



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE ACAPULCO

**"SISTEMA DE ANÁLISIS DE IMÁGENES CON PYTHON PARA LA
DETECCIÓN DE CASOS COVID-19 A TRAVÉS DE IMÁGENES
RADIOLÓGICAS DE LOS PACIENTES"**

TITULACIÓN INTEGRAL
TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE
INGENIEROS EN SISTEMAS COMPUTACIONALES

PRESENTAN:
ROSAS FLORES JUAN JOSÉ
LOEZA SILVERIO JAIME ULISES

DIRECTOR DE TESIS
DR. JUAN JOSÉ BEDOLLA SOLANO

ACAPULCO, GRO. JUNIO, 2024.

Dedicatoria

Dedico esta tesis en particular a mis padres y a mi tío Juan José, quien tuvo un papel importante en mi desarrollo personal y me inculcó los valores de responsabilidad, disciplina y aspiraciones de éxito. Además, me dedico con mucho amor a mi familia, amigos y a todos los que me han apoyado y animado a seguir adelante.

También a mis profesores, quienes compartieron conmigo sus conocimientos y sabiduría a lo largo de mi vida académica para que pueda desarrollarme profesionalmente. A todos ustedes les dedico esta tesis.

Fuente: Rosas Flores Juan José

Dedico esta tesis especialmente a mi pequeña, María Regina Loeza Galeana, quien es mi razón de vivir, me llena de alegría todos los días, siendo mi mayor motivación para salir a delante para luchar todos los días por conseguir el éxito. También a mi madre María de Jesús Silverio Evangelista, por enseñarme a nunca rendirme y perseverar cuando hay adversidad.

Fuente: Loeza Silverio, Jaime Ulises

Agradecimiento

Agradecemos especialmente al Dr. Bedolla Solado Juan José por la confianza depositada en nosotros al encomendarnos este proyecto abierto y por reconocer nuestros conocimientos, habilidades, delicadeza, responsabilidad y diligencia.

También agradecemos de antemano al Tecnológico Nacional de México campus Acapulco por aceptar el sistema de información actual como proyecto de tesis y brindarnos la oportunidad de desarrollarnos profesionalmente para aprender y adquirir nuevas habilidades. Por último, pero no menos importante, nos gustaría expresar nuestro agradecimiento a nuestros compañeros de trabajo, amigos y familiares por su apoyo durante la realización de la tesis. También nos gustaría expresar un agradecimiento especial a nuestros padres por alentarnos siempre a seguir aprendiendo para que podamos ser mejores. Muchas gracias a todos.

Agradezco a Dios por ayudarme a elegir el camino correcto y darme la sabiduría que necesitaba para lograr esta meta en mi vida. Agradezco a mi madre María de Lourdes Flores Manzo a mi padre Juan Regulo Rosas Espinosa y a mi tío Juan José Flores Manzo, por ser el pilar fundamental de mi vida y por brindarme su comprensión y amor. También quiero agradecer a todos los que, directa o indirectamente, creyeron lo suficiente en mis habilidades para avanzar en mi trabajo.

Agradezco sinceramente a todos los miembros de mi familia, incluidos mis padres, hermanos y tíos.

También quiero agradecer a mí por siempre acompañarme durante mi carrera y ayudarme a superarme.

Fuente: Rosas Flores Juan José

Agradezco a mis padres, Armando y María de Jesús, gracias por todo su esfuerzo, por darme una vida digna y darme estudios, por todo su apoyo en cada meta en mi vida y por darme el claro ejemplo de cómo salir a delante de manera honrada. A mi familia, Nicole y María Regina, por darme motivación todos los días para vivir y salir adelante, por los momentos más felices de mi vida que los he vivido con ustedes, el amor que me dan todos los días, lo cual es mi pilar en mi día a día. A mi asesor, el Dr. Juan José

Bedolla Solano, gracias por su tiempo, paciencia y confianza depositada en mí para poder llevar a cabo este proyecto, sobre todo por su guía para poder alcanzar las metas propuestas . A todos mis amigos y familiares que creen en mí, agradezco mucho por sus consejos y apoyo, por todos los buenos momentos que me han dado, además por todas las enseñanzas de vida que me han ayudado a no rendirme.

Fuente: Loeza Silverio Jaime Ulises

Resumen

Esta tesis presenta el desarrollo de un sistema de análisis de imágenes mediante Python para la detección de COVID-19 a partir de imágenes radiológicas de pacientes. La aplicación web implementada utiliza técnicas de procesamiento de imágenes, divididas en fases de preprocesamiento, post-procesamiento y segmentación, aplicadas a un repositorio de imágenes.

El sistema maneja datos de pacientes con posibles casos de COVID-19 y permite el análisis de sus radiografías torácicas. Para acceder al sistema, se requiere un registro que garantice la privacidad del paciente.

Desarrollado con la metodología de Cascada, el sistema utiliza Python como lenguaje de programación backend, HTML, CSS y JavaScript para la interfaz web, y MySQL para almacenar datos, además de emplear Flask como *framework*. Este avance tecnológico proporciona una herramienta rápida y accesible para el diagnóstico de COVID-19, atendiendo la necesidad de diagnósticos oportunos, especialmente para pacientes con enfermedades crónicas y de menor riesgo.

Con su enfoque de código abierto y dominio público, se pretende ofrecer a hospitales y al gobierno de Acapulco una herramienta de diagnóstico preliminar basada en imágenes de pacientes con posibles casos de COVID-19. Además, se considera una opción más económica en comparación con las pruebas convencionales, como las PCR, lo que la hace más viable considerando las características demográficas de la región.

Palabras claves: COVID-19, PCR, Python, Flask, MySQL, Aplicación web, Preprocesamiento, Post-procesamiento, Segmentación.

Abstract

"This thesis presents the development of an image analysis system using Python for COVID-19 detection through radiological images of patients. The implemented web application utilizes image processing techniques, divided into pre-processing, post-processing, and segmentation phases applied to an image repository.

The system manages data from patients with potential COVID-19 cases and allows for the analysis of their chest radiographs. Access to the system requires registration to ensure patient privacy.

Developed with the Cascade methodology, the system employs Python as the backend programming language, HTML, CSS, and JavaScript for the web interface, and MySQL for data storage, alongside using Flask as a framework. This technological advancement provides a fast and accessible tool for COVID-19 diagnosis, addressing the need for timely diagnoses, especially for patients with chronic illnesses and lower risk.

With its open-source approach and public domain focus, it aims to offer hospitals and the Acapulco government a preliminary diagnostic tool based on patient images with potential COVID-19 cases. Furthermore, it is considered a more economical option compared to conventional tests like PCR, making it more viable considering the region's demographic characteristics."

Keywords: COVID-19, PCR, Python, Flask, MySQL, Web application, Preprocessing, Postprocessing, Segmentation.

Índice de Ilustraciones

Ilustración.1 “Vidrio despulido” bilateral. Las flechas señalan las áreas afectadas con un incremento en la atenuación. Tomado de: *Diagnóstico clínico-radiológico de neumonía por COVID-19 (SARS-CoV-2) en el Hospital Ángeles Lomas.* **¡Error! Marcador no definido.**

Marcador no definido.

Fuente: Lozano Z. H. et al (2022). *Acta médica Grupo Ángeles*, 18(2), 224-228. **¡Error! Marcador no definido.**

Ilustración 2. “Vidrio despulido” asimétrico. La flecha señala el territorio afectado. Tomado de: *Diagnóstico clínico-radiológico de neumonía por COVID-19 (SARS-CoV-2) en el Hospital Ángeles Lomas.* **¡Error! Marcador no definido.**

Ilustración 3. “Vidrio despulido” asimétrico bilateral. En el pulmón derecho en el segmento 6, y en el izquierdo en los segmentos 5 y 8. Tomado de: *Diagnóstico clínico-radiológico de neumonía por COVID-19.* **¡Error! Marcador no definido.**

Ilustración.4 Consolidación. Extensa zona de ocupación alveolar con formación de broncograma aéreo en territorio del lóbulo inferior izquierdo. Se muestran en planos axial y coronal. Tomado de: *Diagnóstico clínico-radiológico de neumonía por COVID-19 (SARS-CoV-2).* **¡Error! Marcador no definido.**

Ilustración 5. Patrón en “empedrado” (crazy-paving). La flecha muestra aspecto “en empedrado” asociado con “vidrio despulido”. La estrella muestra “vidrio despulido” así como opacidades lineales. **¡Error! Marcador no definido.**

Ilustración 7. TC de Tórax. Patrón de vidrio esmerilado, se observan áreas de parénquima aumentado de densidad sin borrar los vasos pulmonares. Corresponde

a relleno alveolar. Tomado de: INTERPRETACIÓN DE PATRONES RADIOLÓGICOS SUGERENTES DE..... **¡Error! Marcador no definido.**
COVID-19 EN PEDIATRÍA. Por Prado A, F. et al (2020), Neumología Pediátrica, 15(2), 330–338..... **¡Error! Marcador no definido.**

Ilustración 8. Radiografía de tórax de recién nacido de 27 días, SARS-CoV-2 (+). Discreto infiltrado intersticial hiliofugal tenue del lóbulo superior e inferior derecho de predominio basal. Tomado de: INTERPRETACIÓN DE PATRONES RADIOLÓGICOS SUGERENTES DE COVID-19 EN PEDIATRÍA. **¡Error! Marcador no definido.**

Por Prado A, F. et al (2020), Neumología Pediátrica, 15(2), 330–338 **¡Error! Marcador no definido.**

Ilustración 9. TC de tórax con áreas focales de vidrio esmerilado bilaterales y con consolidación y halo circulante. Tomado de: INTERPRETACIÓN DE PATRONES RADIOLÓGICOS SUGERENTES DE COVID-19 EN PEDIATRÍA. **¡Error! Marcador no definido.**

Fuente: Prado A, F. et al (2020), Neumología Pediátrica, 15(2), 3 **¡Error! Marcador no definido.**

Ilustración 10. En radiografía en decúbito se pueden evidenciar opacidades múltiples de consolidación multilobar bilateral. Signo de la silueta con los contornos del corazón y con las hojas diafragmáticas. Tomado de: INTERPRETACIÓN DE PATRONES RADIOLÓGICOS SUGERENTES COVID-19 EN PEDIATRÍA. **¡Error! Marcador no definido.**

Fuente: Prado A, F. et al (2020), Neumología Pediátrica, 15(2), 330–338..... **¡Error! Marcador no definido.**

<i>Ilustración 11. Modelo de calidad del producto de software</i>	<i>76</i>
<i>Fuente: (ISO 25000, 2022).....</i>	<i>76</i>
<i>Ilustración 12. Base de datos de aplicación web. Fuente: Propia (2023).</i>	<i>83</i>
<i>Ilustración 13. Diagrama entidad relación de la base de datos. Fuente: Propia(2023).</i>	<i>86</i>
<i>Ilustración 14 . Diagrama relacional de la base de datos. Fuente: Propia(2023).</i>	<i>87</i>
<i>Ilustración 15. Diagrama de secuencia de inicio de sesión. Fuente: Propia (2023)</i>	<i>104</i>
<i>Ilustración 16. Diagrama de secuencia de los Diagnósticos. Fuente: Propia (2023).....</i>	<i>105</i>
<i>Ilustración 17 . Carpeta creada para el proyecto. Fuente: Propia (2023)</i>	<i>107</i>
<i>Ilustración 18. Instalación exitosa de TensorFlow. Fuente: Propia (2023).....</i>	<i>108</i>
<i>Ilustración 19. Esquema “COVID-19”, creado para guardar los datos de la aplicación. Fuente: Propia (2023).....</i>	<i>109</i>
<i>Ilustración 20. Diagrama de la base de datos utilizada. Fuente: Propia (2023).....</i>	<i>110</i>
<i>Ilustración 21. Diseño de la interfaz de inicio. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).</i>	<i>114</i>
<i>Ilustración 22. Menú principal. Autoría propia. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).</i>	<i>115</i>
<i>Ilustración 23. Lista de los pacientes que ya han tenido diagnóstico. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).....</i>	<i>116</i>
<i>Ilustración 24. Visualización de la imagen. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).....</i>	<i>117</i>
<i>Ilustración 25. Registro del paciente. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).....</i>	<i>118</i>

Ilustración 26. Registrar usuarios. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).....	119
Ilustración 27. Inicio de sesión para administrador y médico. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).....	120
Ilustración 28. Inicio de sesión. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	121
Ilustración 29. Diseño de la interfaz de perfil. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	122
Ilustración 30. Diseño de la interfaz de editar perfil. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	123
Ilustración 31. Diseño de la interfaz actualizar contraseña. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).....	124
Ilustración 32. Módulo de pacientes. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).....	124
Ilustración 33. Módulo médico. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	125
Ilustración 34. Módulo estudios. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	125
Ilustración 35. Barra de navegación del administrador. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	126
Ilustración 36. Barra de navegación del usuario. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	126
Ilustración 37. Dictámenes positivos a COVID-19. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	127

Ilustración 38. Dictámenes Negativos a COVID-19. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	128
Ilustración 39. Ver dictamen del paciente. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	129
Ilustración 40. Dictamen de COVID-19 del paciente. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	130
Ilustración 41. Resultado del inicio de sesión. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	137
Ilustración 42. Resultado de interfaz de bienvenida con el nombre del usuario logeado. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	138
Ilustración 43. Resultado de la interfaz de Registrar usuario. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	139
Ilustración 44. Resultado de la interfaz editar perfil. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	139
Ilustración 45. Resultado de la interfaz Registro paciente. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	140
Ilustración 46. Resultado de la interfaz editar paciente. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	141
Ilustración 47. Visualización de la lista de médicos registrados al sistema. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	142
Ilustración 48. Visualización del paciente registrado al sistema. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	143
Ilustración 49. Subir la imagen del tórax del paciente. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	144

Ilustración 50. Visualización de la imagen subida exitosamente al sistema. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	144
Ilustración 51. Interfaz dictamen. Fuente, adquirida del analizador de imagen COVID-19 (2023).	145
Ilustración 52. Dictamen del paciente. Fuente, adquirida del analizador de imagen COVID-19 (2023).	146
Ilustración 53. Filtro de dictamen (Resultado positivos). Fuente, adquirida del analizador de imagen COVID-19 (2023).	147
Ilustración 54. Filtro de dictamen (Resultado negativos). Fuente, adquirida del analizador de imagen COVID-19 (2023).	147
Ilustración 55. Resultado del diagnóstico a través de la red neuronal convolucional. Fuente, adquirida del analizador de imagen COVID-19 (2023).....	148
Ilustración 56. Resultado del diagnóstico a través del procesamiento de imágenes. Fuente, adquirida del analizador de imagen COVID-19 (2023).....	149
Ilustración 57. Reporte general. Fuente, adquirida del analizador de imagen COVID-19 (2023).....	150
Ilustración 58. Reporte genera de negativo a COVID-19I. Fuente, adquirida del analizador de imagen COVID-19 (2023).	151
Ilustración 59. Respuesta del sistema a una contraseña incorrecta. Fuente: Propia(2023).....	152
Ilustración 60. Resultado en probabilidad por medio de la Técnica, bordes y contornos. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	156

Ilustración 61. Resultado en probabilidad por medio de la Técnica, operaciones morfológicas. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).....	157
Ilustración 62. Resultado en probabilidad por medio de la técnica de Ruido. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	157
Ilustración 63. Resultado en probabilidad por medio de la técnica de Bordes y contornos. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	158
Ilustración 64. Resultado arrojado por la inteligencia artificial implementada. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).	158
Ilustración 65. Prueba de rendimiento utilizando las herramientas para desarrollador de Google Chrome. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).....	159

Índice de Tablas

<i>Tabla 2.1: Descripción de los hallazgos en TC en la literatura mundial y nuestra experiencia.....</i>	11
<i>Tabla 3.2 Esquema metodológico de procesamiento de imágenes.....</i>	49
<i>Tabla 3.5 Técnicas de procesamiento de imágenes.....</i>	58
<i>Tabla 3.6.1 Recursos de software utilizados en proyecto.....</i>	66
<i>Tabla 3.6.2. Costo en recurso de hardware.....</i>	67
<i>Tabla 3.7 Costo en recursos humano para el desarrollo del proyecto.....</i>	67
<i>Tabla 3.7.1. Costo total del desarrollo del proyecto.....</i>	67
<i>Tabla 3.9. Características de norma ISO 9126.....</i>	69
<i>Tabla 3.9.1. Métrica de propiedades.....</i>	73

<i>Tabla 3.9.2. Métrica de niveles</i>	<i>75</i>
<i>Tabla 4.2.1 Dictamen, fuente: Propia(2023).</i>	<i>84</i>
<i>Tabla 4.1.2 Admin, fuente: Propia(2023).</i>	<i>85</i>
<i>Tabla 4.1.3. Paciente Fuente: Propia(2023).</i>	<i>86</i>
<i>Tabla 4.2 Funcionamiento de los módulos de la aplicación.</i>	<i>87</i>
<i>Tabla 4.3.2. Funcionamiento de la aplicación con submódulos</i>	<i>89</i>
<i>Tabla 4.3.2 Técnicas de procesamiento de imágenes.....</i>	<i>91</i>
<i>Tabla 5.1 Prueba de inicio de sesión al sistema.</i>	<i>152</i>

Índice de gráficas

<i>Grafica 1. Resultados en porcentajes de probabilidad de padecer COVID-19 arrojados por las técnicas de procesamiento de imágenes probadas. Fuente: Propia.</i>	156
--	-----

Índice de cuadros

Cuadro 1. <i>Renderizar vista. Fuente. Propia. Adquirida del analizador COVID-19(2023).</i>	108
Cuadro 2. <i>Controladores. Fuente. Propia. Adquirida del analizador COVID-19(2023).</i>	109
Cuadro 3. <i>Código de iniciar sesión. Fuente. Propia. Adquirida del analizador COVID-19(2023).</i>	129
Cuadro 4. <i>Código de registrar. Fuente. Propia. Adquirida del analizador COVID-19(2023).</i>	130
Cuadro 5. <i>Código de la técnica de Umbralización. Fuente. Propia. Adquirida del analizador COVID-19(2023).</i>	132
Cuadro 6. <i>Código de la inteligencia artificial. Fuente. Propia. Adquirida del analizador COVID-19(2023).</i>	133
Cuadro 7. <i>Código de la conexión a la base de datos. Fuente. Propia. Adquirida del analizador COVID-19(2023).</i>	134

Índice general

Dedicatoria	i
Agradecimiento	ii
Resumen	iv
Abstract	v
Índice de Ilustraciones	vi
Índice de Tablas	xii
Índice de gráficas	xiv
Índice de cuadros	xv

Capítulo 1: Contexto General	1
1.1 Introducción	1
1.2 Problemática	2
1.3 Objetivos	3
1.3.1 Objetivo General.....	3
1.3.2 Objetivos específicos.....	4
1.4 Justificación	5
1.5 Hipótesis	7
Capítulo 2: Marco Teórico	9
2.1 Antecedentes	9
2.2 Fundamentos teóricos	19
2.2.1 Lenguajes de programación front-end.....	19
2.2.3 Lenguajes de programación Back-end.....	23
2.2.4 Herramientas de desarrollo de software.....	24
2.2.5 Framework de programación.....	27
2.2.6 Fundamentos teóricos de COVID-19.....	29
2.2.7 Inteligencia artificial.....	31
2.2.8 Procesamiento de imágenes.....	33
2.2.9 Adquisicióno captura de imágenes.....	33

2.2.10 Preprocesamiento de imágenes	34
2.2.11 Segmentación	36
2.2.12 Post-procesamiento de imagen	38
2.2.13 Operaciones morfológicas	38
2.2.13 Metodología de investigación	39
2.3 Trabajos Relacionados.....	42
Capítulo 3: Metodología.....	47
3.1 Elección de la metodología.....	47
3.2 Esquema metodológico de procesamiento de imágenes	48
3.2.1 Captura de imágenes	50
3.2.2 Preprocesamiento de imágenes	51
3.2.3 Segmentación	52
3.2.4 Post-procesamiento de imágenes	53
3.2.5 Evaluación	54
3.3 Tipo de investigación	55
3.4 Selección de herramienta e instrumento de medición	57
3.5 Procedimientos	58
3.5.1 Técnicas de procesamiento de imágenes	58
3.6 Red neuronal utilizando DenseNet 121	62
3.7 Análisis de factibilidad	65
3.7.1 Factibilidad técnica	65
3.7.2 Factibilidad económica	67
3.8 Recursos Humanos	67
3.7.1 Costo total	67
3.9 Análisis de costos y beneficios	68
3.9.1 Evaluación del sistema	68

Capítulo 4: Desarrollo	82
4.1 Herramientas	82
4.2. Base de datos de la aplicación	83
4.2.1 Diccionario de datos	83
Tabla Dictamen	83
Tabla paciente.....	85
4.2.3 Modelo Entidad-Relación	86
4.2.3 Modelo relacional	87
4.3 Funcionamiento de la aplicación	87
4.3.2 Submódulos de la aplicación.....	89
4.4 Descripción detallada de las técnicas de procesamiento de imágenes ...	95
4.4.1 Preprocesamiento	95
4.4.2 Segmentación	97
4.4.3 Post-Procesamiento	98
4.5 Requerimientos del sistema	99
4.5.1 Específicos	99
4.5.2 No específicos.....	102
4.6 Diagrama de secuencias.....	104
4.7 Procedimientos realizados	105
4.7.1 Instalación de entorno de desarrollo y librerías	106
4.7.2 Creación de la base de datos en <i>MySQL</i>	108
4.7.3 Creación de rutas	110
4.7.4 Controladores para la base de datos	111
4.8 Diseño de interfaz	113
4.9 Codificación	130
Capítulo 5: Pruebas y Resultados	137

5.1 Prueba de interfaz.....	137
5.2 Pruebas de datos e integridad de base de datos.....	151
5.2 Resultados de evaluación aplicando la Norma ISO.....	153
5.3 Resultados de diferentes técnicas de análisis de imágenes.....	155
5.4 Prueba de rendimiento.....	159
5.4 Cumplimiento de los objetivos.....	160
<i>Discusiones.....</i>	162
<i>Conclusión.....</i>	164
<i>Recomendaciones.....</i>	166
<i>Glosario.....</i>	168
<i>Referencia Bibliográfica.....</i>	171
<i>Anexo.....</i>	182
Anexo A: Entrevista.....	182

Capítulo 1: Contexto General

1.1 Introducción

La pandemia de COVID-19 ha dejado un impacto significativo, con cifras alarmantes de contagios y defunciones. Desde el inicio de esta investigación, se han registrado 7.363.037 contagios y 343.609 defunciones, según datos de la Secretaría de Salud (2022). La continua evolución del virus SARS-CoV-2 y la emergencia de múltiples variantes subrayan la importancia del reconocimiento temprano de la enfermedad y la necesidad de diagnósticos precisos para evitar la transmisión y proporcionar atención médica oportuna.

El desarrollo de nuevas estrategias de prevención y control de epidemias ha llevado al uso crucial de sistemas de inteligencia artificial. Estas tecnologías han sido fundamentales en la creación de herramientas y modelos computacionales que pueden prever tanto la cantidad de personas infectadas como la gravedad de sus síntomas. Su aplicación no solo se limita a esta pandemia, sino que también ofrece la posibilidad de anticipar y prevenir futuras enfermedades, salvaguardando incontables vidas humanas (Gamero & Chamorro; 2021).

Por otra parte, el procesamiento de imágenes digitales es un campo que abarca diversas áreas, desde el reconocimiento de imágenes en 2D, 3D y secuencias, hasta el análisis, la manipulación y la transmisión de imágenes. En este contexto, la creación de una aplicación web de dominio público y código abierto se presenta como una oportunidad para identificar casos potenciales de COVID-19 a través de

diagnósticos de imágenes digitales, empleando técnicas de inteligencia artificial, específicamente redes neuronales.

Los objetivos de esta investigación comprenden la identificación de técnicas de procesamiento de imágenes, el diseño de una herramienta que las respalde, su desarrollo utilizando tecnologías web basadas en ingeniería de software, y la implementación de una propuesta de asistencia al sector salud relacionada con la identificación de la enfermedad COVID-19.

En este sentido, esta tesis se enfoca en el desarrollo de una herramienta que analiza imágenes digitales de radiografías torácicas para ofrecer una evaluación preliminar de la condición de una persona, incluyendo la detección de una posible infección por COVID-19; con ello, ofrecer una alternativa de desarrollo tecnológico y sostenible en el sector salud.

1.2 Problemática

La necesidad de un diagnóstico preciso y oportuno del SARS-CoV-2 continúa siendo un desafío crítico en la gestión de la salud pública. A pesar de los avances en las pruebas de diagnóstico, persisten limitaciones en su disponibilidad, tiempo de obtención de resultados y costo, lo que podría retrasar la implementación de medidas preventivas y el tratamiento adecuado para los pacientes. La falta de herramientas automatizadas específicas para la identificación rápida de la enfermedad COVID-19 impide un diagnóstico ágil y eficiente.

Esta situación conlleva riesgos significativos, ya que la demora en el diagnóstico y tratamiento adecuado no solo impacta la salud del paciente, sino que también aumenta la posibilidad de propagación del virus. La dependencia de pruebas convencionales, aunque efectivas, carece de la agilidad necesaria para abordar la urgencia de la situación actual.

Se hace imperativo desarrollar un sistema automatizado basado en análisis de imágenes que permita una identificación temprana y precisa de casos de COVID-19, brindando una opinión inicial a los profesionales médicos. Esta herramienta no solo agilizaría el proceso de diagnóstico, sino que también optimizaría el uso de recursos y apoyaría la implementación de medidas de tratamiento y prevención más eficaces, reduciendo así el impacto negativo de la propagación del virus en la salud pública.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar una aplicación de diagnóstico por imágenes centrada en pacientes con y sin la enfermedad del COVID-19, que amplíe las opciones diagnósticas existentes al proporcionar una evaluación preliminar sobre la contagiosidad de un paciente. Esta solución innovadora se diseñará para ser accesible y fácil de usar, ofreciendo un análisis rápido y sencillo a través de una interfaz amigable para el paciente, contribuyendo así a una detección más precisa y ágil de la enfermedad, complementando los métodos de diagnóstico convencionales.

1.3.2 Objetivos específicos

- Comprender la fisiopatología de la infección pulmonar causada por el virus SARS-CoV-2 para identificar patrones específicos en las imágenes radiológicas.
- Conocer, desarrollar y aplicar algoritmos de procesamiento de imágenes digitales para detectar características distintivas de la infección por COVID-19 en radiografías torácicas, permitiendo una evaluación precisa y rápida.
- Explorar y aplicar diversas técnicas de procesamiento de imágenes digitales, incluyendo transformaciones, segmentación y análisis de texturas, para seleccionar mejores alternativas de uso, precisión y confiabilidad del diagnóstico por imágenes.
- Diseñar y construir una plataforma con lenguajes de modelado de datos y tecnologías web intuitiva y fácil de usar que permita a los usuarios cargar imágenes radiológicas, obtener resultados de diagnóstico preliminares y ofrecer recomendaciones claras para una atención médica adicional.
- Evaluar el sistema automatizado de diagnóstico de COVID-19 mediante normas de desarrollo y desempeño de software para realizar una propuesta dirigida al sector salud eficiente y accesible basado en análisis de imágenes, contribuyendo a una detección más ágil y económica de la enfermedad.

1.4 Justificación

Es fundamental identificar y diagnosticar rápidamente los casos positivos de *COVID-19* para aislarlos, así como iniciar un estudio clínico tomando en cuenta el historial clínico del paciente con el fin de determinar el grado de complicaciones que puede llegar a presentar el paciente, cabe mencionar que se tienen diferentes poblaciones y cada una tiene un porcentaje distinto de mortalidad, un claro ejemplo nos lo dice el autor L. Aguirre(2021) “En los pacientes fallecidos predominaron enfermedad cardiovascular (incluye hipertensión) el 42,9%, diabetes y cáncer en un 14, 3% en ambas comorbilidades”. Lo que significa que dependiendo del estado de salud del paciente se puede determinar si está en riesgo o no. La detección precoz es muy importante, por ende, se debe tener como prioridad las pruebas en casos sospechosos y en personas que tuvieron contacto con estos, el autor M. Pérez nos dice lo siguiente,

“Los autores alertan sobre la importancia de la detección temprana de los pacientes en etapa asintomática debido a que el cuadro clínico puede tener una evolución tórpida inesperada y llevar al paciente a la muerte, aunque tenga una carga viral baja que no represente peligro inminente para su vida, es necesario que sea evaluado como corresponde. Algunos de los infectados, confirmados a partir de la realización de la prueba, pueden ser asintomáticos, por lo que los autores aconsejan que las personas que tuvieron contacto cercano con estos pacientes deben ser aisladas y ser monitoreadas por un tiempo determinado para descartar la infección.”

El objetivo es un diagnóstico precoz que permita un mejor manejo del paciente (incluyendo un diagnóstico y tratamiento adecuado), seguimiento del paciente, aplicación de medidas de prevención y control, y vigilancia epidemiológica.

No todos los encuentros tienen el mismo riesgo. Se puede distinguir entre relaciones cercanas o intrusivas y relaciones casuales o no intrusivas. Un contacto estrecho es alguien que ha estado cerca de un caso de positividad durante más de 15 minutos, sin protección, durante las 48 horas desde el inicio de los síntomas o el momento en que se realizó la prueba diagnóstica (si el caso positivo no tenía síntomas).

El precio de una prueba de antígenos va de los 200 a los 600 pesos (entre 10 y 30 dólares), aunque una PCR llega a costar unos 3.200 pesos (poco menos de 160 dólares). Para su familia, eso significa ponerse una mascarilla sobre el cubrebocas y varias capas de ropa para alcanzar uno de los *test* gratuitos que aplica la Secretaría de Salud local. De igual manera, estos lugares mantienen una mala administración que provoca la espera en la toma de pruebas, haciendo que estas personas convivan entre ellas y que en su momento las personas que no tengan el virus se contagien en esas horas de espera.

Al ver la problemática, contribuimos con una solución, puesto que es de suma importancia la detección de un buen diagnóstico para poder frenar el contagio entre personas y estas no tengan que llevar a cabo pruebas que requieren de tiempo y dinero que no todos tienen. Hoy en día existen diferentes pruebas para detectar *COVID-19*, entre las más famosas tenemos: RT-PCR, test de antígenos rápidos y el *test* anticuerpos, la primera siendo la más confiable con un porcentaje de entre 85-90% de efectividad, por otro lado, existe otro tipo de prueba

radiológica que es la Tomografía computarizada, según el diario de salud de América sus autores señalaron que “el PCR en tiempo real es el *Gold estándar* y método más empleado; sin embargo, tiene algunas desventajas es costoso, necesita personal experto en Microbiología molecular y medidas de bioseguridad, equipos y personal específico, además tiene un porcentaje relativo de falsos negativos. En comparación con la PCR, la tomografía computarizada de tórax puede ser un método más fiable, práctico y rápido para diagnosticar y evaluar el *COVID-19*, especialmente en el área de la epidemia”; según un estudio con 1.000 pacientes que se ha publicado en la revista '*Radiology*' la tomografía computarizada (TC) de tórax es mejor que las pruebas de laboratorio para el diagnóstico de coronavirus o *COVID-19*, demostró tener una sensibilidad del 88% (en 1014 pacientes) y en otro 98% en caso de *COVID-19* sospechosos aumentando al 97% en casos asociados a PCR positiva. Por lo que se puede afirmar que es una gran opción como prueba, siendo menos costosa y teniendo un buen porcentaje de sensibilidad.

1.5 Hipótesis

- El desarrollo y aplicación de un sistema para diagnosticar con altas probabilidades de encontrar posibles casos de *COVID-19* en una variedad de imágenes escaneadas permitirá agilizar los procesos de detección de la enfermedad y al mismo tiempo mejorará la administración de la información en los hospitales.
- Administrar la información proporcionada y procesada por el sistema ayudará a

los expertos en el área a eficientar el diagnóstico; y por consiguiente incrementar la atención que permita valorar mayores pacientes y casos de COVID-19 u otras enfermedades que se detecten en el diagnóstico de imágenes.

- El desarrollo de un sistema automatizado que identifique las afectaciones en las radiografías de tórax a través de imágenes digitales, permitirá detectar la enfermedad; y al mismo tiempo ofrecer un tratamiento oportuno con alternativa de desarrollo tecnológico y estratégico para mejorar la calidad de vida y el desarrollo en el sector salud.

Capítulo 2: Marco Teórico

2.1 Antecedentes

Las pruebas de diagnóstico de *COVID-19* son esenciales para detectar el virus, comprender la epidemiología, informar el manejo de casos y detener la propagación de la enfermedad. La familia *Coronaviridae*, que incluye los cuatro géneros (alfa, beta, delta y gamma), así como numerosos subgéneros y especies que se encuentran en una variedad de animales y humanos, incluye el coronavirus. (Ramos et al., 2020).

Las pruebas que se podrían utilizar si hubiera sospecha del nuevo beta-coronavirus SARS-CoV-2 causante de *COVID-19* son las siguientes:

- Amplificación de ácidos nucleicos (NAAT) mediante técnica *RT-PCR*.
- Secuenciación viral: las pruebas para detectar secuencias de genes virales, secuencias de *ARN* polimerasa viral y secuencias de nucleoproteínas se han implementado con éxito para la confirmación de virus y son útiles para mostrar mutaciones del genoma viral.
- No se recomienda el cultivo viral debido a preocupaciones de bioseguridad.
- Estudios serológicos
- Las pruebas de diagnóstico rápido o pruebas de antígenos pueden encontrar las proteasas del virus.

- Las pruebas realizadas. Estas pruebas no se utilizan para diagnosticar *COVID-19*. Estas pruebas para estudios determinan si una persona ha tenido previamente una infección aunque no presente, síntomas.

Al igual que otras patologías, los estudios de imagen son ayudas diagnósticas; en el caso de la infección por *SARS-CoV-2*, la información que brindan es una variedad de patrones y signos radiológicos debido al daño que sufren los pulmones y el área vaginal durante el curso de la enfermedad. La tomografía axial computarizada es más confiable para determinar el tipo y extensión de la parenquimatosa pulmonar. El papel del ecógrafo en situaciones de emergencia es más evidente en adultos que en niños, donde existe una buena correlación con las imágenes de toxinas.

Aunque la radiografía tórax tiene sus limitaciones, es el estudio de imagen de elección para la evaluación inicial de pacientes con sospecha de proceso pulmonar infectado. En el estudio DIAGNÓSTICO CLÍNICO-RADIOLOGICO DE NEUMONÍA POR *COVID-19* (*SARS-CoV-2*) en el Hospital Los Ángeles Lomas en el año 2022 se hicieron 800 pruebas, de las cuales 228 dieron positivo a *COVID-19*. De estos pacientes, se realizaron 108 tomografías computarizadas, que encontraron al menos 58 anormalidades y 50 normalidades. De los 58 casos anormales, tres pacientes tuvieron resultados definitivamente negativos de la PCR en tiempo real, y uno de ellos tuvo un resultado de PCR positivo para el virus de la influenza-B.

Finalizando con los datos de la Tabla 2.1, la descripción de los síntomas más comunes encontrados en pacientes con tomografía computarizada y su correlación con reportes previos en la literatura, además de incluir la tabla con las ilustraciones del estudio (1, 2, 3, 4, 5 y 6), se puede observar en la tabla 2.2.



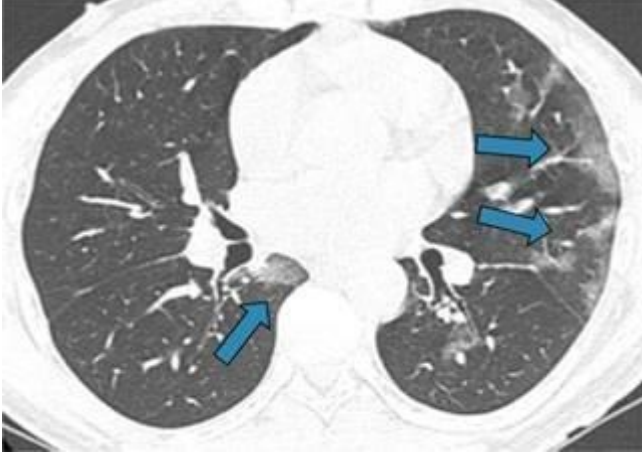
Tabla 2.1: Descripción de los hallazgos en TC en la literatura mundial y nuestra experiencia

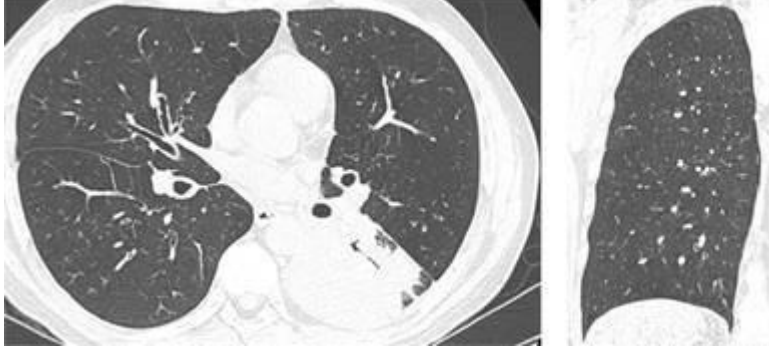

Hallazgos por tomografía computarizada	Frecuencias de alteraciones informadas	Frecuencia de alteraciones encontradas en esta serie
Opacidades “vidrio despulido” (<i>groundglass</i>)	86	97
Consolidación	29	44
Patrón en “empedrado” (<i>crazy-paving</i>)	19	13
Opacidades lineales	14	10
Cavitación	0	0
Nódulos no calcificados	0	2
Derrame pleural	0	0
Adenopatías	0	0
Distribución bilateral	76	60
Distribución periférica	33	55

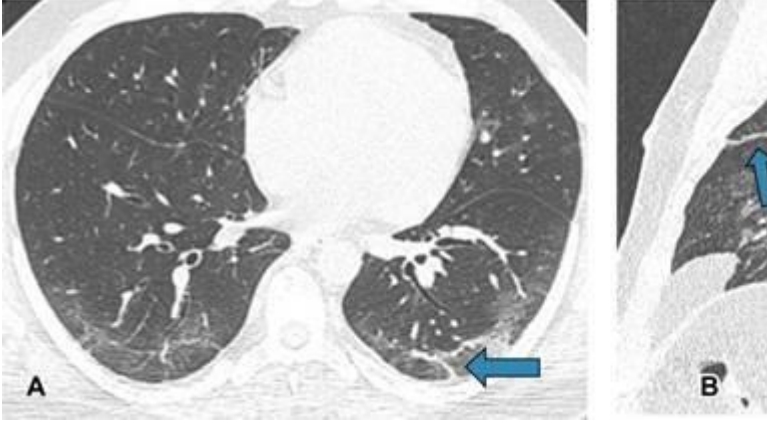
Fuente:(Lozano et al. 2020).

