



**Software para el reconocimiento y análisis predictivo de radiografías en
pacientes de LPH (Labio Paladar Hendido)**

(Autor-es)

**RICHARD ALEJANDRO MORA PERILLA
BRAYAN NICOLAS GETZAMHA PEÑA**

**UNIVERSIDAD EL BOSQUE
PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
Bogotá, 2023**

**Software para el reconocimiento y análisis predictivo de radiografías en
pacientes de LPH (Labio Paladar Hendido)**

(Autor-es)

**RICHARD ALEJANDRO MORA PERILLA
BRAYAN NICOLAS GETZAMHA PEÑA**

**Trabajo de Grado presentado como requisito para optar al título de
INGENIERO DE SISTEMAS**

**Modalidad de Grado:
DESARROLLO TECNOLÓGICO**

Director

**Ing. Wilson Mauro Rojas Reales
Mg. Seguridad Informática**

**UNIVERSIDAD EL BOSQUE
PROGRAMA DE INGENIERÍA DE SISTEMAS
FACULTAD DE INGENIERÍA
Bogotá, año**

DEDICATORIA

En agradecimiento a nuestra familia quienes han sido el motor para el esfuerzo constante invertido en el desarrollo de este proyecto, nuestra carrera y nuestra vida, y a nuestros maestros quienes nos dieron todo el conocimiento necesario para llevarlo a cabo con éxito.

AGRADECIMIENTOS

El desarrollo y culminación de este proyecto se le agradece al Ing. Mg. Wilson Mauro Rojas Reales quien dirigió los pasos a seguir para lograr este proyecto quien con base a sus conocimientos y experiencia ha dado una correcta dirección sobre los participantes.

Tabla de contenido

Contenido

1.	Introducción	4
2.	Descripción Contexto y Justificación del Problema Desde el Modelo Biopsicosocial y Cultural.....	5
3.	Marco Referencial.....	5
3.1.	Antecedentes y Estado Del Arte.....	5
3.2.	Marco Teórico	6
4.	Descripción de la Solución Desde el Modelo Biopsicosocial y Cultural	8
4.1.	Objetivos del Proyecto.....	9
4.2.	Descripción del Artefacto.....	10
4.3.	Componente de Análisis: Descripción de la Transformación Esperada del Contexto.....	10
5.	Diseño Metodológico	10
6.	Resultados y Discusión.....	21
7.	Conclusiones	23
8.	Lecciones Aprendidas y Trabajo Futuro	24
9.	Referencias.....	24
	Anexos.....	26
	MODELO BPSC DEL PROBLEMA	26
	MODELO BPSC DE LA SOLUCIÓN	26
	CRONOGRAMA	27
	REQUERIMIENTOS FUNCIONALES.....	27
	REQUERIMIENTOS NO FUNCIONALES	27
	DIAGRAMA DE CASOS DE USO.....	27
	HISTORIAS DE USUARIO.....	28
	PRODUCT BACKLOG.....	28
	MODELO RELACIONAL DE LA BASE DE DATOS	28
	TABLERO DE CONTROL KANBAN	28
	DIAGRAMA DE ARQUITECTURA	28

Software para el reconocimiento y análisis predictivo de radiografías en pacientes de LPH (Labio Paladar Hendido)

Software for the recognition and predictive analysis of radiographs in LPH (Cleft Lip Palate) patients

Richard A. Mora P. y Brayan N. Getzamha P.
rmorap@unbosque.edu.co, bgetzamha@unbosque.edu.co
Universidad El Bosque, Colombia

Resumen— El Labio Paladar Hendido o LPH es una condición congénita que afecta gran porcentaje de la población en el mundo, esta condición es tratada bajo una serie de tratamientos multidisciplinarios donde en la mayoría de los casos es necesaria una intervención quirúrgica. Con base en lo anterior se planteó el desarrollo de un software que tuviera la capacidad de clasificar el caso y predecir la necesidad de intervención quirúrgica del paciente, siendo este un gran apoyo para los profesionales de la salud en el área. Este software apoyaría el proceso de diagnóstico que ha venido siendo realizado a experiencia sin herramienta alguna, este proyecto acortaría tiempos para una mejor toma de decisiones y ayudaría a sesgar el punto de vista del profesional de la salud en veras de descartar dudas y confusiones sobre la imagen diagnóstica.

