

**FACULTAD DE INGENIERÍA**

Escuela Académico Profesional de Ingeniería Electrónica

Tesis

**Diseño de arquitectura de red neuronal convolucional  
para el diagnóstico del COVID19 mediante imágenes  
de rayos X**

Jose Rafael Galvez Siuce

Para optar el Título Profesional de  
Ingeniero Electrónico

Huancayo, 2023

Repositorio Institucional Continental  
Tesis digital



Esta obra está bajo una Licencia "Creative Commons Atribución 4.0 Internacional" .

## INFORME DE CONFORMIDAD DE ORIGINALIDAD DE TESIS

**A** : Felipe Néstor Gutarra Meza  
Decano de la Facultad de Ingeniería

**DE** : Carlos Quispe Anccasi  
Asesor de tesis

**ASUNTO** : Remito resultado de evaluación de originalidad de tesis

**FECHA** : 14 de noviembre de 2023

---

Con sumo agrado me dirijo a vuestro despacho para saludarlo y en vista de haber sido designado asesor de la tesis titulada: "Diseño de arquitectura de red neuronal convolucional para el diagnóstico del Covid19 mediante imágenes de rayos x ", perteneciente al/la/los/las estudiante(s) JOSE RAFAEL GALVEZ SIUCE, de la E.A.P. de Ingeniería Electrónica; se procedió con la carga del documento a la plataforma "Turnitin" y se realizó la verificación completa de las coincidencias resaltadas por el software dando por resultado 6 % de similitud (informe adjunto) sin encontrarse hallazgos relacionados a plagio. Se utilizaron los siguientes filtros:

- Filtro de exclusión de bibliografía SI  NO
- Filtro de exclusión de grupos de palabras menores (Nº de palabras excluidas: 20) SI  NO
- Exclusión de fuente por trabajo anterior del mismo estudiante SI  NO

En consecuencia, se determina que la tesis constituye un documento original al presentar similitud de otros autores (citas) por debajo del porcentaje establecido por la Universidad.

Recae toda responsabilidad del contenido de la tesis sobre el autor y asesor, en concordancia a los principios de legalidad, presunción de veracidad y simplicidad, expresados en el Reglamento del Registro Nacional de Trabajos de Investigación para optar grados académicos y títulos profesionales – RENATI y en la Directiva 003-2016-R/UC.

Esperando la atención a la presente, me despido sin otro particular y sea propicia la ocasión para renovar las muestras de mi especial consideración.

Atentamente,

## **DECLARACIÓN JURADA DE AUTENTICIDAD**

Yo, JOSE RAFAEL GALVEZ SIUCE, identificado(a) con Documento Nacional de Identidad No. 73175480, de la E.A.P. de INGENIERIA ELECTRONICA de la Facultad de INGENIERIA la Universidad Continental, declaro bajo juramento lo siguiente:

1. La tesis titulada: "Diseño de arquitectura de red neuronal convolucional para el diagnóstico de Covid19 mediante imágenes de rayos x", es de mi autoría, la misma que presento para optar el Título Profesional de INGENIERO ELECTRONICO.
2. La tesis no ha sido plagiada ni total ni parcialmente, para la cual se han respetado las normas internacionales de citas y referencias para las fuentes consultadas, por lo que no atenta contra derechos de terceros.
3. La tesis es original e inédita, y no ha sido realizado, desarrollado o publicado, parcial ni totalmente, por terceras personas naturales o jurídicas. No incurre en autoplagio; es decir, no fue publicado ni presentado de manera previa para conseguir algún grado académico o título profesional.
4. Los datos presentados en los resultados son reales, pues no son falsos, duplicados, ni copiados, por consiguiente, constituyen un aporte significativo para la realidad estudiada.

De identificarse fraude, falsificación de datos, plagio, información sin cita de autores, uso ilegal de información ajena, asumo las consecuencias y sanciones que de mi acción se deriven, sometiéndome a las acciones legales pertinentes.

15 de noviembre de 2023.

# "Diseño de arquitectura de red neuronal convolucional para el diagnóstico del Covid19 mediante imágenes de rayos x"

## INFORME DE ORIGINALIDAD



## FUENTES PRIMARIAS

1	<a href="http://hdl.handle.net">hdl.handle.net</a> Fuente de Internet	2%
2	Submitted to Pontificia Universidad Catolica del Peru Trabajo del estudiante	1%
3	<a href="http://repositorio.unp.edu.pe">repositorio.unp.edu.pe</a> Fuente de Internet	1%
4	Submitted to Universidad Continental Trabajo del estudiante	<1%
5	<a href="http://www.topbooks.com.tw">www.topbooks.com.tw</a> Fuente de Internet	<1%
6	Submitted to Universidad Alfonso X el Sabio Trabajo del estudiante	<1%
7	<a href="http://csscysm.blogspot.com">csscysm.blogspot.com</a> Fuente de Internet	<1%
8	<a href="http://repositorio.urp.edu.pe">repositorio.urp.edu.pe</a> Fuente de Internet	<1%

9	Domingo Mery, Christian Pieringer. "Computer Vision for X-Ray Testing", Springer Science and Business Media LLC, 2021 Publicación	<1 %
10	es.slideshare.net Fuente de Internet	<1 %
11	repository.icesi.edu.co Fuente de Internet	<1 %
12	"Modelación de consideración y elección de rutas en transporte público", Pontificia Universidad Católica de Chile, 2021 Publicación	<1 %
13	Submitted to Escuela Superior Politécnica del Litoral Trabajo del estudiante	<1 %
14	Submitted to UNIBA Trabajo del estudiante	<1 %
15	Submitted to Universidad San Francisco de Quito Trabajo del estudiante	<1 %
16	repositorio.continental.edu.pe Fuente de Internet	<1 %

Excluir citas

Activo

Excluir coincidencias < 20 words

Excluir bibliografía

Activo

## **DEDICATORIA**

A mis padres, que me apoyaron para lograr a en mi crecimiento como profesional y a mi abuelo Crisanto, aunque ya no esté aquí, sé que vela por mí.

## **AGRADECIMIENTOS**

A toda mi familia, a la que estaré eternamente agradecido por el apoyo, consejos y motivación, pues sin su apoyo no se hubiera alcanzado una meta más.

Al Dr. Carlos Quispe Ancasi, por los consejos, ya que con su ayuda he logrado realizar la presente tesis.



## ÍNDICE DE CONTENIDO

DEDICATORIA .....	6
AGRADECIMIENTOS .....	7
ÍNDICE DE CONTENIDO.....	8
ÍNDICE DE TABLAS .....	11
ÍNDICE DE FIGURAS .....	12
RESUMEN .....	13
ABSTRACT .....	14
PLANTEAMIENTO DEL ESTUDIO .....	15
1.1. Planteamiento y formulación del problema .....	15
1.1.1. Problema general .....	17
1.1.2. Problemas específicos .....	17
1.2. Objetivos .....	18
1.2.1. Objetivo general .....	18
1.2.2. Objetivos específicos .....	18
1.3. Justificación .....	18
1.3.1. Técnica.....	18
1.3.2. Teórica.....	18
1.3.3. Económica .....	19
1.4. Importancia .....	19
1.5. Limitaciones .....	20
1.6. Resultados esperados.....	20
1.6.1. General .....	20
1.6.2. Específicos .....	20
1.7. Variables.....	20
1.7.1. Variable independiente .....	20
1.7.2. Variable dependiente .....	21
1.7.3. Operacionalización de variables .....	21
CAPÍTULO II .....	22
MARCO TEORICO.....	22
2.1. Antecedentes del Problema .....	22
2.1.1. Nacionales .....	22
2.1.2. Internacionales .....	24
2.2. Bases teóricas.....	29

2.2.1.	Arquitectura de red neuronal convolucional .....	29
2.2.2.	Capa convolucional.....	30
2.2.3.	Capa de agrupación .....	31
2.2.4.	Capas totalmente conectadas .....	31
2.2.5.	Métricas.....	32
2.2.6.	Matriz de confusión.....	32
2.2.6.1.	Exactitud .....	33
2.2.6.2.	Precisión .....	33
2.2.6.3.	Sensibilidad.....	33
2.2.6.4.	Especificidad .....	34
2.2.6.5.	F-Score .....	34
2.2.7.	Diagnóstico .....	34
2.2.8.	Prueba de diagnostico .....	34
2.2.9.	Diagnóstico del COVID-19.....	34
2.2.10.	Definición de Términos Básicos .....	35
2.2.10.1.	Red neuronal convolucional .....	35
2.2.10.2.	COVID-19 .....	36
2.2.10.3.	Clasificación .....	36
2.2.10.4.	Filtro kernel .....	36
2.2.10.5.	Rayos x del tórax.....	36
2.2.10.6.	Procesamiento de imágenes.....	36
CAPÍTULO III .....		37
METODOLOGIA .....		37
3.1.	Método de Investigación .....	37
3.2.	Tipo de Investigación.....	37
3.3.	Alcance de la Investigación.....	37
3.4.	Diseño de investigación.....	38
3.5.	Población.....	38
3.6.	Muestra .....	39
3.7.	Técnicas e Instrumentos de Recolección de Datos .....	40
3.8.	Diseño de Ingeniería.....	40
3.8.1.	Muestra .....	41
3.8.2.	Exploración .....	41
3.8.3.	Modificación .....	41
3.8.4.	Modelado .....	41

3.8.5.	Evaluación .....	41
3.9.	Materiales y Especificaciones Técnicas.....	42
3.9.1.	Python .....	42
3.9.2.	Hardware .....	42
3.10.	Construcción del algoritmo .....	43
3.10.1.	Muestra de base de datos .....	43
3.10.2.	Exploración de base de datos .....	45
3.10.3.	Modificación del base de datos .....	46
3.10.4.	Modelo de arquitectura convolucional.....	51
3.10.5.	Evaluación del clasificador .....	53
CAPÍTULO IV	.....	54
RESULTADOS Y DISCUSIÓN	.....	54
4.2.	Resultados e interpretación .....	54
4.2.1.	Resultados de preprocesamiento de base de datos .....	54
4.2.2.	Resultados de configuración de capas convolucional y capas totalmente conectadas.....	58
4.2.3.	Resultado de validación de arquitectura .....	62
4.3.	Discusión de Resultados .....	65
CONCLUSIONES	.....	68
RECOMENDACIONES	.....	70
REFERENCIAS BIBLIOGRAFICAS	.....	71
ANEXOS	.....	75

## ÍNDICE DE TABLAS

Tabla 1.	Operacionalización de Variables .....	21
Tabla 2.	Matriz de confusión.....	33
Tabla 3.	Tipo y Nivel de investigación.....	38
Tabla 4.	Base de datos de instituciones .....	38
Tabla 5.	Base de datos .....	39
Tabla 6.	Formato de recolección de datos .....	40
Tabla 7.	Comparación del promedio de intensidad media y desviación estándar entre la base de datos de imágenes originales y las imágenes preprocesadas .....	55
Tabla 8.	Pérdida de intensidad media y desviación estándar entre imágenes originales y ecualizadas.....	57
Tabla 9.	Resultado de entrenamiento del modelo .....	59
Tabla 10.	Métricas del modelo .....	64

## ÍNDICE DE FIGURAS

Figura 1.	Infecciones y muertes mesunuales globales por COVID-19, enero a octubre 2020 .....	15
Figura 2.	Hospitalizaciones por COVID-19 en hospitales del MINSA y porcentaje de fallecidos, Perú marzo 2020 a julio 2021 .....	16
Figura 3.	Arquitectura de la red neuronal convolucional .....	29
Figura 4.	Convolución por superposición .....	31
Figura 5.	Filtro de agrupación 2 x 2 .....	31
Figura 6.	Capas totalmente conectadas .....	32
Figura 7.	Radiografía del torác: (a) paciente normal (b) paciente con COVID.....	35
Figura 8.	Proceso de etapas de SEMMA.....	42
Figura 9.	Diagrama de segmentación de datos .....	44
Figura 10.	Carpetas creadas .....	45
Figura 11.	Dimensiones de imágenes de rayos X clase “COVID” y “Normal” .....	46
Figura 12.	Importación de librerías para el procesamiento .....	46
Figura 13.	Rutas de las carpetas y clases .....	47
Figura 14.	Fórmula máximos y mínimos de tonos de gris .....	47
Figura 15.	Algoritmo de equalization .....	48
Figura 16.	Código de creación de carpeta por clases .....	49
Figura 17.	Código de pre procesamiento .....	50
Figura 18.	Librería de aprendizaje profundo.....	51
Figura 19.	Arquitectura de red neuronal convolucional .....	52
Figura 20.	Algoritmo de arquitectura de red convolucional .....	52
Figura 21.	Algoritmo para iniciar el entrenamiento .....	53
Figura 22.	Algoritmo de matriz de confusión .....	53
Figura 23.	Muestra de imágenes de la base de datos. Las imágenes originales (a) “normal”, (b) “COVID” . Imagen filtrada pasa bajo (c) “normal”, (d) “COVID”. Imagen filtrada pasa alto (e) “normal”, (f) “COVID”. Imagen agudizado (g) “normal”,(h) “COVID”. Imagen ecualizada (i) “normal”, (j) “COVID”.....	54
Figura 24.	Evolución de pérdidas durante el proceso de entrenamiento .....	61
Figura 25.	Evolución de exactitud durante el proceso de entrenamiento .....	61
Figura 26.	Matriz de confusión aplicada el conjunto pruebas .....	62

## RESUMEN

El COVID-19 fue una de las pandemias más mortales y la enfermedad que más rápido se propagó, debido a su alta transmisibilidad. La posibilidad de un diagnóstico erróneo causó varias consecuencias negativas al paciente infectado por COVID-19. Para abordar este problema, la tesis tiene como objetivo diseñar una arquitectura de red neuronal convolucional para el diagnóstico de COVID-19 mediante imágenes de rayos x. El método de desarrollo para la arquitectura es SEMMA; esto ayuda controlar mejor las etapas que se desarrollan. Asimismo, se realizó las pruebas con 753 imágenes rayos x para evaluar el rendimiento de la arquitectura entrenado usando varios parámetros de evaluación. Los resultados muestran que la arquitectura logra la mejor precisión de 90 %, exactitud del 91 % y sensibilidad del 93 %. En general, el modelo realiza un buen desempeño para detectar COVID-19.

**Palabras clave:** COVID-19, aprendizaje automático, red neuronal convolución, imágenes médicas, rayos x del tórax

## ABSTRACT

COVID-19 was one of the deadliest pandemics and the fastest spreading disease due to its high transmissibility. The possibility of a misdiagnosis causes several negative consequences to the patient infected by COVID-19. To address this problem, the thesis aims to design a convolutional neural network architecture for the diagnosis of COVID-19 using x-ray images. The development method for the architecture is SEMMA, this helps to better control the stages that are developed. Tests are performed with 753 x-ray images to evaluate the performance of the trained architecture using various evaluation parameters. The results show that the architecture achieves the best precision of 90 %, accuracy of 91 % and sensitivity of 93 %. In general, the model performs well in detecting COVID-19.

**Keywords:** COVID-19, machine learning, red neural convolution, medical imaging, chest x-ray

# CAPÍTULO I

## PLANTEAMIENTO DEL ESTUDIO

### 1.1. Planteamiento y formulación del problema

En el 2020, la enfermedad más infecciosa es el COVID-19, que causó el colapso en la salud pública en todos los países. Los síntomas iniciales del paciente infectado por el COVID-19 son fiebre, dolor de cabeza, diarrea, dificultad de respirar, dolor de garganta y otras enfermedades respiratorias[1].

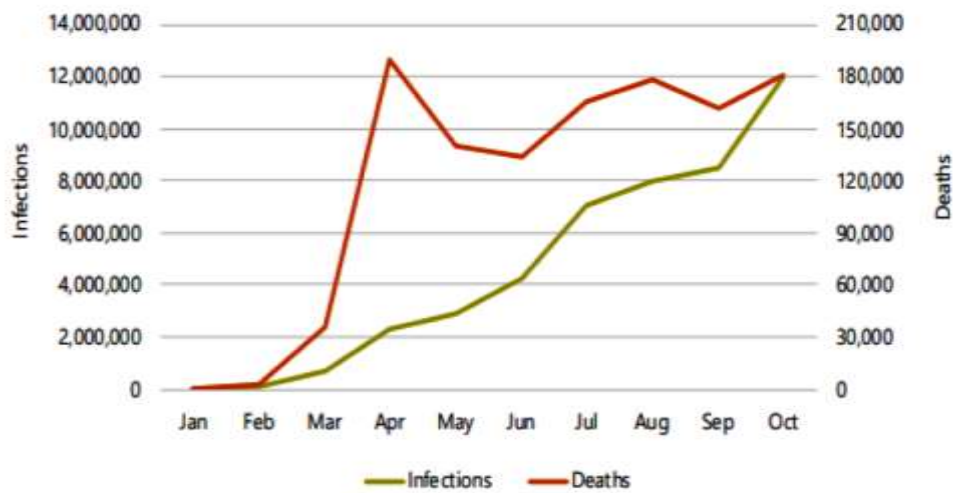


Figura 1. Infecciones y muertes mesuales globales por COVID-19, enero a octubre 2020, por How Data Can Manage Global Health Pandemics-Rupa Mahanti

En el Perú, la primera ola de la pandemia COVID-19 fue asoladora con una tasa de mortalidad alta de 660.5 por 100 mil habitantes [2], registrando 89,884 fallecidos en el territorio peruano de acuerdo con la Resolución Ministerial N.º 95-2021-PCM[3]. Ante la situación, el primer método de diagnóstico era la prueba de reacción en cadena de la polimerasa con transcriptasa (RT-PCR) [4], pero tiene diversos inconvenientes, debido a que la prueba ofrece el diagnóstico en un tiempo de 6 a 8 horas, además la sensibilidad oscila 85-90 %, por lo que los falsos negativos son un problema clínico[5]. Asimismo, la prueba RT-PCR no es el único método de diagnóstico, la otra alternativa es el uso de la radiografía del tórax, mediante los



equipos de rayos X que es un instrumento principal para los radiólogos en el diagnóstico del COVID-19. Durante la primera ola, el Hospital Nacional Arzobispo Loayza en el área de Diagnóstico por Imagen ha realizado más de 3000 exámenes de tomografías, rayos X y ecografías a pacientes internados por el COVID-19 para su diagnóstico[6]. Sin embargo, los resultados dependen a la buena interpretación a las imágenes, debido que las imágenes con COVID-19 presentan opacidad en vidrio, deslustrado similar a la opacidad de las neumonías bacterianas y fúngicas que no están relacionada con el COVID-19. Esto resulta problemático para los radiólogos y aumenta el error de diagnóstico en diferenciar entre estas patologías[7]. Por otra parte, el incremento exponencial de pacientes infectados ver figura 2 está se asocia con una mayor carga laboral para los radiólogos de tórax ocasionado el desgaste profesional. El desgaste profesional es por el aumento de exámenes de imágenes realizadas, incrementó de interpretación de las imágenes por pacientes y jornadas de trabajos prolongados desencadenado los errores de diagnósticos[8].

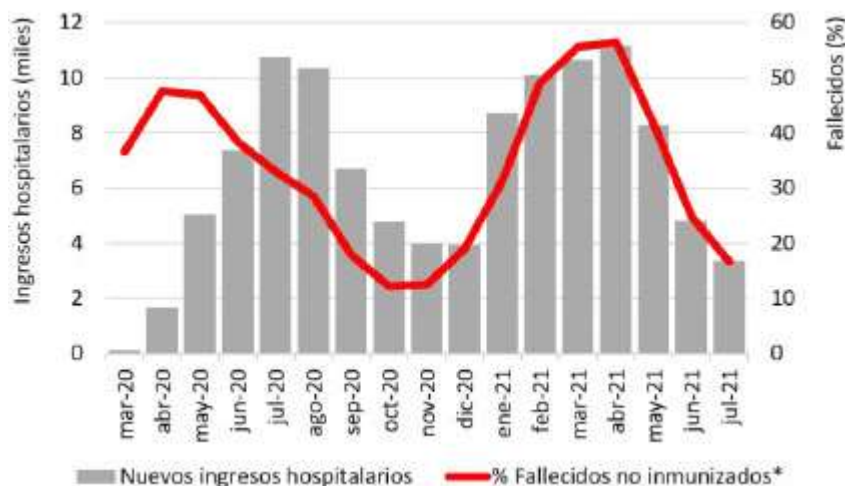


Figura 2. Hospitalizaciones por COVID-19 en hospitales del MINSA y porcentaje de fallecidos, Perú marzo 2020 a julio 2021. Por Percy Mayta.

Los daños generados por los errores de diagnóstico son los falsos negativos y falsos positivos, considerando si se diagnostica un falso negativo malinterpretado las

imágenes de rayos x como normales, corre el riesgo de que el paciente infectado con COVID-19 agrave su salud al no recibir atención médica adecuada o contagie a futuros pacientes. También, los falsos positivos ocurren si se diagnostica erróneamente al paciente que no están infectado con COVID-19 y se le interna con los pacientes infectados y exponiéndolo a la carga viral del coronavirus.

Con el objetivo de abordar este problema, esta tesis tiene como propósito diseñar una arquitectura de red neuronal convolucional utilizando imágenes de rayos X de tórax para el diagnóstico eficaz del COVID-19 en el contexto de la pandemia en Perú, que abarco desde el inicio del estado de emergencia en el 2020 has el fin de 2022.

### **1.1.1. Problema general**

¿Cómo diseñar una arquitectura de red neuronal convolucional para el diagnóstico de COVID-19 mediante imágenes de rayos x?

### **1.1.2. Problemas específicos**

- ¿Cómo preprocesar las imágenes de rayos x para el desarrollo del diagnóstico de la arquitectura de red neuronal convolucional?
- ¿Cómo configurar las capas convolucionales para lograr el diagnóstico de COVID-19 mediante imágenes de rayos x?
- ¿Cómo configurar las capas totalmente conectadas para lograr el diagnóstico de COVID-19 mediante imágenes de rayos x?
- ¿Cómo validar la arquitectura de red neuronal convolucional para el diagnóstico de COVID-19 mediante imágenes de rayos x?

## **1.2. Objetivos**

### **1.2.1. Objetivo general**

Diseñar una arquitectura de red neuronal convolucional para el diagnóstico de COVID-19 mediante imágenes de rayos X.

### **1.2.2. Objetivos específicos**

- Preprocesar las imágenes de rayos X para el desarrollo del diagnóstico de la arquitectura de red neuronal convolucional.
- Configurar las capas convolucionales para lograr el diagnóstico de COVID-19 mediante imágenes de rayos X.
- Configurar las capas totalmente conectadas para lograr el diagnóstico de COVID-19 mediante imágenes de rayos X.
- Validar la arquitectura de red neuronal convolucional para el diagnóstico de COVID-19 mediante imágenes de rayos X.

## **1.3. Justificación**

### **1.3.1. Técnica**

Debido a la alta demanda de imágenes médicas y la capacidad limitada de los radiólogos en el Perú por la pandemia COVID-19, el proyecto de investigación se centra en el diagnóstico mediante el uso de redes neuronales convolucionales. Estas redes serán entrenadas utilizando imágenes de rayos X del tórax, con el propósito de reducir significativamente el tiempo necesario para obtener un diagnóstico preciso.

