



GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

OPTIMIZACIÓN DE RECURSOS HOSPITALARIOS MEDIANTE EL USO DE UNA APP DE DIAGNÓSTICO DERMATOLÓGICO BASADA EN MACHINE LEARNING

Autor: Paula Arroyo Plaza

Director: Atilano Ramiro Fernández-Pacheco Sánchez-Migallón

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
OPTIMIZACIÓN DE RECURSOS HOSPITALARIOS MEDIANTE EL USO DE UNA
APP DE DIAGNÓSTICO DERMATOLÓGICO BASADA EN MACHINE LEARNING

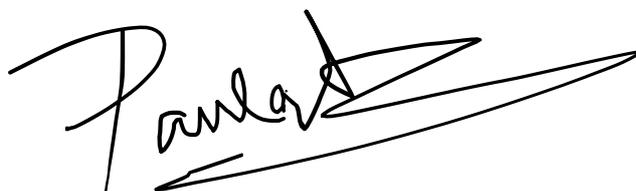
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.



Fdo.: Paula Arroyo Plaza

Fecha: 27/ 6/ 2025

Autorizada la entrega del proyecto

Atilano Ramiro Fernández-Pacheco Sánchez-Migallón

Fdo.: Atilano Ramiro Fernández-Pacheco Sánchez-Migallón

Fecha://

.....



GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

OPTIMIZACIÓN DE RECURSOS HOSPITALARIOS MEDIANTE EL USO DE UNA APP DE DIAGNÓSTICO DERMATOLÓGICO BASADA EN MACHINE LEARNING

Autor: Paula Arroyo Plaza

Director: Atilano Ramiro Fernández-Pacheco Sánchez-Migallón

Madrid

Agradecimientos

Quiero expresar mi más profundo agradecimiento a mis padres, por ser siempre mi pilar fundamental. Gracias por su apoyo incondicional, por confiar en mí en cada paso del camino y por enseñarme, con su ejemplo, los valores del esfuerzo, la constancia y la generosidad. Este trabajo no habría sido posible sin su respaldo, su motivación constante y su cercanía en los momentos más importantes.

OPTIMIZACIÓN DE RECURSOS HOSPITALARIOS MEDIANTE EL USO DE UNA APP DE DIAGNÓSTICO DERMATOLÓGICO BASADA EN MACHINE LEARNING

Autor: Arroyo Plaza, Paula.

Director: Fernández-Pacheco Sánchez-Migallón, Atilano Ramiro.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Este trabajo investiga la viabilidad de una aplicación móvil que, mediante técnicas de *machine learning*, permite el diagnóstico dermatológico de lesiones cutáneas. La herramienta se orienta a la detección temprana y clasificación de lesiones, lo que contribuye a una atención médica más ágil y precisa. Se plantea la integración de tecnologías como Flutter [1] para el desarrollo de la aplicación y Firebase [2] para la gestión segura de datos. La propuesta busca optimizar los recursos hospitalarios, reduciendo tiempos de espera y costos asociados a diagnósticos convencionales. Los resultados preliminares evidencian un alto potencial en la mejora de la eficiencia en el sector salud.

Palabras clave: *Machine Learning*, Aplicación móvil, Diagnóstico dermatológico, Redes Neuronales Convolucionales.

1. Introducción

El *machine learning* se presenta como una tecnología transformadora en el sector salud, mejorando tanto la precisión en el diagnóstico como la experiencia del paciente durante el proceso asistencial. Mediante el análisis de grandes volúmenes de datos, los algoritmos son capaces de identificar patrones sutiles que posibilitan la detección temprana de enfermedades en etapas más tratables, algo que ha demostrado ser crucial en áreas como la medicina y la educación.

En dermatología, estas técnicas se aplican para examinar imágenes digitales de lesiones cutáneas, permitiendo distinguir con rapidez y exactitud entre condiciones benignas y potencialmente malignas. La integración de estas herramientas en aplicaciones móviles no solo facilita el acceso a diagnósticos preliminares, sino que también optimiza el uso de recursos hospitalarios y mejora la calidad de la atención médica.

2. Definición del proyecto

El presente proyecto tiene como finalidad desarrollar una aplicación móvil que, mediante técnicas avanzadas de *machine learning*, permita el diagnóstico automatizado de lesiones dermatológicas para optimizar el uso de recursos hospitalarios. La propuesta se centra en el análisis de imágenes digitales de la piel, con el fin de identificar patrones que diferencien entre condiciones benignas y potencialmente malignas, posibilitando así diagnósticos tempranos y eficientes. Además, la aplicación estará diseñada con una interfaz intuitiva y accesible, garantizando una experiencia de usuario óptima tanto para profesionales de la salud como para pacientes, y sirviendo de apoyo en la toma de decisiones clínicas. Los objetivos de este proyecto son:

- Diseñar y entrenar un modelo de *machine learning* capaz de clasificar lesiones dermatológicas con alta precisión.
- Desarrollar una aplicación móvil que integre este modelo, permitiendo la carga y análisis de imágenes de forma rápida y segura.
- Optimizar la asignación de recursos hospitalarios al facilitar diagnósticos preliminares y reducir tiempos de espera.
- Mejorar la experiencia del usuario a través de una interfaz amigable y accesible, que facilite el seguimiento y la interpretación de resultados.

3. Descripción del modelo/sistema/herramienta

Se ha desarrollado una aplicación móvil multiplataforma en Flutter denominada SkinCheck, diseñada para la detección preliminar de lesiones dermatológicas. Esta aplicación permite a los usuarios capturar imágenes de su piel, que son enviadas a un *backend* en la nube donde se procesan mediante dos modelos de *machine learning* entrenados con la base de datos de ISIC Archive [3].

Uno de los modelos predice la malignidad de la lesión y el otro identifica el tipo específico. El sistema emplea redes neuronales convolucionales optimizadas y utiliza Firebase para la autenticación segura del usuario y Supabase [4] para el almacenamiento de imágenes. La arquitectura implementa un pipeline de inferencia alojado en Render [5] que garantiza un diagnóstico preliminar rápido, escalable y seguro.

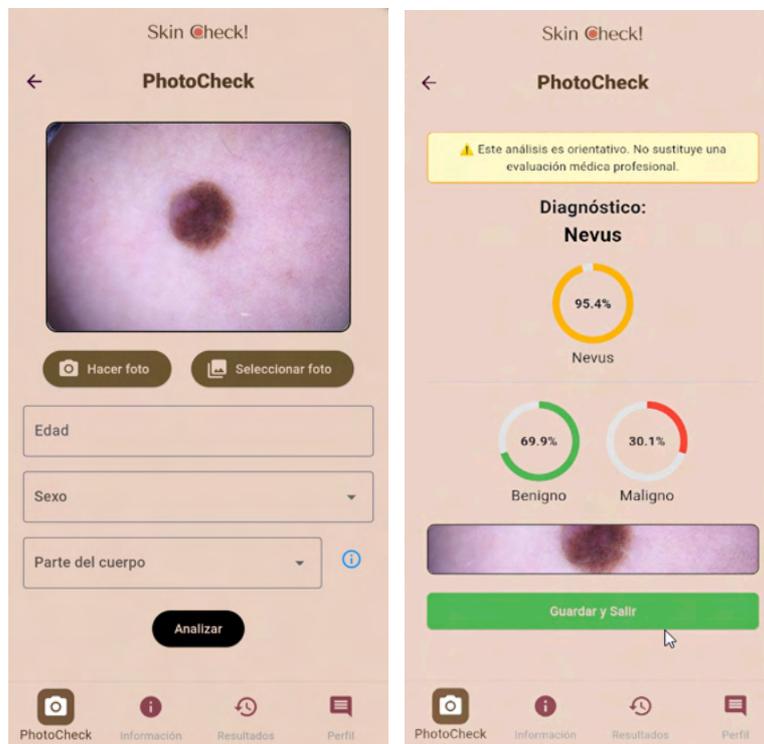


Ilustración 1. Pantallas PhotoCheck y Resultados

4. Resultados

Los modelos desarrollados mostraron un rendimiento robusto y consistente en las pruebas realizadas. El clasificador de malignidad obtuvo un F1-score de 0.85 sobre el conjunto de test, acompañado de una precisión global del 87%, lo que indica un equilibrio adecuado entre sensibilidad y precisión en la detección de lesiones potencialmente cancerígenas. Por su parte, el modelo encargado de clasificar el tipo específico de lesión alcanzó un F1-score de 0.82, demostrando una alta capacidad para distinguir entre diferentes categorías dermatológicas como BCC, MEL, NV, entre otras.

Ambos modelos fueron evaluados con métricas adicionales como sensibilidad, especificidad y el área bajo la curva ROC, obteniendo valores satisfactorios que reflejan una sólida capacidad discriminativa entre lesiones benignas y malignas.

En cuanto al rendimiento funcional de la aplicación, se observó una respuesta eficiente en los dispositivos móviles: el tiempo promedio de procesamiento por imagen fue inferior a 5 segundos desde su envío hasta la obtención del resultado, incluso en condiciones de conectividad estándar. Además, durante las pruebas funcionales realizadas en distintos entornos y con múltiples usuarios, la aplicación mantuvo una tasa de respuesta exitosa superior al 95%, lo cual refuerza su viabilidad como herramienta de diagnóstico preliminar

y su potencial utilidad clínica en contextos reales. Estos resultados sugieren que la solución propuesta no solo es precisa desde el punto de vista técnico, sino también ágil y fiable en su implementación práctica.

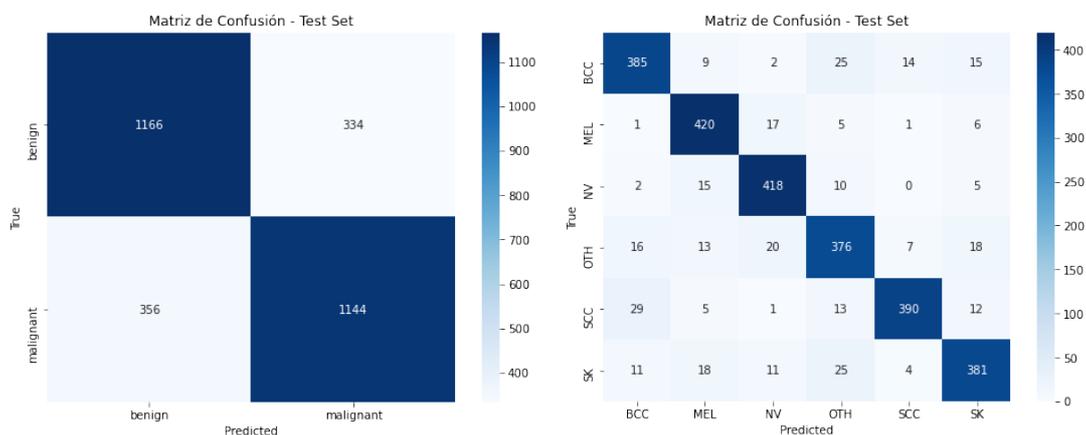


Ilustración 2. Matrices de confusión test set

5. Conclusiones

SkinCheck demuestra ser una herramienta prometedora para el diagnóstico preliminar de lesiones dermatológicas, especialmente en contextos con acceso limitado a dermatólogos. El uso de *machine learning* permite clasificar lesiones con alta precisión y en tiempo real, mejorando la eficiencia del cribado médico. Como trabajo futuro se plantea la validación clínica del sistema y su despliegue en colaboración con centros de salud para su uso en escenarios reales.

6. Referencias

- [1] Flutter Team, «Flutter,» 2024. [En línea]. Available: <https://flutter.dev/>. [Último acceso: 1 2025].
- [2] Firebase Team, «Cloud Firestore,» 2024. [En línea]. Available: <https://firebase.google.com/products/firestore>. [Último acceso: 1 2025].
- [3] ISIC Archive, «ISIC Archive,» International Skin Imaging Collaboration, 2024. [En línea]. Available: <https://www.isic-archive.com/>. [Último acceso: 2 2025].
- [4] Supabase Team, «Supabase,» 2024. [En línea]. Available: <https://supabase.com/>. [Último acceso: 1 2025].
- [5] Render, «Render,» 2024. [En línea]. Available: <https://render.com/>. [Último acceso: 1 2025].

OPTIMIZATION OF HOSPITAL RESOURCES THROUGH THE USE OF A MACHINE LEARNING-BASED DERMATOLOGICAL DIAGNOSTIC APP

Author: Arroyo Plaza, Paula

Director: Fernández-Pacheco Sánchez-Migallón, Atilano Ramiro

Collaborating Entity: ICAI – Comillas Pontifical University

PROJECT SUMMARY

This project explores the feasibility of a mobile application that leverages machine learning techniques to perform dermatological diagnoses of skin lesions. The tool is designed for early detection and classification of lesions, contributing to more agile and accurate medical care. The proposed system integrates technologies such as Flutter [1] for app development and Firebase [2] for secure data management. The main goal is to optimize hospital resources by reducing waiting times and the costs associated with conventional diagnostic methods. Preliminary results reveal strong potential for improving efficiency in the healthcare sector.

Keywords: Machine Learning, Mobile Application, Dermatological Diagnosis, Convolutional Neural Networks.

1. Introduction

Machine learning is emerging as a transformative technology in the healthcare sector, enhancing diagnostic accuracy and improving the patient experience throughout the care process. By analyzing large volumes of data, machine learning algorithms are able to identify subtle patterns that enable early disease detection at more treatable stages, something that has proven essential in fields like medicine and education.

In dermatology, these techniques are applied to analyze digital images of skin lesions, enabling the system to quickly and accurately distinguish between benign and potentially malignant conditions. Integrating these tools into mobile applications not only facilitates access to preliminary diagnoses, but also optimizes hospital resource use and improves the quality of medical care.

2. Project Summary

The objective of this project is to develop a mobile application that uses advanced machine learning techniques to automate the diagnosis of dermatological lesions, with the aim of optimizing hospital resource utilization. The proposal focuses on analyzing digital images of the skin to identify patterns that differentiate between benign and potentially malignant conditions, enabling early and efficient diagnoses.

The application will feature an intuitive and accessible user interface to ensure an optimal experience for both healthcare professionals and patients, serving as a valuable support tool for clinical decision-making. The main goals of the project are:

- To design and train a machine learning model capable of classifying dermatological lesions with high accuracy.
- To develop a mobile application that integrates this model, allowing users to upload and analyze images quickly and securely.
- To optimize hospital resource allocation by facilitating preliminary diagnoses and reducing patient wait times.
- To enhance user experience through a user-friendly and accessible interface that simplifies result tracking and interpretation.

3. Description of the Model / System / Tool

A cross-platform mobile application called SkinCheck has been developed using Flutter, designed for the preliminary detection of dermatological lesions. The app allows users to capture images of their skin, which are then sent to a cloud-based backend where they are processed by two machine learning models trained with the ISIC Archive [3] dataset.

One model predicts the malignancy of the lesion, while the other identifies its specific type. The system uses optimized convolutional neural networks (CNNs) and integrates Firebase for secure user authentication and Supabase [4] for image storage. The architecture implements an inference pipeline hosted on Render [5], ensuring fast, scalable, and secure preliminary diagnostics.

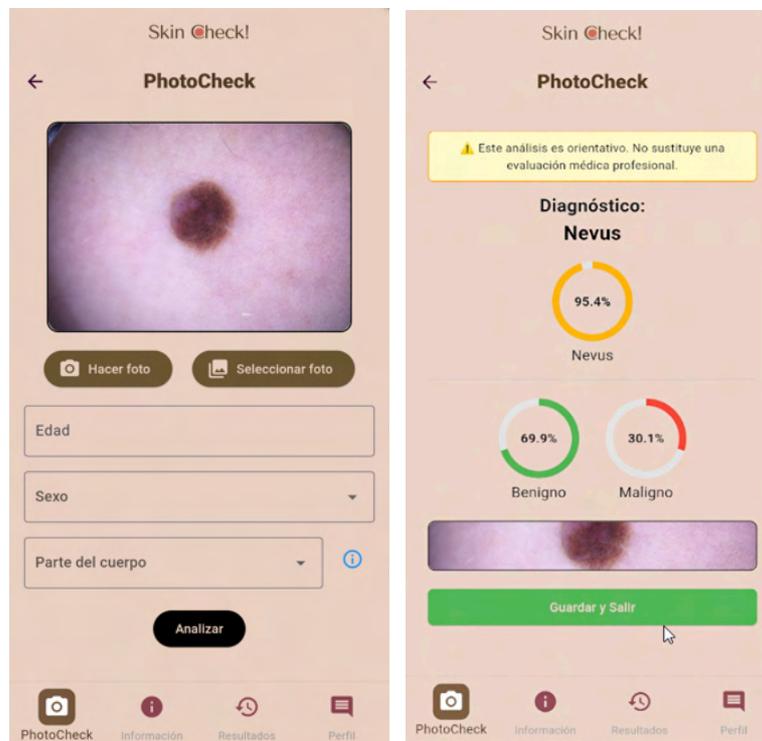


Ilustración 3. PhotoCheck and Results Screens

4. Results

The developed models demonstrated robust and consistent performance during testing. The malignancy classifier achieved an F1-score of 0.85 on the test set, along with an overall accuracy of 87%, indicating a well-balanced trade-off between sensitivity and precision in detecting potentially cancerous lesions. Meanwhile, the model responsible for classifying the specific type of lesion reached an F1-score of 0.82, showing strong capability in distinguishing between various dermatological categories such as BCC, MEL, NV, among others.

Both models were further evaluated using additional metrics such as sensitivity, specificity, and the area under the ROC curve, all of which yielded satisfactory results that reflect a solid discriminative ability between benign and malignant lesions.

Regarding the app's functional performance, efficient processing was observed on mobile devices: the average image processing time was under 5 seconds from submission to result, even under standard connectivity conditions. Furthermore, during functional testing across various environments and users, the app maintained a successful response rate of over 95%, reinforcing its feasibility as a preliminary diagnostic tool and its potential clinical

applicability in real-world settings. These results suggest that the proposed solution is not only technically accurate but also agile and reliable in practical use.

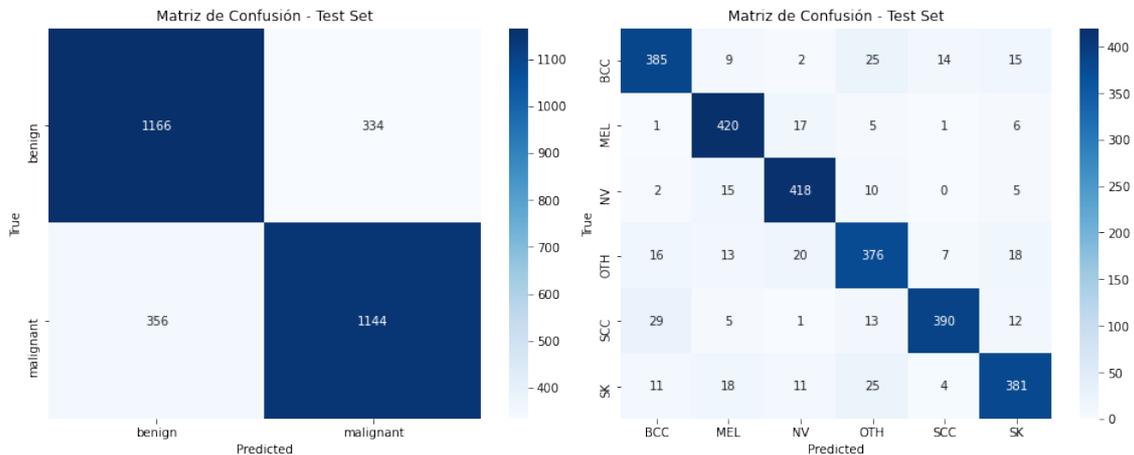


Ilustración 4. Confusion matrix test set

5. Conclusions

SkinCheck proves to be a promising tool for the preliminary diagnosis of dermatological lesions, especially in contexts with limited access to dermatologists. The use of machine learning enables real-time, high-precision lesion classification, improving the efficiency of medical triage. Future work includes clinical validation of the system and its deployment in collaboration with healthcare centers for use in real-world scenarios.

6. References

- [1] Flutter Team, «Flutter,» 2024. [En línea]. Available: <https://flutter.dev/>. [Último acceso: 1 2025].
- [2] Firebase Team, «Cloud Firestore,» 2024. [En línea]. Available: <https://firebase.google.com/products/firestore>. [Último acceso: 1 2025].
- [3] ISIC Archive, «ISIC Archive,» International Skin Imaging Collaboration, 2024. [En línea]. Available: <https://www.isic-archive.com/>. [Último acceso: 2 2025].
- [4] Supabase Team, «Supabase,» 2024. [En línea]. Available: <https://supabase.com/>. [Último acceso: 1 2025].
- [5] Render, «Render,» 2024. [En línea]. Available: <https://render.com/>. [Último acceso: 1 2025].

Índice de la memoria

Índice de la memoria	XV
Índice de figuras	19
Índice de tablas	22
Capítulo 1. Introducción	23
1.1 Motivación del proyecto	23
Capítulo 2. Descripción de las Tecnologías	24
2.1 Lenguajes de programación.....	24
2.1.1 Python	24
2.1.2 Flutter y Dart.....	25
2.2 Firebase.....	26
2.3 Visual Studio Code	28
2.4 Supabase	29
2.5 Android Studio	31
2.6 Github	32
2.7 RenderNet.....	32
Capítulo 3. Estado de la Cuestión	34
3.1 Diagnóstico de Enfermedades Dermatológicas con Inteligencia Artificial.....	34
3.1.1 Panorama actual del uso de IA en dermatología	34
3.1.2 Beneficios del diagnóstico asistido por IA en la detección de lesiones.....	35
3.1.3 Desafíos en la implementación de IA en el ámbito clínico.....	37
3.2 Modelos de Machine Learning para Clasificación de Imágenes Médicas	38
3.2.1 Redes neuronales convolucionales (CNNs) en el diagnóstico dermatológico	40
3.2.2 Comparación de modelos populares (ResNet, EfficientNet, MobileNet, etc.).....	41
3.2.3 Métricas de evaluación en modelos de clasificación de imágenes médicas.....	42
3.3 Bases de Datos de Imágenes Dermatológicas	44
3.3.1 Datasets públicos y privados en dermatología (HAM10000, ISIC, PH2, etc.).....	44

3.3.2 Problemas de sesgo en los conjuntos de datos médicos.....	45
3.4 Desarrollo de Aplicaciones Móviles para Diagnóstico Médico	47
3.4.1 Ventajas y desafíos del desarrollo multiplataforma en salud digital.....	50
3.4.2 SkinCheck como solución innovadora en la detección de lesiones.....	52
3.5 Servicios en la Nube para Procesamiento de Imágenes Médicas	53
3.5.1 Computación en la nube y sus modelos de servicio (IaaS, PaaS, SaaS).....	54
3.5.2 Uso de Firebase para autenticación y supabase para almacenamiento de datos médicos.....	55
Capítulo 4. Ética y Privacidad.....	57
4.1 Consideraciones Éticas y Regulatorias en el Uso de IA en Salud.....	57
4.1.1 Regulaciones y normativas en IA aplicada a la salud (HIPAA, GDPR, etc.).....	57
4.1.2 Responsabilidad médica y riesgos de falsos positivos/negativos en diagnósticos.....	59
4.1.3 Transparencia y explicabilidad en los modelos de IA para el sector médico.....	60
4.2 Consentimiento Informado y Transparencia	61
4.3 Equidad y No Discriminación	62
4.4 Supervisión Humana y Responsabilidad	62
Capítulo 5. Definición del Trabajo	63
5.1 Justificación.....	63
5.2 Objetivos.....	63
5.3 Metodología.....	64
5.3.1 Comparación de frameworks (Flutter, React Native, etc.).....	65
5.4 Planificación y Estimación Económica	66
5.5 Arquitectura Final Futura	71
Capítulo 6. Sistema Desarrollado	73
6.1 Diseño del desarrollo	73
6.1.1 Esquema y Arquitectura del Sistema.....	73
6.1.2 Diagrama de Casos de Uso	76
6.1.3 Requisitos Funcionales.....	78
6.1.4 Requisitos No Funcionales	79
6.1.5 Componentes del Proyecto	80
6.2 Modelo de Machine Learning.....	82
6.2.1 Recopilación de Datos.....	82

6.2.2 Limpieza y Preparación de Datos	83
6.2.3 Gestión de Atípicos	92
6.2.4 Selección de Características.....	93
6.2.5 Entrenamiento y Testeo del Modelo	93
6.2.6 Gestión de la Desproporción entre Clases	94
6.2.7 Elección del Modelo Final.....	103
6.3 Aplicación.....	104
6.3.1 Desarrollo del Backend	104
6.3.2 Desarrollo del Frontend	107
Capítulo 7. Análisis de Resultados.....	110
7.1 Resultados del modelo.....	110
7.1.1 Métricas Obtenidas de Cada Modelo.....	110
7.1.2 Gráficas de desempeño.....	114
7.1.3 Análisis Cualitativo	117
Capítulo 8. Conclusiones y Trabajos Futuros.....	120
8.1 Trabajos futuros.....	122
Bibliografía	123
ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS.....	128
ANEXO II: Manual De Instalación.....	131
Visual Studio Code	131
Flutter	131
Emulador Android.....	132
Verificación Del Entorno De Desarrollo.....	133
Configuración Del Desarrollo Para Android.....	133
Prueba, Debug y Deploy de La App	134
ANEXO III: Manual De Usuario de la App.....	135
Welcome	135
Log in y Register	136
PhotoCheck	137
Información	138

<i>Tipos de Lesiones:</i>	139
<i>Evaluación del Riesgo:</i>	141
<i>Rayos UV:</i>	144
<i>Citas Médicas:</i>	145
Historial	148
Perfil	149
<i>ANEXO IV: Lista De Abreviaturas</i>	150
<i>ANEXO IV: Términos Clave Y Sus Traducciones</i>	151

Índice de figuras

Ilustración 1. Pantallas PhotoCheck y Resultados.....	IX
Ilustración 2. Matrices de confusión test set.....	X
Ilustración 3. PhotoCheck and Results Screens.....	XIII
Ilustración 4. Confusion matrix test set.....	XIV
Ilustración 5. Logo de Python.....	24
Ilustración 6. Logo de Pandas.....	25
Ilustración 7. Logo de Pytorch	25
Ilustración 8. Logo de Scikit Learn	25
Ilustración 9. Logo de Dart en Flutter	25
Ilustración 10. Firebase Authentication.....	27
Ilustración 11. Cómo funciona Firebase.....	28
Ilustración 12. Logo Visual Studio Code	28
Ilustración 13. Logo Supabase.....	30
Ilustración 14. Logo Android Studio.....	31
Ilustración 15. Logo Github.....	32
Ilustración 16. Logo RenderNet	33
Ilustración 17. Modelo de aprendizaje automático.....	35
Ilustración 18. Clasificación de Machine Learning [31]	38
Ilustración 19. Tipos de Machine Learning [29]	39
Ilustración 20. Arquitectura de una CNN [30]	41
Ilustración 21. ISIC-ARCHIVE dataset [34].....	45
Ilustración 22. Eczemaless AI pantalla y logo	47
Ilustración 23. Tibot AI [40].....	48
Ilustración 24. DermExperto pantallas [41]	48
Ilustración 25. SkinVision pantallas y logo [42]	49
Ilustración 26. Arquitectura del Sistem	73
Ilustración 27. Diagrama de Casos de Uso.....	77
Ilustración 28. Requisitos Funcionales.....	79

Ilustración 29. Requisitos No Funcionales	80
Ilustración 30. Componentes del Proyecto	81
Ilustración 31. Funcionamiento modelo [37]	82
Ilustración 32. Dataset ISIC Archive [38]	83
Ilustración 33. Distribución por edad	84
Ilustración 34. Densidad por edad	84
Ilustración 35. Localización Anatómica	85
Ilustración 36. Distribución de Malignidad	85
Ilustración 37. Distribución de Lesiones Dermatológicas	86
Ilustración 38. Imágenes de Nevus	86
Ilustración 39. Imágenes de Melanoma	87
Ilustración 40. Imágenes de Carcinoma Basocelular	87
Ilustración 41. Imágenes Carcinoma Escamoso	88
Ilustración 42. Imágenes Querastosis Seborreica	88
Ilustración 43. Distribución de Género en Maligno	89
Ilustración 44. Distribución de Edad en Maligno	89
Ilustración 45. Distribución Anatómica en Maligno	89
Ilustración 46. Distribución de Género en Benigno	89
Ilustración 47. Distribución de Edad en Benigno	89
Ilustración 48. Distribución Anatómica en Benigno	90
Ilustración 49. Comparativa Género Maligno Benigno	90
Ilustración 50. Comparativa de Edad Maligno Benigno	91
Ilustración 51. Comparativa Anatómica Maligno Benigno	91
Ilustración 52. Dataframe Completo	92
Ilustración 53. Diagrama de Caja Edad Global	93
Ilustración 54. Arquitectura Modelo Lesión. Malignidad	103
Ilustración 55. Arquitectura Modelo Benigno	103
Ilustración 56. Firebase Authentication	104
Ilustración 57. Authentication Métodos de Acceso	105
Ilustración 58. Verificación de dirección del correo electrónico	105

Ilustración 59. Bases de Datos creadas.....	106
Ilustración 60. Imágenes guardadas de cada usuario en Supabase.....	106
Ilustración 61. Logo Canva	107
Ilustración 62. Logo App.....	107
Ilustración 63. Inicio.....	108
Ilustración 64. PhotoCheck.....	108
Ilustración 65. Información	109
Ilustración 66. Historial	109
Ilustración 67. Perfil	109
Ilustración 68. Classification Report on Train Lesion.....	111
Ilustración 69. Classification Report on Test Lesion	111
Ilustración 70. Classification Report on Train Mal	113
Ilustración 71. Classification Report on Test Mal.....	113
Ilustración 72. Matrices de Confusión Tipo de Lesión	115
Ilustración 73. Matrices de Confusión Malignidad	116
Ilustración 74. Análisis de Entrenamiento Lesiones	117
Ilustración 75. Análisis de Entrenamiento de Malignidad	118
Ilustración 76. Pantalla de Inicio	135
Ilustración 77. Pantalla de Log in y de Register.....	136
Ilustración 78. Pantalla de Photocheck.....	137
Ilustración 79. Pantalla Análisis	138
Ilustración 80. Pantalla de Información.....	139
Ilustración 81. Pantalla de Tipos de Lesiones	140
Ilustración 82. Encuesta Evaluación del Riesgo.....	142
Ilustración 83. Pantalla Índice UV	144
Ilustración 84. Pantalla Programar Citas Médicas.....	146
Ilustración 85. Pantalla Historial y 8un Análisis	148
Ilustración 86. Pantalla Mi Perfil.....	149

Índice de tablas

Tabla 2. Comparación de Frameworks.....	65
Tabla 3. Estimación Económica de Recursos Humanos	68
Tabla 4. Estimación Económica de Licencias y Herramientas	68
Tabla 5. Estimación Económica de Infraestructura Tecnológica	69
Tabla 6. Estimación Económica de Gastos Indirectos	70
Tabla 7. Estimación Económica Total.....	70
Tabla 8. Resumen Comparativo Resultados de Modelos	114

Capítulo 1. INTRODUCCIÓN

1.1 MOTIVACIÓN DEL PROYECTO

El proyecto surge como respuesta a una necesidad técnica y asistencial dentro del ámbito sanitario: la optimización del proceso de diagnóstico dermatológico mediante el uso de tecnologías basadas en inteligencia artificial. Actualmente, los servicios de dermatología enfrentan una demanda creciente, lo que conlleva a listas de espera prolongadas y, en consecuencia, a diagnósticos tardíos que pueden comprometer la eficacia del tratamiento.

La incorporación de herramientas automáticas de ayuda al diagnóstico puede contribuir a mitigar estos desafíos. Los modelos de aprendizaje automático han demostrado una alta capacidad para analizar imágenes médicas con precisión, siendo una solución viable para realizar una primera clasificación de lesiones cutáneas de forma rápida y escalable.

Este proyecto plantea el desarrollo de una aplicación móvil capaz de realizar un análisis preliminar de imágenes dermatológicas tomadas por el propio usuario, mediante un sistema de clasificación basado en redes neuronales convolucionales entrenadas con conjuntos de datos médicos. La aplicación proporciona una orientación inicial sobre el tipo de lesión y su posible malignidad. Esto permite una priorización de casos y una mejora en la gestión de recursos clínicos, derivando a los especialistas únicamente los casos que lo requieran.