Abstract— Cleft Lip Palate or LPH is a congenital condition that affects a large percentage of the population in the world, this condition is treated under a series of multidisciplinary treatment where in most cases surgical intervention is necessary. Based on the above, we proposed the development of a software that would have the ability to classify the case and predict the need for surgical intervention of the patient, this being a great support for health professionals in the area. This software would support the diagnostic process that has been done without any tool, this project would shorten times for better decision making and would help to bias the point of view of the health professional in order to rule out doubts and confusion about the diagnostic image.

Palabras Clave— Labio Paladar Hendido, LPH, Red Neuronal, Redes Convolucionales, Diagnóstico médico, clasificación.

1. INTRODUCCIÓN

La condición de Labio Paladar Hendido o LPH es una de las enfermedades más comunes en el mundo, el proceso esta condición inicia a partir del desarrollo embrionario del paciente por lo que su tratamiento inicia desde el nacimiento del infante. Dicho esto, se busca mejorar el proceso de diagnóstico que se realiza sobre los pacientes que poseen la condición de LPH, dado que para realizar este diagnóstico se requiere de gran experiencia a la hora de estudiar una imagen diagnostica de este caso que serían las radiografías maxilofaciales de perfil.

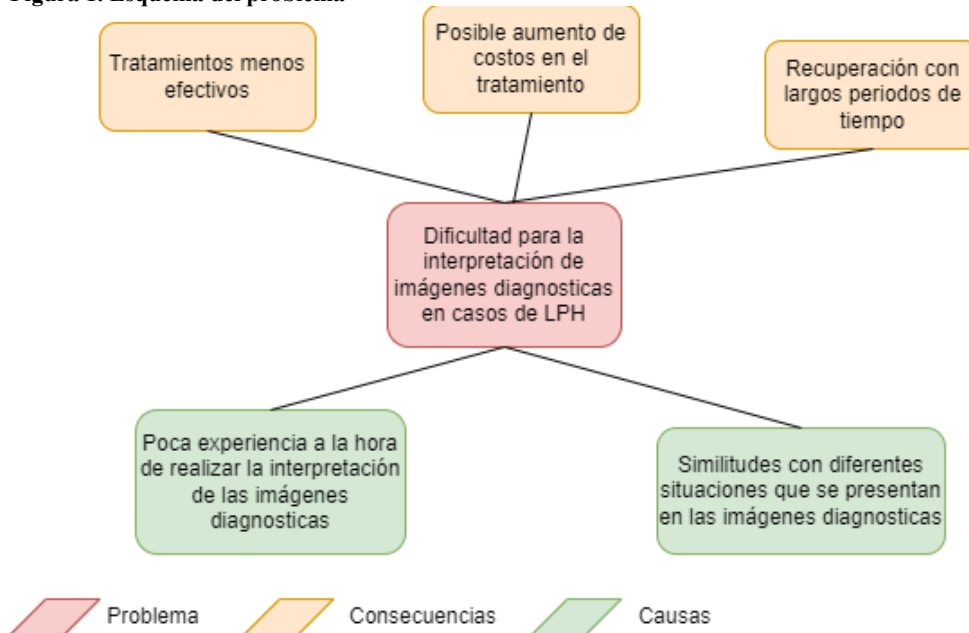
Por lo tanto, el propósito de este producto de software busca apoyar el proceso de análisis sobre las imágenes con el propósito de acortar tiempos en la toma de decisiones de tal manera que le permita al profesional de la salud determinar un tratamiento más efectivo y además minimizar algunas dudas sobre el diagnostico a realizar con base a estas imágenes dado que se presentan casos de confusión con otras condiciones que se puedan presentar sobre el paciente.

Este trabajo se encuentra dividido en dos grandes partes. La primera parte hace referencia a la documentación recolectada mediante encuentros con el grupo de profesionales de odontología y orientaciones recibidas de parte de ellos, la segunda parte es el desarrollo del proyecto desde la ingeniería de sistemas en donde el lector conocerá las actividades que fueron realizadas en cada una de las fases del ciclo de vida de desarrollo del software. Al final, encontrará además algunas recomendaciones a partir de las lecciones aprendidas adquiridas a lo largo del desarrollo de este trabajo.

