

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación

“IMPLEMENTACIÓN DE UN SISTEMA DE ANÁLISIS DE
IMÁGENES MÉDICAS EN LA NUBE PARA DIAGNÓSTICOS
REMOTOS”

EXAMEN COMPLEXIVO - COMPONENTE PRÁCTICO
INFORME PROFESIONAL

Previo a la obtención del Título de:

MAGISTER EN INGENIERÍA BIOMÉDICA

KATHERINE ELIZABETH JÁCOME CORO

GUAYAQUIL – ECUADOR

AÑO: 2025

AGRADECIMIENTOS

Mi más profundo agradecimiento a la Escuela Superior Politécnica del Litoral por brindarme la oportunidad de formar parte de la V cohorte de la Maestría en Ingeniería Biomédica, permitiéndome continuar con mi desarrollo académico y profesional

Agradezco sinceramente a los docentes del programa por compartir sus valiosos conocimientos y experiencias a lo largo de este proceso formativo. En especial, extendiendo mi gratitud al PhD. Washington Velásquez, tutor de este trabajo de titulación, por su acompañamiento.

Finalmente, a mi familia por su respaldo incondicional, motivación durante cada etapa de este camino.

DEDICATORIA

El presente proyecto lo dedico a mi familia, pilar fundamental en mi vida. En especial, a mi querida madre, quien desde el cielo me acompaña y ha sido testigo silenciosa de cada uno de mis logros. Su recuerdo, su ejemplo y su amor incondicional han sido mi mayor fuente de fortaleza y motivación para alcanzar esta meta profesional.

COMITÉ DE EVALUACIÓN

Washington Velásquez, PhD.

PROFESOR TUTOR

Francis R. Loayza, PhD.

PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

Yo Katherine Elizabeth Jácome Coro acuerdo y reconozco que: La titularidad de los derechos patrimoniales de autor del proyecto de graduación corresponderá al autor, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor. El estudiante deberá procurar en cualquier caso de cesión de sus derechos patrimoniales incluir una cláusula en la cesión que proteja la vigencia de la licencia aquí concedida a la ESPOL.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, secreto empresarial, derechos patrimoniales de autor sobre software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que me corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique al autor que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o

divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 2 de julio del 2025.

Katherine Jácome Coro

Autora

RESUMEN

Las personas que residen en áreas rurales son desatendidas enfrentando desafíos significativos para acceder a servicios médicos especializados. Esta situación se debe a varios factores como la falta de infraestructura tecnológica médica, escases de personal de la salud en zonas rurales, afectando directamente a la calidad del diagnóstico y tratamiento de los pacientes [1].

Por lo que se propone el desarrollo y la implementación de un sistema para el análisis de imágenes médicas, que permita realizar diagnósticos remotos asistidos por inteligencia artificial. Buscando mejorar la cobertura de salud en áreas rurales o de difícil acceso, para asistir a los médicos generales en la toma de decisiones clínicas.

La arquitectura del sistema es modular, implementada con tecnologías *Open Source* como *Flask* para el backend, *PostgreSQL* para la administración de la base de datos, *Fernet* para la encriptación simétrica utilizada en la protección de datos sensibles del paciente, *Docker* para el almacenamiento de información de los servicios, *CheXNet* es el algoritmo de Inteligencia Artificial capaz de determinar hasta 14 patologías pulmonares a partir de radiografías de tórax y *NGROK* que permite exponer el sistema local a través de una IP pública para simular el acceso remoto. Esta configuración brinda portabilidad al sistema para su posterior migración en la nube, garantizando escalabilidad y eficiencia en despliegues futuros.

Con las pruebas realizadas se pudo evidenciar el correcto funcionamiento del sistema de forma local y de acceso remoto a través de un dispositivo móvil con acceso a datos móviles evidenciando que es una solución efectiva y escalable para apoyar en el diagnóstico médico en áreas desatendidas.

ÍNDICE GENERAL

AGRADECIMIENTOS.....	2
DEDICATORIA.....	3
COMITÉ DE EVALUACIÓN.....	4
DECLARACIÓN EXPRESA.....	5
RESUMEN.....	7
INTRODUCCIÓN.....	10
CAPÍTULO 1.....	12
1. PLANTEAMIENTO DEL PROBLEMA Y MARCO TEORICO.....	12
1.1 Descripción del Problema	12
1.2 Justificación del Problema.....	13
1.3 Solución Propuesta	14
1.4 Objetivos.....	15
1.4.1 Objetivo General	15
1.4.2 Objetivos Específicos	15
1.5 Alcance e Impacto	15
1.5.1 Alcance	15
1.5.2 Impacto	16
1.6 Marco Teórico	16
1.6.1 Telemedicina.....	16
1.6.2 Imágenes médicas	18
1.6.3 Procesamiento y Análisis de Imágenes Medicas	19
1.6.4 Estándares	20
1.6.5 Inteligencia Artificial.....	22
CAPÍTULO 2.....	25
2. METODOLOGÍA.....	25
2.1 Diseño del Sistema de comunicación.....	25
2.1.1 Arquitectura del Sistema de Comunicación	26
2.1.2 Usuarios	26

2.1.3	Políticas de Uso	27
2.2	Infraestructura Tecnológica	28
2.2.1	Arquitectura del Sistema de Comunicación Contenedor Backend (Flask – Python)	29
2.2.2	Contenedor del modelo de inteligencia artificial (CheXNet – TensorFlow Lite).....	30
2.2.3	Contenedor de Base de Datos (PostgreSQL).....	30
2.3	Interfaz web del sistema	31
2.3.1	Integración de la interfaz web con el backend	32
2.3.2	Funcionalidades de la interfaz web	32
2.3.3	Creación de usuario automático - perfil paciente	32
2.4	Implementación del módulo de IA para el análisis de imágenes médicas.33	
2.4.1	Proceso de análisis con IA	34
2.4.2	Integración técnica del modelo CheXNet en el sistema ...	34
2.5	Encriptación de datos sensibles.	35
2.6	Validación remota del sistema.	36
CAPÍTULO 3.....		37
3.	RESULTADOS.....	37
3.1	Funcionamiento del sistema de comunicación	37
3.2	Verificación de conexión entre Backend y Base de Datos.....	37
3.3	Validación de la encriptación y desencriptación de los datos sensibles.	44
3.4	Validación remota del Sistema	46
3.5	Validación Funcional de Sistemas.....	48
3.6	Métricas básicas de desempeño.	48
CONCLUSIONES Y RECOMENDACIONES		50
BIBLIOGRAFÍA.....		52
ANEXOS		55

INTRODUCCIÓN

En la actualidad, el acceso oportuno y preciso a servicios de diagnóstico médico es un pilar fundamental en la atención médica de calidad [2]. Sin embargo, en muchas áreas rurales o desatendidas, hay una escasez de recursos médicos especializados, lo que impide la detección temprana de enfermedades y disminuye la calidad de vida de los pacientes. En este sentido, tecnologías emergentes como la inteligencia artificial, la computación en la nube y la telemedicina ofrecen soluciones innovadoras a las brechas de acceso al diagnóstico clínico especializado [3].

El presente componente práctico tiene como objetivo diseñar e implementar un sistema para el análisis de imágenes de RX de tórax para diagnósticos remotos. Su principal propósito es asistir a los médicos generales en la toma de decisiones clínicas, permite a los especialistas confirmar los resultados desde cualquier lugar a través de una interfaz segura, escalable y fácil de usar.

El sistema desarrollado incorpora tecnologías basadas en inteligencia artificial entrenadas para diagnosticar patologías a partir de imágenes médicas. Cuenta con una base de datos que permite almacenar registros médicos, usuarios y resultados diagnósticos de forma segura, incluye un sistema de autenticación de usuarios con distintos niveles de privilegios según sus roles, y garantiza la seguridad mediante la encriptación de imágenes médicas y datos sensibles de los pacientes. Todos estos componentes fueron implementados en contenedores Docker, lo que asegura portabilidad, modularidad y facilidad de mantenimiento en entornos locales o en la nube.

Este componente práctico proporciona una solución tecnológica innovadora que promueve a la transformación equitativa en el sector de la salud. El sistema desarrollado cumple con las normativas internacionales sobre

Ciberseguridad y Protección de Datos

En los siguientes capítulos se detalla la fundamentación teórica que sustenta el proyecto, su metodología y los resultados obtenidos tras su implementación, seguidos de una evaluación final donde se justifica su relevancia del sistema en escenarios de telemedicina.

CAPÍTULO 1

1. PLANTEAMIENTO DEL PROBLEMA Y MARCO TEORICO

1.1 Descripción del Problema

El análisis de imágenes médicas como radiografías, Tomografías Computarizadas (TC) y Resonancias Magnéticas es fundamental para el diagnóstico y tratamiento de diversas enfermedades. Sus análisis se realizan de manera local o regional en hospitales o clínicas. Esto puede restringir significativamente el acceso a diagnósticos especializados en regiones remotas o desatendidas.

Los sistemas basados en la nube para el análisis de imágenes médicas permiten diagnósticos remotos utilizando recursos de computación en la nube. Sin embargo, este enfoque presenta muchas dificultades:

- **Seguridad y Privacidad de los Datos:** La información sensible, como imágenes contenidas en el archivo médico de un paciente, requiere una protección adecuada. Su transmisión y almacenamiento en sistemas en la nube deben proteger la confidencialidad, integridad y disponibilidad de los datos de acuerdo con las regulaciones de: la Ley de Portabilidad y Responsabilidad de Seguros de Salud (siglas en Ingles HIPAA), y el Reglamento General de Protección de Datos (GDPR) [4], [5].
- **Fiabilidad y Precisión del Diagnóstico:** El análisis automatizado de imágenes médicas debe ser lo más preciso y fiable posible. Los algoritmos de Inteligencia Artificial (IA) y Aprendizaje Automático (siglas en Ingles ML) empleados deben ser entrenados y validados adecuadamente para prevenir diagnósticos erróneos.
- **Latencia y Accesibilidad:** En lugares remotos con mala conectividad, adjuntar grandes conjuntos de datos puede llevar mucho tiempo. Es importante asegurarse de que el sistema sea accesible y funcione de manera óptima en tiempo real.
- **Interoperabilidad:** El sistema debe ser compatible con varios formatos de imágenes médicas y sistemas de información hospitalaria para garantizar la

facilidad integración y el uso de entornos clínicos múltiples

1.2 Justificación del Problema

Las personas que residen en áreas rurales son desatendidas enfrentando desafíos significativos para acceder a servicios médicos especializados. Esta situación se debe a varios factores.

Las instalaciones médicas en áreas rurales son generalmente limitadas debido a que cuentan con menor infraestructura tecnológica médica, el acceso oportuno a atención médica especializada es escasa afectando directamente a la calidad del diagnóstico y tratamiento de los pacientes [1]. Debido al poco número de profesionales de la salud en estas zonas, llevando a tiempos prolongados de espera y atención menos personalizada.

La telemedicina emerge como una solución prometedora para abordar estas limitaciones geográficas, permitiendo a profesionales médicos compartir diagnósticos, opciones de tratamiento y estrategias preventivas a través de tecnologías de la información y comunicación [6]. Dentro de la telemedicina, el análisis de imágenes médicas es crucial, pues facilita la identificación precisa de enfermedades que no siempre son detectables mediante síntomas o exámenes clínicos convencionales.

Sin embargo, el proceso tradicional de análisis de imágenes médicas es complejo y demanda una gran experiencia, resultando en procedimientos lentos que retrasan diagnósticos críticos. La integración de inteligencia artificial (IA) representa una alternativa eficiente para este desafío, permitiendo automatizar la identificación de patrones e irregularidades en las imágenes médicas, mejorando significativamente la precisión y rapidez del diagnóstico [7].

Actualmente existen diversas aplicaciones clínicas de IA en productos aprobados por organismos regulatorios, como la Administración de Alimentos y Medicamentos de los Estados Unidos (FDA), lo que demuestra su confiabilidad en entornos clínicos reales. Ejemplos claros de estas soluciones son IDx-DE, un sistema autónomo aprobado para detectar retinopatía diabética mediante imágenes del fondo ocular, y Aidoc, un software especializado en la detección de patologías críticas como hemorragias

intracraneales y embolias pulmonares mediante tomografías computarizadas.

Por otra parte, la creciente afluencia de pacientes en instituciones médicas genera cuellos de botella que disminuyen la eficiencia del servicio, lo que deriva en retrasos en la detección de enfermedades. Este retardo en el diagnóstico afecta negativamente la salud de los pacientes, al complicarse en las enfermedades que podrían haberse tratado fácilmente en etapas iniciales.

Finalmente, dado que la privacidad y la confidencialidad de los datos de salud es esencial en cualquier sistema de telemedicina, existen regulaciones internacionales estrictas para asegurar la protección adecuada a la información médica sensible. Normativas como el Reglamento General de Protección de Datos (GDPR) en Europa y la Ley de Portabilidad y Responsabilidad del Seguro Médico (HIPAA) en Estados Unidos establecen estándares claros y obligatorias para proteger la privacidad y confidencialidad de los datos de los pacientes.

1.3 Solución Propuesta

Se propone la implementación de un sistema de análisis de imágenes de RX de tórax, orientado a facilitar diagnósticos remotos de manera segura, eficiente y precisa. Este sistema permitirá procesar radiografías desde cualquier ubicación geográfica mejorando el acceso al diagnóstico especializado en zonas rurales o desatendidas.

La seguridad de la información constituye un eje fundamental en el diseño del sistema se integrará protocolos de cifrado para garantizar la confidencialidad, integridad y disponibilidad de los datos.

El sistema integrará un algoritmo de inteligencia artificial entrenado para el análisis de imágenes de Rx de tórax. Este algoritmo permitirá detectar patologías como: neumonía, neumotórax, fibrosis facilitando un diagnóstico temprano y confiable incluso con ausencia de especialistas médicos en el punto de atención. De esta manera se busca reducir los tiempos de atención para así mejorar la toma de decisiones clínicas y contribuir a la equidad en el acceso de servicios de salud.

