EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCIÓN I.- TESIS

TRABAJO PROFESIONAL

"Análisis de características para la detección temprana de DMAE variante seca a partir del análisis de imágenes del fondo del ojo utilizando técnicas de Visión Artificial y Aprendizaje Profundo"

QUE PARA OBTENER EL GRADO DE:
MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA:
I.S.C. Augusto Javier Reyes Delgado

DIRECTOR DE TESIS:
Dr. José Luis Sánchez Cervantes

CODIRECTOR DE TESIS:
MIDS. Jorge Ernesto González Díaz

ORIZABA, VERACRUZ, MÉXICO SEPTIEMBRE 2024

Orizaba, Veracruz, 26/septiembre/2024
Dependencia: **División de Estudios de Posgrado e Investigación**
Asunto: **Autorización de Impresión**
OPCION: I

C. AUGUSTO JAVIER REYES DELGADO
CANDIDATO A GRADO DE MAESTRO EN:
SISTEMAS COMPUTACIONALES
P R E S E N T E.-

De acuerdo con el Reglamento de Titulación vigente de los Centros de Enseñanza Técnica Superior, dependiente de la Dirección General de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

" Análisis de características para la detección temprana de DMAE variante seca a partir del análisis de imágenes del fondo del ojo utilizando técnicas de Visión Artificial y Aprendizaje Profundo."

comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
CIENCIA – TÉCNICA - CULTURA®

DRA. OFELIA LANDETA ESCAMILLA
JEFA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

EDUCACIÓN SECRETARÍA DE EDUCACIÓN PÚBLICA | TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLOGICO DE ORIZABA
OG-13-F06
DIV. DE EST. DE POSGRADO E INVEST.

Orizaba, Veracruz, 19/septiembre/2024
Asunto: **Revisión de trabajo escrito**

C. OFELIA LANDETA ESCAMILLA
JEFA DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN
P R E S E N T E.-

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C.

AUGUSTO JAVIER REYES DELGADO

La cual lleva el título de:

Análisis de características para la detección temprana de DMAE variante seca a partir del análisis de imágenes del fondo del ojo utilizando técnicas de Visión Artificial y Aprendizaje Profundo.

Y concluyen que se acepta.

ATENTAMENTE
Excelencia en Educación Tecnológica®
CIENCIA – TÉCNICA - CULTURA®

PRESIDENTE:	DR. JOSÉ LUIS SÁNCHEZ CERVANTES	FIRMA
SECRETARIO:	DR. GINER ALOR HERNÁNDEZ	FIRMA
VOCAL:	DRA. LISBETH RODRÍGUEZ MAZAHUA	FIRMA
VOCAL SUP.:	M.C. JORGE ERNESTO GONZÁLEZ DÍAZ	FIRMA

TA-09-18

CARTA DE ORIGINALIDAD

En la ciudad de Orizaba, Veracruz, el día 26 de septiembre del año 2024, el (la) que suscribe **Augusto Javier Reyes Delgado**, alumno (a) del programa de **Maestría en Sistemas Computacionales** con número de control **M17010207**, manifiesta que es autor(a) del trabajo de tesis titulado **"Análisis de características para la detección temprana de DMAE variante seca a partir del análisis de imágenes del fondo del ojo utilizando técnicas de Visión Artificial y Aprendizaje Profundo"** y declaro que el trabajo es original ya que sus contenidos son producto de mi directa contribución intelectual. Todos los datos y las referencia a materiales ya publicados están debidamente identificados con su respectivo crédito e incluidos en las notas bibliográficas y en las citas que se destacan como tal y, en los casos que así lo requieran, cuento con las debidas autorizaciones de quieres poseen los derechos patrimoniales. Por lo tanto, me hago responsable de cualquier litigio o reclamación relacionada con derechos de propiedad intelectual, exonerando de toda responsabilidad al Tecnológico Nacional de México / Instituto Tecnológico de Orizaba.

Augusto Javier Reyes Delgado
Nombre y Firma

CARTA DE CESIÓN DE DERECHOS

En la ciudad de Orizaba, Veracruz, el día 26 del mes de septiembre del año 2024, el (la) que suscribe **Augusto Javier Reyes Delgado**, alumno (a) del programa de **Maestría en Sistemas Computacionales** con número de control **M17010207**, manifiesta que es autor(a) del trabajo de tesis bajo la dirección de José Luis Sánchez Cervantes y cede los derechos del trabajo de tesis titulado **"Análisis de características para la detección temprana de DMAE variante seca a partir del análisis de imágenes del fondo del ojo utilizando técnicas de Visión Artificial y Aprendizaje Profundo"** al **Tecnológico Nacional de México / Instituto Tecnológico de Orizaba** para su difusión y divulgación, con fines académicos y de investigación.

Queda estrictamente prohibido reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del **Tecnológico Nacional de México / Instituto Tecnológico de Orizaba**. Este puede obtenerse escribiendo a la siguiente dirección: msc@orizaba.tecnm.mx. Si el permiso se otorga, cualquier usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

Augusto Javier Reyes Delgado

Nombre y Firma

Agradecimientos

Al Tecnológico Nacional de México – Campus Orizaba por permitirme cursar el programa de posgrado de Maestría en Sistemas Computacionales.

Así como al Consejo Nacional de Humanidades, Ciencias y Tecnologías por el financiamiento otorgado, con el cual pude realizar satisfactoriamente mis estudios de postgrado.

Agradezco a mi madre, la maestra Inés Reyes Delgado, por siempre apoyarme y escucharme, por siempre tolerarme y soportarme, por siempre amarme y perdonarme, por jamás perder la fe y confianza en mí en momentos en los que cualquier otra persona en su lugar lo hubiera hecho. A la autora de mi vida, no me alcanzan las palabras para dar las gracias. A mi padre, el señor Arturo López Soriano, por siempre acompañarnos y cuidarnos. Y a Karamelo, por ser el más real de los compañeros, sin trampas ni cartones, solo gato naranja, goloso y juguetón.

A mis directores de tesis, el Dr. José Luis Sánchez Cervantes y el maestro Jorge Ernesto González Díaz, cuya guía y dirección proporcionaron las herramientas necesarias para la correcta conclusión del presente proyecto de tesis. Agradezco también su mentoría y confianza, la cual me ayudó inmensamente en mi formación como ingeniero e investigador.

A mis hermanos de la cofradía conocida como *Orden Oculta de Sistemas*, por su apoyo en los momentos ásperos de la vida, y por su compañía en los momentos placidos de la misma.

Finalmente, agradezco al Instituto de Oftalmología Fundación Conde de Valenciana, cuya colaboración contribuyó invaluablemente, no solo en la conclusión del proyecto de tesis, sino a la calidad de los resultados obtenidos en el mismo.

Tabla de contenido

Índice de tablas

Índice de figuras

Resumen

La degeneración macular asociada con la edad (DMAE) es una condición médica en la cual, una estructura en el fondo central de la retina, conocida como mácula lútea (encargada de la visión central fina), ve degradadas sus células por diversos factores, tales como la obesidad, la hipertensión, el tabaquismo y por supuesto, la edad. Esta condición médica, es una de las tres principales causas de ceguera entre adultos mayores alrededor del mundo, según la Organización Mundial de la Salud (OMS) y se tornará un mal cada vez más común a medida que la población mundial tiende a envejecer.

En su manifestación más común, conocida como manifestación seca (el 80% de los casos) la degeneración macular es tratable, pero también es gradual e irreversible, por lo que es indispensable la detección temprana de la enfermedad, sin embargo, esta no presenta síntomas hasta etapas intermedias.

De lo anterior, surge la idea del presente proyecto, el cual propone un módulo de software Web para el análisis de características para la detección temprana de la degeneración macular asociada con la edad. Para ello el módulo usará técnicas de visión artificial y aprendizaje profundo para el análisis de imágenes del fondo del ojo, buscando detectar y clasificar la enfermedad en sus diferentes estadios.

El aporte de este proyecto yace en su utilidad para los especialistas de la salud en el diagnóstico de la degeneración macular asociada con la edad en sus diferentes etapas, siendo la detección temprana esencial para evitar la pérdida total de visión en aquellos individuos que padezcan de dicha enfermedad y una condición incapacitante que deteriore su calidad de vida y la de sus familiares.

Introducción

Cataratas, glaucoma, errores de refracción y retinopatía diabética son varias de las causas que lista la OMS hasta octubre del 2022 como principales causas de la ceguera y discapacidad visual, pero entre ellas hay una que se volverá más y más relevante con el tiempo para la población en general, la degeneración macular asociada con la edad. En los últimos años, la población del planeta ha ido aumentando su esperanza de vida, permitiendo que cada vez más personas lleguen a edades en las que presentan enfermedades típicas de la vejez, además del hecho de que la tasa de natalidad ha ido disminuyendo progresivamente, provocando que la proporción de ancianos sea cada vez mayor en los países. De ello que cada vez, una mayor proporción de personas padecerán de degeneración macular asociada con la edad o DMAE.

La DMAE, es una de las tres principales causas de ceguera alrededor del mundo, afectado principalmente a países desarrollados, porque que está fuertemente asociada a condiciones como la obesidad, diabetes e hipertensión (además del tabaquismo), las cuales son muy comunes en México.

La DMAE presenta dos manifestaciones, la húmeda y la seca. Para el alcance de este proyecto, se plantea abordar la manifestación seca, la cual es gradual, progresiva y causa daño permanente, es decir, es irreversible.

De las investigaciones realizadas para la elaboración de este anteproyecto, se ha encontrado que el uso de visión artificial y aprendizaje profundo en la detección de la degeneración macular, es un enfoque altamente efectivo, permitiendo detectar las características correspondientes a la enfermedad.

Ahora, un nuevo enfoque en la visión artificial es el uso de los Transformers (un tipo de algoritmo de inteligencia artificial) para la detección de objetos en imágenes. Con su aparición, se superó en eficacia a las que hasta hace pocos años eran el estado del arte en el análisis y clasificación de imágenes, las redes neuronales convolucionales.

El enfoque de este proyecto es el uso Trasformadores de visión para la clasificación de imágenes de fondo de ojo y encontrar en ellas las características para determinar

si existe o no la condición de DMAE, y en el caso de que exista, el nivel de afectación presente, pues en sus etapas tempranas, la enfermedad es tratable.

El presente trabajo se divide en tres capítulos: El primero aborda el marco teórico, planteamiento del problema, objetivos generales, objetivos específicos y la justificación; el segundo capítulo presenta el estado de la práctica y el análisis comparativo de los artículos estudiados; en el capítulo tres se describe la propuesta de solución, la descripción de la solución, el análisis de las tecnologías, el cronograma de actividades y los respectivos entregables. Por último, se presentan las conclusiones y referencias del trabajo.

Capítulo 1. Antecedentes

En este capítulo se presentan diversos conceptos que resultan relevantes para este proyecto. También se describe la problemática a resolver, el objetivo general, los objetivos específicos y la justificación del presente trabajo.

1.1 Marco Teórico

A continuación. Se describen los conceptos relacionados con el tema del proyecto.

1.1.1 Inteligencia Artificial

La inteligencia artificial se define como cualquier proceso computacional diseñado para emular la inteligencia humana, es decir, replicar comportamientos orientados a la resolución de problema. Para esto la inteligencia artificial hace uso de algoritmos, memorización y aprendizaje. Y el aprendizaje es la clave para diferenciar un algoritmo simple de la inteligencia. Un algoritmo, tal como una máquina de estados, siempre sigue una serie de pasos preestablecidos para alcanzar su propósito. Por otro lado, la inteligencia artificial aprende de experiencias previas y utiliza ese conocimiento adquirido para resolver problemas o ejecutar tareas asignadas [1], [2]. Estas tareas son muy variadas, tales como la visión artificial y clasificación de imágenes, aunque su aplicación también se expande a: 1.- Asistentes virtuales 2.- Su uso en GPS 3.- Procesos selectivos 4.- Plataformas de distribución de contenido digital y 5.- Chatbots [3]. Estos últimos han experimentado un gran auge recientemente debido a su capacidad para mantener conversaciones coherentes con los usuarios, gracias a su mencionada habilidad de aprendizaje. El aprendizaje en inteligencia artificial se divide en dos grandes campos que se detallan a continuación:

1.1.1.1 Aprendizaje Automático

Es una rama de la inteligencia artificial, la cual se centra en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender y mejorar automáticamente a partir de la experiencia en lugar de ser programadas explícitamente para la realización de una tarea específica. Para ello, las máquinas utilizan datos la identificación patrones, aprender de ellos y tomar decisiones con mínima intervención humana. A continuación, se presenta una breve descripción de los tipos de aprendizaje automático [1]:

Aprendizaje supervisado: Es aquel en el que los datos de entrenamiento son previamente organizados o clasificados por humanos, además de requerir de la retroalimentación de estos.

Aprendizaje no supervisado: En este, los datos no son previamente organizados ni clasificados por humanos para indicar como realizar la categorización de la información. Sino que el algoritmo tiene que encontrar la manera de clasificar por su cuenta.

Aprendizaje por refuerzo: Este tipo de aprendizaje contempla algoritmos que se ocupan de determinar qué acciones elegirá un agente de software en un entorno determinado con la finalidad de recibir “refuerzos positivos” cada vez que acierta, es decir, los refuerzos son acumulables con base en los aciertos para la identificación, clasificación, descubrimiento y organización de información según sea el caso.

1.1.1.2 Aprendizaje Profundo

Son un tipo de algoritmos que simulan el comportamiento del cerebro humano, esto mediante el uso de redes neuronales artificiales, las cuales aprenden de los datos con que son entrenadas. Los términos “red neuronal” y “algoritmo de aprendizaje profundo” se consideran sinónimos en el contexto en el que se utilizan para referirse a una arquitectura específica de redes neuronales profundas.

Su arquitectura es la siguiente, se cuenta con tres capas principales: 1.- La capa de entrada, que es la que recibe los datos. 2.- Las capas ocultas, en donde se realizan operaciones sobre los datos que son proporcionados por la capa de entrada, para identificar la relación entre los datos y realizar la clasificación. 3.- La capa de salida, que es la que entrega los datos de salida, es decir, la clasificación. El segundo tipo de capas son las que distinguen el aprendizaje profundo de las redes neuronales tradicionales, una red tradicional permite tener una o dos capas ocultas, mientras que una red de aprendizaje profundo implica la posibilidad de incluir centenas de capas que se recorren por épocas que a su vez se integran por lotes que requieren iteraciones para realizar las clasificaciones necesarias.

Este tipo de algoritmo es muy eficiente, pero requiere de una gran cantidad de recursos, tanto computacionales como de datos. Su uso es muy amplio, se usan en el reconocimiento de voz, procesamiento de lenguaje natural, asistencia al conductor y visión artificial, por mencionar algunos ejemplos [1], [4].

1.1.2 Redes Neuronales Convolucionales

Una red neuronal convolucional (*convolutional neural network*, CNN por sus siglas en inglés) un tipo de red neuronal artificial especializada en la identificación de imágenes. Se les conoce como redes neuronales porque su funcionamiento se basa fuertemente en los procesos de los sistemas nerviosos biológicos, como, por ejemplo, el cerebro. Incluso los componentes (nodos) interconectados de las redes neuronales artificiales toman su nombre de las células que componen el sistema nervioso. Y al igual que las células homólogas, el trabajo colectivo de los nodos de una red neuronal les permite desarrollar un aprendizaje.

Una red neuronal artificial tradicional tiene la siguiente estructura básica: 1.-capa de entrada, 2.-capa oculta y 3.-capa de salida. Siendo la capa de entrada aquella que recibe las características (los datos) con los que se hace la clasificación, las capas ocultas las que hacen la ponderación de la clasificación, y la capa de salida la que entrega la predicción. Es decir, para realizar una predicción, una RNA tradicional recibe una serie de características y de ellas devuelve su predicción, sin embargo,

cuando lo que se le ingresa es una imagen, lo que la computadora interpreta es una serie de píxeles. Un píxel por sí mismo sirve de mucho como característica, pues es la relación del píxel con sus píxeles aledaños lo que da sentido a una imagen. De aquí surge lo que hace diferentes a las redes neuronales convolucionales. Esta agrega dos capas al modelo tradicional de red neuronal, después de la capa de entrada, la capa convolucional y la capa de agrupación.

Capa convolucional: Esta utiliza filtros convolucionales para extraer características importantes de los datos de entrada, como una matriz de pixeles (una imagen). La operación de convolución implica deslizar un pequeño filtro sobre la entrada y calcular el producto punto entre los valores del filtro y los valores correspondientes de la región de entrada cubierta por el filtro. El proceso se repite en toda la entrada, generando un mapa de características que resalta las características relevantes, lo cual permite aprender patrones como bordes, texturas y formas en diferentes niveles de abstracción a medida que se apilan capas convolucionales en una red neuronal.

Capa de agrupación: Esta tiene como función reducir la dimensionalidad de los mapas de características obtenidos después de las capas convolucionales, lo que ayuda a combatir el sobreajuste y a mejorar la eficiencia y el rendimiento del modelo. Su operación implica dividir la entrada en regiones solapadas o no solapadas y luego aplicar una función de resumen, como el máximo o el promedio, a cada región para obtener un único valor representativo. De esta manera, la información relevante se conserva mientras que se reduce el tamaño de la representación.

Posteriormente siguen el resto de capas de una red tradicional, las capas ocultas y de salida, que en este caso son referidas como la capa completamente conectada.

1.1.3 Vision Transformer

Las arquitecturas basadas en Vision Transformer son el actual “Estado del arte” en el campo de la visión artificial, superando en algunos escenarios a las redes neuronales convolucionales. Vision Transformer (ViT) fue un cambio de paradigma

al aplicar la dinámica de los Transformers, de reconocimiento de lenguaje natural a reconocimiento de imágenes. Con la finalidad de tener una comprensión más amplia de los Transformadores y los Transformadores de Visión los siguientes apartados presentan una explicación detallada de ambos conceptos.

1.1.3.1 ¿Qué es un Transformer?

Un Transformer es un modelo de red neuronal artificial muy utilizado en el procesamiento de lenguaje natural. Se basa en los principios de atención y autoatención. Su objetivo es el procesamiento de texto en forma de una secuencia de tokens de manera efectiva y eficiente, esto mediante la captura de las relaciones entre los distintos tokens. De las relaciones entre los distintos tokens, se obtiene un contexto, por ejemplo, en el lenguaje natural, una oración se compone por una serie de palabras, sin embargo, las palabras por sí solas no dan el significado completo de lo que se quiere comunicar, es el orden en que se presentan las palabras y la relación entre ellas lo que transmite un verdadero significado. Como ejemplo la siguiente oración:

“El gato ronroneó con fuerza, su amo sintió en su pecho su potencia”.

Para un humano es claro que la “potencia” se refiere a la fuerza con la que el gato ronroneaba, pero para una máquina esto es más difícil de entender a que pertenece esa “potencia”, si es que la máquina interpreta palabra por palabra.
Entonces la clave de la arquitectura Transformer está en el mecanismo de autoatención. La cual permite que la red determine qué elementos de entrada son relevantes en relación con otros tokens en particular y entonces colocarlos en consecuencia. Esto permite a la red “prestar atención”.

La arquitectura de los Transformer consiste en dos módulos principales, el codificador y el decodificador, los cuales se estructuran en forma de pilas. En el codificador se encuentra la capa de autoatención y alimentación directa. Y el decodificador que tiene la capa de alimentación directa, la capa de autoatención y

el decodificador de atención. En esta última capa los resultados de la autoatención se procesan para generar una salida final [5], [6].

1.1.3.2 ¿Qué es Vision Transformer?

Una vez se realizó una breve introducción a lo que es un Transformer, es más viable comprender el principio del funcionamiento detrás de un Vision Transformer (ViT). El ViT aplica el mismo principio que utilizan los Transformer en el lenguaje natural, es decir, es capaz de conservar un contexto de aquello que analiza para tener una mejor comprensión al respecto. Sin embargo, los Transformers aplican este principio al lenguaje, donde naturalmente se trata de secuencias de caracteres o vectores de caracteres para analizar. En contraste, en el caso de una imagen, lo que se maneja es un mapa de píxeles cuyas relaciones no se limitan a secuencias de izquierda a derecha o viceversa, sino que cada píxel interactúa con todos los píxeles que lo rodean.

En este escenario, es donde ViT pone en funcionamiento su arquitectura. En términos simples, ViT toma una imagen, y la divide en tokens, los ordena en secuencia y los coloca dentro de su clasificador. Como se muestra en la figura 1.3.
Más detalladamente, la arquitectura de ViT se compone de dos módulos principales, el módulo de extracción de características y el módulo de clasificación.
En el módulo de extracción de características, ViT toma como entrada una imagen, la cual divide en una serie de parches de tamaño fijo para realizar la extracción de características, se aplana en vectores y se tratan como tokens separados a los que se les añade información de posición y clase.
Una vez generada la secuencia de tokens, esta se alimenta a través de una serie de capas de atención Transformer. Estas capas se componen de dos subcapas principales, la capa de atención multi cabezal y la capa completamente conectada. En la capa de atención multi cabezal, los tokens se agrupan en múltiples conjuntos y se calcula una atención entre ellos. En la capa completamente conectada, se utiliza una red neuronal tradicional para procesar los resultados de la capa de atención multi cabezal.

El segundo módulo de la arquitectura se encarga de la clasificación, esto a través de la información de características generada en el paso anterior. Esto mediante un perceptrón multicapa, una red neuronal tradicional [7].

Vision Transformer por sí mismo muestra muy buenos resultados en el campo de visión artificial, y de esta arquitectura derivan otras nuevas y más completas, que se describen en el capítulo 3.

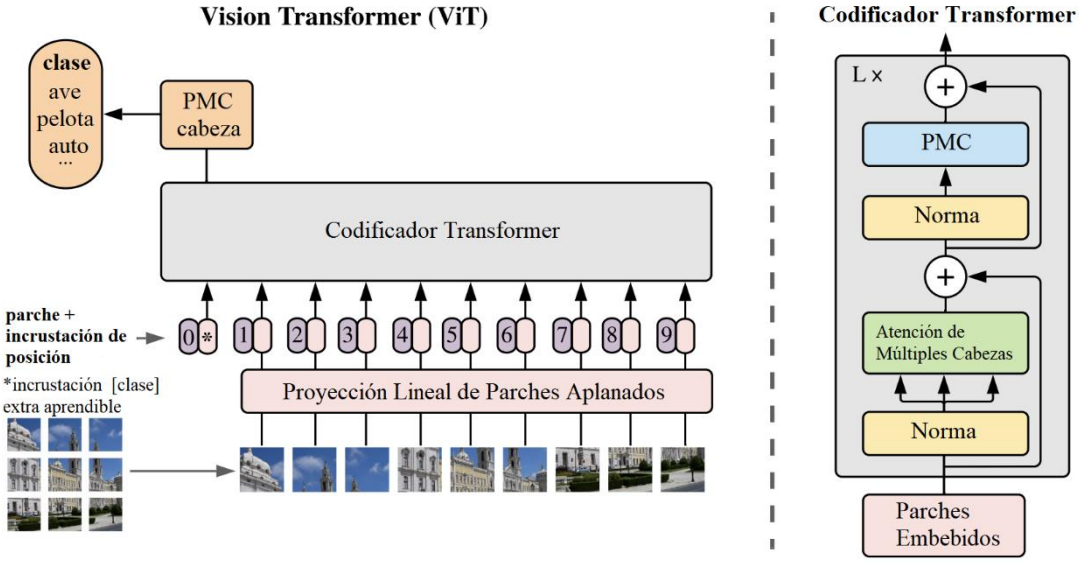


Figura 1.1 *Arquitectura de ViT.* Imagen accedida en[1]

1.1.4 Degeneración Macular Asociada con la Edad

La degeneración macular asociada con la edad (*Age-Related Macular Degeneration*, AMD por sus siglas en inglés) o DMAE es un padecimiento crónico-degenerativo que afecta la mácula de individuos de avanzada edad (a saber, individuos de más de 50 años), causado por diversas anomalías en el fotorreceptor, la membrana de Bruch, el epitelio pigmentario de la retina, la aparición de drusas, atrofia geográfica y/o el desarrollo de neovascularización, produciendo en los que la padecen un deterioro en la agudeza de la visión central, es decir, no es factible apreciar los detalles finos ni cerca ni lejos en el centro focal de la visión, aunque la

[1] Dosovitskiy, A. (2021). An image is worth 16x16 words: transformers for image recognition at scale. ICLR.

visión periférica se mantiene normal [8], [9]. Esta enfermedad causa una pérdida de visión severa y es una de las principales causas de ceguera en países desarrollados [10], siendo según la OMS una de las tres primeras causas de ceguera alrededor del mundo, presentándose de manera moderada o severa en alrededor de 10.4 millones de personas, aunque su incidencia en sus diferentes fases se estima en 196 millones de personas [11]. Este padecimiento se asocia a diferentes hábitos, tales como el tabaquismo y el consumo de alcohol y a enfermedades metabólicas como la obesidad, diabetes e hipertensión [9], además, de aparecer por predisposición genética [10]. Existen dos manifestaciones clínicas de esta enfermedad, las cuales se clasifican como DMAE seca y DMAE húmeda, cuyas diferencias se explican a continuación:

1.1.4.1 DMAE: manifestación seca

También llamada como no neovascular, es la manifestación más común de la enfermedad, con una prevalencia de entre el 80% y el 85% de los casos [14]. Se caracteriza por la aparición de los corpúsculos llamados drusas, entre la mácula lútea y la coroides. Esta manifestación es gradual, progresiva e irreversible, es decir, no existe un tratamiento para revertir el daño causado. La aparición de las drusas entre la coroides y la mácula interrumpe la correcta irrigación sanguínea a esta última, lo cual causa el deterioro en las células fotorreceptoras de la misma, provocando así el deterioro de la visión. En sus estadios tempranos las drusas se manifiestan como pequeñas manchas amarillas, cómo se aprecia en la Figura 1.2:

Figura 1.2 *DMAE: manifestación seca temprana.* Imagen accedida en[2]

En su estadio más avanzado, la presencia de drusas lleva a una condición conocida como atrofia geográfica, en la cual, el daño causa una pérdida total de la visión central. Se manifiesta como es mostrado en la figura siguiente:

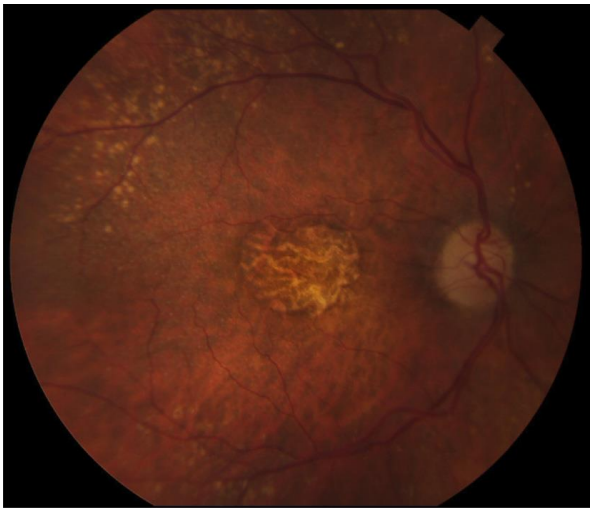


Figura 1.3 *DMAE: manifestación seca con atrofia geográfica.* Imagen accedida en[3]

[2] Jen Hong Tan, "Age-related Macular Degeneration detection using deep convolutional neural network", ELSEVIER, 2018.
[3] Catherine J. Thomas, "Age-Related Macular Degeneration", Medical Clinics of North America, 2021.

1.1.4.2 DMAE: manifestación húmeda

Esta manifestación se caracteriza por la presencia de neovascularización en la mácula, es decir, el crecimiento de vasos sanguíneos anómalos en la membrana de Bruch [8]. Esta manifestación tiene una prevalencia de entre el 15% y el 20%, pero es la que causa una pérdida más severa de visión [12].
Esta manifestación es la que causa una pérdida de visión más acelerada, sin embargo, está fuera del alcance del proyecto propuesto.

1.1.4.3 Mácula

La mácula lútea se encuentra en el fondo del ojo y es la estructura más central de la renina. Esta se encarga de la visión fina y la apreciación de los colores. Es la estructura con mayor densidad de células fotorreceptoras, en específico, conos. Por tener una densidad tan alta de células, esta requiere una excelente irrigación sanguínea que le proporcione suficiente oxígeno a las células fotorreceptoras. La ausencia o insuficiencia de oxígeno provoca la degradación de las células de la mácula, y por ende, una disminución en su función [13].

1.1.4.4 Drusas

El proceso metabólico del organismo genera desechos, los cuales son naturalmente eliminados por el cuerpo, sin embargo, con la edad esta capacidad de eliminación se va deteriorando, no siendo los ojos una excepción, es posible que en los desechos producidos en estos órganos se acumulen corpúsculos amarillos formados por lípidos y proteínas, conocidos como drusas. La presencia de drusas entre las membranas del ojo causa varias complicaciones, entre ellas, la interrupción de la correcta irrigación sanguínea entre la coroides y la mácula lútea, generando la degeneración de esta última [10], [12].

1.1.4.5 Fóvea Central

Es una pequeña área en el centro de la mácula lútea, que se aprecia como una depresión en la retina, esta es el área encargada de la apreciación de los detalles finos y la visión central, y que es afectada por la degeneración macular [14]. Lo descrito previamente se aprecia con y sin drusas en las Figuras 1.6 y 1.7 respectivamente:

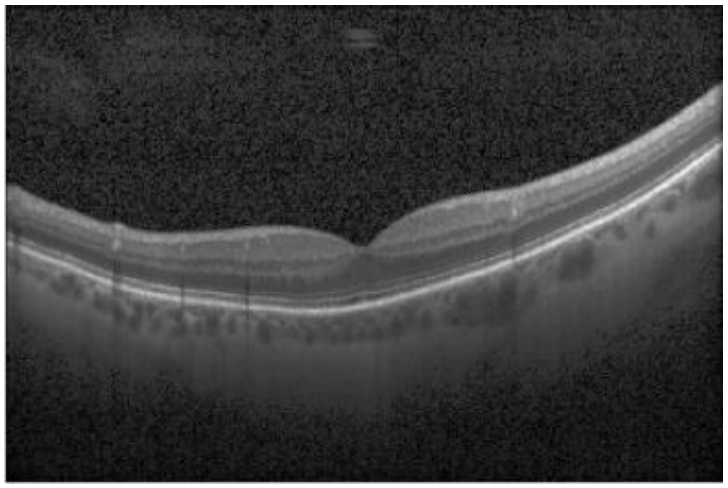


Figura 1.4 *Fóvea en imagen de ojo sano.* Imagen accedida en[4]

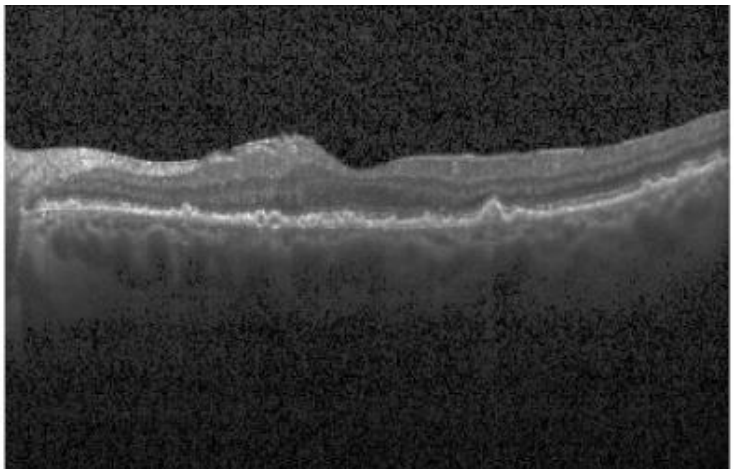


Figura 1.6 *Fóvea en imagen de ojo con drusas.* Imagen accedida en[4]

[4] Yao-Mei Chen, “Classification of age-related macular degeneration using convolutional-neural-network-based transfer learning”, BMC Bioinformatics, 2021.

1.1.4.6 Escotoma Visual

Es un área donde existe una pérdida total o parcial de la visión en el campo visual, que tiene múltiples orígenes, entre los que se consideran las lesiones en el ojo, lesiones en el cerebro, el glaucoma o como la causada por la degeneración macular. Es posible que los escotomas tengan múltiples formas y tamaños, y sean temporales, aunque en el caso de la degeneración macular en su manifestación seca, el escotoma es permanente, manifestándose como un punto ciego en la visión central [15].

1.1.5 Tomografía de Coherencia Óptica

Es una técnica médica no invasiva que utiliza luz a través de un láser de baja potencia para obtener imágenes de alta resolución de la sección transversal de la retina. Con esta técnica, los oftalmólogos tienen la posibilidad de ver y analizar cada una de las distintas capas de la retina, lo cual permite mapear y medir el grosor de esta, mediciones con las cuales se realiza un diagnóstico.

Para su realización, el paciente se coloca frente a una máquina OCT ("*Optical Coherence Tomography",* tomografía de coherencia óptica) y descansa la cabeza sobre un soporte en el cual se mantiene inmóvil. El equipo escaneará el ojo sin tocarlo y tal escaneo dura entre 5 y 10 minutos.

La captura de OCT es útil para la detección de muchas afecciones oculares, como las listadas a continuación: 1.-Agujeo macular. 2.-Fruncido macular. 3.-Edema macular. 4.-Glaucoma. 5.- Retinopatía Serosa Central. 6.- Retinopatía Diabética. 7.- Tracción vítrea y 8.- Degeneración macular asociada con la edad.

Esta última es la que busca clasificar el proyecto propuesto, por lo que, la clasificación de las OCT para el desarrollo del modelo es de mucha utilidad [16], [17].

1.1.6 Imágenes de Fondo de Ojo

Son imágenes de la cara interna posterior del globo ocular (retina) tomadas mediante una técnica médica conocida como oftalmoscopia, la cual se lleva a cabo de diversas maneras, como el uso de oftalmoscopios directos, indirectos o cámaras especiales para la fotografía de retina.

Los métodos anteriores tienen en común el uso luz apuntando a la retina y dispositivos ópticos para la obtención de la imagen.

Estas imágenes son una valiosa herramienta para los oftalmólogos, pues permite a los profesionales de la salud la examinación y el diagnostico de diversas condiciones y enfermedades manifestadas en el interior del globo ocular [18]. Tales afecciones incluyen, aunque no se limitan a: 1.-Retinopatía diabética. 2.-Glaucoma. 3.-Desprendimientos de Retina y 4.-Degeneracion Macular.

1.1.7 Aumento de Datos en Redes Neuronales

El aumento de datos es una técnica que mejora la calidad y la cantidad de datos a utilizar en el entrenamiento de un modelo de aprendizaje automático. Estas técnicas tienen como objetivo el lidiar con diferentes problemas que se presentan cuando la cantidad de datos que se tienen para trabajar es muy pequeña o muy poco variada, lo que implica no generalizar bien, es decir, obtener el llamado sobreajuste [19].

1.1.7.1 ¿Qué es el Sobreajuste en Aprendizaje Automático?

El sobreajuste sucede cuando un modelo estadístico se ajusta completamente a sus datos de entrenamiento, pero no a nuevos datos, cuando este fenómeno sucede, el modelo no es capaz de funcionar correctamente con datos con los que no se entrenó, es decir, no es capaz de generalizar bien aquello que está analizando. Este problema se presenta cuando el modelo es demasiado complejo o se entrena por demasiado tiempo con los datos de muestra. Existen varias técnicas para lidiar con este problema y uno de ellos es el aumento de datos [20].

Entonces, el aumento de datos es una técnica aplicada a una colección de datos, en el caso del alcance del proyecto, los datos son imágenes. Por lo que, las técnicas consisten en aplicar una serie de transformaciones a datos existentes y con ello, generar nuevos datos que ayuden a reducir los sesgos y mejoren la generalización del modelo.

Esta serie de técnicas es muy variada y cambia en función del software empleado para realizar las transformaciones (OpenCV, TensorFlow, YOLOv7, entre otros), tecnologías que se describen más a detalle en el capítulo 3. Sin embargo, las transformaciones más comunes en el proceso de aumento de datos son las siguientes: 1.- Rotación de la imagen. 2.-Zoom. 3.-Volteado de imagen vertical y horizontal. 4.-Reescalado de imagen. 5.-Recorrido de imagen. 6.-Agregar ruido a la imagen. [21].

