



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**APLICACIÓN DE VISIÓN POR COMPUTADORA PARA DETECCIÓN DE
ENFERMEDADES EN RADIOGRAFÍAS DE TÓRAX**

Franklin Estuardo Velásquez Fuentes

Asesorado por el Ing. Herman Igor Veliz Linares

Guatemala, noviembre de 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**APLICACIÓN DE VISIÓN POR COMPUTADORA PARA DETECCIÓN DE
ENFERMEDADES EN RADIOGRAFÍAS DE TÓRAX**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA

POR

FRANKLIN ESTUARDO VELÁSQUEZ FUENTES

ASESORADO POR EL ING. HERMAN IGOR VÉLIZ LINARES

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2021

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Kevin Vladimir Cruz Lorente
VOCAL V	Br. Fernando José Paz González
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Cesar Augusto Fernández Cáceres
EXAMINADOR	Ing. Cesar Rolando Batz Saquimux
EXAMINADOR	Ing. Sergio Arnaldo Méndez Aguilar
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

APLICACIÓN DE VISIÓN POR COMPUTADORA PARA DETECCIÓN DE ENFERMEDADES EN RADIOGRAFÍAS DE TÓRAX

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha 2 de junio de 2021.

A handwritten signature in black ink, enclosed within an oval shape. The signature is stylized and appears to read 'Franklin Estuardo Velásquez Fuentes'.

Franklin Estuardo Velásquez Fuentes

Guatemala, 08 de Octubre de 2021

Ingeniero
Carlos Alfredo Azurdia
Coordinador de Privados y Trabajos de Tesis
Escuela de Ingeniería en Ciencias y Sistemas
Facultad de Ingeniería - USAC

Respetable Ingeniero Azurdia:

Por este medio hago de su conocimiento que en mi rol de asesor del trabajo de investigación realizado por el estudiante **FRANKLIN ESTUARDO VELÁSQUEZ FUENTES** con carné 201602558 y CUI 2992 47945 0101 titulado "APLICACIÓN DE VISIÓN POR COMPUTADORA PARA DETECCIÓN DE ENFERMEDADES EN RADIOGRAFÍAS DE TÓRAX", lo he revisado y luego de corroborar que el mismo que cumple con los objetivos propuestos en el respectivo protocolo, procedo a la aprobación respectiva.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,


~~Ing. HERMAN IGOR VELIZ LINARES~~
Colegiado No. 4836

Herman Igor Veliz Linares
COLEGIADO No. 4836



Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala 18 de octubre de 2021

Ingeniero
Carlos Gustavo Alonzo
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Alonzo:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación del estudiante **FRANKLIN ESTUARDO VELÁSQUEZ FUENTES** con carné **201602558** y CUI **2992 47945 0101** titulado “**APLICACIÓN DE VISIÓN POR COMPUTADORA PARA DETECCIÓN DE ENFERMEDADES EN RADIOGRAFÍAS DE TÓRAX**” y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo aprobado.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,



Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA



FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN
CIENCIAS Y SISTEMAS

*El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación “**APLICACIÓN DE VISIÓN POR COMPUTADORA PARA DETECCIÓN DE ENFERMEDADES EN RADIOGRAFÍAS DE TÓRAX**”, realizado por el estudiante, **FRANKLIN ESTUARDO VELÁSQUEZ FUENTES** aprueba el presente trabajo y solicita la autorización del mismo.*

“ID Y ENSEÑAD A TODOS”

Msc. Carlos Gustavo Alonzo
Director
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 16 de noviembre de 2021



USAC
TRICENTENARIA
Universidad de San Carlos de Guatemala

Decanato
Facultad de Ingeniería
24189101- 24189102
secretariadecanato@ingenieria.usac.edu.gt

DTG. 670.2021

La Decana de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al Trabajo de Graduación titulado: **APLICACIÓN DE VISIÓN POR COMPUTADORA PARA DETECCIÓN DE ENFERMEDADES EN RADIOGRAFÍAS DE TÓRAX**, presentado por el estudiante universitario: **Franklin Estuardo Velásquez Fuentes**, y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Inga. ~~Anabela~~ Cordova Estrada
Decana



Guatemala, noviembre de 2021

AACE/cc

ACTO QUE DEDICO A:

Mi mamá	Aracely Maudaly Fuentes Gómez. Por todo su amor y apoyo incondicional durante toda mi vida y etapa educativa.
Mi papá	Rudy Fredy Velásquez Fuentes. Por todo su apoyo y palabras de aliento durante mi carrera universitaria.
Mi hermano	Fredy Armando Velásquez Fuentes. Por su hermandad y compañía.
Mis abuelos	Por todos los consejos, momentos felices y palabras de aliento.
Mi familia	Por todas las palabras de apoyo, preocupación por mi carrera y salud.
Mis amigos	Por todos los momentos compartidos juntos estudiando, los proyectos realizados y apoyo mutuo durante la carrera.

AGRADECIMIENTOS A:

**Universidad de San
Carlos de Guatemala**

Por ser la casa de estudios que me permitió desarrollar mi carrera profesional y madurar como persona.

Facultad de Ingeniería

Por darme la oportunidad de conocer a personas con intereses similares a los míos y permitirme la oportunidad de estudiar Ingeniería en Ciencias y Sistemas.

Mi asesor

Ingeniero Herman Veliz por ayudarme asesorarme a lo largo de este trabajo.

1.1.6.	Problemas comunes en los algoritmos de visión por computadora.....	13
1.1.6.1.	Underfitting.....	13
1.1.6.2.	Overfitting.....	14
1.1.6.3.	Cambio en la distribución de los datos.	14
1.2.	Detección de enfermedades utilizando visión por computadora	14
1.2.1.	Datasets públicos de radiografías de tórax.....	18
1.2.2.	Estudios realizados	19
1.2.3.	Problemas comunes	21
1.2.3.1.	Falta de datos	21
1.2.3.2.	Desequilibrio en las clases de datos	22
1.2.3.3.	Sobreajuste en tipos de imágenes o en datos locales	23
1.2.4.	Métricas de evaluación del modelo.....	24
1.2.4.1.	Accuracy	24
1.2.4.2.	Confusion Matrix	25
1.2.4.2.1.	Verdaderos positivos y negativos	26
1.2.4.2.2.	Falsos positivos	26
1.2.4.2.3.	Falsos negativos.....	27
1.2.4.3.	Sensitivity	27
1.2.4.4.	Specificity	27
1.2.4.5.	AUC - ROC Curve.....	28
1.2.5.	Beneficios y aplicaciones de la detección de enfermedades utilizando visión por computadora...	29
1.2.5.1.	Herramienta complementaria para realizar diagnósticos preliminares.....	30

1.2.5.2.	Aplicaciones didácticas	31
2.	DISEÑO DEL MODELO DE VISIÓN POR COMPUTADORA PARA ANÁLISIS DE RADIOGRAFÍAS DE TÓRAX	33
2.1.	Tecnologías utilizadas.....	33
2.1.1.	Python	33
2.1.2.	Kaggle.com	34
2.1.3.	Tensorflow y Keras.....	34
2.2.	Dataset de radiografías de tórax utilizado.....	35
2.2.1.	Análisis de balance de clases	36
2.3.	Desarrollo del modelo de visión por computadora	37
2.3.1.	Modelo base de la red utilizando transfer learning.....	38
2.3.2.	Modelo complementario de la red y capa de salida	39
2.4.	Entrenamiento del modelo	40
2.5.	Salidas del modelo	43
3.	DISEÑO DE LA APLICACIÓN WEB	47
3.1.	Vistas.....	47
3.2.	Arquitectura de la aplicación	52
3.3.	Tecnologías utilizadas.....	53
3.3.1.	React JS.....	53
3.3.2.	Amazon Web Services	54
3.3.2.1.	DynamoDB	54
3.3.2.2.	S3	55
3.3.2.3.	Cognito	55
3.3.2.4.	IAM.....	55
3.3.2.5.	Digital Ocean Droplets	55

4.	RESULTADOS DE LA EVALUACIÓN DEL MODELO	57
4.1.	Confusion Matrix	57
4.1.1.	Verdaderos Negativos.....	60
4.1.2.	Verdaderos Positivos	61
4.1.3.	Falsos Positivos	61
4.1.4.	Falsos Negativos.....	62
4.2.	Accuracy	62
4.3.	ROC Curve.....	63
	CONCLUSIONES.....	65
	RECOMENDACIONES	67
	BIBLIOGRAFÍA.....	69
	APÉNDICES	71

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Ejemplo de proceso de entrenamiento de un modelo.....	3
2.	Proceso de evaluación del modelo	5
3.	Arquitectura de una red neuronal.....	6
4.	Funcionamiento de una neurona	7
5.	Comparación de modelos de Aprendizaje automático.....	8
6.	Vista de descenso del gradiente en 3 dimensiones.....	10
7.	Arquitectura de una red neuronal convolucional.....	12
8.	Ejemplo de capa de convolución	12
9.	Modelo de clasificación de imágenes en Histopatología	16
10.	Modelo de análisis de resonancias magnéticas en cerebros.....	17
11.	Modelo de clasificación de enfermedades en tórax.....	18
12.	Rendimiento en función de la cantidad de datos	22
13.	Ejemplo de <i>Confusion Matrix</i> para tres clases.....	25
14.	AUC – ROC Curve.....	28
15.	Ejemplos de radiografías utilizadas para el entrenamiento	35
16.	Algoritmo para crear un histograma desarrollado en <i>Python</i>	36
17.	Gráfica de distribución de clases en el conjunto de datos.....	36
18.	Arquitectura de red neuronal <i>DenseNet121</i>	38
19.	Código escrito en <i>Python</i> para ensamblaje de modelo base y modelo complementario.....	39
20.	Funciones escritas en <i>Python</i> para soportar el desequilibrio en la distribución de los datos durante el entrenamiento.....	40
21.	Función escrita en <i>Python</i> para crear un conjunto de entrenamiento...	41

22.	Muestra de imágenes etiquetadas en el conjunto de datos	42
23.	Proceso de entrenamiento del modelo	42
24.	Arquitectura del modelo entrenado.....	43
25.	Salida con probabilidades para cada tipo de enfermedad.....	44
26.	Mapas de calor creados en <i>Python</i> como salida del modelo	45
27.	Página de ingreso a la plataforma	47
28.	Página para el registro de nuevo usuario	48
29.	Vista para confirmación de código de seguridad	49
30.	Vista para realizar análisis de una nueva radiografía	50
31.	Fase de envío de imagen al servidor y espera de resultados del análisis	51
32.	Página con los resultados del análisis	51
33.	Diagrama de arquitectura de la aplicación	52
34.	Creación de un conjunto de evaluación en <i>Python</i>	57
35.	Programa escrito en <i>Python</i> para obtener las predicciones en imágenes nuevas para calcular y graficar la <i>confusion matrix</i>	58
36.	Valores de <i>confusion matrix</i> obtenidos en el conjunto de evaluación para cada una de las enfermedades	59
37.	Métricas de evaluación del modelo calculadas en <i>Python</i> para cada tipo de enfermedad	60
38.	Cálculo de la métrica de <i>Accuracy</i> del modelo en <i>Python</i>	62
39.	<i>ROC Curve</i> del modelo creado para analizar radiografías de tórax en el conjunto de evaluación	63

LISTA DE SÍMBOLOS

Símbolo	Significado
AWS	<i>Amazon Web Services</i> . Empresa que provee servicios de computación en la nube.
AUC	Área bajo la curva.
API	Conjunto de funciones, rutinas y procedimientos que se publican en forma de interfaz para su uso.
ROC	Representación gráfica de la sensibilidad de un sistema al variar los valores de umbral.
IAM	Servicio de identificación y acceso de usuarios proveído por AWS.

GLOSARIO

<i>Backend</i>	Tecnología encargada de correr del lado del servidor, además es el encargado de manejar la lógica de la aplicación.
<i>Cloud Computing</i>	Se refiere al conjunto de servicios que una organización provee a sus clientes y que les permite ejecutar programas en servidores remotos, así como acceder a una diversa gama de servicios computacionales a través de internet.
<i>Deep Learning</i>	Área de inteligencia artificial especializada en redes neuronales y aprendizaje de sistemas informáticos para resolver problemas complejos.
<i>Framework</i>	Es un entorno de trabajo definida con módulos de software que sirve como base de desarrollo de software.
<i>Fronted</i>	Toda tecnología que corre del lado del cliente, es el encargado de lo que el cliente puede ver.
<i>Machine Learning</i>	Campo de las ciencias de la computación que se encarga de estudiar el aprendizaje de sistemas informáticos para resolver problemas.

ROC

Característica operativa del receptor.

RESUMEN

Se implementó un modelo de inteligencia artificial capaz de analizar imágenes de radiografías de tórax y clasificarlas según su probabilidad en catorce tipos de enfermedades. Este modelo fue previamente entrenado con un gran conjunto de imágenes de radiografías de tórax dispuestas al público por distintas instituciones de salud americanas y promovido por la universidad de Stanford en un concurso público de inteligencia artificial, así como en recursos académicos *online*.

Se desarrolló una aplicación utilizando el *framework React* para crear una página web que pueda ser accedida desde cualquier dispositivo y que sirva de interfaz gráfica para que usuarios puedan interactuar con el modelo de inteligencia artificial subiendo imágenes de radiografías de tórax, enviándolas al servidor y posteriormente recibir los resultados del análisis de la imagen en forma de una tabla, con las probabilidades para cada tipo de enfermedad y un mapa de calor indicando las zonas de la imagen, en la cual el modelo de inteligencia artificial encontró factores determinantes para el análisis.

Se utilizaron servicios de computación en la nube de las empresas *Amazon Web Services* y *Digital Ocean* para desplegar el modelo de inteligencia artificial junto a una aplicación desarrollada en el lenguaje de programación *Python* que permite recibir peticiones de internet y que de esta manera el modelo de análisis de radiografías de tórax pueda ser accedido a través de la aplicación web construida para los usuarios.

OBJETIVOS

General

Desarrollar una aplicación de visión por computadora para detección de enfermedades en radiografías de tórax.

Específicos

1. Implementar un modelo de visión por computadora utilizando redes neuronales convolucionales.
2. Entrenar el modelo de visión por computadora para detectar enfermedades en radiografías de tórax en base a conjuntos de datos públicos de radiografías.
3. Desarrollar una aplicación web que provea una interfaz gráfica de usuario que permita enviar imágenes de radiografías y obtener los resultados del análisis hecho por el modelo.
4. Crear una herramienta para recopilar nuevos datos de radiografías de tórax etiquetadas por humanos con la finalidad de obtener datos locales para mejorar progresivamente el rendimiento del modelo.

INTRODUCCIÓN

Los algoritmos de inteligencia artificial basados en aprendizaje automático y redes neuronales empezaron a desarrollarse desde mediados del siglo XX, sin embargo, en aquella época el poder de procesamiento y memoria de las computadoras era demasiado limitado para aplicar dichos modelos a aplicaciones reales. En un inicio las computadoras y sistemas informáticos eran vistos simplemente como herramientas para realizar una gran cantidad de cálculos matemáticos de forma repetitiva.

Hoy en día, con el aumento del poder de cómputo y el mayor acceso a computación en la nube, es posible desarrollar sistemas informáticos que utilicen inteligencia artificial para resolver tareas más complejas, incluso es posible desarrollar sistemas con capacidades de reconocimiento de texto, voz e imágenes, las cuales se atribuían como capacidades únicamente humanas y que actualmente se han incorporado de forma natural a muchos dispositivos electrónicos que se utilizan en la vida cotidiana.

Existe un consenso bastante grande en la industria tecnológica con respecto a que la aplicación de inteligencia artificial en actividades humanas está siendo y será uno de los puntos más importantes en innovación tecnológica. Cada año se realizan proyectos y aplican modelos de análisis de datos e inteligencia artificial a nuevas áreas del conocimiento que van desde la agricultura y ganadería hasta el estudio del cosmos en astronomía. En definitiva, en todas las áreas del conocimiento, es necesaria la aplicación de tecnología que permita al ser humano facilitar las tareas complejas y repetitivas,

para que, de esta manera, los investigadores puedan invertir mayor tiempo y concentración en las tareas y detalles más importantes de sus investigaciones

Una de las áreas más sensibles y donde se requieren nuevas tecnologías para mejorar la atención y llegar a más personas es el campo de los estudios médicos y biológicos. Existe una gran demanda de médicos y equipos especializados para atender la gran gama de enfermedades que afecta a toda la población global, sin mencionar los casos extremos como emergencias sanitarias que pueden llevar al colapso de la capacidad hospitalaria en un país.

Actualmente, existen diversos proyectos que pretenden integrar tecnologías de inteligencia artificial en el campo de la medicina, algunos de estos ejemplos son modelos para reconocimiento de cáncer o enfermedades de la piel en imágenes de biopsias, detección de puntos de ceguera en imágenes de estudios en retinas y detección de tumores en cerebros, analizando estudios de resonancia magnética en 3D. Muchos de estos modelos presentan altos niveles de precisión en las distintas métricas de evaluación de modelos de inteligencia artificial, en algunos casos incluso superando la tasa de precisión de expertos.

Es importante resaltar que en el campo de la medicina es necesario tener mucha responsabilidad y cuidado, puesto que se está a cargo de vidas humanas. Debido a ello es normal que siempre se exija que toda nueva tecnología primero pase por un largo proceso de pruebas, análisis y validación para poder ser implementada a gran escala y actualmente los expertos en el campo se encuentran trabajando en dicho proceso para poder aplicar estas tecnologías de forma masiva. En el presente trabajo se expone el caso particular de un modelo de reconocimiento de imágenes especializado en la clasificación de enfermedades en radiografías de tórax, se ha tomado como

referencia y se ha implementado una arquitectura de red neuronal y un conjunto extenso de imágenes de radiografías de tórax publicadas en un concurso público de inteligencia artificial de la universidad de Stanford. También se ha construido una aplicación web que permite a los usuarios interactuar con el modelo.

1. MARCO TEÓRICO

A continuación, se exponen los conceptos principales y bases teóricas sobre los cuales se creó el modelo de inteligencia artificial de clasificación de imágenes de tórax.

