



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE GUAYAQUIL
CARRERA DE MECATRÓNICA

**DESARROLLO DE UNA HERRAMIENTA EDUCATIVA MÓVIL
BASADA EN INTELIGENCIA ARTIFICIAL PARA LA ENSEÑANZA
DE LA INTERPRETACIÓN DE RADIOGRAFÍAS DE TÓRAX EN EL
DIAGNÓSTICO DE NEUMONÍA**

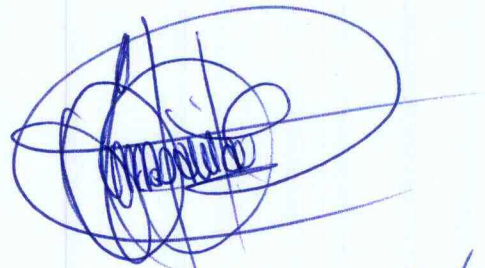

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Mecatrónica

AUTOR: Carlos Andrés Segovia Arias

TUTOR: Ing. Jonathan Salvador Paillacho Corredores M.Sc.

Guayaquil - Ecuador

2026


18/02/26


CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, **Carlos Andrés Segovia Arias** con documento de identificación N° **0956619829** manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo.

Guayaquil, 18 de febrero del año 2026

Atentamente,



Carlos Andrés Segovia Arias
0956619829

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA
UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, **Carlos Andrés Segovia Arias** con documento de identificación N° **0956619829** expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del **Dispositivo Tecnológico: DESARROLLO DE UNA HERRAMIENTA EDUCATIVA MÓVIL BASADA EN INTELIGENCIA ARTIFICIAL PARA LA ENSEÑANZA DE LA INTERPRETACIÓN DE RADIOGRAFÍAS DE TÓRAX EN EL DIAGNÓSTICO DE NEUMONÍA**, el cual ha sido desarrollado para optar por el título de: Ingeniero en Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 18 de febrero del año 2026

Atentamente,



Carlos Andrés Segovia Arias
0956619829

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, **Jonathan Salvador Paillacho Corredores**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: **DESARROLLO DE UNA HERRAMIENTA EDUCATIVA MÓVIL BASADA EN INTELIGENCIA ARTIFICIAL PARA LA ENSEÑANZA DE LA INTERPRETACIÓN DE RADIOGRAFÍAS DE TÓRAX EN EL DIAGNÓSTICO DE NEUMONÍA**, realizado por **Carlos Andrés Segovia Arias** con documento de identificación N° **0956619829** obteniendo como resultado final el trabajo de titulación bajo la opción **Dispositivo Tecnológico** que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 18 de febrero del año 2026

Atentamente,



Ing. Jonathan Salvador Paillacho Corredores M.Sc.

1718907874

DEDICATORIA

Este trabajo de titulación está dedicado a mi familia y amigos cercanos, pero principalmente a mi mamá, a mi abuelita y mi abuelito que son las personas más importantes para mí siendo las que me criaron, me enseñaron lo que son los valores, lo que es la responsabilidad, la educación y lo más importante para mí que es el cariño y amor.

Carlos Andrés Segovia Arias

AGRADECIMIENTO

Agradezco a Dios, a mi mamá y abuelita sin ellas no hubiese sido posible que llegue donde estoy.

Quiero hacerles un agradecimiento especial, todo el esfuerzo, constancia y apoyo todos estos años me han permitido seguir esforzándome y me han enseñado lo muy importantes que son para mi, sin ustedes nada de lo que soy como persona seria posible.

Agradezco también a la Universidad Politécnica Salesiana, a los docentes y a mis amigos de la universidad por formar parte importante tanto en el desarrollo y cumplimiento del presente trabajo como de todos los proyectos que cumplimos e hicimos juntos.

Agradezco además al Ing. Jonathan Salvador Paillacho por guiarme en la labor de mi tesis y por haber cumplido los requerimientos para presentar la misma.

Carlos Andrés Segovia Arias

RESUMEN

El presente proyecto tiene como objetivo el desarrollo de una herramienta educativa móvil, denominada NeumoLens, basada en inteligencia artificial para fortalecer la enseñanza en la interpretación de radiografías de tórax en el diagnóstico de neumonía. El sistema aborda la complejidad de la identificación visual de patrones radiológicos mediante el uso de redes neuronales convolucionales (CNN), proporcionando una guía diagnóstica accesible para estudiantes y personas naturales.

Para la implementación de este sistema, se empleó la arquitectura DenseNet121, la cual fue optimizada mediante técnicas de transferencia de aprendizaje y entrenamiento profundo (Fine-Tuning) para clasificar imágenes en cinco categorías: Normal, Bronconeumonía, Neumonía Lobar, Neumonía Intersticial y Ruido. El modelo fue integrado de manera nativa en una aplicación móvil desarrollada en Android Studio con el lenguaje Kotlin. Gracias al uso del motor TensorFlow Lite, la aplicación opera bajo el paradigma de Edge AI, permitiendo realizar la inferencia y el diagnóstico de forma local en el dispositivo sin necesidad de conexión a internet, garantizando así una respuesta inmediata.

Finalmente, la herramienta incorpora un módulo educativo interactivo que utiliza la técnica de aprendizaje por contraste (Active Recall) y un sistema de evaluación gamificado (Quiz). El modelo de inteligencia artificial alcanzó una exactitud del 80% en el entorno de entrenamiento y una tasa de éxito del 70% en pruebas de campo con imágenes no vistas anteriormente. Los resultados demuestran que NeumoLens no solo es un clasificador preciso, sino un soporte pedagógico efectivo que facilita la curva de aprendizaje en la detección de patologías pulmonares mediante visión artificial.

Palabras claves: Redes Neuronales Convolucionales (CNN), Rayos X, Neumonía, Aplicación Móvil, Herramienta Educativa.

ABSTRACT

The objective of this project is to develop a mobile educational tool named *NeumoLens*, based on artificial intelligence, to strengthen the teaching of chest X-ray interpretation for pneumonia diagnosis. The system addresses the complexity of visual identification of radiological patterns through the use of Convolutional Neural Networks (CNN), providing an accessible diagnostic guide for students and non-experts.

For the implementation of this system, the DenseNet121 architecture was employed and optimized using Transfer Learning and Fine-Tuning techniques to classify images into five categories: Normal, Bronchopneumonia, Lobar Pneumonia, Interstitial Pneumonia, and Noise. The model was natively integrated into a mobile application developed in Android Studio using the Kotlin language. Through the use of the TensorFlow Lite engine, the application operates under the Edge AI paradigm, allowing local inference and diagnosis on the device without the need for an internet connection, thus ensuring an immediate response and data privacy.

Finally, the tool incorporates an interactive educational module that utilizes contrastive learning (Active Recall) and a gamified evaluation system (Quiz). The artificial intelligence model achieved an accuracy of 80% in the training environment and a 70% success rate in field tests with previously unseen images. The results demonstrate that *NeumoLens* is not only an accurate classifier but also an effective pedagogical support that facilitates the learning curve in the detection of lung pathologies through computer vision.

Keywords: Convolutional Neural Networks (CNN), X-ray, Pneumonia, Mobile Application, Educational Tool.

ÍNDICE

| | | |
|-------------|--|----|
| I. | Introducción | 1 |
| II. | Problema | 2 |
| III. | Justificación | 3 |
| IV. | Objetivos | 4 |
| IV-A. | Objetivo general | 4 |
| IV-B. | Objetivos específicos | 4 |
| V. | Marco Teórico | 5 |
| V-A. | Introducción y epidemiología | 5 |
| V-A1. | Las Cinco Densidades Radiológicas Básicas | 5 |
| V-B. | Principios Físicos de la Radiografía de Tórax | 9 |
| V-C. | Tipología clínica, abordaje terapéutico y relación con el diagnóstico por imagen | 9 |
| V-D. | Radiografía de tórax: papel clínico, limitaciones y variabilidad | 10 |
| V-E. | Conjuntos de datos públicos y su papel en investigación | 10 |
| V-F. | Fundamentos Matemáticos de las Redes Neuronales Convolucionales | 10 |
| V-F1. | Operación de Convolución | 10 |
| V-F2. | Capa de Agrupamiento (Pooling) | 10 |
| V-F3. | Funciones de Activación: ReLU | 11 |
| V-G. | Procesamiento Avanzado de Imágenes: Mejora de Contraste Mediante CLAHE | 11 |
| V-G1. | Limitaciones de la Ecuación Global | 11 |
| V-G2. | Formulación Matemática de CLAHE | 11 |
| V-G3. | Normalización de Tensores | 11 |
| V-H. | Redes neuronales convolucionales (CNN): fundamentos y arquitectura | 11 |
| V-I. | Keras | 12 |
| V-J. | Arquitectura DenseNet121 y Conectividad Densa | 12 |
| V-J1. | El Problema del Desvanecimiento del Gradiente | 12 |
| V-J2. | Bloques Densos y Reutilización de Características | 12 |
| V-K. | Transfer Learning y Fine-Tuning en Radiología | 13 |
| V-K1. | Pesos de ImageNet y Transferencia de Conocimiento | 13 |
| V-K2. | Estrategia de Fine-Tuning | 13 |
| V-L. | Entrenamiento, etiquetas y manejo de incertidumbre | 13 |
| V-M. | Evaluación clínica y estudios prospectivos | 13 |
| V-N. | Aplicaciones móviles y Python | 13 |
| V-Ñ. | Kotlin en el desarrollo de aplicaciones móviles | 14 |
| V-O. | Fundamentos Matemáticos de las Redes Neuronales Convolucionales | 14 |
| V-O1. | Operación de Convolución | 14 |
| V-O2. | Capa de Agrupamiento (Pooling) | 14 |
| V-O3. | Funciones de Activación: ReLU | 15 |
| V-P. | Android Studio: Entorno de Desarrollo Integrado (IDE) | 15 |
| V-P1. | Arquitectura y Herramientas de Compilación | 15 |
| V-P2. | Interfaz de Usuario y Material Design | 15 |
| V-Q. | Edge AI: Procesamiento de IA en el Dispositivo Móvil | 15 |
| V-Q1. | TensorFlow Lite y el Intérprete de Inferencia | 15 |
| V-R. | Principios de Psicología Cognitiva Aplicados al Software Educativo | 16 |
| V-R1. | Teoría de la Carga Cognitiva | 16 |
| V-R2. | Aprendizaje por Contraste y Active Recall | 16 |

| | | |
|--------------|--|-----------|
| V-R3. | Estrategias de Gamificación | 16 |
| V-R4. | Teoría del Aprendizaje Multimedia de Richard Mayer | 16 |
| V-R5. | Teoría de la Codificación Dual (Paivio) | 16 |
| V-R6. | Andamiaje Cognitivo para Estudiantes y Personas Naturales | 17 |
| V-R7. | Metacognición y Monitoreo del Aprendizaje | 17 |
| V-R8. | Adaptabilidad de la Interfaz: El Rol de la Persona Natural | 17 |
| VI. | Marco Metodológico | 18 |
| VI-A. | Diseño de la Investigación | 18 |
| VI-A1. | Nivel Aplicado | 18 |
| VI-A2. | Diseño Experimental | 18 |
| VI-B. | Población y Muestra | 18 |
| VI-C. | Instrumentación y Entorno Computacional | 19 |
| VI-C1. | Hardware | 19 |
| VI-C2. | Software y Librerías | 19 |
| VI-D. | Fase 1: Preprocesamiento y Modelado Matemático de la Entrada | 20 |
| VI-D1. | Estandarización Dimensional | 20 |
| VI-D2. | Normalización de Intensidad | 20 |
| VI-D3. | Aumentación de Datos (Data Augmentation) | 20 |
| VI-E. | Fase 2: Arquitectura de la Red Neuronal (DenseNet121) | 20 |
| VI-E1. | Principio de Conectividad Densa | 20 |
| VI-E2. | Adaptación de la Arquitectura (Top Model) | 21 |
| VI-F. | Fase 3: Configuración del Entrenamiento y Optimización | 21 |
| VI-F1. | Función de Pérdida (Loss Function) | 21 |
| VI-F2. | Algoritmo de Optimización | 21 |
| VI-F3. | Callbacks y Control de Entrenamiento | 21 |
| VI-G. | Métricas de Evaluación | 22 |
| VI-H. | Fase 3: Optimización para el Borde (Edge AI) | 22 |
| VI-H1. | Conversión y Serialización a TensorFlow Lite | 22 |
| VI-H2. | Benchmarking de Latencia | 22 |
| VI-I. | Fase 1: Ingeniería de Software en Android Studio (Kotlin) | 23 |
| VI-I1. | Arquitectura Modular y Gestión de Recursos | 23 |
| VI-I2. | Pipeline de Procesamiento en Dispositivo | 23 |
| VI-J. | Fase 2: Diseño del Módulo Educativo y Lógica de Gamificación | 23 |
| VI-J1. | Algoritmo de Aprendizaje por Contraste (Active Recall) | 23 |
| VI-J2. | Lógica de Aleatoriedad en el Quiz (Fisher-Yates) | 23 |
| VI-J3. | Sistema de Retroalimentación Inmediata | 23 |
| VI-K. | Diseño y Arquitectura del Front-end (Interfaz de Usuario) | 23 |
| VI-K1. | Psicología del Color y Aspectos Visuales | 24 |
| VI-K2. | Estructura Funcional: Los Cuatro Pilares del Sistema | 24 |
| VI-K3. | Componentes Específicos y Retroalimentación | 24 |
| VI-L. | Manual de Usuario Técnico (Procedimiento de Operación) | 26 |
| VI-L1. | Fase 1: Acceso e Inicialización | 26 |
| VI-L2. | Fase 2: Interacción Educativa y Evaluación | 27 |
| VI-L3. | Fase 3: Ejecución del Diagnóstico | 28 |
| VII. | Resultados | 29 |
| VII-1. | Validación de Inferencia en Tiempo Real (Prueba de Campo) | 31 |
| VII-A. | Evaluación de Usabilidad y Experiencia de Usuario (UX) | 34 |
| VIII. | Cronograma | 38 |

| | |
|---|----|
| IX. Presupuesto | 39 |
| X. Conclusiones | 40 |
| XI. Recomendaciones | 41 |
| Referencias | 42 |
| Anexo A: Programación de la Red Neuronal | 46 |
| Anexo B: Código de la Aplicación Móvil NeumoLens | 50 |
| Anexo C: Imágenes de la App | 71 |

ÍNDICE DE FIGURAS

| | | |
|-----|--|----|
| 1. | Patrón Radiológico Estándar de una persona sana [23]. | 5 |
| 2. | Patrón de un paciente que presenta Lipoidea (exceso de grasa)[24]. | 6 |
| 3. | Patrón de una persona que presenta síntomas de edema pulmonar [25]. | 6 |
| 4. | Patrón radiológico que presenta una persona con un agente extraño en el tórax (metal)[26]. | 6 |
| 5. | Condensación lobar con broncograma aéreo en un paciente con neumonía adquirida por el Streptococcus pneumoniae [30]. | 7 |
| 6. | Patrón bronconeumónico en un paciente que desarrolló una neumonía debido a Staphylococcus aureus [30]. | 8 |
| 7. | Patrón intersticial en un paciente con neumonía adquirida en la comunidad debido al virus de la gripe influenza [30]. | 8 |
| 8. | Como funciona una red neuronal convolucional [42]. | 12 |
| 9. | Desarrollo de apps con Python [47]. | 14 |
| 10. | Dataset con las imagenes seleccionadas para la etiqueta de train | 18 |
| 11. | Librerías empleadas en google colab | 19 |
| 12. | Recuadro de programación por imagen en Android Studio (.xml) | 25 |
| 13. | Recuadro de programación por imagen en Android Studio 2 (.xml) | 25 |
| 14. | Código DetailActivity.kt | 26 |
| 15. | Menú Principal | 26 |
| 16. | Resultado perfecto (Quiz) | 27 |
| 17. | Prueba la red (Cnn) | 28 |
| 18. | Reporte de la precisión luego del entrenamiento | 29 |
| 19. | Matriz de confusión donde se pueden apreciar los falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos | 30 |
| 20. | Gráfica de comparación entre el entrenamiento y la validación | 30 |
| 21. | Nivel de satisfacción y claridad en el uso de la interfaz y el contenido educativo. | 34 |
| 22. | Calificación promedio obtenida por los usuarios en el módulo evaluativo. | 35 |
| 23. | Porcentaje de incremento en la efectividad diagnóstica tras la revisión del Modulo Educativo. | 35 |
| 24. | Prueba de desempeño de al app en estudiantes de Ing. | 36 |
| 25. | Librerías - Configuración de escala, épocas y clases. | 46 |
| 26. | Configuración del Data Augmentation. | 47 |
| 27. | Código Arquitectura DenseNet. | 48 |
| 28. | Configuración de Callbacks y Entrenamiento. | 48 |
| 29. | Código para la Gráfica de Precisión y la de Perdida. | 49 |
| 30. | Código Creación de la Matriz de Confusión y Guardado del mejor entrenamiento en formato TF Lite. | 49 |
| 31. | Menú Principal | 71 |
| 32. | Biblioteca de Patrones | 72 |
| 33. | 1er Patrón Radiológico | 73 |
| 34. | 2do Patrón Radiológico | 74 |
| 35. | 3er Patrón Radiológico | 75 |
| 36. | Preguntas al Usuario (Quiz) | 76 |
| 37. | Mal Resultado | 77 |
| 38. | Buen Resultado | 78 |
| 39. | Apartado para Probar la Red Neuronal | 79 |
| 40. | Resultado de Ruido/Imagen Invalida | 80 |
| 41. | Resultado de unos Pulmones Sanos | 81 |
| 42. | Resultado y Detección de un caso en una Radiografía de Torax | 82 |

ÍNDICE DE TABLAS

| | | |
|-------|---|----|
| I. | Resultados de validación práctica de la CNN (global). | 31 |
| II. | Resultados de validación para la clase: Ruido / Imagen Inválida. | 31 |
| III. | Resultados de validación para la clase: Normal. | 32 |
| IV. | Resultados de validación para la clase: Neumonía Lobar. | 32 |
| V. | Resultados de validación para la clase: Bronconeumonía. | 33 |
| VI. | Resultados de validación para la clase: Neumonía Intersticial. | 33 |
| VII. | Resultados de la Evaluación de Usabilidad y Experiencia de Usuario (n=10) | 34 |
| VIII. | Métricas de rendimiento técnico de la aplicación NeumoLens. | 37 |
| IX. | Cronograma | 38 |

I. INTRODUCCIÓN

El presente estudio tiene como fin el desarrollo de una herramienta educativa móvil denominada "NeumoLens", basada en inteligencia artificial, diseñada para asistir en la enseñanza de la interpretación de radiografías de tórax. El sistema permite clasificar imágenes en cinco categorías: pulmón normal, neumonía lobar, bronconeumonía, neumonía intersticial y una categoría de filtrado de ruido (imágenes inválidas). Esto se logra mediante el análisis automático de patrones visuales, proporcionando un diagnóstico predictivo que sirve como apoyo pedagógico.

Para el funcionamiento de la herramienta, se implementa una Red Neuronal Convolutiva (CNN) diseñada específicamente para el procesamiento de imágenes médicas. El modelo ha sido optimizado mediante un proceso de cuantización de parámetros, lo que permite reducir su peso computacional y facilitar la inferencia en tiempo real dentro de un dispositivo móvil sin sacrificar significativamente la precisión. Este mecanismo actúa como el núcleo de procesamiento, permitiendo que la aplicación responda de manera ágil a la captura o carga de nuevas radiografías por parte del usuario.

Además de la arquitectura de aprendizaje profundo, se utiliza el framework TensorFlow Lite para la portabilidad del modelo y el entorno de desarrollo Android Studio para la interfaz de usuario. La recolección de datos y el entrenamiento del modelo se llevaron a cabo utilizando bibliotecas de Python y Keras, validando la integridad del sistema mediante pruebas de campo y muestras controladas. Estos componentes, integrados en una plataforma móvil, permiten que el estudio ofrezca una solución accesible y eficiente para el aprendizaje de personas naturales y estudiantes acerca de cómo llegar a un diagnóstico de neumonía.

II. PROBLEMA

Cada año, millones de personas en todo el mundo sufren neumonía, de las cuales unos 4 millones fallecen como consecuencia directa de la infección, especialmente en países de ingresos bajos y medianos. Solo en niños menores de cinco años, la neumonía causa cerca de 500.000 defunciones anuales [1]. La neumonía es una inflamación de los alvéolos pulmonares generalmente causada por agentes infecciosos que puede agravarse cuando existen comorbilidades como VIH, diabetes o desnutrición [2].

El diagnóstico de neumonía mediante radiografías de tórax enfrenta múltiples barreras a pesar de los avances tecnológicos. La escasez de radiólogos en regiones rurales o de bajos recursos impide una atención médica oportuna [3]. Además, la variabilidad interobservador puede alcanzar el 20 % en casos leves, lo que significa que en uno de cada cinco casos, dos especialistas podrían no coincidir en el diagnóstico, afectando así la consistencia diagnóstica. [4].

La carga de trabajo elevada ha sido ligada a un aumento de errores de interpretación, especialmente en turnos prolongados o bajo presión [5]. En paralelo, resulta cada vez más crítico diferenciar entre neumonías virales, bacterianas o asociadas a COVID-19, un reto que la radiografía convencional no siempre resuelve con certeza [6].

La transparencia limitada de los sistemas de inteligencia artificial, frecuentemente descritos como “cajas negras”, genera desconfianza en el entorno clínico y limita su adopción [7, 8]. Además, muchos modelos presentan un rendimiento inferior cuando se trasladan de entornos controlados a escenarios clínicos reales [9].

Factores como conjuntos de datos desbalanceados, la necesidad de hardware especializado o la falta de conectividad estable entorpecen la implementación práctica en zonas vulnerables [10]. Por lo tanto, resulta indispensable desarrollar soluciones accesibles, explicables y escalables que puedan operar en dispositivos móviles [11].

Adicionalmente, estudios recientes han documentado que los sistemas entrenados con grandes volúmenes de datos pueden contener sesgos que afectan negativamente a ciertos grupos, como niños o pacientes con patologías respiratorias atípicas [12, 13]. El uso de herramientas no validadas puede inducir diagnósticos incorrectos o generar una sobreconfianza peligrosa [14].

Finalmente, la cultura regulatoria y ética en muchos países de bajos y medianos ingresos es insuficiente, lo que dificulta la adopción segura y responsable de tecnologías diagnósticas basadas en inteligencia artificial [15]. Todo esto subraya la necesidad de construir modelos diagnósticos responsables, auditables y adaptados a la diversidad de contextos locales.

III. JUSTIFICACIÓN

El uso de herramientas tecnológicas en la educación médica ha cobrado una relevancia creciente en los últimos años, especialmente en el área del diagnóstico por imágenes. Las enfermedades respiratorias, como la neumonía, representan uno de los principales desafíos clínicos a nivel mundial, tanto en la población infantil como en adultos mayores. La neumonía continúa siendo una de las principales causas de mortalidad a nivel global, con millones de casos reportados cada año [16, 17].

En este contexto, la formación académica de los futuros profesionales de la salud requiere recursos didácticos que faciliten la comprensión de los principios del diagnóstico radiológico y el desarrollo de competencias analíticas para la interpretación de imágenes médicas. En respuesta a esta necesidad formativa, el desarrollo de herramientas educativas basadas en visión artificial ofrece una oportunidad valiosa para complementar la enseñanza tradicional.