1.2 Planteamiento del Problema

La degeneración macular asociada con la edad (DMAE) es una de las principales causas de pérdida de visión entre la población mayor de 50 años, afectando especialmente a los países desarrollados [9]. Siendo esta la tercera causa de ceguera alrededor del mundo, con una prevalencia moderada o severa estimada en 10.4 millones de personas según la OMS [11]. La DMAE causa la pérdida de la visión central, esto significa que no se aprecian los detalles finos ni cerca ni lejos en el centro del foco de visión, aunque la visión periférica se mantiene normal. La DMAE se asocia, además de la edad, con el sobrepeso, el tabaquismo, la hipertensión, enfermedades cardiacas y un alto consumo de grasas, condiciones que son muy comunes en países desarrollados y en México, donde más de 30 millones padecen hipertensión y más del 70% de la población presenta obesidad, según datos del gobierno de México y el Instituto Nacional de Salud Pública [22] [23]. Esta afección se divide en dos categorías, siendo la más común la DMAE “seca” (aproximadamente 80% de los casos) y que se caracteriza por la aparición de Drusas (cúmulos de grasa en la retina), y las DMAE “húmeda” que se produce por la aparición de vasos sanguíneos anormales bajo de la retina [12].

La enfermedad es altamente tratable es sus etapas tempranas, sin embargo, esta no presenta síntomas hasta que se encuentra en un estado intermedio, cuando el daño ya es irreversible, por lo que la creación de un módulo que ayude a la detección temprana del padecimiento se hace necesaria en el apoyo a la detección y diagnóstico de la enfermedad. La observación de imágenes de coherencia óptica es uno de los métodos más comunes para el diagnóstico de la afección. En este caso se propone a las técnicas de visión artificial y aprendizaje profundo como ayuda en el análisis de características en las imágenes de fondo de ojo, enfocándose en la categoría más común de DMAE, es decir, la denominada “seca”.

Actualmente, existen modelos de inteligencia artificial, principalmente basados en redes neuronales convolucionales, que abordan esta problemática. Sin embargo, en los últimos años, las técnicas de visión artificial han emergido como el nuevo estándar de referencia para el análisis de imágenes, dada su capacidad para integrar todo el contexto de la imagen analizada. Por tanto, esta propuesta se centra en el desarrollo de un módulo de inteligencia artificial para el análisis de características que permita la detección y clasificación de la Degeneración Macular Asociada a la Edad (DMAE), mediante el uso de técnicas de visión artificial (ViT) y aprendizaje profundo.

1.3. Objetivo General y Objetivos Específicos

1.3.1. Objetivo general

Desarrollar un módulo para el análisis de características para la detección temprana de la Degeneración Macular Asociada con la Edad utilizando técnicas de Visión Artificial y Aprendizaje Profundo que permita establecer el nivel de afectación en el ojo de la persona.

1.3.2. Objetivos específicos

1. Analizar trabajos relacionados para establecer la principal aportación del proyecto a través de la identificación de diferencias entre iniciativas existentes y el tema de tesis propuesto.

2. Analizar los algoritmos visión por computadora para la detección de elementos dentro de una región de interés en una imagen.

3. Identificar las características preponderantes a partir de imágenes del fondo del ojo para la identificación de Degeneración Macular Asociada con la Edad.

4. Identificar los repositorios de imágenes del fondo ojo más adecuados para la adquisición de imágenes que serán la entrada principal del módulo a desarrollar.

5. Diseñar y entrenar el modelo basado en transformadores de visión para la clasificación de valores multiclase que permitan establecer el nivel de afectación en el ojo de la persona.

6. Implementar los servicios e interfaces Web y el repositorio de información con los que se integrará el modelo entrenado para la identificación de Degeneración Macular Asociada con la Edad.

7. Llevar a cabo al menos un caso de estudio como prueba de concepto que permita describir los resultados y conclusiones obtenidas.

1.4 Justificación

La degeneración macular asociada con la edad es una de las afecciones oculares con mayor incidencia a nivel mundial y con mayor riesgo de provocar ceguera, a pesar de ser altamente tratable en sus etapas tempranas, pues esta no presenta síntomas hasta un estado intermedio, por tal, no se detecta a tiempo para iniciar su tratamiento y evitar la pérdida de visión. Dado que el envejecimiento demográfico va en constante aumento, es esperable que se convierta en un problema de salud cada vez más común, la cual es especialmente preocupante por dejar en un estado de incapacidad a aquellos que la padecen, y causar además una carga extra para

los sistemas de salud el estado en términos de recursos humanos, económicos y de infraestructura, añadido a repercusiones para los familiares del afectado o afectada. Por otro lado, su asociación con la obesidad y la diabetes la hace de especial relevancia para México, el cual presenta altas incidencias de ambas condiciones.

De lo expuesto anteriormente, se deriva la necesidad de desarrollar un mecanismo que permita detectar la degeneración macular asociada a la edad en sus etapas tempranas, mediante el análisis de imágenes de fondo de ojo. Con el avance de la inteligencia artificial como soporte para el diagnóstico médico y el advenimiento de las técnicas de visión artificial, la cual reconoce objetos en imágenes de una manera cada vez más precisa, la creación de modelos para la detección de enfermedades (en este caso una retinopatía) como herramienta para el trabajo de los oftalmólogos, resulta altamente provechoso.

La principal aportación de este proyecto reside en su capacidad para asistir a los especialistas de la salud en la detección temprana de la degeneración macular asociada con la edad en sus diversas etapas. La detección precoz es crucial para prevenir la pérdida total de visión en los individuos afectados por esta enfermedad. Por ello, se propone el desarrollo de un módulo que emplee técnicas de aprendizaje profundo y visión artificial para identificar características de la enfermedad en imágenes del ojo, lo cual permitirá determinar la presencia y el grado de avance de la patología.

Capítulo 2. Estado de la práctica

Se recopilaron una serie de artículos científicos relacionados al tema en diversas bases de datos de publicaciones científicas, como son: *Association for Computing Machinery*, IEEE, Elsevier, entre otras. Los términos utilizados para la búsqueda de los artículos científicos, fueron los siguientes o combinaciones de los mismos: "*age-related macular degeneration detection with machine learning, fundus image recognition with machine learning, diabetic retinopathies detection with machine learning, convolutional neural networks for eye diseases detection, vision transformer for eye diseases detection, age-related macular degeneration detection with vision transformer".*

La búsqueda de artículos se enfocó en aquellas publicaciones relacionadas con métodos de Aprendizaje Máquina (*machine learning)* para la detección de AMD (*Age-related Macular Degeneration*, Degeneración Macular Relacionada con la Edad). A saber, las redes neuronales convolucionales (CNN, *Convolutional Neural Networks,* por sus siglas en inglés) y Transformadores de Visión (*Vision Transformer*, ViT). El primero, siendo uno de los principales algoritmos utilizados en el campo de la inteligencia artificial, con diversas aplicaciones. Entre ellas, utilizado para la detección de retinopatías por medio del análisis de imágenes de ojos. Y el segundo (ViT), de aparición relativamente reciente, y siendo el nuevo "estado del arte" para el análisis de imágenes con inteligencia artificial, teniendo como ventaja el analizar el contexto completo de la imagen.

2.1 Trabajos relacionados

2.1.1 Clasificación de retinopatías mediante redes neuronales convolucionales.

El artículo [24] presentó un modelo de CNN (*Convolutional Neural Network*, red neuronal convolucional) para la detección temprana de AMD, la cual se evaluó usando estrategias de *cross-validation* y *blindfold*. Como conjunto de datos,

utilizaron 402 imágenes con fondo de ojo normales, 583 imágenes con evidencia de AMD en estados temprano, intermedio o GA (*Geographic Atrophy*, Atrofia Geográfica) también conocida como AMD seca y 125 imágenes con presunta AMD húmeda. La arquitectura que se propuso tiene un total de 14 capas, 7 capas convolucionales, 4 capas de agrupación máxima y 3 capas completamente conectadas. Para su procesamiento, las imágenes fueron reescaladas a una dimensión de 180x180. Las imágenes pasaron por las distintas capas de la arquitectura. Las capas de convolución tomaron las principales características de las imágenes del fondo de ojo. Las capas de operación de agrupación máxima tomaron los valores más altos de cada núcleo, reduciendo el tamaño de los mapas de características en cada salida. Y las capas completamente conectadas usan la función de activación *softmax* para la capa de salida. La computadora donde se entrenó el modelo propuesto contó con 2 *Intel Xeon E5-2650 v4* y 512GB de memoria RAM, no se utilizó ninguna GPU (*Graphics Processing Unit*, Unidad de procesamiento gráfico) para el proceso. Obtuvieron como resultados una precisión, sensibilidad y especificidad del 91.17%, 92.66% y 88.56% respectivamente utilizando la estrategia *blindfold*. En el caso *cross-validation* obtuvieron como resultados una precisión del 95.45%, sensibilidad del 96.43% y especificidad de 93.75%. Los autores concluyeron que su modelo obtuvo un alto nivel de precisión. Además de ser altamente portable y con una buena relación costo-beneficio.

En [25] propusieron el uso de aprendizaje profundo (*Deep Learning*) para la detección automática AMD (*Age-related Macular Degeneration*, Degeneración Macular Relacionada con la Edad) en imágenes SD-OCT (*Spectral Domain Optical Coherence Tomography*, Tomografía de Coherencia Óptica de Dominio Espectral). Para el entrenamiento, se utilizaron 1012 imágenes SD-OCT de sección transversal, 701 de las cuales estaban etiquetadas como AMD positivo, las 311 imágenes restantes estaban etiquetadas como muestras sanas, se reservaron 100 imágenes para pruebas (50 AMD positivas y 50 sanas). La computadora para el procesamiento de datos fue un MacBook Pro con un procesador *Intel Core i7* a 2.6GHz y 8GB de memoria RAM, con el sistema operativo *system OS X Yosemite*

Versión 10.10.5. (no se hace mención del uso de algún GPU). Para la creación del modelo de clasificación automática, se utilizó el programa *TensorFlow*, el cual provee de una DCNN (*Deep Convolutional Neural Network*, Red Neuronal Convolucional Profunda). Dicha DCNN se entrenó previamente con un conjunto de imágenes superior al millón, pertenecientes a distintas categorías. Por lo que se modificó la última capa de la misma para entrenar con el conjunto de imágenes correspondientes al estudio. Se realizaron 500 pasos para el entrenamiento y validación del modelo. Se realizaron las pruebas con el conjunto de 100 imágenes SD-OCT reservadas, obteniendo resultados del 99.7% ± 0.3% de precisión para los casos de AMD y 92.03% ± 8.5% para aquellas imágenes clasificadas como saludables. Por tanto, los autores concluyeron que es viable utilizar un modelo de aprendizaje profundo con una alta sensibilidad (100%), especificidad (92%) y precisión (96%).

El artículo [26] propuso un método basado en una versión modificada del *framework U-Net*, la arquitectura *m-Net* y DFS (*Depth-First-Select Graph*, Gráfico de Selección de Profundidad Primero) para la detección del disco óptico (OD por sus siglas en inglés), mediante la combinación de localización y la segmentación del OD. Los autores escribieron que la arquitectura *m-Net* contiene dos componentes: codificador y decodificador. El codificador del *U-Net* consiste en dos capas convolucionales y una capa de agrupamiento, *m-Net* adopta los bloques de red residual como cuerpo principal. La red residual se aplicó con éxito en la detección de objetos en imágenes y agrega atajos de conexión entre las capas vecinas. El decodificador toma dos entradas, la salida de la última capa en el codificador y la capa correspondiente con el codificador del mismo tamaño. Dado a que m-Net permite obtener más de un resultado como OD, se utilizó el algoritmo DFS para indicar que localización tiene la máxima posibilidad de ser el OD. Se realizaron las evaluaciones sobre dos conjuntos de datos: *ORIGA* (650 imágenes) que se dividió en grupos de entrenamiento y prueba (325 respectivamente). Y *Messidor* (1200 imágenes) dividido en grupos de 600 imágenes para entrenamiento y prueba. Ambos conjuntos llevaron un preprocesamiento de re escalado de imagen y de

rotación para aumentar el volumen de datos. El algoritmo para el procesamiento de imágenes se implementó en *Python* y el marco de trabajo de *Tensorflow* para el entrenamiento de la CNN. El banco de pruebas tenía el sistema *Ubuntu16.04* y una *GPU NVidia GTX TITAN XP*. Los resultados obtenidos alcanzaron una precisión del 100% para ORIGA y un 99.8% para *Messior*. Los investigadores agregaron que su método permite su aplicación para otros algoritmos de diagnóstico de imagen, tales como glaucoma y la degeneración macular relacionada con la edad.

En el artículo [27] se propuso un esquema automatizado para la clasificación de las exudaciones suaves y duras basado en CNN. Esto extrayendo las secciones donde existen exudaciones del resto de la imagen del fondo del ojo. Además, se separó el parche con la exudación del fondo de la imagen, creando así dos conjuntos de imágenes para entrenar el modelo CNN. Se evaluó el algoritmo de clasificación con el conjunto de 550 imágenes, 275 de exudaciones suaves y 275 fuertes (existen dos conjuntos aun, con y sin el fondo recortado). Los conjuntos se dividieron al azar en conjuntos de entrenamiento (220 suaves y 220 fuertes) y pruebas (55 suaves y 55 fuertes). El modelo propuesto se llamó *ExudateNet*, tiene 5 capas que extraen las características de la imagen usando diferentes números de núcleos 3x3, acompañados por una unidad lineal rectificadora. Se siguió el procesamiento de los mapas de características mediante otras capas de agrupación de 2x2 para reducir la dimensión de la imagen. Los mapas de características se transforman en un vector de características mediante otras dos capas conectadas. La capa final, que proporciona la salida tiene dos neuronas que entregan la clasificación. Los experimentos se realizaron en una computadora con un CPU *Intel(R) Xeon(R) E5-1620* a 3.50GHz, 8GB de RAM y un GPU *NVidia GeForce GTX 1050*. El modelo se entrenó de la siguiente manera: con el conjunto de imágenes de parches de exudación con fondo (*ExudateNet1*), con el conjunto sin fondo (*ExudateNet2*), y una mezcla de los anteriores (*ExudateNet3*). Los resultados revelaron que las imágenes con fondo tienen un mejor desempeño que aquellas sin fondo, por lo que el área que rodea al área estudiada es importante para el algoritmo. Sin embargo, el conjunto *ExudateNet3*, obtuvo un desempeño mejor a los dos anteriores,

obteniendo una precisión del 93.41% sobre los 90.80% y 87.41% de *ExudateNet1* y *ExudateNet2* respectivamente.

En el artículo [28] se tuvo como propósito la validación del sistema de aprendizaje profundo *RetCad v.1.3.0*, sistema comercialmente disponible para la clasificación de DR y AMD en imágenes de fondo de ojo (imágenes las cuales presentan afecciones mixtas). Con el propósito de evaluar la capacidad del sistema para la detección simultánea de las dos condiciones, para compararlo con otros sistemas de clasificación y humanos expertos. Tomaron dos conjuntos de datos para la evaluación, siendo estos *Messidor* (1,200 imágenes) y AREDS (*Age-Related Eye Disease Study*, Estudio de Enfermedades Oculares Relacionadas con la Edad) el cual contaba con 133,821 imágenes. Para la evaluación se realizaron diferentes experimentos sobre los conjuntos de datos. Se realizaron pruebas de clasificación binaria en distintas combinaciones, como DR vs AMD (más controles) y AMD vs DR (más controles). Como resultados para la validación conjunta de DR-AMD se obtuvo una precisión del 95.1% para la detección de casos de DR, en el caso de AMD se obtuvo una precisión del 94.9%. Respecto al conjunto de datos *Messidor*, se obtuvo una precisión del 97.5% en la clasificación de DR. Para el caso de AREDS se obtuvo una precisión del 92.7% en la clasificación de AMD. Tales resultados fueron cercanos a los promedios humanos, los cuales son de 97.5% de precisión para DR y 96.1% para AMD. De lo anterior los investigadores concluyeron que el sistema tiene un desempeño similar al humano y es una herramienta de apoyo para los expertos en el diagnóstico de enfermedades oculares.

En la publicación [29] se evaluó una metodología para el reconocimiento de AMD en imágenes del fondo del ojo mediante el procesamiento de Imagen y 4 modelos de Aprendizaje Máquina: *Naive Bayes*, *Neural Network*, *Support Vector Machine* y *Random Forest*. El proceso realizado se da en el siguiente orden: se recopilaron imágenes de fondo de ojos, usando una cámara retinal *TOPCON TRC-NW8F*. Una cantidad de 13,105 imágenes sin marcar. Del conjunto de imágenes se etiquetan 532 como positivo para AMD, más 52 imágenes etiquetadas como normales. Para

evitar el sobreentrenamiento, se agregaron imágenes de ojos normales al conjunto, provenientes de otras bases de imágenes (*STARE*, *DRIVE* y *Dr. Hossien Rabbani's retinal database*), quedando el conjunto de imágenes de condición normal con 18 elementos y se redujo a 268 al conjunto positivo a AMD. Posteriormente se llevó a cabo un preprocesamiento de imágenes. Este consistió en un ajuste de intensidad, mediante el método de multiplicación de píxeles, un filtrado bilateral (técnica para la reducción de ruido en la imagen), una detección y extracción del disco óptico de la imagen (pues es posible que su presencia cause confusión con drusas). Como último paso se implementó SLIC (*Simple Linear Iterative Clustering*, agrupamiento iterativo lineal simple). El siguiente paso de la metodología fue la extracción y normalización de características. Para el entrenamiento y clasificación se dividió el conjunto de imágenes en una proporción de 70/30 para entrenamiento y pruebas respectivamente. El entrenamiento se realizó sobre los 4 modelos mencionados, aplicando *cross-validation* de 10 pliegues. Se realizó el entrenamiento 2 veces: con el conjunto de imágenes con el preprocesamiento completo y con el conjunto sin aplicar SLIC en el preprocesamiento. Los resultados presentados señalan un mejor comportamiento cuando SLIC es aplicado. Los resultados con SLIC, del mejor al peor fueron: *Neural Network* 95.61%, *Random Forest* 94.74%, *Naive Bayes* 91.23% y *Support Vector Machine* 90.35%.

La propuesta de los autores en el artículo [30] fue un método automático para la segmentación de la ILM (*Inner Limiting Membrane*, Membrana Limitante Interna), el RPEDC (*Retinal Pigment Epithelium and Drusen Complex,* Epitelio Pigmentario Retinal y el Complejo de Drusas) y los límites de capa de BM (*Bruch's Membrane*, Membrana de Bruch) en imágenes OCT (*Optical Coherence Tomography*, Tomografía de Coherencia Óptica) positivas a AMD (*Age-related Macular Degeneration*, Degeneración Macular Relacionada con la Edad). Para su estudio, los autores usaron en su método a lo que ellos denominaron DF-LS (*Deep Forest for Layer Segmentation*, Bosque Profundo para la Segmentación de Capas) el cual se entrena con segmentos de imágenes. Se usaron 3 conjuntos de datos (uno con imágenes sanas y dos con presencia de AMD), que en su conjunto dieron un total

de 489 imágenes. La arquitectura cuenta con tres pasos principales, los cuales son el preprocesamiento, la clasificación de límites de capa y la refinación de límites de capa. Para el preprocesamiento se utilizó una normalización intensiva, la cual mantiene la diferencia de escala de grises de las distintas capas de la retina mientras reduce (incluso elimina) la escala de gris inconsistente. Para el proceso de clasificación de límites de capa, se encontraron regiones cercanas al límite de capa. Durante la clasificación, se extrajeron parches de la imagen. El modelo DF-LS se entrenó con dichos parches. Ya en el tercer paso, utilizaron una versión modificada de GTDP (*Grapy Theory and Dynamic Programming*, Teoría de Grafos y Programación Dinámica) para encontrar los límites de las capas, para el proceso de refinamiento de los límites de mapas de probabilidad. Los resultados que apuntan los autores demuestran que su método es efectivo tanto en imágenes con presencia de patologías como en imágenes de ojos sanos.

En el trabajo [31] los autores presentaron una plataforma basada en *Deep Learning* para la detección de diversas anomalías retinianas. El modelo se entrenó con imágenes OCT (*Optical Coherence Tomography*, Tomografía de Coherencia Óptica), además utilizó *data augmentation* y transfer training para paliar los desafíos del *Deep Learning* en oftalmología. Se entrenó al modelo en seis categorías: neovascularización coroidea (*Choroidal Neovascularization*, CNV por sus siglas en inglés), edema macular diabético (*Diabetic Macular Edema*, DME), drusas, retina normal, retinopatía serosa central (*Central Serous Retinopathy*, CSR) y agujero macular (*Macular Hole*, MH). Las imágenes se obtuvieron de cuatro fuentes, tres de las cuales por sí solas no contenían todas las categorías. La más grande, con más de 200 mil imágenes tenía las categorías de normal, CNV, DME y drusas. Los autores escribieron que se realizó a propósito para causar un problema de desbalance. Para aumentar el número de muestras de imágenes, se llevó a cabo un proceso de *data augmentation*, que consistía en una transformación al azar de las imágenes durante cada época de entrenamiento. Se usó un modelo pre entrenado *VGG16*, que usa imágenes con tres canales de RGB y 224x224 pixeles. El modelo tiene 13 capas convolucionales y 3 capas completamente conectadas.

Fue previamente entrenado con más 14 millones de imágenes de *ImageNet*, de 1000 clases diferentes. El modelo se modificó para ajustarse a las necesidades de clasificación, la última capa se reemplazó por una capa lineal totalmente conectada de seis neuronas de salida (una por clase). El clasificador se desarrolló con *Python* y *PyTorch*, y entrenado usando un GPU *Nvidia GTX 1080 Ti*. Se evaluaron varios modelos. El modelo con mejores resultados fue el *VGG16 CNN* con *transfer training*. Comparado con los demás modelos, demostró una mejora en la precisión y en *test loss*, con una precisión promedio del 96%, siendo su presión más alta las clasificaciones de CVN y retina normal con un 100% y la más baja MH con 83.3%.

En la publicación [32] se analizaron 4 modelos de aprendizaje profundo altamente difundidos, diseñados para la segmentación de tres fluidos retinales: líquido intrarretiniano (*Intraretinal Fluid*, IRF por sus siglas en inglés), líquido subretiniano (*Subretinal Fluid*, SRF) y desprendimiento del epitelio pigmentario (*Pigment Epithelial Detachment*, PED). Para demostrar que un enfoque basado en parches (secciones de la imagen), mejora el rendimiento en cada método. Los modelos analizados fueron: FNC (*Fully Convolutional Networks*, Redes Completamente Convolucionales), *U-Net*, *Seg-net* y *Deeplabv3+*. Se usaron dos conjuntos de datos de acceso público, *RETOUCH* y *OPTIMA*, obteniendo un total de 100 volúmenes de datos, en una proporción de 70/30 para *RETOUCH* y *OPTIMA* respectivamente. Los volúmenes pasaron por un proceso de preprocesamiento que incluía el redimensionamiento y eliminación de ruido de las imágenes. Para el estudio se utilizaron dos computadoras con diferente hardware, una de ellas contaba con un procesador *Intel i7* de séptima generación, un GPU *Nvidia GTX 1080Ti* y 128 GB de memoria RAM. La segunda de ellas tenía un procesador Intel i5, un GPU Nvidia GTX 1070 Ti y 16 GB de memoria RAM. En cuanto al software, ambos dispositivos contaban con *Cuda* versión 10 y *cuDNN* en la versión 7.5; las arquitecturas se codificaron con el marco de trabajo de *MATLAB*. Los resultados de los experimentos muestran que *Deeplabv3+* y *U-net* extendido tuvieron un desempeño superior a los otros dos modelos (aunque no tuvieron un mejor rendimiento en todas las clases). Por tanto, se concluye que el enfoque basado en parches mejoró el rendimiento de

los modelos cuando se les entregan imágenes con diferentes resoluciones espaciales.

El artículo [33] propuso un método para la mejora adaptativa de las imágenes de fondo de ojo, restaurando tanto color como ajustando brillo de la imagen. El algoritmo de proceso de imagen convencional incrementó el contraste sin perder el matiz de color. El método incluyó tres pasos: mejora de iluminación, incremente de contraste y restauración de color. Para el primer paso se convirtió la imagen RGB en una imagen de intensidad mediante un algoritmo de decoloración que conserva el contraste en tiempo real. Para el siguiente paso la imagen obtenida se pasó por un filtro bilateral, usado un núcleo convolucional para adquirir la iluminación de la imagen. El filtro conservó la información tanto de los contornos como de las texturas de las características de la imagen. El siguiente paso fue el método de restauración de color para obtener una imagen sin distorsión. Se realizó un experimento del modelo con diez imágenes (con poca iluminación) elegidas al azar de dos bases de datos públicas. El desempeño se evaluó en los aspectos de brillo, contraste y matiz de color. El modelo se comparó con otros métodos de mejora de imagen, en los aspectos de brillo, distribución de matiz de color y en el error cuadrático medio. El método obtuvo un resultado competitivo comparado con los otros modelos. El modelo fue capaz de mejorar el contraste en un 132.37% y el brillo en un 86.35% para los diez casos de prueba. Los autores señalaron que esto ayudará a la mejor identificación de características de patologías en los ojos.

En la publicación [34] se propuso una Red Neuronal Convolucional con capacidad de transferencia de aprendizaje (método que reduce el tiempo y recursos computacionales, así como la experticia necesaria para la resolución de problemas), y con hiper parámetros adecuados para la clasificación de AMD (*Age-related Macular Degeneration*, Degeneración Macular Relacionada con la Edad) y DME (*Diabetic Macular Edema,* Edema Macular Diabético) a través del análisis de imágenes OCT (*Optical Coherence Tomography*, tomografía de coherencia óptica). El conjunto de datos que se utilizó contaba con 4 clases (drusas, macular

relacionada con la edad, neovascularización coroidea y normales) y se dividió de la manera siguiente: 83,484 imágenes para el entrenamiento, de las cuales 37,205 eran imágenes de CNV (*Choroidal Neovascularization*, Neovascularización Coroidea), 11,384 imágenes de AMD, 8,616 imágenes de drusas y 26,315 imágenes clasificadas como normales. El conjunto de pruebas contó con 968 imágenes, segmentado en cuatro con 242 imágenes por clase. En cuanto al método, para el proceso de transferencia de aprendizaje se utilizó un modelo de CNN pre-entrenado como punto de arranque para la CNN propuesta. Los hiper parámetros (parámetros configurados antes del proceso de aprendizaje) fueron algoritmos que alteraron la velocidad y calidad del aprendizaje. En el desarrollo de la investigación, se utilizaron los siguientes modelos de CNN: *Alexnet* de 8 capas, *Googlenet* de 22 capas, *VGG* de 16 capas, *VGG* de 19 capas, *Resnet* de 18 capas, *Resnet* de 50 capas y *Resnet* de 101 capas. Se hicieron 5 corridas independientes de experimentos se obtuvieron los siguientes resultados para la clasificación de imágenes: *Alexnet* 95.5%, *Googlenet* 95.31%, *VGG16* 95.94%, *VGG19* 99.42%, *Resnet18* 98.47%, *Resnet50* 99.09% y *Resnet101* 99.19%.

El trabajo del artículo [35] propuso un acercamiento basado en aprendizaje profundo para la detección de distintas clases de anomalías en la retina, mediante el análisis de imágenes OCT (*Optical Coherence Tomography*, Tomografía de Coherencia Óptica). Tales anomalías no estando limitadas a las clases definidas en el conjunto de datos de entrenamiento, es decir, el sistema es capaz de detectar nuevas anomalías desconocidas durante el diseño y entrenamiento. Como conjunto de datos utilizaron la base pública *UCSD V3* y la base *Kaggle*, las cuales tienen las categorías de NORMAL, DME (*Diabetic Macular Edema,* Edema Macular Diabético), Drusas y CNV (*Choroidal Neovascularization*, Neovascularización Coroidea). Además, el conjunto de datos *OCTID* se agregó para la evaluación del modelo AD (*Abnormality Detector*, Detector de Anomalías). Mientras que la red neuronal se diseñó con *Keras API* bajo el marco de trabajo *TensorFlow*. Para el entrenamiento y pruebas se utilizó *Jupyter Notebook* con *Google Colab* con soporte de GPU como interfaz. Se propusieron dos modelos para la detección y clasificación

de anomalías, el AD y el CL (*Classification Model*, Modelo de Clasificación). Tales modelos se crearon usando el modelo CD (*Class Detector*, detector de clases) el cual compara la imagen OCT de entrada usando el modelo FE (*Feature Extractor*, Extractor de Características) y el modelo DIS (discriminador). Los resultados presentados para el detector de anomalías fueron del 94.59%, 99.8% y 99.9% de precisión en los conjuntos de datos *OCTID*, *UCSD* y *Kaggle* respectivamente. También se probó el desempeño del modelo clasificador, el cual entregó resultados del 97.7% y 99.79% para *UCSD* y *Kaggle* respectivamente.

Lo propuesto en [36] fue un *framework* para eliminación de ruido en las imágenes OCT (*Optical Coherence Tomography*, Tomografía de Coherencia Óptica) basado en *Deep Learning*, mediante la implementación de CNNs y *Deep Neural Networks* (DNNs). El *framework* gira en torno a la creación de imágenes de referencia limpias para alimentar la CNN a partir de eliminadores de ruido, permitiendo la eliminación de ruido de las imágenes OCT sin la necesidad de una gran base de datos de imágenes limpias. Se introdujeron imágenes sin retocar a través de 4 eliminadores de ruido de última generación (obteniendo cuatro diferentes salidas): *BM3D*, *BM3DDEB*, *Weiner* y *HWT*. Posteriormente se colocó la imagen original y las 4 imágenes resultantes en una red que entregó una imagen significativamente pulida. Se probaron diferentes arquitecturas: *Autoencoder*, *DNCNN*, *GAN*, *DenseNet*, *YOLO*, *SqueezeNet* y *HighwayNet*. Estas se modificaron para recibir y entregar una imagen. La base de datos consistía en 18 sujetos con ojos sanos y ojos afectados por AMD, con un tamaño de 500x900 pixeles. Las CNN se implementaron con diferentes ratios de aprendizaje (5.0×10-3, 1.0×10-3, 1.0×10-4, 1.0×10-1, 1.0×10-2), épocas (200, 500 y 1000), lotes (2 y 4) y tamaños de datos de prueba (18 y 4) para obtener los resultados óptimos. Como métricas de prueba se tenían: pico de relación señal-ruido, relación contraste-ruido y número equivalente de vistas. *Autoencoder* y *DenseNet* obtuvieron los mejores resultados. Destacando *DenseNet* por conseguir remover el ruido de fondo y suavizar el frente de la imagen. *DenseNet* se implementó con ratio de aprendizaje de 10x10-4, 500 épocas y lotes de 2 (este último fue general para las CNN). El *framework* se comparó con *BM3D* y *NLM*,

obteniendo mejores resultados en las 3 métricas. Se concluyó que el método propuesto obtiene resultados satisfactorios entrenando con una sola imagen y que permite su implementación para otras imágenes OCT.

En el artículo [37] propuso un marco de trabajo para detectar enfermedades en los ojos a través de la extracción de las venas retinianas en una imagen del ojo. Utilizaron sistemas de tamizado para la eliminación de la conmoción e impedancia ecológica de la imagen. Tal marco de trabajo se ejecutó con *MATLAB* y eliminó elementos indeseables de la imagen a la vez que aplicó cálculos para la extracción de los vasos sanguíneos de la imagen. Los autores mencionan que su algoritmo tiene la ventaja de ser aplicable a cualquier tipo de imagen retinal (además de ser muy útil para la detección temprana de diabetes); trabajaron con un conjunto de 100 imágenes retinales. La arquitectura de su sistema contaba con 2 grandes fragmentos, en primer lugar, el preprocesamiento de imágenes y extracción de características, en segundo lugar, el proceso de aprendizaje automático que utilizó detección de patrones. El primer paso fue la conversión RGB, donde la imagen es transformada en 3 diferentes imágenes en rojo, verde y azul. Luego la imagen es re escalada y el ruido es retirado. Posteriormente la imagen se pasa a escala de grises. En el proceso de aprendizaje supervisado el conjunto de datos no estaba etiquetado, por tal motivo utilizaron los métodos *K-means* y *fuzzy C-Means*. En el paso final se realiza un proceso de validación usando un algoritmo de agrupamiento de vectores de soporte, para detectar si el paciente padece de diabetes o no. Teniendo un total de 5 pasos. Los autores concluyen que su sistema es capaz de diferenciar imágenes saludables de aquellas que presentan retinopatías como la degeneración macular asociada con la edad y la retinopatía diabética. Sin embargo, el hallazgo más importante de la investigación es que el sistema distingue entre clases, evitando la segmentación previa de las lesiones retinales.

La publicación [38] comparó el desempeño y aplicabilidad de una CDNN (*Convolutional Deep Neural Network*, Red Neuronal Profunda Convolucional) de multitarea para la detección de nAMD (*Neovascular Age-dependent Macular*

Degeneration, Degeneración Macular Neovascular Dependiente de la Edad), que detecta simultáneamente IRF (*Intraretinal Fluid*, Líquido Intrarretiniano) y SRF (*Subretinal Fluid*, Líquido Subretiniano). Para el estudio, se realizó la recolección de datos, en este caso OCT (*Optical Coherence Tomography*, Tomografía de Coherencia Óptica) provenientes de un grupo de 70 pacientes (46 mujeres y 24 hombres), de los cuales se obtuvieron un total de 3,762 imágenes (2011 del ojo derecho y 1751 del ojo izquierdo) de 440 x 512 píxeles. Se llevaron a cabo procesos de aumento de datos y preprocesamiento. Para el aumento de datos durante el entrenamiento, se utilizó *mixup*, tecnología la cual genera ejemplos artificiales mediante la combinación aleatoria de puntos de datos muestreados. También se manipularon la escala y el brillo de las imágenes. En el estudio se comparó el modelo multitarea con modelos monotarea y se desarrollaron tres DNN. De las observaciones se extrajo que el modelo multitarea sobrepasó el desempeño de las redes monotarea en la precisión para la detección de AMD. Se obtuvieron los siguientes resultados: 91.7% en detección de SRF, 93.7% en la detección de IRF y para la detección de nAMD se tuvo una precisión del 94.2% (el modelo monotarea sólo alcanzó una precisión del 91.4%). Detalles más profundos sobre la arquitectura del modelo no se reportaron por los investigadores.

En el artículo [39] los autores propusieron una DCNN (*Deep Convolutional Neural Network*, Red Neuronal Convolucional Profunda) con una arquitectura de 13 capas, para distinguir automáticamente imágenes con presencia de DMAE de imágenes con la ausencia de esta. Los conjuntos de datos utilizados para la investigación fueron *ARIA* y *IChallenge-AMD*. *ARIA* es una base de datos con imágenes de fondo de ojo a color. El estudio se enfocó en la clasificación binaria, por lo que solo se usaron las imágenes con DMAE (23 imágenes) y de control (61 imágenes). Para el caso de *IChallenge-AMD*, se tomaron del conjunto 89 imágenes de DMAE y 311 de control. Para el entrenamiento del modelo DCNN se utilizó una computadora con un procesador *Intel Core i5-8500* a 3.00GHz y 16GB de memoria RAM (no se utilizó ningún apoyo de GPU). La arquitectura de la DCNN propuesta es la siguiente: 5 capas convolucionales, 5 capas de máxima agrupación y 3 capas completamente

conectadas. Las capas convolucionales consisten en una serie de filtros que extraen características de la imagen de entrada y convolucional con la capa actual. De ellas se obtienen mapas de características que pasan por las capas de máxima agrupación, que toma las características más nítidas y reduce la dimensión de la salida. Las capas completamente conectadas utilizan la función de activación *softmax*, que calcula las posibilidades de la clase de salida. El modelo propuesto se entrenó, desarrolló y validó en *Python* usando *TFLearn* y *TensorFlow*, Las métricas de desempeño se calcularon con *Scikit-leand* y las gráficas se generaron usando *Matplotlib*. Los resultados entregaron una precisión de clasificación del 89,75 %, 91,69 % y 99,45 % en las versiones original y aumentada de *iChallenge-AMD* y 90,00 %, 93,03% y 99,55% para *ARIA*, utilizando una técnica de *cross-validation* de 10 veces.