1.1. Inteligencia artificial

El término inteligencia artificial hace referencia a la rama de las ciencias de la computación encargada del estudio de algoritmos que tratan de imitar el razonamiento y aprendizaje humano. En los comienzos de la informática, las computadoras tenían como principal objetivo el realizar cálculos principalmente numéricos de forma rígida y determinista, siendo una versión más potente de las calculadoras convencionales. Con el avance tecnológico y la aplicación de las computadoras en muchas más áreas de estudio, fue necesario crear algoritmos para resolver tareas cada vez más complejas e imitar la forma en la que los humanos aprenden y toman decisiones para resolver problemas.

1.1.1. Machine Learning

Dentro del campo de la inteligencia artificial se encuentra el área del *Machine Learning*, que se encarga del estudio y desarrollo de los algoritmos que permiten a las computadoras aprender automáticamente por medio de la experiencia o a través de grandes cantidades de datos, es por ello que al traducir al castellano se prefiere el término “Aprendizaje Automático”.

Este tipo de algoritmos ha tomado gran relevancia en los últimos años en una gran cantidad de áreas de estudio e industrias; esto debido a la inmensa cantidad de datos e información que se genera diariamente, la mejora en la capacidad de procesamiento de las computadoras, y la aplicación de sistemas inteligentes para resolver multitud de problemas y automatizar tareas.

1.1.2. Aprendizaje supervisado

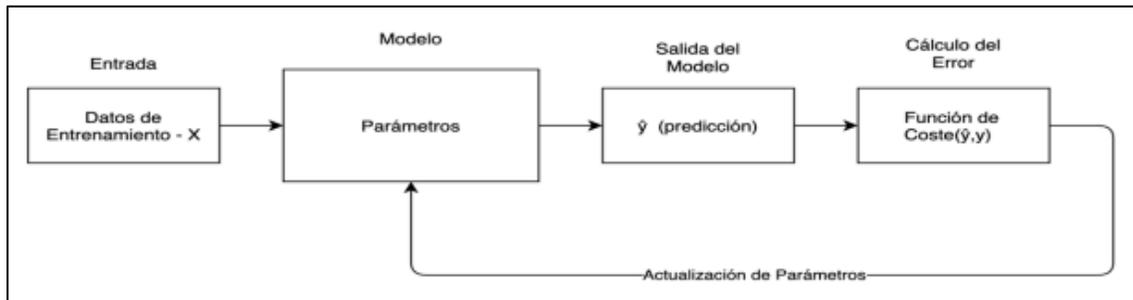
Se suele clasificar a los algoritmos de *Machine Learning* según su forma de aprender a partir de los datos, dando como resultado las categorías de algoritmos de aprendizaje supervisado y no supervisado.

Los algoritmos de aprendizaje supervisado se caracterizan por contar con datos etiquetados previamente por humanos, es decir se conoce el valor real. A partir de esto se implementa un modelo que aprenda a partir de los datos con el objetivo de posteriormente utilizarlo para calcular predicciones sobre datos nunca antes vistos por el modelo y que estas predicciones se acerquen lo más posible al valor real.

1.1.2.1. Proceso de aprendizaje

Al proceso de aprendizaje en un algoritmo de *Machine Learning* se le denomina “entrenamiento” y el resultado de este es un modelo con parámetros optimizados para realizar predicciones sobre nuevos datos; esto puede ser visto como un proceso iterativo de optimización de parámetros.

Figura 1. **Ejemplo de proceso de entrenamiento de un modelo**



Fuente: elaboración propia, empleando Draw.io.

El proceso de entrenamiento puede variar dependiendo del modelo o algoritmo escogido, pero en general consta de los siguientes pasos.

- División de los datos: se toma el conjunto de datos que se posee, se ordenan de forma aleatoria y se dividen en dos, los datos de entrenamiento y los datos de prueba. Estos son llamados usualmente por sus respectivos términos en inglés: *training set* y *test set*.
- Entrada del modelo: se ingresan los datos de entrenamiento al modelo. Se pueden visualizar en forma de matriz, en donde las columnas son todas aquellas variables que se desean utilizar para realizar predicciones (también llamadas variables independientes o *features*) y las filas son todos los valores de ejemplos que se tienen con estas variables.
- Modelo: se define como una forma de representar la realidad, en este caso hace referencia a un modelo matemático que intenta representar la forma en que se comportan los datos. Consta de variables o parámetros

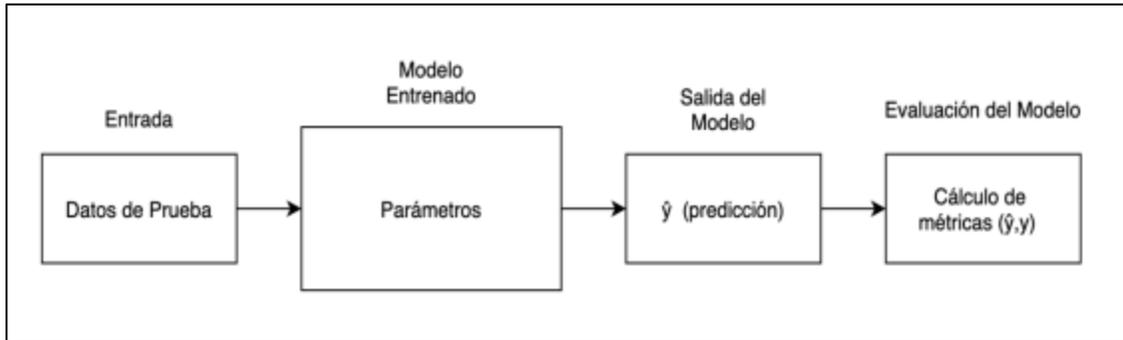
cuyos valores deben optimizarse durante el entrenamiento, con el objetivo de aprender a realizar predicciones.

- Salida del modelo: son las predicciones realizadas por el modelo, estas se obtienen al operar matemáticamente los valores de los parámetros del modelo con los valores de entrada. Por lo general, al inicio del aprendizaje estas predicciones suelen ser incorrectas, pero durante el entrenamiento se van ajustando los parámetros del modelo para mejorar progresivamente.
- Función de coste: es una clave esencial del entrenamiento, esta función compara las predicciones hechas por el modelo con el valor real de la variable objetivo (también llamada dependiente) para medir el error actual del modelo. Esta función varía dependiendo del tipo de predicción que se desea obtener o del problema que se desea resolver.
- Actualización de parámetros: a partir del valor del error obtenido en la función de coste, se actualizan los valores de los parámetros con el objetivo de minimizar el error. Este es un proceso iterativo de optimización que busca encontrar los parámetros que generen las mejores predicciones.

1.1.2.2. Evaluación básica del modelo

Una vez finalizado el proceso de entrenamiento, es necesario evaluar si el modelo resultante es capaz de dar predicciones correctas sobre nuevos datos. Es por ello que se utilizan los datos de prueba que no fueron utilizados durante el entrenamiento para calcular las métricas de precisión del modelo.

Figura 2. **Proceso de evaluación del modelo**



Fuente: elaboración propia, empleando Draw.io.

La métrica más utilizada es la precisión o *accuracy* en inglés, esta se calcula dividiendo el número de predicciones correctas dentro del número de ejemplos ingresados en los datos de prueba, usualmente se representa en forma de porcentaje para indicar la precisión del modelo. Existen diversos tipos de métricas que tratan de evaluar aspectos particulares del modelo, dependiendo del problema que se esté tratando y su contexto.

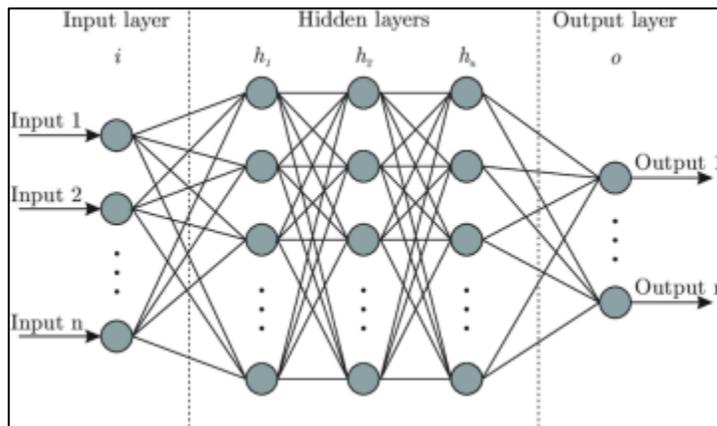
1.1.3. **Redes neuronales**

Las redes neuronales son un modelo de aprendizaje automático que se caracteriza por su potencial en la solución de tareas complejas que imitan el aprendizaje o comportamiento humano. Su nombre se debe a la arquitectura del modelo en forma de red conformado por unidades llamadas neuronas, que se inspira en la forma del cerebro humano, aunque no llega a representar el funcionamiento real del mismo que aún no es comprendido completamente por la neurociencia y que presenta una mayor complejidad.

1.1.3.1. Arquitectura

La unidad básica de la red son las neuronas, que al igual que en otros modelos de aprendizaje automático consta de una serie de parámetros que al ser operados con los datos de entrada producen una salida o señal. Estas se agrupan formando capas, las cuales a su vez se apilan para formar redes profundas.

Figura 3. **Arquitectura de una red neuronal**



Fuente: Artificial Intelligence in Plain English. *How Neuronal Networks process data.*

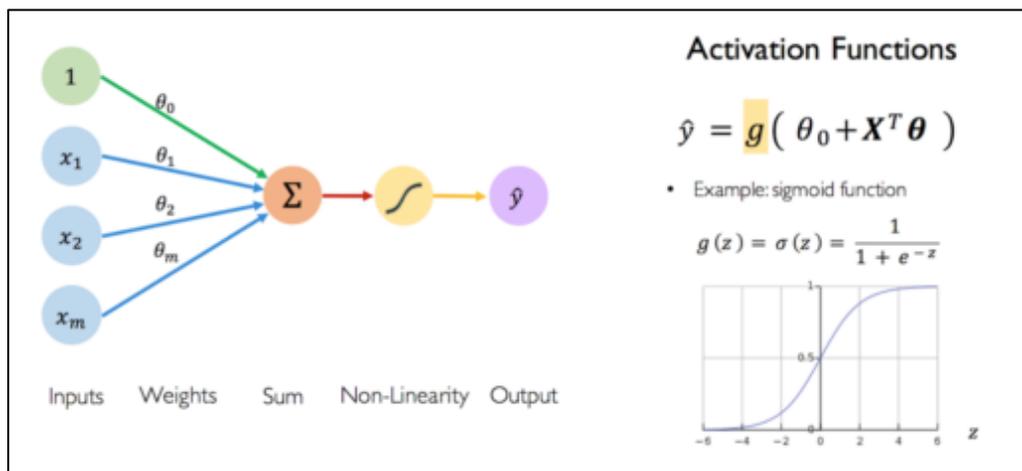
https://miro.medium.com/proxy/1*ub-ifcgdi9xgryqvo0_GRA.png.

Consulta: 22 de febrero de 2021.

Dentro de una neurona la operación básica que se realiza es similar a un modelo de regresión lineal, en donde los parámetros a ser optimizados son los coeficientes o pesos usualmente denotados con las letras W ó θ y los valores operados son las variables de entrada o señales de capas anteriores denotados por la letra X . Posteriormente se aplica una función de activación al resultado que permite capturar comportamientos no lineales dentro de los datos; entre las

funciones de activación más comunes se encuentran: la función sigmoide, ReLU y tangente inversa. El resultado final de la operación se propaga a las capas posteriores en la red.

Figura 4. **Funcionamiento de una neurona**



Fuente: MIT Intro to Deep learning. *Introduction to D.L.*

http://introtodeeplearning.com/slides/6S191_MIT_DeepLearning_L1.pdf.

Consulta: 23 de febrero de 2021.

Una sola neurona no puede capturar comportamientos complejos en los datos, sin embargo, al crear redes neuronales densas y profundas es posible capturar patrones cada vez más complejos, esta es una de las características que provee a las redes neuronales un gran potencial para resolver tareas que otros modelos de aprendizaje automático no resuelven de forma eficiente.

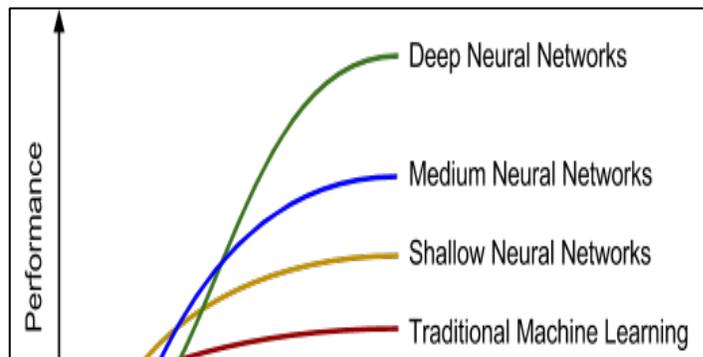
Al área del *Machine Learning* que estudia y utiliza modelos de redes neuronales para resolver problemas se le denomina *Deep Learning* o “Aprendizaje Profundo”. Entre las aplicaciones comunes que se han dado a las redes neuronales se encuentran:

- Detección de objetos en imágenes.
- Reconocimiento de rostros.
- Procesamiento de lenguaje natural.
- Reconocimiento de sonidos, como el habla humana y su transcripción a texto.
- Generación de imágenes artificiales.

1.1.3.2. Comparación con otros modelos de Machine Learning

Se han realizado estudios comparando el rendimiento de los modelos que utilizan redes neuronales con otros modelos clásicos de *Machine Learning*. Para ello se toma en cuenta la cantidad de datos disponibles para el entrenamiento.

Figura 5. **Comparación de modelos de Aprendizaje automático**



Fuente: ALBAHNSSEN, Alejandro. *Building AI Applications Using Deep Learning*.

<https://albahnsen.files.wordpress.com/2018/05/11.png>.

Consulta: 23 de febrero de 2021.

Como es posible apreciar en la figura 5, con forme aumenta la cantidad de datos disponibles para el entrenamiento, los modelos tradicionales de aprendizaje automático no presentan una mejora significativa en su aprendizaje. Por otra parte, los modelos de redes neuronales pueden aumentar su rendimiento al existir una mayor cantidad de datos y al aumentar la profundidad de la red, es decir, al agregar más capas y parámetros a entrenar.

1.1.3.3. Forward Propagation

La propagación hacia adelante, o *Forward Propagation*, es el primer paso en el entrenamiento de una red neuronal, durante esta fase se realizan las operaciones de los valores de entrada del modelo con los parámetros de las neuronas, se calculan los valores de salida y se pasan a las redes posteriores, propagándose de esta manera por toda la red. En la última capa de la red se obtienen las predicciones del modelo.

1.1.3.4. Función de costo

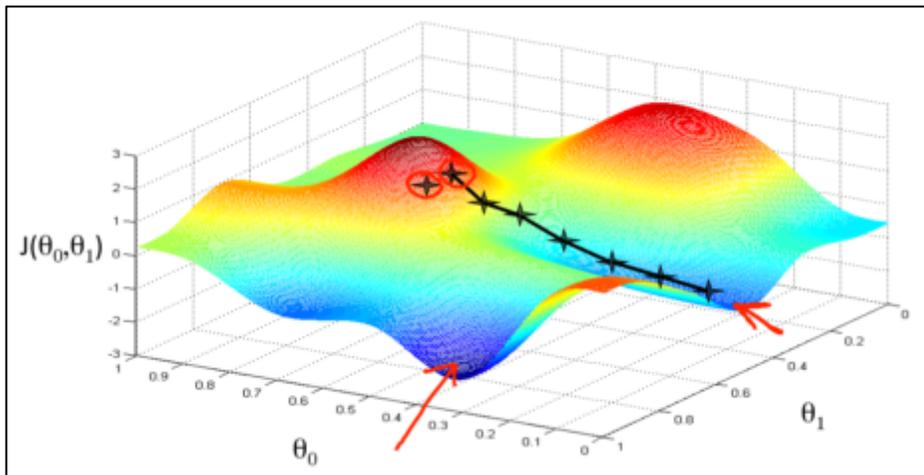
Utilizando la función de costo se calcula el error o diferencia existente entre las predicciones dadas por el modelo y los valores reales según los datos de entrenamiento. Esto se utiliza para evaluar cómo deben ser actualizados los parámetros dentro de las neuronas de la red para mejorar las predicciones y aprender a representar el comportamiento de los datos.

1.1.3.5. Descenso del gradiente

Es el algoritmo de optimización utilizado para encontrar los valores en los parámetros que minimicen la función de costo. Al existir miles o millones de parámetros a optimizar, encontrar una solución matemática analítica puede ser

muy complicado y el coste computacional para calcularla sería muy alto, es por ello que el descenso del gradiente es un método numérico que permite aproximar los mejores valores para los parámetros. Debe su nombre a su representación gráfica, que es la de un vector descendiendo por la gráfica de la función de coste utilizando los gradientes o derivadas en cada paso para intentar hallar el punto mínimo.

Figura 6. **Vista de descenso del gradiente en 3 dimensiones**



Fuente: Medium, Gradient Descent. *All You Need to Know*.

https://miro.medium.com/proxy/1*f9a162GhpMbiTVTAua_ILQ.png.

Consulta: 2 de febrero de 2021.

1.1.3.6. **Back Propagation**

En la fase de propagación hacia atrás o *Back Propagation* se actualizan los valores en los parámetros de la red. Debe su nombre a que los cálculos se realizan en sentido inverso; se inicia calculando la función de coste en las capas de salida y se va retrocediendo en la red para ir actualizando las capas de atrás

hacia adelante, calculando las derivadas de cada parámetro de acuerdo con la función de coste y actualizando su valor.

1.1.4. Transferencia de aprendizaje

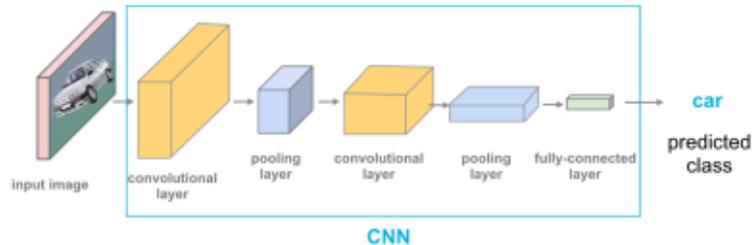
La transferencia de aprendizaje consiste en tomar los valores de los parámetros ya entrenados de un modelo y transferirlos a otro modelo para resolver una nueva tarea; posteriormente, se pueden agregar más capas con neuronas al nuevo modelo para obtener mejores predicciones en nuevos datos. Esto permite reutilizar modelos y tomar ventaja de su aprendizaje, ahorrando de esta manera tiempo y recursos computacionales.