Tabla 2.2: Imágenes radiológicas de los estudios

Nombre	Ilustración	Descripción
		n

<p>Vidrio despulido” bilateral</p>		<p>Las flechas señalan las áreas afectadas con un incremento en la atenuación .</p>
<p>“Vidrio despulido” asimétrico</p>		<p>La flecha señala el territorio afectado.</p>
<p>Vidrio despulido” asimétrico bilateral.</p>		<p>En el pulmón derecho en el segmento 6, y en el izquierdo en los segmentos 5 y 8.</p>

<p>Consolidación.</p>		<p>Extensa zona de ocupación alveolar con formación de broncograma aéreo en territorio del lóbulo inferior izquierdo.</p>
<p>Patrón en “empedrado” (crazy-paving).</p>		<p>). La flecha muestra aspecto “en empedrado” asociado con “vidrio despulido”. La estrella muestra “vidrio despulido” así como opacidades lineales.</p>

<p>opacidades lineales</p>		<p>opacidades lineales (flechas) de localización periférica en lóbulo inferior izquierdo en plano axial. (B) Reconstrucción sagital en lóbulo superior izquierdo.</p>
----------------------------	--	---

Fuente:(Lozano et al. 2020).

Por otro lado, el estudio INTERPRETACIÓN DE PATRONES RADIOLOGICAS SUGERENTES DE COVID-19 EN PEDIATRIA (Prado et al., 2020) describe cómo la enfermedad progresa desde una infección viral respiratoria (Etapa I) hasta un ataque viral más grave en el que los principales El órgano blanco es el pulmón (Etapa II), hasta la actualidad, donde la enfermedad progresa en correlación con una respuesta inflamatoria secundaria que es (Etapa III). Además, existe un daño significativo en los espacios alveolocapilar y endotelial de la microvascular pulmonar.

Describiendo con ella las observaciones posteriores en cada radiografía, describiendo específicamente las firmas de pacientes pediátricos positivos a COVID-19.

1. Vidrio esmerilado.

Corresponde a un tenue aumento de la densidad parenquimatosa pulmonar que típicamente no oblitera los márgenes de los vasos pulmonares ni las paredes bronquiales. Este fenómeno se explica por una ocupación aérea parcial, como se muestra en la ilustración 7. Obedece a que el parénquima pulmonar se llena de líquido (edema) o de células inflamatorias.

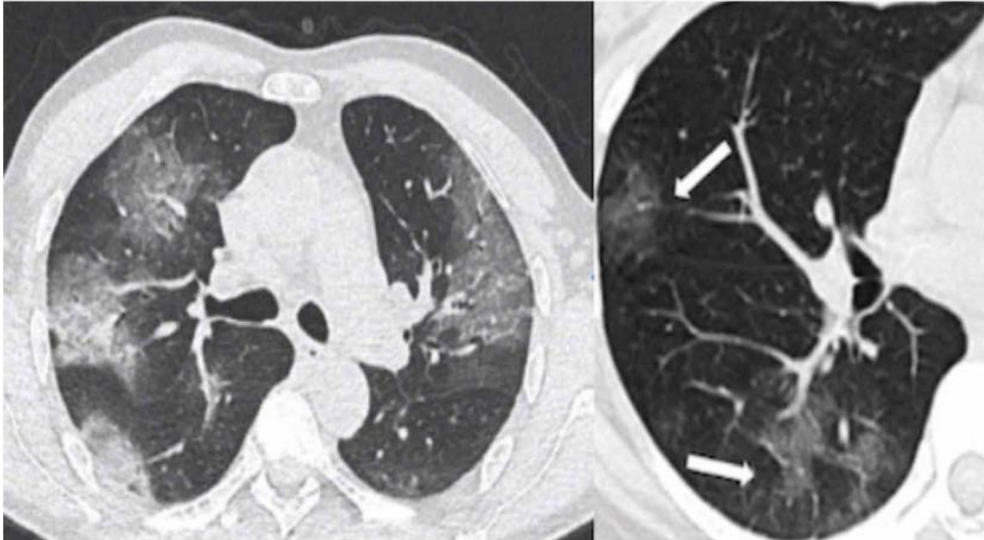


Ilustración 1TC de Tórax. Patrón de vidrio esmerilado, se observan áreas de parénquima aumentado de densidad sin borrar los vasos pulmonares. Corresponde a relleno alveolar. Tomado de: INTERPRETACIÓN DE PATRONES RADIOLÓGICOS SUGERENTES DE

2. Patrón en empedrado (crazy paving appearance)

Este tipo de acuerdo consiste en un patrón de vidrio esmerilado al que se le agregan imágenes lineales de engrosamiento intersticial e interlobulillar que crean una apariencia de material maleable (Ilustración 8).



Ilustración 2. Radiografía de tórax de recién nacido de 27 días, SARS-CoV-2 (+). Discreto infiltrado intersticial hiliopulmonar tenue del lóbulo superior e inferior derecho de predominio basal. Tomado de: INTERPRETACIÓN DE PATRONES RADIOLÓGICOS SUGERENTES DE COVID-19 EN

3. Consolidación más halo circulante.

Relleno alveolar es confluyente y completo de los espacios aéreos rodeados (circundante) por densidades en vidrio *halo-ready*, como se muestra en la ilustración 9. Este patrón puede aparecer en vasculitis u otras lesiones microangiopáticas. Se ha informado que los adultos tienen trombosis de vasculitis muy sugestiva en la microcirculación pulmonar. Estos factores pueden ayudar a explicar el extenso compromiso parenquimatoso pulmonar no exudativo que se desarrolla con SDRA en particular, en el que la distensibilidad pulmonar está menos comprometida que en otras etiologías virales.

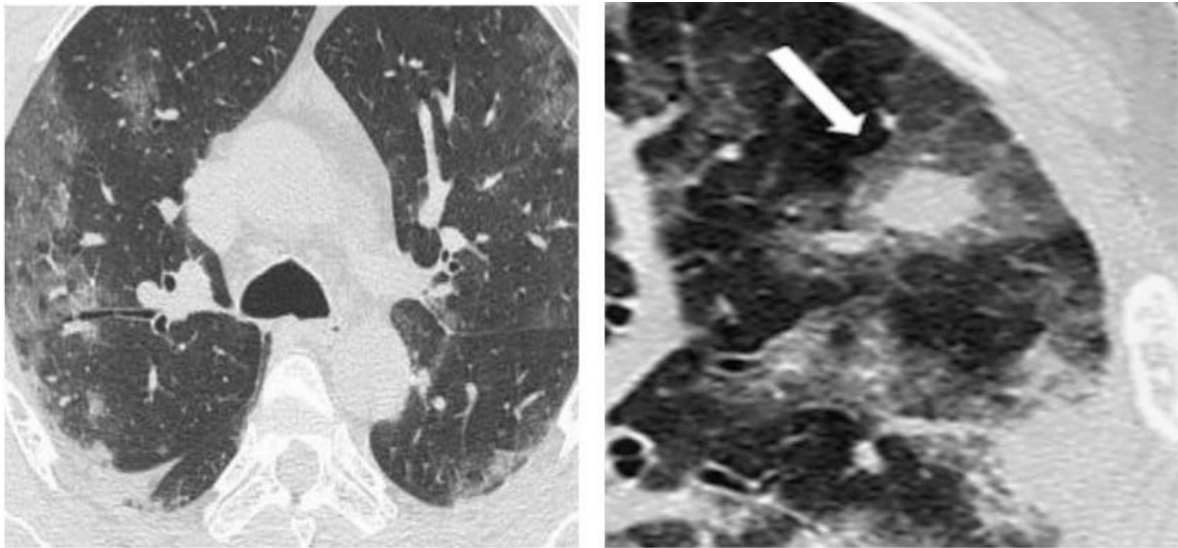


Ilustración 3TC de tórax con áreas focales de vidrio esmerilado bilaterales y con consolidación y halo circulante. Tomado de: INTERPRETACIÓN DE PATRONES RADIOLÓGICOS SUGERENTES DE COVID-19 EN PEDIATRÍA.

1. Consolidación multifocal, bilateral y en áreas preferentes de decúbito

Consolidación multilobar bilateral extensa con aspecto de pulmón blanco con signo de la silueta con los contornos del corazón y con las hojas diafragmáticas (Ilustración 10). Se presenta coexistencia con vidrio esmerilado, broncograma aéreo y engrosamiento de los septos Inter lobares. La representación de estos cambios se asocia con compromiso muy agudo y una insuficiencia respiratoria aguda hipoxémica severa.



Ilustración 4. En radiografía en decúbito se pueden evidenciar opacidades múltiples de consolidación multilobar bilateral. Signo de la silueta con los contornos del corazón y con las hojas diafragmáticas.

A partir de estos resultados, podemos identificar las metodologías más eficaces para detectar los efectos del virus en los pulmones. Además, en el ámbito de la salud, los modelos de aprendizaje automático están adquiriendo un rol crucial en la medicina de precisión, facilitando decisiones terapéuticas basadas en datos provenientes de imágenes médicas, estudios genómicos y otras variables clínicas. Estos modelos son capaces de detectar patrones sutiles, trazas de la evolución de la enfermedad y respuestas al tratamiento, lo que permite predecir el riesgo y la evolución futura de la enfermedad en pacientes individuales. En el campo médico, el aprendizaje automático permite la introducción de nuevas métricas y observaciones cuantitativas, además de fomentar el descubrimiento de nuevos conocimientos sobre patologías (Lands G., 2021).

Gracias al aprendizaje automático, se han identificado nuevos patrones dinámicos de enfermedades, los cuales están relacionados con la progresión de estas y la

respuesta terapéutica correspondiente. Este avance facilita la medición de cambios sutiles en el tejido pulmonar y sus características imagenológicas, transformando estos datos en un conocimiento renovado sobre las huellas de enfermedades, que se manifiestan en patrones distintivos observables en los pacientes.

2.2 Fundamentos teóricos

2.2.1 Lenguajes de programación front-end

Programación

Corrales (2015) menciona que la programación se refiere a: la actividad de la creación de programas; conjuntos de instrucciones que explican un algoritmo escrito en un lenguaje de programación y que, mediante la ejecución de tales instrucciones por la computadora, permiten desempeñar un trabajo determinado.

El hecho de tener un sujeto y un objeto (la computadora), junto con lo anterior, conduce un proceso de comunicación y organización de procesos que determinan tres requisitos previos a la actividad de programación.

Lenguaje de programación

Un programa se escribe con instrucciones en un lenguaje de programación, el cual, a su vez, está definido por su sintaxis, que establece e indica las reglas de escritura (la gramática), y por la semántica de los tipos de datos, instrucciones, definiciones, y todos los otros elementos que constituyen un programa. Un lenguaje de programación

es un caso particular del lenguaje informático; este último permite hacer programas, pero también describir datos, configuraciones físicas y protocolos de comunicación entre equipos y programas (Juganaru Mathieu, 2012: 5).

Front-end

El aspecto del software con el que interactúan los usuarios se conoce como front-end en el diseño de software (Nieto, 2016). El front-end de una página web se compone de texto, imágenes, botones, colores y otros elementos. Los lenguajes de programación utilizados en el desarrollo del sistema se detallan en este apartado:

HTML

Básicamente, los documentos escritos en *HTML* constan de texto y etiquetas. Las etiquetas permiten definir el formato del texto, el título que mostrará en la barra de título del navegador, los elementos multimedia que aparecerán incrustados en el documento, pero que se almacenan en archivos externos, etc. (Ferrer et al; 2003)

para poder darle un estilo más agradable al documento *HTML* es recomendable utilizar las Hojas de Estilo en Cascada (*CSS*) el cual es un lenguaje que maneja el diseño y presentación de las páginas *web*, es decir, cómo lucen cuando un usuario las visita (Santos, D; 2022). Con el que puedes crear reglas para decirle a su sitio *web* cómo desea que aparezca la información al separar los comandos para diseñar elementos (como fuente, color, tamaño, etc.) de los comandos para configurar el contenido. Además, puede crear formatos específicos que se pueden utilizar para transmitir sus

ideas y proporcionar una experiencia visual más agradable para los usuarios del sitio *web*.

CSS3

CSS utiliza las reglas asignadas a elementos *HTML*, un documento *HTML* completo o varios de ellos. Estas reglas son procesadas por el navegador web cuando carga el archivo *HTML*. Para poder generar una regla tenemos que tener en claro los componentes de esta, los cuales según el autor Santos (2022) son:

- *Selector CSS*. Este indica la parte del documento donde se aplica la regla. Y al selector le siguen una o más declaraciones dentro de corchetes. Existen varias formas de escribir un selector, es decir, a cada elemento *HTML* a modificar se menciona por su nombre.
- *Declaración CSS*. Dentro de un par de corchetes se contienen una o más declaraciones CSS. Una de estas declaraciones le dice al navegador qué estilo brindar al elemento seleccionado gracias a sus dos aspectos: una propiedad y un valor. Cada declaración termina con un punto y coma (;).
- *Propiedad CSS*. Como la primera parte de una declaración, una propiedad CSS le indica al navegador cuál característica de estilo del elemento debe cambiarse. Existen muchas propiedades CSS que afectan diferentes cosas: color, tamaño, fuente, forma o locación en una página. Una propiedad se acompaña de al menos un valor y estos elementos siempre aparecen separados por dos puntos (:).

- *Valor CSS*. Cada propiedad tiene un paquete de valores, los cuales especifican el estilo de la propiedad. Algunas de sus propiedades pueden ser: color, ancho, largo, la familia de fuente, etc.

JavaScript

Pérez (2007) considera que el lenguaje de *JavaScript* no demanda compilación. Este proyecto fue impulsado en la compañía *Netscape Communications* a cargo de Brendan Erich. Su realización se basaba en la plataforma *Java*, aunque *JavaScript* es un lenguaje que se dirige a objetos que son empleados generalmente en las páginas. Los navegadores actualizados pueden disponer este lenguaje y de todas sus funciones.

Según Dimes (2015), es un lenguaje de programación de computadoras dinámicas que se usa con frecuencia en navegadores web para controlar cómo se comportan las páginas web y cómo interactúan con los usuarios. permite la comunicación asincrónica y tiene la capacidad actualizar partes de una página web o incluso reemplazar completamente su contenido.

JavaScript (JS) es un lenguaje de programación ligero, interpretado y orientado a objetos con funciones de primera clase, y mejor conocido como el lenguaje de programación para las páginas *Web*, pero también se utiliza en muchos entornos que no son de navegador. Es un lenguaje de scripts que es dinámico, multiparadigma, basado en prototipos y admite estilos de programación orientados a objetos, imperativos y funcionales. (Mozilla, 2022). El cual se ejecuta en el lado del cliente de

la *web*, y se puede utilizar para estilizar/programar cómo se comportan las páginas *web* cuando ocurre un evento.

2.2.3 Lenguajes de programación Back-end

El back-end es responsable del procesamiento de los datos recibidos desde el front-end, que también controla la vista y conecta la base de datos. (Nieto, 2016).

Bases de datos

Según el autor del Busto (2012), las bases de datos no relacionales son un sistema de almacenamiento de información que se caracteriza por no usar el lenguaje *SQL* para las consultas. La respuesta a la necesidad de gestionar volúmenes masivos de información surge de la base de datos *NoSQL*.

Normalmente, la estructura de una base de datos relacional se basa la organización de la información a través de documentos. Este tipo de *database* están pensadas para ofrecer mayor escalabilidad horizontal y no tienen identificadores que permitan establecer relaciones entre diferentes conjuntos de datos. Cada uno de los documentos almacenados en la base de datos incluye todos los atributos del elemento, por lo que resultan muy útiles a la hora de guardar información poco estructurada o de la que no se tiene un esquema claro de inicio. (Tablado, F.; 2020)

Python

Python es un lenguaje de programación creado por Guido van Rossum a principios de los años 90, cuyo nombre está inspirado en el grupo de cómicos ingleses “Monty Python”. Es un lenguaje similar a *Perl*, pero con una sintaxis muy limpia y que favorece un código legible. Se trata de un lenguaje interpretado o de *script*, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos (González Duque; 2015: 7). Por diseño, *Python* implementa una sintaxis deliberadamente simple y legible y un modelo de programación altamente coherente. Como lo atestigua un eslogan en una reciente conferencia de *Python*, el resultado neto es que *Python* parece "ajustarse a su cerebro", es decir, las características del lenguaje interactúan de manera consistente y limitada y se derivan naturalmente de un pequeño conjunto de conceptos básicos (Lutz; 2009: 4-5).

2.2.4 Herramientas de desarrollo de software

Para el desarrollo del sistema se implementaron varias herramientas que ayudaron en cada una de las etapas, desde el diseño de los diagramas hasta la codificación del sistema. Para la elección de las herramientas fue necesario investigar las características de cada una de ellas y las ventajas que ofrecen este sobre otras herramientas.

MySQL

El sistema gestor de bases de datos (SGBD, también conocido como DBMS por sus siglas en inglés) MySQL es muy conocido y ampliamente utilizado por su simplicidad

y su extraordinario rendimiento. Por su facilidad de uso y tiempo de inicio rápido, es una opción atractiva tanto para aplicaciones comerciales como de entretenimiento. Sin embargo, carece de algunas características avanzadas disponibles en otros SGBD del mercado. Contar con una gran estabilidad y un rápido desarrollo son dos beneficios adicionales, además de su distribución gratuita en Internet bajo la licencia GPL.

MySQL es un SGBD que ha ganado popularidad por una serie de atractivas características:

- Está desarrollado en C/C++.
- Se distribuyen ejecutables para cerca de diecinueve plataformas diferentes.
- La API se encuentra disponible en C, C++, *Eiffel*, Java, Perl, PHP, *Python*, Ruby y TCI.
- Está optimizado para equipos de múltiples procesadores.
- Es muy destacable su velocidad de respuesta.
- Se puede utilizar como cliente-servidor o incrustado en aplicaciones.
- Cuenta con un rico conjunto de tipos de datos
- Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD en cada caso concreto.
- Su administración se basa en usuarios y privilegios.
- Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.

- Sus opciones de conectividad abarcan TCP/IP, sockets UNIX y sockets NT, además de soportar completamente ODBC.
- Los mensajes de error pueden estar en español y hacer ordenaciones correctas con palabras acentuadas o con la letra 'ñ'.
- Es altamente confiable en cuanto a estabilidad se refiere.

(Casillas et. al, 2017)

React

Dentro de este lenguaje tenemos la librería *React* diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. También es una aplicación de una sola página. Por tanto, en lugar de enviar una petición al servidor cada vez que hay que renderizar una nueva página, el contenido de la página se carga directamente desde los componentes de *React*. Esto conduce a una renderización más rápida sin recargas de la página. (Kinsta, 2022).

Next js

Una de las aplicaciones de este *framework* es la optimización de imágenes que nos proporciona optimizaciones de imagen automáticas con construcciones instantáneas. La optimización de imágenes es una potente función preintegrada en *Next.js* porque la gestión y optimización de imágenes requiere muchas configuraciones, y la optimización manual de imágenes puede suponer un peaje en tu tiempo productivo. (Copes, 2020)

Rest api

Las *API* son conjuntos de definiciones y protocolos que se utilizan para diseñar e integrar el software de las aplicaciones. Suele considerarse como el contrato entre el proveedor de información y el usuario, donde se establece el contenido que se necesita por parte del consumidor (la llamada) y el que requiere el productor (la respuesta) (RedHat,2020).

La *API RESTful* es una interfaz que dos sistemas de computación utilizan para intercambiar información de manera segura a través de Internet. La mayoría de las aplicaciones para empresas deben comunicarse con otras aplicaciones internas o de terceros para llevar a cabo varias tareas. Donde La transferencia de estado representacional (*REST*) es una arquitectura de software que impone condiciones sobre cómo debe funcionar una *API*. En un principio, *REST* se creó como una guía para administrar la comunicación en una red compleja como Internet. Es posible utilizar una arquitectura basada en *REST* para admitir comunicaciones confiables y de alto rendimiento a escala. Puede implementarla y modificarla fácilmente, lo que brinda visibilidad y portabilidad entre plataformas a cualquier sistema de *API*.(Amazon, 2022)

2.2.5 Framework de programación

En cuanto a la definición utilizada en esta investigación, se tiene que es un esquema de reutilización del software conformado por componentes y relaciones entre estos, por ejemplo: la abstracción de clases, objetos o componentes que la conforman; además, provee diferentes componentes de conexión a base de datos, como

controladores para conexión directa (*MySQL, SQL Server, Oracle*) o de manera general, mediante el estándar *ODBC (Open DataBase Connectivity)* (Zabala y Ochoa, 2008). Existen varios tipos de *Frameworks*, usualmente los enfocados en el desarrollo Web, ofrecen una capa de controladores de acuerdo con el patrón MVC (Model View Controller) (Cui y otros, 2009) facilitando así la integración con otras herramientas para la implementación de las capas de negocio y presentación.

Framework Flask

Flask es un “micro” *Framework* escrito en *Python* y concebido para facilitar el desarrollo de Aplicaciones *Web* bajo el patrón MV. Con el que no se necesita una infraestructura con un servidor *web* para probar las aplicaciones, sino de una manera sencilla se puede correr un servidor web para ir viendo los resultados que se van obteniendo (Domingo J.; 2017).

La página oficial *Flask*(2017), nos describe que al devolver *HTML* (el tipo de respuesta predeterminado en *Flask*), todos los valores proporcionados por el usuario representados en la salida deben escaparse para protegerse de los ataques de inyección. *escape()*, se puede utilizar manualmente. Se omite en la mayoría de los ejemplos por razones de brevedad, pero siempre debe tener en cuenta cómo está utilizando datos que no son de confianza.

2.2.6 Fundamentos teóricos de COVID-19

A finales del 2019, el mundo se vio afectado debido a una enfermedad que provocó múltiples muertes a causa de un brote que provoca neumonía severa. Teniendo su inicio en china en la ciudad de Wuhan (Díaz & Toro, 2020) este virus denominado *SARS-CoV-2* mayormente conocido como *COVID-19*, es un síndrome respiratorio agudo severo de tipo 2 que ataca directamente a los pulmones, se ha evidenciado que este virus genera síntomas en ciertas personas, en la cual existen pacientes asintomáticos que no presentan indicios de estar contagiados con la enfermedad. Por otro lado, los pacientes sintomáticos evidencian síntomas leves de fiebre, tos seca, dolor de cabeza, mareos, dolor abdominal, diarrea, náusea y vómito. Además, puede provocar infecciones severas desencadenando insuficiencia renal aguda y síndrome de dificultad de respirar provocando la muerte (Melián, Calcumil, Boin, & Carrasco, 2020). De acuerdo a distintos estudios, el *COVID-19* se transmite directa o indirectamente de persona a persona por medio del contacto. Entre las técnicas para diagnosticar si una persona está contagiada por el virus, están las pruebas rápidas PCR, en la cual este tipo de prueba lo que hace es detectar el material genético del virus en la persona, por lo general no siempre da un pronóstico eficiente, dando falsos resultados en donde si existen evidencia de síntomas, en esos casos es recomendable realizar otra prueba (Melián, Calcumil, Boin, & Carrasco, 2020).

Rayos X

Las radiografías de rayos X nos permiten observar la parte interna de nuestro cuerpo a través de ondas electromagnéticas, por lo general este tipo de radiografías nos muestra una imagen en tono blanco y negro. Su utilización es muy variada, ya que se pueden realizar radiografías de distintas partes de nuestro cuerpo, como son radiografías abdominales, radiografías de huesos, tórax, dientes, extremidad inferior o superior, de la mano, pelvis, cráneo, cuello, entre otros (Moore, Dailey, & Agur, 2013).

Este tipo de prueba se puede realizar rápidamente y sin dolor, pero nadie puede leer los resultados que se muestran en las imágenes de rayos X; en cambio, hay expertos disponibles que pueden interpretar los resultados y proporcionar el diagnóstico adecuado.

Radiografías de Tórax

Las radiografías de tórax o torácica permiten realizar diagnósticos de los pulmones, en donde se detecta si existen infecciones o afecciones pulmonares como son la presencia de neumonía, tuberculosis o cáncer a los pulmones, incluso por la pandemia mundial por *COVID-19* las radiografías de tórax también están siendo utilizadas para detectar si una persona presenta esta enfermedad (Maguiña, Murguía, Verona, Gomero, & Véliz, 2021).

2.2.7 Inteligencia artificial

Cuando se habla de inteligencia artificial se refiere a la capacidad de las máquinas de imitar al comportamiento humano, de acuerdo a (Takeyas, 2007) “La IA es una rama de las ciencias computacionales encargada de estudiar modelos de cómputo capaces de realizar actividades propias de los seres humanos con base en dos de sus características primordiales: el razonamiento y la conducta”. Por otro lado, también se dice que la inteligencia artificial “tiene como objeto que los ordenadores hagan la misma clase de cosas que puede hacer la mente” (Boden, 2016). En otras palabras, la inteligencia artificial busca solucionar procedimientos repetitivos, en los que una máquina lo haga por nosotros.

El uso de la visión artificial, el reconocimiento de voz, el procesamiento del lenguaje natural, los sistemas expertos, la robótica y el aprendizaje automático son solo algunos ejemplos de las tecnologías en las que la inteligencia artificial está presente actualmente en la mayoría de las industrias.

Inteligencia artificial en la Medicina

La medicina en los últimos años ha logrado grandes avances en los cuales se aprovechan las distintas técnicas de inteligencia artificial existentes, “formadas por una serie de algoritmos lógicos suficientemente entrenados a partir, de los cuales las máquinas son capaces de tomar decisiones para casos concretos a partir de normas generales” (Avila, Mayer, & Quesada, 2020). Esto quiere decir que la inteligencia artificial ofrece grandes beneficios en el área de la medicina, ya que a través de ella

se puede realizar diagnósticos precisos y rápidos, mediante el procesamiento de datos médicos se han automatizado procesos como la detección de patologías y realizar tratamientos con anticipación a los pacientes con un índice de error mínimo (Lugo Reyes, Guadalupe, & Murata, 2014).

Es vital recordar que el proceso de diagnóstico involucra una variedad de conocimientos previos, habilidades y experiencias para reconocer los síntomas y diagnosticar la enfermedad para evaluar la condición del paciente. Por lo tanto, para que los sistemas basados en inteligencia artificial funcionen bien, deben ser resistentes y tener una gran cantidad de datos de entrada.

Redes neuronales

De acuerdo a (Matich, 2001) las redes neuronales “son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico”. Por lo tanto, se puede decir que las redes neuronales artificiales emulan el comportamiento de las redes neuronales humanas.

Redes neuronales convolucionales

Este tipo de redes neuronales se caracterizan por tener un alto potencial de procesamiento de datos. El funcionamiento de las redes neuronales convolucionales se asemeja a la forma en que los seres humanos perciben el mundo a través de los

ojos. En donde “si vemos una cara, la reconocemos porque tiene orejas, ojos, una nariz, cabello, etc. Entonces, para decidir si algo es una cara, lo hacemos como si tuviéramos unas casillas mentales de verificación de las características que vamos marcando” (Torres, 2018).

2.2.8 Procesamiento de imágenes

El procesamiento de imágenes es un método para convertir una imagen en formato digital y realizar algunas operaciones en ella, con el fin de obtener una imagen mejorada o extraer información útil de ella. Es un tipo de distribución de señal en el que la entrada es una imagen, como un cuadro de video o una fotografía, y la salida puede ser una imagen o características asociadas con esa imagen. Por lo general, el sistema de procesamiento de imágenes incluye el tratamiento de imágenes como señales bidimensionales mientras se les aplican métodos de procesamiento de señales ya establecidos. (Alaa y El Abidne; 2021: 6)

2.2.9 Adquisición o captura de imágenes

Según el autor Escalante (2006) La adquisición de la imagen está a cargo de algún transductor o conjunto de transductores que, mediante la manipulación de la luz o de alguna otra forma de radiación que es emitida o reflejada por los cuerpos, se logra formar una representación del objeto dando lugar a la imagen. Ejemplos: el ojo humano, sensores de una cámara fotográfica o de vídeo, tomógrafos.

Por otra parte, podemos decir que la adquisición de imagen es el proceso por el cual se obtiene una imagen

2.2.10 Preprocesamiento de imágenes

Un preprocesamiento de imágenes de entrada es un conocimiento base dentro del manejo *Deep Learning* en el procesamiento de los datos. De manera que toda buena *data scientist* debe tener en cuenta sus procesos para facilitar el análisis del Big Data. (KeepCoding, 2022)

Modelo de color

El color, como cualquier otro recurso, también tiene su técnica y está sometido a ciertas leyes, y según la aplicación que se desea, se trabaja con distintos modelos de color. Los modelos de color describen los colores que se ven en las imágenes digitales e impresas y el trabajo con ellos. Permiten, no sólo establecer un espacio único común a todos los equipos que forman parte de la cadena de adquisición y reproducción de color, sino que también permiten simular cómo lucirá la imagen y su color en otro dispositivo de la cadena; así por ejemplo, podemos ver en la pantalla del computador cómo saldrá la imagen impresa en el papel, después de que haya pasado por las tintas que se usan normalmente en prensa y con diferentes papeles, permitiendo un trabajo de edición de imagen y color mucho más sencilla y fiel a los resultados finales 2 . Cada modelo de color como, por ejemplo, RGB, CMYK o HSB representa un método diferente (y por lo general, numérico) de descripción de los colores. (Soto,2000)

RGB

Este modelo es un subespacio del espacio euclidiano conformado por el cubo unitario. (González y Woods, 2002), Los tres colores primarios del sistema de color aditivo son rojo, verde y azul. El modelo de representación cromática conocido como *RGB* se basa únicamente en los colores rojo, verde y azul para crear cada tono de color.

Ruido

Es la información no deseada que contamina la imagen el ruido se encuentra presente en absolutamente todas las imágenes digitales existentes, Se considera ruido a la aparición aleatoria de señales que no pertenecen a la imagen original, es decir en otras palabras, la aparición de píxeles o conjunto de píxeles de un color que no corresponde con el de la imagen. Se puede definir el ruido como cambios que presentan los píxeles en una imagen. De alguna forma modifican las características de dicha imagen (bordes, nitidez, brillo, contraste). (Bentacourt et al., 2003)

Binarización

Es una técnica que consiste en la realización de un barrido en la matriz de la imagen digital, por medio de bucles o recursividad, con el fin de que el proceso produzca la reducción de la escala de grises a dos únicos valores. (Magro, 2013)

Umbralización

Uno de los métodos que se emplean para generar la segmentación de una imagen es el umbralado. El umbralado obtiene el promedio de una imagen a escala de grises, la cual determina y resalta los elementos con mayor o con menor intensidad del fondo

de la imagen. Después de la detección de los elementos que afectan el material industrial se resaltan estos y se destacan a través de la binarización de la imagen. Este consiste en cambiar en blanco los elementos encontrados y el resto de la imagen en negro, permitiendo así ubicar y destacar la presencia de los agentes que están dañando al material que es objeto de estudio. (Rosales Méndez et al; 2011: 1)

2.2.11 Segmentación

La segmentación subdivide una imagen en sus partes constituyentes u objetos, con el fin de separar las partes de interés del resto de la imagen, por lo tanto, el nivel al que se lleva a cabo esta subdivisión depende del problema a resolver. Algunas de las tareas fundamentales del procesamiento digital de imágenes son: 1) Mejoramiento de una imagen digital con fines interpretativos, y 2) Toma de decisiones de manera automática de acuerdo al contenido de la imagen digital. Los algoritmos de segmentación de imágenes monocromáticas generalmente se basan en una de las dos propiedades básicas de los valores del nivel de gris: discontinuidad y similitud. En la discontinuidad el método consiste en dividir una imagen basándose en los cambios bruscos del nivel de gris. Los temas más importantes en la discontinuidad son: a) detección de puntos aislados, y b) detección de líneas y c) detección de bordes de una imagen. En la similitud, se presenta la regularidad en los valores del nivel de gris, los principales métodos están basados en a) umbralización, b) crecimiento de región, y c) división y fusión de regiones (La Serna y Román; 2009: 9).

Bordes y contornos

Uno de los métodos relacionados con la detección de bordes es el uso de la primera derivada, la que es usada porque toma el valor de cero en todas las regiones donde no varía la intensidad y tiene un valor constante en toda la transición de intensidad. Por tanto, un cambio de intensidad se manifiesta como un cambio brusco en la primera derivada [1], característica que es usada para detectar un borde, y en la que se basa el algoritmo de Canny.(Gonzales,2001)

Similitud

Antes de dar una definición formal de similitud entre dos imágenes, se tiene en cuenta que, para cada imagen se obtienen las firmas sobre cada una de las regiones de la imagen (cada región es una porción de la imagen) utilizando ventanas deslizantes aplicadas a la imagen. (Natsev et al., 2003)

Posteriormente, se agrupan estos resultados generando una estructura que permita almacenar los resultados de manera compacta, es decir, guardando solo el centroide (promedio entre las firmas) de las firmas calculadas anteriormente cuando las mismas sean similares. Un concepto más a tener en cuenta es la métrica de distancia que se utiliza en este trabajo. Esta métrica es la distancia Euclídea, la cual establece la distancia que existe entre los centroides calculados previamente de las dos imágenes a comparar. (Natsev et al., 2003)

Discontinuidad

. Se divide la imagen basándose en cambios bruscos de nivel de gris

2.2.12 Post-procesamiento de imagen

Post-processing (post-procesamiento) es el proceso de aplicar filtros y efectos de pantalla completa al *buffer* de imagen de la cámara antes de que se muestre en la pantalla. Puede mejorar drásticamente las imágenes de su producto con poco tiempo de configuración. (Technologies, s. f.)

2.2.13 Operaciones morfológicas

Erosión

Básicamente, erosión equivale a eliminación de material, que como consecuencia produce una pérdida de masa y, por tanto, una disminución de volumen o tamaño. (Marquez,1996)

Dilatación

En la práctica, la erosión y dilatación se utilizan en pares, la erosión seguida de dilatación o viceversa. En ambos casos, el resultado de aplicarlos iterativamente es la eliminación de un detalle específico de la imagen menor al tamaño del elemento estructurante, sin una distorsión geométrica global de las características no suprimidas. Por ejemplo, efectuar una apertura (erosión-dilatación) por un elemento estructurante con forma de disco, suaviza el contorno, rompe istmos angostos, elimina

pequeñas islas y agudiza picos. Realizar un cierre (dilatación-erosión) por un elemento estructurante con forma de disco, suaviza el contorno, une roturas angostas y golfos delgados, elimina pequeños agujeros y completa, espacios en el contorno (Coca et al, 2008).

2.2.13 Metodología de investigación

Las metodologías de investigación que se utilizan son la aplicada, investigación acción y documental. La investigación aplicada dirigida a proyectos tecnológicos, funciona como el método a seguir que conlleva a todos aquellos conocimientos adquiridos siendo necesarios para poder aplicar tecnologías o procesos de desarrollo; en pocas palabras significa aplicar los conocimientos resultantes durante la investigación en este caso son aquellas técnicas, herramientas y metodologías pertinentes para llevar a cabo el desarrollo del sistema, dicho esto podemos tomar como un claro ejemplo lo que plantea, Bedolla (2022) “ la investigación aplicada, dirigida para trabajos que incursionan las tecnologías, fue considerada para este estudio debido a la presentación y apoyo del sistema automatizado b-learning que fue desarrollado como una alternativa para la educación a distancia debido al confinamiento por la pandemia COVID-19 y en donde se consideran las plataformas digitales que apoyan el servicio educativo; se examinó el manejo de cada una de las plataformas educativas (Google Classrom, Zoom, Skype y Microsoft Teams) considerando actividades aplicadas para el aprendizaje; en este tipo de

investigación se plantea el objetivo de hacer una utilización inmediata del conocimiento resultante de la investigación“ (p. 11).

