que se iban realizando en el proyecto no se veían afectadas entre ellas y que se implementaban correctamente. Pero al implementar las distintas interfaces, se decidió hacer pruebas manuales, las cuales proporcionan una mayor comprensión de la interacción que tiene el usuario con la aplicación, buscando siempre que el usuario entienda el sistema.

Trabajos relacionados

6.1. Proyectos

Ret-iN CaM

Ret-iN CaM es la aplicación que se ha tomado de referencia para hacer el proyecto. Es una aplicación iOS que posteriormente sacó una versión para Android. Es una aplicación que permite realizar imágenes y vídeos de la retina con una gran resolución, proporcionando los informes del paciente, los cuales se pueden exportar para ser compartidos con otros especialistas. A su vez, tiene una interfaz simple e intuitiva, lo que facilita a los usuarios interactuar fácilmente con ella. Principal motivo por el que se ha escogido esta aplicación.

D-EYE 2.0

Es un proyecto que permite la toma de imágenes y vídeos de alta calidad; permite a los médicos ver el nervio óptico sin necesidad de dilatar las pupilas; permite a los médicos ver si el paciente tiene trastornos neurológicos relacionados con el ojo.

6.2. Comparativa del proyecto

Características	RetinAI	Ret-iN CaM	D-EYE 2.0
Aplicación Android	X	X	
Aplicación iOS		X	X
Creación de usuarios			X
Cambio de modo oscuro y claro	X		
Permite iniciar sesión como invitado	X		
Permite guardar la sesión		X	X
Permite elegir el paciente	X	X	X
Ver historial del paciente	X	X	X
Crear nuevos informes	X	X	X
Permite diferenciar entre ojos	X	X	X
Permite hacer imágenes	X	X	X
Permite hacer vídeos		X	X
Escoger imágenes desde la galería	X		
Red neuronal para los resultados	X		
Versión gratuita	X	X	X

Tabla 6.1: Comparativa de las características de los proyectos.

De esta forma, se puede ver las ventajas que ofrece el proyecto.

- Actualmente, hay más móviles con sistema operativo Android que con iOS, por tanto, se ha realizado la aplicación en un sistema Android por este motivo.
- No se permite la creación de usuarios, puesto que como la aplicación esta destinada a médicos de la sanidad publica, la entidad encargada les proporcionará las cuentas para la aplicación.
- La aplicación tiene la opción de cambiar entre modo oscuro y modo claro, de esta forma, permite al usuario adaptarla a su preferencia.
- Al iniciar sesión como usuario, los médicos podrán tener un diagnostico rápido de un paciente, sin que se almacene el informe en la base de datos.
- A la hora de seleccionar pacientes, se ha considerado la protección de datos de los pacientes y para que el médico seleccione a uno, tendrá que poner el DNI.

- Como es posible que se analice una foto tomada desde otro dispositivo. Se ha considerado esta idea mostrando en el explorador de archivos las imágenes.
- Ofrece una red neuronal ya entrenada, la cual determina el grado de retinopatía diabética que tiene el paciente. Característica en la que se basa la aplicación.

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

Bibliografía

- [1] Organización Mundial de la Salud. Ceguera y discapacidad visual, 2022. URL <https://www.who.int/es/news-room/fact-sheets/detail/blindness-and-visual-impairment>. [Online; Accedido 10-febrero-2023].
- [2] Organización Mundial de la Salud. Diabetes, 2022. URL <https://www.who.int/es/news-room/fact-sheets/detail/diabetes>. [Online; Accedido 18-febrero-2023].
- [3] Google. Convolutional layer, 2023. URL https://developers.google.com/machine-learning/glossary?hl=es-419#convolutional_layer. [Online; Accedido 24-mayo-2023].
- [4] Google. Convolutional operation, 2023. URL https://developers.google.com/machine-learning/glossary?hl=es-419#convolutional_operation. [Online; Accedido 24-mayo-2023].
- [5] Google. Convolutional neural network, 2023. URL <https://developers.google.com/machine-learning/glossary?hl=es-419#convolutional-neural-network>. [Online; Accedido 24-mayo-2023].
- [6] Google. fully connected layer, 2023. URL https://developers.google.com/machine-learning/glossary?hl=es-419#fully_connected_layer. [Online; Accedido 24-mayo-2023].
- [7] Google. Pooling, 2023. URL <https://developers.google.com/machine-learning/glossary?hl=es-419#pooling>. [Online; Accedido 24-mayo-2023].

- [8] Cecelia Koetting OD, FAAO. The four stages of diabetic retinopathy. *Modern Optometry*, 2019.
- [9] TensorFlow. Tfliteconverter tensorflow lite api, 2022. URL https://www.tensorflow.org/lite/api_docs/python/tf/lite/TFLiteConverter. [Online; Accedido 27-mayo-2023].
- [10] TensorFlow. Resnet50v2, 2023. URL https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet_v2/ResNet50V2. [Online; Accedido 27-mayo-2023].
- [11] TensorFlow. Vgg16, 2023. URL https://www.tensorflow.org/api_docs/python/tf/keras/applications/vgg16/VGG16. [Online; Accedido 27-mayo-2023].
- [12] David Turbert. La enfermedad ocular diabética, 2022. URL <https://www.aao.org/salud-ocular/enfermedades/la-enfermedad-ocular-diabetica>. [Online; Accedido 20-mayo-2023].