1.4 Objetivos

1.4.1 Objetivo General

Implementar un sistema de análisis de imágenes médicas en la nube utilizando software libre que permita realizar diagnósticos remotos de manera segura, precisa y eficiente, mejorando el acceso a servicios de diagnóstico especializado en áreas remotas o desatendidas.

1.4.2 Objetivos Específicos

- Aplicar un protocolo de seguridad que asegure la transferencia y almacenamiento de imágenes médicas en la nube, garantizando la confidencialidad, integridad y cumplimiento con normativas internacionales de protección de datos.
- Ejecutar algoritmos de inteligencia artificial y aprendizaje automático para el análisis automatizado de imágenes médicas, asegurando alta precisión y fiabilidad en los diagnósticos remotos.
- Optimizar la transferencia de datos y la accesibilidad del sistema asegurando que funcione de manera eficiente en tiempo real, incluso en entornos con baja calidad de conectividad, y que sea compatible con diversos formatos y sistemas de información hospitalaria.

1.5 Alcance e Impacto

1.5.1 Alcance

- Implementación de un sistema de análisis de imágenes médicas en la nube, empleando software libre, contenedores Docker y tecnologías de inteligencia artificial para facilitar diagnósticos remotos.
- Desarrollo de una interfaz web intuitiva que permita la interacción de distintos tipos de usuarios (administrador, médico general, médico especialista y paciente), con acceso personalizado según el asignado.
- Aplicación de Mecanismos de seguridad que garanticen la confidencialidad, integridad y trazabilidad de los datos médicos mediante autenticación por roles, control de acceso y gestión segura de la información en entornos aislados.
- Integración de algoritmos de inteligencia artificial especializados en el análisis

automatizado de imágenes médicas, como radiografías de tórax, permitiendo generar diagnósticos preliminares con alta precisión que puedan ser validadas por profesionales de salud.

1.5.2 Impacto

- El sistema propuesto tiene un impacto significativo en el ámbito social al contribuir directamente a la mejora de la equidad en el acceso a servicios de salud de calidad. Permitiendo reducir la brecha existente entre áreas urbanas y rurales en cuanto a diagnósticos especializados.
- Identificación temprana de enfermedades, mejorando las tasas de tratamiento efectivo y disminuyendo la progresión de patologías.

1.6 Marco Teórico

1.6.1 Telemedicina

La Organización Mundial de la Salud (OMS), define la telemedicina como el conjunto de servicios de atención sanitaria que se brinda cuando la distancia entre el paciente y el proveedor de salud es un factor determinante. Esta modalidad usa tecnologías de la información y comunicación (TIC) para facilitar el intercambio de datos clínicos con el objetivo de apoyar el diagnóstico y la prevención de enfermedades [8].

En el transcurso de la evolución tanto de la medicina como de las tecnologías de la información y comunicación, se han desarrollado diversos tipos de telemedicina los cuales definen su alcance actual. A continuación, se describirán los 4 tipos más importantes.

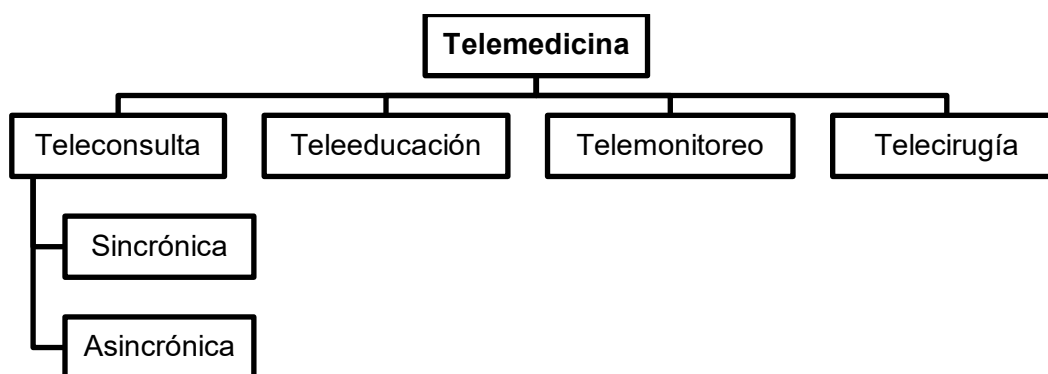


Figura 1.1. Tipos de Telemedicina

- Teleconsulta.** - es la indagación de información médica o sugerencia por parte del personal médico, mediante tecnologías de la información y comunicación. Se desarrolla entre el paciente y el profesional de la salud. La comunicación se establece mediante dos formas como son de forma directa o por intermediarios siendo su interacción sincrónica o asincrónica.
 Las teleconsultas asíncronas se caracterizan por el envío de información clínica por parte del paciente o del personal de salud permitiendo que el profesional médico revise y responda en un momento posterior. En cambio, las teleconsultas sincrónicas se llevan a cabo en tiempo real entre el paciente y el profesional de salud facilitando una comunicación directa [9].
- Teleeducación.** – Es la utilización de las tecnologías de la información y comunicación para la práctica educativa médica a distancia. Existen distintos, estudios que indican la utilización de tecnologías de la información una de ellas la videoconferencia, son los más utilizados por los profesionales en salud mejorando sus conocimientos [9].
- Telemonitoreo.** - Se manipula tecnologías de información y comunicación para obtener datos de rutina con respecto a la condición de los pacientes. Este tipo permite a los profesionales en salud obtener y monitorear las variables fisiológicas, resultados de exámenes e imágenes [8][9].
- Telecirugía.** Es la aplicación de cirugías en donde el cirujano no participa con el paciente, de tal manera, la visualización y manipulación es ejecutada a distancia mediante dispositivos tele-electrónicos y alta tecnología en telecomunicaciones.

1.6.2 Imágenes médicas

Es el conjunto de técnicas y procesos usados para crear imágenes del cuerpo humano, o una fracción de él, con propósitos clínicos, procedimientos médicos que buscan diagnosticar enfermedades, de igual manera para propósitos científicos como el estudio de la anatomía física y metabólica [10].

Tipo de Imágenes Médicas

- **Radiografía**

Es una técnica de imagenología que se maneja radiaciones ionizantes para obtener imágenes del interior del cuerpo humano, ayudando a los profesionales de la salud a examinar y diagnosticar diversas afecciones, desde fracturas óseas hasta enfermedades pulmonares y tumores.

El principio esencial de la radiografía es la capacidad de ciertos tipos de energía para traspasar el cuerpo humano y crear imágenes en una placa o detector [11].

- **Tomografía**

Es la obtención de imágenes por secciones o en cortes de distintas partes internas del cuerpo humano. Mediante el uso de distintas tecnologías la tomografía permite visualizar a detalle la anatomía y la composición de los órganos tejidos y huesos, facilitando el diagnóstico y tratamiento de una extensa lista de enfermedades y trastornos [12].

Las tomografías tienen distintas tecnologías y aplicaciones clínicas las más utilizadas son la computarizada y por emisión de positrones [12].

La tomografía por emisión de positrones conocida como (PET) es un método de diagnóstico en el campo de la medicina nuclear que emplea radiotrazadores con isótopos que emiten positrones, como el flúor-18, para adquirir imágenes del metabolismo y la función del organismo. El paciente recibe el radiotrazador, y las emisiones de positrones son detectadas por un equipo conocido como tomocámara o cámara gamma. Esta técnica se utiliza principalmente para diagnosticar enfermedades oncológicas, neurológicas y cardíacas, al brindar información sobre el

metabolismo celular y la circulación en los tejidos.

- **Resonancia Magnética**

Es una técnica de imagen no invasiva que se manipula un campo magnético y ondas de radiofrecuencia para conseguir imágenes de distintas superficies del cuerpo humano.

Se obtiene imágenes de alta resolución para observar las alteraciones del cuerpo sin utilizar radiaciones ionizantes (rayos X). No poseen efectos perjudiciales para su organismo [13].

- **Ecografía**

Es un examen de diagnóstico rápido permitiéndonos obtener imágenes de nuestros órganos y estructuras internas del cuerpo a través de ondas de ultra frecuencia [14].

1.6.3 Procesamiento y Análisis de Imágenes Médicas

El procesamiento de imágenes médicas hace referencia a técnicas computacionales para mejorar, analizar y visualizar imágenes mediante diferentes modalidades, como resonancia magnética (RM), tomografía computarizada (TC), ultrasonido, y radiografía [15].

Las técnicas más conocidas de procesamiento de imágenes se encuentra el filtrado espacial, la segmentación, la reconstrucción tridimensional y registro de imágenes estas útiles herramientas permiten a los profesionales de la salud a descifrar imágenes médicas con mayor precisión, facilitando la toma de decisiones clínicas.

- **Filtrado Espacial:** Es utilizado para mejorar la calidad de la imagen eliminando ruido y resaltando características relevantes.
- **Segmentación de Imágenes:** Se utiliza para dividir una imagen en regiones off o segmentos que representan diferentes estructuras anatómicas
- **El registro de imágenes:** es una técnica que alinea imágenes de diferentes modalidades o tiempos así permitiendo una comparación y un análisis más efectivo

- **Reconstrucción 3D:** es una conversión de imágenes bidimensionales en modelos tridimensionales así facilitando la visualización de estructuras anatómicas complejas

La inteligencia artificial (IA) ha mejorado el procesamiento de imágenes médicas, ayudando a desarrollar algoritmos que permitan procesar grandes volúmenes de datos con rapidez y precisión. La enseñanza ha demostrado ser eficaz en la codificación de imágenes y detección de patrones médicos que son desapercibidos por los humanos entre los algoritmos más comunes de IA son:

- **Redes Neuronales Convolucionales (CNN):** Utilizadas para la clasificación y segmentación de imágenes médicas, siendo especialmente útiles en el diagnóstico de cáncer y enfermedades oculares [16].
- **Máquinas de Soporte Vectorial (SVM):** Aplicadas en el análisis de señales ECG para la detección de arritmias y otras anomalías cardíacas.
- **Redes Recurrentes (RNN):** Empleadas en el análisis de series temporales de señales médicas, como el EEG, para la predicción de convulsiones.

1.6.4 Estándares

Los estándares para manejar y compartir información de datos médicos garantizan la coherencia y la eficiencia en todos los sistemas de salud. HL7 y DICOM son claves para que gestionen estos datos complejos [17].

Imágenes Digitales y Comunicaciones en Medicina (DICOM)

DICOM ha sido creado específicamente para gestionar, almacenar, imprimir y enviar datos en imágenes médicas. Incorpora un conjunto de protocolos para manejar los datos de imágenes y los metadatos asociados a los dispositivos de diagnóstico por imágenes médicas, tales como radiografías, resonancias magnéticas y tomografías computarizadas.

- **Estructura de imagen y datos:** DICOM estandariza el tipo de archivo y el protocolo de comunicaciones de red que se emplea en las imágenes. Esto asegura el acceso a las imágenes y a la información relacionada utilizando en

distintos sistemas y dispositivos, sin importar el fabricante.

- **Metadatos:** además de los datos de imagen, los archivos DICOM incluyen metadatos detallados, como la identificación del paciente, el tipo de escaneo, las dimensiones de la imagen y la información específica del dispositivo, que son esenciales para el diagnóstico.

Nivel de Salud Siete Internacional (HL7)

Health Level Seven (HL7) es un estándar diseñado para el intercambio de información médica entre los proveedores de atención médica. Su ámbito de aplicación engloba la entrada de diversos tipos de órdenes, la emisión de informes de resultados de pruebas, recetas, así como los procesos de admisión, alta y transferencia de pacientes, etc. Es uno de los estándares médicos más destacados.

HL7 se ocupa del intercambio, la integración y la obtención de información médica en formato electrónico. Los estándares definen y ofrecen una estructura para el intercambio de información relacionado con los pacientes entre distintos sistemas de salud [17].

- **Protocolos de intercambio de datos:** HL7 establece los estándares de comunicación para el intercambio de información entre los sistemas de atención médica. Estos mensajes pueden incluir los registros de los pacientes, los informes de laboratorio y los datos de facturación.
- **Flexibilidad y extensibilidad:** los mensajes HL7 son personalizables, lo que permite el envío de distintos datos médicos y administrativos. Es compatible con diversas necesidades y flujos de trabajo en un contexto de atención médica.

Diferencias: DICOM vs HL7

Tabla 1.1. Diferencias DICOM vs HL7 [17].

Aspecto	DICOM	HL7
Enfoque principal	Estandarizar el manejo, el almacenamiento y la transmisión de los datos de imágenes médicas.	Estandarizar el intercambio de una amplia gama de datos administrativos y clínicos.
Tipo de datos	Archivos binarios grandes (imágenes) junto con los metadatos asociados.	Datos de texto estructurados, incluidos los registros de los pacientes, los resultados de laboratorio y la información de facturación.
Casos de uso	Se usa principalmente en departamentos de imágenes médicas como radiología, cardiología e imágenes dentales.	Amplia aplicación en varios espacios de la atención médica más allá del diagnóstico por imágenes, como laboratorios, facturación, registros de pacientes, etc.
Interoperabilidad	Garantiza la compatibilidad e interoperabilidad de los equipos y sistemas de imágenes entre distintos proveedores.	Facilita el intercambio integral de datos entre diferentes sistemas y aplicaciones, ayudando a las operaciones de atención médica más amplias.
Características principales	Formatos de archivo estandarizados para imágenes y protocolos, integrando dispositivos de imágenes y sistemas PACS. Incluye metadatos para la información del paciente, el tipo de escaneo y los detalles del dispositivo.	Estándares de mensajería personalizables ha adaptándose a diversas necesidades de intercambio de información de atención médica.
Flexibilidad	Más técnico para las necesidades de procedimientos y equipos de diagnóstico por imágenes.	Altamente flexible y adaptable a las diferentes necesidades de intercambio de información hacia la atención médica.
Alcance técnico	Gestiona los datos basados en imágenes y se concentra en almacenar, recuperar y archivar archivos de imágenes complejos.	Gestiona el intercambio de información basado en texto en una extensa lista de servicios de salud.