### **1.3.2. Teórica**

La aplicación del aprendizaje automático en los recientes años tuvo un crecimiento de popularidad y en su utilidad. Unas de las categorías son las redes convolucionales en el área médica, que a nivel internacional tuvo más desarrollo para

el diagnóstico, clasificación en anomalías en el cuerpo debido a su gran capacidad de aprendizaje en imágenes radiológicas. El proyecto de investigación no solo cumple con los requisitos de línea de investigación de la bioingeniería, sino también fomenta estas áreas de investigación, debido al último censo nacional de investigación y desarrollo del 2016 solo se utilizó el 0.08 % del PBI, ya que hay bajo nivel de investigación en nuestro país[9].

### **1.3.3. Económica**

El proyecto de investigación se enfoca al diseño de la red neuronal convolucional cuya inversión no es muy costosa, debido que se utiliza el intérprete Python, además va a ser beneficioso para los centros de salud que cuentan con el área de imágenes medica debido que la licencia de Python es libre.

### **1.4. Importancia**

El aprendizaje automático ha presentado una herramienta alternativa de diagnóstico en el sector salud, revelando grandes innovaciones en la identificación de automática de enfermedades como detección de células cancerígenas, reconocimiento cáncer a la piel, detección de tumores, entre otras enfermedades con la ayuda de las redes neuronales convolucionales. Si bien existe muchas aplicaciones en aprendizaje profundo se aplican en países desarrollado, pero en países de en vías de desarrollo como en Perú, todavía no hay muchos estudios en el campo del aprendizaje automático aplicado en el sector salud.

La tesis propone un modelo de CNN que fortalece el diagnóstico en radiografías del tórax en pacientes con COVID-19 de manera automatizada gracias a la potencia informática y grandes conjuntos de datos digitales. Además, va a fortalecer a los profesionales de la salud para mejorar la práctica médica para avanzar en el

conocimiento medico basado en los datos que se genera en los centros de salud del Perú.

## **1.5. Limitaciones**

Para el desarrollo del proyecto de investigación, las limitaciones encontradas son las siguientes: los centros de salud públicas cuentan con banco de imágenes médicas, pero su acceso es restringido, además estas imágenes no se encuentran correctamente clasificadas de acuerdo con la enfermedad diagnosticada.

## **1.6. Resultados esperados**

### **1.6.1. General**

Diseñar la arquitectura de red neuronal convolucional diagnosticar el COVID-19 mediante imágenes de rayos X.

### **1.6.2. Específicos**

- Preprocesar las imágenes de rayos X permiten el desarrollo de la arquitectura de red neuronal convolucional.
- Configurar las capas convolucionales logran el diagnóstico de COVID-19 mediante imágenes de rayos X.
- Configurar las capas totalmente conectadas logran el diagnóstico de COVID-19 mediante imágenes de rayos X.
- Validar la arquitectura de red neural convolucional logran óptimas métricas el diagnóstico de COVID-19 mediante imágenes de rayos X.

## **1.7. Variables**

### **1.7.1. Variable independiente**

Arquitectura de red neuronal convolucional

### 1.7.2. Variable dependiente

Diagnóstico

### 1.7.3. Operacionalización de variables

Tabla 1. Operacionalización de variables

<b>Problema</b>		<b>¿Cómo diseñar una arquitectura de red neuronal convolucional para el diagnóstico del COVID-19 mediante imágenes de rayos x?</b>		
		<b>Independiente</b>		<b>Dependiente</b>
<b>Variab</b>	<b>Arquitectura de red neuronal convolucional</b>	<b>Diagnostico</b>		
<b>Definición conceptual</b>	La arquitectura de red neuronal convolucional fue diseñada para trabajar con imágenes para la detección de objetos, clasificación y segmentación. La CNN está formado de diferentes capas como las siguientes: capas convolucionales, capas de agrupación y capas completamente conectadas.	Diagnóstico es la recopilación de información para analizar e identificar anomalías y lograr un pronóstico. El diagnóstico en la salud se aplican diversos procesos para obtener información adicional del estado del paciente y esto es adquirida mediante la prueba de diagnóstico.		
<b>Definición operacional</b>	La variable expresa parámetros que son pesos numéricos que aprendieron durante el entrenamiento entre las capas convolucionales y completamente conectadas.	La variable expresa mediante la sensibilidad y la especificidad		
<b>variable independiente</b>	<b>Arquitectura de red neuronal convolucional</b>			
<b>Dimensiones</b>	<b>Indicador</b>	<b>Unidad</b>	<b>Tipo de Variable</b>	<b>Instrumento</b>
<b>Capa convolucional</b>	Numero de pesos que aprendieron	Unidades	Numérica continua	Visual Studio Code
<b>Capa de agrupación</b>	Tamaño	Unidades	Numérica continua	Visual Studio Code
<b>Capa completamente conectada</b>	Numero de pesos que aprendieron	Unidades	Numérica continua	Visual Studio Code
<b>Variable dependiente</b>	<b>Diagnostico</b>			
<b>Dimensiones</b>	<b>Indicador</b>	<b>Unidad</b>	<b>Tipo de Variable</b>	<b>Instrumento</b>
<b>Prueba de diagnóstico</b>	Sensibilidad es que la prueba sea positiva en alguien con la enfermedad	(%) Porcentaje	Numérica continua	Visual Studio Code
	Especificidad es que la prueba sea negativa en alguien que no tenga la enfermedad	(%) Porcentaje	Numérica continua	Visual Studio Code

Fuente: Elaboración propia

## **CAPÍTULO II**

### **MARCO TEORICO**

#### **2.1. Antecedentes del Problema**

##### **2.1.1. Nacionales**

García, en el 2020, realizó la tesis doctoral “Sistemas inteligentes móvil basado en redes neuronales convolucionales para el diagnóstico previo de enfermedades renales”, donde se propone generar un instrumento que permita diagnosticar las enfermedades renales. La investigación es de tipo aplicado que busca automatizar el diagnóstico a un nivel explicativo. Los resultados obtenidos a través de cálculo de los indicadores de sensibilidad, eficacia y especificad, cuyo resultado va a medir el rendimiento de la red neuronal convolucional propuesto por el autor, lo cual es entrenado con un total de 268 imágenes de tiras con resultados del paciente distribuidos entre 154 enfermos y 154 sanos, Se logró una sensibilidad de 100 % y especificad de 99 %; se concluye que en la red neuronal convolucional propuesto es una herramienta efectiva por lo que se obtiene resultados aceptables[10].

Por su parte, Caya et al. (2020), realizó la tesis de pregrado “Evaluación de modelos de redes neuronales convolucionales aplicado a radiografías de tórax, para apoyar al proceso de diagnóstico de neumonía asociada el COVID-19”. El trabajo consiste en evaluar tres modelos de redes convolucionales en imágenes de rayos x del tórax por clasificación de imágenes aplicando, el lenguaje de programación utilizado es Python, las arquitecturas a evaluar son ResNet50 y InceptionV3 con la base de datos extraídos de la Sociedad Italiana de Radiología Medica e Intervencionista e la plataforma Kaggle para el desarrollo. Asimismo, la base de datos fue dividida en dos conjuntos entrenamiento con 848 y validación 212 imágenes del tórax. Los resultados se presentan en la matriz de confusión para su comparación, que se visualiza los éxitos

y fracasos de cada uno de los modelos de red neuronal convolucional con los siguientes parámetros sensibilidad, especificidad y precisión. Para ResNET50 se obtuvo 95.45 %, 97.73 % y 95.55 %. InceptionV3 obtuvo 98.86 %, 99.24 % y 98.86 %. Se llegó a la conclusión de que el modelo más efectivo de los dos fue Inception [11].

A su vez, Fernández, en el 2019, realizó la tesis de pregrado “Redes convolucionales para la determinación de cáncer de piel a partir de imágenes digitales”. El trabajo propone una arquitectura de red convolucional para dar un diagnóstico de lunares malignos o benignos, se utiliza las bases de datos de The International Skin Imaging Collaboration (ISIC) y PH2Dataset, con la base recopilada por médicos dermatólogos especialistas de la patología. Además, la base de datos cuenta con 700 imágenes que serán aplicadas en la red neuronal convolucional. La metodología inicia procesando y segmentando las imágenes el área de los lunares malignos y benignos luego pasa a ser entrenada por la propuesta de red neuronal convolucional para determinar con un “0” si la lesión es benigna y “1” si es maligna. Estos resultados de la CNN son con una precisión de 97 %, sensibilidad de 98 % [12].

Mientras que Calero y Chirinos, en el 2021, efectuó la tesis de pregrado “Detección del uso correcto de mascarillas utilizando red neuronal convolucional para el ingreso de personas a un laboratorio de una universidad”. El trabajo consistió en implementar una red neuronal convolucional para el acceso a laboratorios universitarios. Asimismo, la metodología propuesta abarcó seis etapas: adquisición de imágenes. Creación de base de datos, determinación del modelo, interfaz de la aplicación y pruebas. Además, durante el análisis de los resultados, se observa que el tercer modelo de red neuronal convolucional con un mayor número de capas mostró un rendimiento sobresaliente en la identificación de las categorías de mascarillas,



ausencia de mascarillas y uso incorrecto de la misma. Se logró una clasificación con alta precisión y exactitud del 99 %[13].

A su turno, Rodriguez et al., en el 2022, realizó la tesis de pregrado “Diseño de una red neuronal convolucional para el reconocimiento facial”. El objetivo de la tesis fue el desarrollo de una red neuronal capaz de realizar reconocimiento facial. Asimismo, la metodología adopta un enfoque mixto con un diseño experimental, siguiendo el método *deming* para la planificación, ejecución, evaluación de métricas y mejora continua. Además, las técnicas e instrumentos involucran observación y generación de base de datos a partir de algoritmos para extraer rostro de videos grabados. Se logró general una base de datos con un total de 7127 imágenes, de las cuales 5627 se destinaron para el entrenamiento y 1500 para la validación de la red neuronal. Tras el entrenamiento, obtiene un porcentaje promedio de acierto en el reconocimiento de rostros de 91.97 %[14].

### **2.1.2. Internacionales**

Sagar et al. 2021, en el artículo “A multi model ensemble deep convolutional neural network structure for detection of COVID-19”, tuvo el objetivo de hallar un modelo de arquitectura de red neuronal convolucional profunda (DCNN) aplicando transferencia de aprendizaje de cuatros redes preentrenadas: VGGNet, DenseNet, NASNet y GoogleNet para ayudar a los radiólogos a detectar el COVID-19 mediante imágenes de rayos x del tórax. Asimismo, la metodología empleada por los investigadores es cinco: iniciando con el preprocesamiento de datos, transferencia de aprendizaje, modelo de conjunto para la extracción de características, modelo propuesto de arquitectura DCNN y la implementación y capacitación en dos escenarios para el modelo propuesto. Los resultados obtenidos en el primer escenario, el modelo entrena y valida con 5441 de clase normal, 7966 de neumonía adquirida por comunidad

(CAP) y 755 de COVID-19 se obtiene una exactitud del 88.92 %. Para el segundo escenario solo se aplican dos clases COVID-19 y no COVID-19. La clase no COVID-19 es la fusión de las clases normal y CAP, con las dos clases definidas y aplicando transferencia de aprendizaje se obtiene una mayor exactitud de 98,58 % [15].

Por su parte, Hatice Catal el 2021, en el artículo “COVID-19 diagnosis *with deep learning* con el objetivo de detectar el COVID-19 automáticamente a partir de datos de tomografía computarizada de tórax. La metodología empleada es la obtención de las imágenes de tomografía computarizada de pacientes de con coronavirus y no COVID-19 para ser preprocesada, debido que las imágenes no tenían una característica estándar, segmentación de la base de datos en entrenamiento, validación y prueba, luego son entrenadas con 757 imágenes con tres modelos de red neuronal convolucional (CNN): CNN personalizada, ZFnet y DensesNet21 para su correcta clasificación. Asimismo, la CNN personalizada conformada con 13 convoluciones y 13 activaciones obtiene la precisión de 100 % y exactitud de 95 % mientras que la arquitectura ZFnet con 5 capas de convolución obtienen la precisión de 85 % y exactitud 86 %. DensesNet21 con 86 % y 84 %[16].

Mientras que Arkja Saxena et al. (2020), en el artículo “A glaucoma detection using convolutional neural network”, desarrolló un modelo de aprendizaje profundo para la detección del glaucoma con el uso de la red neuronal convolucional con el fin de lograr un rendimiento para la detección de glaucoma. Asimismo, la metodología propuesta por los autores es la obtención de la base de datos del fondo óptico, obtenidas de ORIGA con 168 imágenes de glaucoma y 482 de fondo óptico normal mientras la base SCES tiene 46 con glaucoma y 1676 fondo óptico normal. Estas imágenes son redimensionadas a la resolución de 256 \* 256 y divididas en cuadrículas con el método de ARGALI para la detección del nervio óptico, para ser aplicada a la arquitectura

propuesta por los autores que está diseñando con seis capas. Las 4 primeras son capas convolucionales y las dos últimas las copas completamente conectadas, esta última capa entrega al clasificador para la detección del glaucoma. La arquitectura propuesta fue aplicado a cada base de datos por separado. La base de datos ORIGA obtuvo una exactitud de 82 % mientras SCES obtuvo 88 %, llegando a la conclusión que la arquitectura diseñada funciona muy bien con resultados satisfactorio[17].

A su vez, Enas M.F et al. 2021, en el artículo “Malignant and nonmalignant classification of breast lesions in mammograms using convolutional neural networks”, plantearon los objetivos de la investigación donde buscan desarrollar un sistema para clasificar lesiones mamarias malignas y no malignas. Asimismo, el método propuesto por los autores consiste en una fase de preprocesamiento, donde se realiza la mejora de las imágenes, eliminación de ruido y siguiendo con la fase de construcción de la red neuronal convolucional (CNN), donde se extraerán las características de las imágenes y pasar por el clasificador utilizando bases datos diferentes MIAS y INbreast, por lo que se obtiene un alto rendimiento en la sensibilidad, especificad, exactitud alcanzaron 96,55 %, 96,49 % y 96,52 con la base de datos INbreast y con MIAS alcanzaron 98 %, 92,6 % y 95,3 % [18].

Mientras que Tariq Rahim et al. (2021), en el artículo “A deep convolutional neural network for the detection of polyps in colonoscopy images”, proponen un modelo basado en redes neuronales convolucionales profundas (DCNN) para la detección de pólipos en imágenes de colonoscopia. Asimismo, la metodología planteada es por fases como las siguientes: preprocesamiento, construcción de la arquitectura de la red neuronal convolucional profunda (DCNN) que está formada 16 capas convolucionales con 2 capas totalmente conectadas (FCN) de la última fase de entrenamiento de 100000 épocas, en la cual se obtuvieron una precisión de 94,44 %,

sensibilidad de 82,92 %. Estos resultados se compararon con el desafío de MICCAI 2015, reflejando que la propuesta de la red neuronal convolucional tienen un mejor desempeño en términos de precisión, sensibilidad[19].

A su turno, Shanthi. T et al. 2020, en el artículo “Automatic diagnosis of skin diseases using convolution neural network”, propusieron una red neuronal convolucional con la capacidad de detectar y categorizar cuatro tipos de enfermedades de la piel como el acné, queratosis, eccema herpético y urticaria. Asimismo, la base de datos utilizada fue la de DermNet con 174 imágenes, donde se segmentó en dos, fase de entrenamiento involucrando 105 imágenes y la de prueba con 69 imágenes. Para comenzar con el entrenamiento se configura la arquitectura de red neuronal convolucional que constituye cinco capas convolucionales, una capa de unidad lineal rectificadora (ReLU), tres capas de agrupación y acabando con la última capa completamente rectificadora. El modelo 1000 se completó en tres días, por lo que se obtiene una exactitud del 98,6 % al 99,04 para el diagnóstico de enfermedad de la piel acné, queratosis, eccema herpético y urticaria respectivamente. [20].

Por su parte, Agustín et al. (2019) realizaron la tesis de pregrado “Modelo de red neuronal convolucional para el diagnóstico de neumonía en imágenes radiológicas”, que consiste en usar redes neuronales convolucionales para generar un modelo de aprendizaje automático para diagnosticar neumonía en imágenes radiológicas en DICOM. Asimismo, la metodología aplicada fue Team Data Science Process (TDSP) que es un ciclo de pasos para proyectos de modelos de inteligencia artificial para encontrar patrones ocultos y convertirlo en información de conocimiento en una población de 10000 imágenes de pacientes sanos y con neumonía extraído del Instituto Nacional de la Salud de América del Norte (NIH), para el desarrollo se tomaron 630 imágenes de muestra 270 de test y 270 para la evaluación los resultados

obtenido que la red neuronal convolucional con los entrenamientos tiene una exactitud de 73 % y 21 % de error[21].

A su vez, Camilo et al. (2022) realizó la tesis de postgrado “Modelo de machine learning para clasificación de pacientes con glaucoma en la población del Valle del Cauca”. Asimismo, el objetivo de investigación fue desarrollar y validar un modelo de aprendizaje automático para el tamizaje de glaucoma en la población del Valle del Cauca. La investigación utilizó el método CRIPS-DM (Cross Industry Standard Process for Data Mining) presentado en seis fases: fase 1 de entendimiento del negocio, fase 2 compresión de los datos, fase 3 preparación de datos, fase 4 modelización, fase 5 evaluación, fase 6 despliegue. El modelo de *deep learning* usado es Inception V3, por lo cual estos resultados son analizados en la matriz de confusión, arrojando una precisión de 77.8 % y una sensibilidad de 76.6 % [22].

Por su parte, Nieves en el 2019, efectuó la tesis de pregrado “Identificación automática de glaucoma a partir de imágenes oct circumpapilares mediante el uso de redes neuronales convolucionales”. Asimismo, el objetivo de la investigación es desarrollar un modelo de clasificación automática empleando imágenes oct circumpapilares para diferenciar en pacientes enfermos y pacientes con glaucoma. La metodología empleada consta de seis etapas: preprocesamiento de datos, diseño de arquitectura, compilación del modelo, entrenamiento y validación del modelo, realización del test y la evaluación. Se obtuvo como resultado bajo la arquitectura VGG16, el valor de 92,4 % de precisión y una sensibilidad de 92.4 %[23].

## 2.2. Bases teóricas

### 2.2.1. Arquitectura de red neuronal convolucional

Las redes convolucionales, también conocidas como redes neuronales de convolución (CNN), son técnicas del aprendizaje profundo[24] que fueron diseñadas para trabajar mayormente con imágenes bidimensionales, debido a que presentan entradas de estructura cuadrada [25]. Una arquitectura de red neuronal convolucional compone de múltiples capas de convolución, capas de agrupación y capas completamente conectadas como se ve en la figura 3. La red neuronal convolucional extrae la información de cada pixel de la imagen de entrada y lo analiza para ser procesada mediante las convoluciones y agrupaciones.

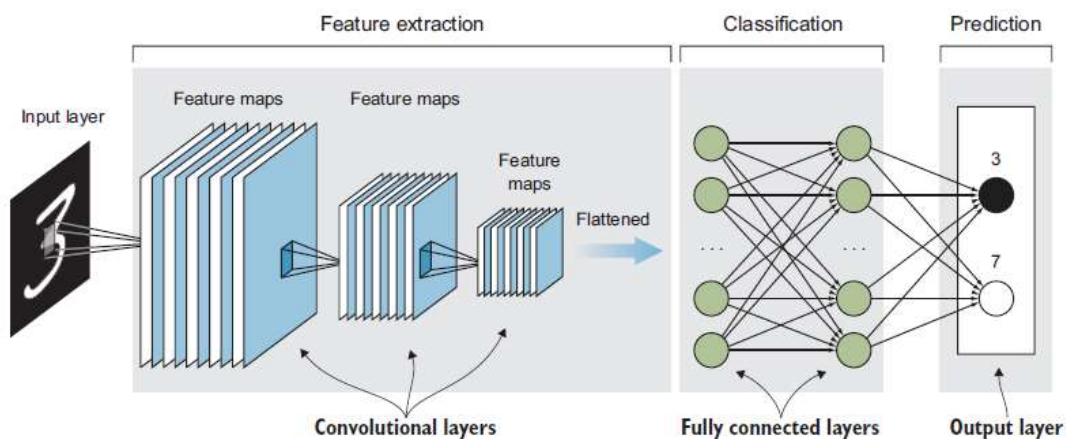


Figura 3. Arquitectura de la red neuronal convolucional, por Deep Learning for Vision Systems-Mohamed Elgendy

Las aplicaciones de las redes neuronales convolucionales en el área de la medicina son numerosas como la clasificación de imágenes médicas en la identificación ciertas características de enfermedades o anomalías anatómicas para diferenciar entre una imagen normal y anormal. La otra aplicación es la detección de un objeto en interés como órganos, células, tumores, etc.[26]. A diferencia con la clasificación, la detección tiene la habilidad de ver la imagen, dividir en regiones más pequeñas y etiquetar cada región con el nombre a la clase que pertenece en una

imagen[27] y la segmentación es una técnica más avanzadas y consume mucho tiempo, este proceso separa una estructura anatómica, lesiones de la imagen original y es esencial el análisis de parámetros clínicos como forma volumen, etc.[26]

### 2.2.2. Capa convolucional

La capa convolucional o *convolutional layer* es el elemento central de una arquitectura de red convolucional y responsable del trabajo pesado computacional. Estas capas realizan búsqueda de características con los filtros kernel que se desplazan a través de los píxeles de la imagen para la extracción de características importantes generando una nueva imagen o también llamada mapa de característica[27].

Asimismo, la operación convolución en imágenes es el cálculo matemático que aplica el producto de dos funciones matriciales, donde la primera función es una imagen “A” con un ancho “k” y alto “l” y la segunda función es el filtro kernel “B”, para generar el mapa de características[28]. Esta operación se repite sobre todas las posiciones espaciales de la imagen y se expresan en la siguiente fórmula.

$$C(i, j) = \sum_K \sum_l A(k, l).B(i - k + 1, j - l + 1)$$

Donde:

A = Imagen bidimensional de entrada

B = Filtros kernel o máscaras de convolución

La convolución aplica la técnica de la superposición entre dos matrices.

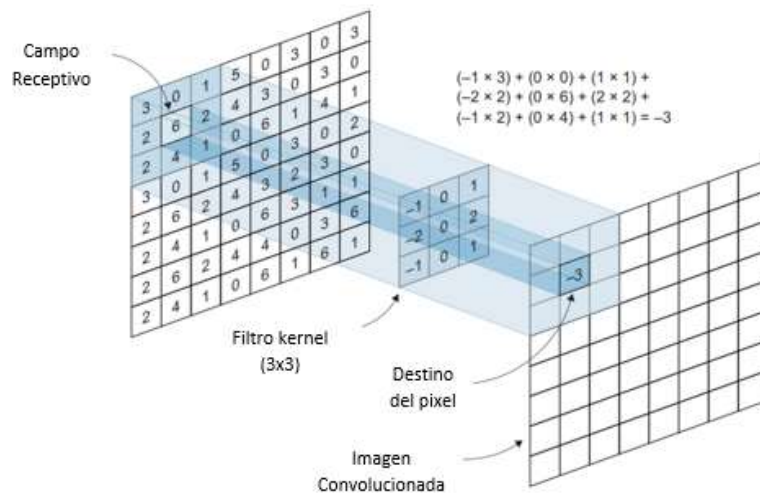


Figura 4. Convolución por superposición, por Neural Networks and Deep Learning-Charu C.A

### 2.2.3. Capa de agrupación

La capa de agrupación o *pooling layer* se encarga de reducir las dimensiones de las nuevas imágenes producidos por la convolución; este proceso conserva los datos de características importantes además reduce el costo computacional en el cálculo en la CNN[29]. Asimismo, hay dos técnicas de agrupación: la agrupación máxima y la agrupación promedio; entre estas técnicas la más común y efectiva en la arquitectura de red neuronal convolucional es la agrupación máxima[24].