1.1.5. Visión por computadora y redes neuronales convolucionales

Se conoce como visión por computadora al campo de la inteligencia artificial que procesa y analiza imágenes de forma automática utilizando algoritmos con el objetivo de extraer información útil, detectar objetos o interpretarlas como lo haría un ser humano.

Las redes neuronales convolucionales son un conjunto de modelos de *Deep Learning* que se caracterizan por contar con capas de “convolución” en su arquitectura, además de especializarse en problemas de visión por computadora.

Figura 7. **Arquitectura de una red neuronal convolucional**



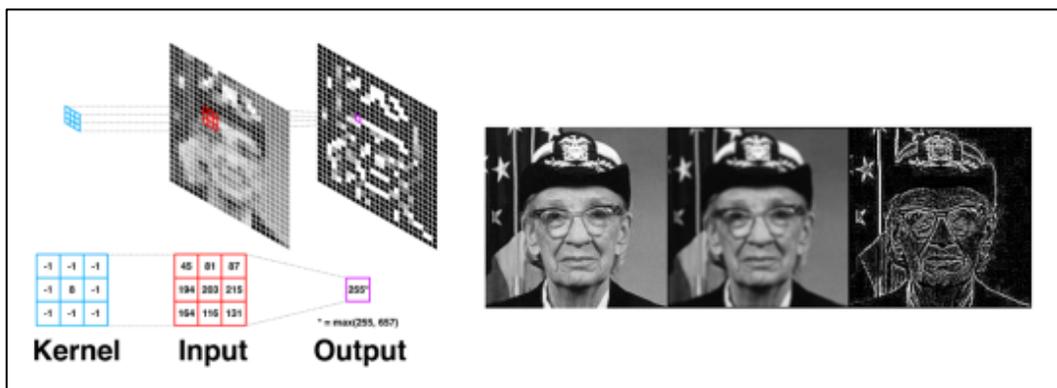
Fuente: CAMACHO, Cezanne. *Convolutional Neural Networks*.

https://cezannec.github.io/assets/cnn_intro/CNN_ex.png.

Consulta: 05 de marzo de 2021.

Una capa de convolución es una serie de matrices con valores numéricos, también llamadas *Kernel*, que se operan con los valores de los píxeles en los canales de color de una imagen. Esta operación puede ser interpretada como una serie de filtros que reconocen algún tipo de característica en una imagen, como líneas verticales, líneas horizontales o esquinas.

Figura 8. **Ejemplo de capa de convolución**



Fuente: GUNDERSEN, Gregory. *From Convolution to Neural Network*.

<http://gregorygundersen.com/image/cnns/41847277995.png>. Consulta: 05 de marzo de 2021.

Al interpretar las operaciones que se realizan en las capas de la red neuronal, es posible observar que en las capas de inicio se reconocen características simples en la imagen, mientras que en las capas profundas se reconocen características más complejas.

Los valores numéricos en las matrices, es decir los filtros a aplicar, se encuentran mediante el entrenamiento del modelo, de tal manera que se optimizan para obtener las mejores predicciones al analizar una imagen.

1.1.6. Problemas comunes en los algoritmos de visión por computadora

Para medir la exactitud con la cual se están reconociendo las imágenes y la calidad de los resultados, se establecen métricas de evaluación para los modelos de aprendizaje automático. En el caso de los modelos de visión por computadora se analizan los resultados que se obtienen al analizar imágenes nunca vistas por los modelos en la fase de entrenamiento, a partir de ahí se han podido observar problemas que normalmente se presentan al crear y evaluar un modelo.

1.1.6.1. Underfitting

El problema del *underfitting* o ajuste pobre, se presenta cuando un modelo tiene un porcentaje de precisión bajo, tanto en los datos de entrenamiento como en los de prueba. Dependiendo del contexto del problema a resolver se establece un valor inferior de precisión a partir del cual se puede considerar que un modelo produce un 60 % de predicciones correctas en los datos de entrenamiento.

1.1.6.2. Overfitting

Se dice que un modelo presenta *overfitting* o sobre ajuste, cuando se obtiene un porcentaje de precisión muy alto al dar predicciones en los datos de entrenamiento, pero tiene una caída en la precisión al intentar analizar datos que no había visto antes, es decir con los datos de prueba o en producción.

En general, se desea crear un modelo que aprenda a generalizar y a dar predicciones correctas sobre nuevos datos.

1.1.6.3. Cambio en la distribución de los datos

En el caso del área de visión por computadora, suele suceder que el problema del sobreajuste se deba a que el modelo se acostumbra a analizar solamente imágenes de un tipo de distribución y tiene problemas al intentar analizar imágenes que proceden de unas fuentes diferentes.

Algunos de los factores que pueden afectar en el reconocimiento de imágenes son los cambios entre los datos de entrenamiento y de prueba con respecto a variables como la resolución de la imagen, escala de color, rotación, tamaño, elementos externos o ruido.

1.2. Detección de enfermedades utilizando visión por computadora

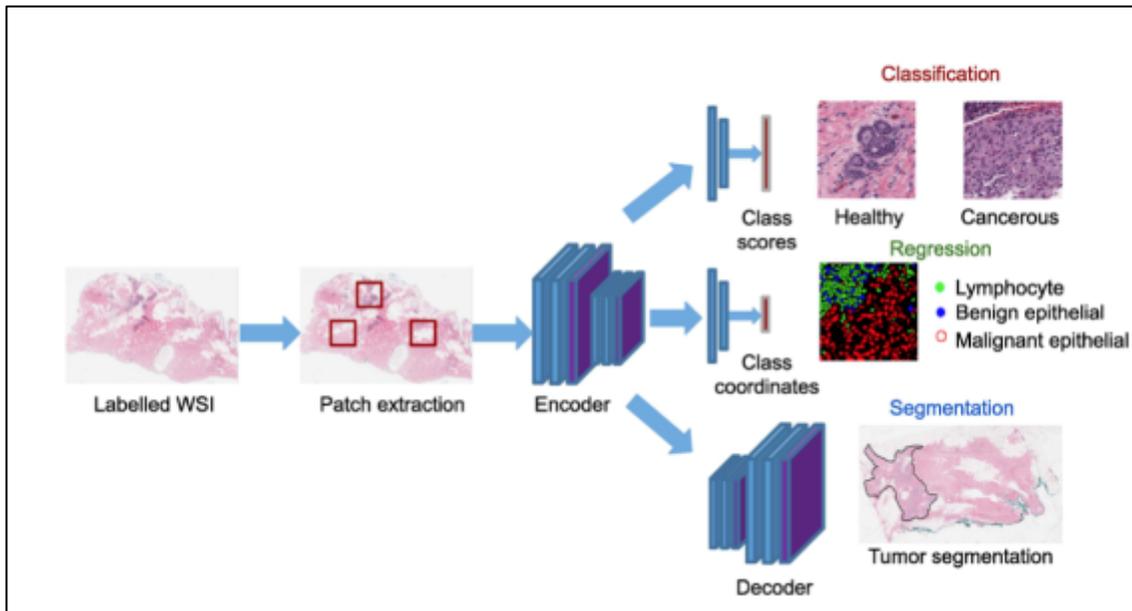
Con el avance tecnológico y de la capacidad de cómputo, ha sido posible la creación de modelos de visión por computadora y algoritmos de inteligencia artificial cada vez más complejos y que se pueden aplicar a muchas áreas del conocimiento. Una de estas áreas ha sido la medicina, en donde se han aplicado una gran variedad de tipos de modelos de inteligencia artificial en las

diversas fases del proceso de atención, diagnóstico, tratamiento y monitorización de los pacientes, así como en el análisis y detección de enfermedades.

Una de las fases comunes en el tratamiento de un paciente es realizar estudios que ayuden a diagnosticar la posible enfermedad o condición que una persona presenta. Existe una gran variedad de estudios que se pueden realizar, en este caso se prestará atención a los exámenes que tienen como resultado una imagen que representa la condición de un miembro o parte del cuerpo, debido a que estas pueden ser analizadas por algoritmos de visión por computadora. A continuación, se presentan algunos ejemplos.

- Detección de cáncer en imágenes de tejidos: Se han desarrollado modelos de visión por computadora que tienen como entrada imágenes de estudios realizados en tejidos de órganos del cuerpo humano y que se encargan de detectar dentro de las imágenes regiones con posibles anomalías o que estén siendo afectadas por alguna enfermedad. Según el Instituto Nacional del Cáncer de Estados Unidos, se define como Histopatología al estudio de las células y el tejido enfermo bajo un microscopio.

Figura 9. **Modelo de clasificación de imágenes en Histopatología**

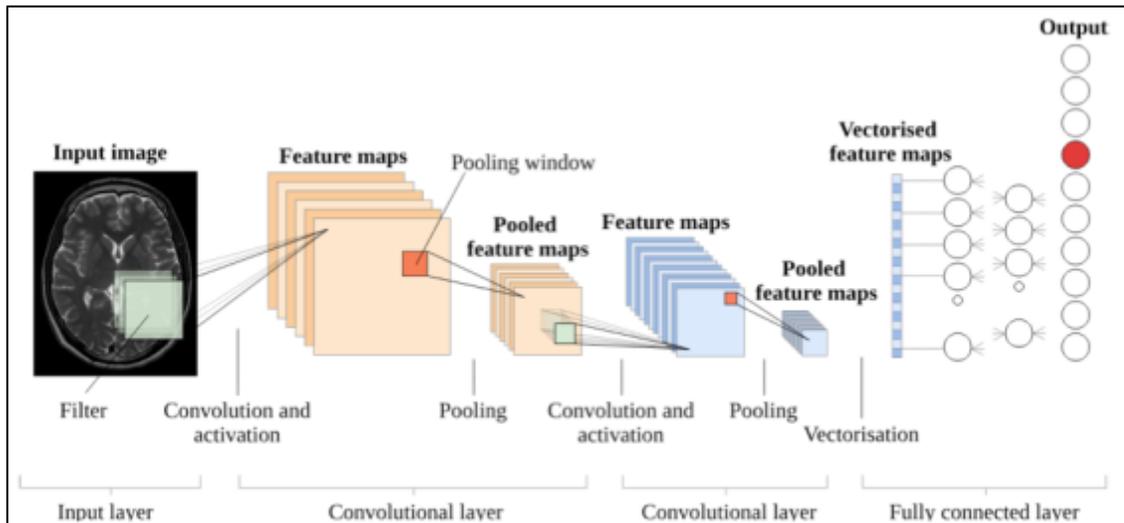


Fuente: Department of Medical Biophysics, University of Toronto, Canada. *Deep neural network models for computational histopathology*. <https://arxiv.org/pdf/1912.12378.pdf>.

Consulta: 06 de marzo de 2021.

- Análisis cerebral en resonancias magnéticas: Los estudios de resonancia magnética crean una representación tridimensional del cerebro de una persona utilizando radiación electromagnética. Los modelos especializados de visión por computadora pueden analizar esta representación para intentar identificar enfermedades y clasificar estudios.

Figura 10. **Modelo de análisis de resonancias magnéticas en cerebros**

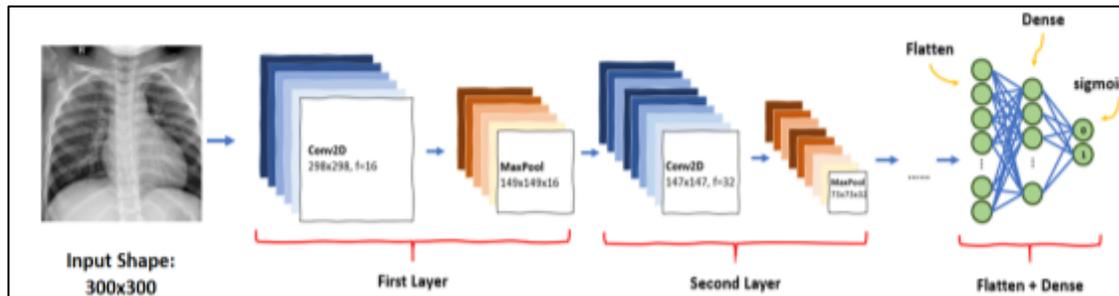


Fuente: SELVIKVAG LUNDERVOLD, Alexander. *Haukeland University Hospital. An overview of deep learning in medical imaging focusing on MRI.* <https://arxiv.org/pdf/1811.10052.pdf>.

Consulta: 6 de marzo de 2021.

- Clasificación de enfermedades en radiografías de tórax: Las radiografías de tórax son uno de los estudios más comunes que se utilizan para realizar diagnósticos a enfermedades relacionadas con el pulmón y sistema respiratorio. Se han aplicado modelos de redes neuronales para realizar análisis sobre estas imágenes de forma automática entrenando modelos en base a diagnósticos hechos previamente por profesionales.

Figura 11. **Modelo de clasificación de enfermedades en tórax**



Fuente: ARCHIE, Christie. *Chest X-rays Pneumonia Detection using CNN*.

https://miro.medium.com/max/700/1*caVi5_pTsarvYlqkarijOg.png

Consultado el 8 de marzo de 2021.

1.2.1. **Datasets públicos de radiografías de tórax**

Diversas instituciones han buscado recopilar grandes conjuntos de datos con fotografías de radiografías de tórax y los diagnósticos dados sobre ellas por profesionales como médicos especialistas y radiólogos. Todo esto con el objetivo de entrenar modelos capaces de analizar radiografías de tórax para diagnóstico de enfermedades. Muchos de estos conjuntos de datos o *Datasets* se han licenciado para uso público con el objetivo de fomentar el desarrollo de modelos de inteligencia artificial a nivel universitario y profesional.

- “*NIH Chest X-Ray Dataset*: Conjunto de datos proporcionado por el Instituto Nacional de Salud de Estados Unidos, o NIH por sus siglas en inglés. En este conjunto se recopila 112 120 fotografías de radiografías de tórax hechas a 30 805 pacientes. Para obtener las etiquetas de las

enfermedades asignadas a cada radiografía, se utilizó un modelo de inteligencia artificial para procesamiento de lenguaje natural.”¹

- *CheXpert*: Este conjunto de datos fue recopilado por la Universidad de Stanford, principalmente de su hospital universitario. Las imágenes de radiografías y sus etiquetas correspondientes respecto a las enfermedades presentadas fueron tomadas por profesionales de la salud entre 2002 y 2019.

1.2.2. Estudios realizados

Diversas instituciones educativas como universidades e institutos técnicos, así como entidades especializadas en salud como hospitales y laboratorios de investigación han realizado proyectos de desarrollo de modelos de visión por computadora para detección de enfermedades en radiografías de tórax. Se han utilizado conjuntos de datos tanto públicos como privados y en su desarrollo se han visto involucrados tanto personal del área de informática y ciencias de la computación como profesionales de la salud para obtener ideas y conocimiento para desarrollar los modelos de inteligencia artificial. A continuación, se detallan algunos casos.

- *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*: “Este modelo fue desarrollado por un equipo de profesores y docentes tanto en el área de ciencias de la computación como en salud de la Universidad de Stanford. El modelo es capaz de

¹ *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning* Artículos académicos. <https://arxiv.org/abs/1711.05225>. Consulta: 5 de marzo de 2021.

detectar más de 14 tipos de enfermedades en radiografías de tórax, entre los cuales destaca un notable rendimiento en la detección de casos de neumonía, obteniendo un rendimiento en diversas métricas que es incluso mayor al rendimiento obtenido en comparación con radiólogos profesionales.”²

- *Robust Deep AUC Maximization A New Surrogate Loss and Empirical Studies on Medical Image Classification*: Este proyecto fue desarrollado en conjunto por la Universidad de Iowa y la Universidad Estatal de Washington, los investigadores decidieron tomar un camino distinto al tradicional, enfocando sus esfuerzos a aumentar la eficiencia del modelo incrementando particularmente la métrica AUC, que se basa en la efectividad del modelo en hacer predicciones cambiando el valor frontera para lo toma de la decisión.
- *ChestNet – A Deep Neural Network for Classification of Thoracic Diseases on Chest Radiography* : Esta propuesta de modelo de red neuronal para el diagnóstico de enfermedades de tórax fue desarrollada por HongyuWang en la escuela de Ciencias de la Computación e Ingeniería de la Universidad Politécnica de Xi’an, China. Se realizó el entrenamiento utilizando la base de datos ChestX-ray14, “la cual contiene más de 100 000 imágenes de radiografías de tórax etiquetadas con reportes radiológicos realizados por expertos. La arquitectura del modelo propone el uso de dos ramas en la red neuronal, una para realizar la clasificación y otra para llevar el paso de atención del modelo

² CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning Artículos académicos. <https://arxiv.org/abs/1711.05225>. Consulta: 5 de marzo de 2021.

utilizando mapas de calor de activación en la imagen. Entre los resultados se obtuvieron valores de hasta un 74 % de efectividad promedio en las predicciones, según la métrica AUC.”³

1.2.3. Problemas comunes

En los diversos trabajos de investigación y modelos de detección de enfermedades que se han realizado, se han visto algunos problemas comunes con respecto al desarrollo de los modelos de visión computadora, así como en los resultados obtenidos en las distintas métricas de evaluación.