2. DESCRIPCIÓN CONTEXTO Y JUSTIFICACIÓN DEL PROBLEMA DESDE EL MODELO BIOPSIOSOCIAL Y CULTURAL

Doctores de la facultad de odontología de la Universidad El Bosque durante años realizan tratamientos sobre casos de Labio Paladar Hendido (en adelante LPH) a través de estudios sobre la imagen diagnóstica, específicamente radiografías maxilofaciales de perfil. Este estudio se realiza observando puntos de medida sobre la imagen diagnóstica que se le toma al paciente y este proceso se realiza con experiencia de un profesional de la salud. La característica primordial de estos pacientes es que son menores en edad temprana, es decir 2 años de edad. (Ver Figura 1).

Figura 1. Esquema del problema



Fuente: Elaboración propia

3. MARCO REFERENCIAL

En el presente capítulo se mostrarán los diferentes aspectos documentales sobre los estudios realizados los cuales tienen particular relación con el artefacto perteneciente al presente proyecto.

3.1. Antecedentes y Estado Del Arte

Computer-aided diagnosis of periodontal diseases: Current status and future perspectives

Este artículo proporciona una revisión exhaustiva del estado actual y las perspectivas futuras de la aplicación de la inteligencia artificial en el diagnóstico de enfermedades periodontales, incluyendo LPH. Se describen diferentes técnicas de aprendizaje automático, como redes neuronales, algoritmos de clasificación y regresión, y técnicas de procesamiento de imágenes, que se han utilizado para desarrollar sistemas de ayuda al diagnóstico asistidos por computadora en periodoncia.

Deep Learning for Periodontal Disease Classification on Full-Mouth Radiographs

Este artículo describe un enfoque de aprendizaje profundo (deep learning) para la clasificación de enfermedades periodontales, incluyendo LPH, utilizando radiografías de boca completa. Los autores desarrollaron un modelo de clasificación basado en redes neuronales convolucionales (CNN) que demostró una alta precisión en la detección de diferentes enfermedades periodontales, incluyendo LPH.

Automated Diagnosis of Periodontal Disease using Deep Learning Algorithms: A Systematic Review

Este artículo es una revisión sistemática de la literatura sobre el uso de algoritmos de aprendizaje profundo en el diagnóstico automatizado de enfermedades periodontales, incluyendo LPH. Los autores examinan varios estudios y enfoques de aprendizaje profundo, incluyendo redes neuronales convolucionales (CNN) y redes neuronales recurrentes (RNN), y discuten su precisión y utilidad en la clasificación automatizada de enfermedades periodontales.

Automated classification of periodontal bone loss using deep learning-based radiomics

Este estudio propone un enfoque de radiómica basado en aprendizaje profundo para la clasificación automatizada de la pérdida ósea periodontal en radiografías panorámicas. Los autores desarrollaron un modelo de aprendizaje profundo basado en redes neuronales convolucionales (CNN) que demostró una alta precisión en la clasificación de diferentes niveles de pérdida ósea periodontal, incluyendo LPH.

Machine learning for periodontal diagnosis: A systematic review

Este artículo es una revisión sistemática de la literatura sobre el uso de técnicas de aprendizaje automático, incluyendo inteligencia artificial, en el diagnóstico de enfermedades periodontales, incluyendo LPH. Los autores revisan diferentes estudios que utilizan algoritmos de clasificación y regresión para la detección y diagnóstico de LPH y otras enfermedades periodontales, y discuten la precisión y limitaciones de estos enfoques.

Artificial intelligence in the diagnosis of periodontal diseases: A review

Este artículo ofrece una revisión completa de la aplicación de la inteligencia artificial en el diagnóstico de enfermedades periodontales, incluyendo LPH. Los autores revisan diferentes enfoques de aprendizaje automático, como redes neuronales, algoritmos de clasificación y regresión, y técnicas de procesamiento de imágenes, que se han utilizado en la detección y diagnóstico automatizado de enfermedades periodontales, y discuten los desafíos y oportunidades en este campo.

