



# **UNIVERSIDAD DE GUAYAQUIL**

**FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS**

**CARRERA DE INGENIERÍA EN SISTEMAS**

**COMPUTACIONALES**

**ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES**

**NEURONALES ARTIFICIALES**

## **PROYECTO DE TITULACIÓN**

Previa a la obtención del Título de:

**INGENIERO EN SISTEMAS COMPUTACIONALES**

**AUTORES:**

**MOROCHO SANDE KEVIN LUIS**

**PULIG CESÉN DANNY XAVIER**

**TUTOR:**

**ING. DARWIN PATIÑO PÉREZ M.Sc.**

**GUAYAQUIL – ECUADOR**

**2021**

  		
<b>REPOSITORIO NACIONAL EN CIENCIAS Y TECNOLOGÍAS</b>		
<b>FICHA DE REGISTRO DE TRABAJO DE TITULACIÓN</b>		
<b>TÍTULO:</b> “ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES NEURONALES ARTIFICIALES”		
<b>AUTORES:</b> Kevin Luis Morocho Sande Danny Xavier Pulig Cesén	<b>REVISOR:</b> Ing. Gilberto Fernando Castro Aguilar, PhD.	
<b>INSTITUCIÓN:</b> Universidad de Guayaquil	<b>FACULTAD:</b> Ciencias Matemáticas y Físicas	
<b>CARRERA:</b> Ingeniería en Sistemas Computacionales		
<b>FECHA DE PUBLICACIÓN:</b>	<b>N° DE PAGS:</b> 167	
<b>AREA TEMÁTICA:</b> <i>Investigación</i>		
<b>PALABRAS CLAVE:</b> Covid-19, rayos X, machine learning, modelo de predicción, redes neuronales convolucionales.		
<p><b>RESUMEN:</b> En la actualidad, las pruebas más efectivas para la detección de Covid-19 como el análisis de imágenes por tomografía axial computarizada (TAC) y las evaluaciones médicas de laboratorio como la prueba de reacción en cadena de la polimerasa (PCR), son uno de los métodos más efectivos para el diagnóstico de esta enfermedad. La única desventaja es que estos métodos tradicionales son muy costosos y eso implicaba que la población de clase media-baja no tenga acceso a estas pruebas de detección. Las redes neuronales artificiales juegan un papel muy importante en el campo de la medicina y en investigaciones tecnológicas que contribuyan a la detección de enfermedades como el Covid-19. Este proyecto se centralizó con el fin de proporcionar un apoyo a los médicos en la toma de decisiones, utilizando herramientas tecnológicas. Se realizó un modelo de machine learning para el análisis de imágenes de rayos X para la detección de Covid-19 a través de redes neuronales convolucionales. El algoritmo fue realizado en el lenguaje de programación Python con el uso del entorno de desarrollo en la nube de Google Colaboratory. Por medio de los repositorios de acceso público de GitHub y Kaggle, se recolectaron bases de datos de imágenes de rayos X de tórax de pacientes con Covid-19 y pacientes normales (sanos) para generar un dataset con imágenes de entrenamiento y validación. Se realizaron pruebas experimentales con tres tipos de arquitectura de redes neuronales convolucionales para los modelos A, B y C. Siendo el modelo C el que obtuvo los mejores resultados.</p>		
<b>N° DE REGISTRO:</b>	<b>N° DE CLASIFICACIÓN:</b>	
<b>DIRECCIÓN URL: (PROYECTO DE TITULACION EN LA WEB)</b>		
<b>ADJUNTO PDF</b>	<b>SI</b> <input checked="" type="checkbox"/>	<b>NO</b> <input type="checkbox"/>
<b>CONTACTO CON AUTORES:</b> Kevin Luis Morocho Sande Danny Xavier Pulig Cesén	<b>Teléfono:</b> 0939844463 0980230148	<b>Email:</b> <a href="mailto:kevin.morochos@ug.edu.ec">kevin.morochos@ug.edu.ec</a> <a href="mailto:danny.puligc@ug.edu.ec">danny.puligc@ug.edu.ec</a>
<b>CONTACTO DE LA INSTITUCIÓN</b>	<b>Nombre:</b> Ab. Juan Chávez Atocha	
	<b>Teléfono:</b> 2307729	
	<b>Email:</b> <a href="mailto:juan.chaveza@ug.edu.ec">juan.chaveza@ug.edu.ec</a>	

## **APROBACIÓN DEL TUTOR**

En mi calidad de Tutor del Trabajo de Titulación, Análisis de imágenes de rayos X de Covid-19 a través de redes neuronales artificiales, elaborado por los Sres. Morocho Sande Kevin Luis, Pulig Cesén Danny Xavier, estudiantes no titulados de la Carrera de Ingeniería en Sistemas Computacionales, Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil, previo a la obtención del Título de Ingeniero en Sistemas Computacionales, me permito declarar que luego de haber orientado, estudiado y revisado, la apruebo en todas sus partes.

**Atentamente,**

Ing. Darwin Patiño Pérez M.Sc.

**TUTOR**

## DEDICATORIA

Dedico este proyecto a mis padres y hermanos, mis seres amados, quienes son parte fundamental de este proceso a través de mis años de estudio, quienes hicieron posible este logro.

*Kevin Luis Morocho Sande*

Dedico este trabajo de titulación principalmente para mi madre Mariana Cesén Paguay y a mi abuelita Ramona Paguay Contero por el apoyo que me han brindado durante toda mi carrea universitaria, también a mi padre y hermanos por sus palabras de apoyo que recibía cada día

*Danny Xavier Pulig Cesén*

## AGRADECIMIENTO

A Dios, por el regalo más preciado que poseo, la vida. Por otorgarme la sabiduría, la fortaleza y el conocimiento para cumplir mis sueños. A mis padres Xavier Morocho Yupa y Glenda Sande León por ser mi motor, mi pilar e inspiración para cumplir mis metas. Gracias por su sacrificio, amor y apoyo incondicional. A mis hermanos Danny y Allison, a quienes amo con toda mi alma por todo el cariño y por creer en mí. A mi familia y a mis amigos más cercanos, este logro es de todos ustedes.

*Kevin Luis Morocho Sande*

En primer lugar le doy gracias a Dios por permitirme terminar con éxito mi carrera universitaria, luego a mi madre Marina Cesén y a mi abuelita Ramona Paguay Contero ellas han sido mis dos pilares principales para poder convertirme en todo un profesional como ingeniero, gracias a sus esfuerzos y consejos pude superar cualquier obstáculo, a mi Padre Virgilio Pulig y mis hermanos Mauricio Pulig y Joel Pulig quienes nunca dudaron que podría conseguir dicho logro académico, y a mis amigos por darme sus palabras de aliento. A los Dres. Javier Mendieta y Peter Arteaga por la contribución de sus valiosos conocimientos y su experiencia.

*Danny Xavier Pulig Cesén*

## TRIBUNAL PROYECTO DE TITULACIÓN

---

Ing. José González Ruiz, M.Sc.  
DECANO DE LA FACULTAD  
CIENCIAS MATEMÁTICAS Y FÍSICAS

---

Ing. Lorenzo Cevallos Torres, Mgs.  
DIRECTOR DE LA CARRERA DE  
INGENIERÍA EN SISTEMAS  
COMPUTACIONALES

---

Ing. Darwin Patiño Pérez, M.Sc.  
PROFESOR TUTOR DEL PROYECTO  
DE TITULACIÓN

---

Ing. Gilberto Fernando Castro Aguilar, PhD.  
PROFESOR REVISOR DEL PROYECTO  
DE TITULACIÓN

---

Ab. Juan Chávez Atocha, Esp.  
SECRETARIO

## **DECLARACIÓN EXPRESA**

“La responsabilidad del contenido de este Proyecto de Titulación, me corresponden exclusivamente; y el patrimonio intelectual de la misma a la UNIVERSIDAD DE GUAYAQUIL”.

Kevin Luis Morocho Sande

Danny Xavier Pulig Cesén



## **CESIÓN DE DERECHOS DE AUTOR**

Ingeniero

José González Ruiz, M.Sc.

**DECANO DE LA FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS**

Presente.

A través de este medio indico a usted que procedo a realizar la entrega de la cesión de derechos de autor en forma libre y voluntaria del trabajo de titulación “**ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES NEURONALES ARTIFICIALES**”, realizado como requisito previo para la obtención del Título de Ingeniero en Sistemas Computacionales de la Universidad de Guayaquil.

Guayaquil, marzo de 2021.

---

**Kevin Luis Morocho Sande**  
**C.I. N° 0931516132**

---

**Danny Xavier Pulig Cesén**  
**C.I. N° 0930417753**



**UNIVERSIDAD DE GUAYAQUIL**

**FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS**

**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES  
NEURONALES ARTIFICIALES**

Proyecto de Titulación que se presenta como requisito para optar por el título de  
**INGENIERO EN SISTEMAS COMPUTACIONALES**

**Autores:** Morocho Sande Kevin Luis

**C.I. N° 0931516132**

Pulig Cesén Danny Xavier

**C.I. N° 0930417753**

**Tutor:** Ing. Darwin Patiño Pérez M.Sc.

Guayaquil, marzo de 2021

## CERTIFICADO DE ACEPTACIÓN DEL TUTOR

En mi calidad de Tutor del Proyecto de Titulación, nombrado por el Consejo Directivo de la Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil.

### CERTIFICO:

Que he analizado el Proyecto de Titulación presentado por los estudiantes **KEVIN LUIS MOROCHO SANDE, DANNY XAVIER PULIG CESÉN**, como requisito previo para optar por el Título de Ingeniero en Sistemas Computacionales cuyo proyecto es:

### **ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES NEURONALES ARTIFICIALES**

Considero aprobado el trabajo en su totalidad.

Presentado por:

\_\_\_\_\_  
Morocho Sande Kevin Luis

\_\_\_\_\_  
0931516132

\_\_\_\_\_  
Pulig Cesén Danny Xavier

\_\_\_\_\_  
0930417753

**Firma Tutor:** \_\_\_\_\_

Guayaquil, marzo de 2021



**UNIVERSIDAD DE GUAYAQUIL**  
**FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS**  
**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**AUTORIZACIÓN PARA PUBLICACIÓN DE PROYECTO DE TITULACIÓN EN FORMATO DIGITAL**

**1. Identificación del Proyecto de Titulación**

<b>Nombre del Estudiante:</b> Kevin Luis Morocho Sande	
<b>Dirección:</b> Coop. Virgen del Cisne Mz. 677 Solar 8B	
<b>Teléfono:</b> 0939844463	<b>Email:</b> kevin.morochos@ug.edu.ec

<b>Nombre del Estudiante:</b> Danny Xavier Pulig Cesén	
<b>Dirección:</b> Azuay y la 19	
<b>Teléfono:</b> 0980230148	<b>Email:</b> danny.puligc@ug.edu.ec

<b>Facultad:</b> Ciencias Matemáticas y Físicas
<b>Carrera:</b> Ingeniería en Sistemas Computacionales
<b>Proyecto de Titulación al que opta:</b> Ingeniería en Sistemas Computacionales
<b>Profesor Tutor:</b> Ing. Darwin Patiño Pérez M.Sc.

<b>Título del Proyecto de Titulación:</b> ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES NEURONALES ARTIFICIALES
---

<b>Palabras Clave:</b> Covid-19, rayos X, machine learning, modelo de predicción, redes neuronales convolucionales.
---

**2. Autorización de Publicación de Versión Electrónica del Proyecto de Titulación**

A través de este medio autorizo a la Biblioteca de la Universidad de Guayaquil y a la Facultad de Ciencias Matemáticas y Físicas a publicar la versión electrónica de este Proyecto de Titulación.

**Publicación Electrónica:**

Inmediata		Después de 1 año	
-----------	--	------------------	--

Firma Estudiante:

\_\_\_\_\_  
Morocho Sande Kevin Luis

\_\_\_\_\_  
0931516132

\_\_\_\_\_  
Pulig Cesén Danny Xavier

\_\_\_\_\_  
0930417753

**3. Forma de envío:**

El texto del Proyecto de Titulación debe ser enviado en formato Word, como archivo .docx, .RTF o Puf para PC. Las imágenes que la acompañen pueden ser: .gif, .jpg o .TIFF.

DVDROM

CDROM

## ÍNDICE GENERAL

<b>FICHA DE REGISTRO DE TRABAJO DE TITULACIÓN .....</b>	<b>II</b>
<b>APROBACIÓN DEL TUTOR.....</b>	<b>III</b>
<b>DEDICATORIA.....</b>	<b>IV</b>
<b>AGRADECIMIENTO .....</b>	<b>V</b>
<b>TRIBUNAL PROYECTO DE TITULACIÓN .....</b>	<b>VI</b>
<b>DECLARACIÓN EXPRESA.....</b>	<b>VII</b>
<b>CESIÓN DE DERECHOS DE AUTOR .....</b>	<b>VIII</b>
<b>CERTIFICADO DE ACEPTACIÓN DEL TUTOR .....</b>	<b>X</b>
<b>AUTORIZACIÓN PARA PUBLICACIÓN DE PROYECTO DE TITULACIÓN EN FORMATO DIGITAL .....</b>	<b>XI</b>
<b>ÍNDICE GENERAL .....</b>	<b>XII</b>
<b>ÍNDICE DE TABLAS.....</b>	<b>XXI</b>
<b>ÍNDICE DE FIGURAS.....</b>	<b>XXII</b>
<b>ABREVIATURAS.....</b>	<b>XXIV</b>
<b>RESUMEN.....</b>	<b>XXV</b>
<b>ABSTRACT.....</b>	<b>XXVI</b>
<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO I.....</b>	<b>4</b>
<b>PLANTEAMIENTO DEL PROBLEMA .....</b>	<b>4</b>
<b>Descripción de la Situación Problemática .....</b>	<b>4</b>
Ubicación del Problema en un Contexto .....	4

Situación conflicto nudos críticos.....	5
Delimitación del problema.....	7
Evaluación del Problema .....	8
<b>Causas y consecuencias del problema .....</b>	<b>9</b>
<b>Formulación del problema .....</b>	<b>10</b>
<b>Objetivos del proyecto .....</b>	<b>10</b>
Objetivo general.....	10
Objetivos específicos .....	10
<b>Alcance del proyecto .....</b>	<b>11</b>
Justificación e importancia .....	11
<b>Limitaciones del estudio .....</b>	<b>12</b>
<b>CAPÍTULO II .....</b>	<b>13</b>
<b>MARCO TEÓRICO .....</b>	<b>13</b>
<b>Antecedentes del estudio.....</b>	<b>13</b>
<b>Fundamentación teórica.....</b>	<b>15</b>
<b>COVID-19.....</b>	<b>15</b>
<b>Rayos X (Radiografías).....</b>	<b>17</b>
¿Cómo funcionan los rayos X?.....	18
<b>Tipos de Radiografías Médicas.....</b>	<b>19</b>
Radiografía de Tórax .....	19
TC (tomografía computarizada).....	20

Presencia de Covid-19 en Tomografías computarizadas (TC) .....	20
Beneficios de los rayos X .....	20
<b>Inteligencia Artificial .....</b>	<b>21</b>
Machine Learning .....	21
Deep Learning.....	21
Modelo de predicción .....	21
<b>Tipos de Machine Learning .....</b>	<b>22</b>
Aprendizaje Supervisado .....	22
Técnica de clasificación.....	22
Aprendizaje no Supervisado .....	22
<b>Estudio e Importancia de las Redes Neuronales Artificiales .....</b>	<b>23</b>
Redes Neuronales Artificiales .....	23
<b>Clasificación de las Redes Neuronales Artificiales .....</b>	<b>23</b>
Red Neuronal Monocapa - Perceptrón simple .....	23
Red Neuronal Multicapa - Perceptrón Multicapa .....	24
Red Neuronal Convolutacional (CNN) .....	25
Estructura de la CNN .....	26
Capa de entrada.....	27
Capa Convolutacional.....	27
Capa de reducción o pooling.....	28
Capa clasificadora totalmente conectada .....	29

Capa de salida .....	29
Red neuronal Recurrente (RNN) .....	30
Algoritmo de Backpropagation (Propagación inversa) .....	31
<b>Imagen Digital .....</b>	<b>32</b>
¿Qué es una imagen digital? .....	32
Procesamiento Digital de Imágenes (PDI).....	33
Niveles de Procesamiento de Imágenes .....	33
Captación .....	33
Pre procesamiento .....	33
Segmentación.....	33
Detección de objetos y clasificación.....	33
Análisis de imagen.....	34
<b>Tipos de Imágenes Digitales.....</b>	<b>35</b>
Imágenes Binarias.....	35
Imágenes Indexadas .....	36
Imágenes RGB (Red-Green-Blue).....	36
Imágenes en Escala de Grises .....	37
<b>Tipos de Formatos de Imágenes Digitales .....</b>	<b>38</b>
Formato de Imagen JPG O JPEG .....	38
Formato de Imagen PNG .....	38
Formato de Imagen GIF.....	39

<b>¿Qué es un dataset? .....</b>	<b>40</b>
<b>Herramientas y Plataformas de Desarrollo .....</b>	<b>41</b>
Lenguaje de Programación Python .....	41
Historia.....	41
Características .....	42
Usos y Aplicaciones.....	43
Python en el desarrollo web.....	43
Python en Data Science .....	43
Python para Big Data .....	44
Python en la Inteligencia Artificial (AI) .....	44
<b>Entorno de Desarrollo en la Nube .....</b>	<b>44</b>
Google Colab .....	44
Google Drive.....	48
<b>Librerías de Machine Learning.....</b>	<b>49</b>
TensorFlow .....	49
Keras .....	49
<b>Machine Learning Aplicado al Análisis de Imágenes de Rayos X. ....</b>	<b>52</b>
Casos de estudio.....	52
<b>Revisiones sistemáticas .....</b>	<b>58</b>
Meta-análisis .....	58
<b>Preguntas científicas a contestarse .....</b>	<b>60</b>

<b>Variables de la Investigación .....</b>	<b>60</b>
<b>Definiciones conceptuales .....</b>	<b>61</b>
<b>CAPÍTULO III.....</b>	<b>64</b>
<b>METODOLOGÍA DE LA INVESTIGACIÓN.....</b>	<b>64</b>
<b>Análisis de Factibilidad .....</b>	<b>64</b>
Factibilidad Operacional.....	65
Factibilidad Técnica.....	65
Factibilidad Legal .....	66
Factibilidad Económica .....	67
<b>Tipo de Investigación.....</b>	<b>67</b>
Investigación Experimental .....	67
<b>Diseño Metodológico de la Investigación .....</b>	<b>68</b>
<b>Metodología de Investigación.....</b>	<b>69</b>
Definición del problema .....	69
<b>Fase 1.....</b>	<b>70</b>
<b>Análisis de la Data.....</b>	<b>70</b>
Descarga y Extracción de Archivos .....	70
Análisis del Dataset .....	70
Clasificación de Imágenes. ....	70
Importar Archivos a Google Drive .....	71
Conexión de Google Colab a Google Drive .....	71

Selección del Dataset .....	71
<b>Fase 2.....</b>	<b>72</b>
<b>Elaboración del Modelo.....</b>	<b>72</b>
Diseño de las Arquitecturas .....	73
Modelo A .....	73
Modelo B .....	74
Modelo C .....	75
Ajustes de Hyperparámetros .....	76
Función de pérdida (Loss) .....	76
Binary cross_entropy .....	77
Optimizador ADAM.....	78
Pre-Procesamiento y Normalización .....	78
Normalización de los datos.....	78
Data augmentation .....	78
Batch_size .....	78
<b>Fase 3.....</b>	<b>79</b>
<b>Entrenamiento.....</b>	<b>79</b>
Análisis Gráfico de Precisión y Pérdida Durante el Entrenamiento .....	82
<b>Fase 4.....</b>	<b>84</b>
<b>Evaluación de las Métricas de Calidad y Desempeño .....</b>	<b>84</b>
Matriz de Confusión .....	85

Precisión.....	86
Sensibilidad (Recall).....	86
F1-score .....	86
Pruebas de Modelos por Conjuntos de Datos .....	88
<b>Fase 5.....</b>	<b>90</b>
<b>Predicción (Ponderación y Resultados) .....</b>	<b>90</b>
<b>Beneficiarios Directos e Indirectos del Proyecto.....</b>	<b>92</b>
Beneficiarios Directos.....	92
Beneficiarios Indirectos .....	92
<b>Entregables del Proyecto.....</b>	<b>92</b>
<b>Propuesta .....</b>	<b>92</b>
<b>Criterios de Validación de la Propuesta .....</b>	<b>94</b>
Análisis de datos .....	94
<b>Resultados.....</b>	<b>94</b>
<b>CAPÍTULO IV .....</b>	<b>95</b>
<b>CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>95</b>
<b>Conclusiones .....</b>	<b>95</b>
<b>Recomendaciones .....</b>	<b>97</b>
<b>Trabajos Futuros .....</b>	<b>97</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>98</b>
<b>Anexo 1. Planificación De Actividades Del Proyecto.....</b>	<b>107</b>

<b>Anexo 2. Geo-Localización Del Problema .....</b>	<b>108</b>
<b>Anexo 3. Fundamentación Legal .....</b>	<b>109</b>
<b>Anexo 4. Validación De Expertos .....</b>	<b>112</b>
<b>Anexo 5. Manual técnico .....</b>	<b>122</b>
<b>Anexo 6. Artículo Científico.....</b>	<b>129</b>

## ÍNDICE DE TABLAS

Tabla 1. Delimitación del Problema .....	8
Tabla 2. Matriz de Causas y Consecuencias del Problema .....	9
Tabla 3. Comparativa de los Tres Tipos de Formatos de Imagen más Usados.....	39
Tabla 4. Estructura de una matriz de datos .....	40
Tabla 5. Funciones principales de Google Colab.....	45
Tabla 6. Características y componentes de una GPU .....	46
Tabla 7. Diferencias entre CPU y GPU .....	47
Tabla 8. Revisión de librerías de Machine Learning por categorías .....	51
Tabla 9. Meta-análisis .....	58
Tabla 10. Características de los ordenadores .....	65
Tabla 11. Entorno de desarrollo.....	66
Tabla 12. Presupuesto del proyecto .....	67
Tabla 13. Estructura de los conjuntos de datos .....	72
Tabla 14. Estructura de los conjuntos de datos por clases .....	72
Tabla 15. Pruebas experimentales.....	73
Tabla 16. Hyperparámetros utilizados .....	76
Tabla 17. Modelo de una matriz de confusión.....	85
Tabla 18. Resultados obtenidos por cada modelo.....	89
Tabla 19. Resultados de las predicciones.....	90
Tabla 20. Criterios de validación Expertos .....	94

**ÍNDICE DE FIGURAS**

Figura 1. Representación gráfica de la transmisión por COVID-19 .....	17
Figura 2. Representación gráfica de la toma de una radiografía.....	18
Figura 3. Radiografía del interior del tórax.....	19
Figura 4. Tomografías computarizadas del tórax.....	20
Figura 5. Estructura de un perceptrón simple .....	24
Figura 6. Estructura de una red neurona artificial.....	25
Figura 7. Ilustración de una imagen en forma de cuadrícula .....	26
Figura 8. Arquitectura de una red neuronal convolucional.....	26
Figura 9. Proceso de convolución de una imagen.....	28
Figura 10. Max-pooling .....	29
Figura 11. Función de activación sigmoid .....	30
Figura 12. Red neuronal recurrente.....	31
Figura 13. Diagrama de funcionamiento del algoritmo backpropagation.....	32
Figura 14. Niveles de procesamiento de imágenes .....	34
Figura 15. Valores de píxeles en una imagen binaria .....	35
Figura 16. Estructura de una imagen indexada .....	36
Figura 17. Formato de color RGB .....	37
Figura 18. Modelo en escala de grises .....	38
Figura 19. Proyecciones de los lenguajes de programación a futuro .....	43
Figura 20. Arquitectura de una CPU y GPU.....	48
Figura 21. Comparación de los marcos de aprendizaje profundo. ....	50
Figura 22. Comparación por búsqueda en la web .....	50

Figura 23. Diagrama de flujo del proceso.....	53
Figura 24. Arquitectura de la red neuronal convolucional.....	55
Figura 25. Diagrama de flujo de los modelos implementados en dispositivos móviles .....	55
Figura 26. Arquitectura de la red neuronal profunda VGG16 .....	56
Figura 27. Comunicación MVC que usa Laravel.....	57
Figura 28. Visualización de imágenes de rayos X de tórax .....	71
Figura 29. Arquitectura del modelo A .....	74
Figura 30. Arquitectura del modelo B .....	74
Figura 31. Arquitectura del modelo C .....	75
Figura 32. Función de pérdida .....	77
Figura 33. Resultado del entrenamiento del modelo A.....	79
Figura 34. Resultado del entrenamiento del modelo B.....	81
Figura 35. Resultado del entrenamiento del modelo C.....	82
Figura 36. Precisión y pérdida durante el entrenamiento de los modelos A y B .....	83
Figura 37. Precisión y pérdida durante el entrenamiento del modelo C .....	84
Figura 38. Modelo A.....	87
Figura 39. Modelo B.....	87
Figura 40. Modelo C.....	88
Figura 41. Imágenes de rayos X de tórax modelo A.....	91
Figura 42. Imágenes de rayos X de tórax modelo B.....	91
Figura 43. Imágenes de rayos X de tórax modelo C.....	91

**ABREVIATURAS**

ANN	Redes neuronales artificiales
CNN	Redes neuronales Convolucionales
CPU	Unidad central de procesamiento
DL	Aprendizaje Profundo
GPU	Unidad gráfica de procesamiento
IA	Inteligencia Artificial
ING	Ingeniero
ML	Aprendizaje Automático
OMS	Organización mundial de salud
PCR	Prueba de reacción en cadena de la polimerasa
SVM	Máquinas de vectores de soportes
TAC	Tomografía axial computarizada
TPU	Unidad de procesamiento tensorial
RNN	Redes neuronales recurrentes



**UNIVERSIDAD DE GUAYAQUIL**  
**FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS**  
**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES**  
**NEURONALES ARTIFICIALES**

**Autores:** Kevin Luis Morocho Sande  
C.I. N° 0931516132  
Danny Xavier Pulig Cesén  
C.I. N° 0930417753

**Tutor:** Ing. Darwin Patiño Pérez M.Sc.

**RESUMEN**

En la actualidad, las pruebas más efectivas para la detección de Covid-19 como el análisis de imágenes por tomografía axial computarizada (TAC) y las evaluaciones médicas de laboratorio como la prueba de reacción en cadena de la polimerasa (PCR), son uno de los métodos más efectivos para el diagnóstico de esta enfermedad. La única desventaja es que estos métodos tradicionales son muy costosos y eso implicaba que la población de clase media-baja no tenga acceso a estas pruebas de detección. Las redes neuronales artificiales juegan un papel muy importante en el campo de la medicina y en investigaciones tecnológicas que contribuyan a la detección de enfermedades como el Covid-19. Este proyecto se centralizó con el fin de proporcionar un apoyo a los médicos en la toma de decisiones, utilizando herramientas tecnológicas. Se realizó un modelo de machine learning para el análisis de imágenes de rayos X para la detección de Covid-19 a través de redes neuronales convolucionales. El algoritmo fue realizado en el lenguaje de programación Python con el uso del entorno de desarrollo en la nube de Google Colaboratory. Por medio de los repositorios de acceso público de GitHub y Kaggle, se recolectaron bases de datos de imágenes de rayos X de tórax de pacientes con Covid-19 y pacientes normales (sanos) para generar un dataset con imágenes de entrenamiento y validación. Se realizaron pruebas experimentales con tres tipos de arquitectura de redes neuronales convolucionales para los modelos A, B y C. Siendo el modelo C el que obtuvo los mejores resultados.

**Palabras clave:** Covid-19, rayos X, machine learning, modelo de predicción, redes neuronales convolucionales.



**UNIVERSIDAD DE GUAYAQUIL  
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS  
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**X-RAY IMAGING ANALYSIS OF COVID-19 THROUGH ARTIFICIAL NEURONAL  
NETWORKS**

**Authors:** Kevin Luis Morocho Sande  
C.I. N° 0931516132  
Danny Xavier Pulig Cesén  
C.I. N° 0930417753

**Tutor:** Ing. Darwin Patiño Pérez M.Sc.

**ABSTRACT**

Currently, the most effective tests for the detection of Covid-19 such as computed axial tomography (CT) image analysis and laboratory medical evaluations such as the polymerase chain reaction (PCR) test, are one of the most effective methods for the diagnosis of this disease. The only disadvantage is that these traditional methods are very expensive and that meant that the lower-middle class population did not have access to these screening tests. Artificial neural networks play a very important role in the field of medicine and in technological research that contributes to the detection of diseases such as Covid-19. This project was centralized in order to provide support to doctors in decision-making, using technological tools. A machine learning model was performed for the analysis of X-ray images for the detection of Covid-19 through convolutional neural networks. The algorithm was carried out in the Python programming language with the use of the Google Colaboratory cloud development environment. Through the public access repositories of GitHub and Kaggle, databases of chest X-ray images of patients with Covid-19 and normal (healthy) patients were collected to generate a dataset with training and validation images. Experimental tests were carried out with three types of convolutional neural network architecture for models A, B and C. Model C being the one that obtained the best results.

**Key words:** Covid-19, X-ray, machine learning, prediction model, convolutional neural networks.

## INTRODUCCIÓN

El descubrimiento del nuevo coronavirus 2019 (Covid-19), es una enfermedad con un alto grado de contagio que se propaga de manera muy fácil y rápida en todo el mundo, afecta la salud del ser humano produciendo infecciones en los pulmones (neumonía), problemas para respirar, fatiga y cansancio, es la principal causa de muerte en pacientes que presentan esta enfermedad. China fue el primer país en donde se reportó una gran cantidad de muertes provocado por el contagio del virus. Brasil fue el segundo país más afectado reportando tres millones de casos confirmados y más de 100.000 muertes en los primeros días del mes de agosto (Grando et al., 2020).

En varias industrias de trabajo el uso de una herramienta tecnológica se ha vuelto muy importante para resolver una gran variedad de problemas con el fin de brindar soluciones ante enfermedades infecciosas. Específicamente en el campo de la medicina los doctores usan imágenes médicas creadas por medio de rayos X para detectar y tratar varios tipos de enfermedades o dolores que se presentan en el cuerpo humano, tales como, infecciones en los pulmones, huesos rotos, tumores, masas anormales, problemas dentales, etc.

El presente trabajo de titulación está enfocado en realizar el análisis de imágenes de rayos X mediante un modelo de machine learning para la predicción de Covid-19 a través de redes neuronales convolucionales, la misma que servirá como herramienta tecnológica alternativa a las pruebas médicas de diagnóstico tradicionales. Con la ayuda de Google Colab como entorno gratuito de Jupyter Notebook se ejecuta en la nube porque permite desarrollar código Python y proporciona entornos de ejecución tales como CPU, GPU y TPU.

Con la ayuda de un modelo de predicción se puede detectar si un paciente se encuentra sano o enfermo de Covid-19 por medio de una radiografía torácica. Se lo realiza por medio del big data (grandes cantidades de imágenes radiográficas) junto con el Deep learning un subcampo del aprendizaje automático, mediante el uso de redes neuronales artificiales en conjunto con técnicas de clasificación. Este trabajo de estudio se llevó a cabo por investigadores del CVB Lab, de la Universidad Politécnica de Valencia (Marcos, 2020).

Mediante el uso de las redes neuronales convolucionales, principal algoritmo utilizado en machine learning sirve para la clasificación de imágenes, detección de objetos, reconocimiento de voz, entre otros. Utiliza métricas de calidad que sirven para indicar que tan bueno o malo es el rendimiento del modelo de predicción, son datos numéricos que podrían variar dependiendo del número de capas y neuronas que tenga la arquitectura, el número de épocas o ciclos de aprendizaje del entrenamiento, matriz de confusión, verdaderos positivos, verdaderos negativos, falsos positivos, falsos negativos, precisión, recall (exhaustividad), F1-score y accuracy (exactitud).

La estructura del proyecto de investigación está formada por 4 capítulos que se detallan a continuación:

**Capítulo I:** Se describe el planteamiento del problema de la investigación, seguido de las causas y consecuencias presentes en el proyecto, formulación del problema, objetivos generales y específicos, alcance, justificación y limitaciones del estudio.

**Capítulo II:** Se encuentra el marco teórico en donde se detallan los temas más relevantes del proyecto de investigación, antecedentes de estudio, fundamentación teórica, pregunta científica y definiciones conceptuales.

**Capítulo III:** Es donde se va a elegir la metodología propuesta para realizar el trabajo de investigación seguido del tipo de investigación, diseño metodológico, criterios de validación y resultados obtenidos.

**Capítulo IV:** Por último, se presentan las conclusiones, recomendaciones y trabajos a futuros.

## **CAPÍTULO I**

### **PLANTEAMIENTO DEL PROBLEMA**

#### **Descripción de la Situación Problemática**

##### **Ubicación del Problema en un Contexto**

Para la comprensión del estudio y el uso de las tecnologías como contribución a la resolución de problemas en los que representa la detección de enfermedades raras es necesario conocer la importancia que estas tienen dentro del campo de la medicina. Una de ellas es la inteligencia artificial, el uso de algoritmos y machine learning.

La inteligencia artificial (IA) es una tecnología que se basa en el modelo de las redes neuronales del cerebro humano, ya que utiliza numerosas capas de información y conexiones que le permite detectar coincidencias de patrones, reglas, aprendizaje profundo mediante la computación cognitiva con el fin de ayudar a aprender y comprender los datos (IBM, s. f.).

Estos datos pueden ser interpretados a través de algoritmos de aprendizaje de máquina con el objetivo de automatizar los procesos manuales que son difíciles de interpretar para los seres humanos.

Un algoritmo es un procedimiento bien definido que permite que una computadora resuelva un problema. Otra forma de describir un algoritmo es una secuencia de instrucciones inequívocas (Robledano, 2019).

La función principal de los algoritmos de inteligencia artificial es que están delineados para la toma de decisiones. Los algoritmos asiduamente utilizan datos en tiempo real y poseen muchas diferencias con respecto a las máquinas pasivas que solo brindan respuestas específicamente de forma mecánica. Los algoritmos de inteligencia artificial combinan los

datos de diversas fuentes y examinan los componentes inmediatamente, los cuales actúan de apoyo con base a los conocimientos derivados de la información. Las mejoras en los sistemas de almacenamiento, la velocidad de procesamiento en conjunto con técnicas analíticas permite una enorme sofisticación en el análisis y toma de decisiones (West & Allen, 2018).

Con la ayuda de los algoritmos de aprendizaje automático las máquinas tienen la capacidad de reconocer los patrones dentro de una imagen médica. Para que los algoritmos aprendan correctamente es necesario una gran cantidad de imágenes con el fin de obtener un resultado esperado al momento de clasificarlos.

La utilidad del aprendizaje automático radica en que las imágenes se encuentren digitalizadas para ser procesadas con el objetivo de detectar cualquier tipo de enfermedades a través de un diagnóstico mediante la visión por computadora. Existen varios tipos de visión por computadora tales como: radiografías de tórax, tomografías computarizadas, imágenes de la piel, imágenes de resonancia magnética cardíaca, etc. Los algoritmos de aprendizaje automático tienden a realizar predicciones que servirán como base a los expertos para generar un diagnóstico sugiriendo el tratamiento adecuado (Schmitt, s. f.).

La elaboración del modelo de machine learning para la predicción de pacientes con Covid-19 por medio de imágenes de rayos X a través de redes neuronales convolucionales servirá como herramienta tecnológica alternativa a las pruebas de diagnóstico tradicionales. Las principales características de los exámenes de rayos X son: efectividad, asequibilidad, coste reducido, rapidez e imperceptibilidad. Por estas razones las personas de bajos recursos optan por este método de diagnóstico debido a su rendimiento y aplicación.

### **Situación conflicto nudos críticos**

Con el transcurso del tiempo se presentaron diversas situaciones respecto a la pandemia en diferentes lugares del mundo. El problema surge a partir de la preparación y la falta de conocimiento que tuvieron los países ante el estado de emergencia del Covid-19. En

consecuencia, se vieron afectados los sistemas de salud y el sector socioeconómico. Ante la problemática generada señalaremos los siguientes factores:

La no disposición de herramientas tecnológicas como modelos computacionales para la predicción de Covid-19 mediante el análisis de imágenes de rayos X.

La falta de métodos para diagnosticar la enfermedad. Poco después se aplicó el método de detección a través de tomografía axial computarizada (TAC).

El alto costo de las TAC implicaba que no todos los usuarios tenían acceso a una prueba ya que el tiempo estimado para realizar estas radiografías en 3D es alrededor de 10 a 30 minutos. Para comprender el uso de los métodos diagnósticos y la importancia de la tecnología en la medicina se menciona lo siguiente:

El TAC tiene una precisión del 97-98%, ya que con ella se obtiene una imagen en 3D del pulmón y se puede diagnosticar la enfermedad sin margen de error alguno, el único problema es que estos dispositivos de imágenes sofisticados solo se encontraban en hospitales grandes, y por lo tanto los hospitales y casas de salud más pequeños no tenían acceso a estos sistemas de detección (Jáuregui & Herrera, 2020).

Alrededor del tiempo se implementaron nuevos métodos diagnósticos tales como las pruebas PCR. Las pruebas de reacción en cadena de la polimerasa (Polymerase Chain Reaction, PCR) o mejor conocida como la prueba de amplificación de ácidos nucleicos (Nucleic acid amplification test, NAAT). Es un tipo de prueba molecular viral, la cual consiste en realizar un hisopado nasal o bucal, la cual sirve para hallar y extraer el material genético del virus. En la actualidad se considera como la prueba más efectiva para la detección de Covid-19 (*Información Sobre Los Distintos Tipos de Pruebas de COVID-19* / SCDHEC, 2019).

Los médicos del hospital de las fuerzas armadas en Guayaquil – Ecuador consideran lo siguiente ”entre sus desventajas se indica que deben ser realizadas por expertos en el área de microbiología molecular, es costoso, requiere de equipos y personal específico, tiene un porcentaje relativo de falsos positivos”(Meza et al., 2020).

Ante la ausencia de métodos rápidos para el diagnóstico de Covid-19 la población de la clase media-baja se arriesgaba por una consulta médica tradicional. El efecto de estas medidas ocasionó la saturación de los sistemas de salud aumentando el riesgo de contagio.

La contribución del *machine learning* a la medicina servirá para crear modelos computacionales basados en técnicas de aprendizaje automático con el fin de clasificar e identificar correctamente el caso de una persona sana con una infectada. Al hacer una clasificación correcta se prevé disminuir la propagación del virus.

Las redes neuronales convolucionales utilizan capas de entrada para captar, memorizar y extraer patrones encontrados en imágenes. Se encarga de asignar una clase determinada un valor ponderativo para obtener un resultado. Con esto se evita los altos costos por pruebas de Covid-19, el congestionamiento en los centros de salud y la ralentización de los procedimientos médicos para la toma de decisiones.

### **Delimitación del problema**

Como medida de apoyo se requiere el manejo de herramientas tecnológicas que agilicen los procesos para analizar el estado de los usuarios detectados por Covid-19 y clasificarlos con una etiqueta correspondiente para saber si están infectados o no.

Anteriormente se consideró un tiempo aproximado de 5 meses entre diciembre de 2019 y abril de 2020, siendo esta época el punto de inflexión donde la pandemia aumentó y se extendió a todos los países del mundo, por los miles de casos confirmados y posibles contagios; al aplicar estas medidas de prevención se pretende disminuir el riesgo de contagios y la tasa de mortalidad

**Tabla 1.***Delimitación del Problema*

<b>Delimitador</b>	<b>Descripción</b>
Campo	Tecnología
Área	Servicios
Aspecto	Predicción por medio de imágenes de rayos X.
Tema	Análisis de imágenes de rayos X de Covid-19 a través de redes neuronales artificiales.

*Nota.* Datos de la investigación. La elaboración es propia

**Evaluación del Problema**

- **Delimitado:** El aprendizaje de nuevos métodos para el desarrollo de modelos de redes neuronales y el análisis de datos simula algo complejo a la hora de realizar el proyecto.
- **Claro:** Elaborar un modelo de predicción para el análisis de imágenes de rayos x para Covid-19 a través de redes neuronales convolucionales con el fin de reducir el tiempo en la entrega de resultados médicos.
- **Concreto:** Conocer los resultados de la precisión del modelo para realizar la predicción y ofrecer un diagnóstico oportuno basado en aprendizaje profundo.
- **Relevante:** Clasificar correctamente a un paciente sano de uno infectado disminuyendo la propagación del virus y de casos positivos.
- **Factible:** La computación en la nube y el uso de librerías de alto nivel para la creación de modelos simula una ayuda extra para los usuarios que no poseen computadores potentes para la ejecución de múltiples tareas. Una de sus ventajas es: GOOGLE COLAB es un entorno gratuito de Jupyter Notebook que no requiere configuración y que se ejecuta completamente en la nube. Jupyter es un entorno interactivo que nos permite desarrollar código Python de manera dinámica, la

máquina en un inicio cuenta con 12 GB de RAM y 50 GB de almacenamiento en disco disponibles para el uso con tres tipos de unidad de procesamiento como CPU, GPU Y TPU (De la fuente, 2019).