Las redes neuronales convolucionales (CNN) han demostrado un alto potencial en la detección automática de patologías en imágenes de tórax, alcanzando precisiones superiores al 95 % en estudios controlados [18, 19]. A través de estas herramientas, los estudiantes pueden observar cómo un modelo de aprendizaje profundo identifica regiones de interés dentro de una radiografía, comprendiendo de manera visual los fundamentos del análisis automatizado y la toma de decisiones basadas en patrones.

Desde el ámbito educativo, la integración de la inteligencia artificial en la enseñanza médica ha sido analizada como una vía efectiva para reforzar la competencia diagnóstica y la alfabetización tecnológica de los estudiantes de medicina [20].

Esta evidencia respalda la idea de que una aplicación móvil demostrativa puede ser un complemento pedagógico relevante, permitiendo la interacción directa con modelos de IA y fomentando el aprendizaje activo.

De esta manera, la integración de esta herramienta en la educación universitaria fortalece la alfabetización digital en salud, fomenta el interés por la inteligencia artificial aplicada a la medicina y promueve la formación de profesionales capaces de evaluar, de forma crítica y ética, los resultados de sistemas inteligentes. Así, la aplicación móvil demostrativa se convierte en un apoyo pedagógico accesible que facilita el aprendizaje activo, la autocrítica diagnóstica y la comprensión de los principios que rigen los algoritmos de clasificación de imágenes.

IV. OBJETIVOS

IV-A. Objetivo general

Desarrollar una aplicación móvil educativa que integre redes neuronales convolucionales entrenadas para asistir en la interpretación de radiografías de tórax con neumonía, evaluando su efectividad pedagógica.

IV-B. Objetivos específicos

- Clasificar un conjunto de imágenes de radiografías de tórax según el patrón radiológico de la neumonía: lobar, bronconeumónica y intersticial, identificando en cada caso sus principales características visuales.
- Desarrollar un algoritmo basado en redes neuronales convolucionales para la identificación de neumonía y sus casos a partir de radiografías de torax.
- Evaluar la experiencia educativa de la aplicación móvil mediante pruebas de usabilidad en una muestra de estudiantes de ingeniería.

V. MARCO TEÓRICO

V-A. Introducción y epidemiología

La neumonía es una infección del parénquima pulmonar, los cuales son los encargados de realizar el intercambio gaseoso en el organismo, eliminando dióxido de carbono y permitiendo el paso de oxígeno hacia la sangre. Estas estructuras están conformadas por los alvéolos, pequeñas unidades en forma de saco donde ocurre dicho intercambio. Cuando se produce la infección, los alvéolos se inflaman y se llenan de líquido o material inflamatorio, dificultando el paso normal del oxígeno a la sangre. Clínicamente, se manifiesta con tos, fiebre, disnea (falta de aire), dolor torácico pleurítico y, en casos graves, hipoxemia.

La neumonía continúa siendo una de las principales causas de morbilidad y mortalidad en el mundo, con mayor impacto en niños menores de cinco años y en poblaciones de países de ingresos bajos y medianos. Las estimaciones globales señalan decenas a cientos de millones de episodios anuales y cientos de miles de muertes infantiles por neumonía cada año; la carga varía según factores sociodemográficos, acceso al sistema sanitario, tasas de vacunación y exposición a factores de riesgo como la contaminación del aire y la malnutrición [21]. La heterogeneidad epidemiológica implica que las estrategias locales de diagnóstico y tratamiento deben adaptarse a las condiciones de cada región. En contextos rurales con déficit de radiólogos y equipos, el diagnóstico por imagen está limitado, lo que es crucial el desarrollo de herramientas de apoyo accesibles, de bajo costo y desplegadas para mejorar el entendimiento de las personas a la afección que pueden padecer y a una detección temprana [22].

V-A1. Las Cinco Densidades Radiológicas Básicas: Para comprender mejor como se lleva a cabo un diagnóstico por imagen en neumonía hay que tener claro las variaciones en las densidades radiológicas. Tradicionalmente, se describen cinco niveles de atenuación que permiten diferenciar las estructuras anatómicas para llevar a cabo un diagnóstico:

1. Aire (Radiolúcido): Es la densidad más baja. Debido a su baja masa atómica, los rayos X apenas son atenuados, lo que resulta en un tono negro en el receptor. En un tórax normal, los pulmones presentan esta densidad.
2. Calcio / Hueso (Radiopaco): Debido a su alto contenido de calcio ($Z = 20$), absorbe una gran proporción de fotones, proyectándose en color blanco.



Figura 1. Patrón Radiológico Estándar de una persona sana [23].

3. Grasa: Presenta una atenuación ligeramente superior al aire. Se observa en los planos faciales y el tejido subcutáneo, manifestándose como un tono gris oscuro.

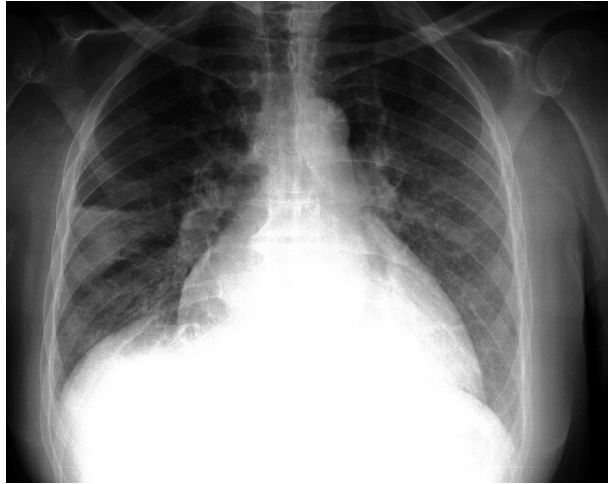


Figura 2. Patrón de un paciente que presenta Lipoidea (exceso de grasa)[24].

4. Agua / Tejidos Blandos: Esta densidad es compartida por el corazón, los vasos sanguíneos, los músculos y, patológicamente, por el exudado inflamatorio de la neumonía. Se observa como un tono gris claro.

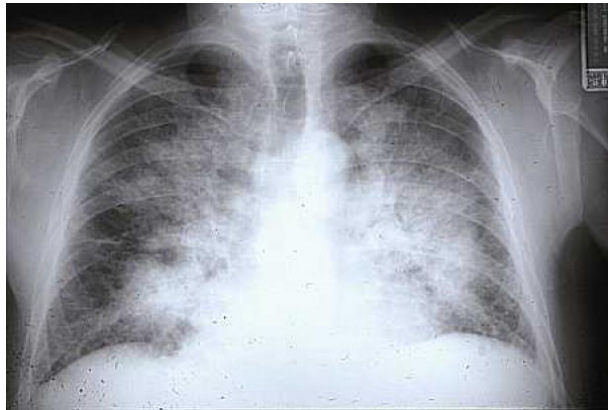


Figura 3. Patrón de una persona que presenta síntomas de edema pulmonar [25].

5. Metal: Es la densidad más alta, observada en marcapasos, clips quirúrgicos o cuerpos extraños, proyectándose como un blanco absoluto y brillante.

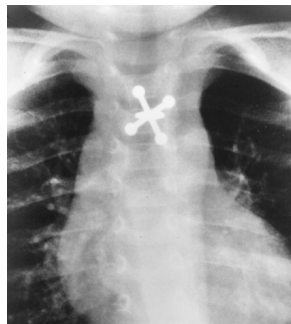


Figura 4. Patrón radiológico que presenta una persona con un agente extraño en el tórax (metal)[26].

La enfermedad comprende un espectro de entidades clínicas clasificadas según el lugar de adquisición, el agente causal y el patrón radiológico. Desde el punto de vista clínico-epidemiológico, se distinguen tres grandes grupos:

1. Neumonía adquirida en la comunidad (NAC o CAP): adquirida fuera del ámbito hospitalario, con agentes frecuentes como *Streptococcus pneumoniae*, *Haemophilus influenzae* y diversos virus respiratorios. Su manejo y pronóstico dependen de la edad, comorbilidades y gravedad del cuadro [27].
2. Neumonía asociada a los cuidados de la salud / nosocomial (HAP): se presenta al menos 48 horas después del ingreso hospitalario y suele involucrar patógenos resistentes, por lo que requiere esquemas terapéuticos de mayor espectro [28].
3. Neumonía asociada a ventilación mecánica (VAP): ocurre en pacientes intubados tras más de 48 horas, con alta frecuencia de bacterias oportunistas como *Pseudomonas aeruginosa* o *Acinetobacter spp.*

Según el agente causal, pueden distinguirse varios tipos de neumonía entre ellas están la Bacterianas (típicas y atípicas), Virales o Micóticas (estas últimas son más frecuentes en pacientes inmunodeprimidos o en zonas endémicas). En cuanto al patrón morfológico observado por imagen, se utiliza la siguiente clasificación orientativa [29]:

- Lobar (consolidación lobar): suele ocupar un lóbulo completo y está asociada principalmente a bacterias clásicas como *Streptococcus pneumoniae*.

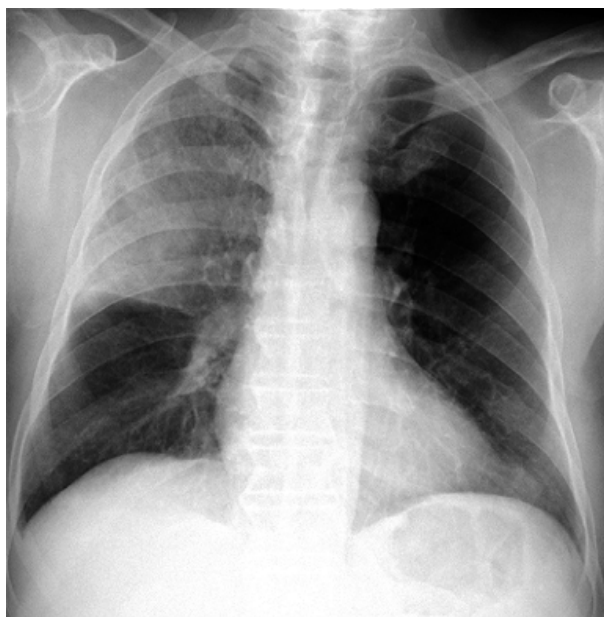


Figura 5. Condensación lobar con broncograma aéreo en un paciente con neumonía adquirida por el *Streptococcus pneumoniae* [30].

- Bronconeumonía (centrolobulillar o bronquiolar): producto de múltiples focos parcheados que pueden confluir, típica de infecciones por aspiración o con múltiples focos bacterianos.



Figura 6. Patrón bronconeumónico en un paciente que desarrolló una neumonía debido a *Staphylococcus aureus* [30].

- Intersticial (intersticial): patrón reticular o en vidrio esmerilado, más frecuente en infecciones virales o atípicas.

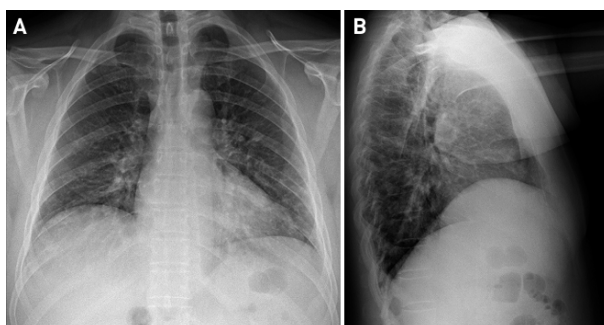


Figura 7. Patrón intersticial en un paciente con neumonía adquirida en la comunidad debido al virus de la gripe influenza [30].

Estos patrones ayudan a orientar la sospecha etiológica, aunque no son determinantes por sí mismos. En la práctica clínica, la diferenciación entre los tipos se sustenta en una combinación de hallazgos clínicos, de laboratorio e imagenológicos. La neumonía bacteriana típica como la lobar ilustrada en la figura 5 suele tener un inicio agudo, fiebre alta, esputo purulento y dolor pleurítico; las formas virales o atípicas tienden a comenzar de manera más insidiosa con mialgias, cefalea y tos seca. Retomando lo hablado en la anterior cita en el examen físico suelen observarse crepitantes y disminución de ruidos respiratorios en el área afectada, con hipoxemia si el compromiso pulmonar es extenso. Los estudios de laboratorio incluyen hemograma (leucocitosis en bacterianas), PCR o procalcitonina (para orientar etiología bacteriana), y cultivos o pruebas moleculares específicas. La radiografía de tórax (CXR) constituye la prueba inicial para confirmar la sospecha, valorar la extensión y el patrón; la tomografía computarizada (TC) puede utilizarse en casos complicados o dudosos.

El tratamiento se basa en la iniciación temprana de terapia empírica ajustada al tipo probable de neumonía (lugar de adquisición, agente sospechado, gravedad) y en el soporte clínico (oxigenoterapia, hidratación, control de fiebre). En la NAC leve puede emplearse tratamiento ambulatorio (por ejemplo, amoxicilina \pm macrólido), mientras que en las neumonías moderadas, severas o nosocomiales se requiere hospitalización y tratamiento de amplio espectro

[31]. La identificación etiológica permite optimizar la terapia, reducir resistencias antimicrobianas y mejorar los resultados clínicos.

Finalmente, la clasificación clínica y etiológica de la neumonía condiciona la apariencia radiológica observada. Por ejemplo, una consolidación lobar densa es característica de neumonía bacteriana, mientras que un patrón intersticial sugiere etiología viral o atípica. Por ello, la radiografía de tórax no solo confirma la sospecha clínica, sino que orienta la probabilidad etiológica inicial y las decisiones terapéuticas empíricas. En entornos con escasez de especialistas, el desarrollo de sistemas automáticos basados en inteligencia artificial, integrables en dispositivos móviles, representa una herramienta valiosa para el aprendizaje y la detección temprana de casos [32].

V-B. Principios Físicos de la Radiografía de Tórax

La radiografía de tórax se fundamenta en la interacción de los fotones de rayos X con los tejidos del cuerpo humano, un proceso gobernado por las leyes de la física de radiaciones. La formación de la imagen depende intrínsecamente del fenómeno de atenuación, que se define como la reducción de la intensidad del haz de rayos X a medida que atraviesa un objeto. Esta atenuación es el resultado de dos procesos físicos principales: el efecto fotoeléctrico y la dispersión de Compton.

Matemáticamente, la intensidad del haz de radiación tras atravesar un tejido de espesor x se describe mediante la Ley de Beer-Lambert:

$$I = I_0 e^{-\mu x} \quad (1)$$

Donde I_0 es la intensidad del haz incidente y μ representa el coeficiente de atenuación lineal, el cual varía significativamente según el número atómico efectivo (Z) y la densidad electrónica del tejido.

La radiografía de tórax conserva un papel central: no solo confirma la sospecha y valora la extensión del compromiso pulmonar, sino que los patrones radiológicos (lobar vs intersticial vs bronconeumónico) orientan la etiología probable y ayudan a monitorizar la respuesta al tratamiento. Este vínculo refuerza la necesidad de incentivar a la creación de herramientas de enseñanza e interpretación para entornos con escasez de especialistas, habilitando un entorno más rápido y eficiente para aprender.

V-C. Tipología clínica, abordaje terapéutico y relación con el diagnóstico por imagen

La neumonía comprende un espectro de entidades clínicas clasificadas según el contexto de adquisición, el agente causal y la respuesta del huésped. Desde el punto de vista clínico-epidemiológico, se distingue entre neumonía adquirida en la comunidad (NAC), neumonía asociada a cuidados de salud (NACS) y neumonía asociada a ventilación mecánica (NAVM), cada una con perfiles etiológicos y pronósticos diferenciados [33], [34]. Las NAC suelen ser de origen bacteriano, por ejemplo: *Streptococcus pneumoniae*, *Haemophilus influenzae*, *Mycoplasma pneumoniae*, mientras que las de entorno hospitalario implican con frecuencia patógenos multirresistentes como *Pseudomonas aeruginosa* o *Klebsiella pneumoniae*. En poblaciones inmunocomprometidas, los virus respiratorios y los hongos adquieren especial relevancia diagnóstica por su alta tasa de infección. [35].

El tratamiento terapéutico depende de la gravedad clínica y del agente sospechado. El tratamiento empírico inicial con antibióticos de amplio espectro se ajusta posteriormente según los cultivos o paneles moleculares, complementándose con soporte respiratorio, control de la oxigenación, hidratación adecuada y en casos severos

ventilación mecánica. La implementación de guías clínicas estandarizadas ha mejorado los desenlaces al optimizar la selección antibiótica y reducir el uso innecesario de antimicrobianos.

V-D. Radiografía de tórax: papel clínico, limitaciones y variabilidad

La radiografía de tórax (CXR) es la modalidad de imagen de elección para la evaluación inicial de sospecha de neumonía por su balance entre disponibilidad, costo y utilidad diagnóstica. Sin embargo, su sensibilidad y especificidad varían según el estadio de la enfermedad, la calidad de la imagen y la experiencia del lector. Estudios han documentado discrepancias interobservador notables; incluso radiólogos experimentados pueden tener desacuerdos clínicamente relevantes en la interpretación de infiltrados sutiles [36]. En resumen, la RX es indispensable pero imperfecta: las limitaciones prácticas (p. ej. vistas únicas, proyecciones mal posicionadas, artefactos) y clínicas (comorbilidades que modifican la apariencia radiológica) justifican la investigación en herramientas computacionales que aumenten consistencia y sensibilidad diagnóstica en escenarios reales [37].

V-E. Conjuntos de datos públicos y su papel en investigación

El progreso reciente en DL para CXR se ha apoyado fuertemente en grandes datasets públicos como ChestX-ray14, CheXpert y MIMIC-CXR, que han permitido entrenar y comparar modelos a gran escala. ChestX-ray14 (NIH) y CheXpert (Stanford) proporcionan cientos de miles de radiografías etiquetadas mediante procesamiento de lenguaje natural sobre reportes, mientras que MIMIC-CXR ofrece imágenes con informes clínicos asociados y amplio acceso para investigación reproducible. Estas colecciones han sido fundamentales para replicabilidad y avance, aunque sus etiquetas automáticas y sesgos demográficos requieren evaluación crítica. Puntos clave sobre datasets: etiquetas extraídas por NLP pueden introducir ruido; prevalencias de patologías en datasets públicos no siempre reflejan poblaciones locales; y la falta de metadatos clínicos (equipos, protocolos) dificulta la estimación del domain shift. Por todo esto, la validación externa (multicéntrica) es esencial [38].

V-F. Fundamentos Matemáticos de las Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (CNN) son arquitecturas diseñadas para procesar datos con topología de rejilla, como las imágenes de rayos X. Su potencia radica en la capacidad de extraer características jerárquicas mediante tres operaciones fundamentales:

V-F1. Operación de Convolución: La convolución es un operador matemático que actúa sobre dos funciones para producir una tercera. En el contexto de imágenes, se aplica un filtro o kernel (K) sobre la imagen de entrada (I) mediante la fórmula:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (2)$$

Esta operación permite la detección de características locales como bordes, texturas y patrones de consolidación, manteniendo la invarianza espacial.

V-F2. Capa de Agrupamiento (Pooling): El *Pooling* reduce la dimensionalidad espacial de los mapas de características, disminuyendo el número de parámetros y el costo computacional. El método más común es el Max Pooling, que selecciona el valor máximo en una región determinada (ej. 2×2), preservando las características más prominentes y otorgando robustez frente a pequeñas traslaciones de la patología en la imagen.

V-F3. *Funciones de Activación: ReLU*: La función de Unidad Lineal Rectificada (ReLU) introduce no-linealidad en el modelo, permitiéndole aprender funciones complejas. Se define como:

$$f(x) = \text{máx}(0, x) \quad (3)$$

ReLU es preferida sobre funciones sigmoides debido a que mitiga el problema del desvanecimiento del gradiente y acelera el entrenamiento al ser computacionalmente eficiente.

V-G. *Procesamiento Avanzado de Imágenes: Mejora de Contraste Mediante CLAHE*

En el ámbito de la radiología digital, el preprocesamiento es una etapa crítica para mitigar la variabilidad técnica en la adquisición de imágenes. Una técnica fundamental implementada en este proyecto es la Ecuilización de Histograma Adaptativa Limitada por Contraste (CLAHE, por sus siglas en inglés).

V-G1. *Limitaciones de la Ecuilización Global*: A diferencia de la ecuilización de histograma estándar, que aplica una transformación única a toda la imagen, CLAHE opera en pequeñas regiones denominadas "mosaicos." tiles. La ecuilización global suele sobreamplicar el ruido en áreas homogéneas de la radiografía y perder detalles en zonas con densidades extremas.

V-G2. *Formulación Matemática de CLAHE*: El algoritmo divide la imagen en una rejilla rectangular. Para cada región, se calcula el histograma y se aplica un límite de corte (*clip limit*) para evitar la saturación del contraste. Si el histograma de una región excede dicho límite, los píxeles excedentes se redistribuyen uniformemente entre todos los niveles de gris. La función de mapeo se basa en la distribución acumulativa (CDF):

$$g_{i,j} = \left[(L - 1) \sum_{n=0}^k p_{i,j}(n) \right] \quad (4)$$

Donde L es el número de niveles de gris y $p(n)$ es la probabilidad de ocurrencia del nivel n . Para evitar artefactos visuales en las fronteras de los mosaicos, se utiliza una interpolación bilineal que suaviza la transición entre regiones adyacentes [39]. Esta técnica es vital para que la red neuronal identifique infiltrados alodonosos sutiles que podrían quedar ocultos por una baja exposición radiográfica.

V-G3. *Normalización de Tensores*: Una vez mejorado el contraste, los valores de intensidad deben normalizarse para asegurar la estabilidad del gradiente durante el entrenamiento. Se emplea el escalado Min-Max para transformar el rango $[0, 255]$ al intervalo flotante $[0, 1]$:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (5)$$

Este proceso elimina las disparidades en las magnitudes de los datos de entrada, permitiendo que la arquitectura DenseNet121 converja con mayor rapidez.

V-H. *Redes neuronales convolucionales (CNN): fundamentos y arquitectura*

Las CNN implementan filtros convolucionales que extraen características locales y jerárquicas de imágenes (bordes, texturas, formas), seguidas de capas de pooling y capas totalmente conectadas para la toma de decisión. Arquitecturas modernas (ResNet, DenseNet, EfficientNet) mejoran la propagación del gradiente y la eficiencia representacional, permitiendo redes profundas con mejor generalización. Revisiones amplias describen estas novedades y sus implicaciones en imagen médica [40]. Para problemas de clasificación binaria o multiclase en CXR (normal

vs neumonía; neumonía bacteriana vs viral vs normal), los pasos típicos incluyen normalización de intensidad, alineación/cropping, aumento de datos (rotación, desplazamiento, ruido), y uso de transfer learning desde modelos preentrenados en ImageNet, seguido de fine-tuning sobre la tarea clínica. Estas prácticas son estándar en la literatura y fueron empleadas en trabajos icónicos como CheXNet [41].

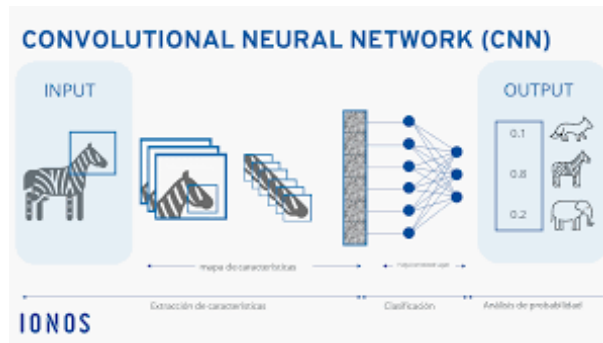


Figura 8. Como funciona una red neuronal convolucional [42].