3.2. Marco Teórico

La implementación de una red neuronal artificial (CNN) es fundamental para el proceso de análisis de imágenes diagnósticas en casos de LPH puesto que existen distintos tipos o clasificaciones de la enfermedad los cuales se determinan al cumplir con ciertas características basadas en la anatomía facial de los pacientes.

La red neuronal convolucional se compone de forma similar a otras redes, pero esta cuenta con la particularidad de que en sus capas ocultas se realizan convoluciones y max pooling, lo cual busca reducir la imagen conforme recorre sus capas, en cada capa quedan los elementos considerados más importantes, por lo que su objetivo fundamental es obtener patrones o características relevantes como profundidad y sombras, logrando identificar formas complejas [8]. La identificación del LPH se logra mediante el análisis de las áreas involucradas las cuales pueden ser el área bucal como también el área nasal. A lo largo del tiempo se han venido desarrollando CNN las cuales han demostrado ser efectivas, exactas y con un alto porcentaje de confiabilidad con respecto a los aciertos.

En la actualidad existen diversos tipos de lenguajes de programación aplicables al desarrollo de redes neuronales artificiales, sin embargo, al momento de realizar la codificación podemos encontrar que algunos de estos lenguajes se pueden identificar distintas ventajas por lo cual destacan unos más que otros [8]. Los lenguajes de programación más fuertes y demandados en la actualidad al momento de realizar análisis de datos son el lenguaje R y Python, estos lenguajes ofrecen la simplicidad a nivel de código para construcciones robustas de sistemas.

La comunidad y paquetes incluidos en el lenguaje de programación son fundamentales a la hora de escoger, puesto que esto nos podrá garantizar la mantenibilidad a lo largo del tiempo, como también un nivel bajo esfuerzo para la implementación de redes neuronales profundas, el lenguaje Python cuenta paquetes como TensorFlow y Keras que fueron desarrollados con el objetivo de simplificar la construcción y entrenamiento de

las redes neuronales buscando permitir que se desarrollen en el menor tiempo posible [1], [2].

Labio Paladar Hendido

El LPH es una de las malformaciones congénitas más comúnmente padecidas en el rostro, donde se puede ver involucrada el área bucal como también el área nasal. El padecimiento de esta malformación tiene complicaciones que podrían afectar sustancialmente el desarrollo normal de la persona, puesto que se podrían presentar problemas a nivel respiratorio, visual, auricular, dental, alimenticio, estético, social y emocional.

La importancia de un control a tiempo del tratamiento de este padecimiento está fundamentada en que, si el control no se realiza a tiempo, la supervivencia del paciente es la que podría verse afectada buscando disminuir las posibles complicaciones que conlleva esta malformación. Esto debido a que desde el nacimiento del bebé podría verse expuesto a infecciones y/o complicaciones respiratorias y alimentarias dependiendo del tipo del tipo de LPH. El traumatismo generado en estos pacientes es debido a que normalmente se requiere de varias cirugías las cuales pueden ser de tipo reconstructivo o estético, sin tener en cuenta los distintos controles con especialistas que se tendrán que llevar a cabo puesto que se necesita de un tratamiento multidisciplinario de las distintas áreas de la salud [3].

Clasificaciones

- Fisura palatina completa (Involucramiento labial)
- Fisura paladar anterior (Fosa anterior)
- Fisura paladar posterior (Detrás de la fosa)
- Fisura submucosa

Existe un tipo de clasificación adicional que se encuentra basado en la anatomía de los pacientes. En los casos donde se ve afectado el labio, podrá ser nombrado como LPH unilateral en el caso que solo sea en el lado derecho o en el lado izquierdo exclusivamente, o también podría ser nombrado como LPH bilateral en el caso es que comprometa ambos lados del labio y por último dependiendo de si se ve o no afectado el área nasal se le agregaría al nombramiento la palabra completo o incompleto correspondientemente [4].

Redes Neurales

El funcionamiento de las redes neuronales artificiales en su estado más básico está basado en el uso de nodos, estos nodos buscan tener comportamientos como el de una neurona humana, cada uno de estos nodos podrían estar conformados a partir de otro conjunto de nodos.