La investigación acción (IA) (Baskerville, 1999) la cual propone 5 fases iterativas: Diagnóstico, Plan de Acción, Tomar Acción, Evaluación y Especificación de Conocimiento, las cuales se explican a continuación.

1. Diagnóstico: Identifica la evolución y situación actual de los *Frameworks*, su complejidad que puede limitar su uso. Una solución a esta problemática la constituye un *Framework* bien diseñado, usando los principios de la Ingeniería de Software que mantenga su simplicidad, y además que esté bien documentado.

2. Planificación de la acción: establece el desarrollo sistemático de *Elementary* a partir del estudio de la metodología para el desarrollo de *Frameworks* propuesta por Bosch y otros (1999).

3. Tomar la acción: en esta fase se aplica la metodología de Bosch y otros (1999). Asimismo, en la actividad 6 de esta metodología, referente a la Documentación, se realiza con el enfoque para documentación de *Frameworks* de Aguiar y David (2000, 2011).

4. Evaluación: se realizan las pruebas y validaciones de *Elementary*.

5. Especificación del conocimiento: se elaboran conclusiones y recomendaciones, dejando abierta la posibilidad de otro ciclo de IA.

Asimismo, la propuesta de Bosch aplicada en la fase 2 de IA, incorpora los Diagramas de Clases y de Secuencia de UML (Miles y Hamilton, 2006) y comprende las siguientes actividades:

1. Análisis del Dominio: se identifican los requisitos necesarios según el dominio del *Framework*. El resultado de esta actividad es un Modelo de Análisis de Dominio, con los requisitos, conceptos del dominio y sus relaciones.
2. Diseño Arquitectónico: se considera el Análisis del dominio como entrada para decidir cuál estilo de arquitectura se adecúa para el desarrollo. Esto se denomina el diseño de alto nivel en el cual se basa el *Framework*. La salida de esta actividad consta de un diagrama de Estados y Transiciones.
3. Diseño del *Framework*: Se refina el diseño de alto nivel del *Framework* obtenido en la actividad anterior y se diseñan las clases adicionales necesarias, los resultados son: el alcance de las funcionalidades del *Framework*, las consideraciones y lineamientos de diseño basados en la arquitectura usada, y el diagrama de clases del *Framework*.
4. Implementación del *Framework*: codificación.
5. Prueba: se instancia el *Framework* para determinar si provee las funcionalidades propuestas, además se evalúa las funcionalidades del mismo. Se realizan pruebas unitarias, análisis de características y un estudio de caso de Yin. En esta investigación la fase de Evaluación de IA se corresponde con esta actividad.
6. Documentación: en este paso, se utiliza el enfoque de Aguiar y David (2000, 2011) para indicar como se estructura la documentación del *API* y el *Cookbook*, que consiste en un manual de instalación, configuración y algunos ejemplos de las funcionalidades básicas del *Framework*.

La investigación documental está presente en todos los trabajos de investigación, es necesaria para profundizar y ampliar el conocimiento sobre el tema a tratar, apoyándose de trabajos anteriores de la misma naturaleza, con el fin de poder

otorgar una base de conocimiento, utilizando medios como electrónicos o impresos divulgados en fuentes fidedignas, de acuerdo a esto los conocimientos o estudios recopilados se utilizan para poder comprender a mayor profundidad la problemática a resolver y también así poder hacer una reflexión exhaustiva de los pasos a seguir en el desarrollo del sistema tomando en cuenta los objetivos del proyecto. Es decir, se trata de un método válido para analizar diversos discursos escritos en sus propios contextos, emitidos por personas o instituciones, sin alterar su contenido, los documentos pueden ser oficiales, gubernamentales, políticos, institucionales y de gestión pública o privada (Sanchez,2020).

2.3 Trabajos Relacionados

Análisis de imágenes de rayos-x de COVID-19 usando aprendizaje profundo

Las afecciones por *COVID-19* deja evidencias radiológicas de neumonía visibles bajo los rayos-x, ya que, produce sombras en los pulmones, por lo cual, el objetivo de la investigación es analizar las imágenes de radiografías torácicas usando técnicas de aprendizaje profundo para el diagnóstico de pacientes con *COVID-19*. En este trabajo se planteó utilizar dos modelos de redes neuronales convolucionales preentrenadas en arquitectura *vgg-16* e *inception v3*, desarrollados mediante *Google Colab*, siendo entrenados con conjunto de datos, el primero fue obtenido de la unión de dos *dataset* y el segundo conjunto de datos fue referenciado a otra tesis con el fin de realizar comparaciones entre el modelo propuesto de dicha tesis contra los modelos planteados en dicho trabajo, se observó el comportamiento de los modelos siendo entrenados con grandes y pequeñas cantidades de datos, dando los siguientes

resultados: para el primer *dataset* el modelo que obtuvo mayor precisión y exhaustividad fue el modelo *Inception v3*, y para el segundo conjunto de datos el modelo que obtuvo mejores resultados fue el que usó la arquitectura *vgg16*.

(Almeida, J.,Moreira,L: 2022).

Análisis de imágenes de rayos x de COVID-19 a través de redes neuronales artificiales

En este proyecto nos recalca que las pruebas más comunes para detección de *COVID-19* que son el análisis de tomografía axial computarizada (TAC) y las evaluaciones médicas de laboratorio como la prueba de reacción en cadena de la polimerasa(PCR) son unos de los métodos más efectivos, pero tienen la desventaja de ser muy costosos y eso implica que la clase media-baja no tenga acceso a estas pruebas. Aquí entran las Redes neuronales, puesto que juegan un campo importante en la medicina e investigaciones que contribuyan a la detección de enfermedades como del *COVID-19*, este trabajo se planteó como un apoyo a los médicos en la toma de decisiones mediante el uso de herramientas tecnológicas. Se desarrolló un modelo de *Machine Learning* para el análisis de imágenes para la detección de *COVID-19* a través de redes neuronales convolucionales, el código fue realizado en *Python* con el uso de entorno de *Google Colab*, además de uso de otros repositorios como *GitHub* y *Kaggle*, se recabaron bases de datos de imágenes de rayos X de tórax de pacientes con *COVID-19* y pacientes sanos con el fin de generar un *dataset* para entrenamiento y validación. Se realizaron pruebas experimentales con tres tipos de arquitectura de

redes neuronales convolucionales para los modelos A, B y C. Siendo el modelo C el que obtuvo los mejores resultados. (Morocho, K., Publig,D: 2021)

Análisis de imágenes de rayos x por medio de redes neuronales artificiales

Este trabajo expone sobre un proceso de análisis de imágenes de rayos x mediante el uso de inteligencia artificial, mediante la aplicación de una técnica de *machine learning*, específicamente el uso de un algoritmo basado en redes neuronales artificiales, se logra un sistema capaz de aplicar aprendizaje automático que aprenda a reconocer patrones en imágenes de tórax de pacientes contagiados y sanos, para que pueda clasificar, predecir y detectar una si una nueva imagen es de un paciente contagiado o sano.(Patiño Pérez, D. et al., 2021)

Detección y análisis basados en aprendizaje profundo de COVID-19 en imágenes de rayos x de tórax

Este trabajo se centra en el análisis de imágenes de rayos x mediante el uso del *Deep Learning* porque es la técnica más exitosa de aprendizaje automático que proporciona un análisis útil para estudiar una gran cantidad de imágenes de rayos x de tórax que pueden tener un impacto mayor en la detección de *COVID-19*. Se toma la vista PA de exploraciones de rayos X de tórax para pacientes con *COVID-19*, así también para pacientes sanos. Después de depurar las imágenes y aplicar el aumento de datos, se utilizan modelos *CNN* basados en aprendizaje profundo y se compara su rendimiento. Se compararon los modelos *Inception V3*, *Xception* y *ResNeXt* y se examinó su

especificidad. Para determinar el rendimiento del modelo, se recopilaron 6432 muestras de radiografías de tórax del repositorio de *Kaggle*, de las cuales 5467 se usaron para entrenamiento y 965 para validación. En el análisis de resultados, el modelo *Xception* brinda la mayor precisión (es decir, 97,97 %) para detectar imágenes de rayos X de tórax en comparación con otros modelos. (Jain, R.)

et al.,2020) Detección y análisis de *COVID-19* en imágenes médicas mediante técnicas de aprendizaje profundo

Esta investigación recaba y compara técnicas de aprendizaje profundo aplicadas a imágenes de rayos x y tomografías computarizadas para la detección de *COVID-19*, se centra en cuatro modelos, los cuales son, *CNN* preentrenados, *VGG16*, *DenseNet121* y *ResNet152*. Donde se pudo concluir que el *VGG16* de transferencia profunda es la más precisa, logrando un 99% de precisión, por lo que se puede utilizar para dar respaldo en la detección de *COVID-19* de manera más rápida. (Yang, D. et al.,2021)

Detección de COVID-19 y neumonía en imágenes rayos x de pulmones utilizando redes neuronales convolucionales

En el presente trabajo se creó un sistema basado en redes neuronales convolucionales capaz de diferenciar si una persona se encuentra enferma de *COVID-19* o *neomenia*, se entrenaron dos modelos con el fin de compararlos para ver cuál es más efectivo. El primer modelo se basa en entrenar dos redes neuronales

convolucionales binarias para realizar la detección de tres clases. En la primera se detecta si está enfermo o no, en la segunda se analiza entre sí es *COVID-19* o neumonía. El segundo modelo solo tiene tres salidas donde detecta si no está enfermo o tiene neumonía o *COVID*. Estas redes basadas en *ResNet 152*. Se esperaba que el algoritmo de pulmón sano tuviera mayor exactitud, ya que se entiende que cuando es sano no presenta manchas y debe ser más sencillo analizarlo, pero arrojaron los siguientes porcentajes, el algoritmo para detectar neumonía y *COVID-19* tuvo una exactitud de 98.416 %, el algoritmo para detectar sano y enfermo tuvo 91.792 %. (Gomez, M; 2022).

Capítulo 3: Metodología

3.1 Elección de la metodología

De acuerdo a un análisis profundo, se determinó cual es la metodología de desarrollo correcta conforme a los requerimientos y necesidades del proyecto, por decir sus requerimientos, el alcance, la implementación así también la importancia de este proyecto para la comunidad, se llegó a la conclusión de utilizar la metodología cascada, ya que el proyecto tuvo un corto plazo de desarrollo debido a las entregas estipuladas por el DR. Juan José Bedolla Solano así también como su importancia para el sector salud de nuestra región y la necesidad de contar con un sistema que sirva como apoyo para detectar de manera temprana y oportuna casos sospechosos de COVID-19 con el fin de poder salvar la vida de un paciente, sin mencionar el entorno cambiante donde fue desarrollado el proyecto, ya que esta metodología se adapta a cambios siendo útil para cubrir esta gran necesidad. Esta metodología permitió organizar y controlar el proceso de desarrollo del sistema, siendo eficaz a la hora de poder adaptarse a los requisitos nuevos que fueron requeridos conforme se iba avanzando en el proceso, aportando a los médicos una forma más sencilla y rápida de obtener un resultado preliminar utilizando las imágenes radiológicas del paciente a tratar, teniendo dos opciones, una mediante el análisis de modelo de color y la otra por medio de una Red neuronal, brindando un acceso a los resultados de distintos pacientes con el fin de poder tener un control en el proceso de detección de COVID-19.

Se usó la metodología de desarrollo en cascada para desarrollar el trabajo porque toma las actividades fundamentales de los procesos de especificación, desarrollo, validación y evolución y luego las representa como etapas distintas del procedimiento, como elaboración de requisitos, diseño de software, implementación, pruebas, etcétera.

Sugiere un enfoque secuencial y sistemático para el desarrollo de *software* que comienza en el nivel del sistema y progresa a través del análisis, diseño, codificación, prueba y mantenimiento. La aplicación del modelo en cascada se adapta mejor al desarrollo de proyectos de corto plazo, de baja innovación, detallados e integrales; proyectos cuyos requisitos y procesos pueden ser descritos con precisión durante la etapa de planificación.

Se siguieron los procedimientos acordes a las Técnicas de recopilación y análisis de imágenes consultadas, ordenando selectivamente cada elemento en su respectiva fase, considerando las técnicas más adecuadas para cada una, el esquema metodológico se dividió en 5 fases inicialmente, esperando que puedan prolongarse en más conforme se progresa, se inició con la recopilación de los datos mediante la aplicación de los instrumentos anteriormente mencionados, seguido de preprocesamiento, segmentación, post-procesamiento y la última fase de evaluación(véase la tabla 3).

3.2 Esquema metodológico de procesamiento de imágenes

Tabla 2.2 Esquema metodológico de procesamiento de imágenes

PRIMERA FASE	SEGUNDA FASE	TERCERA FASE
Captura de imágenes	Preprocesamiento	Segmentación
<ul style="list-style-type: none"> • Recolección de datos y clasificación. • Uso de Google Drive. • Recursos electrónicos, herramientas y aplicaciones. 	<ul style="list-style-type: none"> • Modelo de color. • Ruido. • Binarización. • Umbralización. • Histograma. 	<ul style="list-style-type: none"> • Tipos de segmentación. • Selección del área de interés. • Extracción de características.

CUARTA FASE	QUINTA FASE
Post-procesamiento	Evaluación
<ul style="list-style-type: none"> • Operaciones morfológicas. • Erosión. • Dilatación. 	<ul style="list-style-type: none"> • Interpretación de imágenes. • Prototipo del sistema. • Aplicaciones futuras.

Reconocimiento de formas y contornos.	<ul style="list-style-type: none">• Mejoras aplicables.
---------------------------------------	---

Fuente: Elaboración propia (2023).

3.2.1 Captura de imágenes

En primera fase se establece la recogida y clasificación de imágenes en imágenes sanas e imágenes con *COVID-19*. Se utilizan diferentes métodos para obtener un mayor número de imágenes, asegurando la estabilidad del sistema y mejorando la precisión de cada diagnóstico. Se optó por el uso imágenes existentes en la *web* publicadas por diferentes usuarios, todas estas imágenes fueron almacenadas para su posterior análisis y se tomaron como referencia el aspecto que se tenía que buscar para identificar el virus y así se hicieran capturas adecuadas a la que se quería que el sistema analizara y reconociera en el preprocesamiento y el post-procesamiento.

Para esta fase, recopilar y almacenar estas imágenes era esencial, por lo que se utilizó *Google Drive* por varias razones, incluido el hecho de que permite un almacenamiento limitado, pero es gratuito para todos los usuarios, que todos los colaboradores tienen cuentas y, quizás lo más importante que se puede vincular a *Google Colab* para el trabajo colaborativo e incorpora el lenguaje de programación *Python*. Codificación de

forma libre Esta última es crucial para el marco y construcción del sistema de reconocimiento de imágenes, preprocesamiento y post-procesamiento.

3.2.2 Preprocesamiento de imágenes

La segunda fase consiste en organizar nuestra base de datos de imágenes de manera que podamos procesar cada imagen y discutir sus puntos clave con un experto en la materia usando el mismo formato. Se comenzó con la adquisición de la imagen recortada. Luego, se realizaron operaciones de procesamiento previo para mejorar la imagen eliminando la distorsión no deseada y se mejoraron las características deseadas para el procesamiento posterior. Se aplicaron técnicas de mejora de la calidad de imagen, reduciendo el ruido y mejorando la nitidez de la imagen para una mayor precisión. Este proceso se llevó a cabo para mejorar la calidad de la imagen y así se pudo obtener una mejor detección de dicha enfermedad.

Para procesar la imagen efectivamente se consideraron el proceso de formación y las características de interés de la imagen, se usaron técnicas para mejorar la imagen. Algunas de estas técnicas de mejora fueron la disminución del ruido, se mejoró el contraste de la imagen, se ajustó el brillo, se suavizaron bordes, se mejoró el enfoque, se realzaron contornos, se hicieron pasar por los modelos de color *RGB* (rojo, verde, azul), el modelo *CMY* (cian, magenta, amarillo) y el modelo *CMYK* (cian, magenta, amarillo, negro) que son los modelos de impresión en color, y el modelo *HSI* (tono, saturación, intensidad) que es el más cercano a como los seres humanos

interpretan el color. La binarización es definida como una técnica que consiste en la realización de un barrido en la matriz de la imagen digital, por medio de bucles o recursividad, con el fin de que el proceso produzca la reducción de la escala de grises a dos únicos valores. Negro (= 0) y blanco (= 255), o lo que es lo mismo, un sistema binario de ausencia y presencia de color 0-1. La comparación de cada píxel de la imagen viene determinada por el umbral de sensibilidad (valor $T = Threshold$). Por ejemplo, los valores que sean mayores que el umbral toman un valor 255 (blanco) y los menores 0 (negro). El ruido permite agregar píxeles a las imágenes, es la creación de señales aleatorias o puntos (más concretamente, píxeles) que no son los puntos originales de la imagen. En este caso, los puntos permitieron conocer otros aspectos a la imagen que fueron de interés para el análisis de esta, así como la detección de características en la imagen que permitió descubrir otras características. El histograma mostró características de las imágenes como el contraste y el rango dinámico. Se redistribuyeron los niveles de gris de la imagen por la reasignación de los valores de brillo de los píxeles (Viera G. 2017: 67).

3.2.3 Segmentación

Cuando se obtuvo la imagen procesada, fue muy importante y necesario identificar los objetos de interés en la imagen. La segmentación se puede definir como la etapa donde la imagen de entrada se separó en cada una de las piezas que la integran, siendo cada una de ellas homogéneas respecto a cierto atributo como textura, intensidad o color. Esta etapa fue una de las más difíciles de la visión artificial debido a que influyó directamente en el análisis de imágenes digitales. Por esta razón, se buscó tener una segmentación robusta. La mayoría de los métodos de segmentación

fueron basados en propiedades de intensidad como la similitud (referida a las regiones) y la discontinuidad (referida a los bordes). Una región se definió como un área de la imagen en la que sus *píxeles* poseen características similares y los bordes son líneas que separaron regiones diferentes. Cuando se obtuvo la imagen segmentada se pudo determinar los objetos presentes solo agrupando los *píxeles* que compartían la misma etiqueta y, simultáneamente, sean vecinos. Utilizando los objetos o regiones etiquetadas, se procedió a su clasificación o reconocimiento automático, y se pudo extraer rasgos o características discriminantes e información relevante sobre estos objetos contenidos en la imagen (Viera G. 2017: 44).

3.3.4 Post-procesamiento de imágenes

La tercera fase se asistió con operaciones morfológicas. Dentro de la operación morfológica se encontró la dilatación y erosión, la operación de erosión junto a la dilatación son la base de cualquier transformación morfológica (Montesdeoca, 2022: 11). La dilatación hizo posible el agregar *píxeles* a un objeto u imágenes, es decir, añadió puntos, cerca del objeto de estudio, al fondo de la imagen. Por otro lado, la erosión eliminó aquellos detalles insignificantes que se encuentran alrededor del objeto de estudio. Se pudo decir que la erosión es la parte contraria de la dilatación. La erosión y la dilatación son operaciones duales con respecto a la complementación. Esto significa que una erosión fue equivalente a la complementación de la dilatación de la imagen complementada con el mismo elemento estructurante y viceversa. (Ortiz, 2022: 52). Los filtros morfológicos se

podieron aplicar también para la detección de bordes en imágenes. El procedimiento consistió en la aplicación de forma independiente de una operación de dilatación y otra de erosión, ambas sobre la imagen original. Posteriormente, se realizó una resta, *píxel a píxel*, de la imagen dilatada menos la imagen erosionada, obteniendo como resultado el realce de los bordes (Ruiz Fernández; 2020: 9). En esta fase se pretendió poder analizar las imágenes, al ser manipuladas con las operaciones morfológicas, que nos permitió detectar anomalías en las imágenes y compararlas con las características comunes del *COVID-19* que deja en nuestros pulmones. Para esta fase también se utilizó librerías de edición y manipulación de imágenes que ofreció el lenguaje de programación *Python*.

3.2.5 Evaluación

El reconocimiento fue una fase de selección de características anteriormente extraídas y otra de clasificación. En la selección de características, se eliminó o se balanceó el peso de ciertas características en función de su importancia en la aplicación. La interpretación consistió en la asignación de un significado a un conjunto de objetos reconocido. El reconocimiento de patrones fue el área de conocimiento que trabajo la clasificación, descripción y agrupamiento automático de objetos, que tuvo por objetivo clasificar información extraída de un conjunto de datos. El resultado del proceso fue un descriptor o modelo, que tomó la decisión sobre la clase a la que pertenece un patrón desconocido en función de sus características. (Viera G. 2017: 46). En la clasificación final se clasificó todos los *píxeles* de la imagen en diferentes clases según sus similitudes o diferencias para poder identificar de toda

la hoja de plaga que la afectaba. Dos tipos de métodos de clasificación fueron la clasificación supervisada y la clasificación no supervisada. Por otro lado, en la clasificación de plagas no supervisada, los grupos de *píxeles* fueron basados en el análisis de *software* de una imagen sin que el usuario proporcione un conjunto de datos de entrenamiento. La computadora usó técnicas para determinar qué píxeles están relacionados y los agrupo en clases. La clasificación supervisada se basó en la idea de que un usuario puede seleccionar *píxeles* de muestra en una imagen que representa tutoriales. Luego, el *software* de procesamiento de imágenes utilizó estos conjuntos de datos de entrenamiento como referencias para el proceso de clasificación de todos los píxeles de la imagen. (Gómez C. et al; 2022: 7).

Es importante que al llegar a esta fase se obtuvo la mayor calidad en la imagen para que se pudiera interpretar con el sistema las posibles afecciones, se buscó que al evaluar el sistema se corrigieron y aplicaron mejoras en las imágenes.

3.3 Tipo de investigación

Investigación aplicada: Esta investigación fue de tipo aplicada porque este tipo se distingue por buscar formas de poner en uso el conocimiento recién adquirido. La investigación aplicada está estrechamente relacionada con la investigación básica, ya que depende de sus hallazgos y avances porque toda investigación aplicada necesita un marco teórico. De acuerdo con Murillo (2008), la característica de la investigación aplicada es que busca aplicar o utilizar conocimientos previamente adquiridos y, al mismo tiempo, adquirir nuevos conocimientos luego de implementar y sistematizar la práctica basada en la investigación. la aplicación del conocimiento y

los resultados de la investigación, lo que conduce a una forma completa, organizada y sistemática de entender la realidad.

Investigación documental: La investigación documental recolecta, recopila y selecciona información de diferentes medios, tales como revistas, libros, grabaciones, etc. Esta permite tener un conocimiento general del objeto estudiado, así como contextualizar al lector (Reyes-Ruiz y Carmona Alvarado, 2020). Esto ayudo a tener una noción más amplia de distintas fuentes de información disponibles que nos ayudaron aún más a entender cómo funciona el COVID-19 en los pulmones de los pacientes, así mismo desde un punto de vista técnico es indispensable estar documentando información sobre las funciones que te puede otorgar un lenguaje de programación al momento de codificar las funciones del sistema.

Investigación Cualitativa: La cual se basa en el entendimiento de los seres vivos y las acciones de estos, utilizando la observación, anotaciones, grabaciones o documentos que le permitan interpretarlo para encontrarle un sentido (Fernández-Collado y Baptista-Lucio, 2014).

Investigación Cuantitativa: Es objetiva en lo medidamente posible, pues este debe de ser cauteloso que sus temores, creencias, deseos o tendencias no influyan en los resultados de la investigación. Este generaliza los resultados utilizando datos numéricos (Fernández-Collado y Baptista-Lucio, 2014).

Se realizaron profundas investigaciones en línea de revistas y publicaciones científicas, así como libros, documentos gubernamentales y manuales relacionados tanto con la radiografía del tórax, como el procesamiento de imágenes. Se aplicó una entrevista a un Médico de Acapulco, Guerrero, México, para conocer sobre el trabajo de estos, así sobre cómo trabajan con este tipo de enfermedad, cuestionarios

relacionados con el funcionamiento y eficacia del prototipo desarrollado para conocer los aspectos a mejorar u corregir. Por otra parte, se observaron y analizaron las afectaciones ocasionadas por el *COVID-19* sobre los pulmones para un análisis correcto en las imágenes. Se utilizó el lenguaje de programación *Python*, así como sus librerías, funciones y herramientas que permitieron el procesamiento de las imágenes.

3.4 Selección de herramienta e instrumento de medición

Se definirá el lenguaje a utilizar para el desarrollo de la aplicación, así como el análisis de documentos como medio de diagnóstico médico para que podamos recopilar datos para nuestro proyecto. Usando estudios imagenológicos de *COVID-19*, podemos identificar imágenes de personas con el virus en sus pulmones. Y a partir de ese momento, será posible utilizar imágenes de estos mismos estudios como referencia para el funcionamiento del sitio *web*. Al mismo tiempo, se están tomando medidas para crear un almacén de datos que mantendrá imágenes digitalizadas de personas con este diagnóstico. Además, se llevó a cabo una entrevista con un experto médico en el campo de las imágenes digitales de apoyo para obtener su evaluación y explicación de cómo el *COVID-19* afecta los pulmones.

3.5 Procedimientos

3.5.1 Técnicas de procesamiento de imágenes

Tabla descriptiva de técnicas

La tabla 3.5 describe las fases llevadas a cabo en la metodología, las técnicas utilizadas en dichas fases, así como una breve descripción de cada una que permite conocer lo que se llevó a cabo, y por último las librerías utilizadas en cada técnica, pertenecientes al lenguaje de programación *Python*. Cada fase de técnica de esta tabla nos permitió procesar y manipular las imágenes, haciendo posible que las afectaciones de la plaga fueran destacadas. El lenguaje de programación fue la herramienta más fuerte, pues cuenta con gran variedad de librerías que hacen posible la automatización de este proceso, al solo necesitar ser adaptado al enfoque deseado.

Tabla 3.5 Técnicas de procesamiento de imágenes

FASE	TÉCNICA	DESCRIPCIÓN	LIBRERÍAS
Preprocesamiento	Modelo de color	Se basa en la representación de un solo color utilizando una mezcla de los tres colores primarios (es decir, rojo, azul y verde).	<i>numpy</i> <i>cv2</i> <i>cv2_imshow</i>
	Ruido	Es la creación de señales o puntos aleatorios (más	<i>cv2</i> <i>numpy</i> <i>matplotlib</i>

		específicamente, píxeles), que no son originales a la imagen.	<i>pyplot</i>
	Binarización	Es la conversión de una imagen a color a una tonalidad de blanco y negro sin afectar las propiedades de esta.	<i>cv2</i> <i>numpy</i> <i>matplotlib</i> <i>pyplot</i>
	Umbralización	Se basa en la extracción de píxeles de una imagen a blanco y negro a partir de la manipulación de la intensidad.	<i>cv2</i> <i>numpy</i> <i>matplotlib</i> <i>pyplot</i>
	Histograma	Permite representar de manera gráfica la exposición en la fotografía.	<i>cv2</i> <i>numpy</i> <i>matplotlib</i>
Segmentación	Tipos de segmentación	Se trata de la división u clasificación de los píxeles de una imagen para posteriormente ser analizados en base a sus características. Esta se divide en similitud y discontinuidad.	<i>cv2</i> <i>pyplot</i> <i>numpy</i>
	Selección de áreas de interés	Se basa en el enfoque de un área de interés específica en la imagen para su análisis y procesamiento.	<i>cv2</i> <i>numpy</i>

			<i>matplotlib</i> <i>pyplot</i>
	Extracción de características	Es la selección de características en una imagen que coincidan con el objeto de estudio.	<i>cv2</i> <i>pyplot</i> <i>numpy</i>
Post-procesamiento	Operaciones morfológicas	Se basa en extraer partes de una imagen que permitan representar y describir ciertas regiones de esta que sean útiles.	<i>cv2</i> <i>numpy</i> <i>matplotlib</i>
	Erosión	Consiste en eliminar los píxeles cerca del borde de una imagen que sean innecesarios en el análisis.	<i>cv2</i> <i>numpy</i>
	Dilatación	Es una forma de estirar una imagen al añadir píxeles a los bordes de esta para así resaltarla.	<i>cv2</i> <i>numpy</i> <i>matplotlib.pyplot</i>
	Bordes y contornos	Es la detección de bordes en una imagen que permita identificar y extraer objetos de estas.	<i>cv2</i> <i>numpy</i> <i>cv2_imshow</i>

Fuente: Elaboración propia (2023).

3.6 Red neuronal utilizando DenseNet 121

Se estudio una red neuronal, donde se concluyó que era adecuada para cumplir con las necesidades del proyecto, ya que esta está hecha en el mismo **framework** que es *Flask* por ende se codificó utilizando Python representando poca dificultad al momento de adaptar la función encargada de realizar los análisis sobre las imágenes radiológicas mediante el empleo de un modelo de DenseNet121, la cual es una red densa preconfigurada que tiene 121 capas de profundidad. El modelo clasifica los dos resultados como: COVID y no COVID, emplea un *dataset* clasificado de igual forma, por ello entrenaron el modelo con un numero de 64 muestras para actualizar los pesos en 50 épocas, al momento de entrenar la red.

```
disease_types=['COVID', 'non-COVID']
data_dir = '/kaggle/input/sarscov2-ctscan-dataset/SARS-Cov-2/'
train_dir = os.path.join(data_dir)
```

Ilustración 11. Declaración de los dos tipos de clasificación de COVID-19 así como del dataset de imágenes a utilizar en el entrenamiento del modelo. Fuente: SHAWON A. (2020). Covid-19 Diagnosis from Images Using DenseNet121.

```
BATCH_SIZE = 64

# Split the train and validation sets
X_train, X_val, Y_train, Y_val = train_test_split(X_Train, Y_train, test_size=0.2, random_state=SEED)
```

Ilustración 12. Declaración del **batch_size**, que es el número de muestras que se utilizan para el entrenamiento de la red neuronal en múltiplos de dos de acuerdo a la memoria del computador. Fuente: SHAWON A. (2020). Covid-19 Diagnosis from Images Using DenseNet121.

```
EPOCHS = 50  
SIZE=64  
N_ch=3
```

Ilustración 13. Declaración de las épocas en las que se entrenó el modelo. Fuente: SHAWON A. (2020). Covid-19 Diagnosis from Images Using DenseNet121.

Al momento de declarar el modelo, se utiliza una función específica, donde se establece los parámetros de la misma función así como sus entradas y por ende los resultados que arroja el modelo de acuerdo a las dos categorías que se tienen sobre la enfermedad COVID-19.

La matriz de confusión es de gran utilidad ya que nos ayuda a determinar la exactitud del modelo a la hora de poner a prueba la confiabilidad de los resultados en la siguiente imagen se puede observar los resultados falsos y verdaderos siendo los verdaderos en color verde oscuro y los falsos en verde claro, si aplicamos la fórmula de la exactitud podemos ver que el resultado se observa en la siguiente ilustración.

$$\frac{(250 + 210)}{(250 + 23 + 15 + 210)} = 0.9236$$

Ilustración 14. Aplicación de la fórmula de la exactitud en matrices de confusión. Fuente: elaboración propia(2023).

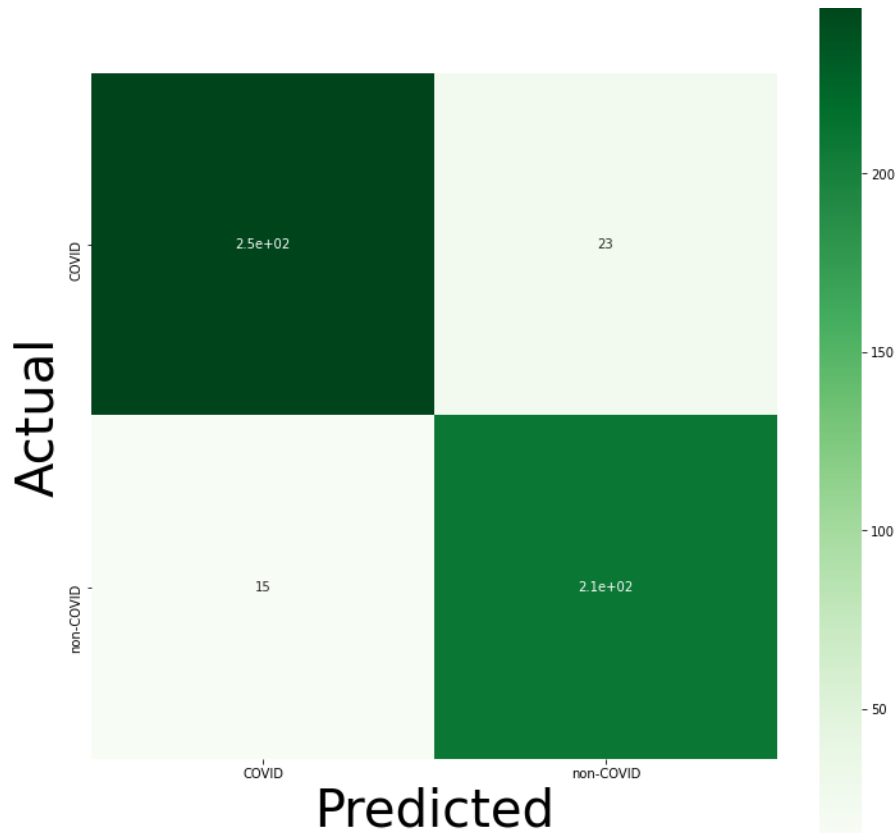


Ilustración 15. matriz de confusión de los resultados verdaderos positivos, falsos positivos, falsos negativos y verdaderos negativos. Fuente: SHAWON A. (2020). Covid-19 Diagnosis from Images Using DenseNet121.

```
#model = load_model('./output/kaggle/working/model.h5')
final_loss, final_accuracy = model.evaluate(X_val, Y_val)
print('Final Loss: {}, Final Accuracy: {}'.format(final_loss, final_accuracy))
```

```
497/497 [=====] - 1s 2ms/step
Final Loss: 0.20675026198148247, Final Accuracy: 0.923541247844696
```

Ilustración 16. Exactitud final calculada en el entrenamiento del modelo. Fuente: SHAWON A. (2020). Covid-19 Diagnosis from Images Using DenseNet121.

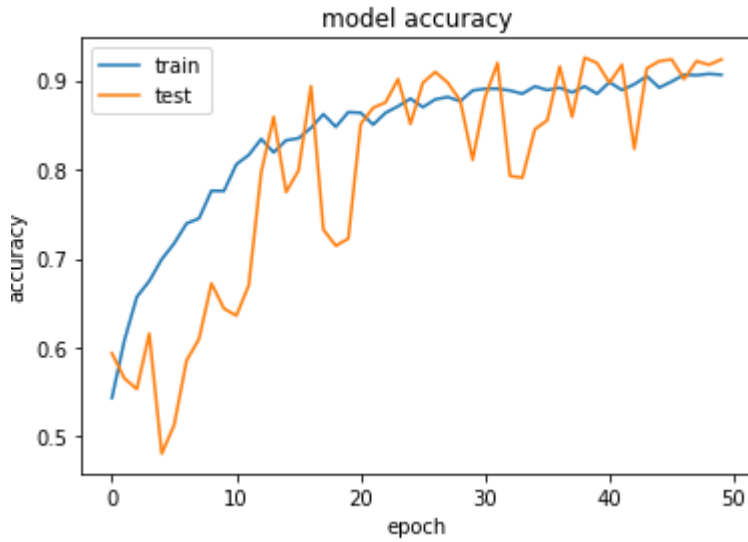


Ilustración 17. Grafica de la exactitud del modelo en entrenamiento y pruebas. Fuente: SHAWON A. (2020). Covid-19 Diagnosis from Images Using DenseNet121.

3.7 Análisis de factibilidad

A continuación, se analizará la factibilidad técnica del proyecto en donde se especificarán los recursos, herramientas, costos y tecnologías utilizadas.

3.7.1 Factibilidad técnica

Hardware

Tabla 3.6 Recursos de hardware utilizado en el proyecto

Hardware	Características
Laptop HP	Intel(R) Core(TM) i3. 8.00 GB RAM 480 GB SSD

OMEN by HP - 15-ce005la	Intel® Core™ i7-7700HQ (frecuencia base de 2,8 GHz, hasta 3,8 GHz con tecnología Intel® Turbo Boost, 6 MB de caché, 4 núcleos) Memoria, estándar 8 GB de SDRAM DDR4-2133 (1 x 8 GB) Gráficos de vídeo NVIDIA® GeForce® GTX 1050 (GDDR5 de 4 GB dedicados) Disco duro SATA de 1 TB y 7200 rpm
-------------------------	--

Fuente: Elaboración propia (2023).

Tabla 3.6.1 Recursos de software utilizados en proyecto

Software	Versión
<i>Python</i>	v. 3.11
<i>Bootstrap</i>	v. 5.3
<i>Flask</i>	v. 2.3.2
<i>MySQL</i>	v. 8.0.33
<i>Google Colab</i>	
<i>Visual Code</i>	v.1.78
<i>Git</i>	v. 2.40
<i>OpenCv</i>	v.4.7.029
<i>Matplotlib</i>	v. 3.7.1
<i>TensorFlow</i>	v. 2.12

Fuente: Elaboración propia (2023).