1.2.3.1. Falta de datos

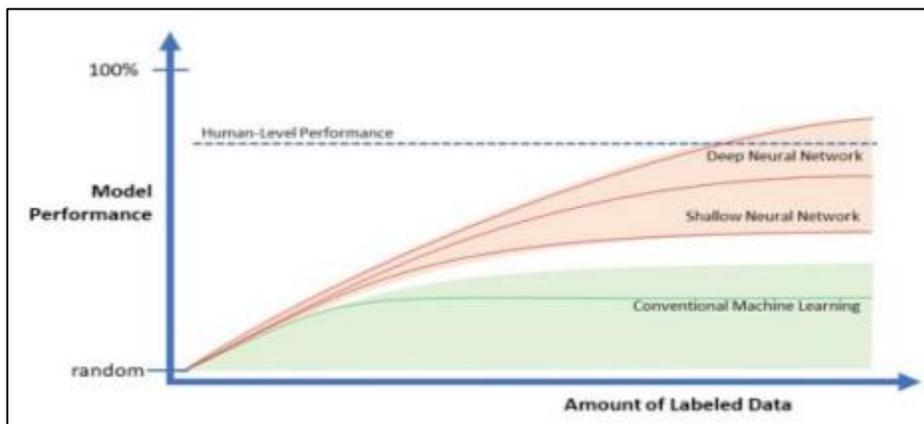
En el campo médico es especialmente complicado obtener un gran volumen de datos para entrenar modelos de inteligencia artificial, principalmente debido a las políticas de privacidad de datos con las que cuentan los centros de salud. A pesar de que no es necesario conocer los detalles de la identidad de los pacientes, si es necesario contar con datos clínicos y las imágenes de los estudios médicos realizados a los mismos.

Los modelos de redes neuronales y de visión por computadora se ven especialmente afectados en su rendimiento y efectividad al existir un bajo volumen de datos para realizar su entrenamiento, debido a que el proceso de aprendizaje automático en inteligencia artificial funciona por medio de un

³ ChestNet. *A Deep Neural Network for Classification of Thoracic Diseases on Chest Radiography*. <https://arxiv.org/abs/1807.03058>. Consulta: 6 de marzo de 2021.

proceso iterativo mediante el cual se van extrayendo patrones e información de los datos con los que se cuentan.

Figura 12. **Rendimiento en función de la cantidad de datos**



Fuente: HACKATHRON, Richard. *How Managers Should Prepare for Deep Learning*. https://miro.medium.com/max/500/1*BEQZJvZXLOQXnyX1tpLpvA.jpeg. Consulta: 7 de marzo de 2021.

Actualmente muchas instituciones de salud han visto el gran potencial que tienen los modelos de inteligencia artificial para mejorar la atención al paciente y avanzar en el desarrollo tecnológico en el ámbito de la medicina, por lo que estas políticas de privacidad se han abierto recientemente para permitir que se pueda recopilar una mayor cantidad de datos con el objetivo de utilizarlos para crear modelos de detección de enfermedades que sean más potentes y efectivos.

1.2.3.2. Desequilibrio en las clases de datos

Dentro del área de la creación de modelos de aprendizaje automático se dice que existe un desequilibrio en los datos cuando se tiene una gran cantidad

de ejemplos respecto a un grupo de clases o casos en particular y muy pocos ejemplos de otras clases.

En el caso de las bases de datos de radiografías de tórax y los reportes hechos por médicos que se tienen al respecto, existe claramente un mayor número de casos de algunas enfermedades que son comunes en la población en comparación con otras enfermedades que no los son, este desequilibrio en los datos puede causar que los modelos de visión por computadora tengan un sesgo en sus predicciones.

Este problema de desequilibrio se suele compensar modificando la función de coste en el algoritmo de entrenamiento para el modelo de visión por computadora, colocando un mayor peso de relevancia a las clases menos representadas en comparación con aquellas en donde se tienen muchos ejemplos, para que de esta manera el modelo aprenda a dar predicciones de una forma más balanceada.

1.2.3.3. Sobreajuste en tipos de imágenes o en datos locales

El problema del sobreajuste en modelos de visión por computadora suele tener orígenes muy diversos. En ocasiones, se ha visto que puede suceder debido a la distribución poco equilibrada de los datos, al formato que tienen las imágenes durante el entrenamiento, o debido a que el modelo aprende a detectar objetos particulares dentro de las imágenes que son poco relevantes al problema.

En el caso de la clasificación de enfermedades en radiografías de tórax, se han notado problemas de sobreajuste en los modelos, debido a la gran

variedad de formatos para tomas de imágenes de radiografías de tórax que se dan alrededor del mundo, así como en la calidad de las imágenes y las máquinas utilizadas para obtener las radiografías. En general es preferible contar con imágenes de alta calidad para poder entrenar un modelo de inteligencia artificial de forma eficiente, especialmente en tareas tan especializadas como detectar enfermedades en estudios clínicos.

Otros casos problemáticos que pueden conducir al sobreajuste de los modelos de radiografías de tórax, es el entrenar los modelos solamente con datos locales de una región geográfica. Si se evalúa el modelo utilizando radiografías de otras regiones, pueden existir problemas si estas difieren en la calidad de la imagen o en el formato con el cuál se extrajo la radiografía.

1.2.4. Métricas de evaluación del modelo

Con el objetivo de medir el rendimiento y eficiencia de un modelo de inteligencia artificial se establecen métricas de evaluación en base a las predicciones que se obtienen de un modelo. En el campo de algoritmos enfocados a áreas sumamente delicadas como medicina y el cuidado de la salud, la evaluación del modelo toma aún más relevancia, es por ello que se utilizan varias métricas que evalúan el rendimiento del modelo desde distintas perspectivas.

1.2.4.1. Accuracy

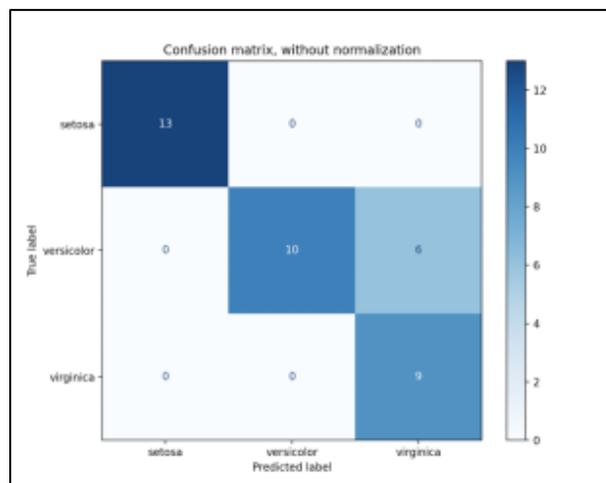
La métrica más sencilla para medir la eficiencia de un modelo es la de *Accuracy* o en español denominada como “exactitud”, para obtener este valor primero se utiliza el modelo para hacer predicciones sobre datos nuevos, los cuales no se han utilizado durante el entrenamiento. Posteriormente se calcula

la exactitud dividiendo el número de predicciones correctas dentro del total de predicciones realizadas, como resultado se obtiene un número entre cero y uno, si se desea en forma de porcentaje se multiplica por cien.

1.2.4.2. Confusion Matrix

Si se desea obtener un tipo de evaluación más detallado se puede utilizar una matriz en la cual se pueden colocar los totales de las predicciones dadas por el modelo para cada clase (por ejemplo: en columnas) y los valores reales con los que corresponde en otro eje perpendicular al primero (por ejemplo: en filas).

Figura 13. Ejemplo de *Confusion Matrix* para tres clases



Fuente: Scikit-Learn. *Confusion Matrix*. https://scikit-learn.org/stable/_images/sphx_glr_plot_confusion_matrix_001.png.

Consulta: 19 de marzo de 2021.

1.2.4.2.1. Verdaderos positivos y negativos

A partir de las celdas que conforman la *confusion matrix* se obtiene el número de predicciones en las cuales el modelo realizó un buen trabajo clasificando a una imagen correctamente con la clase que le corresponde, a este valor se le conoce como verdaderos positivos (TP). En el caso de clasificación de radiografías de tórax, esto sucede cuando el modelo predice que una imagen pertenece a un tipo de enfermedad y esto concuerda con el diagnóstico hecho por profesionales de la salud.

Se conoce como verdaderos negativos (TN) al número de predicciones hechas por el modelo en las cuales acertadamente predice que un caso no pertenece a una clase particular. En el modelo de diagnóstico de enfermedades en radiografías esto sucede cuando el modelo analiza una imagen y da una probabilidad muy baja de que pertenezca a una clase de enfermedad determinada y los profesionales médicos dan un diagnóstico similar.

1.2.4.2.2. Falsos positivos

Utilizando nuevamente valores de la *confusión matrix* es posible calcular el número de predicciones en las cuales el modelo predijo incorrectamente el valor de una clase, afirmando pertenecer a una clase en particular cuando en realidad pertenece a otra, a este valor se le conoce como falsos positivos (FP).

En el caso particular de un modelo para análisis de radiografías de tórax, esto sucede cuando el modelo predice que la imagen corresponde a una enfermedad determinada, cuando en realidad esto es falso y pertenece a otro tipo de enfermedad o trata de un paciente sano.

1.2.4.2.3. Falsos negativos

Se conoce como falsos negativos (FN) al número de predicciones en las cuales el modelo predijo incorrectamente que el elemento ingresado al modelo no pertenece a una clase determinada, cuando en realidad si pertenece. En el caso de clasificación de enfermedades en radiografía de tórax, esto sucede cuando el modelo predice que una imagen no pertenece a un tipo de enfermedad, cuando el resultado real analizado por profesionales dice lo contrario.

1.2.4.3. Sensitivity

Se refiere a la fracción o porcentaje de todos los casos que son positivos o que pertenecen a una clase y que el modelo clasificó correctamente. También se conoce como la tasa de verdaderos positivos, y se calcula dividiendo el número de verdaderos positivos dentro del valor de la suma de verdaderos positivos y falsos negativos. Para calcular el valor de *sensitivity* (TPR) se pueden utilizar los valores anteriormente vistos para la *confusion matrix* : $TPR = TP / (TP + FN)$.

1.2.4.4. Specificity

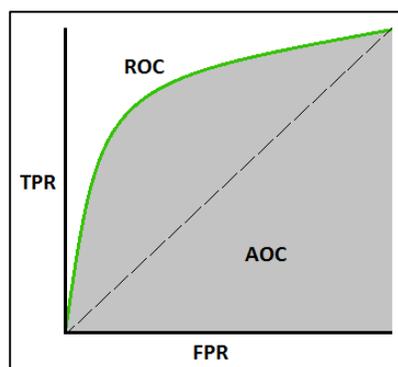
Esta cantidad indica la fracción de todos los ejemplos negativos que son correctamente clasificados por el modelo de inteligencia artificial. Se calcula dividiendo el número de verdaderos negativos dentro de la suma entre los verdaderos negativos y falsos positivos. Es posible calcular el valor de *specificity* de un modelo utilizando la *confusion matrix*: $specificity = TN / (TN + FP)$.

1.2.4.5. AUC - ROC Curve

La métrica *AUC-ROC Curve* es una de las más utilizadas para medir el nivel de rendimiento de un modelo de inteligencia artificial enfocado a realizar tareas de clasificación, en ella se varían los valores de umbral para realizar las predicciones.

ROC viene del acrónimo en inglés para “Características de funcionamiento del receptor” y es una función que relaciona las métricas vistas anteriormente de *sensitivity* (TPR) y un valor llamado FPR que es igual a uno menos el valor calculado anteriormente llamado *specificity*. Graficando estos dos valores se obtiene una curva llamada *ROC Curve*.

Figura 14. AUC – ROC Curve



Fuente: NARKHEDE, Sarang. *Understanding AUC-ROC Curve*.
https://miro.medium.com/max/722/1*pk05QGzoWhCgRiiFbz-oKQ.png.

Consulta: 3 de abril de 2021.

El término AUC viene del acrónimo en inglés para “área bajo la curva” y su valor se obtiene calculando el área bajo la curva de la gráfica de ROC. La interpretación de esta métrica es que un modelo con rendimiento excelente

tendrá un valor AUC cercano a uno, lo cual quiere decir que tiene una capacidad perfecta para clasificar las imágenes dando las predicciones correctas.

En el otro extremo se tiene el caso en el cual el valor del área bajo la curva es igual a 0,5, este valor indica que el modelo no tiene capacidad para dar buenas predicciones y que no puede distinguir entre las distintas clases a la hora de analizar una imagen. Si se tiene un valor cercano a cero, quiere decir que el modelo da predicciones de forma invertida.

El valor del área bajo la curva de la función ROC también puede ser interpretado como la probabilidad que tiene un modelo de distinguir correctamente entre las diferentes clases a la hora de analizar una imagen y dar una predicción.

1.2.5. Beneficios y aplicaciones de la detección de enfermedades utilizando visión por computadora

En las últimas décadas, ha sido posible observar la gran cantidad de beneficios que se obtienen al implementar soluciones tecnológicas en las distintas áreas del conocimiento y trabajo humano. Existen áreas en donde la mayoría de trabajo manual y repetitivo se ha podido sustituir efectivamente por sistemas automatizados mediante computadoras.

Es importante notar que existen áreas en donde el trabajo a realizar es muy delicado y va cargado de una gran responsabilidad, por lo que aún se necesita de una atenta supervisión humana; existen áreas en donde aún no es posible implementar procesos totalmente automatizados. Tal es el caso del área de medicina y cuidado de la salud, en donde un sistema de inteligencia artificial

aún no puede tomar decisiones de forma autónoma respecto a la salud de un paciente. Sin embargo, el implementar modelos de inteligencia artificial en el área de salud trae consigo una serie de beneficios y avances que pueden llegar a ser muy importantes en la actualidad y en el futuro inmediato.

1.2.5.1. Herramienta complementaria para realizar diagnósticos preliminares

Uno de los beneficios más notables de implementar sistemas informáticos es la facilidad que se tiene para colocarlos de forma pública y que lleguen a estar al alcance de muchas personas de una forma rápida. En países en vías de desarrollo, existe una amplia demanda de profesionales de salud y cuidado médico, especialmente en áreas remotas y de difícil acceso.

Un modelo de visión por computadora para detección de enfermedades puede ayudar a realizar análisis preliminares de forma automática y rápida en lugares donde exista poco personal médico, o incluso en áreas pobladas donde exista una gran demanda por una subida en los casos de enfermedades o porque no se cuente con un profesional disponible de forma inmediata.

Uno de los casos más exitosos ha sido el programa implementado en la India, en el cual diversos centros médicos en el país utilizan modelos de visión por computadora para realizar análisis preliminares sobre estudios en los ojos de pacientes y detectar indicios de ceguera. Si un estudio resulta ser positivo, el paciente viaja hacia la ciudad más cercana para ser atendido por un profesional. De esta manera se puede atender a más personas en un país con una gran población y territorio, en donde existe carencia de personal médico.

Durante la pandemia del coronavirus SARS-CoV-2, en los años 2020 y 2021, se utilizaron modelos de inteligencia artificial para detectar automáticamente personas con temperatura elevada en ciertas partes del cuerpo utilizando cámaras de infrarrojos, así como para detectar si una persona utiliza mascarilla al ingresar a un sitio. De esta manera se pudieron automatizar los procesos de bio-seguridad y de una forma preliminar combatir la propagación del virus.

1.2.5.2. Aplicaciones didácticas

Otra área en la cual se pueden aplicar modelos de inteligencia artificial para detectar enfermedades es en la de enseñanza y entrenamiento de futuros profesionales de la salud, tanto médicos como radiólogos. Es posible crear plataformas didácticas para estudiantes que se encuentren aprendiendo a analizar radiografías o demás exámenes médicos; en ellas las personas pueden aprender las partes importantes de una imagen en las que se fija un modelo de inteligencia artificial para realizar su análisis, o las áreas en donde detecta anomalías.

2. DISEÑO DEL MODELO DE VISIÓN POR COMPUTADORA PARA ANÁLISIS DE RADIOGRAFÍAS DE TÓRAX

Con el objetivo de crear un sistema especializado en la clasificación de imágenes de radiografías de tórax, es necesario diseñar una arquitectura de acuerdo al problema que se desea resolver. En el área de reconocimiento de imágenes con redes neuronales, la arquitectura de la red tiene un rol fundamental para obtener mejores predicciones.

2.1. Tecnologías utilizadas

Se describen las herramientas utilizadas para el desarrollo del modelo de visión por computadora que busca analizar y clasificar radiografías de tórax según las enfermedades que podría presentar.

2.1.1. Python

“Es un lenguaje de programación desarrollado en 1980 por Guido van Rossum. Entre sus principales características se encuentra su sintaxis simple, ser orientado a objetos, interpretado y multiplataforma.”⁴

Debido a su simplicidad y potencia, *Python* se ha convertido en un lenguaje muy popular, especialmente en áreas enfocadas a la ciencia de datos

⁴ Python Institute. *What is Python*. <https://pythoninstitute.org/what-is-python/>. Consulta: marzo de 2021

como Matemáticas, Estadística e Inteligencia Artificial. Actualmente existe un gran número de librerías de código abierto para la creación de modelos de redes neuronales escritas en *Python*.

A pesar de no ser uno de los lenguajes de programación más potentes y eficientes, el desarrollo de aplicaciones de inteligencia artificial en *Python* se compensa con una mayor velocidad de desarrollo, acceso a múltiples recursos de código abierto y fácil conexión con otros sistemas.

2.1.2. Kaggle.com

Es una aplicación online enfocada en el desarrollo y competencia de sistemas de *Machine Learning* y Ciencias de Datos en general. A través de esta plataforma se tiene acceso de forma gratuita a una gran multitud de bases de datos disponibles para su análisis y entrenamiento de modelos de inteligencia artificial, además de contar con un ambiente de desarrollo de programas utilizando *Python* como lenguaje de programación.”⁵

2.1.3. Tensorflow y Keras

TensorFlow es una plataforma de código abierto de extremo a extremo para el aprendizaje automático. Cuenta con un ecosistema integral y flexible de herramientas, bibliotecas y recursos de la comunidad que les permite a los investigadores impulsar un aprendizaje automático innovador y, a los desarrolladores, compilar e implementar con facilidad aplicaciones con tecnología de AA.”⁶

Para utilizar los modelos y recursos de *Tensorflow* de forma simplificada a través de un programa en *Python* se utiliza la librería *Keras* como interfaz de programación de aplicaciones.

⁵ Kaggle.com. *Machine Learning*. <https://www.kaggle.com/>. Consulta: 2 de septiembre de 2021.