Además de la mejora en eficiencia diagnóstica, la solución propuesta busca aumentar la accesibilidad a herramientas de cribado, especialmente en zonas con limitada cobertura dermatológica, fomentando un sistema sanitario más equitativo y tecnológicamente avanzado.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

2.1 LENGUAJES DE PROGRAMACIÓN

2.1.1 PYTHON

Python es un lenguaje de programación de alto nivel, interpretado y de tipado dinámico, conocido por su sintaxis clara y legible. Su versatilidad lo hace ideal para una amplia gama de aplicaciones, desde el desarrollo web y la automatización de tareas hasta el análisis de datos, la inteligencia artificial y la ciencia computacional. Además, cuenta con una extensa comunidad y un vasto ecosistema de librerías y *frameworks*, lo que facilita tanto la creación de prototipos rápidos como el desarrollo de soluciones complejas. [1]

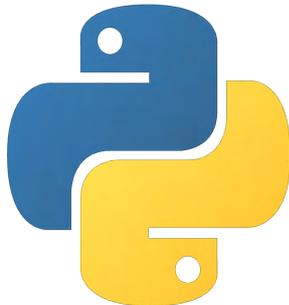


Ilustración 5. Logo de Python

Se ha utilizado Python para desarrollar los modelos de *machine learning*, aprovechando su robusto ecosistema de librerías especializadas. Herramientas como Pandas [2] han sido fundamentales para la manipulación, limpieza y análisis de los datos, mientras que bibliotecas

como NumPy [3] y Scikit-Learn [4] han permitido implementar y optimizar los algoritmos de clasificación. Además, se ha empleado PyTorch [5] como framework principal para la construcción y entrenamiento de redes neuronales profundas, lo que ha permitido definir arquitecturas personalizadas. Esta combinación de recursos ha facilitado el manejo eficiente de grandes volúmenes de información, esencial para entrenar modelos precisos y fiables en el ámbito del diagnóstico dermatológico.



*Ilustración 6. Logo de
Pandas*



Ilustración 8. Logo de Scikit Learn



Ilustración 7. Logo de Pytorch

2.1.2 FLUTTER Y DART

Flutter es un kit de desarrollo de interfaces creado por Google, que permite construir aplicaciones multiplataforma con una sola base de código. Gracias a sus widgets nativos y a un potente motor de renderizado propio, Flutter logra un alto rendimiento y transiciones fluidas en dispositivos Android, iOS, web y escritorio. Uno de los atractivos más relevantes de Flutter es el concepto de “*hot reload*”: una funcionalidad que acorta drásticamente los tiempos de desarrollo al permitir ver los cambios en tiempo real, sin necesidad de compilar todo el proyecto cada vez. [6]



Ilustración 9. Logo de Dart en Flutter

Dart es el lenguaje de programación oficial que da vida a Flutter. También creado por Google, se caracteriza por su sintaxis sencilla y familiar para quienes conocen lenguajes como Java o JavaScript. Dart está diseñado para la eficiencia en entornos clientes: puede compilarse tanto a código nativo como a JavaScript. Además, ofrece un recolector de basura optimizado y cuenta con un conjunto amplio de librerías estándar que facilitan la construcción de aplicaciones robustas. [7]

El uso de Flutter y Dart en conjunto trae consigo beneficios significativos. En primer lugar, reduce costes y tiempo de desarrollo al reutilizar la misma base de código para múltiples plataformas, manteniendo la consistencia en la experiencia de usuario. Por otro lado, su arquitectura declarativa y basada en widgets promueve una programación más ordenada y modular, lo que facilita escalar proyectos de forma estructurada y añadir nuevas funcionalidades sin sacrificar calidad.

2.2 FIREBASE

Firebase es una plataforma desarrollada por Google que ofrece una colección de servicios integrados para acelerar y simplificar el desarrollo de aplicaciones. Con un enfoque en aplicaciones móviles y web, facilita la gestión de la base de datos, la autenticación de usuarios, el alojamiento de archivos y la analítica, entre otras funcionalidades. Una de sus grandes ventajas es que permite concentrarse en la creación de la lógica de negocio y la experiencia de usuario, sin tener que preocuparse por la infraestructura o la escalabilidad del servidor.

Dentro de la plataforma, Firebase Realtime Database [8] y Cloud Firestore [9] destacan como soluciones de base de datos en tiempo real, permitiendo sincronizar datos de forma instantánea en todas las instancias de la aplicación. Esto resulta especialmente útil en escenarios donde se requiere colaboración en vivo o notificaciones instantáneas. Por otra parte, Firebase Authentication [10] ofrece un conjunto de métodos seguros para la gestión de usuarios, con la posibilidad de iniciar sesión utilizando correo y contraseña, redes sociales (Google, Facebook, etc.) o incluso proveedores de identidad personalizados.



Ilustración 10. Firebase Authentication

Otro componente notable de Firebase es Cloud Storage [11], que permite almacenar y servir archivos (imágenes, videos, documentos) de forma sencilla y escalable. Además, Cloud Functions [12] posibilita la ejecución de código en la nube en respuesta a eventos o peticiones concretas, sin necesidad de administrar servidores de manera manual. Por último, Firebase Analytics [13] ofrece estadísticas detalladas del comportamiento del usuario, lo que ayuda a tomar decisiones informadas para mejorar la aplicación con datos de uso reales.

En conjunto, todos estos servicios trabajan de forma integrada para brindar una experiencia de desarrollo fluida y robusta. Gracias a su escalabilidad y a la amplia comunidad de soporte, Firebase se ha convertido en una de las soluciones preferidas para construir aplicaciones modernas que necesitan una backend rápida, segura y confiable.

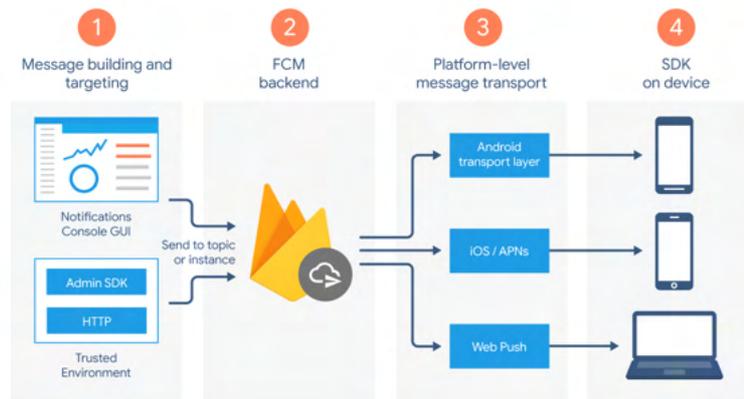


Ilustración 11. Cómo funciona Firebase

2.3 VISUAL STUDIO CODE



Ilustración 12. Logo Visual Studio Code

Visual Studio Code es un editor de código abierto y multiplataforma desarrollado por Microsoft, muy apreciado por su rapidez, ligereza y amplia capacidad de personalización. Cuenta con un sistema de extensiones muy robusto, lo que permite a la comunidad crear y compartir complementos para casi cualquier tecnología o lenguaje de programación. Esto lo convierte en una herramienta versátil que puede utilizarse tanto para el desarrollo web, como para aplicaciones móviles, entornos de escritorio o proyectos de *machine learning*. [14]

Entre sus características más destacadas se encuentra IntelliSense, un sistema de autocompletado inteligente que sugiere funciones, variables y propiedades a medida que se escribe el código. Adicionalmente, incluye un depurador integrado que facilita el proceso de detección y corrección de errores, y una terminal incorporada para ejecutar comandos sin salir del entorno de desarrollo. Asimismo, su diseño limpio y la posibilidad de personalizar temas, atajos y vistas hace que la experiencia de programar sea fluida y agradable.

Al trabajar con herramientas como Flutter, Visual Studio Code ofrece extensiones oficiales que facilitan la configuración y el autocompletado de las clases de widgets. De este modo, se acelera la creación de interfaces y se mejora la productividad general del desarrollador. Además, su perfecta integración con sistemas de control de versiones, como Git, agiliza la colaboración en equipo y el seguimiento de cambios. En conjunto, estas prestaciones hacen de Visual Studio Code uno de los editores preferidos para el desarrollo de aplicaciones modernas.

2.4 SUPABASE

Supabase es una plataforma de código abierto que proporciona una alternativa moderna y escalable a Firebase, especialmente en lo que respecta a la gestión de bases de datos. Está construida sobre PostgreSQL [15], lo que le otorga una gran capacidad de almacenamiento y consulta de datos de manera estructurada. Supabase se destaca por su simplicidad y facilidad de integración, permitiendo crear aplicaciones de base de datos completas sin necesidad de gestionar servidores ni infraestructura compleja. Una de sus características más destacadas es la capacidad de ofrecer funcionalidades en tiempo real, lo que facilita el desarrollo de aplicaciones interactivas y dinámicas que necesitan actualizaciones constantes de datos sin requerir recarga de la página. [16]



Ilustración 13. Logo Supabase

La integración de Supabase con Flutter es directa, lo que facilita el proceso de desarrollo, permitiendo una sincronización eficaz entre la aplicación móvil y la base de datos en la nube. Además, Supabase ofrece un sistema de autenticación robusto, lo que permite gestionar usuarios de manera segura y eficiente, garantizando la privacidad de los datos de los pacientes.

Supabase ofrece varias características clave como la autenticación y gestión de usuarios, base de datos relacional, almacenamiento de archivos y funciones en tiempo real.

2.5 *ANDROID STUDIO*

Android Studio es el entorno de desarrollo integrado (IDE) oficial para la creación de aplicaciones Android, basado en la plataforma IntelliJ IDEA de JetBrains [17]. Entre sus características más sobresalientes destacan el editor de código con soporte para autocompletado y refactorizaciones inteligentes, el sistema de creación de interfaces gráficas mediante el Layout Editor, así como el emulador de Android integrado, que facilita la prueba y depuración de las aplicaciones en diferentes versiones y tamaños de pantalla. [18]



Ilustración 14. Logo Android Studio

Gracias a su estrecha integración con el SDK de Android, Android Studio permite compilar, firmar y publicar fácilmente las aplicaciones en Google Play. A su vez, ofrece herramientas avanzadas para el seguimiento del rendimiento, el análisis de uso de memoria, la detección de cuellos de botella y la optimización de la velocidad de la app. Además, su sistema de control de versiones basado en Git o Subversion se integra sin problema con la interfaz de desarrollo, lo que facilita el trabajo en equipo. [19]

Por otra parte, Android Studio cuenta con características como el “instant run” (para ver cambios en la aplicación sin reinstalarla por completo en el emulador o en un dispositivo), plantillas de proyectos para casos de uso comunes (como actividades, fragmentos y vistas),

y un robusto soporte para lenguajes como Kotlin [20] y Java [21]. Estas prestaciones, junto a su comunidad activa, hacen de Android Studio el IDE preferido para desarrollar aplicaciones nativas con rendimiento y experiencia de usuario de alta calidad.

2.6 GITHUB

GitHub es una plataforma de alojamiento de código basado en la nube que utiliza el sistema de control de versiones Git como eje central. Ofrece herramientas para la colaboración entre equipos de desarrolladores, permitiendo rastrear los cambios en el código, fusionar aportaciones de diferentes ramas y revisar propuestas de manera ágil. Esta infraestructura facilita la transparencia y la organización de los proyectos, ya sean proyectos individuales, aplicaciones comerciales o iniciativas de código abierto.



Ilustración 15. Logo Github

Una de las mayores virtudes de GitHub es su comunidad activa, conformada por millones de desarrolladores de todo el mundo. A través de un modelo de “Pull Requests”, los colaboradores pueden sugerir cambios y correcciones, mientras que el dueño o los administradores de un repositorio deciden si integrarlos. Además, la plataforma ofrece herramientas como GitHub Actions [22], que permiten la automatización de flujos de trabajo, desde la ejecución de pruebas unitarias hasta la generación de compilaciones y despliegues en producción. Con su enfoque en la colaboración y la facilidad de uso, GitHub se ha convertido en una pieza fundamental para el desarrollo de software moderno. [23]

2.7 RENDERNET

RenderNet es una plataforma de cloud hosting que permite desplegar aplicaciones web, microservicios y APIs de manera sencilla, sin necesidad de gestionar la infraestructura subyacente. Ofrece compatibilidad con diversos entornos de ejecución, incluyendo

lenguajes como Python, Node.js [24], Ruby [25], Go [26] y otros, así como con bases de datos y static sites.



Ilustración 16. Logo RenderNet

La plataforma permite la implementación continua conectando directamente con repositorios en GitHub o GitLab [27], lo que facilita el despliegue automático tras cada push. RenderNet gestiona aspectos clave como el escalado automático, balanceo de carga, aprovisionamiento de dominios y certificados SSL, además de ofrecer monitoreo básico del estado del servicio.

Dispone de un plan gratuito con recursos limitados, en el que las aplicaciones pueden entrar en estado de suspensión por inactividad, lo que puede generar un pequeño retardo al ser reactivadas. Para proyectos más exigentes, RenderNet ofrece planes de pago con mayor disponibilidad, recursos dedicados y rendimiento optimizado. [28]

Capítulo 3. ESTADO DE LA CUESTIÓN

Ante la idea inicial de desarrollar un proyecto, siempre debemos realizarnos la pregunta: ¿hay algo similar en el mercado? ¿Hay algún trabajo de investigación que haya aportado los resultados que quiero alcanzar?

Es fundamental investigar y analizar el estado del arte para identificar brechas o áreas de mejora en proyectos similares. De esta manera, podemos asegurarnos de que nuestra propuesta tenga un valor diferencial y aporte algo nuevo al campo.

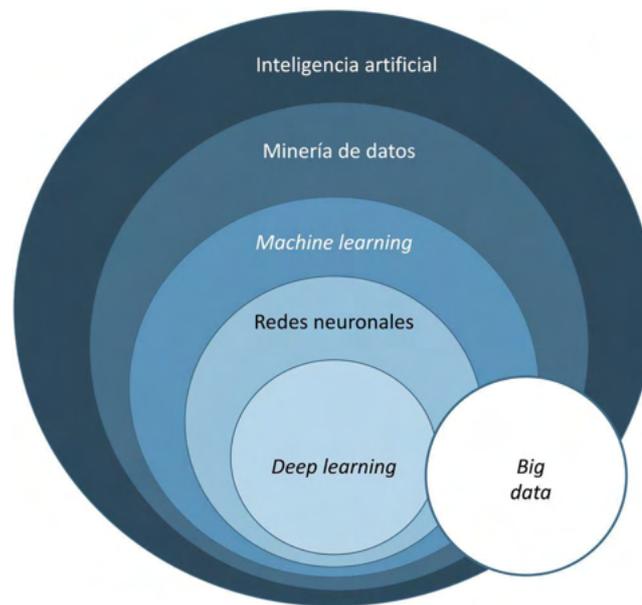
3.1 DIAGNÓSTICO DE ENFERMEDADES DERMATOLÓGICAS CON INTELIGENCIA ARTIFICIAL

3.1.1 PANORAMA ACTUAL DEL USO DE IA EN DERMATOLOGÍA

En los últimos años, la inteligencia artificial ha demostrado un gran potencial en el ámbito de la dermatología, especialmente en la identificación y clasificación de lesiones cutáneas. El desarrollo de modelos de aprendizaje profundo, como las redes neuronales convolucionales (CNNs), ha permitido que los sistemas de IA alcancen niveles de precisión comparables, e incluso superiores, a los dermatólogos experimentados en la detección de enfermedades como el melanoma, el carcinoma basocelular y otras patologías cutáneas.

Estudios recientes han mostrado que los algoritmos de IA pueden analizar imágenes dermatoscópicas y clasificar lesiones con alta precisión. Bases de datos como HAM10000 [29], ISIC [30] y PH2 [31] han sido fundamentales en el entrenamiento de estos modelos, proporcionando conjuntos de imágenes etiquetadas por expertos. Además, grandes compañías tecnológicas y *startups* han desarrollado aplicaciones móviles que permiten a los usuarios obtener una evaluación preliminar de sus lesiones a partir de una simple fotografía.

El uso de IA en dermatología no solo se limita a la clasificación de imágenes. También se están desarrollando herramientas que permiten la integración de estos modelos en sistemas hospitalarios para asistir en la toma de decisiones clínicas, priorizando casos urgentes y reduciendo la carga de trabajo de los especialistas. Sin embargo, a pesar de los avances, la adopción generalizada de estas tecnologías en la práctica médica aún enfrenta diversos desafíos técnicos, éticos y regulatorios.



Actas Dermosifiliogr. 2022;113:30-46

Ilustración 17. Modelo de aprendizaje automático [37]

3.1.2 BENEFICIOS DEL DIAGNÓSTICO ASISTIDO POR IA EN LA DETECCIÓN DE LESIONES

El diagnóstico asistido por IA en dermatología presenta múltiples ventajas tanto para pacientes como para profesionales de la salud. Entre los beneficios más destacados se encuentran: [29]

1. Mayor Precisión Diagnóstica

Los modelos de IA pueden analizar grandes volúmenes de datos y detectar patrones en imágenes dermatológicas con gran precisión. En algunos estudios, se ha demostrado que ciertas CNNs superan a dermatólogos en la clasificación de melanomas y carcinomas.

2. Accesibilidad y Detección Temprana

Muchas personas no tienen acceso inmediato a dermatólogos especializados, lo que retrasa el diagnóstico y tratamiento de enfermedades cutáneas. Las aplicaciones móviles impulsadas por IA pueden proporcionar evaluaciones preliminares en cuestión de segundos, permitiendo a los usuarios detectar lesiones sospechosas a tiempo y buscar atención médica si es necesario.

3. Optimización del Tiempo en Consultas Médicas

La IA puede filtrar y priorizar casos según su nivel de riesgo, ayudando a los dermatólogos a enfocarse en los pacientes con lesiones potencialmente malignas. Esto mejora la eficiencia en clínicas y hospitales, reduciendo el tiempo de espera para un diagnóstico especializado.

4. Reducción de Costos Médicos

Un diagnóstico asistido por IA puede disminuir la necesidad de biopsias innecesarias y consultas presenciales, optimizando los recursos sanitarios y reduciendo los costos asociados a pruebas y tratamientos.

5. Estandarización del Diagnóstico

La subjetividad en el diagnóstico clínico es un problema común en dermatología. Un sistema basado en IA permite ofrecer diagnósticos más objetivos y consistentes, disminuyendo la variabilidad entre distintos especialistas.

Gracias a estos beneficios, el diagnóstico asistido por IA se está convirtiendo en una herramienta clave para mejorar la detección y tratamiento de enfermedades de la piel. Sin embargo, su implementación a gran escala aún enfrenta ciertos retos.

3.1.3 DESAFÍOS EN LA IMPLEMENTACIÓN DE IA EN EL ÁMBITO CLÍNICO

A pesar del potencial de la IA en dermatología, su integración en la práctica médica presenta diversos desafíos que deben ser abordados antes de su adopción generalizada: [30]

- **Calidad y Sesgo en los Datos de Entrenamiento**

Muchos modelos de IA han sido entrenados con bases de datos que no reflejan la diversidad de la población. La mayoría de los datasets incluyen imágenes de piel clara, lo que puede llevar a sesgos en el diagnóstico y afectar la precisión en pacientes con tonos de piel más oscuros.

- **Regulación y Aprobación Médica**

La implementación de IA en salud requiere cumplir con regulaciones estrictas, como las normativas de la FDA (EE.UU), CE (Europa) y otras agencias sanitarias. La validación de modelos de IA en entornos clínicos es un proceso complejo que requiere estudios extensivos y aprobación por parte de organismos reguladores.

- **Explicabilidad y Confianza en los Modelos de IA**

Los sistemas de IA a menudo funcionan como “cajas negras”, lo que significa que sus decisiones no siempre son interpretables por los médicos. La falta de explicabilidad puede generar desconfianza en los especialistas y limitar su adopción. Se están desarrollando técnicas de IA explicable para mejorar la transparencia de estos modelos.

- **Integración con los Sistemas de Salud Existentes**

Para que la IA sea útil en entornos clínicos, debe integrarse con los sistemas de gestión hospitalaria y registros electrónicos de pacientes. Esto requiere compatibilidad con infraestructuras existentes y el cumplimiento de normativas de protección de datos como HIPAA y GDPR.

- **Responsabilidad Médica y Diagnósticos Erróneos**

En caso de un diagnóstico erróneo realizado por un modelo de IA, surge la pregunta: ¿quién es responsable? La falta de regulación clara sobre la responsabilidad médica en el uso de IA genera incertidumbre legal y limita su adopción en entornos clínicos.

- **Aceptación por Parte de los Profesionales de la Salud**

Muchos médicos ven la IA como una herramienta complementaria, pero otros pueden ser escépticos respecto a su fiabilidad y su impacto en la práctica médica. Es necesario capacitar a los profesionales de la salud en el uso de estas tecnologías y demostrar su utilidad a través de estudios clínicos validados.

A pesar de estos desafíos, la investigación y desarrollo de IA en dermatología continúa avanzando. Con la combinación adecuada de regulación, validación científica y mejoras en la calidad de los datos, es posible que en el futuro la IA se convierta en una herramienta estándar para la detección temprana y el tratamiento de enfermedades de la piel.

3.2 MODELOS DE MACHINE LEARNING PARA CLASIFICACIÓN DE IMÁGENES MÉDICAS

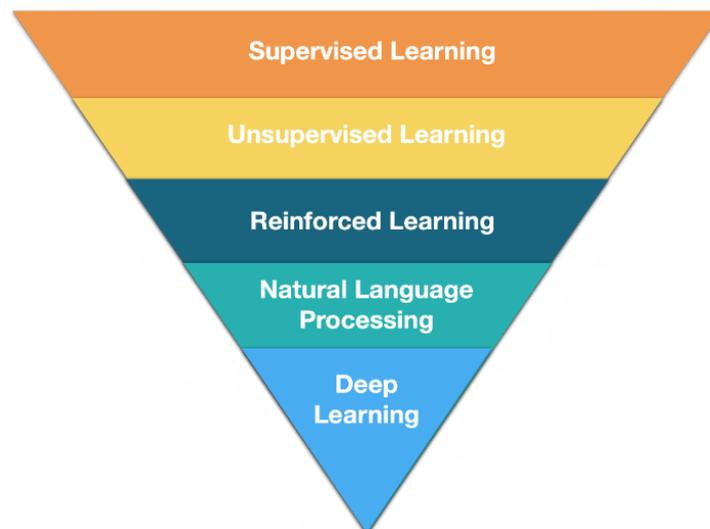


Ilustración 18. Clasificación de Machine Learning [31]

El avance en inteligencia artificial ha permitido el desarrollo de modelos de aprendizaje automático capaces de analizar imágenes médicas con una precisión comparable a la de los especialistas. En el campo de la dermatología, el uso de redes neuronales convolucionales (CNNs) ha revolucionado el diagnóstico de enfermedades cutáneas, mejorando la rapidez y precisión en la detección de lesiones benignas y malignas.

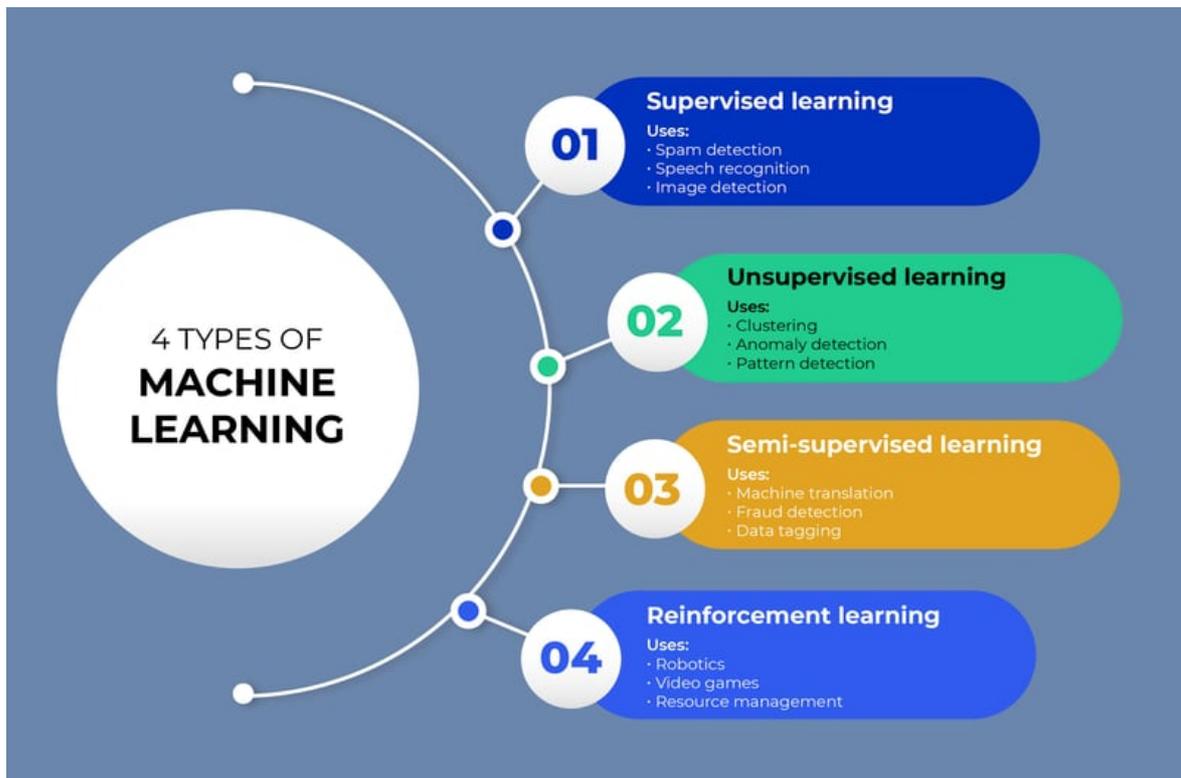


Ilustración 19. Tipos de Machine Learning [29]

A continuación, se describen los principales enfoques utilizados en la clasificación de imágenes dermatológicas, la comparación entre los modelos más populares y las métricas utilizadas para evaluar su desempeño.

3.2.1 REDES NEURONALES CONVOLUCIONALES (CNNs) EN EL DIAGNÓSTICO DERMATOLÓGICO

Las redes neuronales convolucionales (CNNs) han demostrado ser la arquitectura más eficiente para el análisis de imágenes médicas debido a su capacidad para extraer automáticamente patrones relevantes en imágenes sin requerir ingeniería de características manual. En el diagnóstico dermatológico, estas redes se entrenan con grandes volúmenes de imágenes de lesiones cutáneas, permitiendo clasificar diferentes tipos de anomalías con alta precisión.

El funcionamiento de las CNNs se basa en los siguientes elementos: [29]

- **Entrada:** La red recibe como entrada los píxeles de una imagen, representados por su altura, anchura y profundidad. Esta profundidad corresponde al número de canales de color: será 1 en imágenes en escala de grises o 3 en imágenes a color (rojo, verde y azul – RGB).
- **Capas convolucionales:** Esta capa analiza regiones locales de la imagen, es decir, grupos de píxeles contiguos. Utiliza filtros o *kernels* que recorren la imagen y calculan productos escalares entre sus pesos y los valores de los píxeles en esas regiones. El número de filtros utilizados determina el número de mapas de características generados como salida.
- **Capa Relu:** Aplica una función de activación a la salida de la capa convolucional, introduciendo no linealidad al modelo y eliminando valores negativos de los mapas de activación.
- **Capas de pooling o subsampling:** Reduce las dimensiones espaciales (alto y ancho) de los mapas de características, conservando la información más relevante. Esta operación permite disminuir la cantidad de parámetros y el coste computacional, sin afectar a la profundidad de la representación.
- **Capa tradicional:** Se trata de una red neuronal clásica que recibe como entrada la salida “aplanada” de la última capa de *pooling*. Estas capas conectan todas sus

neuronas con las de la capa siguiente y se encargan de realizar la clasificación final, generando una salida con tantas neuronas como clases se quieran predecir.

ARQUITECTURA DE UNA CNN

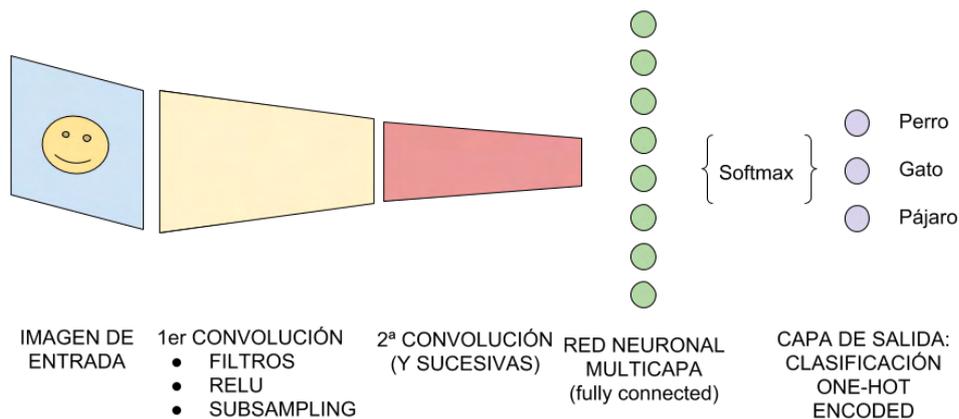


Ilustración 20. Arquitectura de una CNN [30]

Estudios recientes han demostrado que las CNNs pueden igualar o incluso superar el desempeño de los dermatólogos en la detección de lesiones malignas como el melanoma. Sin embargo, su eficacia depende en gran medida de la calidad y diversidad del conjunto de datos utilizado para su entrenamiento. [31]

3.2.2 COMPARACIÓN DE MODELOS POPULARES (RESNET, EFFICIENTNET, MOBILENET, ETC.)

En 2023, Wang et al. realizaron un estudio comparativo para clasificar nódulos pulmonares en imágenes de tomografía computarizada, evaluando tres arquitecturas de redes neuronales convolucionales: *ResNet50*, *MobileNetV2* y *EfficientNetB0*. El objetivo era analizar su rendimiento diagnóstico y su eficiencia computacional en un contexto clínico.

Los resultados revelaron que *EfficientNetB0* obtuvo la mayor precisión (98,62 %) con un equilibrio sobresaliente entre rendimiento y cantidad de parámetros. Por su parte, *MobileNetV2* destacó por su bajo coste computacional y velocidad de inferencia, siendo

especialmente adecuada para dispositivos con recursos limitados. *ResNet50*, aunque más exigente en términos de procesamiento, mantuvo una alta precisión y robustez gracias a sus conexiones residuales, permitiendo entrenar modelos profundos sin degradación del aprendizaje. [32]

Estos hallazgos coinciden con otras aplicaciones en el ámbito médico, incluyendo el diagnóstico dermatológico, donde también se emplean estas arquitecturas por sus respectivas ventajas. *ResNet* ha demostrado un excelente rendimiento en tareas de clasificación de imágenes complejas; *EfficientNet*, gracias a su escalado compuesto, maximiza la eficiencia en entornos clínicos con recursos moderados; y *MobileNet*, con sus convoluciones separables en profundidad, resulta especialmente útil en aplicaciones móviles como *SkinCheck*, donde se requieren modelos ligeros y rápidos.

La elección del modelo adecuado depende del equilibrio entre precisión deseada, disponibilidad de recursos computacionales y el entorno de implementación. En general, *MobileNet* es más apropiado para entornos con limitaciones técnicas, mientras que *EfficientNet* y *ResNet* se adaptan mejor a contextos clínicos con mayor capacidad de procesamiento y necesidad de alta precisión.

3.2.3 MÉTRICAS DE EVALUACIÓN EN MODELOS DE CLASIFICACIÓN DE IMÁGENES MÉDICAS

Para evaluar el desempeño de un modelo de clasificación de imágenes médicas, se utilizan diversas métricas que permiten medir su precisión, sensibilidad y capacidad para minimizar errores en el diagnóstico. Algunas de las métricas más importantes son:

Accuracy (Precisión global):

Mide el porcentaje de imágenes clasificadas correctamente. Útil en *datasets* balanceados, pero puede ser engañosa en conjuntos de datos desbalanceados.

Sensitivity / Recall (Sensibilidad o tasa de verdaderos positivos):

Mide la capacidad del modelo para detectar correctamente las lesiones malignas. Es crucial en aplicaciones médicas, donde la detección temprana de enfermedades es prioritaria.

Specificity (Especificidad o tasa de verdaderos negativos):

Evalúa la capacidad del modelo para evitar falsos positivos. Importante para reducir diagnósticos erróneos y evitar biopsias innecesarias.

F1-Score:

Promedio armónico entre precisión y sensibilidad. Útil cuando el *dataset* es desbalanceado y se busca un equilibrio entre falsos positivos y falsos negativos.

AUC-ROC (Área bajo la curva ROC):

Evalúa la capacidad del modelo para diferenciar entre lesiones benignas y malignas. Una mayor área bajo la curva indica un mejor desempeño en la clasificación. Estas métricas permiten validar la eficacia de un modelo antes de su implementación en entornos clínicos o aplicaciones móviles, por lo que un modelo con alta precisión pero baja sensibilidad podría no ser útil en un contexto médico, ya que podría pasar por alto casos críticos.