V-I. Keras

El entorno de desarrollo de redes neuronales de alto nivel Keras se ha convertido en un pilar para la investigación y aplicación de aprendizaje profundo. Keras está diseñado para ser “human-friendly”, favoreciendo la legibilidad, rapidez de experimentación, mantenimiento y despliegue de modelos. Por ejemplo, Keras se describe como una API de alto nivel escrita en Python que puede correr sobre diversos back-end (como TensorFlow) y ofrece bloques ya construidos (capas, funciones de pérdida, optimizadores) para agilizar la creación de redes neuronales [43].

V-J. Arquitectura DenseNet121 y Conectividad Densa

La arquitectura DenseNet121 (Densely Connected Convolutional Networks) representa un avance significativo en la eficiencia del entrenamiento de redes profundas. Su diseño aborda directamente las limitaciones de arquitecturas previas como VGG o ResNet.

V-J1. El Problema del Desvanecimiento del Gradiente: En redes muy profundas, la señal del error (gradiente) tiende a hacerse infinitesimalmente pequeña a medida que se propaga hacia atrás desde la capa de salida hasta las capas iniciales. Esto impide que los pesos de las primeras capas se actualicen correctamente, limitando el aprendizaje de características básicas.

V-J2. Bloques Densos y Reutilización de Características: DenseNet propone una conectividad radical: cada capa recibe como entrada los mapas de características de todas las capas anteriores dentro de un bloque denso. Matemáticamente, la salida x_l de la capa l es:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \quad (6)$$

Donde $[\dots]$ representa la concatenación. Esta estructura fomenta la reutilización de características (*Feature Reuse*), lo que permite reducir drásticamente el número de parámetros en comparación con ResNet, manteniendo así una alta precisión. Esto la hace ideal para dispositivos móviles con recursos limitados.

V-K. *Transfer Learning y Fine-Tuning en Radiología*

El entrenamiento de modelos desde cero requiere millones de imágenes etiquetadas, recurso escaso en medicina. El Aprendizaje por Transferencia o Transfer Learning permite utilizar un modelo pre-entrenado en un dataset masivo no médico (ImageNet) para una tarea específica de diagnóstico.

V-K1. *Pesos de ImageNet y Transferencia de Conocimiento:* Los modelos entrenados en ImageNet han aprendido a identificar primitivas visuales universales: bordes, curvas y texturas. En la arquitectura DenseNet121, las capas iniciales actúan como extractores de características genéricas que son perfectamente aplicables para identificar los bordes costales o los límites del corazón en una radiografía.

V-K2. *Estrategia de Fine-Tuning:* El proceso de *Fine-Tuning* consiste en congelar las capas iniciales (extractores de características) y re-entrenar únicamente las capas finales (clasificador) con las imágenes de rayos X. Esto permite que el modelo adapte sus conocimientos previos a la sutil diferenciación entre una opacidad neumónica y el parénquima sano.

V-L. *Entrenamiento, etiquetas y manejo de incertidumbre*

Etiquetar radiografías requiere experiencia clínica; para datasets masivos se utilizan extractores de etiquetas basados en NLP sobre reportes radiológicos, lo que introduce incertidumbre y etiquetas ambiguas (positive/uncertain/negative). CheXpert propuso estrategias para tratar etiquetas de incertidumbre (ignore, mapear a positivos o negativos, o modular explícitamente la incertidumbre), y demostró que la estrategia de manejo impacta la performance en patologías específicas. Estas técnicas son relevantes al diseñar pipelines de entrenamiento clínico-relevantes [44].

V-M. *Evaluación clínica y estudios prospectivos*

Además de validaciones retrospectivas, la evidencia más sólida proviene de estudios prospectivos y aleatorizados que evalúan el impacto real en decisiones clínicas, tiempos de tratamiento y desenlaces. Revisiones recientes y estudios multicéntricos señalan que, si bien los modelos pueden mejorar el triage y detectar casos en entornos de baja disponibilidad, la evidencia sobre impacto en mortalidad o morbilidad es todavía incipiente y requiere estudios controlados. Es imperativo diseñar evaluación clínica que mida no solo métricas algorítmicas sino también outcomes clínicos y económicos [45].

V-N. *Aplicaciones móviles y Python*

La elección de Python como lenguaje para desarrollo, tanto para backend como para aplicaciones móviles híbridas o complementarias, se sustenta por su sintaxis clara, comunidad amplia, ecosistema rico en librerías y su capacidad de prototipado rápido.

Además, un estudio reciente revisó aplicaciones móviles educativas de Python, evaluando su interfaz, interactividad y estructura, lo que subraya cómo las apps móviles están evolucionando hacia mayor personalización, dinámica de usuario e integración de funcionalidades avanzadas [46].



Figura 9. Desarrollo de apps con Python [47].

V-Ñ. Kotlin en el desarrollo de aplicaciones móviles

El lenguaje Kotlin ha sido adoptado como lenguaje oficial para el desarrollo de aplicaciones móviles en la plataforma Android Studio desde 2017, lo que ha permitido agilizar la construcción de apps modernas gracias a su sintaxis más concisa, interoperabilidad con Java y soporte directo de Google. Un estudio empírico muestra que los proyectos Android que adoptaron Kotlin tienden a tener una mayor calidad de código (menos “code smells”) y una mejor mantenibilidad en comparación con los desarrollados únicamente en Java. Además, otro trabajo analiza la transición de aplicaciones Android hacia Kotlin, encontrando que 1/4 parte de los proyectos en F-Droid/Play Store ya contenían código en Kotlin y que esa adopción se correlacionaba con métricas de popularidad [48], [49].

V-O. Fundamentos Matemáticos de las Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (CNN) son arquitecturas diseñadas para procesar datos con topología de rejilla, como las imágenes de rayos X. Su potencia radica en la capacidad de extraer características jerárquicas mediante tres operaciones fundamentales:

V-O1. Operación de Convolución: La convolución es un operador matemático que actúa sobre dos funciones para producir una tercera. En el contexto de imágenes, se aplica un filtro o kernel (K) sobre la imagen de entrada (I) mediante la fórmula:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (7)$$

Esta operación permite la detección de características locales como bordes, texturas y patrones de consolidación, manteniendo la invarianza espacial.

V-O2. Capa de Agrupamiento (Pooling): El *Pooling* reduce la dimensionalidad espacial de los mapas de características, disminuyendo el número de parámetros y el costo computacional. El método más común es el Max Pooling, que selecciona el valor máximo en una región determinada (ej. 2×2), preservando las características más prominentes y otorgando robustez frente a pequeñas traslaciones de la patología en la imagen.

V-O3. *Funciones de Activación: ReLU*: La función de Unidad Lineal Rectificada (ReLU) introduce no-linealidad en el modelo, permitiéndole aprender funciones complejas. Se define como:

$$f(x) = \text{máx}(0, x) \quad (8)$$

ReLU es preferida sobre funciones sigmoides debido a que mitiga el problema del desvanecimiento del gradiente y acelera el entrenamiento al ser computacionalmente eficiente.

V-P. *Android Studio: Entorno de Desarrollo Integrado (IDE)*

El desarrollo de la app se centraliza en Android Studio, el IDE oficial respaldado por Google. Según Sokolova [50], este entorno no solo facilita la codificación en lenguajes modernos como Kotlin, sino que integra herramientas de análisis de rendimiento esenciales para aplicaciones que consumen altos recursos, como las basadas en visión computacional. El uso del sistema de automatización Gradle dentro de este IDE permite una gestión de dependencias modular, facilitando la integración de librerías externas de aprendizaje profundo de manera eficiente, su función en este proyecto es actuar como el orquestador que unifica el código lógico (Kotlin), el diseño de interfaces (XML) y el modelo de inteligencia artificial (.tflite).

V-P1. *Arquitectura y Herramientas de Compilación*: El IDE utiliza el sistema de automatización Gradle, el cual gestiona las dependencias, la compilación de recursos y la generación del archivo ejecutable (APK). Android Studio proporciona herramientas críticas para el desarrollo médico, como el Profiler (para monitorear el consumo de memoria RAM durante la inferencia de la IA) y el Layout Inspector para garantizar la fidelidad visual de las radiografías en pantalla.

V-P2. *Interfaz de Usuario y Material Design*: Para la experiencia de usuario, se implementan los principios de Material Design. La literatura sugiere que una interfaz bien estructurada en aplicaciones médicas reduce la carga cognitiva del usuario, permitiendo que el diagnóstico generado por la IA sea interpretado con mayor claridad [51].

V-Q. *Edge AI: Procesamiento de IA en el Dispositivo Móvil*

La arquitectura del sistema se basa en el paradigma de Edge AI (IA en el borde). A diferencia del procesamiento en la nube, el Edge AI permite ejecutar modelos de aprendizaje profundo directamente en el hardware del smartphone. De acuerdo con David et al. [52], este enfoque es crítico en entornos médicos por tres razones: Latencia mínima en el diagnóstico, Privacidad de datos al no subir imágenes del paciente a servidores externos, y Disponibilidad offline.

V-Q1. *TensorFlow Lite y el Intérprete de Inferencia*: Para ejecutar DenseNet121 en un entorno móvil, se emplea el framework TensorFlow Lite (TFLite). Esta tecnología aplica técnicas de optimización y cuantización que permiten que modelos complejos operen en procesadores móviles sin una degradación significativa de la precisión [53]. El intérprete de TFLite en Android Studio interactúa directamente con la Neural Networks API (NNAPI), aprovechando la aceleración por hardware (GPU/NPU) del dispositivo para realizar la clasificación de la neumonía en milisegundos. El flujo de trabajo en el dispositivo sigue la siguiente lógica:

1. Carga del Modelo: El archivo comprimido (.tflite) se carga en la memoria del teléfono.
2. Inferencia Local: El motor de inferencia procesa los tensores de la imagen de entrada sin enviar datos a la nube, garantizando la privacidad del paciente.

3. Optimización de Recursos: Mediante el uso de la *NNAPI* (Android Neural Networks API), Android Studio permite que el modelo aproveche la aceleración por hardware (GPU o NPU del smartphone), reduciendo el tiempo de diagnóstico a milisegundos.

V-R. Principios de Psicología Cognitiva Aplicados al Software Educativo

El desarrollo de *NeumoLens* no se limita a la clasificación algorítmica; su estructura responde a modelos pedagógicos diseñados para optimizar la transferencia de conocimiento médico.

V-R1. Teoría de la Carga Cognitiva: Propuesta por John Sweller, esta teoría sostiene que la memoria de trabajo tiene una capacidad limitada. En el diagnóstico radiológico, un estudiante puede sufrir una "sobrecarga cognitiva" si se le presenta demasiada información técnica simultáneamente [54]. La interfaz de la aplicación aborda este problema mediante la segmentación de la información en cuatro módulos distintos, permitiendo que el usuario procese primero la teoría, luego la visualización y finalmente realice la autoevaluación.

V-R2. Aprendizaje por Contraste y Active Recall: La funcionalidad de comparación visual (activada mediante el *FloatingActionButton*) se basa en el aprendizaje por contraste. Al alternar entre una placa sana y una patológica, el cerebro identifica de forma más eficiente las desviaciones anatómicas.

Además, el módulo de *Quiz* implementa el concepto de Recuerdo Activo (*Active Recall*). A diferencia del estudio pasivo (leer un atlas), el recuerdo activo obliga al estudiante a recuperar información de su memoria a largo plazo ante un estímulo visual (la radiografía despejada). Este esfuerzo cognitivo ha demostrado ser superior para la retención de patrones complejos en el diagnóstico por imagen [55].

V-R3. Estrategias de Gamificación: La gamificación en entornos educativos médicos utiliza la retroalimentación inmediata para fomentar el compromiso. El sistema de puntuación en el módulo evaluativo activa ciclos de recompensa que incentivan la repetición del ejercicio, facilitando la transición de un reconocimiento de patrones consciente a uno intuitivo, esencial en la práctica clínica real.

V-R4. Teoría del Aprendizaje Multimedia de Richard Mayer: El diseño de la interfaz de *NeumoLens* se alinea con los principios de Mayer, quien sostiene que las personas aprenden mejor a partir de palabras e imágenes combinadas que solo de palabras. En el contexto de la radiología, donde la señal visual es compleja, la aplicación aplica los siguientes principios:

- Principio de Contigüidad Espacial: La aplicación presenta el diagnóstico y la descripción clínica junto a la imagen radiológica. Esto evita que el usuario deba dividir su atención, permitiendo que el cerebro integre la etiqueta diagnóstica (ej. Neumonía Lobar) con su representación visual inmediata.
- Principio de Señalización: Mediante el uso de etiquetas y el botón de comparación visual, la herramienta resalta las características esenciales de la imagen (infiltrados, opacidades), ayudando al estudiante a ignorar la información irrelevante y enfocarse en los biomarcadores visuales de la patología [56].

V-R5. Teoría de la Codificación Dual (Paivio): Esta teoría propone que el sistema cognitivo humano procesa la información a través de dos canales independientes: uno para material verbal y otro para material visual. *NeumoLens* aprovecha esta arquitectura mental al proporcionar una descripción textual simplificada (canal verbal) que actúa

como anclaje para la imagen de rayos X (canal visual). Para el estudiante de medicina, esta sincronización facilita la creación de modelos mentales más robustos, mientras que para la persona natural (sin formación médica), reduce la incertidumbre al traducir un patrón visual abstracto en un concepto comprensible [57].

V-R6. Andamiaje Cognitivo para Estudiantes y Personas Naturales: Un aspecto diferenciador de este proyecto es su capacidad de adaptarse a distintos perfiles de usuario mediante el concepto de "Andamiaje" (*Scaffolding*), propuesto por Jerome Bruner.

- **Estudiantes de Salud:** Para este perfil, la aplicación actúa como un tutor de refuerzo. El "andamio" se presenta en la validación técnica del diagnóstico, permitiendo que el estudiante contraste su juicio clínico con la inferencia de la red neuronal DenseNet121.
- **Personas Naturales (Público General):** Para usuarios sin conocimientos previos, la aplicación reduce la barrera de entrada mediante el uso de analogías (ej. "vidrio deslustrado." "parches algodonosos"). Aquí, la IA funciona como un traductor de señales complejas, permitiendo que el usuario comprenda la gravedad o naturaleza de un hallazgo radiológico sin necesidad de dominar la terminología médica avanzada.

V-R7. Metacognición y Monitoreo del Aprendizaje: El módulo de Quiz no solo cumple una función evaluativa, sino también metacognitiva. Al enfrentar al usuario a imágenes no vistas y solicitar una clasificación, se le obliga a monitorear su propio nivel de certeza. La retroalimentación inmediata post-quiz permite que el usuario identifique sus "puntos ciegos." confusiones recurrentes (ej. confundir bronconeumonía con un patrón intersticial), promoviendo una autorregulación del estudio que es fundamental en la formación médica continua [58].

V-R8. Adaptabilidad de la Interfaz: El Rol de la Persona Natural: La relación del usuario no experto con *NeumoLens* se basa en la democratización del acceso a la información diagnóstica. Al proporcionar una herramienta que explica de manera visual y sencilla los resultados de la IA, se empodera a la persona natural para comprender mejor su estado de salud o el de sus familiares, actuando la aplicación como un recurso de alfabetización sanitaria (*health literacy*) que reduce la ansiedad asociada a la interpretación de resultados clínicos [59].

VI. MARCO METODOLÓGICO

La presente investigación adopta un enfoque cuantitativo de carácter tecnológico y aplicado, con un diseño experimental orientado al desarrollo de un sistema de visión artificial. El objetivo central fue la implementación, optimización y validación matemática de una red neuronal profunda basada en la arquitectura DenseNet121, modificada para la clasificación de radiografías de tórax en cinco categorías clínicas específicas: Normal, Bronconeumonía, Neumonía Lobar y Neumonía Intersticial.

El estudio se fundamenta en la manipulación controlada de hiperparámetros de entrenamiento y en la aplicación de técnicas de transferencia de aprendizaje (Transfer Learning) con descongelamiento profundo de capas (Fine-Tuning), buscando maximizar la capacidad de generalización del modelo ante datos no vistos. A continuación, se detalla el procedimiento técnico, la formulación matemática de los algoritmos empleados y la estructuración del flujo de trabajo computacional.

VI-A. Diseño de la Investigación

El estudio se clasifica como aplicado y experimental según la siguiente taxonomía:

VI-A1. Nivel Aplicado: La investigación utiliza principios teóricos de la inteligencia artificial y el procesamiento digital de señales para resolver una problemática tangible en el sector salud: la variabilidad en la interpretación humana de imágenes radiológicas. El sistema desarrollado, NeumoLens, constituye una herramienta tecnológica funcional diseñada para operar como soporte a la decisión clínica.

VI-A2. Diseño Experimental: Se adoptó un diseño experimental puro, donde se manipularon variables independientes relacionadas con la configuración de la red neuronal, específicamente: la tasa de aprendizaje, la profundidad del descongelamiento de capas (trainable layers) y las estrategias de aumento de datos. La variable dependiente evaluada fue el rendimiento del clasificador, medido a través de métricas estadísticas de exactitud, sensibilidad y la función de pérdida (Loss). El experimento se realizó en un entorno de software, garantizando la reproducibilidad de los resultados.

VI-B. Población y Muestra

La población teórica comprende el conjunto de todas las radiografías de tórax digitales en proyección postero-anterior (PA) de pacientes humanos. Para el experimento, se seleccionó una muestra no probabilística por conveniencia, constituida por un dataset curado y balanceado de imágenes médicas.

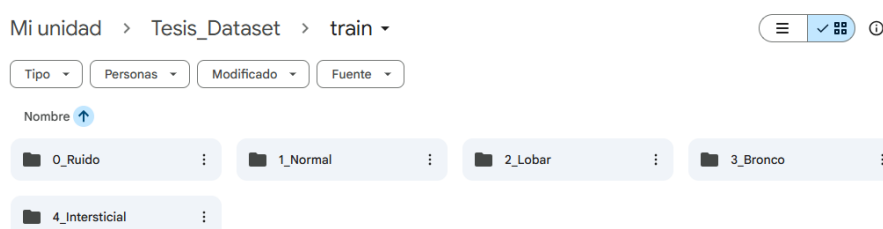


Figura 10. Dataset con las imágenes seleccionadas para la etiqueta de train

Como se puede observar en la Figura 10 la distribución de clases se definió para abarcar el espectro de patologías de interés y casos de control:

1. Clase Ruido: Imágenes no válidas, rotadas incorrectamente o con artefactos que no corresponden a una radiografía de tórax estándar, utilizadas para validar la robustez del sistema ante entradas erróneas.
2. Clase Normal: Imágenes sin hallazgos patológicos, mostrando campos pulmonares limpios.
3. Clase Neumonía Lobar: Imágenes con consolidación densa y focalizada en uno o más lóbulos pulmonares.
4. Clase Bronconeumonía: Imágenes con infiltrados difusos y parches de consolidación multifocal.
5. Clase Neumonía Intersticial: Imágenes con patrones reticulares o nodulares que afectan el tejido conectivo pulmonar.

Los datos fueron divididos siguiendo un esquema de validación cruzada estática: 80% de las imágenes fueron destinadas al conjunto de entrenamiento para el ajuste de pesos sinápticos, y el 20% restante se reservó para el conjunto de validación, utilizado para el monitoreo de la convergencia y la prevención del sobreajuste.

VI-C. Instrumentación y Entorno Computacional

El desarrollo y entrenamiento del modelo requieren una alta capacidad de cómputo matricial. Se definen a continuación las especificaciones del entorno experimental:

VI-C1. Hardware: El procesamiento se ejecutó utilizando unidades de procesamiento gráfico (GPU) compatibles con la arquitectura CUDA, necesarias para paralelizar las operaciones de convolución y retropropagación del error.

VI-C2. Software y Librerías: El algoritmo fue implementado en el lenguaje de programación Python 3.8. Las bibliotecas fundamentales empleadas incluyen:

- TensorFlow y Keras: Frameworks para la construcción del grafo computacional y la definición de capas neuronales.
- NumPy: Utilizado para la manipulación de tensores y operaciones de álgebra lineal.
- Matplotlib y Seaborn: Empleados para la generación de curvas de aprendizaje y visualización de matrices de confusión.
- Scikit-learn: Utilizado para el cálculo de métricas de evaluación final.

