



**UNIVERSIDAD NACIONAL DE INGENIERÍA
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN**

**TRABAJO MONOGRAFICO PARA OPTAR AL TÍTULO DE INGENIERO EN
COMPUTACIÓN**

Tema:

**IMPLEMENTACIÓN DE UN MODELO DE RED NEURONAL CONVOLUCIONAL
PARA EL RECONOCIMIENTO DE ENFERMEDADES DE LA PIEL.**

Autor(es):

BR. Bernard Antonio Obando Mena

BR. Venus Raquel García Benavidez

Tutor:

MSc. Ing. Adilson Isaac González López

Noviembre, 2023

Managua, Nicaragua

DEDICATORIA

BERNARD ANTONIO OBANDO MENA

A mi familia, quienes han sido de gran apoyo constante en mi vida. Su amor incondicional y sacrificio han sido la fuerza impulsora detrás de cada paso que he dado. Este logro es tan suyo como mío.

VENUS RAQUEL GARCIA BENAVIDEZ

Este trabajo monográfico se dedica con profundo agradecimiento a aquellos que han sido pilares fundamentales en mi camino académico.

En primer lugar, quiero dedicar este logro a Dios, quien fue una fuente de sabiduría y guía constante, cuya presencia ha sido invaluable durante todo el proceso de desarrollo de este trabajo y quien me ha ayudado de gran manera en cada paso que he dado.

A mis padres Marcio Alfredo Garcia Diaz y Silvia Yesenia Benavidez Mairena, así como también a mí tía Wanda Azucena Cuba Mairena, quienes me han brindado su amor incondicional, apoyo constante y sacrificio, siendo la base sólida que me ha permitido alcanzar este logro académico.

A nuestro tutor el Msc. Ing. Adilson Isaac González López, cuyo compromiso y orientación experta han sido esenciales en el desarrollo y finalización de esta tesis.

Asimismo, le dedico este logro a nuestra alma mater UNI, institución educativa que ha brindado las herramientas y oportunidades necesarias para nuestra formación académica.

AGRADECIMIENTO

BERNARD ANTONIO OBANDO MENA

Quiero expresar mi profundo agradecimiento a todas las personas que han contribuido de manera significativa a la realización de este trabajo monográfico. En primer lugar, agradezco a nuestro tutor por su guía, valiosos comentarios y paciencia durante todo el proceso de investigación. También quiero reconocer el gran trabajo realizado por mi compañera a lo largo de todo el proyecto. Además, agradezco a mis familiares por su constante apoyo y comprensión. Este proyecto no hubiera sido posible sin el respaldo de cada uno de ustedes y aprecio profundamente la generosidad que me han demostrado.

VENUS RAQUEL GARCIA BENAVIDEZ

Agradezco primeramente a Dios, padre celestial, rey de reyes y señor de señores, por darnos la fuerza, el conocimiento y la motivación suficiente para finalizar esta etapa de nuestra vida.

A mi familia, especialmente mis padres quienes se han esforzado por brindarme educación y me han acompañado y motivado en noches de desvelos apoyándome de manera extraordinaria en todo el camino recorrido académicamente.

A nuestro tutor, el Msc. Ing. Adilson Isaac González López, por su excelente seguimiento en todo el proceso de desarrollo y finalización de este trabajo monográfico, así como también a mi compañero que me ha acompañado a lo largo de este arduo trabajo monográfico.

Finalmente, le decimos gracias a nuestra alma mater UNI, por su excelente docencia y plan de estudio, sin los cuales no habríamos podido llegar a ser lo que somos.

Resumen

Este documento presenta un proyecto monográfico centrado en el uso de una Red Neuronal Convolutacional (CNN) para identificar enfermedades de la piel. El proyecto siguió la metodología CRISP-DM y conllevó la recopilación de datos (Imágenes de enfermedades de piel), el análisis y procesamiento de estos datos, seguido del diseño y entrenamiento de la CNN para lograr un modelo preciso. Este modelo se integró en una aplicación móvil denominada DermaDI, desarrollada con el propósito de integrar el modelo y brindar accesibilidad al mismo, permitiendo a los usuarios cargar imágenes de lesiones cutáneas para recibir diagnósticos precisos. Se presentó DermaDI en el Hospital Central Dermatológico Nacional para evaluar la recepción de la herramienta, y se recopilaron comentarios para futuras mejoras. El modelo CNN demostró alta precisión en la identificación de las enfermedades dermatológicas contenidas. Este documento de trabajo monográfico proporciona detalles completos sobre el desarrollo de cada fase del proyecto.

Índice de Contenido

1.	Introducción.....	1
2.	Antecedentes.....	2
3.	Objetivos.....	5
3.1.	Objetivo General.....	5
3.2.	Objetivos Específicos.....	5
4.	Justificación	6
5.	Marco Teórico	7
5.1.	Inteligencia Artificial (IA)	7
5.2.	Machine Learning (ML)	7
5.3	Técnicas del machine learning.....	8
5.3.1.	Aprendizaje Supervisado	8
5.3.1.1.	Técnicas de Clasificación.....	9
5.4.	Deep Learning.....	9
5.5.	Redes neuronales artificiales.....	10
5.5.1.	Redes neuronales convolucionales (CNN)	10
5.5.1.1.	Entrenamiento de una red neuronal convolucional (CNN)	11
5.6.	Herramientas utilizadas.....	15
5.6.1.	Google Colaboratory.....	15
5.6.2.	TensorFlow	15
5.6.3.	Keras	15
5.6.4.	Python	16
5.6.5.	Google Drive.....	16
5.6.6.	Android Studio.....	16
5.6.7.	Flutter	16
5.6.8	Dart	17
5.6.9.	Gestor de base de datos SQLite	17
6.	Análisis y Presentación de Resultados	18
6.1.	Modelo CRISP-DM	18
6.2.	Fases del modelo CRISP-DM	18
6.2.1.	Business Understanding (Comprensión del negocio)	19
6.2.1.1.	Determinar los objetivos del negocio.....	19
6.2.1.2.	Evaluación de la situación.....	20

6.2.1.3. Estudio de Factibilidad.....	22
6.2.1.4. Análisis y Selección de Alternativa	30
6.2.1.5. Factibilidad Operativa del proyecto.....	31
6.2.1.6. Objetivos de la minería de datos	32
6.2.1.7. Producir el plan del proyecto	34
6.2.2. Data Understanding (Comprensión de los datos).....	37
6.2.2.1. Recolectar los datos iniciales	38
6.2.2.2. Descripción y exploración de los datos.....	39
6.2.2.3. Verificar la calidad de los datos.....	40
6.2.3. Data Preparation (Preparación de los datos)	40
6.2.3.1. Seleccionar los datos.....	41
6.2.3.2. Limpiar los datos.....	42
6.2.3.3. Construir los datos	43
6.2.4. Modeling (Modelado)	43
6.2.4.1. Selección de la técnica de modelado.....	44
6.2.4.2. Generar el plan de Prueba.....	44
6.2.4.3. Construcción del modelo	45
6.2.4.4. Evaluar el modelo	53
6.2.5. Evaluation (Evaluación de los resultados)	66
6.2.5.1. Evaluar los resultados	67
6.2.5.2. Revisar el proceso	68
6.2.5.3. Determinar los próximos pasos.....	71
6.2.6. Deployment (Despliegue. Puesta en producción)	72
6.2.6.1. Preparación del entorno de producción.....	73
6.2.6.2. Desarrollo y configuración de la solución	74
7. Conclusión	90
8. Recomendaciones	91
9. Bibliografía	92
10. Anexos	97
10.1 Cronograma de actividades del proyecto	97
10.2. Diseño del Formulario de recepción y usabilidad de la app móvil DermaDI	98

Índice de Tablas

Tabla 1. Alternativa 1. Recursos de Hardware	23
Tabla 2. Alternativa 1. Recursos de Software	24
Tabla 3. Alternativa 1. Recursos Humanos	24
Tabla 4. ALternativa 1. Factibilidad Economica	25
Tabla 5. Alternativa 2.Recursos de Hardware	27
Tabla 6. Alternativa 2. Recursos de Software	27
Tabla 7. Alternativa 2. Recursos Humanos	28
Tabla 8. Alternativa 2. Factibilidad Economica	29

Índice de Figuras

Figura 1. Tècniques del Machine Learning	8
Figura 2. Matriz de píxeles.....	12
Figura 3. Conjunto de Kernels	13
Figura 4. Fases del Modelo CRISP-DM.....	18
Figura 5. Importaciòn de librerias	46
Figura 6. Arquitectura de red del modelo CNN.....	51
Figura 7. Cálculo y Resultados de métricas de evaluación global del modelo CNN.	55
Figura 8. Código para el cálculo de métricas de evaluación por clase.....	57
Figura 9. Resultados de cálculo de métricas de evaluación por clase	60
Figura 10. Resultados de evaluación de métricas de dominio	66
Figura 11. Diseño logo DermaDI	73
Figura 12. Pantalla principal de la app movil DermaDI.....	75
Figura 13. Pantalla de confirmación de imagen a clasificar	76
Figura 14. Pantalla de resultado de la clasificación de la imagen	77
Figura 15. Pantalla de información sobre la enfermedad identificada.....	78
Figura 16. Pantalla de información de la app y metodos de aporte al proyecto	79
Figura 17. Pantalla de galeria de enfermedades contenidas en el modelo.....	80
Figura 18. Gráfica estimado de casos atendidos por dia	82
Figura 19. Gráfica de enfermedades mas comunes atendidas en el HCND	83
Figura 20. Valoración de la estetica de DermaDI	84
Figura 21. Valoración de la usabilidad de DermaDI.....	85
Figura 22. Valoración del contenido de DermaDI	86
Figura 23. Valoración de la funcionalidad de DermaDI	87
Figura 24. Valoración de la satisfacciòn del usuario con DermaDI.....	88

1. Introducción

Este documento presenta la implementación de un modelo de machine learning para el reconocimiento de ocho tipos de enfermedades de piel.

Se construyó un modelo de red neuronal convolucional (CNN) con tecnologías de machine learning y usando la librería TensorFlow para generar un modelo eficiente de predicción. El modelo está entrenado para clasificar ocho tipos de enfermedades dermatológicas a partir de un conjunto de datos correspondiente a imágenes de enfermedades de piel.

Este trabajo monográfico considera la metodología CRISP-DM orientada a la ciencia de datos y aprendizaje automático, la cual incluye la fase de despliegue. Bajo esta fase, se desarrolló la aplicación móvil nombrada DermaDI, que integra el modelo de red neuronal construido.

Este documento se desarrolla bajo la siguiente estructura:

- **Antecedentes:** Describe la forma tradicional de la identificación de enfermedades de la piel, documentación relacionada al uso de la Inteligencia Artificial (IA) para el reconocimiento de enfermedades de la piel, además de la mención de aplicaciones móviles similares encontradas.
- **Justificación:** Expresa las razones para llevar a cabo la implementación del modelo de CNN.
- **Objetivos:** Presentan el resultado a alcanzar tomando en cuenta la metodología CRISP-DM.
- **Marco teórico:** Describe los conceptos que se abordan en el proceso de implementación de un modelo de Machine Learning.
- **Diseño metodológico:** Aborda la metodología a seguir para la construcción del modelo de CNN.
- **Conclusión:** Presenta los resultados finales en torno a la construcción y despliegue del modelo de CNN construido.
- **Recomendaciones:** Presenta propuestas de mejora para el trabajo monográfico realizado.

2. Antecedentes

Las enfermedades de la piel comprenden una amplia gama de patologías e incluyen enfermedades causadas por infecciones bacterianas, virales, por hongos, reacciones alérgicas, cánceres de piel, por cuestiones hereditarias o por causas desconocidas.

El diagnóstico de enfermedades de la piel se realiza tradicionalmente mediante la inspección visual y la observación continua. El dermatólogo puede utilizar herramientas como una lente de mano o un dermatoscopio¹ para inspeccionar las áreas de interés. Las características que ayudan a revelar la enfermedad incluyen el tamaño, la forma, el color y la ubicación de la anomalía, así como la presencia o ausencia de otros síntomas o signos. En su artículo “Diagnosis of skin disorder” (Benedetti, 2021) proporciona información sobre los métodos tradicionales de diagnóstico de enfermedades de piel y nombra algunas herramientas auxiliares utilizadas por los especialistas dermatólogos.

Las enfermedades de la piel son de las afecciones más frecuentes; su aspecto visual es más notable en comparación con otras enfermedades. Debido a esto y al interés en el tema, el número de estudios sobre el reconocimiento de enfermedades de la piel que utilizan métodos de aprendizaje profundo ha aumentado rápidamente desde 2016, según indica (Li, et. all, 2020) destacando la importancia de este campo en la actualidad.

El acceso a la atención dermatológica puede no ser tan accesible en ciertas regiones del país. En el proceso de triaje², que es realizado por médicos de atención primaria, puede no diagnosticarse y referenciar correctamente la afección del paciente, dado que no en todos los puestos de salud existe la permanencia de médicos especialistas dermatólogos.

Se realizó una búsqueda en los repositorios bibliográficos de tres universidades nacionales: UNI, UNAN y UNAN-León, para encontrar información relacionada con el tema del trabajo monográfico descrito. Sin embargo, solo se encontró una tesis titulada "

¹ Microscopio manual de superficie (aumento de 10x) que permite la visualización de estructuras pigmentadas de la epidermis y de la unión dermopigmentaria.

² Sistema de selección y clasificación de pacientes en los servicios de urgencia, basado en sus necesidades terapéuticas y los recursos disponibles.

Enfermedades dermatológicas en la población de 1 a 14 años que habitan en la comunidad rural de Haulover, Municipio de Laguna de Perlas, R.A.A.S en el año 2013" (García & Urbina, 2014). A pesar de que esta tesis trata sobre las enfermedades de la piel, no está relacionada con el reconocimiento mediante modelos de aprendizaje profundo. No obstante, se encontraron trabajos extranjeros relevantes, como “**A Smartphone-Based Skin Disease Classification Using MobileNet CNN**” (Velasco, et. all, 2019), un proyecto desarrollado por un grupo de estudiantes de la Technological University of the Philippines en 2019, que consiste en una aplicación móvil que permite predecir enfermedades de la piel usando un modelo entrenado con la arquitectura MobileNet V2. Otros trabajos a mencionar son: “**Automatic Skin disease diagnosis using deep learning from clinical image and patient information**” (K. A. Muhaba; K. Dese; T. M. Aga; F. T. Zewdu; G. L. Simegn, 2021) y “**Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM**” (Srinivasu, et. all, 2021), que consisten en artículos científicos que se centran en el uso de técnicas de aprendizaje profundo y proponen la implementación de modelos de clasificación para el reconocimiento de enfermedades de la piel. También, se encontraron herramientas tecnológicas para dispositivos móviles, como las apps Aysa y AI Dermatologist, que sirvieron como referencia para el diseño de la herramienta DermaDI desarrollada para la integración y despliegue del modelo de clasificación resultante de este proyecto.

Los trabajos mencionados tienen limitantes, como la falta de accesibilidad a los modelos de clasificación resultantes de su implementación y que toda su documentación se encuentra solamente en idioma inglés. En el caso de la aplicación móvil Aysa, requiere una conexión permanente a internet, mientras que AI Dermatologist requiere pagos después de un diagnóstico gratuito, además de solicitar información adicional, como el género, color de piel, ojos y cabello. Cabe mencionar que ambas apps solo están disponibles en inglés.

Se buscaron trabajos que tuvieran relación con el tema presentado en este documento que hubieran sido realizados bajo la metodología CRISP-DM. Se encontraron dos trabajos de titulación de fin de carrera, “**Aplicativo web para la identificación de deficiencia de zinc en naranjos mediante redes neuronales convolucionales**” (Farfan Vergara, 2021) y

"Aplicación móvil para clasificación automática de malas hierbas en imágenes usando aprendizaje profundo" (Casa Lema & Tumbaco Sango, 2020). Estos trabajos sirvieron de consulta y guía para la implementación de la metodología CRISP-DM en el presente trabajo monográfico.

3. Objetivos

3.1. Objetivo General

- Implementar un modelo de red neuronal convolucional para el reconocimiento de enfermedades de la piel.

3.2. Objetivos Específicos

- Preparar un set de datos con imágenes de enfermedades de piel, aplicando técnicas de preprocesamiento de imágenes, considerando la metodología CRISP-DM.
- Construir el modelo de red neuronal convolucional a partir del conjunto de imágenes trabajado.
- Evaluar el modelo para garantizar el correcto funcionamiento en el reconocimiento de enfermedades de la piel.
- Desarrollar un aplicativo móvil para la integración y despliegue del modelo generado.

4. Justificación

La incorporación de sistemas de Inteligencia Artificial para identificar imágenes ha mejorado la captura y procesamiento de datos. El reconocimiento de imágenes utilizando algoritmos proporciona una tasa de acierto de hasta un 98% (Vincle Redaccion, 2021).

La implementación de modelos de redes neuronales convolucionales (CNN) es la base de este proyecto, ya que estas redes han demostrado ser altamente eficaces en la identificación y clasificación de patrones visuales complejos en imágenes. Por lo que, las CNN pueden lograr altos niveles de precisión en la clasificación de enfermedades de la piel.

El uso de redes neuronales convolucionales representa una oportunidad para explorar y aprovechar al máximo la tecnología y su impacto positivo en la atención médica y el alcance a los pacientes. Con la implementación del modelo para el reconocimiento de enfermedades de la piel descrito en este documento y su integración en una aplicación móvil (DermaDI), se pone a disposición una herramienta de apoyo a los médicos de atención primaria que no son especialistas en dermatología.

La aplicación móvil DermaDI, que integra el modelo de CNN para el reconocimiento de enfermedades de la piel; proporciona Información de conocimiento general sobre las enfermedades contenidas en el modelo de clasificación, cuidando de omitir Información alguna sobre tratamiento o medicación.

Este proyecto podrá utilizarse como referencia documentaria para posteriores trabajos académicos que consideren las tecnologías de Machine Learning y la técnica de redes neuronales convolucionales. Tanto el paquete de instalación para la aplicación DermaDI en su versión de prototipado utilizada en este proyecto, como toda la documentación construida y referencial trabajada en el desarrollo de este proyecto monográfico están disponibles en un repositorio de libre acceso en GitHub.

5. Marco Teórico

En este apartado se definen los conceptos generales y técnicos que son base en el desarrollo de este proyecto.

5.1. Inteligencia Artificial (IA)

La inteligencia artificial es una rama de la informática que busca imitar la inteligencia humana mediante la programación de algoritmos en un entorno computacional. Para lograr esto, se requieren tres componentes clave: sistemas computacionales potentes, gestión eficiente de los datos y algoritmos avanzados de inteligencia artificial. Cuanto más parecido sea el comportamiento deseado al de los humanos, más datos y potencia de procesamiento serán necesarios (NetApp, s.f.).

La referencia a NetApp subraya la importancia de la potencia de procesamiento y la cantidad de datos necesarios para lograr un comportamiento similar al humano en aplicaciones de IA.

5.2. Machine Learning (ML)

El Machine Learning (ML) es una rama de la IA que permite a las computadoras aprender de forma autónoma a partir de los datos y la experiencia. Los algoritmos de ML usan métodos computacionales para analizar grandes cantidades de datos y encontrar patrones naturales en ellos, lo que les permite mejorar su rendimiento en tareas específicas, tomar mejores decisiones y realizar predicciones precisas. Con el tiempo, la computadora puede continuar mejorando su aprendizaje y rendimiento gracias a la disponibilidad de más datos para el aprendizaje (Itelligent, 2020).

El concepto de Machine Learning obtenido de (Itelligent, 2020). Destaca la autonomía del ML para aprender de datos y experiencia, así como su capacidad para mejorar el rendimiento en tareas y la toma de decisiones precisas a través del análisis de datos.

5.3 Técnicas del machine learning

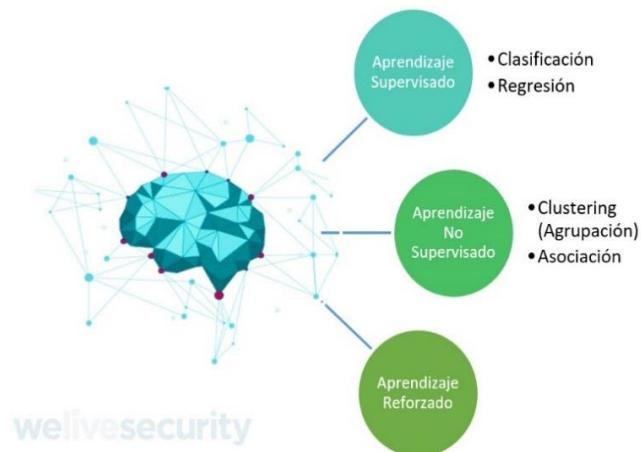
El Machine Learning se basa en diversas técnicas de aprendizaje, incluyendo:

- **Aprendizaje Supervisado:** Con esta técnica el modelo se entrena con datos etiquetados, donde las entradas se asocian con salidas conocidas.
- Aprendizaje No Supervisado: Con esta técnica el modelo se entrena con datos no etiquetados y busca patrones o estructuras ocultas en los datos.
- Aprendizaje Semi-Supervisado: Con esta técnica se combinan datos etiquetados y no etiquetados para entrenar el modelo, útil si es difícil obtener datos etiquetados.
- Aprendizaje por Refuerzo: Con esta técnica el modelo aprende tomando decisiones secuenciales en un entorno y recibe recompensas o penalizaciones en función de sus acciones, lo que permite aprender a través de la retroalimentación del entorno.

En la figura 1 se presenta una representación gráfica de las técnicas de aprendizaje más utilizadas en ML, junto con las técnicas específicas asociadas a cada tipo de aprendizaje.

Figura 1

Técnicas del Machine Learning



Nota: Tomada de: (Gonzalez, 2021)

5.3.1. Aprendizaje Supervisado

El aprendizaje automático supervisado construye un modelo que puede hacer predicciones basadas en pruebas, aunque con un cierto grado de incertidumbre. Para lograr esto, un algoritmo de aprendizaje supervisado se basa en un conjunto conocido de datos de entrada

y sus respuestas correspondientes para entrenar un modelo con el objetivo de generar predicciones precisas ante nuevos datos (MathWorks, s.f.). Es decir, según el aprendizaje automático supervisado se centra en la construcción de modelos que pueden realizar predicciones basadas en datos de prueba, considerando un nivel de incertidumbre.

Este trabajo monográfico se desarrolla bajo la técnica de aprendizaje supervisado, ya que se dispone de un conjunto de datos clasificado y categorizado para entrenar la red neuronal y generar un modelo que pueda predecir con una elevada tasa de precisión.

5.3.1.1. Técnicas de Clasificación

La clasificación se utiliza cuando los datos pueden ser etiquetados, categorizados o separados en grupos específicos. Por ejemplo, las aplicaciones de reconocimiento de escritura usan clasificación para identificar letras y números (MathWorks, s.f.).

En el aprendizaje automático existen diferentes métodos que se pueden aplicar a los problemas de clasificación. Entre ellos, se puede mencionar las máquinas de vectores de soporte, los árboles de decisión, los bosques aleatorios, las redes neuronales y el aprendizaje profundo (Heras, 2020).