- **Evidente:** Evaluar la función objetivo para minimizar el error considerando la utilización de hiperparámetros para ajustar y mejorar el modelo.
- **Variabes:** Las variables de entrada del modelo serán las imágenes del conjunto de datos clasificados en dos tipos: Covid-19 y Normal. Estas variables serán cualitativas.

### Causas y consecuencias del problema

Las causas y consecuencias del problema se describen en la tabla 2.

**Tabla 2.**

*Matriz de Causas y Consecuencias del Problema*

Causas	Consecuencias
C1. El tiempo de espera para atenciones médicas es muy prolongado debido al aumento de personas contagiadas.	E1. Hospitales congestionados.
C2. Crisis económicas en sectores laborales.	E2. Reducción de empleados en todos los sectores públicos y privados del país.
C3. Escasez de trabajo.	E3. Recursos económicos limitados para subsistir.
C4. Contagios de persona a persona	E4. Dificultades en el sistema respiratorio provocando la muerte.
C5. Cierre total de establecimientos educativos.	E5. La falta de recursos tecnológicos para clases virtuales impide el desarrollo sostenible académico de niños y jóvenes.
C6. Reducción en los precios del petróleo.	E6. Afecta directamente a las ventas y exportaciones que ofrece el país a nivel internacional.

*Nota.* Datos de la investigación. La elaboración es propia

## **Formulación del problema**

¿Cuál será el algoritmo más adecuado de redes neuronales artificiales para la elaboración de un modelo de machine learning en la detección de Covid-19 a través de imágenes de rayos X?

## **Objetivos del proyecto**

### **Objetivo general**

- Elaborar un modelo de machine learning para la predicción de pacientes con Covid-19 por medio del análisis de imágenes de rayos X a través de redes neuronales artificiales.

### **Objetivos específicos**

1. Elaborar un marco teórico en el que se refleje el estudio más apropiado que involucre usar redes neuronales artificiales para la detección de Covid-19 por medio de un análisis de imágenes de rayos X.
2. Implementar un modelo de redes neuronales artificiales mediante algoritmos codificados en el lenguaje Python para identificar o detectar Covid-19.
3. Identificar y obtener bases de datos de imágenes de rayos x y/o realizar recolección de imágenes para generar el dataset de entrenamiento y dataset de testing.
4. Realizar experimentos y validar los resultados obtenidos para determinar que pacientes son identificados con Covid-19 a través del análisis de imágenes de rayos X.
5. Elaborar un artículo científico tomando como referencia el modelo de redes neuronales convolucionales para su publicación.

### **Alcance del proyecto**

- Revisión y comprensión del tema acerca de los algoritmos de redes neuronales artificiales.
- Implementar un modelo de identificación de imágenes de rayos x mediante el uso de redes neuronales convolucionales usando Python.
- Entrenar el modelo utilizando imágenes de rayos X para la detección y predicción de Covid –19 a través de redes neuronales convolucionales.
- Validar los resultados obtenidos para deducir las conclusiones del modelo de predicción para pacientes con Covid – 19.

### **Justificación e importancia**

Para realizar el presente trabajo de investigación, se debe tomar en cuenta que el coronavirus proviene de una gran gama de virus, quienes causan enfermedades tanto en animales como en seres humanos.

El coronavirus causa infecciones respiratorias muy parecidas a un resfriado común hasta enfermedades más peligrosas como es el Síndrome Respiratorio de Oriente Medio (MERS) y el Síndrome Respiratorio Agudo Severo (SARS).

El virus denominado como Covid-19 es muy letal y peligroso en algunos pacientes porque deja secuelas de la infección en los órganos del cuerpo humano y en otros casos produciendo la muerte. Actualmente el Covid-19 es una pandemia que afecta a muchos países de todo el mundo.

La propagación del virus se transmite de persona a persona a través de gotículas respiratorias esparcidas por medio de la tos o por individuos que presenten síntomas como fiebre o cansancio.

La finalidad de este proyecto se basa en la detección del virus mediante imágenes de rayos x, con la capacidad de diferenciar entre un paciente infectado y un paciente sano.

El modelo de machine learning para la predicción de pacientes con Covid-19, servirá como herramienta de apoyo a los médicos para solventar y disminuir el gran abastecimiento de tantos casos positivos. Ofrecerá una ayuda inmediata para determinar que pacientes están sanos o enfermos realizando un análisis con el menor tiempo posible.

A través del modelo se pretende agilizar los procedimientos realizados mediante pruebas y evaluaciones a las personas que lo requieran de manera rápida con resultados eficientes, esto se da con el fin de evitar el congestionamiento en los hospitales y centros de salud.

### **Limitaciones del estudio**

1. Estará basado en tesis de grado, artículos científicos, sitios web.
2. Se utilizará Google Colab como entorno de desarrollo gratuito en la nube para la elaboración del modelo en código Python.
3. Recursos y equipo computacional limitado.
4. Las bases de datos de imágenes de rayos X serán recolectados a través de repositorios de acceso público totalmente gratuitos proporcionada por médicos y científicos de datos.
5. El modelo de machine learning será exclusivamente genérico.
6. No estará enfocado a una población determinada.
7. No se realizarán pruebas funcionales dentro de las instituciones de salud.

## CAPÍTULO II

### MARCO TEÓRICO

#### Antecedentes del estudio

Los coronavirus (CoV) son una gran familia de virus que causan enfermedades que van desde el resfriado común hasta enfermedades más graves. La epidemia de Covid-19 fue declarada por la OMS una emergencia de salud pública de preocupación internacional el 30 de enero de 2020. El Director General de la Organización Mundial de la Salud (OMS), el doctor Tedros Adhanom Ghebreyesus, anunció el 11 de marzo de 2020 que la nueva enfermedad por el coronavirus 2019 (Covid-19) puede caracterizarse como una pandemia. La caracterización de pandemia significa que la epidemia se ha extendido por varios países, continentes o todo el mundo, y que afecta a un gran número de personas.(OMS, 2020).

La consecuencia de este problema, es la deficiencia del sistema de salud y con ello la saturación de los hospitales por la demanda de pacientes que requieren atención. Por ello los especialistas de la salud hacen hincapié en la necesidad de nuevas herramientas de diagnósticos, test rápidos que permitan conocer la afección real de la infección por Covid-19 (SARS-COV-2), para que de esta forma influya tanto en el seguimiento de las normas de seguridad y el distanciamiento social para disminuir la propagación.(ISGLOBAL, 2020).

Los métodos vigentes para diagnosticar el Covid-19 en la actualidad son los siguientes.

Las **pruebas serológicas de anticuerpos**, como su nombre lo indica, este tipo de prueba tiene como objetivo detectar anticuerpos que el organismo contiene para combatir el virus a través de un análisis en la sangre con el fin de determinar si un individuo

precisa de una infección o ha poseído el virus del Covid-19 (Centro para el Control y la Prevención de enfermedades, 2020). Se conoce por medio de los estudios realizados, donde se indica que la sensibilidad para estos test serológicos rápidos (Prueba de anticuerpos combinados IgG-IgM de Covid-19 son variables, observando sensibilidad del 88,6% y especificidad del 90,63% (Rocha, 2020).

El **método de diagnóstico o test de antígenos**, se utiliza para reconocer el revestimiento de las proteínas del virus, mientras que por otra parte las PCR solo identifican el material genético que está hospedado en el mismo. Se realiza por medio de un hisopado nasofaríngeo para extraer las muestras que son depositadas en un kit de análisis, donde se observan los resultados en un corto periodo de tiempo de 15 a 30 minutos por medio de un sistema colorimétrico. En cuanto se refiere a la especificidad, el test arroja un valor aproximado de 99.68% para pacientes evaluados con síntomas, lo cual indica que está a la par con la PCR, la única diferencia es que puede obtener presuntamente cifras negativas, tomando como consideración un juicio clínico se descartara mediante una prueba molecular (Coello Cristina & Nuñez Byron, 2020).

La **tomografía axial computarizada (TAC)** utiliza una tecnología sofisticada de rayos X, es una prueba de diagnóstico por imágenes que permite capturar en varios tomos las diferentes partes del cuerpo, los combina en una computadora y los muestra en una imagen detallada de 3 dimensiones. Permite crear imágenes con detalles de los órganos, huesos y vasos sanguíneos. Permiten conocer, detectar y confirmar la presencia de cáncer, un tumor o alguna enfermedad respiratoria (*Exploración Por Tomografía Computarizada (TC) | Cancer.Net*, 2018).

Según los estudios publicados por la revista 'Radiology', indican que el diagnóstico por tomografía computarizada es mejor que las pruebas de laboratorio para la detección de Covid-19. Este estudio basó su investigación y fue aplicado a 1000

pacientes. Lo que demuestra que las TC a comparación de las pruebas PCR, son más eficientes, prácticas y fiables para analizar la infección por el virus. Se demostró que la sensibilidad fue del 98% para la infección por Covid-19. Luego para un nuevo estudio sobre 1014 personas determinó obtener un 88% de casos positivos para imágenes de tórax y aumentando al 97% en casos asociados a PCR positivo (Press, 2020).

El modelo de predicción está basado en un análisis de clasificación de clases, que le permite conocer los resultados de un conjunto de datos a través de una red neuronal convolucional. Los modelos de predicción utilizan uno o más clasificadores para intentar determinar la probabilidad de un conjunto de datos pertenecientes a otro conjunto.

El modelo de análisis de imágenes de rayos X mediante redes neuronales convolucionales utiliza aprendizaje supervisado, este método de aprendizaje fue seleccionado con base a las necesidades de la investigación.

Para determinar el tipo del método de aprendizaje se añade lo siguiente:

El aprendizaje supervisado se caracteriza por el proceso de aprendizaje que se realiza mediante un entrenamiento controlado por un supervisor que determina la respuesta que se debe generar para cada entrada. El supervisor controla la salida y si esta no es correcta, modifica los pesos de las conexiones, con el fin de que la salida obtenida se aproxime a la deseada (Calvo, 2017a).

### **Fundamentación teórica**

#### **COVID-19**

En la actualidad el mundo entero se encuentra viviendo con una enfermedad muy grave que afecta directamente a la salud del ser humano. El epicentro del brote de esta enfermedad se originó en China en los mercados populares de la ciudad de Wuhan en la provincia de Hubei. En el mes de diciembre del 2019 se detectó la presencia de un nuevo virus causante de una neumonía muy letal. El gobierno chino tomó acciones para controlar la

enfermedad que ya era noticia a nivel internacional. Para el día 7 de enero del 2020 los científicos chinos tenían en observación el primer caso de un paciente con sospechas de esta rara enfermedad. Debido a este problema se realizó el aislamiento de personas que presenten síntomas de neumonía. Estas investigaciones servirían para determinar si más personas habrían sido contagiadas por este nuevo virus (Wang et al., 2020).

El nuevo coronavirus que ha surgido es el responsable de causar enfermedades de resfriado común o casos de neumonía severa. “Este virus ahora se conoce como el síndrome respiratorio agudo grave coronavirus 2 (SARS-CoV-2). La enfermedad que causa se llama enfermedad del coronavirus 2019 (Covid-19)” (*Enfermedad Del Coronavirus 2019 (COVID-19) - Síntomas y Causas - Mayo Clinic, 2020, párrafo segundo*).

Las personas de todas las edades pueden contagiarse con coronavirus, pero afecta principalmente a personas mayores de 70 años que padezcan de asma, diabetes y enfermedades cardíacas. Esta población es estrechamente vulnerable en vista de que los casos reportados registran una tasa de muerte mayor al 8%, donde la mayoría fueron pacientes con 56 años de edad quienes presentaban enfermedades cardíacas o cerebro vascular (Palacios Cruz et al., 2020).

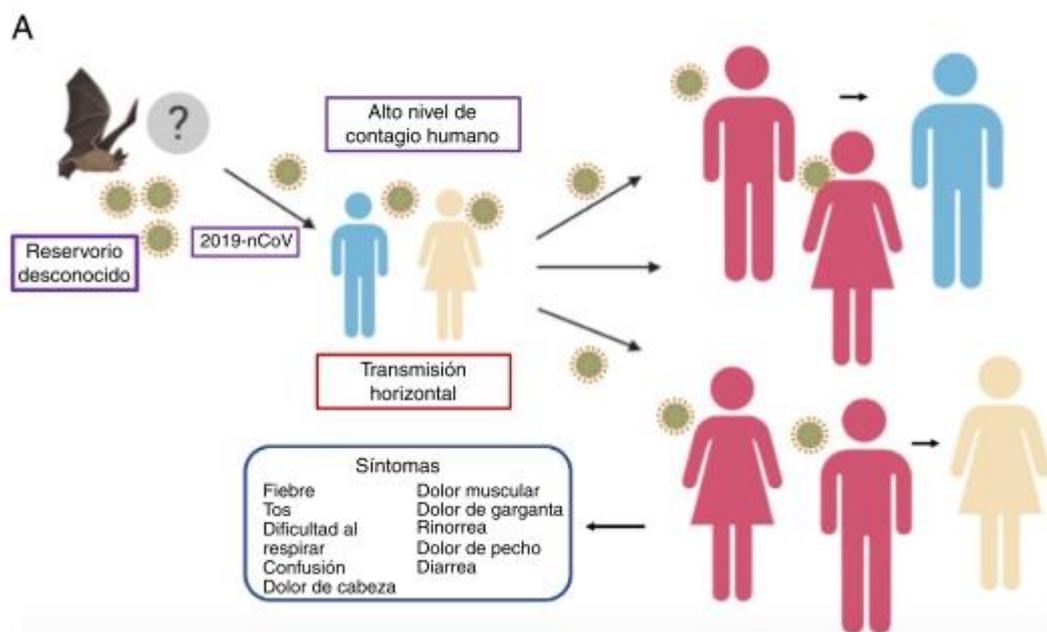
El virus en más de 160 países ha llegado a tener una tasa de mortalidad de 8000 muertes, considerando que los casos confirmados por Covid-19 se aproximan a 200.000, por tal motivo la organización mundial de la salud (OMS), ha declarado que se lo determine como una pandemia. Después de que en Wuhan y China lugar en donde se inició el brote del virus, el país de Italia ubicado en Europa ha sido el más afectado ante esta situación, en solo 2 semanas el virus se propagó rápidamente, por lo cual 1000 pacientes resultaron positivos al coronavirus (Spinelli & Pellino, 2020).

Los síntomas llegan aparecer entre cinco días o dos semanas después de haberse contagiado con el virus, durante ese periodo de tiempo en que se encuentra el paciente con

los síntomas se lo denomina proceso de incubación. Afecta al sistema respiratorio apareciendo tos, gripe, fiebre, cansancio, vómito, dificultad para respirar, dolores de cabeza, pérdida del gusto y el olfato, diarrea, escalofríos y en algunos casos puede llegar a ocasionar la muerte. Los médicos recomiendan lavarse las manos con abundante jabón constantemente, usar desinfectante para manos, usar mascarilla y mantener un metro de distancia de las demás personas para evitar el contagio y descartar cualquier problema de salud.

### Figura 1.

*Representación gráfica de la transmisión por COVID-19*



*Nota.* El Covid-19 proviene de los murciélagos. Todo empieza con la formulación de la hipótesis que los científicos han investigado con respecto a la transmisión de animal a humano. A este proceso se le denomina zoonosis. Tomado de (Palacios Cruz et al., 2020).

### Rayos X (Radiografías)

Los rayos X producen radiación y emiten energía parecida a la luz y ondas electromagnéticas, al momento de hacer contacto con el cuerpo humano es absorbida por los tejidos, órganos internos y huesos. Con la ayuda de un monitor similar al de una computadora se pueden crear imágenes médicas, permitiendo al radiólogo como especialista encargado del

área observar anomalías dentro de la estructura del cuerpo (*Seguridad Del Paciente: Dosis de Radiación En Exámenes de Rayos X y TC*, s. f.).

Para la creación de imágenes se utiliza una película fotográfica que es la encargada de detectar los rayos X, por lo tanto, una vez realizado el proceso las imágenes de rayos X que se obtienen se las conocen con el nombre de radiografías (*Rayos X*, s. f.).

### ¿Cómo funcionan los rayos X?

Una radiografía se crea ubicando al paciente en la máquina y la parte del cuerpo que se va a examinar debe colocarse entre la fuente y el detector de rayos X. Cuando los rayos X atraviesan el cuerpo, los tejidos absorben la radiación en diferentes cantidades (*Rayos X*, s. f.).

La mayor cantidad de rayos X es absorbido por el calcio que se encuentra en los huesos, por ese motivo las radiografías se observan de color blanco, mientras que en los pulmones, tejidos y grasa, se absorbe una cantidad mínima de radiación y el color que se visualiza es de color negro o gris (*Rayos X: MedlinePlus En Español*, s. f.).

### Figura 2.

*Representación gráfica de la toma de una radiografía*



**Nota.** El paciente se encuentra en una posición vertical ubicado en la máquina de rayos X para la toma de una radiografía del tórax. Tomado de (*Rayos X*, s. f.-b).

## Tipos de Radiografías Médicas

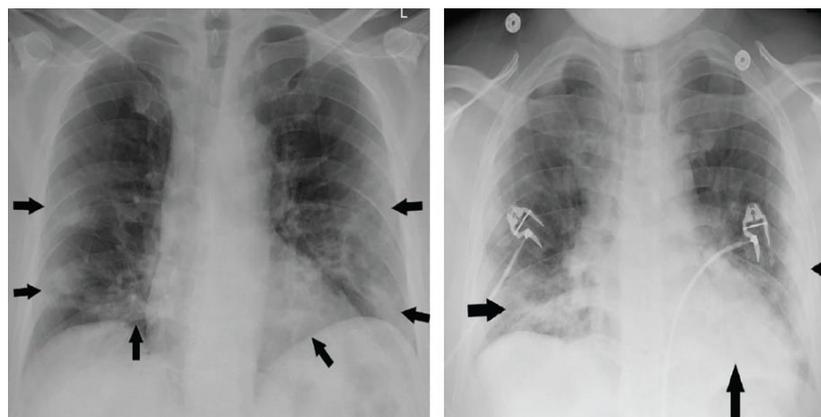
La radiología es una rama de la medicina que utiliza la tecnología imagenológica para diagnosticar y tratar una enfermedad. Los tipos más comunes de exámenes radiológicos de diagnóstico incluyen: radiografía de tórax, mamografía, (TC) tomografía computarizada, Fluoroscopia.

### Radiografía de Tórax

Las imágenes de radiografía de tórax, son muy fáciles y rápidas de obtener ya que sirven de gran ayuda en casos de emergencia. Generan imágenes del interior del tórax utilizando muy poca radiación para realizar una revisión completa de los pulmones, corazón y pared torácica. Por medio de estas pruebas se puede realizar una valoración médica sobre el tipo de enfermedad que padece una persona para brindar un tratamiento adecuado y seguro. Síntomas como fiebre, problemas respiratorios, tuberculosis, molestias en el pecho, neumonía y cáncer, son los casos más comunes que se pueden presentar (*Radiografía de Tórax (Radiografía)*, 2020).

#### Figura 3.

*Radiografía del interior del tórax*



**Nota.** Pruebas radiográficas realizadas a pacientes con Covid-19 para determinar en qué sectores del pulmón se encuentran las lesiones provocadas por dicha enfermedad. Tomado de (Yadav et al., 2020).

## TC (tomografía computarizada)

La tomografía computarizada permite ver la estructura interna del cuerpo más detalladamente a diferencia de las radiografías simples, estas se generan en diferentes ángulos permitiéndole al médico obtener un dictamen más claro (*Rayos X*, s. f.-b).

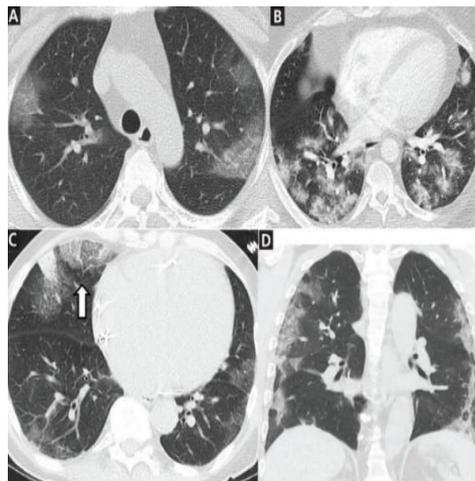
### *Presencia de Covid-19 en Tomografías computarizadas (TC)*

Según la valoración del especialista es importante mencionar lo siguiente:

“Los pulmones son el sitio más común de infección en COVID-19, y la progresión a insuficiencia respiratoria es la causa más común de muerte” (Yadav et al., 2020).

#### **Figura 4.**

##### *Tomografías computarizadas del tórax*



*Nota.* Pruebas de TC realizadas a diferentes pacientes con COVID-19 que muestran varias lesiones dentro de los pulmones. Tomado de (Yadav et al., 2020).

## **Beneficios de los rayos X**

Las pruebas de radiografías se encuentran disponibles en cualquier centro de salud, salas de emergencia y otros lugares. Son accesibles, económicas y rápidas de realizar para una atención inmediata (*Radiografía de Tórax (Radiografía)*, 2020).

## **Inteligencia Artificial**

La inteligencia artificial (IA) se refiere a la simulación de la inteligencia humana en máquinas que están programadas para pensar como humanos e imitar sus acciones. El término también se puede aplicar a cualquier máquina que exhiba rasgos asociados con una mente humana, como el aprendizaje y la resolución de problemas (Tablado, 2020).

### **Machine Learning**

El aprendizaje automático es un método de análisis de datos que automatiza la construcción de modelos analíticos. Es una rama de la inteligencia artificial basada en la idea de que los sistemas pueden aprender de los datos, identificar patrones y tomar decisiones con una mínima intervención humana (*Qué Es Machine Learning, Cómo Funciona y a Qué Se Aplica* / APD, 2019).

### **Deep Learning**

El aprendizaje profundo es una función de inteligencia artificial (IA) que imita el funcionamiento del cerebro humano en el procesamiento de datos y la creación de patrones para su uso en la toma de decisiones. El aprendizaje profundo es un subconjunto del aprendizaje automático en inteligencia artificial que tiene redes capaces de aprender sin supervisión a partir de datos que no están estructurados o etiquetados. También conocido como aprendizaje neuronal profundo o red neuronal profunda (*¿Qué Es El Deep Learning?* | SmartPanel, s. f.).

### **Modelo de predicción**

El análisis predictivo se refiere al uso de datos históricos, aprendizaje automático e inteligencia artificial para predecir lo que sucederá en el futuro. Estos datos históricos se introducen en un modelo matemático que considera tendencias y patrones clave en los datos. (Velazquez, 2020).

## Tipos de Machine Learning

Dentro del machine learning existen dos tipos de aprendizaje, el supervisado y no supervisado, a continuación, se detallan los siguientes:

### Aprendizaje Supervisado

El aprendizaje supervisado es una técnica que consiste en proporcionar datos etiquetados a un modelo de aprendizaje automático. El conjunto de datos etiquetado generalmente son datos recopilados de la experiencia, también llamados datos empíricos. Además, los datos a menudo requieren preparación para aumentar su calidad, llenar sus vacíos o simplemente optimizarlos para la capacitación (*Qué Son Regresión y Clasificación En Machine Learning*, 2020).

El aprendizaje supervisado se puede dividir en 2 categorías: clasificación y Regresión.

#### Técnica de clasificación

La clasificación es un proceso de categorizar un determinado conjunto de datos en clases. Se puede realizar tanto en datos estructurados como no estructurados. El proceso comienza con la predicción de la clase de puntos de datos dados. Las clases a menudo se denominan destino, etiqueta o categorías. Existen dos tipos de clasificación: binaria y multiclase.

**Clasificación Binaria:** Es un tipo de clasificación en el que tan solo se pueden asignar dos clases diferentes (0 o 1), falso o verdadero, anormal y normal.

### Aprendizaje no Supervisado

El aprendizaje no supervisado, también conocido como aprendizaje automático no supervisado, utiliza algoritmos de aprendizaje automático para analizar y agrupar conjuntos de datos sin etiquetar. Estos algoritmos descubren patrones ocultos o agrupaciones de datos sin necesidad de intervención humana (*Tipos Del Algoritmo de Aprendizaje Automático — DATA SCIENCE*, 2020).

## **Estudio e Importancia de las Redes Neuronales Artificiales**

En el siguiente contexto se describen los temas relacionados con las redes neuronales artificiales (ANN), las mismas que juegan un papel muy importante en el campo de la medicina y en investigaciones tecnológicas que contribuyan a la detección de enfermedades, por ejemplo, Covid-19 por lo que a continuación se exponen los siguientes temas:

### **Redes Neuronales Artificiales**

Las redes neuronales artificiales (también conocidas como sistemas conexionistas) son modelos computacionales formados por un conjunto de nodos conocidos como neuronas artificiales quienes transmiten señales y están interconectadas entre sí. Estas señales se transmiten desde la entrada hasta generar una salida. Las redes neuronales artificiales tienen como objetivo imitar el comportamiento del cerebro humano para resolver problemas complejos basados en datos (Delgado et al., 2019).

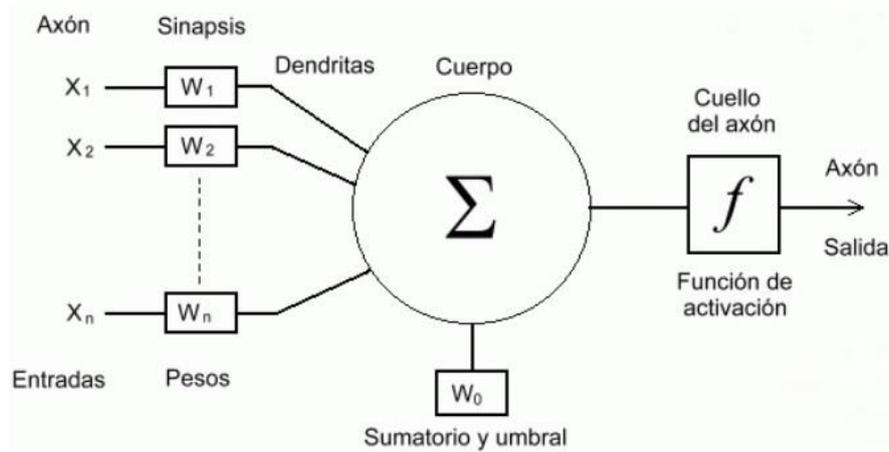
### **Clasificación de las Redes Neuronales Artificiales**

Las redes neuronales artificiales según su arquitectura y topología de red se clasifican en: redes neuronales monocapa y redes neuronales multicapa.

#### **Red Neuronal Monocapa - Perceptrón simple**

Un perceptrón es un algoritmo de clasificación binaria simple, ayuda a dividir un conjunto de señales de entrada en dos partes: "sí" y "no". Un perceptrón es una máquina de aprendizaje muy simple porque puede tomar algunas entradas, cada una de las cuales tiene un peso para indicar qué tan importante es para generar una decisión de salida de "0" o "1" (Moreno, 2016).

En la figura 5 observamos que el perceptrón simple posee nodos de entradas interconectadas y asociadas a un valor numérico determinado donde se conectan al núcleo y luego pasan a la función de activación quien será la encargada de realizar los cálculos para obtener una salida esperada.

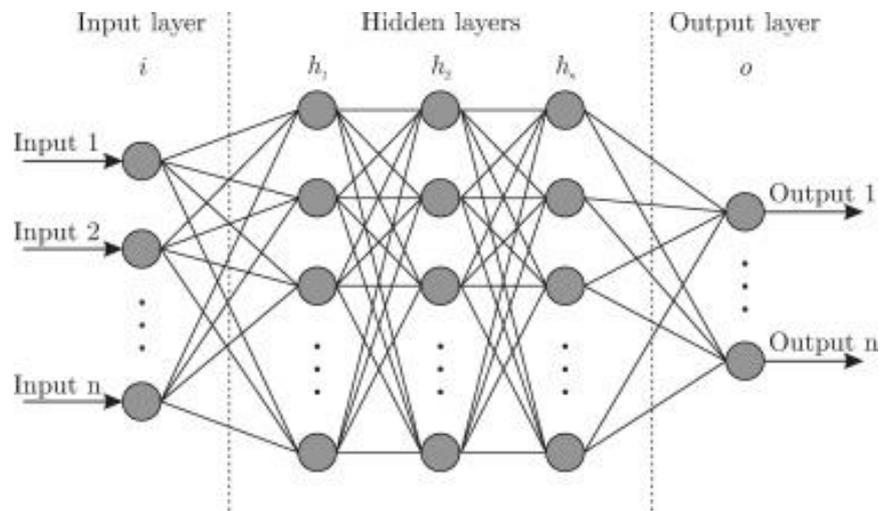
**Figura 5.***Estructura de un perceptrón simple*

Tomado de (Sancho, 2019)

**Red Neuronal Multicapa - Perceptrón Multicapa**

Los perceptrones multicapa también se conocen como "redes neuronales de alimentación hacia adelante", estas son redes más complejas que el perceptrón, ya que consisten en múltiples neuronas que están organizadas en capas. Entre las capas, se puede distinguir una capa de entrada, capas ocultas y una capa de salida, los perceptrones multicapa suelen estar completamente conectados, esto significa que hay una conexión de cada perceptrón en una capa específica a cada perceptrón en la siguiente capa (Delgado et al., 2019).

En la figura 6 las capas ocultas se colocan entre la capa de entrada y la capa de salida las cuales son el verdadero motor computacional ya que esta red dispone de neuronas agrupadas en varios niveles. Dado que este tipo de redes disponen de varias capas, las conexiones entre neuronas pueden ser del tipo feedforward (conexión hacia adelante) o del tipo feedback (conexión hacia atrás).

**Figura 6.***Estructura de una red neurona artificial*

Tomado de (Bre et al., 2018)

**Red Neuronal Convolutiva (CNN)**

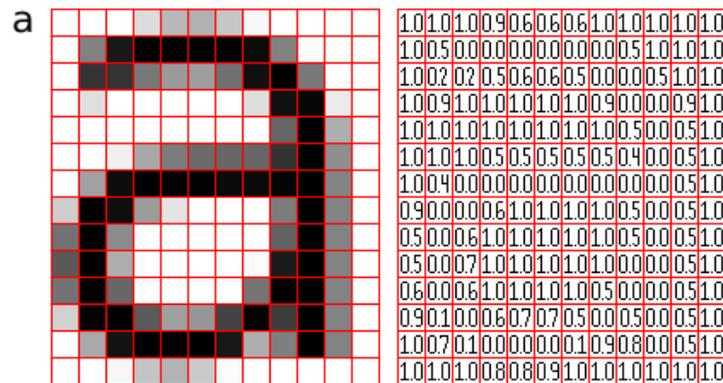
Las redes neuronales convolucionales son muy similares a las redes neuronales ordinarias, están formadas por neuronas que tienen pesos y sesgos que se pueden aprender. Cada neurona recibe algunas entradas, realiza un producto escalar y, opcionalmente, lo sigue con una no linealidad. Toda la red todavía expresa una única función de puntuación diferenciable: desde los píxeles de la imagen sin procesar en un extremo hasta las puntuaciones de la clase en el otro (R. Lopez, 2016).

Estas redes son muy utilizadas para el análisis, procesamiento y extracción de características de imágenes, esto se debe al funcionamiento que realizan la capa de entrada, la capa oculta y la capa de salida. Por lo tanto, al momento de analizar la imagen en la primera capa se obtienen los bordes o líneas de la imagen, la siguiente capa tiene la tarea de encontrar un grupo de patrones para reconocer estructuras más sencillas de la imagen, mientras que la última capa es la encargada de agrupar todos los patrones encontrados para realizar una predicción final (Antona, 2017).

Las imágenes contienen una gran cantidad de píxeles, los cuales se encuentran en forma de cuadrícula, son indicadores que determinan el color y la intensidad que tiene la imagen.

### Figura 7.

*Ilustración de una imagen en forma de cuadrícula*



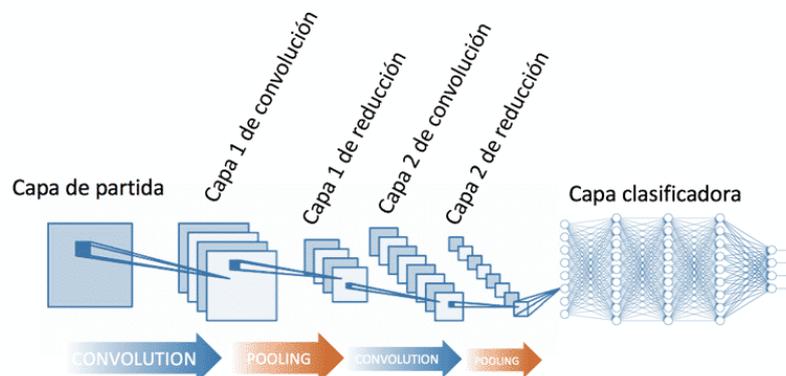
Tomado de (*Sample\_grid\_a\_square.Png (437×256)*, s. f.)

### Estructura de la CNN

Los elementos que integran una red neuronal convolucional están formadas por cinco tipos de capas, tales como: capa de entrada, capa convolucional, capa de reducción o pooling, capa clasificadora totalmente conectada y capa de salida.

### Figura 8.

*Arquitectura de una red neuronal convolucional*



Tomado de (Calvo, 2017b)

### ***Capa de entrada***

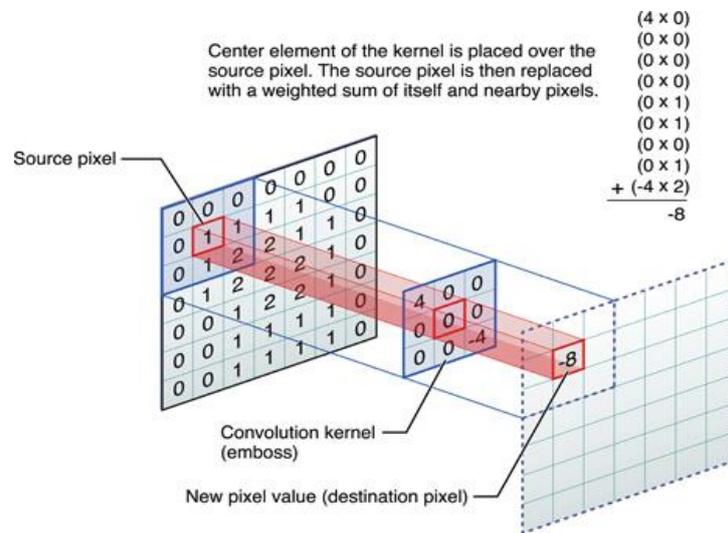
La capa de entrada tomará los píxeles de la imagen y estará formada por los siguientes componentes: (alto, ancho, canales).

- Donde alto y ancho corresponden a las dimensiones y el canal estará determinado por la representación del color, estos pueden ser: 1 sí la imagen es en escala de grises o 3 si es para RGB (rojo, verde, azul).
- Por ejemplo, tenemos una imagen con las siguientes dimensiones: 28x28x3 (alto, ancho, canales).

### ***Capa Convolutiva***

Esta segunda capa toma los píxeles de la imagen de entrada, se ejecuta matemáticamente a través de varios kernel o filtros, una pequeña matriz que va recorriendo todas las neuronas de entradas dando como resultado una nueva matriz de salida, lo que significa que sería una nueva capa de neuronas ocultas (Bagnato, 2018).

- La función de los filtros consiste en captar una porción de la imagen, la extrae en una imagen pequeña con mayor profundidad, en los que cada filtro identifica diferentes características propias de la imagen, y cada elemento es agrupado en una nueva matriz resultante con las características encontradas.
- Existen dos parámetros importantes dentro de la capa convolutiva. El tamaño del filtro, contiene la altura y la longitud del filtro que usaremos para recorrer generando una nueva abstracción. Y mientras que la longitud del filtro, es el número de filtros que tenemos en la capa y se encarga de detectar nuevos elementos, a los filtros de este proceso se le conoce como mapa de características. En la figura 8 se observa el proceso de convolución.

**Figura 9.***Proceso de convolución de una imagen*

Tomado de (*Performing Convolution Operations*, s. f.)

**Capa de reducción o pooling**

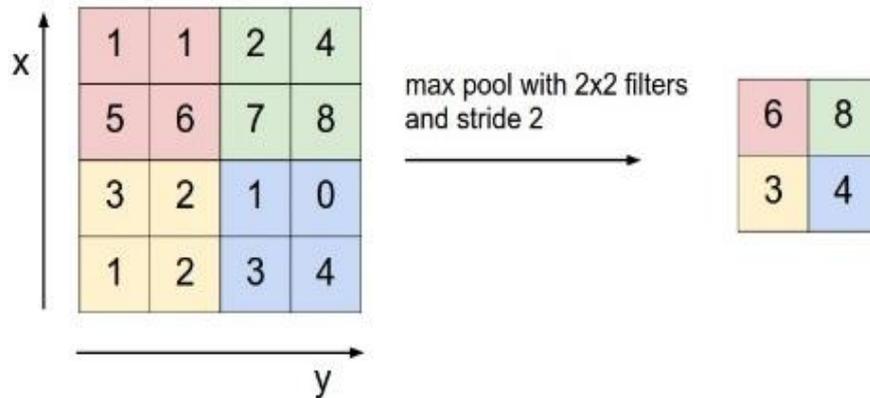
La tercera capa de la CNN, es la encargada de reducir las dimensiones de la altura y anchura de la matriz obtenida en la primera capa. Utiliza una técnica de agrupación Max-pooling (agrupación máxima) que es la más usada en estas redes, la cual consiste en escoger el pixel con el valor más alto que toma el kernel en una región (Sharma, 2017).

- Esta capa tiene como finalidad reducir y agrupar varios pixeles en uno.
- Sirve para no sobre ajustar y para reducir la carga de computación que se utilizará en la siguiente capa haciendo la imagen más pequeña.
- La operación de agrupación se especifica, en lugar de aprender. Dos funciones comunes que se utilizan en la operación de agrupación son:
  - Agrupación promedio: calcula el valor promedio para cada parche en el mapa de características.
  - Agrupación máxima (o agrupación máxima): calcula el valor máximo para cada parche del mapa de características

En la figura 10 se muestra el proceso de reducción de la imagen, donde se obtiene una matriz resultante de menor tamaño.

**Figura 10.**

*Max-pooling*



Tomado de (*Redes Neuronales Convolucionales CS231n Para Reconocimiento Visual*, s. f.)

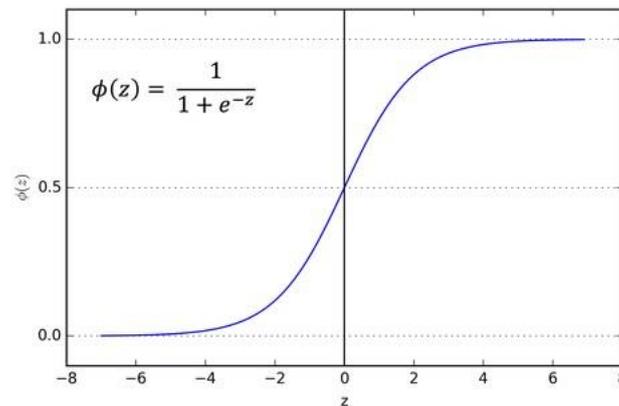
### ***Capa clasificadora totalmente conectada***

Es muy parecida a un perceptrón multicapa, esto se debe a que luego de obtener las características más importantes de una imagen en la capa de convolución y reducción, toma cada pixel como una neurona separada y por capas interconectadas entre sí. Ahora esta última capa clasificadora va tener varias clases y neuronas para poder obtener una salida de datos esperada (Calvo, 2017b).

### ***Capa de salida***

Una función de activación en una red neuronal define cómo la suma ponderada de la entrada se transforma en una salida de un nodo o nodos en una capa de la red.

La función sigmoid toma cualquier valor real como valores de entrada y de salida en el rango de 0 a 1. Cuanto mayor sea la entrada (más positiva), más cerca estará el valor de salida a 1.0, mientras que cuanto más pequeña sea la entrada (más negativa), más cerca estará la salida será 0.0 (Funciones de Activación de Capa, s. f.).