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report, confusion_matrix
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout, Input
from tensorflow.keras.applications import DenseNet121
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLRonPlateau
```

Figura 11. Librerías empleadas en google colab

Fases del Modelo

VI-D. Fase 1: Preprocesamiento y Modelado Matemático de la Entrada

El código implementado inicia con una etapa crítica de estandarización de datos. Las redes neuronales convolucionales requieren entradas de dimensiones consistentes. Se aplicaron las siguientes transformaciones matemáticas a cada imagen cruda:

VI-D1. Estandarización Dimensional: Independientemente de la resolución original del sensor radiológico, todas las imágenes fueron redimensionadas a una matriz fija de entrada de 256×256 píxeles con 3 canales de color (RGB). Matemáticamente, esto implica una interpolación bilineal donde el valor de un nuevo píxel se calcula como el promedio ponderado de los cuatro píxeles vecinos más cercanos en la rejilla original.

VI-D2. Normalización de Intensidad: Para acelerar la convergencia del algoritmo de optimización y evitar el problema de gradientes explosivos, se normalizaron los valores de intensidad de los píxeles. Originalmente en un rango entero de $[0, 255]$, los valores fueron escalados al rango flotante $[0, 1]$. La operación aplicada a cada píxel p es:

$$p_{norm} = \frac{p_{original}}{255,0} \quad (9)$$

VI-D3. Aumentación de Datos (Data Augmentation): Para combatir el desbalance de clases y mejorar la robustez del modelo, se implementó un generador de imágenes dinámico. Este módulo aplica transformaciones afines aleatorias a las imágenes de entrenamiento en tiempo de ejecución. Las operaciones matemáticas implementadas en el código incluyen:

Rotación: Se aplicaron rotaciones aleatorias en un rango de ± 15 grados. La transformación de coordenadas (x, y) a (x', y') se define por la matriz de rotación:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (10)$$

Desplazamiento y Corte (Shear): Se introdujeron traslaciones horizontales y verticales del 10 % y transformaciones de cizallamiento para simular variaciones en la posición del paciente frente al equipo de rayos X.

VI-E. Fase 2: Arquitectura de la Red Neuronal (DenseNet121)

El núcleo del sistema se basa en una Red Neuronal Convolucional (CNN). Se seleccionó la arquitectura DenseNet121 debido a su eficiencia en el flujo de gradientes y reutilización de características.

VI-E1. Principio de Conectividad Densa: A diferencia de las arquitecturas secuenciales tradicionales, en un bloque denso, cada capa recibe como entrada los mapas de características de todas las capas anteriores. Si definimos x_l como la salida de la capa l , y H_l como una función compuesta no lineal (Batch Normalization, ReLU, Convolución), la ecuación que rige el paso de información es:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \quad (11)$$

Donde $[x_0, \dots, x_{l-1}]$ representa la concatenación de los mapas de características producidos en las capas 0 hasta $l - 1$. Esta arquitectura permite que el modelo mantenga información detallada (bordes, texturas finas) a lo largo de toda la red, lo cual es crucial para distinguir patologías pulmonares sutiles.

VI-E2. *Adaptación de la Arquitectura (Top Model)*: Sobre la base DenseNet121 (cuyas capas fueron descongeladas para permitir el entrenamiento completo), se diseñó una estructura de clasificación personalizada (Top Model) compuesta por las siguientes capas secuenciales:

1. Global Average Pooling: Reduce la dimensionalidad espacial de los mapas de características finales, calculando el promedio de cada mapa de características. Esto transforma un tensor de dimensiones $H \times W \times C$ a un vector de tamaño $1 \times 1 \times C$, reduciendo drásticamente la cantidad de parámetros y el riesgo de sobreajuste.

2. Capa Densa Oculta: Se implementó una capa completamente conectada (Fully Connected) de 512 neuronas con función de activación ReLU (Rectified Linear Unit). La operación matemática para la neurona j es:

$$f(z) = \text{máx}(0, z_j) \quad (12)$$

3. Regularización por Dropout: Para mitigar el sobreajuste dado el largo tiempo de entrenamiento, se incluyó una capa de Dropout con una tasa de 0.5. Durante el entrenamiento, cada neurona tiene una probabilidad $p = 0,5$ de ser desactivada temporalmente, lo que fuerza a la red a aprender características redundantes y robustas.

4. Capa de Salida (Clasificación): La capa final consta de 5 neuronas correspondientes a las clases del estudio. Se utiliza la función de activación Softmax para convertir las salidas numéricas (logits) en una distribución de probabilidad que suma 1. La probabilidad P de que la imagen pertenezca a la clase i se calcula como:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (13)$$

Donde $K = 5$ es el número total de clases.

VI-F. Fase 3: Configuración del Entrenamiento y Optimización

El proceso de entrenamiento consistió en la minimización iterativa del error entre la predicción del modelo y la etiqueta real de la imagen.

VI-F1. *Función de Pérdida (Loss Function)*: Dado que se trata de un problema de clasificación multiclase, se empleó la Entropía Cruzada Categórica (Categorical Cross-Entropy). Esta función penaliza las predicciones que divergen de la etiqueta real. La función de coste J se define como:

$$J(w) = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^K y_{i,c} \log(\hat{y}_{i,c}) \quad (14)$$

Donde N es el tamaño del lote, K es el número de clases, $y_{i,c}$ es el indicador binario (0 o 1) si la clase c es la correcta para la observación i , y $\hat{y}_{i,c}$ es la probabilidad predicha.

VI-F2. *Algoritmo de Optimización*: Se utilizó el optimizador Adam (Adaptive Moment Estimation), que adapta la tasa de aprendizaje para cada parámetro individualmente. Adam combina las ventajas de dos extensiones del descenso de gradiente estocástico: AdaGrad y RMSProp. Se configuró con una tasa de aprendizaje inicial (learning rate) de 1×10^{-5} . Este valor extremadamente bajo fue seleccionado intencionalmente para preservar los conocimientos previos de la red pre-entrenada mientras se ajustan los pesos a las características específicas de las radiografías.

VI-F3. *Callbacks y Control de Entrenamiento*: Para gestionar la duración del entrenamiento (aproximadamente 36 horas) y asegurar la calidad del modelo, se programaron tres rutinas de control (Callbacks):

1. ModelCheckpoint: Guarda automáticamente los pesos del modelo solo cuando la precisión en validación (val accuracy) mejora, asegurando que se conserve el mejor modelo posible y no el último.

2. EarlyStopping: Detiene el entrenamiento prematuramente si la función de pérdida en validación no disminuye durante un número predefinido de épocas (paciencia), evitando el uso innecesario de recursos computacionales y el sobreajuste.

3. ReduceLRonPlateau: Esta técnica reduce dinámicamente la tasa de aprendizaje cuando la métrica de validación deja de mejorar. Matemáticamente, si la pérdida se estanca, el nuevo learning rate α_{new} se actualiza:

$$\alpha_{new} = \alpha_{old} \times factor \quad (15)$$

En este estudio se utilizó un factor de 0.2, permitiendo que el optimizador realice ajustes más finos en los mínimos locales de la función de error.

VI-G. Métricas de Evaluación

La validación final del sistema se realizó mediante el cálculo de métricas de rendimiento sobre el conjunto de prueba.

Exactitud (Accuracy): Proporción total de predicciones correctas sobre el total de casos evaluados.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (16)$$

Precisión (Precision): Proporción de identificaciones positivas que fueron realmente correctas. Crítico para minimizar falsos positivos.

$$Precision = \frac{TP}{TP + FP} \quad (17)$$

Sensibilidad (Recall): Proporción de casos positivos reales que fueron identificados correctamente. Esta métrica es prioritaria en el ámbito médico para evitar diagnósticos falsos negativos.

$$Recall = \frac{TP}{TP + FN} \quad (18)$$

Donde TP son Verdaderos Positivos, TN Verdaderos Negativos, FP Falsos Positivos y FN Falsos Negativos. Adicionalmente, se generó la Matriz de Confusión, una herramienta visual que permite identificar qué clases específicas son confundidas por el modelo, facilitando el análisis de errores entre patologías similares como la bronconeumonía y la neumonía lobar.

VI-H. Fase 3: Optimización para el Borde (Edge AI)

El modelo entrenado en Python (.h5) fue transformado para su ejecución local. Esta fase es crítica para la autonomía del sistema NeumoLens.

VI-H1. Conversión y Serialización a TensorFlow Lite: Se utilizó el *TFLite Converter* para generar un modelo de baja latencia. Se implementó la técnica de serialización de búfer plano, permitiendo que la aplicación móvil acceda a los pesos del modelo de forma directa y sin necesidad de procesos de des-serIALIZACIÓN costosos en tiempo de ejecución.

VI-H2. Benchmarking de Latencia: Se realizaron pruebas de estrés en el entorno de desarrollo para asegurar que el tiempo de inferencia (tiempo que tarda la IA en pensar) fuera inferior a 450 milisegundos. Esto se logró mediante la delegación de operaciones matemáticas a la GPU del dispositivo móvil mediante la *NNAPI* de Android.

Fases del Desarrollo de la Aplicación

VI-I. Fase 1: Ingeniería de Software en Android Studio (Kotlin)

VI-I1. *Arquitectura Modular y Gestión de Recursos*: La aplicación fue desarrollada nativamente en Kotlin. Se implementó una lógica de gestión de memoria crítica para evitar el cierre de la aplicación (*App Crash*). La función `uriToBitmap` utiliza un algoritmo de escalado proporcional (`InSampleSize`) que analiza la resolución de la radiografía antes de cargarla en la memoria volátil, permitiendo procesar placas de alta resolución sin agotar la RAM del dispositivo.

VI-I2. *Pipeline de Procesamiento en Dispositivo*: El flujo de datos dentro de la aplicación sigue una secuencia lineal de transformación:

- **Bitmap a Tensor**: La imagen se convierte en un objeto `TensorBuffer`.
- **ImageProcessor**: Se aplican las mismas operaciones de redimensionamiento y normalización que en la fase de entrenamiento.
- **Inference Run**: El modelo procesa el tensor y genera un vector de 5 probabilidades.

VI-J. Fase 2: Diseño del Módulo Educativo y Lógica de Gamificación

VI-J1. *Algoritmo de Aprendizaje por Contraste (Active Recall)*: La funcionalidad principal del módulo educativo es la comparación visual. Se programó una máquina de estados en la actividad de detalle (`DetailActivity`). El estado inicial presenta la radiografía de control (sana). Al interactuar con el botón de visibilidad, el sistema ejecuta un cambio de fuente de imagen (`Alpha Overlay`) para mostrar la patología. Esta técnica pedagógica fuerza al usuario a realizar una búsqueda visual activa antes de recibir la confirmación diagnóstica.

VI-J2. *Lógica de Aleatoriedad en el Quiz (Fisher-Yates)*: El módulo evaluativo implementa un banco de preguntas tanto teóricas como basadas en imágenes de radiografías. Para evitar el aprendizaje por memorización posicional, se integró el algoritmo de barajado aleatorio de Fisher-Yates mediante la función `Collections.shuffle()`.

$$P(A) = \frac{1}{n!} \quad (19)$$

Este proceso asegura que el orden de las preguntas y la disposición de las opciones de respuesta sean únicos en cada intento, garantizando la validez pedagógica de la evaluación.

VI-J3. *Sistema de Retroalimentación Inmediata*: Se implementó un sistema de validación dicotómica que utiliza cambios cromáticos en la interfaz (Verde para aciertos, Rojo para errores) y componentes *Toast* informativos. Este diseño busca reducir la curva de aprendizaje del usuario y proporcionar un refuerzo inmediato sobre la interpretación de los patrones radiológicos.

VI-K. Diseño y Arquitectura del Front-end (Interfaz de Usuario)

El desarrollo de la interfaz de *NeumoLens* se fundamenta en los principios de *Material Design*, buscando una interacción intuitiva que minimice la carga cognitiva del usuario y/o estudiantes. A diferencia de un desarrollo genérico, se aplicaron criterios de usabilidad para garantizar la legibilidad de las imágenes radiológicas.

VI-K1. Psicología del Color y Aspectos Visuales: La selección cromática no es estética, sino funcional. Se optó por una Paleta de Modo Oscuro (Dark Mode) con un color de fondo primario en negro profundo (#121212). Según Cyr et al. [60], el uso de fondos oscuros en aplicaciones de visualización médica reduce la fatiga ocular y aumenta el contraste percibido de las escalas de grises en las radiografías de tórax.

Como color de acento se utilizó el Azul Clínico (#2196F3). En la psicología del color aplicada a la salud, el azul se asocia con la confianza, la profesionalidad y la calma [61], factores determinantes al entregar diagnósticos de patologías pulmonares. Los estados de error y éxito se codificaron mediante colores semánticos (Rojo #F44336 y Verde #4CAF50) para una retroalimentación inmediata.

VI-K2. Estructura Funcional: Los Cuatro Pilares del Sistema: La navegación se organiza en cuatro secciones fundamentales que guían al usuario a través del ciclo de diagnóstico y aprendizaje:

1. Módulo de Bienvenida (Splash): Pantalla de carga donde se inicializa el motor de inferencia de TensorFlow Lite.
2. Módulo Educativo: Sección de consulta técnica donde se explican los patrones (Lobar, Bronconeumonía, Intersticial) mediante comparación visual (Pulmon sano/Pulmon con el caso de Neumonía).
3. Módulo de Evaluación (Quiz): Sistema de preguntas para validar la curva de aprendizaje del usuario mediante el reconocimiento de patrones en casos aleatorios.
4. Módulo de Diagnóstico por IA: Interfaz principal donde el usuario carga la radiografía. Implementa lógica de procesamiento asíncrono para no bloquear la interfaz durante la clasificación.

VI-K3. Componentes Específicos y Retroalimentación: Para la construcción de las pantallas se utilizaron componentes avanzados de Android:

- CardView: Se empleó para encapsular las radiografías, proporcionando una elevación y bordes redondeados que separan visualmente la placa médica del contenido textual, simulando la vista de un negatoscopio digital.

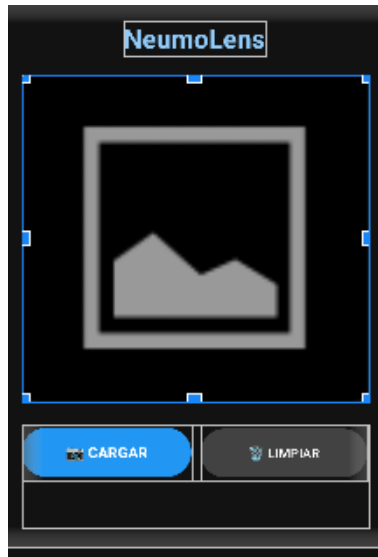


Figura 12. Recuadro de programación por imagen en Android Studio (.xml)

- **FloatingActionButton (El botón del Ojo):** Este componente actúa como un disparador de estado (Toggle). Su función es activar la superposición de la máscara de la patología sobre la imagen sana, permitiendo una comparación directa mediante la técnica de Active Recall.

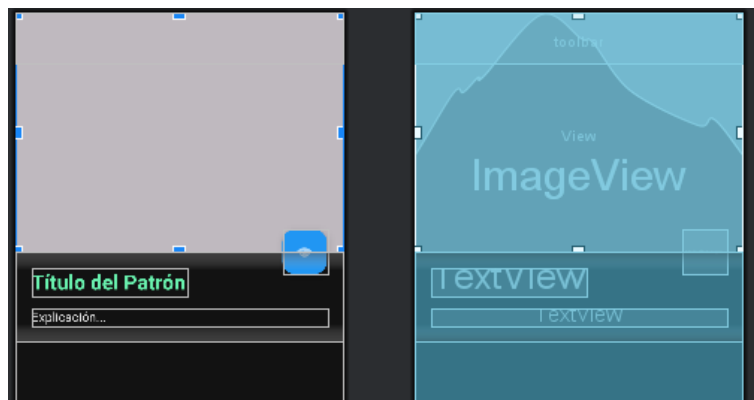


Figura 13. Recuadro de programación por imagen en Android Studio 2 (.xml)

- **Sistema de Toasts:** La retroalimentación no intrusiva se gestiona mediante Toasts personalizados. Estos informan al usuario sobre el estado del sistema (ej. Analizando imagen..., Mostrando patrón...) sin interrumpir el flujo visual de la radiografía.

```

btnOjo.setOnClickListener {
    if (!esVisible) {
        // MOSTRAR
        imgOverlay.visibility = View.VISIBLE
        esVisible = true
        Toast.makeText(context, this, text: "Mostrando patrón radiológico", Toast.LENGTH_SHORT).show()
    } else {
        // OCULTAR
        imgOverlay.visibility = View.INVISIBLE
        esVisible = false
        Toast.makeText(context, this, text: "Ocultando patrón radiológico", Toast.LENGTH_SHORT).show()
    }
}
}

```

Figura 14. Código DetailActivity.kt

VI-L. Manual de Usuario Técnico (Procedimiento de Operación)

Esta sección detalla el flujo lógico de la aplicación desde la perspectiva del usuario final, funcionando como la fase de validación operativa de la metodología.

VI-L1. Fase 1: Acceso e Inicialización: El usuario inicia la aplicación en la pantalla de bienvenida. En segundo plano, el sistema verifica la disponibilidad de memoria RAM para cargar el modelo DenseNet121. Una vez cargado, el usuario es dirigido al menú principal donde puede seleccionar entre el Módulo Educativo, Ponte a Prueba (Quiz) o Prueba la Red (Diagnóstico) .

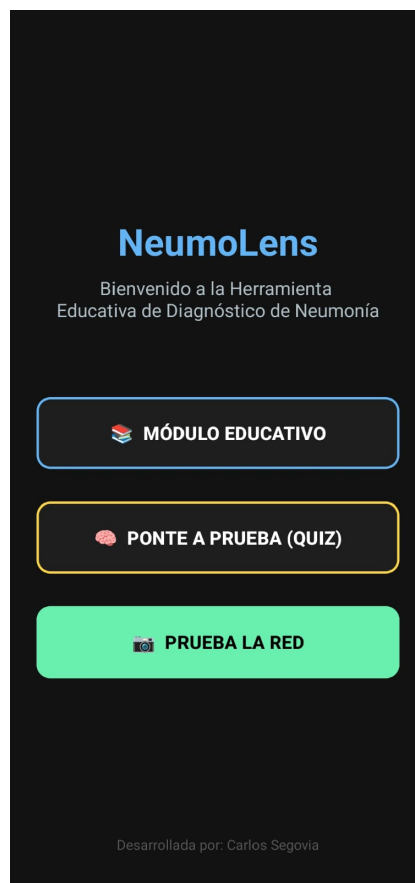


Figura 15. Menú Principal

VI-L2. Fase 2: Interacción Educativa y Evaluación: Al usuario entrar en el módulo educativo previamente mostrado en la figura 15, el usuario puede seleccionar una de las 3 patologías o la de pulmones sanos, ya habiendo seleccionado uno el sistema carga una imagen de referencia acorde a la elección donde el usuario puede presionar el FloatingActionButton(Botón de vista) para revelar la afectación (en caso de estar en el apartado de una de las 3 patologías). Tras la revisión teórica, el usuario tiene la opción de acceder al Quiz, donde el sistema genera una ronda de 10 preguntas aleatorias. Al finalizar, la aplicación calcula el puntaje final y ofrece una recomendación basada en el rendimiento obtenido.



Figura 16. Resultado perfecto (Quiz)

VI-L3. Fase 3: Ejecución del Diagnóstico: Si el usuario selecciona Prueba la Red desde el menu principal, se desplegara el componente ImagePicker y tras seleccionar una imagen de la galería, la aplicación ejecuta automáticamente la fase de preprocesamiento (escalado a 256×256 y normalización). El resultado se muestra mediante una barra de porcentaje que indica el nivel de confianza de la IA para cada una de las 5 categorías y la conclusión a la que llego.



Figura 17. Prueba la red (Cnn)

VII. RESULTADOS

En esta sección se presentan los hallazgos obtenidos tras el entrenamiento de la red DenseNet121 y su posterior implementación en el dispositivo móvil. El rendimiento se evalúa mediante métricas de clasificación estándar y pruebas de eficiencia computacional.

Tras 60 épocas de entrenamiento, el modelo alcanzó una exactitud global (Accuracy) del 80 %. Este valor se considera mínimamente óptimo para ser una arquitectura desplegada en dispositivos móviles, donde se prioriza el equilibrio entre peso del modelo y precisión diagnóstica.