Este proyecto se construyó considerando la técnica de clasificación de redes neuronales convolucionales (CNN) dado que esta técnica ha demostrado ser altamente efectiva en la resolución de problemas de clasificación complejos y se considera una de las más avanzadas y eficaces en el campo de la IA y el aprendizaje automático.

5.4. Deep Learning

El Deep Learning es una rama del ML que se enfoca en desarrollar sistemas capaces de aprender de manera autónoma mediante la identificación de patrones y la ejecución de tareas similares a las de los humanos. Para ello, se emplean redes neuronales, que son un tipo de algoritmo específico que cuenta con varias capas y que se basa en la imitación de la función de las redes neuronales del cerebro humano. Estos algoritmos son utilizados en

aplicaciones como el reconocimiento de voz, la detección de objetos y la identificación de imágenes (Martinez, 2020).

5.5. Redes neuronales artificiales

UNIR, a través de un artículo en su revista digital publicado en 2021, compara la estructura y el funcionamiento de un modelo de red neuronal artificial, con el proceso de las neuronas en el cerebro humano. Explicado a continuación.

Las redes neuronales artificiales son un modelo computacional que permite a las máquinas simular el comportamiento del cerebro humano, es decir, dotar a las máquinas de la capacidad de aprender de una manera similar a como lo hace nuestro cerebro.

Un modelo de red neuronal artificial se compone de unidades artificiales llamadas "neuronas", que reciben información del exterior o de otras neuronas al igual que las neuronas en el cerebro humano reciben impulsos nerviosos. Estas neuronas artificiales procesan la información y generan un valor de salida que alimenta a otras neuronas en la red o actúa como la respuesta externa del modelo (UNIR Revista, 2021).

5.5.1. Redes neuronales convolucionales (CNN)

Las CNN son un tipo de red neuronal artificial que funciona de manera similar a las neuronas en la corteza visual primaria de un cerebro humano. Este tipo de red puede ser utilizada para operaciones con matrices bidimensionales, lo que la hace muy efectiva en tareas de visión artificial, como la clasificación y segmentación de imágenes (Barrios, s.f.).

Las CNN aprenden a través de diferentes niveles de abstracción, cada capa se concentra en diferentes características de la imagen, como formas, colores y bordes. Para esto, utilizan filtros que escanean la imagen, definen y clasifican los objetos. La combinación de estos permite a la red identificar objetos complejos en la imagen (Redaccion KeepCoding, 2022).

Consolidando los conceptos previos obtenidos de (Barrios, s.f.) y (Redacción KeepCoding, 2022), se puede redefinir una CNN como un tipo de neurona artificial que intenta imitar el comportamiento de las neuronas del cerebro humano a través de cálculos matemáticos que le permiten realizar funciones que le dan la capacidad de aprender y combinar características de los datos para así identificarlos y clasificarlos.

5.5.1.1. Entrenamiento de una red neuronal convolucional (CNN)

Las Redes Neuronales Convolucionales (CNN) tienen la capacidad de identificar objetos en imágenes después de un proceso de entrenamiento con una gran cantidad de ejemplos. La red neuronal se alimenta con un gran conjunto de datos de imágenes etiquetadas, cada una con su correspondiente etiqueta de clase. La CNN procesa cada imagen asignando valores aleatorios y luego realiza comparaciones con la etiqueta de clase de la imagen de entrada. Durante el entrenamiento, las neuronas de la red aprenden a reconocer las características únicas de cada objeto y a generalizar ese conocimiento para inferir imágenes desconocidas. La capacidad de la CNN para reconocer objetos y su capacidad de inferencia se derivan de su habilidad para capturar patrones en los datos de entrada. (Barrios, s.f.). A continuación, se definen conceptos ligados a redes neuronales convolucionales y el entrenamiento de este tipo de redes.

5.5.1.1.1 Píxeles

El píxel (px) es el elemento más pequeño de una imagen reproducida digitalmente. En un monitor o en la pantalla de un teléfono móvil, suelen alinear varios píxeles en una trama. La combinación de varios píxeles constituye una imagen rasterizada (Ionos digital guide, 2021).

Al ampliar una imagen a través de un zoom, es posible observar los píxeles que permiten la creación de la misma. Ante la vista se presentan como pequeños cuadrados o rectángulos en blanco, negro o matices de gris (Porto & Merino, 2009).

En la figura 2 se puede visualizar una matriz de píxeles para imágenes en blanco y negro, o bien, imágenes a color las cuales se componen de los 3 canales RGB.

Figura 2

Matriz de píxeles



Nota: Tomada de: (Barrios, s.f.)

5.5.1.1.2. Neuronas

En términos de IA, la neurona de McCulloch-Pitts es una unidad de procesamiento que busca imitar el comportamiento de las neuronas naturales presentes en el cerebro humano. Su combinación y conexión forman una red neuronal artificial. Esta unidad fundamental de procesamiento, también conocida como nodo o unidad, recibe información de otras unidades o de una fuente externa (Ramis, 1943).

La neurona artificial, tiene entradas, pesos asociados a esas entradas, realiza una suma ponderada de estas entradas, y luego aplica una función de activación para producir una salida. La combinación de múltiples neuronas en capas forma una red neuronal.

5.5.1.1.3. Preprocesamiento de datos

El preprocesamiento de datos es una etapa crucial que debe llevarse a cabo antes de cualquier manipulación o transformación de los datos. A menudo, esta etapa se realiza en más de una ocasión, no solo al comienzo del proyecto. Las actividades que se realizan en esta etapa incluyen la recolección de datos de diversas fuentes y la manipulación de las cabeceras y filas de datos para obtener un conjunto de datos de mayor calidad y homogeneidad (Stradata aml, s.f.)

5.5.1.1.4. Convoluciones

“Las redes neuronales convolucionales se han hecho famosas gracias a su capacidad para detectar patrones que después clasifican. Pues bien, esos detectores de patrones son las convoluciones” (Redaccion KeepCoding, 2022).

Las convoluciones son una operación matemática que se utiliza en procesamiento de imágenes y en aprendizaje automático para analizar patrones en datos. En otras palabras, una convolución implica tomar un conjunto de datos (como una imagen o un audio) y aplicar un filtro (una pequeña matriz de números) para producir un nuevo conjunto de datos que resalte ciertas características en los datos originales. En aprendizaje automático, las convoluciones se utilizan en las CNN para procesar datos de entrada y extraer características útiles (Barrios, s.f.). En síntesis, una convolución implica aplicar un filtro a un conjunto de datos que permita resaltar características específicas en dichos datos.

5.5.1.1.5. Filtro: conjunto de kernels

Los filtros convolucionales son un conjunto de kernels que se encuentran en cada capa convolucional de una red. Cada filtro está compuesto por diferentes kernels que extraen características específicas de una imagen. Esto significa que la salida de una capa en particular se compone de diferentes imágenes que han sido filtradas de la imagen de entrada, lo que se conoce como mapas de características. (Ibarrola, 2022).

Figura 3

Conjunto de Kernels

	0,6	0,6		
0,6			0,6	
0,6	0,6	0,6	0,6	
0,6			0,6	

-1,2	-0,6	0,6	1,2
-1,2	0,6	-0,6	1,2
-1,2	1,2	-1,2	1,2
-0,6	1,2	-1,2	0,6

Nota. Se muestra un ejemplo del kernel realizando el producto matricial con la imagen de entrada y desplazando de a un píxel de izquierda a derecha y de arriba-abajo, generando así una nueva matriz que compone al mapa de características. Tomada de: (Barrios, s.f.)

5.5.1.1.6. Función de activación: ReLU

“La función de activación más utilizada para CNN es la llamada ReLU por Rectifier Linear Unit y consiste en una función $f(x) = \max(0, x)$ ”.

En el contexto de las redes neuronales, la función de activación del rectificador o ReLU es una función de activación definida como la parte positiva de su argumento; Donde x es la entrada a una neurona $f(x) = x^+ = \max(0, x)$. ReLU establece todos los valores negativos en la matriz x a cero y todos los demás valores se mantienen constantes, se calcula después de la convolución y es una función de activación no lineal (Barrios, s.f.).

5.5.1.1.7. Muestreo: Max-Pooling

El pooling es una técnica utilizada en el análisis de imágenes para extraer información relevante de la imagen mediante la división de la misma en regiones. El método de max-pooling es una variante que reduce la cantidad de datos que deben ser procesados en cada capa, preservando la información más importante. Para aplicar el max-pooling, se divide la imagen en regiones y se selecciona el valor máximo en cada una de ellas. Esto permite obtener una representación más compacta y relevante de la imagen, lo que mejora la eficiencia del proceso de análisis. (Sotaquirá, 2019).

5.5.1.1.8. Aplanamiento

El aplanamiento se refiere al proceso de convertir una matriz o tensor multidimensional en un vector unidimensional. Este proceso es comúnmente utilizado para conectar una capa de neuronas completamente conectada a la salida de una capa convolucional o de agrupamiento en una red neuronal convolucional. Al aplanar los datos, se pierde la estructura espacial original de los datos y se obtiene un vector de características que se pueden utilizar como entrada para una capa completamente conectada.

“El proceso de aplanamiento simplemente significa reordenar el Mapa de características agrupadas en una sola columna. Hacemos eso, porque este vector ahora está listo para usarse como entrada de una red neuronal artificial para su posterior procesamiento” (Perlato, s.f.).

5.5.1.1.9. Softmax

Después de agregar una nueva capa oculta "tradicional" (tras el aplanamiento) al modelo, se aplica la función conocida como Softmax. Esta función conecta la capa oculta con la capa de salida final, que debe tener un número de neuronas igual al número de clases que se están clasificando. Por ejemplo, si se están clasificando perros y gatos, se necesitan dos neuronas; si se está clasificando coches, aviones y barcos, se necesitan tres.

5.6. Herramientas utilizadas

En este apartado se presentan las herramientas utilizadas en el desarrollo del proyecto.

5.6.1. Google Colaboratory

Google Colaboratory o Colab es una solución creada por Google con el propósito de brindar a cualquier individuo, sin importar su ubicación, acceso gratuito a unidades de procesamiento gráfico (GPU) y unidades de procesamiento tensorial (TPU). Esta plataforma ha sido especialmente diseñada para facilitar el desarrollo de aplicaciones de inteligencia artificial y análisis de datos (Redaccion Tokio, 2022).

5.6.2. TensorFlow

TensorFlow fue desarrollado por Google para satisfacer las necesidades a partir de redes neuronales artificiales. Permite construir y entrenar redes neuronales para detectar patrones y razonamientos usados por los humanos (Prada, 2022).

“TensorFlow Lite es una biblioteca para dispositivos móviles con la que puedes implementar modelos en dispositivos móviles o perimetrales y microcontroladores.” (TensorFlow, s.f.)

5.6.3. Keras

Keras es una biblioteca de código abierto escrita en Python que se basa en el trabajo de François Chollet en el proyecto ONEIROS. Fue lanzada en 2015 y tiene como objetivo acelerar la creación de redes neuronales. Keras funciona como una interfaz de uso intuitivo

que permite acceder a varios frameworks de aprendizaje automático y desarrollarlos, como Theano, Microsoft Cognitive Toolkit y TensorFlow (Ionos Digital Guide, 2020).

5.6.4. Python

Python es un lenguaje de programación muy popular en diversos campos, como aplicaciones web, desarrollo de software, ciencia de datos y machine learning (ML). Los programadores optan por utilizar Python debido a su eficiencia y facilidad de aprendizaje, y también porque puede ejecutarse en múltiples plataformas. Python está disponible de forma gratuita, se integra perfectamente con diversos sistemas y agiliza significativamente el proceso de desarrollo (aws.amazon, s.f.).

5.6.5. Google Drive

Google Drive es una plataforma de almacenamiento de información en la nube, lo que significa que los datos se guardan en línea y pueden ser accedidos desde cualquier dispositivo con conexión a internet. Esta herramienta cuenta con potentes capacidades de búsqueda proporcionadas por google, lo que asegura rapidez, confiabilidad y opciones para trabajar en colaboración. (Google, s.f.)

5.6.6. Android Studio

Android Studio es el IDE oficial para el desarrollo de aplicaciones de Android basado en IntelliJ IDEA. Ofrece herramientas para desarrolladores y una gran cantidad de funciones para aumentar la productividad. Está disponible para varios sistemas operativos, incluyendo GNU/Linux, MacOS, Microsoft Windows y ChromeOS. (Developer Android, s.f.).

5.6.7. Flutter

Flutter es un framework para desarrollar aplicaciones para diferentes plataformas elaborado por Google. Este kit de desarrollo ofrece un gran número de bibliotecas para elementos estándar de la interfaz de usuario de Android y iOS, pero también sirve para desarrollar aplicaciones web de escritorio (Ionos Digital Guide, 2020).

Se utiliza para desarrollar aplicaciones multiplataforma para Android, iOS, Linux, MacOS, Windows, Google Fuchsia y la web a partir de un único código base.

5.6.8 Dart

Dart es un lenguaje de programación de propósito general de código abierto. Originalmente fue desarrollado por Google y luego aprobado como estándar por ECMA. Se utiliza para crear aplicaciones web, de escritorio y móviles (Alvarez Corredera, s.f.).

Dart es un lenguaje de programación integrado en el framework Flutter, este tiene la ventaja de no requerir de otro lenguaje de programación para la creación de aplicaciones ya que es posible trabajar frontend y backend con código Dart.

5.6.9. Gestor de base de datos SQLite

SQLite es una herramienta de software libre que permite almacenar información de manera sencilla, eficaz y rápida en dispositivos embebidos con limitaciones de hardware, como PDA o teléfonos móviles. Implementa el estándar SQL92 y ofrece extensiones que facilitan su uso en cualquier ambiente de desarrollo. Es compatible con diversas plataformas y se puede utilizar tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos. Esto hace que la portabilidad entre dispositivos y plataformas sea transparente (Rommel, s.f.).

6. Análisis y Presentación de Resultados

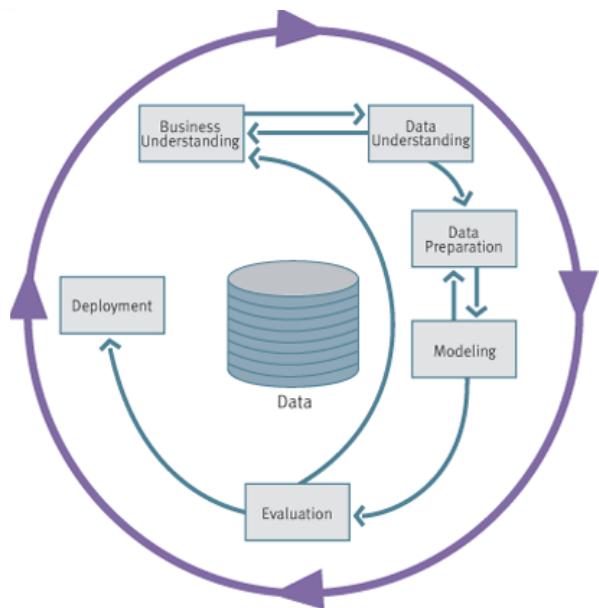
6.1. Modelo CRISP-DM

CRISP-DM (Cross Industry Standard Process for Data Mining), en español, Proceso Estándar Interindustrias o Transversal para la Minería de Datos, proporciona una descripción normalizada del ciclo de vida de un proyecto estándar de análisis de datos. Esto es análogo a como se realiza en la ingeniería del software con los modelos de ciclo de vida de desarrollo de software (singular, s.f.).

6.2. Fases del modelo CRISP-DM

Figura 4

Fases del Modelo CRISP-DM



Nota: Tomada de: (Goicochea, 2009)

El modelo CRISP-DM permite flexibilidad en la secuencia de las fases, y el resultado de cada fase determina la siguiente tarea. El proyecto no termina después de implementar la solución, ya que pueden surgir nuevas iteraciones del modelo. Los procesos posteriores de análisis se benefician de las experiencias previas obtenidas (Dataprix, 2007).

6.2.1. Business Understanding (Comprensión del negocio)

En el contexto del Deep Learning, la primera fase se enfoca en comprender los objetivos del proyecto y definir un problema específico de Deep Learning. Luego, se desarrolla un plan preliminar que se ajuste a las necesidades del proyecto y considere los recursos disponibles (singular, s.f.).

En esta primera fase, se definieron los objetivos del negocio, que están estrechamente relacionados con los objetivos del proyecto monográfico. También se establecieron criterios de éxito para el proyecto, se evaluaron las condiciones para el desarrollo del proyecto en torno a criterios técnicos y financieros, se definieron los objetivos en torno a data analytics, se definieron los criterios de evaluación para el proyecto y se estableció el plan de ejecución del proyecto.

6.2.1.1. Determinar los objetivos del negocio

Entiéndase aquí los objetivos del negocio como el alcance y criterios de éxito del proyecto.

El objetivo de este proyecto es, como su título indica, implementar un modelo de red neuronal convolucional para el reconocimiento de enfermedades de la piel. Sujeto a este objetivo general están los objetivos específicos del proyecto que se enfocan en el cumplimiento de cada una de las fases del modelo CRISP-DM y fueron definidos previamente, desde el enfoque del negocio, los objetivos a alcanzar son los siguientes:

- Construir un modelo de machine learning para el reconocimiento de enfermedades de la piel aplicando la técnica de redes neuronales convolucionales.
- Desarrollar una aplicación móvil como herramienta de despliegue para el modelo construido.
- Implantar el modelo en la aplicación móvil para su usabilidad.
- Construir una base documental para proyectos que consideren la rama del machine learning.

Criterios de éxito

Como criterios de éxito para este proyecto, se definieron los siguientes:

1. Precisión en los resultados: El modelo resultante es capaz de identificar las enfermedades de la piel con un alto porcentaje de precisión en las categorías contenidas.
2. Accesibilidad: El modelo está integrado en una aplicación móvil, brindando accesibilidad al mismo, la app que integra el modelo cuenta con una interfaz intuitiva y fácil de usar.
3. Documentación: Se consolidó la documentación construida en el transcurso de la ejecución de cada una de las fases del modelo CRISP-DM.

6.2.1.2. Evaluación de la situación

Este apartado detalla los requerimientos funcionales y no funcionales para el proyecto, además, del análisis de la disposición de los recursos técnicos, financieros y humanos en cuestión de ejecución del proyecto.

6.2.1.2.1. Requerimientos Funcionales

Requerimientos Funcionales para aplicación móvil DermaDI

- Captura de imagen: Permitir a los usuarios capturar una imagen utilizando la cámara del dispositivo.
- Cargar imagen: Permitir a los usuarios seleccionar y cargar una imagen desde la galería del dispositivo.
- Galería de enfermedades: Mostrar una galería que exhiba las enfermedades de piel con las que trabaja el modelo.
- Análisis de imagen: Procesar y analizar la imagen cargada o capturada utilizando el modelo de red neuronal convolucional para determinar las posibles enfermedades de la piel.

- Múltiples salidas y nivel de prioridad: Proporcionar más de una salida posible después del análisis, con cada salida acompañada de un nivel de prioridad basado en la precisión del modelo.
- Información de enfermedades: Brindar información detallada sobre las enfermedades de piel identificadas, incluyendo descripción de la enfermedad, sintomatología, posibles causas y nivel de riesgo.
- Comunicación con los desarrolladores: Incluir un botón para que los usuarios puedan comunicarse con el equipo de desarrolladores en caso de consultas, sugerencias o problemas técnicos.
- Activación/desactivación del flash: Incorporar un botón para activar o desactivar el flash de la cámara en el módulo de captura de imagen.

Requerimientos Funcionales para el modelo CNN

- Entrenamiento del modelo: Realizar el entrenamiento del modelo utilizando un conjunto de datos adecuado y la técnica de aprendizaje profundo de redes neuronales convolucionales.
- Clasificación de enfermedades: El modelo debe ser capaz de clasificar y reconocer las enfermedades de piel con un alto nivel de precisión.
- Múltiples salidas y nivel de prioridad: El modelo debe ser capaz de proporcionar más de una salida posible, asignando un nivel de prioridad basado en la confiabilidad de los resultados.

6.2.1.2.2. Requerimientos No Funcionales

Requerimientos No Funcionales para aplicación móvil DermaDI

- Usabilidad: La aplicación debe ser intuitiva, fácil de usar y contar con una interfaz de usuario atractiva.
- Rendimiento: El procesamiento y análisis de las imágenes deben realizarse de manera rápida y eficiente.
- Precisión y confiabilidad: El modelo de reconocimiento de enfermedades de piel debe tener un alto nivel de precisión y asertividad en sus resultados.

- Interoperabilidad: La aplicación debe poder comunicarse correctamente con el modelo de red neuronal convolucional y los servicios asociados.

Requerimientos No Funcionales para el modelo CNN

- Precisión y confiabilidad: El modelo debe tener una precisión y confiabilidad altas en la identificación de enfermedades de piel.
- Eficiencia computacional: El modelo debe ser eficiente en términos de uso de recursos computacionales para su despliegue en la aplicación móvil.

6.2.1.3. Estudio de Factibilidad

La factibilidad técnica, según (Kendall & Kendall, 2011), se refiere a la evaluación de si se cuenta con los conocimientos y habilidades necesarios en el manejo de métodos, procedimientos y funciones requeridos para el desarrollo e implementación o implantación de un proyecto. Además, aborda la disponibilidad de equipos y herramientas esenciales para llevarlo a cabo. En caso de que los recursos no estén disponibles, se considera la viabilidad de generarlos o crearlos dentro del plazo establecido por el proyecto y posteriormente implementarlos.

6.2.1.3.1. Alternativa 1.

6.2.1.3.1.1. Factibilidad Técnica

En esta alternativa se establecen los requisitos mínimos de hardware y software necesarios para el desarrollo e implantación de este proyecto desde una perspectiva académica y con la obtención de un producto prototipo. La Tabla 1 describe los recursos de hardware existentes dispuestos para el desarrollo de este proyecto, mientras que la Tabla 2 detalla los recursos necesarios a nivel de software, los cuales son recursos existentes, gratuitos o de código abierto.

En cuanto al factor humano, en esta alternativa de factibilidad del proyecto, el personal se compone de 2 personas, quienes desempeñan los roles de analista y programador en

conjunto. Cada persona involucrada es considerada recurso humano y debe ser tomada en cuenta en el análisis de disposición de recursos, en la tabla 3 se describe el rol y tareas del recurso humano dispuesto para este proyecto.

Tabla 1

Alternativa 1. Recursos de Hardware

Ítems	Nombre del Recurso	Descripción	Cant
1	Computador de escritorio Dell Precision T3600	Procesador: Intel(R) Xeon(R) E5 1620 0 3.8GHz	1
		GPU: RX570 4GB	1
		1 TB de almacenamiento	1
		16 GB RAM	1
		Otros: Monitor, Teclado, Mouse	1
2	Computador Laptop Asus TUF F15	Procesador: Intel(R) Core(TM) i5-11400H 2.70GHz	1
		GPU: NVIDIA GeForce RTX3050 4GB	1
		500 GB de Almacenamiento	1
		8GB RAM	1
3	Móvil Nokia C12	Procesador: Unisoc 9863A1	1
		64 GB de Almacenamiento	1
		2 GB RAM	1
		Otros: Cámara 8 MP	1
4	Móvil Samsung A34	Procesador: Mediatek Dimensity 1080 2.6GHz	1
		128 GB de Almacenamiento	1
		6 GB RAM	1
		Otros Cámara Triple, 48MP+8MP+5MP	1

Nota: Obtenida por elaboración propia.