El uso de redes neuronales convolucionales en el diagnóstico dermatológico ha abierto nuevas posibilidades para la detección temprana de enfermedades cutáneas. Modelos como *ResNet*, *EfficientNet* y *MobileNet* han demostrado ser eficaces en la clasificación de imágenes médicas, y su evaluación mediante métricas como AUC-ROC, sensibilidad y precisión es fundamental para garantizar su fiabilidad en la práctica clínica.

La selección del modelo y su implementación en una aplicación como *SkinCheck* debe equilibrar la precisión con la eficiencia computacional, asegurando que la IA pueda ser utilizada tanto en dispositivos móviles como en entornos clínicos para mejorar el acceso al diagnóstico y la atención médica en dermatología.

3.3 BASES DE DATOS DE IMÁGENES DERMATOLÓGICAS

Para el entrenamiento y validación de modelos de inteligencia artificial en el diagnóstico dermatológico, es fundamental contar con bases de datos de imágenes médicas de alta calidad. Estos conjuntos de datos permiten a los algoritmos aprender patrones visuales en lesiones cutáneas y mejorar su precisión en la clasificación de enfermedades.

A continuación, se describen los principales *datasets* utilizados en dermatología, los desafíos asociados al sesgo en los datos y las consideraciones éticas y de privacidad en la recopilación de imágenes médicas.

3.3.1 DATASETS PÚBLICOS Y PRIVADOS EN DERMATOLOGÍA (HAM10000, ISIC, PH2, ETC.)

Existen varias bases de datos públicas y privadas que han sido utilizadas para entrenar modelos de *machine learning* en la detección de lesiones cutáneas. Algunas de las más relevantes incluyen:

HAM10000 (Human Against Machine with 10,000 images)

- Contiene 10,015 imágenes de lesiones cutáneas, clasificadas en siete tipos de enfermedades dermatológicas.
- Es uno de los *datasets* más utilizados en la investigación sobre IA aplicada a dermatología.
- Las imágenes provienen de diferentes poblaciones y dispositivos de captura, lo que mejora la generalización de los modelos entrenados con este conjunto de datos. [33]

ISIC (International Skin Imaging Collaboration)

- Es una de las mayores iniciativas para la recopilación de imágenes dermatológicas.
- Incluye miles de imágenes dermatoscópicas etiquetadas por expertos.
- Ha sido la base de múltiples competiciones de *machine learning* enfocadas en el diagnóstico del melanoma y otras lesiones cutáneas. [34]

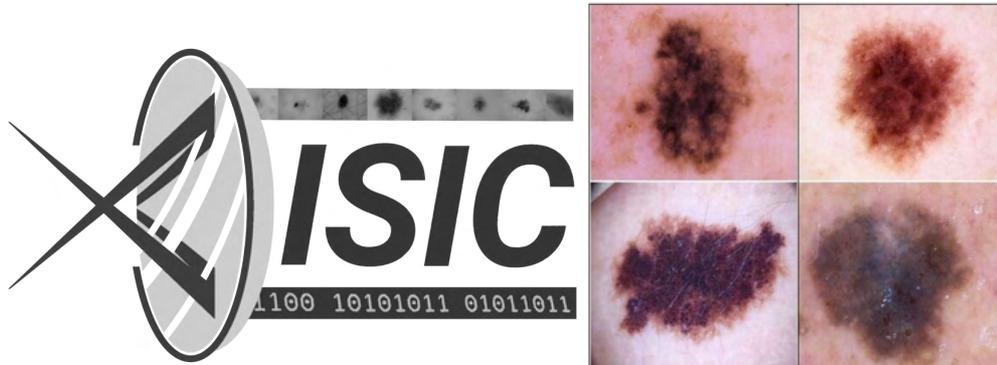


Ilustración 21. ISIC-ARCHIVE dataset [34]

PH2 (Pedro Hispano Dataset)

- Consta de 200 imágenes dermatoscópicas de alta calidad.
- Cada imagen está acompañada de anotaciones clínicas y segmentaciones manuales realizadas por dermatólogos.
- Es útil para el desarrollo de modelos de segmentación de lesiones en imágenes médicas. [35]

Además de estos conjuntos de datos públicos, existen bases de datos privadas utilizadas en hospitales y centros de investigación. Sin embargo, el acceso a estos recursos suele estar restringido por regulaciones de privacidad y confidencialidad médica.

3.3.2 PROBLEMAS DE SESGO EN LOS CONJUNTOS DE DATOS MÉDICOS

Como señalan Tschandl et al. [36], uno de los principales desafíos en la implementación de IA en dermatología es el sesgo en los conjuntos de datos utilizados para entrenar los modelos. Este problema puede afectar negativamente el desempeño del sistema y generar desigualdades en la calidad del diagnóstico. Algunos de los principales factores que contribuyen al sesgo incluyen:

Falta de Diversidad en el Tono de Piel:

La mayoría de los *datasets* dermatológicos contienen predominantemente imágenes de personas con piel clara. Esto puede llevar a una menor precisión en el diagnóstico de enfermedades en pacientes con piel más oscura, aumentando el riesgo de diagnósticos erróneos o tardíos en estas poblaciones.

Diferencias en los Dispositivos de Captura:

Algunas bases de datos incluyen imágenes tomadas con dermatoscopios, mientras que otras utilizan fotografías estándar. Un modelo entrenado con un solo tipo de imagen puede no generalizar bien a diferentes dispositivos de captura.

Desbalance en la Representación de Enfermedades:

Algunas enfermedades cutáneas son más comunes que otras, lo que puede resultar en conjuntos de datos desbalanceados. Si un modelo es entrenado con más imágenes de lesiones benignas que malignas, puede tener dificultades para detectar casos de alto riesgo, como el melanoma.

Etiquetado Inconsistente:

En algunos *datasets*, las imágenes son etiquetadas por diferentes expertos, lo que puede generar discrepancias en la clasificación de las lesiones. Esto puede afectar la precisión del modelo y su capacidad para generalizar en nuevos datos.

Para mitigar estos problemas, se han propuesto estrategias como la recolección de datos más diversa, el uso de técnicas de *data augmentation* para balancear clases y la aplicación de métodos de *fairness* en *machine learning* para reducir el sesgo en los modelos.

3.4 DESARROLLO DE APLICACIONES MÓVILES PARA DIAGNÓSTICO MÉDICO

El avance de las tecnologías móviles ha permitido la creación de aplicaciones destinadas a la detección y monitoreo de enfermedades, proporcionando herramientas de diagnóstico asistido por inteligencia artificial accesibles desde cualquier móvil. En el ámbito de la dermatología, estas aplicaciones permiten a los usuarios tomar fotografías de lesiones cutáneas y recibir una evaluación preliminar mediante modelos de *machine learning* entrenados con bases de datos médicas.

En un análisis realizado por Eczemaless, se destacan cinco aplicaciones de diagnóstico con IA que están revolucionando el cuidado de la piel. [39]

1. **EczemaLess AI:** especializada en el manejo del eccema, esta aplicación utiliza algoritmos avanzados para analizar síntomas, recomendar tratamientos personalizados, registrar brotes y ofrecer recursos de comportamiento y apoyo comunitario bajo supervisión clínica.

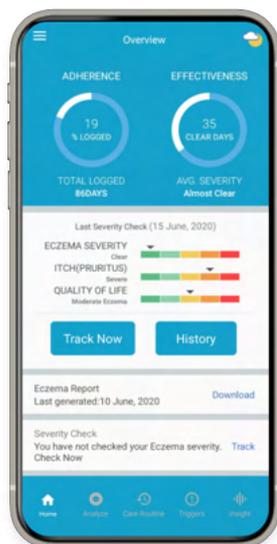


Ilustración 22. Eczemaless AI pantalla y logo

2. **Tibot AI:** su IA de última generación diagnostica múltiples afecciones (acné, psoriasis, cáncer de piel...), ofrece consultas virtuales con dermatólogos certificados y proporciona asesoramiento de tratamiento inteligente.



Ilustración 23. Tibot AI [40]

3. **DermExpert:** basada en algoritmos de última generación, permite subir imágenes de lesiones y obtener informes detallados sobre diagnóstico y recomendaciones prácticas.

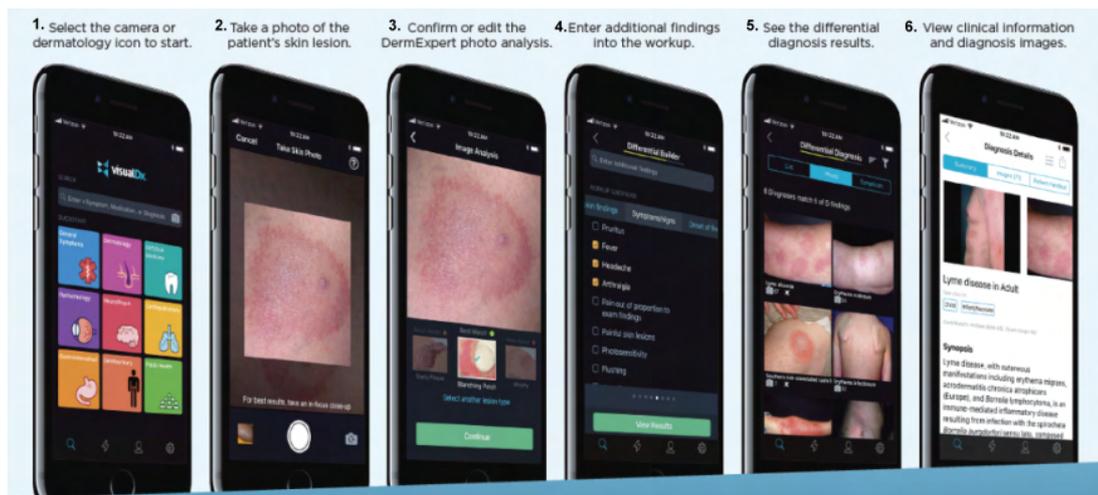
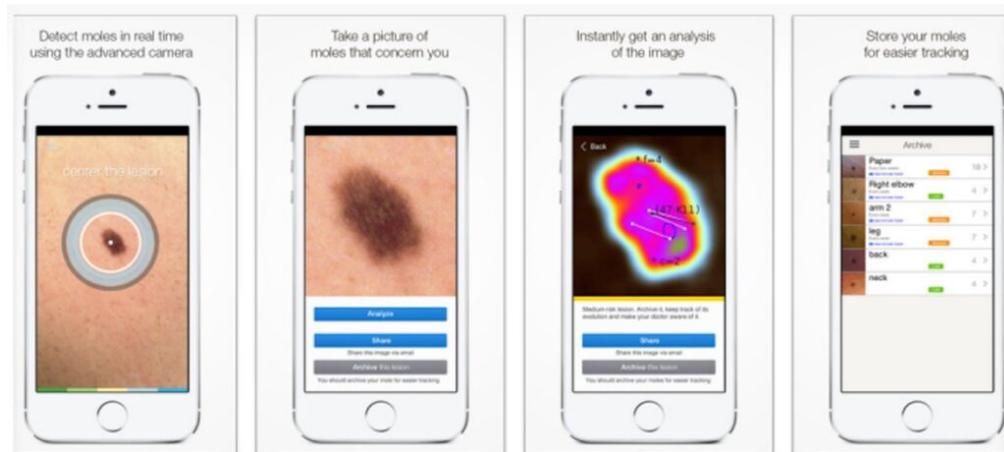


Ilustración 24. DermExperto pantallas [41]

4. **SkinVision**: pionera en la detección de melanoma, clasifica lesiones en riesgo bajo, medio o alto y brinda recomendaciones ágiles sobre el siguiente paso clínico.



SkinVision

Ilustración 25. SkinVision pantallas y logo [42]

Estas aplicaciones demuestran la capacidad de combinar uso móvil, accesibilidad y algoritmos avanzados de IA para aportar diagnósticos rápidos, personalizados y comparables a la evaluación dermatológica experta.

3.4.1 VENTAJAS Y DESAFÍOS DEL DESARROLLO MULTIPLATAFORMA EN SALUD DIGITAL

El desarrollo de aplicaciones médicas requiere una estrategia que garantice accesibilidad, eficiencia y compatibilidad con diferentes dispositivos. Una de las soluciones más utilizadas es el desarrollo multiplataforma, que permite crear una sola aplicación que funcione tanto en Android como en iOS.

Ventajas del Desarrollo Multiplataforma en Aplicaciones Médicas

1- Mayor Alcance y Accesibilidad

Al permitir su uso en múltiples sistemas operativos, las aplicaciones multiplataforma pueden llegar a un mayor número de usuarios sin la necesidad de desarrollar versiones separadas para cada plataforma.

2- Reducción de Costos y Tiempo de Desarrollo

La reutilización del código en diferentes sistemas operativos reduce los costos de desarrollo y mantenimiento. Se optimiza el tiempo de lanzamiento al mercado, lo que es fundamental en aplicaciones de salud que buscan ofrecer soluciones rápidas a los pacientes.

3- Consistencia en la Experiencia de Usuario

La interfaz y el funcionamiento de la aplicación se mantienen uniformes en diferentes dispositivos, mejorando la usabilidad y la satisfacción del usuario.

4- Actualizaciones Simultáneas

Las mejoras y correcciones pueden implementarse de manera centralizada sin necesidad de desarrollar actualizaciones separadas para Android e iOS.

5- Acceso a Medicina de Calidad en Zonas Rurales y Desatendidas

Al no requerir dispositivos específicos o costosos, las aplicaciones médicas multiplataforma permiten acercar servicios de diagnóstico y seguimiento clínico a poblaciones que tradicionalmente tienen menos acceso a atención médica, como comunidades rurales o zonas con infraestructura sanitaria limitada. Esto contribuye a reducir desigualdades en salud y promueve una atención más equitativa y universal.

Desafíos del Desarrollo Multiplataforma en Salud Digital

1- Rendimiento y Consumo de Recursos

En aplicaciones médicas que requieren procesamiento intensivo de imágenes, los *frameworks* multiplataforma pueden presentar limitaciones en comparación con las soluciones nativas.

Se debe optimizar el procesamiento para garantizar tiempos de respuesta rápidos, especialmente en diagnósticos en tiempo real.

2- Integración con Sensores y Hardware Médico

Algunas aplicaciones requieren acceso a cámaras de alta precisión, sensores de temperatura o dispositivos médicos específicos, lo que puede ser más complicado en un entorno multiplataforma.

3- Cumplimiento de Normativas de Seguridad y Privacidad

Las aplicaciones médicas deben cumplir con regulaciones como HIPAA [43] y GDPR [44], lo que implica implementar estrictos protocolos de seguridad en el almacenamiento y transmisión de datos.

A pesar de estos desafíos, los *frameworks* modernos han mejorado significativamente su rendimiento y compatibilidad con dispositivos médicos, permitiendo el desarrollo de aplicaciones eficaces para el diagnóstico asistido.

3.4.2 SKINCHECK COMO SOLUCIÓN INNOVADORA EN LA DETECCIÓN DE LESIONES

SkinCheck es una aplicación móvil diseñada para facilitar la detección temprana de lesiones cutáneas mediante el uso de inteligencia artificial. Su enfoque combina un modelo de *machine learning* entrenado con bases de datos dermatológicas y una interfaz intuitiva para el usuario, permitiendo realizar diagnósticos preliminares de manera rápida y accesible.

Características Principales de *SkinCheck*

1. Análisis Automático de Imágenes

Los usuarios pueden tomar una fotografía de su lesión cutánea y recibir un análisis basado en modelos de IA.

El sistema clasifica la lesión como benigna o maligna, además de proporcionar información sobre el tipo de anomalía detectada.

2. Desarrollo con Flutter y Firebase

Se ha elegido Flutter como framework para garantizar una experiencia fluida en dispositivos Android e iOS. Se utiliza Firebase para la autenticación de usuarios y el almacenamiento seguro de imágenes.

3. Privacidad y Seguridad de los Datos

Todas las imágenes son anonimizadas y almacenadas de acuerdo con normativas de privacidad médica. Los datos de los usuarios son protegidos mediante cifrado y acceso restringido.

4. Interfaz Intuitiva y Accesibilidad

La aplicación está diseñada para usuarios sin conocimientos médicos, permitiendo una navegación sencilla y resultados claros.

Impacto de SkinCheck en la Dermatología Digital

- **Accesibilidad Global:** Facilita el acceso a diagnósticos preliminares para personas en zonas con escasez de dermatólogos.
- **Detección Temprana:** Reduce el tiempo de espera para la identificación de lesiones potencialmente malignas.
- **Reducción de Costos:** Minimiza la necesidad de consultas médicas innecesarias, optimizando los recursos sanitarios.

El desarrollo de aplicaciones médicas multiplataforma ha permitido mejorar el acceso a herramientas de diagnóstico asistido por IA. *SkinCheck*, basada en Flutter y Firebase, representa una solución innovadora en la detección de lesiones dermatológicas, combinando precisión, accesibilidad y seguridad para brindar un diagnóstico preliminar eficiente a los usuarios.

3.5 *SERVICIOS EN LA NUBE PARA PROCESAMIENTO DE IMÁGENES MÉDICAS*

El uso de computación en la nube ha transformado la forma en que se procesan y almacenan datos en el ámbito de la salud. En aplicaciones médicas, la nube permite manejar grandes volúmenes de información, ejecutar modelos de inteligencia artificial con alto poder de cómputo y garantizar el acceso seguro a los datos de los pacientes.

En el caso de *SkinCheck*, la nube juega un papel fundamental en la autenticación de usuarios, el almacenamiento de imágenes dermatológicas y la ejecución de modelos de *machine learning* para diagnóstico en tiempo real. En esta sección, se describen los modelos de servicio en la nube, el uso de Firebase en la aplicación y la implementación de inferencia en la nube para mejorar la eficiencia del sistema.

3.5.1 COMPUTACIÓN EN LA NUBE Y SUS MODELOS DE SERVICIO (IAAS, PAAS, SAAS)

La computación en la nube permite acceder a recursos de procesamiento, almacenamiento y redes de manera escalable y bajo demanda [43]. Dependiendo del nivel de control y gestión de los recursos, existen tres modelos principales de servicio en la nube:

Infrastructure as a Service (IaaS):

Proporciona infraestructura virtualizada, como servidores, almacenamiento y redes [44]. Ejemplos: Google Cloud Compute Engine, AWS EC2, Microsoft Azure Virtual Machines. Útil para ejecutar modelos de *machine learning* con alto consumo de GPU en servidores remotos.

Platform as a Service (PaaS):

Ofrece una plataforma con herramientas y entornos preconfigurados para el desarrollo de aplicaciones [45]. Ejemplos: Google Cloud AI Platform, AWS SageMaker, Firebase. Permite entrenar y desplegar modelos de IA sin necesidad de administrar servidores físicos.

Software as a Service (SaaS):

Permite acceder a aplicaciones directamente desde la nube sin necesidad de instalación local [46]. Ejemplos: Google Drive, Dropbox, Microsoft OneDrive. En el caso de *SkinCheck*, se podría desarrollar como una aplicación SaaS accesible desde cualquier dispositivo.

Para *SkinCheck*, se utilizan soluciones PaaS y SaaS para la autenticación, almacenamiento de imágenes y procesamiento de IA en la nube, asegurando una infraestructura eficiente y escalable.

3.5.2 USO DE FIREBASE PARA AUTENTICACIÓN Y SUPABASE PARA ALMACENAMIENTO DE DATOS MÉDICOS

Firebase, un servicio de Google. Ha sido seleccionado para gestionar la autenticación de usuarios en *SkinCheck*, mientras que Supabase se encarga del almacenamiento de imágenes médicas. Esta combinación permite garantizar seguridad, escalabilidad y facilidad de implementación.

Autenticación de Usuarios con Firebase Authentication

Soporte para múltiples métodos de autenticación (Google, *email/password*). Gestión segura de sesiones y control de acceso a datos médicos. Cumplimiento con normativas de seguridad, aplicando cifrado y protección contra accesos no autorizados.

Almacenamiento de Imágenes Médicas con Supabase Storage

Las imágenes de lesiones dermatológicas se almacenan de forma segura en Supabase. Se generan URLs públicas de acceso, que son guardadas en la base de datos de Firebase para su consulta. Integración con *machine learning* para enviar imágenes automáticamente a un modelo de IA alojado en la nube.

La combinación de Firebase Authentication y Supabase garantiza que los datos de los usuarios sean manejados de forma segura y accesibles solo por los propietarios de la información, cumpliendo con normativas como HIPAA [47] y GDPR [48].

Pipeline de Inferencia en la Nube en *SkinCheck*

1. El usuario captura una imagen de la lesión desde la aplicación.
2. La imagen es enviada de forma segura a Supabase.
3. El *backend* procesa la imagen con un modelo de CNN previamente entrenado.
4. El resultado del análisis es devuelto a la aplicación con una clasificación de la lesión (benigna/maligna y tipo de lesión).

5. El usuario recibe el diagnóstico preliminar y una recomendación sobre si debe consultar a un dermatólogo.

Esta arquitectura garantiza una experiencia rápida y eficiente, permitiendo que *SkinCheck* ofrezca diagnósticos preliminares en cuestión de segundos sin comprometer la privacidad ni la seguridad de los datos del usuario.

El uso de servicios en la nube ha revolucionado la forma en que se procesan y almacenan imágenes médicas. *SkinCheck* aprovecha la computación en la nube para ofrecer una solución escalable y eficiente, utilizando Firebase para la autenticación y almacenamiento de datos.

Esta infraestructura permite ofrecer un diagnóstico rápido, seguro y accesible, facilitando la detección temprana de lesiones cutáneas y mejorando la experiencia del usuario sin requerir hardware especializado.

Capítulo 4. ÉTICA Y PRIVACIDAD

4.1 *CONSIDERACIONES ÉTICAS Y REGULATORIAS EN EL USO DE IA EN SALUD*

El desarrollo y uso de inteligencia artificial en el sector médico plantea desafíos éticos y regulatorios que deben ser abordados para garantizar la seguridad y confianza de los pacientes. La implementación de modelos de IA en salud no solo debe enfocarse en su precisión y eficiencia, sino también en el cumplimiento de normativas legales, la mitigación de riesgos asociados a errores en el diagnóstico y la transparencia en la toma de decisiones del sistema.

Una referencia clave en este contexto es el nuevo Reglamento de Inteligencia Artificial de la Unión Europea (AI Act) [49], aprobado en 2024, que establece un marco jurídico común para el desarrollo, comercialización y uso de sistemas de IA dentro del territorio europeo. Este reglamento clasifica las aplicaciones de IA en función del nivel de riesgo, siendo los sistemas utilizados en el ámbito de la salud considerados de alto riesgo. Por tanto, deben cumplir con requisitos estrictos en cuanto a seguridad, calidad de los datos, documentación técnica, supervisión humana y transparencia.

En esta sección se detallan las principales regulaciones aplicadas al uso de IA en el sector salud, la responsabilidad médica ante falsos positivos y negativos en los diagnósticos, y la importancia de la explicabilidad en los modelos de IA.

4.1.1 *REGULACIONES Y NORMATIVAS EN IA APLICADA A LA SALUD (HIPAA, GDPR, ETC.)*

El manejo de datos médicos está sujeto a estrictas normativas que buscan proteger la privacidad de los pacientes y garantizar que la información sensible no sea utilizada de manera indebida. Las principales regulaciones aplicables al uso de IA en salud incluyen:

HIPAA (*Health Insurance Portability and Accountability Act* - EE.UU.) [47]

Regula la protección y privacidad de los datos médicos en Estados Unidos. Establece requisitos para el almacenamiento y transmisión segura de información de salud. Aplica a cualquier sistema que procese datos médicos, incluyendo aplicaciones basadas en IA como *SkinCheck*.

GDPR (*General Data Protection Regulation* - Unión Europea) [48]

Garantiza la privacidad y control de los datos personales de los ciudadanos de la UE. Obliga a las empresas a solicitar consentimiento explícito antes de recopilar datos médicos. Exige que los pacientes puedan eliminar o solicitar acceso a su información en cualquier momento.

Regulación MDR (*Medical Device Regulation* - UE) [49]

Clasifica las aplicaciones médicas basadas en IA como dispositivos médicos, lo que las somete a estrictos controles de seguridad y validación. Exige que cualquier sistema de IA utilizado en diagnóstico clínico sea auditado y aprobado antes de su implementación en hospitales.

Otras Normativas Importantes

FDA (*Food and Drug Administration* - EE.UU) [50]: Regula el uso de software médico y sistemas de IA en el diagnóstico clínico. Ley de Protección de Datos Biométricos (BIPA - EE.UU) [51]: Regula el uso de datos biométricos, incluyendo imágenes médicas.

Para que *SkinCheck* cumpla con estas regulaciones, se han implementado medidas como anonimización de imágenes, encriptación de datos y consentimiento explícito antes de procesar cualquier información médica.

4.1.2 RESPONSABILIDAD MÉDICA Y RIESGOS DE FALSOS POSITIVOS/NEGATIVOS EN DIAGNÓSTICOS

Uno de los principales desafíos en el uso de IA en salud es la precisión del diagnóstico y las consecuencias de posibles errores en la clasificación de enfermedades.

Falsos Positivos

Ocurren cuando el sistema clasifica una lesión benigna como maligna. Puede generar ansiedad innecesaria en los pacientes y llevar a procedimientos médicos innecesarios (biopsias, cirugías). Puede aumentar los costos en el sistema de salud debido a diagnósticos erróneos.

Falsos Negativos

Ocurren cuando la IA no detecta una lesión maligna, clasificándola erróneamente como benigna. Representa un riesgo grave para la salud del paciente, ya que podría retrasar el diagnóstico y tratamiento de enfermedades como el melanoma. Puede derivar en problemas legales y éticos si se demuestra que el error del sistema perjudicó la salud de una persona.

Responsabilidad Médica y Legalidad del Diagnóstico Asistido por IA

¿Quién es responsable si la IA comete un error en el diagnóstico?

- El desarrollador del sistema (por fallos en el modelo de IA).
- El médico que interpreta el resultado (si confía ciegamente en la IA sin validación adicional).
- La institución de salud que adopta el sistema sin realizar pruebas adecuadas.

La mayoría de regulaciones actuales consideran que la IA debe ser utilizada como herramienta de apoyo, pero la decisión final debe recaer en un profesional de la salud.

Para minimizar riesgos, *SkinCheck* incluirá un mensaje claro en sus resultados indicando que el análisis de IA no reemplaza el diagnóstico médico y recomendando consultar a un especialista en caso de dudas.

4.1.3 TRANSPARENCIA Y EXPLICABILIDAD EN LOS MODELOS DE IA PARA EL SECTOR MÉDICO

Uno de los desafíos más importantes en la implementación de IA en salud es la necesidad de que los modelos sean interpretables y explicables para médicos y pacientes.

El Problema de la “Caja Negra” en la IA Médica

Muchos modelos de *deep learning* funcionan como cajas negras, donde el usuario no puede entender cómo se llegó a una decisión. Esto puede generar desconfianza en los profesionales médicos, quienes necesitan interpretar el diagnóstico antes de tomar una decisión clínica.

Soluciones para Mejorar la Transparencia en Modelos de IA

1. IA Explicable (*Explainable AI - XAI*)

Utiliza técnicas como mapas de calor para visualizar qué áreas de la imagen influyeron en la decisión del modelo. Permite que los médicos validen si el modelo está analizando correctamente la lesión dermatológica.

2. Reportes Detallados en los Diagnósticos

En lugar de solo indicar "benigno" o "maligno", la IA debe proporcionar información adicional, como nivel de confianza del modelo, comparación con casos similares y recomendaciones basadas en evidencia.

3. Validación Clínica del Modelo

Los modelos de IA deben ser validados con estudios clínicos antes de ser utilizados en entornos reales. Para garantizar su confiabilidad, *SkinCheck* se basará en *datasets*

dermatológicos reconocidos y en métricas de evaluación validadas (sensibilidad, especificidad, AUC-ROC).

4. Ética en la Toma de Decisiones Automáticas

Se debe evitar que la IA tome decisiones médicas definitivas sin intervención humana. La responsabilidad final del diagnóstico debe seguir recayendo en médicos capacitados.

El uso de IA en salud debe cumplir con estrictos principios éticos y normativas legales para garantizar su seguridad y fiabilidad. *SkinCheck* implementará medidas de seguridad como anonimización de datos, cumplimiento de regulaciones como HIPAA y GDPR, y herramientas de explicabilidad de IA para mejorar la transparencia del sistema.

Además, se enfatiza que la IA no reemplaza el juicio clínico de un especialista, sino que funciona como una herramienta de apoyo para la detección temprana de enfermedades cutáneas.

4.2 **CONSENTIMIENTO INFORMADO Y TRANSPARENCIA**

Aunque los datos utilizados son públicos y anonimizados, es esencial destacar la importancia del consentimiento informado en la recopilación y uso de datos médicos. En aplicaciones futuras que impliquen la recopilación de datos directamente de los usuarios, se deberá asegurar que los pacientes comprendan claramente cómo se utilizarán sus datos, los fines del procesamiento y sus derechos en relación con la información proporcionada.

La transparencia en el funcionamiento de los sistemas de IA también es crucial. Los usuarios deben ser informados de que están interactuando con un sistema automatizado y deben tener acceso a explicaciones comprensibles sobre cómo se toman las decisiones diagnósticas. Esto no solo cumple con principios éticos, sino que también fortalece la confianza en la tecnología.

4.3 EQUIDAD Y NO DISCRIMINACIÓN

Un desafío significativo en el desarrollo de sistemas de IA en salud es evitar sesgos que puedan conducir a decisiones discriminatorias. Es fundamental que los conjuntos de datos utilizados para entrenar los modelos sean representativos de la diversidad de la población, incluyendo variaciones en edad, género, etnia y otras características relevantes. La falta de representatividad puede resultar en diagnósticos menos precisos para ciertos grupos, perpetuando desigualdades existentes en el sistema de salud.

4.4 SUPERVISIÓN HUMANA Y RESPONSABILIDAD

La implementación de sistemas de IA en el ámbito médico debe complementarse con la supervisión de profesionales de la salud. La IA debe ser vista como una herramienta de apoyo que mejora la capacidad diagnóstica, pero no sustituye el juicio clínico. Además, es necesario establecer mecanismos claros de responsabilidad en caso de errores o fallos en el sistema, asegurando que existan protocolos para la revisión y corrección de decisiones automatizadas.

Capítulo 5. DEFINICIÓN DEL TRABAJO

5.1 JUSTIFICACIÓN

La creciente incidencia de enfermedades dermatológicas, sumada a la importancia de un diagnóstico precoz, hace indispensable contar con herramientas que faciliten la detección y clasificación de lesiones cutáneas. El proyecto "Skin Check" surge como respuesta a la necesidad de mejorar el acceso a diagnósticos de calidad, especialmente en áreas con limitaciones en la disponibilidad de especialistas. La aplicación se fundamenta en técnicas de inteligencia artificial y machine learning para analizar imágenes dermatológicas y determinar la probabilidad de malignidad, lo que puede contribuir significativamente a la prevención y tratamiento oportuno. Además, el desarrollo de esta herramienta representa una oportunidad para integrar tecnologías emergentes en el ámbito de la salud, potenciando la eficiencia del proceso diagnóstico y fomentando la innovación en el sector sanitario.

5.2 OBJETIVOS

El objetivo principal del proyecto es diseñar y desarrollar una aplicación móvil que permita a los usuarios capturar imágenes de lesiones dermatológicas y obtener una evaluación preliminar sobre su posible naturaleza maligna o benigna. Para alcanzar este fin, se plantean los siguientes objetivos específicos:

- **Desarrollar un modelo de *machine learning* preciso y eficiente** que clasifique imágenes de lesiones cutáneas en categorías clínicas relevantes (maligno, benigno, intermedios), optimizando su rendimiento para su integración en dispositivos móviles.
- **Implementar una aplicación móvil en Flutter con una interfaz accesible**, que permita la captura, carga y análisis de imágenes, e integre servicios en la nube para autenticación, almacenamiento y gestión de datos.

- **Validar el sistema con datos reales y elaborar documentación técnica y de usuario**, asegurando la utilidad clínica preliminar, la reproducibilidad del proyecto y su posible extensión en futuros desarrollos.