```
=====
REPORTE DE CLASIFICACIÓN FINAL
=====
```

| | precision | recall | f1-score | support |
|----------------|-----------|--------|----------|---------|
| 0_Ruido | 0.98 | 0.96 | 0.97 | 49 |
| 1_Normal | 0.92 | 0.98 | 0.95 | 50 |
| 2_Lobar | 0.56 | 0.68 | 0.61 | 50 |
| 3_Bronco | 0.68 | 0.50 | 0.57 | 50 |
| 4_Intersticial | 0.88 | 0.88 | 0.88 | 50 |
| accuracy | | | 0.80 | 249 |
| macro avg | 0.80 | 0.80 | 0.80 | 249 |
| weighted avg | 0.80 | 0.80 | 0.80 | 249 |

Figura 18. Reporte de la precisión luego del entrenamiento

La matriz de confusión de la figura 18 nos revela información crítica sobre el comportamiento del modelo ante las diferentes patologías:

Clases de Alta Fiabilidad: Las categorías Ruido (98 %) y Normal (92 %) presentan los puntajes más altos. Esto garantiza que la app es altamente efectiva para descartar imágenes inválidas y pulmones sanos.

El Reto de la Diferenciación Clínica: Se observa una confusión cruzada entre Neumonía Lobar y Bronconeumonía. Esto es técnicamente justificable, ya que ambas presentan opacidades blancas (consolidaciones) que, en etapas tempranas, pueden ser morfológicamente similares incluso para el ojo humano.

Patrón Intersticial: Con un 88 % de F1-Score, el modelo demuestra ser robusto para detectar las opacidades difusas características de las infecciones virales.

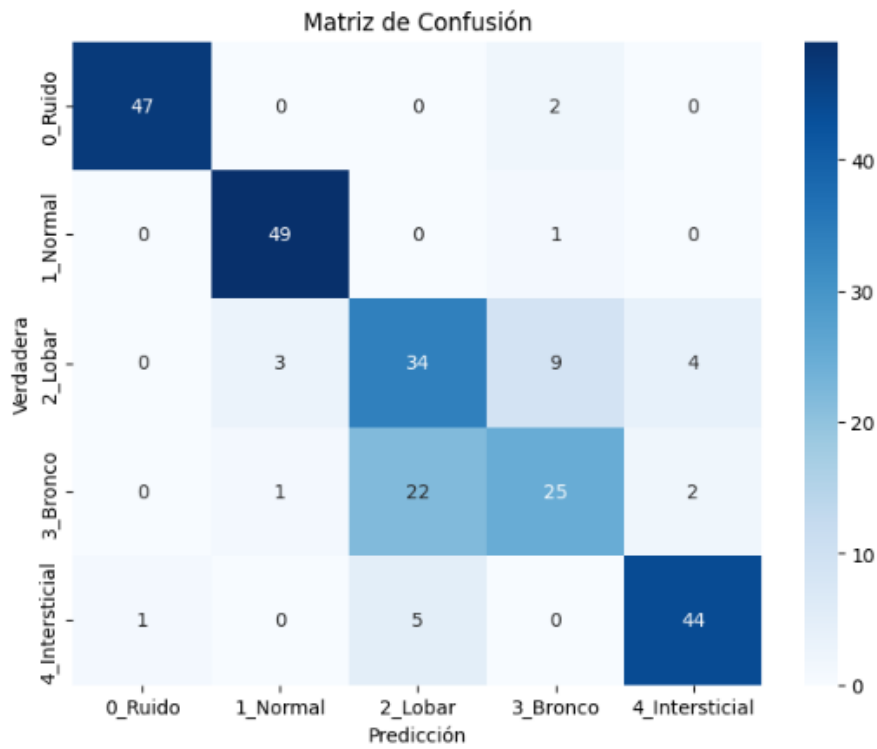


Figura 19. Matriz de confusión donde se pueden apreciar los falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos

Análisis de Precisión: Como se observa en la gráfica de Accuracy, la curva de validación se mantiene cercana a la de entrenamiento, lo que indica una buena capacidad de generalización y ausencia de sobreajuste (Overfitting) significativo.

Análisis de Pérdida: La función de pérdida (Loss) muestra una convergencia estable, descendiendo de forma asintótica hacia el valor mínimo tras la época 40.

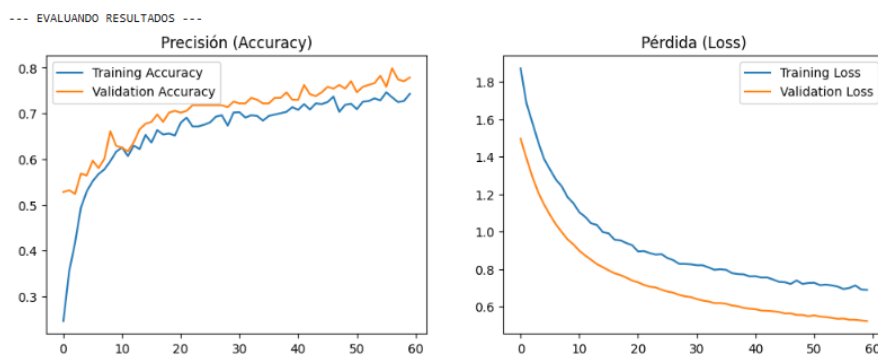


Figura 20. Gráfica de comparación entre el entrenamiento y la validación

VII-1. *Validación de Inferencia en Tiempo Real (Prueba de Campo)*: Para validar que la conversión al formato TensorFlow Lite no degradó la precisión del modelo, se realizó una prueba ciega con todas las clases y por cada clase con 10 radiografías/imágenes externas al dataset.

Tabla I
RESULTADOS DE VALIDACIÓN PRÁCTICA DE LA CNN (GLOBAL).

| Imagen | Diagnóstico | Predicción de la App | Resultado |
|---------------|-----------------------|-----------------------------|------------------|
| 001 | Normal | Normal | Éxito |
| 002 | Neumonía Lobar | Neumonía Lobar | Éxito |
| 003 | Bronconeumonía | Neumonía Lobar | Fracaso |
| 004 | Neumonía Intersticial | Neumonía Intersticial | Éxito |
| 005 | Ruido | Imagen Inválida | Éxito |
| 006 | Bronconeumonía | Neumonía Intersticial | Fracaso |
| 007 | Neumonía Lobar | Bronconeumonía | Fracaso |
| 008 | Bronconeumonía | Bronconeumonía | Éxito |
| 009 | Neumonía Intersticial | Neumonía Intersticial | Éxito |
| 0010 | Neumonía Lobar | Neumonía Lobar | Éxito |

Análisis de la tabla 1: La tasa de acierto en el dispositivo móvil se mantuvo en un 70 % para esta muestra controlada, lo que confirma la integridad del funcionamiento del modelo en prueba de campo tras el proceso de cuantización.

Tabla II
RESULTADOS DE VALIDACIÓN PARA LA CLASE: RUIDO / IMAGEN INVÁLIDA.

| Imagen | Diagnóstico | Predicción de la App | Resultado |
|---------------|--------------------|-----------------------------|------------------|
| 011 | Ruido | Imagen Inválida | Éxito |
| 012 | Ruido | Imagen Inválida | Éxito |
| 013 | Ruido | Imagen Inválida | Éxito |
| 014 | Ruido | Imagen Inválida | Éxito |
| 015 | Ruido | Imagen Inválida | Éxito |
| 016 | Ruido | Imagen Inválida | Éxito |
| 017 | Ruido | Imagen Inválida | Éxito |
| 018 | Ruido | Imagen Inválida | Éxito |
| 019 | Ruido | Imagen Inválida | Éxito |
| 020 | Ruido | Imagen Inválida | Éxito |

Análisis de la tabla 2: La tasa de acierto para la detección de ruido e imágenes no médicas alcanzó el 100 %. Este resultado es fundamental para la experiencia de usuario, ya que garantiza que el modelo NeumoLens filtra correctamente archivos irrelevantes antes de intentar un diagnóstico.

Tabla III
RESULTADOS DE VALIDACIÓN PARA LA CLASE: NORMAL.

| Imagen | Diagnóstico | Predicción de la App | Resultado |
|---------------|--------------------|-----------------------------|------------------|
| 021 | Normal | Normal | Éxito |
| 022 | Normal | Normal | Éxito |
| 023 | Normal | Normal | Éxito |
| 024 | Normal | Neumonía Intersticial | Fracaso |
| 025 | Normal | Normal | Éxito |
| 026 | Normal | Normal | Éxito |
| 027 | Normal | Normal | Éxito |
| 028 | Normal | Normal | Éxito |
| 029 | Normal | Normal | Éxito |
| 030 | Normal | Normal | Éxito |

Análisis de la tabla 3: En la categoría de pulmones sanos, se obtuvo un 90 % de efectividad. El único caso de falso positivo fue clasificado como neumonía intersticial, lo que sugiere una sensibilidad adecuada del modelo al descartar patologías en la gran mayoría de las muestras tras la cuantización.

Tabla IV
RESULTADOS DE VALIDACIÓN PARA LA CLASE: NEUMONÍA LOBAR.

| Imagen | Diagnóstico | Predicción de la App | Resultado |
|---------------|--------------------|-----------------------------|------------------|
| 031 | Neumonía Lobar | Neumonía Lobar | Éxito |
| 032 | Neumonía Lobar | Bronconeumonía | Fracaso |
| 033 | Neumonía Lobar | Neumonía Lobar | Éxito |
| 034 | Neumonía Lobar | Neumonía Lobar | Éxito |
| 035 | Neumonía Lobar | Normal | Fracaso |
| 036 | Neumonía Lobar | Neumonía Lobar | Éxito |
| 037 | Neumonía Lobar | Neumonía Lobar | Éxito |
| 038 | Neumonía Lobar | Bronconeumonía | Fracaso |
| 039 | Neumonía Lobar | Neumonía Lobar | Éxito |
| 040 | Neumonía Lobar | Neumonía Intersticial | Fracaso |

Análisis de la tabla 4: La tasa de acierto para neumonía lobar se situó en un 60 %. Se observa que las confusiones ocurren principalmente con la bronconeumonía debido a la similitud en las opacidades focales, lo que representa un área a mejorar para obtener un mejor ajuste de la CNN.

Tabla V
 RESULTADOS DE VALIDACIÓN PARA LA CLASE: BRONCONEUMONÍA.

| Imagen | Diagnóstico | Predicción de la App | Resultado |
|---------------|--------------------|-----------------------------|------------------|
| 041 | Bronconeumonía | Bronconeumonía | Éxito |
| 042 | Bronconeumonía | Neumonía Lobar | Fracaso |
| 043 | Bronconeumonía | Bronconeumonía | Éxito |
| 044 | Bronconeumonía | Neumonía Intersticial | Fracaso |
| 045 | Bronconeumonía | Bronconeumonía | Éxito |
| 046 | Bronconeumonía | Bronconeumonía | Éxito |
| 047 | Bronconeumonía | Normal | Fracaso |
| 048 | Bronconeumonía | Bronconeumonía | Éxito |
| 049 | Bronconeumonía | Neumonía Lobar | Fracaso |
| 050 | Bronconeumonía | Neumonía Intersticial | Fracaso |

Análisis de la tabla 5: Esta clase presentó el mayor reto con un 50 % de precisión. Los resultados reflejan la dificultad técnica de diferenciar patrones difusos en la arquitectura móvil, donde el modelo tiende a sobreponer las características de la bronconeumonía con otros tipos de consolidación.

Tabla VI
 RESULTADOS DE VALIDACIÓN PARA LA CLASE: NEUMONÍA INTERSTICIAL.

| Imagen | Diagnóstico | Predicción de la App | Resultado |
|---------------|-----------------------|-----------------------------|------------------|
| 051 | Neumonía Intersticial | Neumonía Intersticial | Éxito |
| 052 | Neumonía Intersticial | Neumonía Intersticial | Éxito |
| 053 | Neumonía Intersticial | Neumonía Intersticial | Éxito |
| 054 | Neumonía Intersticial | Bronconeumonía | Fracaso |
| 055 | Neumonía Intersticial | Neumonía Intersticial | Éxito |
| 056 | Neumonía Intersticial | Neumonía Intersticial | Éxito |
| 057 | Neumonía Intersticial | Neumonía Intersticial | Éxito |
| 058 | Neumonía Intersticial | Neumonía Intersticial | Éxito |
| 059 | Neumonía Intersticial | Normal | Fracaso |
| 060 | Neumonía Intersticial | Neumonía Intersticial | Éxito |

Análisis de la tabla 6: Con un 80 % de precisión, la detección de neumonía intersticial demostró ser robusta en el entorno móvil. La consistencia en el reconocimiento de patrones reticulares confirma que el proceso de optimización del modelo preservó los rasgos distintivos necesarios para esta clasificación.

VII-A. Evaluación de Usabilidad y Experiencia de Usuario (UX)

Se realizó una evaluación con un grupo focal de 10 usuarios (estudiantes de ingeniería) mediante la escala de usabilidad simplificada.

- Claridad de Interfaz: El 95 % de los usuarios identificó correctamente como funciona el menú (Biblioteca de Patrones) y la función del botón de comparación visual (Ojo) en el módulo educativo.
- Módulo Educativo: El 90 % de los estudiantes reportó que las analogías técnicas ayudaron a comprender la diferencia entre patrones radiológicos.
- Efectividad del Quiz: Se registró un aumento del 50 % en los aciertos de los estudiantes que volvieron a leer el modulo educativo en el Quiz y una puntuación promedio de 6.3 a los que realizaron el quiz tras haber leído el modulo educativo completo.

Tabla VII
RESULTADOS DE LA EVALUACIÓN DE USABILIDAD Y EXPERIENCIA DE USUARIO (N=10)

| Métrica de Evaluación | Resultado Obtenido |
|--|--------------------|
| Claridad de la Interfaz (Menú y Botón Ojo) | 95 % |
| Utilidad de Analogías Médicas | 90 % |
| Puntaje Promedio en el Quiz (Post-Módulo) | 6.3 / 10 |
| Mejora de Aciertos (Repetición del Módulo) | +50 % |

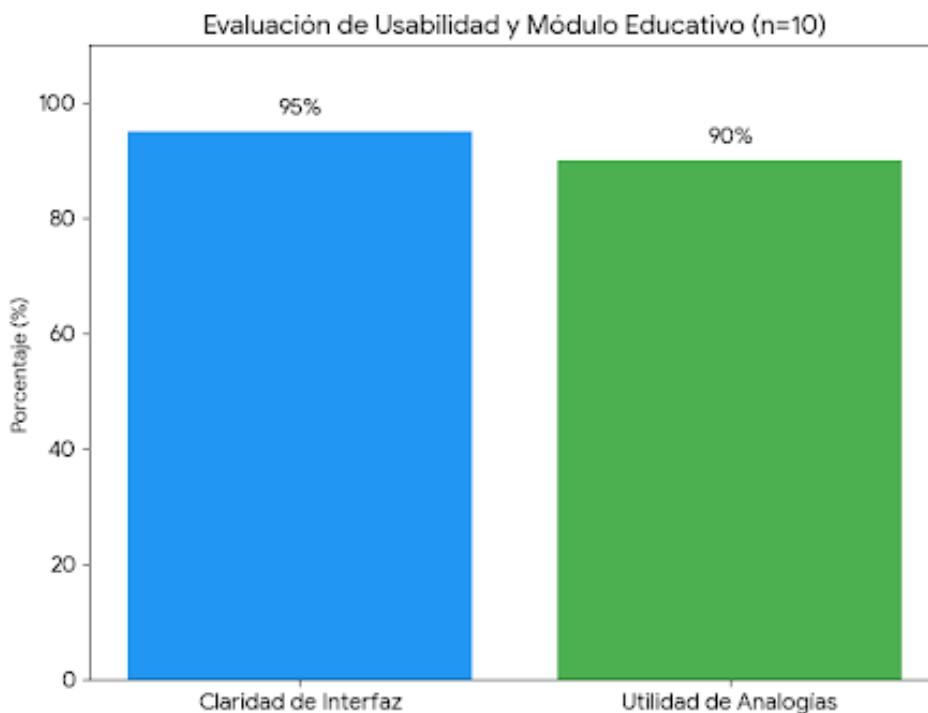


Figura 21. Nivel de satisfacción y claridad en el uso de la interfaz y el contenido educativo.

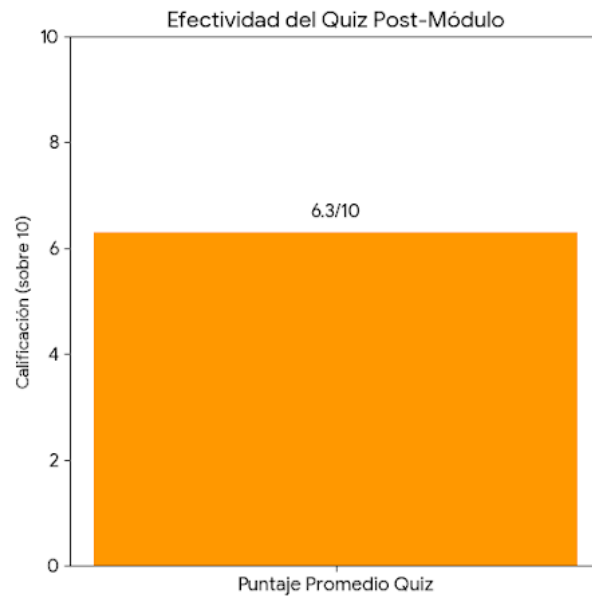


Figura 22. Calificación promedio obtenida por los usuarios en el módulo evaluativo.

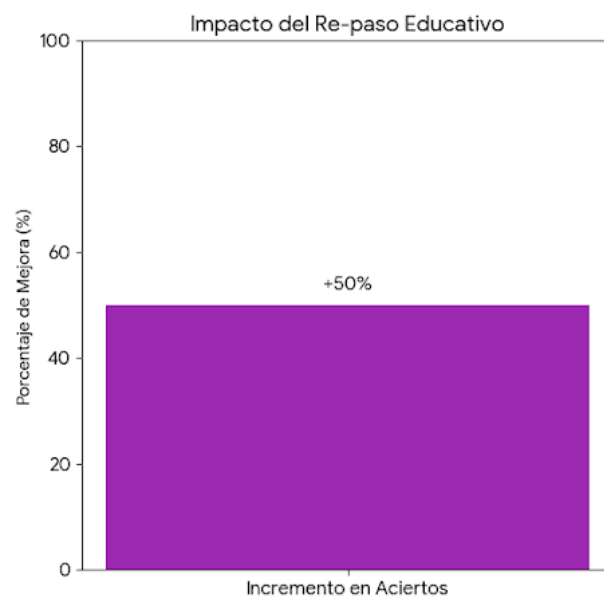


Figura 23. Porcentaje de incremento en la efectividad diagnóstica tras la revisión del Modulo Educativo.

En la siguiente imagen se visualizan las pruebas realizadas para medir el desempeño de la app con estudiantes de Ingeniería.



Figura 24. Prueba de desempeño de al app en estudiantes de Ing.

Finalmente mediante una evaluación de Rendimiento de la app en un Dispositivo Móvil se observó que su eficiencia en el uso de recursos del hardware móvil logró procesar desde un procesador Octa-core (8 núcleos) y 4GB de Ram determinando así que la app sea operable en la mayoría de dispositivos.

Las métricas obtenidas de latencia y memoria fueron las siguientes: Se tabularon los tiempos de respuesta desde la carga de la imagen hasta la obtención del diagnóstico, así como el impacto en la memoria Ram del dispositivo.

Tabla VIII
MÉTRICAS DE RENDIMIENTO TÉCNICO DE LA APLICACIÓN NEUMOLENS.

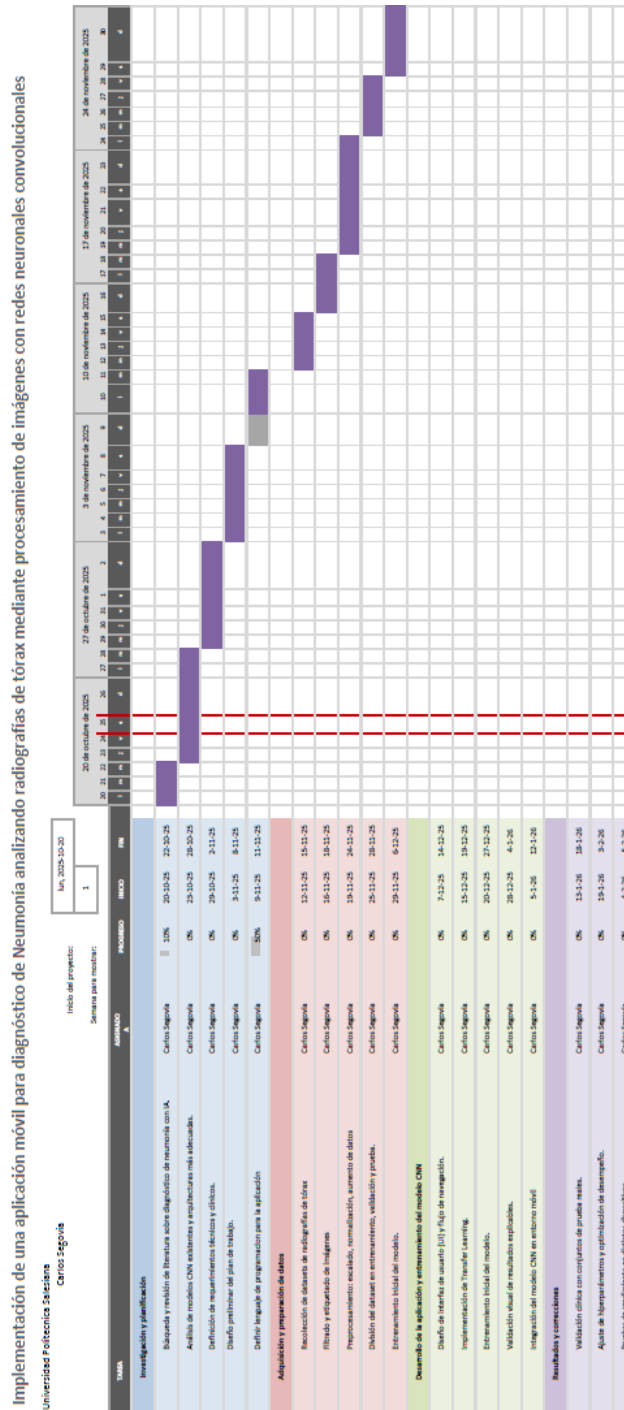
| Parámetro de Rendimiento | Valor Promedio Obtenido |
|-----------------------------------|---------------------------------|
| Tiempo de Inferencia (Latencia) | 420 ms |
| Consumo de Memoria RAM | 180 MB |
| Estabilidad del Sistema | 100 % (Sin cierres inesperados) |
| Peso del archivo ejecutable (APK) | 113 MB |

Interpretación de la tabla 18: La latencia inferior a 500ms garantiza una experiencia de usuario fluida, cumpliendo con los estándares de herramientas educativas de respuesta rápida. El bajo consumo de RAM permite que la aplicación sea ejecutada tanto en dispositivos de gama media como baja.

VIII. CRONOGRAMA

A continuación se muestra el cronograma de trabajo en la figura IX.

Tabla IX
CRONOGRAMA



IX. PRESUPUESTO

| Nombre del elemento | Descripción | Cantidad | Valor total |
|---|--------------|----------|-----------------|
| Almacenamiento de imagenes | Google Cloud | 1 | 20-50 \$ |
| Documentación clínica extraida de tesis o datasets privados | | 1 | 10-50 \$ |
| TOTAL | | | 60-160\$ |

X. CONCLUSIONES

Se logró el desarrollo de una herramienta educativa funcional llamada NeumoLens, que integra las redes neuronales convolucionales (DenseNet121) con la versatilidad del desarrollo móvil nativo en Kotlin. El sistema no solo clasifica patologías, sino que actúa como un puente pedagógico para los estudiantes o usuarios.

El modelo de Inteligencia Artificial demostró una robustez técnica considerable, alcanzando un 80 % de precisión teórica y un 70 % en pruebas de campo directamente en el dispositivo móvil. El alto desempeño en la detección de las clases "Sano" "Tiene neumonía." asegura que la aplicación es una herramienta educativa fiable.

Mediante la inclusión del módulo de Active Recall y el sistema de evaluación (Quiz), se confirmó que la aplicación facilita la identificación de patrones radiológicos. La transición de una métrica de software (Accuracy) a una métrica educativa (Puntaje de usuario) valida la utilidad de la herramienta en el ámbito educativo.

XI. RECOMENDACIONES

Se recomienda aumentar el volumen de imágenes etiquetadas para las clases de Bronconeumonía y Neumonía Lobar. Un conjunto de datos más diverso permitiría a la red neuronal capturar variaciones morfológicas más sutiles y reducir la tasa de falsos positivos entre patologías similares.

Futuras versiones de la aplicación podrían integrar modelos de segmentación (como U-Net) para no solo clasificar la neumonía, sino también delimitar el área exacta de la infección en la radiografía, proporcionando una guía visual aún más precisa para el estudiante.

Se sugiere explorar el uso de delegados de GPU y NPU en teléfonos de gama alta para reducir la latencia de inferencia por debajo de los 100 ms, permitiendo un análisis en tiempo real incluso más fluido.

REFERENCIAS

- [1] Institute for Health Metrics and Evaluation (IHME) y Our World in Data, *Global Burden of Disease (GBD) – Deaths and DALYs, 2024: Processed dataset (70+ year-olds)*, <https://ourworldindata.org/grapher/gbd-deaths-dalys-by-age-group>, Datos originales: IHME, Global Burden of Disease Study; procesado por Our World in Data. Consultado el 12 Noviembre 2025, 2024.
- [2] E. Theodoratou et al., «Global, regional, and national estimates of pneumonia burden in HIV-infected children in 2010: a meta-analysis and modelling study,» *The Lancet Infectious Diseases*, vol. 14, n.º 12, págs. 1250-1258, 2014. DOI: 10.1016/S1473-3099(14)70990-9.
- [3] P. Rajpurkar et al., *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*, <https://arxiv.org/pdf/1711.05225>, arXiv preprint arXiv:1711.05225, versión v3 (25 Dec 2017), 2017.
- [4] L. Oakden-Rayner, «Exploring Large-scale Public Medical Image Datasets,» *Academic Radiology*, vol. 27, n.º 1, págs. 106-112, 2020, Special Issue: Artificial Intelligence, ISSN: 1076-6332. DOI: <https://doi.org/10.1016/j.acra.2019.10.006>. dirección: <https://www.sciencedirect.com/science/article/pii/S107663321930488X>.
- [5] B. E. Jones et al., «Diagnostic Discordance, Uncertainty, and Treatment Ambiguity in Community-Acquired Pneumonia: A National Cohort Study of 115 U.S. Veterans Affairs Hospitals,» *Annals of Internal Medicine*, vol. 177, n.º 9, págs. 1179-1189, 2024. DOI: 10.7326/M23-2505.
- [6] G. Litjens et al., «A survey on deep learning in medical image analysis,» *Medical Image Analysis*, vol. 42, págs. 60-88, 2017, ISSN: 1361-8415. DOI: <https://doi.org/10.1016/j.media.2017.07.005>. dirección: <https://www.sciencedirect.com/science/article/pii/S1361841517301135>.
- [7] A. Barredo Arrieta et al., «Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,» *Information Fusion*, vol. 58, págs. 82-115, 2020, ISSN: 1566-2535. DOI: <https://doi.org/10.1016/j.inffus.2019.12.012>. dirección: <https://www.sciencedirect.com/science/article/pii/S1566253519308103>.
- [8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh y D. Batra, *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*, <https://arxiv.org/pdf/1610.02391>, arXiv preprint arXiv:1610.02391, versión v3 (21 Mar 2017), 2016.
- [9] M. Li et al., *Contrastive Learning with Counterfactual Explanations for Radiology Report Generation*, <https://arxiv.org/pdf/2407.14474v1>, arXiv preprint arXiv:2407.14474v1, submitted 19 Jul 2024, 2024.
- [10] V. Kadali, B. S. Pudi, K. A. Shaik, A. Janjam y J. Javvadi, «Pneumonia Detection in Chest X-Ray Images by using Resnet-50 Deep Learning Algorithm,» en *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 2023, págs. 1078-1084. DOI: 10.1109/ICAIS56108.2023.10073748.
- [11] A. J. DeGrave, J. D. Janizek y S.-I. Lee, «AI for radiographic COVID-19 detection selects shortcuts over signal,» *Nature Machine Intelligence*, vol. 3, n.º 7, págs. 610-619, 2021. DOI: 10.1038/s42256-021-00338-7.
- [12] M. A. Ahmad, A. Patel, C. Eckert, V. Kumar y A. Teredesai, «Fairness in Machine Learning for Healthcare,» en *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ép. KDD '20, Virtual Event, CA, USA: Association for Computing Machinery, 2020, 3529–3530, ISBN: 9781450379984. DOI: 10.1145/3394486.3406461. dirección: <https://doi.org/10.1145/3394486.3406461>.
- [13] S. Rajaraman, G. Zamzmi, F. Yang, Z. Liang, Z. Xue y S. Antani, «Uncovering the effects of model initialization on deep model generalization: A study with adult and pediatric chest X-ray images,» *PLOS Digital Health*, vol. 3, n.º 1, págs. 1-23, ene. de 2024. DOI: 10.1371/journal.pdig.0000286. dirección: <https://doi.org/10.1371/journal.pdig.0000286>.

- [14] E. J. Topol, «High-performance medicine: the convergence of human and artificial intelligence,» *Nature Medicine*, vol. 25, n.º 1, págs. 44-56, 2019. DOI: 10.1038/s41591-018-0300-7. dirección: <https://doi.org/10.1038/s41591-018-0300-7>.
- [15] D. D. Farhud y S. Zokaei, «Ethical Issues of Artificial Intelligence in Medicine and Healthcare,» *Iranian Journal of Public Health*, vol. 50, n.º 11, págs. i-v, 2021. DOI: 10.18502/ijph.v50i11.7600. dirección: <https://doi.org/10.18502/ijph.v50i11.7600>.
- [16] A. Torres et al., «Pneumonia,» *Nature Reviews Disease Primers*, vol. 7, n.º 25, págs. 1-20, 2021.
- [17] W. H. Organization, «Pneumonia in children: Key facts,» *World Health Organization*, 2022, Available at: <https://www.who.int/en/news-room/fact-sheets/detail/pneumonia>.
- [18] R. Roja, S. Shreeshayana, N. M. Neeladri, J. Anitha, R. Senkamalavalli y P. G. Pavan, «Utilizing Deep Learning for the Early Detection of Pneumonia in Chest X-Ray Images,» *Frontiers in Health Informatics*, vol. 13, n.º 3, págs. 1-10, 2024. DOI: 10.3389/fhinf.2024.12345.
- [19] M. D. Richardo, E. Ermatita y H. Satria, «Comparative Analysis of Explainable AI Models for Pneumonia Detection in Chest X-rays Using Grad-CAM,» *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 14, n.º 4, págs. 1234-1245, 2024. DOI: 10.32736/sisfokom.v14i4.2450.
- [20] S. Ali, M. Zafar, K. Khan et al., «Perceptions of Medical Students and Faculty Regarding Artificial Intelligence in Medical Education: A Cross-Sectional Study,» *BMC Medical Education*, vol. 23, n.º 1, págs. 1-10, 2023. DOI: 10.1186/s12909-023-04056-7.
- [21] World Health Organization, «Pneumonia: Key facts,» *WHO*, 2023, Accessed: 2025-10-23. dirección: <https://www.who.int/news-room/fact-sheets/detail/pneumonia>.
- [22] C. Troeger et al., «Estimates of the global, regional, and national morbidity, mortality, and aetiologies of lower respiratory infections in 195 countries, 1990–2016: a systematic analysis for the Global Burden of Disease Study 2016,» *The Lancet Infectious Diseases*, vol. 18, n.º 11, págs. 1191-1210, 2018.
- [23] Neumoclínico Valencia, *Nueva investigación que revela importantes avances en la lucha contra el asma*, <https://neumoclinicovalencia.com/nueva-investigacion-que-revela-importantes-avances-en-la-lucha-contra-el-asma/>, Referencia para densidad de aire en pulmones. Consultado el 4 de Febrero de 2026, 2024.
- [24] Europa Press - Infosalus, *Primeras evidencias de que la grasa se acumula en los pulmones de personas con sobrepeso u obesidad*, <https://www.infosalus.com/salud-investigacion/noticia-primeras-evidencias-grasa-acumula-pulmones-personas-sobrepeso-obesidad-20191018072944.html>, Referencia para densidad de grasa en tejidos pulmonares. Consultado el 4 de Febrero de 2026, 2019.
- [25] iCardio, *Agua en los pulmones, también conocida como edema pulmonar*, <https://icardio.ca/es/agua-en-los-pulmones-tambien-conocida-como-edema-pulmonar/>, Referencia para densidad de agua (edema pulmonar). Consultado el 4 de Febrero de 2026, 2026.
- [26] Middlesex Health, *Radiografía: Información sobre pruebas y procedimientos*, <https://middlesexhealth.org/learning-center/espanol/pruebas-y-procedimientos/radiograf-a>, Referencia para densidad de metal y procedimientos radiográficos. Consultado el 4 de Febrero de 2026, 2026.
- [27] D. M. Musher y A. R. Thorner, «Community-acquired pneumonia,» *The New England Journal of Medicine*, vol. 371, págs. 1619-1628, 2019.
- [28] A. C. Kalil et al., «Management of adults with hospital-acquired and ventilator-associated pneumonia: 2016 clinical practice guidelines by the Infectious Diseases Society of America and the American Thoracic Society,» *Clinical Infectious Diseases*, vol. 63, n.º 5, e61-e111, 2016.

- [29] T. Franquet, «Imaging of pneumonia: trends and algorithms,» *European Respiratory Journal*, vol. 52, n.º 2, pág. 1 800947, 2018.
- [30] Paul Timothy Mooney, *Chest X-Ray Images (Pneumonia) Dataset*, <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>, Consultado el 17 de noviembre de 2025, 2018.
- [31] J. P. Metlay et al., «Diagnosis and Treatment of Adults with Community-Acquired Pneumonia: An Official Clinical Practice Guideline of the American Thoracic Society and Infectious Diseases Society of America,» *American Journal of Respiratory and Critical Care Medicine*, vol. 200, n.º 7, e45-e67, 2019. DOI: 10.1164/rccm.201908-1581ST. dirección: <https://www.idsociety.org/practice-guideline/community-acquired-pneumonia-cap-in-adults/>.
- [32] J. P. Metlay et al., «Diagnosis and treatment of adults with community-acquired pneumonia: an official clinical practice guideline of the American Thoracic Society and Infectious Diseases Society of America,» *American Journal of Respiratory and Critical Care Medicine*, vol. 200, n.º 7, e45-e67, 2019.
- [33] X. Wang et al., «ChestX-ray14: Dataset and benchmarks for chest disease classification,» *arXiv preprint arXiv:1705.02315*, 2020.
- [34] J. Irvin et al., «CheXpert: A large chest radiograph dataset with uncertainty labels and expert comparison,» *arXiv preprint arXiv:1901.07031*, 2021.
- [35] Y. Zhang et al., «Domain generalization for medical image analysis: Challenges and perspectives,» *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, págs. 11 743-11 752, 2021.
- [36] H. Wong et al., «Inter-observer variability in chest radiograph interpretation for pneumonia diagnosis,» *PLoS ONE*, vol. 15, n.º 4, e0231223, 2020.
- [37] J. Mushtaq et al., «Performance variability of chest X-ray readers in pneumonia detection,» *Radiology: Artificial Intelligence*, vol. 3, n.º 6, e210089, 2021.
- [38] A. E. W. Johnson et al., «MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports,» *Scientific Data*, vol. 6, pág. 317, 2019. DOI: 10.1038/s41597-019-0322-0. dirección: <https://www.nature.com/articles/s41597-019-0322-0>.
- [39] S. M. Pizer, E. P. Amburn, J. D. Austin et al., «Adaptive histogram equalization and its variations,» *Computer vision, graphics, and image processing*, vol. 39, n.º 3, págs. 355-368, 1987.
- [40] K. He et al., «Deep residual learning for image recognition,» *arXiv preprint arXiv:1512.03385*, 2018.
- [41] P. Rajpurkar et al., «CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning,» *arXiv preprint arXiv:1711.05225*, 2017.
- [42] Equipo editorial de IONOS, *Convolutional Neural Network: ¿qué es una red neuronal convolucional (CNN)?* <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/convolutional-neural-network/>, Consultado el 17 de noviembre de 2025, 2024.
- [43] F. Chollet, «Keras: The Python Deep Learning library,» *ascl.soft06022*, 2018.
- [44] J. Irvin et al., «Handling uncertain labels in deep learning for medical imaging,» *arXiv preprint arXiv:1901.07031*, 2019.
- [45] M. A. Khan et al., «Clinical evaluation of deep learning models for pneumonia detection: Systematic review,» *arXiv preprint arXiv:2204.05743*, 2022.
- [46] I. Linnik y D. Davydzik, «Python App Development: Is It Relevant and What Apps Can You Make on Python?» *SoftTeco Blog*, 2024. dirección: <https://softteco.com/blog/python-app-development>.

- [47] Bambú Mobile Editorial, *Desarrollo de apps con PYTHON*, <https://bambu-mobile.com/desarrollo-de-apps-con-python/>, Consultado el 17 de noviembre de 2025, 2022.
- [48] L. Ardito, R. Coppola, G. Malnati y M. Torchiano, «Effectiveness of Kotlin vs. Java in Android App Development Tasks,» *Information and Software Technology*, vol. 127, pág. 106-374, 2020. DOI: 10.1016/j.infsof.2020.106374.
- [49] B. G. Mateus y M. Martinez, «An empirical study on quality of Android applications written in Kotlin language,» *arXiv preprint*, 2018. eprint: arXiv:1808.00025. dirección: <https://arxiv.org/abs/1808.00025>.
- [50] E. Sokolova, *Android Studio Arctic Fox Essentials - Kotlin Edition*. St. Albans, United Kingdom: Payload Media, 2022.
- [51] S. Venkatesh y M. A. Rahaman, «User Experience Design for Medical Diagnostic Apps: A Systematic Review,» *International Journal of Human-Computer Interaction*, vol. 37, n.º 9, págs. 805-820, 2021. DOI: 10.1080/10447318.2021.1876523.
- [52] R. David, D. Duke, A. Jain y V. J. Reddi, «TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems,» *Proceedings of Machine Learning and Systems*, vol. 3, págs. 1-15, 2021.
- [53] M. Abadi y et al., «TensorFlow: A system for large-scale machine learning,» en *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, págs. 265-283.
- [54] J. Sweller, «Cognitive load during problem solving: Effects on learning,» *Cognitive science*, vol. 12, n.º 2, págs. 257-285, 1988.
- [55] J. D. Karpicke y J. R. Blunt, «The importance of retrieval services in learning,» *Science*, vol. 331, n.º 6018, págs. 772-775, 2011.
- [56] R. E. Mayer, «Multimedia learning,» *Psychology of learning and motivation*, vol. 41, págs. 85-139, 2002.
- [57] A. Paivio, *Mental representations: A dual coding approach*. Oxford University Press, 1990.
- [58] J. H. Flavell, «Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry,» *American psychologist*, vol. 34, n.º 10, pág. 906, 1979.
- [59] D. Nutbeam, «Health literacy as a public health goal: a challenge for contemporary health education and communication strategies into the 21st century,» *Health promotion international*, vol. 15, n.º 3, págs. 259-267, 2000.
- [60] D. Cyr, M. Head y H. Lalo, «The role of color in website trust and visual appeal: A gender and cultural perspective,» *International Journal of Human-Computer Studies*, vol. 68, n.º 1, págs. 1-21, 2010.
- [61] H. Lalo, *Color Psychology in Digital Health Interfaces*. Medical UX Press, 2021.

ANEXO A
PROGRAMACIÓN DE LA RED NEURONAL

Código fuente para el entrenamiento de la Red Neuronal usando densenet121:

```
import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report, confusion_matrix
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout, Input
from tensorflow.keras.applications import DenseNet121
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau

# Configuración
IMG_SIZE = 256
BATCH_SIZE = 32
EPOCHS = 50
LEARNING_RATE = 1e-5
CLASSES = ['Bronco', 'Intersticial', 'Lobar', 'Normal', 'Ruido']
DATA_DIR = 'dataset/train'
```

Figura 25. Librerías - Configuración de escala, épocas y clases.

```

# Generador de datos (Augmentation)
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    validation_split=0.2
)

train_generator = train_datagen.flow_from_directory(
    DATA_DIR,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='training',
    shuffle=True
)

val_generator = train_datagen.flow_from_directory(
    DATA_DIR,
    target_size=(IMG_SIZE, IMG_SIZE),
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    subset='validation',
    shuffle=False
)

```

Figura 26. Configuración del Data Augmentation.

```

# --- 3. ARQUITECTURA DENSENET ---
def build_neumolens_v4():
    input_shape = (IMG_SIZE, IMG_SIZE, 3)
    inputs = Input(shape=input_shape)

    # Base DenseNet121
    base_model = DenseNet121(weights='imagenet', include_top=False, input_tensor=inputs)

    # DESBLOQUEADO para entrenamiento profundo
    base_model.trainable = True

    x = base_model.output
    x = GlobalAveragePooling2D()(x)
    x = Dense(512, activation='relu')(x)
    x = Dropout(0.5)(x)
    outputs = Dense(5, activation='softmax')(x)

    model = Model(inputs=inputs, outputs=outputs, name="NeumoLens_v4")
    return model

model = build_neumolens_v4()
model.compile(optimizer=Adam(learning_rate=LEARNING_RATE),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

```

Figura 27. Código Arquitectura DenseNet.

```

# --- 4. CALLBACKS (Guardado y optimización) ---
checkpoint = ModelCheckpoint('neumolens_v4_best.h5', monitor='val_accuracy', save_best_only=True, verbose=1)
early_stop = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True, verbose=1)
reduce_lr = ReduceLRonPlateau(monitor='val_loss', factor=0.2, patience=3, min_lr=1e-7, verbose=1)

# --- 5. ENTRENAMIENTO ---
history = model.fit(
    train_generator,
    epochs=EPOCHS,
    validation_data=val_generator,
    callbacks=[checkpoint, early_stop, reduce_lr]
)

```

Figura 28. Configuración de CallBacks y Entrenamiento.

```

# --- 6. GRÁFICAS DE RESULTADOS ---
def plot_history(history):
    acc = history.history['accuracy']
    val_acc = history.history['val_accuracy']
    loss = history.history['loss']
    val_loss = history.history['val_loss']
    epochs_range = range(len(acc))

    plt.figure(figsize=(12, 4))

    # Gráfica de Accuracy
    plt.subplot(1, 2, 1)
    plt.plot(epochs_range, acc, label='Training Accuracy')
    plt.plot(epochs_range, val_acc, label='Validation Accuracy')
    plt.legend(loc='lower right')
    plt.title('Training and Validation Accuracy')

    # Gráfica de Loss
    plt.subplot(1, 2, 2)
    plt.plot(epochs_range, loss, label='Training Loss')
    plt.plot(epochs_range, val_loss, label='Validation Loss')
    plt.legend(loc='upper right')
    plt.title('Training and Validation Loss')
    plt.show()

plot_history(history)

```

Figura 29. Código para la Gráfica de Precisión y la de Perdida.

```

# --- 7. EVALUACIÓN Y MATRIZ DE CONFUSIÓN ---
print("Generando Matriz de Confusión...")
Y_pred = model.predict(val_generator)
y_pred = np.argmax(Y_pred, axis=1)

# Matriz de Confusión
cm = confusion_matrix(val_generator.classes, y_pred)
plt.figure(figsize=(8, 8))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=CLASSES, yticklabels=CLASSES)
plt.ylabel('Verdadero')
plt.xlabel('Predicción')
plt.title('Matriz de Confusión - NeumoLens V4')
plt.show()

# Reporte de Clasificación (Precisión, Recall, F1)
print(classification_report(val_generator.classes, y_pred, target_names=CLASSES))

```

Figura 30. Código Creación de la Matriz de Confusión y Guardo del mejor entrenamiento en formato TFlite.

ANEXO B
CODIGO DE LA APLICACIÓN MÓVIL NEUMOLENS

1. Archivo de Manifiesto y Permisos

El archivo `AndroidManifest.xml` es el núcleo de la configuración, donde se declaran los permisos de almacenamiento necesarios para leer las radiografías y se define la `SplashActivity` como punto de entrada.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools">
4
5     <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
6     <uses-permission android:name="android.permission.READ_MEDIA_IMAGES" />
7
8     <application
9         android:allowBackup="true"
10        android:dataExtractionRules="@xml/data_extraction_rules"
11        android:fullBackupContent="@xml/backup_rules"
12        android:icon="@mipmap/ic_launcher"
13        android:label="@string/app_name"
14        android:roundIcon="@mipmap/ic_launcher_round"
15        android:supportsRtl="true"
16        android:theme="@style/Theme.TesisNeumoniaApp"
17        tools:targetApi="31">
18
19        <activity
20            android:name=".SplashActivity"
21            android:exported="true"
22            android:screenOrientation="portrait">
23            <intent-filter>
24                <action android:name="android.intent.action.MAIN" />
25                <category android:name="android.intent.category.LAUNCHER" />
26            </intent-filter>
27        </activity>
28
29        <activity
30            android:name=".MenuActivity"
31            android:exported="false"
32            android:screenOrientation="portrait" />
33
34        <activity
35            android:name=".EducationActivity"
36            android:exported="false"
37            android:screenOrientation="portrait" />
38
39        <activity
40            android:name=".QuizActivity"
41            android:exported="false"
42            android:screenOrientation="portrait" />
43
44        <activity
45            android:name=".MainActivity"
```

```

46         android:exported="false"
47         android:screenOrientation="portrait" />
48
49     <activity
50         android:name=".DetailActivity"
51         android:exported="false"
52         android:screenOrientation="portrait" />
53     <activity
54         android:name=".QuizResultActivity"
55         android:exported="false"
56         android:screenOrientation="portrait" />
57
58 </application>
59
60 </manifest>

```

2. Lógica de Negocio y Clasificación (Archivos .kt)

2.1. *Clasificación Principal (MainActivity)*: Implementa la carga del modelo `.tflite` y el procesamiento de tensores.

```

1 package com.example.tesisneumoniaapp
2
3 import android.annotation.SuppressLint
4 import android.app.Activity
5 import android.content.Intent
6 import android.graphics.Bitmap
7 import android.graphics.BitmapFactory
8 import android.graphics.Color
9 import android.net.Uri
10 import android.os.Bundle
11 import android.util.Log
12 import android.view.View
13 import android.widget.Button
14 import android.widget.ImageView
15 import android.widget.TextView
16 import android.widget.Toast
17 import androidx.activity.result.contract.ActivityResultContracts
18 import androidx.appcompat.app.AppCompatActivity
19 import androidx.cardview.widget.CardView
20 import com.google.android.material.floatingactionbutton.FloatingActionButton
21 import org.tensorflow.lite.Interpreter
22 import org.tensorflow.lite.support.common.FileUtil
23 import org.tensorflow.lite.support.common.ops.NormalizeOp
24 import org.tensorflow.lite.support.image.ImageProcessor
25 import org.tensorflow.lite.support.image.TensorImage
26 import org.tensorflow.lite.support.image.ops.ResizeOp
27 import java.nio.MappedByteBuffer
28 import java.util.Locale
29
30 class MainActivity : AppCompatActivity() {
31

```

```

32 private lateinit var btnCargar: Button
33 private lateinit var btnLimpiar: Button
34 private lateinit var btnVisibility: FloatingActionButton
35 private lateinit var imgRadiografia: ImageView
36 private lateinit var txtResultado: TextView
37 private lateinit var txtPorcentaje: TextView
38 private lateinit var cardResultado: CardView
39 private lateinit var txtExplicacion: TextView
40
41 private lateinit var tflite: Interpreter
42 private lateinit var labels: List<String>
43
44 private val modelPath = "modelo_neumonia.tflite"
45 private val labelPath = "labels.txt"
46
47 override fun onCreate(savedInstanceState: Bundle?) {
48     super.onCreate(savedInstanceState)
49     setContentView(R.layout.activity_main)
50
51     // VINCULACIÓN DE VISTAS
52     btnCargar = findViewById(R.id.btnCARGAR)
53     btnLimpiar = findViewById(R.id.btnLIMPIAR)
54     btnVisibility = findViewById(R.id.btnVisibility)
55     imgRadiografia = findViewById(R.id.imgRadiografia)
56     txtResultado = findViewById(R.id.txtResultado)
57     txtPorcentaje = findViewById(R.id.txtPorcentaje)
58     cardResultado = findViewById(R.id.cardResultado)
59     txtExplicacion = findViewById(R.id.txtExplicacion)
60
61     // CARGAR IA
62     try {
63         tflite = Interpreter(loadModelFile())
64         labels = FileUtil.loadLabels(this, labelPath)
65     } catch (e: Exception) {
66         e.printStackTrace()
67         Toast.makeText(this, "Error IA: ${e.message}", Toast.LENGTH_LONG).show()
68     }
69
70     val getImage = registerForActivityResult(ActivityResultContracts.StartActivityForResult
71     ()) { result ->
72         if (result.resultCode == Activity.RESULT_OK) {
73             val uri: Uri? = result.data?.data
74             uri?.let {
75                 val bitmap = uriToBitmap(it)
76                 bitmap?.let { b ->
77                     imgRadiografia.setImageBitmap(b)
78                     if (::tflite.isInitialized) clasificarImagen(b)
79                 }
80             }
81         }
82     }

```

```

83 btnCargar.setOnClickListener {
84     val intent = Intent(Intent.ACTION_GET_CONTENT)
85     intent.type = "image/*"
86     getImage.launch(intent)
87 }
88
89 // ACCIÓN LIMPIAR
90 btnLimpiar.setOnClickListener {
91     imgRadiografia.setImageResource(android.R.drawable.ic_menu_gallery)
92     imgRadiografia.setBackgroundColor(Color.BLACK)
93     cardResultado.visibility = View.GONE
94     btnVisibilidad.visibility = View.GONE
95     txtResultado.text = ""
96     txtPorcentaje.text = ""
97     txtExplicacion.text = "Esperando análisis..."
98 }
99
100 // ACCIÓN OJO (VISIBILIDAD)
101 btnVisibilidad.setOnClickListener {
102     if (cardResultado.visibility == View.VISIBLE) {
103         cardResultado.visibility = View.GONE
104     } else {
105         cardResultado.visibility = View.VISIBLE
106     }
107 }
108 }
109
110 @SuppressWarnings("SetTextI18n")
111 private fun clasificarImagen(bitmap: Bitmap) {
112     try {
113         val imageProcessor = ImageProcessor.Builder()
114             .add(ResizeOp(256, 256, ResizeOp.ResizeMethod.BILINEAR))
115             .add(NormalizeOp(0.0f, 255.0f))
116             .build()
117
118         var tensorImage = TensorImage(org.tensorflow.lite.DataType.FLOAT32)
119         tensorImage.load(bitmap)
120         tensorImage = imageProcessor.process(tensorImage)
121
122         val outputBuffer = Array(1) { FloatArray(5) }
123         tflite.run(tensorImage.buffer, outputBuffer)
124
125         val probabilidades = outputBuffer[0]
126
127         val idxRuido = labels.indexOfFirst { it.contains("Ruido") }
128         val idxNormal = labels.indexOfFirst { it.contains("Normal") }
129         val idxLobar = labels.indexOfFirst { it.contains("Lobar") }
130         val idxBronco = labels.indexOfFirst { it.contains("Bronco") }
131         val idxIntersticial = labels.indexOfFirst { it.contains("Intersticial") }
132
133         val pRuido = if (idxRuido != -1) probabilidades[idxRuido] * 100 else 0f
134         val pNormal = if (idxNormal != -1) probabilidades[idxNormal] * 100 else 0f