Tabla 2*Alternativa 1. Recursos de software*

Ítems	Nombre del Recurso	Descripción	Cant
1	Sistema Operativo Dell Precision 3600	Fedora Linux	1
2	Sistema Operativo Laptop Asus Tuf F15	Windows 11 Home	1
3	Sistema Operativo Nokia C12	Android 12 Go Edition	1
4	Sistema Operativo Samsung A34	Android 13	1
5	Gestor de Base de Datos	SQL Lite	1
6	Lenguajes de Programación	Python	1
		Dart	1
7	Entorno de Desarrollo (IDE)	Android Studio	1
8	Frameworks	Flutter	1
9	Herramientas Auxiliares	Google Colaboratory	1
		Google Drive	1
		TensorFlow	1

Nota: Obtenida por elaboración propia.**Tabla 3***Alternativa 1. Recursos Humanos*

Ítem s	Nombre del Recurso	Descripción / Características	Cant
1	Analista – Programador. Especialista en datos.	<ul style="list-style-type: none"> - Analizar los objetivos del negocio - Construir modelo de red neuronal convolucional - Desarrollar aplicativo para despliegue del modelo 	2

Nota: Obtenida por elaboración propia.

6.2.1.3.1.2. Factibilidad Económica.

En el estudio de factibilidad económica, se realizó el presupuesto de costos de los recursos técnicos, humanos y materiales requeridos para el desarrollo y ejecución del proyecto. En esta alternativa, los recursos de hardware son dispuestos por el equipo de desarrollo, y aunque no generan gastos directos, se considera el costo que implica este equipo.

En relación con el recurso de software, bajo esta alternativa no genera gastos considerando que el S.O Windows viene incluido en el equipo de hardware correspondiente y el S.O Fedora Linux es un producto de código abierto, por lo tanto, está disponible de forma gratuita. Además, los demás elementos de software fueron utilizados en sus versiones de comunidad o gratuitas.

En los costos de desarrollo, además de los aspectos técnicos y de software, se contempla el pago al equipo de desarrollo constituidos por los especialistas en datos, quienes analizan, planifican y ejecutan todas las actividades del proyecto. En el contexto de este proyecto, se asume el pago de dos personas, considerando un salario promedio para un especialista en datos de \$900.00 mensuales por persona, durante un periodo de 6 meses. Los detalles de costos se describen en la Tabla 4.

Tabla 4

Alternativa 1. Factibilidad Económica

RECURSOS DE HARDWARE		
Cant	Recurso	Precio total USD (\$)
1	Computador de escritorio Dell Precision T3600	\$ 830.00
1	Computador Laptop Asus TUF F15	\$ 1,040.00
1	Móvil Nokia C12	\$ 100.00
1	Móvil Samsung A34	\$ 305.00
Sub Total		\$ 2,275.00

RECURSOS DE SOFTWARE						
Cant	Recurso	Descripción del recurso		Precio total USD (\$)		
1	Sistema Operativo	Windows 11 Home		\$ 00.00		
1	Sistema Operativo	Fedora Linux		\$ 00.00		
1	Gestor de BD	SQL Lite		\$ 00.00		
1	Lenguaje de programación	Python		\$ 00.00		
1	Lenguaje de programación	Dart		\$ 00.00		
1	Entorno de desarrollo (IDE)	Android Studio		\$ 00.00		
1	Framework	Flutter		\$ 00.00		
1	Herramienta auxiliar	Google Colaboratory		\$ 00.00		
1	Herramienta auxiliar	Google Drive		\$ 00.00		
1	Herramienta auxiliar	TensorFlow		\$ 00.00		
Sub Total				\$ 00.00		
RECURSO HUMANO						
Cant	Recurso	Salario mes USD (\$)	Tiempo estimado del proyecto	Salario USD (\$)		
2	Analista – Programador, Especialista en datos	\$ 900.00	6 meses	\$ 10,800.		
Sub Total				\$ 10,800.00		
TOTAL NETO				\$ 13,075.00		

Nota. El salario asumido para este estudio se obtuvo del sitio tusalario.org (TuSalario, 2023). Obtenida por Elaboración propia.

6.2.1.3.2. Alternativa 2.

6.2.1.3.2.1. Factibilidad Técnica

En esta alternativa, se establecen los requisitos de hardware y software óptimos para el desarrollo e implantación del proyecto desde una perspectiva de negocio, buscando desarrollar un proyecto a mayor escala con un producto resultante productivo que ofrezca un modelo de negocio para los desarrolladores.

Por tanto, respecto a recursos de hardware, se requiere de una Workstation para trabajar el desarrollo de manera local, además de cualquier dispositivo móvil con S.O Android en la versión 7 o superior. La Tabla 5 describe los recursos de hardware necesarios, mientras que la Tabla 6 detalla los recursos de software requeridos. Respecto al recurso humano se es el mismo de la alternativa 1.

Tabla 5.

Alternativa 2. Recursos de Hardware.

Ítems	Nombre del Recurso	Descripción	Cant
1	WX-620 Dual GPU	Procesador: Intel i9 13900K	1
		GPU: RTX 4090 24GB	2
	Workstation KALAMIT	Almacenamiento SSD NVMe 16 TB	1
		64 GB RAM	1
		Otros: Monitor, Teclado, Mouse	1
2	Móvil Nokia C12	Procesador: Unisoc 9863A1	1
		64 GB de Almacenamiento	1
		2 GB RAM	1
		Otros: Cámara 8 MP	1

Nota: Obtenida por elaboración propia.

Tabla 6.

Alternativa 2. Recursos de Software.

Ítems	Nombre del Recurso	Descripción	Cant
1	Sistema Operativo WK	Fedora Linux	1
3	Sistema Operativo Mokia C12	Android 12 Go Edition	1
5	Gestor de Base de Datos	SQL Lite	1
6	Lenguajes de Programación	Python	1
		Dart	1

7	Entorno de Desarrollo (IDE)	Android Studio	1
8	Frameworks	Flutter	1
	Herramienta Auxiliar	TensorFlow	1

Nota: Obtenida por elaboración propia.

Tabla 7.

Alternativa 2. Recursos Humanos

Ítem s	Nombre del Recurso	Descripción / Características	Cant
1	Analista – Programador. Especialista en datos.	<ul style="list-style-type: none"> - Analizar los objetivos del negocio - Construir modelo de red neuronal convolucional - Desarrollar aplicativo para despliegue del modelo 	2

Nota: Obtenida por elaboración propia.

6.2.1.3.2.2. Factibilidad Económica

En el estudio de factibilidad económica, para la alternativa 2 se realizó el presupuesto de costos de los recursos técnicos, humanos y materiales requeridos para el desarrollo y ejecución del proyecto. En esta alternativa, los recursos de hardware deberán ser comprados o financiados por el potencial cliente interesado en el proyecto. Los costos pueden variar a lo estipulado en este análisis, sin embargo, el equipo debe contar con los requerimientos mínimos de hardware detallados en la Tabla 5 en dependencia de la calidad y tamaño del modelo de CNN a construirse.

En relación con el recurso de software, bajo esta alternativa no genera gastos considerando que el S.O Fedora Linux es un producto de código abierto, por lo tanto, está disponible de forma gratuita. Además, los demás elementos de software fueron utilizados en sus versiones de comunidad o gratuitas. Con respecto a los costos de Recurso Humano

prevalecen los mismos que para la alternativa 1. Los detalles de costos se describen en la Tabla 8.

Tabla 8.

Alternativa 2. Factibilidad Económica.

RECURSOS DE HARDWARE				
Cant	Recurso			Precio total USD (\$)
1	Workstation KALAMIT			\$ 6,999.00
1	Móvil Nokia C12			\$ 100.00
Sub Total				\$ 7,099.00
RECURSOS DE SOFTWARE				
Cant	Recurso	Descripción del recurso		Precio total USD (\$)
1	Sistema Operativo	Fedora Linux		\$ 00.00
1	Gestor de BD	SQL Lite		\$ 00.00
1	Lenguaje de programación	Python		\$ 00.00
1	Lenguaje de programación	Dart		\$ 00.00
1	Entorno de desarrollo (IDE)	Android Studio		\$ 00.00
1	Framework	Flutter		\$ 00.00
1	Herramienta auxiliar	TensorFlow		\$ 00.00
Sub Total				\$ 00.00
RECURSO HUMANO				
Cant	Recurso	Salario mes USD (\$)	Tiempo estimado del proyecto	Salario USD (\$)
2	Analista – Programador, Especialista en datos	\$ 900.00	6 meses	\$ 10,800.
Sub Total				\$ 10,800.00
TOTAL NETO				\$ 17,899.00

Nota. El salario asumido para este estudio se obtuvo del sitio tusalario.org (TuSalario, 2023). Obtenida por Elaboración propia.

6.2.1.4. Análisis y Selección de Alternativa

Considerando los objetivos y el alcance perseguido con este proyecto de trabajo monográfico, se determinó la alternativa de recursos y costos más factible para su ejecución. A continuación, se describen los factores influyentes en la selección de la alternativa de factibilidad más conveniente.

- Factor Hardware: Dado que el equipo de desarrollo cuenta con los recursos de un computador de escritorio tipo server y de una laptop, y que ambos equipos cumplen con los requerimientos mínimos necesarios para ejecutar el proyecto en una versión prototipo. Además, el equipo de desarrollo dispone de los equipos móviles con los requerimientos necesarios para la integración del modelo resultante. Este factor no es un aspecto decisivo para la selección de alternativa.
- Factor Software: El desarrollo de este proyecto considera muchos productos de software, sin embargo, estos son gratuitos, de código abierto o cuentan con una versión de comunidad, lo que convierte a este factor en no decisivo.
- Factor Humano: similar al factor hardware, este tampoco tiene relevancia en la elección de alternativas, Los especialistas de datos en cualquiera de las alternativas son los integrantes del equipo monográfico, quienes, en cumplimiento con los requisitos para la obtención de título de ingenieros, no generan costos de contratación por el desarrollo de este proyecto.

Considerando estos aspectos, se determina que, dado que el objetivo de este proyecto monográfico se alcanza con la implementación del modelo y la obtención de un producto resultante en una versión de prototipo, no se requiere la inversión de recursos adicionales a los ya existentes y dispuestos por el equipo de desarrollo. Esta es la alternativa más factible y a tomar en cuenta en el desarrollo de este proyecto monográfico.

Es importante destacar que la factibilidad de recursos técnicos y económicos considerada en la alternativa 2 es rescatable en un momento que este proyecto sea escalable y maximice su alcance desde un enfoque de emprendimiento.

6.2.1.5. Factibilidad Operativa del proyecto

El modelo de red neuronal para el reconocimiento de enfermedades de piel debía ser implantado en una herramienta que proporciona accesibilidad al mismo. Por tanto, se desarrolló la herramienta DermaDI, la cual contiene el modelo de clasificación, y proporciona imágenes e información sobre las enfermedades incluidas en el modelo.

La app DermaDI puede ser utilizada por cualquier usuario, aunque está dirigida principalmente al uso por médicos de atención primaria. No requiere almacenar ningún tipo de información del usuario, ya que no es necesario una suscripción para el uso de la herramienta. Las funcionalidades de la app fueron descritas en este documento en el apartado de “Evaluación de la situación”, estas incluyen la apertura de la cámara al iniciar la aplicación para facilitar la captura de fotografías, así como la opción de cargar imágenes desde la galería del dispositivo. El modelo clasifica la imagen dada y muestra al usuario los posibles resultados con los que encuentra similitud de características. Posterior a la tarea de clasificación y tras devolver el resultado, la imagen dada es destruida. Por tanto, la app no requiere de asignación de roles, ni adjudicación o retención de permisos más que los permisos administrados por el dispositivo. La app cuenta con un apartado denominado “acerca de”, donde se refleja la información sobre la aplicación DermaDI; además, se muestran los correos de contacto de los desarrolladores, facilitando la retroalimentación de los usuarios al equipo de desarrollo.

La herramienta DermaDI es ejecutable en dispositivos con sistema operativo Android, y la versión mínima requerida para el óptimo aprovechamiento de la app es a partir de la versión 7 o Android Nougat, esto para los usuarios finales del modelo. Considerando este proyecto escalado a un mercado remunerado, deberá de considerarse otros aspectos para la

operatividad, como la utilización constante de un servidor que administre el modelo y que el dispositivo usado por el usuario final cuente con conexión wifi.

6.2.1.6. Objetivos de la minería de datos

En esta actividad se determinaron los objetivos de data analytics que permiten alcanzar el logro de los objetivos del negocio. También, se definen los criterios de evaluación a aplicar para determinar el éxito del proyecto.

Los objetivos en términos de minería de datos son los siguientes:

- Recopilar y analizar un conjunto de datos representativo.
 - Desarrollar un modelo de clasificación efectivo, considerando múltiples salidas.
 - Evaluar y validar el rendimiento del modelo.
1. Recopilar y analizar un conjunto de datos representativo: Este objetivo se alcanzó con la recopilación de datos de imágenes con diferentes enfermedades de la piel. La colección de la mayor cantidad posible de datos es importante para mejorar la precisión y la capacidad predictiva del modelo de reconocimiento. **Para este proyecto se establecieron 8 categorías de enfermedades de piel de las cuales se recolectó la mayor cantidad posible de imágenes por categoría.**
 2. Desarrollar un modelo de clasificación efectivo, considerando múltiples salidas: Este objetivo se alcanzó tras la construcción, entrenamiento y optimización del modelo de redes neuronales convolucionales, siendo capaz de clasificar con precisión diferentes enfermedades de la piel, considerando un resultado con múltiples salidas priorizando aquella con el porcentaje de precisión más alto.
 3. Evaluar y validar el rendimiento del modelo: Este objetivo se alcanzó tras la implementación de métricas de evaluación para medir el rendimiento del modelo de redes neuronales convolucionales.

Definición de Criterios de evaluación

El modelo de reconocimiento está construido de modo que proporciona **múltiples salidas**, teniendo en cuenta que las enfermedades de piel en múltiples casos tienen similitudes entre sí. Por lo tanto, los criterios de evaluación para el modelo se definen considerando esto.

A continuación, se describen los criterios de evaluación definidos para medir el rendimiento del modelo de manera cuantitativa.

- Precisión por clase: Calcula la precisión para cada clase de salida por separado. Esto implica medir la proporción de predicciones correctas para cada clase en relación con el total de muestras pertenecientes a esa clase. La precisión por clase ayuda a identificar las fortalezas y debilidades del modelo en la clasificación de cada clase en particular.
- Precisión global: Calcula la precisión global del modelo considerando todas las salidas en conjunto. Esto implica medir la proporción de predicciones correctas en relación con el total de muestras. Con la evaluación de la precisión global del modelo se evalúa el rendimiento del modelo de forma general.
- Matriz de confusión: Muestra las predicciones del modelo en relación con las clases reales. La matriz de confusión permite visualizar el desempeño del modelo para cada clase y determinar si hay algún patrón de errores o confusiones entre las salidas.
- F1-score: El F1-score es una métrica que combina la precisión y el recall (sensibilidad), y es especialmente útil cuando hay clases desbalanceadas en el conjunto de datos. Este criterio proporciona una evaluación más equilibrada del modelo y es de utilidad para evaluar el rendimiento del modelo.
- Métricas específicas de dominio: Estas métricas se definen dependiendo del dominio de aplicación (en este caso detección médica), existen métricas específicas que son relevantes para evaluar el rendimiento del modelo, para este proyecto se aplicaron las siguientes:
 - **Sensibilidad (Exhaustividad) / especificidad:** La sensibilidad se refiere a la capacidad del modelo para identificar correctamente los casos positivos, mientras

que la especificidad se refiere a la capacidad del modelo para identificar correctamente los casos negativos.

- **Valor predictivo positivo (VPP)/Valor predictivo negativo (VPN):** Estas métricas están relacionadas con la precisión de las predicciones positivas y negativas. El VPP es la proporción de casos positivos correctamente identificados por el modelo, mientras que el VPN es la proporción de casos negativos correctamente identificados por el modelo.

6.2.1.7. Producir el plan del proyecto

6.2.1.7.1. Plan del proyecto

El proyecto se desarrolló según la siguiente estructura:

- Etapa 1: Definición de objetivos y planeación (3 semanas): Se establecieron los objetivos, se evaluó la situación actual del negocio y se planificó la ejecución del proyecto.
- Etapa 2: Recolección de datos (4 semanas): Se llevaron a cabo las actividades relacionadas con la obtención de imágenes para la construcción del conjunto de datos. También se revisó la calidad de los datos recopilados.
- Etapa 3: Selección de datos y limpieza de datos (4 semanas): Se seleccionaron las imágenes relevantes y se realizó una limpieza de datos para garantizar la calidad y coherencia del conjunto de datos.
- Etapa 4: Diseño del test (5 semanas): Se llevó a cabo la selección de los datos dividiéndolos en conjunto para entrenamiento y conjunto para pruebas. Se construyó y entreno el modelo de reconocimiento. También se evaluó el rendimiento del mismo considerando distintas métricas de evaluación

- Etapa 5: Evaluación de resultados (4 semanas): Los resultados obtenidos se evaluaron en función de los objetivos del proyecto y sus criterios de éxito. También se realizó una revisión general del proyecto y las actividades realizadas hasta ese punto.
- Etapa 6: Desarrollo de DermaDI e implantación del modelo (5 semanas): Durante esta etapa se desarrolló la herramienta tecnológica a utilizar para implantar el modelo y brindar accesibilidad al mismo. Bajo la actividad “Implantación” se describe el diseño y funcionalidad de la app DermaDI.
- Etapa 7: Evaluación del modelo en producción (2 semanas): En esta etapa del proyecto se llevó a cabo la visita al Hospital Central Dermatológico Nacional, se presentó la aplicación DermaDI, y se levantó un formulario de encuesta dirigida a los médicos del hospital para evaluar la recepción de la app y recoger retroalimentación por parte de los especialistas. Luego, se analizó y documentó los resultados obtenidos tras la visita.
- Etapa 8: Documentación y conclusiones (5 semanas): Se consolidó toda la documentación generada a lo largo de las fases del proyecto. También se brindaron conclusiones y recomendaciones en base al proyecto monográfico.

6.2.1.7.2. Evaluación inicial de herramientas y técnicas.

La evaluación inicial de las herramientas y técnicas utilizadas en este proyecto demuestra su relevancia y aplicabilidad en el desarrollo de la aplicación DermaDI y en la implementación de los modelos de clasificación de imágenes. Cada una de estas herramientas y técnicas desempeña un papel fundamental en diferentes aspectos del proyecto. A continuación, se presenta una evaluación de su participación y aplicación:

Herramientas

Google Colaboratory: Esta herramienta brinda un entorno de desarrollo colaborativo y proporciona accesibilidad de forma remota a recursos computacionales altamente

eficientes para la creación del modelo y la realización de las pruebas necesarias para su evaluación.

TensorFlow y Keras: Ambas herramientas son librerías de aprendizaje automático, estas son utilizadas para la construcción y el entrenamiento del modelo de reconocimiento de enfermedades de piel, ayudan a simplificar el proceso de desarrollo y configuración del modelo.

Python: Es el lenguaje de programación utilizado por Colab, relacionado a este lenguaje de programación se encuentran las bibliotecas más utilizadas para la implementación de Deep learning, como lo son Keras y TensorFlow, entre otras.

Google Drive: Esta herramienta es utilizada para almacenar el conjunto de datos utilizados para el entrenamiento del modelo, además, es utilizada para almacenar la información resultante de los formularios de usabilidad y recolección de imágenes.

Android Studio: Esta herramienta proporciona el entorno de desarrollo para la creación de la aplicación móvil DermaDI permitiendo la instalación del plugin necesario para la utilización del framework Flutter y el lenguaje de programación Dart. También, permite compilar e instalar la aplicación directamente en el teléfono agilizando la realización de pruebas de funcionalidad.

Flutter: El framework flutter proporciona las bibliotecas utilizadas para el desarrollo de la interfaz gráfica de la app DermaDI.

Dart: Es lenguaje de programación utilizado por el framework flutter.

SQLite: SQLite es utilizado para la creación de la base de datos que contiene la información de consulta sobre cada clase perteneciente al modelo, es decir la información que se proporciona al usuario sobre las enfermedades de piel que el modelo clasifica.

Las herramientas seleccionadas para este proyecto han demostrado su valía al desempeñar roles clave en el desarrollo de la aplicación DermaDI y la implementación del modelo de clasificación de imágenes. Su participación contribuye a la funcionalidad, rendimiento y éxito general del proyecto.

Técnicas

En este proyecto se implementaron diversas técnicas, tanto para el desarrollo, como para la evaluación.

Redes Neuronales Convolucionales (CNN): Esta técnica fue definida desde el nacimiento de la idea del proyecto, dado que es una técnica de aprendizaje profundo ideal para el reconocimiento de imágenes. La utilización de esta técnica permite extraer las características y patrones relevantes de las imágenes de enfermedades de piel.

Preprocesamiento de Imágenes: Se realizaron técnicas de preprocesamiento de imágenes para garantizar la calidad y consistencia de los datos de entrada. Esto incluyó la normalización de las imágenes.

Validación Cruzada: Esta técnica es utilizada para evaluar el rendimiento del modelo de manera robusta. Esto implicó dividir el conjunto de datos en conjuntos de entrenamiento y prueba, y realizar múltiples iteraciones de entrenamiento y evaluación para obtener medidas de rendimiento más confiables.

Métricas de Evaluación: Se emplearon métricas de evaluación estándar, como precisión, puntuación F1, y técnicas de dominio para medir el rendimiento del modelo en la clasificación de enfermedades de la piel. Estas técnicas permiten cuantificar la capacidad del modelo para identificar correctamente las diferentes afecciones dermatológicas.

Es importante destacar que estas técnicas son adaptadas específicamente para abordar los desafíos y requisitos del proyecto. La combinación de estas técnicas permite alcanzar el cumplimiento de los criterios de éxito de este proyecto.

6.2.2. Data Understanding (Comprensión de los datos)

Durante esta fase se realiza la recolección y exploración inicial de los datos con el objetivo de familiarizarse con el problema. Es crucial que esta fase se realice adecuadamente ya que

cualquier malinterpretación de los datos podría tener consecuencias graves, como aumentar el tiempo total del proyecto y disminuir las posibilidades de éxito. (Alvarez Gil, 2021).

En esta fase se llevó a cabo la recolección de los datos, los cuales consisten en imágenes de enfermedades de piel. Se validó la fuente de obtención de datos, y se revisó y analizó los datos recolectados para garantizar que estos sean correctos y correspondan a las etiquetas adecuadas.

6.2.2.1. Recolectar los datos iniciales

La recolección de datos para este proyecto se basó en la recopilación de imágenes de enfermedades de la piel. Debido a la indisponibilidad para la recolección de datos manualmente, se optó por utilizar un conjunto de datos proporcionado por Kaggle. En particular, se utilizó el conjunto de datos recogido por DermNet, una plataforma en línea que recopila y comparte imágenes dermatológicas con fines educativos. DermNet NZ (DermNet, s.f.) es un sitio web educativo de dermatología de Nueva Zelanda que proporciona información detallada sobre diversas condiciones de la piel, así como fotografías de enfermedades cutáneas.