5.3 METODOLOGÍA

Para alcanzar los objetivos planteados se propone una metodología de desarrollo ágil que combine fases de investigación, desarrollo técnico y validación. El proceso se estructura en las siguientes etapas:

1. **Revisión bibliográfica y análisis del estado del arte:** Se realizará un estudio exhaustivo de las técnicas de procesamiento de imágenes y algoritmos de *machine learning* aplicados al diagnóstico dermatológico, identificando las mejores prácticas y posibles desafíos.
2. **Definición de requisitos y diseño del sistema:** Con base en la revisión bibliográfica, se establecerán los requerimientos funcionales y no funcionales de la aplicación. Se diseñará la arquitectura general del sistema, incluyendo la integración entre la aplicación móvil, el *backend* y el modelo de inteligencia artificial.
3. **Desarrollo del modelo de *machine learning*:** Se recopilarán y preprocesarán conjuntos de datos de imágenes dermatológicas. Se procederá a la selección y entrenamiento del modelo, realizando iteraciones de validación y ajuste para optimizar la precisión y robustez.
4. **Implementación de la aplicación móvil:** Se desarrollará la aplicación utilizando Flutter, integrando funcionalidades de captura, procesamiento y envío de imágenes. Se incorporará la autenticación y gestión de datos a través de Firebase.
5. **Integración y pruebas del sistema:** Se integrarán el modelo de *machine learning* y la aplicación móvil, realizando pruebas unitarias y de integración para garantizar la

correcta comunicación entre componentes. Se planificarán pruebas piloto con usuarios para evaluar la experiencia y la precisión del diagnóstico.

6. **Evaluación y refinamiento:** A partir de los resultados obtenidos en las pruebas, se realizarán ajustes y mejoras en el modelo y la interfaz, asegurando que el sistema cumpla con los estándares de calidad y eficacia requeridos.
7. **Documentación y difusión de resultados:** Se preparará la documentación técnica y de usuario, y se planificarán actividades de divulgación científica y potenciales colaboraciones con centros de salud.

5.3.1 COMPARACIÓN DE FRAMEWORKS (FLUTTER, REACT NATIVE, ETC.)

Existen varias opciones para el desarrollo de aplicaciones multiplataforma en el ámbito de la salud digital. A continuación, se comparan los *frameworks* más utilizados:

Framework	Lenguaje	Ventajas	Desventajas
Flutter [6]	Dart	Alto rendimiento, interfaz nativa y compatibilidad con machine learning.	Mayor curva de aprendizaje, comunidad más pequeña.
React Native [52]	JavaScript	Amplia comunidad, integración con APIs nativas, uso de JavaScript.	Menor rendimiento en tareas intensivas, mayor uso de recursos.
Xamarin [53]	C#	Soporte de Microsoft, integración con Visual Studio.	Apps más pesadas, requiere conocimientos de .NET.

Tabla 1. Comparación de Frameworks

Flutter ha ganado gran popularidad en el desarrollo de aplicaciones médicas debido a su alto rendimiento y compatibilidad con modelos de *machine learning*, lo que lo convierte en una opción ideal para aplicaciones como *SkinCheck*.

5.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

La planificación del proyecto se distribuye en varias fases que abarcan desde la investigación inicial hasta la fase de validación y documentación final. Se establece un cronograma tentativo de 9 a 12 meses, dividido de la siguiente manera:

- **Fase 1: Investigación y diseño (Meses 1-2)**
 - Revisión bibliográfica y definición de requisitos.
 - Elaboración del diseño de la arquitectura del sistema.
- **Fase 2: Desarrollo del modelo y la aplicación (Meses 3-7)**
 - Recolección y preprocesamiento de datos.
 - Entrenamiento y validación del modelo de *machine learning*.
 - Desarrollo de la aplicación móvil y su integración con Firebase.
- **Fase 3: Pruebas y validación (Meses 8-9)**
 - Realización de pruebas de integración y pruebas piloto.
 - Recopilación de feedback y ajustes en el sistema.
- **Fase 4: Documentación y difusión (Meses 10-12)**
 - Elaboración de la documentación técnica y de usuario.
 - Preparación de materiales para la presentación y difusión del proyecto.

Estimación económica:

- **Recursos humanos:** Coste asociado a horas de trabajo del equipo de desarrollo, incluyendo programadores, especialistas en *machine learning* y colaboradores clínicos.
- **Licencias y herramientas:** Gastos en software y licencias necesarias para el desarrollo, así como el uso de servicios en la nube (por ejemplo, Firebase y Supabase) durante el periodo de desarrollo y pruebas.
- **Infraestructura tecnológica:** Inversión en hardware (por ejemplo, dispositivos móviles para pruebas, servidores de desarrollo) y recursos de almacenamiento de datos.
- **Gastos indirectos:** Incluyen costos de comunicación, formación y difusión del proyecto.

Recursos Humanos:

Concepto	Descripción	Cantidad	Coste Unitario	Total	Observaciones
Programador	Desarrollo de la app en Flutter	1	2000	2000	Contratación o asignación de horas dedicadas
Especialista en ML	Entrenamiento y validación del modelo	1	1500	1500	Incluye optimización y ajustes continuos

Colaborador clínico	Asesoría en validación y pruebas con datos reales	1	1000	1000	Consultoría part-time
---------------------	---	---	------	------	-----------------------

Tabla 2. Estimación Económica de Recursos Humanos

Licencias y Herramientas:

Concepto	Descripción	Cantidad	Coste Unitario	Total	Observaciones
Firestore y Supabase	Subscripción y uso de servicios en la nube (almacenamiento y autenticación)	1	200	200	Plan básico/mediano según necesidades
Software de Diseño y Edición	Herramientas para diseño UI/UX y edición de imágenes	1	150	150	Licencias temporales
IDE y Herramientas de Desarrollo	Licencias para IDEs y plugins necesarios	1	150	150	Ejemplo: JetBrains o similares

Tabla 3. Estimación Económica de Licencias y Herramientas

Infraestructura tecnológica:

Concepto	Descripción	Cantidad	Coste Unitario	Total	Observaciones
Dispositivos móviles	Dispositivos móviles para pruebas y validación	2	300	600	Equipos de gama media para testeo
Servidor en la Nube	Recursos para entrenamiento y pruebas del modelo	1	400	400	Servidor virtual o plan de <i>cloud computing</i>
Equipo de Pruebas	Equipos adicionales (cámaras, accesorios, etc.)	1	300	300	Recursos complementarios para ensayos prácticos

Tabla 4. Estimación Económica de Infraestructura Tecnológica

Gastos Indirectos:

Concepto	Descripción	Cantidad	Coste Unitario	Total	Observaciones
Comunicaciones	Servicios de internet y herramientas de colaboración	1	200	200	Incluye plataformas para reuniones virtuales
Materiales de Difusión	Producción de folletos, presentaciones y otros materiales promocionales	1	150	150	Materiales para presentaciones y divulgación del proyecto
Formación y Capacitación	Cursos o workshops en ML, desarrollo móvil y tecnología	1	100	100	Formación complementaria para el equipo
Imprevistos	Otros gastos imprevistos o de contingencia	1	100	100	Reserva para imprevistos

Tabla 5. Estimación Económica de Gastos Indirectos

Total general:	6.850 €
----------------	----------------

Tabla 6. Estimación Económica Total

Se prevé que la inversión total se mantenga en un rango moderado, aprovechando recursos ya disponibles y potenciales colaboraciones con centros de investigación y salud. Un análisis detallado de cada partida permitirá optimizar el uso de los fondos, garantizando la viabilidad y sostenibilidad del proyecto.

5.5 ARQUITECTURA FINAL FUTURA

Aunque durante el desarrollo del proyecto se ha optado por soluciones gratuitas tanto en Render como en Firebase o Supabase, que permiten probar y validar la funcionalidad del sistema, en un entorno profesional sería necesario desplegar la aplicación utilizando arquitecturas más robustas, escalables y seguras que garanticen un rendimiento óptimo y disponibilidad continua.

Una posible arquitectura de producción basada en servicios *cloud* y herramientas de uso profesional sería:

- **Frontend**

La aplicación móvil desarrollada en Flutter se mantendría como cliente principal, encargada de capturar imágenes, metadatos y mostrar los resultados al usuario. Se conectaría a una API propia desplegada en la nube y a servicios externos si fuera necesario.

- **Backend**

Los modelos de *machine learning* se desplegarían como servicios independientes en contenedores *Docker* usando una plataforma como Google Cloud Run, AWS ECS o Azure Container Apps. Esto permitiría escalar automáticamente el número de instancias dependiendo de la carga, y además facilitaría el versionado de modelos.

- **Servicio de almacenamiento de imágenes**

Las imágenes de entrada serían almacenadas en un servicio de almacenamiento escalable como Amazon S3, Google Cloud Storage o Azure Blob Storage, lo que permitiría acceso rápido, seguro y económico a gran escala.

- **Base de datos**

Para almacenar usuarios, diagnósticos y metadatos asociados, se recomienda utilizar un sistema de base de datos gestionado como Firebase Firestore, Amazon DynamoDB o una base de datos relacional como PostgreSQL desplegada en Google Cloud SQL o AWS RDS, dependiendo de los requisitos estructurados del proyecto.

- **Autenticación y seguridad**

El sistema de autenticación debería estar respaldado por un servicio como Firebase Authentication, Auth0 o Amazon Cognito, que permite gestionar usuarios, permisos y sesiones de forma segura y conforme al RGPD.

- **Ciclo de vida de modelos**

Para mantener y actualizar los modelos de machine learning en producción, se recomienda integrar una solución de MLOps como Vertex AI, SageMaker, o herramientas open source como MLflow o Kubeflow, que permiten versionar modelos, monitorizar su rendimiento y automatizar el reentrenamiento.

Esta arquitectura permitiría a la aplicación escalar horizontalmente, soportar miles de usuarios concurrentes y mantener los modelos de IA actualizados y desplegados sin interrumpir el servicio. Además, el uso de servicios gestionados reduce significativamente los esfuerzos de mantenimiento y mejora la seguridad general del sistema.

Capítulo 6. SISTEMA DESARROLLADO

6.1 DISEÑO DEL DESARROLLO

6.1.1 ESQUEMA Y ARQUITECTURA DEL SISTEMA

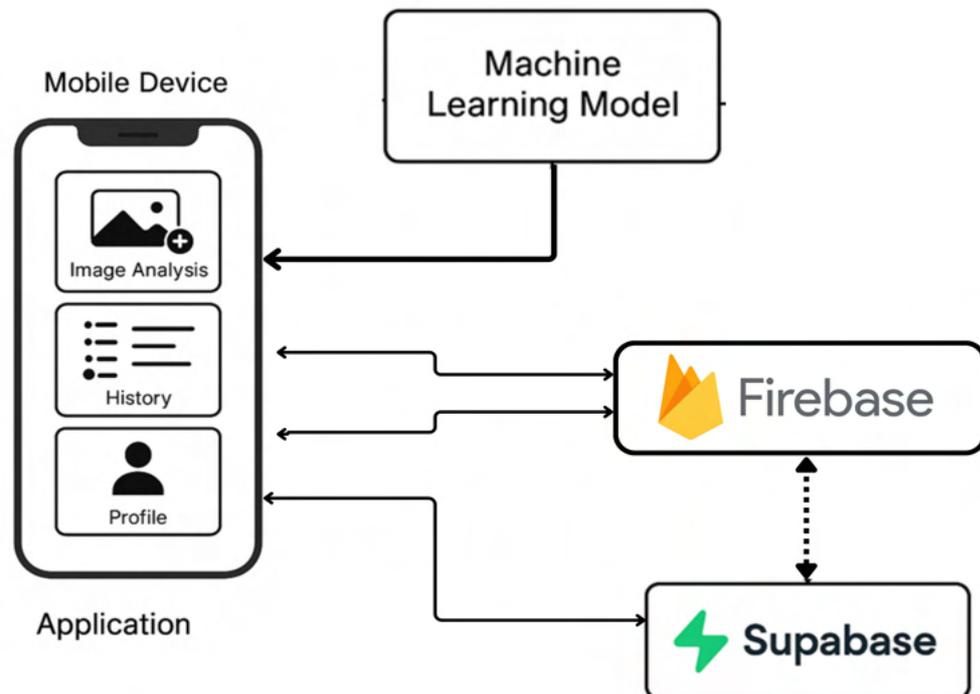


Ilustración 26. Arquitectura del Sistem

SkinCheck está diseñada como una plataforma integral para el análisis de lesiones dermatológicas mediante un modelo de Machine Learning (ML). La arquitectura del sistema se basa en una estructura cliente-servidor con integración de tecnologías modernas y escalables. Esta arquitectura se compone de tres capas principales:

Capa de *Frontend* (Cliente): La aplicación móvil, desarrollada en Flutter, permite al usuario interactuar de manera sencilla y directa. La aplicación se compone de cuatro pantallas principales:

1. **Pantalla de análisis de imagen:** Permite al usuario cargar una imagen de una lesión dermatológica para ser procesada por el modelo de ML y obtener un diagnóstico.
2. **Pantalla de historial:** Muestra los análisis previos realizados por el usuario, con detalles sobre el diagnóstico y las imágenes asociadas.
3. **Pantalla de perfil:** Permite a los usuarios gestionar sus datos dentro de la aplicación.
4. **Pantalla de información:** Contiene cuatro secciones informativas:
 - Descripción e imágenes de las lesiones dermatológicas.
 - Test de riesgo de piel para evaluar la predisposición a enfermedades cutáneas.
 - Calendario de citas para que el usuario pueda agendar visitas con profesionales.
 - Información sobre rayos UV y su impacto en la salud de la piel.

Capa de *Backend*: El *backend* de *SkinCheck* está encargado de procesar las imágenes cargadas por el usuario, ejecutar los modelos de *machine learning* para generar el diagnóstico y gestionar la lógica de negocio relacionada con la autenticación y el acceso a los datos. El *backend* se comunica con Firebase para la autenticación de usuarios y Supabase para almacenar tanto las imágenes como el modelo.

Capa de Base de Datos y Almacenamiento: Firebase se utiliza para gestionar la autenticación de los usuarios, permitiendo que los usuarios se registren e inicien sesión de manera segura. También se utiliza como base de datos, haciendo uso de Firebase Authentication y Firestore Database.

Supabase almacena las imágenes de las lesiones dermatológicas, y otros datos importantes del usuario.

El flujo de trabajo del sistema es el siguiente: el usuario carga una imagen o la hace desde la aplicación móvil, esta es enviada al *backend*, donde los modelos de *machine learning*

realizan la predicción. Existen dos opciones para gestionar este proceso: cargar el modelo directamente en la aplicación móvil o utilizar una API para hacer las predicciones en el *backend*.

Cargar el modelo directamente en la aplicación móvil tiene la ventaja de permitir que las predicciones se realicen sin necesidad de una conexión a Internet, lo cual es útil si el usuario se encuentra en áreas con cobertura limitada o si se desea que la aplicación funcione de manera offline. Además, este enfoque puede reducir la latencia, ya que el procesamiento se realiza localmente en el dispositivo, lo que ofrece respuestas más rápidas al usuario. También proporciona mayor control sobre los datos, ya que no es necesario enviarlos a servidores externos, lo cual puede ser una preocupación en aplicaciones relacionadas con datos sensibles, como los de salud. Por último, este enfoque puede reducir los costos asociados con el uso de servidores y servicios en la nube.

Por otro lado, utilizar una API en el *backend* ofrece la ventaja de poder ejecutar modelos más grandes y complejos que requieren mayores recursos de procesamiento. Esto es especialmente útil si el modelo de *machine learning* es pesado o si se desea usar técnicas avanzadas de inferencia que no serían viables en dispositivos móviles con recursos limitados. Este enfoque también facilita la actualización del modelo, ya que las mejoras o entrenamientos adicionales se pueden hacer directamente en el servidor sin necesidad de que los usuarios actualicen su aplicación. Además, al centralizar el procesamiento en el servidor, se facilita la escalabilidad del sistema y se pueden manejar más usuarios sin afectar el rendimiento de la aplicación en dispositivos individuales. En resumen, la opción de utilizar una API permite mantener la aplicación más ligera y ágil, delegando las tareas de procesamiento pesado a servidores con mayores capacidades.

Ambas opciones tienen sus ventajas, y la elección dependerá de los requisitos específicos del proyecto, como la complejidad del modelo, la necesidad de funcionamiento offline, y la capacidad de actualización y escalabilidad del sistema.

En este caso, he optado por desplegar los modelos como servicios web separados accesibles mediante *endpoints* REST, en lugar de integrarlos directamente en la aplicación móvil. Para

ello, he utilizado la plataforma Render, donde he desplegado dos APIs independientes: una dedicada al modelo de clasificación de malignidad, y otra para el modelo de tipo de lesión dermatológica. Cada servicio expone un *endpoint* específico que acepta una imagen y sus metadatos en formato JSON, procesa la información en el servidor y devuelve una predicción en tiempo real.

Desde la aplicación Flutter, realizo peticiones HTTP POST a estos *endpoints* utilizando los datos proporcionados por el usuario (imagen y metadatos). Esta arquitectura permite actualizar o reentrenar los modelos de forma centralizada sin necesidad de modificar ni redistribuir la aplicación móvil. Además, delegar el procesamiento en servidores externos reduce la carga computacional en el dispositivo del usuario, lo que mejora el rendimiento general de la aplicación, especialmente en móviles con recursos limitados.

El resultado es mostrado en la interfaz móvil, mientras que las imágenes y los datos del diagnóstico se guardan en el almacenamiento de Supabase y Firebase para su posterior consulta.

6.1.2 DIAGRAMA DE CASOS DE USO

El diagrama de casos de uso de *SkinCheck* representa las interacciones entre los actores principales (usuario y administrador) y las funcionalidades del sistema. Los actores y los casos de uso más relevantes incluyen:

Usuario:

- **Cargar imagen de lesión:** El paciente carga una imagen desde la aplicación para obtener un diagnóstico.
- **Ver diagnóstico:** El sistema procesa la imagen y devuelve el diagnóstico.
- **Consultar historial de análisis:** El usuario puede acceder al historial de sus análisis pasados, con imágenes y resultados.
- **Gestionar perfil:** Los usuarios pueden editar sus datos personales y preferencias.

- **Acceder a información:** Los usuarios tienen acceso a información sobre lesiones dermatológicas, test de riesgo, calendario de citas y rayos UV.

Administrador:

- **Gestión de usuarios:** El administrador puede ver y gestionar los datos de los usuarios.
- **Supervisión de análisis:** El administrador puede acceder a los diagnósticos y los resultados generados por el modelo de *machine learning*.

El diagrama de casos de uso también refleja la interacción entre las distintas capas del sistema, asegurando que el *frontend*, el *backend* y la base de datos trabajen de manera conjunta.

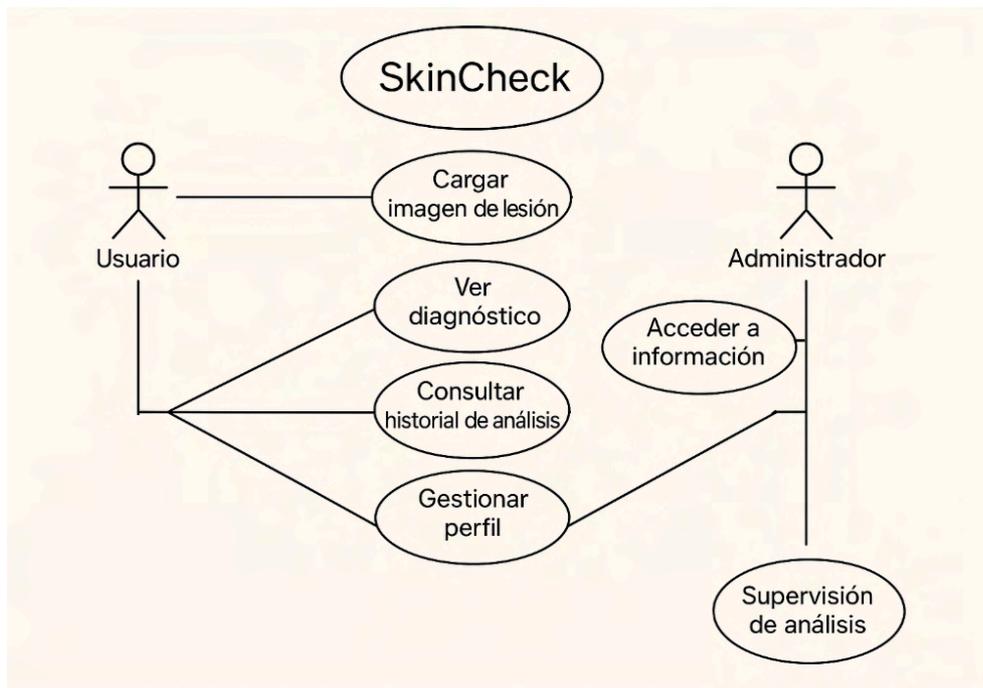


Ilustración 27. Diagrama de Casos de Uso

6.1.3 REQUISITOS FUNCIONALES

Los requisitos funcionales definen las acciones y tareas que el sistema debe ser capaz de realizar para satisfacer las necesidades del usuario. Los requisitos funcionales de *SkinCheck* incluyen:

Autenticación de usuarios: Los usuarios deben poder registrarse y acceder a su cuenta mediante un sistema de autenticación seguro proporcionado por Firebase.

Carga y análisis de imágenes: El sistema debe permitir al usuario cargar imágenes de lesiones dermatológicas y analizar estas imágenes con el modelo para obtener un diagnóstico.

Visualización de resultados: Los usuarios deben poder ver los resultados del diagnóstico.

Historial de análisis: Los usuarios deben poder acceder a un historial de los análisis previos, con las imágenes asociadas y los diagnósticos previos.

Gestión del perfil: Los usuarios pueden gestionar su perfil y editar sus datos personales.

Acceso a información relevante: La aplicación debe proporcionar acceso a información educativa sobre lesiones dermatológicas, un *test* de riesgo, un calendario de citas y datos sobre los rayos UV.

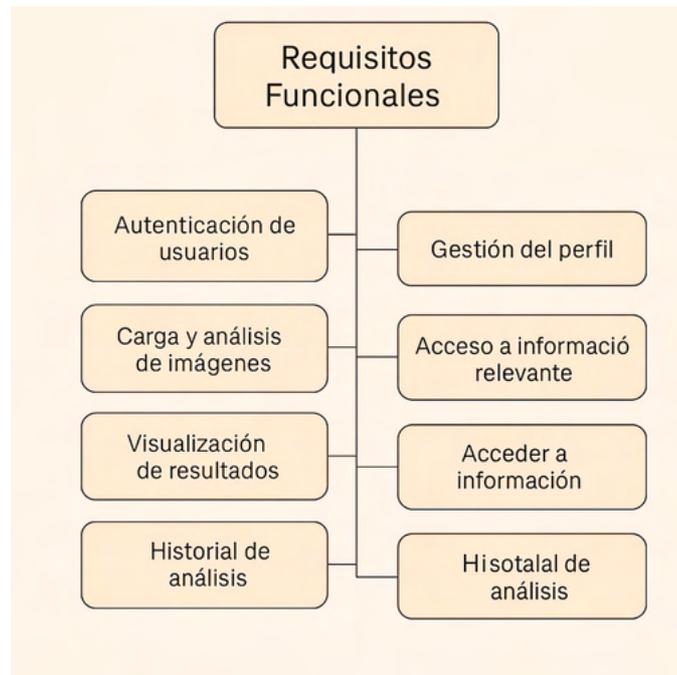


Ilustración 28. Requisitos Funcionales

6.1.4 REQUISITOS NO FUNCIONALES

Los requisitos no funcionales definen las características de calidad del sistema y cómo debe comportarse bajo ciertas condiciones. Los principales requisitos no funcionales de *SkinCheck* son:

Escalabilidad: El sistema debe ser capaz de manejar al menos 5000 usuarios activos simultáneos y datos sin comprometer su rendimiento. Esto es especialmente importante a medida que se agregan más imágenes de usuarios y se realizan más análisis.

Seguridad: El sistema debe garantizar la privacidad de los datos de los pacientes, cumpliendo con normativas como el GDPR. Los datos deben ser almacenados de forma segura mediante autenticación robusta en Firebase y cifrado de archivos en Supabase.

Rendimiento: El tiempo de respuesta desde que se envía una imagen hasta que se recibe el diagnóstico no debe superar los 5 segundos en condiciones de red normales. La aplicación

debe mantener una latencia inferior a 600 ms en la interfaz de usuario para asegurar una experiencia fluida y sin retrasos perceptibles.

Disponibilidad: El sistema debe estar operativo al menos el 99.5% del tiempo mensual, permitiendo el acceso y uso continuo de la aplicación, incluso durante actualizaciones menores. Las actualizaciones de los modelos de *machine learning* y los datos del usuario deben sincronizarse en segundo plano sin afectar la usabilidad.



Ilustración 29. Requisitos No Funcionales

6.1.5 COMPONENTES DEL PROYECTO

Los componentes principales del sistema *SkinCheck* incluyen:

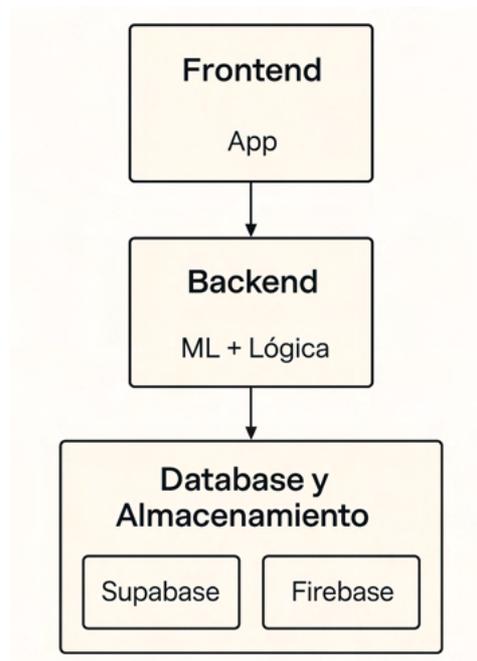


Ilustración 30. Componentes del Proyecto

Aplicación Móvil (Frontend): Desarrollada en Flutter, proporciona una interfaz interactiva que permite a los usuarios cargar imágenes, ver diagnósticos, gestionar su perfil y acceder a información relevante.

Backend: Gestiona la lógica del sistema, incluyendo la comunicación entre el *frontend* y la base de datos. El *backend* se encarga de procesar las imágenes y de ejecutar el modelo.

Firebase: Se utiliza para gestionar la autenticación de usuarios y asegurar que solo los usuarios registrados accedan a la aplicación y sus datos.

Supabase: Proporciona almacenamiento de archivos (imágenes de lesiones) y base de datos relacional para gestionar los metadatos (resultados de diagnósticos, historial de usuarios) de manera escalable y eficiente.

Modelos de *Machine Learning*: Los modelos entrenados se utilizan para clasificar las imágenes según su malignidad y tipo de lesión. El modelo ha sido desplegado como un

servicio web, y su acceso se realiza a través de una API externa. Las imágenes se procesan desde la aplicación mediante llamadas a esta API.

6.2 *MODELO DE MACHINE LEARNING*

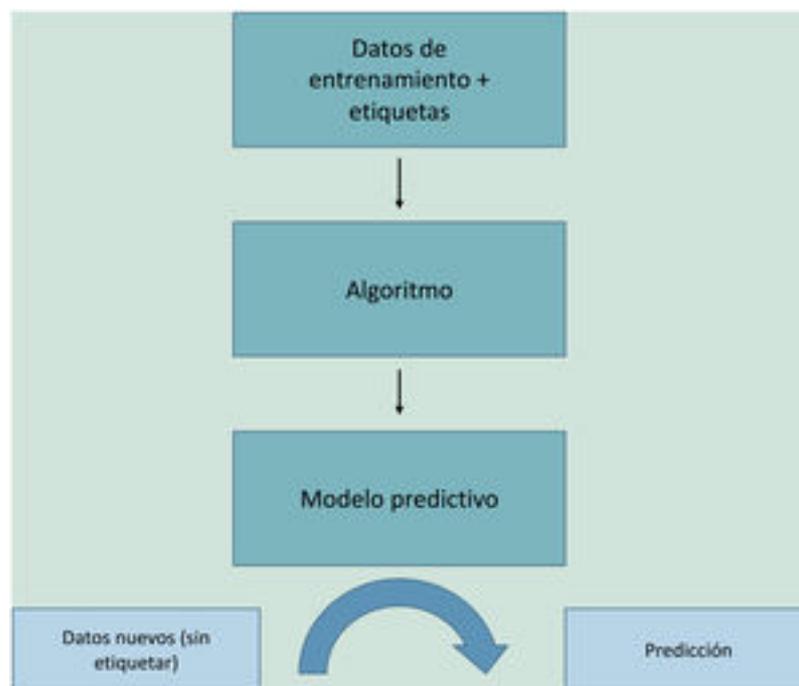


Ilustración 31. Funcionamiento modelo [37]

6.2.1 *RECOPILACIÓN DE DATOS*

Para el desarrollo del modelo de clasificación de lesiones dermatológicas, se empleó el conjunto de datos del archivo ISIC Archive, que contiene miles de imágenes de lesiones etiquetadas con diagnóstico clínico. Además de las imágenes, se descargaron metadatos asociados, incluyendo información sobre la edad, sexo, ubicación anatómica y naturaleza (benigna/maligna) de las lesiones.

Los datos fueron cargados y explorados en profundidad utilizando Pandas y Seaborn. La distribución por sexo, edad y ubicación corporal fue representada gráficamente para tener una visión inicial de la composición del *dataset*.

Challenge 2024: Training

Actions ▾

Name: Challenge 2024: Training
Number of images: 401,059
Number of lesions: 22,058
Number of patients: 1,042
Attribution:

- CC-BY: Memorial Sloan Kettering Cancer Center (129.068)
- CC-BY-NC: Department of Dermatology, Hospital Clínic de Barcelona (105.724)
- CC-BY-NC: Title: ISIC 2024 Basel Source: University Hospital of Basel License: Creative Commons Attribution-NonCommercial (CC BY-NC) (65.218)
- CC-BY: Frazer Institute, The University of Queensland, Dermatology Research Centre (51.768)
- CC-0: Courtesy of ACEMID MIA (28.665)
- CC-BY-NC: ViDIR Group, Department of Dermatology, Medical University of Vienna (12.640)
- CC-BY: Department of Dermatology, University of Athens, Andreas Syggros Hospital of Skin and Venereal Diseases, Alexander Stratigos, Konstantinos Liopyris (7976)

Description:
Official collection

Page 1 of 13,369 [next](#) [last](#) [»](#)
[fewer columns](#) | [more columns](#)



Ilustración 32. Dataset ISIC Archive [38]

6.2.2 LIMPIEZA Y PREPARACIÓN DE DATOS

Tras la recopilación inicial de metadatos clínicos asociados a las imágenes del archivo ISIC, se llevó a cabo un proceso exhaustivo de depuración y organización del conjunto de datos, con el objetivo de preparar una base sólida para el entrenamiento del modelo de clasificación de lesiones dermatológicas.

Distribución general y selección de variables relevantes:

En primer lugar, se seleccionaron las columnas más relevantes: `isic_id`, `age_approx`, `sex`, `anatom_site_general`, `benign_malignant` y `diagnosis`. Se visualizan tres aspectos clave del conjunto de datos tras la carga inicial:

La distribución por sexo muestra una representación equilibrada entre hombres y mujeres, con una ligera mayoría masculina.

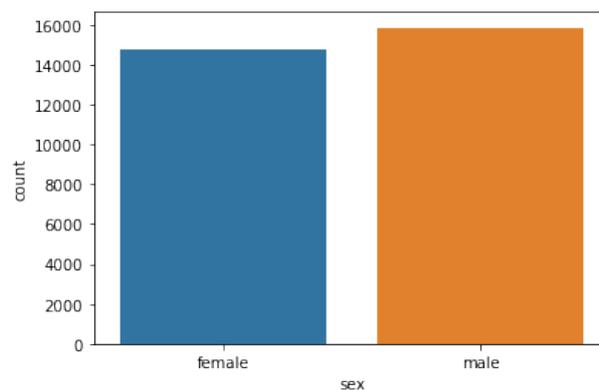


Ilustración 33. Distribución por edad

La densidad de edad (`age_approx`) presenta una distribución unimodal con un pico en torno a los 50 años, lo que es coherente con la prevalencia de lesiones cutáneas a edades medias-avanzadas.

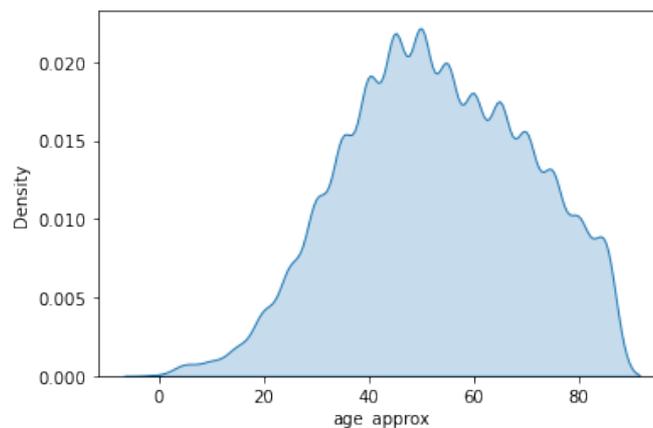


Ilustración 34. Densidad por edad

En cuanto a la localización anatómica, se observa que las extremidades inferiores y el torso anterior son las zonas más frecuentemente analizadas.