3.7.2 Factibilidad económica

Hardware

Tabla 3.6.2. Costo en recurso de hardware

Hardware	Costo
Laptop HP	\$10,000.00 MX
OMEN by HP - 15-ce005la	\$20,000.00 MX

Fuente: Elaboración propia (2023).

3.8 Recursos Humanos

Tabla 3.7 Costo en recursos humano para el desarrollo del proyecto

Recurso Humano	Cantidad	Tiempo	Costo
Programador	2	3 meses	\$24,000.00 MX

Fuente: Elaboración propia (2023).

3.7.1 Costo total

Tabla 3.7.1. Costo total del desarrollo del proyecto

Detalle	Costo
Hardware	\$ 30,000.00 MX

Software	\$0
Recurso Humano	\$24,000.00 MX
Total	\$ 54,000.00MX

Fuente: Elaboración propia (2023).

3.9 Análisis de costos y beneficios

El desarrollo del proyecto tendrá un costo total de \$54,000.00 pesos mexicanos. En donde, la aplicación *web* permitirá automatizar los procesos de detección de *COVID-19* en los pacientes a través de imágenes de radiografías de tórax. Uno de los beneficios del desarrollo del proyecto es el uso de tecnologías y herramientas de código abierto, que no requieren ningún tipo de licencia para operar y no generan costos adicionales.

Por otro lado, el sistema también permite recibir reportes de casos confirmados de *COVID* para seguir el comportamiento del virus, permitiendo identificar el total de pacientes positivos a *COVID*.

3.9.1 Evaluación del sistema

Para la evaluación del sistema se aplicaron normas de evaluación de *software*, estas son Norma *ISO/IEC 9126* e *ISO/IEC 25010*.

Norma ISO/IEC 9126: Es un reporte técnico que incluye las métricas internas que se pueden aplicar a un producto de *software*; cabe destacar que al ser métricas internas se aplican a productos de *software* no ejecutables (González Pinzón y González Sanabria, 2013: 117).

El estándar *ISO-9126* estipula que cualquier componente de la calidad del software se puede describir mediante seis atributos, y cada atributo se explica en detalle mediante un conjunto de subatributos, de modo que la evaluación de la calidad del producto de *software* se puede analizar y profundizar (Tabla 3. 9).

Tabla 3.9. Características de norma ISO 9126

ISO 9126	
CARACTERÍSTICA	SUBCARACTERÍSTICA
Funcionalidad	Idoneidad
	Exactitud
	Interoperabilidad
	Seguridad
Fiabilidad	
	Madurez
	Tolerancia a fallos
	Facilidad de recuperación
Usabilidad	Facilidad de comprensión
	Facilidad de aprendizaje
	Operatividad
Eficiencia	Tiempo de uso
	Recursos utilizados
Mantenibilidad	Facilidad de análisis
	Facilidad de cambio
	Estabilidad
	Facilidad de prueba
Portabilidad	Facilidad de instalación
	Facilidad de ajuste
	Facilidad de adaptación al cambio

Fuente: González Pinzón y González Sanabria, 2013: 117.

A continuación, se detalla cada una de las características que establece el estándar *ISO-9126*.

- **Funcionalidad:** En este grupo se conjunta una serie de atributos que permiten calificar si un producto de *software* maneja en forma adecuada el conjunto de funciones que satisfagan las necesidades para las cuales fue diseñado. Para este propósito se establecen los siguientes atributos:
 - **Adecuación.** Se enfoca a evaluar si el software cuenta con un conjunto de funciones apropiadas para efectuar las tareas que fueron especificadas en su definición.
 - **Exactitud.** Este atributo permite evaluar si el *software* presenta resultados o efectos acordes a las necesidades para las cuales fue creado.
 - **Interoperabilidad.** Permite evaluar la habilidad del *software* de interactuar con otros sistemas previamente especificados.
 - **Conformidad.** Evalúa si el software se adhiere a estándares, convenciones o regulaciones en leyes y prescripciones similares.
 - **Seguridad.** Se refiere a la habilidad de prevenir el acceso no autorizado, ya sea accidental o premeditado, a los programas y datos.
- **Confiabilidad:** Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del *software* de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido. Las subcaracterísticas que el estándar sugiere son:
 - **Nivel de Madurez.** Permite medir la frecuencia de falla por errores en el *software*.

- Tolerancia a fallas. Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del *software* o de cometer infracciones de su interfaz específica.
- Recuperación. Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesarios para lograrlo.
- Usabilidad: Consiste en un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema.
 - Comprensibilidad. Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del *software*.
 - Facilidad de Aprender. Establece atributos del *software* relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.
 - Operabilidad. Agrupa los conceptos que evalúan la operación y el control del sistema.
- Eficiencia: Esta característica permite evaluar la relación entre el nivel de funcionamiento del *software* y la cantidad de recursos usados. Los aspectos a evaluar son:
 - Comportamiento con respecto al Tiempo. Atributos del *software* relativos a los tiempos de respuesta y de procesamiento de los datos.
 - Comportamiento con respecto a Recursos. Atributos del *software* relativos a la cantidad de recursos usados y la duración de su uso en la

realización de sus funciones.

- **Mantenibilidad:** Se refiere a los atributos que permiten medir el esfuerzo necesario para realizar modificaciones al *software*, ya sea por la corrección de errores o por el incremento de funcionalidad. En este caso, se tienen los siguientes factores:
 - Capacidad de análisis. Relativo al esfuerzo necesario para diagnosticar las deficiencias o causas de fallas, o para identificar las partes que deberán ser modificadas.
 - Capacidad de modificación. Mide el esfuerzo necesario para modificar aspectos del *software*, remover fallas o adaptar el *software* para que funcione en un ambiente diferente.
 - Estabilidad. Permite evaluar los riesgos de efectos inesperados debidos a las modificaciones realizadas al *software*.
 - Facilidad de Prueba. Se refiere al esfuerzo necesario para validar el *software* una vez que fue modificado.
- **Portabilidad:** En este caso, se refiere a la habilidad del *software* de ser transferido de un ambiente a otro, y considera los siguientes aspectos:
 - Adaptabilidad. Evalúa la oportunidad para adaptar el *software* a diferentes ambientes sin necesidad de aplicarle modificaciones.
 - Facilidad de Instalación. Es el esfuerzo necesario para instalar el *software* en un ambiente determinado.
 - Conformidad. Permite evaluar si el *software* se adhiere a estándares o convenciones relativas a portabilidad.

- Capacidad de reemplazo. Se refiere a la oportunidad y el esfuerzo usado en sustituir el *software* por otro producto con funciones similares. (Abud Figueroa, 2022, 1-2)

Métricas:

Las métricas muestran la puntuación de cada atributo de calidad. Va de 0 a 10, donde 0 es la calificación menos satisfactoria y 10 es la calificación más satisfactoria. Los pesos se multiplican para obtener la nota, que es la nota final. Contiene una columna que muestra el valor deseado para cada subpropiedad (Tabla 3.9.1). Para obtener la puntuación final, se suman los subatributos y se comparan con la puntuación idea.

Tabla 3.9.1. Métrica de propiedades

Atributo	Subatributo	Peso
Funcionalidad		
	Idoneidad	2
	Exactitud	2
	Interoperabilidad	2
	Seguridad	2
	Conformidad	2
	Confiabilidad	2
Confiabilidad		

Madurez	2
Recuperación	2
Operatividad	2
Usabilidad	
Aprendizaje	2
Compresión	2
Operatividad	2
Eficiencia	
Comportamiento en el tiempo	2
Comportamiento de recursos	2
Mantenibilidad	
Estabilidad	2
Facilidad de análisis	2
Facilidad de cambio	2
Facilidad de pruebas	2
Portabilidad	
Capacidad de instalación	2
Capacidad de reemplazamiento	2
Adaptabilidad	2
Coexistencia	2

Fuente: *Elaboración propia (2023).*

Teniendo en cuenta, también se clasifica el nivel obtenido por cada subcategoría en A (alto), M (medio) y B (bajo), según un rango dado (Tabla 3.9.2).

Tabla 3.9.2. Métrica de niveles

RANGO	NIVEL
1-3	B
4-7	M
8-10	A

Fuente: Elaboración propia (2023).

Norma *ISO/IEC 25010*: De acuerdo al sitio oficial de *la ISO (ISO 25000, 2022)* el modelo de calidad representa la piedra angular en torno a la cual se establece el sistema para la evaluación de la calidad del producto. En este modelo se determinan las características de calidad que se van a tener en cuenta a la hora de evaluar las propiedades de un producto *software* determinado. La calidad del producto *software* se puede interpretar como el grado en que dicho producto satisface los requisitos de sus usuarios, aportando de esta manera un valor. Son precisamente estos requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas.

El modelo de calidad del producto definido por la *ISO/IEC 25010* se encuentra compuesto por las ocho características de calidad que se muestran en la siguiente ilustración:



Ilustración 18. Modelo de calidad del producto de software

Fuente: (ISO 25000, 2022).

Adecuación Funcional

Representa la capacidad del producto *software* para proporcionar funciones que satisfacen las necesidades declaradas e implícitas, cuando el producto se usa en las condiciones especificadas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Completitud funcional. Grado en el cual el conjunto de funcionalidades cubre todas las tareas y los objetivos del usuario especificados.
- Corrección funcional. Capacidad del producto o sistema para proveer resultados correctos con el nivel de precisión requerido.
- Pertinencia funcional. Capacidad del producto *software* para proporcionar un

conjunto apropiado de funciones para tareas y objetivos de usuario especificados.

Eficiencia de desempeño

Esta característica representa el desempeño relativo a la cantidad de recursos utilizados bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Comportamiento temporal. Los tiempos de respuesta y procesamiento y las ratios *de throughput* de un sistema cuando lleva a cabo sus funciones bajo condiciones determinadas en relación con un banco de pruebas (*benchmarking*) establecido.
- Utilización de recursos. Las cantidades y tipos de recursos utilizados cuando el *software* lleva a cabo su función bajo condiciones determinadas.
- Capacidad. Grado en que los límites máximos de un parámetro de un producto o sistema *software* cumplen con los requisitos.

Compatibilidad

Capacidad de dos o más sistemas o componentes para intercambiar información y/o llevar a cabo sus funciones requeridas cuando comparten el mismo entorno, *hardware* o *software*. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Coexistencia. Capacidad del producto para coexistir con otro *software* independiente, en un entorno común, compartiendo recursos comunes sin detrimento.

- Interoperabilidad. Capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada.

Usabilidad

Capacidad del producto *software* para ser entendido, aprendido, usado y resultar atractivo para el usuario, cuando se usa bajo determinadas condiciones. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Reconoscibilidad de la adecuación. Capacidad del producto que permite al usuario entender si el *software* es adecuado para sus necesidades.
- Aprendizabilidad. Capacidad del producto que permite al usuario aprender su aplicación.
- Operabilidad. Capacidad del producto que permite al usuario operarlo y controlarlo con facilidad.
- Protección contra errores de usuario. Capacidad del sistema para proteger a los usuarios de hacer errores.
- Estética de la interfaz de usuario. Capacidad de la interfaz de usuario de agradar y satisfacer la interacción con el usuario.
- Accesibilidad. Capacidad del producto que permite que sea utilizado por usuarios con determinadas características y discapacidades.

Fiabilidad

Capacidad de un sistema o componente para desempeñar las funciones especificadas, cuando se usa bajo unas condiciones y periodo de tiempo

determinado. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Madurez. Capacidad del sistema para satisfacer las necesidades de fiabilidad en condiciones normales.
- Disponibilidad. Capacidad del sistema o componente de estar operativo y accesible para su uso cuando se requiere.
- Tolerancia a fallos. Capacidad del sistema o componente para operar según lo previsto en presencia de fallos *hardware* o *software*.
- Capacidad de recuperación. Capacidad del producto *software* para recuperar los datos directamente afectados y restablecer el estado deseado del sistema en caso de interrupción o fallo.

Seguridad

Capacidad de protección de la información y los datos de manera que personas o sistemas no autorizados no puedan leerlos o modificarlos. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Confidencialidad. Capacidad de protección contra el acceso de datos e información no autorizados, ya sea accidental o deliberadamente.
- Integridad. Capacidad del sistema o componente para prevenir accesos o modificaciones no autorizados a datos o programas de ordenador.
- No repudio. Capacidad de demostrar las acciones o eventos que han tenido lugar, de manera que dichas acciones o eventos no puedan ser repudiados

posteriormente.

- Responsabilidad. Capacidad de rastrear de forma inequívoca las acciones de una entidad.
- Autenticidad. Capacidad de demostrar la identidad de un sujeto o un recurso.

Mantenibilidad

Esta característica representa la capacidad del producto *software* para ser modificado efectiva y eficientemente, debido a necesidades evolutivas, correctivas o perfectivas. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Modularidad. Capacidad de un sistema o programa de ordenador (compuesto de componentes discretos) que permite que un cambio en un componente tenga un impacto mínimo en los demás.
- Reusabilidad. Capacidad de un activo que permite que sea utilizado en más de un sistema *software* o en la construcción de otros activos.
- Analizabilidad. Facilidad con la que se puede evaluar el impacto de un determinado cambio sobre el resto del *software*, diagnosticar las deficiencias o causas de fallos en el *software*, o identificar las partes a modificar.
- Capacidad para ser modificado. Capacidad del producto que permite que sea modificado de forma efectiva y eficiente sin introducir defectos o degradar el desempeño.
- Capacidad para ser probado. Facilidad con la que se pueden establecer criterios

de prueba para un sistema o componente y con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios.

Portabilidad

Capacidad del producto o componente de ser transferido de forma efectiva y eficiente de un entorno *hardware*, *software*, operacional o de utilización a otro. Esta característica se subdivide a su vez en las siguientes subcaracterísticas:

- Adaptabilidad. Capacidad del producto que le permite ser adaptado de forma efectiva y eficiente a diferentes entornos determinados de *hardware*, *software*, operacionales o de uso.
- Capacidad para ser instalado. Facilidad con la que el producto se puede instalar y/o desinstalar de forma exitosa en un determinado entorno.
- Capacidad para ser reemplazado. Capacidad del producto para ser utilizado en lugar de otro producto software determinado con el mismo propósito y en el mismo entorno (ISO 25000, 2022).

Capítulo 4: Desarrollo

4.1 Herramientas

Dentro de nuestro proyecto estaremos requiriendo como recursos humanos a:

(2) Desarrollador web

Los cuales estarán apoyándose mutuamente para que la página *web*, tenga un resultado más acertado a las evaluaciones que haría un especialista, así mismo la retroalimentación constante de los especialistas a los desarrolladores.

Como recursos materiales requeriremos un ordenador con los siguientes requerimientos mínimos:

- Procesador: *Intel Atom® processor o Intel® Core™ i3*
- Disco Duro: *1 GB*
- Sistema Operativo: *Windows* 7 o posterior, macOS, o Linux.*
- Herramientas de desarrollo empleadas: *Visual Studio Code*
- Librerías incluidas: *OpenCV*

4.2.Base de datos de la aplicación

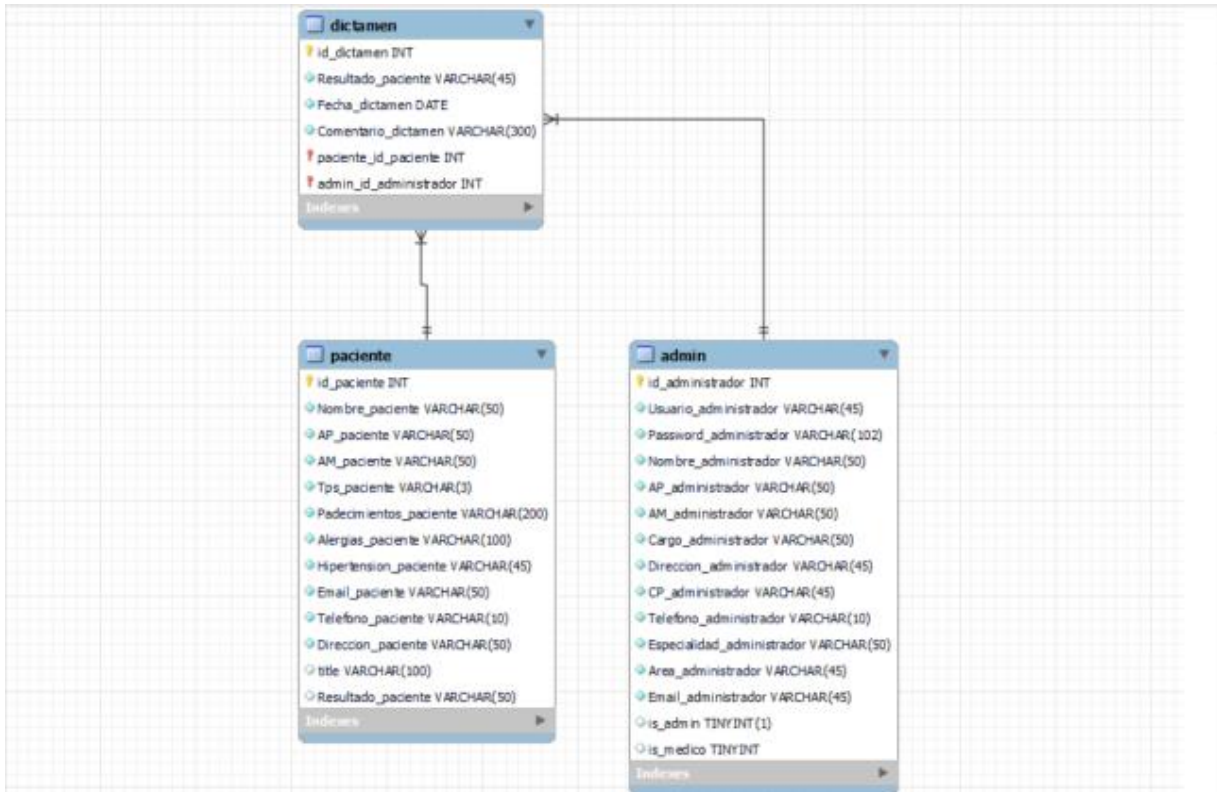


Ilustración 19. Base de datos de aplicación web. Fuente: Propia (2023).

4.2.1 Diccionario de datos

El diccionario de datos está basado en el diagrama relacional de la base de datos del sistema, tiene el propósito de describir cada tabla de la base de datos, además de sus atributos como tipos de datos, si son nulos o la descripción de los mismos.

Tabla Dictamen

La tabla dictamen, conlleva toda aquella información que se consideró pertinente para poder guardar el dictamen final hecho por el médico que visualizó los resultados previamente arrojados por el sistema y en base a su conocimiento determinó este mismo, esta tupla cuenta con un identificador de tipo entero, la fecha en la que se registró dicho dictamen además del comentario donde se da una explicación detallada de lo que determina el Doctor, sin mencionar que relaciona todo esto al paciente que le corresponde dichos análisis.

COLUMN NAME	DATA TYPE	NOT NULL	COMMENTS
-------------	-----------	----------	----------

Id_dictamen	Int	No	Clave del dictamen.
Resultado_paciente	Varchar(45)	No	Valor en positivo o negativo a COVID-19.
Fecha_dictamen	date	No	Fecha en la cual se registró el dictamen
Comentario_dictamen	Varchar(300)	No	Observaciones del médico.
Paciente_id_paciente	Int	No	Clave del paciente al cual pertenece el dictamen.
Admin_id_administradore	Int	No	Clave del médico al cual pertenece el dictamen.

Tabla 4.2.1 Dictamen, fuente: Propia(2023).

Tabla Administrador

En esta tabla se guardan los datos correspondientes al médico o trabajador del área de Salud encargado de gestionar el sistema, así como de aplicar los análisis a las imágenes radiológicas del paciente que se está tratando, se le asigna un identificador único al registro, nombre de usuario, también se capturan datos personales como son los nombres, apellidos, dirección y número de teléfono, en cuanto a datos profesionales se toman en cuenta la especialidad del médico y su área hospitalaria. Se tiene un campo llamado "is_admin" este campo tiene una función importante porque por ser booleano, se puede comprobar si el usuario que está entrando tiene el rol de administrador o es un usuario normal, de acuerdo si en este atributo este valor se presenta como verdadero, en caso de ser así, se podrá tener acceso a la información de los usuarios del sistema, como dar de alta y eliminar un usuario ya existente, por el caso contrario solo se le podrá dar acceso al sistema de análisis de imágenes y registro de pacientes.

COLUMN NAME	DATA TYPE	NOT NULL	COMMENTS
Id_administrador	Int	No	Clave del usuario.
Usuario_administrador	Varchar(45)	No	Nombre de usuario.
Password_administrador	Varchar(102)	No	Contraseña del usuario
Nombre_administrador	Varchar(50)	No	Nombre del usuario.
AP_administrador	Varchar(50)	No	Apellido paterno del usuario
AM_administrador	Varchar(50)	No	Apellido Materno del usuario

Cargo_administrador	Varchar(50)	No	Cargo que ocupa en un hospital el usuario.
Direccion_administrador	Varchar(45)	No	Dirección del usuario.
CP_administrador	Varchar(45)	No	Código Postal del usuario.
Telefono_administrador	Varchar(10)	No	Número de teléfono.
Especialidad_administrador	Varchar(50)	No	Especialidad médica del usuario.
Area_administrador	Varchar(45)	No	Área hospitalaria a la que pertenece el usuario.
Email_administrador	Varchar(45)	No	Correo electrónico.
Is_admin	Boolean	No	Es administrador o no.

Tabla 4.1.2 Admin, fuente: Propia(2023).

Tabla paciente

El registro de los pacientes se obtiene de la tabla pacientes, esta tupla tiene un número de identificador único, se guardan datos personales del paciente, como son su nombre, apellidos, tipo de sangre, padecimientos del paciente en caso de contar, alergias, ya sean a medicamentos u otras cosas, sí padece de hipertensión arterial siendo este un factor clave para poder determinar el nivel de mortalidad del virus en el paciente, el nombre de la imagen radiológica del paciente se guarda en este registro esta misma se puede actualizar en caso de ser necesario, ya que el sistema guarda el nombre de la imagen haciendo referencia a la misma por lo que se debe tener cuidado a la hora de nombrar el archivo del paciente, teniendo que ser exactamente el mismo nombre.

COLUMN NAME	DATA TYPE	NOT NULL	COMMENTS
Id_paciente	Int	No	Clave del paciente.
Nombre_paciente	Varchar(50)	No	Nombre del paciente.
AP_paciente	Varchar(50)	No	Apellido paterno del paciente.
AM_paciente	Varchar(50)	No	Apellido Materno del paciente.
Tps_paciente	Varchar(3)	No	Tipo de sangre del paciente.
Padecimientos_paciente	Varchar(200)	No	Dirección del paciente.
Alergias_paciente	Varchar(100)	No	Alergias del paciente.
Hipertensión_paciente	Varchar(45)	No	Tiene hipertensión el paciente.
Email_paciente	Varchar(50)	No	Correo electrónico del paciente.

Telefono_paciente	Varchar(10)	No	Número de teléfono del paciente.
Direccion_paciente	Varchar(50)	No	Dirección del paciente.
Title	Varchar(100)	Si	Nombre de la imagen radiológica del paciente.
Resultado	Varchar(50)	Si	Resultado positivo o negativo al análisis.

Tabla 4.1.3. Paciente Fuente: Propia (2023).

4.2.3 Modelo Entidad-Relación

En este modelo las entidades se representan como rectángulos y las relaciones son los rombos, la nomenclatura de las relaciones se insertan en las líneas que conectan a las entidades, los atributos son representados por medio de círculos ovalados, en la siguiente figura (Ilustración 58) se puede observar las tres entidades y las relaciones entre estas, los atributos que son llaves primarias están subrayados, como se puede observar paciente y admin tienen una relación de uno a muchos por lo que siguiendo la lógica de que el paciente puede tener muchos dictámenes y el admin realizar más de un dictamen, esta es la estructura que se obtiene como resultado.

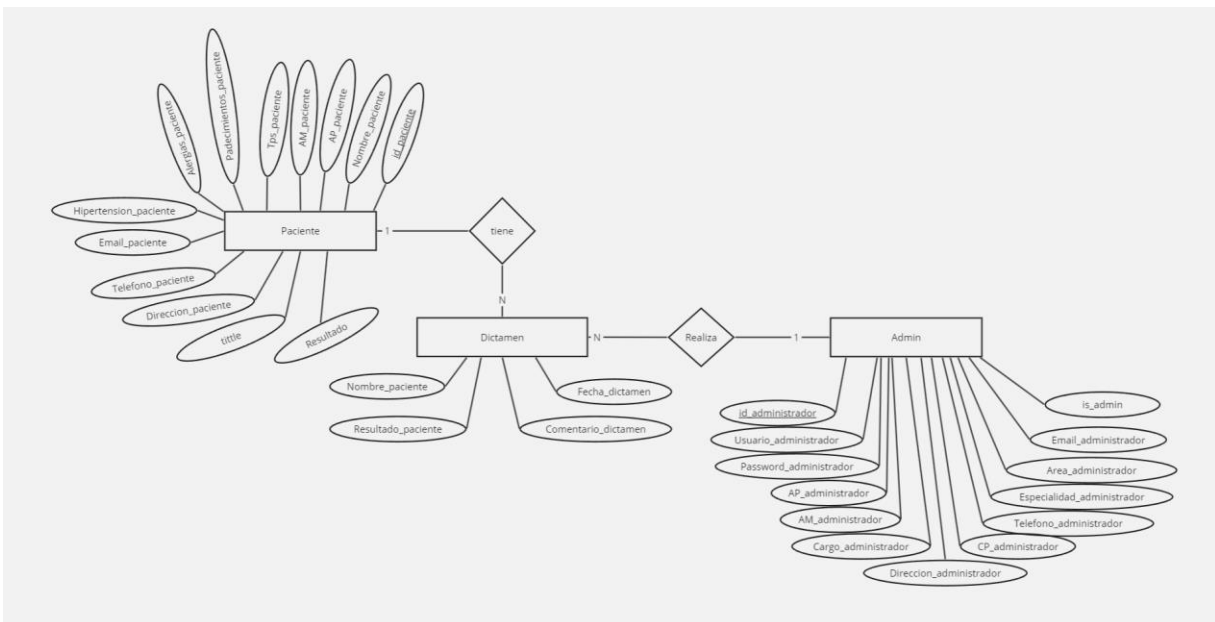


Ilustración 20. Diagrama entidad relación de la base de datos. Fuente: Propia(2023).

4.2.3 Modelo relacional

El modelo relacional está basado en el conjunto de datos y lógica de predicados, este modelo representa los datos en conjunto por tupla y la relación entre estas, como se observa en la siguiente ilustración podemos ver existen tres tablas, la tabla paciente tiene una relación identificadora de muchos a uno con dictamen y por otro lado la tabla admin tiene la misma relación con la misma tabla, ya que se entiende que un admin puede realizar muchos dictámenes y un dictamen puede ser realizado por un solo administrador, lo mismo con la tupla paciente este puede llegar a tener más de un dictamen, pero un dictamen pertenece a un solo paciente.

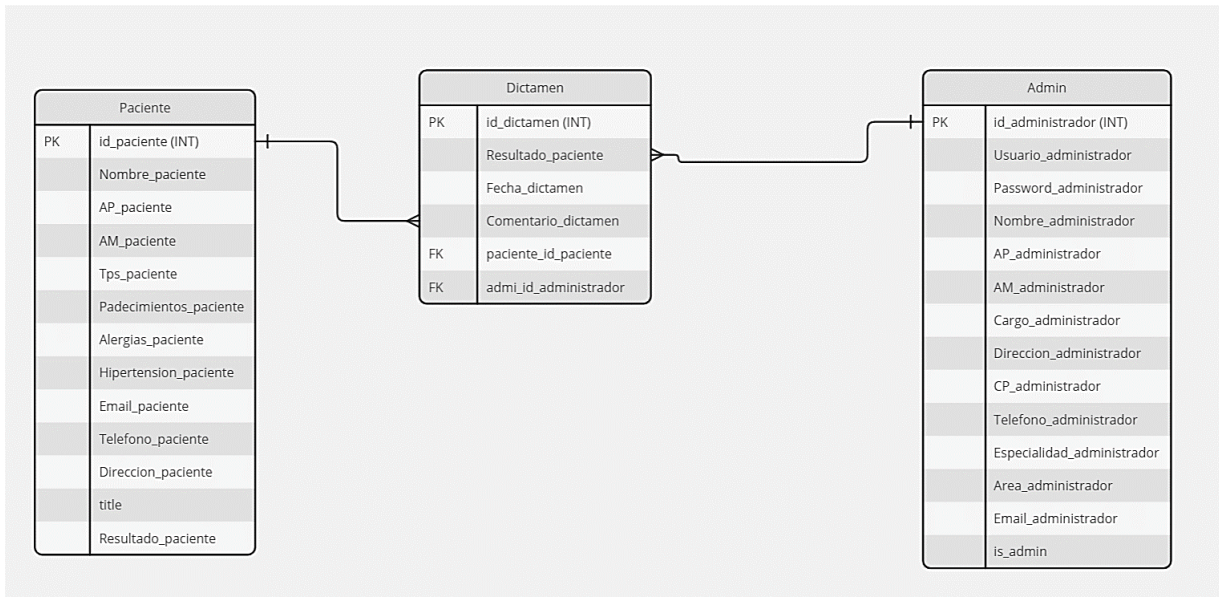


Ilustración 21. Diagrama relacional de la base de datos. Fuente: Propia(2023).

4.3 Funcionamiento de la aplicación

Tabla 3.2 Funcionamiento de los módulos de la aplicación.

Módulo de la aplicación	Descripción	Grupo de usuario

Menú	En este módulo se gestionan y crean los menús de la aplicación para cada grupo de usuarios	Administrador Médico
Perfil	En este módulo se ve la información del médico y así mismo se edita dicha información.	Administrador
Médico	En este módulo el administrador podrá ver los usuarios registrados y por otro lado, agregar usuario que él desee, ya que no solo podrá agregar médico sino también enfermera o secretarías.	Administrador
Lista de estudios	En este módulo el administrador como el usuario podrán ver los estudios realizados	Administrador Médico

Estudios	En este módulo se podrá consultar los resultados de los estudios, ya sea (positivos, sospechosos, negativos).	Administrador Médico
Ayuda	En este módulo se podrá observar indicaciones de cómo usar el programa o también si el usuario tiene problema con la aplicación, poder resolver sus dudas	Administrador Médico

Fuente: Elaboración propia (2023).

4.3.2 Submódulos de la aplicación

Tabla 4.3.2. Funcionamiento de la aplicación con submódulos

Submódulo de la aplicación	Descripción	Grupo de usuario
Registrar paciente	En este submódulo se podrá registrar pacientes ya sea por el médico o el mismo administrador o	Administrador Médico

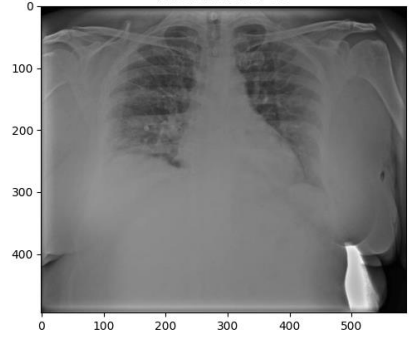
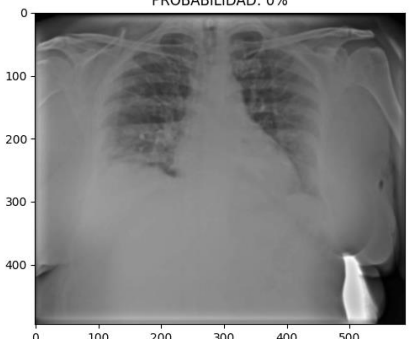

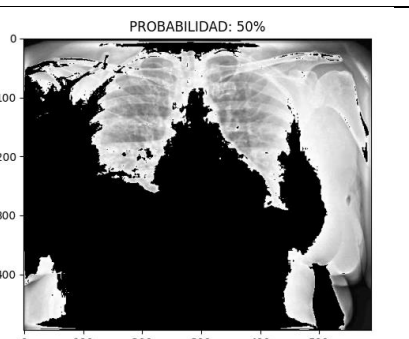
	todo aquel que tenga acceso a la aplicación	
Lista de pacientes	En este submódulo es de los más importantes, ya que aquí se realiza el <i>crud</i> es decir, aquí se puede eliminar, actualizar, insertar la imagen del tórax del paciente y el apartado de analizar imagen.	Administrador Médico
Registrar usuario	En este submódulo es donde se realiza la acción de agregar usuario, ya sea médico o aquel que el administrador vea conveniente	Administrador
Ver lista de usuarios	En este submódulo es para que el administrador vea y gestione la cantidad de	Administrador

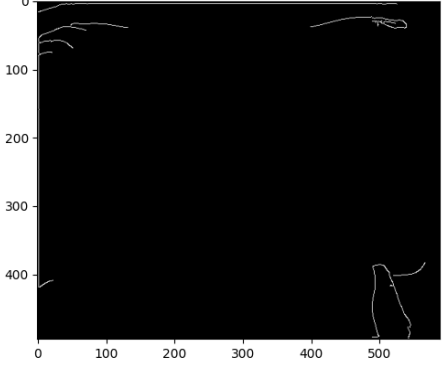
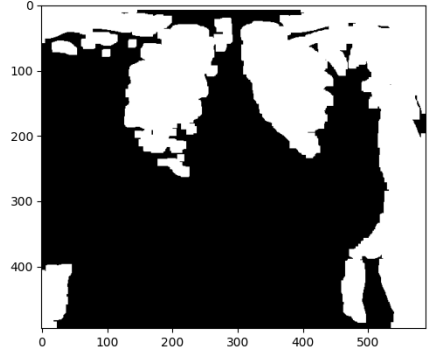
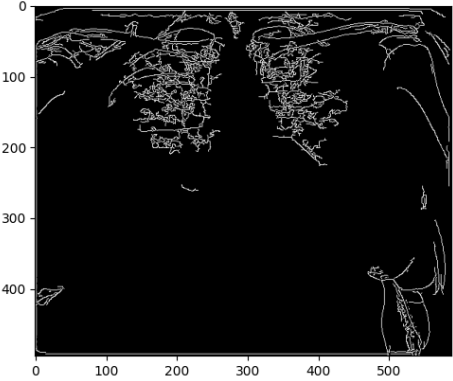
	usuario que tiene acceso a la aplicación	
Total	En este submódulo se podrá ver el total estudios realizados	Administrador Médico
Positivos	En este submódulo se podrá ver el total estudios que ha dado positivo a <i>COVID-19</i>	Administrador Médico
Sospechoso	En este submódulo se podrá ver el total estudios que el médico considere sospechosos	Administrador Médico
Negativos	En este submódulo se podrá ver el total estudios que ha dado negativo a <i>COVID-19</i>	Administrador Médico

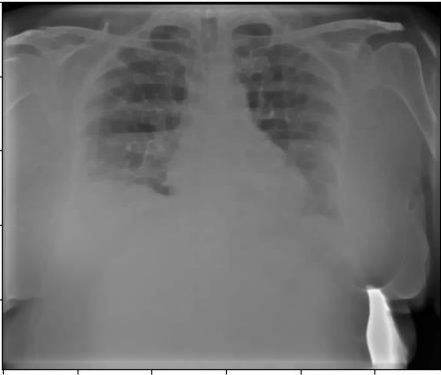
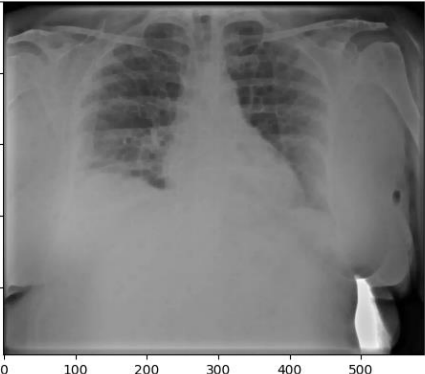
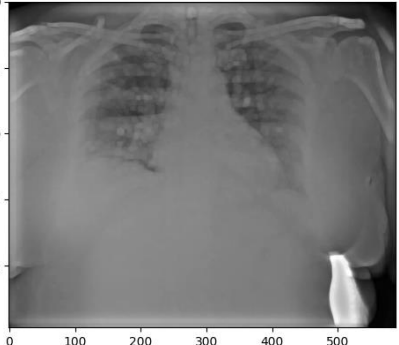
Fuente: Elaboración propia (2023).