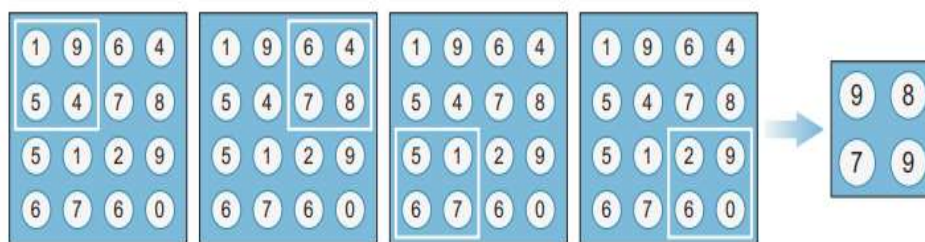


Figura 5. Filtro de agrupación 2 x 2, por Deep Learning for Vision Systems-Mohamed Elgendy

### 2.2.4. Capas totalmente conectadas

Las capas totalmente conectadas o Fully Connected Layers (FCL) se encuentran las neuronas completamente conectadas con todas las neuronas de las capas



anteriores y es un componente importante de las CNN, debido que realiza un razonamiento de alto nivel. Asimismo, para llevar a cabo las capas totalmente conectadas se tuvieron que realizar operaciones de convolución y agrupación para la extracción de característica, estas características son aplanadas en un vector de características unidimensional antes de pasar a las capas totalmente conectadas[28].

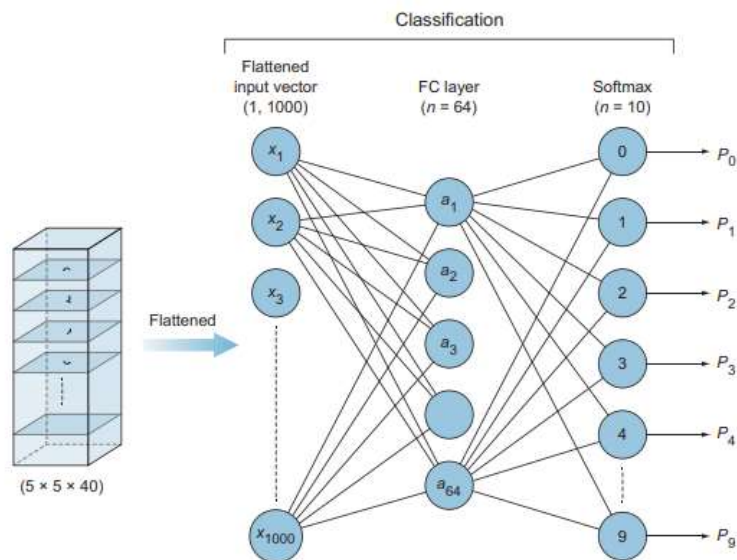


Figura 6. Capas totalmente conectadas, por Deep Learning for Vision Sytems- Mohamed Elgendy

### 2.2.5. Métricas

Las métricas son las que evalúan la efectividad del rendimiento de un algoritmo del aprendizaje automático diseñado. Se utilizan tres métricas como precisión, recuperación y matriz de confusión[30].

### 2.2.6. Matriz de confusión

La matriz de confusión es una representación gráfica de valores del rendimiento alcanzado, además es una matriz de  $K \times K$ , donde  $K$  es el número de clases [31]. La matriz compara los valores objetivos reales como los predichos por el modelo. Esto da una visión holística de que tan bien esta funcionando el modelo de clasificación

y los errores que se está cometiendo. En la tabla (2) se observan cuatro métricas: verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falso negativos (FN)[29].

Tabla 2. *Matriz de confusión*

		Predicción	
		Positivo	Negativo
Clases	Positivo	TP	FN
	Negativo	FP	TN

Fuente: Elaboración propia

A partir de estas estadísticas obtenemos las siguientes definiciones:

#### 2.2.6.1. Exactitud

Exactitud (*accuracy*) indica el número de cuantas veces el modelo predice correctamente como clase positiva y negativa. La exactitud se calcula por la siguiente ecuación.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

#### 2.2.6.2. Precisión

Precisión es la relación entre los verdaderos positivos y todos los resultados previstos. La precisión se calcula usando la siguiente ecuación.

$$PR = \frac{TP}{TP + FP}$$

#### 2.2.6.3. Sensibilidad

Sensibilidad o también llamada *recall* es la proporción de observaciones positivas precisas en la clase real. La sensibilidad se calcula por la siguiente ecuación.

$$RE = \frac{TP}{TP + FN}$$

#### **2.2.6.4. Especificidad**

Especificidad es la proporción de observaciones negativas precisas en la clase real. La especificidad se calcula por la siguiente ecuación.

$$SP = \frac{TN}{TN + FP}$$

#### **2.2.6.5. F-Score**

F-score es la métrica que tiene en cuenta la precisión y la sensibilidad de la misma manera. Es la media armónica entre precisión y sensibilidad.

$$F - score = \frac{2 * PR * RE}{PR + RE}$$

#### **2.2.7. Diagnóstico**

El diagnóstico clínico es la aplicación procesos para obtener información adicional del estado del paciente mediante las pruebas de diagnósticos.

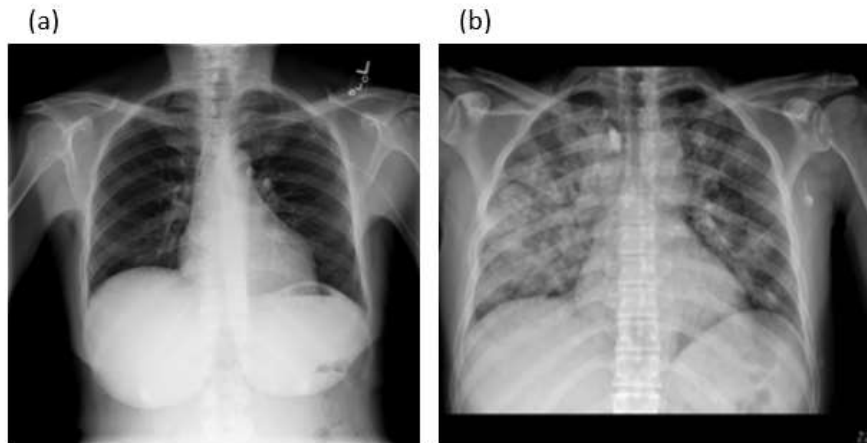
#### **2.2.8. Prueba de diagnostico**

Las pruebas de diagnóstico es un método para evaluar el estado del paciente con cuatro combinaciones de posibles. Estos resultados de la prueba, dos las que son correctas: verdadero positivo y verdadero negativo; y dos en los que son incorrectas: falso positivo y falso negativo[32].

#### **2.2.9. Diagnóstico del COVID-19**

Para determinar el diagnóstico del COVID-19 se realizan pruebas de radiografía al tórax para identificar procesos inflamatorios de los pulmones y rastrear el proceso de la enfermedad.

La neumonía viral producida por el coronavirus genera el aumento de la densidad pulmonar, exactamente en los alveolos, debido a la presencia de agua, proteínas plasmáticas y células inmunitarias en esta área. Este proceso se visualiza en la radiografía del tórax como opacidades de vidrio esmerilado [33].



*Figura 7.* Radiografía del tórax: (a) paciente normal (b) paciente con COVID, por Computational Modelling and Imaging for SARS-CoV-2 and COVID-19-S. Prabha

La radiografía (a) de la figura 7 pertenece a un paciente sano, generalmente el área de los pulmones debe aparecer oscura, debido a la presencia de aire, porque todavía hay presencias de tejidos, mientras la radiografía (b) de un paciente diagnosticado con COVID-19, el pulmón presenta opacidad en vidrio esmerilado indicando que el pulmón ya este enfermo.

## **2.2.10. Definición de Términos Básicos**

### **2.2.10.1. Red neuronal convolucional**

La red neuronal convolucional es una categoría de las redes neuronales artificiales convolucional que fue diseñanda para trabajar con datos unidimensional, bidimensional y tridimensional para determinar una predicción[34].

#### **2.2.10.2. COVID-19**

El COVID-19 es una nueva enfermedad de la familia de coronavirus y se debe a una infección causada por el virus SARS CoV2 y causa infección aguda con síntomas respiratorias[35].

#### **2.2.10.3. Clasificación**

La clasificación se emplea para predecir el tipo o clase de un objeto dentro de un conjunto limitado de opciones y la variable de salida es categórica[36].

#### **2.2.10.4. Filtro kernel**

El filtro kernel es una matriz de valores aleatorios con el propósito de desenfocar, agudizar, detectar bordes para la extracción de características de la imagen[29].

#### **2.2.10.5. Rayos x del tórax**

La radiografía del tórax es una representación en 2D de una víscera torácica en 3D y se usa para ayudar a diagnosticar y controlar el tratamiento afecciones pulmonares como el cáncer, neumonía y enfisema, etc. [37].

#### **2.2.10.6. Procesamiento de imágenes**

El procesamiento de imagen es el proceso de manipulación de imágenes a través de operaciones matemáticas para mejorar la imagen o extraer información por medio de una computadora digital [30].

## CAPÍTULO III

### METODOLOGIA

#### **3.1. Método de Investigación**

La investigación adopta el enfoque cuantitativo, que implica una medición precisa y una evaluación basada en métricas numéricas, como la exactitud, precisión, sensibilidad, especificidad y F1-score. Estas métricas cuantitativas proporcionan una comprensión detallada y cuantificable del rendimiento del modelo diseñado en el diagnóstico del COVID-19 a partir de imágenes de rayos X.

#### **3.2. Tipo de Investigación**

El estudio es tecnológico, en palabras de Ñaupás, “la investigación tecnológica tiene como propósito resolver problemas mejorando, perfeccionando u optimizando el funcionamiento de los sistemas en beneficio de la vida productiva de la sociedad aplicando los conocimientos científicos “[38]. Por ello, el estudio es de tipo tecnológico, en vista que se diseña una arquitectura de red neuronal convolucional con imágenes de rayos x del tórax para realizar diagnóstico de COVID-19, lo cual se tomó en cuenta los conocimientos científicos y tecnológicos para el correcto diseño.

#### **3.3. Alcance de la Investigación**

El estudio es de nivel aplicada, de acuerdo con Ciro, el nivel de investigación aplicado tiene como propósito solucionar problemas de la sociedad diseñando tecnología blanda [39].

Tabla 3. *Tipo y nivel de investigación*

<b>Tipo Investigación</b>	<b>Nivel de Investigación</b>
<b>Investigación Básica</b>	Investigación descriptiva
	Investigación explicativa
<b>Investigación Tecnológica</b>	Investigación explicativa
	Investigación aplicada

*Fuente: Metodología de la investigación tecnológica-Ciro Ezpinoza*

El estudio es tecnología blanda, debido a que son diseños intangibles como el desarrollo de programas que mejoran el funcionamiento en las áreas administrativas, industrias, medicas ,etc.[38]

### **3.4. Diseño de investigación**

El trabajo de investigación es de diseño transversal; en palabras de Ñaupas, el estudio transversal es la variable se examina simultáneamente durante un año[38].

### **3.5. Población**

Para el desarrollo de la tesis, la población enfocada consistió en las bases de datos generadas por organizaciones e instituciones radiológicas que se encuentran públicamente en sus sitios web.

Tabla 4. *Base de datos de instituciones*

<b>Institución/Organizaciones</b>	<b>Cantidad de imágenes</b>
European Society of Radiology[40]	919 imágenes
Italian Society of Medical and Interventional Radiology[41]	3 616 imágenes
Medical Imaging Databank of the Valencia Region[42]	5028 imágenes
The National Institutes of Health[43]	100 000 imágenes
Vision and Image Analysis Group[44]	59 000 imágenes
Stanford-Center for Artificial Intelligence in Medicine & Imagingapli[45]	3 256 imágenes

*Fuente: Elaboración propia*

### 3.6. Muestra

La muestra se obtiene aplicando el muestreo por accidente; de acuerdo con Ñaupás, este consiste en elegir una muestra de la población para una mayor facilidad[38]. Por ello, se optó por la base de datos “Medical Imaging Databank of the Valencia Region (BIMCV)”. Esta elección se justifica por su estrecha relación con la pandemia de COVID-19 en comparación con otras bases de datos, simplificando la recopilación al estar claramente segregada en dos conjuntos: COVID-19 positivo y COVID-19 negativo. La focalización en imágenes de rayos X para el diagnóstico de coronavirus coincide plenamente con la temática de la tesis. Asimismo, la base de datos recibió respaldo de la “Society for Imaging Informatics in Medicine (SIIM)” y la “Radiological Society of Nort America (RSNA)”, y fue evaluada por expertos en radiología de estas instituciones. En su primera versión, la base de datos consta de 5028 imágenes de rayos X del tórax, divididas en 2551 de pacientes sanos y 2477 de pacientes diagnosticados con COVID-19. Es esencial considerar que estos datos están sujetos a modificaciones, actualizaciones, adiciones según las necesidades organizativas.

Tabla 5. *Base de datos*

N.º	Modalidad	Numero de imágenes	Base de datos/ Institución
1	Rayos x del tórax	2551 imágenes de pacientes normales-COVID-19 negativo	Medical Imaging Databank of the Valencia Region (BIMCV)
2	Rayos x del tórax	2477 imágenes de pacientes infectados-COVID-19 positivo	Medical Imaging Databank of the Valencia Region (BIMCV)

*Fuente: Elaboración propia*



### 3.7. Técnicas e Instrumentos de Recolección de Datos

La tesis emplea la técnica empírica mediante la observación, de acuerdo Espinoza “la observación es una recolección de datos que permite acumular y sintetizar información sobre el objeto de estudio de investigación que tiene relación con el problema de investigación”[39].

Como instrumento se utilizó el *software* Visual Studio Code con el lenguaje de Python, el cual permite cómo evoluciona las métricas durante el entrenamiento de la arquitectura de la red neuronal convolucional como las siguientes: precisión de entrenamiento (acc), pérdida de entrenamiento(loss), precisión de validación (val\_acc) y pérdida de validación (val\_loss).

Tabla 6. *Formato de recolección de datos*

N.º épocas	Entrenamiento				
	Tiempo	loss	acc	val_loss	val_acc
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					

*Fuente: Elaboración propia*

### 3.8. Diseño de Ingeniería

Se aplica el método SEMMA, este método fue propuesto por el instituto “Analytics Software & Solutions” para proyectos de minería de datos. El método considera cinco etapas: sample, explore, modify, model, assess [46].

### **3.8.1. Muestra**

En la etapa muestra es la extracción de una muestra representativa de un gran conjunto de datos para manipularla rápidamente, además se aconseja crear conjuntos de datos de entrenamiento, prueba y validación a partir de la división de la muestra para evaluar la precisión[46].

### **3.8.2. Exploración**

La etapa exploración es donde el usuario busca una mejor comprensión del conjunto de datos con la exploración visual o numéricamente en la búsqueda de tendencias[46].

### **3.8.3. Modificación**

Esta etapa es la modificación de los datos mediante creación, selección, transformación para enforzar el proceso de selección del modelo[46].

### **3.8.4. Modelado**

La etapa modelada busca un modelo en particular y apropiada dentro de las situaciones específicas de minería de datos dependiendo de los datos. Las técnicas incluyen redes neuronales artificiales, árbol de decisión, regresión y otros modelos estadísticos[46].

### **3.8.5. Evaluación**

La última etapa es donde se evalúa el modelo para estimar cuan bien funcionan. El medio para evaluar el modelo es aplicar un conjunto de datos reservados durante la etapa del muestreo y que no se haya utilizado durante el proceso del modelo[46].

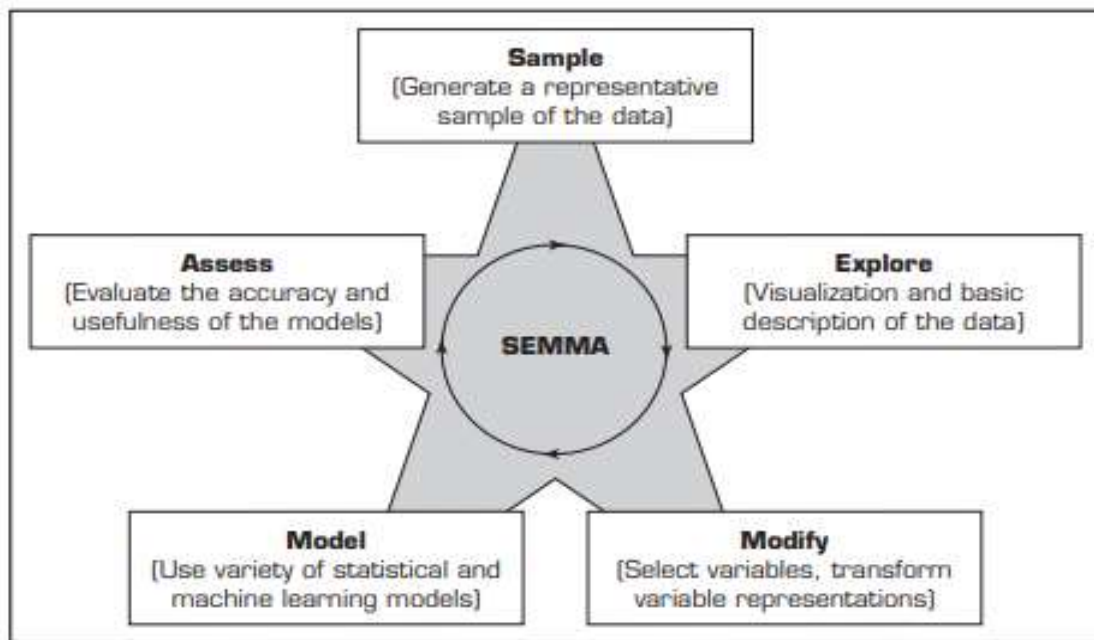


Figura 8. Proceso de etapas de SEMMA, por Business Intelligence and Analytics Systems for Decision Support-Ramesh Sharda

### 3.9. Materiales y Especificaciones Técnicas

#### 3.9.1. Python

La versión de Python utilizada es la 3.8.8. La razón por la que se eligió este lenguaje es por la compatibilidad de librerías de aprendizaje automático como TensorFlow y OpenCV, debido que algunas funciones de estas librerías son incompatibles con versiones más recientes.

#### 3.9.2. Hardware

Para cumplir los objetivos de la tesis se desarrolla en la portátil Asus X556UR cuya especificación se mientras a continuación.

- Procesador Core i5-7200U
- Frecuencia de procesador de 2,5 GHz
- Memoria RAM 8GB

- Procesador grafico NVIDIA GeForce 930x
- Windows10

### **3.10. Construcción del algoritmo**

El algoritmo propuesto para la tesis se implementó utilizando el lenguaje de programación Python y se desarrolló en el editor Visual Studio Code. A lo largo de su ejecución, se siguen todas las etapas del proceso SEMMA, asegurando un enfoque completo y estructurado en el desarrollo y evaluación de la arquitectura de red neuronal convolucional

#### **3.10.1. Muestra de base de datos**

La tesis utilizó la base de datos disponible del Medical Imaging Databank of the Valencia Region (BIMCV), que proporcionó un conjunto de imágenes de rayos X de tórax. Este conjunto de datos está compuesto por un total de 5028 imágenes, divididas en dos clases etiquetadas como “normal” y “COVID-19”, y fue recopilada en el 2020. Estas clases fueron creadas y evaluadas utilizando imágenes de pacientes de diversa edades y géneros, siendo revisadas por radiólogos con una experiencia de 3 a 7 años en el análisis de imágenes de tórax.

Además, el conjunto de bases de datos fue dividida en tres conjuntos distintos: entrenamiento, validación y prueba, siguiendo una proporción comúnmente empleada por investigadores, que es del 70 % para entrenamiento, 15 % para validación y 15 % para prueba[29]. Además, de las 5028 imágenes de clase “normal” y “COVID”, el subconjunto de entrenamiento se compones de 3521 imágenes, de las cuales 1788 de clase “normal” y 1733 son de la clase “COVID”. Asimismo, el conjunto de validación contiene 754 imágenes, con 383 de la clase “normal” y 371 de “COVID”. Por último, el conjunto de prueba está conformada por 753 imágenes, con 383 de la clase “normal” y 371 de “COVID”.

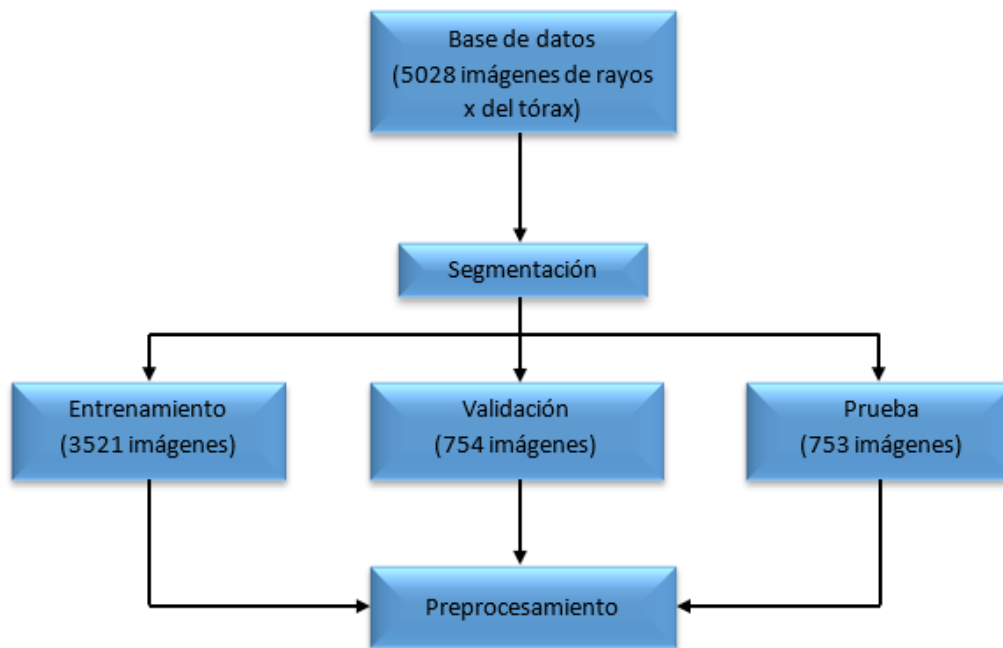


Figura 9. Diagrama de segmentación de datos

La división de la base de datos se realizó de manera aleatoria y se almacena en tres carpetas creadas específicamente: “Train\_set”, “Val\_set” y “Test\_set”. Cada una de estas carpetas contiene dos subcarpetas adicionales, cada una con el nombre de una de las clases, en la cuales se almacenaron los conjuntos de datos ya divididos.