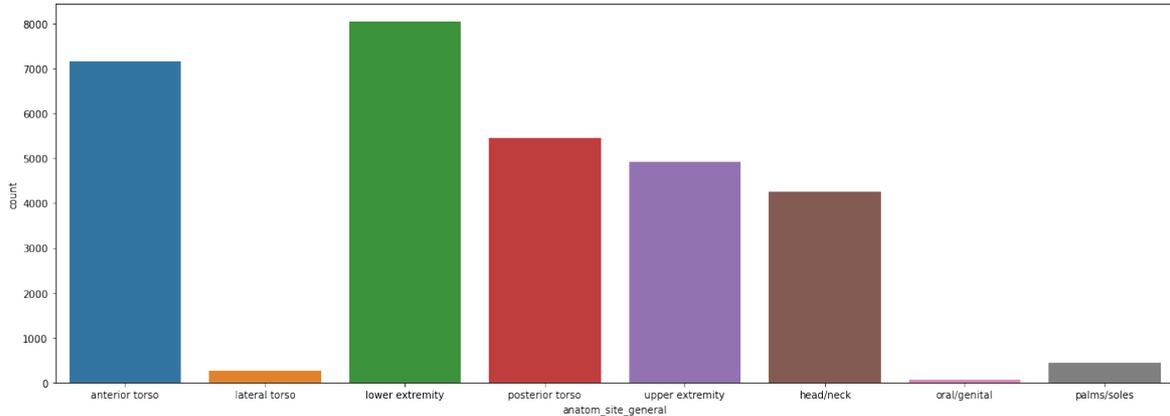


Ilustración 35. Localización Anatómica

Homogeneización de etiquetas y reagrupación de diagnósticos

Se identificaron etiquetas redundantes y clases con bajo soporte estadístico. Se procedió a: Homogeneizar las etiquetas de benign_malignant, eliminando casos “indeterminate” por ser innecesaria y estar casi vacía y quedando dos clases principales.

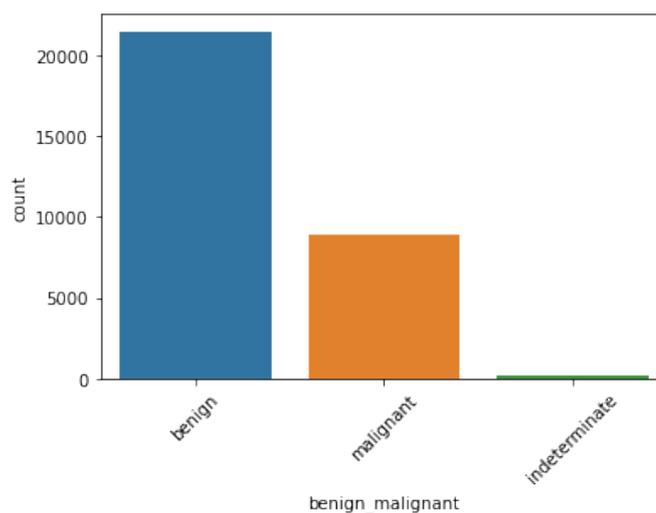


Ilustración 36. Distribución de Malignidad

Agrupar los diagnósticos menos frecuentes bajo la etiqueta OTH (“Other”) y mantener seis clases dominantes que representan la mayoría del dataset. Esta decisión se justifica por la alta desproporción observada, donde el nevus (NV) y el melanoma (MEL) dominan frente a otras clases minoritarias como AK o SL, las cuales eliminaré.

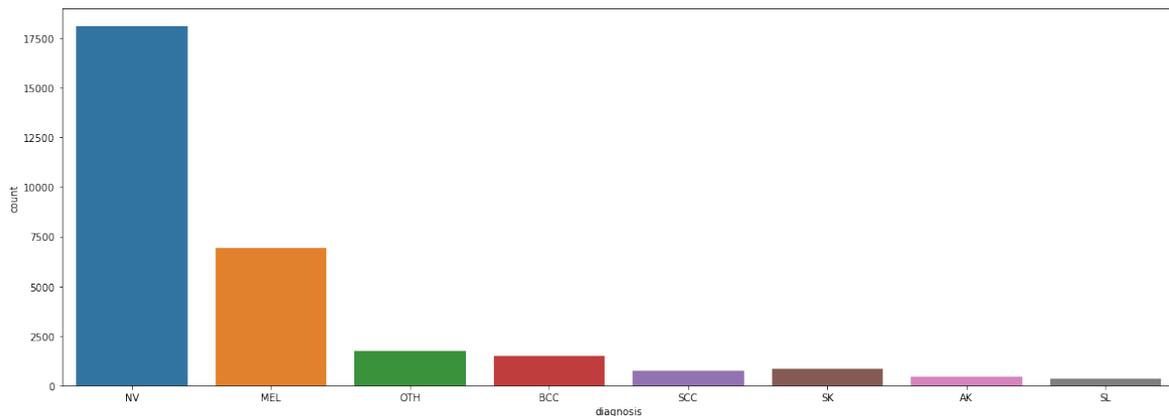


Ilustración 37. Distribución de Lesiones Dermatológicas

Tipos de Lesión

El tipo de lesión más representado en el conjunto de datos es el **nevus melanocítico (NV)**, lo cual es consistente con la prevalencia en la población general. Los nevus suelen ser lesiones benignas y comunes, pero algunos pueden presentar características atípicas o evolucionar hacia melanoma, lo que convierte su correcta clasificación en un punto clave para evitar tanto diagnósticos erróneos como intervenciones innecesarias. En el modelo, los nevus constituyen una clase fundamental para entrenar el sistema a distinguir entre lesiones benignas y potencialmente malignas.



Ilustración 38. Imágenes de Nevus

El segundo grupo en frecuencia corresponde al **melanoma** (MEL), una de las formas más agresivas de cáncer de piel. Aunque menos común que otras lesiones, su detección temprana es crítica, ya que el pronóstico del paciente depende directamente de la rapidez con la que se identifique y trate. La presencia de un número suficiente de ejemplos de melanoma en el conjunto de datos permite al modelo aprender patrones específicos de esta clase, como bordes irregulares, variabilidad de color y asimetrías.



Ilustración 39. Imágenes de Melanoma

Otro tipo relevante es el **carcinoma basocelular** (BCC), el cáncer de piel más frecuente. Aunque su malignidad es generalmente baja y rara vez metastatiza, su identificación sigue siendo importante para evitar daños locales y preservar la estética cutánea del paciente. Su apariencia puede ser variable, lo que justifica su inclusión como clase independiente dentro del modelo.



Ilustración 40. Imágenes de Carcinoma Basocelular

El **carcinoma escamoso** (SCC) también está presente, aunque en menor proporción. Este tipo de cáncer cutáneo tiene una mayor probabilidad de diseminación que el BCC, especialmente si no se trata adecuadamente. Su correcta diferenciación frente a otras lesiones resulta esencial desde el punto de vista clínico, ya que requiere intervenciones más urgentes.



Ilustración 41. Imágenes Carcinoma Escamoso

También se incluyen lesiones como la **queratosis seborreica** (SK), que es una afección cutánea benigna muy común en personas mayores. Aunque inofensiva, su apariencia puede ser confusa y asemejarse a lesiones malignas, lo que plantea un reto diagnóstico tanto para dermatólogos como para sistemas automatizados. Entrenar al modelo a reconocer correctamente estas lesiones evita falsos positivos que podrían generar ansiedad innecesaria en el paciente.

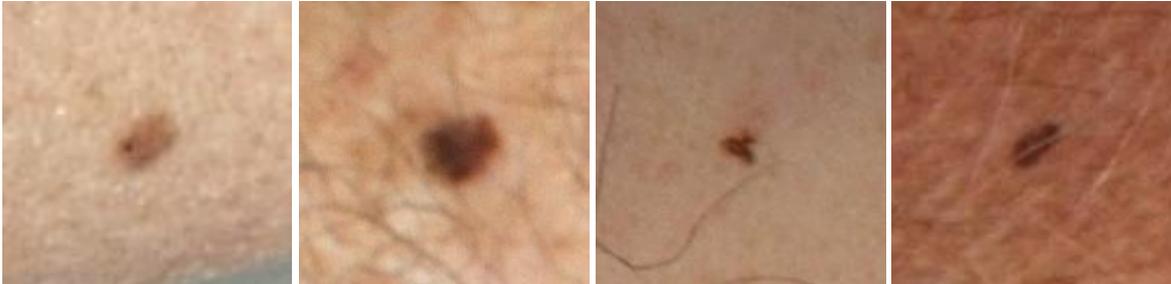


Ilustración 42. Imágenes Queratosis Seborreica

La clase OTH (*Other*) representa un conjunto heterogéneo de lesiones no clasificadas en las categorías principales. Aunque su frecuencia es baja, su inclusión ayuda al modelo a generalizar y a enfrentarse a imágenes que no se ajustan a patrones típicos. Esto refuerza la robustez del sistema ante casos menos comunes o atípicos.

En conjunto, la correcta identificación de estos seis tipos de lesión no solo mejora la precisión diagnóstica del modelo, sino que también aporta valor clínico al permitir una evaluación personalizada del riesgo y una priorización adecuada en el seguimiento dermatológico de cada paciente.

MALIGNANT:

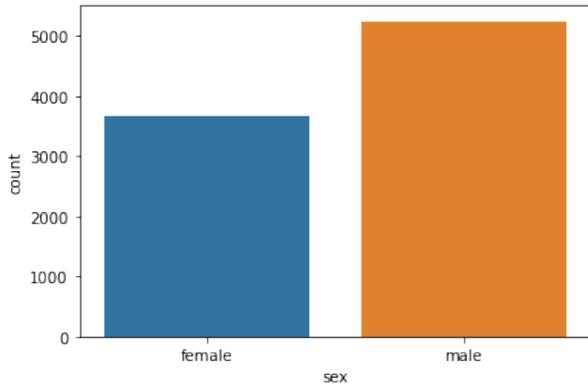


Ilustración 43. Distribución de Género en Maligno

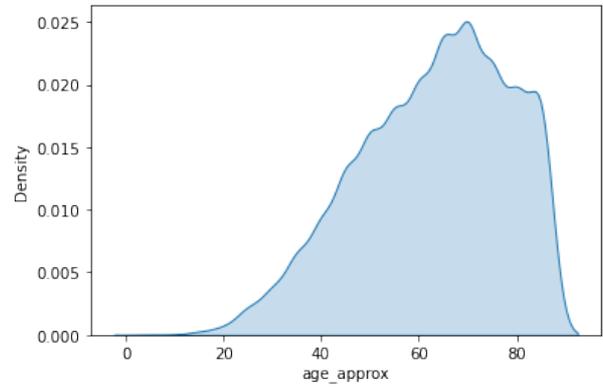


Ilustración 44. Distribución de Edad en Maligno

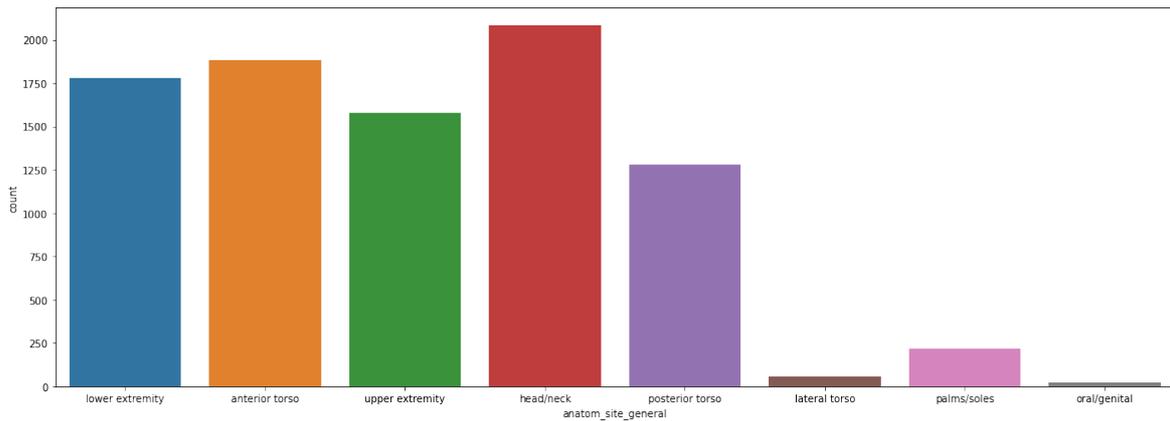


Ilustración 45. Distribución Anatómica en Maligno

BENIGNANT:

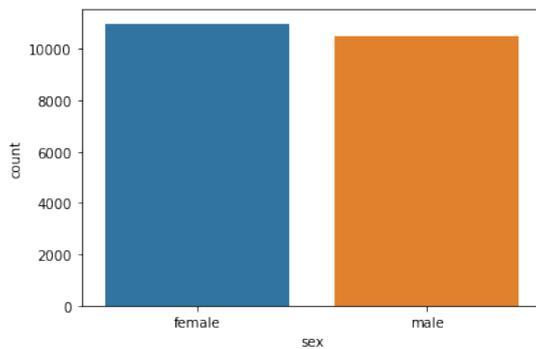


Ilustración 46. Distribución de Género en Benigno

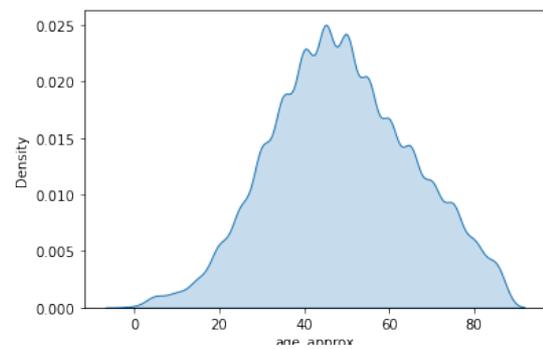


Ilustración 47. Distribución de Edad en Benigno

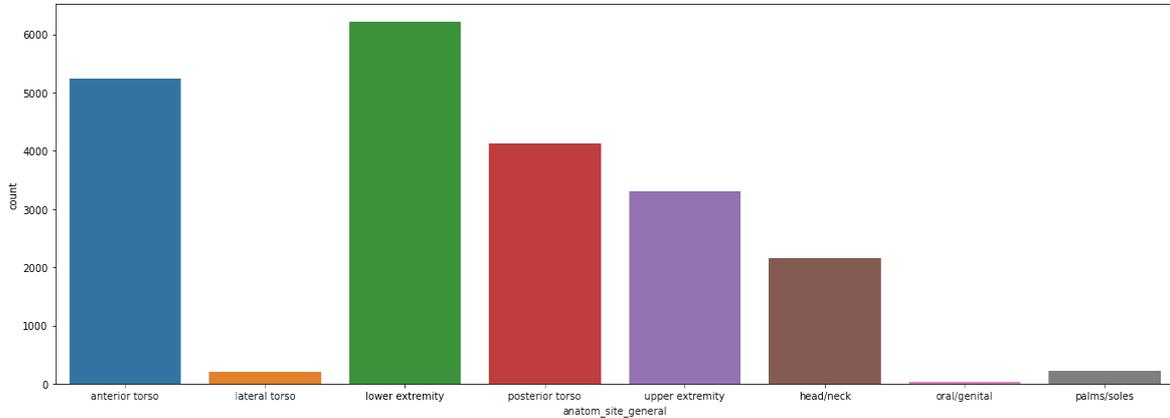


Ilustración 48. Distribución Anatómica en Benigno

Análisis de distribución y equilibrio por clase

Una vez clasificados los datos en benignos y malignos, se compararon sus distribuciones. Se presenta las distribuciones de edad y sexo separadas por tipo de diagnóstico.

Se observa que las lesiones malignas tienden a aparecer con mayor frecuencia en pacientes mayores, lo que valida la inclusión de age_approx como variable relevante.

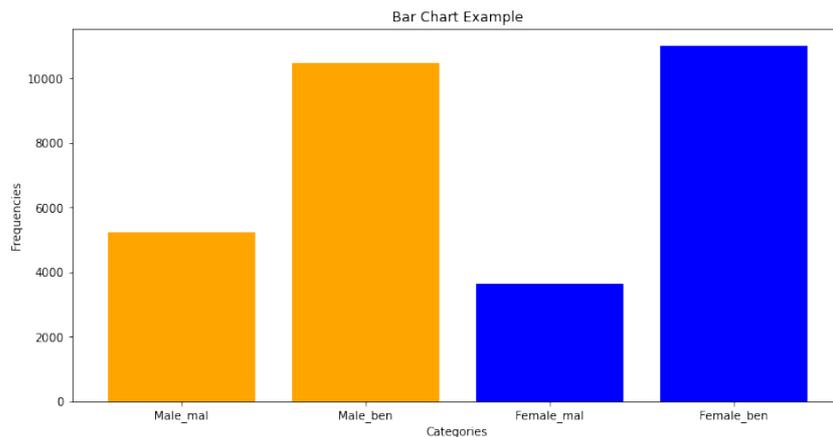


Ilustración 49. Comparativa Género Maligno Benigno

También existen diferencias notables en la localización corporal según la naturaleza de la lesión, siendo más comunes en la cabeza/cuello y extremidades superiores para casos malignos.

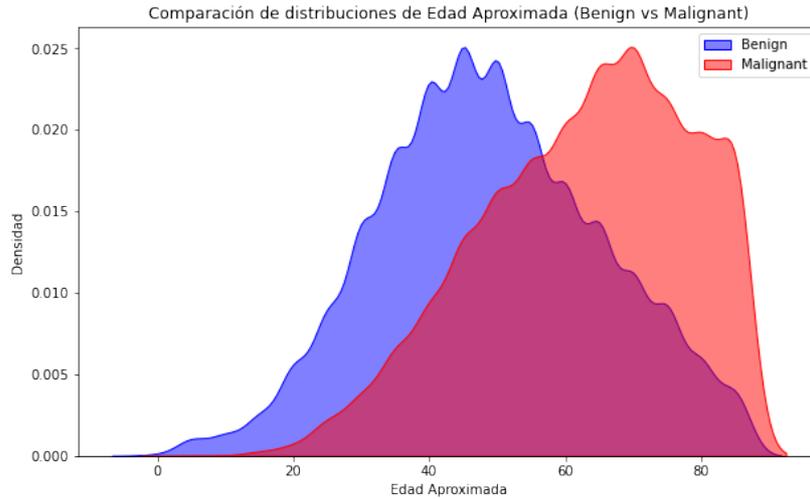


Ilustración 50. Comparativa de Edad Maligno Benigno

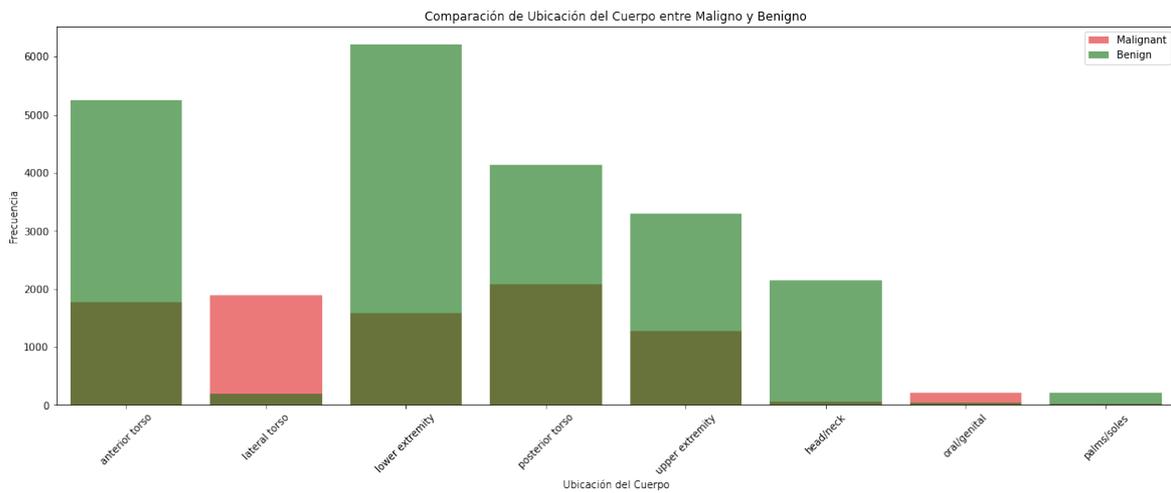


Ilustración 51. Comparativa Anatómica Maligno Benigno

Conclusión de la sección

Este proceso de limpieza no sólo permitió reducir el ruido en el conjunto de datos, sino que sentó las bases para la codificación y transformación de variables en el pipeline de entrenamiento. Se garantizó un equilibrio razonable entre clases, se eliminaron valores atípicos, y se mantuvo la diversidad diagnóstica sin comprometer la generalización del modelo. Estas decisiones han sido fundamentales para el rendimiento final del clasificador.

	isic_id	age_approx	sex	anatom_site_general	benign_malignant	diagnosis
0	ISIC_5257439	40	female	lateral torso	benign	NV
1	ISIC_0469776	50	female	lower extremity	benign	NV
2	ISIC_1648650	40	female	lateral torso	benign	NV
3	ISIC_7529219	30	female	lower extremity	benign	NV
4	ISIC_8568104	30	female	lower extremity	benign	NV
...
25418	ISIC_0000005	40	female	anterior torso	benign	NV
25419	ISIC_0000004	80	male	posterior torso	malignant	MEL
25420	ISIC_0000002	60	female	upper extremity	malignant	MEL
25421	ISIC_0000001	30	female	anterior torso	benign	NV
25422	ISIC_0000000	55	female	anterior torso	benign	NV

25423 rows × 6 columns

Ilustración 52. Dataframe Completo

6.2.3 GESTIÓN DE ATÍPICOS

Durante la etapa de limpieza, también se analizaron posibles valores atípicos en las variables numéricas, como `age_approx`. Se utilizó un gráfico de distribución de densidad para observar el comportamiento de la edad en función del tipo de diagnóstico. Se descartaron registros con valores imposibles o extremadamente inusuales.

Se identificaron diferencias significativas en la distribución por sexo y ubicación anatómica entre lesiones benignas y malignas, lo cual puede aportar información útil para el modelo.

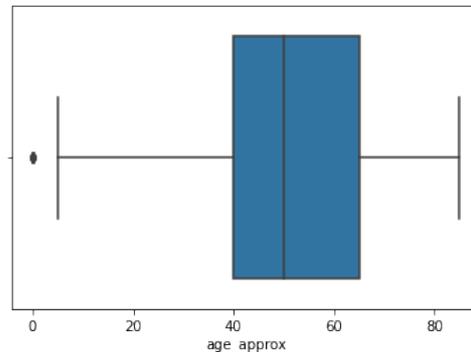


Ilustración 53. Diagrama de Caja Edad Global

6.2.4 SELECCIÓN DE CARACTERÍSTICAS

Se construyó un *dataframe* final con las variables `isic_id`, `age_approx`, `sex`, `anatom_site_general`, `benign_malignant`, `diagnosis` y las variables binarizadas para cada diagnóstico categórico. Esta transformación permite alimentar correctamente el modelo de *machine learning*, ya que las variables categóricas deben ser codificadas en formato numérico.

Estas características combinan información visual (las imágenes) con metadatos clínicos, permitiendo entrenar un modelo robusto capaz de identificar no solo la benignidad o malignidad, sino también el tipo específico de lesión. Esta información ha sido crucial para definir la arquitectura final del modelo CNN que se integra en la aplicación *SkinCheck*.

6.2.5 ENTRENAMIENTO Y TESTEO DEL MODELO

Con el objetivo de mejorar la precisión y la especialización del sistema de clasificación, he optado por desarrollar dos modelos distintos: uno enfocado en predecir la malignidad de una lesión dermatológica (benigna o maligna), y otro dedicado a identificar el tipo específico de lesión (BCC, MEL, NV, entre otros).

Esta decisión se fundamenta en varios motivos. En primer lugar, ambas tareas presentan un grado diferente de complejidad: mientras que la predicción de malignidad es un problema binario, la clasificación del tipo de lesión implica un problema multiclase más desafiante.

Separar los modelos permite optimizar cada arquitectura y función de pérdida para su respectivo objetivo. Además, esta división favorece la interpretabilidad del sistema y permite realizar un diagnóstico más modular, aplicando distintos umbrales de confianza o estrategias de validación cruzada según la naturaleza del análisis clínico.

Ambos modelos comparten una arquitectura común basada en un enfoque multimodal, que integra imágenes clínicas de lesiones (resolución 128x128 RGB) junto con metadatos del paciente, como edad, sexo y ubicación anatómica de la lesión. Los metadatos fueron codificados mediante *OneHotEncoding* (para variables categóricas) y normalización estándar (para edad). Posteriormente, se combinaron con las características extraídas por una red convolucional basada en ResNet50 preentrenada, para generar una representación conjunta que alimenta la capa de decisión final.

El *dataset* se dividió siguiendo una estrategia estratificada en tres conjuntos: entrenamiento (68%), validación (17%) y test (15%), preservando la proporción de clases en cada uno de ellos. Esto se realizó con el fin de evaluar de forma robusta el rendimiento del modelo sin sobreajuste.

Durante el entrenamiento, se utilizaron técnicas como *EarlyStopping* y *ReduceLRonPlateau* para evitar el sobreajuste y mejorar la estabilidad del aprendizaje. Se emplearon optimizadores como RMSprop o AdamW, y la función de pérdida utilizada fue la *categorical crossentropy*, tanto para el problema binario como para el multiclase (debido a su formulación con codificación one-hot).

6.2.6 GESTIÓN DE LA DESPROPORCIÓN ENTRE CLASES

Durante las primeras etapas del entrenamiento del modelo de clasificación de malignidad, observé un claro desbalance de clases en el conjunto de imágenes descargado desde el *dataset* original. Esta desproporción reflejaba fielmente la distribución real que se da en la población: un mayor número de lesiones benignas frente a un menor número de lesiones malignas, y una cantidad muy reducida de casos indeterminados.

En cifras concretas, el conjunto de datos contenía 17.835 imágenes benignas, 7.412 malignas y tan solo 176 correspondientes a la clase “indeterminada”. Dado que esta última clase representaba una fracción mínima del total y no aportaba valor clínico al diagnóstico que mi aplicación busca proporcionar, el cual se basa en una decisión clara entre benignidad o malignidad, decidí eliminarla por completo del proceso de entrenamiento.

A continuación, para abordar el desbalance restante entre las clases "benigna" y "maligna", opté por recortar el número de imágenes de ambas clases hasta obtener un conjunto de datos equilibrado, garantizando así que el modelo no desarrollara un sesgo hacia la clase mayoritaria.

Una vez balanceado el número de ejemplos, procedí a aumentar el tamaño del *dataset* aplicando técnicas de *data augmentation*, lo cual no solo compensó la pérdida de imágenes por el recorte, sino que también permitió mejorar la capacidad de generalización del modelo.

```
benign          17835
malignant       7412
indeterminate    176
Name: benign_malignant, dtype: int64
```

Técnicas Utilizadas

- **Undersampling:**

```
balanced_df_bm = (
    train_data.groupby("benign_malignant")
    .apply(lambda x: x.sample(n=min(len(x), 3000), random_state=42))
    .reset_index(drop=True)
)
```

```
malignant      3000
benign         3000
Name: benign_malignant, dtype: int64
```

Como se puede observar en el bloque anterior, he aplicado una técnica de *undersampling* utilizando *groupby()* junto con la función *sample()* de Pandas, limitando el número de

imágenes por clase a 3.000 para lograr un conjunto de entrenamiento inicial balanceado. Esta estrategia, aunque implica una reducción del volumen de datos originales, ayuda a evitar que el modelo aprenda un sesgo hacia la clase más abundante (en este caso, la clase benigna). El resultado de este paso fue un conjunto de datos equilibrado con 3.000 imágenes benignas y 3.000 malignas, como puede verse en la salida impresa. Este nuevo conjunto equilibrado sirvió como base para aplicar las técnicas posteriores de aumento de datos.

- **Data Augmentation:**

```
datagen = ImageDataGenerator(  
    rotation_range=15,  
    width_shift_range=0.1,  
    height_shift_range=0.1,  
    zoom_range=0.15,  
    horizontal_flip=True,  
    fill_mode="nearest"  
)  
  
aug_imgs, aug_labels, aug_meta = [], [], []  
  
def augment_class_to_10000(class_idx, current_count):  
    n_needed = 5000 - current_count  
    indices = np.where(np.argmax(malignant_labels_balanced, axis=1) ==  
class_idx)[0]  
  
    imgs = balanced_images_bm[indices]  
    labels = malignant_labels_balanced[indices]  
    meta = metadata_balanced[indices]  
  
    imgs_uint8 = (imgs * 255).astype('uint8')  
  
    generated = []  
    for batch in datagen.flow(imgs_uint8, batch_size=1, shuffle=True):  
        generated.append(batch[0].astype('float32') / 255.0)  
        if len(generated) >= n_needed:  
            break  
  
    # Selecciona metadatos aleatorios consistentes  
    selected = np.random.choice(len(meta), n_needed, replace=True)
```

```
aug_labels.extend(labels[selected])
aug_meta.extend(meta[selected])
aug_imgs.extend(generated)

# Aplicar aumento
for idx, label in enumerate(label_encoder.classes_):
    count = np.sum(np.argmax(malignant_labels_balanced, axis=1) == idx)
    print(f"{label}: {count} imágenes")
    if count < 5000:
        augment_class_to_10000(idx, count)

# Concatenar datasets
balanced_images_bm = np.concatenate([balanced_images_bm] +
[ np.array(aug_imgs) ])
malignant_labels_balanced = np.concatenate([malignant_labels_balanced] +
[ np.array(aug_labels) ])
metadata_balanced = np.concatenate([metadata_balanced] +
[ np.array(aug_meta) ])
```

benign: 3000 imágenes
malignant: 3000 imágenes

✅ **Dataset balanceado FINAL:**
benign: 5000 imágenes
malignant: 5000 imágenes
🖼️ **Imágenes: (10000, 128, 128, 3)**
🏷️ **Etiquetas: (10000, 2)**
📄 **Metadata: (10000, 12)**

Para compensar la reducción inicial de datos y aumentar la diversidad del conjunto de entrenamiento, implementé una estrategia de *data augmentation* utilizando la clase *ImageDataGenerator* de Keras. Esta herramienta permite generar nuevas imágenes a partir de transformaciones aleatorias, tales como rotaciones, desplazamientos, *zoom* y volteos horizontales, manteniendo al mismo tiempo la coherencia semántica de las lesiones. En el bloque de código se puede ver cómo se generaron muestras sintéticas hasta alcanzar un total de 5.000 imágenes por clase. Esta técnica resultó especialmente útil para mejorar la

capacidad de generalización del modelo y reducir el riesgo de *overfitting*, ya que expone al modelo a una mayor variabilidad sin necesidad de recolectar más datos reales.

Como resultado final del proceso de balanceo y aumento, el conjunto de entrenamiento alcanzó un total de 10.000 imágenes distribuidas equitativamente entre ambas clases, con 5.000 benignas y 5.000 malignas. Además, las etiquetas y los metadatos asociados fueron replicados y asignados de manera coherente para garantizar la calidad del conjunto de datos. Esta estrategia no solo solucionó el problema de desbalance inicial, sino que proporcionó un conjunto sólido, diverso y clínicamente relevante para entrenar un modelo robusto y fiable. La forma final del conjunto de datos, con imágenes de tamaño (128, 128, 3), etiquetas codificadas en *one-hot* y metadatos estructurados, está detallada en la salida impresa como confirmación del éxito del proceso.