El funcionamiento de los nodos que componen la red está dado en un valor de entrada y un valor de salida también llamado valor de activación, si el nodo recibe valor un valor de entrada en base a este se construirá un valor de salida para el caso de las redes con dos capas puesto que las redes neuronales pueden implementar dos o más capas intermedias, también llamadas capas ocultas las cuales están hechas para resolver problemas más complejos.

Entrenamiento de la red neuronal

Las redes neuronales artificiales necesitan de aprendizaje que se lleva a cabo a través de un entrenamiento, entrenamiento que, de ser aplicado de forma correcta, podría garantizar el funcionamiento acorde a la necesidad que se busca resolver.

El entrenamiento se clasifica en dos tipos, entrenamiento supervisado y no supervisado.

- El entrenamiento supervisado busca generar distintos escenarios que apliquen valores de activación distintos con el fin realizar una comparativa frente a las ejecuciones ideales cargadas previamente a la red neuronal y generar así un tipo de historial basado en las relaciones que logró identificar [5].
- El entrenamiento no supervisado busca encontrar algún tipo de patrón de similitud entre los distintos datos para realizar agrupaciones o categorizaciones basadas en estos patrones sin necesidad de realizar ninguna comparativa con casos ideales, solo basada en la experiencia que gana tras cada ejecución. Posterior a este entrenamiento la red neuronal intentará clasificar los nuevos datos en alguna de las agrupaciones definidas

según el patrón que analice en la información testeada [6].

Tipos de redes neuronales

La red neuronal profunda está especializada en el procesamiento de texto, imágenes pequeñas y datos numéricos. Esta red se compone de una capa de entrada, una capa de salida y una capa intermedia donde se almacenan las capas ocultas.

Dentro de la capa oculta se podrían encontrar dos o más capas, capas en las cuales todos sus nodos se encuentran conectados de capa a capa, razón por la cual esta red cuenta con múltiples conexiones en su interior que tiende a degradar el estado computacional cuando se trabaja con datos muy grandes.

La red neuronal recurrente está especializada en el procesamiento de datos secuenciales dado que su valor de salida siempre va a depender del valor de entrada y de un histórico o valor predecesor, razón por la cual el orden de los datos procesados va a ser fundamental. Un ejemplo práctico de estos datos secuenciales son los textos o el lenguaje natural, puesto que el orden de cada palabra es crucial para dar sentido a una frase.

En la estructura de esta red encontramos que en la capa de en medio, la cual es llamada capa recurrente, cuenta con una LSTM (memoria corta a largo plazo), la cual permite almacenar el resultado previo para que la capa oculta permanezca realimentando a ella misma de forma recurrente.

Las redes neuronales convolucionales (CNN) son un tipo de red neuronal artificial, la cual su realiza su procesamiento mediante la estructura multicapa jerarquizada, esta red neuronal tiene su fuerte en el procesamiento de imágenes, aunque también es usada para procesamiento de texto, esta red nace con la necesidad de procesar imágenes de forma efectiva y eficiente.

Esta red se compone de forma similar a otras redes, pero esta cuenta con la particularidad de que en sus capas ocultas se realizan convoluciones y max pooling, buscando reducir la imagen conforme pasa de capa en capa la información, quedando en cada una de las capas los elementos que considera esta los más importantes, por lo cual se puede decir que su objetivo fundamental es obtener patrones o características relevantes en sus valores de entrada partiendo desde los patrones más básicos al inicio de sus capas como líneas y formas, hasta los patrones más complejos en sus capas más profundas como profundidad y sombras, logrando identificar formas complejas [7].

La identificación y clasificación de patrones generales la logra establecer mediante la verificación de grandes cantidades de imágenes a las cuales analiza, normaliza, filtra y procesa.

A lo largo del tiempo se han venido desarrollando CNN las cuales han demostrado ser efectivas, exactas y con un alto porcentaje de confiabilidad con respecto a los aciertos. Algunas de las más usadas son: LeNet, AlexNet, VGG, GoogleNet, ResNet y DenseNet[8].