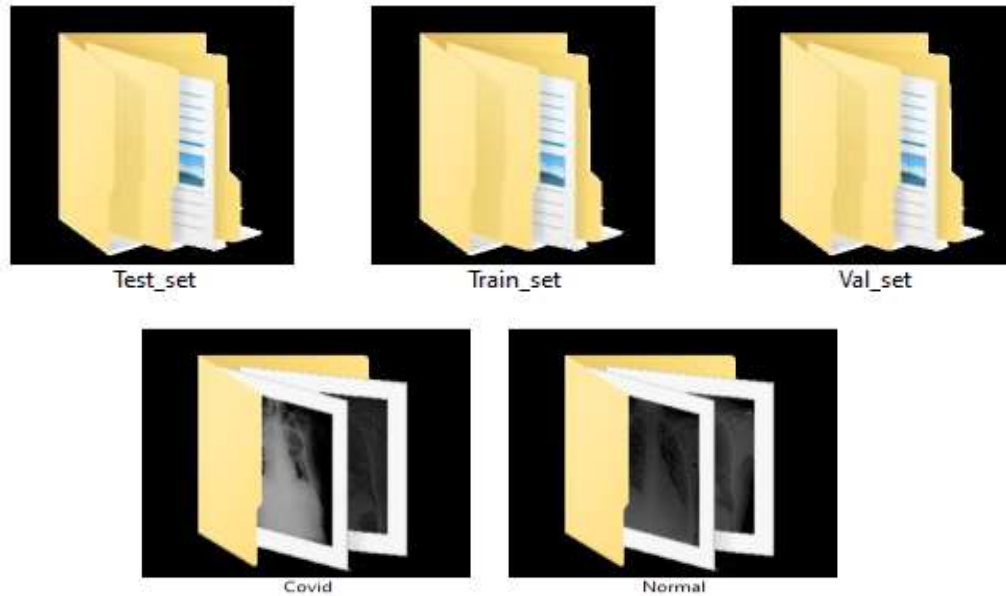


Figura 10. Carpetas creadas

### 3.10.2. Exploración de base de datos

En esta etapa, se observa que todas las imágenes de la base de datos tienen una dimensión tridimensional de 299 x 299 x 3, en formato PNG (Portable Network Graphics). El valor 3 se debe a que las imágenes están compuestas por las capas de colores RGB (rojo, verde y azul). Además, se identifica que muchas de las imágenes con COVID-19 exhiben un patrón recurrente de opacidad de vidrio esmerilado en uno o ambos pulmones, como se ve en la figura 11. Estas características visuales podrían ser indicativo de la presencia de la enfermedad y es esencial para comprender los

signos visuales específicos asociados con COVID-19 en las imágenes de rayos X del tórax.

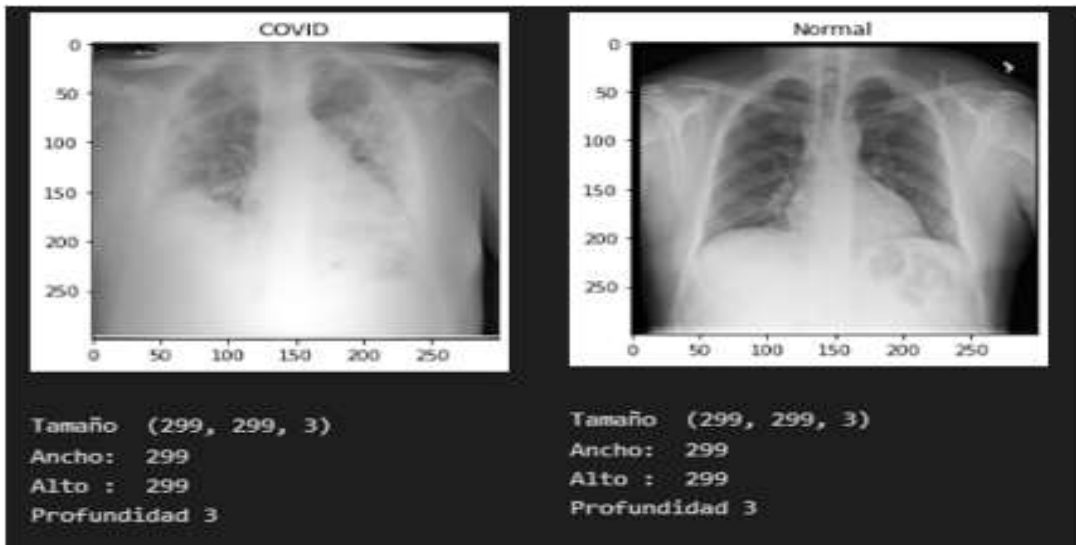


Figura 11. Dimensiones de imágenes de rayos X clase “COVID” y “Normal”

### 3.10.3. Modificación del base de datos

En esta etapa, el proceso se inicia mediante la importación de diversas librerías en el entorno de programación Python a través del editor Visual Studio Code. Entre las librerías esenciales se encuentran OpenCV, numpy, Os, Pytorch y Matplotlib. Estas librerías desempeñan un papel fundamental en la etapa de preprocesamiento de los datos.

```
1 Run Cell | Run Below | Debug Cell
2 #%%Importando librerías
3 import cv2
4 import os
5 import torch
6 import matplotlib.pyplot as plt
7 import numpy as np
```

Figura 12. Importación de librerías para el procesamiento

Una vez que estas librerías han sido importadas, se procede a definir dos variables, “Files” y “Class”, ambas de tipos lista. Estas variables son utilizadas para

acceder a las rutas de las tres carpetas previamente creadas, que están alojadas en la unidad “D”, La variable “Class” corresponde a las dos clases de las imágenes de rayos X del tórax, como se ve en la figura 13.

```
Run Cell | Run Above | Debug Cell
8  √  %% Rutas de las carpetas Test, Train y Val
9
10 √  Files = ['D:/Proyecto de tesis/Poblacion/Test_set',
11           'D:/Proyecto de tesis/Poblacion/Train_set',
12           'D:/Proyecto de tesis/Poblacion/Val_set',]
13
14  Class = ["Covid" , "Normal"]
15
```

Figura 13. Rutas de las carpetas y clases

Se han creado dos funciones esenciales: “equalization” e “image\_processing”, ambas diseñadas para llevar a cabo tareas específicas. La función “equalization” recibe un parámetro único, “img”, que representa la imagen en la que se desea mejorar el contraste. Esta mejora se aplica mediante operaciones matemáticas para determinar los valores mínimos y máximos de tonos de gris, contribuyendo así a la mejora global de la imagen.

$$Y(i, j) = 255 \cdot \frac{X(i, j) - x_{min}}{x_{max} - x_{min}}$$

Figura 14. **Fórmula máximos y mínimos de tonos de gris**

*Fuente: Elaboración propia*

La ecuación de la figura 14 es implementada en la línea 25 del algoritmo presente en la imagen siguiente. Esto permite aplicar de manera efectiva la transformación de ecualización a las imágenes, contribuyendo a la mejora de su calidad y contrasté.



```

Run Cell | Run Above | Debug Cell
16  %% Pre-Procesamiento de la poblacion
17
18  # Funciones
19  def Equalization(img):
20
21      #Copia de la img
22      img_x = img.copy()
23
24      #Ecuacion para valores de 0 a 255
25      img_x = 255 * ((img_x - img_x.min()) / (img_x.max() - img_x.min()))
26
27      # m y n toma los valores del tamaño de la img_x
28      n, m = img_x.shape
29      # Matrix de ceros de n*m de profundidad 1
30      y = np.zeros((n * m , 1), dtype=np.uint8)
31      # Ordena los valores de menor a mayor de acuerdo a su posicion
32      j = np.argsort(img_x.flatten())
33      # Matrix de ceros de n*m de profundidad 1
34      z = np.zeros((n * m , 1), dtype=np.uint8)
35      d = np.int(np.fix(n * m / 256)+ 0.5)
36
37      for i in range(255):
38          z[i * d:(i + 1) * d] = i * np.ones((d, 1)) # , dtype=np.uint8)
39      z[255 * d:n * m] = 255 * np.ones((n * m - 255 * d, 1)) # , dtype=np.uint8)
40
41      y[j] = z
42      y = y.reshape(n, m)
43
44      return y
45

```

Figura 15. Algoritmo de equalization

Por otro lado, la función de “image\_processing”, con los parámetros “ruta” y “clase”, desempeña un papel crucial. El primer parámetro permite elegir entre las tres rutas disponibles en la variable “Files”, mientras que el segundo parámetro se ajusta a las dos opciones en la variable “Class”, tal como se ve en la figura 13. Las primeras líneas de código de esta función involucran el módulo “os” para leer las imágenes de las tres rutas. Adicionalmente, se generan nuevas carpetas para cada clase, en las cuales

se guardan las imágenes después de ser sometidas al preprocesamiento. Esto establece una estructura ordenada y facilita el almacenamiento de las imágenes preprocesadas.

```
46 def image_processing (ruta,Clase):
47
48     #Cargar la ruta de las imagenes en la variable
49     # enlistar las imagenes de la carpeta
50     Normal_files=os.listdir(ruta + '/' + Clase )
51
52     #Crear carpeta para las imagenes procesadas
53     Name_file= ruta[0:31] + "Procesada/" + ruta[-8:] # Nombre y ruta de la carpeta
54     if not os.path.exists(Name_file ): # Si la carpeta no existe, se crea
55         os.makedirs(Name_file) # Se Crea la carpeta
56         print("Carpeta creada", Name_file) |
57
58     #Crear carpeta para las imagenes procesadas para cada clase
59     Clases = ruta[0:31] + "Procesada/" + ruta[-8:] + '/' + Clase
60     if not os.path.exists(Clases ): # Si la carpeta no existe, se crea
61         os.makedirs(Clases) # Se Crea la carpeta
62         print("Carpeta creada", Clases)
63
```

Figura 16. Código de creación de carpeta por clases

El proceso de preprocesamiento de las imágenes se realiza de manera individual, aprovechando las funciones proporcionadas por OpenCV y numpy. La secuencia comienza con la conversión de las imágenes de color de tres canales RGB (rojo,verde,azul) a imágenes en escala grises, utilizando la función “cvtColor”. Esta transformación simplifica la información y permite un procesamiento más efectivo.

Después de obtener las imágenes en escala grises, se redimensiona a un tamaño de 200 x 200 para reducir la carga computacional. Sin embargo, esta modificación puede conllevar tanto beneficios como desafíos para el rendimiento de la red neuronal convolucional. Durante el proceso de redimensionamiento, es posible que surjan ruido en las imágenes y que características importantes se pierdan. Para abordar esto, se recurre a técnicas de filtrado, un enfoque clásico en el preprocesamiento de imágenes.

Se implementa un filtro que pasa bajo de 5x5 para eliminar las altas frecuencias que se generaron durante el redimensionamiento de las imágenes de rayos X del tórax. Posteriormente, se aplica un filtro pasa alto para eliminar las componentes de baja frecuencia, lo que afina los bordes tanto internos como externos de los pulmones. Estos pasos adicionales de filtrado contribuyen con la agudización a mejorar la calidad de las imágenes y a conservar características esenciales para el diagnóstico.

```

64     contador = 1
65
66     for i in Normal_files: # Imprimir las imagenes de la imagenes
67
68         image_read= ruta + '/' + Clase + "/" + i
69         image = cv2.imread(image_read)
70
71         # Preprocesamiento a las imagenes de las carpetas
72         image = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY) # Conversion de BGR a escala gris
73         image = cv2.resize(image, (200,200), interpolation=cv2.INTER_CUBIC ) # Redimension de imagen a 200x200
74
75         # Filtro pasa bajo
76         n = 5
77         Mk = np.ones((n,n),np.float32)/n/n # Matrix 1 de n*n
78         L = cv2.filter2D(image,-1,Mk) # Ejecucion del filtro
79
80         #% Filtro pasa alto
81         Id = image.astype(float) # Convesion de imagen a floante
82         Ld = L.astype(float) # Conversion del filtro pasa bajo a flotante
83         Hd = Id-Ld # Resta de imagen con Filtro pasa bajo
84
85         H = np.abs(Hd)
86         H = H - np.min(H)
87         H = H / np.max(H) * 255
88
89         #Resalte de pixeles de pasa altos
90         Jd = Id + 3 * Hd
91         Jd[Jd<0]= 0
92         Jd = Jd / np.max(Jd) * 255
93
94         # Equalizacion
95         Z = Equalization(Jd) # Ejecucion de la funcion Equalizacion
96
97         # Guardar las imagenes procesadas en la carpeta creada
98         cv2.imwrite(Clases + "/" + Clase + '-' + str(contador) + ".PNG", Jd )
99         contador += 1
100     return

```

Figura 17. Código de preprocesamiento

Para concluir el proceso, se realiza la función de ecualización para enriquecer el contraste de las imágenes antes de ser guardadas en formato PNG. Cada imagen es almacenada en la carpeta correspondiente de la figura 13, con su respectiva clase.

### 3.10.4. Modelo de arquitectura convolucional

Con las imágenes ya preprocesadas, se avanza hacia la etapa del modelo de la arquitectura de red neuronal convolucional para el entrenamiento. Este proceso, se utiliza la librería TensorFlow, que se centra en el aprendizaje profundo, y pandas para el análisis estadístico

```
121 import pandas as pd
122 import seaborn as sns
123 import tensorflow as tf
124 from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img, img_to_array
125 from tensorflow.keras.models import Sequential, Model
126 from tensorflow.keras.layers import Conv2D, Dense, Flatten, Input, MaxPooling2D, Dropout, BatchNormalization
127 from sklearn.metrics import confusion_matrix
128 from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping
129
```

Figura 18. Librería de aprendizaje profundo

La estructura del modelo diseñado se compone de tres capas convoluciones, dos capas de agrupación máxima, una capa totalmente conectada y una capa sigmoid, con el objetivo de diagnosticar el COVID-19. La primera capa del modelo utiliza 32 filtros de tamaño 3 x 3. Luego, aplicamos la agrupación máxima de un tamaño de 3 x 3 a la salida de los filtros, lo que resulta una matriz de 50 x 50. La siguiente capa de convolución emplea 64 filtros, también de tamaño 3 x 3, seguida de otra agrupación máxima de un tamaño de 3 x 3, generado 64 matrices de tamaño 8 x 8. En la última capa de convolución, utilizamos 128 filtros de tamaño 3 x 3, seguidos, lo que produce 128 matrices de tamaño 2 x 2. Estas matrices se conectan a las capas totalmente conectadas con 262656 neuronas, sobre las cuales aplicamos la función de activación lineal ratificada.

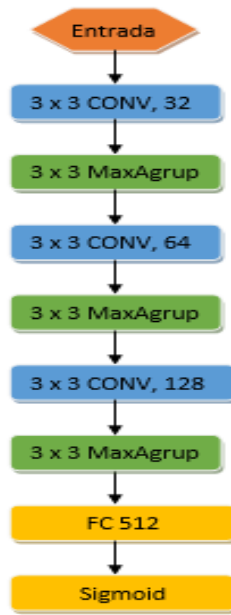


Figura 19. Arquitectura de red neuronal convolucional

La figura 19 muestra el modelo de red neuronal convolucional, mientras que la implementación en código Python se presenta en la figura 20.

```

169 os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
170 cnn=Sequential()
171 cnn.add(Conv2D(32, kernel_size=(3,3), input_shape=(img_size, img_size, 1), activation="relu", padding='same', strides=1))
172 cnn.add(MaxPooling2D(pool_size=(3, 3)))
173 cnn.add(Conv2D(64, kernel_size=(3,3), strides=2, activation="relu", padding='same'))
174 cnn.add(MaxPooling2D(pool_size=(3, 3)))
175 cnn.add(Conv2D(128, kernel_size=(3,3), strides=1, activation="relu", padding='same'))
176 cnn.add(MaxPooling2D(pool_size=(3, 3)))
177 cnn.add(Flatten())
178 cnn.add(Dense(512, activation="relu"))
179 cnn.add(Dense(1, activation='sigmoid'))
180
  
```

Figura 20. Algoritmo de arquitectura de red convolucional

Antes de iniciar el proceso de entrenamiento de la arquitectura de red neuronal convolucional, se realizan las ultimas configuraciones del optimizador y se establece una ruta para guardar los mejores pesos obtenidos por las neuronas durante el entrenamiento. Se eligió el optimizador Adam, debido a su capacidad para acelerar el proceso de entrenamiento de la red neuronal. Para el almacenamiento de los mejores pesos, se utiliza la función ModelCheckpoint de la biblioteca de TensorFlow, como se

muestra en la figura 21 Esta función permite guardar los mejores pesos obtenidos en cada época. Por último, se configura el modelo para un total de 150 épocas de entrenamiento.

```
cnn.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy']) # Optimizador adam

filepath= "Modelo clasificador_COVID-19_cnn.h5" # Nombre del modelo de neuronas entrenadas

checkpoint = ModelCheckpoint(filepath, monitor='val_loss', # Guarda el mejor modelo durante el entrenamiento
                             verbose=1, save_best_only=True,
                             save_weights_only=False)

history=cnn.fit(traindata,validation_data=valdata,epochs=150,callbacks=[checkpoint]) # Inicio del entrenamiento
```

Figura 21. Algoritmo para iniciar el entrenamiento

### 3.10.5. Evaluación del clasificador

Para evaluar el modelo después de las 150 épocas de entrenamiento, se utilizan la función de la biblioteca Scikit-learn para importar dos librerías: `confusion_matrix` (matriz de confusión) y `classification_report` (informe de clasificación). La matriz de confusión agiliza los cálculos de los cuatro métricas principales: verdadero positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN). Esto facilita la evaluación del rendimiento del modelo de manera más eficiente.

```
220 from sklearn.metrics import classification_report, confusion_matrix
221 print(classification_report(y_test,y_pred,target_names = ['Covid', 'Normal'])) # Asignado las clases al reporte de clasificación
222 plt.figure(figsize=(10,8))
223 sns.heatmap(confusion_matrix(y_test,y_pred),annot=True,
224             fmt='.3g',xticklabels=['Normal','Covid'], # configuración de ejes matriz de confusión
225             yticklabels=['Covid','Normal'], cmap='Blues')
226 plt.show()
```

Figura 22. Algoritmo de matriz de confusión

## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

#### 4.2. Resultados e interpretación

En este capítulo se presentan los resultados obtenidos en la tesis desde el preprocesamiento de imágenes hasta la validación de la arquitectura de red neuronal convolucional.

##### 4.2.1. Resultados de preprocesamiento de base de datos

En esta sección, se presentan los resultados del preprocesamiento de la base de datos imágenes de rayos x del tórax. Se han analizado datos cuantitativos que involucran la media de los valores de píxeles y la desviación estándar en distintas etapas del preprocesamiento. Estas etapas abarcan desde la imagen original hasta las técnicas de procesamiento de imagen, como se ve en la figura 23. Los resultados de este análisis se presentan en la tabla 7.

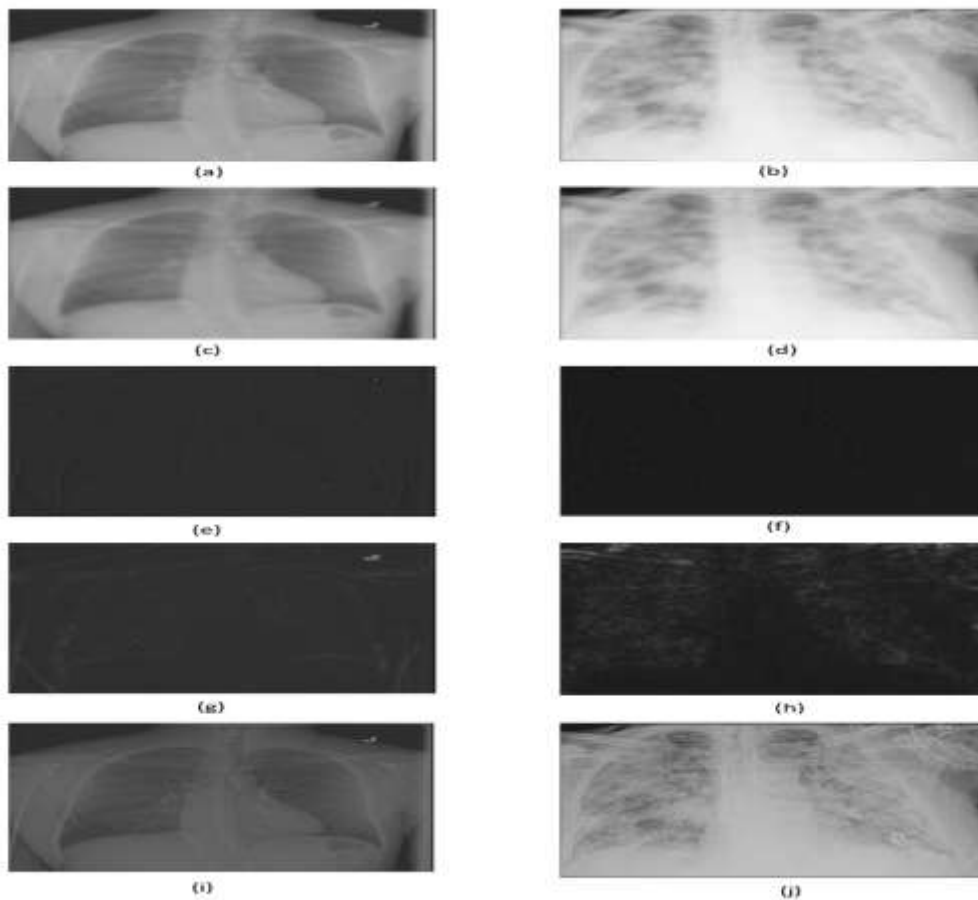


Figura 23. Muestra de imágenes de la base de datos. Las imágenes originales (a) “normal”, (b) “COVID”. Imagen filtrada pasa bajo (c) “normal”, (d) “COVID”. Imagen filtrada pasa alto (e) “normal”, (f) “COVID”. Imagen agudizada (g) “normal” (h) “COVID”. Imagen ecualizada (i) “normal”, (j) “COVID”

Tabla 7. *Comparación del promedio de intensidad media y desviación estándar entre la base de datos de imágenes originales y las imágenes preprocesadas*

	Técnicas de procesamiento de imagen	Imágenes rayos X del tórax	Intensidad media	Desviación estándar
<u>1</u>	Imagen original	Paciente normal	127.72	61.56
		Paciente COVID	143.49	54
<u>2</u>	Filtro pasa bajo	Paciente normal	127.72	60.82
		Paciente COVID	143.49	53.39
<u>3</u>	Filtro pasa alto	Paciente normal	0.004	5.34
		Paciente COVID	0.002	4.83
<u>4</u>	Agudización	Paciente normal	6.30	10.4
		Paciente COVID	7.63	12.62
<u>5</u>	Ecuilización	Paciente normal	62.42	31.57
		Paciente COVID	88.73	33.67

En la tabla 7, se observa que la imagen original de paciente normal revela una intensidad media de 127.72 y una desviación estándar de 61.560. Esto se relaciona con la presencia de zonas de pulmones sanos que exhiben tonos de gris oscuros, como se ve en la figura 23. Por otro lado, la imagen original de paciente con coronavirus exhibe una intensidad media más elevada, alcanzado 143.49, junto con una desviación estándar ligeramente menor, situada en 54. Este contraste en las métricas se debe a que las áreas pulmonares afectadas por la enfermedad presentan tonalidades de gris más claras. Esto resulta de los cambios en los tejidos pulmonares característicos de la afección coronavirus.