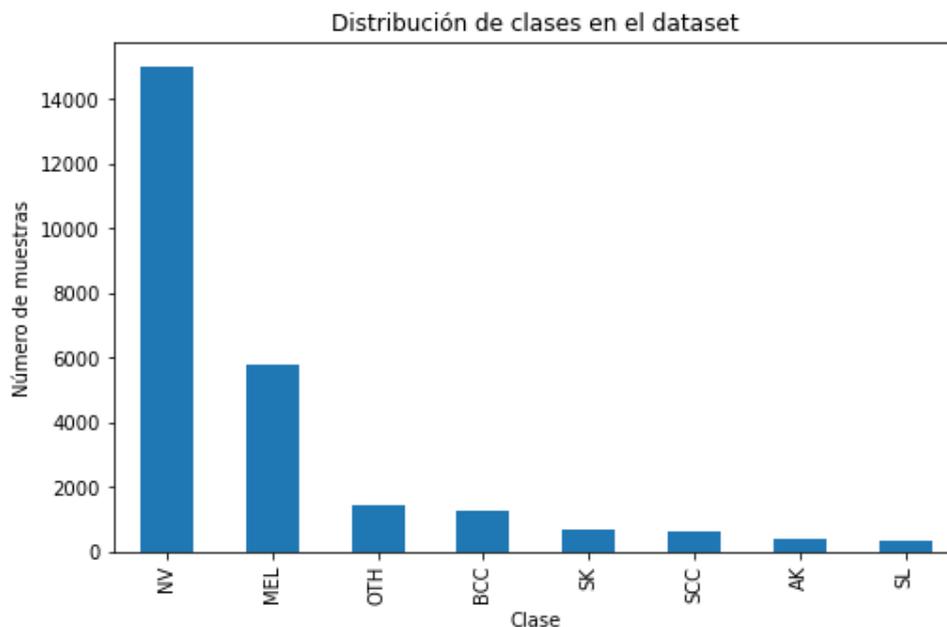
Por otro lado, apliqué una estrategia similar para abordar el desequilibrio en el modelo encargado de clasificar el tipo de lesión dermatológica. Este caso presentaba una complejidad adicional debido tanto al alto número de clases como a su extrema desproporción. Tal como se aprecia en la figura anterior, casi el 60 % de las muestras pertenecen a la clase “NV” (nevus), aproximadamente el 23 % a la clase “MEL” (melanoma), y el resto se reparte en pequeñas fracciones entre “OTH” (otras), “BCC” (carcinoma basocelular), “SK” (queratosis seborreica), “SCC” (carcinoma escamoso), “AK” (queratosis actínica) y “SL” (lesión vascular benigna).

Desde el inicio tuve claro que quería que el modelo pudiera distinguir el mayor número de clases posible, para aumentar su aplicabilidad clínica y su utilidad dentro de la aplicación. Sin embargo, las clases “AK” y “SL” contenían un número de muestras tan reducido (menos del 1.5 % cada una) que su inclusión habría supuesto un riesgo importante de sobreajuste y errores en la predicción. Por esta razón, decidí eliminar únicamente estas dos clases minoritarias y conservar las restantes, buscando así un equilibrio entre representatividad y rendimiento.

Para construir una base de datos balanceada a partir de las clases restantes, recurrí a las mismas técnicas que empleé en el modelo de malignidad, pero aplicadas de forma más

agresiva debido a la mayor cantidad de clases y la mayor disparidad entre ellas. Específicamente, primero apliqué *undersampling* a las clases mayoritarias (“NV” y “MEL”) y luego utilicé técnicas de data *augmentation* intensivo sobre las clases minoritarias, como “BCC”, “SK”, “OTH” y “SCC”.

Estas técnicas de aumento incluyeron rotaciones aleatorias, desplazamientos horizontales y verticales, *zoom*, volteos horizontales y ajustes de brillo y contraste, permitiendo así generar nuevas muestras sintéticas a partir de los datos existentes. Gracias a este enfoque, logré construir un conjunto de datos equilibrado que permitiera entrenar un modelo capaz de distinguir entre seis tipos de lesiones con mayor precisión y robustez.



```
NV      59.072493
MEL     22.629115
OTH      5.597294
BCC      4.889273
SK       2.698344
SCC      2.450537
AK       1.510443
SL       1.152500
Name: diagnosis, dtype: float64
```

- **Undersampling:**

```
balanced_df_diag = (  
    train_data.groupby("diagnosis")  
    .apply(lambda x: x.sample(n=min(len(x), 800), random_state=42))  
    .reset_index(drop=True)  
)
```

```
BCC      800  
NV       800  
OTH      800  
MEL      800  
SK       686  
SCC      623  
Name: diagnosis, dtype: int64
```

En primer lugar, para reducir el desbalance más evidente del conjunto de datos, apliqué una técnica de *undersampling* a las clases mayoritarias. En este caso, recorté las clases “NV” y “MEL” a un máximo de 800 imágenes cada una, usando una función de muestreo aleatorio (*sample*) con una semilla fija para garantizar la reproducibilidad. Esta reducción permitió acercar estas clases a un nivel similar al de las más minoritarias, estableciendo una base inicial más equilibrada. Como resultado de este proceso, obtuve un *dataset* con las siguientes distribuciones: 800 imágenes por clase para BCC, NV, OTH y MEL, y valores ligeramente inferiores para SK y SCC (686 y 623 respectivamente), lo que me permitió aplicar posteriormente un aumento más focalizado en las clases que aún estaban por debajo del umbral deseado.

- **Data Augmentation:**

```
datagen = ImageDataGenerator(  
    rotation_range=10,  
    width_shift_range=0.05,  
    height_shift_range=0.05,  
    zoom_range=0.05,  
    horizontal_flip=True,  
    brightness_range=[0.9, 1.1],
```

```
        fill_mode='reflect')
    )

    aug_imgs, aug_labels, aug_meta = [], [], []

    def augment_class_to_3000(class_idx, current_count):
        n_needed = 3000 - current_count

        indices = np.where(np.argmax(diagnosis_labels_balanced, axis=1) ==
                           class_idx)[0]

        imgs = balanced_images_diag[indices]
        labels = diagnosis_labels_balanced[indices]
        meta = metadata_balanced[indices]

        imgs_uint8 = (imgs * 255).astype('uint8')

        generated = []

        for batch in datagen.flow(imgs_uint8, batch_size=1, shuffle=True):
            generated.append(batch[0].astype('float32') / 255.0)

            if len(generated) >= n_needed:
                break

        aug_imgs.extend(generated)
        aug_labels.extend([labels[0]] * n_needed)
        aug_meta.extend([meta[0]] * n_needed)

    # Aplicar a todas las clases con < 2000
    for idx, label in enumerate(label_encoder.classes_):
        count = np.sum(np.argmax(diagnosis_labels_balanced, axis=1) == idx)
```

```
print(f"{label}: {count} imágenes")

if count < 3000:

    augment_class_to_2000(idx, count)

# Concatenar al dataset existente

balanced_images_diag = np.concatenate([balanced_images_diag] +
[ np.array(aug_imgs) ])

diagnosis_labels_balanced = np.concatenate([diagnosis_labels_balanced] +
[ np.array(aug_labels) ])

metadata_balanced = np.concatenate([metadata_balanced] + [ np.array(aug_meta) ])
```

✅ Dataset balanceado FINAL:

BCC: 3000 imágenes

MEL: 3000 imágenes

NV: 3000 imágenes

OTH: 3000 imágenes

SCC: 3000 imágenes

SK: 3000 imágenes

🖼️ Imágenes: (18000, 128, 128, 3)

🏷️ Etiquetas: (18000, 6)

📄 Metadata: (18000, 12)

Tras el recorte inicial, utilicé nuevamente la clase *ImageDataGenerator* para realizar un aumento de datos, esta vez con una configuración más agresiva que en el modelo de malignidad. Además de rotaciones y desplazamientos, incorporé variaciones de brillo para simular diferentes condiciones de iluminación, mejorando así la robustez del modelo frente a variaciones reales. La función `augment_class_to_3000` generó imágenes hasta alcanzar las 3.000 muestras por clase para aquellas categorías con menos representación. Este procedimiento permitió sintetizar nuevas imágenes válidas, respetando las características esenciales de cada tipo de lesión, a partir de datos ya existentes y asegurando la coherencia con los metadatos y etiquetas originales.

Como resultado del proceso de balanceo, el conjunto de datos final para la clasificación de lesiones quedó compuesto por 18.000 imágenes, con 3.000 por cada una de las seis clases

seleccionadas: BCC, MEL, NV, OTH, SCC y SK. También se actualizaron correctamente las matrices de etiquetas codificadas y los metadatos asociados, resultando en un *dataset* completo y estructurado con dimensiones consistentes: (18000, 128, 128, 3) para las imágenes, (18000, 6) para las etiquetas y (18000, 12) para los metadatos. Este conjunto balanceado y aumentado garantiza una mejor capacidad de aprendizaje del modelo y una mayor precisión en la predicción de cada clase, incluso las menos representadas inicialmente.

6.2.7 ELECCIÓN DEL MODELO FINAL

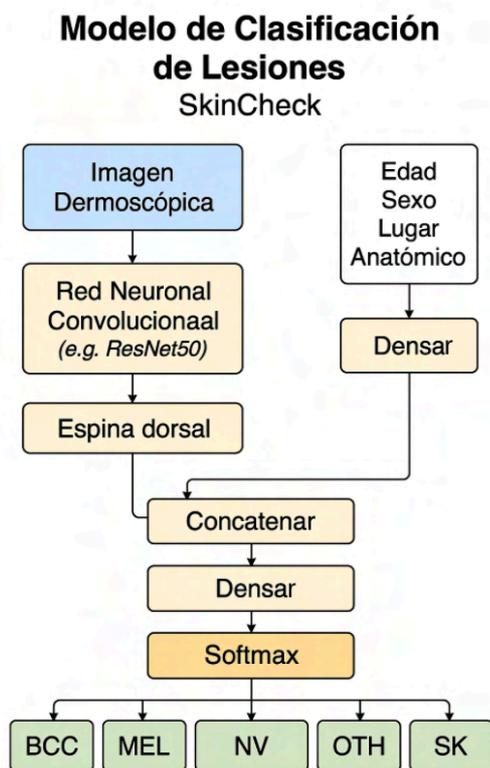


Ilustración 54. Arquitectura Modelo Lesión.

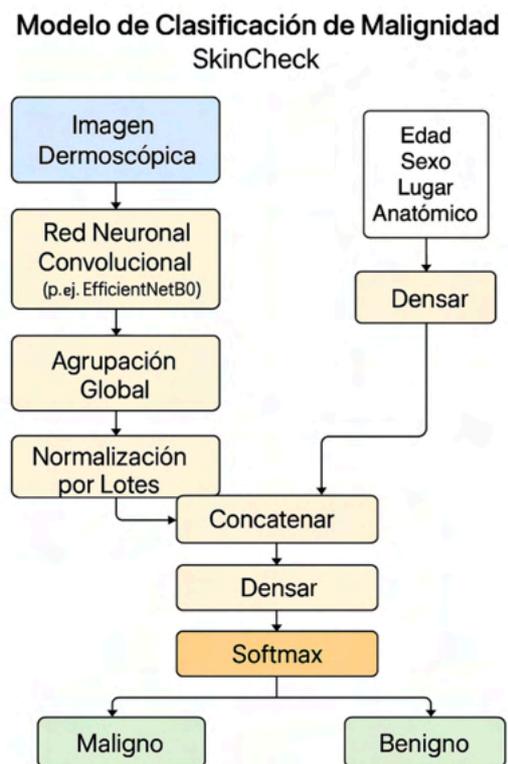


Ilustración 55. Arquitectura Modelo Malignidad

6.3 APLICACIÓN

6.3.1 DESARROLLO DEL BACKEND

6.3.1.1 Autenticación

El sistema de autenticación se ha implementado usando Firebase Authentication, que proporciona una solución robusta y segura para la gestión de usuarios. Los usuarios pueden registrarse e iniciar sesión mediante correo electrónico y contraseña, o bien usando su cuenta de Google gracias a la integración con el proveedor de identidad OAuth 2.0 de Google. Esta opción simplifica el acceso y mejora la experiencia de usuario al reducir la fricción en el login.

Se utilizó el paquete `firebase_auth` de Flutter, junto con el paquete `google_sign_in`, configurando el proyecto en Firebase Console para Android y el identificador de paquete correcto. Se definieron métodos para registrar, autenticar, recuperar contraseñas y cerrar sesión, gestionando posibles errores con mensajes amigables en la interfaz.

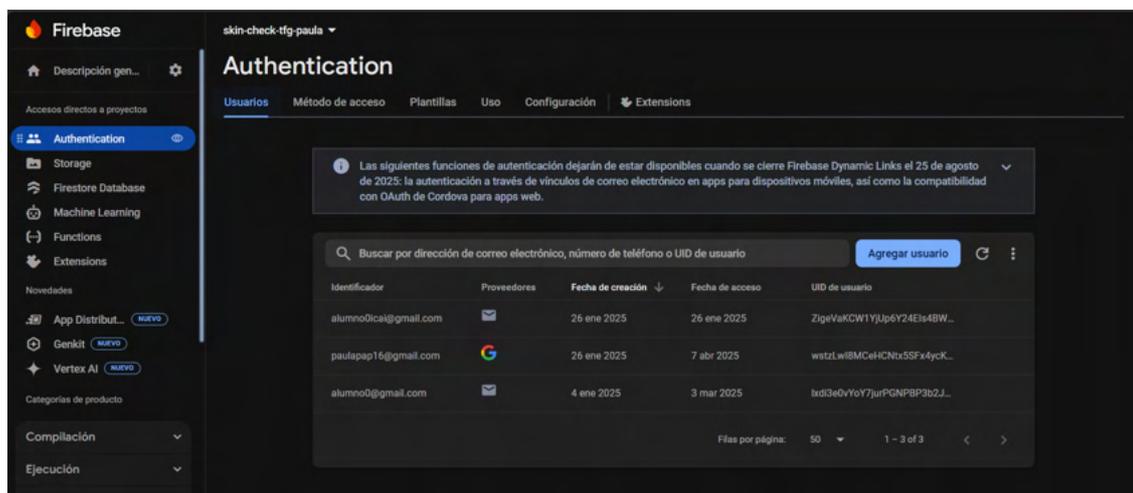


Ilustración 56. Firebase Authentication

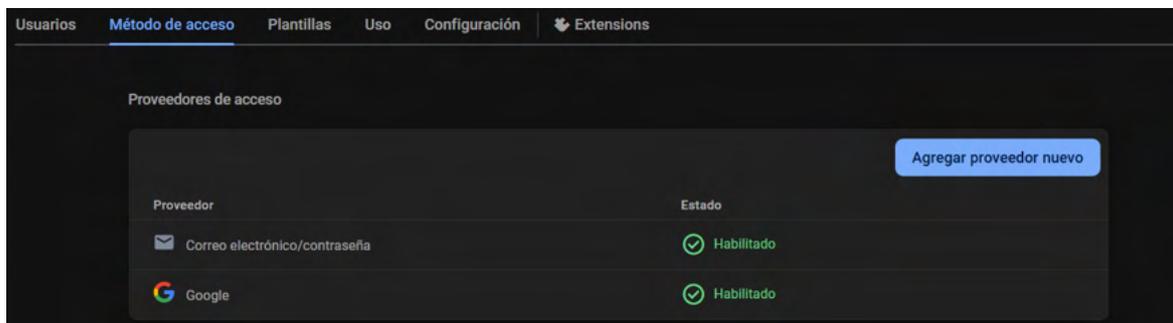


Ilustración 57. Authentication Métodos de Acceso

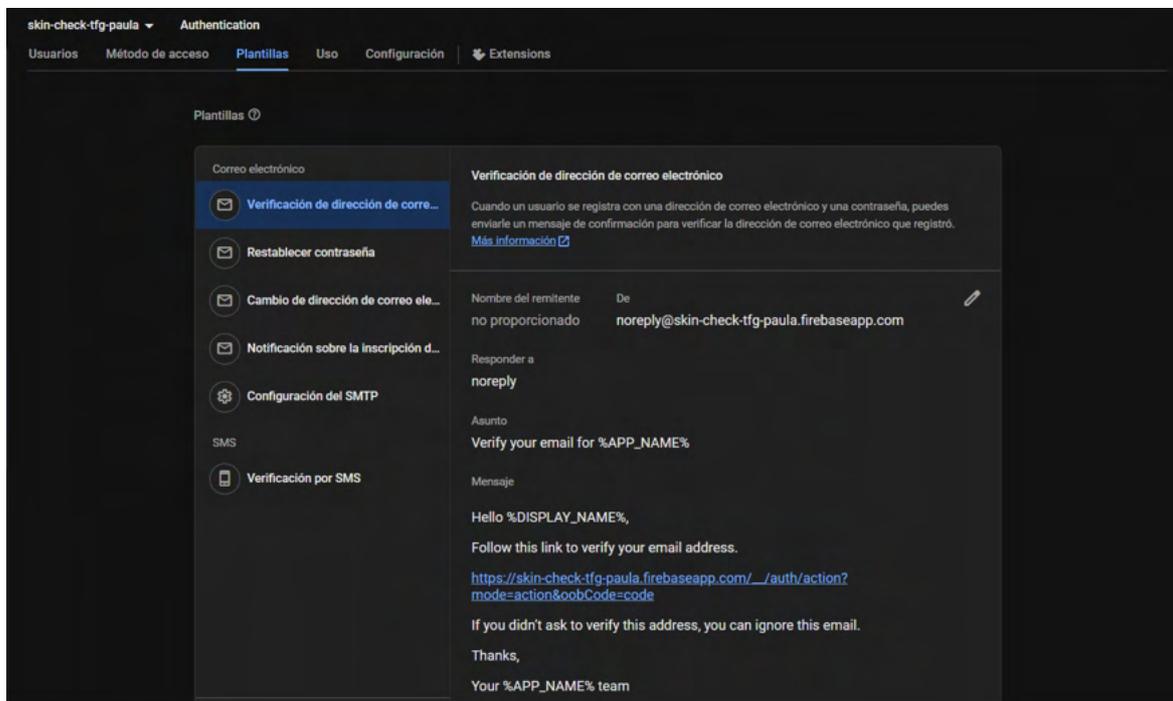


Ilustración 58. Verificación de dirección del correo electrónico

6.3.1.2 Implementación de Firebase

Firestore Database. Cada usuario tiene su propio documento que contiene información como nombre, edad, sexo, historial de diagnósticos (fecha, resultado, tipo de lesión, nivel de confianza del modelo, etc.). Las operaciones de lectura y escritura se realizan con transacciones asíncronas que garantizan integridad y consistencia.

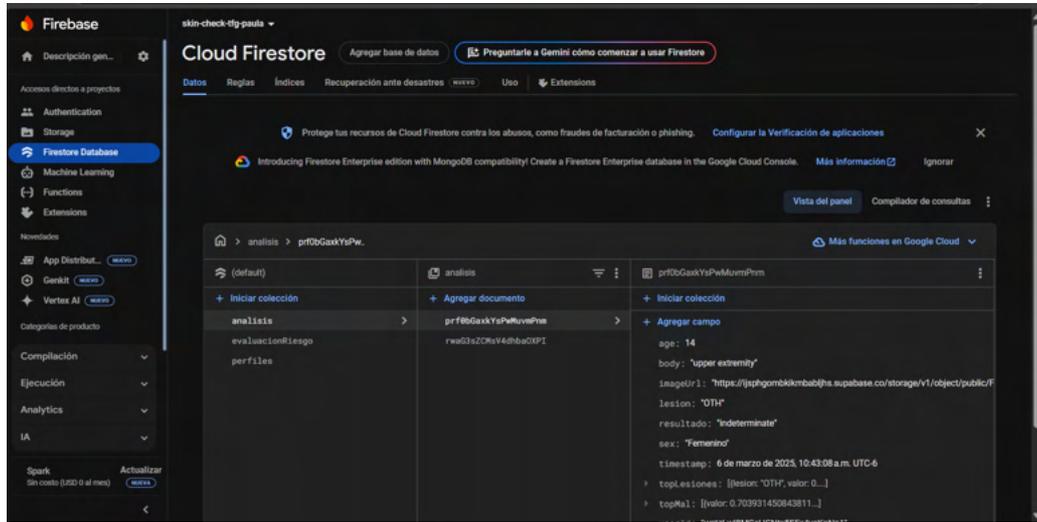


Ilustración 59. Bases de Datos creadas

El *backend* también hace uso de Firebase Storage para subir pequeños archivos auxiliares si es necesario, aunque el grueso de imágenes se maneja con Supabase.

6.3.1.3 Implementación de Supabase

Para el almacenamiento eficiente de imágenes de alta resolución, se optó por Supabase, una alternativa *open-source* a Firebase con una arquitectura basada en PostgreSQL. Las imágenes se suben directamente desde la aplicación móvil al *bucket* correspondiente, usando una API REST generada automáticamente por Supabase.

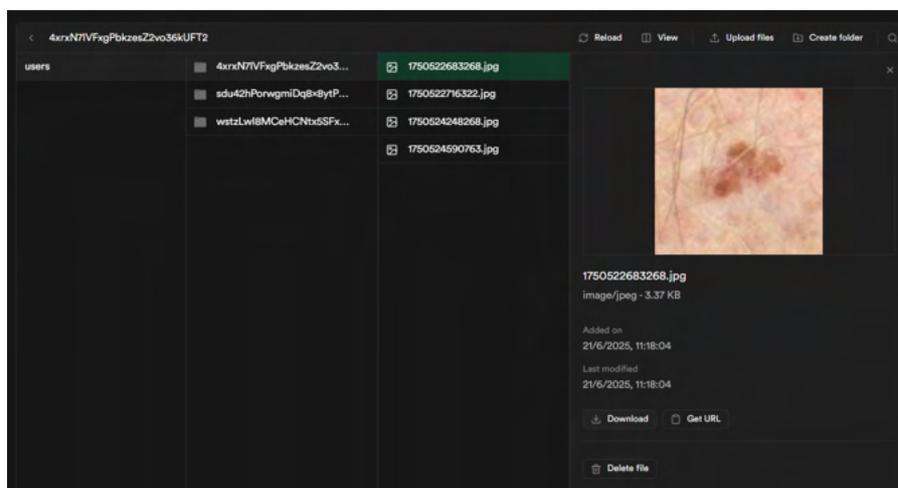


Ilustración 60. Imágenes guardadas de cada usuario en Supabase

Se implementó un sistema de subida autenticada mediante JWTs generados por Firebase y aceptados por Supabase, gracias a una configuración de reglas que valida los *tokens* externos. Esto permite que solo usuarios autenticados puedan subir y consultar imágenes.

Por otra parte, el modelo de *machine learning* entrenado fue convertido a formato `tflite` y almacenado en Supabase Storage, desde donde la aplicación puede descargarlo al primer uso o accederlo en local si ya está cacheado.

6.3.2 DESARROLLO DEL FRONTEND

6.3.2.1 Diseño

La accesibilidad visual es un aspecto fundamental para asegurar que las personas puedan interactuar con la aplicación de manera fácil y agradable. Desde el inicio, se ha buscado un diseño que resulte cómodo para todo tipo de usuarios, con colores suaves y contrastes adecuados que no resulten agresivos a la vista y que faciliten la lectura.

En primer lugar, se trabajó en el desarrollo de la marca “*Skin Check!*”. Se optó por utilizar un color rosado que transmite profesionalidad y calidez al mismo tiempo. Esta gama cromática ayuda a proyectar una imagen confiable y cercana, además de ser agradable a la vista. Se diseñó el logotipo con la intención de que fuera reconocible y sencillo, y a partir de él se definió la identidad visual general: tipografías limpias, tonos pastel, y elementos gráficos redondeados para inspirar cercanía y confianza.

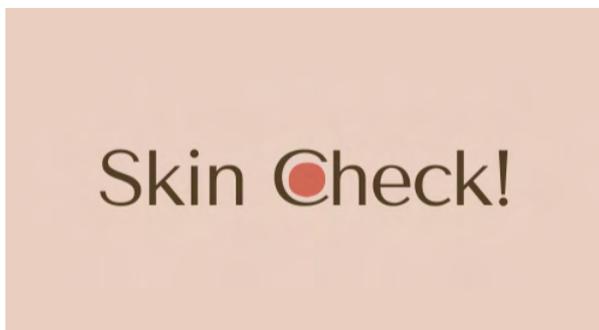


Ilustración 62. Logo App



*Ilustración 61. Logo
Canva*

6.3.2.2 Pantallas Más Relevantes

La navegación es sencilla y clara: una pantalla de bienvenida da paso al inicio de sesión y al registro, mientras que las secciones principales (*PhotoCheck*, Información, Resultados y Perfil) se organizan en una barra inferior. De este modo, el usuario puede tomar o seleccionar una foto para su análisis (*PhotoCheck*), consultar recursos de salud (Información), ver su historial de exámenes (Resultados) o editar sus datos de cuenta (Perfil), todo ello en un entorno visual coherente y fácil de usar.



Ilustración 63. Inicio

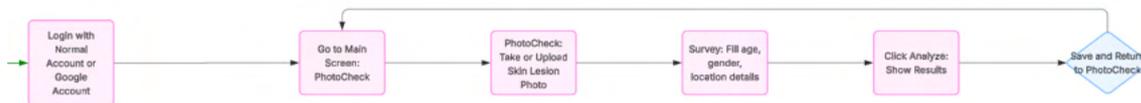


Ilustración 64. PhotoCheck

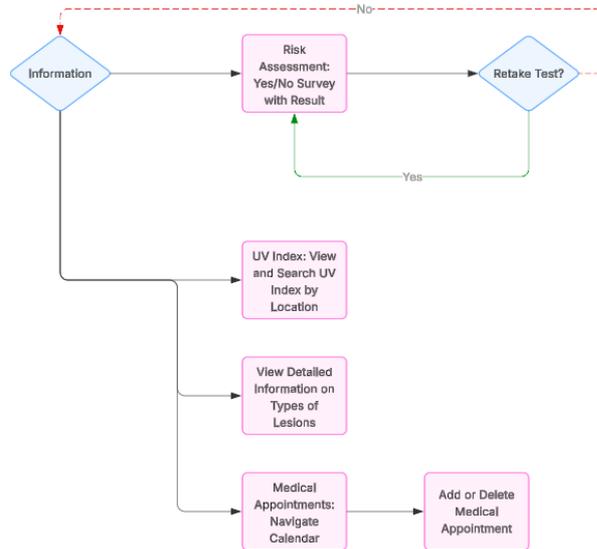


Ilustración 65. Información



Ilustración 66. Historial

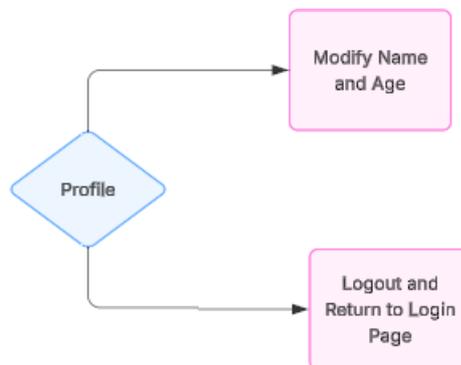


Ilustración 67. Perfil

Capítulo 7. ANÁLISIS DE RESULTADOS

7.1 RESULTADOS DEL MODELO

7.1.1 MÉTRICAS OBTENIDAS DE CADA MODELO

Para evaluar el rendimiento de los modelos desarrollados, se presentan los resultados obtenidos tanto en los conjuntos de entrenamiento como de prueba, considerando las dos tareas principales: clasificación del tipo de lesión dermatológica y predicción de malignidad.

Se ha utilizado el informe de clasificación proporcionado por la biblioteca Sklearn. Este informe incluye varias métricas fundamentales para evaluar modelos de clasificación supervisada, que a continuación se detallan:

- **Precision (Precisión):** mide cuántas de las predicciones positivas realizadas por el modelo son correctas. Es especialmente importante cuando el costo de un falso positivo es alto.
Ejemplo: si el modelo predice que una lesión es maligna, la precisión indica con qué frecuencia esta predicción es acertada.
- **Recall (Sensibilidad o exhaustividad):** mide cuántas de las verdaderas clases positivas fueron correctamente identificadas. Es clave cuando el objetivo es minimizar los falsos negativos.
Ejemplo: cuántas lesiones malignas reales fueron correctamente identificadas por el modelo como malignas.
- **F1-score:** es la media armónica entre la precisión y el *recall*. Proporciona un equilibrio entre ambas métricas, siendo útil cuando hay un desbalance entre clases.
- **Support:** indica la cantidad de muestras reales presentes en cada clase del conjunto de datos.
- **Accuracy:** mide el porcentaje total de predicciones correctas sobre todas las clases. Sin embargo, en problemas con clases desbalanceadas puede resultar engañosa.

- **Macro avg:** es el promedio simple de las métricas de todas las clases, sin tener en cuenta el número de muestras en cada clase.
- **Weighted avg:** es el promedio ponderado de las métricas según la cantidad de muestras en cada clase (*support*). Ofrece una visión más representativa cuando las clases están desbalanceadas.

Clasificación de Lesiones Dermatológicas

Classification report:

	precision	recall	f1-score	support
BCC	0.86	0.86	0.86	510
MEL	0.87	0.89	0.88	510
NV	0.89	0.94	0.91	510
OTH	0.84	0.85	0.84	510
SCC	0.92	0.85	0.88	510
SK	0.84	0.83	0.83	510
accuracy			0.87	3060
macro avg	0.87	0.87	0.87	3060
weighted avg	0.87	0.87	0.87	3060

Ilustración 68. Classification Report on Train Lesion

Classification report:

	precision	recall	f1-score	support
BCC	0.87	0.86	0.86	450
MEL	0.88	0.93	0.90	450
NV	0.89	0.93	0.91	450
OTH	0.83	0.84	0.83	450
SCC	0.94	0.87	0.90	450
SK	0.87	0.85	0.86	450
accuracy			0.88	2700
macro avg	0.88	0.88	0.88	2700
weighted avg	0.88	0.88	0.88	2700

Ilustración 69. Classification Report on Test Lesion

En esta tarea, el modelo debía clasificar imágenes dermatológicas en seis categorías: BCC (Carcinoma Basocelular), MEL (Melanoma), NV (Nevus), OTH (Otras), SCC (Carcinoma Escamoso) y SK (Queratosis Seborreica).

Se observa que el modelo obtiene resultados sólidos tanto en el conjunto de entrenamiento como en el de prueba.

En el conjunto de entrenamiento, se alcanza una precisión global (*accuracy*) del 87%, con una f1-score media (*macro avg*) también del 87%. En el conjunto de prueba, se observa una ligera mejora, con una *accuracy* del 88% y macro f1-score del 88% también.

Estos resultados indican que el modelo generaliza bien y no está sobreajustado (*overfitted*).

En cuanto a las clases individuales:

- **SCC** es consistentemente la clase mejor clasificada, con una precisión de 0.94 y f1-score de 0.90 en el conjunto de prueba. Esto podría deberse a que sus características visuales son más distintivas.
- **NV** también destaca con un *recall* de 0.93 y f1-score de 0.91, lo que indica que es detectado correctamente en la mayoría de los casos.
- **OTH** y **SK** presentan los valores más bajos, lo que sugiere que estas clases podrían tener mayor solapamiento visual con otras o menor representación en el *dataset*.

Esto podría indicar la necesidad de aplicar técnicas de mejora, como aumento de datos (*data augmentation*) o mayor recolección de imágenes para esas clases.

Clasificación de Malignidad:

 Classification report:

	precision	recall	f1-score	support
benign	0.75	0.78	0.77	1700
malignant	0.77	0.74	0.76	1700
accuracy			0.76	3400
macro avg	0.76	0.76	0.76	3400
weighted avg	0.76	0.76	0.76	3400

Ilustración 70. Classification Report on Train Mal

 Classification report:

	precision	recall	f1-score	support
benign	0.77	0.78	0.77	1500
malignant	0.77	0.76	0.77	1500
accuracy			0.77	3000
macro avg	0.77	0.77	0.77	3000
weighted avg	0.77	0.77	0.77	3000

Ilustración 71. Classification Report on Test Mal

Para esta segunda tarea, se utilizó un modelo binario que distingue entre lesiones benignas y malignas. Aunque se trata de una clasificación más simple en número de clases, los resultados no son significativamente superiores, lo que refleja la dificultad clínica de diferenciar lesiones en base únicamente a imágenes.

En el conjunto de entrenamiento, el modelo alcanza una *accuracy* de 76%, con f1-score similares para ambas clases (*benign*: 0.77, *malignant*: 0.76), como se muestra en la Ilustración 57. En el conjunto de prueba, se observa una leve mejora, alcanzando una *accuracy* del 77% y f1-score igualados para ambas clases.

Es importante señalar que el modelo tiene un comportamiento balanceado entre benigno y maligno, lo cual es positivo desde el punto de vista clínico, ya que evita favorecer una clase frente a otra. Sin embargo, aún existe margen de mejora para aumentar la sensibilidad del

modelo a lesiones malignas (*recall* de 0.76), dado el riesgo que implica un falso negativo en contexto médico.