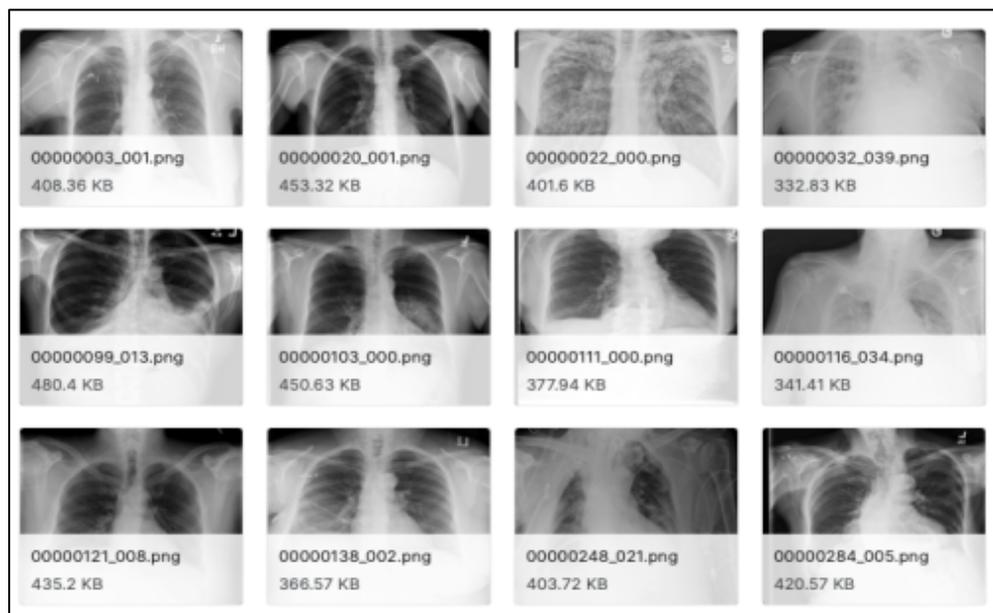
⁶ Tensorflow. *Por qué TensorFlow*. <https://www.tensorflow.org/?hl=es-419>. Consulta: marzo de 2021.

2.2. Dataset de radiografías de tórax utilizado

Para el desarrollo del modelo se utilizó en *dataset* de acceso público llamado “*NIH Chest X-ray Dataset*”, el cual fue publicado por el *National Institutes of Public Health*, es decir el conjunto de instituciones públicas de salud del gobierno de Estados Unidos.

Este conjunto de datos cuenta con 112 120 imágenes de radiografías de tórax realizadas a 30 805 pacientes diferentes, las cuales fueron etiquetadas según el diagnóstico hecho por radiólogos expertos. Estos recursos fueron publicados con el objetivo de proveer una fuente de datos extensa para la creación de modelos de aprendizaje automático capaces de analizar los datos.

Figura 15. Ejemplos de radiografías utilizadas para el entrenamiento



Fuente: National Institute of Health. *NIH Chest X-ray Dataset*. <https://www.kaggle.com/nih-chest-xrays/data>. Consulta: 4 de abril de 2021.

2.2.1. Análisis de balance de clases

Como primer paso para comprender la distribución de los datos con los que se cuentan de forma pública en las bases de datos de radiografías, se utilizó el lenguaje de programación *Python* para crear una gráfica en forma de histograma que ayude a comprender la distribución de cada clase o tipo de enfermedad que se encuentra dentro del conjunto de datos para entrenar el modelo.

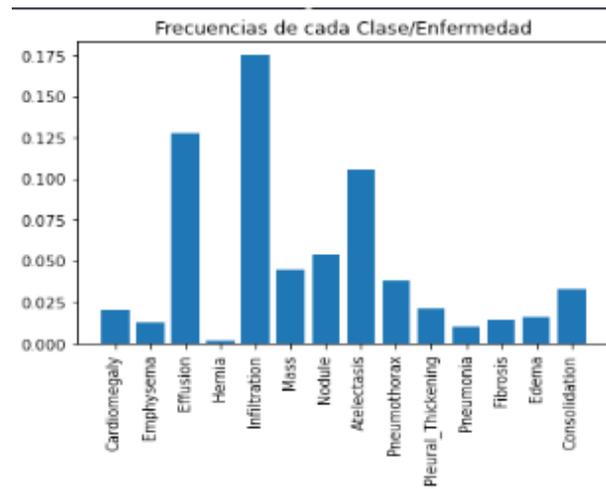
Figura 16. Algoritmo para crear un histograma desarrollado en *Python*

```
train_generator = get_train_generator(train_df, IMAGE_DIR, 'Image', labels)
valid_generator, test_generator = get_test_and_valid_generator(valid_df, test_df, train_df, IMAGE_DIR, 'Image', labels)

plt.xticks(rotation=90)
plt.bar(x=labels, height=np.mean(train_generator.labels, axis=0))
plt.title("Frecuencias de cada Clase/Enfermedad")
plt.show()
```

Fuente: elaboración propia, utilizando Python 2010.

Figura 17. Gráfica de distribución de clases en el conjunto de datos



Fuente: elaboración propia, utilizando Python 2010.

Analizando la gráfica mostrada en la figura 16, es posible observar que existe un desequilibrio en la distribución de los datos. Esto debido a que existen clases de enfermedades que son muy comunes y que por lo tanto tienen una gran cantidad de ejemplos en los conjuntos de imágenes públicas.

Entre las enfermedades que cuentan con una mayor cantidad de ejemplos se encuentran: infiltración, efusión, atelectasia y nodule. Por otra parte, existen enfermedades que no son comunes en la población y que por este motivo tienen una baja cantidad de imágenes de ejemplo disponibles para entrenar el modelo, entre estas clases se tienen: hernia, neumonía, enfisema y fibrosis.

El algoritmo utilizado para entrenar el modelo debe tomar en cuenta este desequilibrio durante el entrenamiento para evitar que el modelo resultante se encuentre sesgado en predecir de forma más probable algunas enfermedades.

2.3. Desarrollo del modelo de visión por computadora

El modelo de inteligencia artificial utilizado para la clasificación de imágenes de radiografías fue desarrollado por la Universidad de Stanford en su concurso de inteligencia artificial “CheXpert” y es posible encontrarlo de forma pública en su repositorio de GitHub y en su curso online “AI For Medical Diagnosis” en la plataforma de aprendizaje Coursera.

Se tomó como base una red neuronal pre-entrenada con el objetivo de reutilizar los pesos optimizados en las neuronas que fueron entrenados en un gran conjunto de datos para clasificar diversos tipos de imágenes en general.

Posteriormente se adaptó sobre la primera red neuronal base una segunda red neuronal con más capas de neuronas que fueron entrenadas

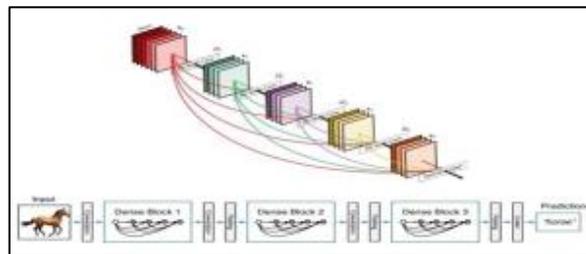
específicamente con imágenes de radiografías de tórax con el objetivo de optimizar el modelo para realizar la tarea de análisis deseada.

2.3.1. Modelo base de la red utilizando transfer learning

El primer modelo utilizado de base es la red *DenseNet121* la cual utiliza capas de convolución para detectar características en las imágenes y posteriormente brindar predicciones en forma de clasificación. Esta red fue entrenada para clasificar más de 1 000 imágenes distintas, utilizando tres conjuntos de datos públicos distintos.

Los conjuntos de datos utilizados para entrenar estas redes fueron CIFAR (50 000 imágenes) e ImageNet (1,2 millones de imágenes) con esto la red cuenta con aprendizaje generalizado para analizar una gran diversidad de imágenes. Este aprendizaje se puede transferir a la tarea de analizar radiografías de tórax al colocar sobre la red base nuevas capas de convolución con redes neuronales que se especialicen en la nueva tarea.

Figura 18. **Arquitectura de red neuronal *DenseNet121***



Fuente: HUANG, Gao. *Densely Connected Convolutional Networks*. .
<https://arxiv.org/pdf/1608.06993.pdf>. Consulta: 5 de abril de 2021

2.3.2. Modelo complementario de la red y capa de salida

Sobre el primer modelo base se agregaron nuevas capas y en base al análisis de distribución de datos se creó una función de coste que pudiera resolver el problema del desequilibrio de clases en la distribución de los datos, con el objetivo de generar un modelo que no se encuentre sesgado en pocos tipos de enfermedades y que pueda brindar atención al aprendizaje de distintas clases de enfermedades de forma más general.

El modelo complementario que se agregó a la red cuenta con una capa de *Pooling* que se encarga de comprimir la información obtenida por el modelo base y posteriormente se agregó una capa con redes que utilizan la función de activación *Sigmoide* para obtener las probabilidades de las predicciones para cada tipo de enfermedad que se soporta en el modelo.

Figura 19. Código escrito en *Python* para ensamblaje de modelo base y modelo complementario

```
base_model = DenseNet121(weights=trained_data_path+'densenet.hdf5', include_top=False)
x = base_model.output
x = GlobalAveragePooling2D()(x)

predictions = Dense(len(labels), activation='sigmoid')(x)
model = Model(inputs=base_model.input, outputs=predictions)
model.compile(optimizer='adam', loss=get_weighted_loss(pos_weights, neg_weights))

model.load_weights(trained_data_path + '/pretrained_model.h5')

predicted_vals = model.predict_generator(test_generator, steps=len(test_generator))
```

Fuente: elaboración propia, utilizando Python 2010.

Se utilizó el *framework* para creación de redes neuronales llamado *Keras* en conjunto con el lenguaje de programación *Python*, con el objetivo de definir

la nueva función de coste que se ajusta a los datos, así como para ensamblar tanto el modelo base como el modelo posterior en una misma red neuronal interconectada.

Figura 20. **Funciones escritas en Python para soportar el desequilibrio en la distribución de los datos durante el entrenamiento**

```
def get_weighted_loss(pos_weights, neg_weights, epsilon=1e-7):  
  
    def weighted_loss(y_true, y_pred):  
  
        loss = 0.0  
        for i in range(len(pos_weights)):  
  
            loss += -K.mean( pos_weights[i]*y_true[:,i]*K.log(y_pred[:,i]+epsilon) + neg_weights[i]*(1-y_true[:,i])*K.log(1-y_pred[:,i]+epsilon) )  
        return loss  
  
    return weighted_loss  
  
  
def compute_class_freqs(labels):  
    N = labels.shape[0]  
  
    positive_frequencies = np.sum(labels,axis=0)/N  
    negative_frequencies = np.sum(labels==0,axis=0)/N  
  
    return positive_frequencies, negative_frequencies
```

Fuente: elaboración propia, utilizando Python 2010.

2.4. Entrenamiento del modelo

Para entrenar el modelo de red neuronal en primer lugar fue necesario crear una función para generar el conjunto de entrenamiento de imágenes de radiografías de tórax en el formato que reconoce el *framework* de *Tensorflow* y *Keras* para el lenguaje de programación *Python*.

Debido al gran tamaño del conjunto de datos en cuanto a la cantidad y calidad de cada una de las imágenes, el proceso de entrenamiento con gráficas en completa definición se convierte en un proceso costoso en términos de capacidad de cómputo y tiempo de procesamiento, requiriendo en muchos

casos el uso de computadoras con *hardware* especializado en *machine learning*. Es por este motivo que se tomó la decisión de reducir las imágenes a un tamaño de 320x320 píxeles con el objetivo de poder procesarlos en una computadora personal.

Figura 21. **Función escrita en *Python* para crear un conjunto de entrenamiento**

```
def get_train_generator(df, image_dir, x_col, y_cols, shuffle=True, batch_size=8, seed=1, target_w = 320, target_h = 320):  
  
    image_generator = ImageDataGenerator(  
        samplewise_center=True,  
        samplewise_std_normalization=True)  
  
    generator = image_generator.flow_from_dataframe(  
        dataframe=df,  
        directory=image_dir,  
        x_col=x_col,  
        y_col=y_cols,  
        class_mode='raw',  
        batch_size=batch_size,  
        shuffle=shuffle,  
        seed=seed,  
        target_size=(target_w, target_h))  
  
    return generator  
  
train_generator = get_train_generator(train_df, IMAGE_DIR, 'Image', labels)
```

Fuente: elaboración propia, utilizando Python 2010.

Con el objetivo de entrenar el modelo también fue necesario leer el archivo con formato CSV provisto junto al conjunto de datos del *National Institute of Health* que contiene el nombre de cada una de las imágenes de radiografías de tórax y su respectiva enfermedad diagnosticada por profesionales en medicina y radiología. Se denota con un número uno cuando una clase de enfermedad se encuentra presente en una radiografía y con un número cero en caso contrario, en la figura a continuación es posible observar una muestra de estos datos.

Figura 22. Muestra de imágenes etiquetadas en el conjunto de datos

```
train_df = pd.read_csv(trained_data_path + "/train-small.csv")
valid_df = pd.read_csv(trained_data_path + "/valid-small.csv")
test_df = pd.read_csv(trained_data_path + "/test.csv")

train_df
```

	Image	Atelectasis	Cardiomegaly	Consolidation	Edema	Effusion	Emphysema	Fibrosis	Hernia
0	00008270_015.png	0	0	0	0	0	0	0	0
1	00029855_001.png	1	0	0	0	1	0	0	0
2	00001297_000.png	0	0	0	0	0	0	0	0
3	00012359_002.png	0	0	0	0	0	0	0	0
4	00017951_001.png	0	0	0	0	0	0	0	0
...
995	00015869_010.png	0	0	0	0	0	0	0	0
996	00020113_005.png	0	0	0	0	1	0	0	0
997	00019939_000.png	0	0	0	0	0	0	0	0
998	00030496_000.png	0	0	0	0	0	0	0	0

Fuente: elaboración propia, utilizando Python 2010.

Como paso final se procedió a realizar el entrenamiento de la red neuronal utilizando la función *fit_generator* que forma parte del *framework* de *Keras* para que se ejecuten los pasos de *forward propagation*, *backward propagation* y descenso del gradiente, con el objetivo de ajustar los pesos de cada una de las neuronas en la red, hasta encontrar los valores óptimos que minimicen el error y que generen las mejores predicciones para el conjunto de datos de imágenes de radiografías de tórax que se introdujeron al modelo.

Figura 23. Proceso de entrenamiento del modelo

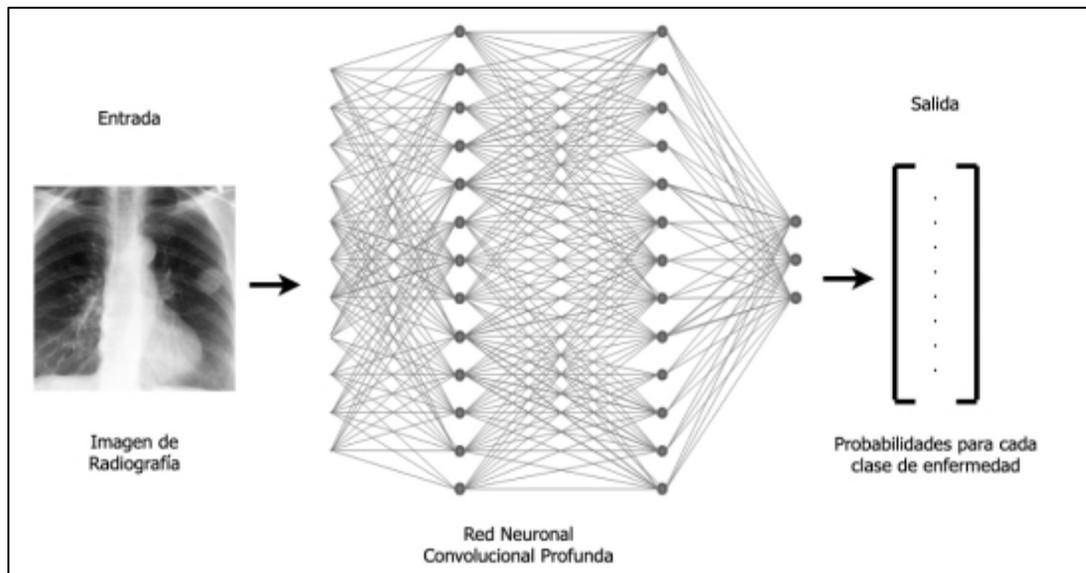
```
history = model.fit_generator(train_generator,
                             validation_data=valid_generator,
                             steps_per_epoch=50,
                             validation_steps=15,
                             epochs = 3)
```

```
Epoch 1/3
50/50 [=====] - 457s 9s/step - loss: 1.0265 - val_loss: 2.7063
Epoch 2/3
50/50 [=====] - 415s 8s/step - loss: 0.8815 - val_loss: 8.3629
Epoch 3/3
```

Fuente: elaboración propia, utilizando Python 2010.

El resultado de la etapa de entrenamiento es un modelo con los parámetros optimizados para realizar predicciones. En este caso el modelo recibe como entrada una radiografía de tórax que pasa a ser procesada a través de las distintas capas de la red neuronal, hasta tener como salida una lista de probabilidades para cada tipo de enfermedad que admite el modelo.

Figura 24. **Arquitectura del modelo entrenado**



Fuente: elaboración propia, utilizando Draw.io.

2.5. Salidas del modelo

Al ingresar una imagen de radiografía de tórax al modelo se obtienen las probabilidades para cada tipo de enfermedad que se encontraba en el conjunto de datos de entrenamiento. Este resultado puede ser interpretado de distintas formas, la primera de ellas consiste en tomar como predicción del modelo a la enfermedad que tiene la probabilidad más alta o bien, a un conjunto de ellas.

Figura 25. **Salida con probabilidades para cada tipo de enfermedad**

Clase ↕	Probabilidad ↕
Mass	0.96
Pleural_Thickening	0.88
Fibrosis	0.84
Emphysema	0.83
Hernia	0.81
Nodule	0.78
Pneumothorax	0.59
Consolidation	0.42

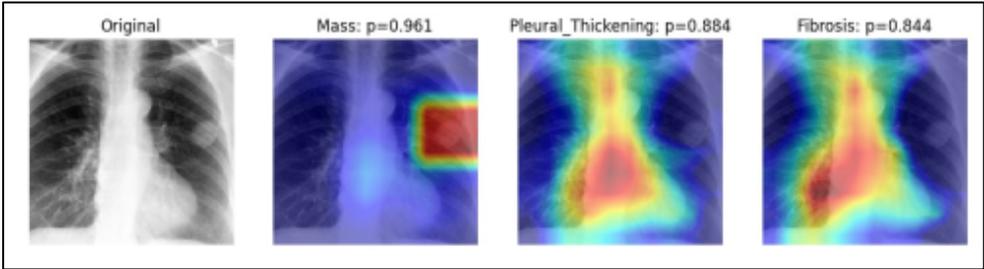
Fuente: elaboración propia, utilizando React JS.