4. DESCRIPCIÓN DE LA SOLUCIÓN DESDE EL MODELO BIOPSIOSOCIAL Y CULTURAL

Se plantea el análisis del modelo Biopsicosocial y Cultural, pero ahora basado en el enfoque de la implementación de la solución web especificada en el presente proyecto de investigación. En primer lugar, se provee un análisis preliminar del modelo Biopsicosocial y Cultural según la percepción después de la implementación de la solución planteada, con base en: artefactos, Medios, creencias y hábitos (Ver Tabla 1).

Adicionalmente, se provee un análisis más detallado, en donde se relacionan las dimensiones de este modelo en el problema en cuestión, para una visualización gráfica del modelo biopsicosocial y cultural consultar el anexo (Ver Figura 19).

Tabla 1. Modelo BPSC de la solución

	Paciente	Profesional de la salud
Artefactos	- El paciente ingresa a la app donde realiza la carga de sus imágenes diagnósticas. - El paciente recibe de parte de la app una clasificación a partir del análisis realizado por la red neuronal.	- El profesional de la salud realiza carga de imágenes diagnósticas masivas para entrenamientos de la red neuronal. - El profesional de la salud carga imágenes diagnósticas de sus pacientes, para realizar consultas de categorización del LPH del paciente.
Medios	El paciente no necesita asistir al centro especializado para recibir la clasificación del LPH diagnosticado.	El profesional de la salud se encuentra prestando sus servicios profesionales en el centro de salud.
Hábitos	- Tras una oportuna categorización del LPH, el paciente asiste a su tratamiento y mejora sustancialmente. - Ya no es necesario que el paciente se realice múltiples estudios radiográficos para una categorización del LPH.	- El profesional de la salud realiza una previa evaluación de las imágenes diagnósticas por el sistema para corroborar su diagnóstico. - Ya no es necesario solicitar al paciente que se realice múltiples estudios radiográficos para corroborar lo visualizado en las imágenes diagnósticas.
Creencias	- Resulta muy fácil realizar la carga de las imágenes diagnósticas y obtener el resultado de la categorización. - El sistema arroja muy rápido el resultado de la clasificación del LPH. - Ya no resulta necesario el asistir a una cita exclusiva de lectura de exámenes para saber el tipo de LPH presentado. - El sistema se puede estar equivocando en la categorización del LPH. - Se puede obtener un diagnóstico sin salir de casa.	- El sistema se puede llegar a tomar bastante tiempo procesando las imágenes diagnósticas. - Se pueden llegar a reducir el número de consultas realizadas a los centros de salud. - El sistema puede apoyar en el diagnóstico de categorización del PLH. - El sistema puede realizar cauterizaciones equivocadas por falta de entrenamiento.

Fuente: Elaboración propia

4.1. Objetivos del Proyecto

Objetivo general

Desarrollar un software basado en inteligencia artificial adoptando redes neuronales para el análisis y clasificación imágenes diagnósticas (radiografías) de estudios realizados de LPH (Labio Paladar Hendido) con el fin de apoyar a los profesionales de la salud -en odontología- en la toma de decisiones para seleccionar el mejor tratamiento a seguir.

Objetivos específicos

1. Seleccionar el mejor modelo de red neuronal para la clasificación de imágenes.

2. Diseñar un modelo de inteligencia artificial que adopte el modelo de red neuronal seleccionado con el fin de clasificar los casos de LPH.
3. Entrenar el modelo de red neuronal hasta que se obtenga un alto porcentaje de precisión para la clasificación de casos de LPH.
4. Realizar una evaluación de la precisión del modelo entrenado frente a un conjunto de datos preseleccionado y categorizado por profesionales de la salud.

4.2. Descripción del Artefacto

Como solución a la problemática planteada la cual se define como, según el objetivo general (Ver capítulo 4.1), un software basado en inteligencia artificial adoptando el uso de redes neuronales que analice y clasifique imágenes diagnósticas seleccionadas para este proyecto como radiografías maxilofaciales o craneofaciales de perfil.

A partir de lo anteriormente descrito como solución, se espera que una vez se haya implementado el software, los profesionales de la salud (odontólogos) logren clasificar por medio de los resultados del software el caso de LPH en estudio de tal manera que en una primera etapa le permita al profesional tomar decisiones en caso de encontrar algún dato relevante al respecto.