1.6.5 Inteligencia Artificial

Machine Learning (ML)

El aprendizaje de máquina es una subdisciplina de inteligencia artificial que permite

el desarrollo de algoritmos y modelos donde las máquinas aprenden y mejoran automáticamente a partir de los datos proporcionados, sin ser programados para una tarea en específico. Este aprendizaje se logra a través de identificación de patrones en los datos con la creación de reglas predictivas [18].

Un modelo de aprendizaje de máquina se logra mediante un proceso de entrenamiento donde el conjunto de datos de entrada es expresado como (features) y sus resultados esperados, son expresados como etiquetas (labels) son utilizados para ajustar los parámetros internos del modelo. Dependiendo de la naturaleza del problema, el aprendizaje de la máquina puede clasificarse en tres categorías principales:

- **Supervisado:** El modelo se entrena con datos etiquetados, aprendiendo a asociar entradas con salidas específicas. Ejemplo: Clasificación de imágenes.
- **No supervisado:** El modelo trabaja con datos no etiquetados y busca patrones o agrupamientos en los datos. Ejemplo: agrupación.
- **Por refuerzo:** El modelo interactúa en un entorno dinámico donde aprender mediante sistemas de recompensa y penalización.

Los algoritmos más comunes en el aprendizaje de máquina incluyen regresión lineal, árboles de decisión, máquinas de soporte vectorial (por sus siglas en inglés, SVM), y métodos de conjunto como Random Forest y Gradient Boosting. Durante el entrenamiento, los algoritmos optimizan los parámetros del modelo utilizando métricas de error y técnicas como el descenso de gradiente, con el objetivo de mejorar la precisión en las predicciones [18].

Deep Learning

El aprendizaje profundo, (en inglés Deep Learning, es un tipo de machine learning enfocado en redes neuronales artificiales multicapa que le permiten aprender y extraer características sobre los datos. A diferencia del aprendizaje automático que solo tiene algoritmos basados en reglas manuales definidas, hoy el deep learning se servirá de la capacidad de los modelos para aprender automáticamente a partir de datos sin procesar [19].

Las redes neuronales artificiales aplicadas al aprendizaje profundo se estructuran como una sucesión de capas de neuronas conectadas. Cada capa recibe parcialmente -vamos a decir así- los datos y las características obtenidas serán utilizadas en capas posteriores para identificar patrones progresivamente más complejos. Este procedimiento continuará hasta que se logren las características más generales del conjunto de datos [20].

Deep learning puede ser aplicada en múltiples áreas tales como computación visual, procesamiento automático del lenguaje llano, traducción automática, detección avanzada y también clasificación del fraude y reconocimiento avanzado. Ya es conocido por su relación con la inteligencia artificial gracias a su habilidad autónoma del aprendizaje y precisión en patrones avanzados ha extendido su uso fundamentalmente tanto dentro del global deep learning como una herramienta valiosa dentro del machine learning.

Redes Neuronales

Las redes neuronales artificiales (ANN por su nombre en inglés) son un modelo computacional que imita la función y la estructura de un cerebro humano. Consiste en una red de nodos interconectados conocidos como neuronas artificiales donde se procesa la información de entrada y genera una salida.

En una red neuronal artificial, cada neurona recibe una o varias entradas las cuales procesa con una función de activación que determina el resultado de aquella neurona. La salida a estas neuronas se comunica a otras mediante conexiones ponderadas usadas para ajustar la contribución de cada neurona emisora.

En el entrenamiento de una red neuronal artificial los sistemas ajustan los pesos en las conexiones entre neuronas con el fin de minimizar las diferencias entre la deseada y real. Se utilizan algoritmos del tipo optimización tales como el “gradiente”.

El uso más frecuente incluye tareas dentro del aprendizaje automático en clasificación, visión por computadora, comprensión del lenguaje natural y análisis predictivo en series temporales [21].

CAPÍTULO 2

2. METODOLOGÍA

2.1 Diseño del Sistema de comunicación.

Para la implementación y el desarrollo del sistema de análisis de imágenes de RX de tórax se empleó contenedores para optimizar recursos computacionales y de esta manera facilitar la portabilidad y la escalabilidad del sistema.

Los contenedores son entornos virtualizados que permite encapsular la aplicación con sus dependencias, permitiendo una ejecución de forma aislada. Estas características facilitan una mayor eficiencia en relación con las máquinas virtuales tradicionales, los contenedores comparten recursos sin necesidad de replicar un sistema operativo.

Para la gestión de los contenedores se utilizó Docker, es una plataforma de código abierto utilizado en el ámbito de desarrollo. Docker permite garantizar la consistencia del entorno de ejecución entre distintos sistemas así evitando conflictos de dependencia entre versiones de software, facilitando la migración y el despliegue en distintas infraestructuras.

En la parte interna de los contenedores se ejecutan diferentes servicios necesarios para el funcionamiento del sistema. El backend se encuentra desarrollado en el lenguaje de programación *Python* utilizando un *Microframework Flask*, se encarga de gestionar la lógica, el sistema de autenticación de roles, la comunicación con la base de datos y el llamado del modelo de inteligencia artificial. Se utilizó el modelo *Chexnet* implementado con *TensorFlow Lite*, permitiendo el análisis automático de imágenes médicas.

La base de datos utilizada *PostgreSQL* es relacional, esta se ejecuta dentro de un contenedor independiente así formando un entorno aislado y controlado para el almacenamiento de los datos. Este contenedor almacena la información de los usuarios (administrador, médico general, médico especialista y paciente), el historial clínico de cada paciente y los resultados de las predicciones de imágenes de RX de

tórax.

2.1.1 Arquitectura del Sistema de Comunicación

En la Figura 2.1 se observa la arquitectura implementada para desarrollar el Sistema de Análisis de Imágenes de RX de tórax para Diagnósticos Remotos. Esta arquitectura está compuesta por contenedores Docker que encapsulan cada uno de los módulos funcionales del sistema, incluyendo el backend desarrollado en Flask, el modelo de inteligencia artificial CheXNet, el módulo de cifrado simétrico – Fernet y la base de datos PostgreSQL. Adicionalmente se presenta la comunicación entre los diferentes componentes, con los puertos utilizados para cada servicio interno.

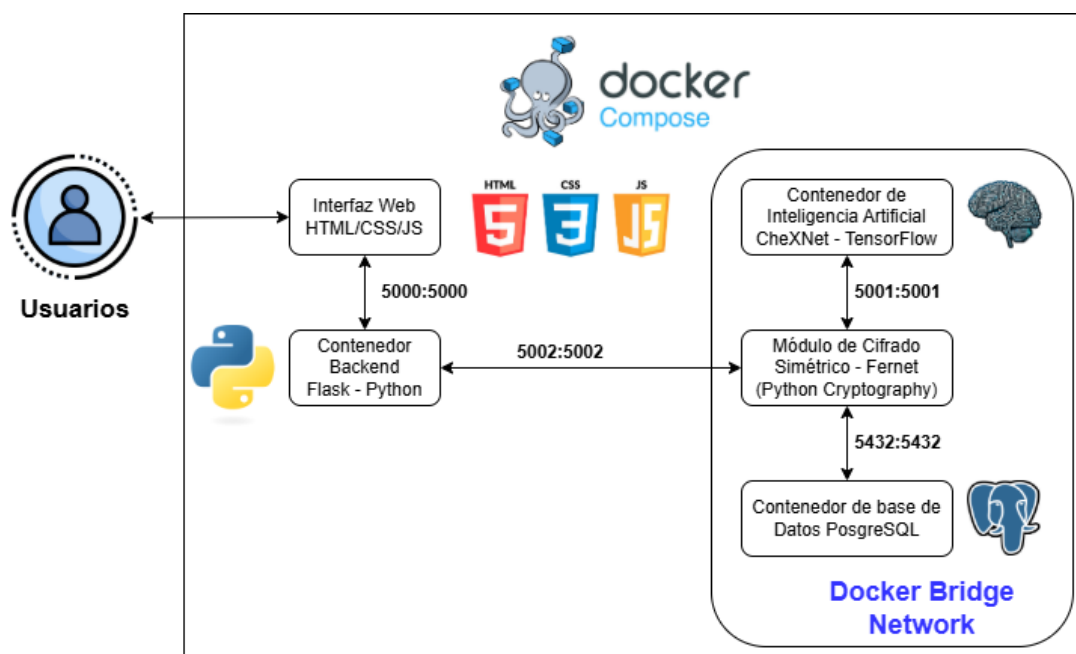


Figura 2.1 Arquitectura del sistema de comunicación

2.1.2 Usuarios

Los usuarios que interactúan con el sistema de análisis de imágenes médicas son:

- **Administrador:** Es el usuario responsable de gestionar el correcto funcionamiento de la plataforma, el acceso de los distintos usuarios. Sus funciones son la creación, modificación y eliminación de cuentas de médicos y pacientes.

- **Médico especialista:** Su rol es validar los diagnósticos generados por el modelo de *CheXNet* en caso de ser erróneos puede realizar correcciones en dicho diagnóstico para así garantizar la precisión del resultado.
- **Médico:** El médico general puede crear, editar historias clínicas, subir imágenes médicas, visualizar el diagnóstico generado por el modelo de *CheXNet* y observar si el diagnóstico fue validado por un médico especialista.
- **Paciente:** Su acceso es restringido, puede visualizar sus diagnósticos generados por el sistema y su historial clínico.

2.1.3 Políticas de Uso

En el sistema de análisis de imágenes médicas para accesos remotos, se consideró la seguridad, la confidencialidad de los datos sensibles que contienen las historias clínicas. Se describen sus políticas de uso.

- Solo pueden acceder los usuarios que fueron registrados y autenticados, la autenticación se gestiona a través del backend, utilizando credenciales suministradas por el administrador según el rol asignado.
- Cada usuario es responsable de mantener la confidencialidad de sus credenciales incluyendo el nombre de usuario y su contraseña en caso de pérdida deberá notificar al administrador del sistema
- Las historias clínicas, las imágenes analizadas, los usuarios y permisos están almacenados en la base de datos PostgreSQL que se encuentra aislada del contenedor Docker estos son accesibles desde los servicios internos de la red Docker Bridge.
- El sistema permite acceder a distintas funcionalidades en referencia al tipo de usuario
 - El médico general puede crear o editar historias clínicas, subir imágenes médicas y visualizarlas.
 - El médico especialista validará y/o modificará los diagnósticos generados por el sistema.
 - El paciente puede visualizar su historial médico y los resultados procesados por el sistema.

- El administrador supervisa el funcionamiento adecuado del sistema y gestiona los usuarios.

Estas políticas permiten que el sistema cumpla con los criterios éticos y normativos en el manejo de información médica, promoviendo un entorno seguro en el diagnóstico remoto asistido por inteligencia artificial.

2.2 Infraestructura Tecnológica

La infraestructura utilizada en el sistema de análisis de imágenes de RX de tórax se basa en el uso de contenedores Docker y tecnologías de código abierto esto permite garantizar características claves como la escalabilidad, portabilidad, tolerancia a fallos y facilidad de despliegue en distintos entornos tanto locales como en la nube

El uso de Docker Compose como herramienta de orquestación facilita la configuración y ejecución conjunta de los servicios necesarios, lo cual asegura una interacción fluida entre los módulos que conforman el sistema: backend, base de datos y modelo de inteligencia artificial. Se visualiza en el Código 2.1 la configuración de los servicios mediante docker-compose.yml

```
version: '3.8'

services:
  backend:
    build: .
    ports:
      - "5000:5000"
    depends_on:
      - postgres
      - modelo_ia
    networks:
      - tesisnet

  postgres:
    image: postgres:13
    environment:
      POSTGRES_DB: tesis_db
      POSTGRES_USER: tesis_user
      POSTGRES_PASSWORD: tesis123
    ports:
```

Código 2.1 Configuración de Servicios

2.2.1 Arquitectura del Sistema de Comunicación Contenedor Backend (Flask – Python)

El *backend* fue desarrollado en *Python* utilizando el *Microframework Flask*, y se encuentra encapsulado en un contenedor Docker. Este componente actúa como núcleo del sistema, permitiendo gestionar:

- El flujo de autenticación de usuarios.
- Las solicitudes del frontend (interfaz web).
- La comunicación con el modelo de inteligencia artificial para el análisis de imágenes médicas.
- La conexión con la base de datos *PostgreSQL*.
- La aplicación del cifrado simétrico Fernet sobre los resultados generados y los datos sensibles.

El contenedor expone el puerto 5000 para la recepción de peticiones externas desde la interfaz web. Se visualiza en el Código 2.2 un fragmento del Backend (Flask – Python)

```
@app.route('/subir_imagen', methods=['POST'])
def subir_imagen():
    imagen = request.files['imagen']
    nombre_archivo = secure_filename(imagen.filename)
    ruta_imagen = os.path.join("imagenes", nombre_archivo)
    imagen.save(ruta_imagen)

    # Realiza análisis con IA
    resultado = chexnet_predict(ruta_imagen)

    # Aplica cifrado homomórfico (simplificado para la tesis)
    resultado_cifrado = cifrar_resultado(resultado)

    # Guarda en PostgreSQL
    conn = conectar_db()
    cursor = conn.cursor()
    cursor.execute("""
        INSERT INTO diagnosticos (usuario_id, imagen, resultado)
        VALUES (%s, %s, %s)
        """, (session['usuario_id'], nombre_archivo, resultado_cifrado))
```

Código 2.2 Fragmento de Backend (Flask – Python).

2.2.2 Contenedor del modelo de inteligencia artificial (CheXNet – TensorFlow Lite)

El modelo de inteligencia artificial utilizado es una adaptación de *CheXNet*, entrenado para el análisis de imágenes de RX de tórax se, implementado con *TensorFlow* Lite para garantizar eficiencia y bajo consumo de recursos. Este módulo se encuentra desplegado como un contenedor independiente que se comunica con el *backend* a través del puerto 5001.