Tabla 4.3.2 Técnicas de procesamiento de imágenes

FASES	TÉCNICAS	LIBRERÍA	IMAGEN PROCESADA
-------	----------	----------	------------------

Preprocesamiento	Modelo de color	<p>Cv2</p> <p><i>Numpy</i></p> <p><i>Matplotlib</i></p> <p><i>Pyplot</i></p>	<p>PROBABILIDAD: 0%</p> 
	Ruido	<p>Cv2</p> <p><i>Numpy</i></p> <p><i>Matplotlib</i></p> <p><i>Pyplot</i></p>	<p>PROBABILIDAD: 0%</p> 
	Binarización	<p>Cv2</p> <p><i>Numpy</i></p> <p><i>Matplotlib</i></p> <p><i>Pyplot</i></p>	<p>PROBABILIDAD: 0%. Valor umbral 118.0</p> 
	Umbralización	<p>Cv2</p> <p><i>Numpy</i></p> <p><i>Matplotlib</i></p> <p><i>Pyplot</i></p>	<p>PROBABILIDAD: 50%</p> 

Segmentación	Bordes y contornos	<code>cv2</code> <code>numpy</code> <code>cv2_imshow</code>	<p>PROBABILIDAD: 0%. Se encontraron 8 objetos</p>  <p>A plot showing edge detection results. The y-axis ranges from 0 to 400, and the x-axis ranges from 0 to 500. The image is mostly black with white lines representing edges. A text label at the top reads 'PROBABILIDAD: 0%. Se encontraron 8 objetos'.</p>
	Similitud	<code>cv2</code> <code>numpy</code> <code>cv2_imshow</code>	<p>PROBABILIDAD: 0%</p>  <p>A plot showing similarity segmentation results. The y-axis ranges from 0 to 400, and the x-axis ranges from 0 to 500. The image is mostly black with white regions representing the segmented objects. A text label at the top reads 'PROBABILIDAD: 0%'.</p>
	Discontinuidad	<code>cv2</code> <code>numpy</code> <code>cv2_imshow</code>	<p>PROBABILIDAD: 0%</p>  <p>A plot showing discontinuity segmentation results. The y-axis ranges from 0 to 400, and the x-axis ranges from 0 to 500. The image is mostly black with white regions representing the segmented objects. A text label at the top reads 'PROBABILIDAD: 0%'.</p>

Post-procesamiento	Operaciones morfológicas	<p>Cv2</p> <p><i>Numpy</i></p> <p><i>Matplotlib</i></p>	<p>PROBABILIDAD: 0%</p>  <p>0 100 200 300 400 500</p>
	Erosión	<p>Cv2</p> <p><i>Numpy</i></p> <p><i>Matplotlib</i></p>	<p>PROBABILIDAD: 0%</p>  <p>0 100 200 300 400 500</p>
	Dilatación	<p>Cv2</p> <p><i>Numpy</i></p> <p><i>Matplotlib</i></p>	 <p>0 100 200 300 400 500</p>

Fuente: Elaboración propia (2023).

4.4 Descripción detallada de las técnicas de procesamiento de imágenes

4.4.1 Preprocesamiento

Modelo de color

Para realizar el modelo de color se utilizaron las siguientes funciones:

- `imread()` que permite cargar una imagen desde un archivo específico.
`cv2.imread(path, flag)`. `COLOR_RGB2GRAY` código de la conversión del espacio de color en este caso la conversión es de RGB a gris.
`img2.shape` características de `img2`, dimensiones y canal.
- `img2` variable usada para almacenar la conversión de espacio de color de `img`.
- `cvtColor` tiene dos argumentos `img` y `COLOR_RGB2GRAY`
- `Cv2.imshow`: El método `cv2.imshow()` se utiliza para mostrar una imagen en una ventana, la ventana se ajusta automáticamente al tamaño de la imagen.
- NumPy es el que permite una generación y manejo de datos extremadamente.
NumPy tiene su propia estructura de datos incorporada llamado arreglo.

Ruido

En el siguiente procedimiento se utilizó un núcleo gaussiano. Lo cual se hizo mediante la función, `cv2.GaussianBlur()`. Introduciendo como argumentos, además de la imagen fuente, el alto y ancho del kernel, que tendrá que ser positivo e impar. Finalmente se introducirá la desviación estándar en las direcciones X e Y, `sigmaX` y `sigmaY`, respectivamente. Se produjo una reducción considerable del ruido en la imagen, sin producirse un excesivo proceso de difuminado sobre los contornos.

Binarización.

Para binarizar la imagen se empleó el método *threshold* (imagen, umbral, máximo valor de píxeles, tipo binarización) en el cual está albergado en la biblioteca cv2. El umbral fue un valor elegido por el usuario y la binarización que se utilizó fue *cv2.THRESH_BINARY*. El motivo por el que puso un guion bajo al principio de la igualdad es porque el método *threshold()* devuelve dos valores, y el único que interesó fue el segundo.

Umbralización.

La función que se utilizó fue *cv2.threshold*. Si el valor del *pixel* es mayor al valor del umbral, se le asigna un valor, puede ser blanco, de otro modo se le asigna otro valor, puede ser negro. El primer argumento es la imagen fuente, que debe encontrarse en escala de grises. El segundo argumento es el valor del umbral que se usa para calificar los valores de *píxeles*. El tercer argumento es el *maxVal* el cual representa el valor dado si el valor del *pixel* es mayor que (a veces menor que) el valor del umbral. La umbralización simple consiste en modificar el valor de un *píxel* si sobrepasa el umbral establecido. Existen varios tipos de umbralización simple como:

THRESH_BINARY, *THRESH_BINARY_INV*, *THRESH_TRUNC*,
THRESH_TOZERO, *THRESH_TOZERO_INV*.5

4.4.2 Segmentación

Tipos de segmentación.

Se utilizó la función `cv2.imread` para la lectura de la imagen, en donde especificó que esta sería transformada a escala de grises con 0.

La segmentación se encuentra dividida en dos: discontinuidad y similitud. Para la discontinuidad primero se encuentra la detección de bordes. Se aplicó un desenfoque *gaussiano* a la imagen con la función `cv.GaussianBlur(image, (3,3), 0)` en el cual `image` es nuestra imagen de entrada, (5,5) es la altura y anchura de nuestro kernel y 0 especifica que nuestra imagen será a escala de grises. Se encontró el gradiente en la dirección X con la función `cv.Sobel(src, cv.CV_16SC1, 1, 0)` en el cual `src` es nuestra imagen de entrada, `CV_16SC1` que especifica la precisión de la imagen a convertir, 1 y 0 dictan la derivada de las direcciones. Hacemos lo mismo para encontrar la gradiente en dirección y con la función `cv.Sobel(src, cv.CV_16SC1, 0, 1)`, esta vez nuestra precisión será `cv.CV_16SC1` y nuestros valores de derivadas serán 1 y 0. Convertimos el valor del gradiente a 8 bits con las funciones `cv.convertScaleAbs(grad_x)` y `cv.convertScaleAbs(grad_y)` en donde `grad_x` y `grad_y` serán nuestras matrices. Se combinó los dos gradientes con la función `cv.addWeighted(x_grad, 0.5, y_grad, 0.5, 0)` en donde `x_grad` y `y_grad` serán nuestras fuentes y 0.5 nuestro alfa y beta. Finalmente, encontramos tres resultados para la detección de bordes. Tenemos nuestro *Canny_Edge 1*, en el cual se utilizó la función `cv.Canny(src1, 50, 100)` donde `src` será nuestra imagen de entrada, 50 es nuestro valor de umbral inferior y 100 nuestro valor de umbral superior. Para *Canny_Edge 2*

se utilizó la función `cv.Canny(grad_x, grad_y, 10, 100)` donde nuestras entradas serán `grad_x` y `grad_y` y nuestros umbrales superiores e inferiores son 10 y 100.

4.4.3 Post-procesamiento

Operaciones morfológicas.

Apertura.

Para la apertura se empleó la función de `cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)`, la apertura es simplemente otro nombre para erosión seguida de dilatación.

Erosión

Se utilizó la función `cv2.imread` para la lectura de la imagen, en donde especificó que esta sería transformada a escala de grises con 0. A continuación se creó el *kernel* con la función `np.ones((5, 5), np.uint8)` en la cual (5,5) es la forma de nuestra matriz, y `np.uint8` es el tipo de dato que tiene nuestra matriz. Después aplicamos la erosión a nuestra imagen con la función `cv2.erode(image, kernel)` donde *image* es nuestra imagen de entrada, y *kernel* es, valga la redundancia, nuestro kernel.

Dilatación

La dilatación es lo opuesto a la erosión; agrega píxeles a los límites de un objeto en la imagen dada. A continuación, se creó el kernel con la función `np.ones((5, 5), np.uint8)` en la cual (5,5) es la forma de nuestra matriz, y `np.uint8` es el tipo de dato

que tiene nuestra matriz. Después usamos la función dilate () para aplicar la dilatación a un objeto de imagen. Los argumentos son la imagen original, el núcleo y las iteraciones donde sugieren cuántas veces desean ejecutar el proceso.

4.5 Requerimientos del sistema

4.5.1 Específicos

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Subir imagen
Características:	Se subirá la imagen a analizar a la página <i>web</i> .
Descripción del requerimiento:	El usuario podrá subir la imagen como un archivo externo, para poder usarlo para su análisis próximo, con las especificaciones del paciente que se va a atender. Así mismo, subiéndola a la base de datos de nuestro sistema.
Prioridad del requerimiento:	
Alta	

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Análisis de imágenes

Características:	Se analizarán las imágenes cargadas.
Descripción del requerimiento:	Se analizará la imagen a partir de técnicas de procesamiento de imágenes, realizando cada una de estas etapas por medio de técnicas de inteligencia artificial (redes neuronales).
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Diagnóstico preliminar
Características:	Dará un diagnóstico preliminar a partir del análisis.
Descripción del requerimiento:	A partir del análisis realizado, el sistema podrá dar un diagnóstico preliminar donde se dará una especificación del análisis de las imágenes. Prediciendo si es un paciente positivo o negativo a <i>COVID-19</i>
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF04
--	------

Nombre del Requerimiento:	Lista de pacientes
Características:	Se visualizarán todos los pacientes que ha atendido usuario.
Descripción del requerimiento:	El usuario tendrá acceso a una lista de los pacientes que ha atendido con datos de este y su posible diagnóstico de la enfermedad de <i>COVID-19</i>
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Análisis pacientes
Características:	Se mostrará un resumen del paciente y sus imágenes analizadas
Descripción del requerimiento:	Se mostrarán los datos generales del paciente, las imágenes ya analizadas junto con un resumen de cada ellas, y el diagnóstico preliminar del paciente.
Prioridad del requerimiento: Alta	

4.5.2 No específicos

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Interfaz del sistema.
Características:	El sistema presentará una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema.
Descripción del requerimiento:	El sistema debe tener una interfaz de uso intuitiva y sencilla.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Diagnósticos pacientes
Características:	Se mostrará un análisis previo de las imágenes ya analizadas del paciente.
Descripción del requerimiento:	Se mostrará una breve descripción del análisis realizado del diagnóstico, es decir, habrá un porcentaje de las anomalías encontradas.
Prioridad del requerimiento:	

Alta

4.6 Diagrama de secuencias

4.6.1 Inicio de sesión

El diagrama de secuencia es uno de los tantos usos de los diagramas UML, que particularmente muestran de forma gráfica la interacción entre el usuario (Actor) y el sistema, de tal manera que utiliza símbolos para representar partes del sistema, estos están hechos por una función específica del sistema, como se observa en la ilustración 15, el actor es el médico quien intenta validar datos de inicio de sesión para lo que el diagrama representa los dos posibles caminos dependiendo de un valor verdadero o falso, nunca superponiendo una parte de encima de la otra por otro lado, en caso de validar, este puede consultar datos y recibirlos desde la base de datos.

Diagrama de secuencia - Inicio de sesión

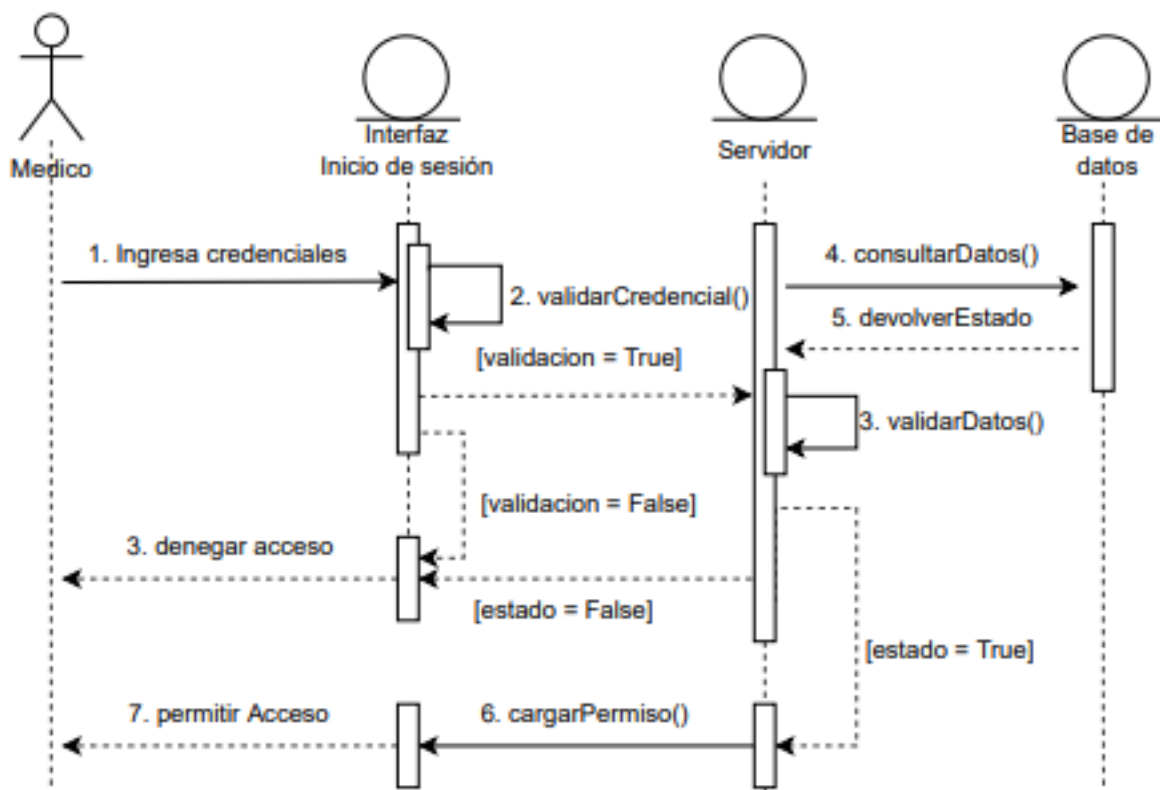


Ilustración 22. Diagrama de secuencia de inicio de sesión. Fuente: Propia (2023)

4.6.2 Consulta del historial de diagnósticos

En este diagrama, se puede observar que el médico ingresa una búsqueda por filtro, está en caso de ser validada, hace una petición a la base de datos, después esta envía los diagnósticos que cumplan dicha condición, el servidor renderiza los datos obtenidos para finalmente enlistarlos en la vista correspondiente.

Diagrama de secuencia - Consulta de historial de diagnósticos

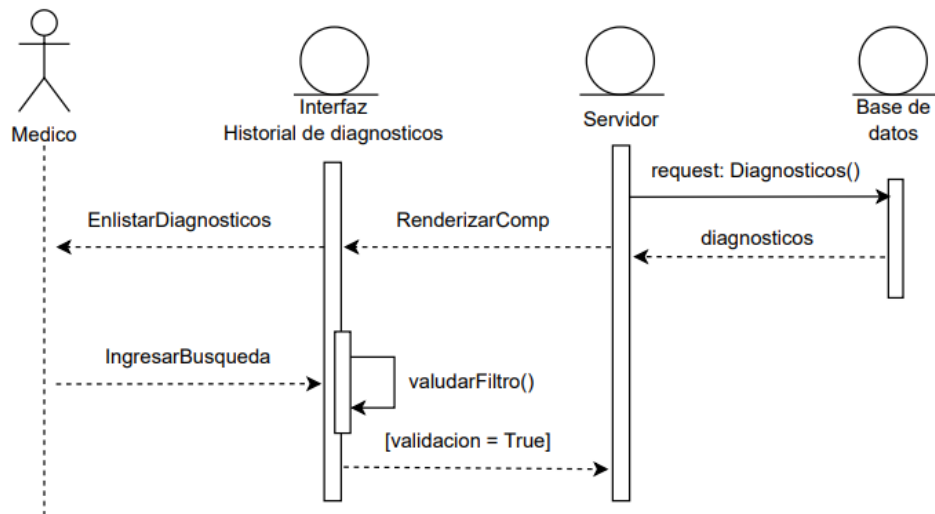


Ilustración 23. Diagrama de secuencia de los Diagnósticos. Fuente: Propia (2023).

4.7 Procedimientos realizados

Al realizar el proyecto el primer paso fue la investigación de cómo encontrar características para detectar el virus *COVID-19*, partiendo de esto, se usó el lenguaje *Python* para el *backend* de la aplicación *web* y para el *frond-end* se usó el lenguaje *JavaScript*, *HTML*, *CSS* y la herramienta *Booststrap*.

Se usó el framework *Flask*, ya que a diferencia de su hermano mayor que es *Django*, este es un *microframework* que ofrece todo lo necesario para la elaboración de este proyecto.

Se usó *MySQL* como nuestro gestor de base de datos, ya que es un gestor que cumple todas las características y exigencia de la aplicación *web*. Se usó el entorno virtual de *Python* en cuál crea una carpeta *env* con todos los componentes del *microframework* se dividió en dos carpetas *env* que esta los componentes y la carpeta *src* que aquí es donde se realizó toda esta carpeta se divide en *static* aquí es donde se encuentra los archivos *CSS* y las imágenes que se usaron para la interfaz y la carpeta *templates* que se encuentra todos los *HTML* de la aplicación y dentro de la carpeta *src* de manera aparte está un archivo *app.py* que es donde se realizó toda la lógica del programa también se encuentra el archivo *bd.py* que es donde se realiza la conexión a la base de datos y los controladores para la comunicación del programa a la base de datos (*MySQL*)

Por otra parte, se usó *Google colab* para implementar las técnicas de preprocesamiento de imágenes que posteriormente se implementó en la *app.py* como se observa en el cuadro 3.

4.7.1 Instalación de entorno de desarrollo y librerías

Este proceso consiste desde la creación de una carpeta en nuestro equipo hasta la instalación de *Python* en nuestro equipo, por ende, primeramente, se procedió a instalarlo, se descargó desde la página principal del desarrollador en su versión 3.10.6 para *Windows 10* de 64 bits. Para poder escribir nuestro código de una manera profesional y organizada se empleó *Visual Studio Code* un editor de código fuente muy conocido, además de contar con las herramientas necesarias para depurar y correr aplicaciones, como se observa en la ilustración 15, se creó una

carpeta y en ella se instalaron todas las herramientas pertinentes para poder realizar la APP.

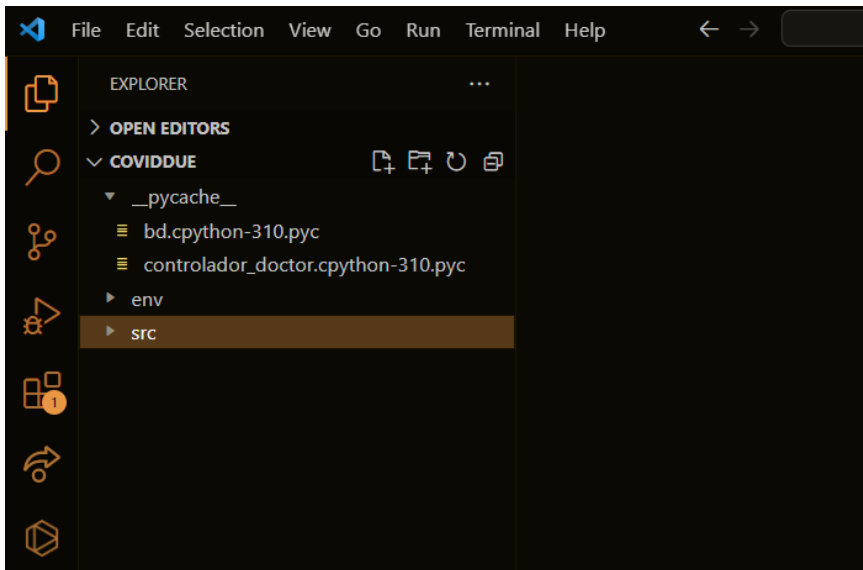


Ilustración 24 . Carpeta creada para el proyecto. Fuente: Propia (2023)

Después de la creación de la carpeta, se procede a crear el entorno virtual con el fin de poder probar los distintos paquetes de *Python* y poder aislar específicamente en una dirección de nuestro sistema, en este caso fue el directorio *SRC* de la carpeta principal del proyecto, mediante la consola de *Visual Studio Code* se ejecutaron los códigos para instalarlo, donde ya se dispone de la dirección donde se instaló y se ejecutó el código, *pip install virtualenv* para después activar el entorno por otro lado, para instalar *Flask* se utilizó *pip install flask*, para poder así contar con sus herramientas para desarrollo, se instaló *flask_mysqldb* la herramienta utilizada para la conexión con la base de datos *MYSQL*, entre otras como son: *TensorFlow* para redes neuronales, *flask_login* para controlar las sesiones de los usuarios, *werkzeug* para poder implementar medidas de seguridad, *numpy* e *matplotlib* con el objetivo de poder analizar y mostrar las imágenes de los pacientes, un ejemplo de ello lo

podemos ver en la siguiente ilustración donde se observa cómo se instaló *TensorFlow* y sus paquetes. (Véase en la ilustración 16)

```
Installing collected packages: tensorflow
Successfully installed tensorflow-2.12.0

[notice] A new release of pip is available: 23.0.1 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\jaime\OneDrive\Escritorio\Ultima copia\COVIDDUE\src>
```

Ilustración 25. Instalación exitosa de *TensorFlow*. Fuente: Propia (2023).

4.7.2 Creación de la base de datos en *MySQL*

Para la base de datos se creó un esquema llamado *COVID-19* en *MySQL* el cual está compuesto de tres tablas como se puede ver en la ilustración 17, la primera fue nombrada como “paciente”, donde se guardan los datos del paciente, otra tabla es la tabla llamada “admin” en esta tabla se guardan los datos de aquellos usuarios que van a utilizar el sistema en este caso se guardan sus datos, entre estos datos se tienen la contraseña y el nombre de usuario, estos son utilizados para iniciar sesión en el sistema por otra parte, cada tupla tiene un campo llamado “is_admin” el cual es de tipo booleano por ende tiene solo valores entre 0 y 1, en pocas palabras falso o verdadero, con el fin de poder diferenciar las sesiones y así poder excluir ciertas funciones al usuario que no tiene rol de administrador.

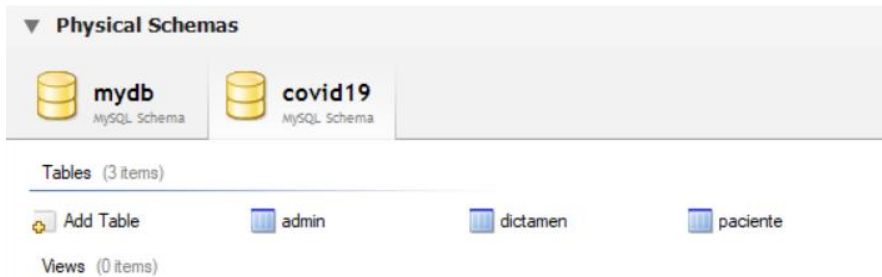


Ilustración 26. Esquema “COVID-19”, creado para guardar los datos de la aplicación. Fuente: Propia (2023).

La última tabla es “dictamen” donde se guardaron los datos de cada dictamen realizado, entre estos se tuvieron el id, la fecha, el comentario por parte del médico y como tiene una relación de uno a muchos de esta a la tabla “paciente” y así también con la tabla “admin” para poder tener los registros de acuerdo con los pacientes y los usuarios que realizaron estos dictámenes, en la siguiente ilustración se puede observar la relación entre las tablas también la nomenclatura de cada una de ellas, teniendo los nombres de cada atributo así como los tipos de atributo correspondientes y su tamaño máximo.

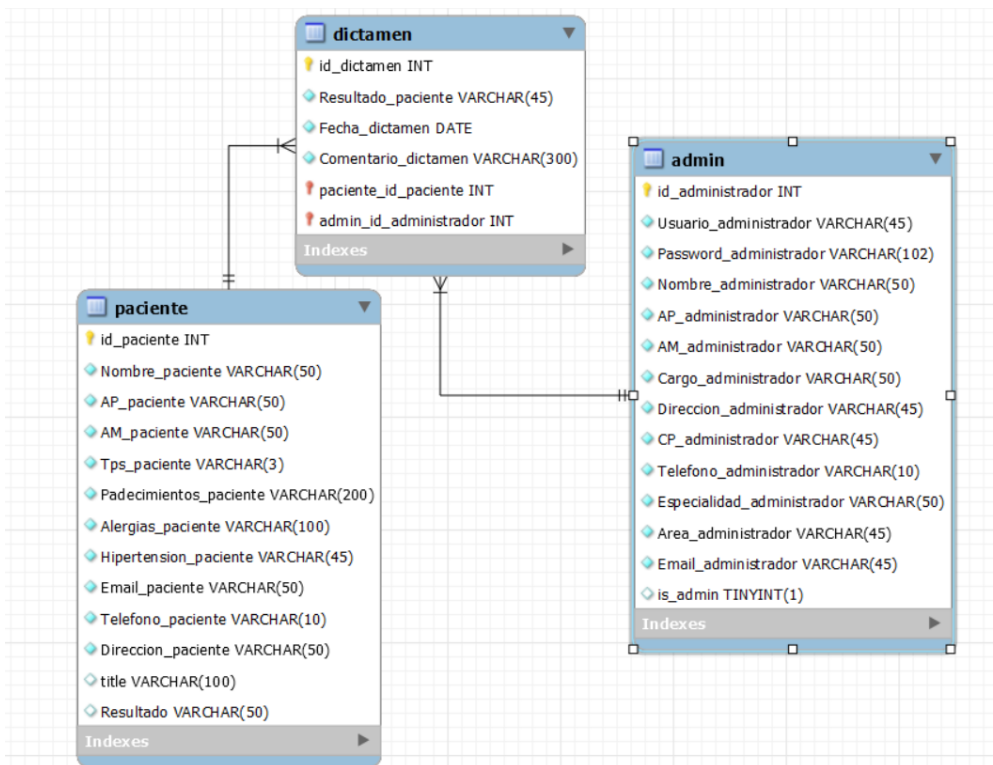


Ilustración 27. Diagrama de la base de datos utilizada. Fuente: Propia (2023).

4.7.3 Creación de rutas

En *Flask* existen métodos para crear rutas, para ello se dispuso de instanciar la app con el método `_name_` con el fin de que el sistema pueda saber dónde se encontraron las rutas, una vez realizado esta configuración solo se utiliza una estructura general, donde se utilizó el identificador `@` y el método `route` donde se instancia el nombre de la plantilla a realizar, en estos casos los nombres fueron de acuerdo a la función de cada pantalla, el siguiente ejemplo se puede ver que la plantilla tiene el nombre de 'Registro' en referencia a la función de registrar usuarios,

además de declarar un parámetro para el método que se utilizó, esta vez fue el método *GET*, este es el encargado de obtener datos, dentro de la función de esta vista llamada registrar(), se hace una referencia al parámetro *is_admin* correspondiente al registro de los usuarios ya que solo sí es administrador se podrá renderizar la vista.(Véase en el cuadro 1)

```
@app.route('/Registro', methods=['GET'])
@login_required
def registrar():
    if current_user.is_admin:
        return render_template('/Registro.html')
    else:
        return redirect(url_for('warning'))
```

Cuadro 1. Renderizar vista. Fuente. Propia. Adquirida del analizador COVID-19(2023).

Todas las rutas utilizaron esta estructura básica, las diferencias fueron los métodos para obtener o enviar datos o en incluso la declaración de variables haciendo uso de otras funciones ya declaradas.

4.7.4 Controladores para la base de datos

Se decidió llamar a este apartado de funciones como controladores, ya que fueron las funciones encargadas de ejercer control sobre la base de datos de acuerdo a la función que se determinó de acuerdo a la vista en las que se emplearon, estas llevaron una estructura simple donde se declara el nombre y por ende los parámetros que sirvieron para alojar los datos de cada atributo de la base de datos, además de

instanciar al método de conexión de base de datos con una variable llamada conexión para invocar el método cursor de conexión de base de datos con el objetivo de ejecutar una sentencia SQL, en caso de haber sido requerido retornar el valor de una función en una variable previamente declarada e instancia da al cursor esto en la selección de datos. En el siguiente ejemplo se puede observar que la variable encargada de obtener los datos de la base se declaró como un arreglo ya que esta es una lista en sí con cada posición correspondiente a un atributo, se utilizó un *Select* en un *inner join* entre la tabla paciente y dictamen por medio de la igualación de los id del paciente para así poder tener el dato correspondiente a cada paciente. (Véase en cuadro 2)

```
from bd import obtener_conexion
def obtener_dictamenesN():
    conexion = obtener_conexion()
    dictamenesN = []
    with conexion.cursor() as cursor:
        cursor.execute("""
SELECT id_dictamen,Nombre_paciente,AP_paciente,AM_paciente,Fecha_dictamen,Resultado_paciente,Comentario_dictamen,paciente_id_paciente
      from paciente
      inner join dictamen
      on paciente.id_paciente = dictamen.paciente_id_paciente
      Where Resultado_paciente='Resultado: Negativo a COVID-
19' """)
        dictamenesN = cursor.fetchall()
    conexion.close()
```

Cuadro 2. Controladores. Fuente. Propia. Adquirida del analizador COVID-19(2023).

4.8 Diseño de interfaz

Interfaz de inicio

En la siguiente ilustración se puede observar el diseño de la interfaz de inicio, en este apartado se puede apreciar una bienvenida al sistema, donde se tendrá información del sistema, así como del equipo de trabajo, la paleta de colores utilizada es alusiva a los colores del Instituto Tecnológico Nacional de México, además de contar con una imagen de un médico sosteniendo una radiografía teniendo que ver con los análisis que efectuará el sistema. El botón de inicio de sesión nos llevará a otra pantalla donde se solicitará la información necesaria para acceder. (Véase en la ilustración 15)

Esta es la interfaz de bienvenida de usuario en cuál está el nombre de la aplicación web “Sistema de análisis con *Python* para la detección de COVID-19 a través de imágenes radiológicas de los pacientes”. Por otra parte, cuenta con botón de inicio de sesión en cuál dirigirá a la interfaz que se puede observar en ilustración 21.

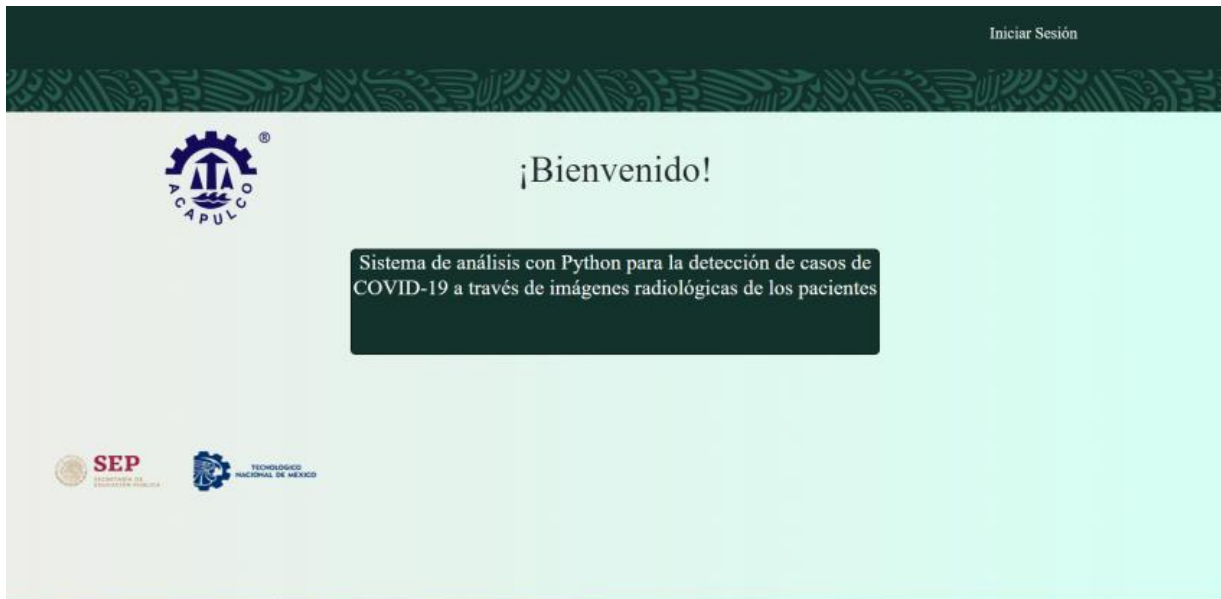


Ilustración 28. Diseño de la interfaz de inicio. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Menú principal

En el menú principal se puede tener acceso a todas las secciones del sistema, se diseñó de tal manera que el usuario no tuviera problemas en identificar cada funcionalidad del sistema, tratando de ser lo más intuitivo posible, en la barra del menú se tienen distintas opciones como son el perfil del especialista que inició sesión, la lista de los pacientes, la sección de diagnosticar que es donde se efectúan los análisis y la lista de estudios realizados. Por último, en la barra de arriba se encuentra el botón de cerrar sesión, el cual nos deslogueará del sistema. (Véase en la ilustración 22)

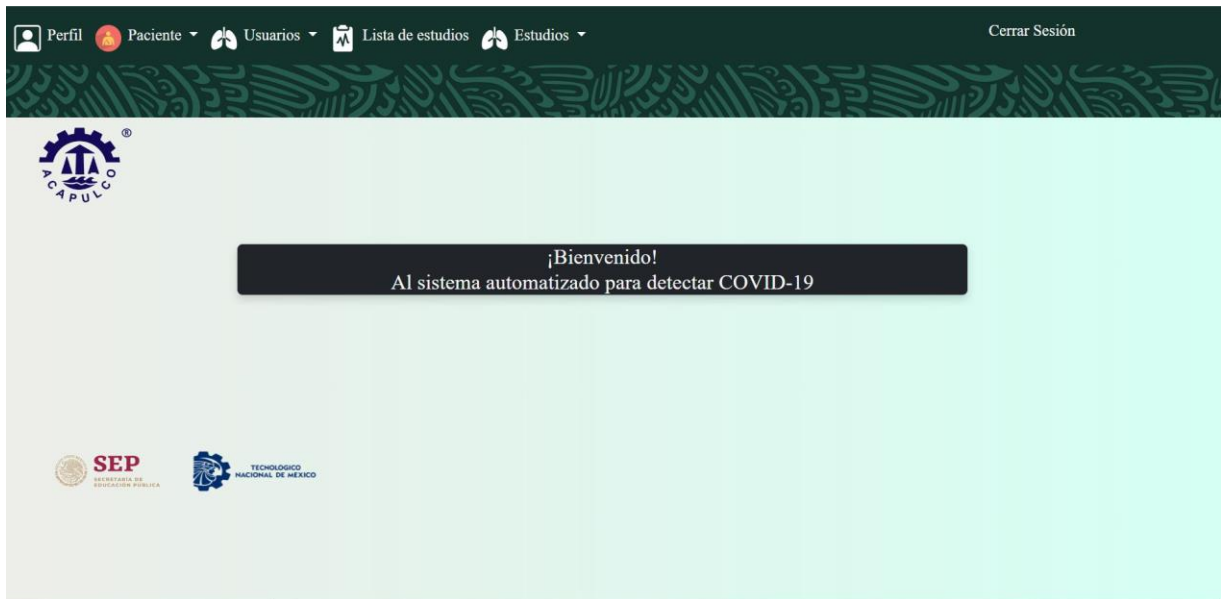


Ilustración 29. Menú principal. Autoría propia. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Lista de pacientes

En esta interfaz se obtendrá la información de todos los pacientes en general, se apreciarán datos importantes como son nombre, número de identificación, fecha de diagnóstico y sexo, en la barra principal también se cuentan con los botones que direccionaran a otras funciones tales como el perfil del usuario que está en sesión del sistema, la lista de estudios y diagnosticar, así también se tiene un buscador de pacientes donde se escribe el nombre del paciente y para poder encontrar su registro en dado caso que sean demasiados.(Véase en la ilustración 23)



Ilustración 30. Lista de los pacientes que ya han tenido diagnóstico. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Diagnóstico

Cuando se haya seleccionado la opción de diagnóstico nos enviará a la pestaña encargada de analizar las imágenes de rayos x, el análisis solicitará el registro del paciente pidiendo datos como su nombre, número de identificación, la fecha en la cual se efectúa el análisis y su edad respectiva, contará con un botón capaz de seleccionar qué acción hacer, por ejemplo seleccionar un método de análisis o incluso tratamiento de imagen por último al dar clic en ejecutar el análisis nos arrojará una ventana donde podremos ver los datos de la imagen y el cálculo de la probabilidad de que el paciente presenta COVID. (Véase en la ilustración 24)

The screenshot shows a web application interface for COVID-19 analysis. At the top, there is a header with the text 'Análisis de imágenes' and the 'GOBIERNO DE MÉXICO' logo on the left. In the center of the header is a green button labeled 'Generar Diagnostico'. On the right is a 'Menú' icon. Below the header, the main content area is titled 'Análisis de COVID-19'. On the left side, there is a dark sidebar with the title 'Técnicas de Análisis de imágenes' and several menu items: 'Técnicas de Pre-procesamiento', 'Técnicas de Post-procesamiento', 'Técnicas de Segmentación', and 'Analizar imagen con I.A.'. At the bottom of the sidebar is a green button labeled 'Analizar imagen'. The main content area contains a form with the following fields: 'Nombre:', 'Apellido paterno:', 'Apellido materno:', 'Tipo de sangre:', 'Padecimientos:', 'Alergias:', and 'Hipertensión:'. Each field is represented by a white input box with a light blue border.