2.1.2 Clasificación de retinopatías mediante Vision Transformer.

En el artículo [40] se revisaron y analizaron métodos de *Deep Learning* de última generación para la detección de clasificación de DR. Estos fueron métodos supervisados, auto supervisados y *transformer*. Para la evaluación de los modelos se hizo una categorización para aquellos que hacían una predicción binaria (presencia o ausencia de DR) y aquellos que hacían una predicción multiclase (graduación de la severidad de DR). Se utilizaron distintos conjuntos de datos para la comparación de los diferentes modelos y técnicas de clasificación, estos fueron: 1. *EyePACS 2015*, 2. *APTOS 2019*, 3. *Messidor*, 4. *Messidor-2*, 5. *IDRiD*, 6. *DRIVE*, 7. *DIARETBD1*, 8. *DIARETDB0*, 9. *ODIR*, 10.*DDR* y 11. *RFMiD*. Se revisaron 11 artículos de CNN supervisada, 3 de CNN auto supervisada y 4 *transformers* (de los modelos *transformer* evaluados, 3 utilizaban *vision transformer* en su arquitectura) mediante el análisis de técnicas y resultados de los respectivos métodos propuestos. De los *transformers* se destaca su correlación positiva con el número de parámetros de entrenamiento y la precisión, su inmunidad a la saturación con grandes conjuntos de datos y una distribución de datos variada. También su capacidad de tener un entendimiento global de la imagen a diferencia de los que sucede con las CNN. Se remarcó su ventaja obtenida mediante sus mecanismos de

atención y su capacidad de detectar lesiones más pequeñas y con más detalle que las CNN. En sus conclusiones, respecto a los *transformers*, los autores escribieron que estos introdujeron nuevos métodos que ayudan a superar las limitaciones de la no generalización y que nuevos indicadores ocultos permiten su detección gracias a acercamientos de contexto enriquecido.

En el artículo [41] se propuso un mecanismo de *Deep Learning* llamado *lesion-aware transformer* (transformador consciente de la lesión o LAT por sus siglas en inglés) para la detección y clasificación de DR, en conjunto con un modelo profundo unificado, mediante una estructura de codificador-decodificador. El codificador, es el codificador de relación de píxeles, diseñado para adaptarse a las variaciones de apariencia de los píxeles mediante el modelado de correlación de los mismos. El modelado de la correlación de pixeles se creó con la intención de capturar la información del contexto completo de una imagen y para la generación de un mapa de características mejorado. Por otra parte, se creó un decodificador basado en un filtro de lesiones, para la detección de las diferentes regiones con lesiones. Como base para su red de extracción de características, los autores utilizaron *ResNet50*. Removieron la capa de conjunto global promedio y la capa completamente conectada. Para los experimentos del proyecto, se utilizaron 3 conjuntos de datos: 1. *Messidor-1* con 1,200 imágenes de fondo de ojo, 2. *Messidor-2* con 1,748 y 3. *EyePACS* con 35,126 imágenes de entrenamiento, 10,906 imágenes de validación y 42,670 imágenes de pruebas. Las imágenes se transformaron a una escala de 512x512 píxeles. El conjunto de imágenes fue aumentado mediante giros horizontales y verticales al azar. El modelo propuesto se comparó con diversos modelos para la clasificación de DR. Las comparaciones se hicieron sobre los tres conjuntos de datos. En todos los conjuntos de datos, el modelo propuesto superó a aquellos modelos con los que se comparó por un considerable margen. En *Messidor-1* obtuvo precisiones del 98.7% y 96.3% para las clasificaciones de remisión y normales respectivamente. En *EyePACS* tubo una precisión del 89.3% en el conjunto de validación y 88.4% en el conjunto de prueba. Se evaluó su capacidad para el descubrimiento de lesiones sobre *Messidor*-2, donde se

obtuvieron resultados positivos. Se concluye que LAT obtiene resultados favorables en la clasificación de DR comparado con otros acercamientos recientes al problema.

La propuesta presentada en [42] fue un algoritmo *Deep Learning* para la detección de neMNV (*Nonexudative Macular Neovascularization*, Neovascularización Macular no Exudativa) mediante la detección de DLS (*Double-Layer Sign*, Señal de Doble Capa) también conocido como "elevación superficial irregular del epitelio pigmentario de la retina", basado en el análisis de imágenes estructurales OCT. Usaron un modelo de segmentación basado en ViT que se entrenó usando imágenes de ojos sanos o con una categoría de neMNV. Se utilizaron 251 ojos de 210 pacientes, 182 de los cuales presentaban DLS, y 115 ojos con presencia de drusas. De un total de 125,500 imágenes, 5,256 se utilizaron para entrenamiento y 1,623 para validación. El modelo propuesto se construyó usando una arquitectura codificador-decodificador completamente basada en ViT. El proceso de la arquitectura fue el siguiente: Se ingresó una imagen que se dividió en una secuencia de parches, los cuales se convirtieron en vectores unidimensionales y usados para alimentar una capa que producía una secuencia de parches embebidos. Se agregaba la información posicional para obtener una secuencia de tokens. El codificador *transformer* generaba una secuencia contextualizada. El decodificador aprendía a mapear las codificaciones a nivel de parche provenientes del codificador, a etiquetas de clase a nivel de parche. Se aplicó una capa puntual lineal para producir *logits* de clase a nivel de parche. Se formó la secuencia en un mapa bidimensional de características y se sobre muestreó mediante interpolación bilineal. Se utilizó *softmax* en la dimensión de clase para obtener el mapa de segmentación final. El modelo alcanzó una sensibilidad, especificidad, valor predictivo positivo y valor predictivo negativo del 82 %, 90 %, 79 % y 91 %, respectivamente. Se concluyó que el modelo propuesto tuvo un desempeño consistente y confiable.

En el artículo [43] se integró *vision transformer* con OCT (*Optical Coherence Tomography*, Tomografía de Coherencia Óptica) para mejorar el diagnóstico de

retinopatías, apuntando principalmente AMD (Age Related Macular Degeneration, Degeneración Macular Asociada con la Edad) y DME (*Diabetic Macular Edema*, Edema Macular Diabético). Se recolectaron imágenes OCT de ambas retinopatías, así como de ojos normales para realizar el entrenamiento. El conjunto de datos contó con 1407 imágenes OCT de ojos normales, 723 con la condición AMD y 1101 con DME. Se realizó un preprocesamiento de las imágenes, ajustando su resolución a 224x224 píxeles. Como entorno de desarrollo se utilizó una computadora con un procesador *Intel Core I7 9700f*, un GPU *Nvidia RTX2060 super* y 16GB de memoria RAM. En cuanto al software, la computadora utilizó Windows 10, *Python* 3.7 y el *framework* de aprendizaje automático *PyTorch*. El conjunto de datos se dividió en un conjunto de entrenamiento, un conjunto de validación y un conjunto de pruebas en una proporción de 8:1:1. Se utilizó *ranger optimizer* como optimizador, configurando la ratio de aprendizaje en 0.0003. Entrenaron su modelo 100 veces. Guardaron los parámetros del modelo con la mayor precisión del modelo en el conjunto de validación. La prueba se presentó en el conjunto de pruebas y se obtuvo la precisión de tal conjunto. Para ambas retinopatías, se obtuvo una precisión en la predicción del 99.69%. Comprobaron que *vision transformer* tiene la mejor habilidad de reconocimiento que las CNN tradicionales (compararon su modelo con VGG16, Resnet50, Densenet121 y EfficentNet) además, tiene una mayor velocidad de reconocimiento.

El trabajo [44] se estudiaron diferentes métodos de *Deep Learning* (CNN y arquitecturas de *transformers*) para el diagnóstico diferentes estados de AMD. Además, presentaron dos métodos para usar modelos de graduación de AMD como modelos binarios. El conjunto de datos de imágenes retinales se creó de la base de datos *UpRetina*. El conjunto de datos contaba con 4,876 imágenes sin AMD, 2803 evaluadas con AMD temprano, 1,563 en estado intermedio y 530 en estado avanzado. Todas las redes se implementaron con *Pytorch* y se entrenaron con las bibliotecas *FastAI* y *Timm*. Todo con una GPU *Nvidia RTX 2080 Ti*. Respecto a las arquitecturas analizadas, se incluyeron 8 CNN y 9 arquitecturas basadas en *transformer*. Para ambos tipos de arquitectura se usaron los siguientes

hiperparámetros: 1. Lotes de tamaño 64, 2. Ratio de aprendizaje de 3e-3, 3. 100 épocas y 4. Paciencia parada temprana = 5. Además de lo anterior, se utilizó un proceso de *transfer learning.* De los análisis realizados se concluyó que hay muchas diferencias entre las arquitecturas CNN y *transformer*, teniendo los *transformer* un desempeño inferior a las CNN (todas las CNN obtuvieron una *AUROC* media mayor a 95%, mientras las *transformer* estuvieron siempre debajo del 75%). Los investigadores comentaron que, a pesar del éxito de los *transformer* con imágenes naturales, el conocimiento obtenido no es fácilmente transferible al análisis de AMD. Se aplicó TTA (*Test-Time Augmentation*, Aumento de Tiempo de Prueba) a ambos tipos de arquitecturas, lo que mejoró el desempeño en 6 de las 9 arquitecturas *transformer*. Respecto los modelos propuestos por los autores, se describió un modelo con la arquitectura *EfficentNet B3*, el cual entregó una exactitud del 92.64% después de aplicar TTA y la aproximación de redimensionamiento PR-224-512. Los autores concluyeron que, en el contexto de clasificación de imágenes con AMD, las arquitecturas CNN dan mejores resultados y que son necesarios nuevos métodos para la correcta aplicación de *transfer learning* para las arquitecturas *transformer*.

Lo propuesto en [45] fue un modelo para la clasificación de imágenes de fondo de ojo, tal método distingue entre 5 estados de DR: 1. No DR, 2. Leve, 3. Moderada, 4. Severa y 5. Proliferativa. La arquitectura del modelo constaba de 2 módulos principales: FEB (*Feature Block Extraction*, Extracción de Bloques de Características) y GPB (*Grading Prediction Block*, Bloque de Predicción de Calificación). FEB, principalmente usado para la extracción de características en imágenes de fondo de ojo mediante el modelo ViT y GPB para la clasificación de las categorías de DR, mediante la captura de las diferentes regiones espaciales ocupadas por diferentes clases de objetos usando atención residual. Se tuvieron 2 conjuntos de datos. *DDR*, el cual contiene 13,673 imágenes de fondo de ojo, 6,835 para entrenamiento, 2,733 para validación y 4,105 para pruebas. También estuvo el conjunto *IDRiD*, con 516 imágenes de fondo de ojo, divididas en dos sets de 413 y 103 imágenes para entrenamiento y pruebas respectivamente. Las imágenes se cambiaron de resolución al azar a una escala de 512x512 pixeles y se llevaron a

cabo pasos de *data augmentation* como la rotación al azar de imágenes. El modelo se implementó en *PyTorch* 1.6 y la computadora en que se ejecutó contaba con un GPU *Nvidia Quadro RTX 6000* con 24GB de memoria VRAM. De los experimentos realizados, se obtuvo una precisión promedio de 91.54% en la clasificación de DR con la base de datos *DDR*, en la base de datos *IDRiD* la precisión promedio fue de 87.96%. Los autores concluyeron que su modelo alcanzó un desempeño competitivo.

2.2 Análisis Comparativo.

La siguiente tabla muestra una comparativa de los diferentes métodos y algoritmos empleados para la detección de degeneración macular asociada con la edad y retinopatía diabética (entre otras afecciones retinianas relacionadas), el tipo detección realizada, la tasa de precisión en la detección de la afección y el estado del proyecto.

Tabla 2.1 *Análisis comparativo de los artículos relacionados.*

Autor	Condición médica	Tipo de detección	Métodos	Resultados	Estado
JenHongTan [24]	AMD húmeda y seca	Binaria (sí/no)	CNN - blindfold CNN - cross-validation	0.9117 0.9545	Finalizado
Maximilian Treder [25]	AMD No especifica la calificación	Binaria (sí/no)	DCNN	0.9970	Finalizado
Zaiwang Gu [26]	Detección del disco óptico	N/A	U-Net (m-Net) DFS	1.0 (ORIGA) 0.9983 (mesidor)	Finalizado
Lei Wang [27]	Detección de exudaciones	N/A	CNN – ExudateNet1 CNN – ExudateNet2 CNN – ExudateNet3	0.9080 0.8741 0.9341	Finalizado
Cristina González-Gonzalo [28]	AMD húmeda o seca RD	No AMD/No RD Temprana Intermedia Avanzada	RetCad v.1.3.0	0.9510 (DR-AMD-DR) 0.9490 (DR-AMD-AMD) 0.9750 (DR) 0.9270 (AMD)	Finalizado
Joel C. De Goma [29]	AMD No especifica la calificación	Binaria (sí/no)	Neural Network - SLIC Random Forest - SLIC Naive Bayes - SLIC Support Vector Machine - SLIC	0.9561 0.9474 0.9123 0.9035	Finalizado
Venci Mihalov [31]	Retinopatías varias: CNV DME Drusas CSR MH	Binaria (sí/no)	CNN CNN – weight balancing DCNN VGG16 – transfer training	1.0000 0.9917 0.9876 0.9167 0.8333	Finalizado
Yao-Mei Chen [34]	AMD húmeda y seca DME Drusas	Binaria (sí/no)	Alexnet Googlenet VGG16	0.9550 0.9531 0.9594	Finalizado

Autor	Condición médica	Tipo de detección	Métodos	Resultados	Estado
	CNV		VGG19 Resnet18 Resnet50 Resnet101	0.9942 0.9847 0.9909 0.9919	
Ashok L R [35]	CNV DME Drusas	Binaria (sí/no)	Keras API	0.9459 (OCTID) 0.9980 (UCSD) 0.9990 (Kaggle)	Finalizado
Murat Seçkin Ayhan [38]	nAMD IRF SRF	Binaria (sí/no)	DNN	0.9420 0.9370 0.9170	Finalizado
Rivu Chakraborty [39]	AMD húmeda y seca	Temprana Intermedia Avanzada no neovascular Avanzada neovascular	DCNN	0.9000 (iChallenge-AMD) 0.9303 (iChallenge-AMD aumentada) 0.9955 (ARIA)	Finalizado
Rui Sun [41]	DR	Normal Leve Moderada Severa no proliferativa Severa proliferativa	Transformer - ResNet50	0.9870 0.9630	Finalizado
Yuka Kihara [42]	neMNV	Binaria (sí/no)	ViT	0.9100	Finalizado
Zhencun Jiang [43]	AMD DME	Binaria (sí/no)	ViT	0.9969	Finalizado
César Domínguez [44]	AMD No especifica la calificación	Binaria (sí/no) – Transformers Multiclase - CNNs	ViT EfficientNet B3	0.6532 0.9264	Finalizado
Zongyun Gu [45]	DR	Saludable Leve Moderada Severa Proliferativa no calificable	ViT	0.8280 0.9635 0.7793 0.9881 0.9581 0.9752	Finalizado

El análisis comparativo muestra que el uso de las redes neuronales convolucionales y ViT es altamente efectivo para la clasificación de DMAE y otras afecciones oculares, por lo que, un proyecto como el presentado en este trabajo tienen una alta viabilidad para su propósito en la detección de características de la DMAE en sus diferentes etapas.

Capítulo 3. Aplicación de la Metodología

En este capítulo se presentan las etapas de desarrollo del módulo para el análisis de características para la detección temprana de la degeneración macular asociada con la edad, mediante el análisis de las características extraídas de imágenes de fondo de ojo usando técnicas de visión artificial y aprendizaje profundo. Las etapas de desarrollo se basan en la metodología XP (*eXtreame Programming*, Programación Extrema). XP es una metodología de enfoque ágil que se constituye de 4 etapas, las cuales son: 1.- Planeación, 2.- Diseño, 3.-Desarrollo y 4.-Pruebas.

3.1 Planificación

En la etapa de planeación se identifican las características principales y prioritarias para entender los requerimientos del sistema. Lo anterior mediante la creación de historias de usuario que se implementan en posteriores iteraciones.

3.1.1 Historias de usuario

La Tabla 3.1 muestra la historia de usuario para el registro de usuarios en el sistema.

Tabla 3.1 *Historia de usuario: registro de usuario.*

Numero: 1	
Nombre de Historia: Registro de usuario	
Prioridad: Media	**Riesgo de Desarrollo:** Bajo
Puntos Estimados: 1	**Iteración Asignada:** 1
Descripción: El usuario ingresa datos en los campos solicitados por un formulario. La información capturada se almacena en la base de datos.	
Observaciones: El registro se trata del usuario (médico) para el que se diseñó el sistema.	

La Tabla 3.2 muestra la historia de usuario para el registro del paciente.

Tabla 3.2 *Historia de usuario: registro de paciente.*

Numero: 2	
Nombre de Historia: Registro de paciente	
Prioridad: Media	**Riesgo de Desarrollo:** Bajo
Puntos Estimados: 1	**Iteración Asignada:** 1
Descripción: Un usuario ingresa los datos de un “Paciente” en un formulario. Los datos capturados se almacenan en la base de datos.	
Observaciones: Se considera “Paciente” al sujeto cuya imagen de fondo de ojo es clasificada. Un paciente no se registra a sí mismo.	

La Tabla 3.3 muestra la historia de usuario para la clasificaron de imagen de fondo de ojo.

Tabla 3.3 *Historia de usuario: clasificación de imagen de fondo de ojo.*

Numero: 3	
Nombre de Historia: Clasificación de imagen de fondo de ojo	
Prioridad: Alta	**Riesgo de Desarrollo:** Alta
Puntos Estimados: 1	**Iteración Asignada:** 1
Descripción: El usuario ingresa al sistema la imagen de fondo de ojo de un paciente. El sistema lleva a cabo una clasificación y muestra un reporte (el resultado) de la evaluación.	
Observaciones: El resultado devuelto muestra el porcentaje de coincidencia en los diferentes grados afectación (no dmae, leve, moderado y avanzado).	

La Tabla 3.4 muestra la historia de usuario consultar historial de clasificaciones.

Tabla 3.4 *Historia de usuario: consultar historial de clasificaciones.*

Numero: 4	
Nombre de Historia: Consultar historial de clasificaciones	
Prioridad: Media	**Riesgo de Desarrollo:** Bajo
Puntos Estimados: 1	**Iteración Asignada:** 1
Descripción: El usuario visualiza el historial de clasificaciones realizadas.	
Observaciones: El usuario tendrá acceso a opciones de filtrado de datos.	

La Tabla 3.5 muestra la historia de usuario recuperación de reporte de clasificación.

Tabla 3.5 *Historia de usuario: recuperación de reporte de clasificación.*

Numero: 5	
Nombre de Historia: Recuperación de reporte de clasificación	
Prioridad: Media	**Riesgo de Desarrollo:** Bajo
Puntos Estimados: 1	**Iteración Asignada:** 1
Descripción: El usuario recibe el reporte de clasificación de un paciente en formato PDF.	
Observaciones: El usuario selecciona un reporte del historial de clasificaciones de acuerdo a los filtros aplicados a la lista de reportes.	

La Tabla 3.6 muestra la historia de usuario actualizar datos de usuario.

Tabla 3.6 *Historia de usuario: actualizar datos de usuario.*

Numero: 6	
Nombre de Historia: Actualizar datos de usuario	
Prioridad: Baja	**Riesgo de Desarrollo:** Bajo
Puntos Estimados: 1	**Iteración Asignada:** 1
Descripción: El usuario tiene la opción de actualizar sus datos mediante un formulario.	
Observaciones: El usuario no puede actualizar datos de otros usuarios.	

La Tabla 3.7 muestra la historia de usuario consultar listado de pacientes.

Tabla 3.7 *Historia de usuario: consultar listado de pacientes.*

Numero: 7	
Nombre de Historia: Consultar listado de pacientes	
Prioridad: Baja	**Riesgo de Desarrollo:** Bajo
Puntos Estimados: 1	**Iteración Asignada:** 1
Descripción: Se presenta al usuario la lista de usuarios registrados.	
Observaciones: La lista se presenta de acuerdo a los filtros aplicados a la lista de pacientes. El usuario únicamente visualiza a sus pacientes.	

La Tabla 3.8 muestra la historia de usuario actualizar datos de paciente.

Tabla 3.8 *Historia de usuario: actualizar datos de paciente.*

Numero: 8	
Nombre de Historia: Actualizar datos de paciente	
Prioridad: Baja	**Riesgo de Desarrollo:** Bajo
Puntos Estimados: 1	**Iteración Asignada:** 1
Descripción: El usuario podrá actualizar los datos de uno de sus pacientes mediante un formulario.	
Observaciones: El usuario solo tiene la opción de modificar datos de sus pacientes.	

La detección (clasificación) del grado de avance de DMAE, se realiza a partir de una imagen de fondo de ojo que el usuario ingresa al sistema (la obtención de esta imagen depende de la técnica utilizada por el oftalmólogo). Una vez ingresada la imagen al sistema, el proceso de clasificación se lleva a cabo mediante un modelo entrenado. Dicho modelo se entrenó con un conjunto de datos conformado por imágenes obtenidas de diferentes repositorios.

Para el ingreso de imágenes, clasificación y visualización de resultados, se utiliza una aplicación web, a la cual el usuario se registra y valida. El usuario que desee realizar una clasificación, debe realizar primero el registro del paciente cuya imagen de fondo de ojo se analiza para obtener su clasificación.

3.1.2 Asignación de roles del proyecto

En la Tabla 3.8 se estipula la asignación de los roles para el desarrollo del proyecto.

Tabla 3.8 *Roles del proyecto.*

Roles	Encargado
Gestor	Augusto Javier Reyes Delgado
Programador	Augusto Javier Reyes Delgado
Encargado de Pruebas	Augusto Javier Reyes Delgado
Encargado de Seguimiento	Augusto Javier Reyes Delgado
Cliente	Augusto Javier Reyes Delgado
Entrenador	Augusto Javier Reyes Delgado

3.1.3 Plan de entrega del proyecto

En la Tabla 3.9 El plan de entrega se define basándose las historias de usuario.

Tabla 3.9 *Plan de entrega del proyecto.*

Historia	Iteración	Prioridad	Riesgo	Fecha Inic.	Fecha Fin.
Historia 1	1	Media	Bajo	01/08/23	31/05/24
Historia 2	1	Media	Bajo	01/08/23	31/05/24
Historia 3	1	Alta	Alto	01/08/23	31/05/24
Historia 4	1	Media	Bajo	01/08/23	31/05/24
Historia 5	1	Media	Bajo	01/08/23	31/05/24
Historia 6	1	Baja	Bajo	01/08/23	31/05/24

Historia 7	1	Baja	Bajo	01/08/23	31/05/24
Historia 8	1	Baja	Bajo	01/08/23	31/05/24

3.2 Diseño

En esta etapa se muestra el diseño de la aplicación, basado en los requerimientos obtenidos de las diversas historias de usuario obtenidas.

3.2.1 Arquitectura del sistema

Se diseñó una arquitectura en capas para el sistema de detección temprana de la degeneración macular asociada con la edad. En la Figura 3.1 se presenta el diseño de la arquitectura con las diferentes capas y módulos que la aplicación utiliza para solucionar el problema planteado en este proyecto.

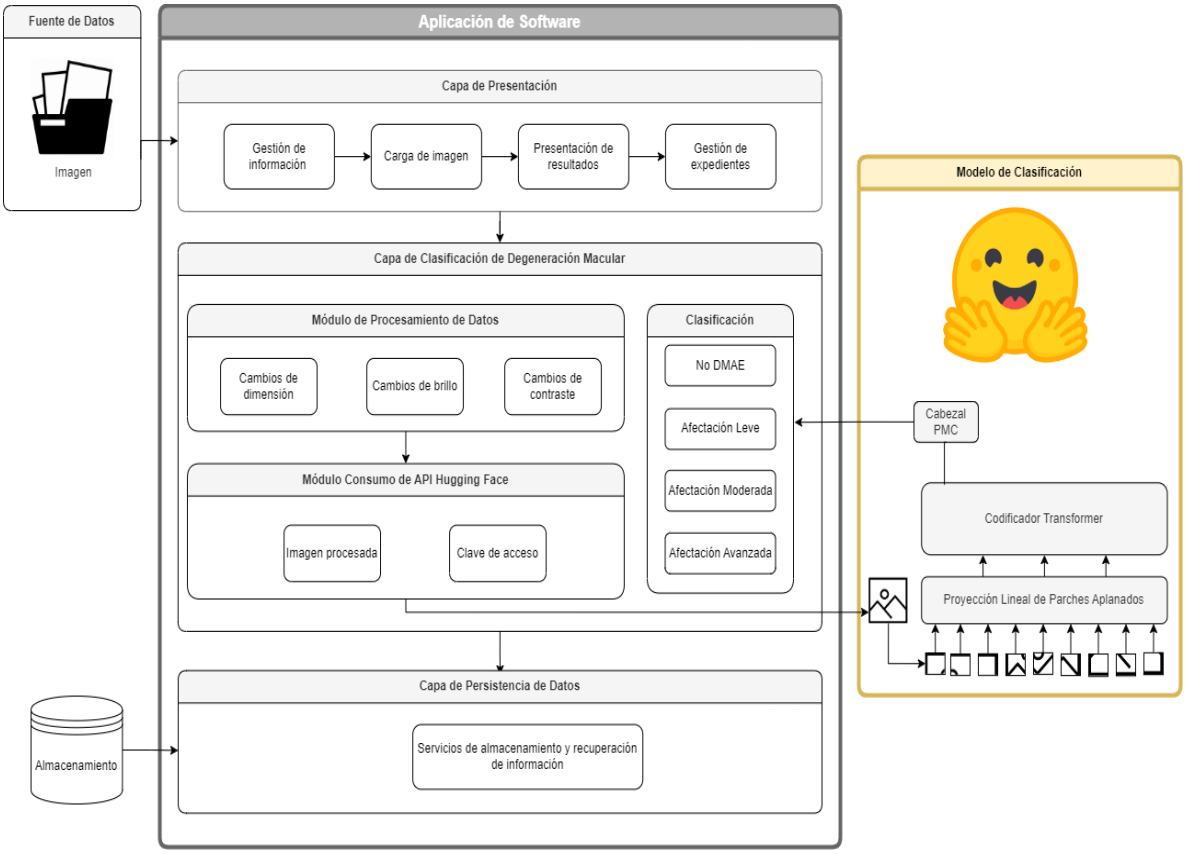


Figura 3.1 *Arquitectura del sistema.*

3.2.1.1 Capas de la arquitectura

La arquitectura mostrada en la Figura 3.1 se conforma de 3 capas, las cuales son descritas a continuación.

- **Capa de presentación:** Esta capa facilita la interacción del usuario con el sistema, en este caso, a través de una aplicación web. Aquí se realizan el registro y la autenticación de usuarios, así como el ingreso de datos de los sujetos, cuyas imágenes de fondo de ojo son clasificadas por el sistema. Adicionalmente, esta capa permite el control y la gestión de los datos del sistema.
- **Capa de clasificación de degeneración macular:** Esta capa se encarga de la clasificación de imágenes. Recibe imágenes de fondo de ojo, las procesa, analiza y devuelve los resultados. Estos resultados se almacenan en la capa de persistencia y se presentan en la capa de presentación.
- **Capa de persistencia de datos:** Esta capa es responsable de almacenar toda la información introducida en el sistema, incluyendo los datos de los usuarios, los datos de los sujetos a quienes pertenecen las imágenes de fondo de ojo, las propias imágenes y los resultados de clasificación obtenidos por la capa de clasificación de degeneración macular.

3.2.1.2 Módulos de la arquitectura

Cada capa de la arquitectura está conformada de diferentes módulos, estos se describen a continuación.

- **Gestión de información**: Módulo que se encarga de la gestión (registro, modificación y eliminación) de datos de los usuarios del sistema, así como de los datos de los pacientes.
- **Carga de imagen**: Módulo por el cual el usuario ingresa al sistema la imagen que es clasificada por el modelo entrenado.

- **Presentación de resultados**: Módulo para la presentación de los resultados de la clasificación de imágenes.
- **Gestión de expedientes**: Módulo para la gestión (recuperación y eliminación) de los reportes generados por el sistema tras realizar la clasificación de una imagen.
- **Módulo de procesamiento de datos**: Se encarga de aplicar un preprocesamiento a la imagen que se ingresó al sistema, previo a la clasificación. Transformaciones como cambio de resolución y cambios de brillo y contraste, esto para resaltar las características buscadas en la imagen para su clasificación.
- **Módulo de aprendizaje profundo:** Este módulo consiste en un modelo de transformador de visión entrenado con un conjunto de imágenes de fondo de ojo. Recibe imágenes preprocesadas, las divide en parches, y añade información de posición. A continuación, convierte los parches en una serie de tokens, incorpora información de posición adicional y procesa la secuencia de tokens para realizar la clasificación.
- **Clasificación**: Extensión del módulo de aprendizaje profundo, este módulo se encarga de la clasificación de la imagen ingresada.
- **Servicios de almacenamiento y recuperación de información**: Módulo encargado del almacenamiento y recuperación de los datos de usuarios y pacientes del sistema.

3.2.1.2 Flujo de trabajo

El flujo de trabajo del sistema se lista a continuación.

1. El usuario ingresa sus credenciales al sistema (se ingresa a través de la web), si estas son correctas, obtiene el acceso.
2. El usuario ingresa los datos correspondientes al sujeto paciente, incluida la imagen de fondo de ojo que pasa por el proceso de clasificación. Los la información se almacena en la base de datos.

3. La imagen ingresada pasa por un proceso que realiza transformaciones que destacan las características buscadas por el modelo entrenado.
4. El modelo entrenado realiza el proceso para la clasificación de la imagen dentro de alguno de los grados de avance de DMAE.
5. Se obtiene la clasificación de la imagen, y se genera un reporte con el resultado. Los datos se almacenan en la base de datos.
6. Obtenida la clasificación, se presenta el reporte con los resultados al usuario. El reporte presenta el porcentaje de posibilidad de cada uno de los cuatro niveles de afectación.

3.2.2 Arquitectura del transformador de visión

El módulo de aprendizaje profundo mencionado en la sección previa se trata de un modelo de transformador de visión. La arquitectura de tal transformador de visión (basada en la original ViT) se representa en la Figura 3.2.

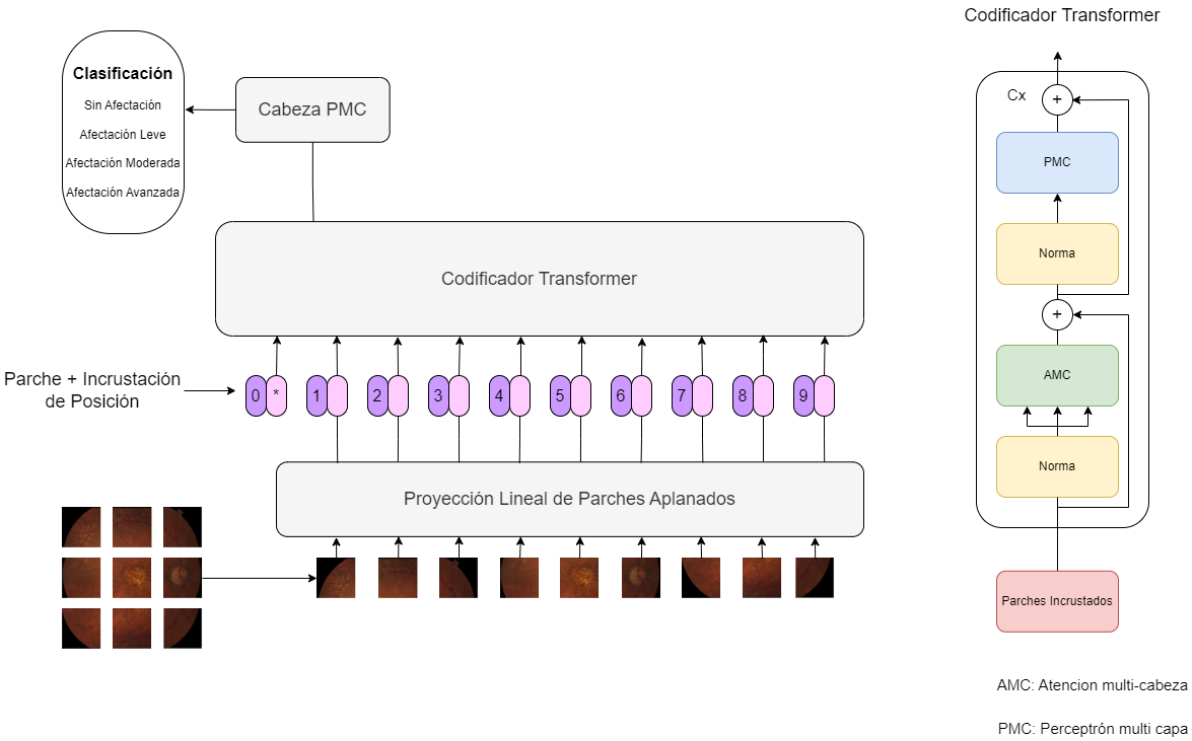


Figura 3.2 *Arquitectura del transformador de visión.* Imagen basada en[1]

[1] Dosovitskiy, A. (2021). An image is worth 16x16 words: transformers for image recognition at scale. ICLR.

La arquitectura común de ViT se compone de varias capas que se describen a continuación.

1. Incrustación de parches de Parches: En esta capa, la imagen de entrada se divide en parches solapados o no solapados, y cada parche se convierte en un vector de características mediante una capa de proyección. Esto da lugar a una secuencia de tokens.
2. Codificación de Posición: Para proporcionar al modelo información sobre la posición relativa de los tokens en la secuencia, se agrega información de posición utilizando codificación de posición sinusoidal o métodos similares.
3. Incrustación de clasificación: Los vectores de características de los parches se aplanan en una secuencia 1D y se pasan a través de una capa de incrustación de clasificación. Esto convierte los parches en una secuencia de tokens que se utilizan para realizar la clasificación.
4. Bloques de Transformadores: El corazón de la arquitectura ViT son los bloques de transformadores. Estos bloques incluyen capas de atención multi-cabezal y capas de feed-forward. Las capas de atención permiten al modelo aprender relaciones entre los tokens en la secuencia, y las capas de feed-forward realizan transformaciones no lineales en los vectores de características.
5. Capa de Normalización: Después de cada bloque de transformador, se aplica una capa de normalización para estabilizar las activaciones y acelerar el entrenamiento.
6. Capa de Clasificación: La salida de la última capa del transformador se pasa a una capa de clasificación que produce la predicción de la clase. Esto se logra mediante una capa de proyección que reduce la dimensionalidad de la salida a un número de clases y una función softmax que calcula las probabilidades de pertenencia a cada clase.

3.2.3 Análisis de requerimientos

El siguiente paso en la etapa de diseño es la identificación de requerimientos del sistema. Este paso se desglosa en los puntos que se presentan a continuación.

3.2.3.1 Requisitos funcionales

Los requisitos funcionales son aquellos que describen las funciones y capacidades específicas que requiere el software para satisfacer las necesidades del usuario y cumplir con los objetivos del sistema. Estos se detallan en las siguientes tablas.

Tabla 3.10 *Requisitos funcionales – Registrar usuario.*

ID	**RF1**
Nombre	Registrar usuario
Tipo	Requisito
Prioridad	Media
Descripción	El sistema muestra una serie de campos requeridos para la realización del registro.
Observación	El usuario ingresa los datos, el sistema los valida y almacena en la base de datos.
Salidas	Registro de datos

Tabla 3.11 *Requisitos funcionales – Registrar paciente.*

ID	**RF2**
Nombre	Registrar paciente
Tipo	Requisito
Prioridad	Media
Descripción	El sistema muestra una serie de campos requeridos para la realización del registro.

Observación	Un usuario registrado capturará los campos del paciente y el sistema los almacenará.
Salidas	Registro de datos

Tabla 3.12 *Requisitos funcionales – Clasificar imagen de fondo de ojo.*

ID	**RF3**
Nombre	Clasificar imagen de fondo de ojo.
Tipo	Requisito
Prioridad	Alta
Descripción	El sistema muestra un campo para la carga de una imagen de fondo de ojo que el usuario subirá al sistema y este último regresa su análisis.
Observación	El sistema almacena los resultados obtenidos.
Salidas	Resultados de Clasificación

Tabla 3.13 *Requisitos funcionales – Consulta de clasificaciones.*

ID	**RF4**
Nombre	Consulta de clasificaciones.
Tipo	Requisito
Prioridad	Media
Descripción	El sistema presenta la lista de los análisis realizados.
Observación	Existirán opciones de filtrado de datos.
Salidas	Ninguna

Tabla 3.14 *Requisitos funcionales – Recuperar reporte de clasificación.*

ID	**RF5**
Nombre	Recuperar reporte de clasificación.

Tipo	Requisito
Prioridad	Media
Descripción	El sistema muestra los resultados detallados de un reporte almacenado en el sistema.
Observación	Se tiene la opción de descargar el reporte en formato PDF.
Salidas	Ninguna

Tabla 3.15 *Requisitos funcionales – Actualizar datos de usuario.*

ID	**RF6**
Nombre	Actualizar datos de usuario.
Tipo	Requisito
Prioridad	Baja
Descripción	El usuario actualiza los datos de un paciente mediante un formulario.
Observación	Ninguna.
Salidas	El usuario modifica los campos que quiera actualizar.