DermNet NZ obtiene los datos y las imágenes que se muestran en su sitio web de varias fuentes, incluyendo:

- Contribuciones de expertos en dermatología: Dermatólogos y profesionales médicos especializados en dermatología contribuyen con información y fotografías para el sitio web.
- Fotografías de pacientes con su consentimiento: Los pacientes que acuden a dermatólogos y dan su consentimiento para que sus imágenes sean utilizadas con fines educativos pueden contribuir con fotografías de sus afecciones cutáneas.
- Imágenes de dominio público: En algunos casos, DermNet puede utilizar imágenes de dominio público o con licencia para ilustrar ciertas condiciones de la piel.

Esto es comprobable visitando el sitio web de Dermnet. En la parte inferior de la pantalla de inicio, se encuentra un enlace proporcionado a todo visitante para enviar sus imágenes,

ya sea médico o paciente, DermNet especifica ciertos requisitos para la subida de alguna imagen, de acuerdo a su política de privacidad y términos de carga de imágenes.

Este conjunto de datos incluye una amplia variedad de imágenes de enfermedades de piel, proporcionando una base sólida para el desarrollo y entrenamiento de nuestro modelo de reconocimiento de enfermedades de piel. Cabe destacar que DermNet permite la descarga y uso gratuito de sus imágenes para fines educativos y de investigación, lo cual cumple con nuestras necesidades para el proyecto y es factible legalmente.

Todas las imágenes recopiladas fueron almacenadas en una carpeta privada de Google Drive. Las imágenes no contienen metadatos que pudiesen exponer datos personales de los pacientes y aquellas donde se muestra un rostro completo están censuradas respecto a las características más relevantes, como ojos, nariz, boca. Esto con el fin de no incumplir así cualquier regulación y/o política en torno a protección de la privacidad.

6.2.2.2. Descripción y exploración de los datos

Los datos utilizados en este proyecto para el entrenamiento y evaluación del modelo de reconocimiento de enfermedades de piel, como se mencionó anteriormente, provienen de un data set obtenido a partir de DermNet y proporcionado por Kaggle. Este data set contiene una amplia variedad de imágenes de diferentes afecciones dermatológicas.

El conjunto de datos facilitado por DermNet contenía un total de 23 etiquetas de datos, entendidas como categorías o clases de enfermedades de piel. Cada etiqueta se refiere a una enfermedad de la piel, y estas clases contienen diversas imágenes que representan las diferentes variaciones o presentaciones de una misma enfermedad.

El data set está compuesto por imágenes en formato JPG de distintas resoluciones. La clasificación de las imágenes en estas clases ha sido realizada por expertos médicos y dermatólogos, lo que garantiza la calidad y precisión de las etiquetas asociadas a cada imagen.

El conjunto de datos proporcionado por DermNet cuenta con un número aproximado de 15,000 imágenes, distribuidas entre las 23 etiquetas. La cantidad de datos por etiqueta varía

desde 400 hasta más de 1000 imágenes por clase, lo que significaba que el conjunto de datos no estaba equilibrado en términos de distribución.

6.2.2.3. Verificar la calidad de los datos.

La verificación de la calidad de los datos es una actividad necesaria para garantizar que los datos utilizados sean precisos, confiables y representativos. Para verificar la calidad de los datos, se realizaron diversas acciones y análisis, que se describen a continuación:

Revisión visual de las imágenes: Se llevó a cabo una revisión visual de las imágenes del data set, esto para validar las etiquetas contenidas y asegurar que estas estuviesen correctamente representadas. También se verificó la calidad de las imágenes, su resolución, y que estuviesen a color. Se pudo observar que todo el conjunto contiene imágenes con variaciones respecto a tamaño y resolución, aspecto que será corregido en la siguiente fase.

Análisis de distribución de clases: Como se mencionó en el apartado anterior “6.2.2.3. Descripción de los datos”, el data set contiene 23 etiquetas, tras el proceso de la revisión visual de los datos se validó que la distribución para estas 23 clases es bastante dispareja considerando que algunas etiquetas contienen un aproximado de poco más 400 imágenes y para otras se cuenta con más de 1000 imágenes aproximadamente. Considerando esto en fases posteriores se normalizó la cantidad de clases y la cantidad de datos que contendría cada clase seleccionada

Limpieza de datos: Se realizó una limpieza superficial de los datos en la que se revisó que no hubiera imágenes duplicadas o que no correspondiera a alguna etiqueta específica. Tras la verificación, se procedió a eliminar los datos no correspondidos.

6.2.3. Data Preparation (Preparación de los datos)

Esta fase se enfoca en todas las actividades requeridas para construir el conjunto final de datos que se utilizará en las herramientas de modelado. Estas actividades incluyen la transformación de los datos brutos iniciales y la realización de ajustes necesarios para garantizar la calidad y la relevancia del conjunto final de datos. (singular, s.f.).

En esta fase se considera el tratamiento de las imágenes. Se estableció la resolución de las imágenes, la cantidad de imágenes que contiene cada etiqueta, se verificó si se debe redimensionar aquellas imágenes que no cumplan con la resolución definida y se dividieron los datos que se utilizarán para el entrenamiento del modelo y para la evaluación del modelo.

6.2.3.1. Seleccionar los datos.

Considerando que parte de los criterios de éxito para este proyecto es que el modelo de reconocimiento de enfermedades de piel sea eficiente y con un alto porcentaje de precisión, se decidió trabajar con un conjunto reducido de etiquetas. En lugar de utilizar las 23 etiquetas originales del conjunto de datos, se enfocaron los esfuerzos en construir el modelo con ocho etiquetas específicas que representan algunas de las enfermedades de piel más comunes y relevantes según el conjunto de datos recogido.

Esta selección estratégica de ocho etiquetas tiene varios propósitos clave:

- Optimización de recursos: Trabajar con un número menor de etiquetas reduce la complejidad del modelo y el esfuerzo computacional requerido durante el entrenamiento. Esto permitió utilizar de manera más eficiente los recursos disponibles y acelerar el proceso de desarrollo.
- Mejora en la precisión del modelo: Al tener un enfoque en un conjunto más pequeño de clases, se dirigieron los esfuerzos a mejorar la precisión y el rendimiento del modelo en estas categorías específicas.
- Reducción del desequilibrio de clases: El data set original presenta un desequilibrio en la cantidad de muestras por cada etiqueta, lo que puede afectar el desempeño del modelo y sesgar sus predicciones hacia las clases más representadas. Al trabajar con un número reducido de etiquetas, se tiene la oportunidad de mitigar este desequilibrio y lograr un entrenamiento más balanceado.

Las ocho categorías o clases seleccionadas para este proyecto son:

- Clase 1. Acné y Rosácea.
- Clase 2. Melanoma.
- Clase 3. Tumores vasculares.
- Clase 4. Carcinoma.
- Clase 5. Queratosis Seborreicas.
- Clase 6. Vasculitis.
- Clase 7. Verrugas vulgares.
- Clase 8. Enfermedades ampollosas.

Al centrar los esfuerzos en estas clases específicas, se busca obtener un modelo de reconocimiento de alta calidad.

6.2.3.2. Limpiar los datos.

Con la definición de las etiquetas a trabajar para el modelo de reconocimiento, se llevó a cabo una revisión más profunda sobre los datos seleccionados. Se realizó una limpieza exhaustiva de dichos datos para asegurar la calidad y consistencia necesarias en el entrenamiento del modelo de reconocimiento de enfermedades de piel.

Como se mencionó en puntos anteriores, el conjunto de datos original de DermNet para algunas clases presentaba una variación considerable en la cantidad de imágenes respecto a otras clases. Para evitar sesgos o inclinación por parte del modelo hacia las clases con mayor representación, se tomó la decisión de igualar la cantidad de imágenes por etiqueta.

Por tanto, se definió que el conjunto de datos para el entrenamiento del modelo esté compuesto por 800 imágenes y para el conjunto de pruebas lo esté de 400 imágenes, esto por cada una de las ocho etiquetas seleccionadas, lo que significó un total de 1,200 imágenes por cada una de las clases, sumando un total de 9,600 imágenes para el conjunto de datos a utilizar en este proyecto. Esta decisión permitió que el modelo se entrene de manera equilibrada en todas las clases, asegurando una mayor objetividad y precisión en el reconocimiento de las distintas enfermedades de piel.

Durante el proceso de limpieza de datos, también se detectó y eliminó imágenes con información irrelevante que pudiesen afectar negativamente el rendimiento del modelo. Se

llevó a cabo técnicas de preprocessamiento como la normalización de tamaños y resoluciones de las imágenes para garantizar que todas las entradas del modelo contasen con una representación homogénea.

Con esta actividad, se buscó garantizar que el conjunto de datos estuviese limpio, equilibrado y listo para ser utilizado en el proceso de construcción del modelo de reconocimiento de enfermedades de piel.

6.2.3.3. Construir los datos

Finalizada la limpieza de datos, se procedió a la construcción del conjunto de datos a utilizar, tanto para el entrenamiento como para la evaluación del modelo de reconocimiento de enfermedades de piel.

En la actividad previa “6.2.3.2. Limpieza de datos” se realizó la división del conjunto de datos de entrenamiento y el conjunto de datos de prueba, manteniendo una proporción del 66.67% de los datos para entrenamiento y el 33.33% restante para pruebas en cada una de las etiquetas. Esta distribución permite evaluar la capacidad del modelo para generalizar a datos no vistos previamente.

Con esta verificación final del proceso de limpieza de datos y la preparación para su uso en la construcción del modelo, se concluyó que el conjunto de datos estaba listo para los siguientes pasos respecto al modelado.

6.2.4. Modeling (Modelado)

En la fase de modelado, se seleccionan y aplican las técnicas más apropiadas y se ajustan los parámetros a valores óptimos. Hay múltiples técnicas disponibles y algunas de ellas pueden requerir una forma específica de los datos, por lo que es posible que sea necesario volver a la fase de preparación de datos para adaptarlos. (singular, s.f.).

6.2.4.1. Selección de la técnica de modelado

Dada la naturaleza visual de los datos y la necesidad de identificar patrones en las imágenes para poder clasificarlas, desde el nacimiento de la idea de este proyecto se definió trabajar con la técnica de redes neuronales convolucionales (CNN)

Las CNN han demostrado ser altamente efectivas en la identificación de patrones visuales en imágenes, son capaces de aprender automáticamente características y representaciones jerárquicas en las imágenes esencial para capturar detalles sutiles. Esto las hace particularmente adecuadas para problemas de clasificación de imágenes como el reconocimiento de enfermedades de la piel (Na8, 2018).

Las redes neuronales convolucionales son altamente flexibles y escalables; es decir, pueden adaptarse a diferentes tamaños y resoluciones de imágenes, un aspecto importante en los casos en que el conjunto de datos contiene imágenes significativamente variables en calidad y detalle. Además, las CNN son una forma de aprendizaje profundo, lo que significa que pueden capturar relaciones complejas y no lineales en los datos. Estos aspectos son fundamentales para identificar o clasificar imágenes de enfermedades de piel que pueden tener múltiples presentaciones y variaciones.

Considerando lo anterior, se decidió que las CNN es la técnica de modelado adecuada para el reconocimiento de enfermedades de piel. Sin embargo, durante la actividad de construcción del modelo, se optimizaron hiperparámetros y otros métodos para garantizar un rendimiento óptimo del modelo en la identificación de enfermedades de la piel.

6.2.4.2. Generar el plan de Prueba

La fase de modelado de este proyecto implica la creación y evaluación de un modelo de reconocimiento de enfermedades de la piel basado en redes neuronales convolucionales (CNN). Para garantizar la evaluación sólida y efectiva de este modelo, se desarrolló un plan de pruebas bien estructurado que define las estrategias y métricas de evaluación.

1. Definición de conjuntos de prueba y validación

Previamente en la fase de ‘6.2.3. Preparación de los datos’ en las actividades de “6.2.3.2. Limpieza de datos” y “6.2.3.3. Construcción de los datos”, se determinó

y llevó a cabo la distribución del conjunto de datos a utilizar en el proyecto. Se estableció la cantidad de 800 imágenes por clase para el conjunto de entrenamiento y 400 imágenes por clase para el conjunto de pruebas, lo que equivale al 66.7% y 33.3% respectivamente del total de los datos. Esto se hizo con el objetivo de asegurar una representación equilibrada por clase.

2. Selección de métricas de evaluación:

En la sección “6.2.1.3. Objetivos de la minería de datos” de este documento, se definieron los criterios de evaluación para este proyecto. Estos criterios establecieron las métricas de evaluación aplicadas al modelo resultante para verificar la calidad del mismo.

Las métricas de evaluación incluyen:

- Precisión: Precisión por clase y precisión global.
- F1-Score
- Matriz de Confusión
- Métricas de dominio:
 - Sensibilidad
 - Especificidad
 - Valor Predictivo Positivo (VPP)
 - Valor Predictivo Negativo (VPN)

3. Ejecución de las métricas de evaluación:

Una vez completada la construcción y el entrenamiento del modelo de reconocimiento, se procedió a aplicar las métricas de evaluación descritas anteriormente y se validó la calidad del modelo respecto a los resultados de estas evaluaciones.

6.2.4.3. Construcción del modelo

Para la construcción y entrenamiento del modelo de red neuronal convolucional (CNN) para el reconocimiento de enfermedades de la piel, se empleó principalmente Google

Colaboratory. Esta plataforma permite la ejecución de código Python y la utilización de recursos informáticos de alto rendimiento de manera remota, aprovechando la infraestructura de servidores proporcionada por Google.

6.2.4.3.1. Preparación de Datos y definición de hiperparametros

Los datos utilizados han pasado por diferentes procesos, como la selección, limpieza y distribución, todo esto con el fin de construir un modelo eficiente. El conjunto de datos de entrenamiento fue la primera parte de los datos en ser utilizada. A continuación, se describe el proceso realizado en la construcción y entrenamiento del modelo CNN:

El proceso de construcción del modelo de reconocimiento se llevó a cabo comenzando por la importación de las librerías necesarias para los procesos de machine learning, principalmente TensorFlow y Keras, como se muestra en la figura 5.

Figura 5

Importación de librerías

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
from PIL import Image
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score, f1_score
import matplotlib.pyplot as plt
```

Nota: Obtenida por elaboración propia.

El conjunto de datos destinado para el entrenamiento del modelo fue almacenado en una carpeta de Google Drive. Desde Colaboratory, estos datos se cargaron al script indicando la ruta de acceso a dicha carpeta, como se aprecia en el siguiente código.

```
# Ruta del directorio de datos
ruta_datos = "/content/drive/MyDrive/TF/basura/traindata2"
```

Después de proporcionar la ruta del directorio de datos, se realizó un preprocesamiento de los mismos, donde se llevó a cabo una transformación de los datos, normalizando los píxeles de las imágenes. Esto se hizo dividiendo cada valor del píxel por 255.0, lo que permitió escalar los valores de los píxeles al rango [0,1]. Esta normalización ayuda al

modelo a converger más rápido durante el entrenamiento, lo que significa que ajusta sus parámetros internos para que sus predicciones se acerquen cada vez más a los valores reales o deseados, como se observa en la línea de código siguiente.

```
datagen = ImageDataGenerator(rescale=1.0/255.0) # Normalización de  
píxeles.
```

Seguidamente, se procedió a cargar el conjunto de datos de entrenamiento y, al mismo tiempo se aplicaron funciones para ajustar el tamaño de las imágenes, el cual debe ser correspondiente a 224 X 224 píxeles. Además, se estableció la cantidad de imágenes a utilizar en cada paso del entrenamiento, para lo cual se definió 32 imágenes, lo que significa que se procesan 32 imágenes a la vez antes de actualizar los pesos del modelo. También se definió que el tipo de tarea realizada por el modelo es “categorical” lo que sugiere que se trabaja una tarea de clasificación de múltiples clases. Estos pasos se muestran en el código a continuación.

```
# Cargar el conjunto de datos de entrenamiento  
conjunto_datos = datagen.flow_from_directory(  
    ruta_datos,  
    target_size=(224, 224), # Ajusta el tamaño de las imágenes  
    según los requisitos del modelo  
    batch_size=32 # Cantidad de imágenes a utilizar en cada paso del  
    entrenamiento  
  
    class_mode='categorical' # Ajusta según el tipo de tarea  
    (categórica, binaria, etc.)  
)
```

6.2.4.3.2. Construcción de arquitectura CNN

La arquitectura utilizada para este modelo corresponde a una arquitectura simple de TensorFlow. Esta arquitectura utiliza varias funciones y capas importantes en la construcción del modelo de reconocimiento de imágenes. A partir del código mostrado a continuación se irá especificando dichas funciones y como ayudan en el proceso de construcción del modelo de CNN.

```
# Definir la arquitectura del modelo  
inputs = tf.keras.Input(shape=(224, 224, 3))
```

```
x = tf.keras.layers.Conv2D(32, (3, 3), activation='relu')(inputs)
x = tf.keras.layers.MaxPooling2D((2, 2))(x)
x = tf.keras.layers.Conv2D(64, (3, 3), activation='relu')(x)
x = tf.keras.layers.MaxPooling2D((2, 2))(x)
x = tf.keras.layers.Flatten()(x)
x = tf.keras.layers.Dense(128, activation='relu')(x)
output = tf.keras.layers.Dense(8, activation='softmax')(x)
```

Inicialmente, con la función “input”, se definió la capa de entrada del modelo, que es la capa donde se introducen las imágenes. La capa de entrada de este modelo tiene una forma (224,224,3) lo que indica que se trabaja con imágenes de 224 X 224 píxeles y 3 canales de color (RGB). Con esto, se calcula el número de neuronas utilizadas en la capa de entrada multiplicando las dimensiones espaciales de la imagen (224x224) por el número de canales de color (3), lo que da como resultado, ((224x224) x 3) = 150,528. Por lo tanto, la primera capa de entrada tiene 150,528 neuronas, una para cada píxel en la imagen RGB. Cada neurona representa un valor de píxel en la imagen de entrada y sirve como la entrada inicial para el modelo.

En continuidad, se agregaron capas convolucionales (Conv2D) y de agrupación (MaxPooling2D) de manera alternada para que el modelo aprendiese características de las imágenes.

- Capas convolucionales (Conv2D en Keras):

Son utilizadas para extraer características relevantes de las imágenes. En este caso, se trata de una capa convolucional de 32 filtros de 3x3 píxeles cada uno, y se utilizó la función de activación “relu” para introducir no linealidad en la red.

Para una mejor comprensión, las capas convolucionales utilizan filtros, también conocidos como kernels, para realizar operaciones de convolución en los datos de entrada. Los filtros actúan como detectores de características. Cada filtro puede aprender a detectar ciertas características en la imagen, como bordes, texturas o patrones específicos. La operación de convolución implica deslizar el filtro sobre la imagen de entrada, multiplicar los valores del filtro por los valores

correspondientes de la región de la imagen y sumar los resultados para producir un solo valor en la salida. Este proceso se repite para diferentes ubicaciones en la imagen. Las capas convolucionales generan mapas de características. Cada mapa de característica es el resultado de aplicar un filtro a la imagen de entrada, representan la presencia o ausencia de características específicas en diferentes partes de la imagen. Ver figura 3.

Las capas convolucionales realizan una detección de características jerárquicas. Las primeras capas tienden a detectar características simples como bordes y colores, mientras que las capas más profundas aprenden a detectar características más complejas y abstractas.

Dentro de la función de convolución se aplicó la función ReLU (Rectified Linear Unit). Esta función introduce no linealidad en la red al reemplazar los valores negativos con cero y mantener los valores positivos sin cambios. Esto ayuda a la red a aprender a representar patrones y relaciones no lineales en los datos.

- Capas de agrupación (MaxPooling2D)

Después de cada capa convolucional, se aplicó una capa de agrupación máxima de 2x2. Esta capa se utiliza para reducir la dimensionalidad de las características extraídas por las capas convolucionales, lo que ayuda a conservar las características más importantes y reduce el costo computacional.

El MaxPooling divide la imagen de entrada en regiones pequeñas y selecciona el valor máximo (el píxel más brillante) de cada región. Al reducir el número de píxeles en cada región seleccionando solo el valor máximo, los mapas de características son más pequeños y contienen menos información, pero conservan las características más importantes. MaxPooling permite que la red se enfocue en características locales y específicas en lugar de considerar cada detalle en la imagen.

Estos bloques de capas convolucionales y de agrupación se repiten dos veces para aprender representaciones cada vez más abstractas de las imágenes de enfermedades de piel.

Luego de aplicar las capas de convolución y agrupación, se utilizó una capa de aplanamiento llamada “Flatten”. Esta capa se emplea para transformar las representaciones 2D en un vector 1D, lo cual es necesario antes de conectar las características a las capas densamente conectadas.

La capa Flatten permite conectar los resultados de las capas convolucionales (que se enfocan en la extracción de características) a las capas densamente conectadas (que realizan la clasificación final). Las capas densamente conectadas requieren una entrada unidimensional, y la función Flatten en Keras es una capa que se utiliza para transformar datos 2D en un vector 1D. Esto es necesario fundamentalmente para conectar las capas convolucionales con las capas densamente conectadas en una red neuronal, lo que posibilita la realización de tareas de clasificación en el espacio de características extraídas.

Luego de aplicar la capa de aplanamiento “Flatten”, se procedió con la función “Dense”. La cual crea una capa completamente conectada o densamente conectada en una red neuronal (también conocida como capa oculta). Se añadio una capa densamente conectada con 128 neuronas y se utilizó la función de activación “relu”. Esta capa tiene como objetivo aprender relaciones más complejas entre las características aplanadas. Cada neurona en una capa densa recibe entradas de todas las neuronas de la capa anterior y envía salidas a todas las neuronas de la capa siguiente. Esta capa se considera también capa oculta porque realiza cálculos y aprende representaciones de características más complejas antes de la clasificación final.

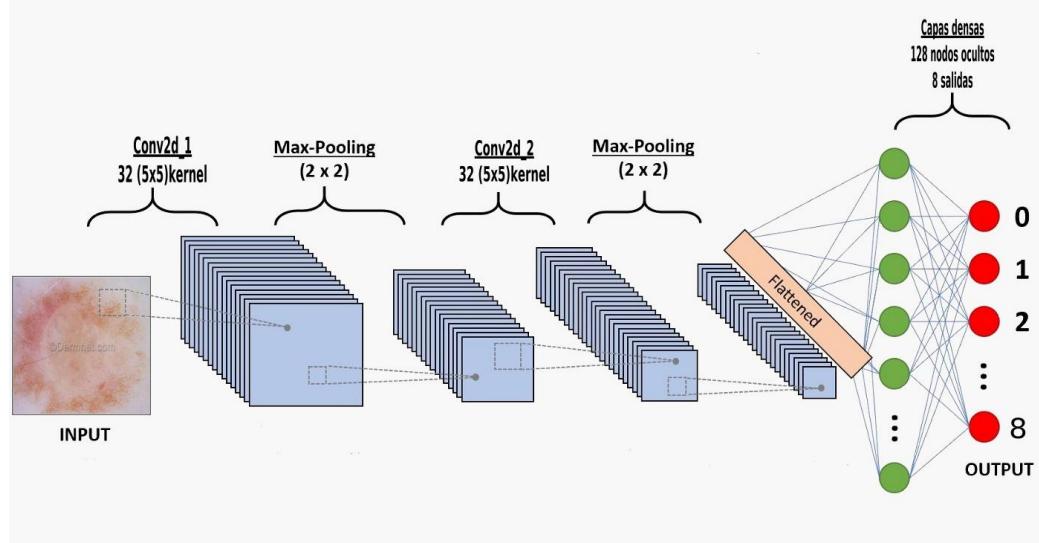
Para la última capa se utilizó nuevamente la función “Dense”, definiendo una capa densa de salida con 8 neuronas, ya que hay 8 clases de salida en este proyecto, junto con la función de activación “Softmax”, la cual calcula las probabilidades de pertenencia a cada una de las 8 clases, lo que facilita la asignación de una clase a una imagen de enfermedad de piel.

En síntesis, para la construcción de la arquitectura de la red neuronal convolucional CNN inicialmente se definieron las capas convolucionales para extraer características de las

imágenes. A estas capas les siguieron las capas de agrupación, que reducen la dimensionalidad de las características. Luego, las características extraídas se aplanaron y pasaron a través de capas densamente conectadas, donde se aprenden relaciones más complejas. Finalmente, las características se dirigen a la capa de salida, que realiza la clasificación de las imágenes de enfermedades de piel. La Figura 6 muestra una representación gráfica de la arquitectura de red construida para este modelo de CNN.

Figura 6.

Arquitectura de red del modelo CNN



Nota: Obtenida por elaboración propia.

6.2.4.3.3. Creación y Compilación del modelo de CNN

Para la creación del modelo de CNN se utilizó la función “tf.keras.Model”, como parámetros de la función se especifica la capa de entrada y la capa de salida definidas en la arquitectura del modelo, como se muestra en la línea de código a continuación.