Al aplicar las técnicas del filtro pasa bajo a ambas imágenes originales, se observa que las intensidades medias y las desviaciones estándar permanecen relativamente estables. En el caso de la imagen del paciente sano, la intensidad media es de 127.72 y la desviación estándar es de 60.82. Para el paciente con coronavirus, la intensidad media es de 143.49 y la desviación estándar es 53.39. A pesar de aplicar la



técnica del filtro pasa bajo, estas métricas apenas varían, lo que resalta cómo esta técnica elimina los ruidos y suaviza la imagen, manteniendo las características de ambas imágenes.

Ahora, al aplicar la técnica de filtro pasa alto, se presentan resultados notables. En la imagen del paciente normal, la intensidad media es de 0.004, como una desviación estándar de 5.34. Por otro lado, en la imagen del paciente con coronavirus, la intensidad media es aún más baja, registrando 0.002, y la desviación estándar es de 4.83. Esta marca de reducción de métrica de intensidad y desviación estándar se debe a que la técnica resalta los detalles finos y los bordes en las imágenes, haciendo que las áreas de interés, como las fronteras entre los tejidos pulmonares, sean más pronunciadas y prominentes. La baja intensidad media y desviación estándar reflejan la naturaleza de los bordes.

Al agudizar, las imágenes filtradas con pase alto, se revelan aspectos aún más detallado de estas imágenes. En el caso de la imagen de paciente normal, la intensidad media se establece en 6.30, acompañado por una desviación estándar de 10.4. Por otro lado, la imagen de paciente con coronavirus presenta una intensidad media ligeramente superior de 7.63 y una desviación estándar de 12.62. Esta técnica es particularmente útil para visualizar con claridad áreas de cambios abruptos de tejido, como los bordes entre diferente estructura pulmonares.

Concluyendo con la técnica de ecualización, se descubre cómo esto influye en la mejora del contraste y visualización de detalles en las imágenes de rayos X del tórax. Además, en las imágenes de pacientes normales, la intensidad media es de 62.42, mientras que la desviación estándar es de 31.57. Por otro lado, en las imágenes de paciente con coronavirus, encontramos una intensidad más alta de 88.73 y una desviación estándar 33.67.

Tabla 8. *Pérdida de intensidad media y desviación estándar entre imágenes originales y ecualizadas*

Imágenes rayos X del tórax	Pérdida de intensidad media (%)	Pérdida de desviación estándar (%)
Paciente normal	-51.13 %	-48.71 %
Paciente coronavirus	-38.16 %	-37.65 %

En la tabla 8, se evidencia pérdidas sustanciales en la intensidad media y la desviación estándar después de haber pasado por las técnicas de procesamiento de imágenes. Para las imágenes de rayos x de paciente normal, observamos una pérdida de 51.13 % en la intensidad media y una reducción del 48.71 % en la desviación estándar. En el caso de las imágenes de rayo x de paciente coronavirus, la pérdida es del 38.18 % en la intensidad media y del 37.82 % en la desviación estándar.

Estos valores negativos indican una pérdida en términos de intensidad al aplicar la ecualización a las imágenes de pacientes normal y coronavirus. A pesar de esta disminución de las métricas, es fundamental resaltar que la que la ecualización mejora notablemente la visualización de detalles anatómicos, especialmente en los bordes de los órganos, que pueden no ser notorios en las imágenes originales. Esta mejora en la visualización es crucial para una interpretación más precisa de las áreas pulmonares y los tejidos afectados en pacientes con COVID-19 y en aquellos sin la enfermedad.

Por tanto, las imágenes ecualizadas van a ser la entrada para la red neuronal convolucional. Al mejorar las características y los detalles de la imagen, la CNN puede reconocer patrones de manera más efectiva, lo que conduce a una mayor precisión en la detección y diagnóstico del COVID-19, así como la identificación de casos normales.

#### **4.2.2. Resultados de configuración de capas convolucional y capas totalmente conectadas**

Tras configurar las capas convolucionales y completamente conectadas en la sección de “Modelo de arquitectura convolucional” del método SEMMA, se lograron resultados sobresalientes en cuanto al rendimiento a lo largo de las 150 épocas de entrenamiento. Estos resultados fueron obtenidos al emplear un conjunto de datos compuestos por 3521 imágenes de rayos X del tórax para el entrenamiento y 753 imágenes para la validación. Estas imágenes se dividieron en dos clases: “normal” y “COVID-19”.

Es fundamental resaltar que se ha adoptado una estrategia para facilitar la comprensión de los resultados obtenidos. En este sentido, se decidió presentar de manera concisa y relevante la información recopilada a lo largo de las 150 épocas de entrenamiento. Para lograr esto, se han seleccionado específicamente 19 épocas. Esta selección permite una representación más clara significativa de la evolución del rendimiento del modelo.

Tabla 9. *Resultado de entrenamiento del modelo*

<b>Entrenamiento</b>					
<b>Época</b>	<b>Tiempo</b>	<b>Loss</b>	<b>Acc</b>	<b>Val_loss</b>	<b>Val_acc</b>
<b>1</b>	317s	0.5129	0.7654	0.6778	0.6379
<b>3</b>	91s	0.3370	0.8600	0.6505	0.6804
<b>4</b>	92s	0.3075	0.8765	0.6080	0.7095
<b>5</b>	93s	0.2972	0.8807	0.5870	0.6976
<b>8</b>	83s	0.2726	0.8929	0.5654	0.7454
<b>9</b>	93s	0.2725	0.8926	0.5196	0.7745
<b>16</b>	77s	0.2386	0.9017	0.5136	0.7613
<b>19</b>	79s	0.2262	0.9108	0.5045	0.7838
<b>23</b>	81s	0.2195	0.9131	0.4470	0.8050
<b>24</b>	82s	0.2145	0.9196	0.4416	0.7931
<b>31</b>	81s	0.1996	0.9165	0.3956	0.8289
<b>74</b>	91s	0.1463	0.9367	0.3918	0.8263
<b>75</b>	84s	0.1469	0.9418	0.3802	0.8422
<b>80</b>	92s	0.1374	0.9404	0.3697	0.8607
<b>81</b>	81s	0.1501	0.9369	0.3663	0.8369
<b>85</b>	91s	0.1404	0.9441	0.3506	0.8342
<b>87</b>	86s	0.1408	0.9395	0.3417	0.8395
<b>105</b>	81s	0.1357	0.9418	0.3285	0.8541
<b>137</b>	103s	0.1115	0.9512	0.3131	0.8740

En la tabla 9, la primera época del proceso de entrenamiento tuvo una duración de 317 segundos, se registró una pérdida (Loss) de 0.5129. Esta cifra refleja la frecuencia con la que el modelo incurrió en errores al realizar predicciones en el conjunto de entrenamiento. Una exactitud (Acc) de 0.7664 indica que el modelo logró clasificar correctamente el 76.54 % de los ejemplos del mismo conjunto. En cuanto al conjunto de validación, la pérdida (Val\_loss) obtenida fue de 0.6778, lo que significa que el modelo tenía una pérdida relativamente alta en el conjunto de validación. Esto indica que el modelo no estaba generalizando bien nuevos datos y que había margen para mejorar su capacidad de predicción en ejemplos no visto previamente. Por otro lado, la exactitud (Val\_acc) fue de 0.6379, lo que significa que el modelo logró clasificar correctamente el 63.79 % de los ejemplos en el conjunto de validación.

Sin embargo, en la época 3, se observa una mejora en ambos indicadores en comparación con la primera época. La pérdida (Loss) disminuyó a 0.3370. Esto surgiere de que el modelo comenzó a ajustarse y a mejorar sus predicciones. La exactitud (Acc) aumentó notablemente a 0.86, lo que indica que el modelo logró clasificar correctamente el 86 % de los ejemplos del conjunto de entrenamiento.

En la época 16, se continúa observando mejoras en los valores. La pérdida (Loss) se redujo aún más a 0.2386 en el conjunto de entrenamiento, y la exactitud (Acc) alcanzo un valor de 0.9017, demostrado la capacidad del modelo para realizar predicciones más precisas y reducir sus errores. En el conjunto de validación, la pérdida (Val\_loss) siguió disminuyendo a 0.5136, mientras la exactitud (Val\_acc) se mantuvo en 0.7613.

Finalmente, en la mejor época, la 137, se alcanzaron los valores óptimos de rendimiento. La pérdida (Loss) se redujo significativa a 0.1115 en el conjunto de entrenamiento, lo que mostró una gran mejora en la capacidad del modelo para realizar predicciones precisas. La exactitud (Acc) alcanzó su puntuación máxima en 0.9512, destacando la alta tasa de aciertos del modelo en el conjunto de entrenamiento. En el conjunto de validación, la pérdida (Val\_loss) continuó disminuyendo a 0.3131, mientras que la exactitud (Val\_acc) aumentó a 0.8740. En las figuras 24 y 25 se presenta la evolución del entrenamiento. La figura 24 se observa la disminución gradual de las pérdidas (Loss) tanto en el conjunto de entrenamiento como el de validación. Desde un valor inicial alto en la época 1, las pérdidas disminuyen a medida que avanzan las épocas, alcanzando un mínimo en la época 137.

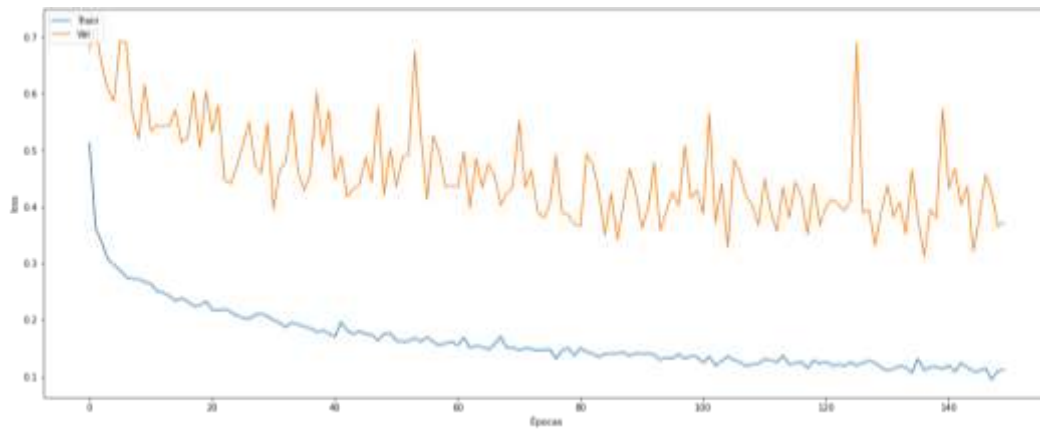


Figura 24. Evolución de pérdidas durante el proceso de entrenamiento

Por otro lado, en la figura 25, se muestra como las exactitudes (Acc) en el conjunto de entrenamiento aumentan de manera constante a lo largo de las épocas. En contraste, las exactitudes en el conjunto de validación no siguen una tendencia uniforme, aunque también experimenta un aumento general. Estas precisiones empiezan con valores bajos y alcanzan su punto máximo en la época 137.

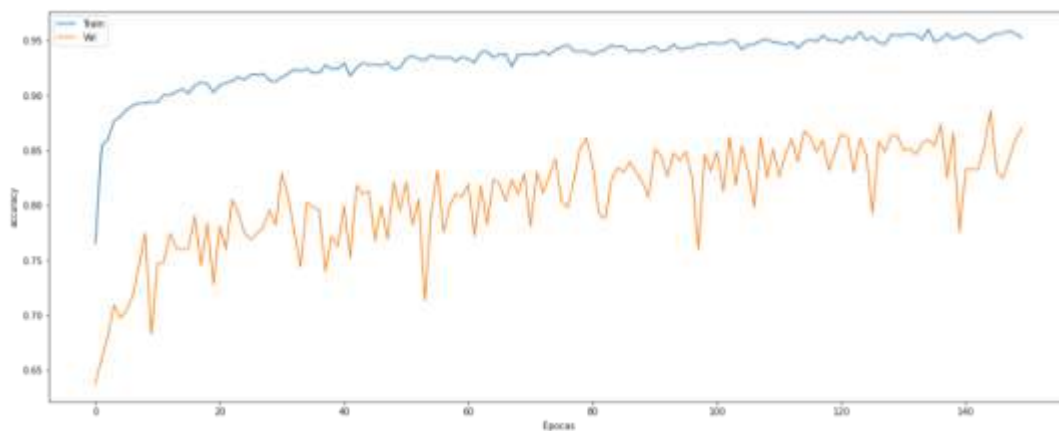


Figura 25. Evolución de exactitud durante el proceso de entrenamiento

La capacidad del modelo para clasificar entre las clases “normal” y “COVID-19” en las imágenes de rayos X del tórax ha mejorado sustancialmente. Estos resultados apoyan la efectividad de la configuración de las capas convolucionales y totalmente conectadas en el modelo.

### 4.2.3. Resultado de validación de arquitectura

Se presenta los resultados de la validación de arquitectura propuesta. Estos resultados desempeñan un papel fundamental en la evaluación del rendimiento del modelo de datos que no han sido utilizado en el proceso de entrenamiento. Los datos utilizados corresponden al conjunto de prueba, el cual estuvo conformada por un total de 754 imágenes. De estas imágenes, 383 pertenecían a la clase “normal”, mientras que los 370 restantes eran de la clase “COVID-19”.

La evaluación del rendimiento del modelo se complementa aplicando la matriz de confusión. La matriz de confusión es una herramienta valiosa para analizar como el modelo clasifica correcta o erróneamente las imágenes de las dos clases: “normal” y “COVID-19”

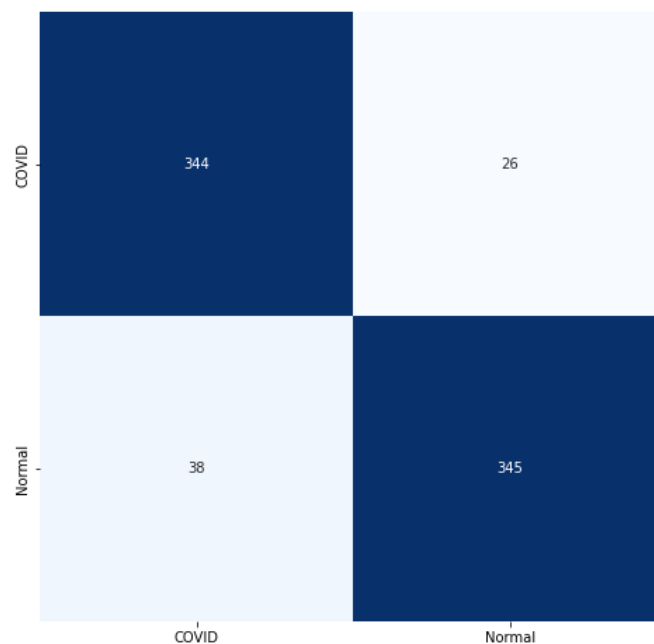


Figura 26. Matriz de confusión aplicada el conjunto pruebas

En la figura 26, se observa la matriz de confusión y los valores en la diagonal principal representan las predicciones correctas, mientras que los valores fuera de la diagonal principal representan las predicciones incorrectas. En este caso, el modelo clasificó correctamente 344 imágenes de la clase “COVID-19” y 345 imágenes de la

clase “normal”. Sin embargo, también hubo 26 imágenes de la clase “COVID-19” que fueron erróneamente clasificados como “normal” y 38 imágenes de la clase “normal” que fueron erróneamente clasificados como “COVID-19”. Estos resultados son útiles para evaluar como el modelo está realizando las predicciones en cada una de las clases.

De la matriz de confusión se extrajeron cuatro valores que son esenciales para evaluar el rendimiento del modelo de clasificación.

- Verdadero positivo (TP): 344
- Verdadero negativo (TN): 345
- Falso positivo (FP): 38
- Falso negativo (FN): 26

A partir de estos valores, se calculó varias métricas de evaluación del rendimiento del modelo.

- Exactitud

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

$$ACC = \frac{344 + 345}{344 + 345 + 38 + 26}$$

$$ACC = 0.9150$$

- Precisión

$$PR = \frac{TP}{TP + FP}$$

$$PR = \frac{344}{344 + 38}$$

$$PR = 0.9005$$



- Sensibilidad

$$RE = \frac{TP}{TP + FN}$$

$$RE = \frac{344}{344 + 26}$$

$$RE = 0.9297$$

- Especificidad

$$SP = \frac{TN}{TN + FP}$$

$$SP = \frac{345}{345 + 38}$$

$$SP = 0.9007$$

- F-score

$$F - score = \frac{2 * PR * RE}{PR + RE}$$

$$F - score = \frac{2 * 0.9005 * 0.9297}{0.9005 + 0.9297}$$

$$F - score = 0.914$$

Tabla 10. *Métricas del modelo*

	Precisión	Sensibilidad	Especificidad	F-score	Exactitud (Acc)
<b>Modelo propuesto</b>	90 %	93 %	90 %	91 %	91 %

Los valores presentados en la tabla 10 muestran el rendimiento del modelo propuesto en cinco métricas claves de evaluación. La precisión alcanzada por el modelo es del 90 %, lo que significa que alrededor del 90 % de las predicciones

positivas realizadas por el modelo fueron correctos. La sensibilidad, se sitúa en un 93 %, indicando que el modelo pudo identificar correctamente aproximadamente el 93 % de los casos positivos de la clase “COVID-19”. La especificidad, que mide la capacidad del modelo para identificar alrededor correctamente los casos negativos, alcanza un 90 %, resaltando su habilidad para identificar alrededor del 90 % de los casos “positivos” de la clase “normal”. Asimismo, el valor F-score del 91 % refleja un equilibrio entre precisión y sensibilidad, brindando una medida compuesta del rendimiento general del modelo. Por último, la exactitud del 91 % muestra que aproximadamente el 91 % de los casos, las predicciones del modelo fueron correctas, proporcionando una visión general del rendimiento global en la clasificación.

#### **4.3. Discusión de Resultados**

En esta sección, se comparan los resultados con investigaciones similares en artículos científicos sobre el tema. Sin embargo, es importante tener en cuenta que las diferencias en los conjuntos de datos utilizados, así como las diferencias lógicas utilizados en la configuración de diferentes modelos de red neuronal convolucional como el tamaño de entrada, cantidad de capas convolucionales. Además, los artículos no facilitan el algoritmo ni datos numéricos del proceso de entrenamiento.

De los resultados obtenidos en la tesis, se llega a la conclusión sobre qué diseño de la arquitectura de red neuronal convolucional contribuye el diagnóstico del COVID-19 mediante las imágenes de rayos x de tórax, esto concuerda con todos antecedentes, ya que con arquitecturas conocidas como VGGNet, DenseNet, NASNet ,GoogleNet y AlexNet o arquitecturas propias por los autores obtiene una exactitud mayor al 90 % utilizando solo imágenes médicas.

Con respecto al primer objetivo específico, el preprocesamiento a la muestra del conjunto de datos se aplican técnicas de procesamiento de imagen para mejorar

características de la imagen y que permita el desarrollo del diagnóstico de la arquitectura de red neuronal convolucional, donde se utilizó el *software* Visual Studio Code, utilizando la librería de OpenCV para obtener una mejor exactitud del 95 % en el entrenamiento y en la validación 87 % con las imágenes preprocesadas. Este resultado no se relaciona con los antecedentes de artículos por Sagar et al. [15], Heatice Catal [16] y Shanthi et al. [20] y las de tesis por Paul Calero y Xavier Chirinos [13], Juan Carlos [11], Agustín et al [21], Carlos García [10], Juan Camilo et al. [22] y Sheyla nieves [23]. Además, los autores de los artículos no aplican mucho las técnicas de procesamiento de imagen, debido que el uso de arquitectura conocidas como VGGNet, DenseNet, NASNet ,GoogleNet y Alexnet no son muy necesarias, ya que estos modelos tienen entradas definidas y solo permiten imágenes con un tamaño determinado por el modelo. Mientras que los antecedentes por Enas M.F et al. [18] y Arkja Saxena at al [17] señalan como el pre procesamiento al conjunto de datos mejoran los resultados de exactitud a un 90 %.

Con respecto al segundo y tercer objetivo específico, la tesis considera que la configuración de las capas convolucionales, capas de agrupación y capas completamente conectadas, del modelo de arquitectura propuesto logra un 95 % de exactitud en el entrenamiento y 87 % de la validación con solo 4275 radiografías del tórax. Este resultado es óptimo comparando con las tesis de Jhan Carlos [11] obtiene una exactitud del 97 % con solo 1060 imágenes médicas y con una capa menos de agrupación de nuestro modelo. Fernández[12] logra una exactitud 87 % de entrenamiento con 91 200 imágenes de cáncer de piel con la configuración cuatro capas de convolución, tres de agrupación y tres activadores ReLU y Agustín et al. [21] con una configuración de dos capas convolucionales y dos de agrupación para la arquitectura del modelo obtiene una exactitud de 73 % con solo 1000 radiografías del tórax.

La tesis tiene en cuenta en validar la arquitectura de red neuronal convolucional para el diagnóstico del COVID-19. Al evaluar se ve el desempeño de la arquitecta con la ayuda de la matriz de confusión aplicando el conjunto de prueba de 753 radiografías del tórax. Se obtuvo una exactitud del 91 %, precisión de 90 %, sensibilidad de 93 % y un F-score de 91 %. Estos Estos resultados concuerdan con las tesis de Carlos García[10] con solo 29 imágenes obtuvo 99 % precisión y 98 % de sensibilidad, Juan Camilo et al. [22] también obtiene un resultado menor de 79 % de precisión y sensibilidad del 77 % con 318 imágenes médicas puestas a prueba, Rodríguez [14] tras el entrenamiento obtiene un resultado de reconocimiento de rostros de 91.97 % y Nieves [23] que evalúa con la misma matriz con 50 tomografía del fondo óptico, obtiene resultados similares del 92 % en ambos precisión y sensibilidad pese a tener menor cantidad de imágenes.

## CONCLUSIONES

1. El diseño y desarrollo de la arquitectura de red neuronal convolucional para el diagnóstico de COVID-19 a través de imágenes de rayos X del tórax destaca por su eficacia y precisión en diferenciar entre imágenes de rayos X que muestran signos de COVID-19 y las que presentan condiciones normales. Al lograr una tasa de precisión del 90 % y una sensibilidad del 93 %, se confirma la habilidad del modelo para identificar con precisión los casos de COVID-19, brindando un nivel adicional de confianza en la capacidad de diagnóstico.
2. El preprocesamiento de imágenes de rayos X del tórax a través de técnicas procesamiento de imagen ha demostrado su capacidad para realzar los detalles anatómicos en imágenes de pacientes normales y con COVID-19. Aunque implica una pérdida del 44.65 % de intensidad durante la ecualización, visualmente se confirma que esta transformación conduce a imágenes más detalladas y resalta las características anatómicas claves, siendo beneficioso para la arquitectura de red neuronal convolucional identifique con mayor precisión patrones relevantes en el diagnóstico.
3. La configuración de las capas convolucionales y totalmente conectada ha demostrado ser fundamental para alcanzado una pérdida mínima del 11.15 % y una exactitud máxima del 95.12 % en el conjunto de entrenamiento, demostrando la eficacia de la CNN en la diferencia precisa entre casos “normal” y “COVID-19” a partir de imágenes de rayos X del tórax.