Tabla 3.16 *Requisitos funcionales – Consultar datos de pacientes.*

ID	**RF7**
Nombre	Consultar datos de pacientes.
Tipo	Requisito
Prioridad	Media
Descripción	El sistema muestra la lista de los usuarios registrados.
Observación	Solo se muestran los pacientes del usuario doctor.
Salidas	Ninguna

Tabla 3.17 *Requisitos funcionales – Actualizar datos de paciente.*

ID	**RF8**
Nombre	Actualizar datos de paciente.
Tipo	Requisito
Prioridad	Baja
Descripción	El usuario actualiza sus datos mediante un formulario.
Observación	Ninguna
Salidas	El usuario modifica los campos que quiera actualizar.

Tabla 3.18 *Requisitos funcionales – Eliminar datos de paciente.*

ID	**RF9**
Nombre	Eliminar datos de paciente.
Tipo	Requisito
Prioridad	Baja
Descripción	El usuario elimina los datos de un paciente seleccionado.
Observación	Se eliminan los reportes relacionados al paciente.
Salidas	Se actualiza la base de datos.

3.2.3.2 Requisitos no funcionales

Estos son los aspectos que no están directamente relacionados con las funciones específicas que el software realiza, sino más bien, a cómo se desempeña en términos de calidad, rendimiento y otros atributos. Se detallan los requisitos no funcionales de este sistema en las tablas a continuación.

Tabla 3.19 *Requisitos no funcionales – Base de datos.*

ID	**RNF1**

Nombre	Base de datos
Tipo	Requisito
Prioridad	Alta
Descripción	Cada vez que se cree registro de usuario, paciente u análisis de imagen, los datos se almacenan en la base de datos.
Observación	Ninguna

Tabla 3.20 *Requisitos no funcionales – Diseño responsivo.*

ID	**RNF2**
Nombre	Diseño responsivo
Tipo	Requisito
Prioridad	Media
Descripción	La capa de presentación del sistema tiene un diseño capaz de adaptarse a cualquier dispositivo para su visualización.
Observación	Ninguna

Tabla 3.21 *Requisitos no funcionales – Disponibilidad.*

ID	**RNF3**
Nombre	Disponibilidad
Tipo	Requisito
Prioridad	Media
Descripción	El sistema está disponible en cualquier lugar y momento que el usuario lo requiera.
Observación	Ninguna

Tabla 3.22 *Requisitos no funcionales – Rendimiento.*

ID	**RNF4**
Nombre	Rendimiento
Tipo	Requisito
Prioridad	Media
Descripción	El sistema tiene una velocidad de carga y respuesta adecuado para su función como herramienta de soporte rápido. En el caso de clasificación de imágenes, no mayor a 10 segundos.
Observación	Ninguna

3.2.3.3 Casos de uso

De los requisitos funcionales, se genera un diagrama de caos de uso, el cual muestra los casos de uso del sujeto usuario. Se muestra gráficamente en la Figura 3.3.

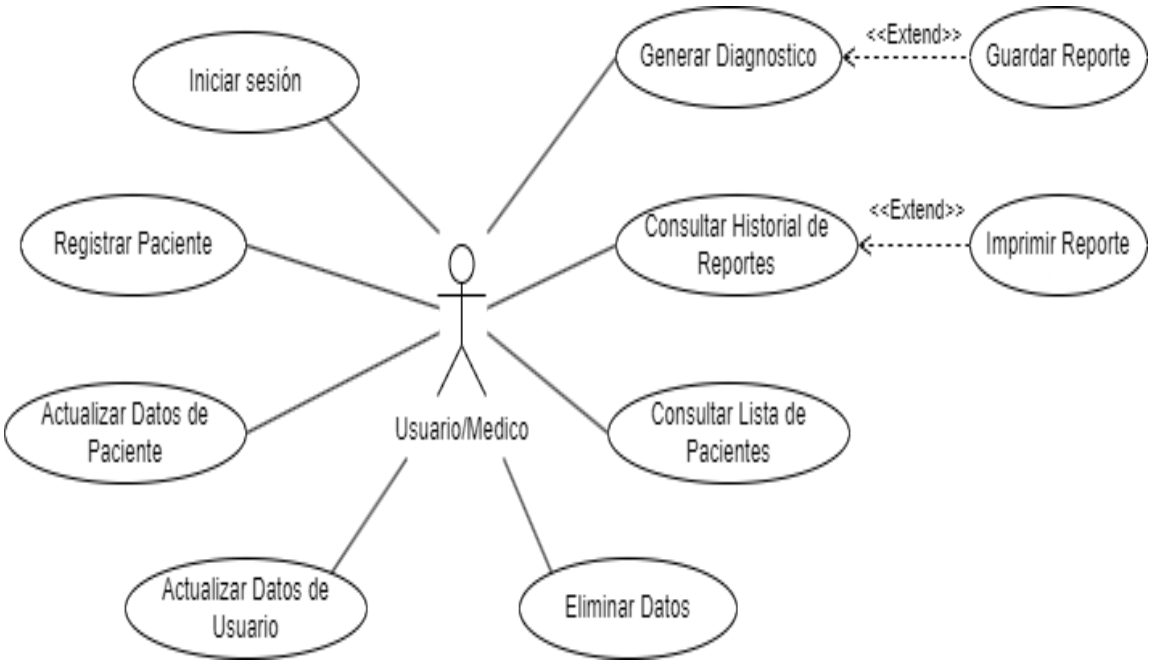


Figura 3.3 *Diagrama de casos de uso.*

3.2.3.4 Narrativas de casos de uso

Habiendo quedado definidos los casos de uso, se detallan a continuación las narrativas de cada respectivo caso de uso.

Tabla 3.23 *Narrativa de caso de uso – Registrar Paciente.*

Caso de Uso	Registrar Paciente
Actor	Usuario
Precondición	El usuario accedió a la aplicación web.
Postcondición	Se registra información en la base de datos.
Fujo Básico	
Actor	**Sistema**
1. El usuario accede a la página de registro de la aplicación web.	
	2. El sistema muestra un formulario de registro que solicita la siguiente información: nombre(s), primer apellido, segundo apellido, fecha de nacimiento, genero, estado y ciudad del paciente.
3. El usuario completa el formulario, ingresando la información requerida y da clic en “Registrar Paciente”.	
	4. El sistema valida que todos los campos sean completados. En caso de excepción, se va al flujo alternativo “Datos de entrada inválidos”. 5. El sistema genera un identificador único para el paciente (última clave más uno).

	6. El sistema almacena los datos ingresados. 7. Fin del caso de uso.
Flujo alternativo – Datos de entrada inválidos.	
Actor	Sistema
	1. El usuario no completó el formulario correctamente, el sistema muestra un mensaje de error y se le pide que corrija los campos incorrectos o incompletos.

Tabla 3.24 *Narrativa de caso de uso – Generar Diagnostico.*

Caso de Uso	Generar Diagnostico
Actor	Usuario
Precondición	El usuario accedió a la aplicación web. El sistema tiene pacientes registrados.
Postcondición	Se registra información en la base de datos.
Fujo Básico	
Actor	**Sistema**
1. El usuario selecciona a un paciente de la lista. 2. Selecciona si se va a analizar el ojo izquierdo o derecho. 3. Carga o arrastra la imagen a analizar en el sistema.	
	4. El sistema recibe la imagen.
5. El usuario da clic en “Generar Diagnostico”.	
	6. El sistema preprocesa la imagen.

	7. Realiza el análisis con el modelo entrenado. 8. Va a la pantalla de resultados y muestra el reporte de análisis con el porcentaje de precisión de cada uno de los posibles grados de avance. Ver flujo alternativo “Guardar Reporte”. 9. Fin de caso de uso.
Flujo alternativo – Guardar Reporte.	
Actor	Sistema
1. El usuario da clic en “Guardar” reporte.	
	2. El sistema guarda el reporte generado, relacionándolo con su respectivo paciente. 3. Regresa a la pantalla principal.

Tabla 3.25 *Narrativa de caso de uso – Consultar Historial de Reportes.*

Caso de Uso	Consultar Historial de Reportes
Actor	Usuario
Precondición	El usuario accedió a la aplicación web. El sistema tiene reportes registrados.
Postcondición	Ninguna.
Fujo Básico	
Actor	**Sistema**
1. El usuario da clic en “Ver Historial de Diagnósticos”. 2. Si lo desea, el usuario selecciona filtros para búsqueda.	

	3. El sistema muestra una lista de los reportes generados.
4. Si lo desea, el usuario selecciona un reporte y da clic en “Imprimir”. Se va al flujo alternativo “Imprimir Reporte”	
	5. Fin del caso de uso.
Flujo alternativo – Imprimir Reporte.	
Actor	Sistema
	1. El sistema regresa un documento en PDF con el reporte seleccionado. El cual muestra los resultados del análisis realizado, así como los datos del paciente al que pertenece el reporte.

Tabla 3.26 *Narrativa de caso de uso – Consultar Lista de Pacientes.*

Caso de Uso	Consultar Lista de Pacientes
Actor	Usuario
Precondición	El usuario accedió a la aplicación web. El sistema tiene pacientes registrados.
Postcondición	Ninguna
Fujo Básico	
Actor	**Sistema**
1. El usuario da clic en “Ver lista de pacientes”.	
	2. El sistema carga la lista de pacientes registrados.
3. Fin del caso de uso.	

Tabla 3.27 *Narrativa de caso de uso – Actualizar Datos de Paciente.*

Caso de Uso	Actualizar Datos de Paciente
Actor	Usuario
Precondición	El sistema tiene pacientes registrados. El usuario accedió a la lista de pacientes.
Postcondición	Se actualiza información en la base de datos.
Fujo Básico	
Actor	**Sistema**
1. El usuario selecciona a un paciente de la lista. 2. Da clic en “Actualizar”	
	3. El sistema carga una pantalla con un formulario con los datos del paciente.
4. El usuario modifica los datos deseados. 5. El usuario da clic en “Guardar”.	
	6. El sistema valida que todos los campos estén completos. En caso de excepción, se va al flujo alternativo “Datos de entrada inválidos”. 7. El sistema actualiza la base de datos. 8. Fin del caso de uso.
Flujo alternativo – Datos de entrada inválidos.	
Actor	Sistema
	1. El usuario no completó el formulario correctamente, el sistema muestra un mensaje de error y se le pide que corrija los campos incorrectos o incompletos.

Tabla 3.28 *Narrativa de caso de uso – Eliminar Datos.*

Caso de Uso	Eliminar Datos
Actor	Usuario
Precondición	El sistema tiene pacientes registrados. El usuario accedió a la lista de pacientes.
Postcondición	Se actualiza la base de datos.
Fujo Básico	
Actor	Sistema
1. El usuario seleccionar un paciente. 2. Da clic en “Eliminar Datos”	
	3. El sistema muestra un mensaje de confirmación al usuario.
4. El usuario da clic en “Aceptar”.	
	5. El sistema elimina los datos del paciente de la base de datos. 6. Fin del caso de uso.

Tabla 3.29 *Narrativa de caso de uso – Actualizar Datos de Usuario.*

Caso de Uso	Actualizar Datos de Usuario
Actor	Usuario
Precondición	El usuario accedió a la aplicación web.
Postcondición	Se actualiza información en la base de datos.
Fujo Básico	
Actor	**Sistema**
1. El usuario da clic en “Actualizar Datos”	

	2. El sistema carga una pantalla con el formulario de datos del usuario.
3. El usuario modifica los datos deseados. 4. El usuario da clic en “Guardar”.	
	5. El sistema valida que todos los campos estén completos. En caso de excepción, se va al flujo alternativo “Datos de entrada inválidos”. 6. El sistema actualiza la base de datos. 7. Fin del caso de uso.
Flujo alternativo – Datos de entrada inválidos.	
Actor	Sistema
	1. El usuario no completó el formulario correctamente, el sistema muestra un mensaje de error y se le pide que corrija los campos incorrectos o incompletos.

Las narrativas de casos de uso son valiosas en el proceso de desarrollo de software porque proporcionan una manera clara y comprensible de comunicar los requisitos del sistema a diferentes partes interesadas, incluidos los desarrolladores, diseñadores y usuarios finales.

3.2.3.5 Diagrama Entidad-Relación

Definidos los casos de uso y sus narrativas, se procede con la presentación del diagrama entidad-relación en la Figura 3.4. El entidad-relación (DER) es una herramienta visual utilizada en el diseño de bases de datos para representar las relaciones entre diferentes entidades.

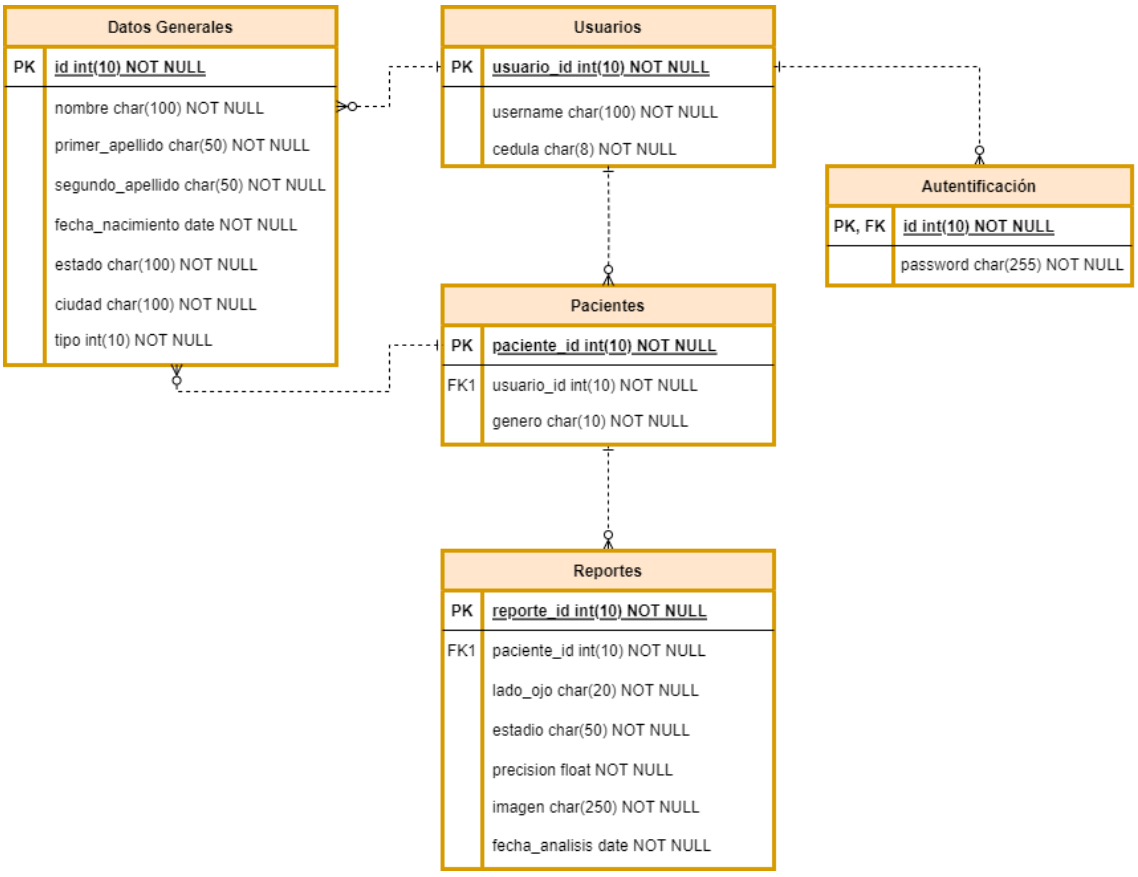


Figura 3.4 *Diagrama entidad-relación.*

3.2.3.6 Diccionarios de datos

A continuación, se describen las entidades y sus campos con el uso de sus respectivos diccionarios de datos.

Tabla 3.30 *Diccionario de datos – Usuario.*

Campo	Llave Primaria	Tipo	Tamaño	Nulo	Descripción
usuario_id	Si	Integer	10	Not Null	Identificador del usuario, se incrementa en uno con cada nuevo registro.
username	No	Varchar	100	Not Null	Nombre de usuario.
nombre	No	Varchar	100	Not Null	Nombre o nombres del usuario.
primer_apellido	No	Varchar	50	Not Null	Primer apellido del usuario.
segundo_apellido	No	Varchar	50	Not Null	Segundo apellido del usuario.
cedula	No	Varchar	8	Not Null	Cedula profesional del usuario.
password	No	Varchar	50	Not Null	Contraseña del usuario.

Tabla 3.31 *Diccionario de datos – Pacientes.*

Campo	Llave Primaria	Tipo	Tamaño	Nulo	Descripción
paciente_id	Si	Integer	10	Not Null	Identificador del paciente, se incrementa en uno con cada nuevo registro.
usuario_id	No	Integer	10	Not Null	Identificador del usuario que registró al paciente.

Campo	Llave Primaria	Tipo	Tamaño	Nulo	Descripción
nombre	No	Varchar	100	Not Null	Nombre o nombres del paciente.
primer_apellido	No	Varchar	50	Not Null	Primer apellido del paciente.
segundo_apellido	No	Varchar	50	Not Null	Segundo apellido del paciente.
fecha_nacimiento	No	Date		Not Null	Fecha de nacimiento del paciente.
genero	No	Varchar	10	Not Null	Genero del paciente.
estado	No	Varchar	100	Not Null	Entidad federativa del paciente.
ciudad	No	Varchar	100	Not Null	Nombre de la ciudad del paciente.

Tabla 3.32 *Diccionario de datos – Reportes.*

Campo	Llave Primaria	Tipo	Tamaño	Nulo	Descripción
reporte_id	Si	Integer	10	Not Null	Identificador del reporte, se incrementa en uno con cada nuevo registro.
paciente_id	No	Integer	10	Not Null	Identificador del paciente a quien pertenece el reporte.

Campo	Llave Primaria	Tipo	Tamaño	Nulo	Descripción
lado_ojo	No	Varchar	20	Not Null	Identifica ojo izquierdo o derecho.
estadio	No	Varchar	10	Not Null	Nivel de afección identificada.
precision	No	Varchar		Not Null	Nivel de precisión de la predicción.
imagen	No	Varchar	250	Not Null	Enlace a la imagen de fondo de ojo.
fecha_analisis	No	Date		Not Null	Fecha en que se realizó el análisis.

En conclusión, los diccionarios de datos explican las entidades y los campos que las componen. En resumen, un usuario (médico) tiene la capacidad de atender varios pacientes, a su vez, un paciente cuenta con la posibilidad de obtener varios reportes. El nivel de avance que es registrado en la base de datos del sistema es aquél que muestra el porcentaje más alto de probabilidad según el modelo entrenado.

3.2.3.7 Diagrama de clases

De lo desarrollado en los puntos anteriores, se generó el diagrama mostrado en la Figura 3.5, en la cual se muestra el diagrama de clases del sistema.

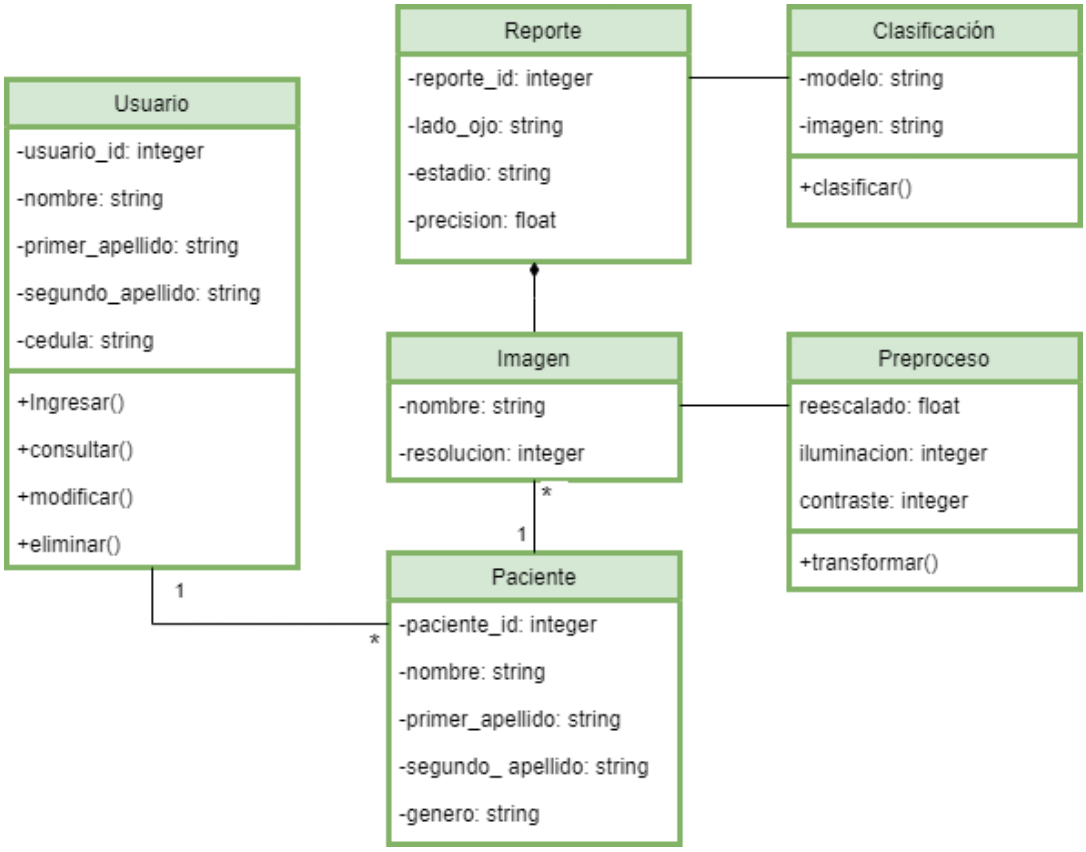


Figura 3.5 *Diagrama de clases.*

3.2.4 Diseño de la aplicación web

Dado el diseño interno del sistema, se procedió a desarrollar el diseño de la capa más externa, la cual facilita la interacción con el usuario. Es crucial asegurar que este diseño sea congruente con las historias de usuario previamente establecidas. Además, es necesario seleccionar un nombre apropiado para la aplicación, que permita a los usuarios identificar fácilmente el sistema.

Para lo escrito en el párrafo anterior, se decidió denominar al sistema con el nombre de "**DES**" (*DMAE Evaluation Support,* Soporte para evaluación DMAE) el cual hace referencia directa a la funcionalidad y propósito del sistema. A su vez se generaron diferentes *wirefremes* de las interfaces graficas que se requieren por la aplicación web del sistema.

El *wireframe* generado se ajusta a los casos de uso detectados. Para que el usuario realice el caso de uso “Registrar Paciente” es necesario que primero acceda a la aplicación web. En la Figura 3.6 se muestra la pantalla de login de la aplicación.

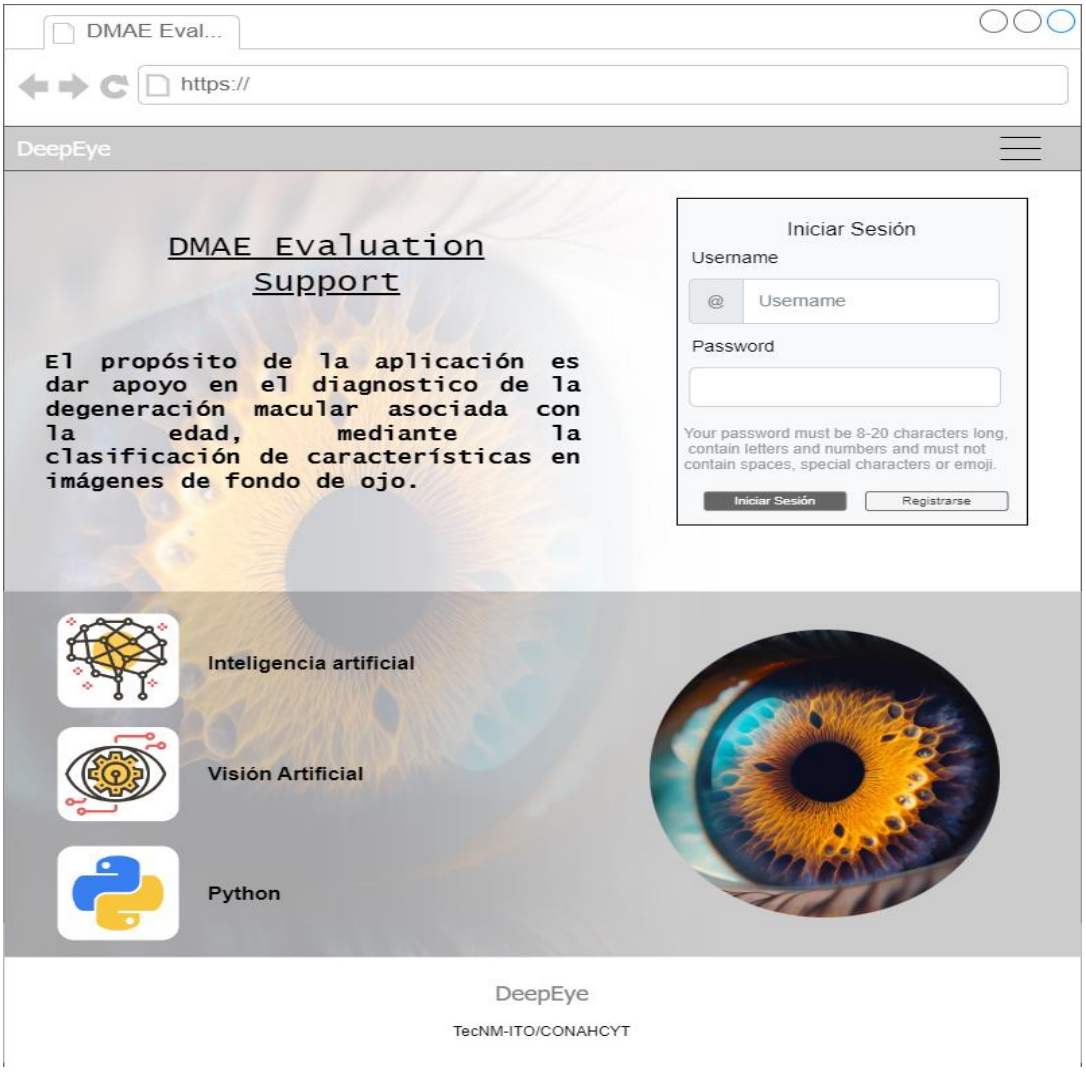


Figura 3.6 *Pantalla - Acceso (Login) de la aplicación.*

El usuario introduce sus credenciales (*Username* y *Password*) para iniciar sesión. Sin embargo, es necesario que previamente el usuario se encuentre registrado en el sistema. Para lo anterior, se requiere de una pantalla en la cual el sujeto (medico) que desee registrarse, ingrese sus datos. Esta pantalla esta graficada en la Figura 3.7.

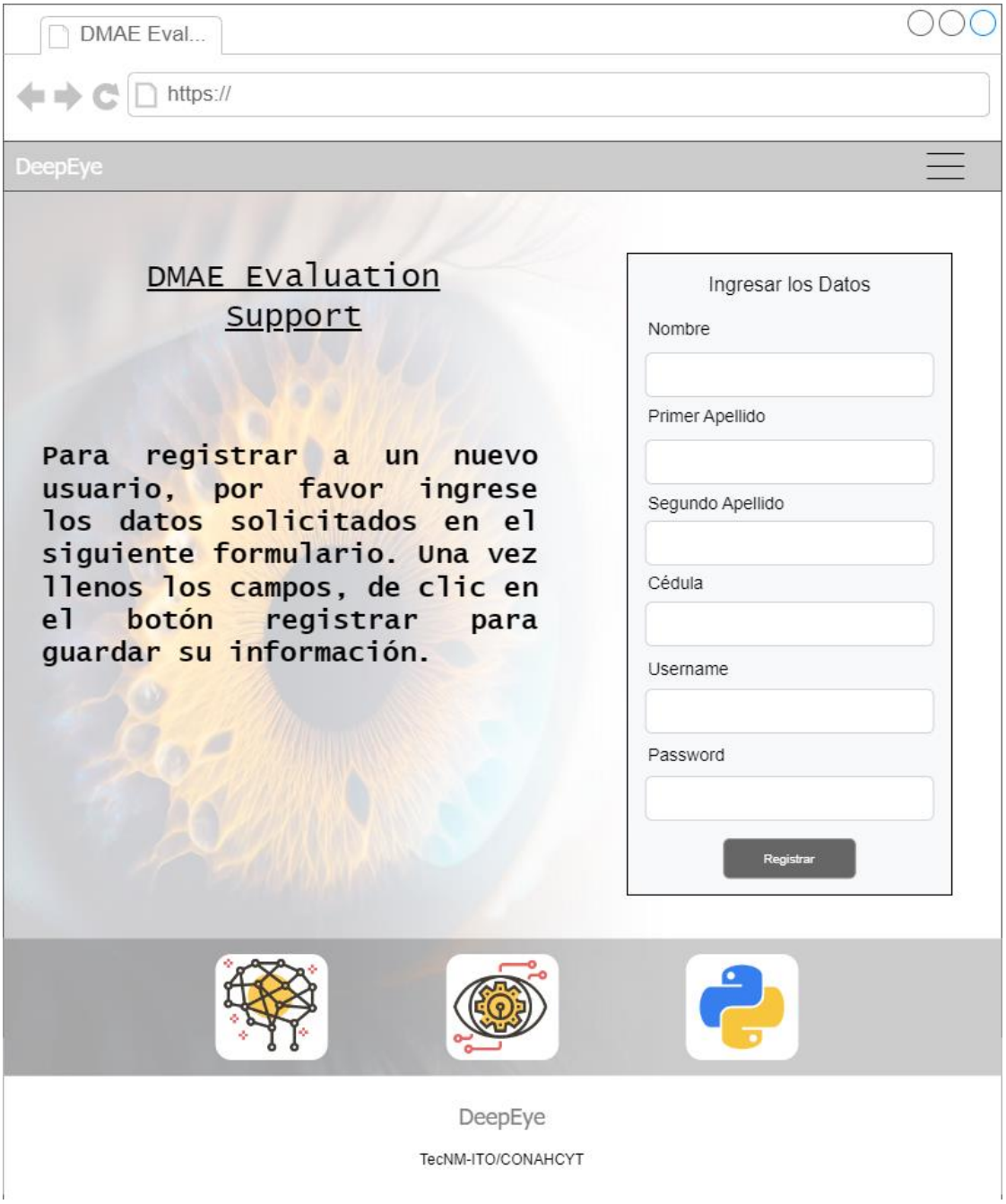


Figura 3.7 *Pantalla - Registro de usuario.*

Una vez el usuario esta registrado e ingresa al sistema, se encuentra con la pantalla principal o de inicio del sistema (Figura 3.8). En esta pantalla el usuario tendrá acceso directo a varias de las funcionalidades del sistema (Registrar Paciente, Generar Diagnostico), además tiene al alcance enlaces (botones) para acceder a las demás funcionalidades del sistema.

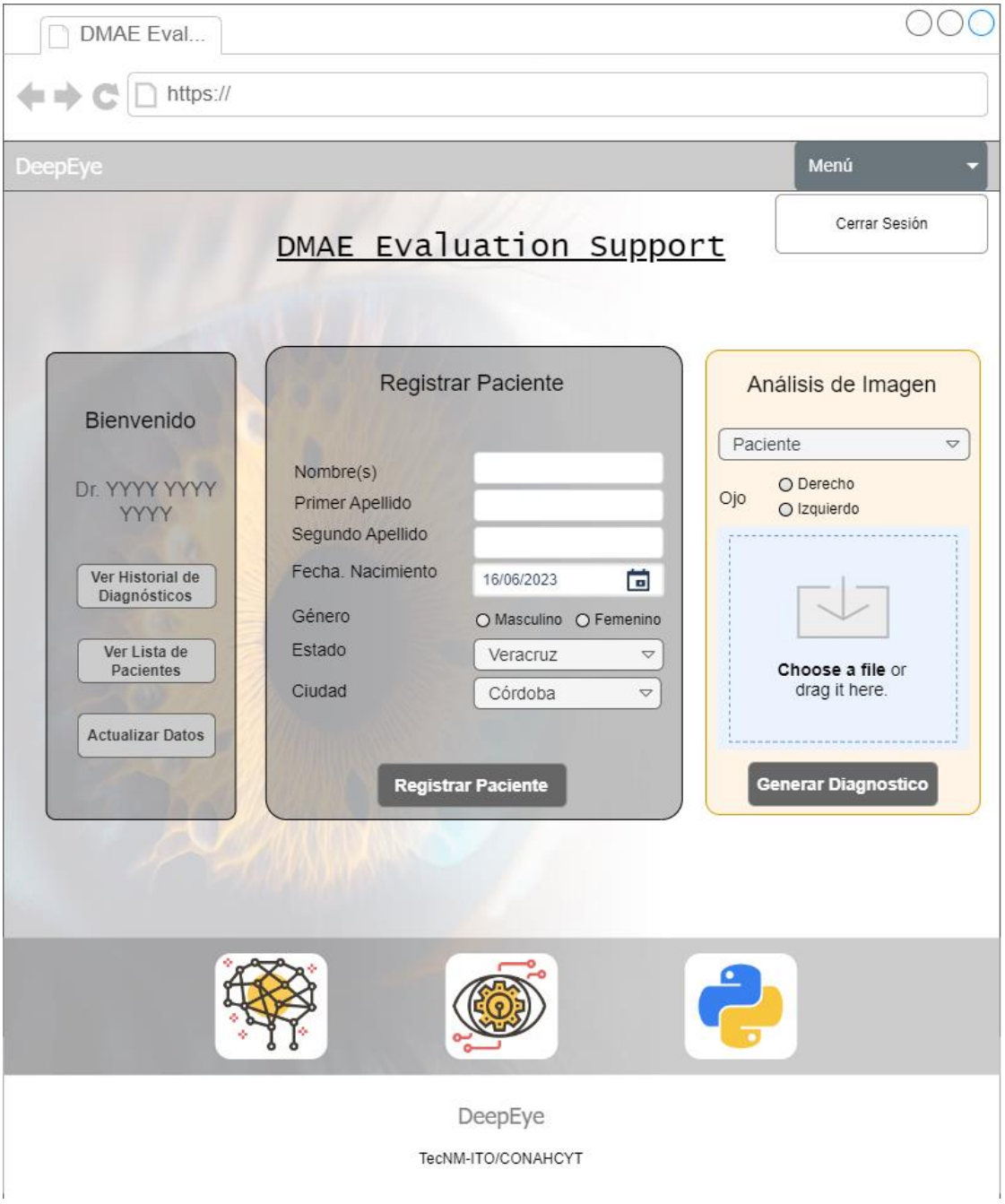


Figura 3.8 *Pantalla - inicio (Registrar Paciente, Generar Diagnostico).*

Si el usuario desea registrar a un paciente, este tiene la opción en la pantalla de inicio. Al realizar el registro de un paciente en el sistema, este último simplemente recargará la pantalla principal y el nombre del nuevo paciente aparece en la lista de pacientes desplegable de la sección de “Análisis de Imagen”. Para usar esta última

funcionalidad, el usuario selecciona un paciente de la lista, e inicia con el ingreso de una imagen (En este caso, la aplicación va dirigida a un usuario final, es decir un médico, por lo que, se espera que este último cargue imágenes de fondo de ojo a la aplicación), seleccionar si el ojo a analizar es el derecho o izquierdo y dar clic en “Generar Diagnostico”. Realizado el paso anterior, el sistema muestra la siguiente pantalla (Figura 3.9) con el reporte del análisis.