Tomando la lista de probabilidades, así como los valores de las últimas capas de la red neuronal es posible crear mapas de calor con aquellas neuronas que se activaron más al momento de analizar y reconocer la imagen dentro del modelo. Estas neuronas dentro de la red neuronal profunda normalmente se especializan en detectar ciertos tipos de características en determinadas áreas de la imagen.

Aquellas áreas en donde las neuronas se activaron más al reconocer la imagen se colorean con tonos más cercanos al rojo, mientras que las áreas donde las neuronas se activaron en menor medida se colorean con tonos cercanos al violeta.

En la figura 26, es posible observar un ejemplo de mapas de calor generados a partir de los resultados del modelo para una imagen de radiografía de tórax en particular, se toman las tres clases de enfermedades con las probabilidades más altas.

Figura 26. **Mapas de calor creados en *Python* como salida del modelo**



Fuente: elaboración propia, utilizando Python 2010.

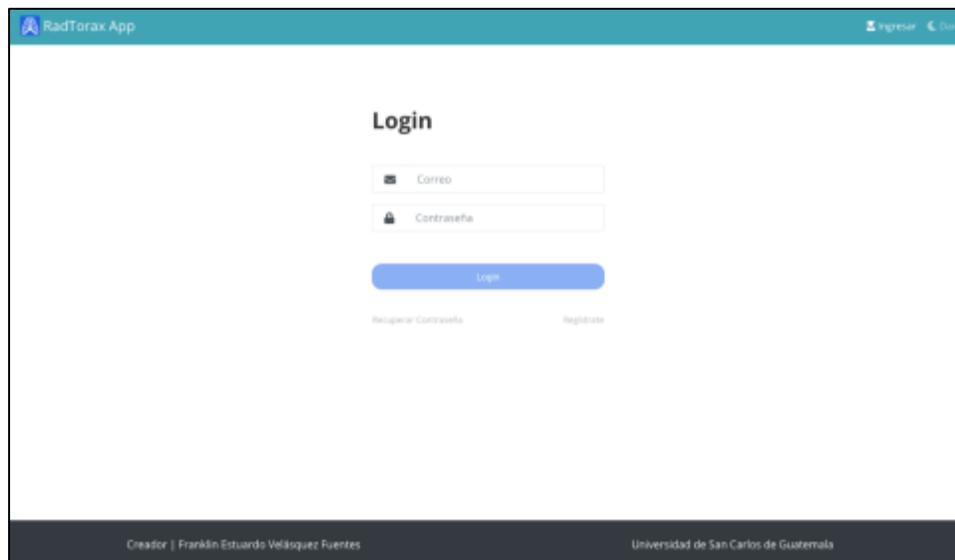
3. DISEÑO DE LA APLICACIÓN WEB

La aplicación web está pensada para que el usuario pueda registrarse en la misma con sus datos personales y acceder a la funcionalidad de realizar análisis de radiografías de tórax.

3.1. Vistas

En primer lugar se tiene el área de ingreso en donde el usuario puede acceder a las funcionalidades de la aplicación ingresando su correo y contraseña, o bien puede dirigirse al área de registro para crear una cuenta nueva.

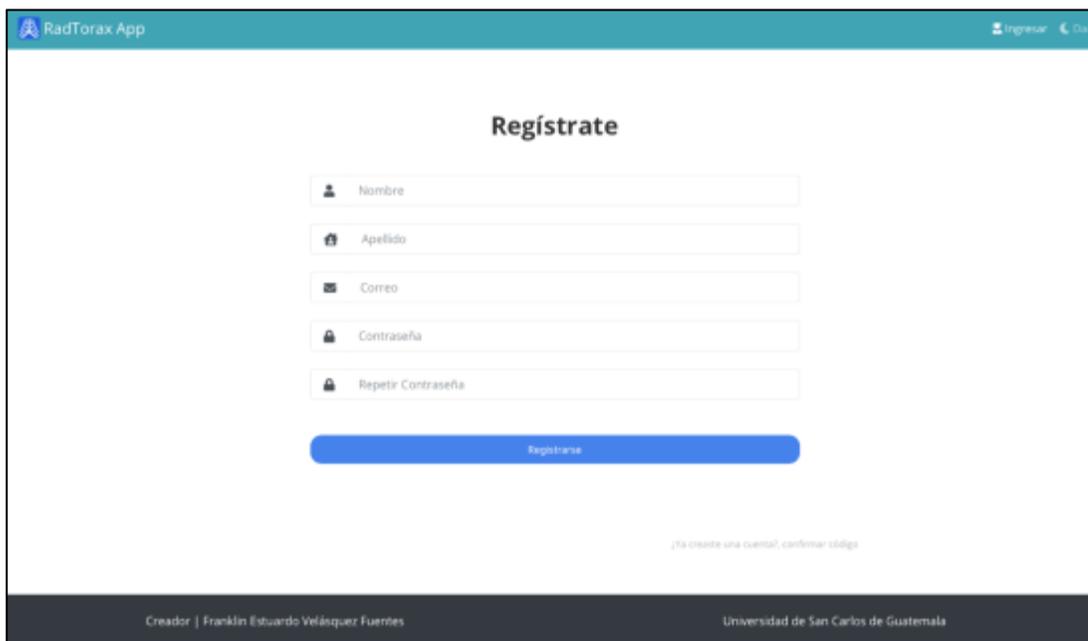
Figura 27. **Página de ingreso a la plataforma**



Fuente: elaboración propia, utilizando RadTorax App.

Un usuario debe registrarse en la aplicación con sus datos personales para crear una cuenta que le permita el acceso a las funcionalidades de la misma.

Figura 28. **Página para el registro de nuevo usuario**



The image shows a mobile application interface for registration. At the top, there is a teal header with the text "RadTorax App" on the left and "Ingresar" and "Salir" on the right. The main content area is white and features the title "Regístrate" in bold black text. Below the title are five input fields, each with a small icon to its left: "Nombre" (person icon), "Apellido" (house icon), "Correo" (envelope icon), "Contraseña" (lock icon), and "Repetir Contraseña" (lock icon). Below these fields is a prominent blue button with the text "Registrarse". At the bottom of the form area, there is a small link that says "¿Ya creaste una cuenta?, confirmar código". The footer is a dark grey bar containing the text "Creador | Franklin Estuardo Velásquez Fuentes" on the left and "Universidad de San Carlos de Guatemala" on the right.

Fuente: elaboración propia, utilizando RadTorax App.

Posteriormente el usuario recibe en el correo ingresado un código de confirmación que debe ingresar en la plataforma para poder terminar con el proceso de creación de su cuenta. Este paso se lleva a cabo con el objetivo de asegurar que se creen cuentas con correos válidos y que los usuarios sean reales, de esta manera se puede evitar la creación de cuentas utilizando robots que suelen atacar sistemas en internet.

Figura 29. Vista para confirmación de código de seguridad

RadTorax App

Ingresar

Confirmar Cuenta

Revisa el correo electrónico que ingresaste y busca el código de confirmación de cuenta. Puede que también debas revisar en spam o correo no deseado. Ingresar el código a continuación.

Correo

Código

Confirmar

Creador | Franklin Estuardo Velásquez Fuentes

Universidad de San Carlos de Guatemala

Fuente: elaboración propia, utilizando RadTorax App.

Una vez creada una cuenta, el usuario puede ingresar a la página con sus credenciales y acceder a la vista en la cual puede realizar análisis de radiografías de tórax utilizando el modelo entrenado y que ahora se encuentra cargado en el servidor.

En esta página se proveen ejemplos de radiografías de tórax para realizar un análisis, así como un enlace para acceder a más de 100,000 radiografías. El usuario puede subir su propia radiografía desde su computadora o dispositivo móvil, para ello es necesario tomar una fotografía de calidad y que la radiografía se encuentre sobre un marco iluminado, los cuáles utilizan los doctores para analizar radiografías y resaltar los detalles de estas.

Figura 30. Vista para realizar análisis de una nueva radiografía



Fuente: elaboración propia, utilizando RadTorax App.

Posterior a seleccionar una radiografía o subir una nueva, el usuario puede hacer ver el botón “Analizar”, el cuál al ser accionado envía la imagen al servidor para su análisis.

Se muestra una barra de progreso conforme se realiza el proceso de análisis y clasificación de la imagen en el servidor, así como la generación de los mapas de calor. En el servidor la imagen pasa por el proceso de compresión y luego se introduce al modelo, los valores en píxeles de la imagen pasan a través de las capas de la red neuronal causando las activaciones necesarias en los filtros de convolución para detectar las características importantes que se determinaron durante el entrenamiento del modelo.

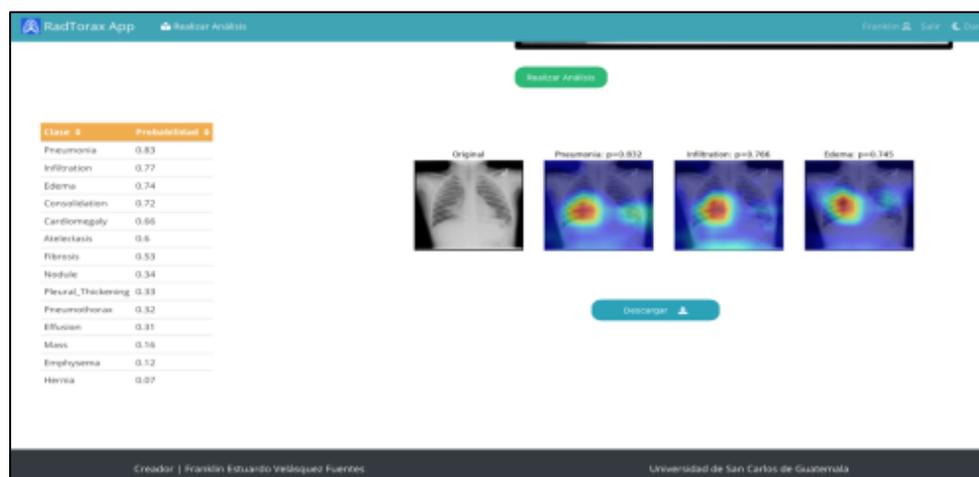
Figura 31. **Fase de envío de imagen al servidor y espera de resultados del análisis**



Fuente: elaboración propia, utilizando RadTorax App.

Como resultado, el servidor responde con la tabla con las probabilidades predichas para cada tipo de enfermedad que el modelo soporta, así como un mapa de calor de las tres enfermedades con probabilidades más altas.

Figura 32. **Página con los resultados del análisis**



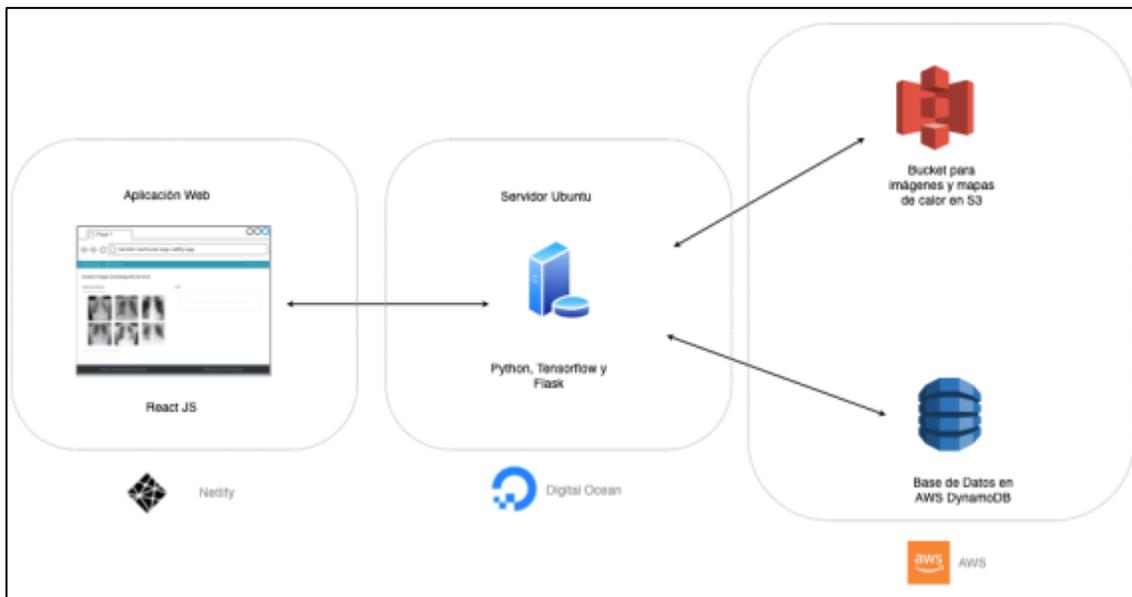
Fuente: elaboración propia, utilizando RadTorax App.

3.2. Arquitectura de la aplicación

Para la creación de la aplicación se tomó una arquitectura de la forma cliente-servidor con infraestructura en la nube, en donde el cliente es la aplicación web utilizada por el usuario para realizar análisis de radiografías y recibir los resultados.

Se tiene un servidor con el sistema operativo Ubuntu que se encarga de analizar las imágenes de radiografías de tórax y que contiene los paquetes en *Python* con los pesos del modelo entrenado anteriormente precargados para realizar el análisis.

Figura 33. Diagrama de arquitectura de la aplicación



Fuente: elaboración propia, realizado con Draw.io.

3.3. Tecnologías utilizadas

A continuación, se detallan las tecnologías y herramientas utilizadas para desarrollar la aplicación web que sirve de interfaz gráfica para que usuarios puedan utilizar el modelo de reconocimiento de radiografías utilizando inteligencia artificial. Dichas tecnologías son:

- React JS
- Amazon Web Services
 - DynamoDB
 - S3
 - Cognito
 - IAM
 - Digital Ocean Droplets

3.3.1. React JS

React es una herramienta de código abierto desarrollada por Facebook que permite crear aplicaciones web del lado del cliente utilizando HTML, CSS y JavaScript. Estas aplicaciones siguen el concepto de aplicaciones de una sola página, lo que permite versatilidad en el manejo de estado y flujo de las rutas propuestas dentro de la aplicación.

Utilizando las herramientas proveídas por el marco de trabajo de *React*, se crearon las páginas y pantallas de la aplicación web que los usuarios pueden utilizar desde su computadora o dispositivo móvil para enviar radiografías de tórax para ser analizadas por el modelo.

Con el objetivo de crear interfaces que sean amigables para los usuarios en su diseño y funcionamiento, se utilizó *Bootstrap* como librería para implementar el estilo de la aplicación con un diseño minimalista y adaptable a distintos tamaños de pantalla.

3.3.2. Amazon Web Services

Amazon Web Services ó *AWS* es un proveedor de computación en la nube que provee de una gran variedad de servicios web que permite a grandes empresas y desarrolladores independientes crear aplicaciones utilizando la infraestructura proveída por *Amazon*. De esta manera es posible crear aplicaciones sin necesidad de comprar una gran cantidad de computadoras o equipo especializados.

En la actualidad utilizar servicios en la nube es una manera conveniente de tener un espacio para poder desarrollar aplicaciones que puedan ser escalables sin necesidad de tener que esforzarse mucho con el mantenimiento e inversión en computadoras e infraestructura.

3.3.2.1. DynamoDB

Servicio de base de datos no relacional, que permite guardar datos en un formato muy similar a *JSON* ó *JavaScript Object Notation*. Tiene como beneficios que al ser una base de datos no relacional permite una gran tasa de transacción, así como la posibilidad de guardar los datos en un formato flexible y ajustable a las necesidades que requieran las interfaces gráficas en la aplicación.

3.3.2.2. S3

Simple Storage Service (S3) es un servicio de AWS especializado en guardar archivos que permiten una gran tasa de escritura y lectura, además provee herramientas para establecer permisos de acuerdo a los usuarios que pueden acceder a las imágenes o archivos generados por la aplicación.

3.3.2.3. Cognito

Servicio de AWS especializado en la creación, autenticación y autorización de usuarios en una aplicación. Provee a los desarrolladores herramientas para crear una base de datos de usuarios con los métodos y protocolos de autenticación más comunes, así como proveer integración con servicios externos como Google y Facebook.

3.3.2.4. IAM

Identity and Access Management (IAM) es la herramienta de AWS que brinda al desarrollador un marco de trabajo para definir políticas de acceso a los datos de la aplicación por parte de los usuarios.

3.3.2.5. Digital Ocean Droplets

Digital Ocean es un proveedor de nube que se *caracteriza* por proveer máquinas virtuales en la nube a precios accesibles. Los precios de *Digital Ocean* en el apartado de alquiler de cómputo en la nube son muy atractivos para individuos y empresas con un presupuesto ajustado y que desean un ambiente que les provea de las herramientas necesarias para implementar un ambiente de desarrollo.

En el caso de la aplicación para el análisis de radiografías de tórax, se optó por utilizar los *Droplets* o máquinas virtuales de *Digital Ocean*, como la herramienta de cómputo para correr los procesos relacionados al análisis de las imágenes de radiografías utilizando el modelo de inteligencia artificial.

4. RESULTADOS DE LA EVALUACIÓN DEL MODELO

Para evaluar el desempeño del modelo y obtener las métricas de evaluación de este, se tomó un conjunto con 420 imágenes nuevas, las cuales no fueron vistas por el modelo durante su fase de entrenamiento. De estas imágenes se conoce su clasificación correcta dada por profesionales de la salud que colaboraron al crear el conjunto de datos del NIH. Se procedió a analizar estas imágenes utilizando el modelo entrenado y se compararon las predicciones obtenidas con los valores reales.