Cuando el médico sube una imagen de RX de tórax, el *backend* envía la imagen al modelo CheXNet, recibe el diagnóstico y posteriormente aplica las medidas de cifrado antes de almacenar los resultados. Se visualiza en el Código 2.3 un fragmento del modelo CheXNet.

```
import tenseal as ts

def cifrar_resultado(vector):
    ctx = ts.context(ts.SCHEME_TYPE.CKKS, poly_modulus_degree=8192, coeff_mod_bit_sizes=[40, 21, 21, 40])
    ctx.global_scale = 2**21
    ctx.generate_galois_keys()
    encrypted_vector = ts.ckks_vector(ctx, vector)
    return encrypted_vector.serialize()
```

Código 2.3 Fragmento modelo CheXNet

2.2.3 Contenedor de Base de Datos (PostgreSQL)

El sistema utiliza PostgreSQL como motor de base de datos relacional, ejecutándose en un contenedor separado expuesto internamente en el puerto 5432. Donde se almacenan:

- Los datos de los usuarios y sus roles.
- Las Historias Clínicas Electrónicas (HCE).
- Los resultados diagnósticos cifrados.
- La trazabilidad del acceso a la información médica.

La conexión con la base de datos es gestionada exclusivamente por el backend,

manteniéndose dentro de una red interna segura (*Docker Bridge Network*). Controlado para el diagnóstico remoto asistido por inteligencia artificial.

```
tesis_db=# kathy@Kathy:~/tesis_medica$ docker exec -it postgres_tesis bash
root@68d4341d6796:/# psql -U tesis_user -d tesis_db
psql (14.18 (Debian 14.18-1.pgdg120+1))
Type "help" for help.

tesis_db=# SELECT * FROM usuarios;
tesis_db=# SELECT * FROM imagenes_medicas;
tesis_db=# /dt
tesis_db=# \dt

      List of relations
Schema |      Name      | Type  | Owner
-----+-----+-----+-----
public | historia_clinica | table | tesis_user
public | imagenes_medicas | table | tesis_user
public | usuarios         | table | tesis_user
(3 rows)
```

Figura 2.3 Contenedor de base de datos

2.3 Interfaz web del sistema

Se desarrolló una interfaz web funcional que permite a los diferentes usuarios (administrador, médicos y pacientes) interactuar con los módulos del sistema de análisis de imágenes de RX de tórax. La interfaz fue implementada utilizando tecnologías estándar del lado del cliente: *HTML5*, *CSS3* y *JavaScript*, sin *Frameworks* externos, con el objetivo de mantener una estructura ligera, portable.



Figura 2.4 Interfaz principal de Diagnóstico de Imágenes Médicas

2.3.1 Integración de la interfaz web con el backend

La interfaz no se encuentra en un contenedor independiente, sus archivos están alojados directamente en el contenedor backend (Flask – Python), dentro de la carpeta *static/*, lo que permite mostrar directamente al usuario el contenido web sin necesidad de un servidor externo. Flask se encarga de exponer las rutas y renderizar las vistas HTML según el rol del usuario autenticado, lo cual permite una lógica dinámica del lado del servidor.

2.3.2 Funcionalidades de la interfaz web

La interfaz web del sistema ha sido diseñada para ofrecer una experiencia intuitiva, accesible y multiplataforma desde cualquier navegador moderno. Cada formulario o vista está conectado con rutas definidas en Flask (@app.route), las cuales procesan y gestionan la información ingresada por el usuario.

Las funcionalidades clave disponibles desde la interfaz web son:

- **Carga de imagen médica:** A través de un formulario con el campo input type="file", permite seleccionar una imagen de RX de tórax y enviarla al backend para su análisis mediante el modelo CheXNet.
- **Visualización del resultado diagnóstico:** Después que el modelo de inteligencia artificial ha procesado la imagen, el resultado se muestra de forma directa y clara al usuario autorizado.
- **Registro de historia clínica:** Incluye campos como nombres, número de cédula, género, fecha de nacimiento, antecedentes y otros datos clínicos. Esta información es almacenada de forma segura en la base de datos PostgreSQL.
- **Autenticación y control de sesión:** Mediante formularios de inicio de sesión que verifican las credenciales y redirigen al usuario al panel correspondiente según su rol (médico, paciente o administrador).

2.3.3 Creación de usuario automático - perfil paciente

Cuando el médico general procede a crear una nueva historia clínica en el sistema se crea de forma automática la creación del usuario tipo paciente

Este usuario se genera a partir de los datos ingresados, utilizando la cédula como nombre de usuario y una contraseña predeterminada que antepone el prefijo SYS seguido del número de cédula del paciente.

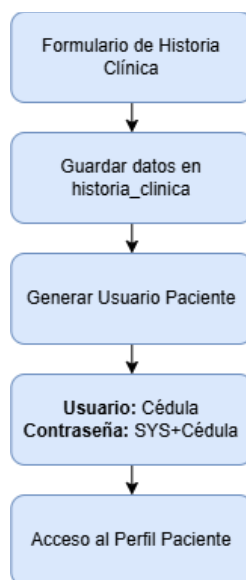


Figura 2.3 Diagrama de creación del perfil paciente

Este proceso facilita a que cada paciente, tenga acceso inmediato a su perfil donde podrá consultar su historial clínico los resultados del análisis de las imágenes por inteligencia artificial y la información relevante.

2.4 Implementación del módulo de IA para el análisis de imágenes médicas.

El sistema propuesto utiliza un módulo de inteligencia artificial basado en el modelo CheXNet, es un algoritmo de aprendizaje profundo especializado en el análisis de radiografías de tórax. Este modelo fue desarrollado por Rajpurkar [22], y utiliza una arquitectura de red neuronal convolucional. El algoritmo ha demostrado resultados comparables con los diagnósticos realizados por radiólogos en la detección de enfermedades pulmonares más comunes, como la neumonía.

En la implementación del sistema se utiliza el modelo de CheXNet de libre acceso, el cual fue integrado en contenedores Docker. El modelo se ejecuta como un servicio autónomo dentro de un contenedor separado, lo que permite su escalabilidad, facilita su mantenimiento y evita comprometer la estabilidad del backend principal.

2.4.1 Proceso de análisis con IA

El proceso de análisis de imágenes de RX de tórax se desarrolla de la siguiente forma.

- 1) El médico general accede a la plataforma web e ingresa al formulario de carga de imágenes de RX de tórax.
- 2) Una vez seleccionada la imagen de Rx de tórax, esta se envía al backend en Flask, donde es preprocesada y posteriormente dirigida al módulo de inteligencia artificial.
- 3) El modelo CheXNet recibe imágenes de Rx de tórax en formato digital (DICOM, JPG, PNG), realiza el proceso de inferencia, y retorna un vector con 14 probabilidades asociadas a las posibles patologías detectadas por el algoritmo CheXNet.
- 4) El resultado es almacenado en la base de datos PostgreSQL, asociado a la historia clínica del paciente.
- 5) El médico especialista accede a este resultado para su validación y, en caso de ser aprobado, queda disponible para visualización por el paciente en su perfil.

2.4.2 Integración técnica del modelo CheXNet en el sistema

El modelo está implementado en PyTorch y ejecutado dentro de un entorno virtual controlado, lo que garantiza compatibilidad con las dependencias necesarias. Para facilitar su integración con el backend, se ha desarrollado un script llamado *chexnet_predict()* que permite:

- Cargar los pesos del modelo pre entrenado,
- Preprocesar la imagen entrante,
- Realizar la predicción y devolver el diagnóstico con las probabilidades asociadas.

A partir del vector resultante, el sistema identifica la clase con mayor probabilidad como diagnóstico preliminar. Esta predicción no reemplaza la validación clínica, sino que actúa como soporte para la toma de decisiones del médico especialista, quien puede confirmar o ajustar el resultado.

2.5 Encriptación de datos sensibles.

Se implemento el Código 2.4 para garantizar la seguridad de los datos médicos sensibles del sistema de análisis de imágenes de Rx de tórax, se optó por el uso de la biblioteca *cryptography* en *Python*, la que permite encriptar los datos antes de ser almacenados en la base de datos y desencriptados temporalmente antes de ser visualizados por el usuario según sus roles. Los datos del paciente considerados sensibles son: cedula, nombres, apellidos, genero dirección, correo electrónico, antecedentes médicos, imágenes médicas.

```
from cryptography.fernet import Fernet
import os

CLAVE_PATH = "clave_secreta.key"

def generar_o_cargar_clave():
    if not os.path.exists(CLAVE_PATH):
        clave = Fernet.generate_key()
        with open(CLAVE_PATH, 'wb') as archivo_clave:
            archivo_clave.write(clave)
```

Código 2.4 Fragmento de encriptación y desencriptación – Fernet

En la Figura 2.4 se muestra el esquema de funcionamiento del algoritmo de encriptación.

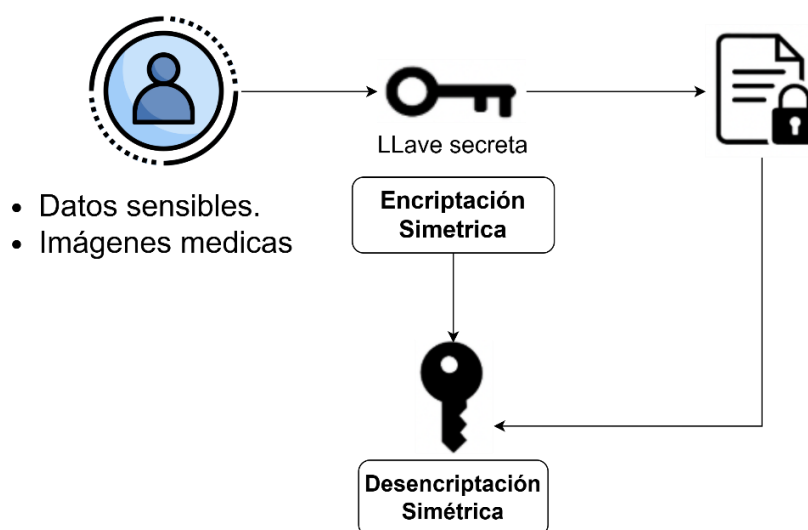


Figura 2.4 Encriptación simétrica Fernet

2.6 Validación remota del sistema.

Para simular un entorno en la nube, se utilizó el software Ngrok el cual permite exponer nuestro servidor local a través de una dirección IP pública en nuestro caso es la siguiente <https://ca5d-191-99-66-57.ngrok-free.app/>, la misma que nos permitió realizar pruebas de accesibilidad mediante datos móviles así evaluando el rendimiento de la carga de imágenes médicas, respuesta del sistema en tiempo real. En la Figura 2.5 visualizamos la simulación de acceso remoto mediante Ngrok.



Figura 2.5 Acceso remoto al sistema mediante Ngrok

CAPÍTULO 3

3. RESULTADOS

3.1 Funcionamiento del sistema de comunicación

Se procedió a verificar la infraestructura del sistema de comunicación validando que los contenedores han sido inicializados de forma correcta tal como se muestra en la Figura 3.1

```
kathy@Kathy:~/tesis_medica$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
8bef3326eb06	postgres:14	"docker-entrypoint.s..."	12 hours ago	Up About a minute	0.0.0.0:5433->5432/tcp, [::]:5433->5432/tcp
a597f83d2818	tesis_medica-backend-1	"python app.py"	12 hours ago	Up About a minute	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp

Figura 3.1. Contenedores Docker

Se comprobó el correcto funcionamiento del contenedor tesis_medica-backend-1 a través del URL <http://localhost:5000>, tal y como se muestra en la Figura 3.2.



Figura 3.2. Contenedores Docker

3.2 Verificación de conexión entre Backend y Base de Datos

Para garantizar que el contenedor que ejecuta *PostgreSql* se encuentra activo y accesible para el backend en *Flash*, se utilizó el siguiente comando:

```
docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
postgres_tesis
```

mostrándonos como resultado la siguiente dirección IP 172.18.0.2 como se muestra en la Figura 3.3. Esta dirección es utilizada en el *backend* para establecer conexión directa con la base de datos PostgreSQL, lo cual valida el correcto funcionamiento de la arquitectura basada en contenedores y la red interna Docker.