Resumen Comparativo de Resultados:

Tarea	Conjunto	Accuracy	Macro F1-score	Clases
Tipo de Lesión	Train	87%	87%	6
Tipo de Lesión	Test	88%	88%	6
Malignidad	Train	76%	76%	2
Malignidad	Test	77%	77%	2

Tabla 7. Resumen Comparativo Resultados de Modelos

7.1.2 GRÁFICAS DE DESEMPEÑO

Además de las métricas numéricas presentadas en el apartado anterior, se han generado las matrices de confusión correspondientes a ambos modelos, tanto en los conjuntos de entrenamiento como de prueba. Estas gráficas permiten visualizar de manera clara los aciertos y errores del modelo en cada clase específica.

Clasificación Tipos de Lesión:

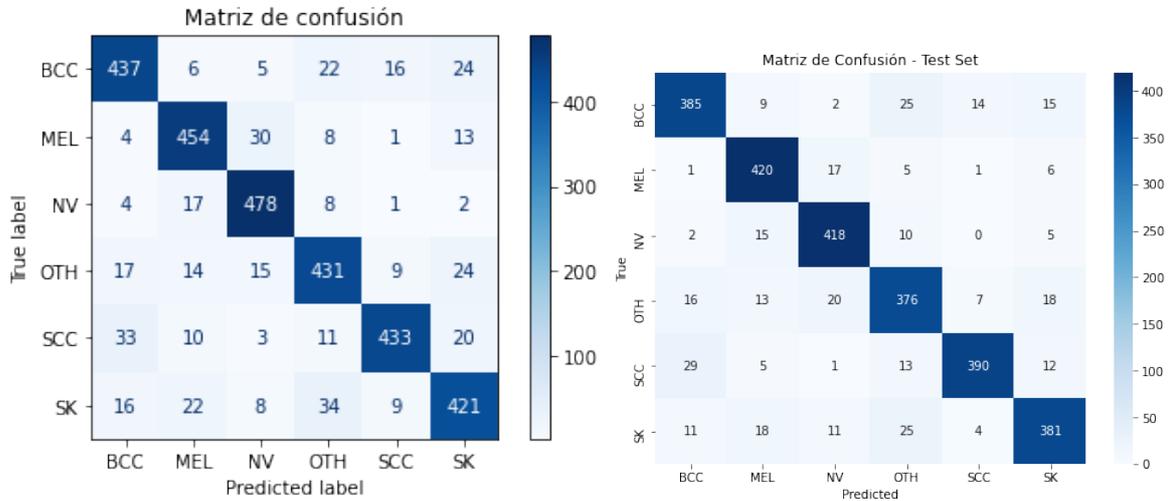


Ilustración 72. Matrices de Confusión Tipo de Lesión

Las dos matrices de confusión representan los resultados para las seis clases dermatológicas: **BCC**, **MEL**, **NV**, **OTH**, **SCC** y **SK**.

En el conjunto de entrenamiento (izquierda), se observa una alta precisión global, con la mayoría de las predicciones correctamente ubicadas en la diagonal. Por ejemplo:

- **NV** presenta 478 aciertos sobre 510 muestras, lo que demuestra que el modelo distingue esta clase con facilidad.
- **SCC** también muestra una fuerte consistencia con 433 aciertos, aunque con cierta confusión con **BCC** y **OTH**.
- Las clases con más confusión son **SK** y **OTH**, donde se aprecian errores cruzados entre sí y con otras clases, lo cual puede deberse a similitudes visuales en las imágenes.

En el conjunto de *test* (derecha), el patrón se mantiene estable:

- **MEL**, **NV** y **SCC** continúan destacando con buenos resultados.
- Las confusiones más frecuentes siguen apareciendo entre **OTH**, **SK** y **BCC**, lo cual sugiere que estos tipos de lesiones podrían beneficiarse de un aumento de datos o un refinamiento en la arquitectura.

Estas gráficas confirman que el modelo tiene un rendimiento equilibrado, aunque se identifican áreas específicas de mejora en clases más ambiguas.

Clasificación Malignidad:

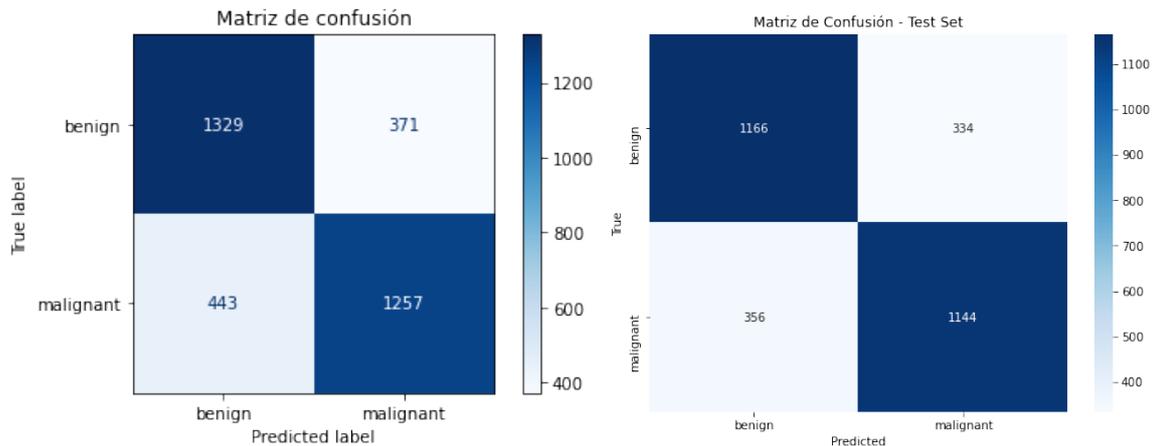


Ilustración 73. Matrices de Confusión Malignidad

En el conjunto de entrenamiento (izquierda), el modelo logra identificar correctamente:

- 1329 lesiones benignas.
- 1257 lesiones malignas.
- Sin embargo, se observan 443 falsos negativos (lesiones malignas clasificadas como benignas), lo cual es especialmente delicado en el ámbito clínico.

En el conjunto de prueba (derecha), la precisión se mantiene similar:

- 1166 benignas correctamente clasificadas.
- 1144 malignas correctamente clasificadas.
- 356 falsos negativos, lo que supone una ligera mejora respecto al conjunto de entrenamiento.

Aunque el modelo muestra un desempeño balanceado, es crucial destacar que el número de falsos negativos sigue siendo un punto crítico, ya que podría conllevar riesgos médicos si

una lesión maligna no se detecta correctamente. Esto resalta la necesidad de utilizar este sistema como apoyo al diagnóstico médico, nunca como sustituto.

7.1.3 ANÁLISIS CUALITATIVO

Con el fin de evaluar el comportamiento del modelo durante el proceso de entrenamiento, se han registrado y graficado las curvas de precisión (*accuracy*) y pérdida (*loss*) tanto en los conjuntos de entrenamiento como de validación para cada una de las tareas: clasificación de tipos de lesión y predicción de malignidad.

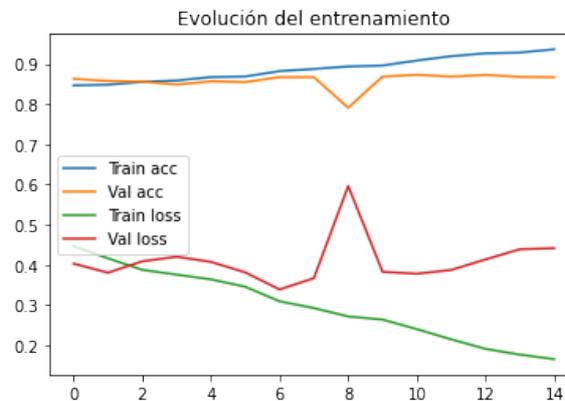


Ilustración 74. Análisis de Entrenamiento Lesiones

En la gráfica se observa una evolución progresiva del entrenamiento:

- La precisión en entrenamiento (train acc) y en validación (val acc) se mantiene alta y cercana al 90%, lo que indica que el modelo logra aprender eficazmente las características relevantes para diferenciar entre las clases.
- La pérdida de entrenamiento (train loss) disminuye de forma constante, mientras que la pérdida de validación (val loss) muestra una ligera oscilación a partir de la época 7, lo cual podría ser un indicio de sobreajuste incipiente (overfitting), aunque no muy acusado.

- A pesar de estas fluctuaciones, la precisión en validación se mantiene estable y no se observa una caída brusca en el rendimiento, lo que confirma una buena generalización del modelo.

Este comportamiento es típico en modelos bien ajustados que han sido entrenados con un número de épocas adecuado y técnicas de regularización como *early stopping* o *dropout*.

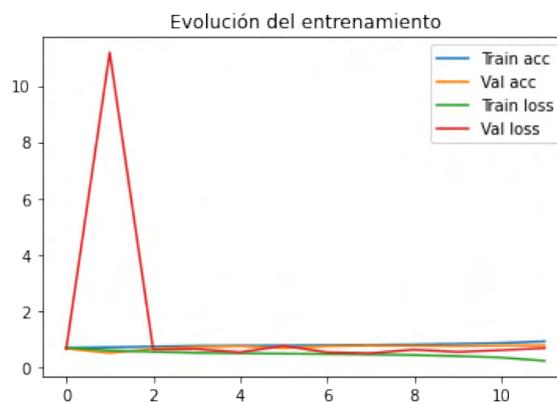


Ilustración 75. Análisis de Entrenamiento de Malignidad

En esta gráfica se aprecian ciertas peculiaridades:

- En las primeras épocas, la pérdida de validación (val loss) muestra un pico elevado, superando incluso un valor de 10. Esta anomalía podría deberse a una mala inicialización del modelo o a una distribución conflictiva de las muestras en los primeros pasos del entrenamiento.
- Sin embargo, tras este pico inicial, el modelo se estabiliza rápidamente y tanto la precisión como la pérdida convergen de forma adecuada.
- Las curvas de precisión (train acc y val acc) se mantienen en torno al 77%, coherente con las métricas evaluadas previamente.

Este tipo de comportamiento es relativamente común cuando se trabaja con datos clínicos complejos y puede mitigarse en futuras iteraciones mediante una mejor normalización de datos o estrategias de *learning rate scheduling* más ajustadas.

En conclusión, ambas gráficas reflejan un proceso de aprendizaje estable y efectivo, con una evolución coherente entre entrenamiento y validación. A pesar de ciertos altibajos puntuales, especialmente en el modelo binario, los resultados obtenidos validan la eficacia de las arquitecturas seleccionadas y su capacidad de generalización sobre datos no vistos.

Capítulo 8. CONCLUSIONES Y TRABAJOS

FUTUROS

A lo largo de este Trabajo de Fin de Grado se ha desarrollado un sistema inteligente capaz de analizar imágenes dermatológicas junto con información complementaria del paciente, con el objetivo de detectar automáticamente el tipo de lesión cutánea y evaluar su posible malignidad. Para ello, se ha diseñado y entrenado un conjunto de modelos de aprendizaje profundo que, a partir de fotografías de lesiones e información clínica (edad, sexo y ubicación anatómica), pueden proporcionar un diagnóstico automatizado con alta precisión.

Una de las principales decisiones de diseño fue la separación del problema en dos tareas diferenciadas: por un lado, la predicción de la malignidad (benigno o maligno), y por otro, la clasificación del tipo de lesión dermatológica (entre seis clases: BCC, MEL, NV, OTH, SCC y SK). Esta aproximación modular ha permitido optimizar cada arquitectura para su respectiva tarea, mejorar el rendimiento global y facilitar la interpretación de los resultados. Ambos modelos fueron entrenados por separado, utilizando técnicas de aumento de datos, regularización y validación estratificada, y su rendimiento fue evaluado sobre un conjunto de *test* independiente.

Además del diseño de los modelos, se ha prestado especial atención al preprocesamiento de los datos, asegurando un equilibrio en la distribución de clases, la limpieza de valores ausentes y la correcta codificación de los metadatos. El modelo ha sido integrado dentro de una arquitectura pensada para ser incorporada en una aplicación móvil, lo que facilita su accesibilidad y su potencial uso por parte de usuarios sin conocimientos técnicos.

Los resultados obtenidos muestran un rendimiento sólido. El modelo de clasificación de tipo de lesión ha alcanzado una precisión superior al 87%, mientras que el modelo de predicción de malignidad ha logrado más del 73% de acierto en el conjunto de validación. Estas cifras avalan la viabilidad técnica del sistema como herramienta de apoyo al diagnóstico clínico,

especialmente en contextos donde el acceso a dermatólogos es limitado o existen listas de espera prolongadas.

Aunque existen múltiples algoritmos clásicos de clasificación supervisada que permiten evaluar distintos métodos de balanceo (como KNN, SVM, o Random Forest), en este trabajo se ha optado por el uso de redes neuronales convolucionales (CNNs) por su alta eficacia demostrada en el tratamiento de imágenes médicas. Las técnicas de *undersampling* y *data augmentation* se han aplicado directamente sobre el *dataset* para entrenar estos modelos de *deep learning*, ya que en este contexto, la prioridad era obtener un modelo funcional y robusto que pudiera integrarse en una aplicación móvil y generalizar correctamente ante nuevas imágenes clínicas.

En trabajos futuros, se podrían comparar diferentes estrategias de preprocesamiento y algoritmos clásicos como parte de un análisis más exhaustivo de técnicas de clasificación.

Aportaciones principales del trabajo:

- Desarrollo de una arquitectura multimodal que combina imágenes con metadatos clínicos del paciente.
- Implementación y entrenamiento de dos modelos de *deep learning* especializados en tareas médicas diferenciadas.
- Aplicación de técnicas de balanceo de clases y *data augmentation* personalizado para mejorar la robustez del sistema.
- Propuesta de un flujo completo desde la carga de datos hasta la predicción, con vistas a su integración en una aplicación de salud.
- Contribución a la detección temprana del cáncer de piel, facilitando un primer cribado automático que puede alertar a los usuarios.

8.1 TRABAJOS FUTUROS

A pesar de los avances logrados, el proyecto abre la puerta a múltiples líneas de trabajo y mejora. Uno de los aspectos más prometedores sería establecer colaboraciones con centros médicos o hospitales, con el fin de validar el sistema en un entorno clínico real. Esta validación externa permitiría ajustar los modelos con datos más específicos y evaluar su utilidad como herramienta de cribado complementaria en consultas dermatológicas.

Además, me gustaría que en un futuro la aplicación integre una pasarela directa de contacto con profesionales sanitarios. Por ejemplo, si el sistema detecta que una lesión presenta un perfil potencialmente maligno, el usuario podría recibir una notificación recomendándole contactar con un médico. Esta notificación incluiría la posibilidad de solicitar automáticamente una cita presencial o incluso de programar una consulta online con un especialista dermatológico. Esta funcionalidad convertiría la aplicación no solo en una herramienta de diagnóstico preliminar, sino en un puente efectivo entre la inteligencia artificial y la atención médica profesional, fomentando una atención más rápida y personalizada.

Asimismo, se plantea la posibilidad de ampliar la base de datos con otras variables clínicas relevantes, como antecedentes médicos, historial de lesiones previas, o incluso imágenes de seguimiento temporal. También sería interesante integrar técnicas de interpretación de modelos (como Grad-CAM) para visualizar qué regiones de la imagen han influido más en la predicción, mejorando la transparencia del sistema ante profesionales médicos.

Finalmente, para alcanzar un rendimiento profesional y asegurar una disponibilidad y capacidad de respuesta adecuadas ante una mayor demanda, sería necesario desplegar la solución en arquitecturas de pago, con servicios escalables que permitan mantener la calidad del servicio en un entorno real de producción.

BIBLIOGRAFÍA

- [1] Flutter Team, «Flutter,» 2024. [En línea]. Available: <https://flutter.dev/>. [Último acceso: 1 2025].
- [2] Firebase Team, «Cloud Firestore,» 2024. [En línea]. Available: <https://firebase.google.com/products/firestore>. [Último acceso: 1 2025].
- [3] ISIC Archive, «ISIC Archive,» International Skin Imaging Collaboration, 2024. [En línea]. Available: <https://www.isic-archive.com/>. [Último acceso: 2 2025].
- [4] Supabase Team, «Supabase,» 2024. [En línea]. Available: <https://supabase.com/>. [Último acceso: 1 2025].
- [5] Render, «Render,» 2024. [En línea]. Available: <https://render.com/>. [Último acceso: 1 2025].
- [6] Python Software Foundation, «Python,» 2025. [En línea]. Available: <https://www.python.org/doc/>. [Último acceso: 10 1 2025].
- [7] Pandas Development Team, «Pandas,» 2024. [En línea]. Available: <https://pandas.pydata.org/>. [Último acceso: 10 1 2025].
- [8] Numpy Developers, «Numpy,» 2024. [En línea]. Available: <https://numpy.org/>. [Último acceso: 1 2025].
- [9] Scikit-learn developers, «Scikit-learn,» 2025. [En línea]. Available: <https://scikit-learn.org/>. [Último acceso: 1 2025].
- [10] PyTorch Team, «PyTorch,» 2024. [En línea]. Available: <https://pytorch.org/>. [Último acceso: 1 2025].
- [11] Dart Team, «Dart,» 2024. [En línea]. Available: <https://dart.dev/>. [Último acceso: 1 2025].
- [12] Firebase Team, «Realtime Database,» 2024. [En línea]. Available: <https://firebase.google.com/products/realtime-database>. [Último acceso: 1 2025].
- [13] Firebase Team, «Firebase Authentication,» 2024. [En línea]. Available: <https://firebase.google.com/products/auth>. [Último acceso: 1 2025].

- [14] Firebase Team, «Cloud Storage,» 2024. [En línea]. Available: <https://firebase.google.com/products/storage>. [Último acceso: 1 2025].
- [15] Firebase Team, «Cloud Functions,» 2024. [En línea]. Available: <https://firebase.google.com/products/functions>. [Último acceso: 1 2025].
- [16] Firebase Team, «Google Analytics for Firebase,» 2024. [En línea]. Available: <https://firebase.google.com/products/analytics>. [Último acceso: 1 2025].
- [17] Microsoft, «Visual Studio Code,» 2024. [En línea]. Available: <https://code.visualstudio.com/>. [Último acceso: 1 2025].
- [18] PostgreSQL Global Development Group, «PostgreSQL,» 2024. [En línea]. Available: <https://www.postgresql.org/>. [Último acceso: 1 2025].
- [19] JetBrains, «IntelliJ IDEA,» 2024. [En línea]. Available: <https://www.jetbrains.com/idea/>. [Último acceso: 1 2025].
- [20] Android Developers, «Android Studio,» 2024. [En línea]. Available: <https://developer.android.com/studio>. [Último acceso: 1 2025].
- [21] Git SCM, «Git,» 2024. [En línea]. Available: <https://git-scm.com/>. [Último acceso: 1 2025].
- [22] JetBrains, «Kotlin,» 2024. [En línea]. Available: <https://kotlinlang.org/>. [Último acceso: 1 2025].
- [23] Oracle, «Java,» 2024. [En línea]. Available: <https://www.oracle.com/java/>. [Último acceso: 1 2025].
- [24] GitHub, Inc., «GitHub Actions,» 2024. [En línea]. Available: <https://github.com/features/actions>. [Último acceso: 1 2025].
- [25] GitHub, Inc, «GitHub,» 2024. [En línea]. Available: <https://github.com/>. [Último acceso: 1 2025].
- [26] OpenJS Foundation, «Node.js,» 2024. [En línea]. Available: <https://nodejs.org/>. [Último acceso: 1 2025].
- [27] Ruby Lang, «Ruby Programming Language,» 2024. [En línea]. Available: <https://www.ruby-lang.org/>. [Último acceso: 1 2025].
- [28] Go Team (Google), «Go Programming Language,» 2024. [En línea]. Available: <https://go.dev/>. [Último acceso: 1 2025].

- [29] GitLab Inc., «GitLab,» 2024. [En línea]. Available: <https://about.gitlab.com/>. [Último acceso: 1 2025].
- [30] P. Tschandl, C. Rosendahl y H. Kittler, «The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions,» *Scientific Data*, vol. 5, p. 180161, 2018.
- [31] ADDI Project – Universidade do Porto, «H² Database – Pedro Hispano Dermoscopic Image Database,» ADDI Project, Universidade do Porto, 2013. [En línea]. Available: <https://www.fc.up.pt/addi/ph2%20database.html>. [Último acceso: 2 2025].
- [32] P. Tschandl, N. Codella, B. Rinner y e. al., «Human–computer collaboration for skin cancer recognition,» *Nature Medicine*, vol. 26, nº 8, p. 1229–1234, 2020.
- [33] J. Alonso-Castro, A. Hernández-Fernández y P. Jaén-Olasolo, «Actas Dermo-Sifiliográficas,» Elsevier España, 1 9 2022. [En línea]. Available: <https://www.actasdermo.org/es-inteligencia-artificial-dermatologia-amenaza-u-articulo-S0001731021002957>. [Último acceso: 1 2025].
- [34] AprendeIA (marca la casilla “Autor corporativo”), «Clasificación de Machine Learning,» AprendeIA, 22 3 2018. [En línea]. Available: <https://aprendeia.com/2018/03/22/clasificacion-de-machine-learning/>. [Último acceso: 2 2025].
- [35] Algotive, «Machine Learning: What is ML and how does it work?,» Algotive, 10 2023. [En línea]. Available: <https://www.algotive.ai/blog/machine-learning-what-is-ml-and-how-does-it-work>. [Último acceso: 2 2025].
- [36] M. Fernández, «Cómo funcionan las Convolutional Neural Networks (visión por ordenador),» Aprende Machine Learning, 11 2018. [En línea]. Available: <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>. [Último acceso: 2 2025].
- [37] H. A. Haenssle, C. Fink, R. Schneiderbauer y e. al., «Man against machine: diagnostic performance of a deep learning convolutional neural network for dermatoscopic melanoma recognition in comparison to 58 dermatologists,» *Annals of Oncology*, 8 2018. [En línea]. Available: [https://www.annalsofoncology.org/article/S0923-7534\(19\)34105-5/fulltext](https://www.annalsofoncology.org/article/S0923-7534(19)34105-5/fulltext). [Último acceso: 2 2025].
- [38] Y. Wang, D. Wu, T. Zhang, Y. Lin y Y. Liu, «Comparative analysis of deep learning CNN models for classification of lung nodules in CT images,» *Computers in Biology and Medicine*, vol. 165, p. 107425, 2023.
- [39] Eczemaless, «Las 5 mejores aplicaciones de dermatología con IA que revolucionan el cuidado de la piel,» Eczemaless, [En línea]. Available: <https://eczemaless.com/es/las-5-mejores-aplicaciones-de-dermatologia-con-ia-que-revolucionan-el-cuidado-de-la-piel/>. [Último acceso: 3 2025].

- [40] Tibot AI, «Tibot AI: A Quantum Leap in Dermatology Through Artificial Intelligence,» Medium, 2023. [En línea]. Available: <https://medium.com/@tibotai9473/tibot-ai-a-quantum-leap-in-dermatology-through-artificial-intelligence-bab5e2f09018>. [Último acceso: 3 2025].
- [41] Mercer University School of Medicine, «DermExpert App Overview,» Mercer University, 2023. [En línea]. Available: <https://med.mercer.edu/library/DermExpert.htm>. [Último acceso: 3 2025].
- [42] R. Paith, «Telehealth application around globe – SkinVision,» rahulpaith.com, 2021. [En línea]. Available: <https://rahulpaith.com/telehealth-application-around-globe-skinvision/>. [Último acceso: 3 2025].
- [43] U.S. Department of Health & Human Services, «HIPAA for Professionals,» U.S. Government, 2024. [En línea]. Available: <https://www.hhs.gov/hipaa/for-professionals/index.html>. [Último acceso: 3 2025].
- [44] European Union, «General Data Protection Regulation (GDPR),» Publications Office of the European Union, 2016. [En línea]. Available: <https://eur-lex.europa.eu/eli/reg/2016/679/oj>. [Último acceso: 3 2025].
- [45] «Wikipedia,» Wikimedia Foundation, 2024. [En línea]. Available: https://es.wikipedia.org/wiki/Computaci%C3%B3n_en_la_nube. [Último acceso: 3 2025].
- [46] Amazon Web Services, «Amazon Web Services,» Amazon, 2024. [En línea]. Available: <https://aws.amazon.com/what-is-cloud-computing/>. [Último acceso: 3 2025].
- [47] Google Cloud, «Google Cloud,» Google LLC, 2024. [En línea]. Available: <https://cloud.google.com/learn/what-is-cloud-computing>. [Último acceso: 3 2025].
- [48] Microsoft Azure, «Microsoft Azure,» Microsoft Corporation, 2023. [En línea]. Available: <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-cloud-computing/>. [Último acceso: 3 2025].
- [49] European Union, «Official Journal of the European Union, Regulation (EU) 2024/1083 on Artificial Intelligence (AI Act),» European Union, 2024. [En línea]. Available: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32024R1083>. [Último acceso: 3 2025].
- [50] European Union, «Medical Device Regulation (MDR),» Publications Office of the European Union, 2017. [En línea]. Available: <https://eur-lex.europa.eu/eli/reg/2017/745/oj>. [Último acceso: 3 2025].
- [51] U.S. Food and Drug Administration, «FDA - Medical Devices,» U.S. Government, 2024. [En línea]. Available: <https://www.fda.gov/medical-devices>. [Último acceso: 3 2025].

- [52] State of Illinois, «Biometric Information Privacy Act (BIPA),» Illinois General Assembly, 2008. [En línea]. Available: <https://www.ilga.gov/legislation/ilcs/ilcs3.asp?ActID=3004>. [Último acceso: 3 2025].
- [53] Meta, «Introduction – React Native,» Meta Platforms, Inc., 2024. [En línea]. Available: <https://reactnative.dev/docs/getting-started>. [Último acceso: 1 2025].
- [54] Microsoft, «What is Xamarin?,» Microsoft, 2024. [En línea]. Available: <https://learn.microsoft.com/en-us/xamarin/>. [Último acceso: 1 2025].
- [55] ISIC Archive, «ISIC 2024 Challenge: Training Dataset,» International Skin Imaging Collaboration, 2024. [En línea]. Available: <https://api.isic-archive.com/collections/390/>. [Último acceso: 2 2025].
- [56] Baseflow, «Geocator for Flutter – API Documentation,» pub.dev, 2024. [En línea]. Available: <https://pub.dev/packages/geocator>. [Último acceso: 3 2025].

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

El proyecto SkinCheck se alinea con varios Objetivos de Desarrollo Sostenible (ODS) establecidos por la Agenda 2030 de las Naciones Unidas, especialmente en lo que respecta a la salud, la innovación tecnológica y la reducción de desigualdades.



Esta aplicación contribuye de manera directa al ODS 3: Salud y Bienestar, al ofrecer una herramienta accesible para la detección temprana de posibles enfermedades dermatológicas, lo cual puede facilitar el diagnóstico precoz de lesiones malignas y, por tanto, mejorar el pronóstico y la calidad de vida de los pacientes. El acceso a este tipo de tecnología fomenta una mayor conciencia sobre la salud de la piel y promueve hábitos de prevención.



Asimismo, SkinCheck se vincula con el ODS 9: Industria, Innovación e Infraestructura, al incorporar tecnologías avanzadas como el machine learning, el desarrollo móvil y el uso de plataformas en la nube como Firebase y Supabase. Esto refleja el compromiso del proyecto con la innovación responsable, promoviendo la creación de soluciones tecnológicas aplicadas al ámbito médico y de la salud pública.



Otro de los ODS con los que se relaciona el proyecto es el ODS 10: Reducción de las Desigualdades, ya que la aplicación está diseñada para ser usada desde cualquier lugar, incluyendo zonas rurales o con difícil acceso a centros médicos. Al integrar el modelo de inteligencia artificial directamente en el dispositivo móvil, se elimina la dependencia de conexión a internet, lo cual garantiza un acceso más equitativo al análisis dermatológico preliminar, independientemente de la ubicación geográfica del usuario.



Por último, el enfoque en la privacidad de los datos y el cumplimiento de normativas como el RGPD refuerza el ODS 16: Paz, Justicia e Instituciones Sólidas, promoviendo una gestión ética, segura y transparente de la información sensible del usuario. Este compromiso ético resulta esencial en el desarrollo de tecnologías aplicadas a la salud.



ANEXO II: MANUAL DE INSTALACIÓN

Este Manual de Instalación tiene como finalidad describir de forma clara y ordenada los pasos necesarios para preparar el entorno de trabajo de la aplicación desarrollada en este Trabajo de Fin de Grado. Se detallarán los elementos esenciales, herramientas y configuraciones que deben instalarse para garantizar un funcionamiento correcto del sistema.

A lo largo del proyecto ya se han mencionado las tecnologías empleadas; sin embargo, en este apartado se profundiza en el proceso de instalación de los softwares más relevantes, permitiendo que cualquier usuario pueda replicar el entorno sin complicaciones técnicas.

VISUAL STUDIO CODE

1. Accede a la página principal de Visual Studio Code: <https://code.visualstudio.com>
2. Selecciona tu sistema operativo y haz clic para iniciar la descarga del instalador
3. Ejecuta el instalador. En Windows será un archivo .exe, en macOS será .dmg y en Linux puede ser .deb o .rpm.
4. Seguir los pasos del asistente de instalación:
 - Acepta términos y condiciones
 - Elige la carpeta donde instalar VS Code
 - Agregar Path
 - Instalar

FLUTTER

1. Accede a la web oficial de Flutter: <https://flutter.dev/docs/get-started/install>
2. Selecciona tu sistema operativo
3. Descarga el SDK de Flutter
4. Extrae el archivo descargado y añade Flutter al Path
5. Verifica la instalación escribiendo en una nueva terminal:

- a. flutter doctor
6. Instalar los requisitos adicionales según lo que indique flutter doctor:
 - a. Android Studio
 - b. Xcode
 - c. Extensiones para Flutter y Dart

EMULADOR ANDROID

1. Descarga Android Studio: <https://developer.android.com/studio>
2. Abre Android Studio y configura los componentes:
 - a. SDK Manager
 - b. Android SDK
 - c. Android SDK Platform-Tools
 - d. Android Emulator
 - e. Una versión de Android
3. “Apply” y “ok”
4. Crear un dispositivo virtual:
 - a. More actions → Virtual Device Manager
 - b. Create device
 - c. Elige un modelo y haz clic en “next”
 - d. Selecciona una imagen del sistema y haz clic en “next”
 - e. Finish
5. Verifica que flutter lo detecta:
 - a. Flutter devices
6. Ejecuta tu app Flutter en el emulador:
 - a. Abre tu proyecto con VSC o Android Studio
 - b. Asegúrate de que el emulador está encendido
 - c. Ejecuta: flutter run

VERIFICACIÓN DEL ENTORNO DE DESARROLLO

Una vez completada la instalación de Flutter, Visual Studio Code, Android Studio y, en su caso, el emulador o dispositivo físico, es fundamental verificar que todo el entorno de desarrollo está correctamente configurado para evitar errores futuros durante el desarrollo de la aplicación.