```
# Crear el modelo
model = tf.keras.Model(inputs=inputs, outputs=output)
```

Luego, el modelo se compilo utilizando el método “compile”, donde se configuraron los siguientes parámetros para el entrenamiento:

- Se eligió el algoritmo de optimización “Adam”, que se adapta de manera dinámica a la tasa de aprendizaje para cada parámetro de la red, acumula el momentum³ de los gradientes y aplica regularización para facilitar el entrenamiento eficiente de la red neuronal.
- La función de pérdida seleccionada fue “categorical_crossentropy”, ya que el modelo trabaja con múltiples clases en la clasificación.
- Como métrica de evaluación del rendimiento del modelo durante el entrenamiento. Se utilizó “Accuracy” para calcular la precisión del modelo.

El código de compilación se ve de la siguiente manera:

```
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
```

6.2.4.3.4. Entrenamiento del modelo CNN

El entrenamiento del modelo, cada paso de entrenamiento implica tomar un lote de datos del conjunto de datos de entrenamiento, calcular el gradiente de la función de pérdida con respecto a los pesos del modelo y luego actualizar los pesos utilizando el algoritmo de optimización Adam.

El objetivo de entrenar el modelo en múltiples épocas es permitir que el modelo ajuste sus pesos a medida que ve el conjunto de datos varias veces. Esto ayuda a mejorar el rendimiento del modelo al aprender patrones generales en los datos en lugar de simplemente memorizarlos. El código de compilación se ve de la siguiente manera:

```
model.fit(conjunto_datos, epochs=10)
```

6.2.4.3.5. Guardado del modelo entrenado

Después de construir y entrenar el modelo, se llevó a cabo la siguiente secuencia de acciones:

³ Es una especie de inercia, que ayuda a acelerar el proceso de optimización.

1. El modelo entrenado se guardó en formato “SavedModel”. Esto permite almacenar el modelo y sus pesos en un formato que se puede recuperar más tarde para su uso o evaluación adicional.
2. Posteriormente, se cargó el modelo guardado para realizar tareas adicionales, como evaluación o inferencia en nuevas imágenes.
3. Finalmente, el modelo se convirtió al formato TensorFlow Lite. Esta conversión era esencial ya que el modelo fue integrado en la aplicación móvil DermaDI, ya que TensorFlow Lite es una versión de TensorFlow optimizada para la implementación en dispositivos móviles y otros sistemas con recursos limitados.

6.2.4.4. Evaluar el modelo

En este apartado, la evaluación del modelo se centró en la aplicación de los criterios de evaluación predefinidos, enfocados en los objetivos de minería de datos del proyecto. De acuerdo con el plan de pruebas, se evaluó el rendimiento del modelo utilizando el conjunto de datos reservado para pruebas. Esto se hizo para verificar como se desempeña el modelo en datos que no había visto durante el entrenamiento.

En la sección “6.2.1.3. Objetivos de la minería de datos” de este documento se definieron los criterios de evaluación a utilizar y se explicó sobre su importancia en la evaluación del modelo. Por tanto, en esta sección se presentan directamente los resultados obtenidos al aplicar dichas métricas.

6.2.4.4.1. Evaluación global del modelo

La evaluación global del modelo arroja los siguientes resultados:

- Exactitud (Accuracy): 0.9073333333333333. La exactitud representa la proporción de predicciones correctas en relación con todas las predicciones. En este caso, el modelo tiene una exactitud del 90.73%, lo que significa que aproximadamente el 90.73% de las predicciones son correctas en el conjunto de datos evaluado.

- Precisión (Precision): 0.9137056854969571. La precisión mide la proporción de verdaderos positivos entre todas las predicciones positivas, En este caso, el modelo tiene una precisión del 91.73%, lo que significa que aproximadamente el 91.73% de las predicciones positivas realizadas por el modelo son realmente correctas.
- Exhaustividad (Recall): 0.9073563957625259. La exhaustividad mide la proporción de verdaderos positivos entre todos los casos positivos reales. El modelo tiene una exhaustividad del 90.74%, lo que indica que aproximadamente el 90.74% de los casos positivos reales se han identificado correctamente por el modelo.
- Puntuación F1 (F1 Score): 0.9087731897878228. La puntuación F1 es una métrica que combina precisión y exhaustividad en una sola métrica. En este caso, el modelo tiene una puntuación F1 del 90.88%, lo que indica un buen equilibrio entre la precisión y la exhaustividad en las predicciones.
- Matriz de confusión. Proporciona información detallada sobre como el modelo ha clasificado las muestras en cada clase. Cada fila representa la clase real, y cada columna la clase predicha. Los valores en la diagonal principal (de izquierda a derecha, superior a inferior) representan las predicciones correctas, mientras que los valores fuera de la diagonal principal son errores de clasificación. Por ejemplo, en la primera fila, se observa que la clase 1 ha sido clasificada correctamente 366 veces, pero ha habido algunos errores en las otras clases. Del mismo modo, en la segunda fila, la clase 2 ha sido clasificada correctamente 348 veces.

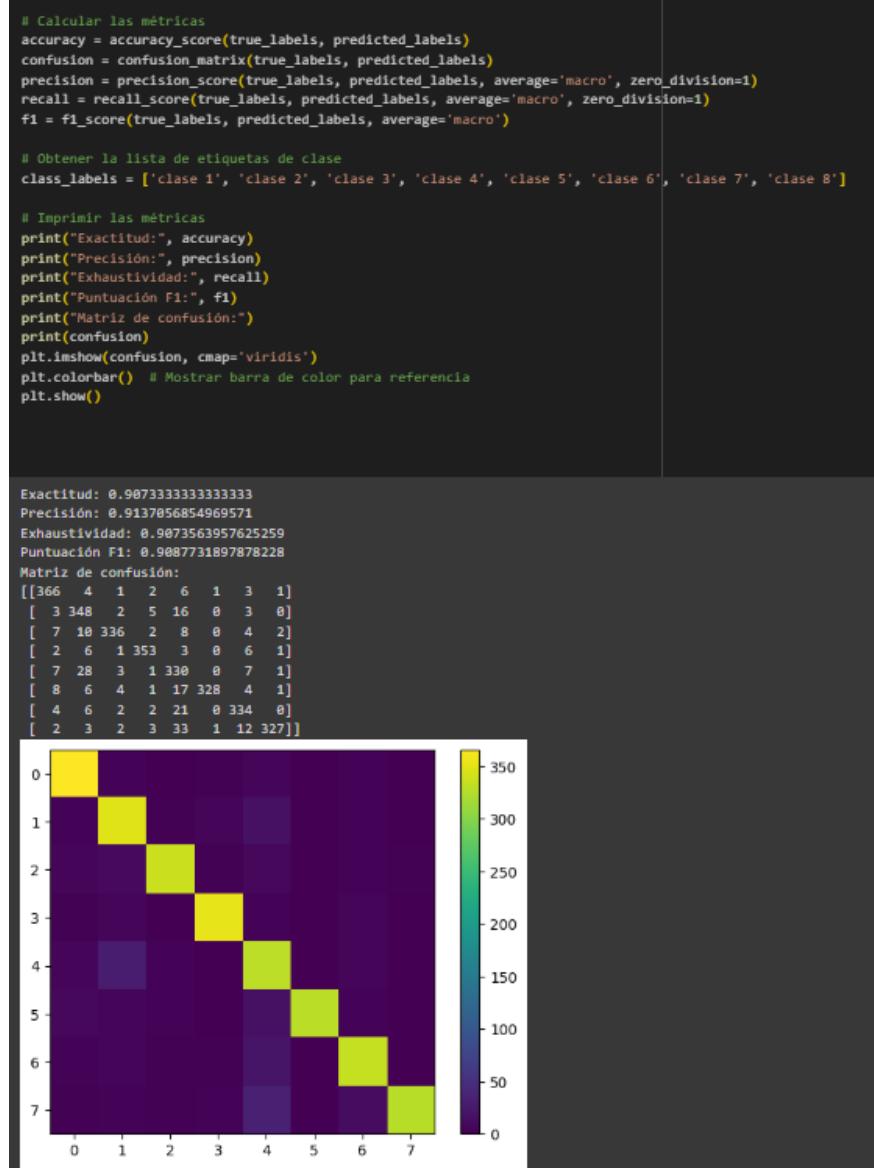
Estos resultados demuestran que el modelo tiene un buen rendimiento general en la clasificación de imágenes, con una alta exactitud, precisión y exhaustividad, así como una puntuación F1 equilibrada.

La figura 7 muestra las funciones utilizadas para el cálculo de las métricas y los resultados descritos anteriormente, tal como fueron devueltos tras la ejecución de las pruebas de validación. La matriz de confusión también es útil para identificar patrones de errores del

modelo y evaluar su rendimiento en cada clase, de lo cual se mostrarán resultados posteriormente.

Figura 7

Cálculo y resultados de métricas de evaluación global del modelo CNN



Nota: Se visualiza la matriz de confusión utilizando una gráfica de calor (heatmap) que resalta las relaciones entre las clases y las predicciones. Figura obtenida de la ejecución del cálculo de matriz de confusión.

A partir de la matriz de confusión, se calculan métricas en función de cada clase. Estas métricas proporcionan información sobre el rendimiento del modelo. Entre estas métricas de evaluación se tiene:

- Verdaderos Positivos (TP)
- Falsos Positivos (FP)
- Falsos Negativos (FN)
- Verdaderos Negativos (TN)

Estas métricas son esenciales para evaluar el rendimiento del modelo de clasificación y cómo se comporta en diferentes situaciones. Ayudan a evaluar que tan bien el modelo distingue entre clases positivas y negativas, y a identificar áreas de mejora en el rendimiento del modelo.

A continuación, se describen los resultados en promedio a nivel global del modelo para estas métricas.

Promedio TP: 340.25
Promedio FP: 34.75
Promedio FN: 34.75
Promedio TN: 2590.25

- Promedio TP (Verdaderos Positivos): 340.25

Esto indica que, en promedio, el modelo clasifica correctamente alrededor de 340.25 ejemplos positivos como positivos en todas las clases del problema.

- Promedio FP (Falsos Positivos): 34.75

En promedio, el modelo clasifica incorrectamente alrededor de 34.75 ejemplos negativos como positivos en todas las clases. Esto significa que en promedio hay aproximadamente 34.75 errores de falsos positivos.

- Promedio FN (Falsos Negativos): 34.75

En promedio, el modelo clasifica incorrectamente alrededor de 34.75 ejemplos positivos como negativos en todas las clases. Esto significa que en promedio hay aproximadamente 34.75 errores de falsos negativos.

- Promedio TN (Verdaderos Negativos): 2590.25

Esto indica que, en promedio, el modelo clasifica correctamente alrededor de 2590.25 ejemplos negativos como negativos en todas las clases del problema.

6.2.4.4.2. Evaluación por clase del modelo

A continuación, se realiza la evaluación de las métricas de Precisión, Exhaustividad y F1-Score, para cada una de las clases del modelo de reconocimiento. Importante reiterar que el modelo de reconocimiento de enfermedades de piel de este proyecto trabaja con 8 clases o categorías. Ver figura 8.

Figura 8

Código para el cálculo de métricas de evaluación por clase

```
import numpy as np

# Matriz de confusión
confusion_matrix = np.array([[366, 4, 1, 2, 6, 1, 3, 1],
                            [3, 348, 2, 5, 16, 0, 3, 0],
                            [7, 10, 336, 2, 8, 0, 4, 2],
                            [2, 6, 1, 353, 3, 0, 6, 1],
                            [7, 28, 3, 1, 330, 0, 7, 1],
                            [8, 6, 4, 1, 17, 328, 4, 1],
                            [4, 6, 2, 2, 21, 0, 334, 0],
                            [2, 3, 2, 3, 33, 1, 12, 327]])]

# Calcula la precisión por clase
num_classes = confusion_matrix.shape[0]
precision_per_class = []

for i in range(num_classes):
    true_positives = confusion_matrix[i, i]
    false_positives = np.sum(confusion_matrix[:, i]) - true_positives
    precision = true_positives / (true_positives + false_positives)
    precision_per_class.append(precision)

# Calcula el recuerdo (recall) por clase
recall_per_class = []

for i in range(num_classes):
    true_positives = confusion_matrix[i, i]
    false_negatives = np.sum(confusion_matrix[i, :]) - true_positives
    recall = true_positives / (true_positives + false_negatives)
    recall_per_class.append(recall)

# Calcula el puntaje F1 por clase
f1_score_per_class = []

for i in range(num_classes):
    precision = precision_per_class[i]
    recall = recall_per_class[i]
    f1_score = 2 * (precision * recall) / (precision + recall)
    f1_score_per_class.append(f1_score)

# Imprime las métricas por clase
for i in range(num_classes):
    print("Clase", i + 1)
    print("Precisión:", precision_per_class[i])
    print("Recuerdo:", recall_per_class[i])
    print("Puntaje F1:", f1_score_per_class[i])
    print()
```

Nota. Figura obtenida del código fuente para el desarrollo del modelo de CNN.

- **Clase 1:** Precisión: 0.9173 → 91.73%; Exhaustividad: 0.9531 → 95.31%; F1-Score: 0.9349 → 93.49%. Estos valores indican que el modelo tiene un alto nivel de precisión para la clase 1, lo que significa que la mayoría de las predicciones positivas para esta clase son correctas. Además, el modelo tiene un buen recuerdo para esta clase, lo que indica que identifica correctamente la mayoría de los casos positivos reales. El puntaje F1, que combina precisión y recuerdo, también es alto y sugiere un rendimiento general sólido para esta clase.
- **Clase 2:** Precisión: 0.8467 → 84.76%; Exhaustividad: 0.9231 → 92.31%; F1-Score: 0.8832 → 88.32%. En la Clase 2, el modelo tiene una precisión menor que en la Clase 1, lo que sugiere que comete más falsos positivos para esta clase. Sin embargo, el recuerdo es bastante alto, lo que indica que el modelo identifica correctamente la mayoría de los casos positivos reales. El puntaje F1 es moderado y refleja un equilibrio entre precisión y recuerdo.
- **Clase 3:** Precisión: 0.9573 → 95.73%; Exhaustividad: 0.9106 → 91.06%; F1-Score: 0.9333 → 93.33%. Para la Clase 3, el modelo muestra una alta precisión, lo que significa que la mayoría de las predicciones positivas son correctas. El cálculo del recuerdo indica que el modelo identifica la mayoría de los casos positivos reales. El puntaje F1 es alto, lo que sugiere un rendimiento sólido en esta clase.
- **Clase 4:** Precisión: 0.9566 → 95.66%; Exhaustividad: 0.9489 → 94.89%; F1-Score: 0.9528 → 95.28%. En la Clase 4, el modelo tiene una alta precisión y un alto recuerdo. Esto indica que el modelo realiza predicciones precisas y captura la mayoría de los casos positivos reales. El puntaje F1 es muy alto, lo que indica un excelente rendimiento en esta clase.
- **Clase 5:** Precisión: 0.7604 → 76.04%; Exhaustividad: 0.8753 → 87.53%; F1-Score: 0.8138 → 81.38%. La Clase 5 muestra una precisión más baja en comparación con las clases anteriores, lo que sugiere que el modelo comete más falsos positivos para esta clase. Sin embargo, el recuerdo sigue siendo

razonablemente alto, lo que indica que se identifican la mayoría de los casos positivos reales. El puntaje F1 es moderado.

- **Clase 6:** Precisión: 0.9940 → 99.40%; Exhaustividad: 0.8889 → 88.89%; F1-Score: 0.9385 → 93.85%. Para la Clase 6, el modelo muestra una precisión excepcionalmente alta, indicando que la mayoría de las predicciones positivas son correctas. El cálculo del recuerdo indica una buena identificación de casos positivos reales. El puntaje F1 es alto, reflejando un rendimiento sobresaliente en esta clase.
- **Clase 7:** Precisión: 0.8954 → 89.54%; Exhaustividad: 0.9051 → 90.51%; F1-Score: 0.9003 → 90.03%. En la Clase 7, el modelo tiene una precisión y un recuerdo razonablemente altos, lo que indica que realiza predicciones precisas y captura la mayoría de los casos positivos reales. El puntaje F1 es moderado y sugiere un rendimiento sólido en esta clase.
- **Clase 8:** Precisión: 0.9820 → 98.20%; Exhaustividad: 0.8538 → 85.38%; F1-Score: 0.9134 → 91.34%. Finalmente, para la Clase 8, el modelo muestra una alta precisión, lo que indica que la mayoría de las predicciones positivas son correctas. El recuerdo es un poco más bajo, lo que sugiere que no identifica todos los casos positivos reales. El puntaje F1 es alto, lo que refleja un buen equilibrio entre precisión y recuerdo en esta clase.

Estos resultados indican que el modelo tiene un rendimiento general sólido en la mayoría de las clases, con altas precisiones y recuerdos en muchas de ellas.

Figura 9

Resultados de cálculo de métricas de evaluación por clase

```
Clase 1
Precisión: 0.9172932330827067
Recuerdo: 0.953125
Puntaje F1: 0.9348659003831417

Clase 2
Precisión: 0.8467153284671532
Recuerdo: 0.9230769230769231
Puntaje F1: 0.883248730964467

Clase 3
Precisión: 0.9572649572649573
Recuerdo: 0.9105691056910569
Puntaje F1: 0.9333333333333333

Clase 4
Precisión: 0.9566395663956639
Recuerdo: 0.9489247311827957
Puntaje F1: 0.9527665317139001

Clase 5
Precisión: 0.7603686635944701
Recuerdo: 0.8753315649867374
Puntaje F1: 0.813810110974106

Clase 6
Precisión: 0.9939393939393939
Recuerdo: 0.8888888888888888
Puntaje F1: 0.938483547925608

Clase 7
Precisión: 0.8954423592493298
Recuerdo: 0.9051490514905149
Puntaje F1: 0.9002695417789759