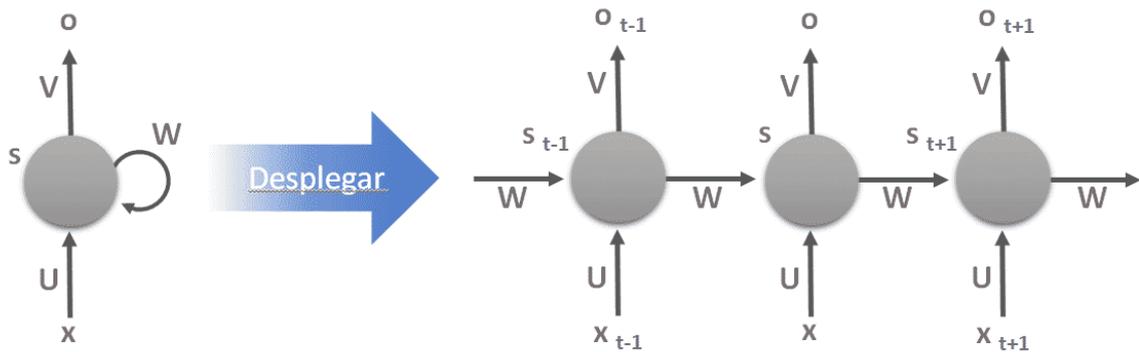
**Figura 11.***Función de activación sigmoid*

Tomado de (Funciones de Activación de Capa, s. f.).

### **Red neuronal Recurrente (RNN)**

La red neuronal recurrente proporciona un circuito de retroalimentación a la neurona anterior. Esta arquitectura permite recordar información importante sobre la entrada que recibió la red y toma en consideración el aprendizaje junto con la siguiente entrada. Los RNN funcionan muy bien con datos secuenciales como sonido, datos de series de tiempo (sensores) o lenguajes naturales escritos (Lazzeri, 2020).

- Las redes neuronales recurrentes aprovechan el algoritmo de retropropagación a través del tiempo (BPTT) para determinar los gradientes, que es ligeramente diferente de la retropropagación tradicional, ya que es específico de los datos de secuencia.
- Comparten parámetros en cada capa de la red. Si bien las redes feedforward tienen diferentes pesos en cada nodo, las redes neuronales recurrentes comparten el mismo parámetro de peso dentro de cada capa de la red.
- Los RNN comparten los parámetros en diferentes etapas de tiempo. Esto se conoce popularmente como Parameter Sharing. Esto da como resultado menos parámetros para entrenar y disminuye el costo computacional.

**Figura 12.***Red neuronal recurrente*

Tomado de (Calvo, 2017a)

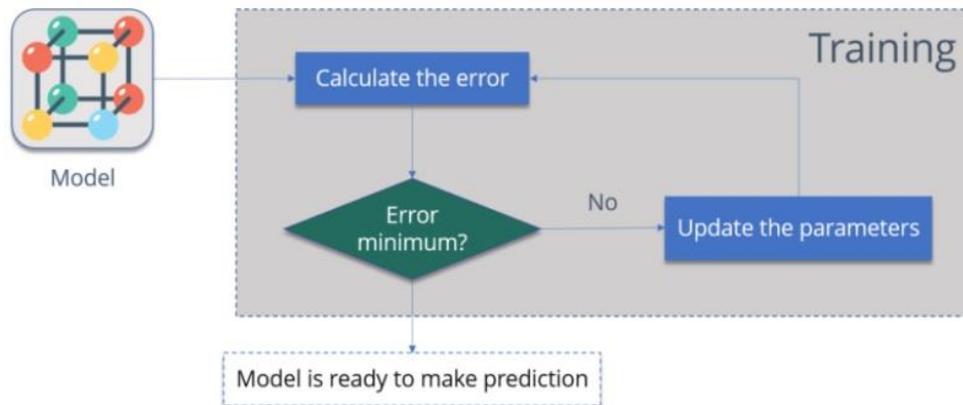
**Algoritmo de Backpropagation (Propagación inversa)**

El algoritmo de Backpropagation se considera una etapa de funcionamiento, la cual se expone en presencia de la red entrenada un patrón de entrada, donde se transfiere a través de sucesivas capas de neuronas hasta obtener una salida. Más adelante, en la etapa de entrenamiento o aprendizaje, se le modificarán los pesos a la red, de forma que esta coincida con la salida deseada por el usuario con la salida obtenida por la red (Piloto Rodríguez, 2017).

El algoritmo de Backpropagation es un algoritmo de aprendizaje supervisado para entrenar perceptrones multicapa. Según el analista de investigación en inteligencia artificial y big data de Edureka (Kulshrestha, 2020), “busca el valor mínimo de la función de error en el espacio de peso mediante una técnica llamada regla delta o descenso de gradiente. Las ponderaciones que minimizan la función de error se consideran una solución al problema de aprendizaje”. Para la comprensión del funcionamiento observar la figura 13.

**Figura 13.**

*Diagrama de funcionamiento del algoritmo backpropagation*



Tomado de (Kulshrestha, 2020)

- **Calcular el error:** Qué tan lejos está la salida del modelo de la salida real.
- **Error mínimo:** Comprobar si el error está minimizado o no.
- **Actualizar los parámetros:** Sí el error es grande, actualiza los parámetros (pesos y sesgos). Luego, verifica nuevamente el error. Repite el proceso hasta que el error sea mínimo.
- **El modelo está listo para hacer una predicción:** Una vez que el error es mínimo, podrá alimentar algunas entradas a su modelo y producirá la salida.

## Imagen Digital

### ¿Qué es una imagen digital?

Una imagen puede ser definida matemáticamente como una función bidimensional  $f(x, y)$ , donde  $X$  y  $Y$  son coordenadas espaciales (en un plano), y  $f$  en cualquier par de coordenadas es la intensidad o nivel de gris en esa coordenada. Cuando  $X$  y  $Y$  los valores de  $f$  son todas cantidades finitas, discretas, decimos que la imagen es una imagen digital. Una imagen digital se compone de un número finito de elementos, cada uno con un lugar y valor específico. Estos elementos son llamados píxeles. Esta información es interpretada

correctamente por una computadora, las copias digitales son perfectas y su reproducción es lograda sin errores (Valle, 2016).

### **Procesamiento Digital de Imágenes (PDI)**

El procesamiento digital de imágenes es una técnica que tiene como finalidad mejorar la calidad de las imágenes digitales, ya que permite extraer a profundidad sus características propias que pueden ser utilizadas para el estudio, comprensión e interpretación de la información dentro de ellas. Realiza una mejora previa para distinguir los componentes que no se encuentran a simple vista, con el afán de mejorar errores frecuentes en cuanto a la matiz del tono y el degrado de colores (Avilés Pincay & Barcia Salavarría, 2016).

### **Niveles de Procesamiento de Imágenes**

#### ***Captación***

Es el proceso por el cual se obtiene una imagen visual y se la toma como un valor de entrada.

#### ***Pre procesamiento***

Son operaciones que sirven para ajustar una imagen, con el fin de determinar un análisis de las características que van a ser extraídas. Por ejemplo, una operación de procesamiento es la modificación de brillo y contraste.

#### ***Segmentación***

Esta operación permite realizar una fragmentación de la imagen en varias extensiones, para figurar los datos necesarios del problema que queremos solucionar.

#### ***Detección de objetos y clasificación***

Esta técnica tiene como función establecer y ordenar los objetos alojados en una imagen.

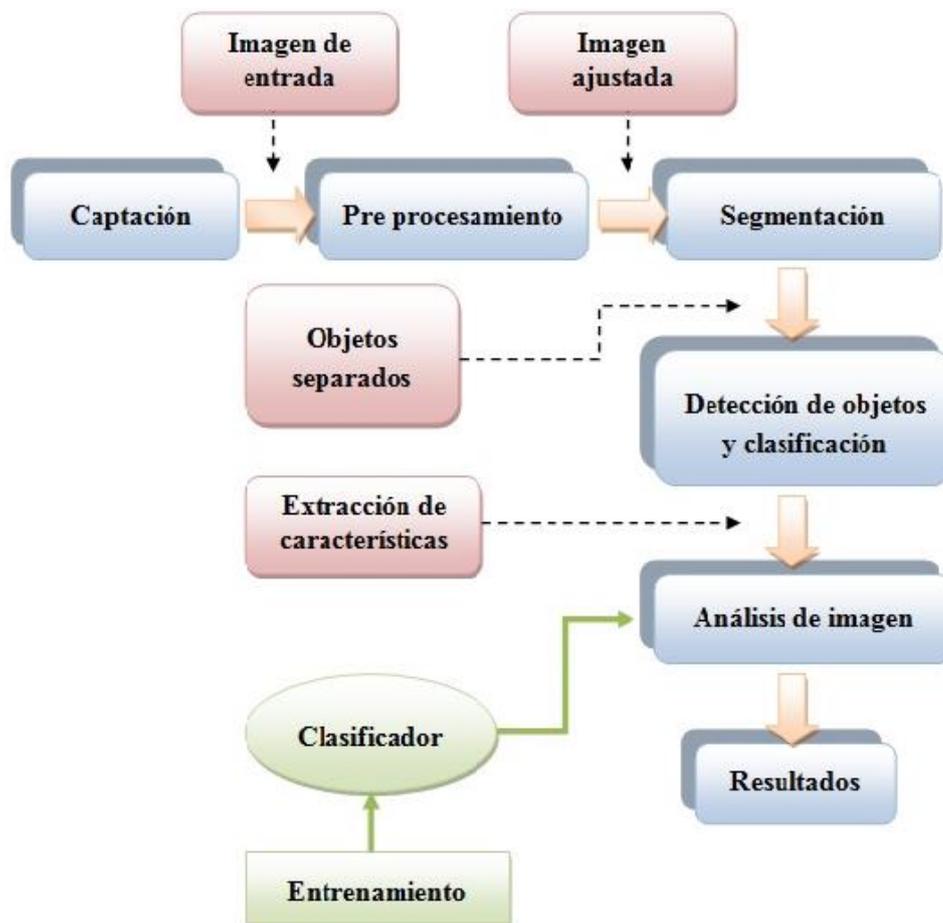
### *Análisis de imagen*

Permite adquirir datos de alto nivel respecto a las características que se presentan en la imagen. De esta forma se utiliza un modelo en la toma de decisiones que permita categorizar a qué lugar pertenecen los objetos.

En la figura 14 se observa cómo funciona el procesamiento digital de imágenes para tratamiento, ya que guarda mucha relación con las fases del desarrollo del proyecto.

**Figura 14.**

*Niveles de procesamiento de imágenes*



*Nota.* Datos de la investigación. La elaboración es propia

## Tipos de Imágenes Digitales

El tipo de imágenes digitales que se usan para el procesamiento existen cuatro tipos, estos son: imágenes binarias, imágenes indexadas, imágenes RGB e imágenes en escala de grises.

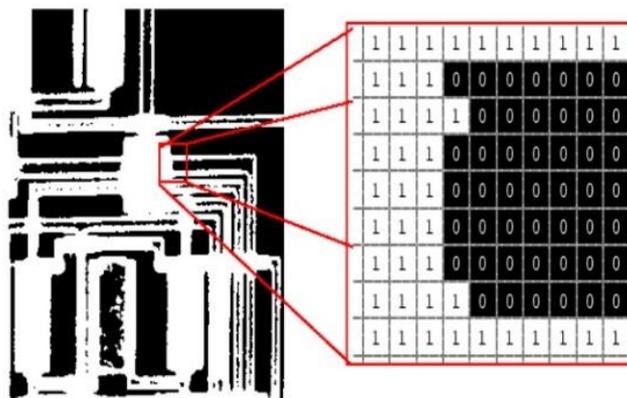
### Imágenes Binarias

La binarización consiste en un proceso de reducción de información. Una imagen binaria se almacena como una matriz, donde las imágenes se caracterizan por presentar dos valores, verdadero o falso, donde 1 es blanco y 0 negro cuyos pixeles toman los valores referenciales del color. En muchos casos una imagen binaria es el resultado del proceso de segmentación por niveles de gris o ya sea por la segmentación de un rango de color en particular, este proceso les permite emplear una selección de zonas de interés, para luego ser tomados como valores de referencia y comparación. Las características de este tipo de imágenes son:

- Requiere un bit por cada pixel.
- Este tipo de imágenes son muy útiles en términos de almacenamiento.

### Figura 15.

*Valores de pixeles en una imagen binaria*



Tomado de (MathWorks, 2020)

## Imágenes Indexadas

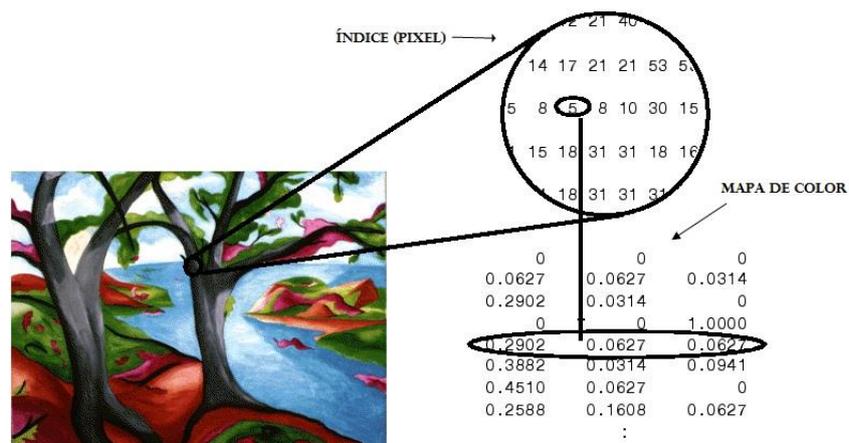
Tiene como finalidad almacenar dos tipos de matrices. La primera matriz contiene el mismo tamaño de la imagen principal y a este se le atribuye un número para cada pixel. La segunda matriz se le denomina mapa de color, cuya dimensión toma su origen al número de colores que se desea emplear. Esta técnica sirve para realizar la compresión de imágenes y reservar espacio en la memoria del computador (Avilés Pincay & Barcia Salavarría, 2016).

Las características de este tipo de imágenes son:

- Admite la simplificación de colores en la imagen hasta un valor máximo de 256.
- Acepta los formatos de imagen GIF y PNG-8 para archivos multimedia.

**Figura 16.**

*Estructura de una imagen indexada*



Tomado de (MathWorks, 2020)

## Imágenes RGB (Red-Green-Blue)

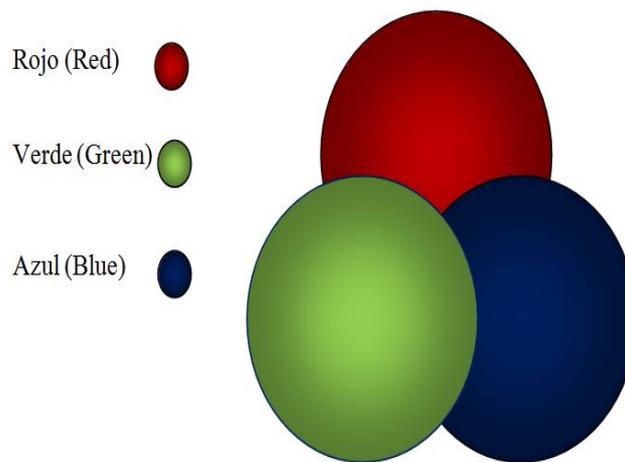
El modelo de color RGB se basa en la adición de los tres colores primarios y está presente en la mayoría de dispositivos electrónicos. Está representando por un sistema de tres dimensiones (R, G, B) el cual está formado entre valores que oscilan desde 0, 0, 0 hasta el 255, 255, 255. Donde 0 significa poca transparencia y 255 mayor luminosidad (Solé, 2020).

Las características de este tipo de imágenes son:

- Está compuesto de (8bits x 3 canales) = 24(número total de bits para cada pixel).
- Este modelo abarca un total de 16. 7 millones de colores.
- Acepta formatos de imágenes JPG, PNG Y BMP.

### Figura 17.

*Formato de color RGB*



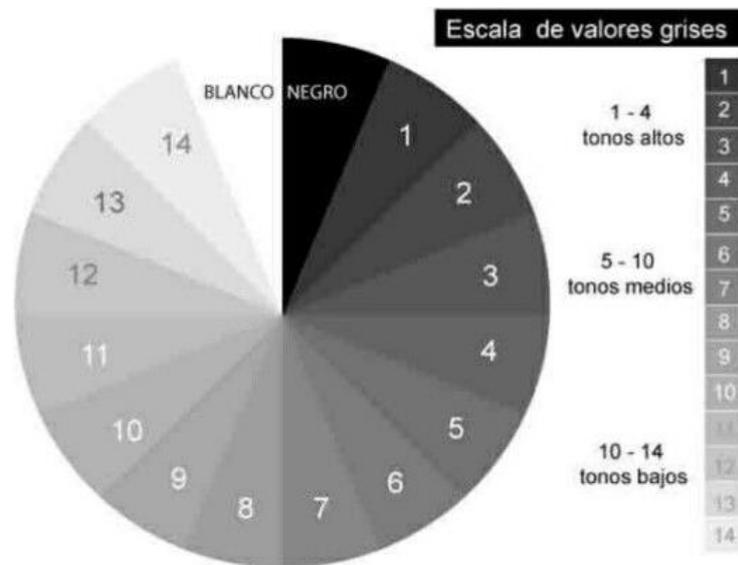
*Nota.* Datos de la investigación. La elaboración es propia

### Imágenes en Escala de Grises

La escala de grises o comúnmente conocida como escala monocromática, es aquella que usa como parámetro un pixel, el cual adquiere un valor numérico individual específico que sirve para representar su luminancia, cuya escala se extiende entre blanco y negro. Está formada por una sola matriz  $M \times N$  y se emplea al modo de color en el que cada componente (píxel o de otro tipo) se detalla dentro de una sucesión limitada de estimaciones de un mismo tono neutro (Valle, 2016). Se observa en la figura 18.

Las características de este tipo de imágenes son:

- Cada píxel tiene un valor aproximado de entre 0 (negro) a 255( blanco).
- Una imagen de 8 bits tendría como resultante hasta 256 tipos de gris.

**Figura 18.***Modelo en escala de grises*

Tomado de (Valle, 2016)

### **Tipos de Formatos de Imágenes Digitales**

#### **Formato de Imagen JPG O JPEG**

El formato de compresión JPG (Joint Photographic Experts Group) debido a su alta demanda de uso por los usuarios, es el que más se utiliza para guardar imágenes. El tamaño del archivo que se está comprimiendo por lo general hace una reducción invariable al descartar información innecesaria y redundante de la imagen (Jackson, 2019).

#### **Formato de Imagen PNG**

El formato de imagen PNG (Portable Network Graphics) a diferencia del JPG, incluye transparencia porque contiene una mayor compresión y no disminuye la calidad de la imagen. Este formato fue inventado para superar los condicionamientos del formato GIF y usa una compresión sin pérdidas ya que no está sujeto a las patentes que pesan sobre el empleo del GIF (B. Lopez, 2020).

## Formato de Imagen GIF

El formato de imagen GIF (Graphics Interchange Format) es uno de los más vistos en redes sociales por su alto contenido visual. Este formato permite la unión de varias imágenes con el fin de conseguir una animación al establecer una reproducción de imágenes en secuencia (Romero, 2020).

**Tabla 3.**

*Comparativa de los Tres Tipos de Formatos de Imagen más Usados*

Formato de imagen	Ventajas	Desventajas
<p><i>JPG o JPEG</i> (Joint Photographic Experts Group)</p>	<ul style="list-style-type: none"> <li>• Son más utilizados en páginas web.</li> <li>• Tamaño reducido.</li> <li>• Velocidad de carga mucho más rápida.</li> </ul>	<ul style="list-style-type: none"> <li>• Pérdida de calidad por su gran compresión.</li> <li>• No incluye el atributo de transparencia.</li> </ul>
<p><i>PNG</i> (Portable Network Graphics)</p>	<ul style="list-style-type: none"> <li>• Contiene un amplio contraste y colorido.</li> <li>• Mayor exhibición a detalle de la imagen.</li> <li>• Incluye el atributo de transparencia.</li> </ul>	<ul style="list-style-type: none"> <li>• Mayor compresión sin pérdida de calidad.</li> <li>• Devuelve archivos más pesados.</li> </ul>
<p><i>PNG 8</i></p>	<ul style="list-style-type: none"> <li>• Formato de color de 8 bits.</li> <li>• Menor tamaño de archivos.</li> </ul>	<ul style="list-style-type: none"> <li>• Limitación de colores hasta 256.</li> </ul>
<p><i>PNG 24</i></p>	<ul style="list-style-type: none"> <li>• Formato de color de 24 bits.</li> <li>• Expansión a 16 millones de colores.</li> <li>• Mayor calidad de color y mejor transparencia.</li> </ul>	<ul style="list-style-type: none"> <li>• Mayor tamaño de archivos.</li> </ul>
<p><i>GIF</i> (Graphics Interchange Format)</p>	<ul style="list-style-type: none"> <li>• Reproducción de imágenes en secuencia.</li> <li>• Contiene 256 colores.</li> <li>• Menor tamaño de archivo.</li> </ul>	<ul style="list-style-type: none"> <li>• No es compatible con todos los programas y aplicaciones.</li> </ul>

*Nota.* La elaboración es propia y la fuente corresponde a (Romero, 2020)

### ¿Qué es un dataset?

Un dataset es una colección de datos tabulada el cual corresponde al contenido de una base de datos o una matriz de datos estadística.

El conjunto de datos está formado por datos estructurados (filas y columnas), en la que cada columna de la tabla corresponde al valor de una variable específica y la fila (registro) hace referencia a un componente en concreto de la colección de datos que se desea emplear. Aquí se encuentran todos los valores que pueden tener una variable y las características importantes de un objeto como son la altura y el peso, los cuales corresponden a cada elemento de la matriz de datos. Así mismo el conjunto de datos puede contener otro tipo de información como números enteros, decimales o caracteres. El dataset puede añadir nuevos valores para uno o más componentes, pero puede variar dependiendo de la función del número de registros (Balagueró, 2018).

#### **Tabla 4.**

##### *Estructura de una matriz de datos*

<b>Registro No.</b>	<b>Nombre</b>	<b>Apellido</b>	<b>Teléfono</b>	<b>Marcas %</b>
001	Austin	Jobs	0945261236	88.90%
002	Marck	Evans	0987451236	78.40%

*Nota.* Datos de la investigación. La elaboración es propia

El conjunto de datos incorpora la relación entre tablas de una base datos. Estos datos son almacenados en memoria donde utiliza un modelo de programación relacional consecuente y autónoma sin importar la procedencia de los datos que esta contenga (Balagueró, 2018).

## Herramientas y Plataformas de Desarrollo

### Lenguaje de Programación Python

Python es un lenguaje de programación que tiene como finalidad dirigirse a los usuarios de forma general, es potente, flexible, simple y fácil de aprender. Por otra parte es considerado como un lenguaje de alto nivel, ya que permite procesar de manera sencilla todo tipo de estructuras de datos, tanto numéricos como de texto (Covantec, 2018).

Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos.

### Historia

El lenguaje de programación Python fue desarrollado por Guido Van Rossum, un ingeniero holandés que laboraba en la CWI de Ámsterdam, más conocido como el Centro de Investigación de Ciencias de la Computación, en la década de los noventa, en los Países Bajos. Python toma las cualidades más importantes de lenguajes antecesores, haciéndolo compatible con soluciones de muchos de ellos: Fortran 1957, Pascal 1970, C 1972, Modula-2 1978 y Java 1991. Este entorno de desarrollo adoptó tres formas de imprimir los valores de una variable, desde el entorno interactivo escribiendo el valor de la variable (Basic), usando la concatenación de componentes usando la función print, con concatenación de componentes (al estilo del write de Pascal) o bien utilizando patrones de formato (al estilo del Printf de C) (Covantec, 2018).

Por esa razón el lenguaje de programación desde sus comienzos nació como proyecto de software libre, lo que debe en parte de su éxito la decisión de realizarlo como código abierto.

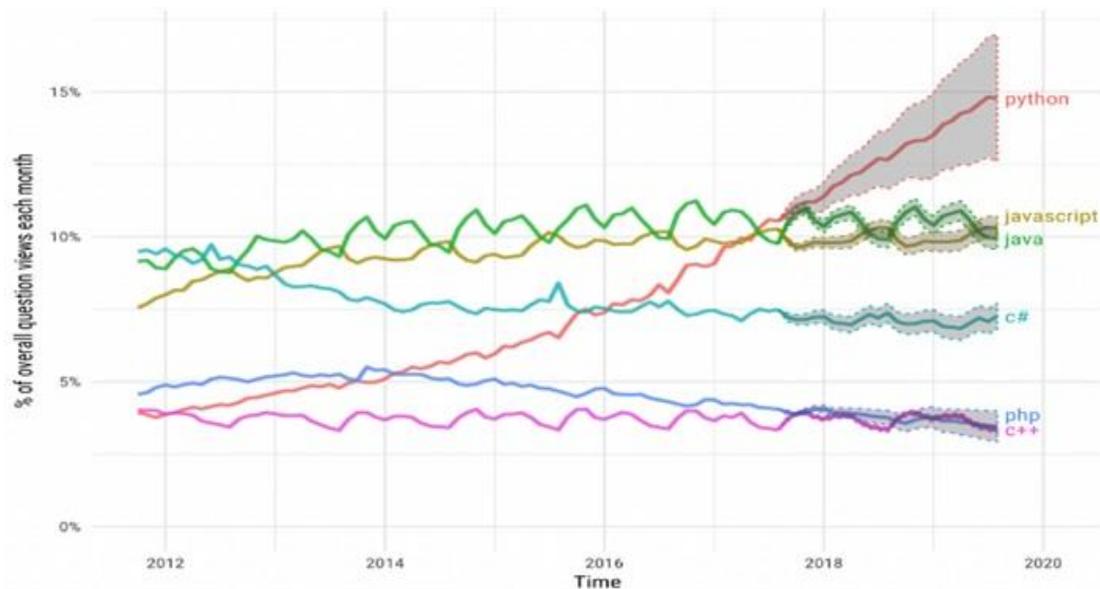
## Características

- Python está desarrollado bajo una licencia Open Source o código abierto.
- Python es un lenguaje multiparadigma.
- Es un lenguaje orientado a objetos.
- Integra propiedades de la programación imperativa, funcional, procedural y reflexiva.
- Es un lenguaje interpretado, esto significa que no se compila a diferencia de otros lenguajes como Java o C/C++, sino que es interpretado en tiempo de ejecución.
- Es de tipado dinámico, aunque opcionalmente desde la versión 3.5 podemos hacer uso de tipado estático (Zaforas, 2017).
- Es un lenguaje multiplataforma, ya que este entorno está disponible para (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, etc.)
- Python es un lenguaje de escritura rápido, escalable, robusta y de código abierto, ventajas que hacen de Python un aliado perfecto para la Inteligencia Artificial (Soloaga, 2018).

Python podría mantenerse a la cabeza o ser superado por Java en el otoño (está aproximadamente dentro de la variación de las predicciones del modelo), pero está claramente en camino de convertirse en la etiqueta más visitada en 2018. STL también sugiere que JavaScript y Java se mantendrán en niveles similares de tráfico entre los países de altos ingresos, tal como lo han hecho durante los últimos dos años. Se observa en la figura 19 las proyecciones que tendrán los lenguajes de programación con respecto a su usabilidad por los usuarios como la etiqueta más visitada. (Robinson, 2017).

**Figura 19.**

*Proyecciones de los lenguajes de programación a futuro*



Tomado por (Robinson, 2017)

## Usos y Aplicaciones

### *Python en el desarrollo web*

Python permite construir mucho más con menos líneas de código, por lo que se crean prototipos de forma más eficiente.

Otras alternativas de frameworks muy maduros y asentados como lo son: Symfony para PHP, Spring para Java, Grails para Groovy o Rails para Ruby (Zaforas, 2017).

### *Python en Data Science*

Algunas de las bibliotecas de Python para Data Science son TensorFlow, Keras, NumPy, Scikitlearn o Pandas. Estas permiten a los usuarios llevar a cabo tareas de forma mucho más eficiente. Por ejemplo, con Pandas podemos manejar datos o limpiar CSV con miles de filas y columnas. TensorFlow, en cambio, nos permite desarrollar redes neuronales de forma más sencilla (Universidad de Alcalá, 2020).

### ***Python para Big Data***

Compatibilidad. Hadoop es la plataforma de big data de código abierto más popular y la compatibilidad inherente de Python es otra razón más para preferirlo a otros lenguajes. Python tiene un poderoso conjunto de paquetes para una amplia gama de necesidades de análisis y ciencia de datos (Universidad Internacional de Valencia, 2019).

### ***Python en la Inteligencia Artificial (AI)***

Permite plasmar ideas complejas con unas pocas líneas de código, lo que no es posible con otros lenguajes. Existen bibliotecas como “Keras” y “TensorFlow”, que contienen mucha información sobre las funcionalidades del aprendizaje automático. Posee bibliotecas gratuitas, las cuales se usan mucho en los algoritmos como Scikitlearn, una biblioteca gratuita de aprendizaje automático que presenta varios algoritmos de regresión, clasificación y agrupamiento (Soloaga, 2018).

### **Entorno de Desarrollo en la Nube**

Para el presente proyecto se ha utilizado el entorno de desarrollo de Google Colab, en este apartado se explicará una de sus funcionalidades más importantes.

### **Google Colab**

Este entorno de desarrollo está proporcionado por Jupyter Notebook, ya que no precisa de configuraciones. Uno de los beneficios que ofrece Google es el almacenamiento de datos y ejecución en la nube. Con Google Colab es posible unir esas aptitudes y sacarle un mayor provecho a la hora de programar un cuaderno en código Python y elaborar modelos computacionales de Machine Learning (De la fuente, 2019). A continuación, se explican sus funciones principales.

**Tabla 5.***Funciones principales de Google Colab*

<b>Características</b>	<b>Funciones</b>
<b>Tipo de cálculo</b>	<ul style="list-style-type: none"> <li>• Hace uso de GPU en lugar de CPU para realizar cálculos complejos.</li> </ul>
<b>Rendimiento</b>	<ul style="list-style-type: none"> <li>• Proporciona potencia y libera la máquina de la carga computacional haciendo que se ejecuten las tareas de forma rápida.</li> </ul>
<b>Tipo de ejecución</b>	<ul style="list-style-type: none"> <li>• Permite realizar ejecuciones en tiempo real por cada comando escrito.</li> </ul>
<b>Recursos y herramientas</b>	<ul style="list-style-type: none"> <li>• Proporciona bibliotecas de alto nivel escritas en Python como TensorFlow y Keras.</li> <li>• Permite trabajar incluso si no se dispone de una computadora potente.</li> <li>• Se conecta a una VM de Google Compute Engine (la infraestructura de máquinas virtuales de Google en la nube) cuando ejecutamos una celda</li> </ul>
<b>Almacenamiento</b>	<ul style="list-style-type: none"> <li>• La máquina en un inicio cuenta con 12 GB de RAM y 50 GB de almacenamiento en disco disponibles para el uso.</li> </ul>

*Nota:* Se muestran las funciones que desempeña el entorno de ejecución de Google Colab.

### ¿Qué es una GPU?

Unos de las ventajas que ofrece este entorno de desarrollo es que hace uso de la unidad de procesamiento gráfico por sus siglas en inglés GPU (Graphics Processing Unit).

La GPU es la parte principal de una tarjeta gráfica al igual que una CPU para los computadores, es decir es la unidad fundamental que sirve para realizar cálculos complejos al momento de ejecutar una tarea, está presente en los videojuegos ya que permite disfrutar de una mejor calidad de imágenes y gráficos en alta resolución mostrados por pantalla (López, 2021).

**Tabla 6.***Características y componentes de una GPU*

<b>Componentes</b>	<b>Características</b>
<b>Procesador de comandos</b>	<ul style="list-style-type: none"> <li>• Lee la lista de pantalla o de instrucciones que le envía la CPU.</li> <li>• Generar gráficos.</li> <li>• Realiza cálculos complejos.</li> </ul>
<b>Unidad de Rasterizado</b>	<ul style="list-style-type: none"> <li>• Realiza la transformación del espacio tridimensional a uno bidimensional.</li> <li>• Se basa en pixeles</li> </ul>
<b>Raster Output:</b>	<ul style="list-style-type: none"> <li>• Trazar los pixeles finales sobre el búfer de imagen.</li> <li>• Posee una Caché de último nivel.</li> <li>• Tiene permisos de escritura sobre la VRAM.</li> </ul>
<b>Unidad de Texturizado</b>	<ul style="list-style-type: none"> <li>• Simula texturas o colores en la vida real aplicado sobre una imagen.</li> </ul>
<b>Unidad de intersección</b>	<ul style="list-style-type: none"> <li>• Calcula la intersección de los rayos de la escena con los objetos.</li> <li>• Es esencial para el Ray Tracing (trazado de rayos).</li> </ul>
<b>CODEC de Vídeo</b>	<ul style="list-style-type: none"> <li>• Procesador independiente.</li> <li>• Descodifica vídeo en varios formatos multimedia y los reproduce.</li> <li>• Genera vídeo e incluso pasar de un formato a otro.</li> </ul>
<b>Interfaz de memoria</b>	<ul style="list-style-type: none"> <li>• Permite a la GPU leer de su memoria RAM, conocida como VRAM.</li> </ul>
<b>DMA</b>	<ul style="list-style-type: none"> <li>• Permite a la GPU leer de la RAM principal del sistema.</li> </ul>

*Nota.* Detalle por cada componente evaluado

Es por ello que la computación acelerada por GPU consiste en acelerar el funcionamiento de las aplicaciones de aprendizaje profundo, análisis e ingeniería donde la computación es más intensiva mientras que el resto del código se ejecuta en la CPU. Por el cual se van a explicar las diferencias en la siguiente tabla comparativa.

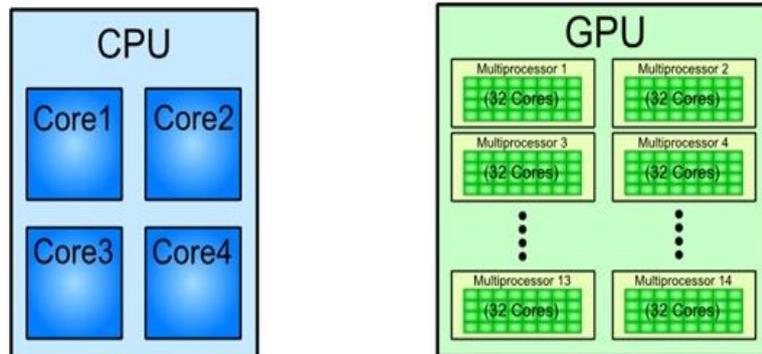
**Tabla 7.***Diferencias entre CPU y GPU*

CPU	GPU
<ul style="list-style-type: none"> <li>• Cachés realmente rápidos (excelente para la reutilización de datos).</li> <li>• Granularidad de ramificación fina.</li> <li>• Muchos procesos / hilos diferentes.</li> <li>• Alto rendimiento en un solo hilo en ejecución.</li> <li>• Las CPU son excelentes para el paralelismo de tareas.</li> <li>• CPU optimizadas para un alto rendimiento en códigos secuenciales (cachés y predicciones de rama).</li> </ul>	<ul style="list-style-type: none"> <li>• Muchas unidades de matemáticas.</li> <li>• Acceso rápido a la memoria integrada.</li> <li>• Ejecutar un programa en cada fragmento / vértice.</li> <li>• Alto rendimiento en tareas paralelas.</li> <li>• Las GPU son excelentes para el paralelismo de datos.</li> <li>• GPU optimizadas para alta intensidad aritmética para naturaleza paralela (operaciones de punto flotante).</li> </ul>

*Nota:* Observamos que tienen similitudes en cuanto al rendimiento para realizar diversos procesos

Para comprender de mejor forma, la CPU ejecuta varias tareas a la vez, pero no arroja resultados en tiempo real, sino que espera un tiempo determinado para devolver un valor, a diferencia de la GPU que si realiza ejecuciones en tiempo real. Para demostrar las diferencias entre CPU y GPU de los fabricantes de componentes y de tecnología “Nvidia” señalan que:

La arquitectura de la CPU, posee un número determinado de núcleos optimizados para realizar procesos en serie secuencial, mientras que la GPU cuenta con una arquitectura en paralelo la cual contiene miles de núcleos más pequeños eficaces y que se diseñaron para resolver varias tareas al mismo tiempo (Nvidia, s.f.).

**Figura 20.***Arquitectura de una CPU y GPU*

Tomado de (Chimp, 2017)

Otros datos importantes para comprobar el estudio acerca de las diferencias entre CPU y GPU son:

Si bien las GPU pueden procesar datos varios órdenes de magnitud más rápido que una CPU debido al paralelismo masivo, las GPU no son tan versátiles como las CPU. Las CPU tienen conjuntos de instrucciones grandes y amplias, que administran todas las entradas y salidas de una computadora, lo que una GPU no puede hacer. (OmniSci, s.f.).

### Google Drive

Google drive es un servicio gratuito de almacenamiento en la nube que le permite a sus usuarios acceder, crear copias de seguridad de todos los archivos y compartir documentos. Por medio de una cuenta de Gmail los usuarios tendrán acceso a todos los servicios que Google ofrece, cuenta con un almacenamiento de 15GB y este puede extenderse utilizando la versión de pago (Uriarte, 2020). Una de las funciones más importantes de Google Drive es que permite trabajar de forma colaborativa con otros servicios de Google, como lo es el entorno de desarrollo de Google Colab para extraer archivos almacenados en la nube y usarlos en la elaboración de modelos de machine learning.

## **Librerías de Machine Learning**

### **TensorFlow**

TensorFlow es una plataforma de código abierto, específicamente es un instrumento de computación numérico creado por Google. Se utiliza para proyectos de aprendizaje automático. Posee dos características importantes. La flexibilidad, permite al desarrollador crear e innovar nuevas ideas a un periodo de tiempo a corto plazo. Mientras que la escalabilidad, permite utilizar los modelos desarrollados para que puedan ser usados por millones de usuarios (Sanseviero, 2019).

Para el proyecto se consideró que TensorFlow compila y entrena modelos de aprendizaje automático mediante API intuitivo de alto nivel para realizar ejecuciones en tiempo real. Accede a la correlación de los modelos y la depuración de los datos de forma ágil e inmediata. Esta herramienta es utilizada para la creación de modelos basados en redes neuronales artificiales para detectar y descifrar patrones.

### **Keras**

Para este proyecto se eligió Keras, ya que es una librería de redes neuronales escrita en Python. Se ejecuta sobre la plataforma de aprendizaje automático de TensorFlow, para realizar experimentos con redes neuronales artificiales. Por lo cual se considera lo siguiente.

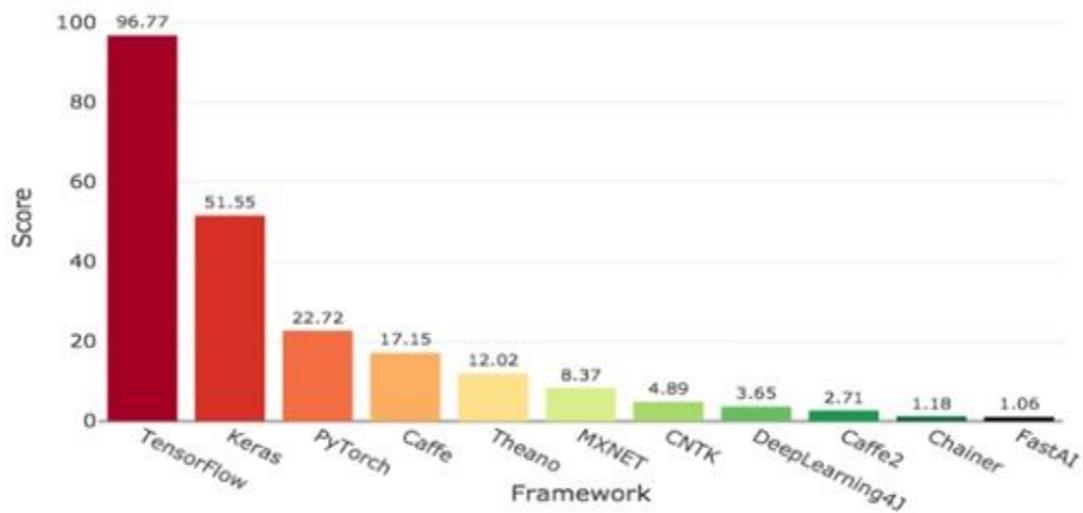
Keras es una API de alto nivel de TensorFlow 2: una interfaz accesible y altamente productiva para resolver problemas de aprendizaje automático, con un enfoque en el aprendizaje profundo moderno. Proporciona abstracciones y bloques de construcción esenciales para desarrollar y enviar soluciones de aprendizaje automático con alta velocidad de iteración (Keras, s.f.).