1. Ejecuta flutter doctor. Este comando analiza el estado del entorno de Flutter y devuelve un resumen del sistema, indicando si existen componentes pendientes de instalar o configurar.
2. Comprobar dispositivos disponibles. Ejecuta flutter devices. Este comando debe mostrar un emulador encendido o un teléfono conectado por USB. Si no aparece ningún dispositivo, asegúrate de que:
 - El emulador esté iniciado desde Android Studio.
 - El teléfono físico tenga activada la depuración USB.
3. Puedes crear y ejecutar un proyecto de prueba para verificar que todo funciona:
 - Flutter create app_prueba
 - Cd app_prueba
 - Flutter run

CONFIGURACIÓN DEL DESARROLLO PARA ANDROID

1. Abre Android Studio
2. Ve a Android SDK y selecciona:
 - a. Android SDK Platform-Tools
 - b. Android SDK Build-Tools
 - c. Android SDK Tools
3. Haz clic en “Apply” y “ok”.
4. En una terminal ejecuta:
 - a. Flutter config -android-sdk [ruta del sdk de Android] para configurar Flutter con el SDK de Android

PRUEBA, DEBUG Y DEPLOY DE LA APP

1. Instala el proyecto de la aplicación en tu dispositivo
2. Abre Visual Studio Code y carga el proyecto Flutter
 - a. flutter pub get para descargar todas las librerías necesarias
3. Conecta tu dispositivo o inicia un emulador
4. Ejecuta flutter run o haz clic en Run and Debug para ejecutar la aplicación
5. Para realizar pruebas puedes
 - a. Ver logs en tiempo real desde la consola
 - b. Usar hot reload para aplicar cambios de forma rápida
 - c. Colocar breakpoints en el código para analizar el comportamiento, línea por línea

ANEXO III: MANUAL DE USUARIO DE LA APP

WELCOME

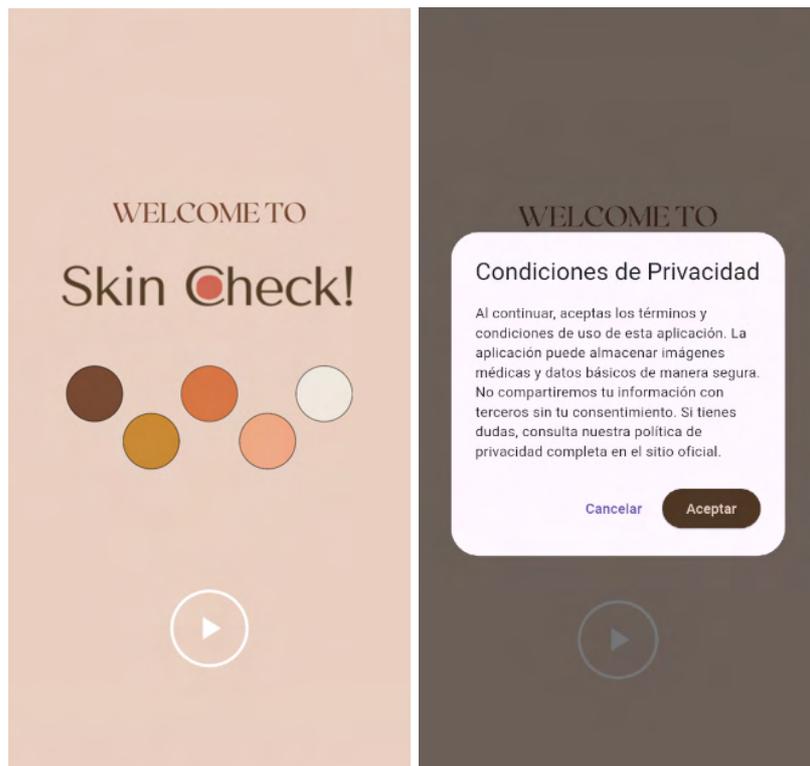
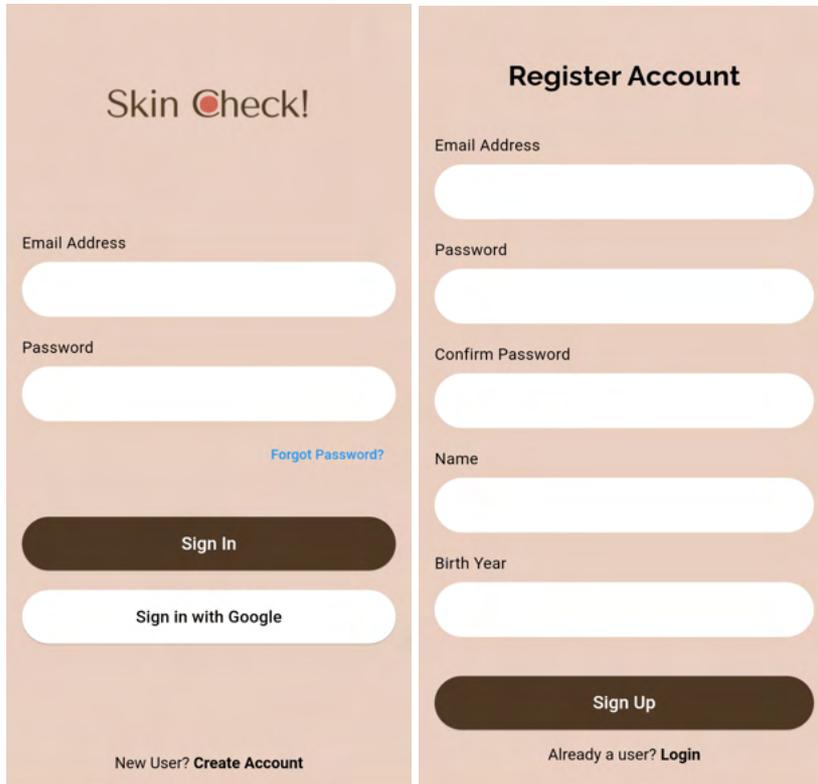


Ilustración 76. Pantalla de Inicio

La pantalla de bienvenida de la aplicación *Skin Check!* recibe al usuario con un diseño cálido y agradable, mostrando el logotipo en el centro y una paleta de tonos suaves que refuerzan la temática de cuidado de la piel. Justo bajo el título, se ofrecen varias opciones de color para adaptar la apariencia de la aplicación a diferentes tonos de piel, transmitiendo inclusión y personalización desde el primer momento. En la parte inferior destaca un gran botón circular con el ícono de reproducción (“*play*”), que al pulsarse conduce a la pantalla de inicio de sesión (*Login*), facilitando así el flujo natural hacia el uso principal de la aplicación.

LOG IN Y REGISTER



The image displays two side-by-side screenshots of a mobile application interface. The left screenshot, titled "Skin Check!", features a login form with fields for "Email Address" and "Password", a "Forgot Password?" link, a "Sign In" button, and a "Sign in with Google" button. At the bottom, it says "New User? Create Account". The right screenshot, titled "Register Account", features a registration form with fields for "Email Address", "Password", "Confirm Password", "Name", and "Birth Year", a "Sign Up" button, and a link for "Already a user? Login".

Ilustración 77. Pantalla de Log in y de Register

La pantalla de *Login* cuenta con un formulario para que el usuario introduzca su correo electrónico y contraseña, junto con un botón que permite recuperar la contraseña en caso de olvido (al pulsar “*Forgot Password?*” se solicita el email y se envía un enlace de restablecimiento). También ofrece la opción de iniciar sesión a través de Google mediante el botón “*Sign in with Google*”. Finalmente, en la parte inferior se encuentra el enlace “*New User? Create Account*”, que redirige a la pantalla de registro para los usuarios que aún no tengan una cuenta creada.

La pantalla de *Register Account* permite al usuario crear una nueva cuenta ingresando su correo electrónico, contraseña y confirmando la misma para evitar errores de tipeo. Además, se pide el nombre y el año de nacimiento como datos adicionales de perfil. Una vez completados todos los campos, con solo pulsar el botón *Sign Up* se realiza el registro. En

caso de ya tener una cuenta, se ofrece el enlace *Login* en la parte inferior, redirigiendo de vuelta a la pantalla de inicio de sesión para quienes deseen iniciar directamente con sus credenciales existentes.

PHOTOCHECK

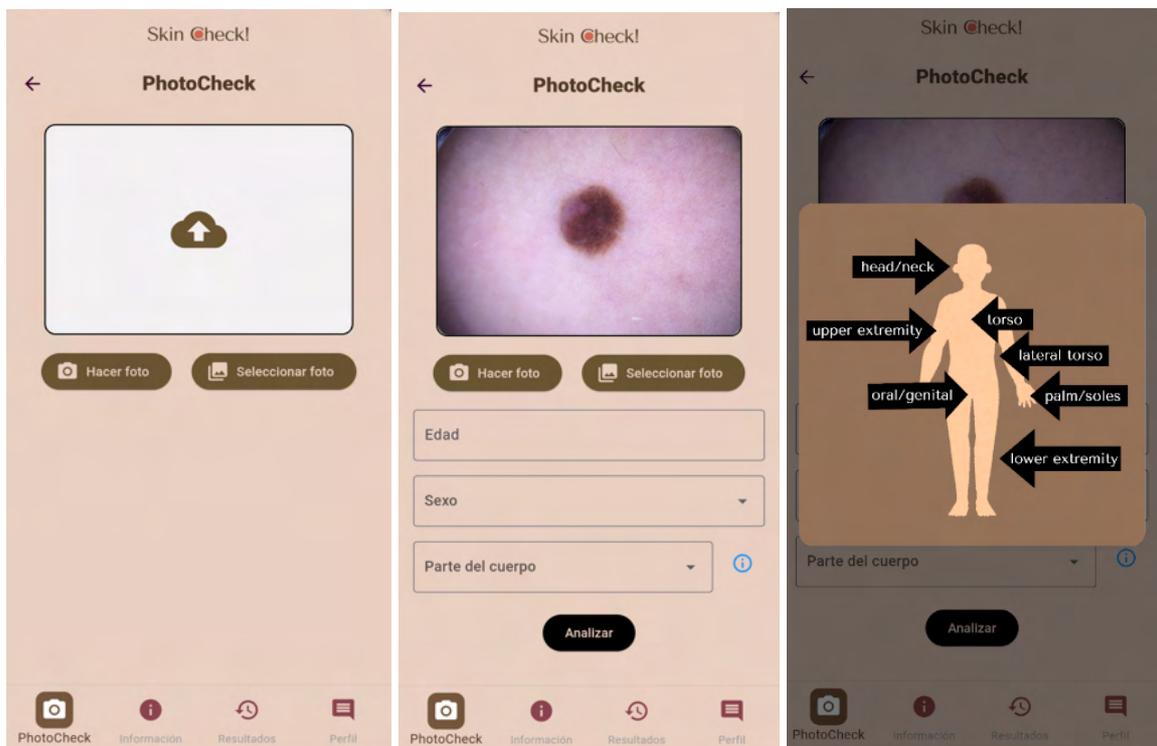


Ilustración 78. Pantalla de Photocheck

La pantalla PhotoCheck permite al usuario tomar o seleccionar una fotografía de la zona de piel que desea analizar. Tras capturar o elegir la imagen, el usuario puede indicar datos adicionales como edad, sexo y parte del cuerpo. Al pulsar el botón Analizar, la aplicación envía la imagen y la información proporcionada al modelo de *machine learning* que se encarga de procesar los datos y ofrecer una primera evaluación o diagnóstico preliminar. De esta forma, se combina la interacción directa (tomar la foto) con la inteligencia artificial para facilitar la monitorización y cuidado de posibles lesiones cutáneas.

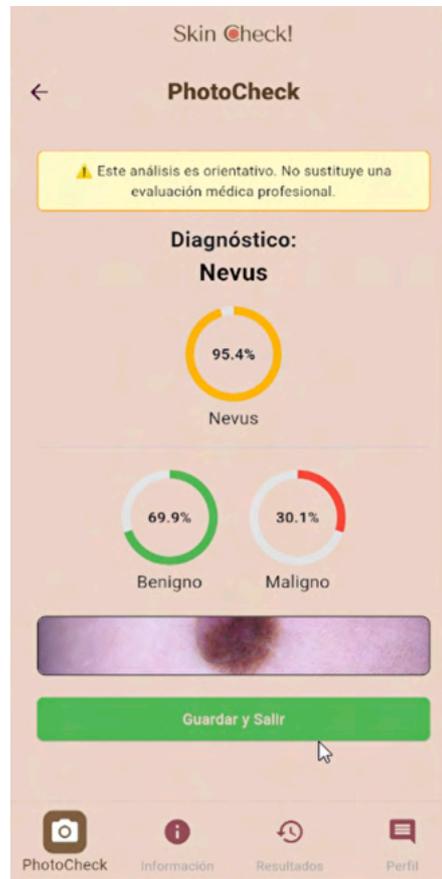


Ilustración 79. Pantalla Análisis

INFORMACIÓN



Ilustración 80. Pantalla de Información

La sección de Información ofrece al usuario un acceso rápido a recursos y contenidos relacionados con el cuidado de la piel. Se organiza en cuatro apartados principales:

TIPOS DE LESIONES:

Donde se describen las distintas lesiones cutáneas que pueden aparecer y se brindan consejos para identificarlas;

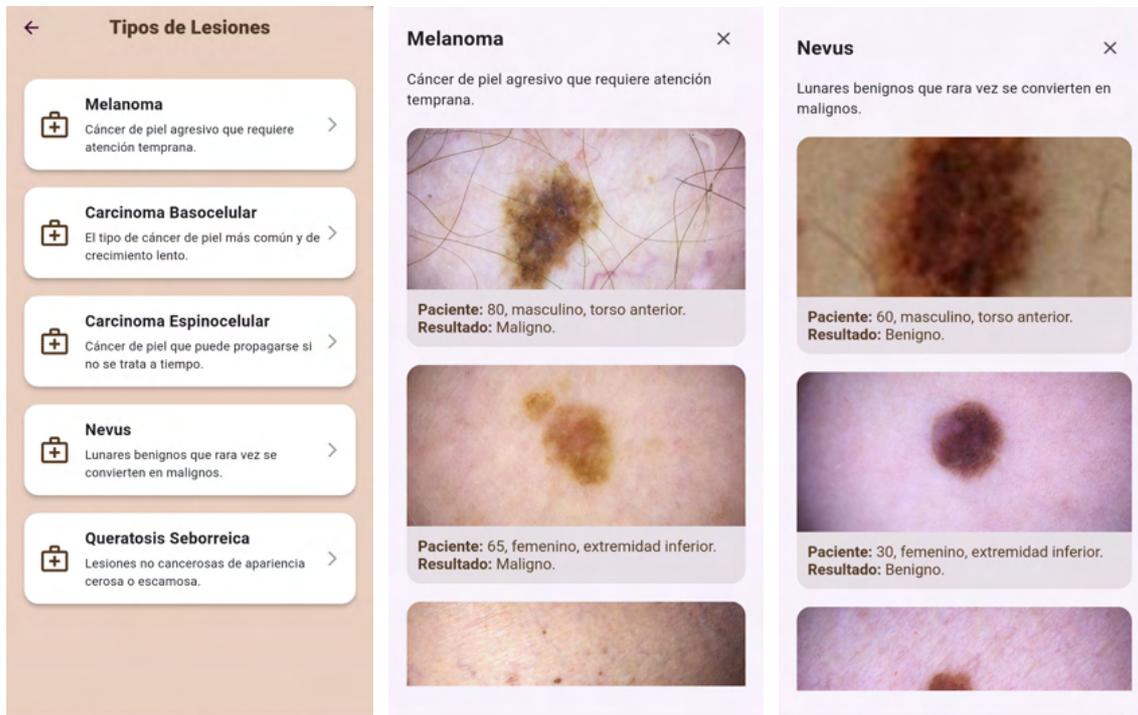


Ilustración 81. Pantalla de Tipos de Lesiones

Esta funcionalidad de la aplicación permite al usuario explorar de forma estructurada los diferentes tipos de lesiones dermatológicas que el modelo es capaz de clasificar. Desde un punto de vista técnico, esta sección tiene como objetivo proporcionar información educativa y visual sobre cada categoría de lesión, contribuyendo así a la interpretabilidad del sistema y a una mejor comprensión por parte del usuario.

La pantalla principal muestra una lista de tarjetas interactivas, cada una correspondiente a una categoría de lesión cutánea, como *Melanoma*, *Carcinoma Basocelular*, *Nevus*, entre otras. Cada tarjeta contiene:

- El **nombre de la lesión**.
- Una **descripción breve** de su naturaleza clínica.
- Un **icono representativo** y una **flecha indicativa** de navegación.

El contenido se carga dinámicamente desde un archivo lesiones.json, lo que facilita la escalabilidad y mantenimiento de la información.

Visualización de imágenes clínicas y metadatos

Al seleccionar una de las tarjetas, se activa un *modal bottom sheet* que presenta:

- Una descripción ampliada de la lesión seleccionada.
- Una galería de imágenes reales correspondientes a dicha clase, tomadas del conjunto de entrenamiento.
- Metadatos asociados a cada imagen, incluyendo:
 - Edad aproximada del paciente.
 - Sexo.
 - Localización anatómica de la lesión.
 - Diagnóstico binario (maligno o benigno).

Estas imágenes se extraen de forma aleatoria de los activos locales (assets) filtrados por código de clase, y se enlazan con sus respectivos metadatos extraídos del archivo `train_todo.csv`. El sistema utiliza un mapa de manifiesto (`AssetManifest.json`) para localizar las imágenes asociadas a cada clase y cruza los identificadores con los datos clínicos para enriquecer la presentación.

EVALUACIÓN DEL RIESGO:

Una guía o herramienta que ayuda a comprender el nivel de riesgo asociado a cada caso, teniendo en cuenta factores como antecedentes personales o características de la lesión;



Ilustración 82. Encuesta Evaluación del Riesgo

Esta sección de la aplicación ofrece un test interactivo de autoevaluación para estimar el riesgo del usuario de padecer cáncer de piel. Su finalidad es preventiva y educativa, permitiendo identificar perfiles con factores de riesgo que podrían beneficiarse de un seguimiento dermatológico más cercano. La funcionalidad se basa en la integración con una base de datos dinámica en Firebase Firestore y está diseñada para ser ligera, intuitiva y adaptable.

Carga dinámica de preguntas

Al iniciar la pantalla, se realiza una consulta a la colección `evaluacionRiesgo` alojada en Firebase Firestore, desde donde se recuperan todas las preguntas disponibles. Posteriormente, el sistema filtra automáticamente aquellas entradas que no contienen preguntas válidas ni opciones bien formateadas. Se seleccionan aleatoriamente hasta 7

preguntas para cada sesión, asegurando variedad en cada ejecución del test. Este enfoque mejora la experiencia de usuario y evita sesgos al repetir siempre las mismas preguntas.

Estructura de la lógica del test

La lógica se organiza de forma secuencial:

- **Presentación de preguntas:** Se muestra una pregunta por pantalla junto a sus posibles respuestas, generalmente del tipo “Sí / No”.
- **Registro de respuestas:** Cada selección se almacena en una lista que mantiene el estado del test.
- **Progreso visual:** Se incluye una barra de progreso (LinearProgressIndicator) que indica al usuario en qué punto del test se encuentra.

Una vez completadas todas las preguntas, se ejecuta la función `_calculateResult`, la cual computa el resultado total basándose en la cantidad de respuestas afirmativas (“Sí”):

- 5 o más respuestas afirmativas → Riesgo **ALTO**.
- 2 respuestas afirmativas → Riesgo **MEDIO**.
- Menos de 2 respuestas afirmativas → Riesgo **MODERADO**.

Este resultado se muestra al usuario mediante un mensaje destacado acompañado de un botón que permite reiniciar el test (`_resetQuiz`), lo cual vuelve a cargar un nuevo conjunto aleatorio de preguntas desde la base de datos.

RAYOS UV:

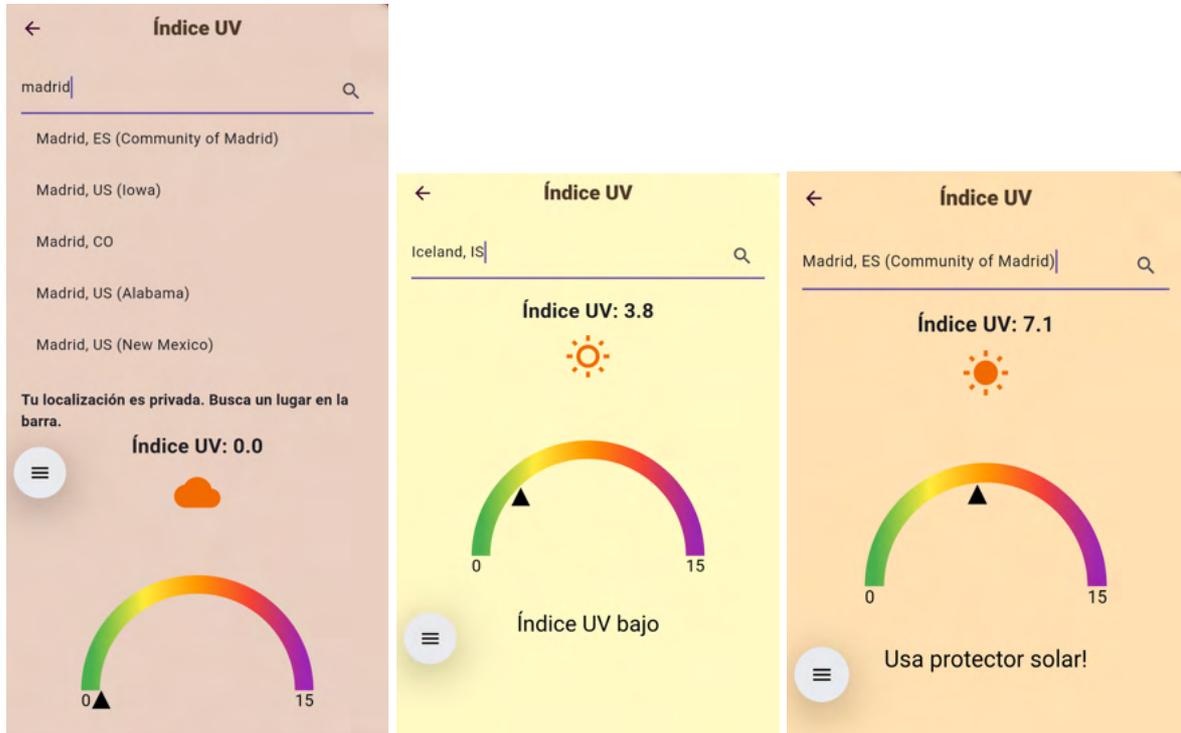


Ilustración 83. Pantalla Índice UV

La pantalla de **Índice UV** proporciona al usuario una estimación en tiempo real de la radiación ultravioleta (UV) presente en su ubicación actual o en cualquier ciudad del mundo. Esta funcionalidad persigue un objetivo preventivo en salud dermatológica, ya que la sobreexposición a la radiación UV es uno de los principales factores de riesgo para el desarrollo de cáncer de piel.

Obtención del índice UV

El sistema implementa dos métodos para obtener la ubicación:

- **Ubicación automática del usuario**, usando la API de Geolocator [40] para acceder a las coordenadas GPS, siempre que el usuario haya otorgado los permisos necesarios.

- **Búsqueda manual por ciudad**, en la que se muestran sugerencias conforme se escribe en el campo de texto, gracias a la API de geolocalización directa de OpenWeather.

Una vez obtenidas las coordenadas (latitud y longitud), se realiza una solicitud HTTP a la API de OpenWeather (<https://api.openweathermap.org/data/2.5/uv>) para recuperar el valor actual del índice UV. El valor es almacenado como un número decimal (double) y se utiliza para personalizar la interfaz visual y emitir recomendaciones al usuario.

Visualización del riesgo UV

La interfaz adapta su diseño dinámicamente al nivel de exposición UV:

- **Colores de fondo** y de la AppBar cambian según el nivel: verde (bajo), amarillo (moderado), naranja (alto), rojo (muy alto), púrpura (extremo).
- Se muestra un **ícono climático** representativo (nube, sol, sol fuerte) y un texto que indica el nivel del índice UV.
- Si no se puede acceder a la ubicación, se presenta un mensaje informativo y se permite al usuario buscar manualmente.

CITAS MÉDICAS:

Facilita la gestión de consultas con profesionales de la salud, guiando al usuario para programar una revisión médica en caso de ser necesario.

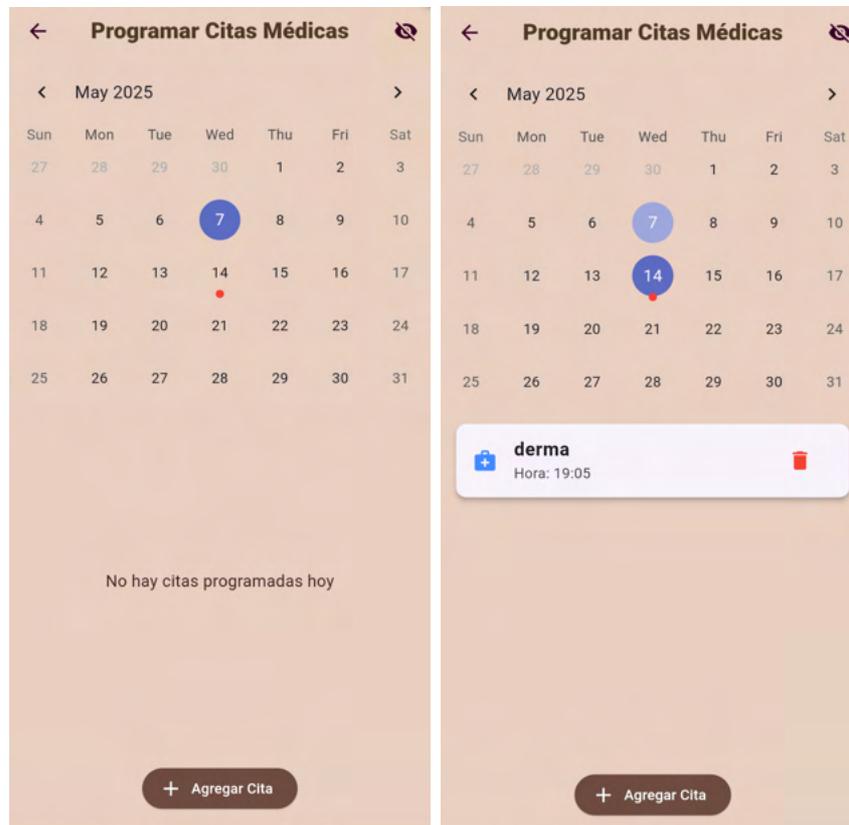


Ilustración 84. Pantalla Programar Citas Médicas

La pantalla de **Pedir Cita** permite a los usuarios autenticados registrar, consultar y gestionar sus citas médicas directamente desde la aplicación móvil. Esta funcionalidad tiene como objetivo facilitar la organización del seguimiento clínico, sirviendo como puente entre el usuario y el profesional sanitario o sistema de gestión.

Arquitectura y tecnologías utilizadas

- **Firestore Authentication** se utiliza para asegurar que las citas se asocien de forma privada e individualizada a cada usuario registrado.
- **Supabase** actúa como base de datos remota para el almacenamiento persistente de las citas médicas.
- **TableCalendar**, un paquete de Flutter, se utiliza para representar el calendario de forma interactiva.
- Las fechas se normalizan (sin hora) para facilitar la agrupación de eventos por día.

Visualización y gestión de citas

- **Calendario interactivo:**

Muestra el mes en curso y permite al usuario seleccionar un día. Los días con citas tienen un marcador visual (punto rojo). Se puede ocultar o mostrar dinámicamente mediante un botón en la AppBar.

- **Listado de citas del día seleccionado:**

Una vez elegido un día, se muestran las citas asociadas en tarjetas con iconos, hora, descripción y un botón para eliminarlas. Si no hay citas, se muestra un mensaje informativo: **Añadir nuevas citas**.

Mediante el botón “**Agregar Cita**”, el usuario puede abrir un diálogo modal en el que:

- Introduce una **descripción de la cita**.
- Selecciona una **fecha** (con showDatePicker).
- Elige una **hora** (con showTimePicker).

Una vez confirmada, la cita se guarda en la base de datos Supabase y se actualiza el estado local para que se refleje inmediatamente en el calendario.

Eliminación de citas

Cada tarjeta de cita tiene un icono de papelera que permite eliminarla. Esta acción realiza una operación de borrado tanto en Supabase como en la estructura local `_citas`, asegurando que los cambios se reflejan instantáneamente en la interfaz.

Cada uno de estos apartados refuerza la finalidad de la aplicación: brindar información útil, fomentar la detección temprana y ofrecer vías de acción concretas para el cuidado adecuado de la piel.

HISTORIAL

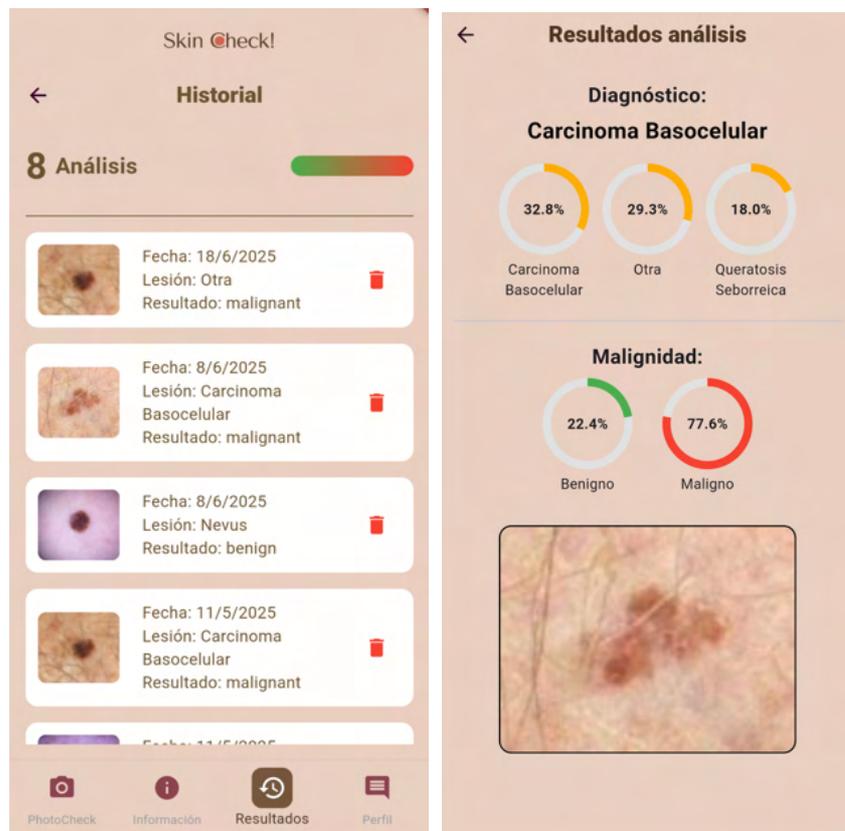


Ilustración 85. Pantalla Historial y 8un Análisis

La pantalla de Resultados muestra el historial de análisis realizados por el usuario, ofreciendo un listado detallado donde cada elemento incluye la fecha, el tipo de lesión examinada y el resultado obtenido. De esta manera, se facilita el seguimiento de la evolución de la piel a lo largo del tiempo. En la parte superior puede presentarse un indicador visual que represente el riesgo o la gravedad de la lesión de forma rápida y comprensible, pasando de verde (bajo riesgo) a rojo (alto riesgo). Cada resultado puede seleccionarse para ver

información más detallada, ayudando al usuario a tener un registro claro de todos sus análisis y a controlar mejor su salud cutánea.

PERFIL

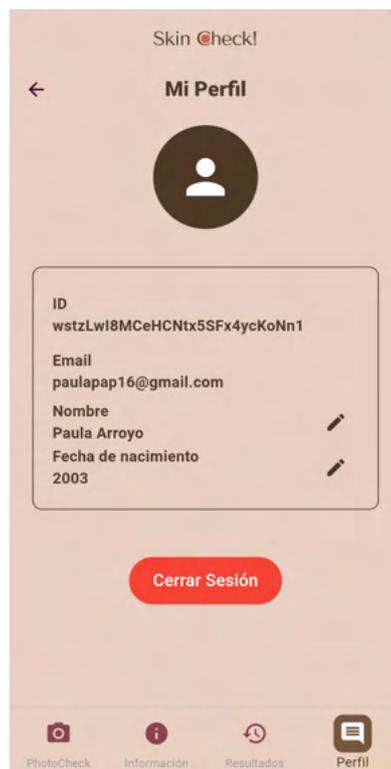


Ilustración 86. Pantalla Mi Perfil

La sección Mi Perfil centraliza la información personal del usuario, mostrando un identificador único (ID), la dirección de correo, el nombre y el año de nacimiento. Además, se incluyen iconos o botones para editar estos datos, garantizando que puedan mantenerse actualizados. En la parte inferior aparece un destacado botón de “Cerrar Sesión”, que permite al usuario salir de la cuenta de forma rápida y segura en cualquier momento. De esta forma, la pantalla de perfil asegura la gestión de la información personal de manera clara y accesible.

ANEXO IV: LISTA DE ABREVIATURAS

Abreviatura	Significado
BCC	Carcinoma Basocelular
MEL	Melanoma
NV	Nevus
OTH	Otras lesiones
SCC	Carcinoma Escamoso
SK	Queratosis Seborreica
CNN	Red Neuronal Convolutacional
F1-Score	Métrica de evaluación basada en precisión y exhaustividad
train acc	precisión en entrenamiento
val acc	precisión en validación

ANEXO IV: TÉRMINOS CLAVE Y SUS TRADUCCIONES

Término	Traducción
CNN	Red Neuronal Convolutacional
Classification report	Informe de clasificación
Accuracy	Precisión global del modelo
Recall	Sensibilidad o exhaustividad
Support	Cantidad de muestras reales presentes en cada clase
Precision	Cuántas de las predicciones positivas son correctas
Macro avg	Visión de cuántas clases están desbalanceadas
Overfitted	Sobreajustado
Underfitted	Subajustado
Oversampling	Sobremuestreo
Undersampling	Submuestreo
Dataset	Conjunto de datos
Data augmentation	Aumentación de datos
Benign	Benigno
Malignant	Maligno
Test	Prueba
Train	Entrenamiento

Loss	Pérdida
Early Stopping	Parada anticipada
Dropout	Eliminación aleatoria de nodos
Learning rate	Tasa de aprendizaje
Scheduling	Programación (del aprendizaje)
Deep learning	Aprendizaje profundo
Backend	Lógica
Frontend	Interfaz