Figura 34. Creación de un conjunto de evaluación en *Python*

```
def get_test_and_valid_generator(test_df, train_df, image_dir, x_col, y_cols,
                               sample_size=100, batch_size=8, seed=1, target_w = 320, target_h = 320):

    test_generator = image_generator.flow_from_dataframe(
        dataframe=test_df,
        directory=image_dir,
        x_col=x_col,
        y_col=y_cols,
        class_mode="raw",
        batch_size=batch_size,
        shuffle=False,
        seed=seed,
        target_size=(target_w, target_h))
    return test_generator

IMAGE_DIR = '../input/images-small-set/images-small-set/'
test_df = pd.read_csv(trained_data_path + "/test.csv")
test_generator = get_test_and_valid_generator(test_df, train_df, IMAGE_DIR, 'Image', labels)
```

Fuente: elaboración propia, utilizando Python 2010.

4.1. Confusion Matrix

La *confusion matrix* se utiliza para colocar en forma de matriz el recuento de las clasificaciones correctas e incorrectas dadas por el modelo para cada una de las distintas enfermedades que el modelo es capaz de clasificar. De esta

manera es posible comparar las predicciones dadas por el modelo con los valores reales y calcular las métricas de evaluación del modelo.

Figura 35. Programa escrito en *Python* para obtener las predicciones en imágenes nuevas para calcular y graficar la *confusion matrix*

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import multilabel_confusion_matrix, classification_report
from sklearn.metrics import confusion_matrix

predicted_vals = model.predict_generator(test_generator, steps=len(test_generator))
gt = test_generator.labels
pred = ( predicted_vals > 0.7)
cf_matrix = multilabel_confusion_matrix(gt,pred)

def print_confusion_matrix(confusion_matrix, axes, class_label, class_names, fontsize=14):

    df_cm = pd.DataFrame(confusion_matrix, index=class_names, columns=class_names,)
    try:
        heatmap = sns.heatmap(df_cm, annot=True, fmt='d', cbar=False, ax=axes)
    except ValueError:
        raise ValueError("Error creating heat map")
    heatmap.yaxis.set_ticklabels(heatmap.yaxis.get_ticklabels(), rotation=0, ha='right', fontsize=fontsize)
    heatmap.xaxis.set_ticklabels(heatmap.xaxis.get_ticklabels(), rotation=45, ha='right', fontsize=fontsize)
    axes.set_ylabel('Valor Verdadero')
    axes.set_xlabel('Predicción')
    axes.set_title( class_label)

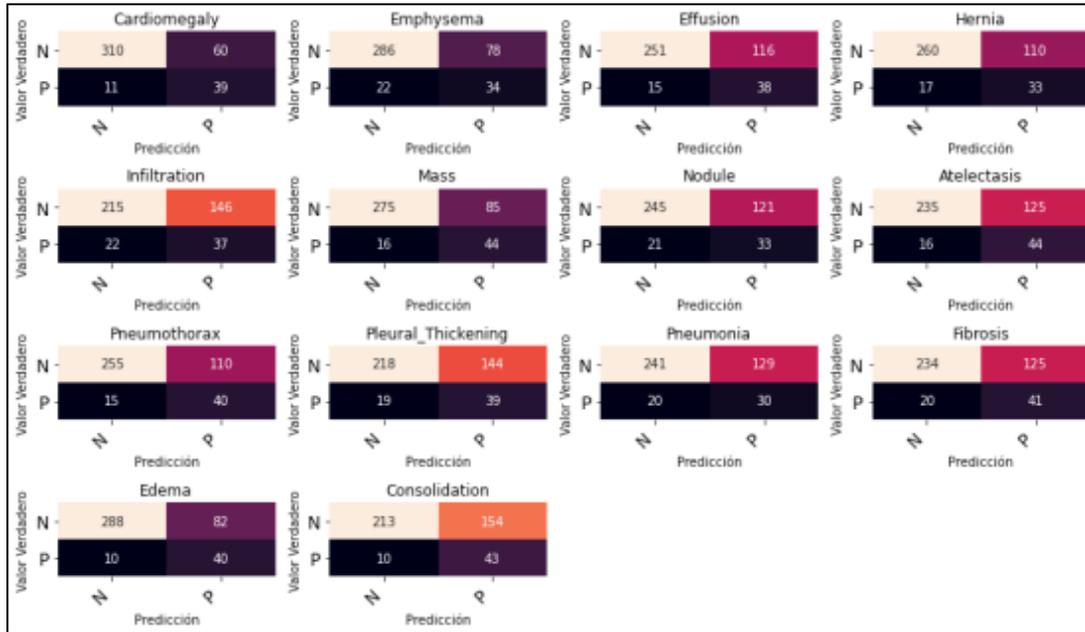
all_classes_cm = cf_matrix
fig, ax = plt.subplots(4, 4, figsize=(12, 7))
axes_list = ax.flatten()

for axes, cfs_matrix, label in zip(axes_list, vis_arr, labels):
    print_confusion_matrix(cfs_matrix, axes, label, ["N", "Y"])

axes_list[14].axis('off')
axes_list[15].axis('off')
fig.tight_layout()
plt.show()
```

Fuente: elaboración propia, utilizando Python 2010.

Figura 36. Valores de *confusion matrix* obtenidos en el conjunto de evaluación para cada una de las enfermedades



Fuente: elaboración propia, utilizando Python 2010.

A partir de las *confusion matrix* para cada tipo de enfermedad se pueden calcular diferentes métricas para la evaluación del modelo. En primer lugar se calculan las métricas de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Posteriormente, utilizando estos resultados se pueden calcular métricas más específicas, como la exactitud y la curva ROC.

Para calcular las métricas se creó un programa escrito en *Python* utilizando las librerías *Pandas*, *Seaborn*, *Matplotlib* y *Skitlearn* para el manejo y análisis de datos utilizando las salidas dadas por el modelo para el conjunto de evaluación.

Figura 37. Métricas de evaluación del modelo calculadas en *Python* para cada tipo de enfermedad

	Enfermedad	Verdaderos Negativos	Verdaderos Positivos	Accuracy	Falsos Negativos	Falsos Positivos
0	Cardiomegaly	310	39	0.830952	11	60
1	Emphysema	286	33	0.759524	23	78
2	Effusion	251	39	0.690476	14	116
3	Hernia	261	33	0.700000	17	109
4	Infiltration	215	37	0.600000	22	146
5	Mass	275	44	0.759524	16	85
6	Nodule	246	32	0.661905	22	120
7	Atelectasis	234	44	0.661905	16	126
8	Pneumothorax	256	41	0.707143	14	109
9	Pleural_Thickening	219	39	0.614286	19	143
10	Pneumonia	241	30	0.645238	20	129
11	Fibrosis	234	40	0.652381	21	125
12	Edema	285	40	0.773810	10	85
13	Consolidation	212	43	0.607143	10	155

Fuente: elaboración propia, utilizando Python 2010.

4.1.1. Verdaderos Negativos

Estos son los valores que se encuentran en las posiciones (0,0) de cada *confusion matrix*, los cuales representan la cantidad de veces que el modelo predijo correctamente que una imagen no pertenecía a esa enfermedad, es decir la predicción del modelo coincidió con el valor real.

4.1.2. Verdaderos Positivos

Indican la cantidad de veces que el modelo predijo correctamente que una radiografía de tórax pertenecía a una enfermedad dada y que coincidió con la clasificación anteriormente dada por los profesionales de la salud que etiquetaron los datos en el conjunto de pruebas. En la *confusion matrix* los verdaderos positivos se localizan en las posiciones (1,1) donde ocurre la intersección para los valores positivos.

Como es posible observar, un gran número de predicciones en la categoría de verdaderos positivos y verdaderos negativos indican un buen rendimiento de un modelo, puesto que son los casos en los cuales las predicciones fueron acertadas.

4.1.3. Falsos Positivos

Se localizan en la posición (0,1) en la *confusion matrix*, e indican la cantidad de veces que el modelo predijo incorrectamente que una imagen pertenece a una enfermedad en específico. Es decir, un gran número de predicciones en esta categoría indica un problema en el modelo, posiblemente sobre ajuste para una clase en específico y falta de generalización en el modelo.

Para un modelo es preferible contar con un número bajo de predicciones durante la evaluación que entren en la categoría de falsos positivos, pues esto afecta en el rendimiento del modelo.

4.1.4. Falsos Negativos

Esta métrica cuenta el número de predicciones en las cuales el modelo incorrectamente predijo que una imagen no presentaba una enfermedad en específico, cuando la realidad dicta lo contrario. Este valor se encuentra en las posiciones (1,0) en cada una de las matrices para las distintas enfermedades.

4.2. Accuracy

Denominada como exactitud en español, esta métrica indica el porcentaje de las predicciones que se realizaron de forma correcta. Se calcula sumando la cantidad de verdaderos positivos y verdaderos negativos y dividiendo el resultado dentro del total de predicciones hechas. Para el modelo anteriormente creado se obtuvo un valor de *accuracy* de 0,82, es decir del 82 %.

Un valor alto en esta métrica indica que un alto porcentaje de las predicciones hechas por el modelo serán correctas y sirve de referencia para determinar la fiabilidad de las predicciones, mientras que un valor bajo indica un bajo rendimiento del modelo al intentar realizar predicciones que sean correctas.

Figura 38. **Cálculo de la métrica de *Accuracy* del modelo en *Python***

```
avg_accuracy = metrics_df['Accuracy'].sum() / len(labels)