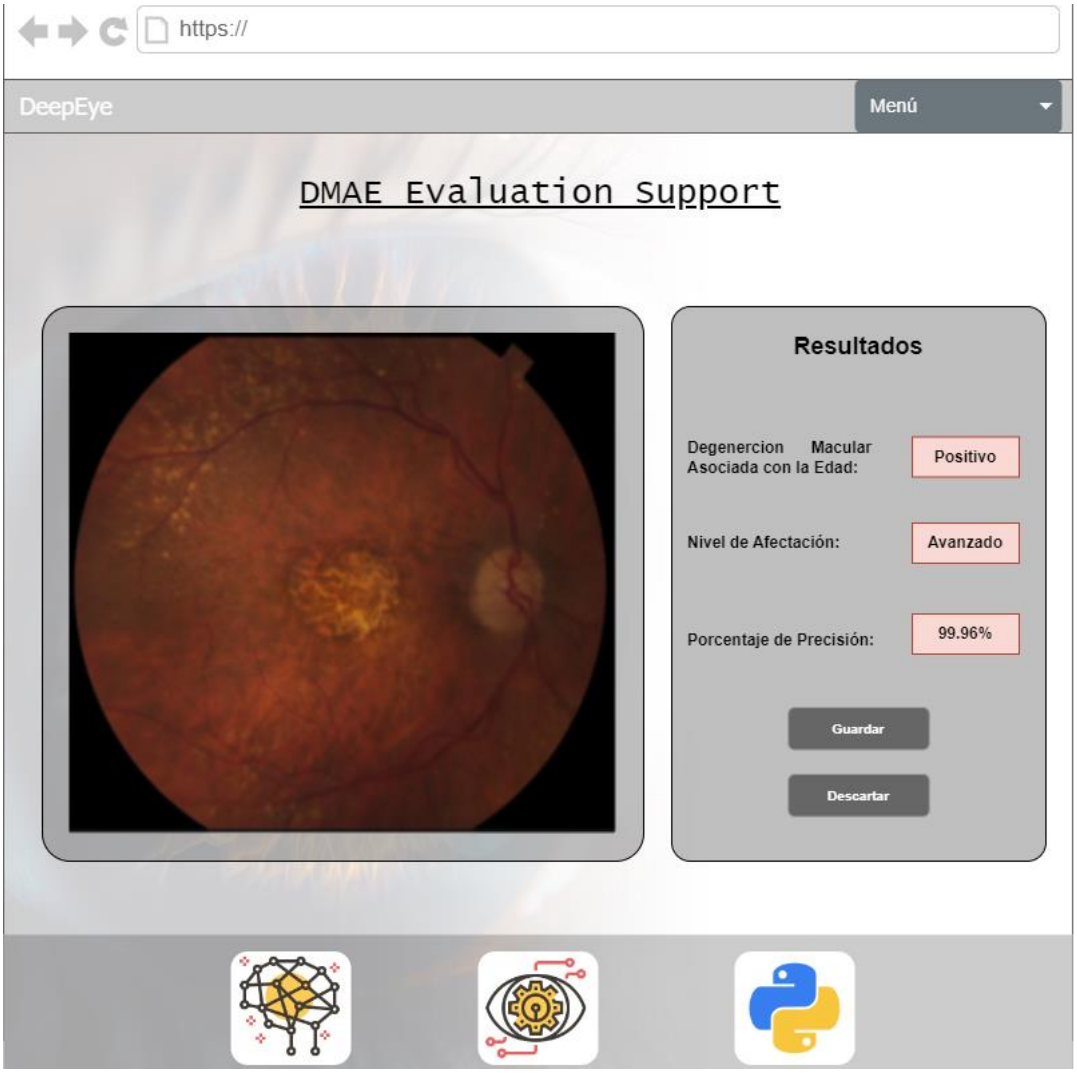


Figura 3.9 *Pantalla - Reporte del análisis.*

Obtenido el reporte del análisis de la imagen, el usuario tiene la opción de “Guardar” o “Descartar” el resultado, después el sistema devuelve al usuario a la página de inicio.

Los resultados guardados son presentados en la pantalla “Historial de Análisis”, a la cual se accede desde la página de inicio. El historial de diagnóstico carga toda la lista de reportes por defecto, como se muestra en la Figura 3.10.

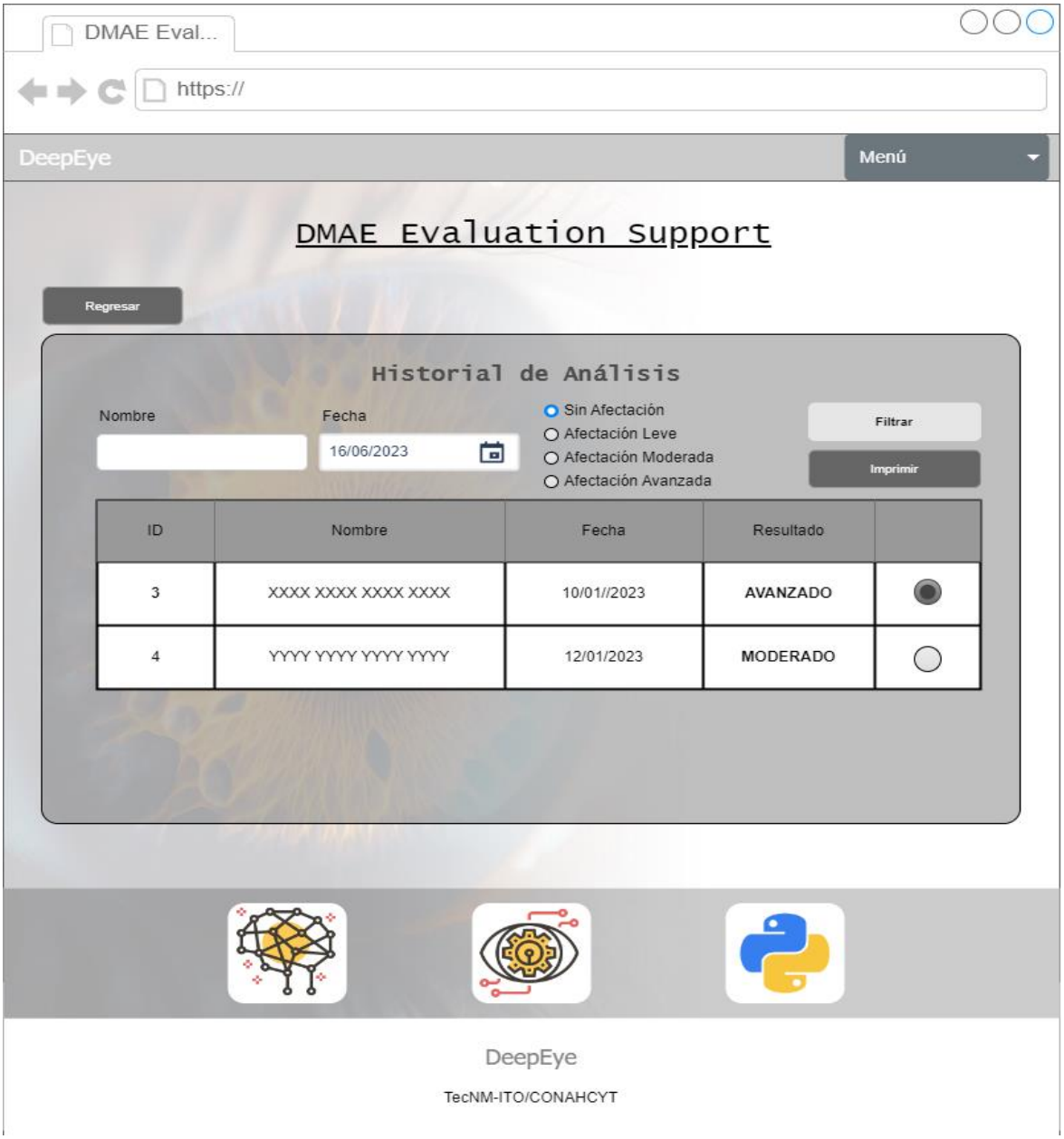


Figura 3.10 *Pantalla - Historial de análisis.*

En la pantalla presentada en la Figura 3.10, el usuario también tiene la opción de imprimir algún reporte seleccionado.

Respecto a la consulta de la lista de pacientes, se cuenta con la pantalla mostrada en la Figura 3.11, en esta aparece la lista todos los pacientes, con opciones de filtrado, además, en esta pantalla aparecen las opciones de “Editar Datos” y “Eliminar Datos”, representados por sus respectivos botones.

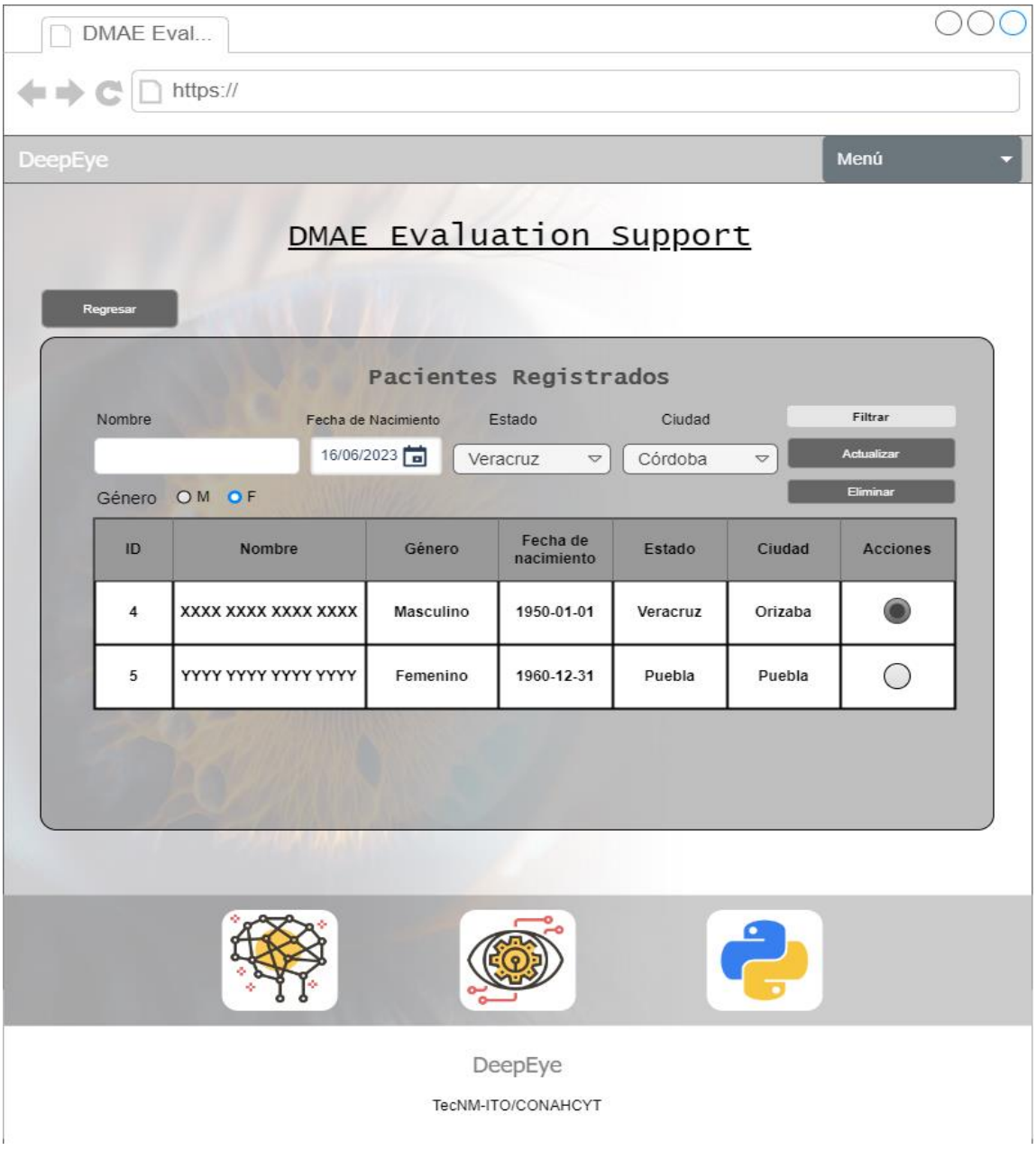


Figura 3.11 *Pantalla – Pacientes registrados.*

Al seleccionar la opción de “Eliminar Datos”, se borra la información de un paciente seleccionado y se recarga la página. Al dar clic en la opción “Editar Datos”, el

sistema redirige al usuario a una pantalla con un formulario con los datos del paciente seleccionado. En la pantalla mostrada en la Figura 3.12 se pueden realizar las modificaciones necesarias.

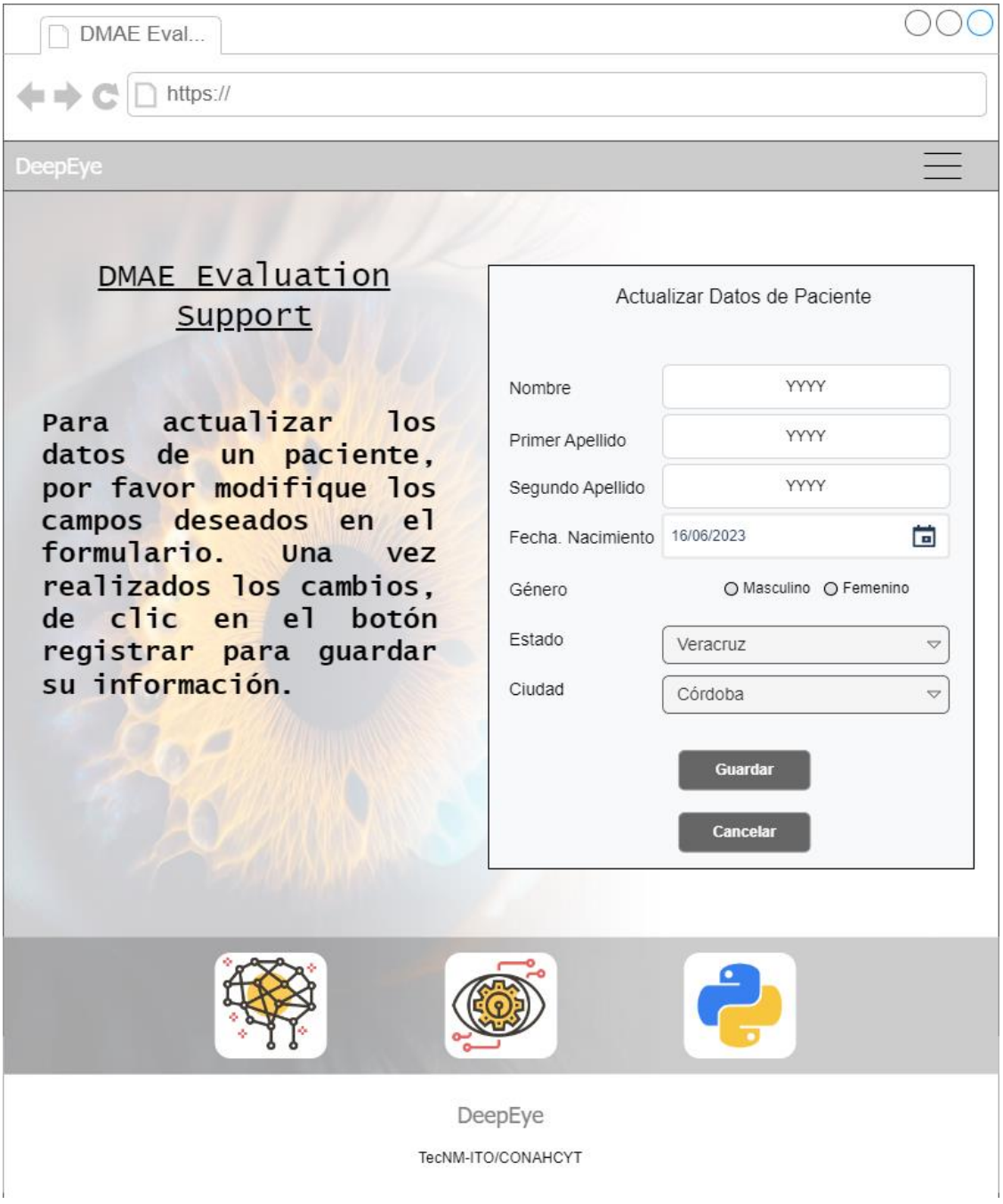


Figura 3.12 *Pantalla – Actualizar datos de paciente.*

La última de las pantallas concierne a la actualización de datos del usuario. Esta opción es accesible desde la página principal, a través del botón “Actualizar Datos”. En la pantalla de actualización de datos, el usuario puede modificar sus propios datos. La pantalla de esta funcionalidad se muestra en la Figura 3.13.

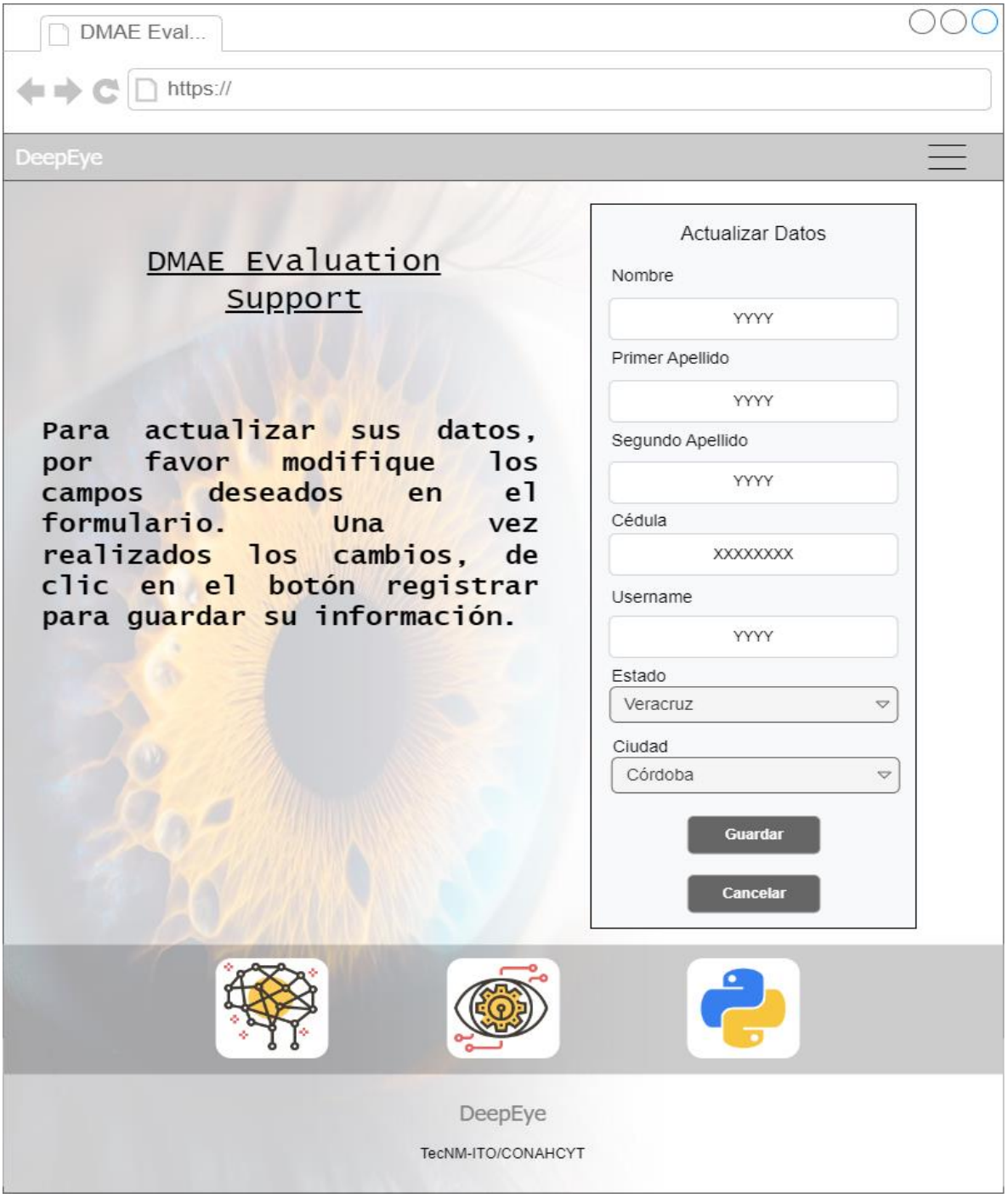


Figura 3.13 *Pantalla – Actualizar Datos.*

3.3 Codificación y Desarrollo

La tercera etapa de la metodología XP es la fase de desarrollo, en esta, se escribe el código para la implementación de las historias de usuario creadas, es decir, crear mecanismos funcionales para el sistema planteado.
En esta sección se documentan los procesos de desarrollo del sistema, es decir, el entrenamiento del modelo de visión artificial con transformadores de visión, y la aplicación que integra el modelo entrenado para dar acceso al mismo al usuario final. Para crear el modelo de Vision artificial, primero es necesario contar con datos de entrenamiento, paso que se detalla en el siguiente punto.

3.1.1 Conjunto de datos de entrenamiento

El modelo de visión artificial identifica las características presentes en las imágenes de fondo de ojo, para realizar la clasificación de manera correcta y coherente. Pare ello es esencial un conjunto de datos de entrenamiento amplio, variado y bien etiquetado.

3.3.1.1 Obtención de conjuntos de datos

Para obtener imágenes de fondo de ojo, las cuales presentan la condición de degeneración macular asociada con la edad, se buscó en diversos repositorios de datos. Se seleccionaron dos conjuntos de datos para la extracción de imágenes y creación de un nuevo conjunto de datos. El conjunto de datos de iChallenge-AMD [46] y un conjunto de datos disponible en Kaggle publicado por Mujib [47], el cual integra a su vez imágenes de distintos conjuntos de datos.
Estas imágenes están clasificadas de manera binaria, es decir, solo se distingue entre aquellas que presentan degeneración macular y aquellas que no. El propósito del proyecto de tesis estipula la clasificación de las imágenes de fondo de ojo en sus diferentes estados de avance, para lo cual es necesario etiquetar las imágenes según la severidad de la degeneración macular.

3.3.1.2 Etiquetado de imágenes

Para el correcto etiquetado de las imágenes del conjunto de datos, es necesaria la participación de expertos en el área. En este caso, se contó con la colaboración del presidente de la unidad periférica (Dr. Yonathan Omar Garfias Becerra) y del subdirector médico (Dr. José Luis Rodríguez Loaiza) del Instituto de Oftalmología Fundación Conde de Valenciana. El ultimo mencionado realizó la clasificación de las imágenes de fondo de ojo obtenidas de los conjuntos de datos seleccionados. El profesional de la oftalmología clasificó las imágenes en conjuntos según el grado de avance (leve, moderado, avanzado y sin dmae). Una muestra de las imágenes etiquetadas se muestra en la Figura 3.14.

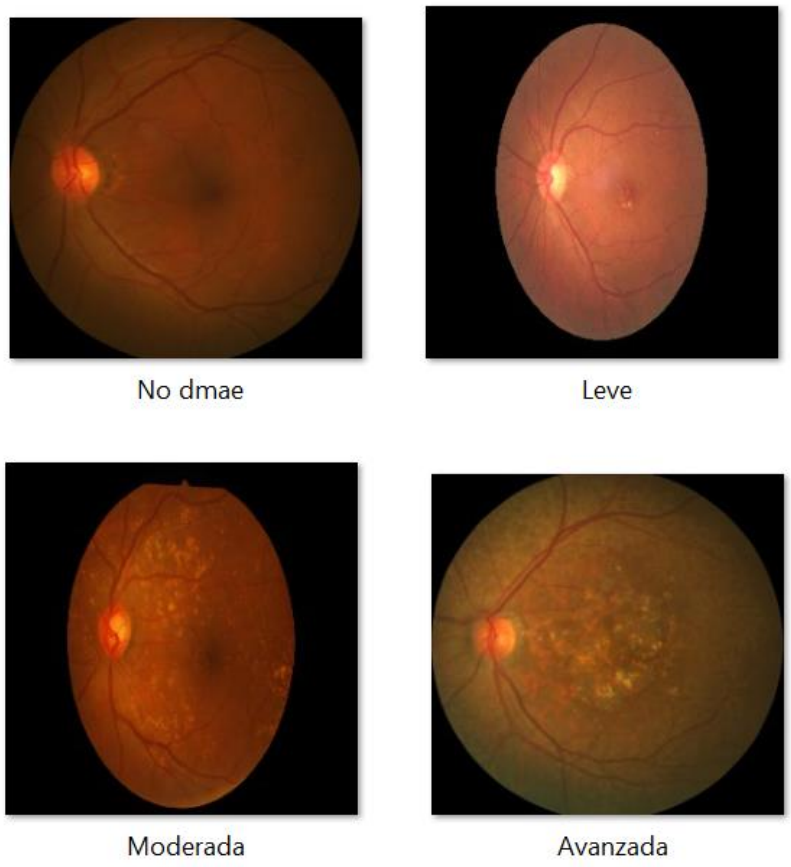


Figura 3.14 *Imágenes clasificadas en los diferentes grados de avance.*

Realizada la clasificación de las imágenes en los diferentes grados de avance presentados, se procedió con un proceso de aumento de datos.

3.3.1.3 Aumento de datos

El proceso de aumento de datos, el cual consiste en aplicar transformaciones a las imágenes para aumentar la muestra, se aplicó en este caso, porque el conjunto de imágenes para el entrenamiento se consideraba pequeño, para evitar los problemas asociados a un conjunto datos reducido. Antes de aplicar el aumento de datos, se apartaron imágenes para los segmentos de test y validación, los cuales no requieren la aplicación del proceso de aumento de datos.

Las transformaciones se aplicaron utilizando la interfaz de Keras (de Tensorflow), con la cual se aplicaron las transformaciones de: 1.-Rotación, 2-Translacion vertical, 3.-Traslación horizontal, 4.- Volteo vertical, 5.-Volteo horizontal, 6.-Cambios de brillo y 7.-Cambios de contraste.

Listado 3.1 *Transformaciones con Keras.*

```
1     from tensorflow.keras.preprocessing.image import
      ImageDataGenerator, array_to_img, img_to_array, load_img
2     import matplotlib.pyplot as plt

3     datagen = ImageDataGenerator(
      rotation_range=10,
      width_shift_range=0.03,
      height_shift_range=0.03,
      horizontal_flip=True,
      vertical_flip=True,
      fill_mode='nearest',
      brightness_range=[0.4,1.5],
      channel_shift_range=80
      )
```

El código mostrado en el bloque de Listado 3.1, muestra las transformaciones que se aplicaron al conjunto de datos. Dichas transformaciones se aplicaron al azar. Con estas transformaciones, se crearon nuevas imágenes para el conjunto final de entrenamiento. Se decidió que se crearon 5 imágenes por cada imagen original (con un tamaño de 300x300 pixeles), sin embargo, una parte de la muestra presentó aberraciones visuales (Exceso de luz o contraste o una combinación de ambos), por lo que se descartaron para su uso en el proceso de entrenamiento.

3.3.1.4 Creación del conjunto de datos

Una vez realizado el proceso de aumento de datos, se procedió a formar el conjunto de datos final con el cual entrenar el modelo de visión artificial. Para ello, el conjunto de imágenes al cual se aplicó aumento de datos, se tomó como segmento de entrenamiento, contando con un total de 1094 imágenes. Este conjunto de entrenamiento se juntó con los conjuntos de test y validación (60 imágenes cada uno) mencionados en el punto anterior para la creación del conjunto de datos final. Este conjunto de datos fue colocado en la plataforma de *Hugging Face* como se aprecia en la Figura 3.15, con el nombre de "*dmae-ve-U5*". Desde dicha plataforma, el conjunto es accedido, descargado y utilizado para el entrenamiento de modelos de visión artificial.

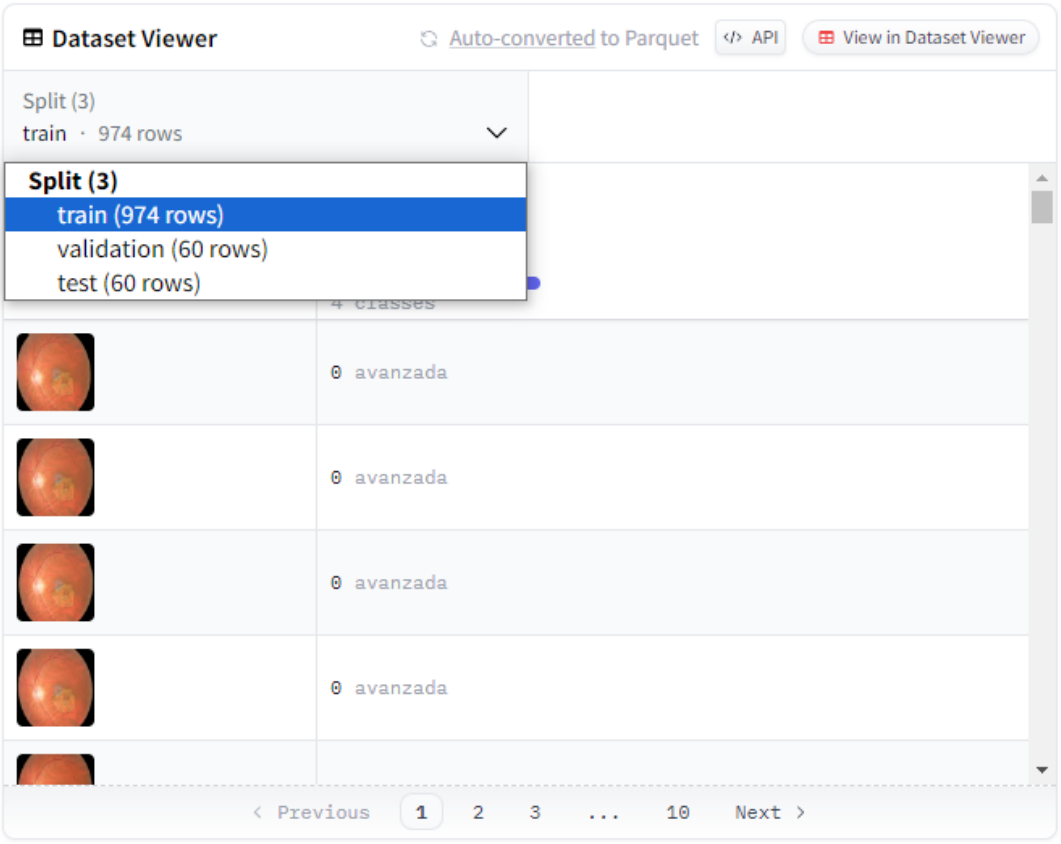


Figura 3.15 *Conjunto de datos en Hugging Face.*

3.3.2 Entrenamiento del modelo

Una vez preparado el conjunto de datos de entrenamiento, se procedió al entrenamiento del modelo de visión artificial. Para este propósito, se empleó una versión afinada de un modelo de Vision Transformer (ViT) proporcionado por Google. Además, se utilizó el framework de PyTorch junto con la librería Transformers. El proceso de entrenamiento se llevó a cabo utilizando notebooks en la plataforma Google Colab, y se aprovechó la plataforma de Hugging Face tanto para acceder al conjunto de datos creado como para alojar el modelo entrenado.

3.3.2.1 Implementación del modelo

Como punto de control para el entrenamiento del modelo se utilizó el modelo "google/vit-base-patch16-224" [48].

Listado 3.2 *Modelo que fue afinado.*

```
model_checkpoint = "google/vit-base-patch16-224" # pre-trained
model from which to fine-tune
```

Como se mencionó, se cargó el conjunto de datos creado al notebook.

Listado 3.3 *Carga del conjunto de datos.*

```
from datasets import load_dataset
...
# option 3: just load any existing dataset from the hub, like ...
dataset = load_dataset("Augusto777/dmae-ve-U5")
```

Para el proceso de entrenamiento, es necesario introducir una serie de parámetros denominados hiperparámetros o "argumentos de entrenamiento". Estos parámetros configuran el proceso y abarcan aspectos como el punto de control, el número de épocas de entrenamiento y la tasa de aprendizaje, entre otros. Un ejemplo de estos hiperparámetros se muestra en el Listado 3.4.

Listado 3.4 *Ajuste de hiperparámetros.*

```
model_name = model_checkpoint.split("/")[-1]

args = TrainingArguments(
      f"{model_name}-dmae-va-U5-40",
      remove_unused_columns=False,
      evaluation_strategy = "epoch",
      save_strategy = "epoch",
      learning_rate=5e-5,
      per_device_train_batch_size=batch_size,
      gradient_accumulation_steps=4,
      per_device_eval_batch_size=batch_size,
      num_train_epochs=42,
      warmup_ratio=0.1,
      logging_steps=12,
      load_best_model_at_end=True,
      metric_for_best_model="accuracy",
      push_to_hub=True,
)
```

Los hiperparámetros se describen brevemente a continuación:

- **Tasa de aprendizaje (learning_rate):** Es un hiperparámetro que controla cuánto se ajustan los pesos del modelo en cada actualización durante el entrenamiento. Está establecido en 5e-5.
- **Tamaño del lote de entrenamiento por dispositivo (per_device_train_batch_size):** Especifica el tamaño del lote para cada dispositivo de procesamiento. El valor se configura con la variable batch_size.
- **Pasos de acumulación de gradientes (gradient_accumulation_steps):** Controla cuántos pasos de retropropagación se acumulan antes de actualizar los pesos del modelo. Establecido en 4.
- **Número total de épocas de entrenamiento (num_train_epochs):** Indica cuántas veces el modelo procesa el conjunto completo de datos de entrenamiento. Fijado en 42.

- **Proporción de calentamiento (warmup_ratio):** Define qué fracción del entrenamiento total se dedica al período de calentamiento, en este caso, el 10%.
- **Frecuencia de registro de métricas (logging_steps):** Determina con qué frecuencia se registran las métricas durante el entrenamiento, establecido en 12 pasos.

Los valores asignados a los hiperparámetros se ajustaron mediante experimentación. Se probó con, por ejemplo, 12, 14 y 16 épocas de entrenamiento. Se determinó que un numero alrededor de las 40 épocas era adecuado para las circunstancias del modelo, pues un incrementó en este hiperparámetro no demostró un incremento en el desempeño hasta muy tarde en el desarrollo.

3.3.2.2 Resultados del modelo

Se llevaron a cabo numerosos entrenamientos, con diferentes hiperparámetros, cuyas precisiones fueron incrementando según los ajustes realizados hasta llegar a una meseta alrededor del 0.93 de precisión, sin embargo, la función pérdida continuaba variando. Se realizaron ajustes hasta llegar a un modelo que rebasara el 0.95 de precisión.

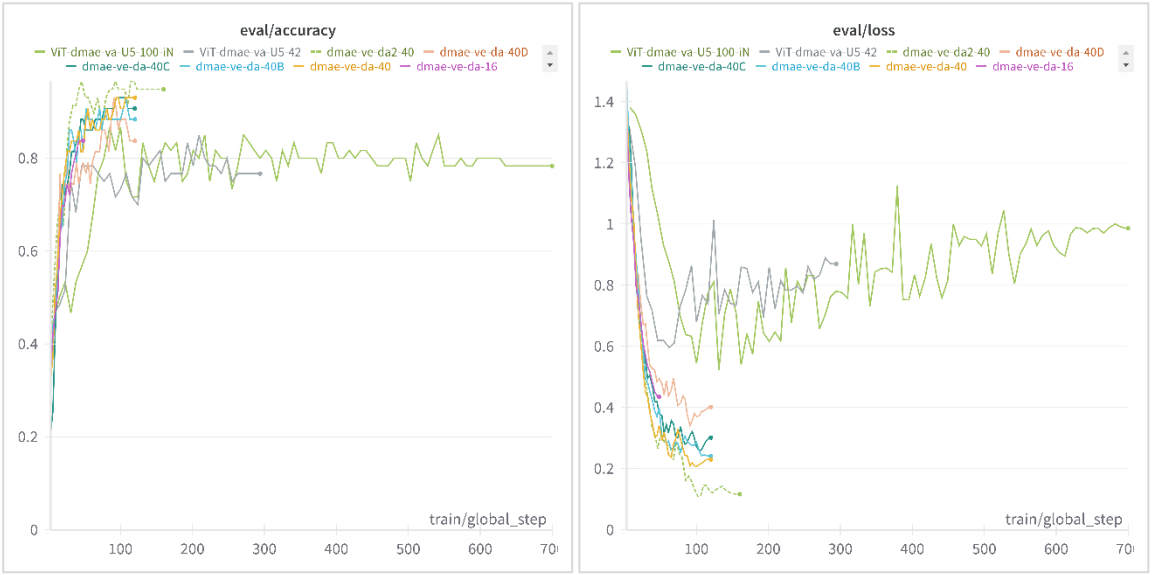


Figura 3.14 *Métricas de entrenamiento (precisión y perdida).*

El modelo entrenado “dmae-va-da2-40” alcanzó una precisión de 0.9655 con una pérdida del 0.2660, como se muestra en la Figura 3.16. Sin embargo, se llegó a la conclusión de que tal nivel de precisión obedecía a un sobre ajuste, pues tal precisión no se reflejaba al evaluar el modelo con el conjunto de pruebas. Por lo que, se seleccionó como modelo final al modelo “vit-base-patch16-224-dmae-va-U5-100-iN” que mostró el desempeño más consistente en los conjuntos de validación y pruebas.

Figura 3.16 *Tarjeta del modelo.*

3.3.2.3 Evaluación del modelo

Creado el modelo, se pasó a evaluar la precisión del mismo, para ello se utilizaron 60 imágenes de fondo de ojo y su evaluaron las meticas de: 1.-*Accuracy* (Exactitud), 2.-*Presicion* (Precisión), 3.- Sensibilidad y 4.- F1 Score.