Proporciona la creación de modelos de forma rápida y sencilla, de otro modo, está desarrollado para simplificar la carga mental y reducir la cantidad de acciones por el usuario en los casos de uso más comunes. A continuación, en la figura 21 se muestra los datos

estadísticos que fueron tomados como base para la elección de las herramientas y el desarrollo del proyecto. Según los estudios de Jeff Hale.

**Figura 21.**

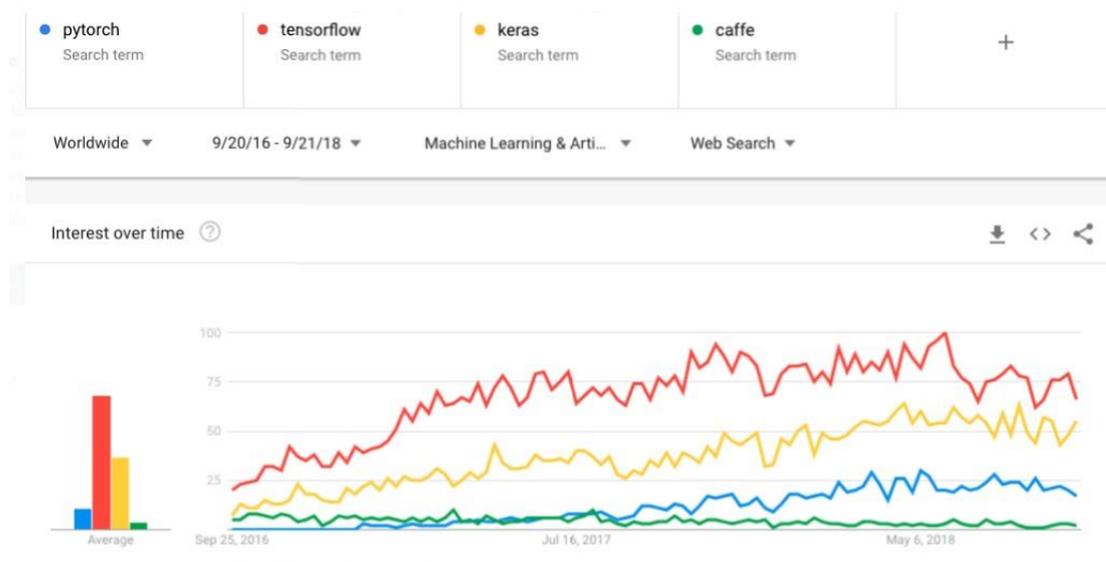
*Comparación de los marcos de aprendizaje profundo.*



Tomado de (Hale, 2018)

**Figura 22.**

*Comparación por búsqueda en la web*



Tomado de (Hale, 2018)

Para la realización del estudio, se tomó en consideración los conceptos aprendidos en la investigación, no solo por el uso de las herramientas, sino también por la demanda de usuarios y científicos de datos en la búsqueda de nuevas tecnologías que aporten al desarrollo y el conocimiento de cada individuo. Se observa en la figura 22, estudio realizado por Jeff Hale y estadísticas proporcionadas por Google Trends.

Keras no estaba lejos de TensorFlow. PyTorch estaba en tercer lugar y otros marcos tenían puntuaciones de volumen de búsqueda relativas de cuatro o menos. Estas puntuaciones se utilizaron para los cálculos de puntuación de potencia. Se observa brevemente cómo ha cambiado el volumen de búsquedas a lo largo del tiempo para brindar un contexto más histórico. El gráfico de Google directamente muestra las búsquedas en los últimos dos años (Hale, 2018).

**Tabla 8.**

*Revisión de librerías de Machine Learning por categorías*

Framework	Online Job Listings					KDnuggets Usage Survey	Google Search Volume	Medium Articles	Amazon Books	ArXiv Articles	GitHub Activity			
	Indeed	Monster	Simply Hired	LinkedIn	Angel List						Stars	Watchers	Forks	Contributors
TensorFlow	2,079	1,253	1,582	2,610	552	29.90%	73	6,200	202	3,700	109,576	8,334	67,551	1,642
Keras	684	364	449	695	177	22.20%	53	9,120	79	1,390	33,558	1,847	12,658	719
PyTorch	486	309	428	665	120	6.40%	19	1,780	18	1,560	18,716	952	4,474	760
Caffe	607	399	515	866	123	1.50%	4	815	14	1,360	25,604	2,218	15,633	270
Theano	356	316	279	508	95	4.90%	0	428	17	652	8,477	585	2,447	328
MXNET	266	154	200	298	29	1.50%	2	524	32	260	15,200	1,170	5,498	587
CNTK	126	96	97	160	12	3.00%	0	223	1	88	15,106	1,368	4,029	189
DeepLearning4J	17	5	9	35	3	3.40%	2	70	11	27	9,615	829	4,441	232
Caffe2	55	51	49	109	12	1.20%	2	335	2	67	8,284	577	2,102	193
Chainer	19	19	19	28	3	0.00%	2	91	3	164	4,128	325	1,095	182
FastAI	0	0	0	0	0	0.00%	0	858	0	11	7,268	432	2,647	195

*Nota.* Investigación proporcionada por Jeff Hale en conjunto con Towards Data Science. Estudio realizado basado en 11 fuentes de datos en 7 categorías distintas para medir el uso, el interés y la popularidad del marco de herramientas de machine learning. Priorizando los criterios como: Listados de trabajos en línea, encuestas realizadas por KDnuggets, búsquedas web y actividades de GitHub. Tomado de (Hale, 2018).

## **Machine Learning Aplicado al Análisis de Imágenes de Rayos X.**

El estudio del proyecto tiene como objetivo conocer la funcionalidad de las redes neuronales convolucionales para realizar modelos de aprendizaje automático, basados en aprendizaje profundo para la detección de objetos y la clasificación de imágenes. A continuación, se mencionan los casos de estudio que guardan relación con el análisis de imágenes de rayos X para Covid-19.

### **Casos de estudio**

#### **Sistema de Aprendizaje Profundo para Detectar la Neumonía de 2019 por la Enfermedad del Nuevo Coronavirus.**

Los especialistas en radiología indicaron que surgían manifestaciones de los casos de COVID-19 vistas a través de TC donde cada imagen tiene sus propias características, esto ayudaba a que los médicos encuentren anomalías al clasificar una imagen por presencia covid-19 o caso sospechoso de neumonía.

El proyecto de aprendizaje profundo tuvo como objetivo la detección prematura de COVID-19 mediante imágenes de tomografía computarizada (TC), el cual se plantea clasificar los casos como covid-19, IAVP (Neumonía Viral por Influenza) e irrelevantes para la infección (ITI). Para el conjunto de datos se extrajeron muestras de 219 pacientes con COVID-19 confirmados, 224 pacientes con neumonía y 175 de pacientes normales (Xu et al., 2020).

Utilizaron los modelos basados en la red residual tradicional ResNet y otro modelo de clasificación de ubicación-atención ResNet-18.

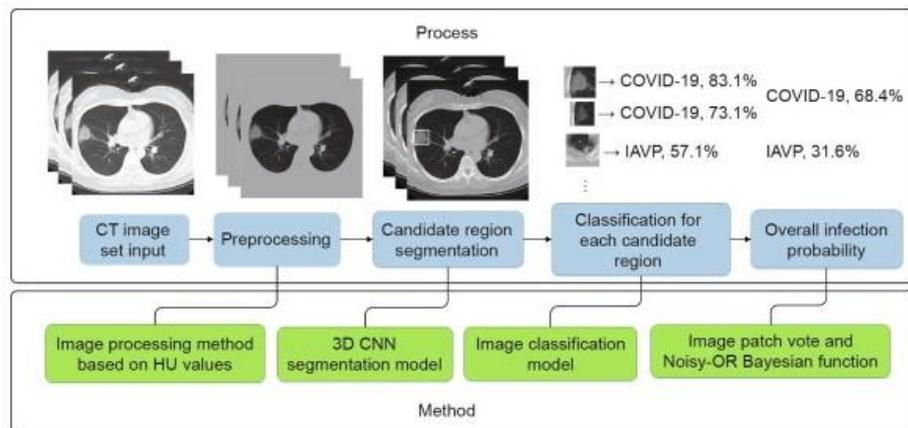
Para el proceso de la generación del informe pulmonar realizaron los siguientes pasos:

- La extracción de las regiones de las afecciones pulmonares se preprocesaron en imágenes de TC.

- Utilizaron un modelo de segmentación de red neuronal convolucional en tres dimensiones para realizar la división de los cubos de imágenes de los candidatos. Por ejemplo, la imagen central del recuadro muestra como cada vecino recopiló datos significativos para los pasos siguientes.
- Luego, utilizaron un modelo de clasificación de imágenes donde se dividirán todos los parches posibles clasificados en tres tipos Covid-19, IAVP e irrelevantes para la infección.
- Finalmente, para realizar el cálculo del análisis general de una muestra de TC utilizaron una función bayesiana Noisy-OR.

**Figura 23.**

*Diagrama de flujo del proceso*



Tomado de (Xu et al., 2020)

Las redes neuronales convolucionales como modelos con mecanismo de atención de ubicación pueden clasificar Covid-19, IAVP y sanos, este modelo obtuvo resultados de una tasa de precisión general del 86.7% para términos usados de todos los casos de TC del conjunto de datos.

## **Aplicación Móvil Para El Reconocimiento De Covid 19 En Imágenes De Radiografía De Tórax.**

TensorFlow Lite permite ejecutar modelos de aprendizaje profundo de redes neuronales convolucionales de forma rápida, lo que lo hace bueno a la hora de desarrollar es que no consume muchos recursos. Utiliza las mismas características que TensorFlow y Keras, pero lo hace más eficiente y más rápido de usar porque reduce el tamaño del archivo (Quezada, 2020).

Para este caso de estudio la arquitectura de la red neuronal convolucional realizará los siguientes procesos: entrenamiento y predicción.

- Primero inicia la carga de las imágenes de rayos x.
- Luego realiza el proceso de selección y extracción.
- Una vez hecho estos pasos se subirán y se creará un proyecto de Firebase.
- Cuando concluye la etapa de entrenamiento se enviará un correo indicándole al usuario que el archivo TF Lite está listo para ser descargado y desarrollado en Android Studio.
- El proyecto realizará una clasificación de dos clases como Covid y No Covid.

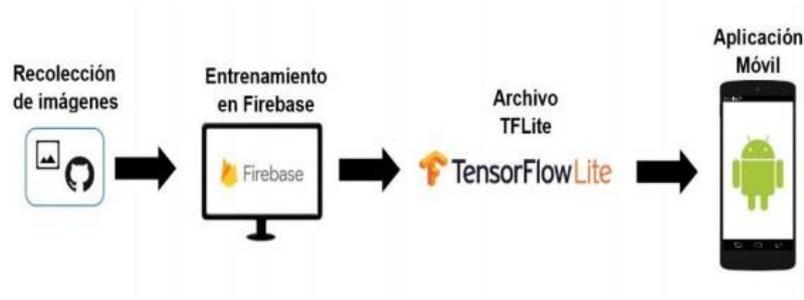
El flujo de trabajo que muestra TensorFlow en las aplicaciones móviles es la siguiente:

- Se construye el modelo inicial.
- Luego se entrena el modelo de TensorFlow Lite.
- Luego de la etapa de entrenamiento se realiza la conversión usando TensorFlow Lite.
- El resultado de la conversión es el archivo del modelo generado.
- La implementación del modelo consta de 2 despliegues, el primero es el sistema operativo Android y el otro el sistema operativo IOS.

- La aplicación Android contiene una API, estas pueden ser Java o C++, también posee un kernel y un intérprete.
- La aplicación en IOS contiene una API de C++, un kernel y un intérprete.

**Figura 24.**

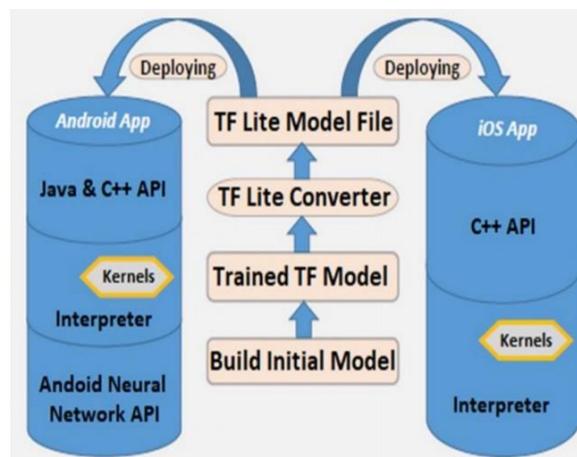
*Arquitectura de la red neuronal convolucional*



Tomado de (Quezada, 2020)

**Figura 25.**

*Diagrama de flujo de los modelos implementados en dispositivos móviles*



Tomado de (Quezada, 2020)

El conjunto de datos está organizado de la siguiente manera 1000 imágenes de entrenamiento y 100 de prueba. Los resultados obtenidos del modelo para la clasificación tienen una exactitud del 98% y una precisión del 96%.

## COVID – XR plataforma de gestión web para Detección de coronavirus en imágenes de tórax de rayos X.

Para este caso de estudio los autores propusieron el diseño e implementación de una red neuronal profunda utilizando un modelo pre entrenado VGG16 para la detección de Covid-19 utilizando imágenes de rayos X torácicas (Orozco et al., 2020).

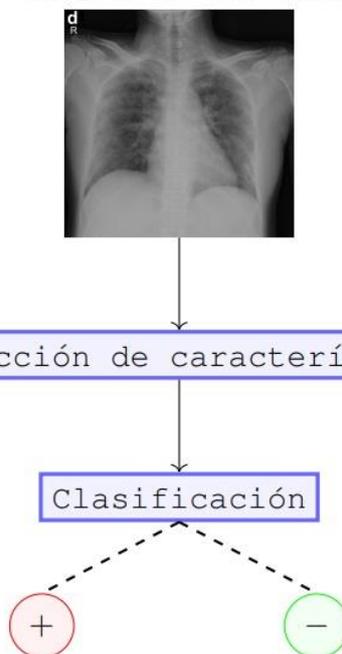
El esquema utilizado contiene las siguientes fases.

- El modelo recibe una imagen de entrada para ser procesada.
- Luego, utiliza la extracción de características de la imagen con la red neuronal. En esta fase se presentan un conjunto de capas, donde cada una de ellas tiene una función principal. Estas capas están compuestas por valores de procesamiento predeterminados que operan sobre las capas anteriores.

### Figura 26.

#### *Arquitectura de la red neuronal profunda VGG16*

Imagen de entrada



Tomado de (Orozco et al., 2020)

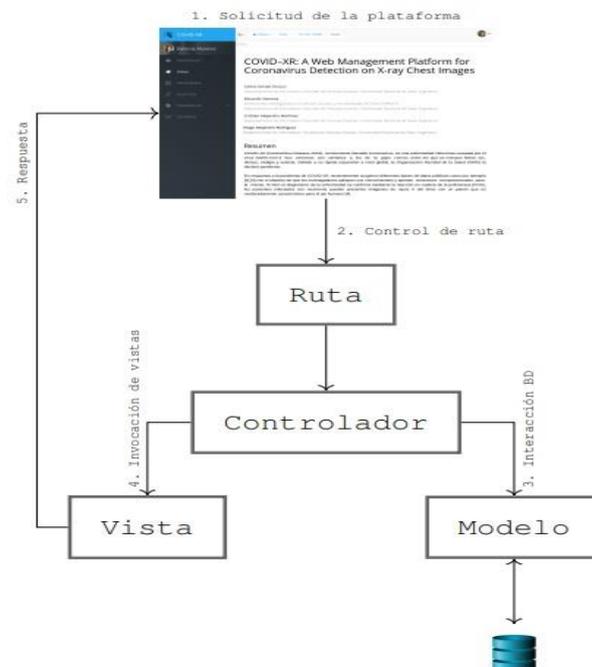
- En la fase de clasificación utiliza las redes neuronales multicapa, las cuales servirán para darle valor a las demás capas mediante.
- La capa de salida proporciona a la red los resultados esperados con etiquetas de las clases definidas Covid-19 y Normal.

El conjunto de datos establecido consta de 1000 imágenes proporcionadas de 70 individuos confirmados, frente a 1431 imágenes de neumonía catalogadas como No Covid de 1008 individuos.

Los resultados que obtuvieron en su estudio usando 3 modelos, donde todos parten del original el cual estuvo sujeto a variaciones en el mismo. El modelo C fue el que presentó mejores resultados en sensibilidad 73.20%, especificidad 96.48 % y AUC de 95.35 %. También se presenta el esquema implementación del sistema con el uso de PHP, Laravel, MySQL y Bootstrap.

**Figura 27.**

*Comunicación MVC que usa Laravel*



Tomado de (Orozco et al., 2020)

### Revisiones sistemáticas

La revisión sistemática de este trabajo de titulación se basó en la búsqueda de artículos científicos, libros educativos, tesis de grado referentes al tema para comprender, observar y analizar el uso y funcionamiento de las redes neuronales convolucionales en modelos de machine learning para la predicción de Covid-19 en imágenes médicas de radiografía de tórax. Se revisó las técnicas y métodos que fueron utilizados para el análisis de los datos, así como los resultados y conclusiones de los investigadores.

Se realizó una búsqueda considerando palabras claves, tales como: Covid-19, rayos X, inteligencia artificial, machine learning, deep learning, modelo de predicción, redes neuronales convolucionales, de entre los años 2019 y 2021.

Para el presente trabajo de titulación se utilizó 25 artículos científicos los cuales forman parte de las referencias bibliográficas, permitiendo visualizar las técnicas y metodología de trabajo.

### Meta-análisis

**Tabla 9.**

*Meta-análisis*

<b>Autores</b>	<b>Universidad/ Revista</b>	<b>Palabras Claves</b>	<b>País</b>	<b>Fecha De Publicación</b>
Antona	Universidad Autónoma de Madrid	<ul style="list-style-type: none"> <li>• Redes neuronales</li> <li>• Keras</li> <li>• Tensorflow</li> <li>• Perceptrón</li> <li>• Redes convolucionales</li> <li>• Redes LSTM</li> <li>• Diabetes</li> <li>• Housing</li> <li>• Mnist</li> <li>• Generador de texto</li> </ul>	España	Enero 2017
Avilés, Barcia	Universidad de Guayaquil	<ul style="list-style-type: none"> <li>• Análisis</li> <li>• Procesamiento digital de imágenes</li> <li>• Matlab</li> <li>• Clasificadores</li> <li>• Análisis digital de imágenes</li> </ul>	Ecuador	Noviembre 2016

<b>Autores</b>	<b>Universidad/ Revista</b>	<b>Palabras Claves</b>	<b>País</b>	<b>Fecha De Publicación</b>
Plaza	Universidad de Chile	<ul style="list-style-type: none"> <li>• Aprendizaje automático</li> <li>• Support Vector Machines</li> </ul>	Chile	Enero 2020
Kumar, Biswas	Revista Internacional de Ciencias Matemáticas, Ingeniería y Gestión	<ul style="list-style-type: none"> <li>• Coronavirus</li> <li>• COVID-19</li> <li>• Diagnóstico</li> <li>• Funciones profundas</li> <li>• SVM</li> </ul>	India	15 de abril de 2020
Bre, Gimenez, Fachinotti	Energía y Edificios	<ul style="list-style-type: none"> <li>• Coeficiente de presión</li> <li>• Ventilación natural</li> <li>• Simulación de rendimiento de edificios</li> <li>• Red neuronal artificial</li> </ul>	Argentina	22 de noviembre de 2017
Grando, Brentano	La revista brasileña de enfermedades infecciosas	<ul style="list-style-type: none"> <li>• Enfermedad por coronavirus 201</li> <li>• Neumonía viral CT</li> <li>• Diagnóstico</li> </ul>	Brasil	31 de octubre de 2020
Choy, Khalilzadeb, Samir	Departamento de Radiología de la Facultad de Medicina de la Universidad de Colorado, Aurora, Colo (J.R.G.).	<ul style="list-style-type: none"> <li>• Red neuronal convolucional</li> <li>• Machine learning</li> </ul>	EE. UU.	2018
Delgado, Ledesma	Universidad de Guanajuato	<ul style="list-style-type: none"> <li>• Inteligencia artificial</li> <li>• Red neuronal artificial</li> <li>• Electroencefalograma</li> <li>• Error cuadrático medio</li> <li>• Validación cruzada</li> </ul>	México	2019
Valle	Universidad Técnica de Ambato	<ul style="list-style-type: none"> <li>• Incendios Forestales</li> <li>• Sistema Electrónico</li> <li>• Procesamiento de Imágenes</li> </ul>	Ecuador	Mayo 2016
Palacios, Santos, Velázquez, León.	Revista Clínica Española	<ul style="list-style-type: none"> <li>• Coronavirus</li> <li>• Diagnóstico</li> <li>• Transmisión</li> <li>• cuadro clínico</li> <li>• Prevención</li> <li>• Número de reproducción básico</li> </ul>	España	4 de Marzo de 2020

<b>Autores</b>	<b>Universidad/ Revista</b>	<b>Palabras Claves</b>	<b>País</b>	<b>Fecha De Publicación</b>
Quezada	Universidad Técnica de Machala	<ul style="list-style-type: none"> <li>• Firebase</li> <li>• Radiografías</li> <li>• Codiv-19</li> <li>• Aplicación móvil</li> <li>• AutoML Vison Edge</li> </ul>	Ecuador	17 de diciembre de 2020
Meza, Estrada, Chabusa, Velasco	Revista de salud americana	<ul style="list-style-type: none"> <li>• RT-PCR</li> <li>• TAC tórax</li> <li>• Pruebas rápidas</li> <li>• Anticuerpos</li> </ul>	EE. UU.	2 de julio de 2020
Xu, Jiang, Ma, Du, Li.	Academia China de Ingeniería y Prensa de Educación Superior	<ul style="list-style-type: none"> <li>• COVID-19</li> <li>• Modelo de clasificación de ubicación-atención</li> <li>• Tomografía computarizada</li> </ul>	China	27 de junio de 2020
Yadav, Sahoo, Graham	Revista de Medicina de la Clínica Cleveland	<ul style="list-style-type: none"> <li>• Radiografía de tórax</li> <li>• Tomografía Computarizada</li> </ul>	EE. UU.	Agosto de 2020
Orozco, Xamena, Martínez, Rodríguez.	Departamento de Informática. Facultad de Ciencias Exactas. Universidad Nacional de Salta	<ul style="list-style-type: none"> <li>• Covid 19</li> <li>• Aprendizaje profundo</li> <li>• Tórax de rayos x</li> <li>• Web plataforma.</li> </ul>	Argentina	2020

**Nota:** Esta tabla refleja los trabajos referentes sobre Covid-19, rayos X, aprendizaje automático y redes neuronales convolucionales.

### **Preguntas científicas a contestarse**

- ¿El modelo de machine learning ayudará a los médicos a la toma de decisiones para la predicción de covid-19?
- ¿El modelo de predicción servirá para reducir los tiempos en que un médico tarde en generar el resultado de un diagnóstico por covid-19?

### **Variables de la Investigación**

**Variable Independiente:** Modelo de clasificación de imágenes para la predicción de Covid-19.

**Variable Dependiente:** Conjunto de datos de imágenes de rayos X.

## **Definiciones conceptuales**

### **Accuracy**

La exactitud es una métrica para evaluar modelos de clasificación. Informalmente, la exactitud es la fracción de predicciones que el modelo realizó correctamente (Martinez, 2020).

### **Algoritmo**

Son procesos que siguen un orden específico mediante un conjunto de instrucciones y reglas establecidas, es decir son pasos que se deben seguir para llegar a una la solución del problema dado.

### **Bit**

Un bit, abreviatura de dígito binario, se define como la unidad de datos más básica en telecomunicaciones e informática. Cada bit está representado por un 1 o un 0 y esto se puede ejecutar en varios sistemas a través de un dispositivo de dos estados (Marker, 2021).

### **Capas ocultas**

Una capa oculta en una red neuronal artificial es una capa entre las capas de entrada y las capas de salida, donde las neuronas artificiales toman un conjunto de entradas ponderadas y producen una salida a través de una función de activación.

### **Épocas**

Una época es un término utilizado en el aprendizaje automático e indica el número de pasadas de todo el conjunto de datos de entrenamiento que ha completado el algoritmo de aprendizaje automático. Los conjuntos de datos generalmente se agrupan en lotes (especialmente cuando la cantidad de datos es muy grande) (Tutivén, 2020).

### **Función de activación ReLU**

Función ReLU (Unidad lineal rectificadora) es una operación inteligente que reemplaza todos los valores de píxeles negativos en el mapa de características con 0 (Cornieles, 2019).

### **Hyperparámetros**

Son aquellos parámetros de un algoritmo de aprendizaje automático que controla el proceso de aprendizaje y la eficiencia del algoritmo de aprendizaje automático en su fase de entrenamiento y, por supuesto, se pueden configurar y optimizar manualmente (D. Rodríguez, 2019).

### **Métricas de evaluación**

Una métrica de evaluación cuantifica el rendimiento de un modelo predictivo. Por lo general, esto implica entrenar un modelo en un conjunto de datos, usar el modelo para hacer predicciones en un conjunto de datos de reserva que no se usó durante el entrenamiento y luego comparar las predicciones con los valores esperados en el conjunto de datos de reserva (Singh Chauhan, 2020).

### **Pixel**

El píxel se define como la más pequeña de las unidades homogéneas en un color que componen una imagen digital. Al ampliar una de estas imágenes mediante un zoom, es posible observar los píxeles que permiten la creación de la imagen. A la vista, se presentan como pequeños cuadrados o rectángulos en blanco, negro o tonos de gris (*Que Es Un Píxel* (Px), 2020).

### **Probabilidad**

La probabilidad es una rama de las matemáticas que se ocupa de calcular la probabilidad de que ocurra un evento dado, que se expresa como un número entre 1 y 0 (Raffino, 2020).

## Tensor

Un tensor es un objeto matemático similar, pero más general, a un vector y, a menudo, representado por una matriz de componentes que describen funciones relevantes para las coordenadas de un espacio. En pocas palabras, un tensor es una matriz de números que se transforman de acuerdo con ciertas reglas bajo un cambio de coordenadas (Recuero de los Santos, 2019).

## **CAPÍTULO III**

### **METODOLOGÍA DE LA INVESTIGACIÓN**

En este capítulo se presentan las fases que se realizaron para el presente trabajo de investigación los cuales constan de análisis de la data, elaboración del modelo, entrenamiento, evaluación de las métricas de calidad y desempeño, predicción (ponderación y resultados). Se especificarán los modelos de redes neuronales convolucionales por cada prueba experimental. El modelo de CNN estará sujeto a cambios a lo largo del proceso, donde dichas variaciones generales corresponden a la arquitectura y al número de imágenes del conjunto de datos. Después se analizarán los resultados obtenidos y se determinará qué modelo es el más óptimo para la investigación.

#### **Análisis de Factibilidad**

El proyecto fue realizado con el fin de cubrir la necesidad de un método de diagnóstico médico, con el objetivo de clasificar imágenes de rayos X de covid-19 a través de redes neuronales convolucionales para predecir si una persona tiene covid-19 o no. Por otra parte, el machine learning le brinda la capacidad a las máquinas que aprendan por sí mismas con el objetivo de ayudar a la toma de decisiones.

Este proceso inicia cuando el modelo se alimenta con data, aprende a medida que va entrenando, reciben órdenes por medio de algoritmos para aprender de ellos y capturar lo aprendido. El modelo realiza una clasificación de imágenes para obtener un resultado esperado. El modelo aumentará su precisión conforme más interprete los datos y los analice, para esto es necesario el uso de métricas para medir el desempeño del mismo. En el campo de la medicina la idea de crear esta herramienta tecnológica sería un gran apoyo para los

médicos de realizar un diagnóstico breve, rápido, sin costo y reduciendo los contagios mediante la prevención.

### **Factibilidad Operacional**

Este modelo funciona con el método de aprendizaje supervisado, es decir que necesita de un ente externo para ajustar los valores que permita optimizar la suficiencia reduciendo el margen de error. Esto servirá a la hora de realizar una predicción aceptando una imagen como dato de entrada, el modelo la analiza y obtiene una respuesta esperada como 1 sola salida.

### **Factibilidad Técnica**

Para este caso se analizará los ordenadores con sus especificaciones, los modelos de machine learning por lo general usan una gran cantidad de datos para ser procesados. Al ser modelos hechos con IA, depende en gran parte de una gran cantidad de recursos provenientes del computador, como lo es el procesador, la memoria RAM y el almacenamiento en disco. A continuación, se presentan las especificaciones de los computadores que trabajan en el proyecto.

**Tabla 10.**

*Características de los ordenadores*

<b>Características</b>	<b>Maquina 1 especificación</b>	<b>Maquina 2 especificación</b>
<b>Procesador</b>	Intel Core I3 CPU M370 2.40GHz	Pentium Dual-Core E5300 CPU 2.60 GHz
<b>Memoria RAM</b>	6,00 GB	3,00 GB DDR3
<b>Disco duro</b>	SSD 120 GB / HDD 500GB	500 GB
<b>Tipo de sistema</b>	Windows 10 Pro 64 bits	Windows 7 64bits

*Nota.* La elaboración es propia

En la tabla 10 se observa que los recursos para ejecutar un algoritmo de machine learning en un computador personal de esas dimensiones no funcionaría adecuadamente, ya que no cuenta con una unidad de GPU para la ejecución de múltiples tareas.

Por lo que la propuesta de tomar un entorno de desarrollo en la nube como Google Colab como herramienta Open Source es factible, ya que contiene una máquina virtual para ejecutar código Python, usar librerías de TensorFlow para el cálculo numérico y Keras para el diseño de la arquitectura del modelo. Además de eso Google Colab ofrece 12GB de RAM y 50 GB de almacenamiento en la nube, cuenta con GPU lo que lo hace ideal para el desarrollo de modelos computacionales. En la tabla 11 se aprecia el entorno de desarrollo.

**Tabla 11.**

*Entorno de desarrollo*

<b>Entorno de desarrollo y almacenamiento</b>	<b>Versión</b>
<b>Python</b>	3.7
<b>TensorFlow</b>	2.0.0
<b>Keras</b>	2.00
<b>Google drive</b>	2.21.02

*Nota.* La elaboración es propia

### **Factibilidad Legal**

En esta etapa se conocerá si el proyecto cumple con los estándares dentro del marco legal, dicha fundamentación se encuentra en el anexo 4. En caso de que estas normas se irrespeten se asumirá la responsabilidad del acto y las consecuencias por el hecho causado

La fundamentación legal autoriza a los desarrolladores de software y de tecnologías afines acogerse a los derechos de autor, por el uso de su código fuente y la documentación descrita dentro del plano académico e investigativo.

## Factibilidad Económica

En esta etapa se detallan los recursos que se destinaron para la realización del modelo.

A continuación, se muestran los costos estimados para llevar a cabo la investigación.

**Tabla 12.**

*Presupuesto del proyecto*

Presupuesto				
	Descripción	Cantidad	Valor Unitario	Valor total
<b>Talento humano</b>	Desarrollador	2	850	1700
<b>Equipos</b>	Laptop	1	600	600
	PC	1	500	500
	Teléfono	2	200	400
<b>Entorno de desarrollo</b>	Colab	1	0	0
	Python	1	0	0
<b>Suministros</b>	Luz Electrica	2	30	60
	Internet	2	30	60
<b>Total</b>				3320

*Nota.* La elaboración es propia

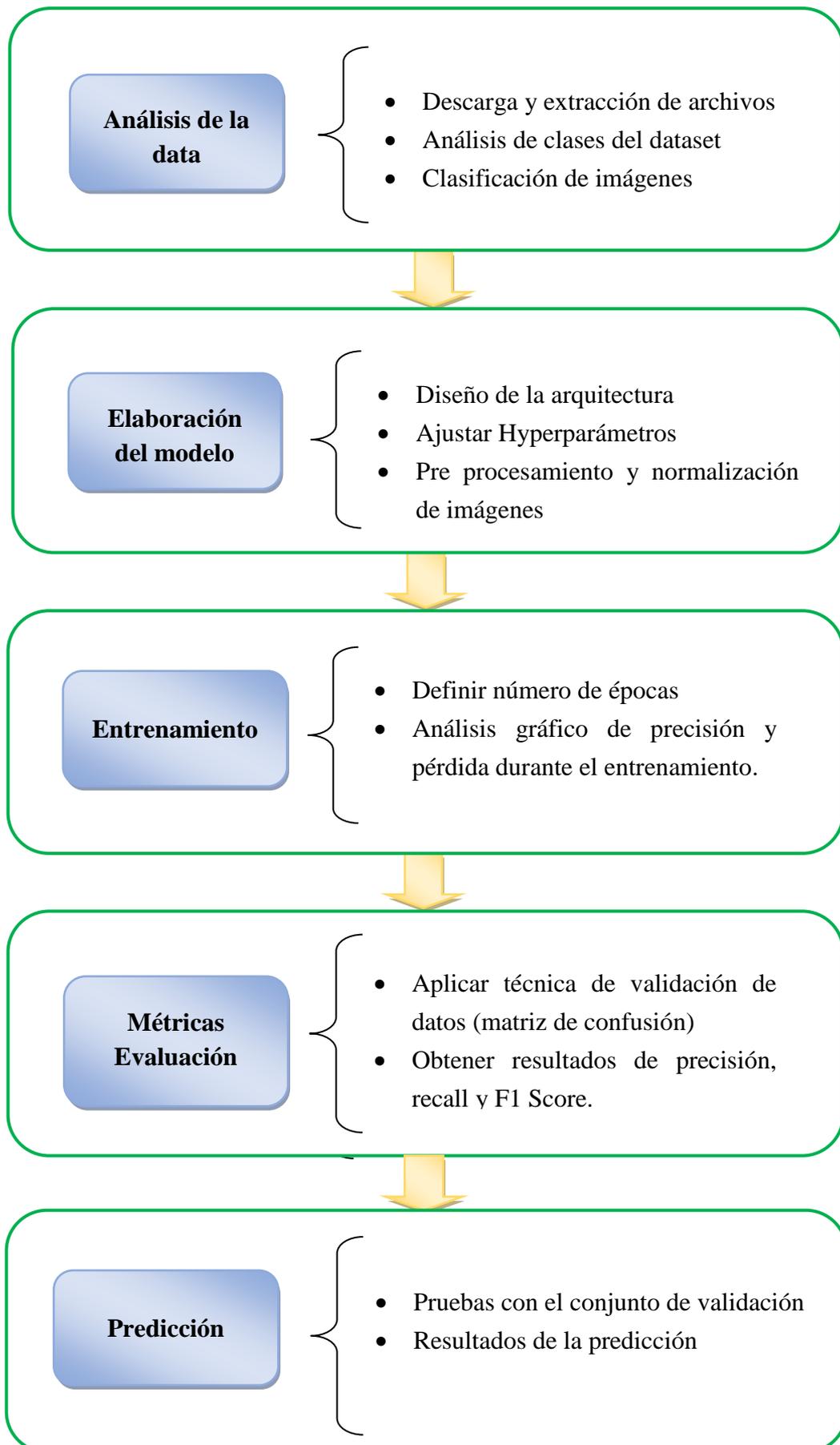
## Tipo de Investigación

### Investigación Experimental

Según Fidias Arias, autor del libro El Proyecto de Investigación 6ta Edición (2012), “la investigación experimental es un proceso que consiste en someter a un objeto o grupo de individuos en determinadas condiciones, estímulos o tratamiento (variable independiente), para observar los efectos o reacciones que se producen (variable dependiente)” (Arias, 2012).

Según (Guevara et al., 2020) autores del artículo científico Metodologías de investigación educativa, "en la investigación de enfoque experimental el investigador manipula una o más variables de estudio, para controlar el aumento o disminución de esas variables y su efecto en las conductas observadas".

## Diseño Metodológico de la Investigación



## **Metodología de Investigación**

### **Definición del problema**

El primer paso para el aprendizaje automático consiste en definir un objetivo, es decir, plantear un problema que requiera una solución a corto, medio y largo plazo.

Existen diferentes formas de estructurar el objetivo de un proyecto en el que se va a implementar tecnología Machine Learning. Se establecerá el objetivo con base a las necesidades y posibilidades en función de los datos disponibles.

Para definir el objetivo, algunas de las preguntas que debemos plantear son:

#### **¿Cuál es el principal objetivo?**

Elaborar un modelo de machine learning para la predicción de pacientes con Covid-19 por medio del análisis de imágenes de rayos X a través de redes neuronales artificiales.

#### **¿Qué se va a predecir?**

Se realizará la predicción de Covid-19 utilizando imágenes de rayos X del tórax.

#### **¿Qué se pretende enseñar a la máquina?**

Que aprenda y extraiga los patrones más significativos de una imagen con Covid-19, con el fin de entrenar y evaluar el funcionamiento del modelo para realizar predicciones.

#### **¿Es posible alcanzar el objetivo con los datos existentes?**

La clasificación de las imágenes estará destinada en 80% entrenamiento y 20% validación, dependiendo del número de imágenes del conjunto de datos.

#### **¿Cómo se van a medir los resultados?**

Los resultados se medirán con base a la técnica de validación para modelos de clasificación binaria (matriz de confusión).

Los pasos para realizar un modelo de machine learning son los siguientes: análisis de la data, elaboración del modelo, entrenamiento, métricas de evaluación y predicción.

## Fase 1

### Análisis de la Data

Para el presente estudio de investigación se procede en la búsqueda del conjunto de datos (*dataset*), usando el navegador web se realiza la búsqueda de repositorios que sean de acceso público, esta información es completamente gratuita. Existen conjunto de datos que contienen más de 1000 elementos, dependiendo de la cantidad y dimensión de las imágenes el tamaño del dataset puede superar los 2GB.

#### Descarga y Extracción de Archivos

Se realiza la descarga de los *dataset* de *GitHub* (Tutivén, 2020) y *Kaggle* (Cohen et al., 2020) (Rahman et al., 2020). Se extrae y se visualiza el contenido del conjunto de datos. Estas bases de datos son públicas y de fácil acceso, se pueden obtener varias bases de datos las cuales pertenecen a una comunidad científica de datos.

#### Análisis del Dataset

Al revisar el conjunto de datos, se pudo conocer que dentro de la carpeta principal llamada data existieron dos clases, una para entrenamiento y otra para prueba, dividida en 3 subcarpetas: Covid-19, Normal y Neumonía.

#### Clasificación de Imágenes.

EL análisis de la data consiste en tomar esas carpetas descargadas como referencia de las clases anteriores para crear un nuevo conjunto de datos. Luego se analizan las clases pertinentes para el proyecto y se separa el conjunto de datos de la siguiente manera: con 80% entrenamiento y 20% validación. Para el proyecto solo se tomarán las clases Covid-19 y Normal. Se observa en la figura 28.

**Figura 28.**

*Visualización de imágenes de rayos X de tórax*



**Covid-19**



**Normal**

Tomado de (Chowdhury et al., 2020)

**Importar Archivos a Google Drive**

Una vez que se tiene las carpetas comprimidas se procede a almacenar los archivos a la nube de google drive.

**Conexión de Google Colab a Google Drive**

Para la creación del cuaderno en Google Colab se realiza un montado para tener acceso a los datos que se encuentran dentro del almacenamiento en la nube para después utilizarlo en el desarrollo del modelo.

**Selección del Dataset**

Se utiliza la reestructuración y normalización, para este caso las imágenes en los dataset por lo general están formadas por dimensiones de 1024 x 1024 pixeles en formatos JPG y PNG, se realiza la normalización de las imágenes a una dimensión de 224 x 224 pixeles manteniendo cualquiera de los formatos, este proceso permite trabajar con lucidez usando las redes neuronales convolucionales. En la tabla 13 se puede observar cómo están distribuidos los *dataset* y el número total de imágenes para cada modelo de prueba.

**Tabla 13.***Estructura de los conjuntos de datos*

BASE DE DATOS DE IMÁGENES	MODELO	CONJUNTO DE DATOS		TOTAL DE IMÁGENES DATASET
		TRAIN	VALIDATION	
GITHUB	A	154	79	233
KAGGLE	B	1091	311	1402
KAGGLE	C	1812	679	2491

*Nota.* La elaboración es propia

A continuación, en la tabla 14 se puede observar la división de las imágenes según las clases Covid-19 y Normal, y el total general de todas las imágenes recopiladas.