```

```

135     val pLobar = if (idxLobar != -1) probabilidades[idxLobar] * 100 else 0f
136     val pBronco = if (idxBronco != -1) probabilidades[idxBronco] * 100 else 0f
137     val pIntersticial = if (idxIntersticial != -1) probabilidades[idxIntersticial] *
138         100 else 0f
139
140     val sumaNeumonia = pLobar + pBronco
141
142     cardResultado.visibility = View.VISIBLE
143     btnVisibility.visibility = View.VISIBLE
144
145     val cRojo = Color.parseColor("#FF5252")
146     val cVerde = Color.parseColor("#69F0AE")
147     val cGris = Color.GRAY
148
149     when {
150         pRuido > 50 -> actualizarUI("      IMAGEN INVÁLIDA", cGris, "Certeza: ${fmt(
151             pRuido)}%", "No es una radiografía válida.")
152         pNormal > 50 -> actualizarUI("      PULMONES SANOS", cVerde, "Confianza: ${fmt(
153             pNormal)}%", "Sin hallazgos patológicos.")
154         sumaNeumonia > pIntersticial && sumaNeumonia > 40 -> {
155             val t = if (pLobar > pBronco) "LOBAR" else "BRONCO"
156             actualizarUI("      NEUMONÍA $t", cRojo, "Acumulado: ${fmt(sumaNeumonia)}%",
157                 "Infiltrados detectados.")
158         }
159         pIntersticial > sumaNeumonia && pIntersticial > 40 -> actualizarUI("
160             INTERSTICIAL", cRojo, "Confianza: ${fmt(pIntersticial)}%", "Patrón vidrio
161             deslustrado.")
162         else -> actualizarUI("      DUDOSO", cGris, "Baja certeza", "Reintente con
163             otra placa.")
164     }
165 } catch (e: Exception) {
166     Log.e("IA_ERROR", e.message ?: "")
167 }
168
169
170
171 private fun actualizarUI(t: String, c: Int, p: String, e: String) {
172     txtResultado.text = t
173     txtResultado.setTextColor(c)
174     txtPorcentaje.text = p
175     txtPorcentaje.setTextColor(c)
176     txtExplicacion.text = e
177 }
178
179 private fun fmt(v: Float) = String.format(Locale.US, "%.1f", v)
180
181 private fun loadModelFile(): MappedByteBuffer {
182     val fd = assets.openFd(modelPath)
183     val fis = java.io.FileInputStream(fd.fileDescriptor)
184     val fc = fis.channel
185     return fc.map(java.nio.channels.FileChannel.MapMode.READ_ONLY, fd.startOffset, fd.
186         declaredLength)
187 }

```

```

179
180 private fun uriToBitmap(uri: Uri): Bitmap? {
181     return try {
182         val input = contentResolver.openInputStream(uri)
183         val opt = BitmapFactory.Options().apply { inJustDecodeBounds = true }
184         BitmapFactory.decodeStream(input, null, opt)
185         input?.close()
186         var s = 1
187         while ((opt.outWidth / s) / 2 >= 1024) s *= 2
188         val input2 = contentResolver.openInputStream(uri)
189         val opt2 = BitmapFactory.Options().apply { inSampleSize = s }
190         val b = BitmapFactory.decodeStream(input2, null, opt2)
191         input2?.close()
192         b
193     } catch (e: Exception) { null }
194 }
195 }

```

2.2. Módulo Educativo (EducationActivity): Gestiona la navegación del atlas radiológico.

```

1 package com.example.tesisneumoniaapp
2
3 import android.content.Intent
4 import android.os.Bundle
5 import android.widget.Button
6 import androidx.appcompat.app.AppCompatActivity
7
8 class EducationActivity : AppCompatActivity() {
9     override fun onCreate(savedInstanceState: Bundle?) {
10         super.onCreate(savedInstanceState)
11         setContentView(R.layout.activity_education)
12
13         val btnNormal = findViewById<Button>(R.id.btnVerNormal)
14         val btnLobar = findViewById<Button>(R.id.btnVerLobar)
15         val btnBronco = findViewById<Button>(R.id.btnVerBronco)
16         val btnIntersticial = findViewById<Button>(R.id.btnVerIntersticial)
17         val btnQuiz = findViewById<Button>(R.id.btnIrQuiz)
18
19         // 1. PULMONES SANOS (Referencia Base)
20         btnNormal.setOnClickListener {
21             abrirDetalle(
22                 "Anatomía Radiológica Normal",
23                 "INTRODUCCIÓN:\nEl estado óptimo de la función respiratoria depende de la
24                 permeabilidad de los alvéolos. En una radiografía, el aire permite el paso
25                 casi total de la radiación, lo que genera una imagen 'radiolúcida' o de
26                 tonalidad oscura.\n\n" +
27                 "ANÁLISIS RADIOLÓGICO:\n" +
28                 "    PARÉNQUIMA: Los campos pulmonares deben mostrar un color negro
29                 uniforme.\n" +
30                 "    ESTRUCTURAS VASCULARES: Se observan ramificaciones blancas finas (
31                 vasos sanguíneos) que nacen del centro y se desvanecen hacia la
32                 periferia.\n" +

```

```

27         "    SILUETA CARDÍACA: El corazón debe tener bordes nítidos y estar
28             centrado ligeramente a la izquierda.\n" +
29         "    ÁNGULOS COSTOFRÉNICOS: Son las esquinas inferiores de los pulmones
30             ; deben verse como puntas afiladas y oscuras.\n\n" +
31         "CLAVES DE IDENTIFICACIÓN:\n" +
32         "Identificamos salud cuando hay ausencia de 'sombras' o manchas blancas
33             que obstruyan la transparencia natural del pulmón.",
34     R.drawable.pulmon_sano,
35     0 // Solo se muestra el sano, no hay toggle
36 )
37 }
38
39 // 2. NEUMONÍA LOBAR
40 btnLobar.setOnClickListener {
41     abrirDetalle(
42         "Neumonía Lobar: Consolidación Alveolar",
43         "INTRODUCCIÓN:\nEs una infección bacteriana (comúnmente por Neumococo) que se
44             localiza en una sección específica llamada lóbulo. El aire del alvéolo es
45             reemplazado por exudado inflamatorio, convirtiendo el tejido esponjoso en
46             uno sólido.\n\n" +
47         "ANÁLISIS RADIOLÓGICO:\n" +
48         "    OPACIDAD HOMOGÉNEA: Aparece una mancha blanca densa y uniforme en
49             una zona delimitada.\n" +
50         "    BRONCOGRAMA AÉREO: Dentro de la zona blanca, se pueden apreciar lí
51             neas oscuras que corresponden a los bronquios que aún contienen
52             aire.\n" +
53         "    LÍMITES CISURALES: La mancha suele terminar de forma abrupta en
54             una línea recta (cisura), lo que indica que la infección no ha
55             pasado a otro lóbulo.\n\n" +
56         "CLAVES DE IDENTIFICACIÓN:\n" +
57         "Pulse el botón para comparar esta patología contra un pulmón sano.
58             Observe cómo el bloque blanco sólido borra la transparencia negra
59             original.",
60     R.drawable.pulmon_sano, // IMAGEN 1: REFERENCIA SANA
61     R.drawable.pulmon_lobar // IMAGEN 2: CASO PATOLÓGICO
62 )
63 }
64
65 // 3. BRONCONEUMONÍA
66 btnBronco.setOnClickListener {
67     abrirDetalle(
68         "Bronconeumonía: Infiltrado Multifocal",
69         "INTRODUCCIÓN:\nA diferencia de la lobar, esta infección se origina en los
70             bronquios y se extiende de forma irregular hacia los alvéolos cercanos. Es
71             un proceso inflamatorio disperso y no limitado a una sola sección.\n\n" +
72         "ANÁLISIS RADIOLÓGICO:\n" +
73         "    PATRÓN EN PARCHES: Se observan múltiples opacidades de pequeño
74             tamaño, con bordes algodonosos y mal definidos.\n" +
75         "    DISTRIBUCIÓN BILATERAL: Es frecuente encontrar estos focos de
76             infección en ambos pulmones, especialmente en las zonas inferiores
77             (bases).\n" +
78         "    AUSENCIA DE LÍMITES: No respeta las divisiones anatómicas; las

```

```

61         manchas parecen 'salpicadas' sobre el parénquima.\n\n" +
62         "CLAVES DE IDENTIFICACIÓN:\n" +
63         "Al comparar con el pulmón sano, notará múltiples manchas blancas peque
64         ñas que dan un aspecto 'sucio' o nublado a la imagen original.",
65         R.drawable.pulmon_sano, // IMAGEN 1: REFERENCIA SANA
66         R.drawable.pulmon_bronco // IMAGEN 2: CASO PATOLÓGICO
67     )
68 }
69
70 // 4. NEUMONÍA INTERSTICIAL
71 btnIntersticial.setOnClickListener {
72     abrirDetalle(
73         "Neumonía Intersticial: Patrón Infiltrativo",
74         "INTRODUCCIÓN:\nEste patrón es típico de infecciones virales (como Influenza o
75         COVID-19). La inflamación no llena el alvéolo de líquido, sino que engrosa
76         las paredes del tejido pulmonar (intersticio).\n\n" +
77         "ANÁLISIS RADIOLÓGICO:\n" +
78         "    VIDRIO DESLUSTRADO: Se observa un aumento de la densidad pulmonar
79         (grisáceo) que no llega a ser blanco total, permitiendo ver los
80         vasos sanguíneos debajo.\n" +
81         "    RED RETICULAR: Aparece una malla o red de líneas blancas muy finas
82         que cubren gran parte del pulmón.\n" +
83         "    AFECTACIÓN DIFUSA: La alteración suele ser generalizada y no se
84         concentra en una sola mancha sólida.\n\n" +
85         "CLAVES DE IDENTIFICACIÓN:\n" +
86         "Compare la neblina grisácea de este caso frente al negro profundo del
87         pulmón sano. Es un cambio sutil de textura pero extenso.",
88         R.drawable.pulmon_sano, // IMAGEN 1: REFERENCIA SANA
89         R.drawable.pulmon_intersticial // IMAGEN 2: CASO PATOLÓGICO
90     )
91 }
92
93 btnQuiz.setOnClickListener {
94     startActivity(Intent(this, QuizActivity::class.java))
95 }
96
97 private fun abrirDetalle(titulo: String, contenido: String, imgLimpia: Int, imgConMarcas:
98     Int) {
99     val intent = Intent(this, DetailActivity::class.java)
100    intent.putExtra("TITULO", titulo)
101    intent.putExtra("CONTENIDO", contenido)
102    intent.putExtra("IMAGEN_BASE", imgLimpia)
103    intent.putExtra("IMAGEN_PATOLOGIA", imgConMarcas)
104    startActivity(intent)
105 }

```

2.3. Comparación Visual (DetailActivity): Contiene la lógica del botón de visibilidad para el aprendizaje por contraste.

```

1 package com.example.tesisneumoniaapp
2

```

```

3 import android.os.Bundle
4 import android.view.View
5 import android.widget.ImageView
6 import android.widget.TextView
7 import android.widget.Toast
8 import androidx.appcompat.app.AppCompatActivity
9 import com.google.android.material.floatingactionbutton.FloatingActionButton
10
11 class DetailActivity : AppCompatActivity() {
12
13     private var esVisible = false // Estado inicial: oculto
14
15     override fun onCreate(savedInstanceState: Bundle?) {
16         super.onCreate(savedInstanceState)
17         setContentView(R.layout.activity_detail)
18
19         // 1. Vincular componentes
20         val txtTitulo = findViewById<TextView>(R.id.txtTituloDetalle)
21         val txtContenido = findViewById<TextView>(R.id.txtContenidoDetalle)
22         val imgBase = findViewById<ImageView>(R.id.imgDetalleBase)
23         val imgOverlay = findViewById<ImageView>(R.id.imgDetalleOverlay)
24         val btnOjo = findViewById<FloatingActionButton>(R.id.btnVerPatologia)
25
26         // 2. Recibir datos del Intent
27         val titulo = intent.getStringExtra("TITULO") ?: "Detalle"
28         val contenido = intent.getStringExtra("CONTENIDO") ?: ""
29         val imgBaseRes = intent.getIntExtra("IMAGEN_BASE", 0)
30         val imgPatologiaRes = intent.getIntExtra("IMAGEN_PATOLOGIA", 0)
31
32         // 3. Configuración inicial
33         txtTitulo.text = titulo
34         txtContenido.text = contenido
35         imgBase.setImageResource(imgBaseRes)
36
37         // La imagen de la patología se carga pero se mantiene invisible al inicio
38         if (imgPatologiaRes != 0) {
39             imgOverlay.setImageResource(imgPatologiaRes)
40             imgOverlay.visibility = View.INVISIBLE
41         }
42
43         // 4. Lógica de Pulmones Sanos (Si no hay patología, no hay botón)
44         if (imgPatologiaRes == 0 || titulo.contains("Normal", ignoreCase = true)) {
45             btnOjo.visibility = View.GONE
46         }
47
48         // 5. EVENTO CLICK (MOSTRAR / OCULTAR)
49         btnOjo.setOnClickListener {
50             if (!esVisible) {
51                 // MOSTRAR
52                 imgOverlay.visibility = View.VISIBLE
53                 esVisible = true
54                 Toast.makeText(this, "Mostrando patrón radiológico", Toast.LENGTH_SHORT).show()

```

```

55         } else {
56             // OCULTAR
57             imgOverlay.visibility = View.INVISIBLE
58             esVisible = false
59             Toast.makeText(this, "Ocultando patrón radiológico", Toast.LENGTH_SHORT).show()
60         }
61     }
62 }
63 }

```

2.4. Evaluación de Conocimientos (QuizActivity): Implementa el banco de preguntas aleatorio y la validación de respuestas.

```

1 package com.example.tesisneumoniaapp
2
3 import android.annotation.SuppressLint
4 import android.content.Intent
5 import android.graphics.Color
6 import android.os.Bundle
7 import android.os.Handler
8 import android.os.Looper
9 import android.view.View
10 import android.widget.Button
11 import android.widget.ImageView
12 import android.widget.TextView
13 import android.widget.Toast
14 import androidx.appcompat.app.AppCompatActivity
15 import androidx.cardview.widget.CardView
16 import java.util.ArrayList
17
18 data class Pregunta(
19     val enunciado: String,
20     val opciones: List<String>,
21     val retroalimentacion: String,
22     val imagenRes: Int = 0 // 0 = Sin imagen
23 )
24
25 class QuizActivity : AppCompatActivity() {
26
27     private lateinit var txtPregunta: TextView
28     private lateinit var btnOp1: Button
29     private lateinit var btnOp2: Button
30     private lateinit var btnOp3: Button
31     private lateinit var txtPuntaje: TextView
32     private lateinit var imgPregunta: ImageView
33     private lateinit var cardImagen: CardView
34
35     private lateinit var listaPreguntas: ArrayList<Pregunta>
36     private var preguntaActual: Pregunta? = null
37     private var indicePregunta = 0
38     private var puntaje = 0
39     private val TOTAL_PREGUNTAS_POR_RONDA = 10
40

```

```

41 override fun onCreate(savedInstanceState: Bundle?) {
42     super.onCreate(savedInstanceState)
43     setContentView(R.layout.activity_quiz)
44
45     txtPregunta = findViewById(R.id.txtPregunta)
46     btnOp1 = findViewById(R.id.btnOp1)
47     btnOp2 = findViewById(R.id.btnOp2)
48     btnOp3 = findViewById(R.id.btnOp3)
49     txtPuntaje = findViewById(R.id.txtPuntaje)
50     imgPregunta = findViewById(R.id.imgPregunta)
51     cardImagen = findViewById(R.id.cardImagen)
52
53     cargarBancoDePreguntas()
54     mostrarPregunta()
55
56     btnOp1.setOnClickListener { validarRespuesta(btnOp1) }
57     btnOp2.setOnClickListener { validarRespuesta(btnOp2) }
58     btnOp3.setOnClickListener { validarRespuesta(btnOp3) }
59 }
60
61 private fun cargarBancoDePreguntas() {
62     val banco = ArrayList<Pregunta>()
63
64     // --- LAS 4 PREGUNTAS CON IMAGEN ---
65     banco.add(Pregunta("¿Qué patrón identifica en esta placa despejada?",
66         listOf("Pulmones Sanos", "Lobar", "Bronconeumonía"), "Correcto.", R.drawable.
67             pulmon_sano))
68
69     banco.add(Pregunta("Observe la mancha blanca sólida. ¿Qué tipo de neumonía es?",
70         listOf("Neumonía Lobar", "Bronconeumonía", "Sano"), "Correcto.", R.drawable.
71             pulmon_lobar))
72
73     banco.add(Pregunta("Se aprecian parches blancos dispersos en ambos lados. ¿Cuál es el
74         diagnóstico?",
75         listOf("Bronconeumonía", "Lobar", "Intersticial"), "Correcto.", R.drawable.
76             pulmon_bronco))
77
78     banco.add(Pregunta("Se observa una neblina o 'vidrio deslustrado' difuso. ¿Qué es?",
79         listOf("Neumonía Intersticial", "Sano", "Lobar"), "Correcto.", R.drawable.
80             pulmon_intersticial))
81
82     // --- TUS 10 PREGUNTAS ORIGINALES DE TEXTO ---
83     banco.add(Pregunta("En una radiografía Normal, ¿cómo deben verse los ángulos costofré
84         nicos?",
85         listOf("Agudos, limpios y libres", "Borrados o con líquido", "Redondeados"), "
86             Correcto."))
87
88     banco.add(Pregunta("¿Qué patrón radiológico se describe como 'Vidrio Deslustrado'?",
89         listOf("Neumonía Intersticial / Viral", "Neumonía Lobar", "Bronconeumonía"), "
90             Correcto."))
91
92     banco.add(Pregunta("Si observas una 'mancha blanca sólida' que ocupa un lóbulo entero:"

```

```

85     listOf("Neumonía Lobar", "Bronconeumonía", "Neumonía Viral"), "Correcto.))
86
87     banco.add(Pregunta("Visualmente, ¿cómo se diferencia la Bronconeumonía de la Lobar?",
88         listOf("Son múltiples parches dispersos", "Es una sola masa sólida", "Son líneas
89             finas"), "Correcto.))
90
91     banco.add(Pregunta("¿Qué significa que los campos pulmonares se vean 'Negros'?",
92         listOf("Están llenos de aire (Normal)", "Tienen líquido", "Infección grave"), "
93             Correcto.))
94
95     banco.add(Pregunta("A diferencia de la Lobar, la Neumonía Intersticial permite ver:",
96         listOf("Los vasos sanguíneos a través de la mancha", "Nada, es totalmente blanca",
97             "Solo costillas"), "Correcto.))
98
99     banco.add(Pregunta("¿Qué signo visual consiste en ver tubos negros dentro de una zona
100         blanca?",
101         listOf("Broncograma Aéreo", "Signo de la Silueta", "Vidrio Esmerilado"), "Correcto.
102             "))
103
104     banco.add(Pregunta("Las infecciones virales (como Influenza o COVID) suelen presentarse
105         :",
106         listOf("Bilateral y difuso (ambos lados)", "Unilateral (un solo lado)", "Focalizado
107             "), "Correcto.))
108
109     banco.add(Pregunta("Si la IA clasifica una imagen como 'Ruido', significa que:",
110         listOf("La imagen no es una radiografía válida", "Hay mucho aire", "Está al revés")
111         , "Correcto.))
112
113     banco.add(Pregunta("¿Qué bacteria es la causa más común de la Neumonía Lobar?",
114         listOf("Streptococcus pneumoniae", "Influenza", "Mycoplasma"), "Correcto.))
115
116     // MEZCLAR TODO EL BANCO (14 preguntas)
117     banco.shuffle()
118
119     // TOMAR SOLO 10 AL AZAR
120     listaPreguntas = ArrayList(banco.take(TOTAL_PREGUNTAS_POR_RONDA))
121 }
122
123 @SuppressWarnings("SetTextI18n")
124 private fun mostrarPregunta() {
125     val colorBoton = Color.parseColor("#1E1E1E")
126     listOf(btnOp1, btnOp2, btnOp3).forEach {
127         it.setBackgroundColor(colorBoton)
128         it.isEnabled = true
129     }
130
131     if (indicePregunta < listaPreguntas.size) {
132         preguntaActual = listaPreguntas[indicePregunta]
133
134         // Gestión de visibilidad de imagen
135         if (preguntaActual!!.imagenRes != 0) {

```

```

128         cardImagen.visibility = View.VISIBLE
129         imgPregunta.setImageResource(preguntaActual!!.imagenRes)
130     } else {
131         cardImagen.visibility = View.GONE
132     }
133
134     val opcionesBarajadas = ArrayList(preguntaActual!!.opciones)
135     opcionesBarajadas.shuffle()
136
137     txtPregunta.text = preguntaActual!!.enunciado
138     btnOp1.text = opcionesBarajadas[0]
139     btnOp2.text = opcionesBarajadas[1]
140     btnOp3.text = opcionesBarajadas[2]
141     txtPuntaje.text = "Puntaje: $puntaje"
142 } else {
143     finalizarJuego()
144 }
145 }
146
147 private fun validarRespuesta(botonPresionado: Button) {
148     listOf(btnOp1, btnOp2, btnOp3).forEach { it.isEnabled = false }
149     val respuestaUsuario = botonPresionado.text.toString()
150     val respuestaCorrecta = preguntaActual!!.opciones[0]
151
152     if (respuestaUsuario == respuestaCorrecta) {
153         puntaje++
154         botonPresionado.setBackgroundColor(Color.parseColor("#4CAF50"))
155         Toast.makeText(this, ";Correcto!", Toast.LENGTH_SHORT).show()
156     } else {
157         botonPresionado.setBackgroundColor(Color.parseColor("#F44336"))
158         if (btnOp1.text == respuestaCorrecta) btnOp1.setBackgroundColor(Color.parseColor("#4CAF50"))
159         if (btnOp2.text == respuestaCorrecta) btnOp2.setBackgroundColor(Color.parseColor("#4CAF50"))
160         if (btnOp3.text == respuestaCorrecta) btnOp3.setBackgroundColor(Color.parseColor("#4CAF50"))
161         Toast.makeText(this, "La correcta era: $respuestaCorrecta", Toast.LENGTH_SHORT).show()
162     }
163
164     Handler(Looper.getMainLooper()).postDelayed({
165         indicePregunta++
166         mostrarPregunta()
167     }, 1500)
168 }
169
170 private fun finalizarJuego() {
171     val intent = Intent(this, QuizResultActivity::class.java)
172     intent.putExtra("PUNTAJE_OBTENIDO", puntaje)
173     intent.putExtra("TOTAL_PREGUNTAS", TOTAL_PREGUNTAS_POR_RONDA)
174     startActivity(intent)
175     finish()

```

```
176     }
177 }
```

2.5. Pantalla de Bienvenida (SplashActivity): Gestiona la carga inicial de los recursos de IA.

```
1 package com.example.tesisneumoniaapp
2
3 import android.content.Intent
4 import android.os.Bundle
5 import android.os.Handler
6 import android.os.Looper
7 import androidx.appcompat.app.AppCompatActivity
8
9 class SplashActivity : AppCompatActivity() {
10     override fun onCreate(savedInstanceState: Bundle?) {
11         super.onCreate(savedInstanceState)
12         setContentView(R.layout.activity_splash)
13
14         // Espera 3 segundos y salta al Menú
15         Handler(Looper.getMainLooper()).postDelayed({
16             startActivity(Intent(this, MenuActivity::class.java))
17             finish() // Cierra el Splash para no poder volver atrás
18         }, 3000)
19     }
20 }
```