4. La validación de la arquitectura de red neuronal convolucional para el diagnóstico de COVID-19 en imágenes de rayos X, utilizando el conjunto de pruebas, Se logró una tasa de precisión del 90 % y una sensibilidad del 93 %. Estos resultados destacan la capacidad del modelo para predecir de manera precisa los casos positivos de COVID-19 y validan la efectividad de la arquitectura.

## RECOMENDACIONES

1. Para mejorar el desempeño del modelo de red neuronal, se recomienda aumentar considerablemente la cantidad de datos disponibles en la base de datos. A medida que se incorpora un mayor número de imágenes de rayos X de diferentes enfermedades, la red neuronal tendrá la oportunidad de aprender patrones más robustos y representativos de más enfermedades respiratorias. Esto permitirá que el modelo distinga con mayor precisión entre las enfermedades y mejore su capacidad de diagnóstico.
2. Establecer una colaboración estrecha con expertos médicos y radiólogos como parte fundamental del proceso. Al trabajar en conjunto con profesionales de la salud, se logrará obtener una perspectiva clínica y especializada en la interpretación de los resultados generados por el modelo. Esta colaboración no solo garantiza la solidez médica en las predicciones, sino que también permitirá incorporar conocimientos esenciales para refinar y ajustar el modelo, asegurando su relevancia y precisión en el diagnóstico de enfermedades a partir de imágenes médica.

## REFERENCIAS BIBLIOGRAFICAS

- [1] Singhal. A Review of Coronavirus Disease-2019 (COVID-19). *Indian J. Pediatr.*, 87(4), 281–286, 2020. ISSN 09737693
- [2] Benítez, C. Velasco, A. R. Sequeira, J. Henríquez, F. M. Menezes, and F. Paolucci. Responses to COVID-19 in five Latin American countries. *Heal. Policy Technol.*, 9(4), 525–559, 2020.
- [3] Gobierno del Perú. Fallecidos por COVID-19 - [Ministerio de Salud - MINSA], *Ministerio de Salud*, 2021. (30, marzo). <https://www.datosabiertos.gob.pe/dataset/fallecidos-por-coronavirus-19-ministerio-de-salud-minsa>.
- [4] Zhong, P. Gong, and G. S. Biging, Diagnostic Performance of CT and Reverse Transcriptase-Polymerase Chain Reaction for Coronavirus Disease 2019: A Meta-Analysis. *Radiology*, 78, 1–15, 2020. ISSN 1527-1315
- [5] Kucirka, L. M. Lauer, O. Laeyendecker, D. Boon, and J. Lessler, Variation in false-negative rate of reverse transcriptase polymerase chain reaction–based SARS-CoV-2 tests by time since exposure. *Ann. Intern. Med.*, 173(4), 262–268, 2020. ISSN 15393704
- [6] Gobierno del Perú, “Departamento de diagnóstico por imágenes del Hospital Loayza realizó más de 3000 controles radiológicos en pacientes con coronavirus. *Hospital Nacional Arzobispo Loayza*, (27, junio, 2020). <https://www.gob.pe/institucion/hospitalloayza/noticias/213853-departamento-de-diagnostico-por-imagenes-del-hospital-loayza-realizo-mas-de-3000-controles-radiologicos-en-pacientes-con-coronavirus>.
- [7] W. C. Serena Low *et al.* An Overview of Deep Learning Techniques on Chest X-Ray and CT Scan Identification of COVID-19 *Comput. Math. Methods Med.*, 2021, ISSN 17486718
- [8] Oprisan, A.; E. Baettig-Arriagada, C. Baeza-Delgado, and L. Martí-Bonmatí. Prevalencia y factores de desgaste profesional en radiólogos durante la pandemia COVID-19. *Radiología (64)2*, 119–127, 2022, ISSN 15781771
- [9] Caparó, F. L. and J. Alhuay-Quispe. La investigación y educación biomédica en el Perú Biomedical research and education in Peru. *Horiz Med*, 19(2), 4–5, 2019, [Online]. <http://dx.doi.org/10.24265/horizmed.2019.v19n2.01>. ISSN 22273530
- [10] García, C. Sistema Inteligente Móvil Basado En Redes Neuronales Convolucionales para el Diagnóstico Previo De Enfermedades Renales. Tesis (Doctor en Ingeniería de Sistemas). Universidad Nacional Federico Villarreal 2020.
- [11] Caya, J. C. Evaluación De Modelos De Redes Neuronales Convolucionales Aplicado a Radiografías De Tórax, Para Apoyar Al Proceso De Diagnóstico De Neumonía Asociada Al Coronavirus-19. Tesis (título de Ingeniero Electrónico). Universidad Ricardo Palma, 2020.
- [12] Baca, R. F. Redes convolucionales para la determinación de cáncer de piel a



- partir de imágenes digital. Tesis (título de Ingeniero Informático) Universidad Católica San Pablo, 2019.
- [13] Manuel, C. S. P. and C. C. Xavier. Detección del uso correcto de mascarillas utilizando una red neuronal convolucional para el ingreso de personas a un laboratorio de una universidad Tesis (título de Ingeniero Electrónico) en Universidad Ricardo Palma, 2021.
- [14] Wanderley, B. C. R. Antonio, and S. C. D. Iván. Diseño de una red neuronal convolucional para el reconocimiento facial,” Tesis (título de Ingeniero Mecatrónico) Universidad Nacional de Piura, 2022.
- [15] Deb, R. K. Jha, K. Jha, and P. S. Tripathi, “A multi model ensemble based deep convolution neural network structure for detection of COVID-19,” *Biomed. Signal Process. Control*, 71, 2022, doi: 10.1016/j.bspc.2021.103126. 17468108
- [16] Reis, H. C. COVID-19 diagnosis with deep learning. *Ing. e Investig.* 42(1), 1–8, 2022, doi: 10.15446/ing.investig.v42n1.88825. ISSN 22488723
- [17] Saxena, A. Vyas, L. Parashar, and U. Singh., A Glaucoma Detection using Convolutional Neural Network. *Proc. Int. Conf. Electron. Sustain. Commun. Syst. ICESC 2020*, 815–820, 2020, doi: 10.1109/ICESC48915.2020.9155930.
- [18] El Houby, E. M. F. and N. I. R. Yassin, “Malignant and nonmalignant classification of breast lesions in mammograms using convolutional neural networks,” *Biomed. Signal Process. Control.* 70, 2021, doi: 10.1016/j.bspc.2021.102954. ISSN 17468108
- [19] Rahim, S. A. Hassan, and S. Y. Shin, “A deep convolutional neural network for the detection of polyps in colonoscopy images,” *Biomed. Signal Process. Control.* 68, 2021, doi: 10.1016/j.bspc.2021.102654. ISSN 17468108
- [20] Shanthi, R. S. Sabeenian, and R. Anand, “Automatic diagnosis of skin diseases using convolution neural network,” *Microprocess. Microsyst.*,76, 2020, doi: 10.1016/j.micpro.2020.103074. ISSN 01419331
- [21] Naar and F. Barreto. *Modelo de red neuronal convolucional para el diagnóstico de neumonía en imágenes radiológicas.* 2019. Tesis (título de Ingeniero de Sistemas.
- [22] Cardona and F. N. Fernández. Modelo de machine learning para clasificación de pacientes con glaucoma en la población del Valle del Cauca. 2022, [Online]. Tesis para obtener el título de Magíster en Ciencia de datos en Universidad ICESI. [https://repository.icesi.edu.co/biblioteca\\_digital/bitstream/10906/95207/1/T02445.pdf](https://repository.icesi.edu.co/biblioteca_digital/bitstream/10906/95207/1/T02445.pdf)
- [23] Nieves, S. Identificación automática de glaucoma a partir de imágenes oct circumpapilares mediante el uso de redes neuronales convolucionales,” tesis para obtener el título de graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación en Universidad Politecnica de Valencia, 2019.
- [24] Huang and T. H. Le, *Principles and Labs for Deep Learning.* London: Academic Press, 2021. ISBN 9780323901987

- [25] Aggarwal, C. C. *Neural Networks and Deep Learning: A Textbook*. Springer, 2018. ISBN 9783319944623
- [26] Kaliraj, P. and T. Devi, *Artificial Intelligence Theory, Models, and Applications*. 2021. ISBN 9781032008097
- [27] Kant Singh, El Hoseny, M, Singh, *Machine Learning and the Internet of Medical Things in Healthcare*. 2021. ISBN 9780128212295
- [28] Rajinikanth, E. Priya, H. Lin, and F. Lin, *Hybrid Image Processing Methods for Medical Image Examination*. CRC Press, 2021. ISBN 9780367534967
- [29] Elgendy, *Deep Learning for Vision Systems*, no. October. Shelter Island, NY: Manning Publications, 2020. ISBN 9781617296192
- [30] S. Tanwar, A. Nayyar, and R. Rameshwar, *Machine Learning in Signal Processing*, vol. 1999, no. December. 2021. ISBN 9780367618902
- [31] Mery and C. Pieringer, *Computer Vision for X-Ray Testing: Imaging, Systems, Image Databases, and Algorithms*, 2nd Editio. 2021. ISBN 9783030567699
- [32] Alan Japp, C. R. *Macleod' s Clinical Diagnosis*, 2nd ed. Elsevier, 2018. ISBN 9780702069635
- [33] Newell, J. D. *Developing the Digital Lung: From First Lung CT to Clinical AI*, 1st ed. Elsevier, 2022. ISBN 9780323795012
- [34] Mirtaheri S. L. and R. Shahbazian, *Machine Learning Theory to Applications*. 2022. ISBN 9780367634537
- [35] Prabha, P. Karthikeyan, K. Kamalanand, and N. Selvaganesan, *Computational Modelling and Imaging for SARS-CoV-2 and COVID-19*. 2021. ISBN 780367695293
- [36] Alpaydin, E. *Introduction to machine learning*, 2010. ISBN 9780262012430
- [37] Christopher, A. D., *Chest X-Rays for Medical Students: CXRs Made Easy*, 2nd ed. Wiley-Blackwell, 2020. ISBN 9781119504054
- [38] Ñaupas, E. Mejía, E. Novo, and A. Villagómez, *Metodología de la Investigación. Cuantitativa - Cualitativa y Redacción de la Tesis*, 4ta ed. 2014. ISBN 9789587628777
- [39] Espinoza, C. A. *Metodología de investigación tecnológica*. 2014. ISBN 9786120016671
- [40] European Society of Radiology, "Chest Database." (12, enero, 2023) [https://www.eurorad.org/advanced-search?filter %5B0 %5D=section %3A40](https://www.eurorad.org/advanced-search?filter%5B0%5D=section%3A40).
- [41] Itali and Society of Medical and Interventional Radiology, "COVID-19 Database." <https://sirm.org/coronavirus-19/>.
- [42] Medical Imaging Databank of the Valencia Region, "BIMCV-COVID-19." <https://bimcv.cipf.es/bimcv-projects/bimcv-COVID-19/>.

- [43] R. Summers, R. “The National Institutes of Health.”  
<https://medpix.nlm.nih.gov/home>.
- [44] VIA/I-ELCAP, (15, enero, 2023) “Vision and Image Analysis Group.”  
<http://www.via.cornell.edu/lungdb.html>.
- [45] Stanford, “Center for Artificial Intelligence in Medicine & Imaging.”  
<https://stanfordaimi.azurewebsites.net/datasets/23c56a0d-15de-405b-87c8-99c30138950c>.
- [46] Sharda, D. Delen, and E. Turban, *Business intelligence and analytics: systems for decision support*, 10th edit. Pearson, 2015. ISBN 9780133050905

## ANEXOS

## ANEXO 1

### Matriz de confusión

Planteamiento de problema	Marco Teórico	Metodología
Formulación del problema general	Antecedentes	Variables
<b>¿Cómo diseñar una arquitectura de red neuronal convolucional para el diagnóstico de COVID-19 mediante imágenes de rayos X?</b>	"A multi model ensemble based deep convolution neuronal network structure for detection of COVID-19" por S.D Deb, R. K, Jha, K. Jha y P. S Tripathi (2021)[15]	<b>Variable independiente</b> Arquitectura de red neuronal convolucional  <b>Variable dependiente</b> Diagnostico  <b>Tipo de investigación</b> Investigación tecnológica  <b>Nivel de investigación</b> Nivel de investigación aplicada  <b>Diseño de investigación</b> Diseño transversal
	"COVID-19 diagnosis with deep learning" por H. C. Reis (2022)[16]	
	"A Glaucoma Detection using Convolutional Neural Network" por A. Saxena, A. Vyas, L. Parashar y U. Singh (2020)[17]	
	"Malignant and nonmalignant classification of breast lesions in mammograms using convolutional neural networks" por E. M. F. El Houby y N. I. R. Yassin (2021)[18]	
<b>Formulación del problema específicos</b>	"A deep convolutional neural network for the detection of polyps in colonoscopy images" por T. Rahim, S. A. Hassan y S. Y. Shin (2021)[19]	
<b>¿Cómo preprocesar las imágenes de rayos X para el desarrollo del diagnóstico de la arquitectura de red neuronal convolucional?</b>	"Automatic diagnosis of skin diseases using convolution neural network" por T. Shanthi, R. S. Sabeenian y R. Anand (2020)[20]	
	"Evaluación De Modelos De Redes Neuronales Convolucionales Aplicado a Radiografías De Tórax, Para Apoyar Al Proceso De Diagnóstico De Neumonía Asociada Al Coronavirus-19" por J. C. Caya Pérez (2020)[11]	
<b>¿Cómo configurar las capas convolucionales para lograr el diagnóstico de COVID-19 mediante imágenes de rayos X?</b>	"Redes convolucionales para la determinación de cáncer de piel a partir de imágenes digitales" por R. F. Baca Santa Cruz (2019)[12]	
	"Modelo de red neuronal convolucional para el diagnóstico de neumonía en imágenes radiológicas" por A. Naar Perez y F. Barreto Martínez (2019)[21]	
<b>¿Cómo configurar las capas totalmente conectadas para lograr el diagnóstico de COVID-19 mediante imágenes de rayos X?</b>	"Sistema Inteligente Móvil Basado En Redes Neuronales Convolucionales para el Diagnóstico Previo De Enfermedades Renales" por C. García (2020)[10]	
	"Modelo de machine learning para clasificación de pacientes con glaucoma en la población del Valle del Cauca" por J. C. Cardona y F. N. Fernández (2022)[22]	
<b>¿Cómo validar la arquitectura de red neuronal convolucional para el diagnóstico de COVID-19?</b>		

Objetivo generar	Marco teórico	Población y Muestra
Diseñar una arquitectura de red neuronal convolucional para el diagnóstico de COVID-19 mediante imágenes de rayos X.	<b>COVID-19:</b> Es una nueva enfermedad de la familia de coronavirus y causa unas infecciones agudas al sistema respiratorio por el virus SARS CoV2.	<b>Población</b>
	<b>Maching learning:</b> El aprendizaje es proceso computacional que utiliza algoritmos y técnicas para solucionar problemas complejos de forma automatizada utilizando base de datos.	Bases de datos
<b>Objetivo específico</b>		<b>Muestra</b>
Preprocesar las imágenes de rayos X para el desarrollo del diagnóstico de la arquitectura de red neuronal convolucional.	<b>Redes convolucionales:</b> Son técnicas del maching learning que fueron diseñados para trabajar mayormente con imágenes bidimensionales, debido que presenta entradas de estructura cuadrangular.	Medical Imaging Databank of the Valencia Region (BIMCV)
Configurar las capas convolucionales para lograr el diagnóstico de COVID-19 mediante imágenes de rayos X.	<b>Convolución:</b> Es una operación matemática que se aplica en dos funciones para obtener una tercera función, la convolución en imágenes es la operación de dos funciones matriciales para la extracción de características y generar una imagen de salida.	<b>Técnica</b>
		Observación
Configurar las capas totalmente conectadas para lograr el diagnóstico de COVID-19 mediante imágenes de rayos X.	<b>Agrupación:</b> Es la encargada de reducir las dimensiones de las imágenes generadas por las convoluciones de la capa anterior, conservando las características importantes y reduciendo el costo computacional.	<b>Instrumento</b>
		Ficha de observación
Validar la arquitectura de red neuronal convolucional para el diagnóstico de COVID-19.		

## ANEXO 2

Resultado de entrenamiento de 150 épocas de la red neuronal convolucional en Python.

```
Epoch 1/150
353/353 [=====] - ETA: 0s - loss: 0.5129 - accuracy: 0.7654
Epoch 1: val_loss improved from inf to 0.67775, saving model to Modelo_Clasificador_COVID-19_cnn.h5
353/353 [=====] - 317s 884ms/step - loss: 0.5129 - accuracy: 0.7654 - val_loss: 0.6778 - val_accuracy: 0.6379
Epoch 2/150
353/353 [=====] - ETA: 0s - loss: 0.3619 - accuracy: 0.8543
Epoch 2: val_loss did not improve from 0.67775
353/353 [=====] - 85s 240ms/step - loss: 0.3619 - accuracy: 0.8543 - val_loss: 0.7190 - val_accuracy: 0.6605
Epoch 3/150
353/353 [=====] - ETA: 0s - loss: 0.3370 - accuracy: 0.8600
Epoch 3: val_loss improved from 0.67775 to 0.65046, saving model to Modelo_Clasificador_COVID-19_cnn.h5
353/353 [=====] - 91s 257ms/step - loss: 0.3370 - accuracy: 0.8600 - val_loss: 0.6505 - val_accuracy: 0.6804
Epoch 4/150
353/353 [=====] - ETA: 0s - loss: 0.3075 - accuracy: 0.8765
Epoch 4: val_loss improved from 0.65046 to 0.60804, saving model to Modelo_Clasificador_COVID-19_cnn.h5
353/353 [=====] - 92s 261ms/step - loss: 0.3075 - accuracy: 0.8765 - val_loss: 0.6080 - val_accuracy: 0.7095
Epoch 5/150
353/353 [=====] - ETA: 0s - loss: 0.2972 - accuracy: 0.8807
Epoch 5: val_loss improved from 0.60804 to 0.58703, saving model to Modelo_Clasificador_COVID-19_cnn.h5
353/353 [=====] - 93s 263ms/step - loss: 0.2972 - accuracy: 0.8807 - val_loss: 0.5870 - val_accuracy: 0.6976
Epoch 6/150
353/353 [=====] - ETA: 0s - loss: 0.2878 - accuracy: 0.8870
Epoch 6: val_loss did not improve from 0.58703
353/353 [=====] - 84s 237ms/step - loss: 0.2878 - accuracy: 0.8870 - val_loss: 0.6933 - val_accuracy: 0.7042
Epoch 7/150
353/353 [=====] - ETA: 0s - loss: 0.2760 - accuracy: 0.8907
Epoch 7: val_loss did not improve from 0.58703
353/353 [=====] - 78s 221ms/step - loss: 0.2760 - accuracy: 0.8907 - val_loss: 0.6895 - val_accuracy: 0.7175
Epoch 8/150
353/353 [=====] - ETA: 0s - loss: 0.2726 - accuracy: 0.8929
Epoch 8: val_loss improved from 0.58703 to 0.56542, saving model to Modelo_Clasificador_COVID-19_cnn.h5
353/353 [=====] - 83s 234ms/step - loss: 0.2726 - accuracy: 0.8929 - val_loss: 0.5654 - val_accuracy: 0.7454
```

Epoch 9/150  
353/353 [=====] - ETA: 0s - loss: 0.2725 - accuracy: 0.8926  
Epoch 9: val\_loss improved from 0.56542 to 0.51964, saving model to Modelo\_Classificador\_COVID-19\_cnn.h5  
353/353 [=====] - 93s 263ms/step - loss: 0.2725 - accuracy: 0.8926 - val\_loss: 0.5196 - val\_accuracy: 0.7745  
Epoch 10/150  
353/353 [=====] - ETA: 0s - loss: 0.2671 - accuracy: 0.8941  
Epoch 10: val\_loss did not improve from 0.51964  
353/353 [=====] - 89s 251ms/step - loss: 0.2671 - accuracy: 0.8941 - val\_loss: 0.6161 - val\_accuracy: 0.6830  
Epoch 11/150  
353/353 [=====] - ETA: 0s - loss: 0.2641 - accuracy: 0.8941  
Epoch 11: val\_loss did not improve from 0.51964  
353/353 [=====] - 76s 214ms/step - loss: 0.2641 - accuracy: 0.8941 - val\_loss: 0.5341 - val\_accuracy: 0.7467  
Epoch 12/150  
353/353 [=====] - ETA: 0s - loss: 0.2509 - accuracy: 0.9003  
Epoch 12: val\_loss did not improve from 0.51964  
353/353 [=====] - 78s 222ms/step - loss: 0.2509 - accuracy: 0.9003 - val\_loss: 0.5445 - val\_accuracy: 0.7480  
Epoch 13/150  
353/353 [=====] - ETA: 0s - loss: 0.2481 - accuracy: 0.9003  
Epoch 13: val\_loss did not improve from 0.51964  
353/353 [=====] - 82s 233ms/step - loss: 0.2481 - accuracy: 0.9003 - val\_loss: 0.5414 - val\_accuracy: 0.7745  
Epoch 14/150  
353/353 [=====] - ETA: 0s - loss: 0.2426 - accuracy: 0.9032  
Epoch 14: val\_loss did not improve from 0.51964  
353/353 [=====] - 81s 231ms/step - loss: 0.2426 - accuracy: 0.9032 - val\_loss: 0.5433 - val\_accuracy: 0.7613  
Epoch 15/150  
353/353 [=====] - ETA: 0s - loss: 0.2335 - accuracy: 0.9060  
Epoch 15: val\_loss did not improve from 0.51964  
353/353 [=====] - 84s 237ms/step - loss: 0.2335 - accuracy: 0.9060 - val\_loss: 0.5722 - val\_accuracy: 0.7599  
Epoch 16/150  
353/353 [=====] - ETA: 0s - loss: 0.2386 - accuracy: 0.9017  
Epoch 16: val\_loss improved from 0.51964 to 0.51358, saving model to Modelo\_Classificador\_COVID-19\_cnn.h5  
353/353 [=====] - 77s 218ms/step - loss: 0.2386 - accuracy: 0.9017 - val\_loss: 0.5136 - val\_accuracy: 0.7613  
Epoch 17/150  
353/353 [=====] - ETA: 0s - loss: 0.2323 - accuracy: 0.9091  
Epoch 17: val\_loss did not improve from 0.51358