Clase 8
Precisión: 0.9819819819819819
Recuerdo: 0.8537859007832899
Puntaje F1: 0.9134078212290504
```

Nota. Figura obtenida del código fuente para el desarrollo del modelo de CNN.

Con la obtención de la matriz de confusión, pueden calcularse las métricas de modo específico para cada clase. Para la evaluación de estas métricas es necesario indicar la matriz de confusión correspondiente al modelo entrenado, como se muestra en la figura 7.

En cada clase, los Verdaderos Positivos (TP) representan la cantidad de ejemplos positivos correctamente clasificados. Los Falsos Positivos (FP) representan la cantidad de ejemplos negativos que fueron incorrectamente clasificados como positivos. Los Falsos Negativos

(FN) representan la cantidad de ejemplos positivos que fueron incorrectamente clasificados como negativos. Los Verdaderos Negativos (TN) representan la cantidad de ejemplos negativos correctamente clasificados.

Resultados de las métricas correspondientes a:

- Verdaderos Positivos (TP)
- Falsos Positivos (FP)
- Falsos Negativos (FN)
- Verdaderos Negativos (TN)

A partir del cálculo de estas métricas, es posible evaluar como se comporta el modelo en términos de errores y aciertos para cada clase individualmente. A continuación, los resultados de la evaluación.

Clase 1

- Verdaderos Positivos (TP): 366.0
- Falsos Positivos (FP): 33.0
- Falsos Negativos (FN): 18.0
- Verdaderos Negativos (TN): 2583.0

El modelo tiene un alto número de verdaderos positivos (366.0) en esta clase, lo que significa que clasifica correctamente la mayoría de los casos positivos. El número de falsos positivos (33.0) y falsos negativos (18.0) es relativamente bajo en comparación con los verdaderos positivos, lo que indica un buen equilibrio. En general, el modelo funciona bien para esta clase.

Clase 2

- Verdaderos Positivos (TP): 348.0
- Falsos Positivos (FP): 63.0
- Falsos Negativos (FN): 29.0
- Verdaderos Negativos (TN): 2560.0

Similar a la Clase 1, el modelo mostro un buen rendimiento con un alto número de verdaderos positivos (348.0). Aunque el número de falsos positivos (63.0) es mayor que en la Clase 1, el equilibrio general es positivo. El número de falsos negativos (29.0) también es relativamente bajo.

Clase 3

- Verdaderos Positivos (TP): 336.0
- Falsos Positivos (FP): 15.0
- Falsos Negativos (FN): 33.0
- Verdaderos Negativos (TN): 2616.0

El modelo tiene un alto número de verdaderos positivos (336.0) en esta clase, pero el número de falsos positivos (15.0) es aún menor que en las clases anteriores, lo que es muy positivo. El número de falsos negativos (33.0) es un poco más alto, pero en general, el modelo funciona bien para esta clase.

Clase 4

- Verdaderos Positivos (TP): 353.0
- Falsos Positivos (FP): 16.0
- Falsos Negativos (FN): 19.0
- Verdaderos Negativos (TN): 2612.0

Al igual que en las clases anteriores, el modelo muestra un buen desempeño en términos de verdaderos positivos (353.0) y falsos positivos (16.0). El número de falsos negativos (19.0) es bajo, lo que indica que el modelo clasifica correctamente la mayoría de los casos positivos en esta clase.

Clase 5

- Verdaderos Positivos (TP): 330.0
- Falsos Positivos (FP): 104.0

- Falsos Negativos (FN): 47.0
- Verdaderos Negativos (TN): 2519.0

Esta clase presenta un desafío, ya que el número de falsos positivos (104.0) es más alto que en las clases anteriores. Sin embargo, el número de verdaderos positivos (330.0) es aún mayor, lo que indica que el modelo tiene un buen rendimiento en la identificación de casos positivos. Se podría buscar reducir los falsos positivos.

Clase 6

- Verdaderos Positivos (TP): 328.0
- Falsos Positivos (FP): 2.0
- Falsos Negativos (FN): 41.0
- Verdaderos Negativos (TN): 2629.0

El modelo muestra un rendimiento excepcional en términos de verdaderos positivos (328.0) en esta clase. El número de falsos positivos (2.0) es muy bajo, lo que es positivo. Los falsos negativos (41.0) son un poco más altos, pero en general, el modelo funciona muy bien en esta clase.

Clase 7

- Verdaderos Positivos (TP): 334.0
- Falsos Positivos (FP): 39.0
- Falsos Negativos (FN): 35.0
- Verdaderos Negativos (TN): 2592.0

El modelo tiene un alto número de verdaderos positivos (334.0) en esta clase, pero el número de falsos positivos (39.0) es un poco más alto de lo deseado. Los falsos negativos (35.0) también son un poco más altos, lo que indica que hay margen de mejora.

Clase 8

- Verdaderos Positivos (TP): 327.0

- Falsos Positivos (FP): 6.0
- Falsos Negativos (FN): 56.0
- Verdaderos Negativos (TN): 2611.0

En esta clase, el número de verdaderos positivos (327.0) es bueno, pero el número de falsos positivos (6.0) y falsos negativos (56.0) muestra un desafío en la clasificación. Aquí, es importante trabajar en la reducción de los falsos positivos y falsos negativos.

En términos generales, el modelo demuestra un buen rendimiento en la mayoría de las clases, con algunos desafíos en las clases 5, 7 y 8.

A continuación, se muestran los resultados de la evaluación de las técnicas de dominio.

- Clase 1:
 - ✓ Sensibilidad: 0.953125. El modelo tiene una alta capacidad para identificar correctamente los ejemplos positivos de esta clase.
 - ✓ Especificidad: 0.9873853211009175. El modelo también es bueno en la identificación de ejemplos negativos de esta clase.
 - ✓ Valor Predictivo Positivo: 0.9172932330827067. La proporción de predicciones positivas que son verdaderos positivos es alta.
 - ✓ Valor Predictivo Negativo: 0.9930795847750865. La proporción de predicciones negativas que son verdaderos negativos es alta.

Estos resultados indican que el modelo es especialmente bueno en la clasificación de la Clase 1, con altas tasas de verdaderos positivos y verdaderos negativos.

- Clase 2:
 - ✓ Sensibilidad: 0.9230769230769231. El modelo tiene una buena capacidad para identificar ejemplos positivos de esta clase.
 - ✓ Especificidad: 0.9759817003431186. El modelo también es bueno en la identificación de ejemplos negativos de esta clase.

- ✓ Valor Predictivo Positivo: 0.8467153284671532. La proporción de predicciones positivas que son verdaderos positivos es un poco menor.
- ✓ Valor Predictivo Negativo: 0.988798764001545. La proporción de predicciones negativas que son verdaderos negativos es alta.

En la Clase 2, el modelo tiene un buen rendimiento en general, aunque la precisión de las predicciones positivas podría ser mejorada.

- Clase 3:
 - ✓ Sensibilidad: 0.9105691056910569. El modelo tiene una buena capacidad para identificar ejemplos positivos de esta clase.
 - ✓ Especificidad: 0.9942987457240593. El modelo es excelente en la identificación de ejemplos negativos de esta clase.
 - ✓ Valor Predictivo Positivo: 0.9572649572649573. La proporción de predicciones positivas que son verdaderos positivos es alta.
 - ✓ Valor Predictivo Negativo: 0.9875424688561721. La proporción de predicciones negativas que son verdaderos negativos es alta.

Para la Clase 3, el modelo tiene un rendimiento muy bueno tanto en la identificación de ejemplos positivos como negativos.

El análisis continúa de manera similar para las otras clases. Ver figura 9. En general, se puede determinar que el modelo tiene un rendimiento sólido en la mayoría de las clases, con altas sensibilidades y especificidades. Sin embargo, en algunas clases, la precisión de las predicciones positivas puede ser mejorada.

Figura 10

Resultados de evaluación de métricas de dominio

```
Clase 1
Sensibilidad: 0.953125
Especificidad: 0.9873853211009175
Valor predictivo positivo: 0.9172932330827067
Valor predictivo negativo: 0.9930795847750865

Clase 2
Sensibilidad: 0.9230769230769231
Especificidad: 0.9759817003431186
Valor predictivo positivo: 0.8467153284671532
Valor predictivo negativo: 0.988798764001545

Clase 3
Sensibilidad: 0.9105691056910569
Especificidad: 0.9942987457240593
Valor predictivo positivo: 0.9572649572649573
Valor predictivo negativo: 0.9875424688561721

Clase 4
Sensibilidad: 0.9489247311827957
Especificidad: 0.9939117199391172
Valor predictivo positivo: 0.9566395663956639
Valor predictivo negativo: 0.9927784112504751

Clase 5
Sensibilidad: 0.8753315649867374
Especificidad: 0.9603507434235609
Valor predictivo positivo: 0.7603686635944701
Valor predictivo negativo: 0.9816835541699143

Clase 6
Sensibilidad: 0.8888888888888888
Especificidad: 0.9992398327632079
Valor predictivo positivo: 0.9939393939393939
Valor predictivo negativo: 0.9846441947565543

Clase 7
Sensibilidad: 0.9051490514905149
Especificidad: 0.9851767388825542
Valor predictivo positivo: 0.8954423592493298
Valor predictivo negativo: 0.9866768176627332