avg_accuracy

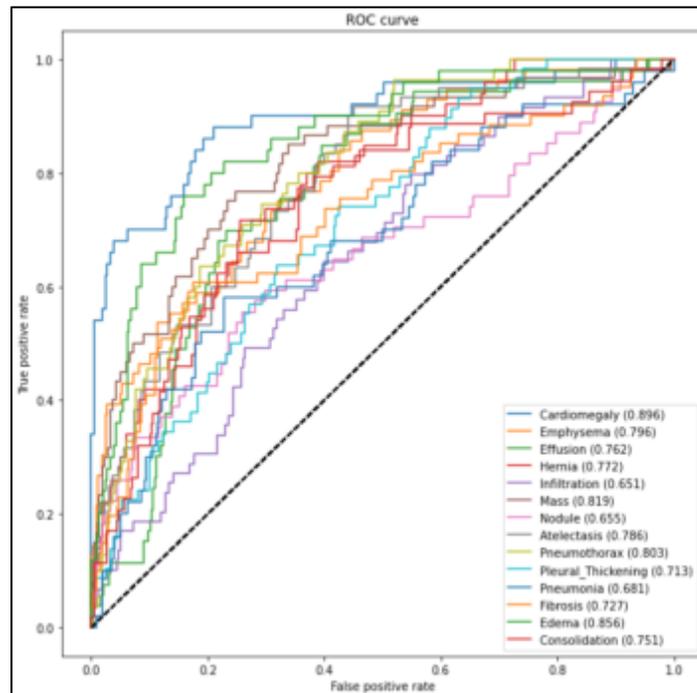
0.8263605442176872
```

Fuente: elaboración propia, utilizando Python 2010.

4.3. ROC Curve

Para cada tipo de clase de enfermedad se graficó la métrica ROC y se calculó el área bajo de la misma. Como se indica en el primer capítulo del presente trabajo, los valores cercanos a uno en esta métrica indican un modelo que sabe distinguir entre las distintas clases a la hora de hacer una predicción, mientras que un valor más bajo cercano a 0,5 indica que el modelo no es capaz de distinguir entre las distintas clases a la hora de realizar una predicción.

Figura 39. **ROC Curve del modelo creado para analizar radiografías de tórax en el conjunto de evaluación**



Fuente: elaboración propia, utilizando Python 2010.

CONCLUSIONES

1. Se implementó un modelo de inteligencia artificial especializado en el análisis de radiografías de tórax, con el objetivo de obtener predicciones acerca de las posibles enfermedades que una persona puede presentar de acuerdo con un conjunto de datos.
2. Se entrenó el modelo de reconocimiento de radiografías con conjuntos de datos públicos, con aproximadamente 112 000 imágenes de radiografías de tórax.
3. Se desplegó el modelo de inteligencia artificial entrenado a través un servidor con arquitectura en la nube, con el objetivo de recibir peticiones de usuarios que envíen nuevas imágenes de radiografías para obtener un análisis y predicciones sobre las mismas.
4. Se implementó una aplicación web que sirve de interfaz gráfica para que usuarios puedan enviar radiografías de tórax a un servidor con el modelo de inteligencia artificial implementado y que de esta manera puedan obtener los resultados de su análisis.
5. Se midieron los niveles de exactitud, tasa de falsos positivos y negativos, así como la curva ROC del modelo utilizando un conjunto de las imágenes de radiografías de tórax no utilizadas durante el entrenamiento del modelo de inteligencia artificial.

RECOMENDACIONES

1. Recopilar nuevas imágenes de radiografías de tórax de pacientes en hospitales de Guatemala, esto con el objetivo de entrenar al modelo con el tipo de radiografías que se obtienen en los laboratorios del país y de esta manera mejorar el rendimiento del modelo.
2. Fomentar los nuevos avances en el campo de la inteligencia artificial dentro de la Escuela de Ingeniería en Ciencias y Sistemas, así como la creación de talleres y laboratorios donde se imparta a los estudiantes el uso de nuevas herramientas para crear modelos avanzados de aprendizaje automático.
3. Brindar información a los profesionales de la salud sobre los posibles casos de uso de herramientas de *software* y de inteligencia artificial en la atención y cuidado de los pacientes.

BIBLIOGRAFÍA

1. HONGYUWANG, Yong. *ChestNet: A Deep Neural Network for Classification of Thoracic Diseases on Chest Radiography*, 2018. [en línea]. <<https://arxiv.org/pdf/1807.03058.pdf>>. [Consulta: 2 de marzo 2021].
2. IRVIN & RAJPURKAR. *et. al. CheXpert – A Large Chest Radiograph Dataset with Uncertainty Labels and Expert Comparison*, 2019. [en línea]. <<https://arxiv.org/pdf/1901.07031.pdf>>. [Consulta: 5 de marzo 2021].
3. METZ, Cade. *India fights diabetic blindness with help from A.I*, *The New York times*. [en línea]. <<https://www.nytimes.com/2019/03/10/technology/artificial-intelligence-eye-hospital-india.html>>. [Consulta: 11 de abril 2021].
4. _____ . *CheXNet – Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning*. [en línea]. <<https://arxiv.org/pdf/1711.05225.pdf> >. [Consulta: 1 de marzo 2021].
5. RAJPURKAR, Pranav. *AI For Medical Diagnosis, Online Course*. [en línea]. <<https://www.coursera.org/learn/ai-for-medical-diagnosis#syllabus>>. [Consulta: 15 de abril 2021].

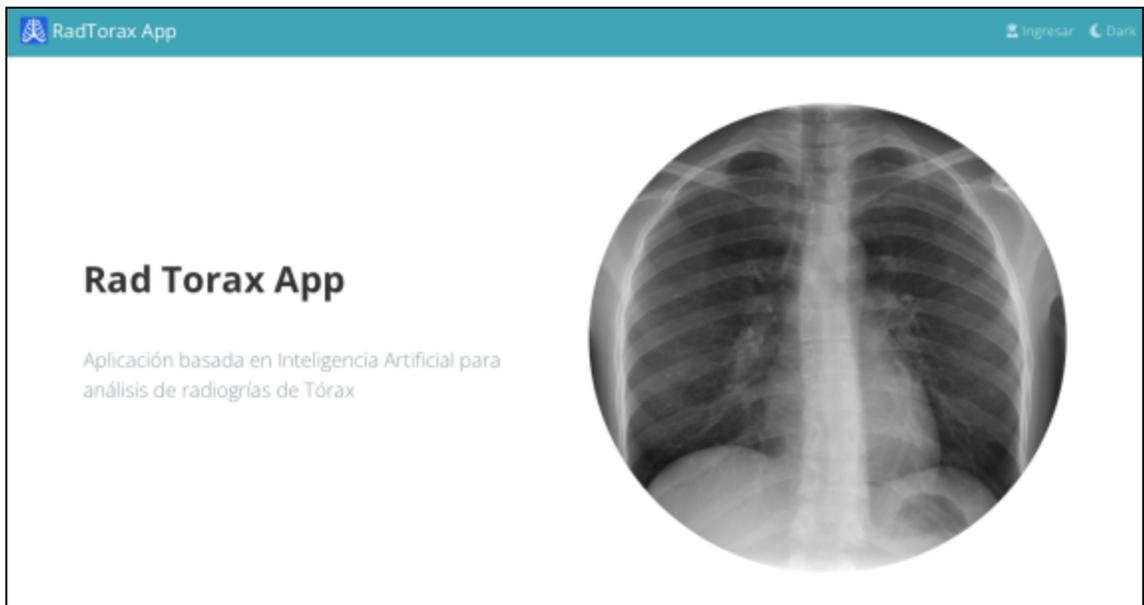
6. ZHUONING, Yuan. *et. al. Roboust Deep AUC Maximization: A New Surrogate Loss and Empirical Studies on Medical Image Classification*, 2020. [en línea]. <<https://arxiv.org/pdf/1711.05225.pdf>>. [Consulta: 22 de marzo 2021].

APÉNDICES

Manual del usuario

- Para utilizar la aplicación se debe ingresar al sitio web: <https://franklin-rad-torax-app.netlify.app/>.

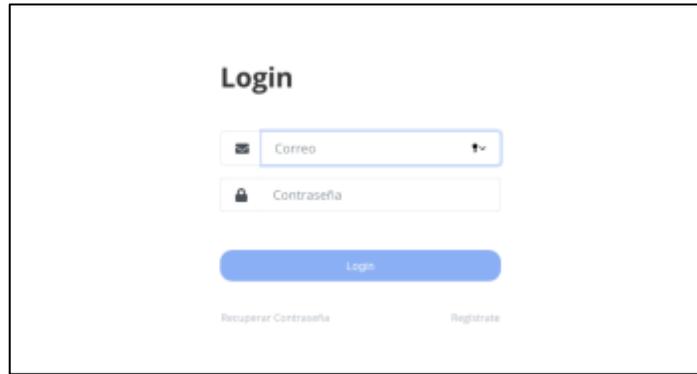
Apéndice 1. Rad Torax App



Fuente: elaboración propia, utilizando RadTorax App.

En primer lugar, el usuario debe autenticarse dentro de la plataforma, haciendo clic en “Ingresar”, posteriormente debe ingresar sus credenciales para iniciar sesión dentro de la plataforma.

Apéndice 2. Login



Login

Correo

Contraseña

Login

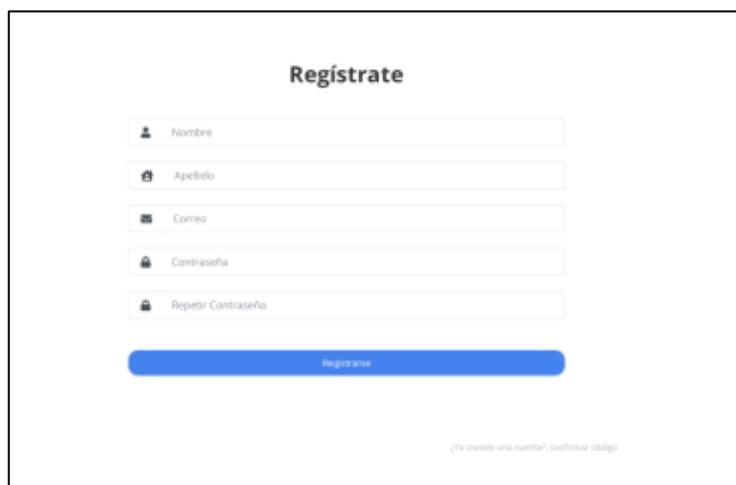
Recuperar Contraseña

Regístrate

Fuente: elaboración propia, utilizando RadTorax App.

Si el usuario aún no tiene una cuenta registrada, puede crear una haciendo clic en “Regístrate”. Posteriormente deberá llenar el formulario con los datos de registro que se le solicitan.

Apéndice 3. Regístrate



Regístrate

Nombre

Apellido

Correo

Contraseña

Repetir Contraseña

Regístrate

(Ya creaste una cuenta?, confirmar código)

Fuente: elaboración propia, utilizando RadTorax App.

Después de enviar sus datos para crear una cuenta, el usuario recibirá un correo electrónico a la dirección de correo que indicó en el formulario de registro. Este correo contiene un código de confirmación de cuenta que debe ser ingresado en el formulario que se despliega automáticamente al crear la cuenta.

Apéndice 4. **Confirmar cuenta**

Confirmar Cuenta

Revisa el correo electrónico que ingresaste y busca el código de confirmación de cuenta. Puede que también debas revisar en spam o correo no deseado. Ingresa el código a continuación.

✉ Correo

⌵ Código

Confirmar

Fuente: elaboración propia, utilizando RadTorax App.

Al completar la creación y confirmación de su cuenta, el usuario puede ingresar a la plataforma y tener acceso a las funcionalidades de análisis de radiografías de tórax dando clic en “Realizar análisis”.

Apéndice 5. Analizar imagen



Fuente: elaboración propia, utilizando RadTorax App.

El usuario puede elegir entre alguno de los ejemplos que se brinda, o bien subir su propia imagen dando clic en el recuadro para cargar archivos.

Apéndice 6. Elegir archivos



Fuente: elaboración propia, utilizando RadTorax App.

Al subir una imagen desde el ordenador o dispositivo móvil se le solicita al usuario que la seleccione del sistema de archivos de su dispositivo y posteriormente debe utilizar la herramienta de recorte de imagen para seleccionar el área importante de la imagen, en caso de que la imagen contenga bordes o contenido extra a la radiografía de tórax.

Apéndice 7. Ajustar imagen

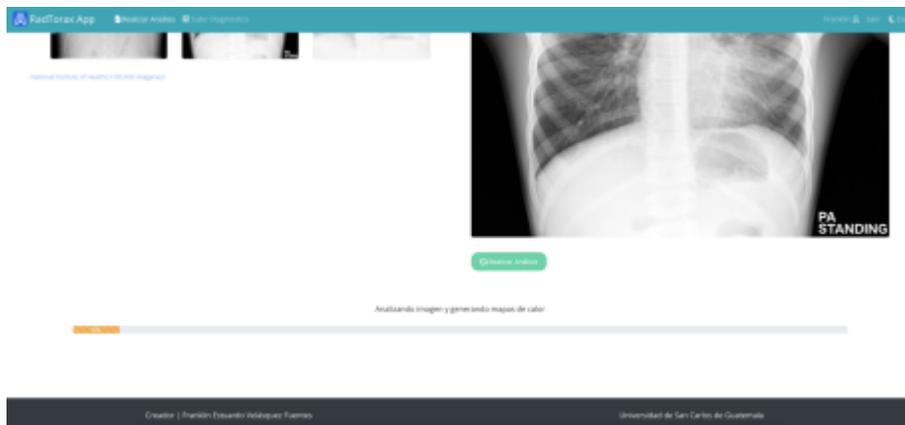


Fuente: elaboración propia, utilizando RadTorax App.

Una vez finalizados los pasos para seleccionar la imagen, el usuario podrá ver la imagen desplegada en el sitio web, debajo de la misma encontrará el botón con la etiqueta de “Realizar Análisis”. Presionando en dicho botón, el usuario puede enviar la imagen al servidor para ser procesada por el modelo de inteligencia artificial.

Al usuario se le mostrará una barra que indica el progreso de su solicitud, esto puede tardar entre 3 a 5 minutos, debido a que el modelo de reconocimiento de imágenes analiza la imagen enviada a través de las diversas capas de la red neuronal para calcular las probabilidades de las etiquetas, además de generar los mapas de calor en formato de imagen.

Apéndice 8. Analizar imagen



Fuente: elaboración propia, utilizando RadTorax App.

Una vez completado el análisis de la radiografía enviada, al usuario se le desplegará automáticamente una tabla mostrando los resultados del cálculo de las clases de enfermedades que soporta el modelo y la probabilidad de que la imagen enviada contenga dicha enfermedad.

De igual manera, se despliega una imagen que contiene la radiografía enviada, así como mapas de calor para las primeras tres clases de enfermedades más probables y las regiones de la imagen en la cual el modelo tuvo un mayor análisis y que fueron determinantes para el cálculo de las probabilidades.

Apéndice 9. Resultados del análisis

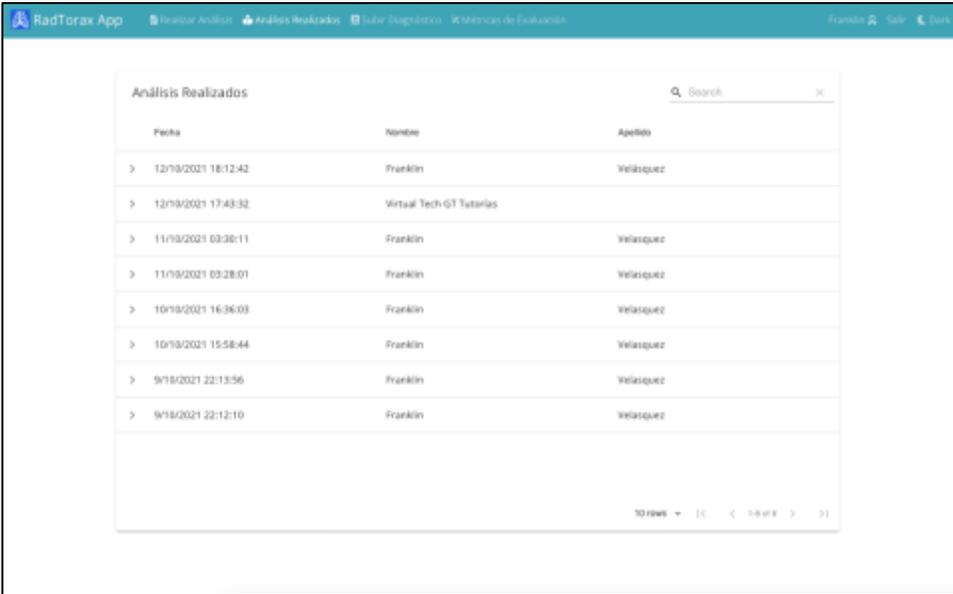


Fuente: elaboración propia, utilizando RadTorax App.

En la pestaña de “Análisis realizados”, el usuario puede ver un listado de los análisis de radiografías de tórax hechos por el modelo de inteligencia artificial. De igual manera, puede ordenarlos por fecha de creación y nombre de usuario.

Además, es posible que doctores especialistas en enfermedades pulmonares puedan corregir las predicciones dadas por el modelo, dando clic en editar y seleccionando las categorías reales a las cuales pertenece dicha radiografía, o bien los propios usuarios pacientes con un diagnóstico pueden ayudar subiendo sus resultados.

Apéndice 10. Análisis realizados

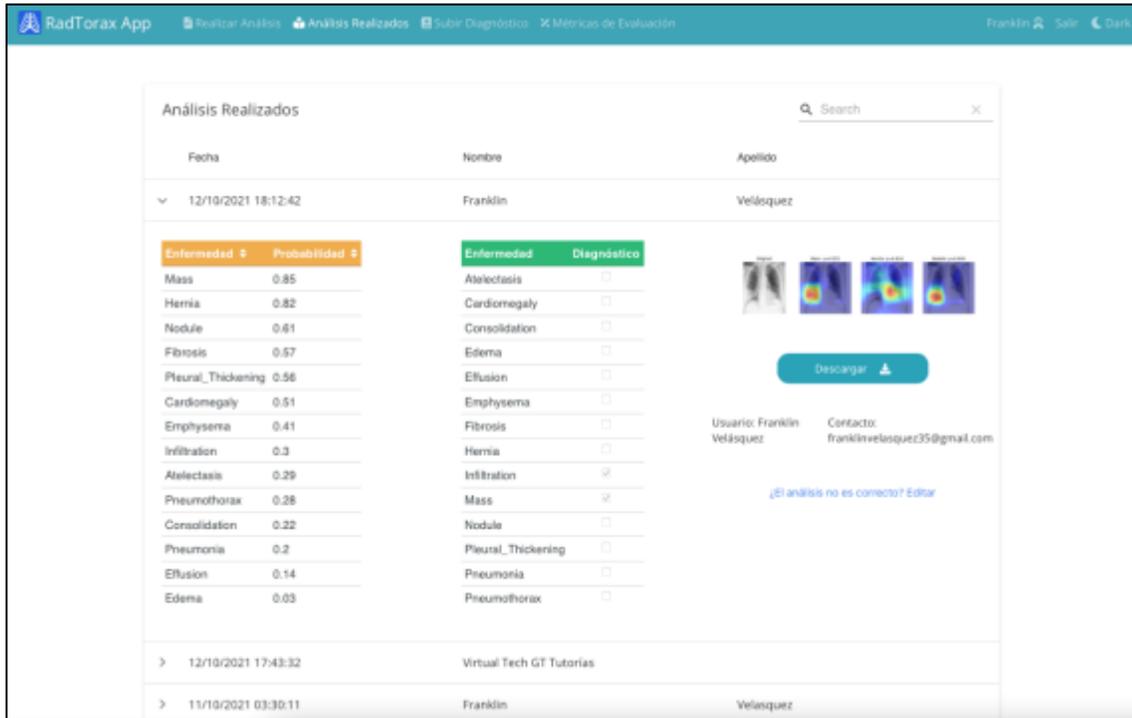


The screenshot shows the 'RadTorax App' interface with a navigation bar at the top containing 'Inicio', 'Realizar Análisis', 'Análisis Realizados', 'Cuentas Registradas', and 'Métricas de Evaluación'. The main content area is titled 'Análisis Realizados' and features a search bar. Below the search bar is a table with three columns: 'Fecha', 'Nombre', and 'Apellido'. The table contains nine rows of data, each with a chevron icon on the left. At the bottom right of the table, there is a pagination control showing '10 rows' and navigation arrows.

Fecha	Nombre	Apellido
> 12/19/2021 18:12:42	Franklin	Velasquez
> 12/19/2021 17:43:32	Virtual Tech GT Testarías	
> 11/19/2021 03:26:11	Franklin	Velasquez
> 11/19/2021 03:28:01	Franklin	Velasquez
> 10/19/2021 16:36:03	Franklin	Velasquez
> 10/19/2021 15:58:44	Franklin	Velasquez
> 9/18/2021 22:13:56	Franklin	Velasquez
> 9/18/2021 22:12:10	Franklin	Velasquez

Fuente: elaboración propia, utilizando RadTorax App.

Apéndice 11. Descargar resultados



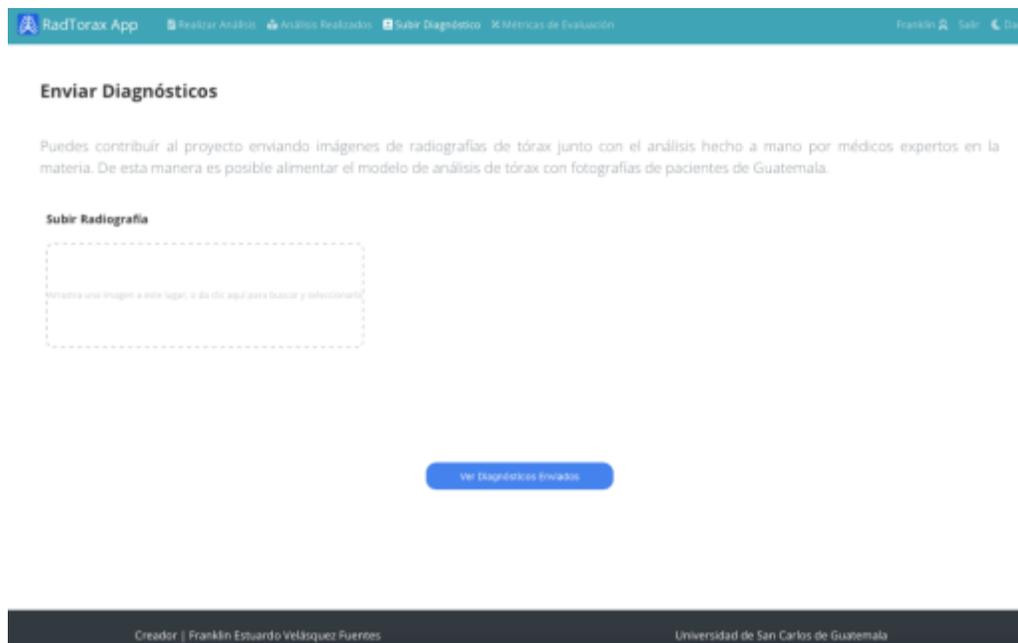
Fuente: elaboración propia, utilizando RadTorax App.

En la pestaña de “Enviar diagnósticos”, se encuentra el módulo que permite a los usuarios contribuir con nuevas imágenes de radiografías de tórax.

De esta manera, es posible recolectar un conjunto de datos con el cual entrenar posteriormente al modelo de reconocimiento de imágenes y de esta forma mejorar las predicciones y adecuarse al tipo de radiografías de tórax que se toman en los hospitales locales.

El usuario puede dar click en el recuadro para subir su radiografía de tórax desde cualquier dispositivo, así como seleccionar en la tabla las enfermedades a las cuáles pertenece su diagnóstico hecho por profesionales de la salud.

Apéndice 12. Enviar diagnóstico



Fuente: elaboración propia, utilizando RadTorax App.

Apéndice 13. Diagnósticos enviados

RadTorax App Realizar Análisis Análisis Realizados Subir Diagnóstico Métricas de Evaluación Franklin Salir Dark

Diagnósticos Enviados Search X

Fecha	Nombre	Apellido																														
10/10/2021 16:14:58	Franklin	Velasquez																														
<table border="1"><thead><tr><th>Enfermedad</th><th>Diagnóstico</th></tr></thead><tbody><tr><td>Atelectasis</td><td><input type="checkbox"/></td></tr><tr><td>Cardiomegaly</td><td><input type="checkbox"/></td></tr><tr><td>Consolidation</td><td><input type="checkbox"/></td></tr><tr><td>Edema</td><td><input type="checkbox"/></td></tr><tr><td>Effusion</td><td><input type="checkbox"/></td></tr><tr><td>Emphysema</td><td><input checked="" type="checkbox"/></td></tr><tr><td>Fibrosis</td><td><input checked="" type="checkbox"/></td></tr><tr><td>Hernia</td><td><input checked="" type="checkbox"/></td></tr><tr><td>Infiltration</td><td><input checked="" type="checkbox"/></td></tr><tr><td>Mass</td><td><input checked="" type="checkbox"/></td></tr><tr><td>Nodule</td><td><input checked="" type="checkbox"/></td></tr><tr><td>Pleural_Thickening</td><td><input checked="" type="checkbox"/></td></tr><tr><td>Pneumonia</td><td><input checked="" type="checkbox"/></td></tr><tr><td>Pneumothorax</td><td><input checked="" type="checkbox"/></td></tr></tbody></table>			Enfermedad	Diagnóstico	Atelectasis	<input type="checkbox"/>	Cardiomegaly	<input type="checkbox"/>	Consolidation	<input type="checkbox"/>	Edema	<input type="checkbox"/>	Effusion	<input type="checkbox"/>	Emphysema	<input checked="" type="checkbox"/>	Fibrosis	<input checked="" type="checkbox"/>	Hernia	<input checked="" type="checkbox"/>	Infiltration	<input checked="" type="checkbox"/>	Mass	<input checked="" type="checkbox"/>	Nodule	<input checked="" type="checkbox"/>	Pleural_Thickening	<input checked="" type="checkbox"/>	Pneumonia	<input checked="" type="checkbox"/>	Pneumothorax	<input checked="" type="checkbox"/>
Enfermedad	Diagnóstico																															
Atelectasis	<input type="checkbox"/>																															
Cardiomegaly	<input type="checkbox"/>																															
Consolidation	<input type="checkbox"/>																															
Edema	<input type="checkbox"/>																															
Effusion	<input type="checkbox"/>																															
Emphysema	<input checked="" type="checkbox"/>																															
Fibrosis	<input checked="" type="checkbox"/>																															
Hernia	<input checked="" type="checkbox"/>																															
Infiltration	<input checked="" type="checkbox"/>																															
Mass	<input checked="" type="checkbox"/>																															
Nodule	<input checked="" type="checkbox"/>																															
Pleural_Thickening	<input checked="" type="checkbox"/>																															
Pneumonia	<input checked="" type="checkbox"/>																															
Pneumothorax	<input checked="" type="checkbox"/>																															
10/10/2021 16:06:31	Franklin	Velasquez																														
12/10/2021 17:05:31	Virtual Tech GT Tutorías																															



Fuente: elaboración propia, utilizando RadTorax App.