353/353 [=====] - 78s 220ms/step - loss: 0.2323 - accuracy: 0.9091 - val\_loss: 0.5223 - val\_accuracy: 0.7905  
Epoch 18/150  
353/353 [=====] - ETA: 0s - loss: 0.2243 - accuracy: 0.9117  
Epoch 18: val\_loss did not improve from 0.51358  
353/353 [=====] - 79s 224ms/step - loss: 0.2243 - accuracy: 0.9117 - val\_loss: 0.6033 - val\_accuracy: 0.7454  
Epoch 19/150  
353/353 [=====] - ETA: 0s - loss: 0.2262 - accuracy: 0.9108  
Epoch 19: val\_loss improved from 0.51358 to 0.50454, saving model to Modelo\_Clasificador\_COVID-19\_cnn.h5  
353/353 [=====] - 79s 224ms/step - loss: 0.2262 - accuracy: 0.9108 - val\_loss: 0.5045 - val\_accuracy: 0.7838  
Epoch 20/150  
353/353 [=====] - ETA: 0s - loss: 0.2327 - accuracy: 0.9026  
Epoch 20: val\_loss did not improve from 0.50454  
353/353 [=====] - 83s 236ms/step - loss: 0.2327 - accuracy: 0.9026 - val\_loss: 0.6045 - val\_accuracy: 0.7281  
Epoch 21/150  
353/353 [=====] - ETA: 0s - loss: 0.2183 - accuracy: 0.9094  
Epoch 21: val\_loss did not improve from 0.50454  
353/353 [=====] - 79s 222ms/step - loss: 0.2183 - accuracy: 0.9094 - val\_loss: 0.5318 - val\_accuracy: 0.7812  
Epoch 22/150  
353/353 [=====] - ETA: 0s - loss: 0.2171 - accuracy: 0.9114  
Epoch 22: val\_loss did not improve from 0.50454  
353/353 [=====] - 88s 248ms/step - loss: 0.2171 - accuracy: 0.9114 - val\_loss: 0.5801 - val\_accuracy: 0.7599  
Epoch 23/150  
353/353 [=====] - ETA: 0s - loss: 0.2195 - accuracy: 0.9131  
Epoch 23: val\_loss improved from 0.50454 to 0.44703, saving model to Modelo\_Clasificador\_COVID-19\_cnn.h5  
353/353 [=====] - 81s 229ms/step - loss: 0.2195 - accuracy: 0.9131 - val\_loss: 0.4470 - val\_accuracy: 0.8050  
Epoch 24/150  
353/353 [=====] - ETA: 0s - loss: 0.2145 - accuracy: 0.9165  
Epoch 24: val\_loss improved from 0.44703 to 0.44157, saving model to Modelo\_Clasificador\_COVID-19\_cnn.h5  
353/353 [=====] - 82s 233ms/step - loss: 0.2145 - accuracy: 0.9165 - val\_loss: 0.4416 - val\_accuracy: 0.7931  
Epoch 25/150  
352/353 [=====>] - ETA: 0s - loss: 0.2078 - accuracy: 0.9142  
Epoch 25: val\_loss did not improve from 0.44157  
353/353 [=====] - 85s 240ms/step - loss: 0.2077 - accuracy: 0.9142 - val\_loss: 0.4705 - val\_accuracy: 0.7745

Epoch 26/150  
353/353 [=====] - ETA: 0s - loss: 0.2042 - accuracy: 0.9188  
Epoch 26: val\_loss did not improve from 0.44157  
353/353 [=====] - 88s 248ms/step - loss: 0.2042 - accuracy: 0.9188 - val\_loss: 0.5103 - val\_accuracy: 0.7692  
Epoch 27/150  
353/353 [=====] - ETA: 0s - loss: 0.2012 - accuracy: 0.9191  
Epoch 27: val\_loss did not improve from 0.44157  
353/353 [=====] - 87s 246ms/step - loss: 0.2012 - accuracy: 0.9191 - val\_loss: 0.5500 - val\_accuracy: 0.7745  
Epoch 28/150  
353/353 [=====] - ETA: 0s - loss: 0.2091 - accuracy: 0.9196  
Epoch 28: val\_loss did not improve from 0.44157  
353/353 [=====] - 98s 277ms/step - loss: 0.2091 - accuracy: 0.9196 - val\_loss: 0.4748 - val\_accuracy: 0.7798  
Epoch 29/150  
353/353 [=====] - ETA: 0s - loss: 0.2112 - accuracy: 0.9134  
Epoch 29: val\_loss did not improve from 0.44157  
353/353 [=====] - 88s 249ms/step - loss: 0.2112 - accuracy: 0.9134 - val\_loss: 0.4588 - val\_accuracy: 0.7958  
Epoch 30/150  
353/353 [=====] - ETA: 0s - loss: 0.2068 - accuracy: 0.9125  
Epoch 30: val\_loss did not improve from 0.44157  
353/353 [=====] - 84s 239ms/step - loss: 0.2068 - accuracy: 0.9125 - val\_loss: 0.5492 - val\_accuracy: 0.7825  
Epoch 31/150  
353/353 [=====] - ETA: 0s - loss: 0.1996 - accuracy: 0.9165  
Epoch 31: val\_loss improved from 0.44157 to 0.39560, saving model to Modelo\_Clasificador\_COVID-19\_cnn.h5  
353/353 [=====] - 81s 229ms/step - loss: 0.1996 - accuracy: 0.9165 - val\_loss: 0.3956 - val\_accuracy: 0.8289  
Epoch 32/150  
353/353 [=====] - ETA: 0s - loss: 0.1955 - accuracy: 0.9196  
Epoch 32: val\_loss did not improve from 0.39560  
353/353 [=====] - 81s 229ms/step - loss: 0.1955 - accuracy: 0.9196 - val\_loss: 0.4653 - val\_accuracy: 0.8077  
Epoch 33/150  
353/353 [=====] - ETA: 0s - loss: 0.1872 - accuracy: 0.9236  
Epoch 33: val\_loss did not improve from 0.39560  
353/353 [=====] - 92s 262ms/step - loss: 0.1872 - accuracy: 0.9236 - val\_loss: 0.4801 - val\_accuracy: 0.7759  
Epoch 34/150  
353/353 [=====] - ETA: 0s - loss: 0.1958 - accuracy: 0.9219  
Epoch 34: val\_loss did not improve from 0.39560

353/353 [=====] - 79s 223ms/step - loss: 0.1958 - accuracy: 0.9219 - val\_loss: 0.5701 - val\_accuracy: 0.7440  
Epoch 35/150  
353/353 [=====] - ETA: 0s - loss: 0.1917 - accuracy: 0.9245  
Epoch 35: val\_loss did not improve from 0.39560  
353/353 [=====] - 77s 217ms/step - loss: 0.1917 - accuracy: 0.9245 - val\_loss: 0.4608 - val\_accuracy: 0.8024  
Epoch 36/150  
353/353 [=====] - ETA: 0s - loss: 0.1887 - accuracy: 0.9205  
Epoch 36: val\_loss did not improve from 0.39560  
353/353 [=====] - 84s 239ms/step - loss: 0.1887 - accuracy: 0.9205 - val\_loss: 0.4292 - val\_accuracy: 0.7984  
Epoch 37/150  
353/353 [=====] - ETA: 0s - loss: 0.1854 - accuracy: 0.9202  
Epoch 37: val\_loss did not improve from 0.39560  
353/353 [=====] - 81s 229ms/step - loss: 0.1854 - accuracy: 0.9202 - val\_loss: 0.4580 - val\_accuracy: 0.7958  
Epoch 38/150  
353/353 [=====] - ETA: 0s - loss: 0.1788 - accuracy: 0.9276  
Epoch 38: val\_loss did not improve from 0.39560  
353/353 [=====] - 92s 260ms/step - loss: 0.1788 - accuracy: 0.9276 - val\_loss: 0.6020 - val\_accuracy: 0.7401  
Epoch 39/150  
353/353 [=====] - ETA: 0s - loss: 0.1815 - accuracy: 0.9242  
Epoch 39: val\_loss did not improve from 0.39560  
353/353 [=====] - 87s 247ms/step - loss: 0.1815 - accuracy: 0.9242 - val\_loss: 0.5045 - val\_accuracy: 0.7719  
Epoch 40/150  
353/353 [=====] - ETA: 0s - loss: 0.1760 - accuracy: 0.9242  
Epoch 40: val\_loss did not improve from 0.39560  
353/353 [=====] - 80s 225ms/step - loss: 0.1760 - accuracy: 0.9242 - val\_loss: 0.5708 - val\_accuracy: 0.7626  
Epoch 41/150  
353/353 [=====] - ETA: 0s - loss: 0.1707 - accuracy: 0.9296  
Epoch 41: val\_loss did not improve from 0.39560  
353/353 [=====] - 87s 246ms/step - loss: 0.1707 - accuracy: 0.9296 - val\_loss: 0.4482 - val\_accuracy: 0.7997  
Epoch 42/150  
353/353 [=====] - ETA: 0s - loss: 0.1961 - accuracy: 0.9179  
Epoch 42: val\_loss did not improve from 0.39560  
353/353 [=====] - 84s 237ms/step - loss: 0.1961 - accuracy: 0.9179 - val\_loss: 0.4882 - val\_accuracy: 0.7520  
Epoch 43/150  
353/353 [=====] - ETA: 0s - loss: 0.1808 - accuracy: 0.9250  
Epoch 43: val\_loss did not improve from 0.39560

353/353 [=====] - 82s 233ms/step - loss: 0.1808 - accuracy: 0.9250 - val\_loss: 0.4174 - val\_accuracy: 0.8183  
Epoch 44/150  
353/353 [=====] - ETA: 0s - loss: 0.1751 - accuracy: 0.9296  
Epoch 44: val\_loss did not improve from 0.39560  
353/353 [=====] - 90s 254ms/step - loss: 0.1751 - accuracy: 0.9296 - val\_loss: 0.4319 - val\_accuracy: 0.8103  
Epoch 45/150  
353/353 [=====] - ETA: 0s - loss: 0.1804 - accuracy: 0.9276  
Epoch 45: val\_loss did not improve from 0.39560  
353/353 [=====] - 85s 241ms/step - loss: 0.1804 - accuracy: 0.9276 - val\_loss: 0.4394 - val\_accuracy: 0.8130  
Epoch 46/150  
353/353 [=====] - ETA: 0s - loss: 0.1750 - accuracy: 0.9281  
Epoch 46: val\_loss did not improve from 0.39560  
353/353 [=====] - 82s 233ms/step - loss: 0.1750 - accuracy: 0.9281 - val\_loss: 0.4873 - val\_accuracy: 0.7679  
Epoch 47/150  
353/353 [=====] - ETA: 0s - loss: 0.1739 - accuracy: 0.9270  
Epoch 47: val\_loss did not improve from 0.39560  
353/353 [=====] - 101s 285ms/step - loss: 0.1739 - accuracy: 0.9270 - val\_loss: 0.4440 - val\_accuracy: 0.7997  
Epoch 48/150  
353/353 [=====] - ETA: 0s - loss: 0.1638 - accuracy: 0.9301  
Epoch 48: val\_loss did not improve from 0.39560  
353/353 [=====] - 89s 251ms/step - loss: 0.1638 - accuracy: 0.9301 - val\_loss: 0.5751 - val\_accuracy: 0.7692  
Epoch 49/150  
353/353 [=====] - ETA: 0s - loss: 0.1757 - accuracy: 0.9239  
Epoch 49: val\_loss did not improve from 0.39560  
353/353 [=====] - 83s 234ms/step - loss: 0.1757 - accuracy: 0.9239 - val\_loss: 0.4215 - val\_accuracy: 0.8210  
Epoch 50/150  
353/353 [=====] - ETA: 0s - loss: 0.1765 - accuracy: 0.9250  
Epoch 50: val\_loss did not improve from 0.39560  
353/353 [=====] - 81s 229ms/step - loss: 0.1765 - accuracy: 0.9250 - val\_loss: 0.5023 - val\_accuracy: 0.7958  
Epoch 51/150  
353/353 [=====] - ETA: 0s - loss: 0.1633 - accuracy: 0.9338  
Epoch 51: val\_loss did not improve from 0.39560  
353/353 [=====] - 84s 238ms/step - loss: 0.1633 - accuracy: 0.9338 - val\_loss: 0.4356 - val\_accuracy: 0.8210  
Epoch 52/150  
353/353 [=====] - ETA: 0s - loss: 0.1615 - accuracy: 0.9358  
Epoch 52: val\_loss did not improve from 0.39560

353/353 [=====] - 78s 222ms/step - loss: 0.1615 - accuracy: 0.9358 - val\_loss: 0.4876 - val\_accuracy: 0.7825  
Epoch 53/150  
353/353 [=====] - ETA: 0s - loss: 0.1625 - accuracy: 0.9335  
Epoch 53: val\_loss did not improve from 0.39560  
353/353 [=====] - 86s 245ms/step - loss: 0.1625 - accuracy: 0.9335 - val\_loss: 0.4928 - val\_accuracy: 0.8050  
Epoch 54/150  
353/353 [=====] - ETA: 0s - loss: 0.1685 - accuracy: 0.9318  
Epoch 54: val\_loss did not improve from 0.39560  
353/353 [=====] - 82s 232ms/step - loss: 0.1685 - accuracy: 0.9318 - val\_loss: 0.6745 - val\_accuracy: 0.7135  
Epoch 55/150  
353/353 [=====] - ETA: 0s - loss: 0.1613 - accuracy: 0.9367  
Epoch 55: val\_loss did not improve from 0.39560  
353/353 [=====] - 83s 234ms/step - loss: 0.1613 - accuracy: 0.9367 - val\_loss: 0.5252 - val\_accuracy: 0.7918  
Epoch 56/150  
353/353 [=====] - ETA: 0s - loss: 0.1697 - accuracy: 0.9341  
Epoch 56: val\_loss did not improve from 0.39560  
353/353 [=====] - 80s 228ms/step - loss: 0.1697 - accuracy: 0.9341 - val\_loss: 0.4129 - val\_accuracy: 0.8316  
Epoch 57/150  
353/353 [=====] - ETA: 0s - loss: 0.1612 - accuracy: 0.9344  
Epoch 57: val\_loss did not improve from 0.39560  
353/353 [=====] - 79s 224ms/step - loss: 0.1612 - accuracy: 0.9344 - val\_loss: 0.5261 - val\_accuracy: 0.7759  
Epoch 58/150  
353/353 [=====] - ETA: 0s - loss: 0.1552 - accuracy: 0.9347  
Epoch 58: val\_loss did not improve from 0.39560  
353/353 [=====] - 93s 263ms/step - loss: 0.1552 - accuracy: 0.9347 - val\_loss: 0.4951 - val\_accuracy: 0.7997  
Epoch 59/150  
353/353 [=====] - ETA: 0s - loss: 0.1584 - accuracy: 0.9313  
Epoch 59: val\_loss did not improve from 0.39560  
353/353 [=====] - 88s 248ms/step - loss: 0.1584 - accuracy: 0.9313 - val\_loss: 0.4351 - val\_accuracy: 0.8103  
Epoch 60/150  
353/353 [=====] - ETA: 0s - loss: 0.1617 - accuracy: 0.9350  
Epoch 60: val\_loss did not improve from 0.39560  
353/353 [=====] - 82s 231ms/step - loss: 0.1617 - accuracy: 0.9350 - val\_loss: 0.4374 - val\_accuracy: 0.8077  
Epoch 61/150  
353/353 [=====] - ETA: 0s - loss: 0.1553 - accuracy: 0.9330  
Epoch 61: val\_loss did not improve from 0.39560

353/353 [=====] - 86s 244ms/step - loss: 0.1553 - accuracy: 0.9330 - val\_loss: 0.4344 - val\_accuracy: 0.8196  
Epoch 62/150  
353/353 [=====] - ETA: 0s - loss: 0.1689 - accuracy: 0.9298  
Epoch 62: val\_loss did not improve from 0.39560  
353/353 [=====] - 94s 265ms/step - loss: 0.1689 - accuracy: 0.9298 - val\_loss: 0.4972 - val\_accuracy: 0.7719  
Epoch 63/150  
353/353 [=====] - ETA: 0s - loss: 0.1505 - accuracy: 0.9395  
Epoch 63: val\_loss did not improve from 0.39560  
353/353 [=====] - 85s 242ms/step - loss: 0.1505 - accuracy: 0.9395 - val\_loss: 0.3983 - val\_accuracy: 0.8183  
Epoch 64/150  
353/353 [=====] - ETA: 0s - loss: 0.1548 - accuracy: 0.9401  
Epoch 64: val\_loss did not improve from 0.39560  
353/353 [=====] - 95s 268ms/step - loss: 0.1548 - accuracy: 0.9401 - val\_loss: 0.4860 - val\_accuracy: 0.7825  
Epoch 65/150  
353/353 [=====] - ETA: 0s - loss: 0.1518 - accuracy: 0.9344  
Epoch 65: val\_loss did not improve from 0.39560  
353/353 [=====] - 92s 260ms/step - loss: 0.1518 - accuracy: 0.9344 - val\_loss: 0.4340 - val\_accuracy: 0.8236  
Epoch 66/150  
353/353 [=====] - ETA: 0s - loss: 0.1480 - accuracy: 0.9372  
Epoch 66: val\_loss did not improve from 0.39560  
353/353 [=====] - 86s 243ms/step - loss: 0.1480 - accuracy: 0.9372 - val\_loss: 0.4760 - val\_accuracy: 0.8196  
Epoch 67/150  
353/353 [=====] - ETA: 0s - loss: 0.1563 - accuracy: 0.9375  
Epoch 67: val\_loss did not improve from 0.39560  
353/353 [=====] - 86s 243ms/step - loss: 0.1563 - accuracy: 0.9375 - val\_loss: 0.4547 - val\_accuracy: 0.8037  
Epoch 68/150  
353/353 [=====] - ETA: 0s - loss: 0.1709 - accuracy: 0.9262  
Epoch 68: val\_loss did not improve from 0.39560  
353/353 [=====] - 85s 240ms/step - loss: 0.1709 - accuracy: 0.9262 - val\_loss: 0.4029 - val\_accuracy: 0.8236  
Epoch 69/150  
353/353 [=====] - ETA: 0s - loss: 0.1504 - accuracy: 0.9372  
Epoch 69: val\_loss did not improve from 0.39560  
353/353 [=====] - 81s 229ms/step - loss: 0.1504 - accuracy: 0.9372 - val\_loss: 0.4233 - val\_accuracy: 0.8103  
Epoch 70/150  
353/353 [=====] - ETA: 0s - loss: 0.1508 - accuracy: 0.9372  
Epoch 70: val\_loss did not improve from 0.39560

```

353/353 [=====] - 82s 233ms/step - loss: 0.1508 - accuracy:
0.9372 - val_loss: 0.4349 - val_accuracy: 0.8289
Epoch 71/150
353/353 [=====] - ETA: 0s - loss: 0.1465 - accuracy: 0.9375
Epoch 71: val_loss did not improve from 0.39560
353/353 [=====] - 93s 262ms/step - loss: 0.1465 - accuracy:
0.9375 - val_loss: 0.5530 - val_accuracy: 0.7812
Epoch 72/150
353/353 [=====] - ETA: 0s - loss: 0.1497 - accuracy: 0.9369
Epoch 72: val_loss did not improve from 0.39560
353/353 [=====] - 86s 242ms/step - loss: 0.1497 - accuracy:
0.9369 - val_loss: 0.4343 - val_accuracy: 0.8302
Epoch 73/150
353/353 [=====] - ETA: 0s - loss: 0.1487 - accuracy: 0.9404
Epoch 73: val_loss did not improve from 0.39560
353/353 [=====] - 87s 247ms/step - loss: 0.1487 - accuracy:
0.9404 - val_loss: 0.4650 - val_accuracy: 0.8117
Epoch 74/150
353/353 [=====] - ETA: 0s - loss: 0.1463 - accuracy: 0.9367
Epoch 74: val_loss improved from 0.39560 to 0.39185, saving model to
Modelo_Classificador_COVID-19_cnn.h5
353/353 [=====] - 91s 257ms/step - loss: 0.1463 - accuracy:
0.9367 - val_loss: 0.3918 - val_accuracy: 0.8263
Epoch 75/150
353/353 [=====] - ETA: 0s - loss: 0.1469 - accuracy: 0.9418
Epoch 75: val_loss improved from 0.39185 to 0.38023, saving model to
Modelo_Classificador_COVID-19_cnn.h5
353/353 [=====] - 84s 239ms/step - loss: 0.1469 - accuracy:
0.9418 - val_loss: 0.3802 - val_accuracy: 0.8422
Epoch 76/150
353/353 [=====] - ETA: 0s - loss: 0.1485 - accuracy: 0.9438
Epoch 76: val_loss did not improve from 0.38023
353/353 [=====] - 82s 231ms/step - loss: 0.1485 - accuracy:
0.9438 - val_loss: 0.4034 - val_accuracy: 0.8037
Epoch 77/150
353/353 [=====] - ETA: 0s - loss: 0.1313 - accuracy: 0.9458
Epoch 77: val_loss did not improve from 0.38023
353/353 [=====] - 81s 228ms/step - loss: 0.1313 - accuracy:
0.9458 - val_loss: 0.4918 - val_accuracy: 0.7984
Epoch 78/150
353/353 [=====] - ETA: 0s - loss: 0.1456 - accuracy: 0.9404
Epoch 78: val_loss did not improve from 0.38023
353/353 [=====] - 84s 238ms/step - loss: 0.1456 - accuracy:
0.9404 - val_loss: 0.3891 - val_accuracy: 0.8236
Epoch 79/150

```

353/353 [=====] - ETA: 0s - loss: 0.1506 - accuracy: 0.9398  
Epoch 79: val\_loss did not improve from 0.38023  
353/353 [=====] - 95s 269ms/step - loss: 0.1506 - accuracy:  
0.9398 - val\_loss: 0.3860 - val\_accuracy: 0.8528  
Epoch 80/150  
353/353 [=====] - ETA: 0s - loss: 0.1374 - accuracy: 0.9404  
Epoch 80: val\_loss improved from 0.38023 to 0.36965, saving model to  
Modelo\_Clasificador\_COVID-19\_cnn.h5  
353/353 [=====] - 92s 260ms/step - loss: 0.1374 - accuracy:  
0.9404 - val\_loss: 0.3697 - val\_accuracy: 0.8607  
Epoch 81/150  
353/353 [=====] - ETA: 0s - loss: 0.1501 - accuracy: 0.9369  
Epoch 81: val\_loss improved from 0.36965 to 0.36632, saving model to  
Modelo\_Clasificador\_COVID-19\_cnn.h5  
353/353 [=====] - 81s 230ms/step - loss: 0.1501 - accuracy:  
0.9369 - val\_loss: 0.3663 - val\_accuracy: 0.8369  
Epoch 82/150  
353/353 [=====] - ETA: 0s - loss: 0.1432 - accuracy: 0.9398  
Epoch 82: val\_loss did not improve from 0.36632  
353/353 [=====] - 78s 221ms/step - loss: 0.1432 - accuracy:  
0.9398 - val\_loss: 0.4936 - val\_accuracy: 0.7931  
Epoch 83/150  
353/353 [=====] - ETA: 0s - loss: 0.1413 - accuracy: 0.9415  
Epoch 83: val\_loss did not improve from 0.36632  
353/353 [=====] - 78s 221ms/step - loss: 0.1413 - accuracy:  
0.9415 - val\_loss: 0.4746 - val\_accuracy: 0.7878  
Epoch 84/150  
353/353 [=====] - ETA: 0s - loss: 0.1341 - accuracy: 0.9455  
Epoch 84: val\_loss did not improve from 0.36632  
353/353 [=====] - 85s 240ms/step - loss: 0.1341 - accuracy:  
0.9455 - val\_loss: 0.4243 - val\_accuracy: 0.8223  
Epoch 85/150  
353/353 [=====] - ETA: 0s - loss: 0.1404 - accuracy: 0.9441  
Epoch 85: val\_loss improved from 0.36632 to 0.35060, saving model to  
Modelo\_Clasificador\_COVID-19\_cnn.h5  
353/353 [=====] - 91s 258ms/step - loss: 0.1404 - accuracy:  
0.9441 - val\_loss: 0.3506 - val\_accuracy: 0.8342  
Epoch 86/150  
353/353 [=====] - ETA: 0s - loss: 0.1410 - accuracy: 0.9441  
Epoch 86: val\_loss did not improve from 0.35060  
353/353 [=====] - 90s 255ms/step - loss: 0.1410 - accuracy:  
0.9441 - val\_loss: 0.4244 - val\_accuracy: 0.8302  
Epoch 87/150  
353/353 [=====] - ETA: 0s - loss: 0.1408 - accuracy: 0.9395