```
kathy@Kathy:~/tesis_medica/backend$ cd ..
kathy@Kathy:~/tesis_medica$ docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' postgres_tesis
172.18.0.2
```

Figura 3.2. Contenedores Docker

El sistema fue desarrollado como una aplicación web utilizando tecnologías estándar como *HTML5*, *CSS3* y *Javascript* para ofrecer una experiencia clara y responsiva como se muestra en la Figura 3.3

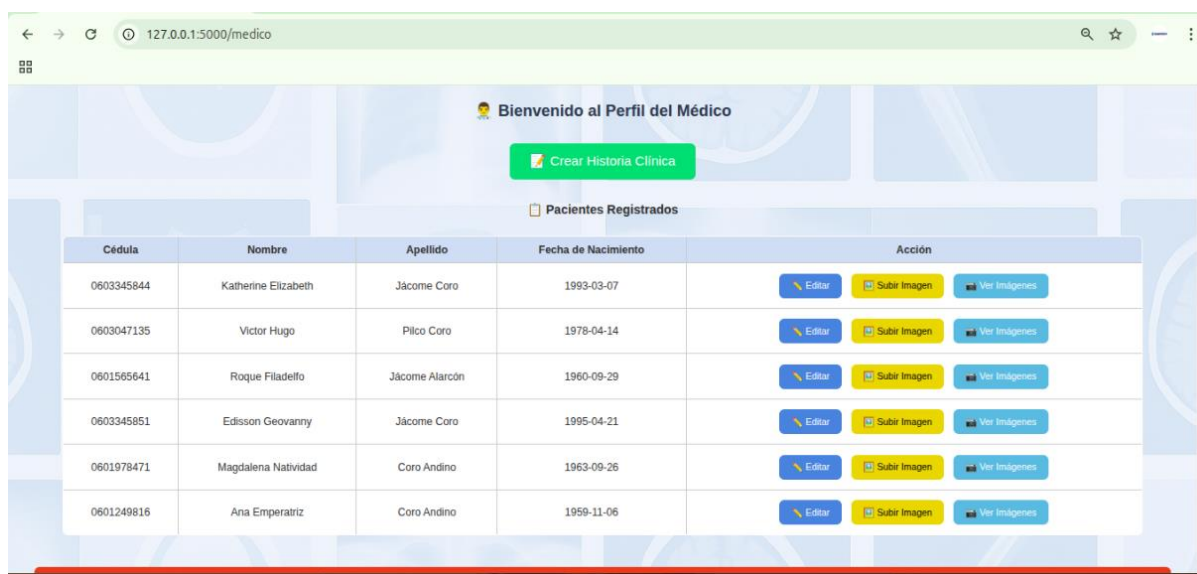


Figura 3.3. Interfaz Web accedida desde el navegador

Para validar su correcto funcionamiento se puede observar en la Figura 3.4 la interfaz

de médico general qué le permite tener distintas funciones entre ellas la creación de historia clínicas como muestra en la Figura 3.5, al momento de la creación de la historia clínica automáticamente se crea el perfil de paciente siendo su **Usuario:** #cedula y su **contraseña:** SYS#cedula como se observa en la Figura 3.6.

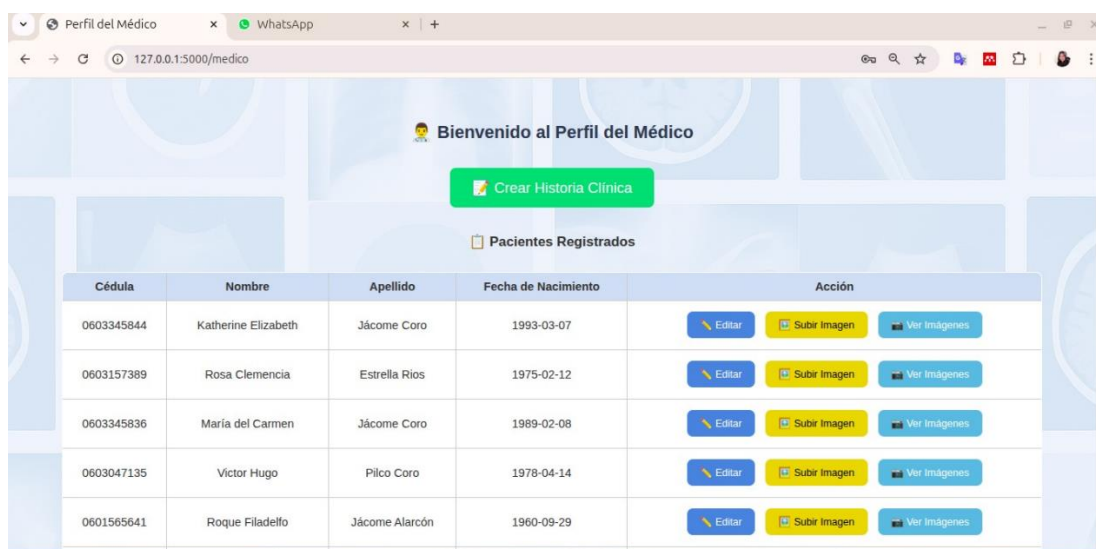


Figura 3.4. Interfaz médico general

Formulario Historia Clínica

Cédula:

Nombres:

Apellidos:

Fecha de Nacimiento:

Edad:

Género:

Figura 3.5. Interfaz historia clínica

```
kathy@Kathy: ~/tesis_medica
kathy@Kathy: ~/tesis_medica
kathy@Kathy:~/tesis_medica$ docker exec -it postgres_testis bash
root@88bef3326eb06:/# psql -U tesis_user -d tesis_db
psql (14.18 (Debian 14.18-1.pgdg120+1))
Type "help" for help.

tesis_db=# SELECT * FROM historia_clinica;
tesis_db=# SELECT * FROM usuarios;
 id | nombres | apellidos | usuario | password | rol
-----+-----+-----+-----+-----+-----
 1 | Adnin   | Principal | admin   | admin123 | admin
 2 | Carlos  | Pérez    | medico1 | medico123 | medico_general
 3 | Juan    | Especialista | especialista1 | clave123 | medico_especialista
 4 | Ana Emperatriz | Coro Andino | 0601249816 | SYS0601249816 | paciente
 5 | Magdalena Natividad | Coro Andino | 0601978471 | SYS0601978471 | paciente
 6 | Edis Geovanny | Jácome Coro | 0603345851 | SYS0603345851 | paciente
 7 | Roque Filadelfo | Jácome Alarcón | 0601565641 | SYS0601565641 | paciente
 8 | Katherine Elizabeth | Jácome Coro | 0603345844 | SYS0603345844 | paciente
 9 | Victor Hugo | Pilco Coro | 0603047135 | SYS0603047135 | paciente
```

Figura 3.6. Tabla usuarios

En el perfil de médico general, además cuenta con la función de editar historia clínica como muestra en la Figura 3.7. que fue editado el nombre del paciente y en la Figura 3.8 se puede observar que contenía el nombre erróneo del paciente.

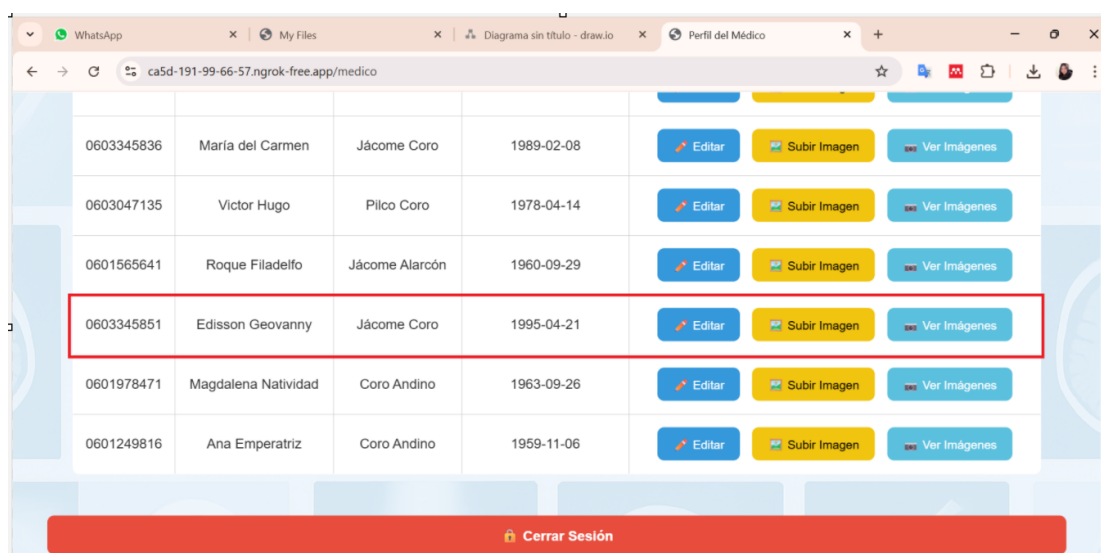


Figura 3.7. Editar historia clínica


```

kathy@Kathy: ~/tesis_medica
kathy@Kathy: ~/tesis_medica
kathy@Kathy:~/tesis_medica$ docker exec -it postgres_testis bash
root@08bef3326eb06:/# psql -U tesis_user -d tesis_db
psql (14.18 (Debian 14.18-1.pgdg120+1))
Type "help" for help.

tesis_db=# SELECT * FROM historia_clinica;
tesis_db=# SELECT * FROM usuarios;
id | nombres | apellidos | usuario | password | rol
-----+-----+-----+-----+-----+-----
1 | Admin | Principal | admin | admin123 | admin
2 | Carlos | Pérez | medico1 | medico123 | medico_general
3 | Juan | Especialista | especialista1 | clave123 | medico_especialista
4 | Ana Emperatriz | Coro Andino | 0601249816 | SYS0601249816 | paciente
5 | Maedalea Natividad | Coro Andino | 0601978471 | SYS0601978471 | paciente
6 | Edis Geovanny | Jácome Coro | 0603345851 | SYS0603345851 | paciente
7 | Roque Filadelfo | Jácome Alarcon | 0601565641 | SYS0601565641 | paciente

```

Figura 3.8. Base de datos de historia clínica

Otra función del médico general le permite adjuntar imágenes médicas para ser diagnosticadas como se muestra en la Figura 3.9, se puede observar que se almacena la imagen en una dirección que contiene el número de cédula del paciente. Después de dar un clic en analizar imagen nos muestra temporalmente el análisis del algoritmo de Inteligencia Artificial.

Para verificar el funcionamiento del algoritmo CheXNet se utilizó imágenes del conjunto de datos de radiografías de tórax de los NIH [23], se encuentran en formato PNG, sus imágenes están desidentificadas esta información está disponible en: <https://cloud.google.com/healthcare-api/docs/resources/public-datasets/nih-chest?hl=es-419>

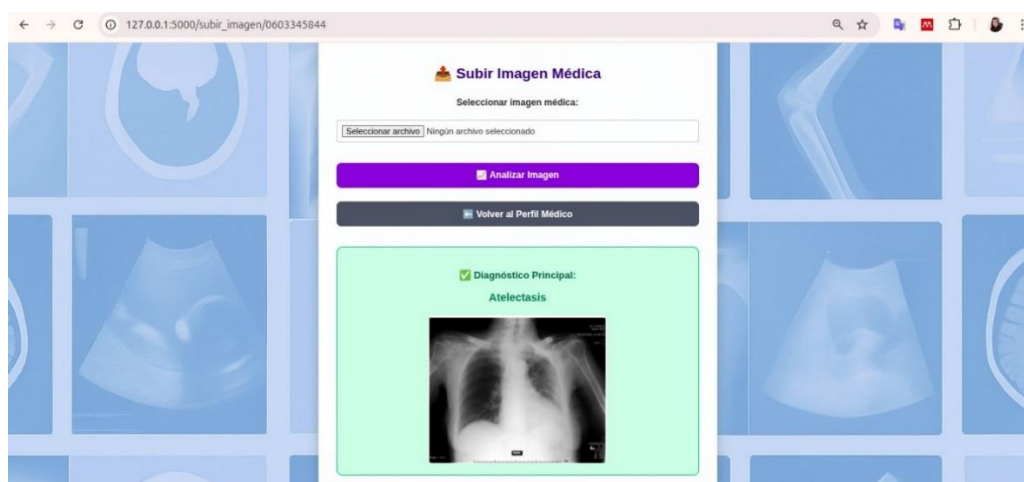


Figura 3.9. Subir imagen médica

Su última función es observar las imágenes analizadas por el algoritmo de Inteligencia Artificial como se muestra en la Figura 3.10, cumpliendo con el objetivo específico 2 de ejecutar algoritmos de inteligencia artificial. Además, se puede observar si el diagnóstico de la imagen fue validado por un especialista como indica la Figura 3.11.

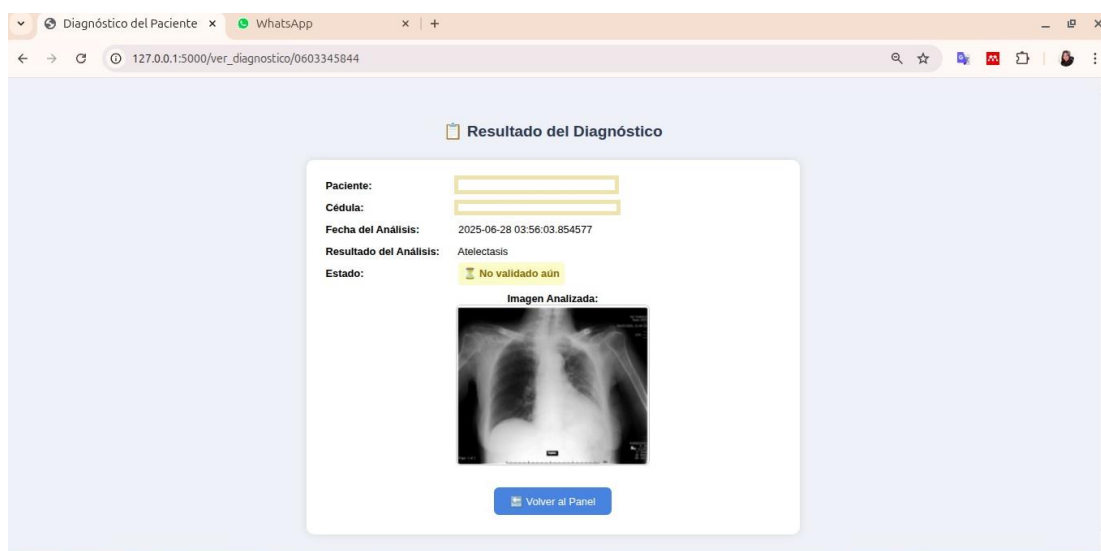


Figura 3.10. Diagnóstico por validar

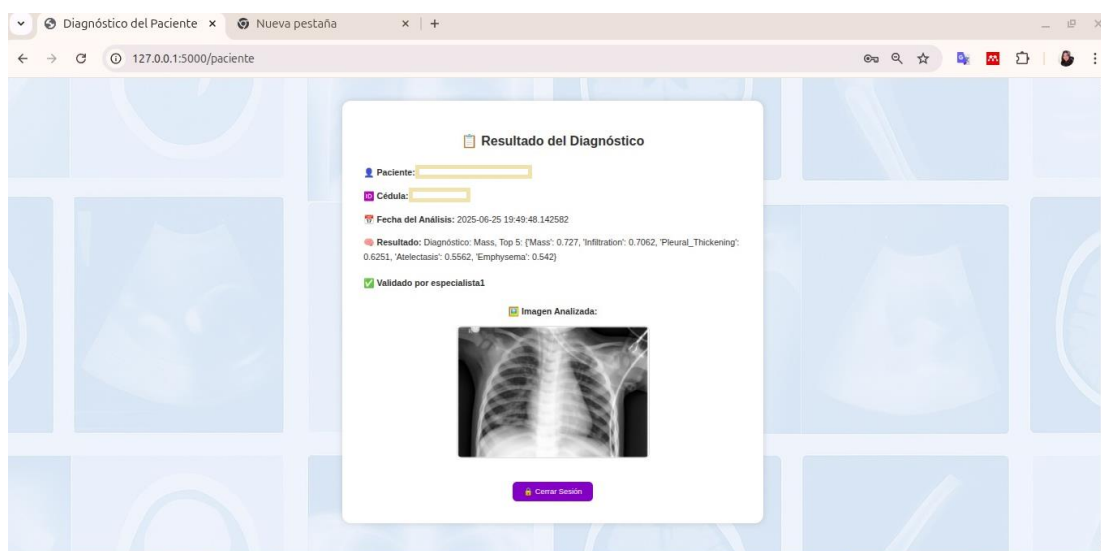


Figura 3.11. Diagnóstico validado

En la Figura 3.12 se observa la tabla de imágenes médicas, donde se almacena la información si fue validada la imagen analizada por un

especialista conjuntamente con la fecha de validación.