Clase 8
Sensibilidad: 0.8537859007832899
Especificidad: 0.9977072984333206
Valor predictivo positivo: 0.9819819819819819
Valor predictivo negativo: 0.979002624671916
```

Nota. Figura obtenida del código fuente para el desarrollo del modelo de CNN.

6.2.5. Evaluation (Evaluación de los resultados)

Esta fase implica una evaluación minuciosa del modelo construido y una comparación con los objetivos de negocio establecidos. Es crucial identificar cualquier problema importante que no se haya considerado adecuadamente. Al final de esta fase, se espera tomar una decisión sobre la aplicación de los resultados obtenidos. (singular, s.f.). En esta fase se revisó la evaluación realizada al modelo y se relacionó con el cumplimiento de los objetivos.

6.2.5.1. Evaluar los resultados

Los resultados de las evaluaciones del modelo son fundamentales para comprender su rendimiento y como contribuye al logro de los objetivos del negocio (proyecto). En este contexto, se presentan los resultados clave de las evaluaciones realizadas y se analiza su relación con los objetivos del proyecto.

- Precisión Global: La precisión global del modelo se evaluó en un 90.73%. Esto significa que el modelo clasifica correctamente el 90.73% de todas las instancias. Este resultado es esencial para cumplir con el objetivo de construir un modelo de CNN con un alto porcentaje de precisión.
- Métricas Clase por Clase: Las métricas por clase revelan como el modelo se desempeña en la clasificación de clases específicas. Cada clase representa un producto o categoría de interés para el negocio. Se observó que el modelo logró una alta precisión y exhaustividad en la mayoría de las clases, lo que indica una capacidad confiable para identificar y clasificar las distintas enfermedades de piel contenidas en el modelo.
- Matriz de Confusión: Al analizar la matriz de confusión, se identificó que el modelo tiene una tendencia a cometer errores en la clasificación de dos categorías específicas. Esto proporciona información valiosa sobre dónde se deben concentrar los esfuerzos de mejora y optimización. Estas son la clase 5 y la clase 8.
- Sensibilidad y Especificidad: La sensibilidad (Recall) y especificidad son métricas críticas para comprender como el modelo afecta la detección de casos positivos y negativos. El modelo logró una alta sensibilidad, lo que significa que es eficiente en la identificación de casos positivos.
- Valor Predictivo Positivo y Negativo: Estas métricas miden la precisión del modelo en la predicción de casos positivos y negativos.

✓ VPP (Valor Predictivo Positivo):

Para las clases 1, 3, 4, 6 y 8, el VPP es alto, lo que indica que cuando el modelo predice que un producto pertenece a una de estas categorías, es probable que sea una predicción precisa. Por ejemplo, para la Clase 6, el VPP es 0.994, lo que sugiere que cuando el modelo predice que un producto pertenece a la Clase 6, es muy probable que sea correcto en un 99.4% de las ocasiones. Sin embargo, para las clases 2, 5 y 7, el VPP es más bajo, lo que significa que el modelo tiende a cometer errores al predecir estas categorías. Por ejemplo, para la Clase 5, el VPP es 0.760, lo que indica que cuando el modelo predice que un producto pertenece a la Clase 5, tiene una precisión del 76%, pero aún comete errores en el 24% de las ocasiones.

✓ VPN (Valor Predictivo Negativo):

En general, el VPN es alto para todas las clases, lo que indica que cuando el modelo predice que un producto no pertenece a una categoría en particular, es muy probable que sea una predicción precisa. Esto sugiere que el modelo es efectivo para identificar productos que no pertenecen a las categorías específicas.

El objetivo fundamental del proyecto era desarrollar un modelo e CNN altamente preciso para el reconocimiento de enfermedades de piel. Esto se traduce en la capacidad de diagnosticar con precisión las enfermedades y minimizar los falsos diagnósticos.

El modelo de CNN ha demostrado un alto grado de precisión en la clasificación de enfermedades de la piel. La exactitud general del modelo es del 90.73%, lo que indica que se cumple con el criterio de éxito de obtener un modelo resultante con una precisión de más del 90%, esto demuestra que el modelo es efectivo en su tarea de reconocimiento.

6.2.5.2. Revisar el proceso

En este apartado, se llevó a cabo una revisión exhaustiva del proceso de minería de datos que se siguió a lo largo del proyecto de implementación de un modelo de red neuronal convolucional (CNN por sus siglas en inglés) para el reconocimiento de enfermedades de la piel. El objetivo de esta revisión es evaluar la efectividad del proceso y determinar las lecciones aprendidas.

6.2.5.2.1. Metodología de Minería de Datos

La metodología seguida durante el proceso de minería de datos se basó y guio en el modelo CRISP-DM, que comprende las siguientes etapas:

- ✓ Comprensión del Negocio

En esta etapa, se definieron los objetivos del negocio y se establecieron las metas y criterios de éxito relacionadas con el reconocimiento de enfermedades de la piel.

- ✓ Comprensión de los Datos

Se realizó una exploración detallada de los datos disponibles, incluyendo la recopilación de imágenes de enfermedades de la piel y la preparación de un conjunto de datos de entrenamiento y prueba.

- ✓ Preparación de los datos

Se llevaron a cabo tareas de preprocesamiento, como el escalado y la normalización de imágenes y la distribución del data set, en conjuntos de entrenamiento y prueba.

- ✓ Modelado

Se implementó el modelo de red neuronal convolucional (CNN) para el reconocimiento de enfermedades de la piel y se optimizaron los hiperparámetros para mejorar el rendimiento del modelo.

- ✓ Evaluación

Se evaluó el modelo utilizando métricas de rendimiento, como la precisión, la sensibilidad y el Valor Predictivo Positivo (VPP), para determinar su eficacia en el diagnóstico de enfermedades de la piel.

- ✓ Despliegue

El modelo entrenado se desplegó en un formato que permite su uso en aplicaciones médicas, lo que incluyó la conversión a TensorFlow Lite para su ejecución en dispositivos móviles.

6.2.5.2.2. Resultados de la Revisión

La revisión del proceso de minería de datos ha arrojado los siguientes resultados clave:

✓ Cumplimiento de etapas:

Se ha cumplido con éxito cada una de las etapas de la metodología CRISP-DM. La comprensión del negocio llevó a la definición de objetivos claros, la comprensión de los datos permitió la creación de un conjunto de datos adecuado y el modelado condujo al desarrollo de un CNN efectivo.

✓ Calidad de los datos:

La calidad de los datos ha demostrado ser adecuada para la tarea de diagnóstico de enfermedades de la piel, con un conjunto de imágenes representativo y bien preparado.

✓ Eficiencia del modelo:

El modelo de CNN implementado ha demostrado ser altamente eficiente, logrando una precisión del 90.73% en la clasificación de enfermedades de la piel.

✓ Impacto en el trabajo monográfico:

El proyecto ha tenido un impacto positivo al cumplir con los objetivos planteados tras aumentar la precisión del diagnóstico.

6.2.5.2.3. Lecciones aprendidas.

Durante el proceso de minería de datos, se han identificado las siguientes lecciones importantes:

- La cantidad y calidad de los datos es fundamental para el éxito del modelo.
- Los recursos computacionales dispuestos deben ser altamente eficientes y de alto rendimiento al trabajar con modelos de aprendizaje profundo.
- La selección y optimización de modelos de aprendizaje automático son aspectos críticos del proceso.

6.2.5.2.4. Conclusiones del proceso de revisión del proyecto

La revisión del proceso de minería de datos confirma que se ha cumplido con éxito con los objetivos del proyecto y se han obtenido resultados positivos en términos de precisión y eficacia en el diagnóstico de enfermedades de la piel.

6.2.5.3. Determinar los próximos pasos

Después de completar las actividades de modelado, evaluación del modelo y revisión del proceso, los siguientes pasos se enfocaron en la implementación y puesta en marcha del proyecto. Aquí hay una descripción general de los próximos pasos:

- Implementación del Modelo en la Aplicación Móvil DermaDI: Dado que el modelo de reconocimiento de enfermedades de la piel ha demostrado ser efectivo en las pruebas y evaluaciones, el siguiente paso consistía en integrarlo en la aplicación móvil que se desarrolló para este propósito. Esto implicó la programación de la lógica que permite a los usuarios cargar o tomar fotos, ejecutar el modelo y mostrar los resultados posibles.
- Desarrollo de la Interfaz de Usuario (UI): La aplicación se desarrolló de modo que fuese intuitiva al usuario y de fácil uso.
- Pruebas de Integración: Se realizaron pruebas exhaustivas para asegurarse de que el modelo se integra correctamente con la aplicación móvil. Las pruebas se realizaron desde máquinas virtuales y dispositivos móviles físicos para asegurar su funcionalidad en dispositivos con distintas características técnicas y versiones de Android diversas.
- Generación de la Base de Datos de Enfermedades de Piel: Se creó una base de datos, desde SQLite, donde se almacena la información referente a las enfermedades pertenecientes a las clases contenidas en el modelo, la cual, se integró en la app móvil. Esta base de datos es esencial para proporcionar información detallada sobre las enfermedades identificadas por el modelo.

- Implementación de la Comunicación con los Desarrolladores: Se verifico el hecho de proporcionar datos de contacto con los desarrolladores, para permitirle a los usuarios reportar problemas y hacer sugerencias.
- Pruebas de Usuario: Se lleva a cabo pruebas de usuario en las que personas reales utilizaron la aplicación para verificar su facilidad de uso, eficacia y capacidad para proporcionar diagnósticos precisos. Estas pruebas fueron realizadas por médicos especialistas en Dermatología, del Hospital Central Dermatológico Nacional, Se hizo recopilación de comentarios y pruebas de usabilidad soportadas con formato de satisfacción entregado a los usuarios participantes en las pruebas.
- Documentación del trabajo monográfico: El proyecto se ha documentado a medida que se ha avanzado en cada una de las fases.
- Despliegue de la Aplicación Móvil: La apk, código fuente y toda la documentación relacionada a este trabajo monográfico fue cargada a un repositorio público, para brindar libre acceso a cualquiera que desee hacer uso de la app o bien documentarse sobre el proceso de implantación de redes neuronales convolucionales.

6.2.6. Deployment (Despliegue. Puesta en producción)

La creación del modelo no significa el final del proyecto. Es importante organizar y presentar la información obtenida para poder utilizarla. La complejidad de esta etapa dependerá de las necesidades específicas, pudiendo ser desde un simple informe hasta la implementación de un proceso de análisis de datos regular y automatizado o bien el desarrollo de una herramienta tecnológica que integre el modelo resultante (singular, s.f.).

Bajo esta fase se lleva a cabo el desarrollo de la aplicación móvil DermaDI para la integración del modelo generado, además de la consolidación de la documentación del proyecto.

6.2.6.1. Preparación del entorno de producción

Entendiendo la preparación del entorno de producción, como los pasos previos para el desarrollo de la herramienta DermaDI, se describen a continuación las actividades realizadas en torno a ello.

El computador utilizado para el desarrollo de la app, trabaja con el sistema operativo Fedora, una de las distribuciones de Linux. El primer paso fue la instalación de Package Manager Snap⁴, este como administrador de paquetes. A través de Snapstore se procedió con la descarga e instalación de Android Studio y el paquete SDK flutter, seguido se accedió a la consola de Linux y se inicializó flutter, se ejecutó el comando “flutter doctor” requerido para completar la instalación y configuración de flutter.

Gracias al package manager Snap, Android studio automáticamente detecta a flutter, por lo que solo se instaló el add-on⁵ de flutter en el entorno y se reinició el mismo. Tras el reinicio de Android Studio, el entorno quedó listo para la creación del nuevo proyecto flutter, seguido se instaló las librerías a utilizar para empezar a codificar. Cabe recalcar que flutter trabaja con su propio lenguaje de programación llamado Dart, el cual viene incluido en los paquetes del framework.

Es válido documentar en este apartado que, para brindarle una identidad propia a la aplicación, además, de denominarla con el naming DermaDI como referencia de “Dermatological Diseases Identifier”, se definieron los colores azul y blanco como colores representativos, también, se realizó el diseño de un logo para el uso exclusivo de la app. El diseño del logo se muestra en la figura 11.

Figura 11

Diseño logo DermaDI

⁴Administrador de paquetes, Es un conjunto de servicios integrados que facilitan la descarga, instalación, actualización, eliminación y configuración de paquetes / programas en una computadora.

⁵ subcategoría de los plugins, pequeños programas que añaden otras funciones al programa principal



Nota. El diseño de este logo fue creado por Entre Líneas – Kevin Rojas, diseño gráfico e ilustraciones.

6.2.6.2. Desarrollo y configuración de la solución

Bajo esta actividad se documenta en detalle de cada una de las pantallas contenidas en la app móvil DermaDI, presentando interfaces y las funcionalidades implementadas en la herramienta, para el completo aprovechamiento del modelo de reconocimiento de enfermedades de la piel.

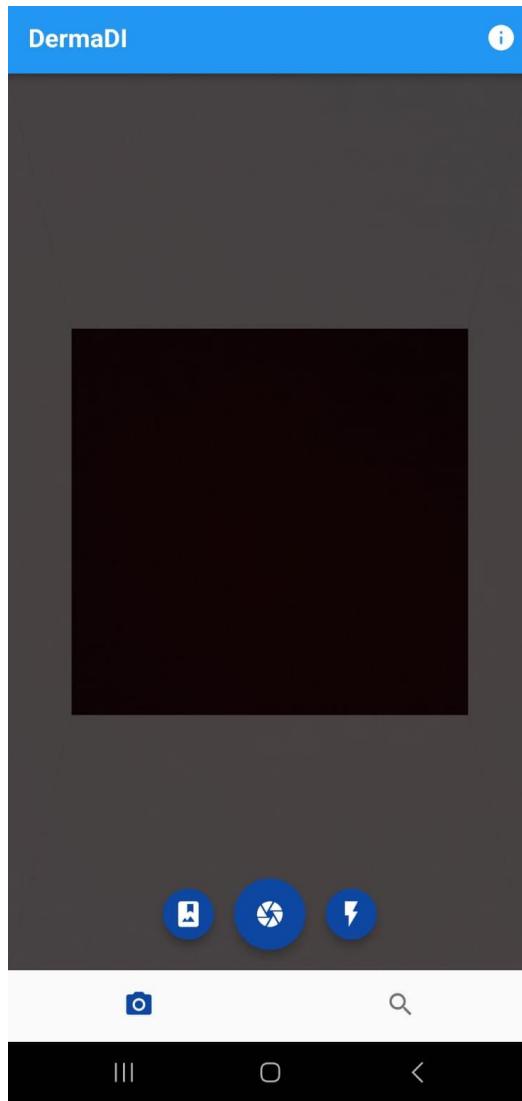
6.2.6.2.1. Diseño de interfaces

Pantalla principal

La pantalla principal de la app, es un acceso directo a la cámara del dispositivo móvil usado, esto para permitirle al usuario poder hacer la tarea de reconocimiento de forma directa. En esta misma pantalla están las funciones para activar o desactivar el flash de cámara para mejorar la captura de la fotografía de la lesión, además, la función de carga de imagen para subir una imagen de la enfermedad a clasificar que se encuentre en la galería del dispositivo.

Figura 12

Pantalla Principal de la app móvil DermaDI



Nota: Obtenida a partir de captura de pantalla de la app móvil DermaDI

Pantalla “Confirmación” (Pantalla de confirmación de imagen a clasificar)

Tras la captura o carga de la imagen a analizar, se muestra la pantalla de confirmación, donde el usuario tiene una vista previa de la imagen a clasificar. Esto para verificar superficialmente la calidad de la imagen.

Figura 13

Pantalla de confirmación de imagen a clasificar



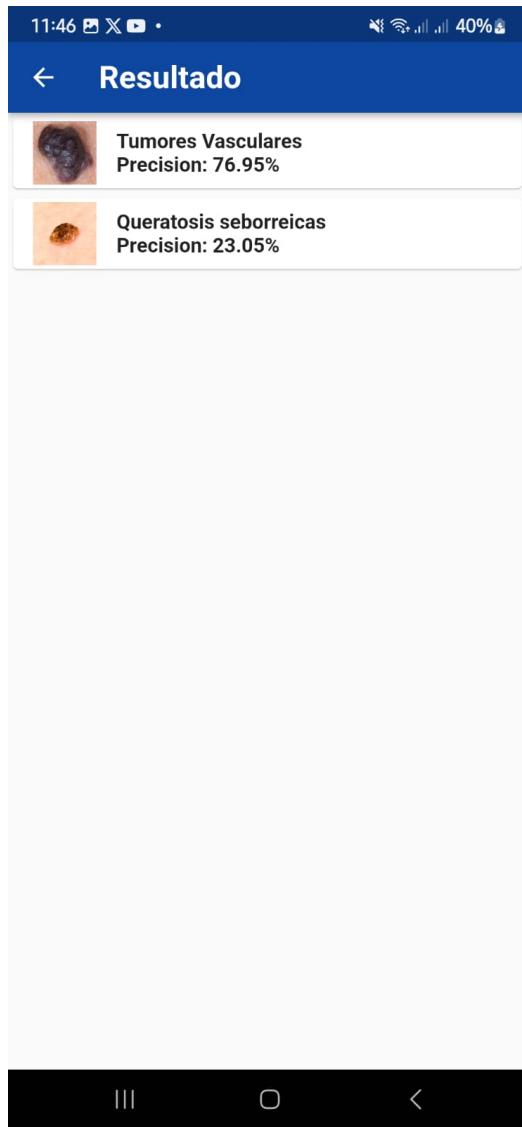
Nota: Obtenida a partir de captura de pantalla de la app móvil DermaDI

Pantalla “Resultado”

La pantalla de Resultado muestra las salidas del modelo, como bien se ha mencionado, el modelo de reconocimiento de enfermedades de piel puede devolver más de una salida mostrando el porcentaje de precisión para cada una.

Figura 14

Pantalla de Resultado de la clasificación de la imagen



Nota: Obtenida a partir de captura de pantalla de la app móvil DermaDI

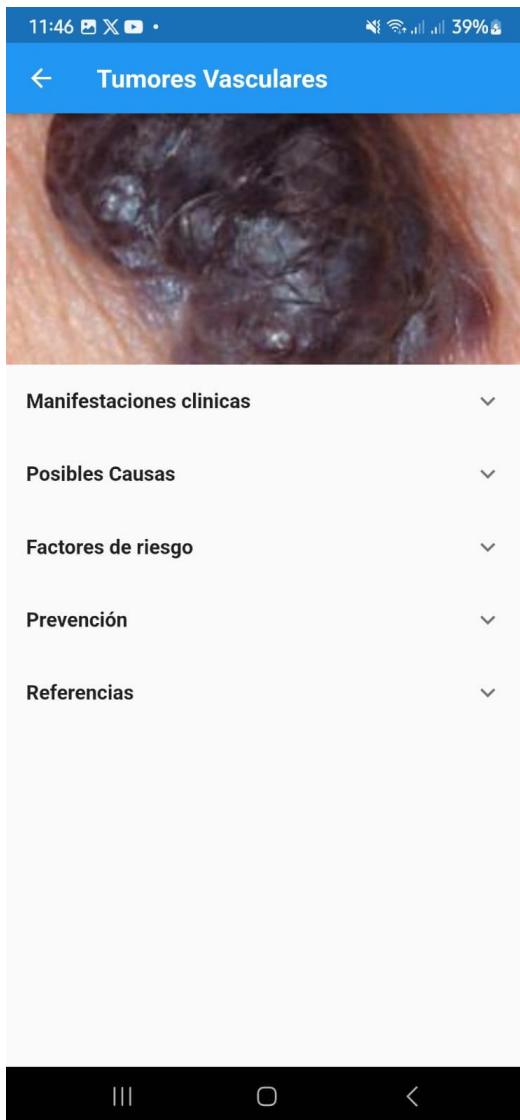
Pantalla “Información de la enfermedad”

Al generar los resultados de la clasificación, el usuario puede acceder a Información de referencia sobre la enfermedad identificada, la Información brindada incluye Información general de la enfermedad, sintomatología, posibles causas, factores de riesgo, medidas de

prevención posibles, incluyendo la referencia bibliográfica de donde se obtuvo la Información mostrada.

Figura 15

Pantalla de Información sobre la enfermedad identificada



Nota: Obtenida a partir de captura de pantalla de la app móvil DermaDI

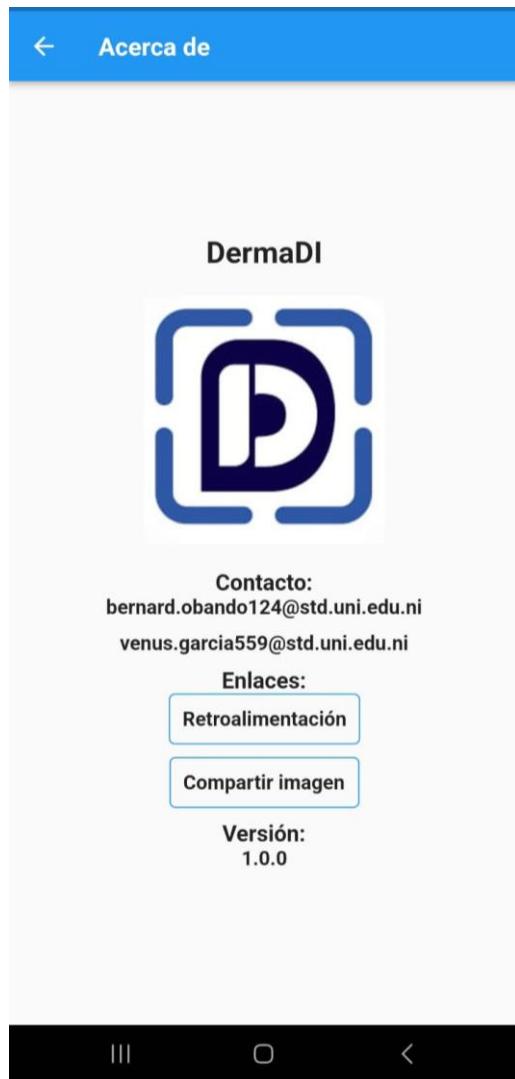
Pantalla “Acerca de”

Se incluye la pantalla ‘Acerca de’, donde, se muestra Información técnica y de contacto. Mostrando el logo de la app, la versión de la app, correos de contacto con los

desarrolladores y se proporcionan los links de acceso a formularios para la evaluación de usabilidad de la app y para retroalimentación al proyecto brindando datos de imágenes a los desarrolladores para mejorar el rendimiento del modelo.

Figura 16

Pantalla de Información técnica de la app y métodos de aporte al proyecto.



Nota: Obtenida a partir de captura de pantalla de la app móvil DermaDI

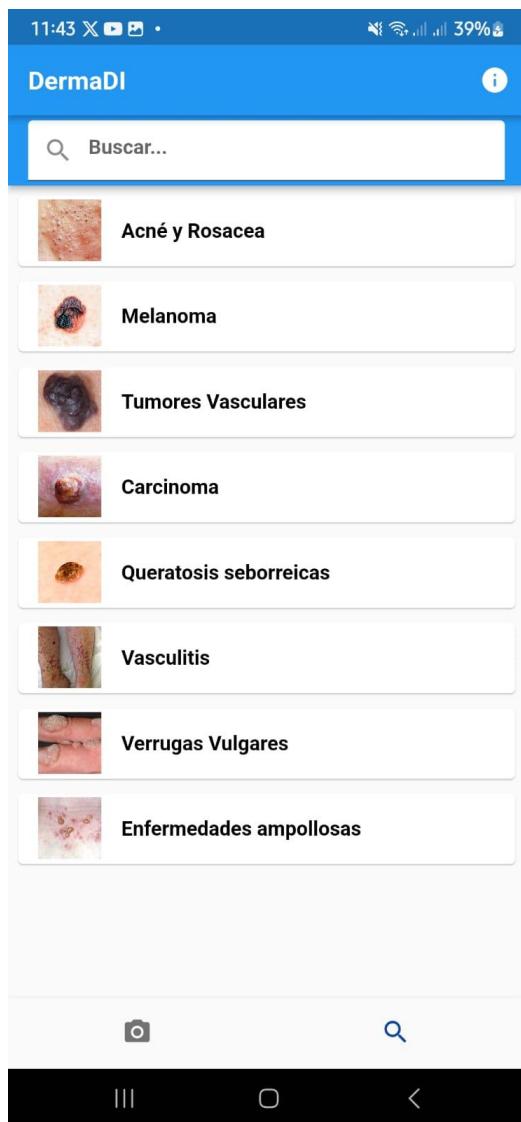
Pantalla “Galería de enfermedades”

En la galería de enfermedades se muestra a modo de lista las enfermedades contenidas en el modelo, es decir, las categorías o clases con las que fue entrenado el modelo de

reconocimiento, y a la vez se proporciona el acceso a la Información referente a la lesión identificada.

Figura 17

Pantalla de galería de enfermedades contenidas en el modelo



Nota: Obtenida a partir de captura de pantalla de la app móvil DermaDI

6.2.6.2.2. Configuración de la base de datos:

Con el desarrollo de la app DermaDI, y considerando el requisito funcional “Información de enfermedades”, se construyó una pequeña base de datos SQLite, utilizando DB Browser for SQLite.

El fin de esta base de datos es almacenar la Información mostrada en la app para cada una de las enfermedades contenidas en el modelo de reconocimiento. Tras construir la base de datos, se llenaron las tablas con la Información correspondiente. Luego se movió el archivo de la BD a la carpeta de desarrollo de la app DermaDI.

6.2.6.2.3. Pruebas de validación

En fases anteriores se documentaron evaluaciones enfocadas a la validación del modelo construido. Sin embargo, en este apartado las pruebas de validación están dirigidas a una evaluación general considerando la herramienta de despliegue utilizada para la integración del modelo.

En este apartado se documenta los resultados de la evaluación del modelo integrado en la aplicación móvil DermaDI. La evaluación se llevó a cabo por medio de la realización de un formulario de usabilidad dirigido a médicos y estudiantes residentes del Hospital Central Dermatológico Nacional, esto con el fin de que estos médicos especialistas conocieran la herramienta, la exploraran, y brindaran sus opiniones al respecto del proyecto. Validando así, el recibimiento a este proyecto en el entorno clínico.

En los Anexos de este documento, se adjunta el diseño del formulario de usabilidad, que fue completado por los encuestados. A continuación, se describen los resultados obtenidos a través del formulario de recepción y usabilidad de la app DermaDI.

El levantamiento de encuestas se llevó a cabo en tres visitas al centro. En el día 1, se presentó la aplicación al médico asignado para dar seguimiento en todo el proceso. En el día 2, se presentó la app a 12 personas, y en el día 3, a otras 12 personas. Durante las presentaciones, se explicó brevemente que es un modelo de red neuronal convolucional y

su importancia en la construcción de un modelo para el reconocimiento de enfermedades de la piel. Se describieron las funcionalidades y contenido de la app DermaDI, además de permitirles a los encuestados descargar y probar la app en sus dispositivos Android.

Por tanto, se recolecto un total de 25 encuestas donde se obtuvieron los siguientes resultados:

1. Se establecieron dos preguntas de interés estadístico, para conocer la situación en el Hospital Central Dermatológico Nacional, las cuales fueron:

1.1. ¿Cantidad estimada de casos atendidos en un día, por medico?

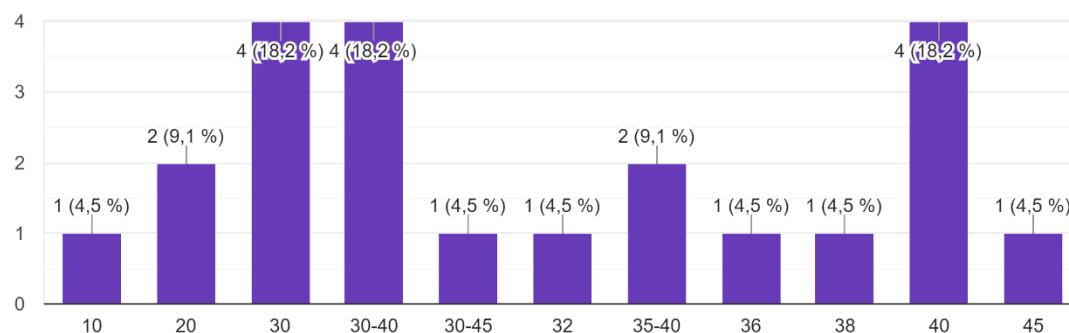
Los resultados de la encuesta mostraron que la cantidad de casos atendidos por medico en un día, en el centro dermatológico oscila entre 30 y 40. Ver figura 18.

Figura 18

Grafica estimado de casos atendidos por día.

Estimado de casos atendidos por día?

22 respuestas



Nota: Obtenida por elaboración propia

1.2. A criterio personal (del encuestado), ¿Cuáles son las 10 enfermedades más comunes atendidas en el centro?

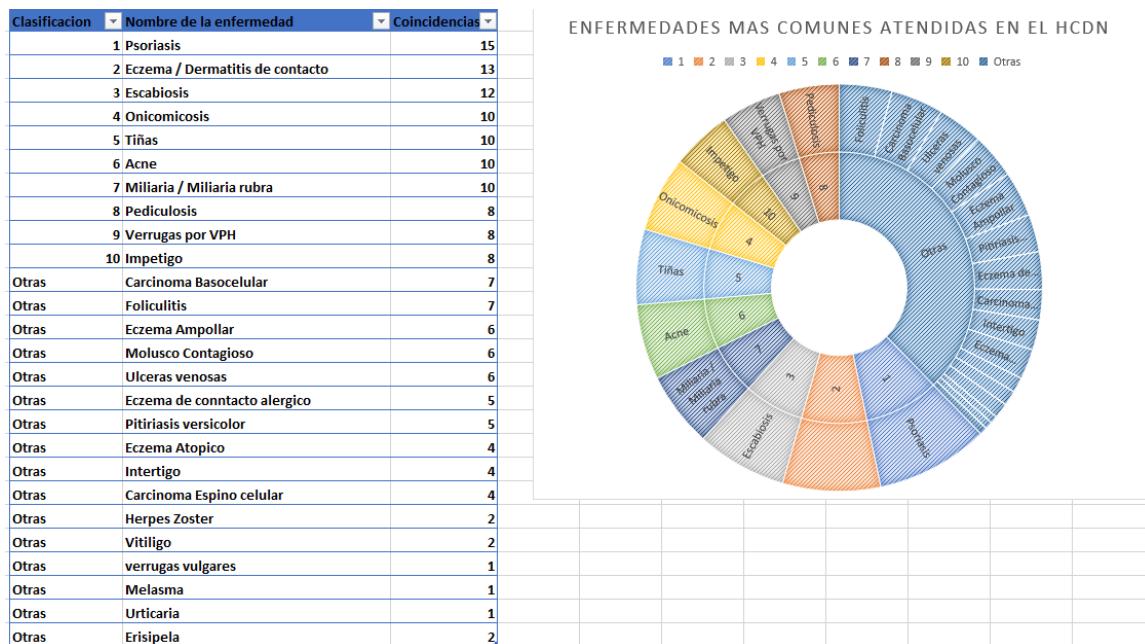
Se tomaron las respuestas de los encuestados y se procesó la Información proporcionada asignando categorías y considerando las coincidencias en la mención de las patologías clínicas para estimar cuales son las patologías más frecuentes tratadas en el Hospital

Central Dermatológico Nacional (HCND), tras el análisis se resolvió que las enfermedades más comunes son las siguientes

1. Psoriasis
2. Eczema / Dermatitis de contacto
3. Escabiosis
4. Onicomicosis
5. Tiñas
6. Acné
7. Miliaria / Miliaria rubra
8. Pediculosis
9. Verrugas por VPH
10. Impétigo

Figura 19

Grafica de enfermedades más comunes atendidas en el HCND



Nota: Obtenida por elaboración propia

2. Se solicito al encuestado evaluar su experiencia con la aplicación DermaDI, calificándola considerando afirmaciones dadas en 5 categorías (estética, usabilidad, contenido, funcionalidad y satisfacción.), mediante un rango calificativo, donde:

- 1: Muy en desacuerdo
- 2: Algo en desacuerdo
- 3: Ni en desacuerdo, Ni de acuerdo

4: Algo de acuerdo

5: Muy de acuerdo

2.1. Estética

Para la evaluación de la estética de la aplicación, se dieron las siguientes afirmaciones:

Afirmación 1: Los colores y el diseño visual de la aplicación son atractivos.

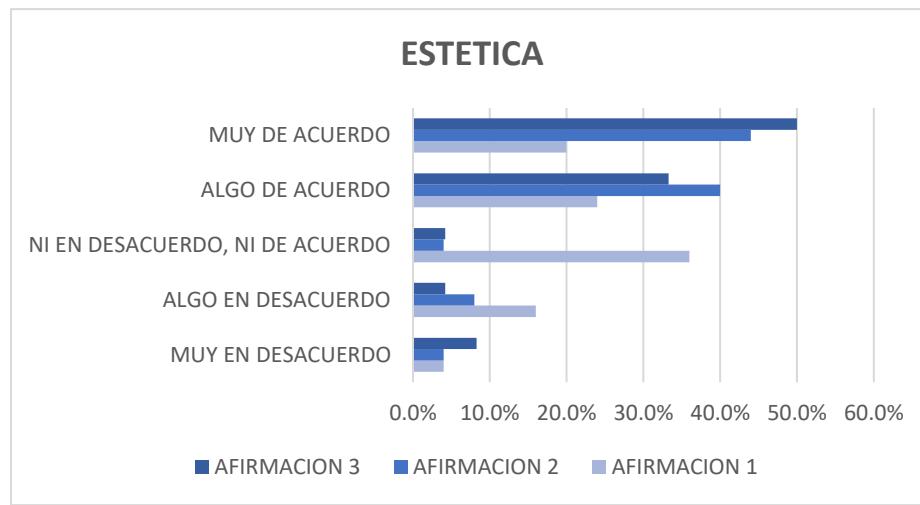
Afirmación 2: El tamaño, fuente y color de letras respetan la comodidad visual del usuario.

Afirmación 3: Los iconos y las imágenes utilizados en la aplicación son fáciles de entender.