Listado 3.5 *Código evaluación del modelo.*

```
from transformers import ViTFeatureExtractor,
ViTForImageClassification, Trainer, TrainingArguments
from datasets import load_dataset
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score

dataset = load_dataset("Augusto777/dmae-ve-U5",
split="validation")
model_name = " Augusto777/vit-base-patch16-224-dmae-va-U5-100-iN"
model = ViTForImageClassification.from_pretrained(model_name)
...
accuracy = accuracy_score(true_labels, predicted_labels)
precision = precision_score(true_labels, predicted_labels,
average=None)
recall = recall_score(true_labels, predicted_labels, average=None)
f1 = f1_score(true_labels, predicted_labels, average=None)
```

Los resultados obtenidos de la evaluación se muestran en la Tabla 3.33.

Tabla 3.33 *Métricas de evaluación por clase.*

Conjunto	Exactitud	Clase	Precisión	Sensibilidad	F1-Score
Validación	0.8666	**No dmae**	0.8333	0.8333	0.8333
		Leve	0.8333	0.9259	0.8771
		Moderada	0.8888	0.8000	0.8421
		Avanzada	1.0000	0.8571	0.9230
Pruebas	0.8166	**No dmae**	0.8000	0.6666	0.7272
		Leve	0.8333	0.9259	0.8771
		Moderada	0.8095	0.8500	0.8292
		Avanzada	0.7500	0.4285	0.5454

Como se observa en la tabla anterior, la precisión varia por clase, pero en general, es igual o superior al 80%.

3.3.3 Desarrollo y control de interfaces graficas

En esta etapa del desarrollo creó la interfaz gráfica de usuario. Para la creación de las mismas (*frontend*) se utilizaron HTML, CSS y EJS (JavaScript pues se utilizó node.js para la estructura del *backend*). Dado a que se trabajó con Node.js, es necesario establecer un motor de plantillas para la renderización de las vistas, como se muestra en el Listado 3.6:

Listado 3.6 *Fragmento de código utilizado como motor de plantillas.*

```
1     const express = require('express');
2     const multer = require('multer');
3     const fs = require('fs');
4     const moment = require('moment');

5     app.set('view engine', 'ejs');

6     app.get('/inicio', (req, res)=>{
7     if(req.session.loggedin){
8     res.render('inicio',{
      login: true,
      id: req.session.idD,
9     ...
```

Se desarrollaron las vistas necesarias para la correcta interacción con el sistema, las cuales se mapean del lado del servidor para su correcto renderizado. Las rutas utilizadas se muestran en el Listado 3.7.

Listado 3.7 *Fragmento de código de rutas de usuario.*

```
1     app.get('/', (req, res)=>{…
2     })
3     app.get('/auth', (req, res)=>{…
4     })
5     app.get('/register', (req, res)=>{…
6     })
7     app.get('/inicio', (req, res)=>{
8     if(req.session.loggedin){
9     res.render('inicio',{
      login: true,
      id: req.session.idD,
      name: req.session.name,
      listapacientes : req.session.pacientes,
      listareportes: req.session.reportes
      });
10    }else{
11    res.render('login',{
      alert: true,
      alertTitle: "Error",
      alertMessage: "Debe Iniciar Sesión",
      alertIcon: 'error',
      showConfirmButton: false,
      time: 2000,
      ruta: ''
      });}})
12    app.get('/records', (req, res)=>{…
13    })
14    app.get('/patients', (req, res)=>{…
15    })
16    app.get('/updateusers', (req, res)=>{…
17    })
18    app.get('/updatepatients', (req, res)=>{…
19    })
20    app.get('/results', (req, res)=>{…
21    })
22    app.get('/registerreport', (req, res)=>{…
23    })
```

Las diversas vistas del sistema comparten encabezado y pie de página, debido a esto, se generaron archivos que sean de uso común para todas las vistas del sistema. Tales archivos se invocan en las respectivas vistas.

Listado 3.8 *Fragmento del código del archivo cabecera.*

```
1     <link rel="stylesheet"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-
      awesome/6.0.0/css/all.min.css">
2     <link rel="stylesheet" type="text/css"
      href="/resources/css/style.css">
3     </head>
4     <body>
5     <header>
6     <nav style="margin-left: 15px;">
7     <ul>
8     <li style="font-size: larger;"><a href="/inicio">DeepEye</a></li>
9     </ul>
10    </nav>
11    <div class="dropdown" style="margin-right: 15px;">
12    <button class="button-T">Menú <i class="fa fa-caret-13  down">
14    </i></button>
15    <div class="dropdown-content">
16    <a href="/logout">Cerrar Sesi&oacute;n</a>
17    </div>
18    </div>
19    </header>
```

Listado 3.9 *Fragmento del código del archivo pie de página.*

```
1     <footer style="clear: both; width: 100%;">
2     <div class="footer-top" style="background-color: #999; padding:
      10px;">
3     <img src="/resources/img/iabrain.png" alt="Imagen 1">
4     <img src="/resources/img/iaeye.png" alt="Imagen 2">
5     <img src="/resources/img/pythonicon.png" alt="Imagen 3">
6     </div>
7     <div class="footer-bottom" style="background-color: #fff; padding:
      10px;">
8     <p style="color: #999; font-size: small;">DeepEye</p>
9     <p style="font-size: small;">TecNM-ITO/CONAHCYT</p>
10    </div>
11    </footer>
```

Establecidos el encabezado y el pie de página, se codificaron las vistas del sistema. parte del código para el index de la aplicación se muestra a continuación.

Listado 3.8 *Fragmento del código de la vista "index".*

```
1     <%- include('header') %>
2     <div style="text-align: center; margin-top: 6vh; font-family:
      lucida console; text-decoration: underline;">
3     <h1>DMAE Evaluation Support</h1>
4     </div>
5     <div class="contenedor">
7     <div class="section-50"><p style="font-size: 26pt;">El propósito de
      la aplicación es dar apoyo en el diagn&oacute;stico
      de la degeneración macular asociada con la edad, mediante la
      clasificación de características en imágenes de fondo de
      ojo.</p></div>
8     <div class="section-30" style="height: 45%;">
9     <h2>Iniciar Sesi&oacute;n</h2>
10    <form action="/auth" method="post">
11    <table>
12    <tr>
13    <td>Nombre de usuario</td>
14    <td><input type="text" name="user" id="user" placeholder="usuario"
      required></td>
15    </tr>
16    <tr>
17    <td>Contraseña</td>
18    <td><input type="password" name="pass" id="pass"
      placeholder="contraseña" required></td>
19    </tr>
20    </table>
21    <br>
22    <button type="submit" class="button-G">Iniciar
23    sesi&oacute;n</button>
24    </form>
25    <a href="register" style="text-decoration: none;">
26    <button style="margin-top: 10px;" class="button-GO">Registrar
27    usuario</button>
28    </a>
29    </div>
30    </div>
31    <%- include('footer') %>
```

La anterior es la estructura común que se utilizó para todas las vistas de la aplicación. La misma vista tiene las opciones de inicio de sesión y registro de usuario. Una vez el usuario registrado confirme sus credenciales, se le presenta su

interfaz de inicio, donde cuanta con las opciones mostradas gráficamente en la Figura 3.8. Parte del código de la vista de inicio se muestra a continuación.

Listado 3.9 *Fragmento del código de la vista "inicio".*

```
1     <div class="section-20">
2     <h1>Bienvenido</h1>
3     <br>
4     <h1>Dr. <%= name %></h1>
5     <br>
6     <a href="/patients" style="text-decoration: none;">
7     <button style="margin-top: 10px;" class="button-GW">Lista de
      Pacientes</button>
      ...
18    </div>

19    <div class="section-30">
20    <h1>Registrar Paciente</h1>
21    <br>
22    <form action="/registropaciente" method="post">
23    <table>
24    <tr>
25    <td>Nombre(s)</td>
26    <td><input type="text" name="nameP" id="nameP"
27    placeholder="Jos&eacute;"></td>
28    </tr>
      ...
105   </div>

106   <form action="/results" method="post" enctype="multipart/form-data"
107   style="margin-top: 10px;">
108   <select style="width: 95%; margin-bottom: 5%;border-radius: 5px;"
109   id="id_p" name="id_p">
110   <% for (var i = 0; i <= listapacientes.length-1; i++){ %>
111   <option value="<%= listapacientes[i].id %>"><%=
112   listapacientes[i].name%> <%= listapacientes[i].flastname%> <%=
113   listapacientes[i].slastname%></option>
114   <% } %>
115   </select>
```

```
116    <br>
117    <label>
118    <input type="radio" name="side" value="derecho" checked>
119    Ojo Derecho
120    </label>
121    <label>
122    <input type="radio" name="side" value="izquierdo">
123    Ojo Izquierdo
124    </label>
125    <br>
126    <br>
127    <div id="dropContainer" ondragover="handleDragOver(event)"
128    ondrop="handleDrop(event)">
129    <p>Selecciona o arrastra y suelta tu imagen aquí</p>
130    </div>
131    <input type="file" name="image" accept="image/*" id="imageInput"
132    style="display: none;">
133    <br><br>
134    <button type="submit" class="button-G">Generar
135    Diagn&oacute;stico</button>
136    </form>
137    </div>
138    </div>
```

En la página de inicio, el usuario final tiene acceso a las funcionalidades de registro de paciente, ver listados de pacientes y reportes, así como de modificar sus datos. Además de mencionado anteriormente, tiene la opción de realizar el análisis imágenes de fondo de ojo. Para esto, selecciona uno de los pacientes que registró y señalar si la imagen a analizar corresponde al ojo izquierdo o derecho, para finalmente seleccionar o arrastrar la imagen para su análisis. El siguiente listado muestra el código en JavaScript para la función de cargado de imagen.

Listado 3.10 *Código para la función de carga de imagen de fondo de ojo.*

```
1      <script>
2      function previewImage(file) {
3      const container = document.getElementById('dropContainer');
4      const dataInput = document.getElementById('imageDataInput');
5      const reader = new FileReader();
```

```
6     reader.onload = function (e) {
7     container.style.backgroundImage = `url('${e.target.result}')`;
8     dataInput.value = e.target.result;
9     };
10    reader.readAsDataURL(file);
11    }
12    function handleDragOver(event) {
13    event.preventDefault();
14    const dropContainer = document.getElementById('dropContainer');
15    dropContainer.classList.add('highlight');
16    }
17    function handleDrop(event) {
18    event.preventDefault();
19    const dropContainer = document.getElementById('dropContainer');
20    dropContainer.classList.remove('highlight');
21    const files = event.dataTransfer.files;
22    if (files.length > 0) {
23    const input = document.getElementById('imageInput');
24    input.files = files;
25    previewImage(files[0]);}}
```

La funcionalidad presentada en el Listado 3.10 es exclusiva de la vista de inicio, sin embargo, se desarrollaron otras funciones comunes entre diferentes vistas de la aplicación, tanto para validación, como para la impresión de reportes y dar alertas al usuario, esto se muestra más a detalle a continuación:

Listado 3.11 *Fragmento de código para la validación de selección.*

```
1     <script>
2     const radioButtons =
      document.querySelectorAll('input[type="radio"][id="idP"]');
3     radioButtons.forEach(radioButton => {
4     radioButton.addEventListener('change', function() {
5     const valorSeleccionado = this.value;
6     document.getElementById('idP1').value = valorSeleccionado;
7     document.getElementById('idP2').value = valorSeleccionado;
8     });
9     });
10    </script>
```

```
11    function validarSeleccionRadio(formularioId) {
12    var radios =
      document.querySelectorAll('input[type="radio"][id="idP"]');
13    var radioSeleccionado = false;
14    radios.forEach(function(radio) {
15    if (radio.checked) {
16    radioSeleccionado = true;
17    }
18    });
19    if (!radioSeleccionado) {
20    alert('Por favor, seleccione un paciente.');
21    event.preventDefault();
22    return false;
23    }
24    return true;
25    }
27    document.getElementById('formularioeditar').addEventListener
      ('submit',function(event) {
28    return validarSeleccionRadio('formulario-e');
29    });
30    document.getElementById('formularioeliminar').addEventListener
      ('submit', function(event) {
31    return validarSeleccionRadio('formulario-e');
32    });
```

Listado 3.12 *Fragmento de código para mostrar alertas.*

```
1     <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
2     <% if(typeof alert != "undefined") { %>
3     <script>
4     Swal.fire({
5     title: '<%= alertTitle %>',
6     text: '<%= alertMessage %>',
7     icon: '<%= alertIcon %>',
8     showConfirmButton: <%= showConfirmButton %>,
9     timer: <%= time %>
10    }).then(()=>{
11    window.location='/<%= ruta %>'
12    })
13    </script>
14    <% } %>
```

La aplicación requería de un estilo consistente para todas sus interfaces de usuario, para ese se escribió código CSS con el cual dotar de un estilo genérico al total de la aplicación.

Listado 3.13 *Fragmento del código CSS.*

```
1     body {
2     margin: 0;
3     padding: 0;
4     box-sizing: border-box;
5     font-family: 'Arial', sans-serif;
6     background-image: url('/resources/…’);
7     background-size: cover;
8     background-position: center;
9     background-repeat: no-repeat;
10    flex-direction: column;
11    height: 100vh;
12    display: flex;
13    }
14    .section {
15    height: 100%;
16    box-sizing: border-box;
17    border: 2px solid #333;
18    }
19    .section-20 {
20    width: 20%;
21    background-color: rgba(135, 135, 135, 0.546);
22    text-align: center;
23    padding: 20px;
24    border-radius: 12px;
25    border: 2px solid  rgba(24, 24, 24, 0.897);
26    margin-top: 2%;
27    margin-right: 5%;
28    margin-left: 6%;
29    }
30    .grid-container {
31    display: grid;
32    grid-template-columns: repeat(4, 1fr);
33    gap: 10px;
34    margin-right: 5%;
35    margin-left: 5%;
36    }
37    ...
```

```
.div-30 {
width: 30%;
background-color: rgba(252, 224, 140, 0.795);
text-align: center;
padding: 20px;
border-radius: 12px;
border: 2px solid  rgb(159, 119, 0);
margin-top: 2%;
 padding-top: 5%;
 font-size: large;
}
.button-G {
padding: 8px 15px;
width: 60%;
border: 2px solid #333;
border-radius: 5px;
background-color: #646464;
color: white;
font-weight: bold;
cursor: pointer;
}
.button-G:hover {
background-color: #a7a7a7;
}
```

3.3.4 Creación de la base de datos

En esta etapa se creó la base de datos del sistema. En esta se almacenan los datos correspondientes a los pacientes y usuarios (médicos) de la aplicación, estos últimos con sus respectivas credenciales de autorización. Así también se almacenan los datos correspondientes a los reportes generados para las imágenes analizadas.

Una muestra del código utilizado para la creación de la base de datos se muestra a continuación en el Listado 3.14.

Listado 3.14 *Fragmento del script para la creación de la base de datos.*

```
1     -- DROP DATABASE IF EXISTS deepeyedb;

2     CREATE DATABASE deepeyedb;

3     CREATE TABLE usuarios (
4     id integer NOT NULL DEFAULT generar_llave_primaria(),
5     user_name character varying(50) COLLATE pg_catalog."default",
6     cedula character varying(8) COLLATE pg_catalog."default",
7     CONSTRAINT usuarios_pkey PRIMARY KEY (id));

8     CREATE TABLE auth (
11    id integer NOT NULL,
12    password character varying(255) COLLATE pg_catalog."default",
13    CONSTRAINT auth_pkey PRIMARY KEY (id));

14    CREATE TABLE reportes (
15    id integer NOT NULL DEFAULT llave_reporte(),
16    id_p integer,
17    side character varying(20) COLLATE pg_catalog."default",
18    phase character varying(50) COLLATE pg_catalog."default",
19    acuracy double precision,
20    image character varying(250) COLLATE pg_catalog."default",
21    analys_date date,
22    CONSTRAINT reportes_pkey PRIMARY KEY (id))

23    ...

24    CREATE OR REPLACE FUNCTION public.generar_llave_primaria()
25    RETURNS integer
26    LANGUAGE 'plpgsql'
27    COST 100
28    VOLATILE PARALLEL UNSAFE
29    AS $BODY$
30    DECLARE
31    next_val INTEGER;
32    BEGIN
33    SELECT nextval('mi_secuencia') INTO next_val;
34    ...
```

3.3.5 Creación del módulo de preprocesamiento de datos

En el desarrollo de este módulo, se emplearon las bibliotecas OpenCV y Jimp. OpenCV, utilizada en su versión OpenCV.js, es una biblioteca orientada a la visión artificial. Por otro lado, Jimp se utilizó para el preprocesamiento de datos. Ambas bibliotecas facilitaron el proceso de transformación de las imágenes: OpenCV se encargó de ajustar las dimensiones, el brillo y el contraste, mientras que Jimp gestionó el manejo y la transformación del tipo de datos. Estas transformaciones se realizaron antes de introducir las imágenes en el módulo de clasificación.

Listado 3.15 *Muestra del código de preprocesamiento de datos.*

```
1     const Jimp = require('jimp');
2     const cv = require('./opencv.js');

3     async function procesarImagen(imagePath) {

4     const jimpSrc = await Jimp.read(imagePath);

5     const resizedImage = changeResolution(src, 300, 300);

6     const adjustedImage = adjustBrightnessContrast(resizedImage, 15,
      30);

7     const dstJimp = await jimpFromMat(adjustedImage);
8     src.delete();
9     resizedImage.delete();
10    adjustedImage.delete();

11    …}

12    function changeResolution(image, newWidth, newHeight) {
      …
13    cv.resize(image, dst, new cv.Size(newWidth, newHeight), 0, 0,
      cv.INTER_LINEAR);}

14    function adjustBrightnessContrast(image, brightnessPercent,
      contrastPercent) {
15    …
16    image.convertTo(dst, -1, contrast, brightness);
17    return dst;}
```

La función recibe una imagen del buffer y la retorna en el mismo formato para enviarla al módulo de clasificación de imagen.

3.3.6 Consumo del API de Hugging Face

Se creó un módulo para el consumo de un API que permite acceder al modelo entrenado, el cual se aloja en la plataforma de Hugging Face. El usuario envía la imagen a través de un formulario, dicha imagen pasa por el preproceso descrito en el punto 3.3.5, posterior al preprocesamiento, se envía módulo para el consumo del del API.

Listado 3.16 *Muestra del código para el manejo de análisis de imagen.*

```
1     app.post('/results', upload.single('image'), async (req, res) => {

2     try {
3      if (!req.file) {
      res.render('inicio', {login: true, n, result: { error: "No se ha
      seleccionado ningún archivo" } });
4      return;}

5     const data = await preproceso.procesarImagen(req.file.buffer);

6     const response = await clasifica.clasifica(data);

7     const result = await response.json();
```

Para lo acceder al servicio, la plataforma otorga una clave de autorización, la cual acompaña a la solicitud para consumir el servicio.

Listado 3.17 *Muestra del código para el consumo de API de Hugging Face.*

```
1     const fs = require('fs');

2     async function clasifica(data){
3     const response = await fetch(
      "https://api-inference.huggingface.co/models/Augusto777/...",
4     { headers: { Authorization: "Bearer hf_..." },
5     method: "POST",
6     body: data,
7      });
8     return response;}
```

La función “clasifica” recibe los datos de la imagen a clasificar y la envía en el body. Una vez analizada la imagen, retorna una respuesta que es en formato JSON.

3.4 Pruebas

Una vez completado el desarrollo de los diferentes módulos que componen la aplicación, se procedió a la integración total de los mismos, para comprobar que el sistema funciona correctamente como unidad. En esta sección se muestra el comportamiento de la aplicación.

3.4.1 Inicio de aplicación Web

Se creó una interfaz de usuario clara y minimalista, con la intensión de dotar al usuario de la información mínima necesaria para operar la aplicación, con el conjunto de botones y enlaces mínimos para evitar confusiones en el usuario. En la Figura 3.17 se muestra la vista de inicio de la aplicación.

Figura 3.17 *Vista de inicio de la aplicación.*

Para acceder a las funcionalidades del sistema, el medico tras estar registrado en el mismo y, además, contar con una cedula profesional valida ante el Registro Nacional de Profesionistas a través del de la Secretaría de Educación Pública (SEP), en caso contrario, se niega el registro (Figura 3.18).

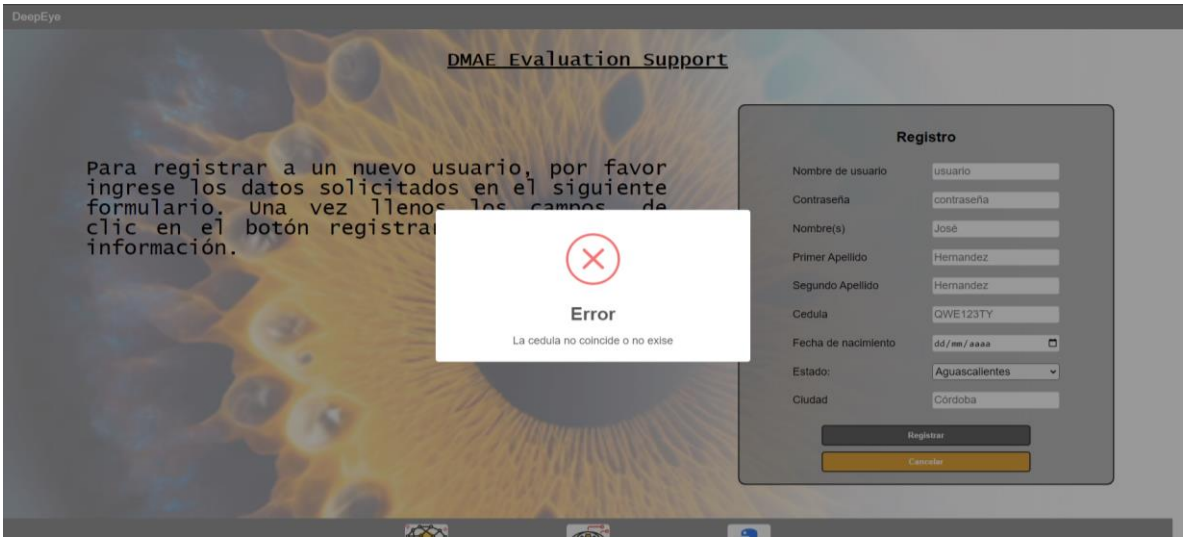


Figura 3.18 *Muestra de error de la aplicación.*

3.4.1 Vistas de usuario

Una vez el medico esta registrado correctamente (convertido en usuario) e inicia sesión, tiene a su alcance las diversas opciones del sistema (listado de pacientes, ver historial, actualizar datos, registrar paciente), la principal de ellas siendo el análisis de imagen, para ello se selecciona uno de entre una lista de pacientes (los pacientes se registran directamente en esta pantalla en el formulario de registrar pacientes, hecho el registro se recarga la vista con la lista actualizada), se señala el ojo a analizar y se carga o arrastra la imagen de fondo de ojo al respectivo campo para generar el diagnostico, como se muestra en la Figura 3.18.

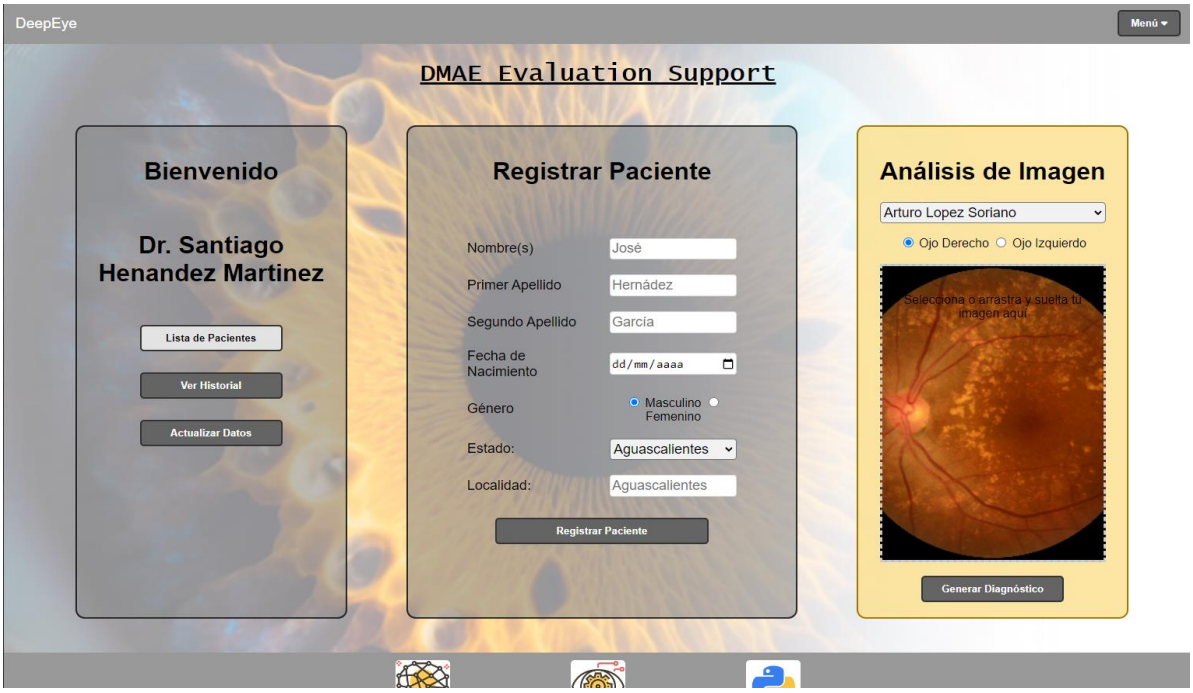


Figura 3.18 *Vista de inicio de usuario.*

Una vez cargada la imagen, el usuario da clic en “Generar Diagnostico” lo cual lleva al usuario a la vista de resultados, en la misma se muestra la clasificación obtenida por el modelo entrenado (Figura 3.19).

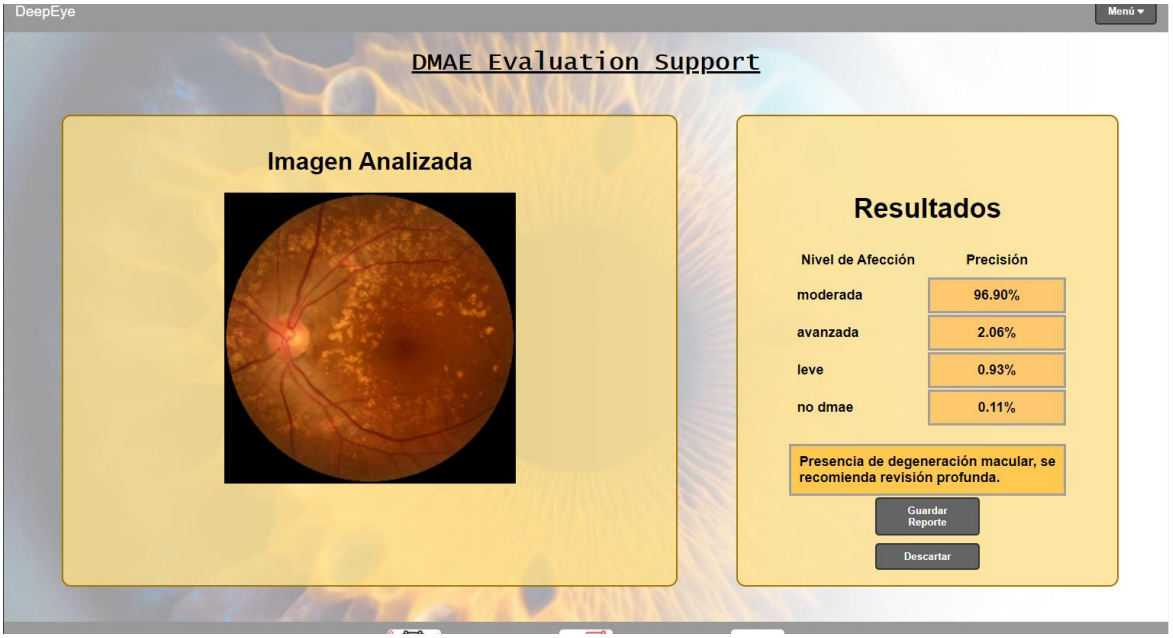


Figura 3.19 *Vista de resultados.*

La pantalla de resultados muestra la imagen analizada, los resultados de la clasificación y da las opciones para guardar o descartar el reporte.

Los reportes guardados tienen la opción para su consulta en la vista de historial de análisis, a la cual se accede desde la vista de inicio de usuario. En el historial de reportes el usuario accede a la lista de reportes guardados, en la que aplica las opciones de filtrado, impresión y eliminación (Figura 3.20).

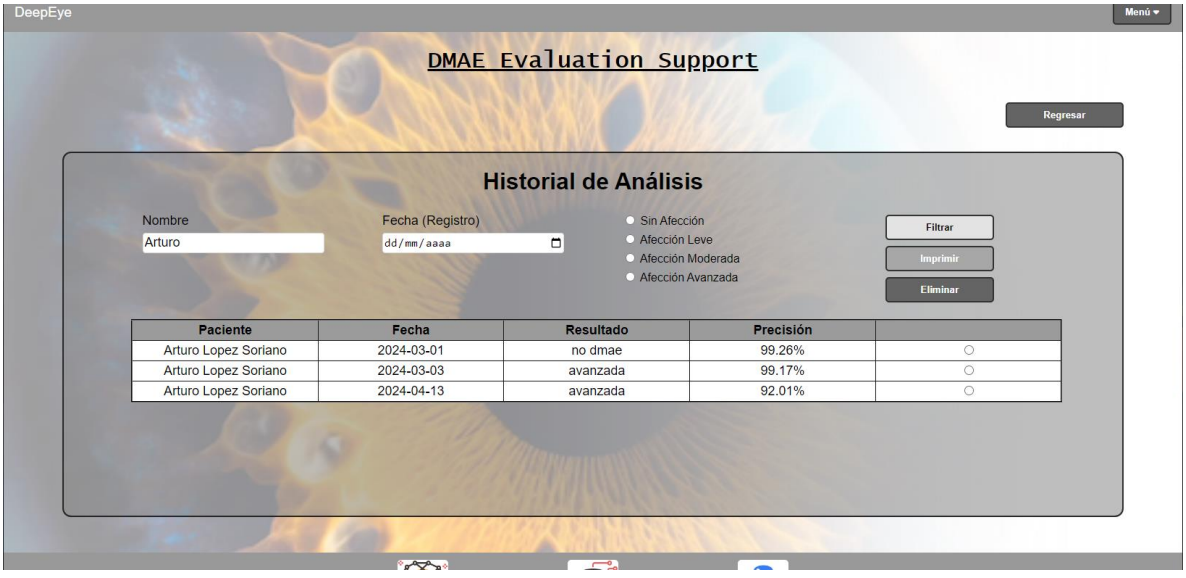


Figura 3.20 *Vista de resultados.*

Si el usuario decide imprimir un reporte seleccionado, es redirigido a una vista con los datos que aparecen en el documento .PDF que se genera para la impresión, teniendo la opción de proceder o no con este paso. En el documento que se genera, aparecen datos básicos del paciente, el nombre del médico, fecha y resultado del análisis, así como el día en que se generó el reporte. Además, se dota de espacio adicional en el que el medico tiene la opción de hacer anotaciones manuales sobre sus observaciones en la imagen, esto se muestra en la Figura 3.21.

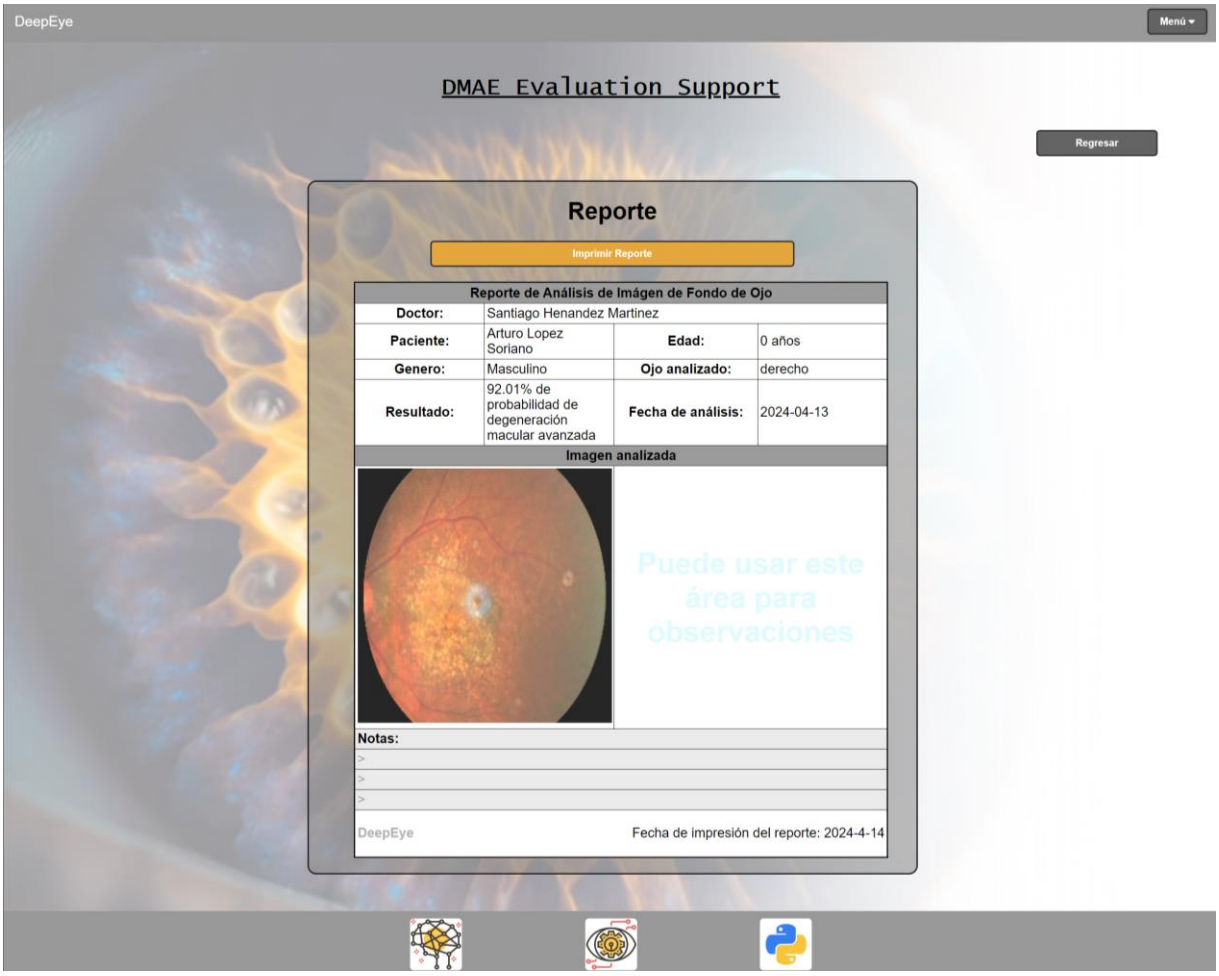


Figura 3.21 *Vista de reporte para impresión.*

La lista de pacientes con sus diversas opciones también esta disponibles como se muestra en la siguiente figura.

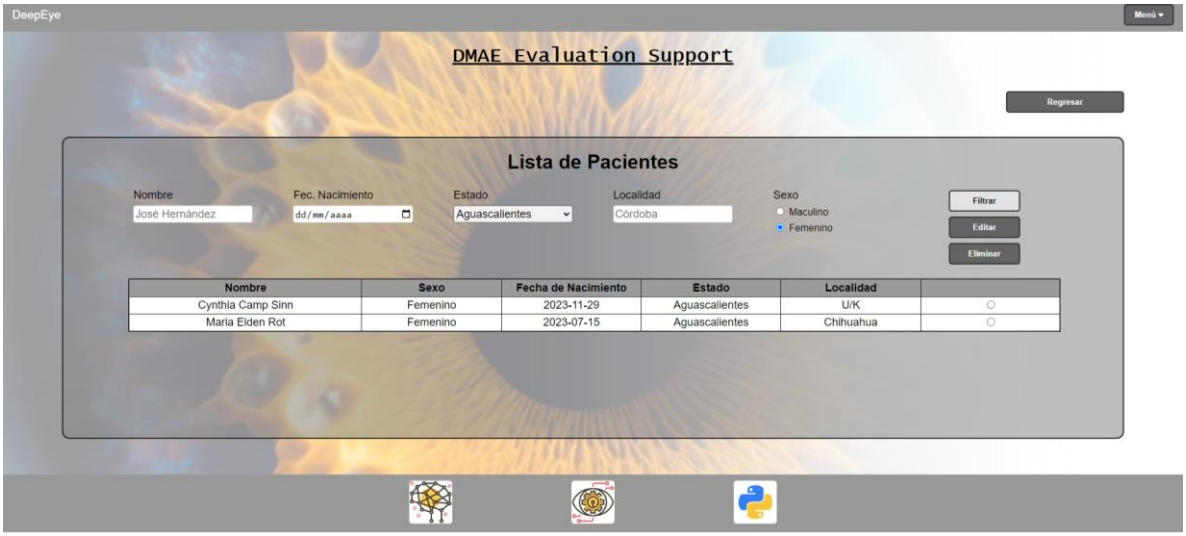


Figura 3.21 *Vista de reporte para impresión.*

Capítulo 4. Resultados

En este capítulo se presentan los resultados que arroja el sistema que se desarrolló en el presente proyecto. A saber, se construyó una aplicación web accesible e intuitiva con la que el usuario final tiene acceso al modelo entrenado para la clasificación de imágenes de fondo de ojo. El sistema en conjunto (aplicación web y modelo entrenado) ayudan en el proceso de detección de degeneración macular en su variante seca en imágenes de fondo de ojo.