```

kathy@Kathy: ~/tesis_medica
kathy@Kathy: ~/tesis_medica

id | cedula | imagen_path | fecha_subida | validado_por | resultado_ia | fecha_analisis | fe
-----+-----+-----+-----+-----+-----+-----+
diagnostico_validado
cha_validacion
-----+-----+-----+-----+-----+-----+-----+
1 | 0601249816 | static/uploads/P.png | Diagnóstico: Mass, Top 5: {'Mass': 0.727, 'Infiltration': 0.7062, 'Pleural_Thickening': 0.6251, 'Atelectasis': 0.5562, 'Emphysema': 0.542} | 2025-06-25 19:49:48.142582 | Diagnóstico: Mass, Top 5: {'Mass': 0.727, 'Infiltration': 0.7062, 'Pleural_Thickening': 0.6251, 'Atelectasis': 0.5562, 'Emphysema': 0.542} | especialista1 | 2025-06-25 19:49:48.142582 | 2025-06-25 21:09:24.227619
2 | 0603345851 | static/uploads/C.png | Diagnóstico: Pneumothorax, Top 5: {'Pneumothorax': 0.6992, 'Cardiomegaly': 0.6454, 'Fibrosis': 0.5621, 'Hernia': 0.5412, 'Edema': 0.5106} | 2025-06-26 01:49:06.112978 | | | 2025-06-26 01:49:06.112978 |

(2 rows)
(END)
Archivos

```

Figura 3.12. Diagnóstico validado

En la Figura 3.13 se muestra la interfaz del médico especialista dónde se despliega las imágenes a validar, mostrándole el mensaje de diagnóstico aún no validado en este apartado. El médico puede validar si está correcto o caso contrario puede editarlo el diagnóstico. En la Figura 3.14 se puede observar que el diagnóstico fue validado correctamente.



Figura 3.13. Imagen por validar

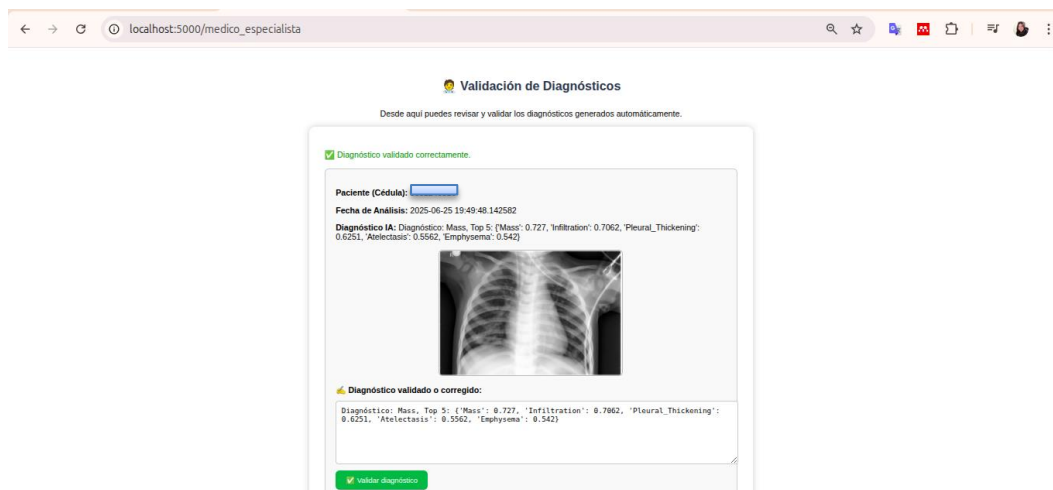


Figura 3.14. Imagen validada

En la Figura 3.15 se puede observar el perfil del paciente donde muestra los datos del paciente, la fecha del análisis de la imagen médica realizado por el sistema y la imagen analizada.

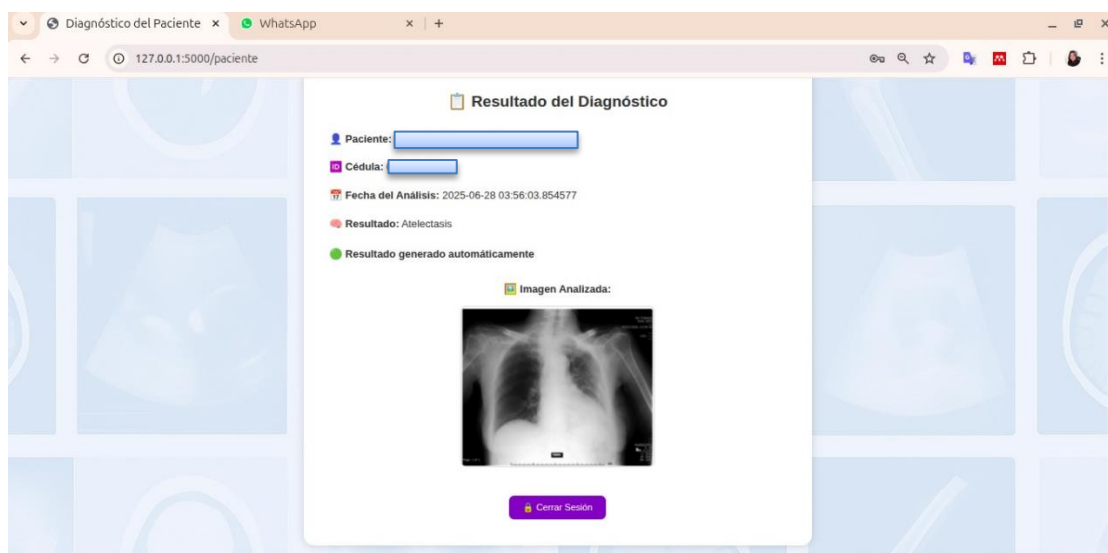


Figura 3.15. Imagen validada

3.3 Validación de la encriptación y desencriptación de los datos sensibles.

En la Figura 3.16 se puede observar la tabla de la historia clínica almacenada en PostgreSQL donde se puede evidenciar la encriptación de los datos sensibles de los pacientes.

En la Figura 3.17 se puede observar que las imágenes médicas se encuentran encriptadas en la tabla de imágenes médicas y en la Figura 3.18 se observa que la imagen no puede ser vista con ningún visualizador de imágenes.

```
kathy@Kathy: ~/tesis_medica
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
kathy@Kathy: ~/tesis... x kathy@Kathy: ~/tesis... x kathy@Kathy: ~/tesis... x kathy@Kathy: ~/tesis... x kathy@Kathy: ~/tesis... x kathy@Kathy: ~/tesis... x
:33.221144 | None
| especialista1 | | 2025-06-29 16:30:07.15209 | f
13 | 0603345848 | static/uploads/encrypted_N.png | | {'diagnostico': 'Pneumothorax', 'top_resultados': {'Pneumothorax': 0.6429, 'Hernia': 0.5976, 'Pneumonia': 0.5568, 'Mass': 0.5444, 'Consolidation': 0.4846}} | 2025-06-29 16:45:58.546608 | {'diagnostico': 'Pneumothorax', 'top_resultados': {'Pneumothorax': 0.6429, 'Hernia': 0.5976, 'Pneumonia': 0.5568, 'Mass': 0.5444, 'Consolidation': 0.4846}} | especialista1 | 2025-06-29 16:45:58.546623 | 2025-06-29 16:46:42.920376 | t
12 | 0603345844 | static/uploads/encrypted_N.png | | {'diagnostico': 'Fibrosis', 'top_resultados': {'Fibrosis': 0.7877, 'Hernia': 0.7479, 'Cardiomegaly': 0.6811, 'Edema': 0.5531, 'Mass': 0.5281}} | 2025-06-29 16:26:10.488478 | {'diagnostico': 'Fibrosis', 'top_resultados': {'Fibrosis': 0.7877, 'Hernia': 0.7479, 'Cardiomegaly': 0.6811, 'Edema': 0.5531, 'Mass': 0.5281}} | especialista1 | 2025-06-29 16:46:46.953847 | f
31 | 0603345844 | static/uploads/encrypted_ENCRI.jpg | | {'diagnostico': 'Emphysema', 'top_resultados': {'Emphysema': 0.6814, 'Hernia': 0.6559, 'Infiltration': 0.6523, 'Atelectasis': 0.6421, 'Edema': 0.5988}} | 2025-06-29 21:03:53.505502 |
| | 2025-06-29 21:03:53.505505 | | t
15 | 0603345844 | static/uploads/encrypted_204c3cbd-4cce-4d51-88ff-a6737c4d1ba0.jpeg | | {'diagnostico': 'Pneumothorax', 'top_resultados': {'Pneumothorax': 0.7359, 'Effusion': 0.6655, 'Consolidation': 0.638, 'Emphysema': 0.6092, 'Atelectasis': 0.5986}} | 2025-06-29 21:53:34.458486 |
| | 2025-06-29 21:53:34.45849 | | t
16 | 0603157389 | static/uploads/encrypted_204c3cbd-4cce-4d51-88ff-a6737c4d1ba0.jpeg | | {'diagnostico': 'Pneumothorax', 'top_resultados': {'Pneumothorax': 0.7359, 'Effusion': 0.6655, 'Consolidation': 0.638, 'Emphysema': 0.6092, 'Atelectasis': 0.5986}} | 2025-06-29 22:03:34.432785 |
| | 2025-06-29 22:03:34.432789 | | t
(8 rows)
(END)
```

Figura 3.17. Encriptación de los datos sensibles en la tabla imágenes médicas.

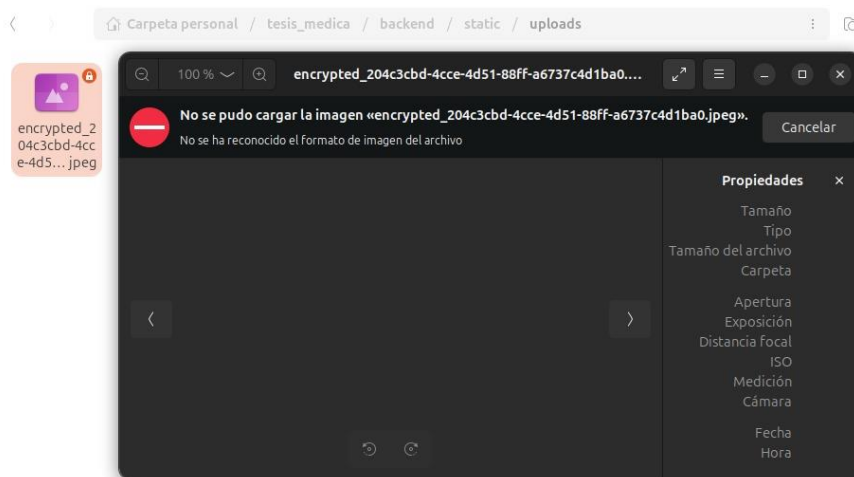


Figura 3.18. Imagen encriptada.

Para la visualización de la interfaz Web, los datos son descifrados temporalmente así garantizando la confidencialidad, integridad y cumplimiento con las normas internacionales de protección de datos.

3.4 Validación remota del Sistema

En la Figura 3.19 se observa la configuración de Ngrok para exponer el servidor local a través de una dirección pública, así facilitando simular el servidor en la nube para acceder en tiempo real a la interfaz.