Ilustración 31. Visualización de la imagen. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Paciente

El sistema cuenta con apartado enfocado en la información del paciente, donde prácticamente se guarda el resultado de su análisis previo, el contenedor tendrá su información registrada, conteniendo datos como nombre, el número de identificación del paciente, su sexo y edad, el sistema consultará la imagen ya analizada con su respectivo análisis, con el fin de poder acceder al registro en caso de ser solicitado por el especialista. (Véase en la ilustración 25)

The image shows a web interface for registering a new patient. At the top, there is a dark green header with a 'Menú' icon on the left, the title 'Registrar nuevo Paciente' in the center, and 'Cerrar Sesión' on the right. Below the header is a light green background with a decorative pattern. In the center, there is a dark grey rectangular form with several input fields and a 'Registrar' button. The input fields are arranged in two columns:

Escriba el nombre del paciente	Apellido paterno
Apellido materno	Escriba el tipo de sangre
Escriba si cuenta con algun padecimiento	Escriba sus alergias
¿Tiene Hipertensión?	Escriba algún email
Escriba algún número de teléfono	Escriba la dirección del paciente

At the bottom center of the form is a green button labeled 'Registrar'.

Ilustración 32. Registro del paciente. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Registro Usuarios

En esta interfaz que lleva como objetivo registrar usuarios, en este apartado el administrador agregara al usuario, en este caso puede ser un especialista, médico, enfermera o personal que él crea o que tenga la capacidad de manejar la aplicación de manera ética, ya que hay datos de valioso valor en este caso de los pacientes.(Véase en la ilustración 26).

The image shows a web interface for registering a new user. At the top, there is a dark green header with a 'Menú' icon on the left, the title 'Registrar nuevo Usuario' in the center, and a 'Cerrar Sesión' link on the right. Below the header is a light green background with a decorative pattern. On the left side, there is a logo for 'CAPULCO' featuring a gear and a scale. The main content is a dark grey registration form with the following fields:

- Nombre
- Apellido paterno
- Apellido materno
- Cargo del usuario
- Dirección del usuario
- Código postal
- Teléfono del usuario
- Especialidad
- Área hospitalaria
- Correo electrónico
- Cree una contraseña
- Cree un nombre de usuario
- Seleccione el tipo de usuario: (dropdown menu)

A green 'Registrar' button is located at the bottom right of the form.

Ilustración 33. Registrar usuarios. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Iniciar sesión usuario y administrador

En esta interfaz es la encargada de proporcionar el formulario para iniciar sesión, en cuál va dependiendo del rol, en este caso, administrador o médico podrá dirigirse al inicio de sesión correspondiente a cada uno y validar si sus datos están en lo correcto o si el administrador le dio el acceso a la aplicación en cuáles sus datos se corroborarán en la base de datos. (Véase en la ilustración 27).

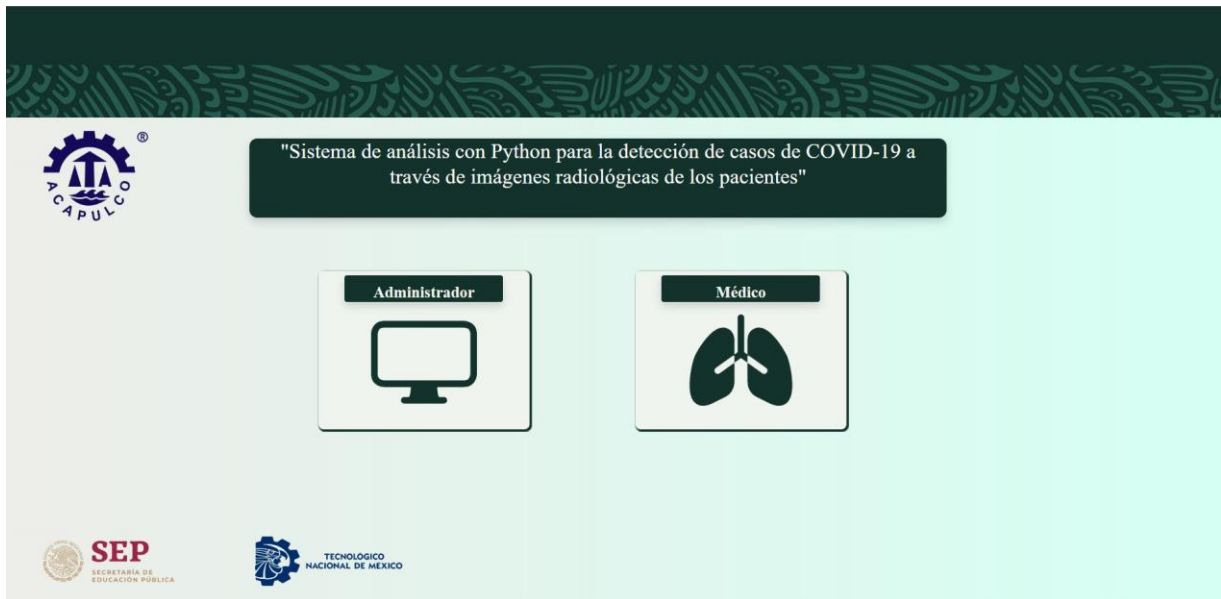


Ilustración 34. Inicio de sesión para administrador y médico. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Iniciar sesión

En esta sección el sistema solicitará al usuario los datos necesarios para acceder al sistema, el correo electrónico o usuario previamente registrado y por ende una contraseña para iniciar la sesión, en dado caso que no esté registrado este podrá registrarse, además de poder recuperar su contraseña, donde se enviara un email al correo registrado de la cuenta, el diseño de la interfaz trata de ser lo más cómodo posible y del lado derecho se tiene el logo del Instituto Tecnológico de Acapulco.(Véase en la ilustración 28).

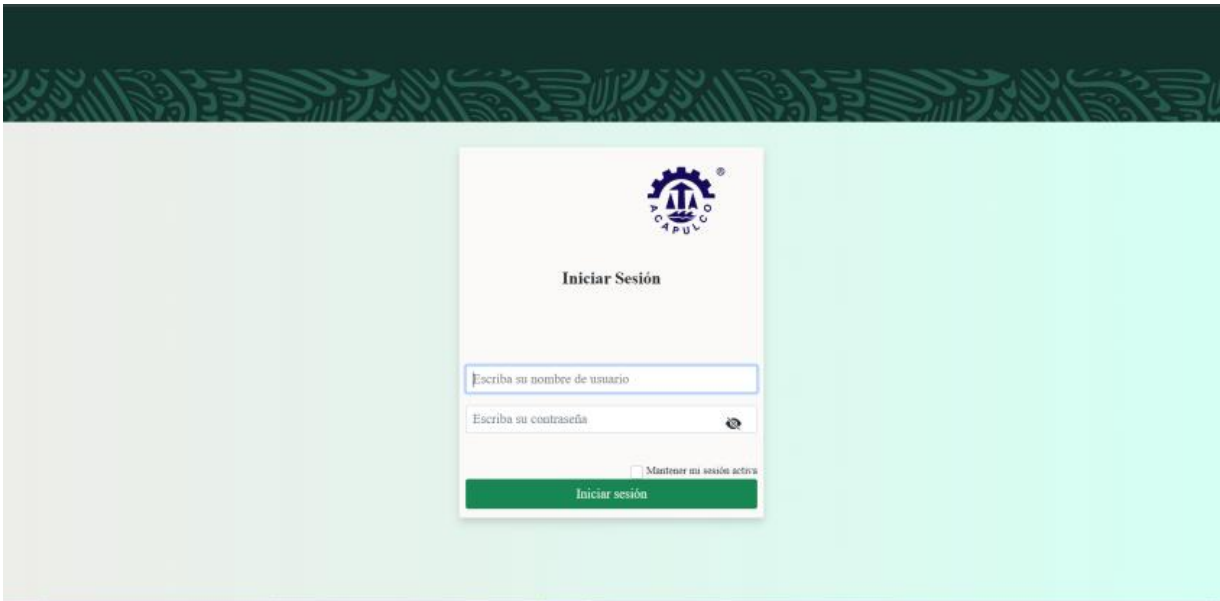


Ilustración 35. Inicio de sesión. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Perfil

En esta interfaz podemos visualizar los datos del usuario como se explicó en figura, puede ser un médico, especialista, enfermera o aquel que administrador vea conveniente. Por otra parte, hay un panel lateral que cuenta con dos botones, uno de ellos lleva por nombre "Editar usuario" en cuál tiene la función de cambiar unos datos del perfil del usuario, este apartado solo está disponible para el administrador, ya que él es responsable del cada usuario agregado y su uso. Por otro lado, tenemos un botón de actualizar contraseña por si desea actualizar dicha contraseña. (Véase en la ilustración, 29).

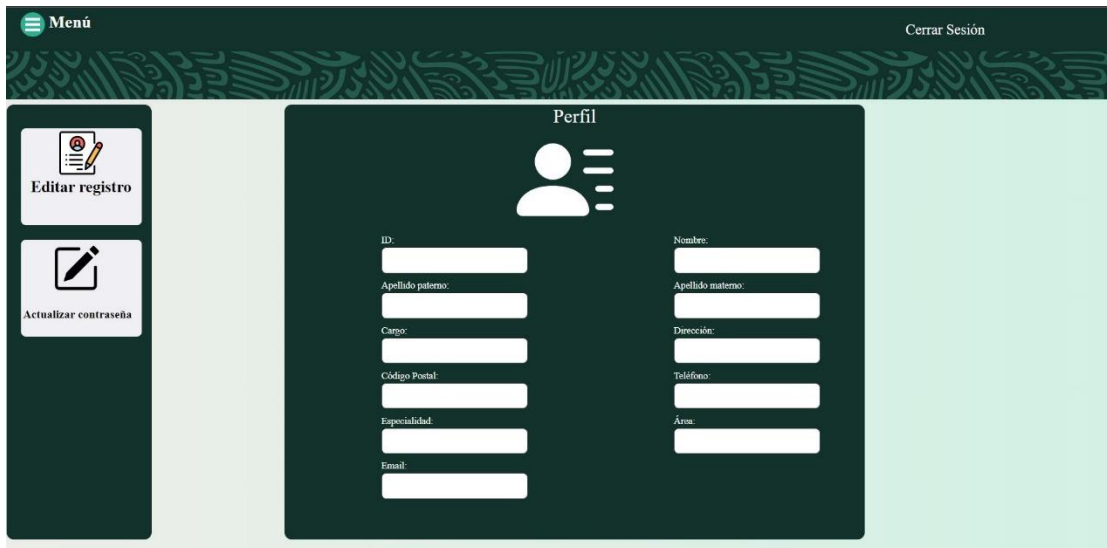


Ilustración 36. *Diseño de la interfaz de perfil. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).*

Editar Perfil

En esta interfaz es donde nos redirecciona el botón, editar perfil en el cual su función o acción a realizar es editar información del usuario o también si hubo un error en su nombre o en cualquier otro campo, así esta se actualiza de manera inmediata en la base de datos. (Véase en la ilustración 28).

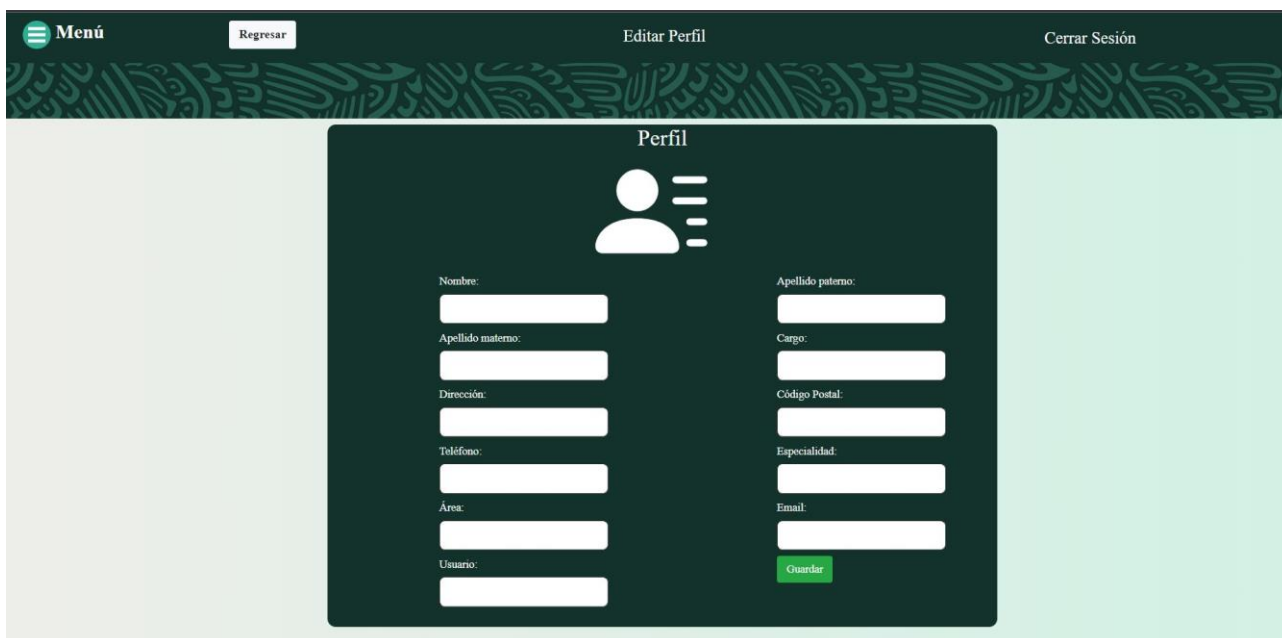


Ilustración 37. Diseño de la interfaz de editar perfil. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Actualizar la contraseña

Esta es la interfaz de actualizar contraseña que se encuentra en el apartado de perfil, a aquí el usuario podrá cambiar su contraseña las veces que él desee, así mismo se incorporó un botón de ver la contraseña y de confirmación para así no haya un error inesperado por parte del usuario al tratar de repetir la contraseña. Esta interfaz se hizo de modo que el usuario se sienta seguro al utilizar la aplicación. (Véase en la ilustración, 31).

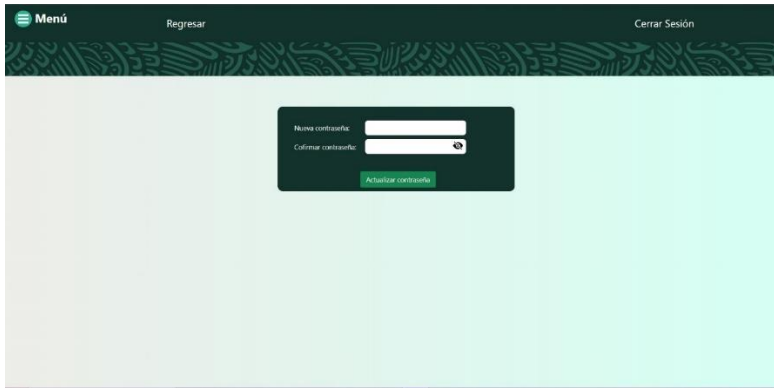


Ilustración 38. Diseño de la interfaz actualizar contraseña. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Menú pacientes

En el módulo Paciente cuenta con dos submódulos en cuál tiene como nombre “Registrar paciente” y “Lista de pacientes” en cuál este módulo tiene como objetivo direccionar a la vista de cada submódulo. (Véase en la ilustración 32)

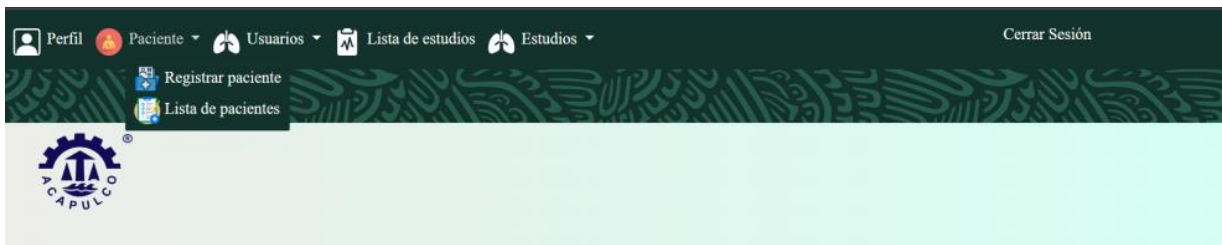


Ilustración 39. Módulo de pacientes. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Menú médico

En el módulo Médico cuenta con dos submódulos en cuál tiene como nombre “Registrar usuario” y “ver lista de usuarios” en cuál este módulo tiene como objetivo direccionar a la vista de cada submódulo. (Véase en la ilustración 33).

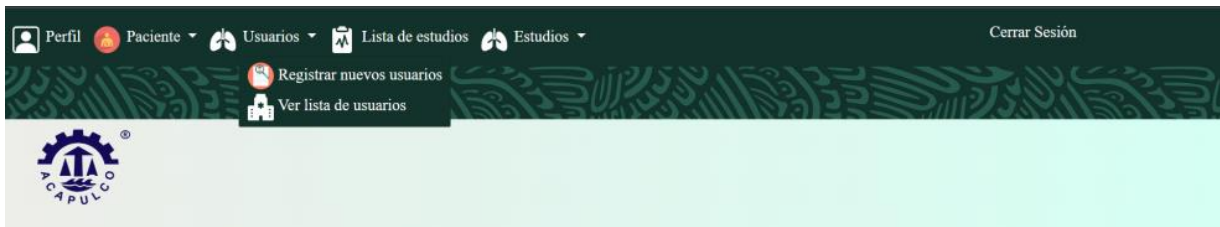


Ilustración 40. Módulo médico. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Menú de estudios

En el módulo Estudios cuenta con dos submódulos en cuál tiene como nombre “Total”, “Positivos”, “Sospechosos” y “Negativos” en cuál este módulo tiene como objetivo direccionar a la vista de cada submódulo. (Véase en la ilustración 34).

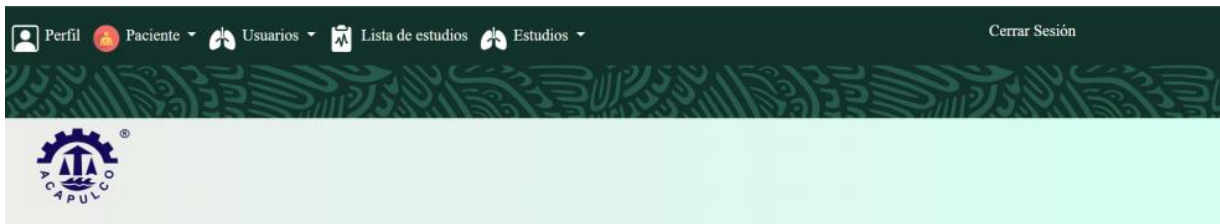


Ilustración 41. Módulo estudios. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Barra administrador

En este apartado muestra cuáles son las opciones disponibles cuando uno inicia sesión como administrador, como se observa tiene todas las funciones del sistema, ya que este es el encargado de administrar la aplicación, es decir él gestiona quien usa la aplicación. (Véase en la ilustración 35)



Ilustración 42. Barra de navegación del administrador. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Barra Médico

En este apartado muestra cuáles son las opciones disponibles cuando uno inicia sesión como médico como se observa no tiene todas las funciones del sistema, ya que este al ser un usuario tiene ciertas limitaciones en comparación con el administrador, ya que este no puede ver el perfil de otros médicos o editar información de otros usuarios. (Véase en la ilustración 36).



Ilustración 43. Barra de navegación del usuario. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Dictámenes positivos

En la siguiente ilustración se muestra la interfaz de dictámenes positivos en cual el administrador como el usuario podrán visualizar todos los pacientes que tiene como resultado al realizar el diagnostico positivo a *COVID-19*, ya sea por las técnicas de preprocesamiento de imágenes o por la inteligencia artificial también podrá ver el dictamen del paciente de manera individual como se visualiza en la ilustración 37. Por otra parte, se puede eliminar el dictamen ya sea porque hubo un error o por un dictamen más reciente, se podrá descargar el dictamen en formato pdf para mayor comodidad del paciente y el usuario se puede visualizar en la ilustración 38. (Véase en la ilustración 37).

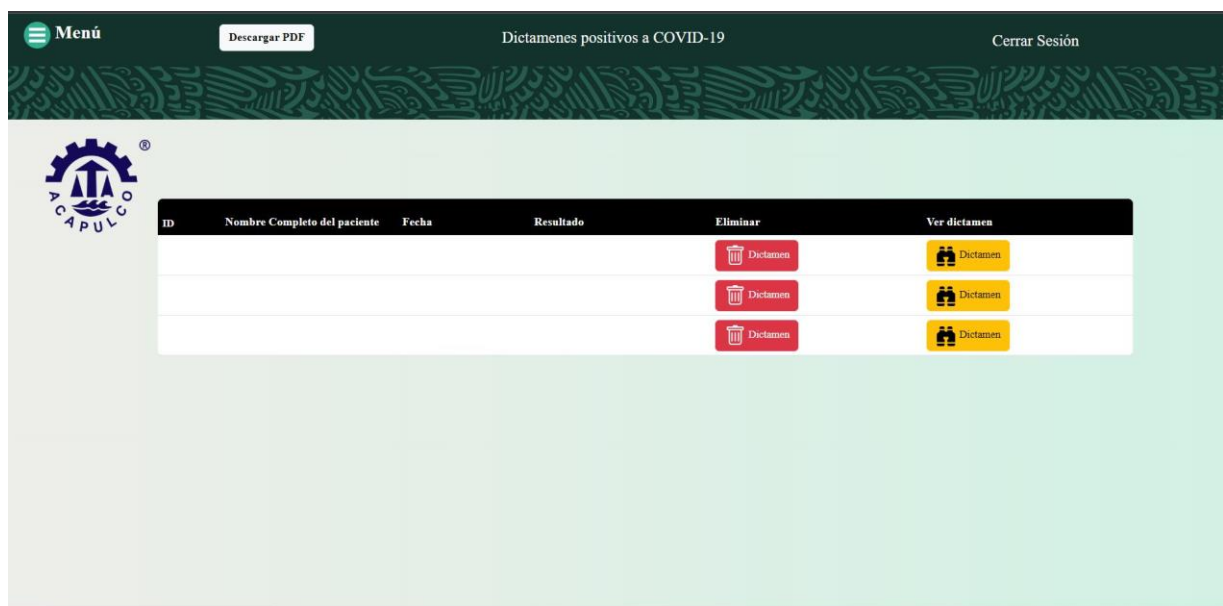


Ilustración 44. *Dictámenes positivos a COVID-19. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).*

Dictámenes negativos

En la siguiente ilustración se muestra la interfaz de dictámenes negativos en cual el administrador como el usuario podrán visualizar todos los pacientes que tiene como

resultado al realizar el diagnostico positivo a COVID-19, ya sea por las técnicas de preprocesamiento de imágenes o por la inteligencia artificial también podrá ver el dictamen del paciente de manera individual como se visualiza en la ilustración 37. Por otra parte, se puede eliminar el dictamen ya sea porque hubo un error o por un dictamen más reciente, se podrá descargar el dictamen en formato pdf para mayor comodidad del paciente y el usuario se puede visualizar en la ilustración 38. (Véase en la ilustración 38)

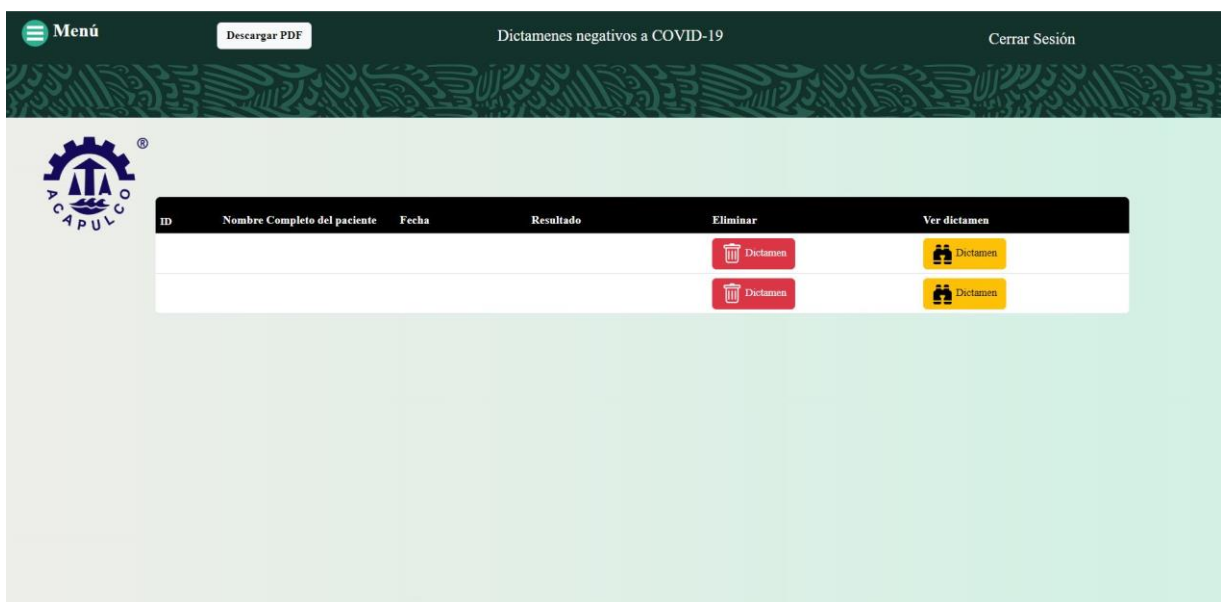


Ilustración 45. Dictámenes Negativos a COVID-19. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Ver dictamen

En esta ilustración se puede observar el dictamen del paciente al cual se le realizo el diagnostico por las técnicas de preprocesamiento de imágenes o inteligencia artificial en cual también hay un apartado de comentario u observaciones aquí el medico podrá describir todo lo que el desee sobre el diagnostico que le arrojó el sistema es

decir si tuvo una duda o no este conforme con el resultado o algún aporte extra al analizador así mismo este tendrá los datos del paciente del usuario quien lo realizo y el resultado del diagnóstico.(Véase en la ilustración 39).

Ver dictamen **Regresar** Menú

Dictamen de COVID-19

Paciente

Nombre: Apellido paterno: Apellido materno:

Tipo de sangre: Padecimientos: Alergias:

Hipertensión:

Información del médico

Nombre: Especialidad:

Resultados

Diagnostico preeliminar:

Dictamen:

Imprimir Dictamen

Ilustración 46. Ver dictamen del paciente. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Dictamen de COVID-19 del paciente

En esta ilustración se puede ver el dictamen del paciente, junto con un apartado para comentarios u observaciones. El médico puede usar esta sección para describir cualquier inquietud que tenga sobre el diagnóstico que ha proporcionado el sistema, incluso si no está de acuerdo con los resultados o si necesita proporcionar información adicional al analizador. En el dictamen también tendrá acceso a los datos del paciente, médico y resultado, así como la fecha que se realizó este. Por otra parte, una opción de guardar dictamen en cual se puede visualizar no solo de un paciente sino de todos los pacientes que se les haya realizado un diagnóstico se puede ver en la ilustración 35. (Véase en la ilustración 40)

Dictamen de COVID-19
Paciente

Nombre: Apellido paterno: Apellido materno:

Tipo de sangre: Padecimientos: Alergias:

Hipertensión:

Información del médico

Nombre: Especialidad:

Resultados

Diagnostico utilizando LA:

Dictamen:
Escriba aquí las observaciones encontradas durante el análisis

Fecha:

Guardar dictamen

Ilustración 47. Dictamen de COVID-19 del paciente. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

4.9 Codificación

En este apartado se muestra algunas partes del código o la más relevantes en siguiente cuadro de texto se muestra el código de iniciar sesión en cual aquí el usuario como el administrador se registran al sistema es por eso que tiene métodos *GET* para renderizar la página en este caso del login y el método *POST* que es el método que pide información al usuario. Por otra parte, se realizó una autenticación para saber quién entra al sistema ya sea el administrador o el usuario este apartado es fundamental ya que tiene diferentes privilegios. (Véase en el cuadro 3)

Cuadro 3. Código de iniciar sesión. Fuente. Propia. Adquirida del analizador COVID-19(2023).

En el siguiente cuadro de texto se puede observar el código de registrar paciente en cual tenemos un método *GET* para redirigir a la página de registrar paciente y por otra parte está el método *POST* para pedirle datos al usuario mediante la página se piden los datos más relevantes o importantes al usuario como el nombre, edad si padece una enfermedad como hipertensión o alguna alergia, estos datos fueron pensados para que el médico se dé una idea general de cómo está de salud el paciente. (Véase en cuadro 4)

```
@app.route('/RegPacientes', methods=['GET'])
@login_required
def regpacientes():
    return render_template('/RegPacientes.html')

@app.route('/RegPacientes', methods=['POST'])
@login_required
def Registro_pacientes():
    Nombre_paciente = request.form['Nombre_paciente']
    AP_paciente = request.form['AP_paciente']
    AM_paciente = request.form['AM_paciente']
    Tps_paciente = request.form['Tps_paciente']
    Padecimientos_paciente = request.form['Padecimientos_paciente']
    Alergias_paciente = request.form['Alergias_paciente']
    Hipertension_paciente = request.form['Hipertension_paciente']
    Email_paciente = request.form['Email_paciente']
```

```

Telefono_paciente = request.form['Telefono_paciente']

Direccion_paciente = request.form['Direccion_paciente']

controlador_doctor.insertar_pacientes(Nombre_paciente,AP_paciente,AM_pacient
e,Tps_paciente,Padecimientos_paciente,Alergias_paciente,
Hipertension_paciente,Email_paciente,Telefono_paciente,Direccion_paciente)

return redirect(url_for('obtener_pacientes'))

```

Cuadro 4. Código de registrar. Fuente. Propia. Adquirida del analizador COVID-19(2023).

En el siguiente cuadro se observa el código de unas de las técnicas de preprocesamiento de imágenes en este caso la técnica de umbralización en cual esta técnica compara los números de *pixeles* blanco y negro a partir de eso da la probabilidad de padecer COVID-19 de acuerdo a los números de pixeles al aplicar la técnica de umbralización.(Véase en cuadro 5)

```

def umbra_an(id):

    paciente = controlador_doctor.obtener_paciente_id(id)

    name=paciente[11].strip() # El strip quita posible retorno de carro

    filename =

rf"C:/Users/jaime/OneDrive/Escritorio/COVIDDUE/src/static/uploads/{name}"

    path = filename

```



```

img = cv2.imread(path,0)

ret,thresh5 = cv2.threshold(img,100,255,cv2.THRESH_TOZERO_INV)

num_black_px = np.sum(thresh5 == 0)

if (num_black_px >= 1105491):

    plt.imshow(thresh5, cmap='gray')

    plt.title("PROBABILIDAD DE PADECER COVID-19: 90%")

elif(num_black_px >= 1000000 and num_black_px <= 1105300):

    plt.imshow(thresh5, cmap='gray')

    plt.title("PROBABILIDAD DE PADECER COVID-19: 70%")

elif(num_black_px >= 100000 and num_black_px <= 199999):

    plt.imshow(thresh5, cmap='gray')

    plt.title("PROBABILIDAD DE PADECER COVID-19: 50%")

elif(num_black_px >= 10000 and num_black_px <= 19999):

    plt.imshow(thresh5, cmap='gray')

    plt.title("PROBABILIDAD DE PADECER COVID-19: 30%")

else:

    plt.imshow(thresh5, cmap='gray')

    plt.title("PROBABILIDAD DE PADECER COVID-19: 0%")

plt.show()

return redirect(url_for('diagnostico', id=paciente[0]))

app.route('/Umbralizacion/<int:id>')

```

Cuadro 5. Código de la técnica de Umbralización. Fuente. Propia. Adquirida del analizador COVID-19(2023).

En el siguiente cuadro se puede observar el código de la inteligencia artificial (red neuronal convolucional) es aquí donde se llama el modelo entrenado y es donde se predice el resultado ya sea “Resultado: Positivo a COVID-19” o “resultado: Negativo a COVID-19”, Por otra parte, se guarda este resultado para que posteriormente se realice un dictamen que mande a llamar el resultado que se obtuvo con el paciente. (Véase en cuadro 6)

```
@app.route('/predict/<int:id>')
def predecir(id):
    paciente = controlador_doctor.obtener_paciente_id(id)
    name=paciente[11].strip() # El strip quita posible retorno de carro
    filename =
rf"C:/Users/jaime/OneDrive/Escritorio/COVIDDUE/src/static/uploads/{name}"
    path = filename

    # Predicción
    preds = model_predict(path, model)
    print(preds[0])

    # x = x.reshape([64, 64]);
    disease_class = ['Resultado: Positivo a COVID-19','Resultado: Negativo a COVID-
19']
    a = preds[0]
```

```

ind=np.argmax(a)

print('Prediction:', disease_class[ind])

result=disease_class[ind]

id_paciente = id

Resultado_paciente = result

controlador_doctor.actualizar_resultado(id_paciente,Resultado_paciente)

flash(result)

return render_template('diagnostico.html',result=result,paciente=paciente)

```

Cuadro 6. Código de la inteligencia artificial. Fuente. Propia. Adquirida del analizador COVID-19(2023).

En el siguiente cuadro se muestra el código que hace posible la conexión de la base de datos a nuestra *API*, lo único que se requiere es el usuario de la base de datos, el *password* y el nombre del *databases* además se agregó un *SECRET_KEY* que su función es encriptar la contraseña del usuario. (Véase en el cuadro 7)

```

import pymysql

class Config:

    SECRET_KEY = 'B!1weNAt1T^%kvhUI*S^'

def obtener_conexion():

```

```
return pymysql.connect(host='localhost',  
user='root',  
passwd='1229',  
db='covid19',  
)
```

Cuadro 7. Código de la conexión a la base de datos. Fuente. Propia. Adquirida del analizador COVID-19(2023).

Capítulo 5: Pruebas y Resultados

5.1 Prueba de interfaz

En la siguiente ilustración se muestra la interfaz de iniciar sesión, en este apartado se comprueba o se verifica si está registrado por el administrador y corrobora que los datos estén correctos como el nombre de usuario y contraseña que se almacenaron en la base de datos. (Véase en la ilustración 41).