En términos de estética, la valoración de la app DermaDI alcanzo una respuesta positiva del 50% con la consideración “muy de acuerdo” y 40% con la consideración “algo de acuerdo” en las afirmaciones dadas. Ver figura 20.

Figura 20

Valoración de la estética de DermaDI



Nota: Obtenida por elaboración propia

2.2. Usabilidad

Para la evaluación de la usabilidad de la aplicación, se dieron las siguientes afirmaciones:

Afirmación 1: El acceso a la cámara como primera pantalla tras iniciar la app es apropiado.

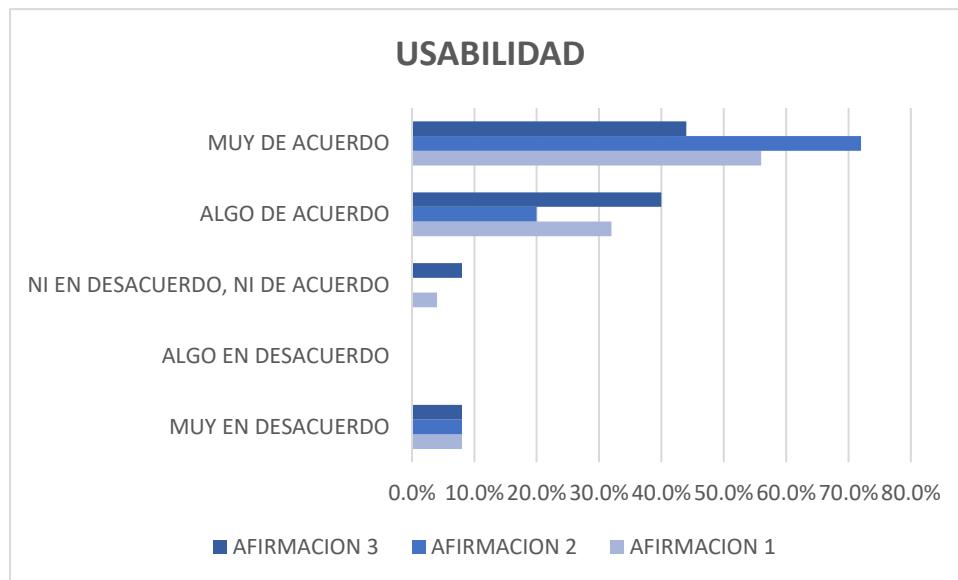
Afirmación 2: Es útil la opción de cargar imagen desde la galería del dispositivo.

Afirmación 3: La aplicación proporciona una respuesta clara y precisa después de tomar una foto o subir una imagen.

En términos de estética, la valoración de la app DermaDI alcanzo una respuesta positiva del 70% con la consideración “muy de acuerdo” y 40% con la consideración “algo de acuerdo” en las afirmaciones dadas. Ver figura 21.

Figura 21

Valoración de la usabilidad de DermaDI



Nota: Obtenida por elaboración propia

2.3. Contenido

Para la evaluación del contenido de la aplicación, se dieron las siguientes afirmaciones:

Afirmación 1: La aplicación proporciona Información útil sobre las enfermedades de la piel.

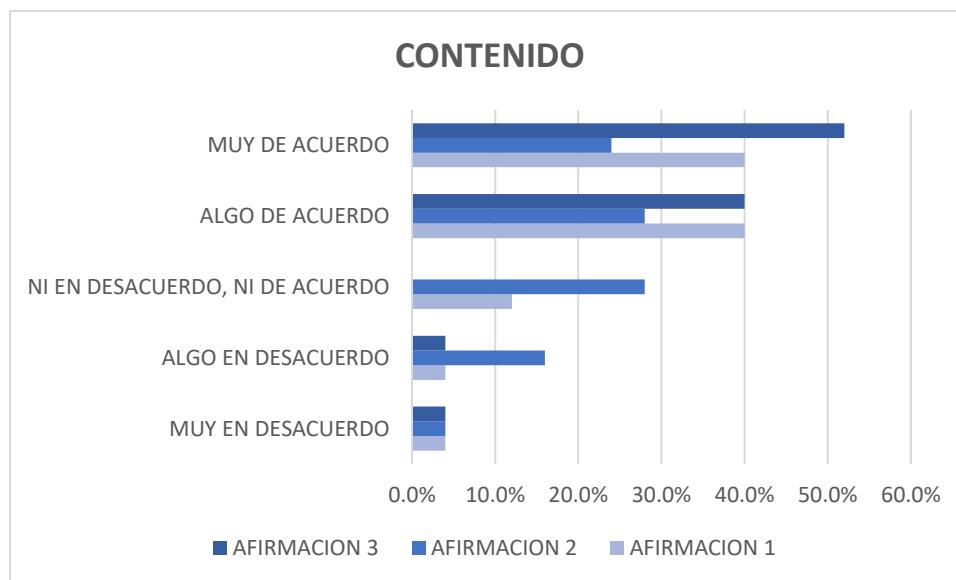
Afirmación 2: La Información proporcionada esta actualizada, es precisa y está debidamente referenciada.

Afirmación 3: El módulo de Galería de enfermedades es útil para dar a conocer las enfermedades más comunes.

En términos de contenido, la valoración de la app DermaDI alcanzo una respuesta positiva del 50% con la consideración “muy de acuerdo” y 40% con la consideración “algo de acuerdo” en las afirmaciones dadas. Ver figura 22.

Figura 22

Valoración del contenido de DermaDI



Nota: Obtenida por elaboración propia

2.4. Funcionalidad

Para la evaluación de la funcionalidad de la aplicación, se dieron las siguientes afirmaciones:

Afirmación 1: La aplicación funciona sin problemas en su dispositivo.

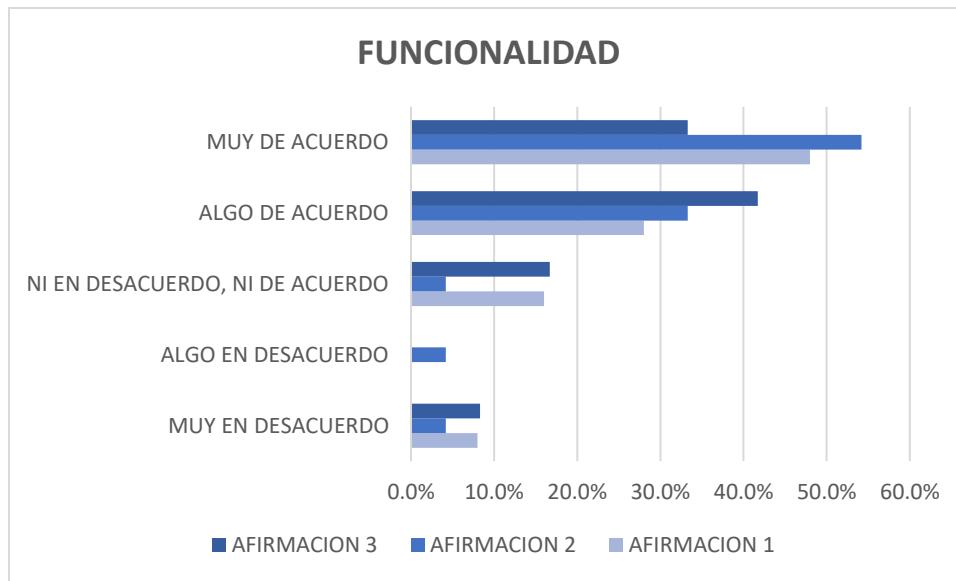
Afirmación 2: La velocidad de reconocimiento de la aplicación fue satisfactoria.

Afirmación 3: El tiempo de respuesta de una pantalla a otra es apropiado.

En términos de funcionalidad, la valoración de la app DermaDI alcanzo una respuesta positiva de más del 50% con la consideración “muy de acuerdo” y más del 40% con la consideración “algo de acuerdo” en las afirmaciones dadas. Ver figura 23.

Figura 23

Valoración de la funcionalidad de DermaDI



Nota: Obtenida por elaboración propia

2.5. Satisfacción

Para la evaluación de la satisfacción del usuario con la aplicación, se dieron las siguientes afirmaciones:

Afirmación 1: De modo general, la aplicación le resulta útil y responde sus expectativas.

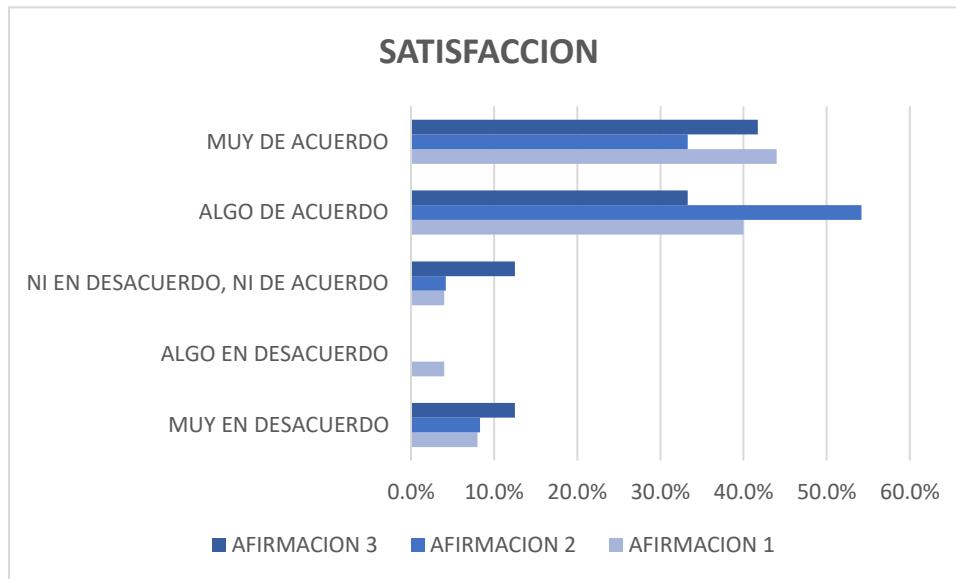
Afirmación 2: Ha obtenido un reconocimiento satisfactorio tras poner a prueba el modelo.

Afirmación 3: Recomendaría la app a otros usuarios.

En términos de satisfacción del usuario, la valoración de la app DermaDI alcanzo una respuesta positiva de más del 45% con la consideración “muy de acuerdo” y más del 50% con la consideración “algo de acuerdo” en las afirmaciones dadas. Ver figura 24.

Figura 24

Valoración de la satisfacción del usuario con DermaDI



Nota: Obtenida por elaboración propia

3. Se pidió a los encuestados brindar sus comentarios considerando sus expectativas al saber de una app para el reconocimiento de enfermedades de la piel, su experiencia con el uso de la app DermaDI y recomendaciones para la mejora de la herramienta. A continuación, se presenta un resumen de las respuestas dadas por parte de los encuestados.

Los comentarios de los encuestados ofrecen una visión completa de sus experiencias con la aplicación DermaDI. En general, se aprecia que la mayoría de los usuarios encuentran la aplicación valiosa. Algunos resaltaron su utilidad tanto para médicos en formación como para aquellos en áreas remotas. Sin embargo, se identificaron áreas de mejora potenciales. Algunos encuestados sugirieron que la aplicación podría ampliar su cobertura de enfermedades dermatológicas y agregar información sobre métodos de diagnóstico y tratamiento.

La importancia de mantener la aplicación actualizada y en constante evolución fue un tema recurrente. También se sugirió consultar el libro de dermatología del Dr. Fitzpatrick como una posible fuente de referencia para mejorar la aplicación, consideración que se tomó en cuenta luego de la primera visita para la continuación de la presentación de la app y levantamiento de encuestas. Las opiniones de los encuestados destacaron tanto las fortalezas de DermaDI como las áreas donde podría realizarse mejoras.

6.2.6.2.4. Impacto social del proyecto monográfico

Este trabajo monográfico conlleva la implementación de técnicas de Machine Learning, una rama de la informática que va en auge en la actualidad y que sin embargo su uso es apenas presente en los países en desarrollo, como Nicaragua. Con este trabajo se proyectó implementar esta tecnología y orientarla a un sector significativo para el país, lo que resultó en determinar el sector salud, como el sector predilecto para este proyecto académico.

Este proyecto trae consigo un impacto significativo al brindar un acceso a la experiencia, donde los médicos no especializados en dermatología, especialmente aquellos ubicados en áreas desatendidas o remotas, podrían contar con una herramienta que les ayude a cerrar estas brechas al proporcionarles una manera de hacer diagnósticos de forma asistida gracias a una herramienta como DermaDI que conlleva el reconocimiento de enfermedades de piel, mejorando los servicios de atención médica en esas áreas.

Este proyecto puede ser llevado más allá y motivar a los especialistas en mejorar la atención médica considerando la sistematización del proceso de enseñanza a los médicos especialistas en formación, permitiéndoles aprender y mejorar sus habilidades de diagnóstico. Podrá servir como un recurso valioso para la educación médica continua y un arranque en la motivación de incluir estas tecnologías en más procesos involucrados en el sector salud.

7. Conclusión.

El presente proyecto monográfico ha logrado cumplir con éxito sus objetivos mediante un enfoque metodológico basado en CRISP-DM. El objetivo central del proyecto consistía en la implementación de un modelo de red neuronal convolucional (CNN) para el reconocimiento de enfermedades de la piel. Dicho objetivo se cumplió al llevar a cabo una serie de actividades correspondientes al modelo CRISP-DM, las cuales, a su vez, con el desarrollo de cada fase, fueron dando cumplimiento a cada uno de los objetivos específicos del proyecto.

Aprovechando fuentes disponibles como Dermnet y Kaggle, la metodología CRISP-DM guio el proceso para la ejecución de las actividades referidas, que comprendieron la recopilación de los datos (el conjunto de datos), la definición de la arquitectura de red, la construcción del modelo, el entrenamiento del modelo, la evaluación del modelo, el desarrollo de la app móvil DermaDI y la integración del modelo resultante en la app DermaDI.

El modelo CNN construido no solo demostró ser preciso en el reconocimiento de enfermedades de la piel con las que fue entrenado, sino que también se integró con éxito en la aplicación móvil DermaDI, haciéndolo accesible a cualquier usuario. Esto representa un hito significativo en la aplicación práctica de la inteligencia artificial en el campo de la salud en Nicaragua.

Este proyecto fue un valioso ejercicio de aprendizaje en el ámbito del Deep learning para el equipo de trabajo. Los conocimientos adquiridos, las habilidades desarrolladas y los resultados obtenidos, todos documentados en el presente, servirán como base de consulta y guía para proyectos futuros que se desarrolleen en el campo de la ciencia de datos y la inteligencia artificial.

8. Recomendaciones

Tras la evaluación de todo el proceso de desarrollo de este proyecto, se han identificado varias recomendaciones para mejorar tanto la precisión del modelo como la eficiencia de su implementación:

- Incremento del Tamaño del Set de Datos: Para aumentar la precisión del modelo, se recomienda considerar la expansión del conjunto de datos con una mayor cantidad de imágenes. Un set de datos más grande puede ayudar al modelo a aprender patrones más complejos y representativos de las enfermedades de la piel.
- Ampliación del Número de Clases: Para abarcar un espectro más amplio de enfermedades de la piel, se sugiere considerar la inclusión de un mayor número de clases en el modelo. Esto permitiría una detección más precisa de una variedad de afecciones dermatológicas.
- Implementación de un Modelo en Red Local: Para un rendimiento óptimo y un control total sobre la infraestructura, se puede considerar la opción de implementar el modelo en una red local ejecutándolo en una computadora potente.
- Exploración de Servicios en Línea: En caso de que los recursos locales sean limitados, se puede explorar la opción de utilizar servicios en línea especializados en el procesamiento de grandes cantidades de información. Plataformas como Runpod.io o soluciones de pago como Google Colab Pro pueden proporcionar la capacidad computacional necesaria para proyectos a gran escala.

Estas recomendaciones se basan en la experiencia adquirida durante el desarrollo de este proyecto y tienen como objetivo mejorar la calidad y el alcance de futuras iteraciones del mismo. La implementación de estas sugerencias puede contribuir significativamente a la eficacia y utilidad del modelo de reconocimiento de enfermedades de la piel y la aplicación DermaDI.

9. Bibliografía

- [1]. Alvarez Corredera, P. (s.f.). *CiberNinjas*. Obtenido de
https://ciberninjas.com/15-mejores-lenguajes-programacion/#11_Dart
- [2]. Alvarez Gil, D. (14 de Enero de 2021). *adictosaltrabajo.com*. Obtenido de
<https://www.adictosaltrabajo.com/2021/01/14/metodologia-crisp-dm/>
- [3]. aws.amazon. (s.f.). *aws.amazon*. Obtenido de
<https://aws.amazon.com/es/what-is/python/>
- [4]. Barrios, J. (s.f.). *Juan Barrios. Healt Big Data*. Obtenido de
<https://www.juanbarrios.com/redes-neurales-convolucionales/>
- [5]. Benedetti, J. (Diciembre de 2021). *Msd Manual*. Obtenido de
<https://www.msdmanuals.com/en-au/home/skin-disorders/biology-of-the-skin/diagnosis-of-skin-disorders>
- [6]. Casa Lema, M., & Tumbaco Sango, J. (Febrero de 2020). *Universidad tecnica de COTOPAXI, Biblioteca general*. Obtenido de
<http://repositorio.utc.edu.ec/handle/27000/6698>
- [7]. Dataprix. (15 de Septiembre de 2007). *Dataprix*. Obtenido de
<https://www.dataprix.com/es/metodologia-crisp-dm-mineria-datos/modelo-referencia-crisp-dm>
- [8]. DermNet. (s.f.). Obtenido de <https://dermnetnz.org/>
- [9]. DermNet. (s.f.). *DermNet*. Obtenido de <https://dermnetnz.org/contact-us/submit-a-photo>
- [10]. Developer Android. (s.f.). *Developer Android*. Obtenido de
<https://developer.android.com/studio/intro?hl=es-419>
- [11]. Farfan Vergara, S. (Abril de 2021). *Repositorio de Tesis UPEU*. Obtenido de Universidad Peruana Union:
<https://repositorio.upeu.edu.pe/handle/20.500.12840/5506>

- [12]. García, L. M., & Urbina, I. Y. (09 de Julio de 2014). Obtenido de
<http://riul.unanleon.edu.ni:8080/jspui/bitstream/123456789/4952/1/230218.pdf>
- [13]. Goicochea, A. (11 de Agosto de 2009). *Anibal Goicochea*. Obtenido de
<https://anibalgoicochea.com/2009/08/11/crisp-dm-una-metodologia-para-proyectos-de-mineria-de-datos/>
- [14]. Gonzalez, S. (10 de Diciembre de 2021). *welivesecurity*. Obtenido de
<https://www.welivesecurity.com/la-es/2021/12/10/por-que-machine-learning-aliado-para-ciberseguridad/>
- [15]. Google. (s.f.). *Google*. Obtenido de <https://www.google.com/intl/es/drive/>
- [16]. Heras, J. M. (29 de Septiembre de 2020). *iArtificial.net*. Obtenido de
<https://www.iartificial.net/clasificacion-o-regresion/>
- [17]. Ibarrola, A. A. (21 de Enero de 2022). Obtenido de <https://academica-e.unavarra.es/bitstream/handle/2454/42439/TFG%20andueza.120902.pdf?sequence=1&isAllowed=y>
- [18]. Ionos Digital Guide. (08 de Octubre de 2020). *IONOS*. Obtenido de
<https://www.ionos.es/digitalguide/online-marketing/marketing-para-motores-de-busqueda/que-es-keras/>
- [19]. Ionos Digital Guide. (09 de 10 de 2020). *IONOS*. Obtenido de
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-flutter/>
- [20]. Ionos digital guide. (07 de Junio de 2021). *IONOS*. Obtenido de
<https://www.ionos.es/digitalguide/paginas-web/diseno-web/que-es-un-pixel/>
- [21]. Itelligent. (19 de Agosto de 2020). *Itelligent*. Obtenido de
<https://itelligent.es/es/machine-learning-que-es-como-funciona/>
- [22]. K. A. Muhaba; K. Dese; T. M. Aga; F. T. Zewdu; G. L. Simegn. (09 de Noviembre de 2021). doi:10.1002/ski2.81

- [23]. Kendall, K. E., & Kendall, J. E. (2011). *Analisis y diseño de sistemas* (Octava ed.). (L. M. Castillo, Ed.) Pearson Educacion. Recuperado el Junio de 2023, de <https://disevapro4a.wordpress.com/que-es-un-estudio-de-factibilidad/>
- [24]. Li, L.-F., Hu, W.-J., Xiong, N. N., Du, Y.-X., Li, B.-S., & Xu, W. (11 de Noviembre de 2020). *IEEEEXPLORE*. doi:10.1109/ACCESS.2020.3037258
- [25]. Martinez, S. (06 de Febrero de 2020). *JasminSoftware*. Obtenido de <https://www.jasminsoftware.es/blog/deep-learning/>
- [26]. MathWorks. (s.f.). *MathWorks*. Obtenido de <https://es.mathworks.com/discovery/machine-learning.html>
- [27]. Na8. (29 de Noviembre de 2018). *aprendemachinelearning.com*. Obtenido de <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>
- [28]. NetApp. (s.f.). *NetApp*. Obtenido de <https://www.netapp.com/es/artificial-intelligence/what-is-artificial-intelligence/>
- [29]. Perlato, A. (s.f.). *AndreaPerlato.com*. Obtenido de <https://www.andreaperlato.com/aipost/cnn-and-softmax/?fbclid=IwAR0SguaNjnDLrKqpU2QUz7aimzE9SC-WCmrPvRlwqmn24eNj-jaRsUI8rs>
- [30]. Porto, J. P., & Merino, M. (2009). *Definicion.De*. Obtenido de <https://definicion.de/pixel/>
- [31]. Prada, J. M. (15 de Junio de 2022). *Incentro*. Obtenido de <https://www.incentro.com/es-ES/blog/que-es-tensorflow>
- [32]. Ramis, M. (06 de Diciembre de 1943). Obtenido de <https://proyectoidis.org/neurona-artificial/>
- [33]. Redaccion KeepCoding. (13 de Enero de 2022). *Keep Coding*. Obtenido de <https://keepcoding.io/blog/redes-neuronales-convolucionales/>

- [34]. Redaccion KeepCoding. (22 de Agosto de 2022). *KeepCoding*. Obtenido de <https://keepcoding.io/blog/que-es-la-convolucion/>
- [35]. Redaccion Tokio. (26 de 08 de 2022). *Tokio*. Obtenido de <https://www.tokioschool.com/noticias/google-colab/>
- [36]. Rommel, F. (s.f.). SG. Obtenido de <https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida>
- [37]. Sandler. (s.f.). *paperswithcode*. Obtenido de <https://paperswithcode.com/method/mobilenetv2>
- [38]. singular. (s.f.). Obtenido de <https://www.sngular.com/es/data-science-crisp-dm-metodologia/>
- [39]. Sotaquirá, M. (06 de Abril de 2019). *Codificandobits*. Obtenido de <https://www.codificandobits.com/blog/padding-strides-maxpooling-stacking-redes-convolucionales/#max-pooling>
- [40]. Srinivasu, P. N., SivaSai, J. G., Ijaz, M. F., Bhoi, A. K., Kim, W., & Kang, J. J. (18 de Abril de 2021). *Nacional Lybrary of Medicine*. doi:10.3390/s21082852
- [41]. Stradata aml. (s.f.). *STRADATA AML*. Obtenido de <https://aml.stradata.co/2017/03/27/preprocesamiento-de-datos-una-forma-de-solucionar-problemas-antes-de-que-aparezcan/>
- [42]. Teachable Machine. (s.f.). *Teachable Machine*. Obtenido de <https://teachablemachine.withgoogle.com/>
- [43]. TensorFlow. (s.f.). Obtenido de <https://www.tensorflow.org/lite?hl=es-419>
- [44]. TuSalario. (2023). *TuSalario.org*. Obtenido de <https://tusalario.org/nicaragua/tu-carrera-profesional/nicaragua-trabajo-y-pago/nicaragua-desarrolladores-de-software>
- [45]. UNIR Revista. (03 de Agosto de 2021). *UNIR La universidad en internet*. Obtenido de <https://www.unir.net/ingenieria/revista/redes-neuronales-artificiales/#:~:text=Las%20redes%20neuronales%20artificiales%20son%20un%20m>

odelo%20computacional%20que%20permite,como%20lo%20hace%20nuestro%20ce
rebro.

- [46]. Velasco, J., Pascion, C., Alberio, J., Apuang, J., Cruz, J., Gomez, M., . . . Romeo Jorda, A. T.-a. (Septiembre de 2019). doi:10.30534/ijatcse/2019/116852019
- [47]. Vincle Redaccion. (16 de Marzo de 2021). *vinkle.com*. Obtenido de <https://www.vinkle.com/es/3-beneficios-del-reconocimiento-de-imagenes/>

10. Anexos

10.1 Cronograma de actividades del proyecto

ACTIVIDADES	MESES Y SEMANAS																								
	Mes 1				Mes 2				Mes 3				Mes 4				Mes 5				Mes 6				
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4	
Fase 1. Entendimiento del problema																									
Definicion de Requerimientos																									
Evaluacion de la situacion actual																									
Definicion de objetivos																									
Fase 2. Compresion de los datos																									
Recolección de datos (Imágenes)																									
Descripcion de los datos																									
Fase 3. Preparacion de datos																									
Clasificacion de los datos																									
Limpieza de los datos																									
Separacion de datos (Datos de entrenamiento y Datos de pruebas).																									
Fase 4. Modelado																									
Construcción y entrenamiento del modelo																									
Fase 5. Evaluacion																									
Pruebas al modelo generado																									
Evaluacion de los resultados																									
Fase 6. Integracion y Despliegue																									
Desarrollo de aplicación móvil DermaDI																									
Creacion de interfaces																									
Creacion y llenado de base de datos SQL Lite																									
Pruebas a la base de datos SQL Lite																									
Integracion del modelo																									
Integracion de la base de datos																									
Conexion del modelo y la base de datos																									
Pruebas a la aplicación y funcionamiento del modelo en la app																									
Documentacion y Defensa																									
Redaccion del documento																									
Revision con el tutor																									
Pre-Defensa																									
Defensa																									

10.2. Diseño del Formulario de recepción y usabilidad de la app móvil DermaDI

RESULTADOS DE PRESENTACIÓN DE LA APLICACIÓN MÓVIL DERMADI

El presente formulario está orientado a la recepción de comentarios sobre la aplicación móvil DermaDI considerando los aspectos de estética, usabilidad, contenido, funcionalidad y satisfacción del usuario. Además de un espacio de preguntas con interés estadístico respecto al comportamiento de enfermedades de la piel.

1. Interrogantes para Información de interés estadístico

Favor contestar las preguntas a continuación, en base a su percepción y experiencia personal.

1.1. ¿Estimado de casos atendidos por día?

1.2. ¿Cuáles son las 10 enfermedades de piel más comunes atendidas en el Centro Dermatológico Nacional? Favor indicarlas en orden de prioridad según su criterio.

1. _____

2. _____

3. _____

4. _____

5. _____

6. _____

7. _____

8. _____

9. _____

10. _____

2. **Califica las siguientes afirmaciones del 1 al 5 según tu grado de acuerdo con las mismas, siendo:**

1: Muy en desacuerdo

2: Algo en desacuerdo

3: Ni en desacuerdo, Ni de acuerdo

4: Algo de acuerdo

5: Muy de acuerdo

2.1 Estética

Los colores y el diseño visual de la aplicación son atractivos.

1	2	3	4	5
<input type="radio"/>				

El tamaño, fuente y color de letras respetan la comodidad visual del usuario

1	2	3	4	5
<input type="radio"/>				

Los iconos y las imágenes utilizados en la aplicación son fáciles de entender.

1	2	3	4	5
<input type="radio"/>				

2.2. Usabilidad

El acceso a la cámara como primera pantalla tras iniciar la app es apropiado considerando el propósito de la aplicación.

1	2	3	4	5
<input type="radio"/>				

Es útil la opción de cargar imagen desde la galería del dispositivo.

1	2	3	4	5
<input type="radio"/>				

La aplicación proporciona una respuesta clara y precisa después de tomar una foto o subir una imagen.

1	2	3	4	5
<input type="radio"/>				

2.3. Contenido.

La aplicación proporciona Información útil sobre las enfermedades de la piel.

1	2	3	4	5
<input type="radio"/>				

La Información proporcionada esta actualizada, es precisa y está debidamente referenciada.

1	2	3	4	5
<input type="radio"/>				

El apartado de búsqueda de enfermedades es útil para conocer sobre las distintas enfermedades que afectan la piel.

1	2	3	4	5
<input type="radio"/>				

2.4. Funcionalidad.

La aplicación funciona sin problemas en su dispositivo.

1	2	3	4	5
<input type="radio"/>				

La velocidad de reconocimiento de la aplicación fue satisfactoria.

1	2	3	4	5
<input type="radio"/>				

El tiempo de respuesta de una pantalla a otra es apropiado.

1	2	3	4	5
<input type="radio"/>				

2.5. Satisfacción

De modo general, la aplicación le resulta útil y responde sus expectativas.

1	2	3	4	5
<input type="radio"/>				

Ha obtenido un reconocimiento satisfactorio tras poner a prueba el modelo.

1	2	3	4	5
<input type="radio"/>				

Recomendaría la app a otros usuarios.

1	2	3	4	5
<input type="radio"/>				

3. Retroalimentación.

Deje sus comentarios considerando sus expectativas al saber de una app para el reconocimiento de enfermedades de la piel, su experiencia con el uso de la app DermaDI y recomendaciones para la mejora de la herramienta.