```

kathy@Kathy: ~/tesis_medica  kathy@Kathy: ~/tesis_medica...  kathy@Kathy: ~/tesis_medica...  kathy@Kathy: ~/tesis_medica  kathy@Kathy: ~/tesis_medica  (Ctrl+C to quit)
ngrok
Want to hang with ngrok on our new Discord? http://ngrok.com/discord

Session Status      online
Account             kathyjc1993@gmail.com (Plan: Free)
Update             update available (version 3.23.3, Ctrl-U to update)
Version             3.23.2
Region             South America (sa)
Latency             186ms
Web Interface       http://127.0.0.1:4040
Forwarding           https://ca5d-191-99-66-57.ngrok-free.app -> http://localhost:5000

Connections         ttl    opn    rt1    rt5    p50    p90
                   45     0      0.00   0.00   0.02   0.04

HTTP Requests
-----
05:55:36.221 -36 POST /login                200 OK
05:55:36.795 -36 GET /especialista            200 OK
05:55:29.035 -29 GET /                      200 OK
05:55:29.665 -29 GET /static/fondo3.png      304 NOT MODIFIED
05:55:28.456 -28 POST /logout                302 FOUND
05:44:03.632 -03 GET /editar_historia/3      200 OK

```

Figura 3.19. Exposición de IP pública con Ngrok

En la Figura 3.20 se observa que gracias a la IP proporcionada por Ngrok, se pudo validar la accesibilidad remota para realizar pruebas mediante un dispositivo utilizando datos móviles, donde la interacción fue fluida permitiendo comprobar que el sistema mantiene la funcionalidad en la transferencia de imágenes, el análisis de las mismas, la visualización de diagnósticos y el acceso a las historias clínicas en tiempo real.

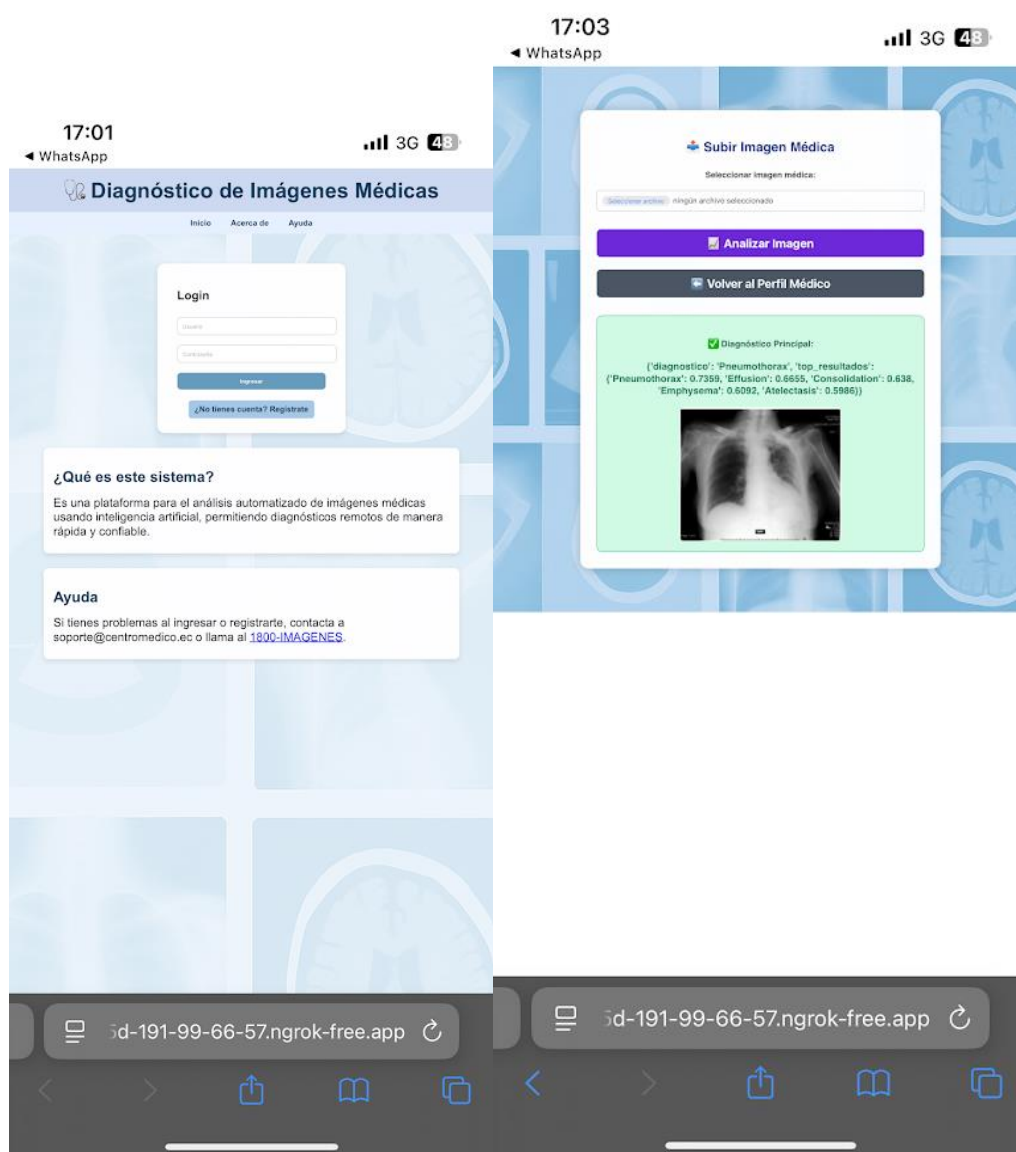


Figura 3.20 Uso del sistema desde un dispositivo móvil

3.5 Validación Funcional de Sistemas

Para validar el correcto funcionamiento del sistema de análisis de imágenes médicas se realizaron pruebas funcionales sobre los distintos módulos. Las pruebas se enfocaron en verificar la operación esperada, como la autenticación de usuarios registro de historia clínica, edición de historia clínica, carga y análisis de imágenes médicas, visualización de resultados según el rol asignado. Como se muestra en la Tabla 3.1 un resumen de las funcionalidades evaluadas.

Tabla 3.1: Funcionalidades evaluadas.

MÓDULO EVALUADO	ACCIÓN REALIZADA	RESULTADO ESPERADO	RESULTADO OBTENIDO
Inicio de Sesión	Ingreso de credenciales válidas	Acceso según el usuario	Acceso exitoso con dirección correcta
Registro de historia clínica	Ingreso de datos del paciente	Almacenamiento en base de datos	Datos guardados correctamente
Edición de historia clínica	Editar datos del paciente	Almacenamiento en base de datos	Datos guardados correctamente
Carga de imagen médica	Seleccionar la imagen	Imagen almacenada y enviada al modelo de IA	Imagen guarda correctamente
Análisis de imagen (IA)	Ejecución del modelo cheXNET	Clasificación automática, mostrado en pantalla	Resultado desplegado correctamente
Validación de diagnóstico	Revisión por parte del médico especializado.	Confirmación o Corrección	Validación realizada y almacenamiento en la base de datos
Visualización del paciente	Acceso al historial de resultados	Consulta de la historia clínica	Datos visibles correctamente

3.6 Métricas básicas de desempeño.

Aunque el sistema se encuentra implementada de forma local se realizaron pruebas de empíricas para estimar el tiempo de respuesta de los procesos

principales. Estas métricas permiten evaluar la viabilidad operativa del sistema y su potencial para escalar a un entorno en la nube. Como indica la Tabla 3.2.

Tabla 3.2. Métricas de desempeño

Proceso	Promedio estimado	Observación
Tiempo de carga del sistema web	1.8 segundos	Desde el acceso al navegador hasta cargar la pantalla principal de login
Tiempo de autenticación	1.2 segundos	Desde el clic en login hasta redireccionar al panel de usuario.
Tiempo de análisis de imagen IA	4.3 segundos	desde la carga de imagen hasta la visualización del diagnóstico
tiempo guardado en base de datos	< 1 segundo	validado mediante inserciones observadas

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

- Se desarrolló un sistema funcional de análisis de imágenes médicas que se puede acceder desde un navegador web, el cual permite realizar diagnósticos remotos con asistencia de inteligencia artificial. A pesar de que la implementación final debió hacerse en local debido a restricciones de recursos, la arquitectura basada en Docker está completamente lista para la nube y cumple con el requisito de portabilidad.
- El sistema incorpora un modelo de acceso con roles diferenciados (administrador, médicos y paciente) que salvaguarda la información médica bajo su ámbito. Además, la autenticación mediante credenciales junto con la estructura modular del backend hace posible incorporar fácilmente protocolos de encriptación durante futuras migraciones a la nube.
- El modelo de inteligencia artificial CheXNet fue integrado y adaptado con éxito, lo que permite la interpretación de imágenes de rayos X torácicos, así como la generación automática de identificación y categorización de patologías. Fue probado funcionalmente con imágenes reales, y los resultados diagnósticos se almacenaron y presentaron de manera accesible a través del sistema web logrando su propósito previsto.
- Se logró una interfaz web simple y responsiva que soporta la carga y análisis de imágenes médicas dentro de un tiempo de respuesta aceptable. Las pruebas funcionales realizadas confirmaron que, además de procesar datos de manera eficiente, el sistema es verdaderamente adaptable incluso a entornos con recursos computacionales limitados, lo que es crítico para su implementación en áreas remotas.
- Se implementó encriptación simétrica utilizando el algoritmo de Fernet de Python, para garantizar la transferencia de los datos sensibles e imágenes médicas dentro del sistema, esta técnica garantiza que los archivos cifrados solo pueden ser descifrados mediante una clave segura generada y almacenada en el sistema.

- Se demostró que el sistema puede ser accedido de forma remota gracias al programa de Ngrok que facilita exponer nuestro sistema local con una dirección pública, de tal forma se pudo realizar pruebas de accesibilidad a través de un dispositivo móvil con datos móviles validando su funcionamiento y fluidez.

RECOMENDACIONES

- Se recomienda migrar el sistema a una plataforma en la nube (como AWS, Azure o GCP) en una etapa posterior, mientras se añaden más medidas de protección de la seguridad de los datos, cifrado y copias de seguridad automáticas.
- Para el uso en entornos clínicos reales, se recomienda complementar el sistema con un módulo de auditoría y trazabilidad que documente el acceso de los usuarios, las ediciones y los diagnósticos realizados por cada usuario, cumpliendo con las regulaciones Internacionales.
- Asegurarse que los resultados automáticos procesados por el sistema alcancen una sensibilidad, especificidad y precisión diagnóstica aceptables en estándares automáticos, es necesario validar el modelo con imágenes clínicas etiquetadas por especialistas en futuras versiones del sistema.
- Para mejorar la experiencia del usuario, se recomienda realizar pruebas de usabilidad con médicos esto ayudará a adaptar el diseño a necesidades específicas.

BIBLIOGRAFÍA

- [1] L. Vera, C. Cuadros, y M. Chavarría, «Impacto de las políticas públicas de salud en zonas rurales ecuatorianas», vol. 10, ago. 2022, Accedido: 7 de diciembre de 2024. [En línea]. Disponible en: http://www.scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2308-01322022000200025
- [2] Organización Mundial de la Salud, «Calidad de la atención». Accedido: 25 de junio de 2025. [En línea]. Disponible en: https://www.who.int/es/health-topics/quality-of-care#tab=tab_1
- [3] A. Llinás-Delgado, A. Alcocer-Olaciregui, y R. Vargas-Moranth, «Telemedicina: revolucionando la prestación de la atención médica en la era digital», *Transdigital*, vol. 4, n.º 8, pp. 1-13, ago. 2023, doi: 10.56162/TRANSDIGITAL259.
- [4] OpenKM, «HIPPA ». Accedido: 25 de junio de 2025. [En línea]. Disponible en: <https://www.openkm.com/es/blog/cumplimiento-hipaa-en-la-gestion-documental.html>
- [5] Intersoft Consulting, «General Data Protection Regulation (GDPR) – Legal Text». Accedido: 25 de junio de 2025. [En línea]. Disponible en: <https://gdpr-info.eu/>
- [6] Comisión Nacional de Productividad, «Regulación de Plataformas Digitales Telemedicina», p. 8, 2021, Accedido: 7 de diciembre de 2024. [En línea]. Disponible en: <https://www.camara.cl/verDoc.aspx?prmTIPO=DOCUMENTOCOMUNICACIONCUENTA&prmID=75386#:~:text=Seg%C3%BAAn%20la%20OMS%2C%20la%20telemedicina,el%20diagn%C3%B3stico%2C%20tratamiento%20y%20prevenci%C3%B3n>
- [7] G. F. Luger y W. A. Stubblefield, «Análisis de Imágenes Médicas mediante IA: Innovación y Precisión», 2023, Accedido: 26 de junio de 2024. [En línea]. Disponible en: <https://revistamedica.com/analisis-diagnostico-imagenes-medicas-mediante-inteligencia-artificial/#:~:text=la%20inteligencia%20artificial.-,La%20IA%20en%20el%20an%C3%A1lisis%20de%20im%C3%A1genes%20m%C3%A9dicas,diagn%C3%B3stico%20m%C3%A1s%20r%C3%A1pido%20y%20preciso.>

- [8] A. Kopec y A. Salazar, *Aplicaciones de Telecomunicaciones en Salud en la Subregión Andina: Telemedicina*. 2002. Accedido: 14 de diciembre de 2024. [En línea]. Disponible en: [http://git.unicauca.edu.co/ehas/docs/Salvador2005/LibroORAS/Resume n-Telemedicina-Aplicaciones%20de%20telecomunicaciones%20en%20salud%20en%20la%20subregion%20andina.pdf](http://git.unicauca.edu.co/ehas/docs/Salvador2005/LibroORAS/Resume%20n-Telemedicina-Aplicaciones%20de%20telecomunicaciones%20en%20salud%20en%20la%20subregion%20andina.pdf)
- [9] C. Ibáñez, Á. Zuluaga, y A. Trujillo, «TELEMEDICINA: Introducción, aplicación y principios de desarrollo», *CES Medicina*, pp. 77-93, 2007, Accedido: 14 de diciembre de 2024. [En línea]. Disponible en: <https://www.redalyc.org/pdf/2611/261120984009.pdf>
- [10] Universidad de Sevilla, «Imágenes Médicas», 2020. Accedido: 14 de diciembre de 2024. [En línea]. Disponible en: [https://biblus.us.es/bibing/proyectos/abreproy/11854/fichero/Volumen+1 %252FCapitulo+2.pdf](https://biblus.us.es/bibing/proyectos/abreproy/11854/fichero/Volumen+1%252FCapitulo+2.pdf)
- [11] Clínica Universidad de Navarra, «Qué es radiografía. Diccionario médico. Clínica U. Navarra». Accedido: 17 de junio de 2025. [En línea]. Disponible en: <https://www.cun.es/diccionario-medico/terminos/radiografia>
- [12] Clínica Universidad de Navarra, «Tomografía: definición médica. Diccionario CUN». Accedido: 17 de junio de 2025. [En línea]. Disponible en: <https://www.cun.es/diccionario-medico/terminos/tomografia>
- [13] Clínica Universidad Navarra, «Resonancia magnética. Pruebas diagnósticas. Clínica Universidad de Navarra». Accedido: 17 de junio de 2025. [En línea]. Disponible en: <https://www.cun.es/enfermedades-tratamientos/pruebas-diagnosticas/resonancia-magnetica>
- [14] Clínica Universidad de Navarra, «Ecografías: Qué son, tipos e información. Diagnóstico. Clínica Universidad de Navarra». Accedido: 17 de junio de 2025. [En línea]. Disponible en: <https://www.cun.es/enfermedades-tratamientos/pruebas-diagnosticas/ecografia>
- [15] C. J. Landín, M. A. Mendoza Pérez, J. R. García Ibarra, y J. M. Sánchez Soto, «Procesamiento Digital de Señales e Imágenes Médicas con Propósito de Diagnóstico Para Reducir Riesgos en Salud», *Ciencia*

Latina Revista Científica Multidisciplinar, vol. 8, n.º 5, pp. 2925-2935, oct. 2024, doi: 10.37811/CL_RCM.V8I5.13751.

- [16] A. Esteva *et al.*, «Dermatologist-level classification of skin cancer with deep neural networks», *Nature*, vol. 542, n.º 7639, pp. 115-118, feb. 2017, doi: 10.1038/NATURE21056,.
- [17] Postdicom, «HL7 frente a DICOM: comprender las principales diferencias y su impacto en las imágenes médicas | PostDICOM». Accedido: 14 de diciembre de 2024. [En línea]. Disponible en: <https://www.postdicom.com/es/blog/hl7-vs-dicom>
- [18] E. Alpaydin, *Introduction to Machine Learning, fourth edition - Ethem Alpaydin - Google Libros*. 2020. Accedido: 14 de diciembre de 2024. [En línea]. Disponible en: <https://books.google.com.ec/books?id=tZnSDwAAQBAJ&printsec=copyright&hl=es#v=onepage&q&f=false>
- [19] Gamco, «Qué es Deep Learning Concepto y definición. Glosario». Accedido: 19 de junio de 2025. [En línea]. Disponible en: <https://gamco.es/glosario/deep-learning/>
- [20] Y. Bengio, «Learning Deep Architectures for AI», *Foundations and Trends® in Machine Learning*, vol. 2, n.º 1, pp. 1-127, nov. 2009, doi: 10.1561/22000000006.
- [21] Gamco, «Qué es Red Neuronal Artificial - ANN Concepto y definición. Glosario». Accedido: 19 de junio de 2025. [En línea]. Disponible en: <https://gamco.es/glosario/red-neuronal-artificial-ann/>
- [22] P. Rajpurkar *et al.*, «CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning», nov. 2017, Accedido: 23 de junio de 2025. [En línea]. Disponible en: <https://arxiv.org/pdf/1711.05225>
- [23] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyon Lu, Mohammadhab Bahheri, y Ronald Summers, «Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases», *Radiology*, vol. 294, n.º 2, pp. 421-431, 2017, doi: 10.1148/RADIOL.2019191293.

ANEXOS

Anexo 1: Docker-compose.yml

En el anexo 1 se muestra el código utilizado para levantar los servicios del sistema de backend y de la base de datos de PostgreSQL en los contenedores Docker.