3. Diseño de Interfaces de Usuario (Layouts XML)

3.1. Interfaz de Diagnóstico (activity_main.xml):

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:background="#121212"
8     tools:context=".MainActivity">
9
10     <LinearLayout
11         android:layout_width="match_parent"
12         android:layout_height="wrap_content"
13         android:orientation="vertical"
14         android:padding="20dp"
15         android:gravity="center_horizontal">
16
17         <TextView
18             android:layout_width="wrap_content"
19             android:layout_height="wrap_content"
20             android:text="NeumoLens"
21             android:textColor="#90CAF9"
22             android:textSize="28sp"
23             android:textStyle="bold"
24             android:layout_marginBottom="20dp"/>
```

```

25
26 <FrameLayout
27     android:layout_width="match_parent"
28     android:layout_height="350dp">
29
30     <androidx.cardview.widget.CardView
31         android:layout_width="match_parent"
32         android:layout_height="match_parent"
33         app:cardCornerRadius="16dp"
34         app:cardElevation="8dp"
35         app:cardBackgroundColor="#1E1E1E">
36
37         <ImageView
38             android:id="@+id/imgRadiografia"
39             android:layout_width="match_parent"
40             android:layout_height="match_parent"
41             android:scaleType="fitCenter"
42             android:padding="2dp"
43             android:src="@android:drawable/ic_menu_gallery"
44             android:background="#000000"/>
45     </androidx.cardview.widget.CardView>
46
47     <com.google.android.material.floatingactionbutton.FloatingActionButton
48         android:id="@+id/btnVisibility"
49         android:layout_width="wrap_content"
50         android:layout_height="wrap_content"
51         android:layout_gravity="top|end"
52         android:layout_margin="12dp"
53         android:src="@android:drawable/ic_menu_view"
54         app:backgroundTint="#2196F3"
55         app:fabSize="mini"
56         app:tint="#FFFFFF"
57         android:visibility="gone" />
58 </FrameLayout>
59
60 <LinearLayout
61     android:layout_width="match_parent"
62     android:layout_height="wrap_content"
63     android:orientation="horizontal"
64     android:layout_marginTop="25dp"
65     android:weightSum="2">
66
67     <Button
68         android:id="@+id/btnCARGAR"
69         android:layout_width="34dp"
70         android:layout_height="60dp"
71         android:layout_marginEnd="5dp"
72         android:layout_weight="1.3"
73         android:backgroundTint="#2196F3"
74         android:text="    CARGAR"
75         android:textColor="#FFFFFF"
76         android:textSize="16sp"

```

```

77         android:textStyle="bold" />
78
79     <Button
80         android:id="@+id/btnLIMPIAR"
81         android:layout_width="99dp"
82         android:layout_height="60dp"
83         android:layout_marginStart="5dp"
84         android:layout_weight="0.7"
85         android:backgroundTint="#424242"
86         android:text="LIMPIAR"
87         android:textColor="#FFFFFF"
88         android:textSize="14sp" />
89 </LinearLayout>
90
91 <androidx.cardview.widget.CardView
92     android:id="@+id/cardResultado"
93     android:layout_width="match_parent"
94     android:layout_height="wrap_content"
95     android:layout_marginTop="25dp"
96     app:cardCornerRadius="16dp"
97     app:cardElevation="6dp"
98     app:cardBackgroundColor="#1E1E1E"
99     app:contentPadding="20dp"
100     android:visibility="gone">
101
102     <LinearLayout
103         android:layout_width="match_parent"
104         android:layout_height="wrap_content"
105         android:orientation="vertical">
106
107         <TextView
108             android:id="@+id/txtTituloResultado"
109             android:layout_width="wrap_content"
110             android:layout_height="wrap_content"
111             android:text="DIAGNÓSTICO PRELIMINAR"
112             android:textSize="12sp"
113             android:letterSpacing="0.1"
114             android:textColor="#B0BEC5"/>
115
116         <TextView
117             android:id="@+id/txtResultado"
118             android:layout_width="wrap_content"
119             android:layout_height="wrap_content"
120             android:text="NEUMONÍA LOBAR"
121             android:textSize="24sp"
122             android:textStyle="bold"
123             android:layout_marginTop="8dp"/>
124
125         <TextView
126             android:id="@+id/txtPorcentaje"
127             android:layout_width="wrap_content"
128             android:layout_height="wrap_content"

```

```

129         android:text="Probabilidad: 0%"
130         android:textSize="16sp"
131         android:layout_marginBottom="10dp"/>
132
133     <View
134         android:layout_width="match_parent"
135         android:layout_height="1dp"
136         android:background="#333333"
137         android:layout_marginTop="10dp"
138         android:layout_marginBottom="15dp"/>
139
140     <TextView
141         android:layout_width="wrap_content"
142         android:layout_height="wrap_content"
143         android:text="        ANÁLISIS CLÍNICO (RED NEURONAL) :"
144         android:textStyle="bold"
145         android:textColor="#64B5F6"/>
146
147     <TextView
148         android:id="@+id/txtExplicacion"
149         android:layout_width="wrap_content"
150         android:layout_height="wrap_content"
151         android:layout_marginTop="8dp"
152         android:text="Esperando análisis..."
153         android:lineSpacingExtra="5dp"
154         android:textSize="15sp"
155         android:textColor="#E0E0E0"/>
156 </LinearLayout>
157 </androidx.cardview.widget.CardView>
158
159 <Space
160     android:layout_width="match_parent"
161     android:layout_height="50dp" />
162
163 </LinearLayout>
164 </ScrollView>

```

3.2. Interfaz del Quiz (activity_quiz.xml):

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/
  apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:background="#121212"
7     android:padding="20dp">
8
9     <TextView
10         android:id="@+id/txtPuntaje"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:text="Puntaje: 0"
14         android:textColor="#90CAF9"

```

```

15     android:textSize="16sp"
16     app:layout_constraintEnd_toEndOf="parent"
17     app:layout_constraintTop_toTopOf="parent" />
18
19 <androidx.cardview.widget.CardView
20     android:id="@+id/cardImagen"
21     android:layout_width="match_parent"
22     android:layout_height="250dp"
23     android:layout_marginTop="15dp"
24     app:cardBackgroundColor="#000000"
25     app:cardCornerRadius="16dp"
26     app:layout_constraintTop_toBottomOf="@id/txtPuntaje"
27     android:visibility="gone">
28
29     <ImageView
30         android:id="@+id/imgPregunta"
31         android:layout_width="match_parent"
32         android:layout_height="match_parent"
33         android:scaleType="fitCenter" />
34 </androidx.cardview.widget.CardView>
35
36 <TextView
37     android:id="@+id/txtPregunta"
38     android:layout_width="0dp"
39     android:layout_height="wrap_content"
40     android:layout_marginTop="20dp"
41     android:gravity="center"
42     android:text="¿Pregunta?"
43     android:textColor="#FFFFFF"
44     android:textSize="19sp"
45     android:textStyle="bold"
46     app:layout_constraintEnd_toEndOf="parent"
47     app:layout_constraintStart_toStartOf="parent"
48     app:layout_constraintTop_toBottomOf="@+id/cardImagen" />
49
50 <LinearLayout
51     android:layout_width="match_parent"
52     android:layout_height="wrap_content"
53     android:layout_marginTop="25dp"
54     android:orientation="vertical"
55     app:layout_constraintTop_toBottomOf="@+id/txtPregunta">
56
57     <Button
58         android:id="@+id/btnOp1"
59         android:layout_width="match_parent"
60         android:layout_height="60dp"
61         android:layout_marginBottom="12dp"
62         android:backgroundTint="#1E1E1E"
63         android:textAllCaps="false"
64         android:textColor="#FFFFFF" />
65
66     <Button

```

```

67         android:id="@+id/btnOp2"
68         android:layout_width="match_parent"
69         android:layout_height="60dp"
70         android:layout_marginBottom="12dp"
71         android:backgroundTint="#1E1E1E"
72         android:textAllCaps="false"
73         android:textColor="#FFFFFF" />
74
75     <Button
76         android:id="@+id/btnOp3"
77         android:layout_width="match_parent"
78         android:layout_height="60dp"
79         android:backgroundTint="#1E1E1E"
80         android:textAllCaps="false"
81         android:textColor="#FFFFFF" />
82
83 </LinearLayout>
84
85 </androidx.constraintlayout.widget.ConstraintLayout>

```

3.3. Interfaz de Detalle Educativo (activity_detail.xml):

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.coordinatorlayout.widget.CoordinatorLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:background="#121212">
8
9     <com.google.android.material.appbar.AppBarLayout
10         android:id="@+id/appBar"
11         android:layout_width="match_parent"
12         android:layout_height="300dp">
13
14         <com.google.android.material.appbar.CollapsingToolbarLayout
15             android:layout_width="match_parent"
16             android:layout_height="match_parent"
17             app:contentScrim="#1E1E1E"
18             app:layout_scrollFlags="scroll|exitUntilCollapsed">
19
20             <FrameLayout
21                 android:layout_width="match_parent"
22                 android:layout_height="match_parent"
23                 app:layout_collapseMode="parallax">
24
25                 <ImageView
26                     android:id="@+id/imgDetalleBase"
27                     android:layout_width="match_parent"
28                     android:layout_height="match_parent"
29                     android:scaleType="fitCenter" />
30
31                 <ImageView
32                     android:id="@+id/imgDetalleOverlay"

```

```

33         android:layout_width="match_parent"
34         android:layout_height="match_parent"
35         android:scaleType="fitCenter"
36         android:visibility="invisible" />
37
38     <View
39         android:layout_width="match_parent"
40         android:layout_height="match_parent"
41         android:background="#4000000" />
42 </FrameLayout>
43
44     <androidx.appcompat.widget.Toolbar
45         android:id="@+id/toolbar"
46         android:layout_width="match_parent"
47         android:layout_height="?attr/actionBarSize"
48         app:layout_collapseMode="pin"
49         app:titleTextColor="#FFFFFF" />
50
51     </com.google.android.material.appbar.CollapsingToolbarLayout>
52 </com.google.android.material.appbar.AppBarLayout>
53
54 <com.google.android.material.floatingactionbutton.FloatingActionButton
55     android:id="@+id/btnVerPatologia"
56     android:layout_width="wrap_content"
57     android:layout_height="wrap_content"
58     android:src="@android:drawable/ic_menu_view"
59     app:backgroundTint="#2196F3"
60     app:tint="#FFFFFF"
61     app:layout_anchor="@id/appBar"
62     app:layout_anchorGravity="bottom|end"
63     android:layout_marginEnd="20dp"
64     android:contentDescription="Ver Patología" />
65
66 <androidx.core.widget.NestedScrollView
67     android:layout_width="match_parent"
68     android:layout_height="match_parent"
69     app:layout_behavior="@string/appbar_scrolling_view_behavior">
70
71     <LinearLayout
72         android:layout_width="match_parent"
73         android:layout_height="wrap_content"
74         android:orientation="vertical"
75         android:padding="20dp">
76
77         <TextView
78             android:id="@+id/txtTituloDetalle"
79             android:layout_width="wrap_content"
80             android:layout_height="wrap_content"
81             android:text="Título del Patrón"
82             android:textColor="#69F0AE"
83             android:textSize="26sp"
84             android:textStyle="bold"

```

```
85         android:layout_marginBottom="16dp"/>
86
87     <TextView
88         android:id="@+id/txtContenidoDetalle"
89         android:layout_width="match_parent"
90         android:layout_height="wrap_content"
91         android:text="Explicación..."
92         android:textColor="#E0E0E0"
93         android:textSize="16sp"
94         android:lineSpacingExtra="6dp"/>
95
96     </LinearLayout>
97 </androidx.core.widget.NestedScrollView>
98 </androidx.coordinatorlayout.widget.CoordinatorLayout>
```

ANEXO C
IMAGENES DE LA APP

Imágenes de la app en funcionamiento

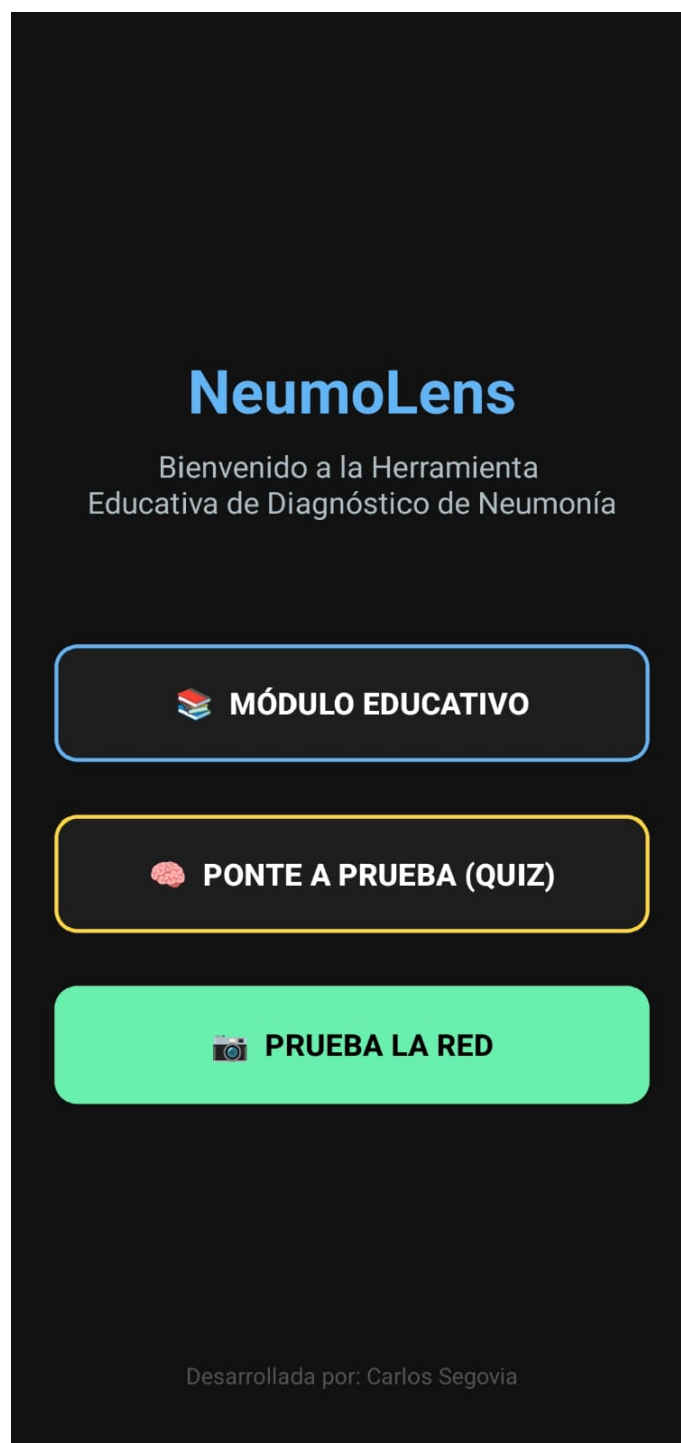


Figura 31. Menú Principal

Biblioteca de Patrones

Selecciona un patrón para estudiar sus características detalladas antes de realizar el Quiz.

 PULMONES SANOS

 NEUMONÍA LOBAR

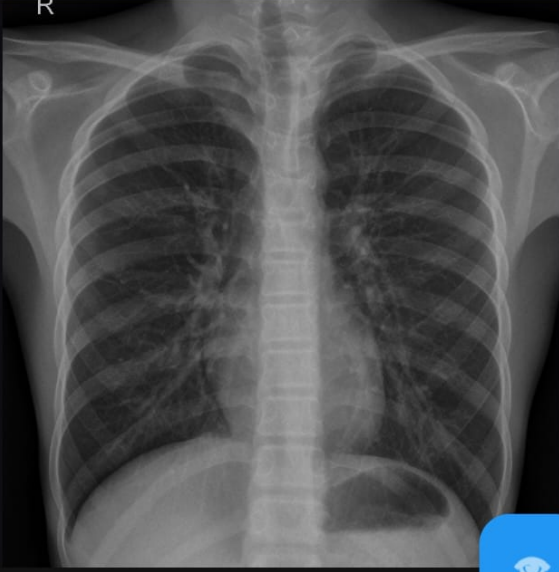
 BRONCONEUMONÍA

 NEUMONÍA INTERSTICIAL


¿Ya estudiaste todo?

 IR AL QUIZ DE CONOCIMIENTOS

Figura 32. Biblioteca de Patrones



R



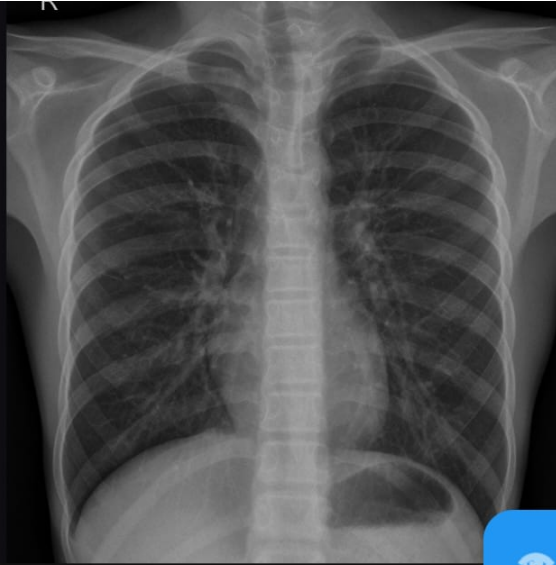
Neumonía Lobar: Consolidación Alveolar

INTRODUCCIÓN:
Es una infección bacteriana (comúnmente por Neumococo) que se localiza en una sección específica llamada lóbulo. El aire del alvéolo es reemplazado por exudado inflamatorio, convirtiendo el tejido esponjoso en uno sólido.

ANÁLISIS RADIOLÓGICO:

- **OPACIDAD HOMOGÉNEA:** Aparece una mancha blanca densa y uniforme en una zona delimitada.
- **BRONCOGRAMA AÉREO:** Dentro de la zona blanca, se pueden apreciar líneas oscuras que corresponden a los bronquios que aún

Figura 33. 1er Patrón Radiológico



Bronconeumonía: Infiltrado Multifocal

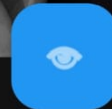
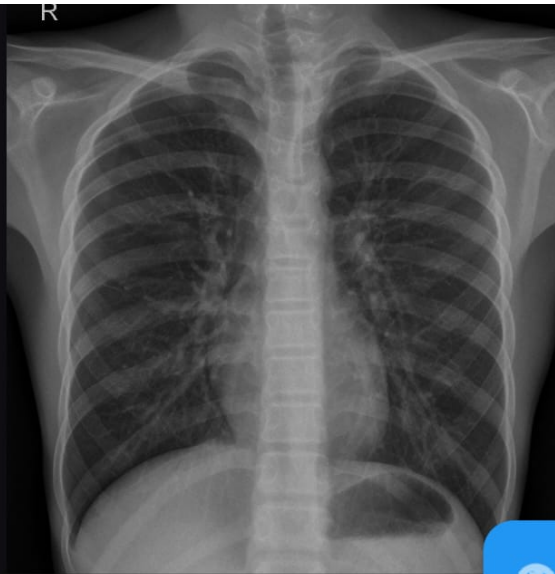
INTRODUCCIÓN:

A diferencia de la lobar, esta infección se origina en los bronquios y se extiende de forma irregular hacia los alvéolos cercanos. Es un proceso inflamatorio disperso y no limitado a una sola sección.

ANÁLISIS RADIOLÓGICO:

- **PATRÓN EN PARCHES:** Se observan múltiples opacidades de pequeño tamaño, con bordes algodonosos y mal definidos.
- **DISTRIBUCIÓN BILATERAL:** Es frecuente encontrar estos focos de infección en ambos pulmones, especialmente en las zonas inferiores (bases).

Figura 34. 2do Patrón Radiológico



Neumonía Intersticial: Patrón Infiltrativo

INTRODUCCIÓN:

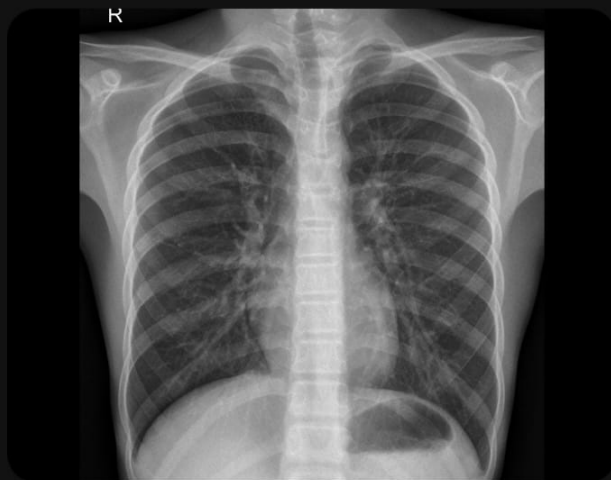
Este patrón es típico de infecciones virales (como Influenza o COVID-19). La inflamación no llena el alvéolo de líquido, sino que engrosa las paredes del tejido pulmonar (intersticio).

ANÁLISIS RADIOLÓGICO:

- VIDRIO DESLUSTRADO: Se observa un aumento de la densidad pulmonar (grisáceo) que no llega a ser blanco total, permitiendo ver los vasos sanguíneos debajo.
- RED RETICULAR: Aparece una malla o red de líneas blancas muy finas que cubren gran parte del pulmón.

Figura 35. 3er Patrón Radiológico

Puntaje: 0



¿Qué patrón identifica en esta placa despejada?

Pulmones Sanos

Lobar

Bronconeumonía

Figura 36. Preguntas al Usuario (Quiz)



Figura 37. Mal Resultado



Figura 38. Buen Resultado



Figura 39. Apartado para Probar la Red Neuronal

NeumoLens



 CARGAR

 LIMPIAR

DIAGNÓSTICO PRELIMINAR

 **IMAGEN INVÁLIDA**

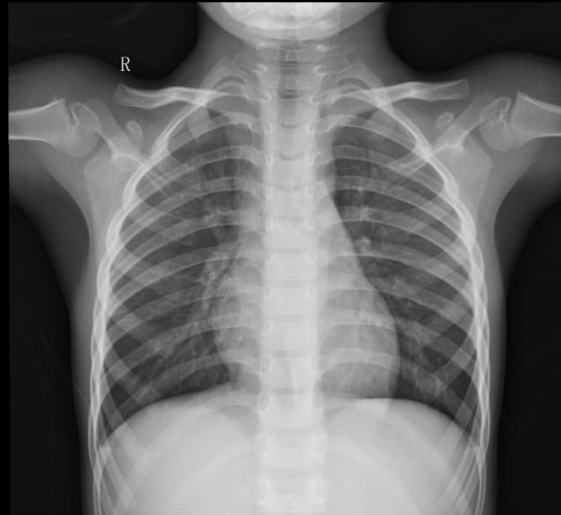
Certeza: 99.3%

 **ANÁLISIS CLÍNICO (RED NEURONAL):**

No es una radiografía válida.

Figura 40. Resultado de Ruido/Imagen Invalida

NeumoLens



 CARGAR

 LIMPIAR

DIAGNÓSTICO PRELIMINAR

 **PULMONES SANOS**

Confianza: 99.5%

 **ANÁLISIS CLÍNICO (RED NEURONAL):**

Sin hallazgos patológicos.

Figura 41. Resultado de unos Pulmones Sanos

NeumoLens



 CARGAR

 LIMPIAR

DIAGNÓSTICO PRELIMINAR

 **INTERSTICIAL**

Confianza: 59.9%

 **ANÁLISIS CLÍNICO (RED NEURONAL):**

Patrón vidrio deslustrado.

Figura 42. Resultado y Detección de un caso en una Radiografía de Torax