El caso de estudio con el que fue realizada la prueba de funcionalidad de la solución, se detalla a continuación.

4.1 Caso de estudio

Para la obtención de las imágenes de fondo de ojo para la realización de las pruebas, se contó con la colaboración del Instituto de Oftalmología FAP Conde de Valenciana. La revisión de las mismas por el Dr. José Luis Rodríguez Loaiza con el número de cédula 2286932.

Se recibieron un total de 14 imágenes de fondo de ojo pertenecientes a diferentes pacientes. finalmente 13 de las imágenes se consideraron para como caso de estudio (Una de estas es un duplicado, por tal, se descartó). Las imágenes se recibieron en formato .jpg y en resoluciones varias.

Algunas de las imágenes presentaban anotaciones generadas por el dispositivo de captura y reflejos de luz provenientes del mismo. Además, en la mayoría de las imágenes, se capturaron también parte de las pestañas y parpados de los pacientes. Condiciones como la presencia de otras partes del ojo o reflejos de luz, no habían sido presenciados en los conjuntos de datos utilizados para el entrenamiento y evaluación del modelo, sin embargo, se considera que, tales condiciones, ayudaran a demostrar la robustes y adaptabilidad el modelo. El conjunto de imágenes recibidas se muestra en la Figura 4.1.

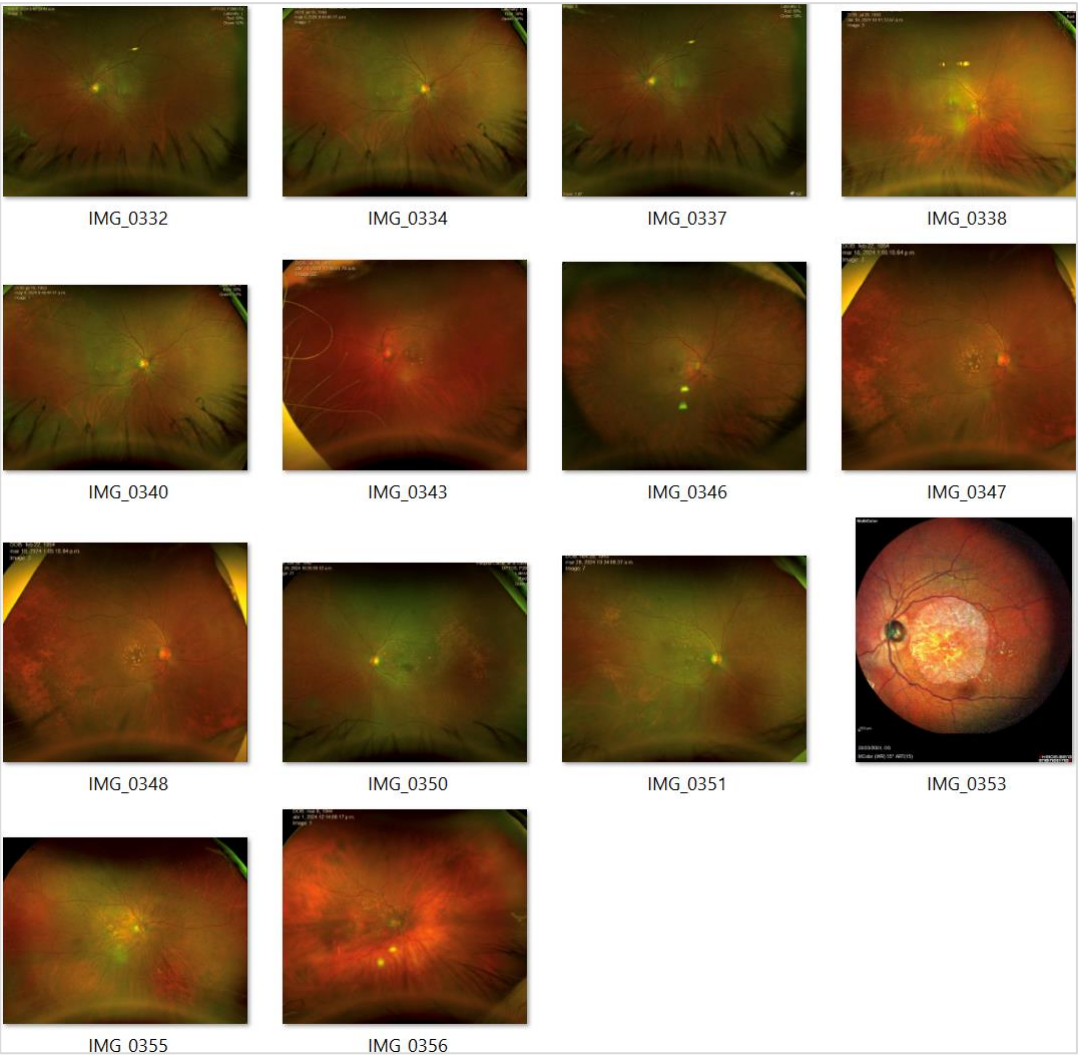


Figura 4.1 *Conjunto de imágenes recibido.*

4.2 Evaluación y resultados

4.2.1 Primera iteración de evaluación

Con este conjunto de datos se llevaron a cabo dos pruebas. La primera, directamente sobre el api de inferencia disponible en la plataforma de Hugging Face, y la segunda, sobre la aplicación desarrollada (DES). Lo anterior es debido a que la aplicación somete a las imágenes a un ajuste/preprocesamiento, buscando destacar las características de la imagen que el modelo entrenado utiliza para hacer su clasificación. Los resultados de las pruebas se muestran en la Tabla 4.1.

Tabla 4.1 *Resultados de primera iteración de pruebas.*

Modelo	vit-base-patch16-224-dmae-va-U5-100-iN			
Plataforma	**Imagen**	**Predicción**	**Exactitud**	**Etiqueta Real**
Hugging Face	IMG_0332	DMAE leve	0.5090	**DMAE leve**
	IMG_0334	DMAE leve	0.5780	**DMAE leve**
	IMG_0337	DMAE leve	0.4770	**DMAE leve**
	IMG_0338	DMAE moderada	0.3760	**DMAE moderada**
	IMG_0340	DMAE leve	0.6170	**DMAE leve**
	IMG_0343	DMAE leve	0.6290	**DMAE moderada**
	IMG_0346	DMAE leve	0.4080	**DMAE leve**
	IMG_0348	DMAE moderada	0.4760	**DMAE moderada**
	IMG_0350	DMAE moderada	0.5890	**DMAE moderada**
	IMG_0351	DMAE leve	0.5560	**DMAE moderada**
	IMG_0353	DMAE avanzada	0.6290	**DMAE avanzada**
	IMG_0355	DMAE leve	0.4160	**DMAE moderada**
	IMG_0356	DMAE leve	0.5900	**DMAE leve**
Aplicación *DES* (Con ajuste de imagen)	IMG_0332	DMAE leve	0.5802	**DMAE leve**
	IMG_0334	DMAE leve	0.5916	**DMAE leve**
	IMG_0337	DMAE leve	0.5801	**DMAE leve**
	IMG_0338	DMAE moderada	0.4394	**DMAE moderada**
	IMG_0340	DMAE leve	0.6342	**DMAE leve**
	IMG_0343	DMAE leve	0.6344	**DMAE moderada**
	IMG_0346	DMAE leve	0.5050	**DMAE leve**
	IMG_0348	DMAE moderada	0.4844	**DMAE moderada**
	IMG_0350	DMAE moderada	0.5973	**DMAE moderada**
	IMG_0351	DMAE leve	0.5593	**DMAE moderada**
	IMG_0353	DMAE avanzada	0.5923	**DMAE avanzada**
	IMG_0355	DMAE leve	0.3719	**DMAE moderada**
	IMG_0356	DMAE leve	0.5506	**DMAE leve**
Aciertos				**0.7692%**

Obsérvese que el preprocesamiento al que la aplicación sometió a las imágenes, en general aumentó la seguridad del modelo en su predicción (en un 0.8461% de las ocasiones), como se ilustra en la Figura 4.2 y la Figura 4.3. Lo anterior demuestra que efectivamente el ajuste aplicado es capaz de destacar las características que el modelo busca para realizar su clasificación.

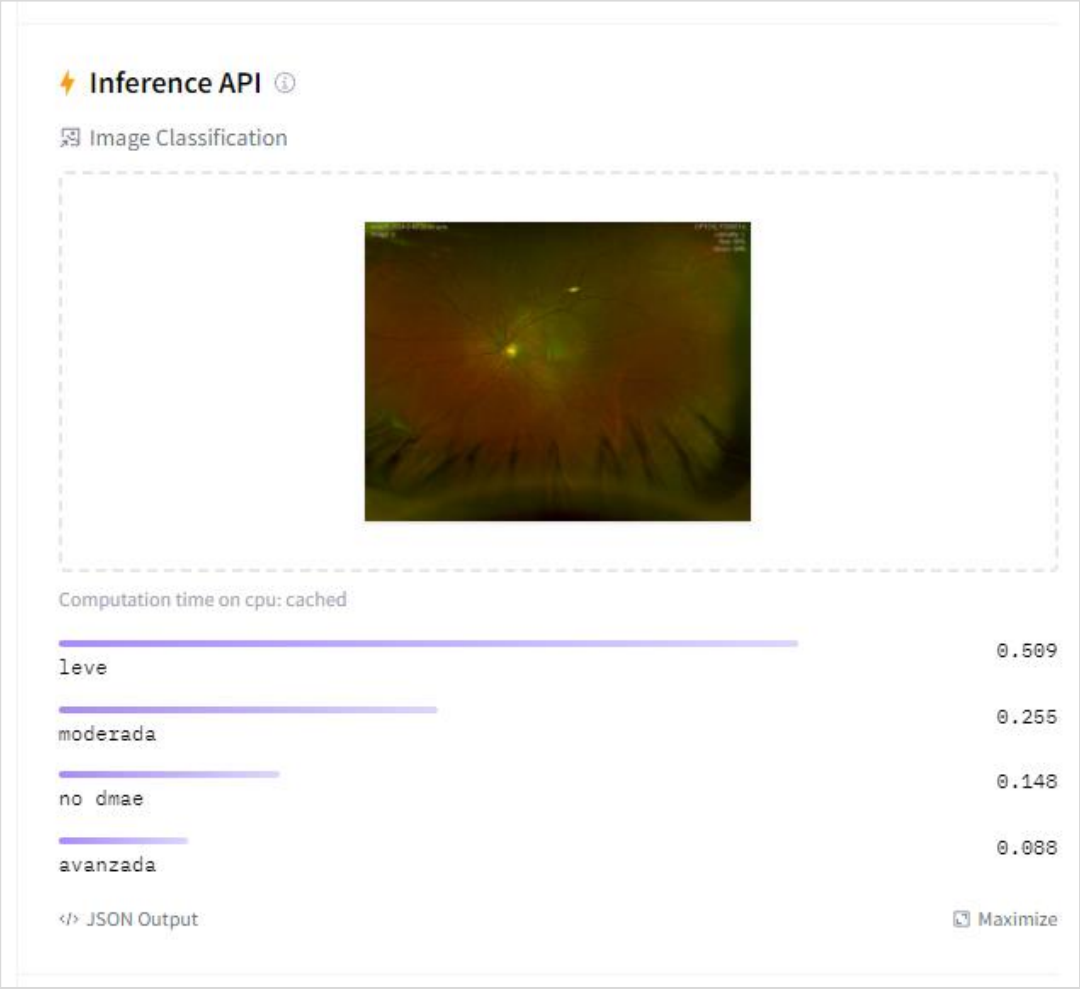


Figura 4.2 *Clasificación realizada sin preprocesamiento (Hugging Face).*

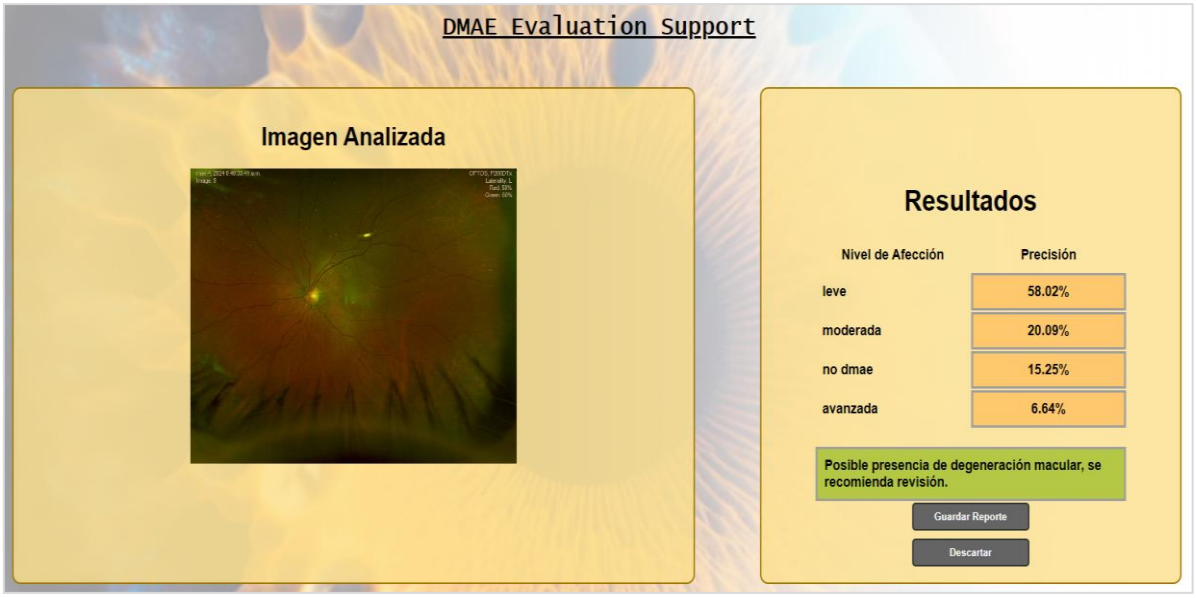


Figura 4.3 *Clasificación realizada con preprocesamiento (aplicación DES).*

Finalmente, como se detalló en la Tabla 4.1, el modelo entrenado obtuvo un porcentaje de aciertos del 0.7692% en la primera iteración de pruebas. Por encima del canal de azar, sin embargo, con ajustes este mejoró su desempeño.

4.2.3 Ajustes

Caso de estudio presenta diferencias claras en su formato respecto a las imágenes que estuvieron disponibles para el conjunto de entrenamiento, como se observa en la Figura 4.4.

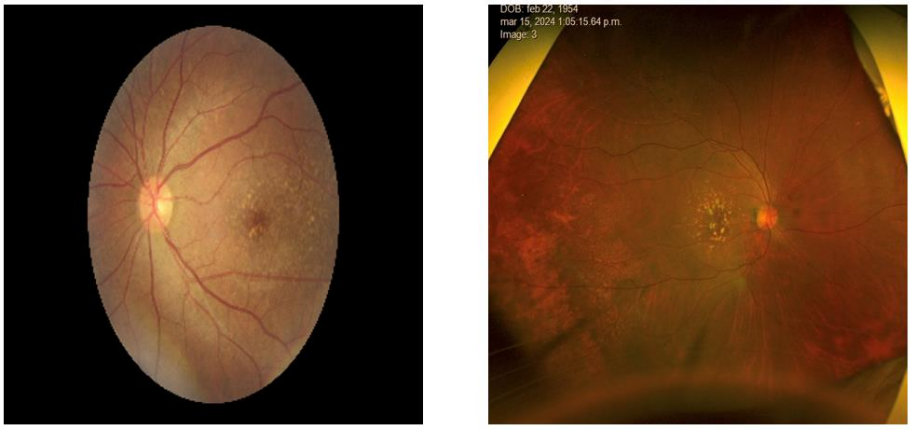


Figura 4.4 *Muestra de conjunto de entrenamiento (izquierda) y caso de estudio (derecha).*

Por lo tanto, se procedió a realizar ajustes en el conjunto de entrenamiento, para realizar una mejor generalización. En este caso, a las transformaciones de aumento de datos descritas en el punto 3.3.1.3, se agregó la transformación de acercamiento de imagen (zoom), un ejemplo se muestra en la Figura 4.5, esto debido a que las imágenes del caso de estudio se recopilaron con grados de acercamiento no presenciados previamente.

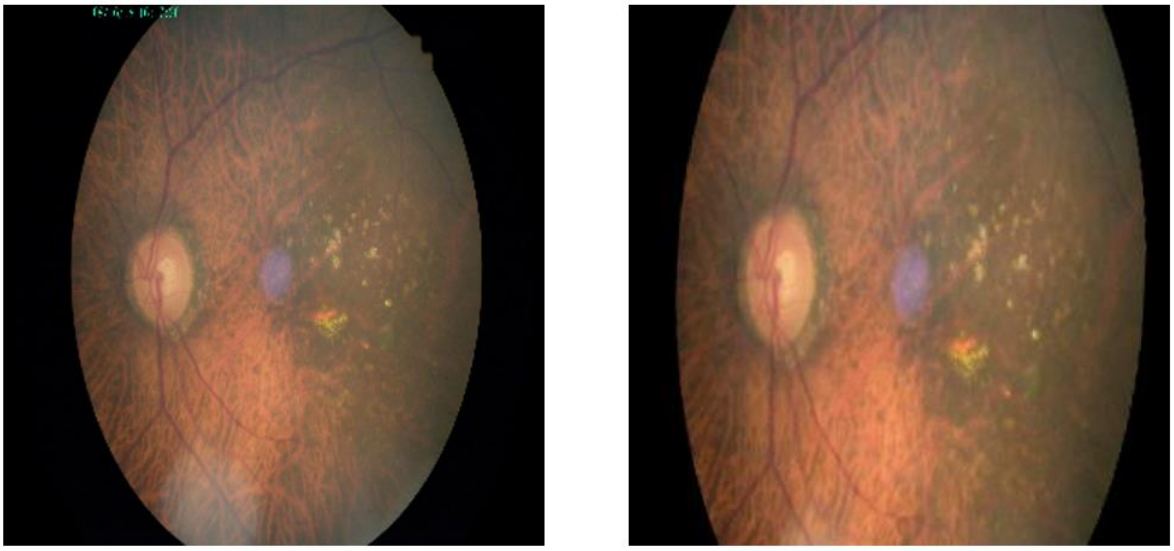


Figura 4.5 *Muestra de imagen con la transformación de acercamiento.*

Las demás transformaciones y el número de imágenes generadas se mantuvieron según lo previamente descrito.

Dado a que ahora el caso de estudio toma el lugar del conjunto de pruebas, el conjunto de imágenes usado en el conjunto de pruebas deja de tener una función, por lo tanto, el conjunto de 305 imágenes se segmentó de nuevo antes de realizar las transformaciones. Teniendo ahora 254 imágenes para el conjunto de entrenamiento y 51 para el conjunto de validación. Lo anterior, buscando una relación de un 85/15 entre cada clase de los conjuntos.

Finalmente, se contó con 2540 imágenes (10 generadas por cada imagen original) para el nuevo conjunto de entrenamiento, 51 para el conjunto de validación, y 13 (el caso de estudio) como conjunto de pruebas.

Se procedió a un nuevo proceso de entrenamiento, bajo los mismos hiperparámetros descritos en el punto 3.3.2, cambiando solo el conjunto de entrenamiento.

Se obtuvo un nuevo modelo, mostrado en la Figura 4.6.

Figura 4.6 *Tarjeta del modelo final.*

Los resultados de evaluación del último modelo se describen en la Tabla 4.2.

Tabla 4.2 *Evaluación del modelo final.*

Conjunto	Exactitud	Clase	Precisión	Sensibilidad	F1-Score
Validación	0.8627	**No dmae**	0.8000	0.8000	0.8000
		Leve	0.8800	0.9166	0.8979
		Moderada	0.8333	0.8823	0.8571
		Avanzada	1.0000	0.6000	0.7500

Con los resultados del modelo final para el conjunto de validación, se pasa a realizar una segunda iteración de pruebas con el caso de estudio.

4.2.4 Segunda iteración de evaluación

Se realizó una segunda iteración de evaluación, cuyos resultados se muestran en la Tabla 4.3.

Tabla 4.3 *Resultados de segunda iteración de pruebas.*

Modelo	vit-base-patch16-224-U8-10b			
Plataforma	**Imagen**	**Predicción**	**Exactitud**	**Etiqueta Real**
Hugging Face	IMG_0332	DMAE leve	0.8060	**DMAE leve**
	IMG_0334	DMAE leve	0.8001	**DMAE leve**
	IMG_0337	DMAE leve	0.5680	**DMAE leve**
	IMG_0338	DMAE moderada	0.7500	**DMAE moderada**
	IMG_0340	DMAE leve	0.8440	**DMAE leve**
	IMG_0343	DMAE leve	0.5560	**DMAE moderada**
	IMG_0346	DMAE leve	0.5760	**DMAE leve**
	IMG_0348	DMAE moderada	0.9000	**DMAE moderada**
	IMG_0350	DMAE moderada	0.8360	**DMAE moderada**
	IMG_0351	DMAE moderada	0.8570	**DMAE moderada**
	IMG_0353	DMAE avanzada	0.7800	**DMAE avanzada**
	IMG_0355	DMAE moderada	0.5250	**DMAE moderada**
	IMG_0356	DMAE leve	0.5490	**DMAE leve**
Aplicación *DES* (Con ajuste de imagen)	IMG_0332	DMAE leve	0.8037	**DMAE leve**
	IMG_0334	DMAE leve	0.8420	**DMAE leve**
	IMG_0337	DMAE leve	0.6122	**DMAE leve**
	IMG_0338	DMAE moderada	0.7671	**DMAE moderada**
	IMG_0340	DMAE leve	0.8592	**DMAE leve**
	IMG_0343	DMAE leve	0.5804	**DMAE moderada**
	IMG_0346	DMAE leve	0.6045	**DMAE leve**
	IMG_0348	DMAE moderada	0.9012	**DMAE moderada**
	IMG_0350	DMAE moderada	0.8239	**DMAE moderada**

	IMG_0351	DMAE moderada	0.8364	**DMAE moderada**
	IMG_0353	DMAE avanzada	0.7795	**DMAE avanzada**
	IMG_0355	DMAE moderada	0.5258	**DMAE moderada**
	IMG_0356	DMAE leve	0.5489	**DMAE leve**
Aciertos				**0.9230%**

Como los muestran los nuevos resultados, con el ajuste realizado, se aumentó considerablemente el desempeño en el proceso de clasificación, alcanzando un máximo del **0.9230%** a exactitud, esto lo pone a la altura de los resultados reportados en la literatura, destacado el hecho de que, los trabajos relacionados se enfocan en la clasificación binaria de la enfermedad, mientras el modelo desarrollado, es capaz de realizar una clasificación multiclase.

De los resultados mostrados anteriormente, se concluye que la aplicación generada es una herramienta eficiente como apoyo en la clasificación, capaz de hacer predicciones por encima del nivel de azar.

Capítulo 5. Conclusiones y Recomendaciones

En este capítulo se exponen las conclusiones obtenidas, así como las recomendaciones derivadas de la solución desarrollada para abordar el problema planteado en el proyecto.

5.1 Conclusiones

En el desarrollo de este proyecto de tesis, se desarrolló una herramienta la cual cumple con el objetivo general del proyecto, así como de los respectivos objetivos específicos. A saber, hacer uso de mecanismos de visión artificial y aprendizaje profundo para la creación de una aplicación la cual ayuda a la clasificación de imágenes de fondo de ojo, para la detección de la degeneración macular en sus diferentes grados de avance.

Para alcanzar los objetivos de este proyecto, se presentaron y analizaron múltiples trabajos e investigaciones en este proyecto de tesis, en los que se muestra la aplicación de la tecnología para la detección y clasificación de la degeneración macular. También, se identificaron las arquitecturas de visión artificial más adecuadas y las bibliotecas de preprocesamiento de datos, para mejorar los procesos de entrenamiento.

De los análisis realizados, se seleccionó a ViT como la arquitectura para generar el modelo de clasificación, así como a PyTorch como la biblioteca para la implementación del modelo. Keras fue la tecnología seleccionada para realizar el aumento de datos. Con la selección lista, se procedió a realizar múltiples iteraciones de entrenamiento, buscando pulir el desempeño de los modelos en su tarea de clasificar imágenes de fondo de ojo.

Con los datos disponibles en los conjuntos de datos públicos, se obtuvo una exactitud del 0.8667 en el conjunto de validación, para el último modelo entrenado (previo a conocer las condiciones del caso de estudio). Con la aplicación

desarrollada para el usuario final (implementada con Node.js) se agregó un módulo de preprocesamiento, la cual ajusta la imagen a analizar, destacando las características buscadas por el modelo, permitiendo así al modelo tener más certeza en sus predicciones.

Una vez teniendo acceso al caso de estudio, se conocieron circunstancias que no estaban presentes en los conjuntos de datos disponibles. Una vez con conocimiento de tales características, se ajustó el modelo para que tuviera en cuenta estas nuevas circunstancias, alcanzando así un 0.9230% de precisión en su proceso de clasificación, lo que lo coloca a la altura de lo reportado en la literatura, con la adición, de que, dichos modelos se limitan a una clasificación binaria, mientras que el modelo desarrollado en este proyecto es multiclase.

Finalmente, con el modelo creado y la aplicación desarrollada, bautizada bajo el nombre de *DES* (*DMAE Evaluation Support o* Soporte para evaluación DMAE), se tiene una herramienta de apoyo para la detección temprana y multiclase de la degeneración macular asociada con la edad.

Se concluye que, la herramienta *DES* beneficiará a la sociedad brindando un mecanismo accesible y rápido para la detección temprana de la degeneración macular. Sobre todo, ahí donde por las circunstancias, los profesionales de la oftalmología no tengan alcance para diagnosticar a las personas. También, el módulo tiene la capacidad de fungir como una segunda opinión que refuerce el diagnostico de los expertos en la oftalmología.

5.2 Recomendaciones

A continuación, se listarán una serie de recomendación para futuros trabajos relacionados.

5.2.1 Mejoras el preprocesamiento de imagen

Las imágenes obtenidas para el caso de estudio varían respecto a las utilizadas en el proceso de entrenamiento, en el aspecto que las imágenes de entrenamiento se

ajustaron previamente (se cortaron) para dejar visible solo el área de la imagen sobre la cual se realiza el análisis, esto debería aplicarse también a las imágenes que analiza el sistema.
El resultado de este recorte debe ser similar a como se muestra en la Figura 5.1.

Figura 5.1 *Imagen del caso de estudio (IMG-0332) recortada.*

Como la finalidad de la aplicación es ser lo más rápida y accesible posible, no es posible dejar la tarea de aplicar tal recorte en la imagen al usuario final, por lo que, en trabajos futuros, se recomienda la adicción de un algoritmo que realice tal recorte durante el preprocesamiento de la imagen; haciendo énfasis en el área que el modelo entrenado está destinado a analizar. Se espera esto aumente el desempeño de futuros proyectos.

5.2.2 Manifestación húmeda de la degeneración macular

La manifestación húmeda de la degeneración macular es un porcentaje importante de los casos avanzados de DMAE, se recomienda entonces que, en futuros trabajos se agregue la clase de “DMAE húmeda”, a las cuatro clases ya existente es este proyecto.

5.2.3 Ajustes en el proceso de aumento de datos

Durante los procesos de entrenamiento se probaron diferentes configuraciones de aumento de datos. Algunas de estas configuraciones no necesariamente mejoraron el desempeño del modelo, sino que, agregaron ruido al conjunto de entrenamiento, se recomienda que, se ajuste el proceso de aumento de datos a una nueva configuración que sea más adecuada con lo planteado en el punto 5.2.2.

Productos Académicos

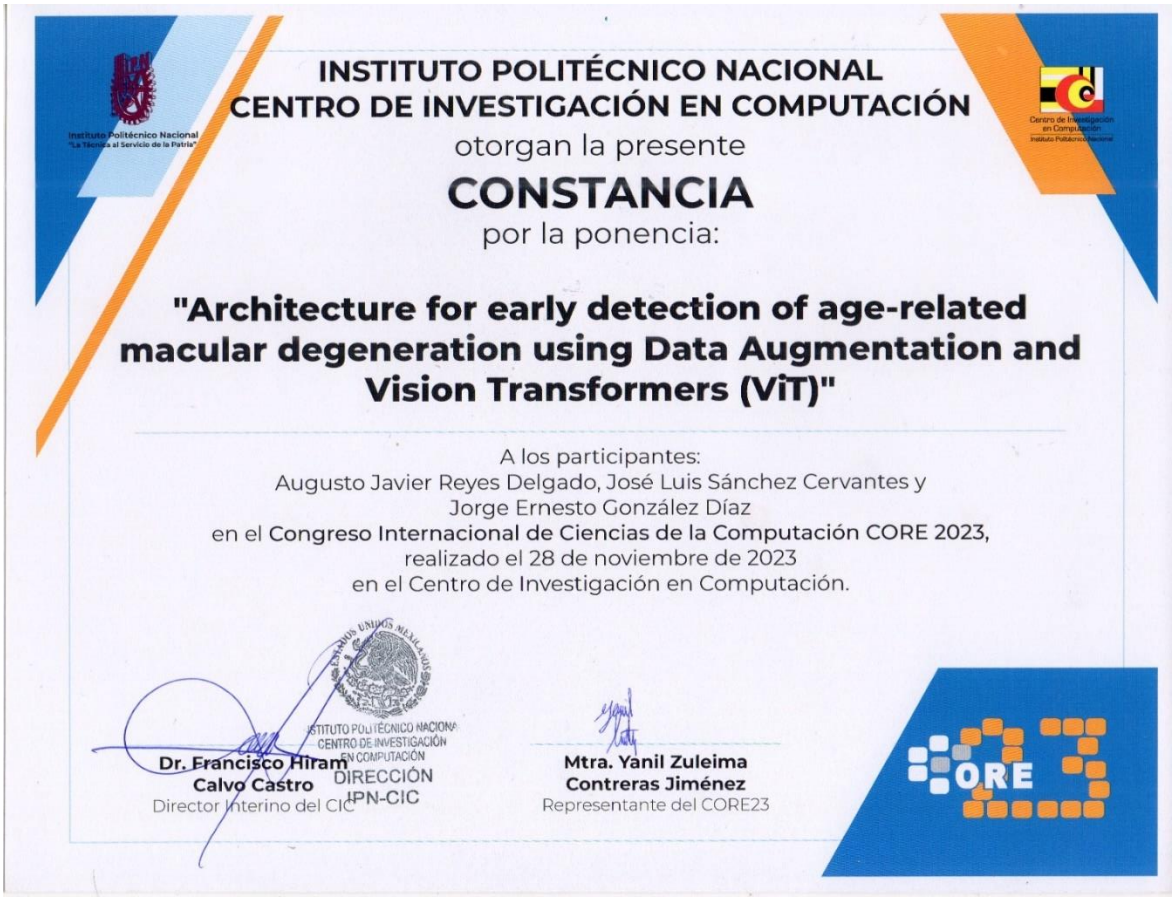
En esta sección se muestran los productos académicos que se han realizado.

6.1 Artículos de congreso

6.1.1 Primer artículo

Architecture for early detection of age-related macular degeneration using Data Augmentation and Vision Transformers (ViT).

Augusto Javier Reyes Delgado, Jorge Ernesto González Díaz, José Luis Sánchez Cervantes, Giner Alor Hernández, José Luis Rodríguez Loaiza, Yara Anahí Jiménez Nieto.



INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

otorgan la presente

CONSTANCIA

por la ponencia:

"Architecture for early detection of age-related macular degeneration using Data Augmentation and Vision Transformers (ViT)"

A los participantes:
Augusto Javier Reyes Delgado, José Luis Sánchez Cervantes y
Jorge Ernesto González Díaz
en el Congreso Internacional de Ciencias de la Computación CORE 2023,
realizado el 28 de noviembre de 2023
en el Centro de Investigación en Computación.

Dr. Francisco Hiram Calvo Castro
Director Interino del CIC

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN
DIRECCIÓN
IPN-CIC

Mtra. Yanil Zuleima Contreras Jiménez
Representante del CORE23

CORE

6.1.1.1 PDF publicación del artículo

ISSN 2395-8618

Architecture for Early Detection of Age-Related Macular Degeneration Using Data Augmentation and Vision Transformers (ViT)

Augusto Javier Reyes-Delgado, Jorge Ernesto González-Díaz, José Luis Sánchez-Cervantes, Giner Alor-Hernández, José Luis Rodríguez-Loaiza, Yara Anahi Jiménez-Nieto

***Abstract*—Age-Related Macular Degeneration (AMD) is a leading cause of blindness among the geriatric population worldwide. While there is no definitive cure for this pathology, early identification of AMD allows for effective treatment administration. This paper introduces an architecture for a specialized module to identifying AMD at various evolutionary stages. An in-depth analysis was conducted examining investigations related to detecting age-related macular degeneration (AMD), with a focus on studies utilizing deep learning techniques and vision transformers. It is important to emphasize that most of these works have only addressed binary disease detection. Our initiative incorporates an architecture that emphasizes data augmentation in the training set and utilizes the ViT vision transformer for analyzing retina images. The main aim is to attain a differentiated categorization (non-AMD, mild, moderate, and advanced) that serves as a basis for diagnosing AMD. The ViT trained model has shown 96.55% accuracy in this classification. Hence, it can be inferred that the outlined module holds noteworthy value as an additional support tool for ophthalmologists in the precise detection of Age-Related Macular Degeneration.**

***Index Terms*—Age-related macular degeneration, vision transformers, data augmentation.**

I. Introduction

Age-Related Macular Degeneration (AMD) is one of the main causes of vision loss among the population over 50, particularly affecting developed countries [1]. It stands as the third cause of blindness worldwide, with a moderate or severe prevalence estimated at 10.4 million people according to the World Health Organization [2].

AMD causes the loss of central vision, implying that fine details are not perceived up close or far away in the center of the visual focus, even though peripheral vision remains normal. In addition to age, AMD is associated with overweight, smoking, hypertension, heart diseases, and high-fat consumption, conditions common in developed countries, as well as some developing countries such as Mexico.

In this context, in Mexico, over 30 million suffer from hypertension, and over 70% of the population is obese, it according to data from the Mexican government and the National Institute of Public Health [3,4]. As far as AMD is concerned, this condition is divided into two categories: 1) The most common being "dry" AMD (approximately 80% of cases), characterized by the appearance of Drusen (fat clusters in the retina, and 2) "wet" AMD, which occurs due to the growth of abnormal blood vessels under the retina [5].

Although the disease is highly treatable in its early stages, it is often asymptomatic until it reaches an advanced state, by which time the damage is irreversible. This fact underscores the urgent need to develop tools for early detection. Optical coherence tomography has become one of the predominant diagnostic methods for this disease (along with fundus images). Observing optical coherence images and fundus images are the most common methods for diagnosing the condition. Our proposal leans on computer vision techniques (Data Augmentation) and deep learning (ViT) to aid in the analysis of features in fundus images, focusing on the most common category of AMD, namely the so-called "dry" type.

Currently, artificial intelligence models, predominantly based on convolutional neural networks, have been developed to address this issue. However, in recent years, computer vision techniques have emerged as the new state-of-the-art in image analysis. These techniques are highlighted for their ability to capture the entire context of the image under analysis. This process involves segmenting the image into patches and subsequently representing them as a vector, which is processed through a transformer-type encoder [6].

Therefore, the contribution of this research, is an architecture for early AMD detection that integrates computer vision techniques (Data Augmentation) and deep learning (Vision Transformers, ViT). This is an essential part of an ongoing technological development integrating an AI module for feature analysis to detect and classify AMD in various stages (No-AMD, mild, moderate, and advanced).

This paper is organized as follows: Section 2 presents the state of the art, Section 3 introduces the design of the module's architecture, Section 4 discusses the training outcome validation, and finally, Section 5 presents the conclusions and future work.