**Tabla 14.***Estructura de los conjuntos de datos por clases*

MODELO	TRAIN		VALIDATION		TOTAL
	COVID-19	NORMAL	COVID-19	NORMAL	
A	77	77	16	63	233
B	530	561	116	195	1402
C	902	910	241	438	2491
<b>TOTAL</b>	<b>1509</b>	<b>1548</b>	<b>373</b>	<b>696</b>	<b>4126</b>

*Nota.* La elaboración es propia

## Fase 2

### Elaboración del Modelo

Para realizar el modelo de predicción de Covid-19 usando redes neuronales convolucionales, se utilizó el entorno de desarrollo de Google Colaboratory. Esta herramienta fue creada por Google porque permite trabajar con el lenguaje de programación Python, ya que ofrece un entorno de ejecución gratuito de unidades de procesamiento gráfico (GPU) que permite acelerar los procesos de entrenamiento presentes en los modelos. En la tabla 15 se muestran las pruebas experimentales realizadas por cada modelo, mediante las características asignadas se verificará que modelo obtiene el mejor resultado.

**Tabla 15.***Pruebas experimentales*

CARACTERISTICAS DE UNA CNN	MODELOS		
	A	B	C
NEURONAS DE LA CAPA DE ENTRADA	80	160	320
CAPAS OCULTAS	3	6	9
NUMERO DE ÉPOCAS	5	10	15
TOTAL DE CAPAS DE LA CNN	5	8	11
TOTAL DE PARAMETROS EJECUTADOS	19.391.161	21.401.551	32.712.737

*Nota.* La elaboración es propia

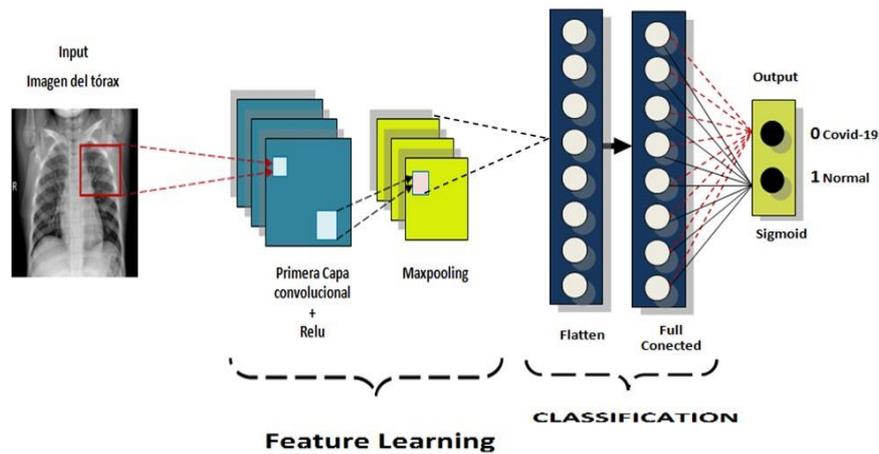
**Diseño de las Arquitecturas**

Para realizar las pruebas experimentales a cada modelo se consideró realizar cambios en la arquitectura de la red neuronal convolucional, como son las capas ocultas, capas Maxpooling y en las capas totalmente conectadas. Una de las funciones más importantes que poseen las redes neuronales convolucionales para realizar modelos computacionales son los siguientes:

- Realiza la detección y genera resultados altamente precisos.
- Aprenden características de forma directa sin necesidad de extraerlas manualmente y se entrenan nuevamente para realizar nuevas tareas aprovechando las redes existentes.

**Modelo A**

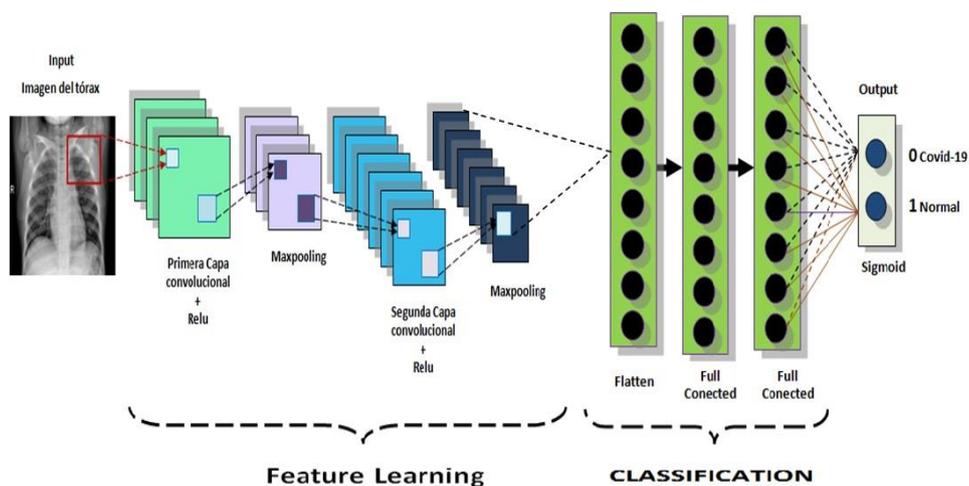
El primer modelo experimental utilizó una arquitectura con 1 capa de entrada convolucional conforme a las dimensiones de las imágenes (224 x 224 x 3), 3 capas ocultas (1 de convolución, 1 de Maxpooling y 1 capa Full conectada) y 1 capa de salida con 2 neuronas full conectada acorde a las 2 opciones posibles de resultado de clasificación. Se observa en la figura 29.

**Figura 29.***Arquitectura del modelo A*

*Nota.* La elaboración es propia

**Modelo B**

El segundo modelo experimental utilizó una arquitectura con 1 capa de entrada convolucional conforme a las dimensiones de las imágenes (224 x 224 x 3), 6 capas ocultas (2 de convolucionales, 2 de Maxpooling y 2 capa Full conectad) y 1 capa de salida con 2 neuronas full conectad acorde a las 2 opciones posibles de resultado de clasificación. Se observa en la figura 30.

**Figura 30.***Arquitectura del modelo B*

*Nota.* La elaboración es propia

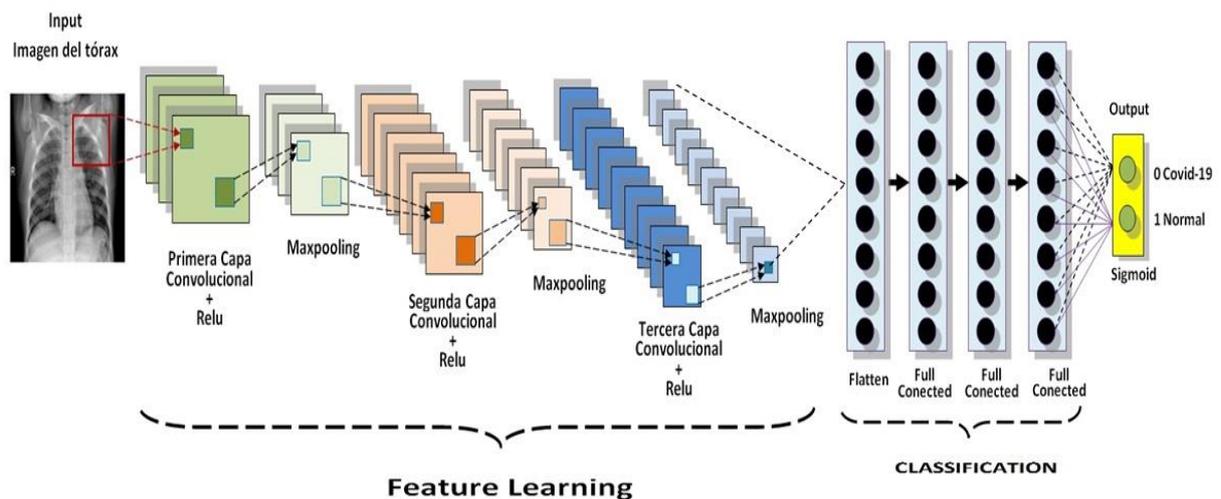
### Modelo C

El tercer modelo experimental utilizó una arquitectura con 1 capa de entrada convolucional conforme a las dimensiones de las imágenes (224 x 224 x 3), 9 capas ocultas (3 de convolucionales, 3 de Max-Pooling y 3 capa Full conectad) y 1 capa de salida con 2 neuronas full conectad acorde a las 2 opciones posibles de resultado de clasificación. Se observa en la figura 31.

- En cada diseño de red se han implementado un número determinado de capas y neuronas. La red neuronal en Keras, ha definido la secuencia de capas, donde cada neurona está relacionada y densamente conectada con la capa anterior.
- Para este caso en Keras se trata de un modelo secuencial, es decir, añade capas en las cuales cada una de ellas destila proporcionalmente los datos de entrada para obtener un resultado esperado.

**Figura 31.**

#### Arquitectura del modelo C



*Nota.* La elaboración es propia

## Ajustes de Hyperparámetros

Una vez finalizada la etapa de diseño de la arquitectura, para el proceso de aprendizaje se definirán 2 argumentos importantes. Uno de estos argumentos es la función de pérdida que servirá para evaluar el grado de error entre las salidas calculadas y los datos deseados.

- El algoritmo de optimización de la red neuronal será de utilidad porque le permitirá calcular los pesos de los parámetros de los datos de entrada y la función de pérdida definida.
- Estos argumentos son denominados hyperparámetros y sirven para optimizar los valores en la curva de aprendizaje. Estos valores son ajustables y el desarrollador puede ajustarlos para obtener mejores resultados. Se observa en la tabla 16 los cambios efectuados para cada modelo.

**Tabla 16.**

*Hyperparámetros utilizados*

HYPERPARÁMETROS	MODELOS		
	A	B	C
<b>Modo de color</b>	Modo RGB	Modo RGB	Modo RGB
<b>Data augmentation</b>	Giro Horizontal con variación de zoom	Giro Horizontal con variación de zoom	Giro Horizontal con variación de zoom
<b>Función de coste / Loss</b>	Bynary Cross_Entropy	Bynary Cross_Entropy	Bynary Cross_Entropy
<b>Optimizador</b>	ADAM	ADAM	ADAM
<b>Taza de aprendizaje -Metrics</b>	ACC	ACC	ACC

*Nota.* La elaboración es propia

### *Función de pérdida (Loss)*

Mejor conocida como función objetivo permite conocer en la etapa de entrenamiento de una red neuronal que tan próximo se encuentra al valor mínimo, también sirve para saber

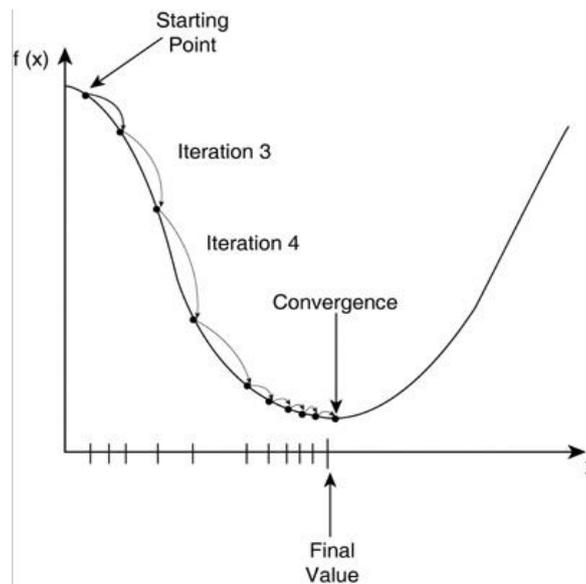
qué tanto es el error aproximado que se desea obtener para saber con exactitud si el modelo está aprendiendo correctamente, de otra forma más exacta muestra una salida del resultado esperado (Gavilán, 2020).

La función consta de dos términos:

1. Término de error, sirve para evaluar el ajuste de la salida de la red neuronal referente al conjunto de datos.
2. Término de regularización, se utiliza para evitar el sobreaprendizaje a través de la comprobación de la dificultad de la red neuronal.

### Figura 32.

*Función de pérdida*



Tomado de (Sancho, 2020)

### ***Binary cross\_entropy***

Esta función de coste calcula la entropía cruzada entre las etiquetas verdaderas con las etiquetas predichas. Dentro del contexto, sirve para clasificar variables binarias donde solo existan 2 clases posibles. Por ejemplo, estos valores posibles son 0 y 1, para cada ejemplo, de la predicción debería haber un único valor de salida.

### ***Optimizador ADAM***

Este algoritmo de optimización es utilizado para solucionar la incertidumbre sobre la fijación que hay entre el ratio de aprendizaje del descenso del gradiente estocástico (SGD) para actualizar de forma iterativa en función de los datos de entrenamiento.

En la tabla 16 se pueden observar los hiperparámetros que se utilizó para determinar el comportamiento del modelo y su funcionamiento a través de valores ajustables.

### **Pre-Procesamiento y Normalización**

#### ***Normalización de los datos***

Distribuye todos los datos para hacerlos más homogéneos. Se realizan estos pasos para hacer más suave la curva de la generación, para que todas las imágenes tengan el mismo modelo de color y la misma resolución, para este caso se escogió un solo canal., esto evitará que el sistema no presente sobreajuste y la red extraiga los valores correctamente, las imágenes están en modelo de color RGB que van de 0 (negro) a 255 (blanco) pasando las imágenes a un solo canal, es decir a escala de grises donde 0 significa (ausencia de color) y 1 (luminosidad).

#### ***Data augmentation***

Sirve para ajustar la imagen en modo horizontal, de manera que la imagen no presente variaciones de acuerdo a la posición en que se encuentra, para este caso se usa este hiperparámetro para que todas las imágenes tengan el mismo tamaño y sean más visibles.

- Por medio de esta técnica se evita el sobreajuste que normalmente se presenta en las redes neuronales convolucionales, porque ayuda a tener un mejor rendimiento en la etapa de entrenamiento.

#### ***Batch\_size***

Es el tamaño de lote, es decir es la cantidad de imágenes que pasará el modelo en los pasos por épocas.

## Fase 3

### Entrenamiento

En el proceso de aprendizaje la red neuronal ejecutará los parámetros de la función de pérdida y el optimizador durante el entrenamiento previo. El algoritmo compara los resultados obtenidos con los esperados, calcula la pérdida en cuanto al optimizador para reducir el error en cada iteración.

Para el estudio del primer modelo se utilizó 3 capas ocultas para una arquitectura básica, utilizando una función ReLU para la capa convolucional y para la capa densa. Luego se realiza el entrenamiento del modelo A, se lo define con 5 épocas, también se utilizó 4 pasos por época y el resultado de la ejecución fue de 36.588 seg aproximadamente como prueba experimental. Se observa en la figura 33.

#### Figura 33.

##### *Resultado del entrenamiento del modelo A*

```
[ ] #Training and validation stages
    history = model.fit(
        train_generator,
        steps_per_epoch=4,
        epochs= 5,
        validation_data=validation_generator,
        validation_steps=60)

Epoch 1/5
4/4 [=====] - 14s 2s/step - loss: 12.4112 - acc: 0.5596 - val_loss: 0.8147 - val_acc: 0.7333
Epoch 2/5
4/4 [=====] - 7s 2s/step - loss: 1.2600 - acc: 0.4687 - val_loss: 0.6665 - val_acc: 0.5333
Epoch 3/5
4/4 [=====] - 7s 2s/step - loss: 0.6702 - acc: 0.6124 - val_loss: 0.6432 - val_acc: 0.7833
Epoch 4/5
4/4 [=====] - 7s 2s/step - loss: 0.6772 - acc: 0.6034 - val_loss: 0.6109 - val_acc: 0.8333
Epoch 5/5
4/4 [=====] - 7s 2s/step - loss: 0.6743 - acc: 0.5809 - val_loss: 0.4817 - val_acc: 0.9333
```

*Nota.* La elaboración es propia

### Dropout

Este método se encarga de deshabilitar neuronas con el fin de reducir el sobreajuste en las imágenes, las neuronas cercanas no tienen una conexión directa con las neuronas deshabilitadas, lo cual les permite trabajar mejor de forma única para obtener patrones

determinados en la etapa de entrenamiento. Toma un valor estimado entre 0 y 1, donde 0.5 indica por defecto la mitad de neuronas deshabilitadas, 1 hace referencia al valor máximo del mayor número de neuronas deshabilitadas. En Keras es lo opuesto, mientras el valor este por debajo de 0.5 mayor serán las neuronas deshabilitadas. Por lo tanto, los modelos en Keras siempre utilizan valores altos para las capas ocultas y en término medio para las capas densas (V. Rodríguez, 2018).

### **Sobreajuste**

El sobreajuste consiste en la pérdida de propagación, esto quiere decir que el modelo tendrá un comportamiento bueno en el proceso de aprendizaje, pero no podrá reconocer nuevos patrones por errores, ruido o desperfectos que contenga una imagen.

En la etapa de aprendizaje se evalúan ciertos criterios importantes a considerar para la etapa de entrenamiento:

- El objetivo del entrenamiento es establecer los valores para el vector de los pesos, en los cuales el error cometido permite conocer si los datos de entrenamiento son mínimos.
- Al realizar el cálculo de los pesos, la red estará lista para recibir nuevos datos, es decir se tomarán los datos de validación que el modelo no haya visto, para poder deducir si los patrones aprendidos durante el entrenamiento son correctos. Con esto se comprueba sí el modelo funciona con entradas que no se han asignado.
- Sí la red falla o no es capaz de identificar los patrones, tendrá que pasar por entrenamiento una vez más hasta que logre ajustarse. Sí la red arroja resultados correctos significa que el modelo está listo para funcionar ya que se podrá incorporar nuevos datos de entrada y predecirlos.

Para el estudio del modelo B, se intensificó a escala intermedia la complejidad al añadir 3 capas ocultas más, obteniendo un total de 6 capas ocultas. Cada capa convolucional

consta de una función ReLU y una función de regularización (*Dropout*) de 0.15 para reducir el sobreajuste (*overfitting*) y para las dos capas densas se usó un valor de 0.25 como regularizador.

El entrenamiento se definió con 10 épocas y 7 pasos por época, obteniendo un tiempo de ejecución de 166.408 seg. Se observa en la figura 34.

### Figura 34.

#### Resultado del entrenamiento del modelo B

```
[ ] #Training and validation stages
history = model.fit(
    train_generator, #carpeta de entrenamiento
    steps_per_epoch=7,
    epochs= 10,
    validation_data=validation_generator,
    validation_steps=60)

Epoch 1/10
7/7 [=====] - 25s 2s/step - loss: 1.1334 - acc: 0.4956 - val_loss: 0.7187 - val_acc: 0.0000e+00
Epoch 2/10
7/7 [=====] - 16s 2s/step - loss: 0.7130 - acc: 0.5915 - val_loss: 0.6625 - val_acc: 0.9667
Epoch 3/10
7/7 [=====] - 15s 2s/step - loss: 0.6438 - acc: 0.6394 - val_loss: 0.7421 - val_acc: 0.3833
Epoch 4/10
7/7 [=====] - 16s 2s/step - loss: 0.5521 - acc: 0.7191 - val_loss: 0.3115 - val_acc: 1.0000
Epoch 5/10
7/7 [=====] - 15s 2s/step - loss: 0.6399 - acc: 0.6716 - val_loss: 0.2255 - val_acc: 0.9667
Epoch 6/10
7/7 [=====] - 15s 2s/step - loss: 0.3935 - acc: 0.8226 - val_loss: 0.3026 - val_acc: 0.9333
Epoch 7/10
7/7 [=====] - 15s 2s/step - loss: 0.4021 - acc: 0.8136 - val_loss: 0.3644 - val_acc: 0.9000
Epoch 8/10
7/7 [=====] - 15s 2s/step - loss: 0.3090 - acc: 0.8828 - val_loss: 0.3168 - val_acc: 0.8500
Epoch 9/10
7/7 [=====] - 15s 2s/step - loss: 0.3932 - acc: 0.8663 - val_loss: 0.1402 - val_acc: 0.9500
Epoch 10/10
7/7 [=====] - 16s 2s/step - loss: 0.3655 - acc: 0.8495 - val_loss: 0.2631 - val_acc: 1.0000
```

*Nota.* La elaboración es propia

Para finalizar en el estudio del modelo C se agregan 3 capas ocultas más obteniendo un total de 9 capas ocultas. Cada capa convolucional consta de una función ReLU y una función de regularización (*Dropout*) de 0.25 y 0.5 para las capas densas. Se realizó el entrenamiento con 15 épocas, utilizando 7 pasos por época. Luego del entrenamiento, se obtuvo como resultado un tiempo de ejecución de 177.644 seg. Se observa en la figura 35.

**Figura 35.**

*Resultado del entrenamiento del modelo C*

```
#Training and validation stages
history = model.fit(
    train_generator, #carpeta de entrenamiento
    steps_per_epoch=7,
    epochs= 15,
    validation_data=validation_generator,
    validation_steps=60)

Epoch 1/15
7/7 [=====] - 24s 2s/step - loss: 1.4175 - acc: 0.5024 - val_loss: 0.7020 - val_acc: 0.0000e+00
Epoch 2/15
7/7 [=====] - 14s 2s/step - loss: 0.6878 - acc: 0.5568 - val_loss: 0.7098 - val_acc: 0.0000e+00
Epoch 3/15
7/7 [=====] - 10s 1s/step - loss: 0.6832 - acc: 0.5367 - val_loss: 0.7145 - val_acc: 0.0000e+00
Epoch 4/15
7/7 [=====] - 11s 1s/step - loss: 0.6842 - acc: 0.6087 - val_loss: 0.6555 - val_acc: 0.9833
Epoch 5/15
7/7 [=====] - 11s 1s/step - loss: 0.7133 - acc: 0.5628 - val_loss: 0.6777 - val_acc: 0.7000
Epoch 6/15
7/7 [=====] - 11s 1s/step - loss: 0.5456 - acc: 0.7783 - val_loss: 0.2173 - val_acc: 0.9833
Epoch 7/15
7/7 [=====] - 11s 1s/step - loss: 0.3625 - acc: 0.8760 - val_loss: 0.4916 - val_acc: 0.9000
Epoch 8/15
7/7 [=====] - 10s 1s/step - loss: 0.3284 - acc: 0.8910 - val_loss: 0.3168 - val_acc: 0.9500
Epoch 9/15
7/7 [=====] - 10s 1s/step - loss: 0.3045 - acc: 0.9101 - val_loss: 0.1600 - val_acc: 0.9500
Epoch 10/15
7/7 [=====] - 11s 1s/step - loss: 0.2871 - acc: 0.9176 - val_loss: 0.1998 - val_acc: 0.9333
Epoch 11/15
7/7 [=====] - 11s 1s/step - loss: 0.2035 - acc: 0.9117 - val_loss: 0.2916 - val_acc: 0.9167
Epoch 12/15
7/7 [=====] - 11s 1s/step - loss: 0.2607 - acc: 0.9283 - val_loss: 0.2996 - val_acc: 0.9333
Epoch 13/15
7/7 [=====] - 11s 1s/step - loss: 0.1961 - acc: 0.9231 - val_loss: 0.1246 - val_acc: 1.0000
Epoch 14/15
7/7 [=====] - 11s 1s/step - loss: 0.1922 - acc: 0.9349 - val_loss: 0.2509 - val_acc: 0.9667
Epoch 15/15
7/7 [=====] - 11s 1s/step - loss: 0.2981 - acc: 0.9024 - val_loss: 0.2600 - val_acc: 0.9667
```

**Nota.** La elaboración es propia

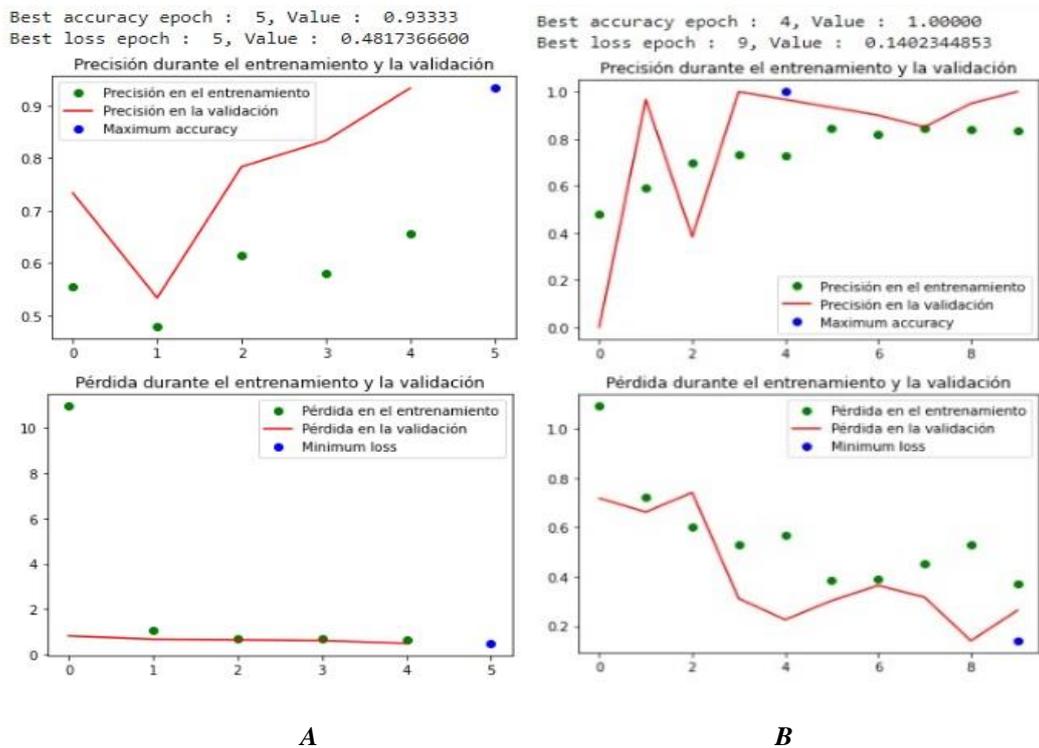
Para realizar la clasificación con una CNN es importante conocer la profundidad y los parámetros de entrada de esta. Las capas de entrada de la CNN definirán el tamaño de la imagen a ser clasificada. Los parámetros establecidos de las CNN utilizadas para la clasificación de imágenes pueden consultarse en la documentación de TensorFlow y Keras (*TensorFlow*, s. f.).

### **Análisis Gráfico de Precisión y Pérdida Durante el Entrenamiento**

Para el caso del modelo A se conoce que obtuvo una precisión máxima (*Maximum accuracy*) de 0.91667, mientras que para la función de pérdida (*Loss*) se aprecia que obtuvo un valor de 0.3283. Estos resultados se presentan en la época 5 para ambas funciones, considerada la mejor época de todas las iteraciones. Se observa en la figura 36.

Figura 36.

*Precisión y pérdida durante el entrenamiento de los modelos A y B*

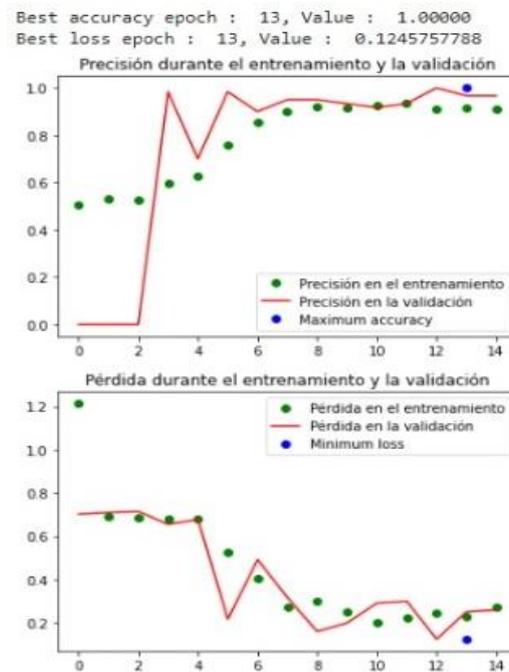


*Nota.* La elaboración es propia

- Para el modelo B se obtuvo una precisión máxima de 1.0000, mientras que para la función de pérdida se obtuvo un valor de 0.1402. Estos resultados se mostraron en la época 4 para el accuracy durante el entrenamiento, y mejor época para la pérdida durante el entrenamiento y la validación. Se observa en la figura 36.
- El modelo C obtuvo una precisión máxima de 1.0000, mientras que para la función de pérdida obtuvo un valor de 0.1245. Estos resultados se detallan en la época 13 para ambas funciones durante el entrenamiento y la validación. Se observa en la figura 37.

**Figura 37.**

*Precisión y pérdida durante el entrenamiento del modelo C*



*Nota.* La elaboración es propia

#### Fase 4

#### Evaluación de las Métricas de Calidad y Desempeño

Para verificar que tan bueno fueron los resultados obtenidos para cada modelo, se usaron las métricas de calidad para realizar las comparaciones de los valores de precisión, sensibilidad (*recall*) y F1-Score para determinar qué modelo es el más óptimo para la predicción de Covid-19.

Para utilizar este método de evaluación de un modelo de clasificación se necesita separar el dataset principal en dos conjuntos de datos.

- Train 80%
- Validation 20%

El conjunto de entrenamiento sirve para ajustar el modelo, lo que significa que en la fase de entrenamiento el algoritmo irá aprendiendo de las características extraídas de las imágenes.

El conjunto de validación ayuda a evaluar los ajustes referentes al modelo y a su vez ajustar los parámetros del modelo.

### Matriz de Confusión

Este elemento es muy importante porque sirve para detallar explícitamente que clases se confundieron con otra, su funcionamiento va acompañado de las métricas de calidad para determinar si el modelo funciona bien. Está compuesto por filas y columnas, en donde las filas son las clases reales y las columnas son clases pronosticadas por el modelo (Recuerdo de los Santos, 2018).

**Tabla 17.**

*Modelo de una matriz de confusión*

		predicción	
		0	1
realidad	0	TN	FP
	1	FN	TP

Tomado de (Martinez, 2020)

Donde:

- **True Positive** es la cantidad de positivos que fueron clasificados correctamente como positivos por el modelo.
- **True Negative** es la cantidad de negativos que fueron clasificados correctamente como negativos por el modelo.
- **False Negative** es la cantidad de positivos que fueron clasificados incorrectamente como negativos.
- **False Positive** es la cantidad de negativos que fueron clasificados incorrectamente como positivos.

### **Precisión**

Se refiere a la calidad de los resultados obtenidos luego de realizar las clasificaciones de casos positivos y el total de predicciones positivas de forma correcta, un valor que se acerca más a una predicción verdadera (Recuero de los Santos, 2020).

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

### **Sensibilidad (Recall)**

Esta métrica indica la cantidad de datos de clase positiva que el modelo ha sido capaz de identificar (Martinez, 2020).

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

### **F1-score**

Esta métrica sirve para combinar y comparar el rendimiento de los valores de precisión y recall (Martinez, 2020).

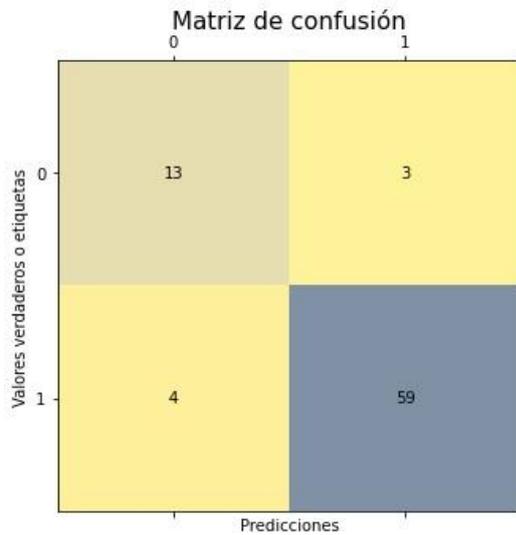
$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

A continuación, se presentan los cálculos de las métricas por cada modelo:

El modelo A presenta los siguientes valores, donde la diagonal principal TN= 13 y TP= 59 corresponden a los valores estimados de forma correcta por el modelo.

Mientras que la otra diagonal, representa los casos en los que el modelo se ha equivocado FN= 4 y FP= 3.

Se puede observar en la figura 38.

**Figura 38.***Modelo A*

$$Precision = \frac{59}{59 + 3} = 0.9516129$$

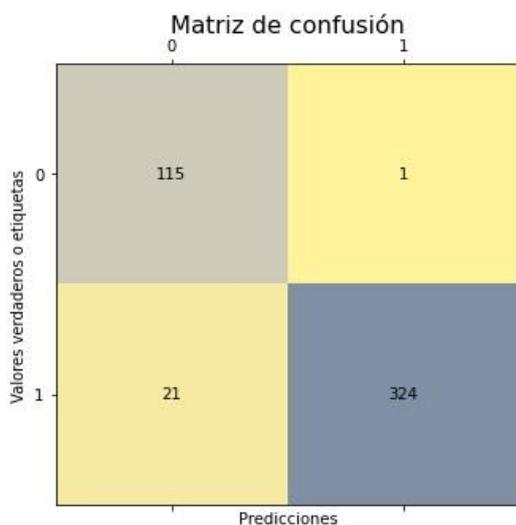
$$Recall = \frac{59}{59 + 4} = 0.9365079$$

$$F1 = 2 * \frac{0.9516129 * 0.9365079}{0.9516129 + 0.9365079} = 0.9440089$$

*Nota.* La elaboración es propia

El modelo B presenta los siguientes valores, donde la diagonal principal TN= 115 y TP= 324 corresponden a los valores estimados de forma correcta por el modelo.

Mientras que la otra diagonal, representa los casos en los que el modelo se ha equivocado FN= 21 y FP= 1. Se observa el resultado en la figura 39.

**Figura 39.***Modelo B*

$$Precision = \frac{324}{324 + 1} = 0.9969231$$

$$Recall = \frac{324}{324 + 21} = 0.9391304$$

$$F1 = 2 * \frac{0.9969231 * 0.9391304}{0.9969231 + 0.9391304} = 0.9671642$$

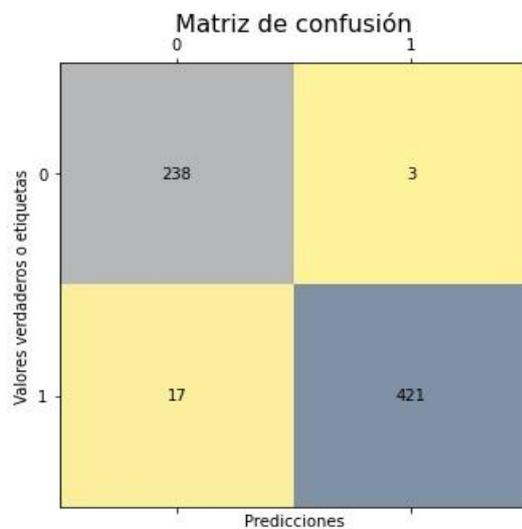
*Nota.* La elaboración es propia

El modelo C presenta los siguientes los valores, donde la diagonal principal TN= 238 y TP= 421 corresponden a los valores estimados de forma correcta por el modelo.

Mientras que la otra diagonal, representa los casos en los que el modelo se ha equivocado FN= 17 y FP= 3. Se observa el resultado en la figura 40.

**Figura 40.**

*Modelo C*



$$Precision = \frac{421}{421 + 3} = 0.9929245$$

$$Recall = \frac{421}{421 + 17} = 0.9611872$$

$$F1 = 2 * \frac{0.9929245 * 0.9611872}{0.9929245 + 0.9611872} = 0.9767981$$

*Nota.* La elaboración es propia

### Pruebas de Modelos por Conjuntos de Datos

El modelo A utilizó un total de 233 imágenes como conjunto de datos extraídas de la plataforma de GitHub. El error aproximado del modelo A para las métricas de precisión, sensibilidad y F1-Score son: 0,0483871, 0,0634921 y 0,0559911.

- Este modelo obtuvo un valor de precisión del 0.9516 con una sensibilidad (Recall) del 0.9365 y un F1-Score de 0.9440. Tiene un rendimiento bueno para ser el primer experimento, pero no llegó a su punto máximo de eficiencia ya que presenta sobreajuste, pero aun así con pocas épocas pudo realizar una clasificación aceptable.

Los resultados se aprecian en la tabla 18.

**Tabla 18.***Resultados obtenidos por cada modelo*

DATASETS	MODELO	NUMERO DE IMÁGENES	PRECISIÓN (ACC)	SENSIBILIDAD (RECALL)	F1-SCORE
GITHUB	A	233	0.9516129	0.9365079	0.9440089
KAGGLE	B	1402	0.9969231	0.9391304	0.9671642
KAGGLE	C	2401	0.9929245	0.9611872	0.9767981

**Nota.** La elaboración es propia

El modelo B a diferencia del modelo A utilizó un conjunto de datos de alrededor de 1402 imágenes extraídas de la plataforma Kaggle. El error aproximado del modelo B para las métricas de precisión, sensibilidad y F1-Score son: 0,0030769, 0,0608696 y 0,0328358.

- Este modelo tiene una pequeña mejora porque obtuvo un valor aproximado de 0.0453102 para la precisión, también obtuvo un aumento en la sensibilidad con un valor aproximado de 0.0026225, y en cuanto al promedio del F1-Score también obtuvo un ligero aumento en el resultado con un valor de 0.0231553. Para este caso se usó una función de regularización para reducir el sobreajuste que presentó el modelo A, cuyas características se encuentran en la etapa de entrenamiento y los resultados en la tabla 18.

El modelo C a diferencia del modelo B también incrementó el número de imágenes del conjunto de datos a 2491, este conjunto de datos también se extrajo desde la plataforma de Kaggle. El error aproximado del modelo C para las métricas de precisión, sensibilidad y F1-Score son: 0,0070755, 0,0388128 y 0,0232019.

- Este modelo obtuvo una precisión de 0.9929245 para este caso obtuvo una diferencia cercana del 0.0039986 a comparación del modelo B. Luego se aprecia el incremento en la sensibilidad y en el F1-Score, obteniendo una diferencia de 0.0220568 y 0.0096339 para ambas métricas. Según indica (Deep Learning: Tres Cosas Que Es Necesario Saber - MATLAB & Simulink, s. f.) “las CNN aprenden

a detectar diferentes características de una imagen mediante decenas o cientos de capas ocultas. Cada capa oculta aumenta la complejidad de las características de la imagen aprendidas”. Por esta razón se nota una pequeña reducción en la precisión.

- Al incrementar el tamaño de datos y el número de capas la extracción de características es más eficiente con respecto al aprendizaje ya que ayuda a reducir la carga computacional.

## Fase 5

### Predicción (Ponderación y Resultados)

Para realizar la predicción el modelo toma una imagen del conjunto de validación que no haya visto en el proceso de entrenamiento para analizar y comprobar si el modelo está cumpliendo la función de clasificación correctamente. Para trabajar con el ambiente de desarrollo de Google Colaboratory, el orden de las carpetas del conjunto de datos está estructurado alfabéticamente. El modelo tendrá la siguiente ponderación:

- Se establece que aquellas predicciones con valores mayores a 0.5 sean clasificadas como Positivas (True).
- Aquellas predicciones con valores menores a 0.5 sean clasificadas como Negativas (False).
  - 0 False = Tiene Covid-19
  - 1 True = Sano

### Tabla 19.

#### *Resultados de las predicciones*

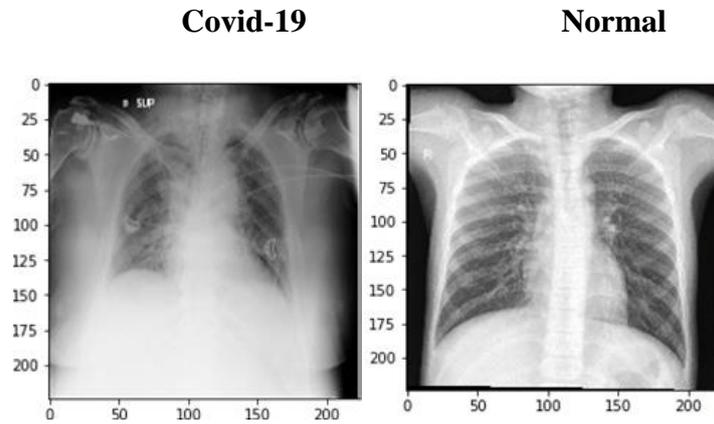
MODELO	PREDICCIONES POR CLASE	
	COVID-19	NORMAL
A	0.36%	0.70%
B	0.34%	0.58%
C	0.05%	0.97%

*Nota.* La elaboración es propia.

El modelo genera la salida de una imagen etiquetada con los dos valores posibles correspondientes a la clasificación: Covid-19 y Normal.