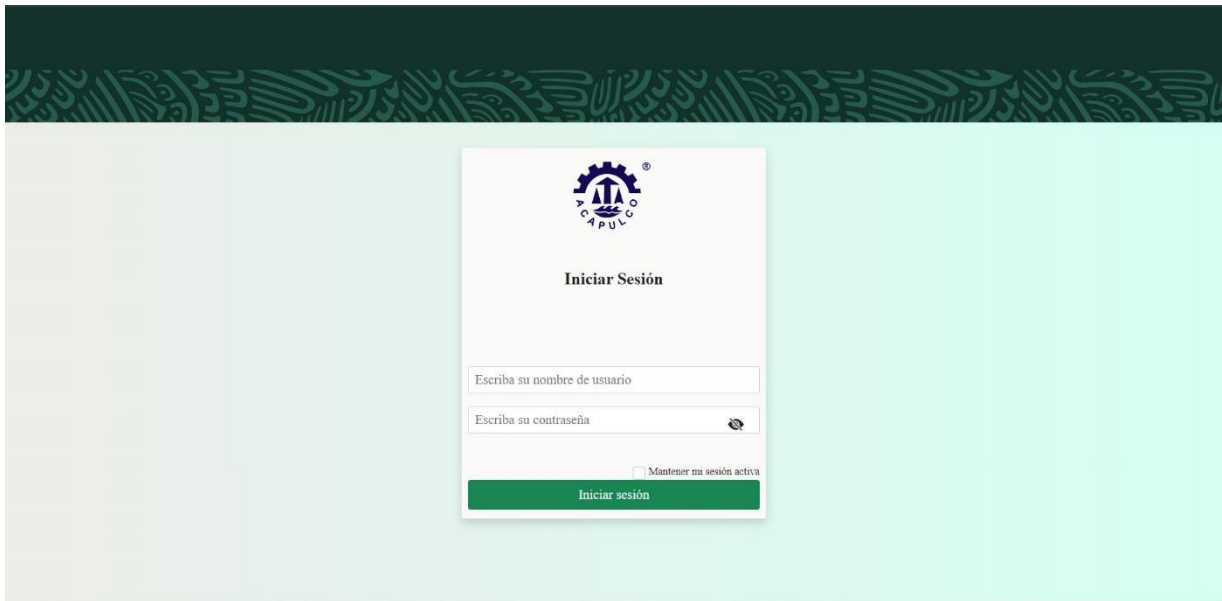


Ilustración 48. Resultado del inicio de sesión. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

En la siguiente ilustración se visualiza la interfaz de bienvenida o menú del administrador, en cuál tiene todas las opciones disponibles al hacer el administrador, en este caso así mismo el sistema se percata al iniciar sesión si fue un usuario o un administrador. Por otra parte, un plus de nuestra parte fue ponerle el nombre completo del usuario que ingresa al sistema, esos datos se traen desde la base de datos. (Véase en la ilustración 42)

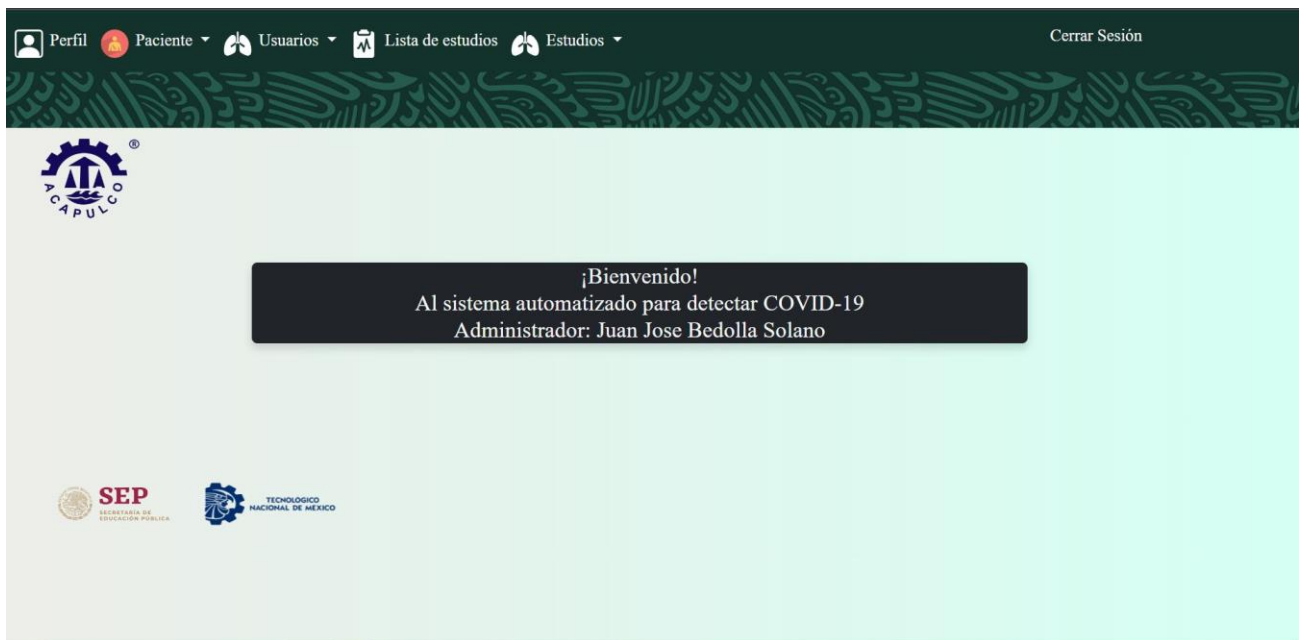


Ilustración 49. Resultado de interfaz de bienvenida con el nombre del usuario logueado. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

En la siguiente Ilustración se muestra la interfaz de registrar usuario y el único que tiene acceso a esta interfaz es administrador, ya que es único usuario en agregar a otros usuarios. Por otra parte, no es necesario que sean médico, especialista o enfermeras, puede ser cualquier persona que el administrador vea que tenga la responsabilidad y la ética de usar el programa. (Véase en la ilustración 43)

Menú Registrar nuevo Usuario Cerrar Sesión

Ilustración 50. Resultado de la interfaz de Registrar usuario. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

En la siguiente figura se muestra la interfaz de perfil en cuál aquí se podrá editar información del usuario, ya sea para cambiar la contraseña o si hubo un error al poner su nombre. (Véase en la ilustración 44)

Menú Perfil Cerrar Sesión

ID: Nombre:
Apellido paterno: Apellido materno:
Cargo: Dirección:
Código Postal: Teléfono:
Especialidad: Área:
Email:

Ilustración 51. Resultado de la interfaz editar perfil. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

En la siguiente figura se visualiza la interfaz de registrar pacientes, esta interfaz la puede usar o visualizar el administrador como el usuario (Médico, especialista o enfermera). (Véase en la ilustración 45)

Menú Registrar nuevo Paciente Cerrar Sesión

Hiromi	Loeza
Silverio	o+
Ninguno	Ninguna
NO	hiromiloe@gmail.com
7443441822	calle 14 mza 26 lote 42 cp.39750

Registrar

Ilustración 52. Resultado de la interfaz Registro paciente. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

En la siguiente ilustración se visualiza la interfaz de editar paciente, aquí se podrá actualizar la información del paciente o si hubo un error en unos de los datos, cambiarlo. (Véase en la ilustración 46)

Nombre:	Apellido paterno:
Jesus	Velazquez
Apellido materno:	Tipo de sangre:
Hernandez	O+
Pacimientos:	Alergias:
Nada	Trabaja:
Hipertensión:	Email:
Si	jesus9@gmail.com
Teléfono:	Dirección:
7443441822	col. Renacimiento calle el puto 3
<input type="button" value="Guardar"/>	<input type="button" value="Volver"/>

Ilustración 53. Resultado de la interfaz editar paciente. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

En la siguiente ilustración se visualiza la lista de usuarios (Médico, especialista o enfermera) en esta interfaz solo la podrá ver o usar el administrador, ya que este podrá visualizar todos los usuarios que tiene acceso a la aplicación. Por otro lado, él podrá ir a la interfaz de editar y también podrá eliminar usuario ya sea por mal uso de la aplicación o cambio de área de trabajo. (Véase en la ilustración 47).

Nombre paciente	Apellido paterno	Apellido materno	Cargo	Dirección	CP	Teléfono	Especialidad	Área	Email	Tipo de usuario	Acciones
Juan	Rosas	Flores	Jefe	calle 14 mza 26 lote 43	39700	1234567890	Covid	pulmon	juan123@hotmail.com	Médico	Eliminar Editar
Juan Jose	Bedolla	Solano	Jefe	calle 14 mza 26 lote 43	39700	1234567810	Neomonología Covid-19		Bedolla@hotmail.com	Administrador	Eliminar Editar
Sergio	Mendoza	Lopez	Jefe	calle 1 mza 12 lote 42	39701	1234567822	Neomonología Covid-19		sergio123@hotmail.com	Administrador	Eliminar Editar
Francisco	Perez	Lopez	Jefe	calle 1 mza 12 lote 42	39700	123457033	Neomonología Covid-19		fran123@hotmail.com	Médico	Eliminar Editar
Cesar	Salgado	Orozco	Cirujia	calle 1 mza 12 lote 41	39700	1234567801	Neomonología Covid		cesar123@hotmail.com	Médico	Eliminar Editar

Ilustración 54. Visualización de la lista de médicos registrados al sistema. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

En esta ilustración se visualiza la interfaz de registro de pacientes en cuál tiene acceso el administrador y el usuario en cuál podrá eliminar, editar información,

insertar la imagen y el apartado de analizar la imagen, pero así mismo mostrará la imagen insertada(Véase en la ilustración 48)

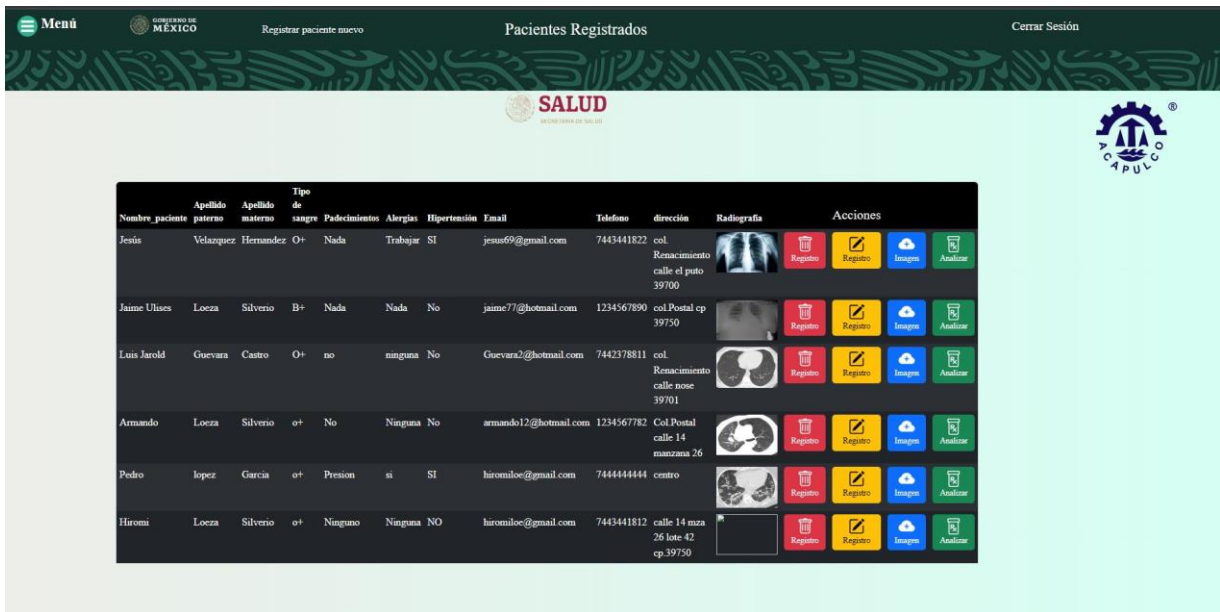


Ilustración 55. Visualización del paciente registrado al sistema. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

En la siguiente ilustración se muestra la interfaz subir imagen en cuál al darle en botón de subir se abrirá el explorador de archivo de la computadora y buscará la imagen que deseas subir al sistema. Por otra parte, al desarrollar el programa hicimos que este apartado pudiera seleccionar cualquier tipo de imagen ya sea png, jpg, jpge. Por último, muestra el nombre de la imagen. (Véase en la ilustración 49)



Ilustración 56. Subir la imagen del tórax del paciente. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

En esta ilustración se muestra la interfaz de la imagen subida de manera exitosa para su posterior diagnóstico (Véase en la ilustración 50).



Ilustración 57. Visualización de la imagen subida exitosamente al sistema. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

En esta ilustración se puede observar la interfaz de “Dictamen” en cual su función o acción a realizar es que el administrador como el usuario puedan visualizar los

resultados de todos los pacientes así mismo generar un reporte que se muestra en ilustración 42. Por otra parte, hay una opción de filtro para que se filtre por casos positivos o negativos. (Véase en la ilustración 51)

ID	Nombre Completo del paciente	Fecha	Resultado	Eliminar	Ver dictamen
11	Jesús Velazquez Hernandez	2023-05-24	Resultado: Positivo a COVID-19	Eliminar Dictamen	Ver dictamen
13	Jaime Ulises Loeza Silverio	2023-05-29	Resultado: Negativo a COVID-19	Eliminar Dictamen	Ver dictamen
16	Luis Jarold Guevara Castro	2023-05-29	Resultado: Negativo a COVID-19	Eliminar Dictamen	Ver dictamen
17	Hiroimi Loeza Silverio	2023-06-01	Resultado: Positivo a COVID-19	Eliminar Dictamen	Ver dictamen
18	Hiroimi Loeza Silverio	2023-06-06	Resultado: Positivo a COVID-19	Eliminar Dictamen	Ver dictamen

Ilustración 58. Interfaz dictamen. Fuente, adquirida del analizador de imagen COVID-19 (2023).

En esta ilustración se observa el dictamen individual de cada paciente, como se puede observar tiene la información de este, así mismo los datos del usuario de quien haya realizado el diagnóstico. Por último, un apartado de resultados en cual arroja el resultado ya sea positivo o negativo así mismo una opción para que el medico escriba un comentario o alguna observación y la fecha que se realizó este. (Véase en la ilustración 52)

The image shows a web-based form titled "Dictamen de COVID-19 Paciente". It is divided into three main sections: "Paciente", "Información del médico", and "Resultados".

Paciente: This section contains input fields for "Nombre:" (Hiromi), "Apellido paterno:" (Loeza), "Apellido materno:" (Silverio), "Tipo de sangre:" (o+), "Padecimientos:" (Ninguno), "Alergias:" (Ninguna), and "Hipertensión:" (NO).

Información del médico: This section contains input fields for "Nombre:" (Juan Jose Bedolla Solano) and "Especialidad:" (Neomonología).

Resultados: This section includes the text "Diagnostico utilizando IA:" followed by a red-bordered box containing the text "Resultado: Positivo a COVID-19". Below this is a "Dictamen:" section with a text area containing the following text: "Mediante el uso de modelo de color se comprobó que el paciente presenta vidrio en los pulmones arrojando un 90% de probabilidad, con la IA también arrojó positivo por lo que se puede determinar que el paciente es positivo a 'COVID-19'". At the bottom of this section is a "Fecha:" field with the value "06/06/2023" and a calendar icon. A "Guardar dictamen" button is located at the bottom right of the form.

Ilustración 59. Dictamen del paciente. Fuente, adquirida del analizador de imagen COVID-19 (2023).

En esta ilustración se puede observar el filtro del dictamen de solos los casos positivos así también la opción de eliminar dicho dictamen. (Véase en la ilustración 53).

ID	Nombre Completo del paciente	Fecha	Resultado	Eliminar	Ver dictamen
11	Jesús Velazquez Hernandez	2023-05-24	Resultado: Positivo a COVID-19		

Ilustración 60. Filtro de dictamen (Resultado positivos). Fuente, adquirida del analizador de imagen COVID-19 (2023).

En esta ilustración se puede observar el filtro del dictamen de solos los casos negativos así también la opción de eliminar dicho dictamen. (Véase en la ilustración 54).

ID	Nombre Completo del paciente	Fecha	Resultado	Eliminar	Ver dictamen
13	Jaime Ulises Loeza Silverio	2023-05-29	Resultado: Negativo a COVID-19		
16	Luis Jarold Guevara Castro	2023-05-29	Resultado: Negativo a COVID-19		

Ilustración 61. Filtro de dictamen (Resultado negativos). Fuente, adquirida del analizador de imagen COVID-19 (2023).

En la siguiente ilustración se puede observar en acción la inteligencia artificial específicamente las redes neuronales convulsionales que entrenamos con el *dataset* de 1000 mil imágenes con *COVID-19* y no *COVID-19*, gracias a esta red neuronal es programa es capaz de diferenciar o encontrar características similares. Por lo cual esto no podría decir que 100% seguro el diagnostico que nos arrojara la aplicación web. (Véase en la ilustración 55).

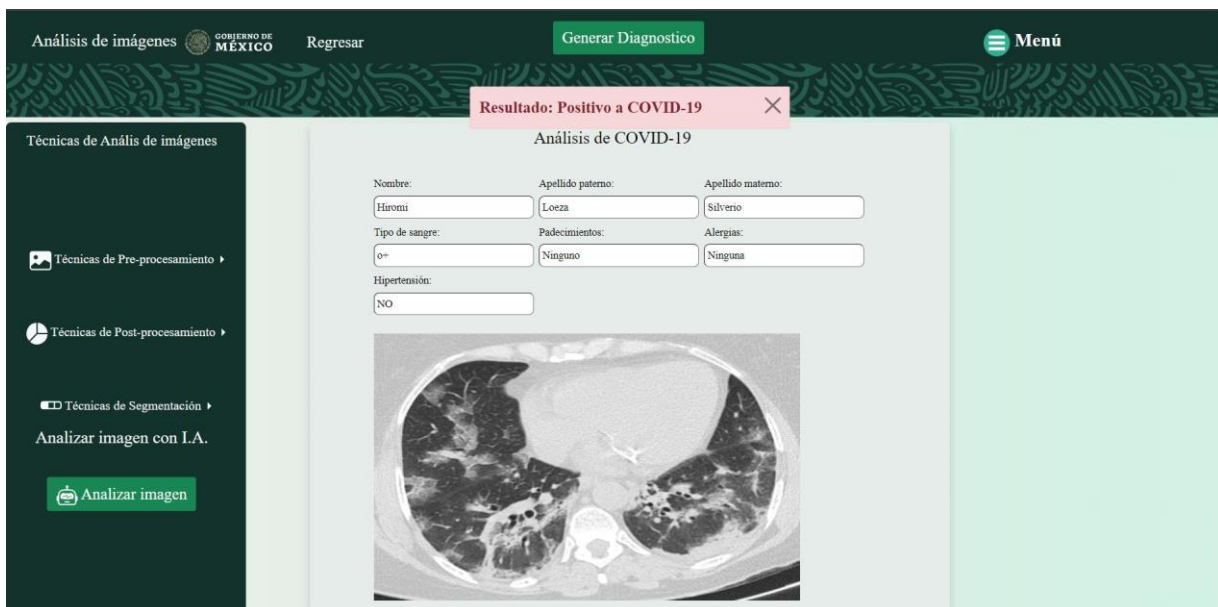


Ilustración 62. Resultado del diagnóstico a través de la red neuronal convolucional. Fuente, adquirida del analizador de imagen COVID-19 (2023).

En la siguiente ilustración se puede observar la realización del diagnóstico a través de las técnicas procesamiento de imágenes (Preprocesamiento, Post-procesamiento y segmentación) en este caso fue con la del preprocesamiento usando modelo de color. Por otra parte, no solo se puede aplicar una técnica ya como se puede observar

tenemos contenedor con cada una de las técnicas y el usuario puede estar probando cada una de ellas. (Véase en la ilustración 56)

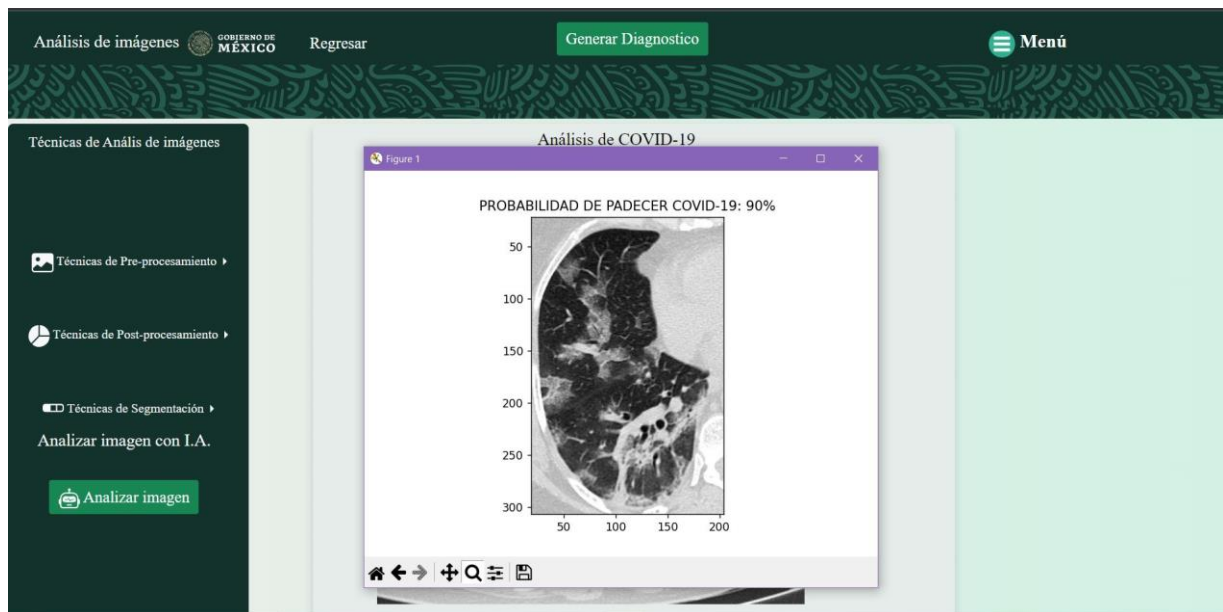


Ilustración 63. Resultado del diagnóstico a través del procesamiento de imágenes. Fuente, adquirida del analizador de imagen COVID-19 (2023).

En la siguiente ilustración se muestra el reporte general de los dictámenes realizados aquí el usuario como el administrador podrán observar todos los dictámenes realizados ya sea con un diagnóstico de resultado positivo o negativo. Por otra parte, el usuario tendrá una idea de la cantidad de paciente que se haya realizado el análisis y que posteriormente se puedan usar para estadísticas propias o para futuras investigaciones o también para tener una excelente organización de los pacientes. (Véase en la ilustración 57)

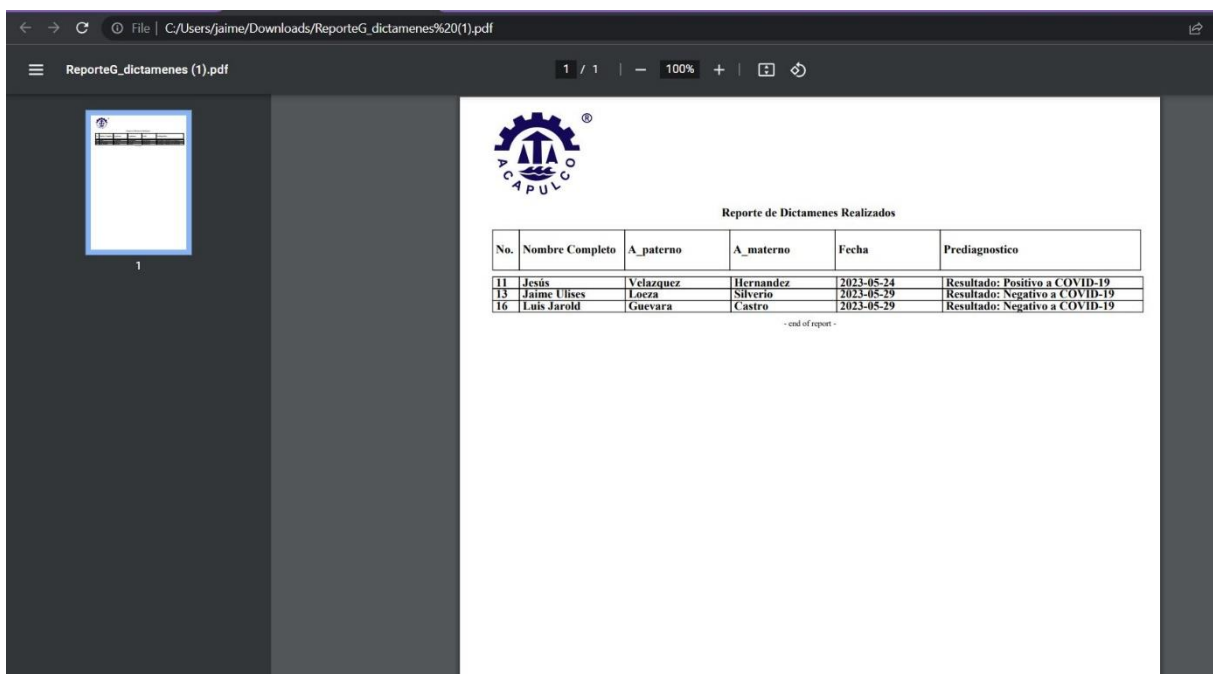


Ilustración 64. Reporte general. Fuente, adquirida del analizador de imagen COVID-19 (2023).

En la siguiente ilustración se muestra el reporte de los pacientes a COVID-19 con un filtro aquí el usuario como el administrador podrán observar todos los dictámenes realizados ya sea con un diagnóstico de resultado negativo. Por otra parte, el usuario tendrá una idea de la cantidad de paciente que se haya realizado el análisis y que posteriormente se puedan usar para estadísticas propias o para futuras investigaciones o también para tener una excelente organización de los pacientes. (Véase en la ilustración 58)

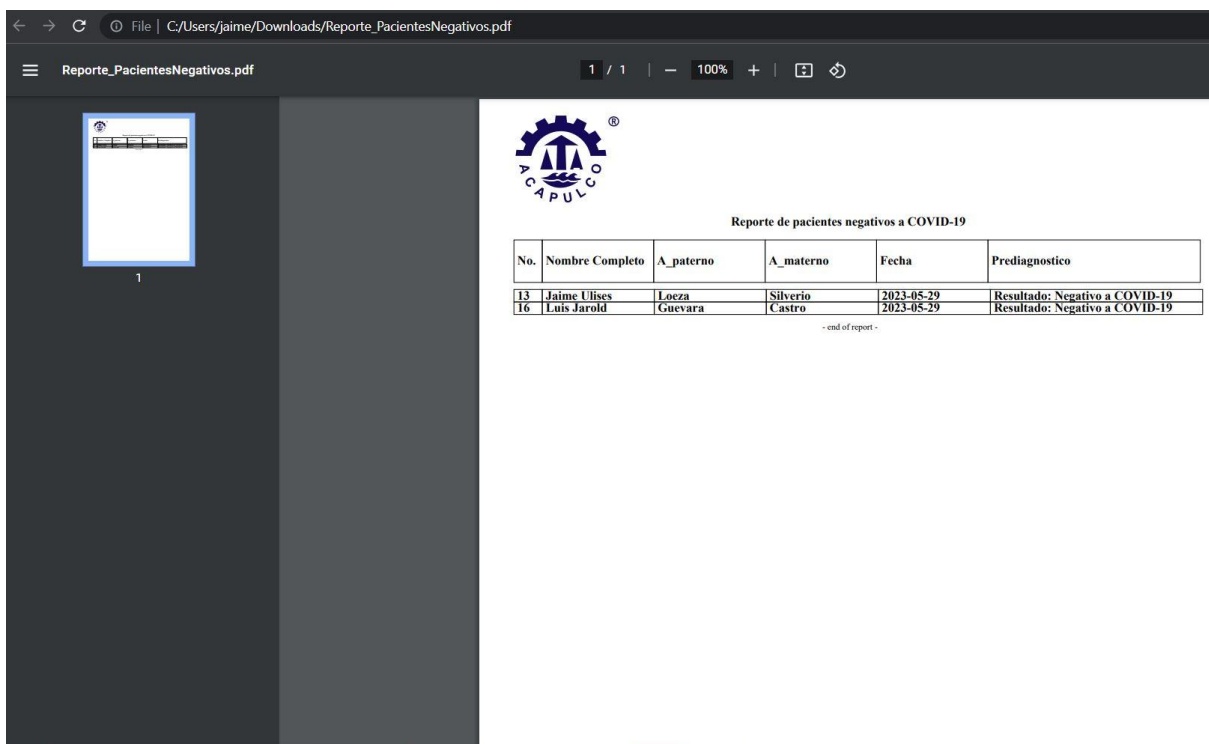


Ilustración 65. Reporte genera de negativo a COVID-19I. Fuente, adquirida del analizador de imagen COVID-19 (2023).

5.2 Pruebas de datos e integridad de base de datos.

En la siguiente tabla se puede observar que el sistema es capaz de denegar el acceso al sistema ya que se tiene como entrada los datos del administrador donde se escribe a propósito de manera incorrecta la contraseña, para así ver sí el sistema puede denegar el acceso por este hecho además de mostrar un aviso en pantalla de este hecho.

Inicio de sesión	
Objetivo:	Comprobar que solo los usuarios registrados puedan iniciar sesión.
Técnicas:	Ingresar al sistema con un nombre y una contraseña incorrecta.

Módulo:	Login
Entrada:	“Bedolla7” “1229”
Salida:	“Contraseña incorrecta”
Resultados	No permite el acceso al sistema en caso de no contar con los datos correctos de comprobación en la base de datos.
Veredicto	Aprobado.

Tabla 5.1 Prueba de inicio de sesión al sistema.

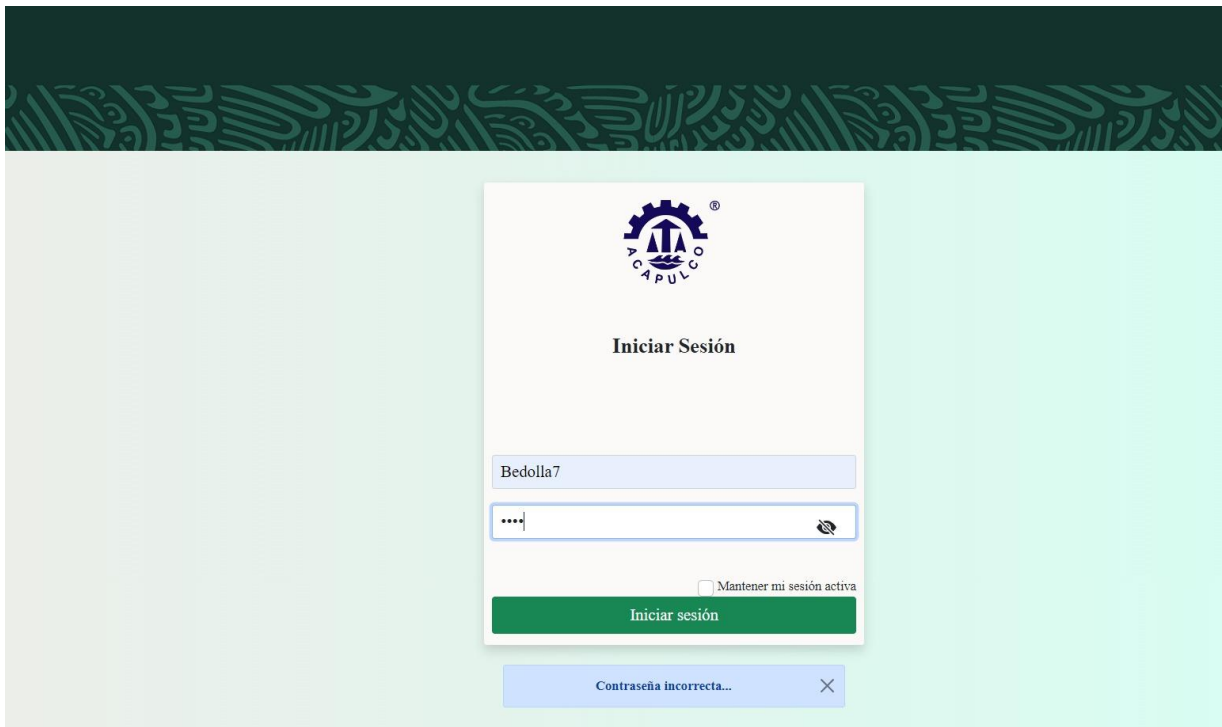


Ilustración 66. Respuesta del sistema a una contraseña incorrecta. Fuente: Propia(2023).

5.2 Resultados de evaluación aplicando la Norma ISO

ISO/IEC 25010

Con base en las características principales del sistema, de esta norma se deriva la 9126 y de manera selectiva toma de las 8 características de calidad las 6 más importantes y sus respectivas métricas.

ISO / IEC 9126

Tomando en cuenta la puntuación de 0 a 10, conseguimos el total multiplicando el peso por esta calificación, comparándolo con el total ideal que debería tener (Tabla 11).

Tabla 5.2 Resultado de la evaluación según métricas de ISO / IEC 9126

Atributo	Subatributo	Peso	Calificación	Total	Total ideal	Nivel
Funcionalidad						
	Idoneidad	2	8	16	20	A
	Exactitud	2	8	16	20	A
	Interoperabilidad	2	0	0	20	B
	Seguridad	2	8	16	20	A
	Conformidad	2	0	0	20	B
Confiabilidad						
	Madurez	2	9	18	20	A

Recuperación	2	7	14	20	M
Operatividad	2	8	16	20	A
Usabilidad					
Aprendizaje	2	9	18	20	A
Compresión	2	10	20	20	A
Operatividad	2	9	18	20	A
Eficiencia					
Comportamiento en el tiempo	2	7	14	20	M
Comportamiento de recursos	2	8	16	20	A
Mantenibilidad					
Estabilidad	2	8	16	20	A
Facilidad de análisis	2	7	14	20	M
Facilidad de cambio	2	9	18	20	A
Facilidad de pruebas	2	9	18	20	A
Portabilidad					
Capacidad de instalación	2	9	18	20	A
Capacidad de reemplazamiento	2	7	14	20	M

Adaptabilidad	2	9	18	20	A
Coexistencia	2	9	18	20	A
			316	400	
			79%	100%	

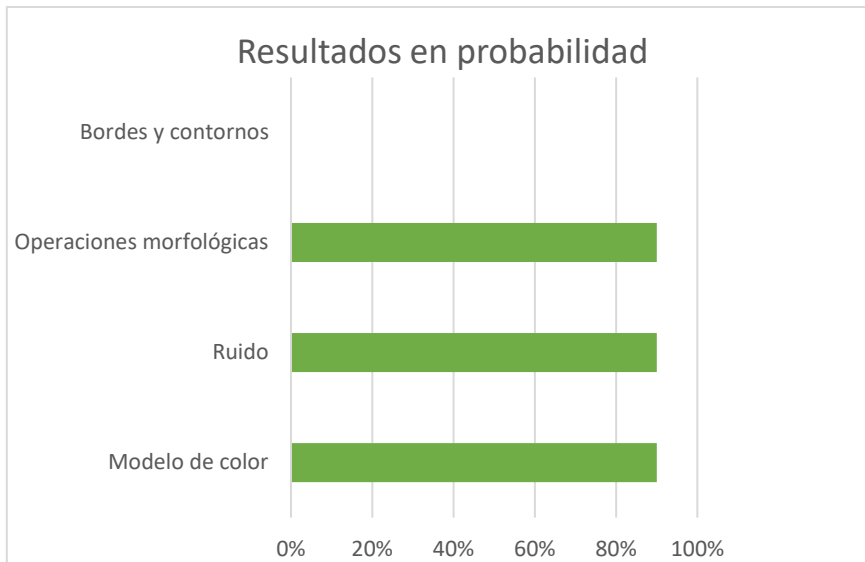
Fuente: Elaboración propia (2023).

Encontramos que el sistema tiene una calificación de 79% sobre 100%, por lo que podemos concluir que el sistema, si bien funciona y cumple con lo que debe hacer, aún requiere de muchas correcciones y mejoras, tanto en la parte de *front-end* como *backend*.

5.3 Resultados de diferentes técnicas de análisis de imágenes.

Se realizaron algunos análisis a una imagen con el fin de poder comprobar el comportamiento en los resultados según cada distinta técnica de análisis de imágenes, en este caso se utilizaron: Modelo de color, Ruido, Operaciones morfológicas además de bordes y contornos, cabe señalar que todos los resultados de técnicas de procesamiento de imágenes se pusieron en comparación con el dado por la Inteligencia Artificial empleada. Se utilizó una imagen con un diagnóstico de COVID-19 previamente diagnosticada como positivo para así comprobar la efectividad del sistema a la hora de realizar un dictamen. Se obtuvieron valores del 90% en casi todas ya que bordes y contornos arrojó un valor erróneo del 0% puesto que la detección de discontinuidad en la intensidad de la imagen, detectando las fronteras de objetos sin embargo no hace una diferenciación entre los objetos que componen normalmente el pulmón y el vidrio despolido, por otro lado la IA arrojó un

valor positivo, por lo que se puede concluir que el sistema arroja valores validos en cuanto a la detección de COVID-19.



Grafica 1. Resultados en porcentajes de probabilidad de padecer COVID-19 arrojados por las técnicas de procesamiento de imágenes probadas. Fuente: Propia.