II. State of the Art

There are several initiatives related to AMD detection using convolutional neural networks and vision transformers, such as those analyzed and briefly described below. A DCNN (Deep Convolutional Neural Network) with a 13-layer architecture to

Manuscript received on 14/04/2023, accepted for publication on 10/06/2023.
A.J. Reyes-Delgado, J. E. González-Díaz, J. L. Sánchez-Cervantes, G. Alor-Hernández, J.L. Rodríguez-Loaiza are with the Instituto de Oftalmología Conde de Valenciana, Tecnológico Nacional de México, Mexico ({m17010207, d04010291,jose.sc, giner.ah}@orizaba.tecnm.mx, jose.rodriguez@ institutodeoftalmologia.org).

Y. Anahí Jiménez-Nieto is with the Universidad Veracruzana, Facultad de Negocios y Tecnologías campus Ixtaczoquitlán, Mexico (yjimenez@uv.mx).

6.1.2 Segundo artículo

Detección temprana de degeneración macular asociada con la edad mediante Arquitecturas basadas en Transformadores de Visión – Un estudio comparativo.

Augusto Javier Reyes Delgado, Jorge Ernesto González Díaz, Yara Anahí Jiménez Nieto, Adolfo Rodríguez Parada, José Luis Sánchez Cervantes, José Luis Rodríguez Loaiza.

6.2 Retribución social

6.2.1 Primera retribución social

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba
División de Estudios de Posgrado e Investigación

Constancia de actividades de retribución social

Actividad 1. Divulgar la ciencia y tecnología a niños y jóvenes, mediante cursos y pláticas.
Mediante la presentación de la ponencia "Arquitectura para la detección temprana de la degeneración macular asociada a la edad mediante Data Augmentation y Vision Transformers (ViT)"

Descripción de la actividad: Se llevó a cabo una ponencia explicando el uso visión artificial y la arquitectura ViT para la detección de la degeneración macular, ante estudiantes de la licenciatura en ingeniería en sistemas computaciones.

Fecha de inicio: 01 de diciembre 2023

Fecha de término: 01 de diciembre 2023

Institución en la que se realizó la actividad: Instituto Tecnológico Superior de Cosamaloapan

Nombre del responsable de supervisar la actividad: José Luis Sánchez Cervantes

Datos de contacto del responsable de la actividad: jose.sc@orizaba.tecnm.mx

Descripción del impacto social de la actividad: La divulgación del uso de la visión artificial para la salud visual tiene el siguiente impacto, aumentar la conciencia sobre la salud visual e introducir a estudiantes del área informática en aplicaciones de la inteligencia artificial.

I.S.C Augusto Javier Reyes Delgado
CVU: 1221852

Dr. José Luis Sánchez Cervantes

Av. Oriente 9 Núm.852, Colonia Emiliano Zapata. C.P. 94320 Orizaba, Veracruz.
Tel. 01 (272)1105360 e-mail: dir_orizaba@tecnm.mx tecnm.mx | orizaba.tecnm.mx

2024 Felipe Carrillo PUERTO

El Instituto Tecnológico Superior de Cosamaloapan
otorga el presente

RECONOCIMIENTO

A: ISC. AUGUSTO JAVIER REYES DELGADO

Por su valiosa participación como ponente de la conferencia: ARQUITECTURA PARA LA DETECCIÓN TEMPRANA DE LA DEGENERACIÓN MACULAR ASOCIADA A LA EDAD MEDIANTE DATA AUGMENTATION Y VISION TRANSFORMERS. En la semana de Ingeniería en Sistemas Computacionales 2023, celebrada del 27 de noviembre al 1ro de diciembre.

COSAMALOAPAN, VERACRUZ, 01 DE DICIEMBRE 2023

Ing. Aldo Fred Hernández Aguirre
Director General

VERACRUZ

6.2.2 Segunda retribución social

Instituto Tecnológico de Orizaba
División de Estudios de Posgrado e Investigación

Constancia de actividades de retribución social

Actividad 2. Participar en foros de intercambio de experiencias sociales/Institucionales.
Mediante la participación en la Charla de Café titulada "La Inteligencia Artificial aplicada en el cuidado de la salud visual".

Descripción de la actividad: Se llevó a cabo una plática explicando el uso visión artificial para el cuidado de la salud visual por los estudiantes de la MSC. Augusto Javier Reyes Delgado y Roberto Márquez Castro, Estudiante del DCI. Jorge Ernesto González Días y el Dr. José Luis Sánchez Cervantes.

Fecha de inicio: 12 de abril 2024

Fecha de término: 12 de abril 2024

Institución en la que se realizó la actividad: Tecnológico nacional de México – Campus Orizaba

Nombre del responsable de supervisar la actividad: José Luis Sánchez Cervantes

Datos de contacto del responsable de la actividad: jose.sc@orizaba.tecnm.mx

Descripción del impacto social de la actividad: La divulgación del uso de la inteligencia artificial para la salud visual tiene como impacto aumentar la conciencia sobre la salud visual e inspirar a estudiantes del área informática a desarrollar habilidades en inteligencia artificial.

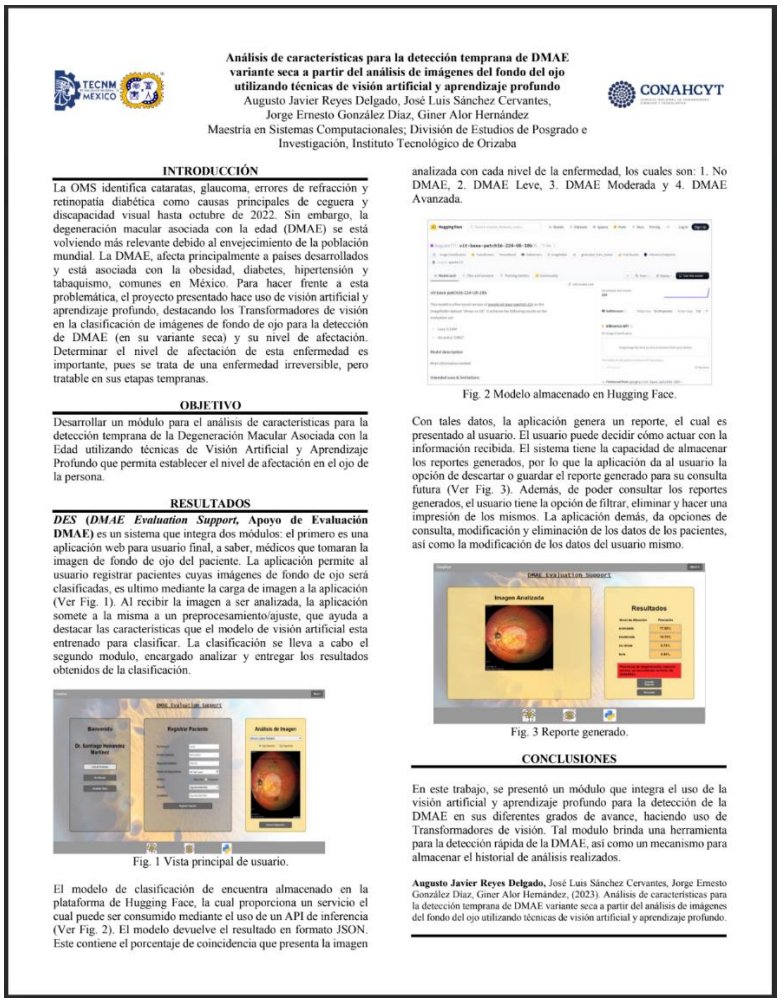
I.S.C Augusto Javier Reyes Delgado
CVU: 1221852

Dr. José Luis Sánchez Cervantes

Av. Oriente 9 Núm.852, Colonia Emiliano Zapata. C.P. 94320 Orizaba, Veracruz.
Tel. 01 (272)1105360 e-mail: dir_orizaba@tecnm.mx tecnm.mx | orizaba.tecnm.mx


EDUCACIÓN
Charlas de café
MSC
0:10 / 1:20:00
TECNM - Orizaba
12 de abril ·
Charla de Café titulada "La Inteligencia Artificial aplicada en el cuidado de la salud visual" por Estudiantes de la MSC. Augusto Javier Reyes Delgado y Roberto... Ver más

6.2.3 Póster y video



Análisis de características para la detección temprana de DMAE variante seca a partir del análisis de imágenes del fondo del ojo utilizando técnicas de visión artificial y aprendizaje profundo

Augusto Javier Reyes Delgado, José Luis Sánchez Cervantes,
Jorge Ernesto González Díaz, Giner Alor Hernández
Maestría en Sistemas Computacionales; División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Orizaba

INTRODUCCIÓN

La OMS identifica cataratas, glaucoma, errores de refracción y retinopatía diabética como causas principales de ceguera y discapacidad visual hasta octubre de 2022. Sin embargo, la degeneración macular asociada con la edad (DMAE) se está volviendo más relevante debido al envejecimiento de la población mundial. La DMAE, afecta principalmente a países desarrollados y está asociada con la obesidad, diabetes, hipertensión y tabaquismo, comunes en México. Para hacer frente a esta problemática, el proyecto presentado hace uso de visión artificial y aprendizaje profundo, destacando los Transformadores de visión en la clasificación de imágenes de fondo de ojo para la detección de DMAE (en su variante seca) y su nivel de afectación. Determinar el nivel de afectación de esta enfermedad es importante, pues se trata de una enfermedad irreversible, pero tratable en sus etapas tempranas.

OBJETIVO

Desarrollar un módulo para el análisis de características para la detección temprana de la Degeneración Macular Asociada con la Edad utilizando técnicas de Visión Artificial y Aprendizaje Profundo que permita establecer el nivel de afectación en el ojo de la persona.

RESULTADOS

***DES* (*DMAE Evaluation Support*, Apoyo de Evaluación DMAE)** es un sistema que integra dos módulos: el primero es una aplicación web para usuario final, a saber, médicos que tomaran la imagen de fondo de ojo del paciente. La aplicación permite al usuario registrar pacientes cuyas imágenes de fondo de ojo será clasificadas, es ultimo mediante la carga de imagen a la aplicación (Ver Fig. 1). Al recibir la imagen a ser analizada, la aplicación somete a la misma a un preprocesamiento/ajuste, que ayuda a destacar las características que el modelo de visión artificial esta entrenado para clasificar. La clasificación se lleva a cabo el segundo modulo, encargado analizar y entregar los resultados obtenidos de la clasificación.

Fig. 1 Vista principal de usuario.

El modelo de clasificación de encuentra almacenado en la plataforma de Hugging Face, la cual proporciona un servicio el cual puede ser consumido mediante el uso de un API de inferencia (Ver Fig. 2). El modelo devuelve el resultado en formato JSON. Este contiene el porcentaje de coincidencia que presenta la imagen analizada con cada nivel de la enfermedad, los cuales son: 1. No DMAE, 2. DMAE Leve, 3. DMAE Moderada y 4. DMAE Avanzada.

Fig. 2 Modelo almacenado en Hugging Face.

Con tales datos, la aplicación genera un reporte, el cual es presentado al usuario. El usuario puede decidir cómo actuar con la información recibida. El sistema tiene la capacidad de almacenar los reportes generados, por lo que la aplicación da al usuario la opción de descartar o guardar el reporte generado para su consulta futura (Ver Fig. 3). Además, de poder consultar los reportes generados, el usuario tiene la opción de filtrar, eliminar y hacer una impresión de los mismos. La aplicación demás, da opciones de consulta, modificación y eliminación de los datos de los pacientes, así como la modificación de los datos del usuario mismo.

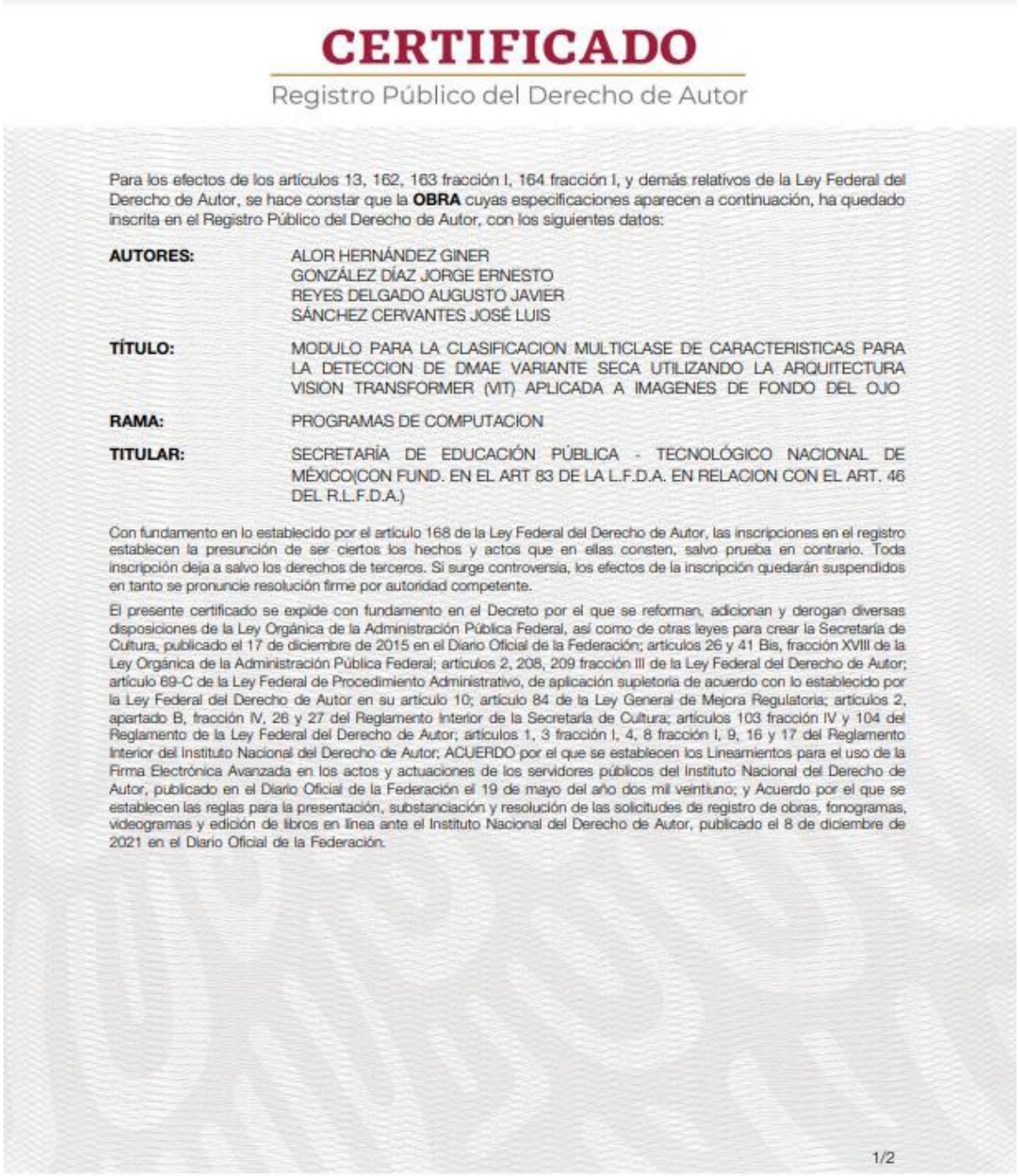
Fig. 3 Reporte generado.

CONCLUSIONES

En este trabajo, se presentó un módulo que integra el uso de la visión artificial y aprendizaje profundo para la detección de la DMAE en sus diferentes grados de avance, haciendo uso de Transformadores de visión. Tal modulo brinda una herramienta para la detección rápida de la DMAE, así como un mecanismo para almacenar el historial de análisis realizados.

Augusto Javier Reyes Delgado, José Luis Sánchez Cervantes, Jorge Ernesto González Díaz, Giner Alor Hernández, (2023). Análisis de características para la detección temprana de DMAE variante seca a partir del análisis de imágenes del fondo del ojo utilizando técnicas de visión artificial y aprendizaje profundo.

6.3 Registro de derechos de autor



CERTIFICADO

Registro Público del Derecho de Autor

Para los efectos de los artículos 13, 162, 163 fracción I, 164 fracción I, y demás relativos de la Ley Federal del Derecho de Autor, se hace constar que la **OBRA** cuyas especificaciones aparecen a continuación, ha quedado inscrita en el Registro Público del Derecho de Autor, con los siguientes datos:

AUTORES:	ALOR HERNÁNDEZ GINER GONZÁLEZ DÍAZ JORGE ERNESTO REYES DELGADO AUGUSTO JAVIER SÁNCHEZ CERVANTES JOSÉ LUIS
TÍTULO:	MODULO PARA LA CLASIFICACION MULTICLASE DE CARACTERISTICAS PARA LA DETECCION DE DMAE VARIANTE SECA UTILIZANDO LA ARQUITECTURA VISION TRANSFORMER (VIT) APLICADA A IMAGENES DE FONDO DEL OJO
RAMA:	PROGRAMAS DE COMPUTACION
TITULAR:	SECRETARÍA DE EDUCACIÓN PÚBLICA - TECNOLÓGICO NACIONAL DE MÉXICO(CON FUND. EN EL ART 83 DE LA L.F.D.A. EN RELACION CON EL ART. 46 DEL R.L.F.D.A.)

Con fundamento en lo establecido por el artículo 168 de la Ley Federal del Derecho de Autor, las inscripciones en el registro establecen la presunción de ser ciertos los hechos y actos que en ellas consten, salvo prueba en contrario. Toda inscripción deja a salvo los derechos de terceros. Si surge controversia, los efectos de la inscripción quedarán suspendidos en tanto se pronuncie resolución firme por autoridad competente.

El presente certificado se expide con fundamento en el Decreto por el que se reforman, adicionan y derogan diversas disposiciones de la Ley Orgánica de la Administración Pública Federal, así como de otras leyes para crear la Secretaría de Cultura, publicado el 17 de diciembre de 2015 en el Diario Oficial de la Federación; artículos 26 y 41 Bis, fracción XVIII de la Ley Orgánica de la Administración Pública Federal; artículos 2, 208, 209 fracción III de la Ley Federal del Derecho de Autor; artículo 69-C de la Ley Federal de Procedimiento Administrativo, de aplicación supletoria de acuerdo con lo establecido por la Ley Federal del Derecho de Autor en su artículo 10; artículo 84 de la Ley General de Mejora Regulatoria; artículos 2, apartado B, fracción IV, 26 y 27 del Reglamento Interior de la Secretaría de Cultura; artículos 103 fracción IV y 104 del Reglamento de la Ley Federal del Derecho de Autor; artículos 1, 3 fracción I, 4, 8 fracción I, 9, 16 y 17 del Reglamento Interior del Instituto Nacional del Derecho de Autor; ACUERDO por el que se establecen los Lineamientos para el uso de la Firma Electrónica Avanzada en los actos y actuaciones de los servidores públicos del Instituto Nacional del Derecho de Autor, publicado en el Diario Oficial de la Federación el 19 de mayo del año dos mil veintiuno; y Acuerdo por el que se establecen las reglas para la presentación, substanciación y resolución de las solicitudes de registro de obras, fonogramas, videogramas y edición de libros en línea ante el Instituto Nacional del Derecho de Autor, publicado el 8 de diciembre de 2021 en el Diario Oficial de la Federación.

1/2

CULTURA | INDAUTOR

CERTIFICADO

Registro Público del Derecho de Autor

El presente documento electrónico ha sido firmado mediante el uso de la firma electrónica avanzada por el servidor público competente, amparada por un certificado digital vigente a la fecha de su elaboración, y es válido de conformidad con lo dispuesto en los artículos 7 y 9, fracción I, de la Ley de Firma Electrónica Avanzada y artículo 12 de su Reglamento.

Número de Registro: 03-2024-061409251300-01

Ciudad de México, a 14 de junio de 2024

EL DIRECTOR DEL REGISTRO PÚBLICO DEL DERECHO DE AUTOR

JESÚS PARETS GÓMEZ

HHTuoBuhX6t1jYtcrtwUUdVNrimbkAzOj0PFaYvvO6eSkAzDvhQrJsziM/EmmU7QI4jHYq63wQ8ImQKks7LmId
Lc9WwOzmOrl7OTGcxeXThZdbgWvtcdaAc7P1VOkG0+Hz2Xb/F8lyn2gSkhql+EGVsA389c1jdvmei/forp2n0EJ
lo2/TtcegR5BlvqYhU6DQSDkFGku5r1v1H3W6xyDDg+YD3KQ5cdqSoZpauvg2KEGaUx6GFj43D0BxOsg4o6f
BaOAdNEFmvc01olGLTcpYhdsYdcNZ1grqKwKFxKOtpKo4GD0dP7aTDHYS6LdZ1SFv/nlnHqoJke1T/l9BxA=

2/2

CULTURA | INDAUTOR

CERTIFICADO

Registro Público del Derecho de Autor

Para los efectos de los artículos 13, 162, 163 fracción I, 164 fracción I, y demás relativos de la Ley Federal del Derecho de Autor, se hace constar que la **OBRA** cuyas especificaciones aparecen a continuación, ha quedado inscrita en el Registro Público del Derecho de Autor, con los siguientes datos:

AUTORES:	ALOR HERNÁNDEZ GINER GONZÁLEZ DÍAZ JORGE ERNESTO REYES DELGADO AUGUSTO JAVIER SÁNCHEZ CERVANTES JOSÉ LUIS
TÍTULO:	MODULO PARA LA OPTIMIZACION DE IMAGENES DE FONDO DE OJO PARA LA CLASIFICACION MULTICLASE DE DMAE VARIANTE SECA UTILIZANDO DATA AUGMENTING
RAMA:	PROGRAMAS DE COMPUTACION
TITULAR:	SECRETARÍA DE EDUCACIÓN PÚBLICA - TECNOLÓGICO NACIONAL DE MÉXICO(CON FUND. EN EL ART 83 DE LA L.F.D.A. EN RELACION CON EL ART. 46 DEL R.L.F.D.A.)

Con fundamento en lo establecido por el artículo 168 de la Ley Federal del Derecho de Autor, las inscripciones en el registro establecen la presunción de ser ciertos los hechos y actos que en ellas consten, salvo prueba en contrario. Toda inscripción deja a salvo los derechos de terceros. Si surge controversia, los efectos de la inscripción quedarán suspendidos en tanto se pronuncie resolución firme por autoridad competente.

El presente certificado se expide con fundamento en el Decreto por el que se reforman, adicionan y derogan diversas disposiciones de la Ley Orgánica de la Administración Pública Federal, así como de otras leyes para crear la Secretaría de Cultura, publicado el 17 de diciembre de 2015 en el Diario Oficial de la Federación; artículos 26 y 41 Bis, fracción XVIII de la Ley Orgánica de la Administración Pública Federal; artículos 2, 208, 209 fracción III de la Ley Federal del Derecho de Autor; artículo 69-C de la Ley Federal de Procedimiento Administrativo, de aplicación supletoria de acuerdo con lo establecido por la Ley Federal del Derecho de Autor en su artículo 10; artículo 84 de la Ley General de Mejora Regulatoria; artículos 2, apartado B, fracción IV, 26 y 27 del Reglamento Interior de la Secretaría de Cultura; artículos 103 fracción IV y 104 del Reglamento de la Ley Federal del Derecho de Autor; artículos 1, 3 fracción I, 4, 8 fracción I, 9, 16 y 17 del Reglamento Interior del Instituto Nacional del Derecho de Autor; ACUERDO por el que se establecen los Lineamientos para el uso de la Firma Electrónica Avanzada en los actos y actuaciones de los servidores públicos del Instituto Nacional del Derecho de Autor, publicado en el Diario Oficial de la Federación el 19 de mayo del año dos mil veintiuno; y Acuerdo por el que se establecen las reglas para la presentación, substanciación y resolución de las solicitudes de registro de obras, fonogramas, videogramas y edición de libros en línea ante el Instituto Nacional del Derecho de Autor, publicado el 8 de diciembre de 2021 en el Diario Oficial de la Federación.

1/2

CULTURA
SECRETARÍA DE CULTURA

INDAUTOR
INSTITUTO NACIONAL DEL DERECHO DE AUTOR

CERTIFICADO

Registro Público del Derecho de Autor

El presente documento electrónico ha sido firmado mediante el uso de la firma electrónica avanzada por el servidor público competente, amparada por un certificado digital vigente a la fecha de su elaboración, y es válido de conformidad con lo dispuesto en los artículos 7 y 9, fracción I, de la Ley de Firma Electrónica Avanzada y artículo 12 de su Reglamento.

Número de Registro: 03-2024-061409213000-01

Ciudad de México, a 14 de junio de 2024

EL DIRECTOR DEL REGISTRO PÚBLICO DEL DERECHO DE AUTOR

JESÚS PARETS GÓMEZ

PaMVMaOqF1s2fbaspQ7YBOaTa+PeGXPosL54RmMb+vamR/OkVmmGIjtY5S8fZvIFIJSZ4gdYx4/Ki0kvvEykI
8PylccC0Q31IJV/kitSpuyQs9HSySLE98QwzygQrAvoZuxB7NII8pb2SqEr2Pv+m4hPaD1Nsu5uduz0eu4VAiif13
D2MlrSMTrvGIFJtwCHKoQRJmBVQuC1zta5IIhszph20KL2+AEamH+ALEZa+XtbG6nnyCNTSd/Rq3TaL8hxSo
TRNxiUW2XqCIKvWZXwppH1X5i9dzVqfhSww5l61BBO9IPYYoHSxVJJsdL0av9XgaIISXdA6eMIlyZtBAA==

2/2

CULTURA | INDAUTOR

Referencias Bibliográficas

[1] L. Rouhiainen, *Inteligencia artificial: 101 cosas que debes saber hoy sobre nuestro futuro*. Alienta, 2018.

[2] S. J. (Stuart J. Russell, Peter. Norvig, J. Manuel. Corchado Rodríguez, and Luis. Joyanes Aguilar, *Inteligencia artificial: un enfoque moderno*. Pearson Prentice Hall, 2004.

[3] National Geographic, "5 usos cotidianos de la inteligencia artificial que la gente no se da cuenta," National Geographic. Accessed: Apr. 03, 2023. [Online]. Available: https://www.nationalgeographicla.com/ciencia/2023/02/5-usos-cotidianos-de-la-inteligencia-artificial-que-la-gente-no-se-da-cuenta

[4] Oracle, "Do you know what deep learning is?," Oracle. Accessed: Apr. 03, 2023. [Online]. Available: https://www.oracle.com/artificial-intelligence/machine-learning/what-is-deep-learning/#deep-learning-defined

[5] J. Rogel-Salazar, "Transformers - Self-Attention to the rescue." Accessed: Apr. 09, 2023. [Online]. Available: https://www.dominodatalab.com/blog/transformers-self-attention-to-the-rescue

[6] A. Vaswani *et al.*, "Attention Is All You Need," Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.03762

[7] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.11929

[8] A. Gheorghe, L. Mahdi, and O. Musat, "AGE-RELATED MACULAR DEGENERATION," *Rom J Ophthalmol*, vol. 59, no. 2, pp. 74–77, 2015.

[9] K. L. Ávila Heras, Y. K. Carrillo Mora, S. N. Cely Jadan, and M. Arcos, "Revisión Bibliográfica: Degeneración Macular Relacionada con la Edad. Prevención y Tratamiento Temprano," *Revista Médica del Hospital José Carrasco Arteaga*, vol. 10, no. 2, pp. 145–149, Jul. 2018, doi: 10.14410/2018.10.2.rb.32.

[10] J. Z. Nowak, "Age-related macular degeneration (AMD): pathogenesis and therapy," 2006.

[11] World Health Organization, "World report on vision," 2019. Accessed: Mar. 01, 2023. [Online]. Available: https://apps.who.int/iris/handle/10665/328717

[12] C. J. Thomas, R. G. Mirza, and M. K. Gill, "Age-Related Macular Degeneration," *Medical Clinics of North America*, vol. 105, no. 3. W.B. Saunders, pp. 473–491, May 01, 2021. doi: 10.1016/j.mcna.2021.01.003.

[13] B. L. Duff, "Facts about the macula of the eye," All About Vision. Accessed: Apr. 04, 2023. [Online]. Available: https://www.allaboutvision.com/resources/macula/

[14] Oftalvist, "Fóvea: definición, función y estructura," Oftalvist. Accessed: Apr. 12, 2023. [Online]. Available: https://www.oftalvist.es/blog/fovea-definicion-funcion-y-estructura

[15] Mácula-Retina, "¿Qué es un escotoma?," Glosario. Accessed: Apr. 12, 2023. [Online]. Available: https://www.macula-retina.es/que-es-un-escotoma/

[16] T. E. de Carlo, A. Romano, N. K. Waheed, and J. S. Duker, "A review of optical coherence tomography angiography (OCTA)," *International Journal of Retina and Vitreous*, vol. 1, no. 1. BioMed Central Ltd., Jul. 24, 2015. doi: 10.1186/s40942-015-0005-8.

[17] D. Turbert, "What Is Optical Coherence Tomography?" Accessed: Apr. 09, 2023. [Online]. Available: https://www.aao.org/eye-health/treatments/what-is-optical-coherence-tomography

[18] M. Ángel Teus, E. Arranz-márquez, and L. Y. López-guajardo Rafael Jiménez-parras, "Fondo de ojo," 2007.

[19] J. Tang, M. Sharma, and R. Zhang, "Explaining the Effect of Data Augmentation on Image Classification Tasks," 2022.

[20] IBM, "What is overfitting?," IBM. Accessed: Apr. 09, 2023. [Online]. Available: https://www.ibm.com/topics/overfitting#What%20is%20overfitting?

[21] O. Elisha, "Overcoming overfitting in image classification using data augmentation," Heartbeat. Accessed: Apr. 09, 2023. [Online]. Available: https://heartbeat.comet.ml/overcoming-overfitting-in-image-classification-using-data-augmentation-9858c5cee986

[22] Secretaría de Salud, “En México, más de 30 millones de personas padecen hipertensión arterial: Secretaría de Salud.”
[23] J. Ángel Rivera Dommarco *et al.*, “La Obesidad en México,” 2018.
[24] J. H. Tan *et al.*, “Age-related Macular Degeneration detection using deep convolutional neural network,” *Future Generation Computer Systems*, vol. 87, pp. 127–135, Oct. 2018, doi: 10.1016/j.future.2018.05.001.
[25] M. Treder, J. L. Lauermann, and N. Eter, “Automated detection of exudative age-related macular degeneration in spectral domain optical coherence tomography using deep learning,” *Graefe’s Archive for Clinical and Experimental Ophthalmology*, vol. 256, no. 2, pp. 259–265, Feb. 2018, doi: 10.1007/s00417-017-3850-3.
[26] Z. Gu, S. Jiang, J. Lee, J. Xie, J. Cheng, and J. Liu, “Automatic localization of optic disc using modified U-Net,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, May 2018, pp. 79–83. doi: 10.1145/3232651.3232671.
[27] L. Wang, Y. Huang, B. Lin, W. Wu, H. Chen, and J. Pu, “Automatic classification of exudates in color fundus images using an augmented deep learning procedure,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Aug. 2019, pp. 31–35. doi: 10.1145/3364836.3364843.
[28] C. González-Gonzalo *et al.*, “Evaluation of a deep learning system for the joint automated detection of diabetic retinopathy and age-related macular degeneration,” *Acta Ophthalmol*, 2019, doi: 10.1111/aos.14306.
[29] J. C. de Goma, O. J. D. Binsol, A. M. T. Nadado, and J. P. A. Casela, “Age-related macular degeneration detection through fundus image analysis using image processing techniques,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Dec. 2019, pp. 146–150. doi: 10.1145/3374549.3374577.
[30] Z. Chen, D. Li, H. Shen, Y. Mo, H. Wei, and P. Ouyang, “Automated retinal layer segmentation in OCT images of age-related macular degeneration,” *IET Image Process*, vol. 13, no. 11, pp. 1824–1834, Sep. 2019, doi: 10.1049/iet-ipr.2018.5304.
[31] V. Mihalov, D. Andreev, and M. Lazarova, “Software Platform for Retinal Disease Diagnosis Through Deep Convolutional Neural Networks,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jun. 2020, pp. 61–65. doi: 10.1145/3407982.3408011.
[32] K. Alsaih, M. Z. Yusoff, T. B. Tang, I. Faye, and F. Mériaudeau, “Deep learning architectures analysis for age-related macular degeneration segmentation on optical coherence tomography scans,” *Comput Methods Programs Biomed*, vol. 195, Oct. 2020, doi: 10.1016/j.cmpb.2020.105566.
[33] Z. Wang, S. P. Tian, X. Fu, and J. Z. He, “An effective image enhancement method for color fundus images,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Dec. 2021. doi: 10.1145/3508546.3508589.
[34] Y. M. Chen, W. T. Huang, W. H. Ho, and J. T. Tsai, “Classification of age-related macular degeneration using convolutional-neural-network-based transfer learning,” *BMC Bioinformatics*, vol. 22, Nov. 2021, doi: 10.1186/s12859-021-04001-1.
[35] L. R. Ashok and K. G. Sreeni, “Abnormality detection and classification of macular diseases from optical coherence tomography images: Using feature space comparison,” in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Nov. 2021. doi: 10.1145/3490035.3490265.
[36] H. T. Ahmed, Q. Zhang, R. Donnan, and A. Alomainy, “Framework of Unsupervised based Denoising for Optical Coherence Tomography,” in *2022 7th International Conference on Biomedical Signal and Image Processing (ICBIP)*, New York, NY, USA: ACM, Aug. 2022, pp. 19–24. doi: 10.1145/3563737.3563741.
[37] Mrs. V. Shelke, Mr. V. M. Shah, Mr. H. Ratnani, and Mr. R. Despande, “Diabetic Retinopathy Detection Using SVM,” *Int J Res Appl Sci Eng Technol*, vol. 10, no. 4, pp. 868–875, Apr. 2022, doi: 10.22214/ijraset.2022.41275.
[38] M. Seçkin Ayhan, H. Faber, L. Kühlewein, W. Inhoffen, F. Ziemssen, and P. Berens, “Multi-task learning for activity detection in neovascular age-related macular degeneration,” *medRxiv*, 2022, doi: 10.1101/2022.06.13.22276315.

[39] R. Chakraborty and A. Pramanik, “DCNN-based prediction model for detection of age-related macular degeneration from color fundus images,” *Med Biol Eng Comput*, vol. 60, no. 5, pp. 1431–1448, May 2022, doi: 10.1007/s11517-022-02542-y.
[40] M. Z. Atwany, A. H. Sahyoun, and M. Yaqub, “Deep Learning Techniques for Diabetic Retinopathy Classification: A Survey,” *IEEE Access*, vol. 10, pp. 28642–28655, 2022, doi: 10.1109/ACCESS.2022.3157632.
[41] R. Sun, Y. Li, T. Zhang, Z. Mao, F. Wu, and Y. Zhang, “Lesion-Aware Transformers for Diabetic Retinopathy Grading,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
[42] Y. Kihara *et al.*, “Detection of Nonexudative Macular Neovascularization on Structural OCT Images Using Vision Transformers,” *Ophthalmology Science*, vol. 2, no. 4, p. 100197, Dec. 2022, doi: 10.1016/j.xops.2022.100197.
[43] Z. Jiang *et al.*, “Computer-aided diagnosis of retinopathy based on vision transformer,” *J Innov Opt Health Sci*, vol. 15, no. 2, Mar. 2022, doi: 10.1142/S1793545822500092.
[44] C. Domínguez, J. Heras, E. Mata, V. Pascual, D. Royo, and M. Á. Zapata, “Binary and multi-class automated detection of age-related macular degeneration using convolutional- and transformer-based architectures,” *Comput Methods Programs Biomed*, vol. 229, Feb. 2023, doi: 10.1016/j.cmpb.2022.107302.
[45] Z. Gu, Y. Li, Z. Wang, J. Kan, J. Shu, and Q. Wang, “Classification of Diabetic Retinopathy Severity in Fundus Images Using the Vision Transformer and Residual Attention,” *Comput Intell Neurosci*, vol. 2023, pp. 1–12, Jan. 2023, doi: 10.1155/2023/1305583.
[46] Broad (Baidu Research Open-Access Dataset), “Refuge - Grand Challenge,” Grand Challenge. Accessed: Aug. 31, 2023. [Online]. Available: https://refuge.grand-challenge.org/iChallenge-AMD/
[47] Rakhshanda Mujib, “ARMD curated dataset 2023,” Kaggle. Accessed: Aug. 31, 2023. [Online]. Available: https://www.kaggle.com/datasets/rakhshandamujib/armd-curated-dataset-2023
[48] Google, “google/vit-base-patch16-224 · Hugging Face,” Hugging face. Accessed: Sep. 24, 2023. [Online]. Available: https://huggingface.co/google/vit-base-patch16-224