Epoch 87: val\_loss improved from 0.35060 to 0.34168, saving model to Modelo\_Clasificador\_COVID-19\_cnn.h5  
353/353 [=====] - 86s 244ms/step - loss: 0.1408 - accuracy: 0.9395 - val\_loss: 0.3417 - val\_accuracy: 0.8395  
Epoch 88/150  
353/353 [=====] - ETA: 0s - loss: 0.1429 - accuracy: 0.9415  
Epoch 88: val\_loss did not improve from 0.34168  
353/353 [=====] - 94s 267ms/step - loss: 0.1429 - accuracy: 0.9415 - val\_loss: 0.4040 - val\_accuracy: 0.8302  
Epoch 89/150  
353/353 [=====] - ETA: 0s - loss: 0.1365 - accuracy: 0.9398  
Epoch 89: val\_loss did not improve from 0.34168  
353/353 [=====] - 87s 248ms/step - loss: 0.1365 - accuracy: 0.9398 - val\_loss: 0.4670 - val\_accuracy: 0.8210  
Epoch 90/150  
353/353 [=====] - ETA: 0s - loss: 0.1415 - accuracy: 0.9429  
Epoch 90: val\_loss did not improve from 0.34168  
353/353 [=====] - 85s 242ms/step - loss: 0.1415 - accuracy: 0.9429 - val\_loss: 0.4274 - val\_accuracy: 0.8077  
Epoch 91/150  
353/353 [=====] - ETA: 0s - loss: 0.1405 - accuracy: 0.9443  
Epoch 91: val\_loss did not improve from 0.34168  
353/353 [=====] - 80s 228ms/step - loss: 0.1405 - accuracy: 0.9443 - val\_loss: 0.3627 - val\_accuracy: 0.8515  
Epoch 92/150  
353/353 [=====] - ETA: 0s - loss: 0.1413 - accuracy: 0.9404  
Epoch 92: val\_loss did not improve from 0.34168  
353/353 [=====] - 80s 226ms/step - loss: 0.1413 - accuracy: 0.9404 - val\_loss: 0.3974 - val\_accuracy: 0.8448  
Epoch 93/150  
353/353 [=====] - ETA: 0s - loss: 0.1387 - accuracy: 0.9412  
Epoch 93: val\_loss did not improve from 0.34168  
353/353 [=====] - 83s 235ms/step - loss: 0.1387 - accuracy: 0.9412 - val\_loss: 0.4783 - val\_accuracy: 0.8263  
Epoch 94/150  
353/353 [=====] - ETA: 0s - loss: 0.1294 - accuracy: 0.9466  
Epoch 94: val\_loss did not improve from 0.34168  
353/353 [=====] - 81s 228ms/step - loss: 0.1294 - accuracy: 0.9466 - val\_loss: 0.3585 - val\_accuracy: 0.8475  
Epoch 95/150  
353/353 [=====] - ETA: 0s - loss: 0.1336 - accuracy: 0.9415  
Epoch 95: val\_loss did not improve from 0.34168  
353/353 [=====] - 86s 244ms/step - loss: 0.1336 - accuracy: 0.9415 - val\_loss: 0.3934 - val\_accuracy: 0.8408  
Epoch 96/150

353/353 [=====] - ETA: 0s - loss: 0.1315 - accuracy: 0.9432  
Epoch 96: val\_loss did not improve from 0.34168  
353/353 [=====] - 81s 229ms/step - loss: 0.1315 - accuracy: 0.9432 - val\_loss: 0.4254 - val\_accuracy: 0.8488  
Epoch 97/150  
353/353 [=====] - ETA: 0s - loss: 0.1398 - accuracy: 0.9435  
Epoch 97: val\_loss did not improve from 0.34168  
353/353 [=====] - 81s 229ms/step - loss: 0.1398 - accuracy: 0.9435 - val\_loss: 0.4018 - val\_accuracy: 0.8249  
Epoch 98/150  
353/353 [=====] - ETA: 0s - loss: 0.1309 - accuracy: 0.9472  
Epoch 98: val\_loss did not improve from 0.34168  
353/353 [=====] - 80s 228ms/step - loss: 0.1309 - accuracy: 0.9472 - val\_loss: 0.5076 - val\_accuracy: 0.7599  
Epoch 99/150  
353/353 [=====] - ETA: 0s - loss: 0.1366 - accuracy: 0.9455  
Epoch 99: val\_loss did not improve from 0.34168  
353/353 [=====] - 81s 228ms/step - loss: 0.1366 - accuracy: 0.9455 - val\_loss: 0.4154 - val\_accuracy: 0.8462  
Epoch 100/150  
353/353 [=====] - ETA: 0s - loss: 0.1343 - accuracy: 0.9480  
Epoch 100: val\_loss did not improve from 0.34168  
353/353 [=====] - 83s 234ms/step - loss: 0.1343 - accuracy: 0.9480 - val\_loss: 0.4292 - val\_accuracy: 0.8316  
Epoch 101/150  
353/353 [=====] - ETA: 0s - loss: 0.1244 - accuracy: 0.9472  
Epoch 101: val\_loss did not improve from 0.34168  
353/353 [=====] - 80s 228ms/step - loss: 0.1244 - accuracy: 0.9472 - val\_loss: 0.3877 - val\_accuracy: 0.8488  
Epoch 102/150  
353/353 [=====] - ETA: 0s - loss: 0.1351 - accuracy: 0.9472  
Epoch 102: val\_loss did not improve from 0.34168  
353/353 [=====] - 80s 225ms/step - loss: 0.1351 - accuracy: 0.9472 - val\_loss: 0.5647 - val\_accuracy: 0.8130  
Epoch 103/150  
353/353 [=====] - ETA: 0s - loss: 0.1193 - accuracy: 0.9494  
Epoch 103: val\_loss did not improve from 0.34168  
353/353 [=====] - 80s 227ms/step - loss: 0.1193 - accuracy: 0.9494 - val\_loss: 0.3716 - val\_accuracy: 0.8621  
Epoch 104/150  
353/353 [=====] - ETA: 0s - loss: 0.1272 - accuracy: 0.9494  
Epoch 104: val\_loss did not improve from 0.34168  
353/353 [=====] - 80s 226ms/step - loss: 0.1272 - accuracy: 0.9494 - val\_loss: 0.4409 - val\_accuracy: 0.8183  
Epoch 105/150

353/353 [=====] - ETA: 0s - loss: 0.1357 - accuracy: 0.9418  
Epoch 105: val\_loss improved from 0.34168 to 0.32853, saving model to  
Modelo\_Clasificador\_COVID-19\_cnn.h5  
353/353 [=====] - 81s 231ms/step - loss: 0.1357 - accuracy:  
0.9418 - val\_loss: 0.3285 - val\_accuracy: 0.8541  
Epoch 106/150  
353/353 [=====] - ETA: 0s - loss: 0.1289 - accuracy: 0.9460  
Epoch 106: val\_loss did not improve from 0.32853  
353/353 [=====] - 86s 243ms/step - loss: 0.1289 - accuracy:  
0.9460 - val\_loss: 0.4839 - val\_accuracy: 0.8316  
Epoch 107/150  
353/353 [=====] - ETA: 0s - loss: 0.1255 - accuracy: 0.9466  
Epoch 107: val\_loss did not improve from 0.32853  
353/353 [=====] - 80s 226ms/step - loss: 0.1255 - accuracy:  
0.9466 - val\_loss: 0.4591 - val\_accuracy: 0.7984  
Epoch 108/150  
353/353 [=====] - ETA: 0s - loss: 0.1182 - accuracy: 0.9494  
Epoch 108: val\_loss did not improve from 0.32853  
353/353 [=====] - 81s 228ms/step - loss: 0.1182 - accuracy:  
0.9494 - val\_loss: 0.4190 - val\_accuracy: 0.8621  
Epoch 109/150  
353/353 [=====] - ETA: 0s - loss: 0.1216 - accuracy: 0.9512  
Epoch 109: val\_loss did not improve from 0.32853  
353/353 [=====] - 80s 226ms/step - loss: 0.1216 - accuracy:  
0.9512 - val\_loss: 0.4026 - val\_accuracy: 0.8249  
Epoch 110/150  
353/353 [=====] - ETA: 0s - loss: 0.1224 - accuracy: 0.9483  
Epoch 110: val\_loss did not improve from 0.32853  
353/353 [=====] - 80s 227ms/step - loss: 0.1224 - accuracy:  
0.9483 - val\_loss: 0.3670 - val\_accuracy: 0.8515  
Epoch 111/150  
353/353 [=====] - ETA: 0s - loss: 0.1309 - accuracy: 0.9477  
Epoch 111: val\_loss did not improve from 0.32853  
353/353 [=====] - 81s 229ms/step - loss: 0.1309 - accuracy:  
0.9477 - val\_loss: 0.4491 - val\_accuracy: 0.8263  
Epoch 112/150  
353/353 [=====] - ETA: 0s - loss: 0.1293 - accuracy: 0.9460  
Epoch 112: val\_loss did not improve from 0.32853  
353/353 [=====] - 80s 227ms/step - loss: 0.1293 - accuracy:  
0.9460 - val\_loss: 0.3917 - val\_accuracy: 0.8462  
Epoch 113/150  
353/353 [=====] - ETA: 0s - loss: 0.1253 - accuracy: 0.9486  
Epoch 113: val\_loss did not improve from 0.32853  
353/353 [=====] - 83s 234ms/step - loss: 0.1253 - accuracy:  
0.9486 - val\_loss: 0.3569 - val\_accuracy: 0.8607

Epoch 114/150  
353/353 [=====] - ETA: 0s - loss: 0.1369 - accuracy: 0.9426  
Epoch 114: val\_loss did not improve from 0.32853  
353/353 [=====] - 82s 232ms/step - loss: 0.1369 - accuracy: 0.9426 - val\_loss: 0.4347 - val\_accuracy: 0.8395  
Epoch 115/150  
353/353 [=====] - ETA: 0s - loss: 0.1215 - accuracy: 0.9486  
Epoch 115: val\_loss did not improve from 0.32853  
353/353 [=====] - 87s 246ms/step - loss: 0.1215 - accuracy: 0.9486 - val\_loss: 0.3804 - val\_accuracy: 0.8674  
Epoch 116/150  
353/353 [=====] - ETA: 0s - loss: 0.1237 - accuracy: 0.9506  
Epoch 116: val\_loss did not improve from 0.32853  
353/353 [=====] - 93s 264ms/step - loss: 0.1237 - accuracy: 0.9506 - val\_loss: 0.4439 - val\_accuracy: 0.8621  
Epoch 117/150  
353/353 [=====] - ETA: 0s - loss: 0.1260 - accuracy: 0.9489  
Epoch 117: val\_loss did not improve from 0.32853  
353/353 [=====] - 85s 240ms/step - loss: 0.1260 - accuracy: 0.9489 - val\_loss: 0.4183 - val\_accuracy: 0.8488  
Epoch 118/150  
353/353 [=====] - ETA: 0s - loss: 0.1146 - accuracy: 0.9546  
Epoch 118: val\_loss did not improve from 0.32853  
353/353 [=====] - 86s 243ms/step - loss: 0.1146 - accuracy: 0.9546 - val\_loss: 0.3513 - val\_accuracy: 0.8594  
Epoch 119/150  
353/353 [=====] - ETA: 0s - loss: 0.1286 - accuracy: 0.9497  
Epoch 119: val\_loss did not improve from 0.32853  
353/353 [=====] - 82s 231ms/step - loss: 0.1286 - accuracy: 0.9497 - val\_loss: 0.4406 - val\_accuracy: 0.8316  
Epoch 120/150  
353/353 [=====] - ETA: 0s - loss: 0.1231 - accuracy: 0.9503  
Epoch 120: val\_loss did not improve from 0.32853  
353/353 [=====] - 85s 241ms/step - loss: 0.1231 - accuracy: 0.9503 - val\_loss: 0.3676 - val\_accuracy: 0.8488  
Epoch 121/150  
353/353 [=====] - ETA: 0s - loss: 0.1259 - accuracy: 0.9480  
Epoch 121: val\_loss did not improve from 0.32853  
353/353 [=====] - 101s 285ms/step - loss: 0.1259 - accuracy: 0.9480 - val\_loss: 0.4007 - val\_accuracy: 0.8647  
Epoch 122/150  
353/353 [=====] - ETA: 0s - loss: 0.1193 - accuracy: 0.9534  
Epoch 122: val\_loss did not improve from 0.32853  
353/353 [=====] - 99s 279ms/step - loss: 0.1193 - accuracy: 0.9534 - val\_loss: 0.4132 - val\_accuracy: 0.8621

Epoch 123/150  
353/353 [=====] - ETA: 0s - loss: 0.1211 - accuracy: 0.9509  
Epoch 123: val\_loss did not improve from 0.32853  
353/353 [=====] - 98s 276ms/step - loss: 0.1211 - accuracy: 0.9509 - val\_loss: 0.4051 - val\_accuracy: 0.8302  
Epoch 124/150  
353/353 [=====] - ETA: 0s - loss: 0.1187 - accuracy: 0.9580  
Epoch 124: val\_loss did not improve from 0.32853  
353/353 [=====] - 100s 283ms/step - loss: 0.1187 - accuracy: 0.9580 - val\_loss: 0.3928 - val\_accuracy: 0.8607  
Epoch 125/150  
353/353 [=====] - ETA: 0s - loss: 0.1252 - accuracy: 0.9503  
Epoch 125: val\_loss did not improve from 0.32853  
353/353 [=====] - 87s 245ms/step - loss: 0.1252 - accuracy: 0.9503 - val\_loss: 0.4118 - val\_accuracy: 0.8462  
Epoch 126/150  
353/353 [=====] - ETA: 0s - loss: 0.1189 - accuracy: 0.9531  
Epoch 126: val\_loss did not improve from 0.32853  
353/353 [=====] - 87s 247ms/step - loss: 0.1189 - accuracy: 0.9531 - val\_loss: 0.6899 - val\_accuracy: 0.7931  
Epoch 127/150  
353/353 [=====] - ETA: 0s - loss: 0.1246 - accuracy: 0.9480  
Epoch 127: val\_loss did not improve from 0.32853  
353/353 [=====] - 92s 259ms/step - loss: 0.1246 - accuracy: 0.9480 - val\_loss: 0.3884 - val\_accuracy: 0.8581  
Epoch 128/150  
353/353 [=====] - ETA: 0s - loss: 0.1280 - accuracy: 0.9466  
Epoch 128: val\_loss did not improve from 0.32853  
353/353 [=====] - 95s 270ms/step - loss: 0.1280 - accuracy: 0.9466 - val\_loss: 0.3955 - val\_accuracy: 0.8488  
Epoch 129/150  
353/353 [=====] - ETA: 0s - loss: 0.1246 - accuracy: 0.9551  
Epoch 129: val\_loss did not improve from 0.32853  
353/353 [=====] - 88s 250ms/step - loss: 0.1246 - accuracy: 0.9551 - val\_loss: 0.3316 - val\_accuracy: 0.8634  
Epoch 130/150  
353/353 [=====] - ETA: 0s - loss: 0.1156 - accuracy: 0.9546  
Epoch 130: val\_loss did not improve from 0.32853  
353/353 [=====] - 90s 254ms/step - loss: 0.1156 - accuracy: 0.9546 - val\_loss: 0.3923 - val\_accuracy: 0.8634  
Epoch 131/150  
353/353 [=====] - ETA: 0s - loss: 0.1105 - accuracy: 0.9546  
Epoch 131: val\_loss did not improve from 0.32853  
353/353 [=====] - 87s 247ms/step - loss: 0.1105 - accuracy: 0.9546 - val\_loss: 0.4375 - val\_accuracy: 0.8501

Epoch 132/150  
353/353 [=====] - ETA: 0s - loss: 0.1137 - accuracy: 0.9560  
Epoch 132: val\_loss did not improve from 0.32853  
353/353 [=====] - 92s 261ms/step - loss: 0.1137 - accuracy:  
0.9560 - val\_loss: 0.3820 - val\_accuracy: 0.8515  
Epoch 133/150  
353/353 [=====] - ETA: 0s - loss: 0.1189 - accuracy: 0.9551  
Epoch 133: val\_loss did not improve from 0.32853  
353/353 [=====] - 91s 258ms/step - loss: 0.1189 - accuracy:  
0.9551 - val\_loss: 0.4085 - val\_accuracy: 0.8462  
Epoch 134/150  
353/353 [=====] - ETA: 0s - loss: 0.1164 - accuracy: 0.9509  
Epoch 134: val\_loss did not improve from 0.32853  
353/353 [=====] - 98s 276ms/step - loss: 0.1164 - accuracy:  
0.9509 - val\_loss: 0.3522 - val\_accuracy: 0.8568  
Epoch 135/150  
353/353 [=====] - ETA: 0s - loss: 0.1069 - accuracy: 0.9600  
Epoch 135: val\_loss did not improve from 0.32853  
353/353 [=====] - 102s 288ms/step - loss: 0.1069 - accuracy:  
0.9600 - val\_loss: 0.4649 - val\_accuracy: 0.8594  
Epoch 136/150  
353/353 [=====] - ETA: 0s - loss: 0.1312 - accuracy: 0.9486  
Epoch 136: val\_loss did not improve from 0.32853  
353/353 [=====] - 97s 276ms/step - loss: 0.1312 - accuracy:  
0.9486 - val\_loss: 0.3768 - val\_accuracy: 0.8541  
Epoch 137/150  
353/353 [=====] - ETA: 0s - loss: 0.1115 - accuracy: 0.9512  
Epoch 137: val\_loss improved from 0.32853 to 0.31306, saving model to  
Modelo\_Clasificador\_COVID-19\_cnn.h5  
353/353 [=====] - 103s 291ms/step - loss: 0.1115 - accuracy:  
0.9512 - val\_loss: 0.3131 - val\_accuracy: 0.8740  
Epoch 138/150  
353/353 [=====] - ETA: 0s - loss: 0.1165 - accuracy: 0.9563  
Epoch 138: val\_loss did not improve from 0.31306  
353/353 [=====] - 87s 245ms/step - loss: 0.1165 - accuracy:  
0.9563 - val\_loss: 0.3946 - val\_accuracy: 0.8249  
Epoch 139/150  
353/353 [=====] - ETA: 0s - loss: 0.1172 - accuracy: 0.9512  
Epoch 139: val\_loss did not improve from 0.31306  
353/353 [=====] - 93s 262ms/step - loss: 0.1172 - accuracy:  
0.9512 - val\_loss: 0.3804 - val\_accuracy: 0.8660  
Epoch 140/150  
353/353 [=====] - ETA: 0s - loss: 0.1133 - accuracy: 0.9534  
Epoch 140: val\_loss did not improve from 0.31306

353/353 [=====] - 90s 255ms/step - loss: 0.1133 - accuracy:  
0.9534 - val\_loss: 0.5734 - val\_accuracy: 0.7759  
Epoch 141/150  
353/353 [=====] - ETA: 0s - loss: 0.1198 - accuracy: 0.9565  
Epoch 141: val\_loss did not improve from 0.31306  
353/353 [=====] - 90s 254ms/step - loss: 0.1198 - accuracy:  
0.9565 - val\_loss: 0.4315 - val\_accuracy: 0.8329  
Epoch 142/150  
353/353 [=====] - ETA: 0s - loss: 0.1086 - accuracy: 0.9534  
Epoch 142: val\_loss did not improve from 0.31306  
353/353 [=====] - 92s 259ms/step - loss: 0.1086 - accuracy:  
0.9534 - val\_loss: 0.4681 - val\_accuracy: 0.8329  
Epoch 143/150  
353/353 [=====] - ETA: 0s - loss: 0.1239 - accuracy: 0.9486  
Epoch 143: val\_loss did not improve from 0.31306  
353/353 [=====] - 93s 264ms/step - loss: 0.1239 - accuracy:  
0.9486 - val\_loss: 0.4032 - val\_accuracy: 0.8329  
Epoch 144/150  
353/353 [=====] - ETA: 0s - loss: 0.1164 - accuracy: 0.9503  
Epoch 144: val\_loss did not improve from 0.31306  
353/353 [=====] - 88s 248ms/step - loss: 0.1164 - accuracy:  
0.9503 - val\_loss: 0.4362 - val\_accuracy: 0.8541  
Epoch 145/150  
353/353 [=====] - ETA: 0s - loss: 0.1091 - accuracy: 0.9540  
Epoch 145: val\_loss did not improve from 0.31306  
353/353 [=====] - 91s 258ms/step - loss: 0.1091 - accuracy:  
0.9540 - val\_loss: 0.3221 - val\_accuracy: 0.8859  
Epoch 146/150  
353/353 [=====] - ETA: 0s - loss: 0.1101 - accuracy: 0.9560  
Epoch 146: val\_loss did not improve from 0.31306  
353/353 [=====] - 95s 268ms/step - loss: 0.1101 - accuracy:  
0.9560 - val\_loss: 0.3834 - val\_accuracy: 0.8302  
Epoch 147/150  
353/353 [=====] - ETA: 0s - loss: 0.1142 - accuracy: 0.9563  
Epoch 147: val\_loss did not improve from 0.31306  
353/353 [=====] - 109s 308ms/step - loss: 0.1142 - accuracy:  
0.9563 - val\_loss: 0.4569 - val\_accuracy: 0.8249  
Epoch 148/150  
353/353 [=====] - ETA: 0s - loss: 0.0944 - accuracy: 0.9588  
Epoch 148: val\_loss did not improve from 0.31306  
353/353 [=====] - 98s 277ms/step - loss: 0.0944 - accuracy:  
0.9588 - val\_loss: 0.4244 - val\_accuracy: 0.8422  
Epoch 149/150  
353/353 [=====] - ETA: 0s - loss: 0.1098 - accuracy: 0.9560  
Epoch 149: val\_loss did not improve from 0.31306

```
353/353 [=====] - 96s 271ms/step - loss: 0.1098 - accuracy:
0.9560 - val_loss: 0.3652 - val_accuracy: 0.8594
Epoch 150/150
353/353 [=====] - ETA: 0s - loss: 0.1119 - accuracy: 0.9529
Epoch 150: val_loss did not improve from 0.31306
353/353 [=====] - 87s 246ms/step - loss: 0.1119 - accuracy:
0.9529 - val_loss: 0.3732 - val_accuracy: 0.8700
```