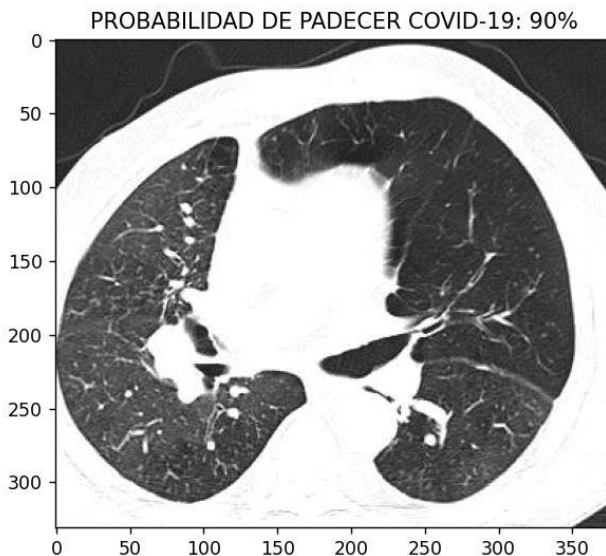


Ilustración 67. Resultado en probabilidad por medio de la Técnica, bordes y contornos. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

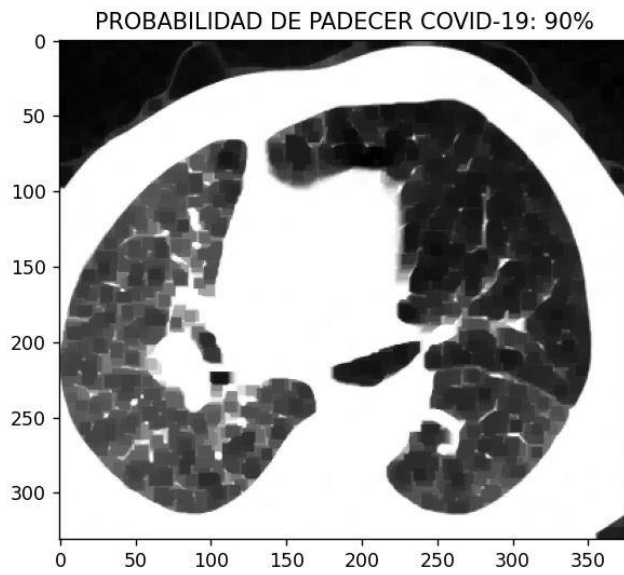


Ilustración 68. Resultado en probabilidad por medio de la Técnica, operaciones morfológicas. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

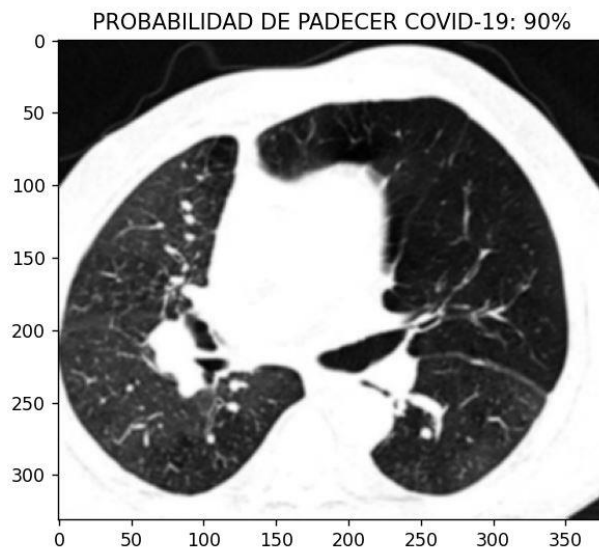


Ilustración 69. Resultado en probabilidad por medio de la técnica de Ruido. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

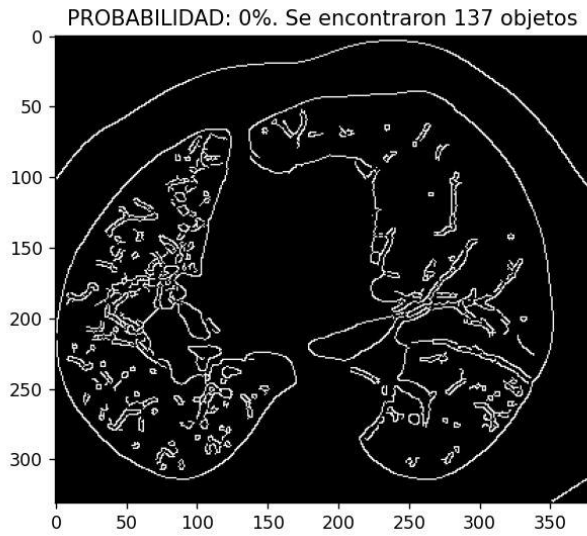


Ilustración 70. Resultado en probabilidad por medio de la técnica de Bordes y contornos. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

Resultado: Positivo a COVID-19 ✕

Análisis de COVID-19

Nombre:	Apellido paterno:	Apellido materno:
<input type="text" value="."/>	<input type="text" value="."/>	<input type="text" value="."/>
Tipo de sangre:	Padecimientos:	Alergias:
<input type="text" value="o+"/>	<input type="text" value="No"/>	<input type="text" value="Ninguna"/>
Hipertensión:		
<input type="text" value="No"/>		

Ilustración 71. Resultado arrojado por la inteligencia artificial implementada. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

5.4 Prueba de rendimiento

Esta prueba se realizó con el fin de comprobar el tiempo de respuesta del sistema en lo que compete a la función de análisis de la imagen del paciente, donde se aplicó la técnica de modelo de color con el fin de medir el tiempo de respuesta de esta función porque prácticamente es el fundamento principal para poder apoyar al médico a la hora de realizar un dictamen, utilizando las herramientas de desarrollador de Google Chrome se pudo comprobar que la respuesta del sistema fue de 3.09 segundos, como se puede observar en la siguiente imagen, siendo un tiempo de respuesta aprobado.

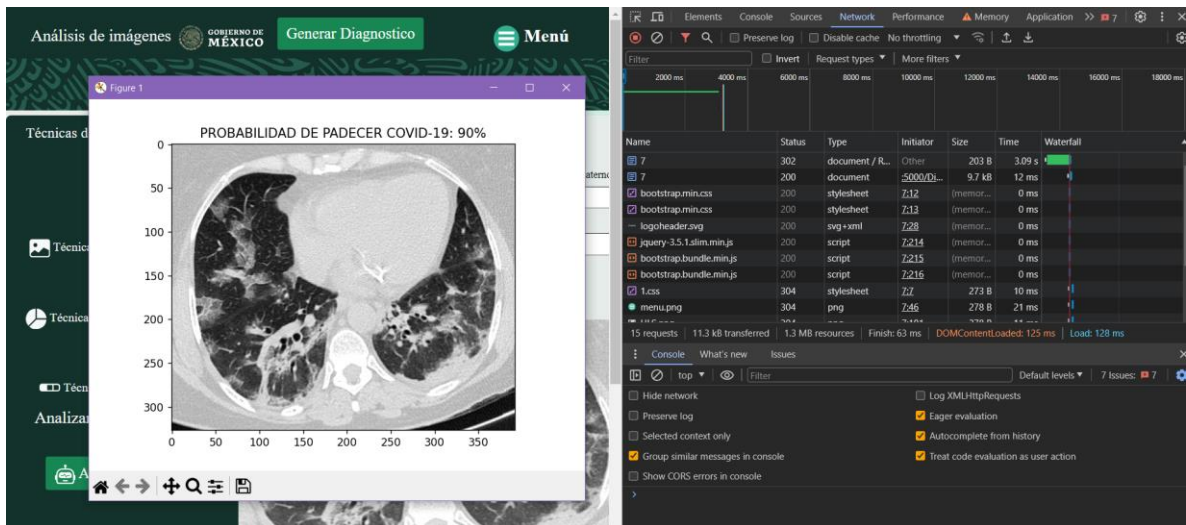


Ilustración 72. Prueba de rendimiento utilizando las herramientas para desarrollador de Google Chrome. Fuente: Propia, adquirida del analizador de imagen COVID-19 (2023).

5.4 Cumplimiento de los objetivos

Al realizar este proyecto se realizó una serie de investigaciones, para que se tuviera una idea de cómo empezar la detección de este virus, por lo cual se entendió de manera correcta como el virus ataca los pulmones.

Creamos un sitio web intuitivo para el usuario, se logró a través de investigaciones, para saber lo que el usuario desea en el uso de una aplicación médica, por lo cual se optó por una interfaz con colores que se presenta en las páginas oficiales de la secretaria de salud de México, con toques propios.

Gracias al analizador de la aplicación *web* se pudo examinar cada fotografía o imagen digital para saber si esta afecta con el virus o no.

Familiarizamos con las diversas técnicas de preprocesamiento de imágenes, al investigar cada una de ellas y se utilizaron las más a acorde a lo que el sistema requería.

Por otra parte, se espera reducir gasto en los estudios de la detección del virus COVID-19, con la implementación de este sistema a los diferentes centros de salud del estado o del país.

Se logró hacer un uso efectivo o adecuado en las imágenes proporcionadas con el resultado de un diagnóstico, ayudando al especialista al realizar un diagnóstico más certero, ya que gracias a la implementación de la red neuronal entrenada con un 92.3 % de exactitud que hace que los resultados arrojados por el sistema sean más precisos.

Se logró obtener un impacto en sector salud ya que con la aplicación *web* se detecta a tiempo o de manera prematura la persona infectada ya sea por las técnicas de preprocesamiento o por la inteligencia artificial.

Para finalizar se logró el desarrollo del software que permite la detección del virus *COVID-19* y que este sea amigable para quien lo use.

Discusiones

Esta investigación tuvo como propósito identificar y analizar las imágenes del tórax de una persona para la detección oportuna para su tratamiento oportuno, además de poner a prueba el uso de herramientas tecnológicas para la detección de esta. De los resultados obtenidos en esta investigación, se puede deducir que el uso o la implementación de herramientas tecnológicas en otras áreas que no solo sean áreas de ingeniería se puede obtener grandes beneficios, en este caso el preprocesamiento de imágenes o el post-procesamiento es una gran alternativa para la detección de *COVID-19* no solo se limita a eso, sino que se puede extender a otro tipo de enfermedades.

El virus COVID-19, que inicialmente fue categorizado como una epidemia, pero rápidamente se convirtió en una pandemia mundial debido a su facilidad de transmisión y severos problemas en el sistema respiratorio, ha causado una catástrofe médica entre los años 2020 hasta el año 2022. Como resultado, varios sistemas de salud colapsaron debido a la gran cantidad de pacientes que ingresaban con condiciones críticas de salud.

El uso de tecnologías en área de salud es una gran alternativa para futuros proyectos como el preprocesamiento de imágenes para la detección de covid19 puede ser un parte agua para futuros proyecto, ya que como lo menciona Fuentes y Vera (2021), para poder realizar una detección de COVID-19 efectiva y accesible, se plantea hacer uso de la Inteligencia Artificial, aplicando Deep Learning, a través del análisis de imágenes de rayos x, específicamente del tórax. Por otro lado, se puede aplicar en diversas áreas, desde identificación de placas hasta reconocimiento facial en

aeropuertos, así mismo que el procesamiento de imágenes está lejos de ser un tema terminado; más bien, está más apropiadamente situado en el medio de los dos, ya que el lenguaje de programación, la plataforma de desarrollo y el proceso mismo proporcionan los recursos necesarios; sin embargo, cómo se utilizan esos recursos depende del desarrollador.

La elección de Python como principal lenguaje de programación fue un acierto ya que es de los más usados para análisis de imagen y uso de inteligencia artificial, como menciona Alfonso, B. (2020) , la elección de *Python* en lugar de otras opciones de programación se debe a que es considerado uno de los lenguajes más apropiados para el procesamiento de imágenes y la implementación de técnicas de aprendizaje automático. Esto se debe a que cuenta con iteraciones rápidas permitiendo concentrar los datos y desarrollar algoritmos. Se puede decir que el uso de Python otorga una infinidad de herramientas como son matplotlib, tensor Flow entre otras cada una con funciones específicas, por ello se definió que el framework de trabajo sería Flask ya que este está hecho en *Python* y tiene una gran adaptabilidad al mismo. Otro trabajo encontrado fue “Detección de anomalías de COVID-19 en radiografías de tórax” prácticamente aquí se entrena una Red neuronal mediante el Resnet50 una función de la librería Keras de Python, en comparación a nuestro trabajo resulta que el análisis es más extenso puesto que su sistema depende totalmente de la Red neuronal en contra parte a nuestro sistema que implementa la librería matplotlib con el fin de poder graficar las imágenes y las operaciones correspondientes en escala de grises, siendo la I.A. como una segunda opción solo para poder estar más seguros, ya que esta fue adaptada a nuestras necesidades siendo hecha utilizando DenseNet otra arquitectura de Keras.

Conclusión

Tras el desarrollo de este proyecto que lleva por nombre “Sistema de análisis de imágenes con *Python* para la detección de casos *COVID-19* a través de imágenes radiológicas de los pacientes”, se puede reafirmar la importancia que hoy en día tiene la ingeniería en sistemas computacionales para el desarrollo de *software*, permitiendo la abstracción de la lógica de los programadores y también involucra de manera indirecta al usuario con lo cual se puede obtener un producto más cercano a las necesidades de éste y en consecuencia maximizando los beneficios para él. Por otra parte, se reafirma que la implementación de *software* no solo en el ámbito de la medicina sino en todos los ámbitos es indispensable, para mejorar la producción y el rendimiento de los procesos, en nuestro caso aumentar la producción y la calidad de los resultados al diagnosticar *COVID-19*, es por eso que el reciente éxito de utilizar técnicas de procesamiento de imágenes digitales, tanto para imágenes fijas como para imágenes en movimiento, es evidente y cada vez más se ven este tipo de tecnologías involucradas en los procesos y negocios.

Por ejemplo, aplicaciones como la captura e identificación de matrículas de vehículos que limitan la velocidad permitida en las calles de la ciudad, o la tecnología de reconocimiento facial utilizada en los aeropuertos para autorizar la entrada a edificios o países, son ampliamente utilizadas en muchas partes del mundo. Por no hablar de los usos industriales de la visión artificial, entre otros.

Sin embargo, el desarrollo de aplicaciones y el procesamiento digital de imágenes con *Python* está lejos de ser un tema cerrado, como se puede ver en las experiencias obtenidas durante este proceso. En cambio, está más cerca de la intersección de un esfuerzo técnico y artístico porque el lenguaje de programación, la plataforma de desarrollo y el proceso en sí proporcionan los recursos necesarios para su uso. Se pudo evaluar la importancia de cada paso del trabajo realizado, incluyendo la adquisición de las imágenes y sus características, el preprocesamiento para mejorar la imagen eliminando características indeseables y resaltando aquellas que son útiles durante el proceso, el preprocesamiento para cambiar la imagen y darle las características necesarias para las etapas finales, y el post procesamiento para verificar errores y posiblemente resaltar detalles importantes. Es indispensable seguir investigando sobre las técnicas de preprocesamiento de imágenes y en que más campo se puede aplicar, así como explorar más a fondo las técnicas de procesamiento de imágenes para que con el tiempo progresen sus capacidades y sea aún más sencilla la detección de anomalías no sólo hablando radiografías del tórax, para aumentar la precisión y ajustarla a la capacidad de cámaras y detectores cada día más avanzados.

Por otra parte, la enfermedad *SARS-CoV-2*, también conocida como *COVID-19*, es una infección viral que ha paralizado los sistemas de salud del mundo. Actualmente, hay muchos millones de casos en todo el mundo, y aunque su letalidad es menor que la de otros *coronavirus*, su efecto es y seguirá siendo importante.

Es importante tener en cuenta que los síntomas clínicos que inicialmente se limitaban a los sistemas respiratorios se han ido expandiendo significativamente a otras áreas

de la economía del paciente. La disponibilidad de pruebas diagnósticas, sus costos y accesibilidad también han sido desafíos para los sistemas de salud. Dado que las manifestaciones en la tomografía computarizada han sido muy razonables, pero no especialmente específicas, la radiología ha tomado protagonismo. Sin embargo, existe una discrepancia entre la clínica, los resultados de laboratorio (*rt-PCR*) y el TC. Además, los avances tecnológicos en el procesamiento de imágenes y la inteligencia artificial pueden ayudar significativamente en la detección rápida y precisa de quienes tienen esta afección, evitando la propagación y manteniendo segura a la población. Por esta razón, se desarrolló este proyecto para brindar a las personas acceso directo a nuestra página *web*, lo que ayudaría a un diagnóstico rápido y ahorraría dinero y tiempo, brindando una nueva opción para realizar una prueba de *COVID-19*.

Para finalizar en transcurso del desarrollo de la aplicación pusimos en práctica todos los conocimientos adquiridos en estos cinco años en la universidad no solo en la programación sino en la demás materia como taller de investigación 1 y 2 por poner un ejemplo, además que también adquirimos nuevos conocimientos que lo llevaremos en nuestros futuros proyectos.

Recomendaciones

- Se recomienda implementar un *dataset*, esto permitirá poder obtener mejores

resultados respecto al entrenamiento de una red neuronal convolucional, dado que mientras mayor sea la cantidad de datos, mejor podrá la red discernir entre las clases definidas.

- Para acortar el tiempo de entrenamiento de la red de sinapsis, así como para reducir la carga computacional en la *PC* o computadora portátil que se utiliza, se recomienda utilizar una *GPU* (unidad de procesamiento de gráficos) o una *TPU* (unidad de procesamiento de tensores).
- Para los próximos proyectos espero que involucren el desarrollo y entrenamiento de redes acopladas neuronalmente, se recomienda realizar un entrenamiento con una gran cantidad de imágenes porque esto permitirá una predicción más precisa.
- Para proyectos de inteligencia artificial se recomienda *Google Colab*, aunque no se disponga de *hardware* de alta gama porque ofrece un alto rendimiento en el procesamiento de datos y recursos gratuitos como el acceso a la *GPU* y *TPU* en la nube, haciendo los procesos mucho más rápidos y sin necesidad para configuraciones previas o instalaciones de *software* porque *Google Colab* proporciona todas las herramientas necesarias.
- Por otro lado, también se recomienda que no solo se trabaje con dos clases de imágenes de rayos x, “COVID” y “normal”, ya que existen otras enfermedades que

afectan a los pulmones como son la neumonía, opacidad pulmonar y neumonía viral, permitiendo tener un diagnóstico más preciso del estado del paciente.

Glosario

Bootstrap: Es un *framework* originalmente creado por *Twitter*, que permite crear interfaces *web* con *CSS* y *JavaScript*, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice.

CSS: Es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en *HTML*.

Framework: Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

Get: Es obtener información del servidor. Traer datos que están en el servidor, ya sea en un archivo o base de datos, al cliente.

HTML: Es un estándar que sirve de referencia del *software* que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código *HTML*) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

Interconexión: Es una técnica que responde a la necesidad de hacer interactuar las distintas infraestructuras (redes) con tecnologías y diseños diferentes, con la finalidad que los usuarios conectados perciban el servicio como si se tratara de una sola red.

Interfaz: Conjunto de elementos de la pantalla que permiten al usuario realizar acciones sobre el Sitio Web que está visitando.

JavaScript: Es un lenguaje de programación que se puede utilizar para construir sitios *Web* y para hacerlos más interactivos.

Librería: Se usa para referirse a un programa que contiene varias funciones para lograr un propósito bien definido y específico.

Link: Correspondiente al enlace del hipertexto a otro documento o recurso.

MySql: Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por *Oracle Corporation* y está considerada como la base datos *open source* más popular del mundo, y una de las más populares en general junto a *Oracle* y *Microsoft SQL Server*, sobre todo para entornos de desarrollo *web*.

Plataforma: Es un sistema que sirve como base para hacer funcionar determinados módulos de *hardware* o de *software* con los que es compatible.

Post: Es enviar información desde el cliente para que sea procesada y actualice o agregue información en el servidor, como sería la carga o actualización en sí de una noticia.

Servidor: Es un ordenador o máquina informática que está al “servicio” de otras máquinas, ordenadores o personas llamadas clientes y que les suministran a estos, todo tipo de información.

URL: Es una secuencia de caracteres que se utiliza para nombrar y localizar recursos, documentos e imágenes en *Internet*.

Usuario: Es un individuo que utiliza una computadora, sistema operativo, servicio o cualquier sistema informático.

Web: Conjunto de información que se encuentra en una dirección determinada de *internet*.

Referencias bibliográficas

Gamero, A. M., & Chamorro, M. R. (2021). La inteligencia artificial en el control de la COVID-19. *Atención Primaria*, 53(10). DOI: 10.1016/j.aprim.2021.102099 .
Procesamiento de imágenes digitales. (s.f.). Alojamiento.us.es. Extraído el 7 de enero de 2023, de <http://alojamiento.us.es/gtocoma/pid/introduccion.html>

JOYANES, L. (2008). *Fundamentos de la programación. Algoritmos y Estructura de Datos*, 4ª Edición. Madrid: McGraw-Hill.

Lands G. (2021) El papel de la inteligencia artificial y la imagenología médica en la lucha contra las enfermedades zoonóticas.. *iaea.org*. Extraído de: <https://www.iaea.org/es/bulletin/inteligencia-artificialimagenologia-medica-lucha-zoonosis>

Secretaría de Salud (2022). Datos Abiertos Dirección General de Epidemiología. Extraído el 7 de enero de 2023, de <https://www.gob.mx/salud/documentos/datos-abiertos152127>

Pérez MR, Gómez Tejeda JJ, Dieguez RA. (2020). Características clínico-epidemiológicas de la COVID-19. *Revista habanera de ciencias médicas*. Recuperado de: <http://www.revhabanera.sld.cu/index.php/rhab/article/view/3254/2505>

Meza Calvache, J. M., Estrada Rodríguez, A. D., Chabusa Martínez, C. B., & Velasco Paucar, V. A. (2020). Utilidad de Pruebas de cadena de polimerasa, pruebas rápidas

y Tomografías en pacientes con COVID-19. *Journal of American Health*, 3(2), 32–39.

<https://doi.org/10.37958/jah.v3i2.28>

Márquez, M. A. (1996). Enseñanza de las ciencias de la tierra: Revista de la Asociación Española para la Enseñanza de las Ciencias de la Tierra,. *Dialnet*. Recuperado 19 de octubre del 2022. C., de <https://dialnet.unirioja.es/servlet/articulo?codigo=234582>

Coca, Ledys, Franco, Zulay, & Pateti, Antonio. (2008). Implementación de filtros morfológicos utilizados en el procesamiento de imágenes digitales en un dispositivo lógico programable. *Universidad, Ciencia y Tecnología*, 12(48), 171-182. Recuperado en 11 de agosto de 2022, de http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S131648212008000300008&lng=es&tlng=es.

Magro, R. (2013, 31 diciembre). BINARIZACIÓN DE IMÁGENES DIGITALES Y SU ALGORITMIA COMO HERRAMIENTA APLICADA A LA ILUSTRACIÓN ENTOMOLÓGICA. *Boletín de la Sociedad Entomológica Aragonesa (S.E.A.)*, 53, 443-464. <http://seaentomologia.org/PDF/Boletin53/443464BSEA53BinarizacionRMagro.pdf>

Alaa, N.; y El Abidne, I.Z. (2021). Introduction to Image Processing with Python. Cadi Ayyad University. DOI: 10.13140/RG.2.2.18870.37444

Lutz, M. (2009). A Python Q&A Session. En Steele, S. (Ed.), *Learning Python* (4ta ed., pp. 4-5). O'Reilly Media.

Gonzalo Pajares, Jesús M. de la Cruz. Visión por Computador. Imágenes digitales y aplicaciones. RaMa, 2001.

La Serna Palomino, N., y Román Concha, U. N. (2009). Técnicas de Segmentación en Procesamiento Digital de Imágenes. *Revista De investigación De Sistemas E Informática*, 6(2), 9–16. Disponible en: <https://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/3299>

Juganaru Mathieu, M. (2014). Del algoritmo al programa. En Enrique Callejas, J. (Ed.), *Introducción a la programación* (5-6). Grupo Editorial. Almeida, J.,Moreira,L.(2022). ANÁLISIS DE IMÁGENES DE RAYOS-X DE COVID-19 USANDO APRENDIZAJE PROFUNDO (2-4). Tesis. Recuperado a partir de <http://repositorio.ug.edu.ec/handle/redug/60030>

Morocho, K., Publig,D.(2021). ANÁLISIS DE IMÁGENES DE RAYOS X DE COVID-19 A TRAVÉS DE REDES NEURONALES ARTIFICIALES (pp.15). Tesis. Universidad de Guayaquil. <http://repositorio.ug.edu.ec/handle/redug/52613>

Patiño Pérez, D., Silva Bustillos, R., Botto-Tobar, M., & Munive Mora, C. (2021). Análisis de Imágenes de Rayos X por Medio de Redes Neuronales Artificiales. *Ecuadorian Science Journal*. 5(1), 55-60. DOI: <https://doi.org/10.46480/esj.5.1.50>

Jain, R., Gupta, M., Taneja, S. Hemanth, J.(2020).DETECCIÓN Y ANÁLISIS BASADOS EN APRENDIZAJE PROFUNDO DE COVID-19 EN IMÁGENES DE RAYOS X DE TÓRAX (51)(1690–1700). Departamento de CSE, Facultad de Ingeniería de Bharati Vidyapeeth, Delhi, India. <https://doi.org/10.1007/s10489-020-01902-1>

Yang, D., Martínez, C., Visuña, L., Khandar, H. Bhatt, C., Carretero, J.(2021). DETECCIÓN Y ANÁLISIS DE COVID-19 EN IMÁGENES MÉDICAS MEDIANTE TÉCNICAS DE APRENDIZAJE PROFUNDO. SCIENTIFIC REPORTS 11. Article number: 19638. <https://www.nature.com/articles/s41598-021-99015-3.pdf>

Gomez,M.(2022). DETECCIÓN DE COVID-19 Y NEUMONÍA EN IMÁGENES RAYOS X DE PULMONES UTILIZANDO REDES NEURONALES CONVOLUCIONALES(p.53).Universidad autónoma de México. http://lapi.fi-p.unam.mx/wpcontent/uploads/Tesis_Mauricio_G_mez310051730ultimaversion.pdf

Ramos González, M. I., Carreto Binaghi, L. E., & Salcedo Chávez, M. (2020). Métodos diagnósticos. Revista Latinoamericana de Infectología Pediátrica, 33(s1), 33–41. DOI: <https://doi.org/10.35366/96669>

Lozano Z. H., Chávez A., Ángel C., Álvarez V. I., Ochoa X., Rodríguez R. & Escamilla P. (2020). Diagnóstico clínico-radiológico de neumonía por COVID-19 (SARS-CoV-2) en el Hospital Ángeles Lomas. Acta médica Grupo Ángeles, 18(2), 224-228. DOI: <https://doi.org/10.35366/93908>

Prado A, F., Oviedo C, V., Valdebenito P, C., Espinoza G, A., & Renedo de la Hoz, S. (2020). INTERPRETACIÓN DE PATRONES RADIOLÓGICOS SUGERENTES DE COVID-19 EN PEDIATRÍA. Neumología Pediátrica, 15(2), 330–338. DOI: <https://doi.org/10.51451/np.v15i2.66>

Murillo, (2008). Enfoque, situación y desafíos de la investigación sobre eficacia escolar en América Latina y el Caribe. En UNESCO, Eficacia escolar y factores asociados en América Latina y el Caribe. Santiago de Chile: UNESCO

Zabala, X. y Ochoa, C. (2010). Estudio de Frameworks para PHP e integración a una herramienta IDE. Tesis de Ingeniería de Sistemas. Escuela Superior Politécnica de Chimborazo. Ecuador.

Cui, W., Huang, L., Liang, L., y Li, J. (2009). The Research of PHP Development Framework Based on MVC Pattern. Proceeding. (Pp. 947-949).

Baskerville, R. (1999). Investigating Information Systems with Action Research. Communications of the Association for Information Systems. Volumen 2, (Pp. 1-32).

Bosch, J. Molin, P. Mattsson, M. Bengtsson, P. y Fayad, M. (1999). Object-oriented foundations of framework design. Documento en línea. Disponible en: <http://www.wiley.com/legacy/compbooks/frameworks/preface.htm>. Consulta: 13/05/2023.

Bentacourt, A., Mujica, A., & Dario, H. (2003). Procesamiento difuso de imágenes: Filtro difuso para eliminar el ruido impulsivo. *Ciencia Investigación Académica Desarrollo*, 8, 40 a 46.

Natsev A., Rastogi R., Shim K., "WALRUS: A Similarity Retrieval Algorithm for Image Databases", Proceedings of the ACM SIGMOD 1999 Philadelphia. Pags. 395-405.

Naiouf M., Tarrío D., De Giusti L., De Giusti A., “ Paralelización del Análisis de Similitud de Imágenes”, En los anales del VI Congreso Argentino de Ciencias de la Computación, CACIC 2000, (Ushuaia Argentina) 2000.

Miles, R. y Hamilton, K. (2006). Learning UML 2.0. United States. O'Reilly. PHP. (2012). Php.net. PHP Nuevas Características. <http://php.net/manual/es/migration54.new-features.php>. Consulta: 13/05/2023.

Aguiar, A. y David, G. (2000). A Minimalist Approach to Framework Documentation. Proceedings. (Pp. 1-11).

Escalante (2006,). *Procesamiento Digital de Imágenes*. UNAM. Recuperado 1 de octubre de 2023, de http://lapi.fi-p.unam.mx/wpcontent/uploads/PDI_Cap1_Introduccion.pdf

Aguiar, A. y David, G. (2011). Patterns for Effectively Documenting Frameworks. Lecture Notes in Computer Science. Volume 6510, (Pp. 79-124).

SOTO Veragua, Jorge. Del trazo al chip: la técnica gráfica y sus sistemas. Ediciones B&B, 2000, Santiago – CHILE.

Dimes, T. (2015). *JavaScript Una Guía de Aprendizaje para el Lenguaje de*

Programación JavaScript. Babelcube Inc.

Reyes-Ruiz, L. & Carmona Alvarado, F. A. (2020). La investigación documental para la comprensión ontológica del objeto de estudio.

<https://hdl.handle.net/20.500.12442/6630> Consultada: 13/05/2023

Lozada(2014) Investigación aplicada [Vol. 3, N°. 1, 2014](#), págs. 47-50

Hernández Sampieri, R., Fernández Collado, C., & Baptista, M. P. (2014). Metodología de la investigación. D.F.: McGrawHill.

Viera, G. (2017). Procesamiento de imágenes usando OpenCV aplicado en Raspberry Pi para la clasificación del cacao (Tesis de licenciatura en Ingeniería Mecánico-Eléctrica). Universidad de Piura, Facultad de Ingeniería. Programa Académico de Ingeniería Mecánico-Eléctrica. Piura, Perú.

<https://hdl.handle.net/11042/2916> Consultada 26/04/2023

Nieto, J. (2016). Desarrollo de una aplicación web, con Front-end y Back-end para compraventa de segunda mano. Valencia: Universidad Politecnica de Valencia.

Montesdeoca, Al. (2022). Aplicación de un algoritmo basado en morfología matemática a la extracción de zonas inundadas en imágenes de satélite. [Trabajo de fin de grado, Universidad Politécnica de Madrid]. Archivo digital UPM <https://oa.upm.es/70931/>

Ortiz, F. (2022). Procesamiento morfológico de imágenes en color: aplicación a la reconstrucción geodésica. Tesis doctoral, Universidad de Alicante. Repositorio Institucional de la Universidad de Alicante. <http://hdl.handle.net/10045/10053>

Ruiz Fernández, L. Á. (2020). Aplicación de filtros morfológicos en imágenes. <http://hdl.handle.net/10251/145903>

González Pinzón, M. F., & González Sanabria, J. S. (2013). Aplicación del estándar ISO/IEC 9126-3 en el modelo de datos conceptual entidad-relación. Revista Facultad de Ingeniería, UPT, 22(35), 113-125. Recuperado en 7 de Febrero del 2023, de <https://www.redalyc.org/pdf/4139/413940774009.pdf>

Abud Figueroa, M. A. (2004). Calidad en la industria del software. La norma ISO 9126. Revista UPIICSA, 34(2). Recuperado en 7 de Marzo del 2023, de <https://www.repositoriodigital.ipn.mx/bitstream/123456789/5321/2/34-2.pdf>

Normas ISO 25000. (2022) ISO 25010. <https://iso25000.com/index.php/normas-iso-25000/iso-25010>

Díaz, C., & Toro, M. A. (2020). SARCoV-2/COVID-19: The virus, the disease and the pandemic. Medicina y Laboratorio, 183-205.

Melián, R. A., Calcumil, H. P., Boin, B. C., & Carrasco, S. ,. (2020). Detección de COVID -19 (SARS-CoV-2) mediante la saliva: Una alternativa diagnóstica

poco invasiva. *International journal of odontostomatology*, 316-320

Moore, K., Dailey, A., & Agur, A. (2013). *Anatomía con orientación clínica*. EE.UU.: Editorial médica panamericana.

Maguiña, Q. J., Murguía, L., Verona, M., Gomero, C. R., & Véliz, J. (2021). Factores asociados a hallazgos anormales en radiografías digitales de tórax en trabajadores asintomáticos en Lima, Perú. *acta médica peruana*.

Takeyas, B. L. (2007). *Introducción a la inteligencia artificial*. Instituto Tecnológico de Nuevo Laredo.

Boden, M. (2016). *Inteligencia Artificial*. Madrid: AI. Its Nature and future KeepCoding,

R. (2022, 26 agosto). Preprocesamiento de imágenes de entrada. *KeepCoding*

Bootcamps. <https://keepcoding.io/blog/preprocesamientoimagenesentrada/#:~:textn>

[%20preprocesamiento%20de%20im%C3%A1genes%20de,el%20an%C3%A1lisis%20del%20Big%20Data.](https://keepcoding.io/blog/preprocesamientoimagenesentrada/#:~:textn)

Technologies, U. (s. f.). *Visión general del post-procesamiento - Unity Manual*.

[https://docs.unity3d.com/es/2017.4/Manual/PostProcessingOverview.html#:~:text=Post%2Dprocessing%20\(post%2Dprocesamiento,con%20poco%20tiempo%20de%20configuraci%C3%B3n](https://docs.unity3d.com/es/2017.4/Manual/PostProcessingOverview.html#:~:text=Post%2Dprocessing%20(post%2Dprocesamiento,con%20poco%20tiempo%20de%20configuraci%C3%B3n)

Matich, D. J. (2001). *Redes Neuronales: Conceptos Básicos y Aplicaciones*.
Universidad Tecnológica Nacional, México.

Torres, J. (2018). *Deep learning Introducción práctica con Keras*. Barcelona: Watchthis space.

García, A. R. (2019). Sistema de percepción de elementos viarios usando técnicas de visión por computador para aplicación en conducción autónoma. *Escuela Técnica Superior de Ingeniería Industrial*.

Fuentes Peralta, N., & Vera Martínez, S. (2021). *Desarrollo De Un Sistema Web Utilizando Procesamiento De Imágenes De Rayos X De Tórax Para La Detección De Covid 19 Mediante Deep Learning*. Universidad Estatal De Milagro Facultad Ciencias E Ingeniería.

Zayat, W., Senvar O.(2020). Framework Study for Agile Software Development Via Scrum and Kanban. *International Journal of Innovation and Technology Management*, 8-10.

Sawalha, S., AbdelNabi,H.(2020). Agile Software Development: Methodologies and Trends. Princess Sumaya University for Technology, Amman, Jordan. 258-261.

Alfonso, B.(2021). Detección precoz de covid-19 a partir de imágenes de radiografías de tórax mediante redes neuronales convolucionales. Universidad Oberta de

Catalunya. Pag 41-42. Recuperado de:

<https://openaccess.uoc.edu/bitstream/10609/132968/6/balfonsoITFM0621memoria.pdf>

f

Bedolla-Solano, J.J., Bedolla-Solano, R., y Miranda-Esteban, A. (2022). Prácticas digitales y sostenibilidad socioambiental en la educación básica de las Smart Cities.

RED. Revista de Educación a Distancia, 23(72). <http://dx.doi.org/10.6018/red.533831>

Alex O. Sánchez Huarcaya. (2020). Los métodos de investigación para la elaboración

de las tesis de maestría en educación. Recuperado de:

<https://repositorio.pucp.edu.pe/index/handle/123456789/195750>

Anexo

Anexo A: Entrevista

Proyecto: Sistema de análisis de imágenes con Python para la detección de casos COVID-19 a través de imágenes radiológicas de los pacientes.

La siguiente entrevista está dirigida a médico en servicio del sector salud del estado Guerrero, México, haciendo énfasis en región de Acapulco de Juárez La entrevista es con fines académicos y servirá de apoyo para el proyecto de investigación “Sistema de análisis de imágenes con Python para la detección de casos COVID-19 a través de imágenes radiológicas de los pacientes”.

Responsable del proyecto los estudiantes:

- Juan José Rosas Flores
- Jaime Ulises Loeza Silverio

1. ¿Cuál es su último nivel de estudio?

Licenciatura en medicina

2. ¿Cuánto tiempo lleva trabajando como Médico?

A partir del 2004

3. En tiempo de COVID-19, ¿Cómo usted diagnosticaba al paciente si tenía dicha enfermedad?

Al principio de pandemia había problemas para diagnosticar dicha enfermedad, ya que la gente tenía una mala información sobre esta enfermedad, ya que la confundía con tos cualquiera o en lo peor de los casos con tuberculosis, por lo cual yo lo enviaba

a que se hicieren la prueba de COVID-19 en su laboratorio de confianza, pero actualmente ya se le puede dar un diagnóstico a través de radiografías

4. En los últimos 3 años trabajando en el sector, ¿ha tenido problema o dificultad al diagnosticar COVID-19 a través de imágenes radiológicas?

Sí, en algunas ocasiones las imágenes están de mala calidad, por lo cual no se le puede dar un resultado 100% seguro

5. ¿Sabe del uso de herramientas tecnológicas para la detección del COVID-19?

Sí, he sabido que algunos lugares utilizan Inteligencia artificial para la detección de dicha enfermedad

6. ¿Cree que es seguro el uso herramientas tecnológicas para la detección de la enfermedad COVID-19?

En mi caso sí, pero depende de cada médico

7. ¿Cómo identifica en una radiografía del tórax la enfermedad COVID-19?

Hay varias formas, pero yo mayormente observo es la opacidad del espacio aéreo en forma de consolidación y/u opacidades en vidrio deslustrado, con distribuciones típicamente bilateral, periférica y de predominio en los campos inferiores

8. ¿Qué tratamiento utiliza cuando el paciente da positivo a COVID-19?

Primordialmente, descanso y de manera general tratamiento antiviral

9. ¿Cree que la aplicación de inteligencia artificial ayudaría para la detección de otras enfermedades que no sean COVID-19?

Si, espero que un futuro esta herramienta contribuyan no solo al sector salud sino a la propia ciudadanía.

10. ¿Cuál es el costo aproximado para el diagnóstico de COVID-19 mediante radiografías del tórax?

Eso depende del médico y del mismo hospital, en mi caso solo cobro la consulta(100.00mx).

11. ¿Usted cree que las herramientas tecnológicas para la detección del COVID-19 ayudaría en optimizar el tiempo de respuesta, así mismo ayudar o ser una guía para un diagnóstico más certero?

Si claro que sí.