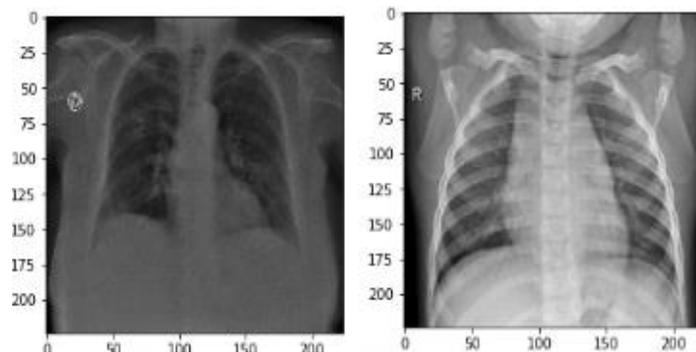
**Figura 41.**

*Imágenes de rayos X de tórax modelo A*



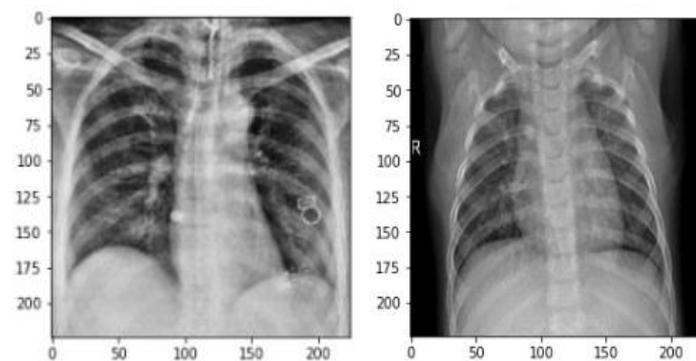
**Figura 42.**

*Imágenes de rayos X de tórax modelo B*



**Figura 43.**

*Imágenes de rayos X de tórax modelo C*



## **Beneficiarios Directos e Indirectos del Proyecto**

### **Beneficiarios Directos**

- Docente Investigador
- Estudiantes que desarrollan este trabajo de titulación

### **Beneficiarios Indirectos**

- Universidad de Guayaquil
- Estudiantes que quieran elaborar modelos de predicción
- Docentes interesados en el proyecto

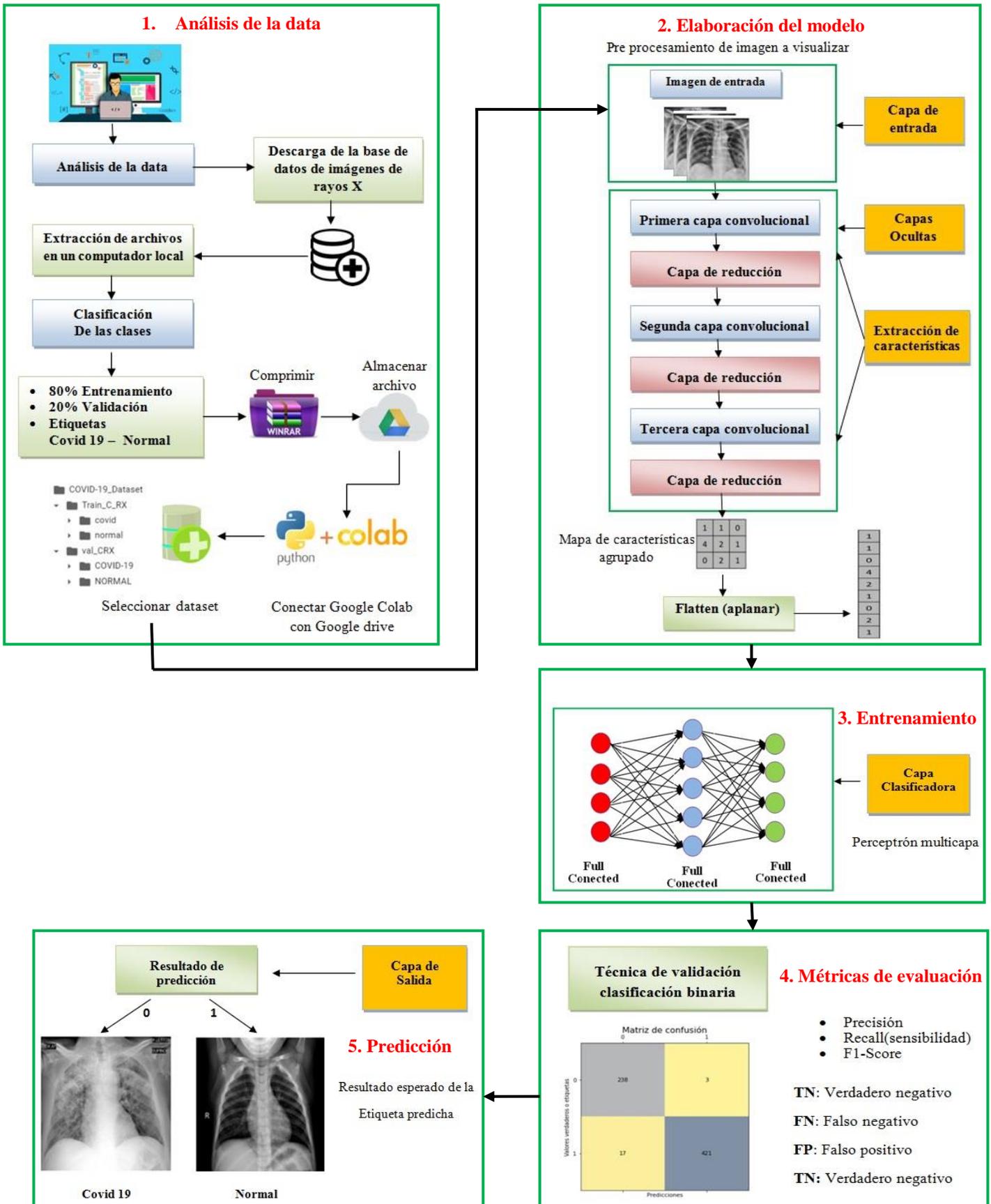
## **Entregables del Proyecto**

- Trabajo de titulación
- Artículo científico
- Código fuente del modelo predicción
- Base de datos de imágenes rayos X

## **Propuesta**

Se ha elaborado el modelo de machine learning para la predicción de Covid-19 en imágenes de rayos X de tórax a través de redes neuronales convolucionales, la misma que podrá ser usada como herramienta tecnológica para ayudar a los médicos en la detección del virus.

## Presentación Esquemática del Modelo



## Criterios de Validación de la Propuesta

### Análisis de datos

Para realizar la validación del proyecto se utilizó el instrumento de juicio de expertos, con la finalidad de efectuar las pruebas de funcionalidad y porcentaje de validación del proyecto elaborado, además los expertos que realicen la validación correspondiente puedan ofrecer valorización para este proyecto y que las técnicas implementadas sean las adecuadas.

En la sección de anexos se encuentra la ficha de juicios de expertos. Se puede observar en el Anexo 4.

### Tabla 20.

#### *Criterios de validación Expertos*

<b>Expertos</b>	<b>Cargos</b>
Mendieta Ojeda Javier	Doctor en medicina general
Arrese Vilche Alfredo	Ingeniero En Sistemas Computacionales
Vélez Salvatierra Heymy	Ingeniero En Sistemas Computacionales
Castañeda Auquilla Ronald	Ingeniero En Sistemas Computacionales
Manzaba Lindao Bryan	Ingeniero En Sistemas Computacionales

*Nota.* La elaboración es propia

### Resultados

Se concluye y se ha validado, con base al estudio y a las pruebas experimentales realizadas de los modelos propuestos, que el modelo C a comparación de los demás ha obtenido los mejores resultados. En consecuencia, se logró una precisión de 0.9929, sensibilidad de 0.9611 y F1 0.9767, lo cual es un resultado muy bueno para lograr la clasificación y detección de patrones mediante imágenes de rayos x con éxito.

## CAPÍTULO IV

### CONCLUSIONES Y RECOMENDACIONES

Una vez finalizado el proyecto de Titulación: “Análisis de imágenes de rayos x de covid-19 a través de redes neuronales artificiales”, se concluye lo siguiente:

#### Conclusiones

- Se ha elaborado el marco teórico en donde se refleja el estudio más apropiado para usar redes neuronales convolucionales. Se concluye que la CNN extraen las características de las imágenes, ya que al ser un modelo de machine learning se alimenta de la data, aprende de ella en el entrenamiento y utiliza los datos aprendidos para identificar o encontrar patrones dentro de un objeto y así obtener un resultado.
- Se concluye que el uso de un entorno de desarrollo en la nube provee los recursos y la paquetería suficiente para elaborar modelos de machine learning cuando no se dispone de un computador con GPU y memoria RAM para almacenar y ejecutar instrucciones.
- Se obtuvo la base de datos del repositorio de GitHub. El conjunto de datos del modelo A, fue captado por Christian Tutivén Gálvez PhD en Automática, robótica y visión de la Universidad Politécnica de Cataluña, España. Quien formó su dataset con referencia a las imágenes de Covid 19 proporcionadas por el Dr. Joseph Cohen, becario posdoctoral de la Universidad de Montreal.
- Se obtuvo la base de datos del repositorio de Kaggle. El conjunto de datos del modelo B, fue captado por (Cohen et al., 2020) y el Comité de Ética de la Universidad de Montreal.
- Para el modelo C se obtuvo el conjunto de datos de un grupo de investigadores de la Universidad de Qatar, Doha, Qatar, y la Universidad de Dhaka, Bangladesh,

junto con sus colaboradores de Pakistán y Malasia en colaboración con médicos, han creado una base de datos de imágenes de rayos X de tórax para casos positivos de Covid-19, junto con imágenes de neumonía normal y viral. Se descargó y analizó tres conjuntos de datos, para realizar la clasificación de los datos se destinó el 80% para entrenamiento y 20% para validación. Concluimos que el modelo mientras más imágenes tenga para procesar, mejor va ser el ciclo de aprendizaje para realizar una correcta clasificación.

- En el modelo A se obtuvo un valor de precisión de 0.9516 y un valor de sensibilidad de 0.9365, utilizando un conjunto de datos con un tamaño de 233 imágenes. Además, el modelo estuvo configurado por 80 neuronas de la capa de entrada, 3 capas de ocultas y 5 números de épocas.
- En el modelo B se obtuvo un valor de precisión de 0.9969 y un valor de sensibilidad de 0.9391, utilizando un conjunto de datos con un tamaño de 1402 imágenes. Además, el modelo estuvo configurado por 160 neuronas de la capa de entrada, 6 capas de ocultas y 10 números de épocas.
- En el modelo C se obtuvo un valor de precisión de 0.9929 y un valor de sensibilidad de 0.9611, utilizando un conjunto de datos con un tamaño de 2491 imágenes. Además, el modelo estuvo configurado por 320 neuronas de la capa de entrada, 9 capas de ocultas y 15 números de épocas.
- De los tres modelos experimentados, el modelo con mejores resultados fue el modelo C, seguido del modelo B y finalmente el modelo A. Los resultados obtenidos tienen una relación directa con el número de imágenes ingresadas al modelo, es decir que, a mayor conjunto de datos, el modelo generalizara de una manera más eficiente.

- Dado los resultados de esta investigación se presenta un artículo científico en el cual se analizan cada uno de los aspectos implementados de un modelo aplicado al Análisis de imágenes de rayos X de Covid-19 a través de redes neuronales artificiales.

### **Recomendaciones**

- Se recomienda ampliar el tamaño del dataset para que se eleve el nivel de predicción del modelo generado.
- Se recomienda usar el modelo desarrollado para futuras predicciones de otro tipo de enfermedades virales.
- El conjunto de imágenes de rayos X de tórax, deben ser claras para cuando entren en la etapa de entrenamiento no presenten inconsistencias.
- Si no se cuenta con un computador de características de RAM y GPU para realizar cálculos complejos y análisis de datos de redes neuronales convolucionales, es recomendable utilizar el entorno de desarrollo virtual de Google Colaboraty, que cuenta con 12 GB de RAM y 50 GB de almacenamiento.

### **Trabajos Futuros**

- Utilizar redes bayesianas para determinar el porcentaje de afectación pulmonar usando el dataset empleado en este trabajo.
- Implementar una aplicación móvil para predecir Covid-19 a través de imágenes de rayos X usando google FireBase TensorFlow Lite y Android estudio.

## REFERENCIAS BIBLIOGRÁFICAS

- ¿Qué es el Deep Learning? | SmartPanel. (n.d.). Retrieved March 10, 2021, from <https://www.smartpanel.com/que-es-deep-learning/>
- Antona, C. (2017). *Herramientas modernas en redes neuronales: la librería keras [Tesis de grado, Universidad Autónoma de Madrid]*. Repositorio Institucional .
- Arias, F. (2012). (PDF) *El proyecto de investigación, 6ta Edición Fidias G. Arias FREELIBROS.ORG | john hamilton - Academia.edu* (EDITORIAL, Vol. 6).
- Avilés Pincay, A. I., & Barcia Salavarría, J. R. (2016). *Elaborar Un Clasificador Para El Procesamiento Digital*. 242.
- Bagnato, J. (2018, November 29). *Convolutional Neural Networks: La Teoría explicada en Español | Aprende Machine Learning*. <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>
- Balagueró, T. (2018). *¿Qué son los datasets y los dataframes en el Big Data?* Deusto Formación.
- Bre, F., Gimenez, J. M., & Fachinotti, V. D. (2018). Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158, 1429–1441. <https://doi.org/10.1016/j.enbuild.2017.11.045>
- Calvo, D. (2017a). *Clasificación de redes neuronales artificiales - Diego Calvo*.
- Calvo, D. (2017b, July 20). *Red Neuronal Convolutiva CNN - Diego Calvo*. <https://www.diegocalvo.es/red-neuronal-convolutiva/>
- Centro para el Control y la Prevención de enfermedades. (2020). *Prueba de detección de infecciones anteriores | CDC*. CDC.
- Chimp, C. (2017, September 27). *El CEO de Nvidia dice que los GPUs reemplazarán la CPU y la ley de Moore está muerta*. <https://www.masgamers.com/nvidia-ley-de-moore-gpu>

- Chowdhury, M. E., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. Bin, Islam, K. R., Khan, M. S., Iqbal, A., Al-Emadi, N., & Reaz, M. B. I. (2020). *¿Puede la IA ayudar a detectar la neumonía viral y COVID-19?* Vol. 8. <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>
- Coello Cristina, & Nuñez Byron. (2020). *Pruebas de antígeno, la nueva herramienta en la detección de la COVID-19.*
- Cohen, J., Morrison, P., Dao, L., Kermany, D., Zhang, K., Goldbaum, M., & Chung, A. (2020). *Chest X-ray (Covid-19 & Pneumonia) | Kaggle.* <https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia>
- Cornieles, P. (2019, February 6). *Entendiendo las redes neuronales: De la neurona a RNN, CNN y Deep Learning - IA Latam.* <https://ia-latam.com/2019/02/06/entendiendo-las-redes-neuronales-de-la-neurona-a-rnn-cnn-y-deep-learning/>
- Covantec. (2018). 3.11. *Tipo diccionarios — Materiales del entrenamiento de programación en Python - Nivel básico.*
- De la fuente, Ó. (2019). Python y machine learning en la nube. *Google Colab: Python y Machine Learning En La Nube.*
- Deep Learning: Tres cosas que es necesario saber - MATLAB & Simulink.* (n.d.). Retrieved February 27, 2021, from <https://es.mathworks.com/discovery/deep-learning.html>
- Delgado, K., Ledesma, S., & Rostro, H. (2019). Análisis de electroencefalograma usando redes neuronales artificiales. *Acta Universitaria*, 29, 1–24. <https://doi.org/10.15174/au.2019.1672>
- Enfermedad del coronavirus 2019 (COVID-19) - Síntomas y causas - Mayo Clinic.* (2020, November 3). <https://www.mayoclinic.org/es-es/diseases-conditions/coronavirus/symptoms-causes/syc-20479963>
- Exploración por tomografía computarizada (TC) | Cancer.Net.* (2018).

- Funciones de activación de capa.* (n.d.). Retrieved January 31, 2021, from <https://keras.io/api/layers/activations/#layer-activation-functions>
- Grando, R. D., Brentano, V. B., Zanardo, A. P., Hertz, F. T., Júnior, L. C. A., Prietto dos Santos, J. F., Galvão, G. S., Zavascki, A. P., & Gazzana, M. B. (2020). Clinical usefulness of tomographic standards for COVID-19 pneumonia diagnosis: Experience from a Brazilian reference center. *Brazilian Journal of Infectious Diseases*, 24(6), 524–533. <https://doi.org/10.1016/j.bjid.2020.10.002>
- Guevara, G., Verdesoto, A., & Castro, N. (2020). *Vista de Metodologías de investigación educativa (descriptivas, experimentales, participativas, y de investigación-acción).*
- Hale, J. (2018). Deep Learning Framework Power Scores 2018. *Towards Data Science.*
- IBM. (n.d.). *Artificial intelligence in medicine.* Retrieved March 23, 2021, from <https://www.ibm.com/watson-health/learn/artificial-intelligence-medicine>
- Información sobre los distintos tipos de pruebas de COVID-19 | SCDHEC.* (2019).
- ISGLOBAL. (2020). *COVID-19: Principales retos y necesidades del ámbito sanitario ante la pandemia - Multimedia - ISGLOBAL.*
- Jackson, B. (2019). *JPG vs JPEG: Entendiendo el formato de archivo de imagen más común.*
- Jáuregui, P., & Herrera, F. (2020). *Inteligencia Artificial para diagnosticar COVID-19 en radiografías del tórax - FBBVA.*
- Keras. (n.d.). *Why choose Keras? Keras Documentation Page.*
- Kulshrestha, S. (2020, April). *What Is Backpropagation? | Training A Neural Network | Edureka.*
- Lazzeri, F. (2020, January 14). *Aprendizaje profundo frente a aprendizaje automático - Azure Machine Learning | Microsoft Docs.* <https://docs.microsoft.com/es-es/azure/machine-learning/concept-deep-learning-vs-machine-learning>
- Lopez, B. (2020). *¿Cuáles son los formatos más importantes de fotos e imágenes digitales?*

- López, J. (2021). *Qué es una GPU: características y especificaciones*.
- Lopez, R. (2016, August 2). *Redes neuronales convolucionales con TensorFlow*.  
<https://relopezbriega.github.io/blog/2016/08/02/redes-neuronales-convolucionales-con-tensorflow/>
- Mapa COVID-19 - Centro de recursos sobre el coronavirus de Johns Hopkins*. (n.d.).  
 Retrieved March 5, 2021, from <https://coronavirus.jhu.edu/map.html>
- Marcos, L. (2020, May 5). *Un algoritmo permite diferenciar al instante a pacientes con COVID-19*. <https://www.muyinteresante.es/tecnologia/articulo/un-algoritmo-permite-diferenciar-al-instante-a-pacientes-con-covid-19-671588677059>
- Marker, G. (2021, January 18). *¿Qué es el Bit? ¿Qué es el Byte? - Tecnología + Informática*.  
<https://www.tecnologia-informatica.com/que-es-el-bit-byte/>
- Martinez, J. (2020, October 9). *Precision, Recall, F1, Accuracy en clasificación - IArtificial.net*. <https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/>
- Meza, J., Estrada, A., Chabusa, C., & Velaszco, Vi. (2020). *Vista de Utilidad de Pruebas de cadena de polimerasa, pruebas rápidas y Tomografías en pacientes con Covid-19 | Journal of America health* (p. 8).
- Moreno, G. (2016, January 21). *Las redes neuronales: qué son y por qué están volviendo*.  
<https://www.xataka.com/robotica-e-ia/las-redes-neuronales-que-son-y-por-que-estan-volviendo>
- Nvidia. (n.d.). *¿GPU vs. CPU? ¿Qué es la computación por GPU? | NVIDIA*.
- OmniSci. (n.d.). *CPU vs GPU | Definition and FAQs | OmniSci*.
- OMS. (2020). *Enfermedad por el Coronavirus (COVID-19) - OPS/OMS | Organización Panamericana de la Salud*. Oms.
- Orozco, C. I., Xamena, E., Mart, C. A., & Rodr, D. A. (2020). *COVID – XR: A Web Management Platform for Coronavirus Detection on X-ray Chest Images. IEEE Latin*

*America Transactions*, 100(1e), 1–8.

Palacios Cruz, M., Santos, E., Velázquez Cervantes, M. A., & León Juárez, M. (2020).

COVID-19, a worldwide public health emergency. In *Revista Clinica Espanola*. Elsevier Doyma. <https://doi.org/10.1016/j.rce.2020.03.001>

*Performing Convolution Operations*. (n.d.). Retrieved September 24, 2020, from

<https://developer.apple.com/library/archive/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html>

Piloto Rodríguez, R. (2017). *Redes Neuronales Artificiales. Conceptos básicos y algunas aplicaciones en Energía*.

Press, E. (2020). *La tomografía de tórax es la mejor prueba para diagnosticar el coronavirus, según estudio*. 1–3.

*Qué es Machine Learning, cómo funciona y a qué se aplica | APD*. (2019, March 4).

<https://www.apd.es/que-es-machine-learning/>

*Que es un Píxel (Px)*. (2020, June 16). <http://chiptecnologico.com/que-es-un-pixel/>

*Qué son regresión y clasificación en Machine Learning*. (2020, October 29).

<https://agenciab12.com/noticia/que-son-regresion-clasificacion-machine-learning>

Quezada, N. (2020). *FACULTAD DE INGENIERÍA CIVIL CARRERA DE INGENIERÍA DE SISTEMAS MACHALA 2020*.

*Radiografía de tórax (radiografía)*. (2020, June 15).

<https://www.radiologyinfo.org/en/info.cfm?pg=chestrax#overview>

Raffino, M. (2020, September 22). *Probabilidad*. <https://concepto.de/probabilidad/>

Rahman, T., Khandakar, A., Qiblawey, Y., Tahir, A., Kiranyaz, S., Abul Kashem, S. Bin, Islam, M. T., Al Maadeed, S., Zughair, S. M., Khan, M. S., & Chowdhury, M. E. H.

(2020). *COVID-19 Radiography Database*.

<https://www.kaggle.com/tawsifurrahman/covid19-radiography-database>

- Rayos X: MedlinePlus en español.* (n.d.). Retrieved November 20, 2020, from <https://medlineplus.gov/spanish/xrays.html>
- Rayos X.* (n.d.). Retrieved November 20, 2020, from <https://www.nibib.nih.gov/espanol/temas-cientificos/rayos-x>
- Recuero de los Santos, P. (2018, January 23). *Machine Learning a tu alcance: La matriz de confusión - Think Big Empresas.* <https://empresas.blogthinkbig.com/ml-a-tu-alcance-matriz-confusion/>
- Recuero de los Santos, P. (2019, January 15). *Deep Learning para todos los públicos: ¿Qué son los tensores? ¿Qué es TensorFlow? - Think Big Empresas.* <https://empresas.blogthinkbig.com/deep-learning-para-todos-los-publicos/>
- Recuero de los Santos, P. (2020, September 9). *Cómo interpretar la matriz de confusión: ejemplo práctico - Think Big Empresas.* <https://empresas.blogthinkbig.com/como-interpretar-la-matriz-de-confusion-ejemplo-practico/>
- Redes neuronales convolucionales CS231n para reconocimiento visual.* (n.d.). Retrieved January 10, 2021, from <https://cs231n.github.io/convolutional-networks/#overview>
- Robinson, D. (2017). The incredible growth of Python. *Stack Overflow Blog.*
- Robledano, Á. (2019, June 18). *Qué es un algoritmo informático | OpenWebinars.* <https://openwebinars.net/blog/que-es-un-algoritmo-informatico/>
- Rocha, C. (2020). *¿Cómo se puede Diagnosticar el COVID-19? ¿Cuál es la Precisión de los Test Diagnósticos?* In *Sociedad Andaluza de Oftalmología.*
- Rodríguez, D. (2019, December 16). *¿Cuál es la diferencia entre parámetro e hiperparámetro? - Analytics Lane.* <https://www.analyticslane.com/2019/12/16/cual-es-la-diferencia-entre-parametro-e-hiperparametro/>
- Rodríguez, V. (2018, November 9). *Dropout y Batch Normalization.* <https://vincentblog.xyz/posts/dropout-y-batch-normalization>

- Romero, B. (2020). *Tipos de formatos de imagen*.  
*sample\_grid\_a\_square.png* (437×256). (n.d.). Retrieved January 8, 2021, from  
[http://pippin.gimp.org/image\\_processing/images/sample\\_grid\\_a\\_square.png](http://pippin.gimp.org/image_processing/images/sample_grid_a_square.png)
- Sancho, F. (2019, December 14). *Redes Neuronales: una visión superficial*.  
<http://www.cs.us.es/~fsancho/?e=72>
- Sancho, F. (2020, December). *Entrenamiento de Redes Neuronales: mejorando el Gradiente Descendente - Fernando Sancho Caparrini*.
- Sanseviero, O. (2019, March 11). *Introducción a TensorFlow (Parte 1) | by Omar Sanseviero | AI Learners | Medium*. <https://medium.com/ai-learners/introducción-a-tensorflow-parte-1-840c01881658>
- Schmitt, M. (n.d.). *Artificial Intelligence in Medicine*. Retrieved March 23, 2021, from  
<https://www.datarevenue.com/en-blog/artificial-intelligence-in-medicine>
- Seguridad del paciente: dosis de radiación en exámenes de rayos X y TC*. (n.d.). Retrieved November 19, 2020, from <https://www.radiologyinfo.org/en/info.cfm?pg=safety-xray>
- Sharma, A. (2017, December 5). *Redes neuronales convolucionales en Python - DataCamp*.  
<https://www.datacamp.com/community/tutorials/convolutional-neural-networks-python>
- Singh Chauhan, N. (2020, September 2). *Métricas De Evaluación De Modelos En El Aprendizaje Automático*. <https://www.datasource.ai/es/data-science-articles/metricas-de-evaluacion-de-modelos-en-el-aprendizaje-automatico>
- Solé, R. (2020). *Qué es el RGB y qué son los LED RGB y sus usos*.
- Soloaga, A. (2018). *Python, los 5 usos más importantes de este lenguaje de programación*. Akademos.
- Spinelli, A., & Pellino, G. (2020). COVID-19 pandemic: perspectives on an unfolding crisis. *British Journal of Surgery*, 107(7), 785–787. <https://doi.org/10.1002/bjs.11627>
- Tablado, F. (2020, June 18). *Inteligencia artificial: Definición, tipos y aplicaciones | Grupo*

- Atico34. <https://protecciondatos-lopd.com/empresas/inteligencia-artificial/>
- TensorFlow. (n.d.). Retrieved February 26, 2021, from <https://www.tensorflow.org/>
- Tipos del algoritmo de aprendizaje automático — DATA SCIENCE. (2020, May). <https://datascience.eu/es/aprendizaje-automatico/los-algoritmos-de-aprendizaje-automatico-mas-populares/>
- Tutivén, C. (2020, March 25). *Detector de COVID-19 usando imágenes de Rayos X / Saturdays.AI*. <https://medium.com/saturdays-ai/detector-de-covid-19-usando-imágenes-de-rayos-x-eaa146efa26>
- Universidad de Alcalá. (2020). *R y Python en Data Science - Máster en Data Science*. Master Data Science.
- Universidad Internacional de Valencia. (2019). *Python para big data: motivos para elegirlo / VIU. CIENCIA Y TECNOLOGÍA*.
- Uriarte, J. (2020, November). *Google Drive: qué es, cómo funciona y características*.
- Valle, F. (2016). *Sistema electrónico de alerta temprana de incendios forestales mediante procesamiento digital de imágenes para la Empresa Municipal del Cuerpo de Bomberos Ambato*. 36–42.
- Velazquez, M. (2020, April 17). *Diferencias entre el análisis predictivo e IA*. <https://recluit.com/diferencias-entre-el-analisis-predictivo-e-ia/#.YErabmhKjIV>
- Wang, C., Horby, P. W., Hayden, F. G., & Gao, G. F. (2020). A novel coronavirus outbreak of global health concern. In *The Lancet* (Vol. 395, Issue 10223, pp. 470–473). Lancet Publishing Group. [https://doi.org/10.1016/S0140-6736\(20\)30185-9](https://doi.org/10.1016/S0140-6736(20)30185-9)
- West, D., & Allen, J. (2018, April 24). *How artificial intelligence is transforming the world*. <https://www.brookings.edu/research/how-artificial-intelligence-is-transforming-the-world/>
- Xu, X., Jiang, X., Ma, C., Du, P., Li, X., Lv, S., Yu, L., Ni, Q., Chen, Y., Su, J., Lang, G., Li,

- Y., Zhao, H., Liu, J., Xu, K., Ruan, L., Sheng, J., Qiu, Y., Wu, W., ... Li, L. (2020). A Deep Learning System to Screen Novel Coronavirus Disease 2019 Pneumonia. *Engineering*, 6(10), 1122–1129. <https://doi.org/10.1016/j.eng.2020.04.010>
- Yadav, R., Sahoo, D., & Graham, R. (2020). Thoracic imaging in COVID-19. *Cleveland Clinic Journal of Medicine*, 87(8), 469–476. <https://doi.org/10.3949/ccjm.87a.ccc032>
- Zaforas, M. (2017). ¿Es Python el lenguaje del futuro? - Paradigma. In *ParadigmaDigital*.



## Anexo 2. Geo-Localización Del Problema



Tomado de (*Mapa COVID-19 - Centro de Recursos Sobre El Coronavirus de Johns Hopkins, s. f.*).

### Anexo 3. Fundamentación Legal

Para el desarrollo y el análisis de nuestro proyecto investigativo nos basaremos en los siguientes artículos:

#### LEY DE PROPIEDAD INTELECTUAL

##### TITULO PRELIMINAR

##### Capítulo 1 (Derecho de autor)

<b>Art. 1.</b>	El Estado reconoce, regula y garantiza la propiedad intelectual adquirida de conformidad con la ley, las decisiones de la Comisión de la Comunidad Andina y los convenios internacionales vigentes en el Ecuador.
----------------	---

##### Sección I - preceptos generales

<b>Art 4.</b>	Se reconocen y garantizan los derechos de los autores y los derechos de los demás titulares sobre sus obras.
<b>Art. 5</b>	El derecho de autor nace y se protege por el solo hecho de la creación de la obra, independientemente de su mérito, destino o modo de expresión.
<b>Art. 6</b>	El derecho de autor es independiente, compatible y acumulable con: <ul style="list-style-type: none"> <li><b>a)</b> La propiedad y otros derechos que tengan por objeto la cosa material a la que esté incorporada la obra.</li> <li><b>b)</b> Los derechos de propiedad industrial que puedan existir sobre la obra; y,</li> <li><b>c)</b> Los otros derechos de propiedad intelectual reconocidos por la ley.</li> </ul>

##### Estatuto Orgánico de la Universidad de Guayaquil, Visión, Misión y Afines

<b>Art. 1.</b>	<b>Fines de la universidad.</b> - La Universidad de Guayaquil, persigue los fines siguientes: Aportar al desarrollo del pensamiento universal, al despliegue de la producción científica y a la promoción de las transferencias e innovaciones tecnológicas;
----------------	--

## LEY ORGANICA DE EDUCACION SUPERIOR, LOES

### TITULO I

#### AMBITO, OBJETO, FINES Y PRINCIPIOS DEL SISTEMA DE EDUCACION SUPERIOR

##### Capítulo 1(Ámbito y objeto)

<b>Art. 1.</b>	<p><b>Ámbito.-</b> Esta Ley regula el sistema de educación superior en el país, a los organismos e instituciones que lo integran; determina derechos, deberes y obligaciones de las personas naturales y jurídicas, y establece las respectivas sanciones por el incumplimiento de las disposiciones Contenidas en la Constitución y la presente Ley.</p>
<b>Art. 2</b>	<p><b>Objeto.-</b> Esta Ley tiene como objeto definir sus principios, garantizar el derecho a la educación superior de calidad que propenda a la excelencia interculturalidad, al acceso universal, permanencia, movilidad y egreso sin discriminación alguna y con gratuidad en el ámbito público hasta el tercer nivel.</p> <p>Nota: Artículo reformado por artículo 2 de Ley No. 0, publicada en Registro Oficial Suplemento 297 de 2 de Agosto del 2018.</p>

##### Capítulo 2(Fines de la educación superior)

<b>Art. 3</b>	<p><b>Fines de la Educación Superior.</b> - La educación superior de carácter humanista, intercultural y científica constituye un derecho de las personas y un bien público social que, de conformidad con la constitución de la República, responderá al interés público y no estará al servicio de intereses individuales y corporativos.</p> <p><b>Nota:</b> Artículo reformado por artículo 3 de Ley No. 0, publicada en Registro Oficial Suplemento 297 de 2 de Agosto del 2018.</p>
<b>Art. 4</b>	<p><b>Derecho a la Educación Superior.</b> - El derecho a la educación superior consiste en el ejercicio efectivo de la igualdad de oportunidades, en función de los méritos respectivos, a fin de acceder a una formación académica y profesional con producción de conocimiento pertinente y de excelencia.</p> <p>Las ciudadanas y los ciudadanos en forma individual y colectiva, las comunidades, pueblos y nacionalidades tienen el derecho y la responsabilidad de participar en el proceso educativo superior, a través de los mecanismos establecidos en la constitución y esta Ley.</p>

**Capítulo 3**  
**Principios del sistema de educación superior**

<b>Art. 13.</b>	<p><b>Funciones del Sistema de Educación Superior.</b> - Son funciones del Sistema de Educación Superior:</p> <p><b>a)</b> Garantizar el derecho a la educación superior mediante la docencia, la investigación y su vinculación con la sociedad, y asegurar crecientes niveles de calidad, excelencia académica y pertinencia.</p> <p><b>b)</b> Promover la creación, desarrollo, transmisión y difusión de la ciencia, la técnica, la tecnología y la cultura.</p> <p><b>c)</b> Formar académicos, científicos y profesionales responsables, éticos y solidarios, comprometidos con la sociedad, debidamente preparados en todos los campos del conocimiento, para que sean capaces de generar y aplicar sus conocimientos y métodos científicos, así como la creación y promoción cultural y artística.</p> <p><b>Nota:</b> Artículo sustituido por artículo 10 de Ley No. 0, publicada en Registro Oficial Suplemento 297 de 2 de Agosto del 2018.</p>
-----------------	--



**CONSTANCIA DE JUICIO DE EXPERTO**

Estimado Ingeniero

Darwin Guillermo Patiño Pérez

**DOCENTE TUTOR DEL TRABAJO DE TITULACIÓN**

Guayaquil.

El presente instrumento certifica que se realizó la revisión del proyecto de titulación **“ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES NEURONALES ARTIFICIALES”** cuyos criterios e indicadores empleados permitieron articular el trabajo según se muestra en el Anexo 4, por tanto, **Danny Pulig Cesén y Kevin Morocho Sande** estudiantes no titulados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, (SI) pueden continuar con el proceso de titulación en vista que (no) existen observaciones.

Por lo actuado en el Anexo 4, se procede a validar el trabajo de titulación.

Sin otro particular.



---

**DR. JAVIER MENDIETA OJEDA**

**C.I. N° 0915471957**

## DATOS GENERALES

APELLIDOS Y NOMBRES DEL EXPERTO		TÍTULO PROFESIONAL DEL EXPERTO		AUTORES																	
ARRESE VILCHE ALFREDO ENRIQUE		INGENIERO EN SISTEMAS COMPUTACIONALES		DANNY PULIG CESÉN								KEVIN MOROCHO SANDE									
TÍTULO DEL PROYECTO		ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES NEURONALES ARTIFICIALES																			
INDICADOR	CRITERIO	DEFICIENTE 0-20				REGULAR 21-40				BUENA 41- 60				MUY BUENA 61- 80				EXCELENTE 81 - 100			
		5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
CLARIDAD	Se utiliza el lenguaje de programación apropiado que facilita la comprensión.																				X
OBJETIVIDAD	Está expresado en conductas observables y medibles.																				X
ACTUALIDAD	Esta acorde a los aportes recientes en la disciplina de estudio.																				X
SUFICIENCIA	Son suficientes la cantidad y calidad de ítems presentados en el instrumento.																				X
INTENCIONALIDAD	Es adecuado para valorar la variable seleccionada.																				X
CONSISTENCIA	Está basado en aspectos teóricos y científicos.																				X
METODOLOGÍA	El instrumento se relaciona con el método planteado en el proyecto.																			X	
APLICABILIDAD	El instrumento es de fácil aplicación.																		X		

**CONSTANCIA DE JUICIO DE EXPERTO**

Estimado Ingeniero

Darwin Guillermo Patiño Pérez

**DOCENTE TUTOR DEL TRABAJO DE TITULACIÓN**

Guayaquil.

El presente instrumento certifica que se realizó la revisión del proyecto de titulación **“ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES NEURONALES ARTIFICIALES”** cuyos criterios e indicadores empleados permitieron articular el trabajo según se muestra en el Anexo 4, por tanto, **Danny Pulig Cesén y Kevin Morocho Sande** estudiantes no titulados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, **(SI)** pueden continuar con el proceso de titulación en vista que (no) existen observaciones.

Por lo actuado en el Anexo 4, se procede a validar el trabajo de titulación.

Sin otro particular.



---

**ARRESE VILCHE ALFREDO ENRIQUE**  
**C.I. N° 0919130096**



**CONSTANCIA DE JUICIO DE EXPERTO**

Estimado Ingeniero

Darwin Guillermo Patiño Pérez

**DOCENTE TUTOR DEL TRABAJO DE TITULACIÓN**

Guayaquil.

El presente instrumento certifica que se realizó la revisión del proyecto de titulación “ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES NEURONALES ARTIFICIALES” cuyos criterios e indicadores empleados permitieron articular el trabajo según se muestra en el Anexo 4, por tanto, **Danny Pulig Cesén** y **Kevin Morocho Sande** estudiantes no titulados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, (SI) pueden continuar con el proceso de titulación en vista que (no ) existen observaciones.

Por lo actuado en el Anexo 7, se procede a validar el trabajo de titulación.

Sin otro particular.

---

**VELEZ SALVATIERRA HEYMY JANNYNA**  
**C.I. N° 0926437708**

## DATOS GENERALES

APELLIDOS Y NOMBRES DEL EXPERTO		TITULO PROFESIONAL DEL EXPERTO		AUTORES																		
CASTAÑEDA AUQUILLA RONALD ANTONIO		INGENIERO EN SISTEMAS COMPUTACIONALES		DANNY PULIG CESÉN								KEVIN MOROCHO SANDE										
TÍTULO DEL PROYECTO		ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES NEURONALES ARTIFICIALES																				
INDICADOR	CRITERIO	DEFICIENTE 0-20				REGULAR 21-40				BUENA 41- 60				MUY BUENA 61- 80				EXCELENTE 81 - 100				
		5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	
CLARIDAD	Se utiliza el lenguaje de programación apropiado que facilita la comprensión.																				X	
OBJETIVIDAD	Está expresado en conductas observables y medibles.																					X
ACTUALIDAD	Esta acorde a los aportes recientes en la disciplina de estudio.																				X	
SUFICIENCIA	Son suficientes la cantidad y calidad de ítems presentados en el instrumento.																				X	
INTENCIONALIDAD	Es adecuado para valorar la variable seleccionada.																				X	
CONSISTENCIA	Está basado en aspectos teóricos y científicos.																				X	
METODOLOGÍA	El instrumento se relaciona con el método planteado en el proyecto.																		X			
APLICABILIDAD	El instrumento es de fácil aplicación.																			X		

**CONSTANCIA DE JUICIO DE EXPERTO**

Estimado Ingeniero

Darwin Guillermo Patiño Pérez

**DOCENTE TUTOR DEL TRABAJO DE TITULACIÓN**

Guayaquil.

El presente instrumento certifica que se realizó la revisión del proyecto de titulación “ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES NEURONALES ARTIFICIALES” cuyos criterios e indicadores empleados permitieron articular el trabajo según se muestra en el Anexo 4, por tanto, **Danny Pulig Cesén** y **Kevin Morocho Sande** estudiantes no titulados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, (SI) pueden continuar con el proceso de titulación en vista que (no) existen observaciones.

Por lo actuado en el Anexo 7, se procede a validar el trabajo de titulación.

Sin otro particular.



---

**CASTAÑEDA AUQUILLA RONALD ANTONIO**  
**C.I. N° 0927114074**



**CONSTANCIA DE JUICIO DE EXPERTO**

Estimado Ingeniero

Darwin Guillermo Patiño Pérez

**DOCENTE TUTOR DEL TRABAJO DE TITULACIÓN**

Guayaquil.