```
version: '3.8'
services:
  backend:
    build: .
    ports:
      - "5000:5000"
    depends_on:
      - postgres
    networks:
      - tesisnet

  postgres:
    image: postgres:13
    environment:
      POSTGRES_DB: tesis_db
      POSTGRES_USER: tesis_user
      POSTGRES_PASSWORD: tesis123
    ports:
      - "5432:5432"
    networks:
      - tesisnet

networks:
  tesisnet:
```

Anexo 2: model_ia.py

En el anexo 2 se puede visualizar el código utilizado para cargar el modelo de CheXNet basado en DenseNet121 donde se realiza el análisis automático de las imágenes médicas.

```
import torch
import torchvision.transforms as transforms
from PIL import Image
from torchvision import models
```

```

def chexnet_predict(image_path):
    model = models.densenet121(pretrained=True)
    model.eval()

    transform = transforms.Compose([
        transforms.Resize(224),
        transforms.CenterCrop(224),
        transforms.ToTensor()
    ])

    image = Image.open(image_path).convert('RGB')
    image = transform(image).unsqueeze(0)

    with torch.no_grad():
        outputs = model(image)

    prediction = torch.sigmoid(outputs).numpy()[0]
    predicted_label = "Anomalía detectada" if max(prediction) > 0.5
else "Sin anomalías"
    return predicted_label

```

Anexo 3: backend – Flask

En el anexo 3 se puede observar el código utilizado para que funcione la aplicación desde la web.

```

from flask import Flask, render_template, request, redirect, session, url_for, jsonify
from werkzeug.utils import secure_filename
from model_ia import chexnet_predict
import psycopg2
import os
from datetime import datetime

app = Flask(__name__)
app.secret_key = 'clave_secreta_segura'

def conectar_db():
    return psycopg2.connect(
        host="postgres",
        database="tesis_db",
        user="tesis_user",
        password="tesis123"
    )

@app.route('/login', methods=['POST'])
def login_post():
    data = request.get_json()

```



```

    usuario = data.get('usuario')
    password = data.get('password')
    conn = conectar_db()
    cur = conn.cursor()
    cur.execute("SELECT password, rol FROM usuarios WHERE usuario =
%s", (usuario,))
    resultado = cur.fetchone()
    cur.close()
    conn.close()
    if resultado and resultado[0] == password:
        return jsonify({"success": True, "rol": resultado[1]})
    return jsonify({"success": False})

@app.route('/subir_imagen/<cedula>', methods=['POST'])
def subir_imagen(cedula):
    if 'imagen' not in request.files:
        return "No se envió ninguna imagen"
    imagen = request.files['imagen']
    filename = secure_filename(imagen.filename)
    ruta = os.path.join('uploads', filename)
    imagen.save(ruta)
    resultado = cheynet_predict(ruta)
    # Guardar en la base de datos
    conn = conectar_db()
    cur = conn.cursor()
    cur.execute("INSERT INTO diagnostico (cedula, resultado, fecha)
VALUES (%s, %s, %s)",
                (cedula, str(resultado), datetime.now()))
    conn.commit()
    cur.close()
    conn.close()
    return "Diagnóstico realizado correctamente"

```

Anexo 4: Estructura base de datos

En el anexo 4 se puede visualizar la estructura de base de datos donde se procedió a crear 3 tablas.

```

CREATE TABLE usuarios (
    id SERIAL PRIMARY KEY,
    nombres VARCHAR(50),
    apellidos VARCHAR(50),
    usuario VARCHAR(50) UNIQUE,
    password VARCHAR(100),
    rol VARCHAR(20)
);

CREATE TABLE historia_clinica (

```

```

        id SERIAL PRIMARY KEY,
        cedula VARCHAR(10) UNIQUE,
        nombre VARCHAR(100),
        edad INTEGER,
        antecedentes TEXT
    );

CREATE TABLE diagnostico (
    id SERIAL PRIMARY KEY,
    cedula VARCHAR(10),
    resultado TEXT,
    fecha TIMESTAMP
);

```

Anexo 5: Docker file

En el anexo 5 podemos visualizar el código utilizado para construir el contenedor Docker del backend (app.py), donde incluye todas las dependencias.

```

FROM python:3.10-slim
WORKDIR /app
COPY . .
RUN pip install --no-cache-dir -r requirements.txt
EXPOSE 5000
CMD ["python", "app.py"]

```

Anexo 6: requirements.txt

En el anexo 6 se puede visualizar la lista de bibliotecas necesarias para el backend Python funcione correctamente, esto permite instalar automáticamente las dependencias.

```

Flask
psycopg2-binary
torch
torchvision
Pillow

```

Anexo 7: app.py

En el anexo 7 se puede observar el código utilizado para el desarrollo del sistema este es el principal porque maneja las rutas web de conexión a base de datos donde

permite el registro, la carga de imágenes y la ejecución del diagnóstico. Coordina toda la lógica del sistema.

```
from flask import Flask, render_template, request, redirect, session, url_for, jsonify
from werkzeug.utils import secure_filename
from model_ia import cheXnet_predict
import psycopg2
import os
from datetime import datetime

app = Flask(__name__, template_folder='templates', static_folder='static')
app.secret_key = 'clave_secreta_segura'

def conectar_db():
    return psycopg2.connect(
        host="postgres",
        database="tesis_db",
        user="tesis_user",
        password="tesis123"
    )

@app.route('/')
def login():
    return render_template('login.html')

@app.route('/login', methods=['POST'])
def login_post():
    data = request.get_json()
    usuario = data.get('usuario')
    password = data.get('password')
    conn = conectar_db()
    cur = conn.cursor()
    cur.execute("SELECT password, rol FROM usuarios WHERE usuario = %s", (usuario,))
    resultado = cur.fetchone()
    cur.close()
    conn.close()
    if resultado and resultado[0] == password:
        return jsonify({"success": True, "rol": resultado[1]})
    return jsonify({"success": False})

@app.route('/registro', methods=['POST'])
def registrar_usuario():
    nombres = request.form['nombres']
    apellidos = request.form['apellidos']
    usuario = request.form['usuario']
    password = request.form['password']
```

```

rol = request.form['rol']
conn = conectar_db()
cur = conn.cursor()
cur.execute("INSERT INTO usuarios (nombres, apellidos, usuario,
password, rol) VALUES (%s, %s, %s, %s, %s)",
            (nombres, apellidos, usuario, password, rol))
conn.commit()
cur.close()
conn.close()
return redirect('/')

@app.route('/guardar_historia', methods=['POST'])
def guardar_historia():
    cedula = request.form['cedula']
    nombre = request.form['nombre']
    edad = request.form['edad']
    antecedentes = request.form['antecedentes']
    conn = conectar_db()
    cur = conn.cursor()
    cur.execute("INSERT INTO historia_clinica (cedula, nombre, edad,
antecedentes) VALUES (%s, %s, %s, %s)",
                (cedula, nombre, edad, antecedentes))
    conn.commit()
    cur.close()
    conn.close()
    return "Historia clínica guardada correctamente"

@app.route('/subir_imagen', methods=['POST'])
def subir_imagen():
    cedula = request.form['cedula']
    imagen = request.files['imagen']
    filename = secure_filename(imagen.filename)
    ruta = os.path.join('uploads', filename)
    imagen.save(ruta)
    resultado = cheXnet_predict(ruta)
    conn = conectar_db()
    cur = conn.cursor()
    cur.execute("INSERT INTO diagnostico (cedula, resultado, fecha)
VALUES (%s, %s, %s)",
                (cedula, resultado, datetime.now()))
    conn.commit()
    cur.close()
    conn.close()
    return f"Diagnóstico: {resultado}"

@app.route('/ver_diagnostico/<cedula>')
def ver_diagnostico(cedula):
    conn = conectar_db()
    cur = conn.cursor()

```

```

        cur.execute("SELECT resultado, fecha FROM diagnostico WHERE ce-
dula = %s ORDER BY fecha DESC LIMIT 1", (cedula,))
        data = cur.fetchone()
        cur.close()
        conn.close()
        if data:
            return render_template('ver_diagnostico.html', cedula=ce-
dula, resultado=data[0], fecha=data[1])
        else:
            return "No hay diagnóstico disponible"

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)

6. style.css
body {
    font-family: Arial, sans-serif;
    background-color: #f4f7fc;
    margin: 0;
    padding: 0;
    text-align: center;
}
form {
    background-color: white;
    margin: 50px auto;
    padding: 30px;
    width: 400px;
    border-radius: 12px;
    box-shadow: 0 0 10px rgba(0,0,0,0.1);
}
input, select, textarea {
    width: 90%;
    margin: 10px 0;
    padding: 10px;
}

```

Anexo 8: registro.html

En el anexo 8, se visualiza el código utilizado para el formulario web donde permite registrar nuevos usuarios con el rol médico y facilita la creación de cuentas seguras para médicos generales y especialistas.

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Registro</title>
</head>

```

```

<body>
  <h2>Registro de Usuario</h2>
  <form action="/registro" method="POST">
    Nombre: <input type="text" name="nombres"><br>
    Apellido: <input type="text" name="apellidos"><br>
    Usuario: <input type="text" name="usuario"><br>
    Contraseña: <input type="password" name="password"><br>
    Rol:
    <select name="rol">
      <option value="medico_general">Médico General</option>
      <option value="medico_especialista">Médico Especia-
lista</option>
    </select><br><br>
    <button type="submit">Registrar</button>
  </form>
</body>
</html>

```

Anexo 9: medico.html

En el anexo 9, se puede observar el código utilizado para la interfaz del médico general donde muestra los distintos accesos para registrar la historia clínica del paciente sobre las imágenes visualizar su análisis haz una de las interfases principales del sistema.

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Perfil Médico</title>
</head>
<body>
  <h2>Bienvenido, Doctor</h2>
  <form action="/guardar_historia" method="POST">
    Cédula del paciente: <input type="text" name="cedula"><br>
    Nombre: <input type="text" name="nombre"><br>
    Edad: <input type="number" name="edad"><br>
    Antecedentes: <textarea name="antecedentes"></textarea><br>
    <button type="submit">Guardar Historia Clínica</button>
  </form>
  <br>
  <form action="/subir_imagen" method="POST" enctype="multi-
part/form-data">
    Cargar imagen médica del paciente:
    <input type="text" name="cedula" placeholder="Cédula del pa-
ciente"><br>
    <input type="file" name="imagen"><br>

```

```

        <button type="submit">Subir Imagen y Analizar</button>
    </form>
</body>
</html>

```

Anexo 10: diagnostico.html

En el anexo 10, se puede observar el código utilizado para que muestre el resultado del análisis realizado por el modelo de inteligencia artificial incluyendo el diagnóstico la cédula, la fecha de la carga de la imagen.

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Diagnóstico</title>
</head>
<body>
    <h2>Diagnóstico del Paciente</h2>
    <p>Cédula: {{ cedula }}</p>
    <p>Resultado del análisis: {{ resultado }}</p>
    <p>Fecha: {{ fecha }}</p>
</body>
</html>

```