El presente instrumento certifica que se realizó la revisión del proyecto de titulación “ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES NEURONALES ARTIFICIALES” cuyos criterios e indicadores empleados permitieron articular el trabajo según se muestra en el Anexo 4, por tanto, **Danny Pulig Cesén** y **Kevin Morocho Sande** estudiantes no titulados de la Carrera de Ingeniería en Sistemas computacionales de la Universidad de Guayaquil, (SI) pueden continuar con el proceso de titulación en vista que (no) existen observaciones.

Por lo actuado en el Anexo 7, se procede a validar el trabajo de titulación.

Sin otro particular.



---

**MANZABA LINDAO BRYAN CRISTOPHER**  
C.I. N° 0929371722

## Anexo 5. Manual técnico

### Se crea la conexión con el Drive para obtener la base de datos

```
from google.colab import drive
drive.mount('/content/drive')
```

### Se descomprimen las carpetas del dataset

```
from zipfile import ZipFile

datasets = ['Train_C_RX', 'val_CRX']
for dataset in datasets:
    with ZipFile('/content/drive/MyDrive/Base_datos_Modelos_covid19/base_
de_datos_RX_PM_1/base_datos_partida_train_val/{}.zip'.format(dataset
), 'r') as zip: #'/content/drive/My Drive/COVID-19_Dataset/Train.zip'
        zip.extractall('COVID-19_Dataset')
```

### Referenciamos los conjuntos de entrenamiento y validación, los cuales se encuentran en el dataset

```
import os
train_dir = os.path.join('COVID-19_Dataset/Train_C_RX')
validation_dir = os.path.join('COVID-19_Dataset/val_CRX')
```

### Se observa una imagen seleccionada a través de la ruta guardada en la variable p

```
#Ruta de alguna imagen que pertenezca al conjunto de entrenamiento
p='/content/COVID-19_Dataset/Train_C_RX/normal/NORMAL (118).png'
p

#Ruta de alguna imagen que pertenezca al conjunto de entrenamiento
img_path = p

# Pre procesamos la imagen en un tensor 4D
from keras.preprocessing import image
import numpy as np

img = image.load_img(img_path, target_size=(224, 224))
img_tensor = image.img_to_array(img)
img_tensor = np.expand_dims(img_tensor, axis=0)

#Recordatorio el modelo será entrenado con imágenes
#de entrada que se pre procesaron de la siguiente manera:
img_tensor /= 255.

# Su dimensión es de (1, 224, 224, 3)
print(img_tensor.shape)
```

**Con la siguiente línea de código visualizamos la imagen del pulmón seleccionada.**

```
import matplotlib.pyplot as plt

plt.imshow(img_tensor[0])
plt.show()
```

### Arquitectura del modelo, red neuronal convolucional

```
from tensorflow.keras import layers
from tensorflow.keras import models

#layers.Conv2D -> Capa convolucional
#layers.MaxPooling2D -> Capa max pooling
#320 -> Volumen de filtros que aplicaremos tras la primera convolucion
#kernel_size -> tamaño del Kernel
model = models.Sequential() #varias capas apiladas entre ellas
model.add(layers.Conv2D(320, kernel_size=(3,3), activation='relu', input_shape=(224,224,3))) #primera capa de redes convolucionales, input_shape(forma de entrada)
model.add(layers.Conv2D(228, (3,3), activation='relu'))
model.add(layers.MaxPooling2D(pool_size=(2,2))) #tamaño de pool
model.add(layers.Dropout(0.25)) #Regularizar el rendimiento de su CNN o mejora el desempeño de su CNN, ayuda a evitar el sobreajuste

model.add(layers.Conv2D(128, (3,3), activation='relu')) #segunda
model.add(layers.MaxPooling2D(pool_size=(2,2)))
model.add(layers.Dropout(0.25))

model.add(layers.Conv2D(128, (3,3), activation='relu')) #tercera
model.add(layers.MaxPooling2D(pool_size=(2,2)))
model.add(layers.Dropout(0.25))

model.add(layers.Flatten()) #los volúmenes de imágenes del último Maxpooling se aplanan, es decir pasan a un vector (Flatten)
model.add(layers.Dense(364, activation='relu')) #primera capa de redes neuronales densamente conectadas
model.add(layers.Dropout(0.5))

model.add(layers.Dense(264, activation='relu')) #segunda
model.add(layers.Dropout(0.5))

model.add(layers.Dense(164, activation='relu')) #tercera
model.add(layers.Dropout(0.5))

model.add(layers.Dense(1, activation='sigmoid')) #capa de salida que da resultado si o no
```

```
model.summary()#funcion, para obtener el resumen o información del modelo creado
```

## Compilamos el modelo

```
from tensorflow.keras import optimizers
#Funcion compile
model.compile(loss='binary_crossentropy', #algoritmo de pérdida
              optimizer='adam', #SGD
              metrics=['acc']) #Eficiencia
```

## Definición de los generadores pre-procesamiento

```
#Using ImageDataGenerator to read imagen from directories/leer imágenes de directorio
from tensorflow.keras.preprocessing.image import ImageDataGenerator

#Preparamos nuestras imágenes
#conjunto de entrenamiento (Normalización)
train_datagen = ImageDataGenerator(rescale=1./255, #Rescalamos los pixeles de la imagen entre 0 y 1 (esto es fundamental para mejorar el entrenamiento)
                                   shear_range = 0.2, #inclinación imágenes, porque las imágenes no siempre vienen derechas
                                   zoom_range = 0.2, #para acercar
                                   horizontal_flip = True)

#conjunto de validación
test_datagen = ImageDataGenerator(rescale=1./255)

#Abre y prepara toda la carpeta de entrenamiento
train_generator = train_datagen.flow_from_directory(
    train_dir, # Directorio de destino
    target_size=(224, 224), #Todas las imágenes cambian de tamaño de 240x320 a 224x224
    batch_size= 32, #va a enviar por cantidades de 32 imágenes para ser procesadas y obtener mejores resultados
    color_mode='rgb',
    class_mode='binary',
    shuffle=True)

#Abre y prepara toda la carpeta de validación
validation_generator = test_datagen.flow_from_directory(
    validation_dir,
    target_size=(224, 224),
    batch_size=1,
    color_mode='rgb',
    class_mode='binary',
    shuffle=False)
```

## Entrenamiento

```
#Training and validation stages
history = model.fit(
    train_generator, #carpeta de entrenamiento
    steps_per_epoch=7,
    epochs= 15,
    validation_data=validation_generator,
    validation_steps=60)
```

## Gráfica de la precisión y pérdida durante el entrenamiento

```
import matplotlib.pyplot as plt

acc = history.history['acc']
val_acc = history.history['val_acc']
max_val_acc = max(val_acc)
max_val_acc_epoch= val_acc.index(max(val_acc)) + 1
loss = history.history['loss']
val_loss = history.history['val_loss']
min_val_loss = min(val_loss)
min_val_loss_epoch= val_loss.index(min(val_loss)) + 1

epochs = range(len(acc))

plt.plot(epochs, acc, 'go', label='Precisión en el entrenamiento')
plt.plot(epochs, val_acc, 'r', label='Precisión en la validación')
plt.plot(max_val_acc_epoch, max_val_acc, 'bo', label='Maximum accuracy'
)
plt.title('Precisión durante el entrenamiento y la validación')
plt.legend()
print("Best accuracy epoch : % d, Value : % .5f" %(max_val_acc_epoch, m
ax_val_acc))
plt.figure()

plt.plot(epochs, loss, 'go', label='Pérdida en el entrenamiento')
plt.plot(epochs, val_loss, 'r', label='Pérdida en la validación')
plt.plot(min_val_loss_epoch, min_val_loss, 'bo', label='Minimum loss')
plt.title('Pérdida durante el entrenamiento y la validación')
plt.legend()
print("Best loss epoch : % d, Value : % .10f" %(min_val_loss_epoch, min
_val_loss))
plt.show()
```

## Guardamos el modelo

```
model.save('/content/drive/MyDrive/Base_datos_Modelos_covid19/algorithmo
_Prediccion_PM_1/modelo_Covid1.h5')
```

## Cargamos el modelo

```
import tensorflow
modelp = tensorflow.keras.models.load_model('/content/drive/MyDrive/Bas
e_datos_Modelos_covid19/algoritmo_Prediccion_PM_1/modelo_Covid1.h5', co
mpile=False)
```

## Matriz de confusión

```
from sklearn.metrics import confusion_matrix

test_predictions = model.predict_generator(validation_generator, 679) #
model.predict_generator(generator, num_lotes_por_epoca)

#Establecemos que aquellas predicciones con valores mayores a 0.5, sean
categorizadas como Positivas (True),
#y que las predicciones con valores menores a 0.5, sean clasificadas co
mo Negativas (False)
test_predictions = (test_predictions > 0.5)
print(test_predictions)

#Etiquetas o valores esperados
validation_generator.classes

from sklearn.metrics import confusion_matrix

cm = confusion_matrix(validation_generator.classes, test_predictions)
print('Confusion Matrix')
print(cm)

import matplotlib.pyplot as plt
import seaborn as sns

fig, ax = plt.subplots(figsize=(5, 5))
ax.matshow(cm, cmap=plt.cm.cividis_r, alpha=0.5)
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        ax.text(x=j, y=i, s=cm[i, j], va='center', ha='center')

plt.title("Matriz de confusión", fontsize=15)
plt.xlabel('Predicciones')
plt.ylabel('Valores verdaderos o etiquetas')

plt.tight_layout()
plt.show()
```

## Calculamos la métrica

```
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score, f1_score

print('Precision: %.7f' % precision_score(y_true=validation_generator.c
lasses, y_pred=test_predictions))
print('Recall: %.7f' % recall_score(y_true=validation_generator.classes
, y_pred=test_predictions))
print('F1: %.7f' % f1_score(y_true=validation_generator.classes, y_pred
=test_predictions))
```

## Se ve cuantas imágenes están en validación y cuantas clases tiene

```
#Test stage
test_generator = test_datagen.flow_from_directory(
    validation_dir,
    target_size=(72, 96),
    batch_size=1,
    color_mode='rgb',
    class_mode='categorical')
```

## Cargamos la imagen y le damos tratamiento para que entre como el modelo espera de 224\*224

```
#Ruta de alguna imagen que pertenezca al conjunto de validación
img_path = '/content/COVID-19 (8).png'

# Pre procesamos la imagen en un tensor 4D
from keras.preprocessing import image
import numpy as np

img = image.load_img(img_path, target_size=(224, 224))
img_tensor = image.img_to_array(img)
img_tensor = np.expand_dims(img_tensor, axis=0)

#Recuerda que el modelo fue entrenado con imágenes
#de entrada que se pre procesaron de la siguiente manera:
img_tensor /= 255.

# Su dimensión es de (1, 224, 224, 3)
print(img_tensor.shape)
```

**Cargamos la imagen al modelo, con dimensiones de 224 \* 224**

```
predic = model.predict_generator(img_tensor)
```

**Se ve cuanto es el valor ya pasado por el modelo**

```
predic
```

**Se especifica que arriba del 0.5 de el resultado True y abajo del 0.5 de false**

- True= Sano
- false= Tiene covid

```
predic = (predic > 0.5)  
print(predic)
```

```
if predic == False:  
    print('Tiene covid, necesita atención médica lo mas pronto posible, p  
or favor')  
    plt.imshow(img_tensor[0])  
    plt.show()  
else:  
    print('Esta sano, no presenta ningún inicio de covid')  
    plt.imshow(img_tensor[0])  
    plt.show()
```

**Anexo 6. Artículo Científico**

# Análisis de imágenes de rayos X de Covid-19 a través de redes neuronales artificiales

## *X-ray imaging analysis of Covid-19 through artificial neuronal networks*



Kevin Morocho Sande – Danny Pulig Cesén.

Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas

### ABSTRACT

- Currently, the most effective tests for the detection of Covid-19 such as computed axial tomography (CT) image analysis and laboratory medical evaluations such as the polymerase chain reaction (PCR) test, are one of the most effective methods for the diagnosis of this disease. The only disadvantage is that these traditional methods are very expensive and that meant that the lower-middle class population did not have access to these screening tests. Artificial neural networks play a very important role in the field of medicine and in technological research that contributes to the detection of diseases such as Covid-19. This project was centralized in order to provide support to doctors in decision-making, using technological tools. A machine learning model was performed for the analysis of X-ray images for the detection of Covid-19 through convolutional neural networks. The algorithm was carried out in the Python programming language with the use of the Google Colaboratory cloud development environment. Through the public access repositories of GitHub and Kaggle, databases of chest X-ray images of patients with Covid-19 and normal (healthy) patients were collected to generate a dataset with training and validation images. Experimental tests were carried out with three types of convolutional neural network architecture for models A, B and C. Model C being the one that obtained the best results.
- **Key Words:** Covid-19, X-ray, machine learning, prediction model, convolutional neural networks.

### RESUMEN

En la actualidad, las pruebas más efectivas para la detección de Covid-19 como el análisis de imágenes por tomografía axial computarizada (TAC) y las evaluaciones médicas de laboratorio como la prueba de reacción en cadena de la polimerasa (PCR), son uno de los métodos más efectivos para el diagnóstico de esta enfermedad. La única desventaja es que estos métodos tradicionales son muy costosos y eso implicaba que la población de clase media-baja no tenga acceso a estas pruebas de detección. Las redes neuronales artificiales juegan un papel muy importante en el campo de la medicina y en investigaciones tecnológicas que contribuyan a la detección de enfermedades como el Covid-19. Este proyecto se centralizó con el fin de proporcionar un apoyo a los médicos en la toma de decisiones, utilizando herramientas tecnológicas. Se realizó un modelo de machine learning para el análisis de imágenes de rayos X para la detección de Covid-19 a través de redes neuronales convolucionales. El algoritmo fue realizado en el lenguaje de programación Python con el uso del entorno de desarrollo en la nube de Google Colaboratory. Por medio de los

repositorios de acceso público de GitHub y Kaggle, se recolectaron bases de datos de imágenes de rayos X de tórax de pacientes con Covid-19 y pacientes normales (sanos) para generar un dataset con imágenes de entrenamiento y validación. Se realizaron pruebas experimentales con tres tipos de arquitectura de redes neuronales convolucionales para los modelos A, B y C. Siendo el modelo C el que obtuvo los mejores resultados

**Palabras Clave:** Covid-19, rayos X, machine learning, modelo de predicción, redes neuronales convolucionales.

### 1. INTRODUCCIÓN

El descubrimiento del nuevo coronavirus 2019 (Covid-19), es una enfermedad con un alto grado de contagio que se propaga de manera muy fácil y rápida en todo el mundo, afecta la salud del ser humano produciendo infecciones en los pulmones (neumonía), problemas para respirar, fatiga y cansancio, este nuevo virus es la principal causa de muertes en pacientes que presentan Covid-19. China fue el primer país en donde se reportó una gran cantidad de muertes provocado por el contagio del virus. Brasil fue el segundo país más afectado reportando tres millones de casos confirmados y más de 100.000 muertes en los primeros días del mes de agosto [1].

En varias industrias de trabajo el uso de una herramienta tecnológica se ha vuelto muy importante para resolver una gran variedad de problemas y brindar soluciones ante enfermedades infecciosas. Específicamente en el campo de la medicina los doctores usan imágenes médicas que son creadas por medio de rayos X, sirven de gran ayuda para poder detectar y tratar varios tipos de enfermedades o dolores que se presentan en el cuerpo humano, tales como, huesos rotos, infecciones en los pulmones, mamografías, ecografías, resonancias magnéticas, entre otras.

El presente trabajo de titulación está enfocado en realizar el análisis de imágenes de rayos X mediante un modelo de machine learning para la predicción de Covid-19 a través de redes neuronales convolucionales, para mejorar el procesamiento de una gran variedad de conjunto de datos y poder obtener resultados más rápidos y precisos. Con la ayuda de Google Colab el cual pertenece a una herramienta de google, permite trabajar con código Python y proporciona entornos de ejecución tales como CPU, GPU y TPU.

Con la ayuda de un modelo de predicción se puede detectar en una radiografía torácica si un paciente se encuentra sano o enfermo por Covid-19, se lo realiza por medio del big data (grandes cantidades de imágenes radiográficas) junto con el Deep learning (es una subcampo del aprendizaje automático), la combinación de ambos métodos permite obtener mejores resultados en la clasificación de imágenes, este sistema de estudio se llevó a cabo por investigadores del CVBLab, de la Universidad Politécnica de Valencia [2].

Mediante el uso de las redes neuronales convolucionales principal algoritmo utilizado en machine learning, son métodos muy

importantes para clasificación de imágenes, detección de objetos, reconocimiento de voz, entre otros. El detector de imágenes de rayos X, usa métricas de calidad que van a servir para indicar que tan bueno o malo es el modelo, son datos numéricos valores que podrían variar dependiendo del número de capas y neuronas que tenga la arquitectura del modelo, el número de épocas o ciclos de aprendizaje del entrenamiento, matriz de confusión, verdaderos positivos, verdaderos negativos, falsos positivos, falsos negativos, precisión, recall (exhaustividad), F1-score y accuracy (exactitud).

### Síntomas del covid-19

Los síntomas que comúnmente se presentan en una persona para determinar si esta posee COVID-19 son la tos seca, fiebre, malestar o cansancio, pérdida del gusto y olfato, congestión nasal y dolores de cabeza. Aproximadamente el 80% de las personas se recuperan solas, sin necesidad de un tratamiento hospitalario determinado. Lo que indica 1 de cada 5 personas presenta un cuadro grave con complicaciones respiratorias, generalmente estos casos se dan en la población de alto riesgo de contagio, como los adultos mayores, diabéticos, personas con hipertensión arterial, problemas cardíacos etc. Por ello, las personas de cualquier edad que tengan estos síntomas deberán acudir al médico como medida de prevención con el fin de evitar la propagación del virus [3].

### Métodos diagnósticos para covid-19

Hoy en día existe gran cantidad de información con respecto al COVID-19, métodos diagnósticos para el estudio y la detección del virus en un individuo. Entre los tipos de pruebas se destacan los RT-PCR, pruebas serológicas de anticuerpos, pruebas de antígenos, tomografía axial computarizada (TAC)[4].

La prueba de reacción en cadena de la polimerasa (Polymerase Chain Reaction, PCR) o mejor conocida como la prueba de amplificación de ácidos nucleicos (Nucleic acid amplification test, NAAT). Es un tipo de prueba molecular viral, la cual consiste en realizar un hisopado nasal o bucal, la cual sirve para hallar y extraer el material genético del virus. En la actualidad se considera como la prueba más efectiva para la detección de covid-19[5]. Entre sus desventajas se indica que deben ser realizadas por expertos en el área de microbiología molecular, es costoso, requiere de equipos y personal específico, tiene un porcentaje relativo de falsos positivos [5].

Las pruebas serológicas de anticuerpos, como su nombre lo indica, este tipo de prueba tiene como objetivo detectar anticuerpos que el organismo contiene para combatir el virus a través de un análisis en la sangre con el fin de determinar si un individuo precisa de una infección o ha poseído el virus del COVID-19[4]. Se conoce por medio de los estudios realizados, se indica que la sensibilidad para estos test serológicos rápidos (Prueba de anticuerpos combinados IgG-IgM de COVID-19 son variables, observando sensibilidad del 88,6% y especificidad del 90,63% [6].

El método de diagnóstico o test de antígenos, se utiliza para reconocer el revestimiento de las proteínas del virus, mientras que por otra parte las PCR solo identifican el material genético que está hospedado en el mismo. Se realiza por medio de un hisopado nasofaríngeo para extraer las muestras que son depositadas en un kit de análisis, donde se observan los resultados en un corto periodo de tiempo de 15 a 30 minutos por medio de un sistema colorimétrico. En cuanto se refiere a la especificidad, el test arroja un valor aproximado de 99.68% para pacientes evaluados con síntomas, lo cual indica que está a la par con la PCR, la única diferencia es que puede obtener presuntamente cifras negativas, tomando como consideración

un juicio clínico se descartara mediante una prueba molecular [7].

La tomografía axial computarizada (TAC) utiliza una tecnología sofisticada de rayos X, es una prueba de diagnóstico por imágenes que permite capturar en varios tomos las diferentes partes del cuerpo, los combina en una computadora y los muestra en una imagen detallada de 3 dimensiones. Nos permite crear imágenes con detalles de los órganos, huesos y vasos sanguíneos. Permiten conocer, detectar y confirmar la presencia de cáncer, un tumor o alguna enfermedad respiratoria [8].

Según los estudios publicados por la revista 'Radiology', indican que el diagnóstico por tomografía computarizada es mejor que las pruebas de laboratorio para la detección de COVID-19. Este estudio basó su investigación y fue aplicado a 1000 pacientes. Lo que demuestra que las TC a comparación de las pruebas PCR, son más eficientes, prácticas y fiables para analizar la infección por el virus. Se demostró que la sensibilidad fue del 98% para la infección por COVID-19. Luego para un nuevo estudio sobre 1014 personas determinó obtener un 88% de casos positivos para imágenes de tórax y aumentando al 97% en casos asociados a PCR positivo [9].

## 2. INTELIGENCIA ARTIFICIAL

La inteligencia artificial (IA) se refiere a la simulación de la inteligencia humana en máquinas que están programadas para pensar como humanos e imitar sus acciones. El término también se puede aplicar a cualquier máquina que exhiba rasgos asociados con una mente humana, como el aprendizaje y la resolución de problemas [10].

### Modelo de predicción

El análisis predictivo se refiere al uso de datos históricos, aprendizaje automático e inteligencia artificial para predecir lo que sucederá en el futuro. Estos datos históricos se introducen en un modelo matemático que considera tendencias y patrones clave en los datos [11]

### Machine learning

El aprendizaje automático es un método de análisis de datos que automatiza la construcción de modelos analíticos. Es una rama de la inteligencia artificial basada en la idea de que los sistemas pueden aprender de los datos, identificar patrones y tomar decisiones con una mínima intervención humana [12].

### Deep learning

El aprendizaje profundo es una función de inteligencia artificial (IA) que imita el funcionamiento del cerebro humano en el procesamiento de datos y la creación de patrones para su uso en la toma de decisiones. El aprendizaje profundo es un subconjunto del aprendizaje automático en inteligencia artificial que tiene redes capaces de aprender sin supervisión a partir de datos que no están estructurados o etiquetados. También conocido como aprendizaje neuronal profundo o red neuronal profunda [13].

### Aprendizaje supervisado

El aprendizaje supervisado es una técnica que consiste en proporcionar datos etiquetados a un modelo de aprendizaje automático. El conjunto de datos etiquetado generalmente son datos recopilados de la experiencia, también llamados datos empíricos. Además, los datos a menudo requieren preparación para aumentar su calidad, llenar sus vacíos o simplemente optimizarlos para la capacitación [14].

### Técnica de clasificación

La clasificación es un proceso de categorizar un determinado conjunto de datos en clases. El proceso comienza con la predicción de la clase de puntos de datos dados. Las clases a menudo se denominan destino, etiqueta o categorías.

### 3. HERRAMIENTAS UTILIZADAS

**GOOGLE COLAB** es un entorno gratuito de Jupyter Notebook que no requiere configuración y que se ejecuta completamente en la nube. Jupyter es un entorno interactivo que nos permite desarrollar código Python de manera dinámica, la máquina en un inicio cuenta con 12 GB de RAM y 50 GB de almacenamiento en disco disponibles para el uso con tres tipos de unidad de procesamiento como CPU, GPU Y TPU [15].

**GOOGLE DRIVE** es un servicio gratuito de almacenamiento en la nube que les permite a sus usuarios acceder, crear copias de seguridad de todos los archivos y compartir documentos. Por medio de una cuenta de Gmail los usuarios tendrán acceso a todos los servicios que GOOGLE ofrece, cuenta con un almacenamiento de 15GB y este puede extenderse utilizando la versión de pago [16]. Una de las funciones más importantes de GOOGLE DRIVE es que permite trabajar de forma colaborativa con otros servicios de GOOGLE, como lo es el entorno de desarrollo de GOOGLE COLAB para extraer archivos almacenados en la nube y usarlos en la elaboración de modelos de machine learning.

**KERAS** es una librería de redes neuronales escrita en Python. Se ejecuta sobre la plataforma de aprendizaje automático de TensorFlow, para realizar experimentos con redes neuronales artificiales. Por lo cual se menciona lo siguiente [17]. Proporciona la creación de modelos de forma rápida y sencilla, de otro modo, está desarrollado para simplificar la carga mental y reducir la cantidad de acciones por el usuario en los casos de uso más comunes.

**PYTHON** es un lenguaje de programación que tiene como finalidad dirigirse a los usuarios de forma general, es potente, flexible, simple y fácil de aprender. Por otra parte, es considerado como un lenguaje de alto nivel, ya que permite procesar de manera sencilla todo tipo de estructuras de datos, tanto numéricos como de texto [18].

**TENSORFLOW** es una plataforma de código abierto, específicamente es un instrumento de computación numérico para resolver cálculos complejos siendo esta un marco de aprendizaje de alto nivel para el desarrollo de modelos basados en ML con inteligencia artificial [19].

### 4. TRABAJOS RELACIONADOS

A continuación, se mencionan los modelos que guardan relación en el campo de la medicina por la pandemia de COVID-19 y el análisis de imágenes basados en redes neuronales convolucionales.

#### Sistema de aprendizaje profundo para detectar la neumonía de 2019 por la enfermedad del nuevo coronavirus

El proyecto de aprendizaje profundo tiene como objetivo la detección prematura de COVID-19 mediante imágenes de tomografía computarizada (TC), en el cual se plantea clasificar los casos como neumonía, sano y otro tipo de neumonía como lo es la IAVP (Neumonía Viral por Influenza). Utiliza un método de clasificación, una estructura de red neuronal convolucional ResNET. Obtuvieron como resultado experimental una accuracy (Exactitud) general del 86.7% para términos usados de todos los casos de TC del conjunto de datos [20].

#### Detección de la enfermedad por coronavirus (covid-19) basada en características profundas y máquina de vectores de soporte

El proyecto tiene el objetivo a consecuencia de la clasificación de personas infectadas para reducir la propagación del virus, esto simula una tarea compleja. Usa el método de la máquina de vectores de soporte (SVM) a través de rayos X, con el modelo pre-entrenado Resnet-50 con los que obtuvo mejores de todos los modelos pre-entrenado que experimento en su estudio. Los resultados que obtuvo en el accuracy (Exactitud) está en 95.33%, realizado con 20 ciclos independientes en entrenamiento [21].

#### Detección automática de la enfermedad por coronavirus (covid-19) mediante imágenes de rayos x y redes neuronales convolucionales profundas

Como medida de apoyo ante las circunstancias, es de mucha importancia la creación de un sistema de detección automática para ofrecer un diagnóstico rápido como alternativa para la disminución de personas contagiadas con COVID-19. Realizo un estudio comparativo con otros modelos pre-entrenados, como resultado final obtuvo una exactitud del 98% utilizando ResNet50 como el mejor entre todos los modelos [22].

#### Modelo predictivo de robo a casas basado en redes neuronales

El proyecto tendrá la capacidad de medir la probabilidad de ocurrencia del evento catalogado como delito de robo a casas, está basado en redes neuronales que permitirán la obtención de zonas de mayor y menor riesgo para un lugar determinado. El método empleado fue el Support Vector Machine SVM (máquina de vectores de soporte), también utilizó un modelo de redes neuronales convolucionales, del estudio comparativo entre los dos modelos, el de redes neuronales convolucionales obtuvo un resultado de (accuracy 69.17%, precisión 66.04%, recall 74.16% y F1 69.72). El resultado del SVM fue (accuracy 71.17%, precisión 69.38%, recall 71.90% y F1 70.40%) cuyo tiempo de ejecución entre los dos modelos fue: 1 minuto con 17 segundos para la red neuronal convolucional, 42 minutos y 36 segundos para la SVM. Del análisis comparativo dedujo que las redes neuronales convolucionales son más eficientes en cuanto al tiempo de ejecución [23].

### 5. IMÁGENES DE RAYOS X

#### ¿Qué son rayos X?

Los rayos X producen radiación y emiten energía parecida a la luz y ondas electromagnéticas, al momento de hacer contacto con el cuerpo humano es absorbida por los tejidos, órganos internos y huesos. Con la ayuda de un monitor similar al de una computadora se pueden crear imágenes médicas, permitiendo al radiólogo el especialista encargado de esa área observar anomalías dentro de la estructura del cuerpo [24].

#### Tipos de rayos X

La radiología es una rama de la medicina que utiliza la tecnología imagenológica para diagnosticar y tratar una enfermedad. Los tipos más comunes de exámenes radiológicos de diagnóstico incluyen: radiografía de tórax, mamografía, (TC) tomografía computarizada, Fluoroscopia.

## Radiografía de tórax

Las imágenes de radiografía de tórax, son muy fáciles rápidas de obtenerlas sirven de gran ayuda cuando se presenta alguna emergencia. Generan imágenes del interior del tórax utilizando muy poca radiación, para poder hacer una revisión completa de los pulmones, corazón y pared torácica. Por medio de estas pruebas se puede realizar una valoración médica sobre el tipo de enfermedad que puede padecer una persona, y obtener un tratamiento adecuado y seguro. Síntomas como fiebre, problemas respiratorios, tuberculosis, molestias en el pecho, neumonía y cáncer, son los casos más comunes que se pueden presentar [25].

## TC (tomografía computarizada)

La tomografía computarizada permite ver las estructuras internas del cuerpo más detalladamente a diferencia de las radiografías simples, se generan en diferentes ángulos formando una imagen tridimensional, permitiéndole al médico tener un mejor diagnóstico [26].

## 6. CLASIFICACIÓN DE LAS REDES NEURONALES ARTIFICIALES

### Red Neuronal Monocapa - Perceptrón simple

Un perceptrón es un algoritmo de clasificación binaria simple, ayuda a dividir un conjunto de señales de entrada en dos partes: "sí" y "no". Un perceptrón es una máquina de aprendizaje muy simple porque puede tomar algunas entradas, cada una de las cuales tiene un peso para indicar qué tan importante es para generar una decisión de salida de "0" o "1" [27].

### Red Neuronal Multicapa - Perceptrón Multicapa

Los perceptrones multicapa también se conocen como "redes neuronales de alimentación hacia adelante", estas son redes más complejas que el perceptrón, ya que consisten en múltiples neuronas que están organizadas en capas. Entre las capas, se puede distinguir una capa de entrada, capas ocultas y una capa de salida, los perceptrones multicapa suelen estar completamente conectados, esto significa que hay una conexión de cada perceptrón en una capa específica a cada perceptrón en la siguiente capa [28].

### 6.1 Red neuronal convolucional (CNN)

Las redes neuronales convolucionales son muy similares a las redes neuronales ordinarias, están formadas por neuronas que tienen pesos y sesgos que se pueden aprender. Cada neurona recibe algunas entradas, realiza un producto escalar y, opcionalmente, lo sigue con una no linealidad. Toda la red todavía expresa una única función de puntuación diferenciable: desde los píxeles de la imagen sin procesar en un extremo hasta las puntuaciones de la clase en el otro [29].

Estas redes son muy utilizadas para el análisis, procesamiento y extracción de características de imágenes, esto se debe al funcionamiento que realizan la capa de entrada, la capa oculta y la capa de salida. Por lo tanto, al momento de analizar la imagen en la primera capa se pueden obtener los bordes o líneas de la imagen, la siguiente capa tiene la tarea de encontrar un grupo de patrones para poder reconocer estructuras más sencillas de la imagen, mientras que la última capa es la encargada de agrupar todos los patrones encontrados para poder realizar una predicción final [30].

## Estructura de la CNN

Los elementos que integran una red neuronal convolucional están formadas por cinco tipos de capas, tales como: capa de entrada, capa convolucional, capa de reducción o pooling, capa clasificadora totalmente conectada y capa de salida:

### Capa de entrada

La capa de entrada tomará los píxeles de la imagen y estará formado por los siguientes componentes: (alto, ancho, canales).

- Donde alto y ancho corresponden a sus dimensiones y el canal estará determinado por la representación del color, estos pueden ser: 1 si la imagen es en escala de grises o 3 si es para RGB (rojo, verde, azul).
- Por ejemplo, tenemos una imagen con las siguientes dimensiones: 28x28x3 (alto, ancho, canales)

### Capa convolucional

Esta segunda capa toma los píxeles de la imagen de entrada, se ejecuta matemáticamente a través de varios kernel o filtros, una pequeña matriz que va recorriendo todas las neuronas de entradas dando como resultado una nueva matriz de salida, lo que significa que sería una nueva capa de neuronas ocultas [31].

### Capa de reducción o pooling

La tercera capa de la CNN, es la encargada de reducir las dimensiones de la altura y anchura de la matriz obtenida en la primera capa. Utiliza una técnica de agrupación Max-pooling (agrupación máxima) que es la más usada en estas redes, la cual consiste en escoger el pixel con el valor más alto que toma el kernel en una región [32].

### Capa clasificadora totalmente conectada

Es muy parecida a un perceptrón multicapa, esto se debe a que luego de obtener las características más importantes de una imagen en la capa de convolución y reducción, toma cada pixel como una neurona separada y por capas interconectadas entre sí. Ahora esta última capa clasificadora va tener varias clases y neuronas para poder obtener una salida de datos esperada[33].

### Capa de salida

Una función de activación en una red neuronal define cómo la suma ponderada de la entrada se transforma en una salida de un nodo o nodos en una capa de la red.

La función sigmoid toma cualquier valor real como valores de entrada y de salida en el rango de 0 a 1. Cuanto mayor sea la entrada (más positiva), más cerca estará el valor de salida a 1.0, mientras que cuanto más pequeña sea la entrada (más negativa), más cerca estará la salida será 0.0[34].

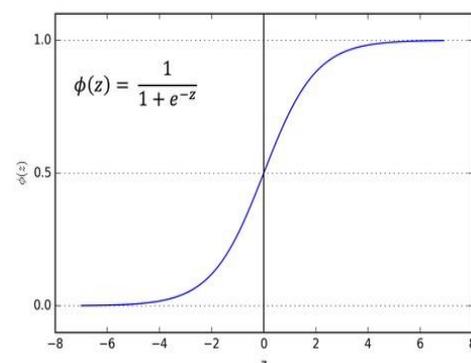


Figura 1: Función sigmoide[34]

## Función activación ReLU

Función ReLU (Unidad lineal rectificadora) es una operación inteligente que reemplaza todos los valores de píxeles negativos en el mapa de características con 0 [35].

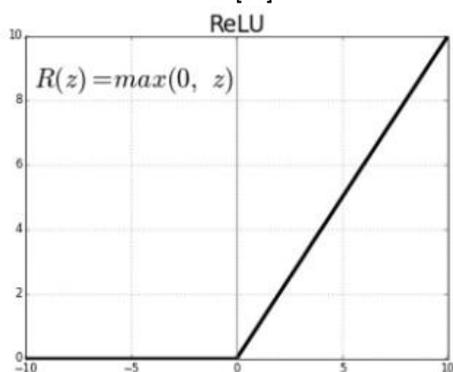


Figura 2: Función ReLU[35]

## Red neuronal recurrente

La red neuronal recurrente proporciona un circuito de retroalimentación a la neurona anterior. Esta arquitectura permite recordar información importante sobre la entrada que recibió la red y toma en consideración el aprendizaje junto con la siguiente entrada. Los RNN funcionan muy bien con datos secuenciales como sonido, datos de series de tiempo (sensores) o lenguajes naturales escritos [36].

## 7. METODOLOGÍA

El presente trabajo de investigación está formado por etapas, las cuales se describen a continuación: análisis de de la data, elaboración del modelo, entrenamiento, evaluación de métricas y predicción, conclusiones.

### ANÁLISIS DE LA DATA - CONJUNTO DE DATOS

#### 7.1 Recopilación de imágenes

Para poder reunir las imágenes de rayos X se acudió a las plataformas de GitHub [37] y Kaggle [38] [39], es un medio público y de fácil acceso se pueden obtener varias bases de datos las cuales pertenecen a una comunidad científica de datos. Los conjuntos de datos (dataset) que han sido seleccionadas para este estudio de análisis fueron, imágenes de radiografía de tórax de pacientes confirmados por Covid-19 (Fig. 3a) y de pacientes que se encuentren sanos (normal) (Fig. 3b).

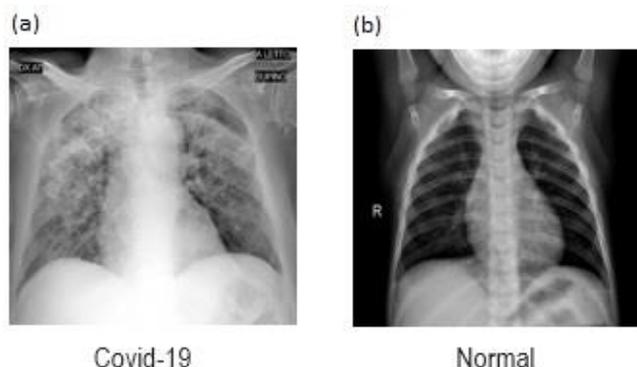


Figura 3: Comparación de las dos clases utilizadas

## 7.1.2 Estructuración y normalización

Las imágenes en los dataset por lo general están formadas por dimensiones de 1024 x 1024 píxeles en formatos JPG y PNG. Luego se realiza la normalización de las imágenes a una dimensión de 224 x 224 píxeles manteniendo cualquiera de los formatos, este proceso permite trabajar con lucidez usando las redes neuronales convolucionales.

En la tabla 1 se puede observar la distribución de los dataset y el número total de imágenes por cada modelo.

BASE DE DATOS DE IMAGENES	MODELO	CONJUNTO DE DATOS		TOTAL DE IMÁGENES DATASET
		TRAIN	VALIDATION	
GITHUB	A	154	79	233
KAGGLE	B	1091	311	1402
KAGGLE	C	1812	679	2491

Tabla 1: Estructura de los conjuntos de datos.

A continuación, en la tabla 2 se observa la división de las imágenes según las clases Covi-19 y Normal. También se presenta el total general de imágenes recopiladas.

MODELO	TRAIN		VALIDATION		TOTAL
	COVID-19	NORMAL	COVID-19	NORMAL	
A	77	77	16	63	233
B	530	561	116	195	1402
C	902	910	241	438	2491
TOTAL	1509	1548	373	696	4126

Tabla 2: Estructura de los conjuntos de datos por clases

## ELABORACIÓN DEL MODELO

### 7.2 Arquitecturas para pruebas experimentales

Para realizar el modelo de predicción de Covid-19 usando redes neuronales convolucionales, se utilizó el entorno de desarrollo de Google Colaboratory. Esta herramienta fue creada por Google porque permite trabajar con el lenguaje de programación Python, ya que ofrece un entorno de ejecución gratuito de unidades de procesamiento gráfico (GPU) que permite acelerar los procesos de entrenamiento presentes en los modelos. En la tabla 3 se muestran las pruebas experimentales realizadas por cada modelo, mediante las características asignadas se va poder verificar cual es el modelo con mejor resultado.

CARACTERÍSTICAS DE UNA CNN	MODELOS		
	A	B	C
NEURONAS DE LA CAPA DE ENTRADA	80	160	320
CAPAS OCULTAS	3	6	9
NUMERO DE ÉPOCAS	5	10	15
TOTAL DE CAPAS DE LA CNN	5	8	11
TOTAL DE PARAMETROS EJECUTADOS	19.391.161	21.401.551	32.712.737

Tabla 3: Pruebas experimentales

## 7.2.2 Diseño de arquitecturas CNN

### Modelo A

Para el primer modelo experimental (Fig. 4), se utilizó una arquitectura con 1 capa de entrada convolucional conforme a las dimensiones de las imágenes (224 x 224 x 3), 3 capas ocultas (1 de convolución, 1 de Max-Pooling y 1 capa Full connected) y 1 capa de salida con 2 neuronas full connected acorde a las 2 opciones posibles de resultado de clasificación.

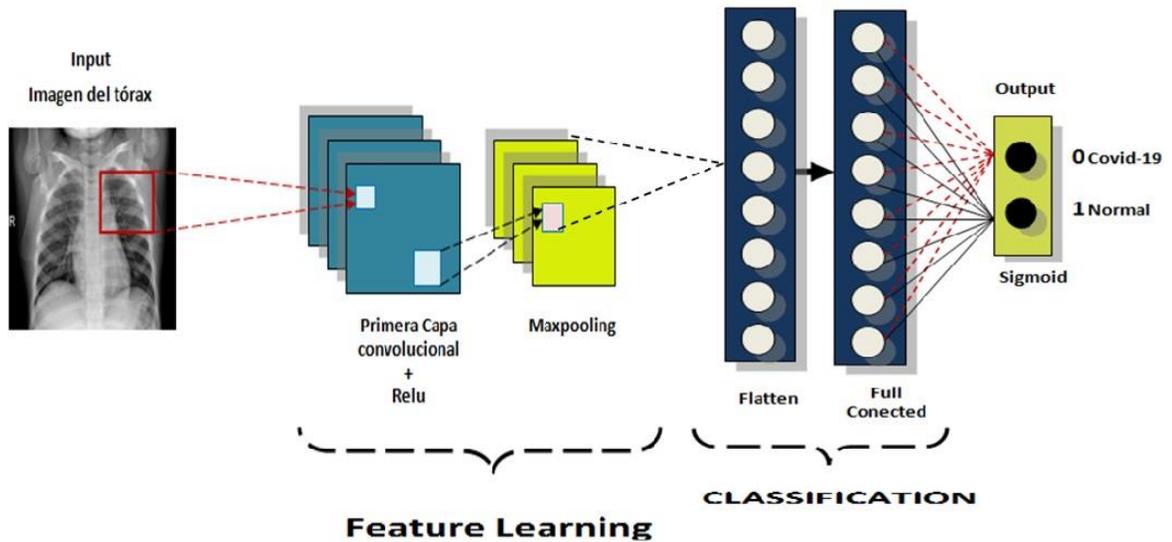


Figura 4: Arquitectura del modelo A

### Modelo B

Para el segundo modelo experimental (Fig. 5), se utilizó una arquitectura con 1 capa de entrada convolucional conforme a las dimensiones de las imágenes (224 x 224 x 3), 6 capas ocultas (2 de convolucionales, 2 de Max-Pooling y 2 capa Full conected) y 1 capa de salida con 2 neuronas full conected acorde a las 2 opciones posibles de resultado de clasificación.

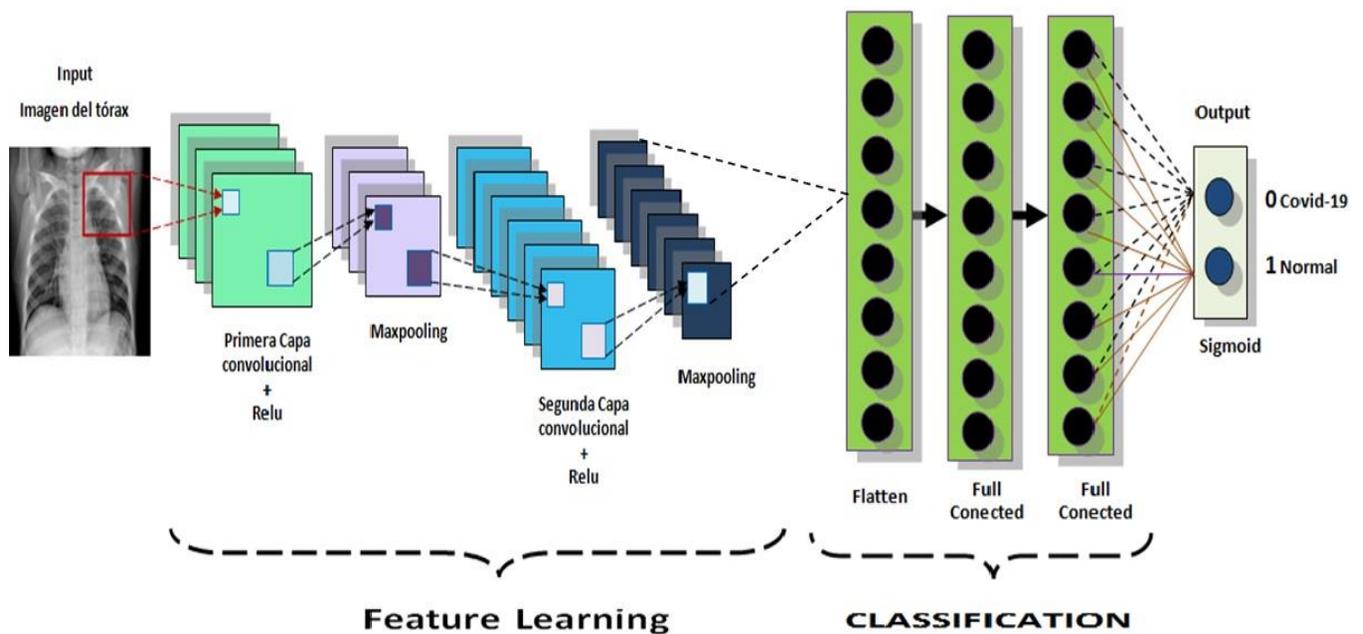


Figura 5: Arquitectura del modelo B

### Modelo C

Para el tercer modelo experimental (Fig. 6), se utilizó una arquitectura con 1 capa de entrada convolucional conforme a las dimensiones de las imágenes (224 x 224 x 3), 9 capas ocultas (3 de convolucionales, 3 de Max-Pooling y 3 capa Full conectad) y 1 capa de salida con 2 neuronas full conectad acorde a las 2 opciones posibles de resultado de clasificación.

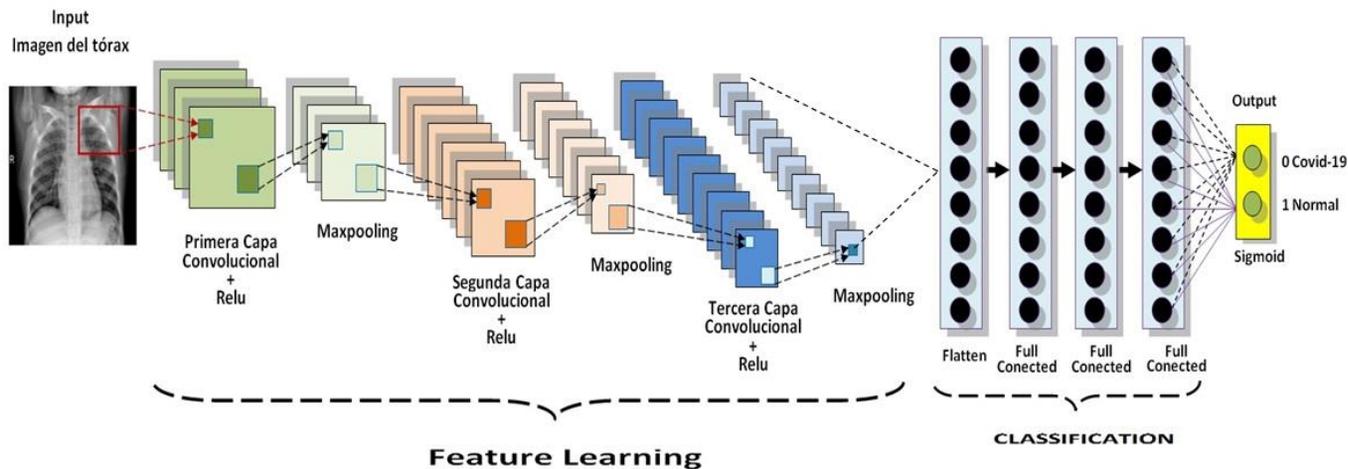


Figura 6: Arquitectura del modelo c

### 7.2.3 Pre-procesamiento de imágenes

Para el estudio y la realización del modelo se tomaron en consideración diferentes hiperparámetros para el pre-procesamiento de imágenes, con el fin de ajustar valores predeterminados que nos servirán para analizar el rendimiento del modelo. En las pruebas experimentales del modelo A, B y C se describen los siguientes:

#### Normalización de los datos

Distribuye todos los datos para hacerlos más homogéneos. Se realizan estos pasos para hacer más suave la curva de la generación, para que todas las imágenes tengan el mismo modelo de color y la misma resolución, para este caso seleccionamos para un solo canal., esto evitara que el sistema no presente fallas y la red extraiga los valores correctamente, las imágenes están en modelo de color RGB que van de 0 (negro) a 255 (blanco) pasando las imágenes a un solo canal es decir a escala de grises donde 0 (ausencia de color) y 1 (luminosidad).

#### Data augmentation

Sirve para ajustar la imagen en modo horizontal, de manera que la imagen no presente variaciones de acuerdo a la posición en que se encuentra, para este caso se usa este hiperparámetro para que todas las imágenes tengan el mismo tamaño y sean más visibles. Por medio de esta técnica se evita el sobreajuste que normalmente se presentan en las redes neuronales convolucionales, ayudando a tener un mejor rendimiento en la etapa de entrenamiento.

#### Batch\_size

Es el tamaño de lote, es decir es la cantidad de imágenes que pasará el modelo en los pasos por épocas. Los valores del tamaño se indican en la (Fig. 7).

#### Optimizador ADAM

Este algoritmo de optimización es utilizado para solucionar la incertidumbre sobre la fijación que hay entre el ratio de aprendizaje del descenso del gradiente estocástico (SGD) para actualizar de forma iterativa en función de los datos de entrenamiento.

HYPERPARÁMETROS	MODELOS		
	A	B	C
Modo de color	Modo RGB	Modo RGB	Modo RGB
Data augmentation	Giro Horizontal con variación de zoom	Giro Horizontal con variación de zoom	Giro Horizontal con variación de zoom
Función de coste / Loss	Bynary Cross_Entropy	Bynary Cross_Entropy	Bynary Cross_Entropy
Optimizador	ADAM	ADAM	ADAM
Taza de aprendizaje -Metrics	ACC	ACC	ACC

Tabla 4: Hiperparámetros

#### Función de regularización Dropout

Este método se encarga de deshabilitar neuronas con el fin de reducir el sobreajuste en las imágenes, las neuronas cercanas no tienen una conexión directa con las neuronas deshabilitadas, lo cual les permite trabajar mejor de forma única para obtener patrones determinados en la etapa de entrenamiento. Toma un valor estimado entre 0 y 1, donde 0.5 indica por defecto la mitad de neuronas deshabilitadas, 1 hace referencia al valor máximo del mayor número de neuronas deshabilitadas. En Keras es lo opuesto mientras el valor este por debajo de 0.5, mayor serán las neuronas deshabilitadas. Por lo tanto, los modelos en Keras siempre utilizan valores altos para las capas ocultas y en término medio para las capas densas [40].

## Función de pérdida (Loss)

Mejor conocida como función objetivo nos permite conocer en la etapa de entrenamiento de una red neuronal que tan próximo se encuentra al valor mínimo, también sirve para saber qué tanto es el error aproximado que debemos de obtener para saber con exactitud si el modelo está aprendiendo bien, de otra forma más exacta nos muestra una salida del resultado esperado. En la fase de entrenamiento la función de pérdida está considerada como un valor de entrada proporcionado por el algoritmo de aprendizaje. La función de pérdida utiliza algoritmos de aprendizaje basados en el descenso de gradiente, estos valores deben ser continuos y derivables [41].

## Binary cross\_entropy

Esta función de coste calcula la entropía cruzada entre las etiquetas verdaderas con las etiquetas predichas. Dentro del contexto, sirve para clasificar variables binarias donde solo existan 2 clases posibles. Por ejemplo, estos valores posibles son 0 y 1, para cada ejemplo, de la predicción debería haber un único valor de salida.

## Sobreajuste (Overfitting)

El sobreajuste consiste en la pérdida de propagación, esto quiere decir que el modelo tendrá un comportamiento bueno en el proceso de aprendizaje, pero no podrá reconocer nuevos patrones por errores, ruido o desperfectos que contenga una imagen.

## 8. ENTRENAMIENTO

Para el estudio del primer modelo se utilizó 3 capas ocultas para una arquitectura básica, utilizando una función ReLU para la capa convolucional y para la capa densa. Luego realizamos el entrenamiento del modelo A, lo definimos con 5 épocas, también se utilizó 4 pasos por época y el resultado de la ejecución fue 36.588 seg aproximadamente como prueba experimental.

Para el modelo B, aumentamos a escala intermedia la complejidad añadiendo 3 capas ocultas más, obteniendo un total de 6 capas ocultas. Cada capa convolucional consta de una función ReLU y una función de regularización (Dropout) de 0.15 para reducir el sobreajuste (overfitting) y para las dos capas densas se usó un valor de 0.25 como regularizador. El entrenamiento se definió con 10 épocas y 7 pasos por época, obteniendo un tiempo de ejecución de 166.408 seg.

Por último, para el modelo C se añadió 3 capas ocultas más obteniendo un total de 9 capas ocultas. Cada capa convolucional consta de una función ReLU y una función de regularización (Dropout) de 0.25 y 0.5 para las capas densas. Se realizó el entrenamiento con 15 épocas, utilizando 7 pasos por época. Para finalizar, se obtuvo como resultado un tiempo de ejecución de 177.644 seg.

Para confirmar el tamaño del dataset y la arquitectura utilizado por los tres modelos se recomienda observar las tablas 1 y 2. Y para visualizar las arquitecturas empleadas observar las (Fig. 4, 5,6)

Para la realización de la clasificación con una CNN es importante conocer la profundidad y los parámetros de entrada de esta. Las capas de entrada de la CNN definirán el tamaño de la imagen a ser clasificada. Los parámetros establecidos de las CNN utilizadas para la clasificación de imágenes pueden consultarse en la documentación de TensorFlow y Keras [42].

## Gráfica de la precisión y pérdida durante el entrenamiento

Para el modelo A observamos que obtuvo una precisión máxima (Maximum accuracy) de 0.91667, mientras que para la función de pérdida (Loss), observamos que obtuvo un valor de 0.3283. Estos resultados se mostraron en la época 5 para ambas funciones, considerada la mejor época de todas las iteraciones (Fig. 7).

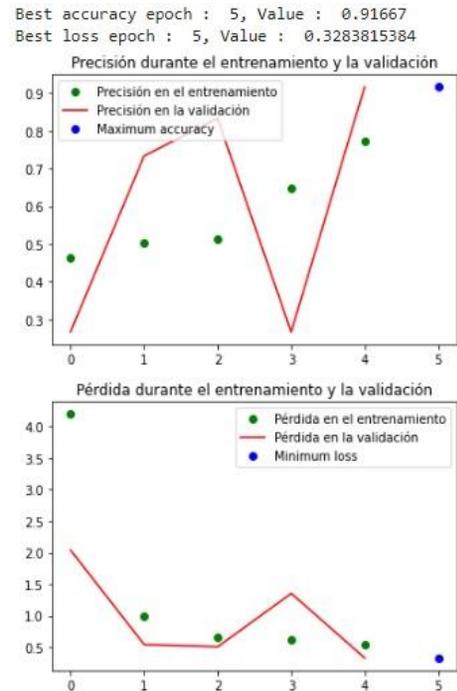


Figura 7: Modelo A

Para el modelo B observamos que obtuvo una precisión máxima de 1.0000, mientras que para la función de pérdida obtuvo un valor de 0.1402. Estos resultados se mostraron en la época 4 para la accuracy durante el entrenamiento, y mejor época para la pérdida durante el entrenamiento y la validación (Fig. 8).

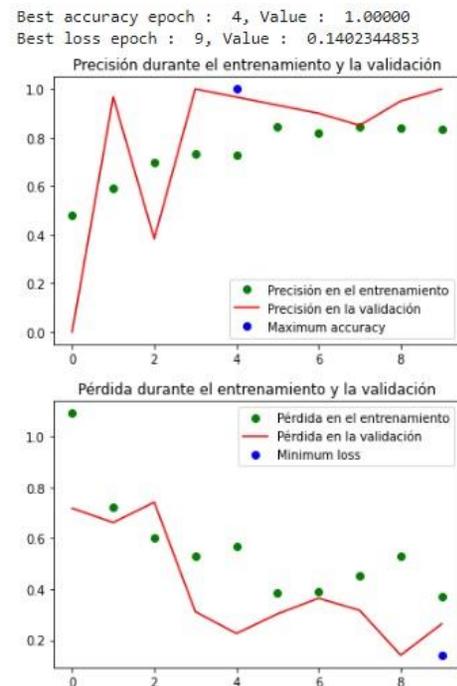


Figura 8: Modelo B

Para el modelo C observamos que obtuvo precisión máxima de 1.0000, mientras que para la función de pérdida obtuvo un valor de 0.1245. Estos resultados se mostraron en la época 13 para ambas funciones durante el entrenamiento y la validación (Fig. 9).

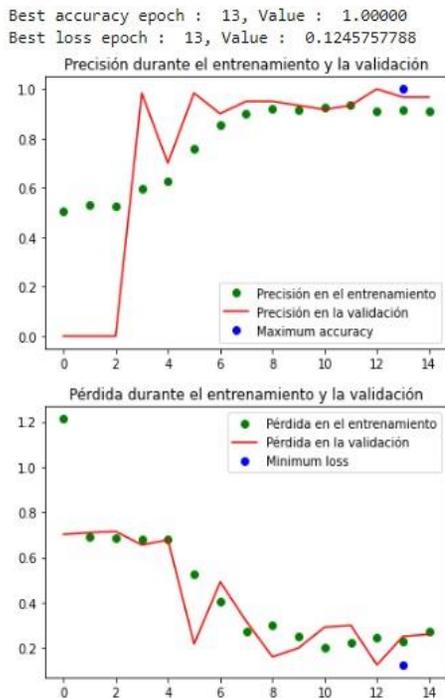


Figura 9: Modelo C

Los resultados obtenidos para todos los modelos se dieron mediante el ajuste de los hiperparámetros tomados de la tabla 4.

## 9. EVALUACIÓN DE LAS MÉTRICAS Y PREDICIÓN

Para verificar que tan bueno fueron los resultados obtenidos para cada modelo, se usaron las métricas de calidad para realizar las comparaciones de los valores de precisión, sensibilidad (recall) y F1-Score para determinar qué modelo es el más óptimo para la predicción de Covid-19.

Para poder utilizar este método de evaluación de un modelo de clasificación necesitamos separar nuestro dataset principal en dos conjuntos de datos.

- Train 80%
- Validation 20%

El conjunto de entrenamiento sirve para ajustar el modelo, lo que significa que en la fase de entrenamiento el algoritmo ira aprendiendo de las características extraídas de las imágenes.

El conjunto de validación nos ayuda a evaluar los ajustes referentes al modelo y a su vez ajustar los parámetros del modelo.

### Matriz de confusión

		predicción	
		0	1
realidad	0	TN	FP
	1	FN	TP

Tabla 5: Modelo de una matriz de confusión [44]

Donde:

- **True Positive** es la cantidad de positivos que fueron clasificados correctamente como positivos por el modelo.
- **True Negative** es la cantidad de negativos que fueron clasificados correctamente como negativos por el modelo.
- **False Negative** es la cantidad de positivos que fueron clasificados incorrectamente como negativos.
- **False Positive** es la cantidad de negativos que fueron clasificados incorrectamente como positivos.

### Precisión

Se refiere a la calidad de los resultados obtenidos luego de realizar las clasificaciones de casos positivos y el total de predicciones positivas de forma correcta, un valor que se acerca más a una predicción verdadera [43]

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

### Sensibilidad (Recall)

Esta métrica, nos indica la cantidad de datos de clase positiva que el modelo ha sido capaz de identificar [44].

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

### F1- Score

Esta métrica sirve para combinar y comparar el rendimiento de los valores de precisión y recall [44].

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

A continuación, se detallan los cálculos de las métricas por cada modelo:

En la (Fig. 10), se puede observar que los valores de la diagonal principal TN= 13 y TP= 59 corresponden con los valores estimados de forma correcta por el modelo.

Mientras que la otra diagonal, representa los casos en los que el modelo se ha equivocado FN= 4 y FP= 3

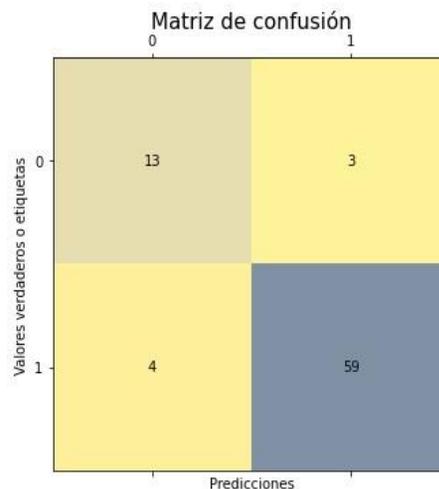


Figura 10: Modelo A

$$\text{Precisión} = \frac{59}{59 + 3} = 0.9516129$$

$$\text{Recall} = \frac{59}{59 + 4} = 0.9365079$$

$$F1 = 2 * \frac{0.9516129 * 0.9365079}{0.9516129 + 0.9365079} = 0.9440089$$

En la (Fig. 11), se puede observar que los valores de la diagonal principal TN= 115 y TP= 324 corresponden con los valores estimados de forma correcta por el modelo.

Mientras que la otra diagonal, representa los casos en los que el modelo se ha equivocado FN= 21 y FP= 1.

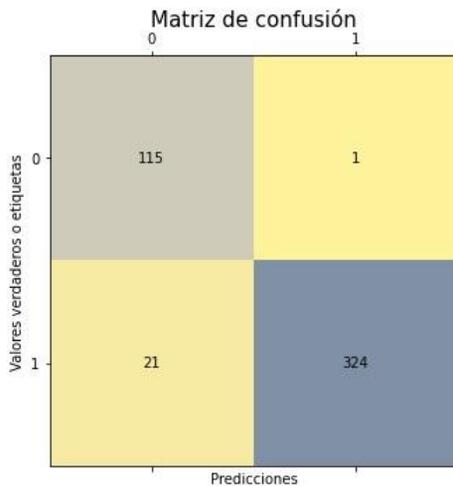


Figura 11: Modelo B

$$\text{Precisión} = \frac{324}{324 + 1} = 0.9969231$$

$$\text{Recall} = \frac{324}{324 + 21} = 0.9391304$$

$$F1 = 2 * \frac{0.9969231 * 0.9391304}{0.9969231 + 0.9391304} = 0.9671642$$

En la (Fig. 12), se puede observar que los valores de la diagonal principal TN= 238 y TP= 421 corresponden con los valores estimados de forma correcta por el modelo.

Mientras que la otra diagonal, representa los casos en los que el modelo se ha equivocado FN= 17 y FP= 3.

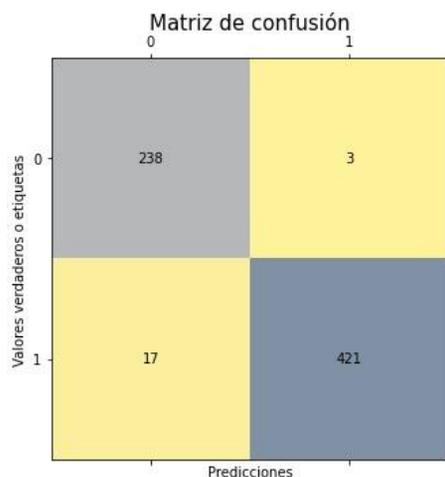


Figura 12: Modelo C

$$\text{Precisión} = \frac{421}{421 + 3} = 0.9929245$$

$$\text{Recall} = \frac{421}{421 + 17} = 0.9611872$$

$$F1 = 2 * \frac{0.9929245 * 0.9611872}{0.9929245 + 0.9611872} = 0.9767981$$

## Pruebas de los modelos por conjuntos de datos

Como podemos observar en la tabla 6, el modelo A obtuvo un valor de precisión del 0.9516 con una sensibilidad (Recall) del 0.9365 y un F1-Score de 0.9440. Tiene un rendimiento bueno para ser el primer experimento, pero no llegó a su punto máximo de eficiencia ya que presentaba sobreajuste (Fig. 7), pero aun así con pocas épocas pudo realizar una clasificación buena.

DATASETS	MODELO	NUMERO DE IMÁGENES	PRECISIÓN	SENSIBILIDAD (RECALL)	F1-Score
GITHUB	A	233	0.9516129	0.9365079	0.9440089
KAGGLE	B	1402	0.9969231	0.9391304	0.9671642
KAGGLE	C	2401	0.9929245	0.9611872	0.9767981

Tabla 6 total de resultados por cada modelo

El modelo B a diferencia del modelo A tiene una pequeña mejora, obtuvo un valor aproximado de 0.0453102 para la precisión, también obtuvo un aumento en la sensibilidad de un valor aproximado de 0.0026225, y en cuanto al promedio del F1-Score también tuvo un ligero aumento de un valor de 0.0231553. Para este caso se usó una función de regularización para reducir el sobreajuste que presentamos en el modelo A, cuyas características se encuentran en la etapa de entrenamiento y en la (Fig. 8).

Para el modelo final se aumentó el número de imágenes del conjunto de datos. Ahora, observamos cómo se comporta el modelo con un tamaño de dataset más grande, para que la extracción de características con el aumento de capas y aumentando la función de regularización ayuden a reducir la carga computacional.

Este modelo obtuvo una precisión de 0.9929245 para este caso obtuvo una diferencia cercana del 0.0039986 a comparación del modelo B, luego observamos que tiene un aumento en la sensibilidad y en el F1-Score, teniendo una diferencia de 0.0220568 y 0.0096339 para ambas métricas. Según indica [45] "las CNN aprenden a detectar diferentes características de una imagen mediante decenas o cientos de capas ocultas. Cada capa oculta aumenta la complejidad de las características de la imagen aprendidas". Es por estas razones que notamos una pequeña reducción en la precisión.

Predicciones (ponderación y resultados)

Para realizar la predicción el modelo toma una imagen del conjunto de validación que no ha visto en el proceso de entrenamiento para analizar y comprobar si el modelo está cumpliendo la función de clasificación correctamente.

Para trabajar con el ambiente de desarrollo de Google Colaboratory, el orden de las carpetas del conjunto de datos está estructurado alfabéticamente.

Establecemos que aquellas predicciones con valores mayores a 0.5, sean categorizadas como Positivas (True), y aquellas predicciones con valores menores a 0.5, sean clasificadas como Negativas (False).

- 0 False = Tiene Covid-19
- 1 True = Sano

A continuación, se muestran los siguientes resultados:

MODELO	PREDICCIONES POR CLASE	
	COVID-19	NORMAL
A	0.36%	0.70%
B	0.34%	0.58%
C	0.05%	0.97%

Tabla 7: Resultados predicciones

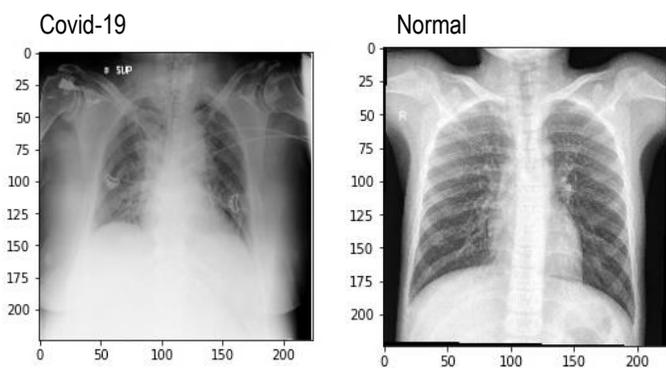


Figura 13: Imágenes de rayos X de tórax modelo A

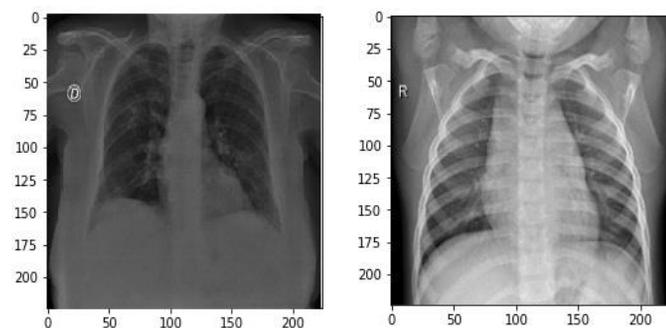


Figura 14: Imágenes de rayos X de tórax modelo B

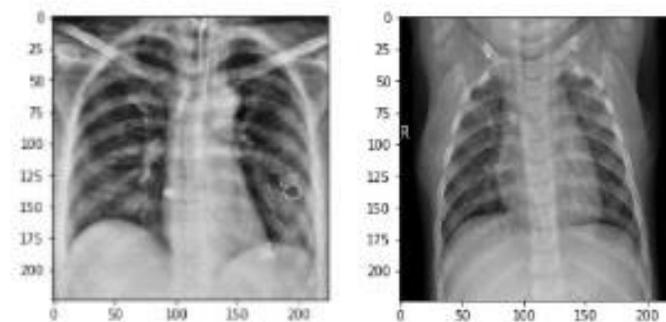


Figura 15: Imágenes de rayos X de tórax modelo C

## 10. CONCLUSIONES

Se concluye con base al estudio y a las pruebas experimentales realizadas de los modelos propuestos, que el modelo C a comparación de los demás ha obtenido los mejores resultados. En consecuencia, se logró una precisión de 0.9929, sensibilidad de 0.9611 y F1 0.9767, lo cual es un resultado muy bueno para lograr la clasificación y detección de patrones mediante imágenes de rayos x con éxito.

Se concluye que el uso de redes neuronales convolucionales para la elaboración de modelos computacionales fue satisfactorio, porque le permite al desarrollador investigar y experimentar el comportamiento de los modelos basados en machine learning para la resolución de problemas de clasificación y la detección de enfermedades raras a través del análisis de imágenes.

Se concluye que el tamaño del conjunto de datos también puede incidir en el rendimiento del modelo, mientras mayor sea el conjunto de datos el modelo aprenderá mucho más. La importancia del uso de las redes neuronales convolucionales para el análisis de datos es efectivo, porque permite crear modelos de ML con tan solo unas pocas épocas para el entrenamiento, con el objetivo de disminuir la carga computacional haciendo más rápido el proceso de ejecución.

Por lo tanto, el modelo de machine learning para la predicción de Covid-19 cumple con los requerimientos necesarios para ser utilizado como una herramienta tecnológica de clasificación confiable y precisa de acuerdo con los resultados obtenidos.

Es por esto la importancia del modelo de machine learning, porque es capaz de clasificar y predecir en cuestión de minutos si un paciente presenta o no síntomas de Covid-19 mediante el análisis de imágenes de rayos X. Durante el entrenamiento del conjunto de datos la red neuronal convolucional aprende y se vuelve más sofisticada para memorizar y extraer los patrones de un objeto en el espacio. Con base a los resultados obtenidos el modelo podrá ser usado como una herramienta tecnológica para el diagnóstico y la toma de decisiones. Permitirá reducir los costos de una prueba tradicional de Covid-19, con el fin de disminuir el congestionamiento en los centros de salud y la ralentización de los procedimientos médicos.

## 11. BIBLIOGRAFÍA

- [1] R. D. Grando *et al.*, "Clinical usefulness of tomographic standards for COVID-19 pneumonia diagnosis: Experience from a Brazilian reference center," *Brazilian J. Infect. Dis.*, vol. 24, no. 6, pp. 524–533, Nov. 2020, doi: 10.1016/j.bjid.2020.10.002.
- [2] L. Marcos, "Un algoritmo permite diferenciar al instante a pacientes con COVID-19," May 2020. .
- [3] Organización Mundial de la Salud (OMS), "Preguntas y respuestas sobre la enfermedad por coronavirus (COVID- 19)," *Organizacion mundial de la salud*, 2020. [https://www.who.int/es/emergencias/diseases/novel-coronavirus-2019/advice-for-public/q-a-coronaviruses?gclid=Cj0KQCQiApsiBhCKARIsAN8o\\_4jPSK-qEFF7WdUgODr-n6oyaEioHe57P3wbKaSiUJ5zFi0WkLmoQ-sAvvfEALw\\_wcB](https://www.who.int/es/emergencias/diseases/novel-coronavirus-2019/advice-for-public/q-a-coronaviruses?gclid=Cj0KQCQiApsiBhCKARIsAN8o_4jPSK-qEFF7WdUgODr-n6oyaEioHe57P3wbKaSiUJ5zFi0WkLmoQ-sAvvfEALw_wcB) (accessed Feb. 22, 2021).
- [4] J. Meza, A. Estrada, C. Chabusa, and Vi. Velaszco, "Vista de Utilidad de Pruebas de cadena de polimerasa, pruebas rápidas y Tomografías en pacientes con Covid-19 | Journal of America health." p. 8, 2020, [Online]. Available: <http://jah-journal.com/index.php/jah/article/view/28/60>.
- [5] "Información sobre los distintos tipos de pruebas de COVID-19 | SCDHEC." 2019. <https://scdhec.gov/covid19-es/informacion-sobre-los-distintos-tipos-de-pruebas-de-covid-19> (accessed Feb. 22, 2021).
- [6] C. Rocha, "¿Cómo se puede Diagnosticar el COVID-19? ¿Cuál es la Precisión de los Test Diagnósticos?," *Sociedad Andaluza de Oftalmología*. 2020, Accessed: Feb. 22, 2021. [Online]. Available: <https://sociedadandaluzadeoftalmologia.es/como-se-puede-diagnosticar-el-covid-19-cual-es-la-precision-de-los-test-diagnosticos/>.
- [7] Coello Cristina and Nuñez Byron, "Pruebas de antígeno, la nueva herramienta en la detección de la COVID-19," 2020. <https://www.edicionmedica.ec/secciones/salud-publica/pruebas-de-antigeno-la-nueva-herramienta-en-la-deteccion-de-la-covid-19--96534> (accessed Feb. 22, 2021).
- [8] "Exploración por tomografía computarizada (TC) | Cancer.Net," 2018. <https://www.cancer.net/es/desplazarse-por-atención-del-cáncer/diagnóstico-de-cáncer/pruebas-y-procedimientos/exploración-por-tomografía-computarizada-tc> (accessed Feb. 22, 2021).
- [9] E. Press, "La tomografía de tórax es la mejor prueba para diagnosticar el coronavirus, según estudio," pp. 1–3, 2020, Accessed: Feb. 22, 2021. [Online]. Available: <https://www.infosalus.com/asistencia/noticia-tomografia-torax-mejor-prueba-diagnosticar-coronavirus-estudio-20200227171650.html>.
- [10] F. Tablado, "Inteligencia artificial: Definición, tipos y aplicaciones | Grupo Atico34," Jun. 18, 2020. <https://protecciondatos-lopdm.com/empresas/inteligencia-artificial/> (accessed Mar. 12, 2021).
- [11] M. Velazquez, "Reclut IT," Apr. 17, 2020. <https://reclut.com/diferencias-entre-el-analisis-predictivo-e-ia/#.YGBYJlVKh1v> (accessed Mar. 28, 2021).
- [12] "Qué es Machine Learning, cómo funciona y a qué se aplica | APD," Mar. 04, 2019. <https://www.apd.es/que-es-machine-learning/> (accessed Mar. 12, 2021).
- [13] "¿Qué es el Deep Learning? | SmartPanel." <https://www.smartpanel.com/que-es-deep-learning/> (accessed Mar. 12, 2021).
- [14] "Qué son regresión y clasificación en Machine Learning," Oct. 29, 2020. <https://agenciab12.com/noticia/que-son-regresion-clasificacion-machine-learning> (accessed Mar. 12, 2021).
- [15] Ó. De la fuente, "Python y machine learning en la nube," *Google Colab: Python y Machine Learning en la nube*, 2019.
- [16] J. M. Uriarte, "Google Drive," Nov. 2020. .
- [17] "¿Por qué elegir Keras?".
- [18] Covantec, "3.11. Tipo diccionarios — Materiales del entrenamiento de programación en Python - Nivel básico," 2018. <https://entrenamiento-python-basico.readthedocs.io/es/latest/leccion1/introduccion.html> (accessed Sep. 07, 2020).
- [19] O. Sanseviero, "Introducción a TensorFlow (Parte 1) | by Omar Sanseviero | AI Learners | Medium," 2019. <https://medium.com/ai-learners/introducción-a-tensorflow-parte-1-840c01881658> (accessed Sep. 09, 2020).
- [20] X. Xu *et al.*, "A Deep Learning System to Screen Novel Coronavirus Disease 2019 Pneumonia," *Engineering*, vol. 6, no. 10, pp. 1122–1129, Oct. 2020, doi: 10.1016/j.eng.2020.04.010.
- [21] P. Kumar Sethy, S. Kumari Behera, P. Kumar Ratha, and P. Biswas, "Detection of Coronavirus Disease (COVID-19) based on Deep Features and Support Vector Machine," *Int. J. Math. Eng. Manag. Sci.*, vol. 5, no. 4, pp. 643–651, 2020, doi: 10.33889/IJMEMS.2020.5.4.052.
- [22] A. Narin, C. Kaya, and Z. Pamuk, "Automatic Detection of Coronavirus Disease (COVID-19) Using X-ray Images and Deep Convolutional Neural Networks."
- [23] P. Plaza, "DESARROLLO DE UN MODELO PREDICTIVO DE ROBO A CASAS BASADO EN REDES NEURONALES," 2020. [http://repositorio.uchile.cl/bitstream/handle/2250/174256/cf-plaza\\_pa.pdf?sequence=1&isAllowed=y](http://repositorio.uchile.cl/bitstream/handle/2250/174256/cf-plaza_pa.pdf?sequence=1&isAllowed=y) (accessed Feb. 25, 2021).
- [24] "Seguridad del paciente: dosis de radiación en exámenes de rayos X y TC." .
- [25] "Radiografía de tórax (radiografía)," Jun. 2020. .
- [26] "Rayos X." .
- [27] G. Moreno, "Las redes neuronales: qué son y por qué están volviendo," Jan. 21, 2016. <https://www.xataka.com/robotica-e-ia/las-redes-neuronales-que-son-y-por-que-estan-volviendo> (accessed Mar. 15, 2021).
- [28] K. Delgado, S. Ledesma, and H. Rostro, "Análisis de electroencefalograma usando redes neuronales artificiales," *Acta Univ.*, vol. 29, pp. 1–24, Apr. 2019, doi: 10.15174/au.2019.1672.
- [29] R. Lopez, "Redes neuronales convolucionales con TensorFlow," Aug. 02, 2016. <https://relopezbriega.github.io/blog/2016/08/02/redes-neuronales-convolucionales-con-tensorflow/> (accessed Mar. 14, 2021).
- [30] C. Antona, "Herramientas modernas en redes neuronales: la librería keras [Tesis de grado, Universidad Autónoma de Madrid]," Repositorio Institucional , 2017.
- [31] J. Bagnato, "Convolutional Neural Networks: La Teoría explicada en Español | Aprende Machine Learning," Nov. 2018. .
- [32] A. Sharma, "Redes neuronales convolucionales en Python - DataCamp," Dec. 2017. .
- [33] D. Calvo, "Red Neuronal Convolucional CNN - Diego Calvo," Jul. 2017. .
- [34] "Funciones de activación de capa." .
- [35] P. Cornieles, "Entendiendo las redes neuronales: De la neurona a RNN, CNN y Deep Learning - IA Latam," Feb. 06, 2019. <https://ia-latam.com/2019/02/06/entendiendo-las-redes-neuronales-de-la-neurona-a-mn-cnn-y-deep-learning/> (accessed Mar. 15, 2021).
- [36] F. Lazzeri, "Aprendizaje profundo frente a aprendizaje automático - Azure Machine Learning | Microsoft Docs," Jan. 14, 2020. <https://docs.microsoft.com/es-es/azure/machine-learning/concept-deep-learning-vs-machine-learning> (accessed Mar. 15, 2021).
- [37] C. Tutivén, "Detector de COVID-19 usando imágenes de Rayos X | Saturdays.AI," Mar. 25, 2020. <https://medium.com/saturdays-ai/detector-de-covid-19-usando-imagenes-de-rayos-x-eea146efa26> (accessed Mar. 28, 2021).
- [38] P. Patel, "Chest X-ray (Covid-19 & Pneumonia) | Kaggle," Sep. 17, 2020. <https://www.kaggle.com/prashant268/chest-xray-covid-19-pneumonia> (accessed Mar. 28, 2021).
- [39] T. Rahman *et al.*, "COVID-19 Radiography Database | Kaggle," 2020. <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database> (accessed Mar. 28, 2021).
- [40] V. Rodríguez, "Dropout y Batch Normalization," 2018. <https://vincentblog.xyz/posts/dropout-y-batch-normalization> (accessed Mar. 03, 2021).
- [41] I. G. R. Gavilán, "Catálogo de componentes de redes neuronales (III): funciones de pérdida | Ignacio G.R. Gavilán," May 2020. .
- [42] "TensorFlow." <https://www.tensorflow.org/> (accessed Sep. 09, 2020).
- [43] P. Recuero de los Santos, "Cómo interpretar la matriz de confusión: ejemplo práctico - Think Big Empresas," Sep. 2020. .
- [44] J. Martinez, "Precision, Recall, F1, Accuracy en clasificación - IArtificial.net," Oct. 2020. .
- [45] "Deep Learning: Tres cosas que es necesario saber - MATLAB & Simulink." .