4.3. Componente de Análisis: Descripción de la Transformación Esperada del Contexto

La solución permitirá una rápida identificación y clasificación de los diferentes tipos de LPH, lo que facilitaría la planificación del tratamiento y mejoraría la precisión en el diagnóstico.

Las mejoras son las siguientes:

Ahorro de tiempo: Un clasificador de casos de LPH puede clasificar automáticamente las imágenes y proporcionar una evaluación rápida y precisa del caso, lo que ahorraría tiempo en el diagnóstico.

Mayor precisión en el diagnóstico: Un clasificador de casos de LPH puede identificar patrones y características específicas de las imágenes que pueden ser difíciles de detectar para el ojo humano, lo que mejora la precisión en el diagnóstico y en la planificación del tratamiento.

Estándares de clasificación: Un clasificador de casos de LPH puede seguir estándares de clasificación predefinidos, lo que proporcionaría una mayor consistencia en la evaluación y clasificación de los casos.

Mejora de la atención al paciente: Al contar con un diagnóstico más preciso, se podría diseñar un tratamiento adecuado para el paciente, mejorando así la atención que se le brinda.

En resumen, un clasificador de casos de LPH puede ser una herramienta muy útil para los odontólogos en la universidad El Bosque, ayudándoles a hacer un diagnóstico más preciso, ahorrar tiempo y mejorar la atención al paciente.

5. DISEÑO METODOLÓGICO

La metodología SCRUM fue seleccionada como marco de trabajo para llevar a cabo el desarrollo de este proyecto. La elaboración del Product Backlog permite priorizar y tener una visión general de la realización de las tareas [4], fueron necesarios tres (3) sprints para realizar la construcción del producto. En cada uno de ellos tal como lo manifiesta Scrum, fueron celebradas las ceremonias propias de este marco de trabajo de tal manera que la realimentación fue fundamental para lograr el cumplimiento de los objetivos.

FASE DE ANÁLISIS

Con base en reuniones previas realizadas con el grupo de odontólogos, se recopiló la información básica necesaria para estudiar el contexto y el entorno de estudio. Adicionalmente, esta información permitió elaborar algunas historias de usuarios. Adicionalmente, en esta etapa se llevó a cabo el estudio e indagación de diversas temáticas acerca de redes neuronales. Este estudio permitió definir una estructura básica desde el punto de vista de arquitectura de la solución. A continuación, se describen fragmentos de Historias de Usuario (Ver Figura 2 y 3).

Figura 2. Historia de Usuario 1

Historia de Usuario	
Código: HU-001	Usuario: Administrador - Usuario
Nombre historia: Crear usuario	
Prioridad en negocio: 10	Riesgo en desarrollo: No aplica
Responsable: Scrum Team	
Descripción: Yo como usuario debo poder agregar un usuario al sistema.	
Resultado: Usuario agregado al sistema.	
Validación: El usuario debe poder visualizarse en la tabla de usuarios.	

Fuente: Elaboración propia

Figura 3. Historia de Usuario 5

Historia de Usuario	
Número: HU-005	Usuario: Usuario
Nombre historia: Cargar imagen	
Prioridad en negocio: 10	Riesgo en desarrollo: No aplica
Responsable: Scrum Team	
Descripción: Yo como usuario debo poder cargar una imagen en el sistema para su respectiva categorización.	
Resultado: Categorización de la imagen exitosa.	
Validación: El usuario debe poder evidenciar la categorización de la imagen en el respectivo modulo.	

Fuente: Elaboración propia

Para definir la mejor estructura del modelo a emplear se realizó una revisión de literatura donde se comparaban los distintos tipos de redes neuronales que se diferencian en su estructura de capas, se determinó que la mejor opción para este caso y no solo este caso sino para la lectura de imágenes en general es el tipo de red neuronal convolucional (Convolutional Neural Network) (Ver Figura 4), adicionalmente se estableció que el mejor lenguaje para desarrollar el modelo sería Python gracias a sus librerías el cual facilitan la construcción, implementación y entrenamiento del modelo, se tomó como base un modelo preconstruido llamado MobileNetV2, dado que presentaba altos niveles de exactitud frente a otras estructuras [9] (Ver Tabla 2 y 3).

Tabla 2. Tabla comparativa de modelos

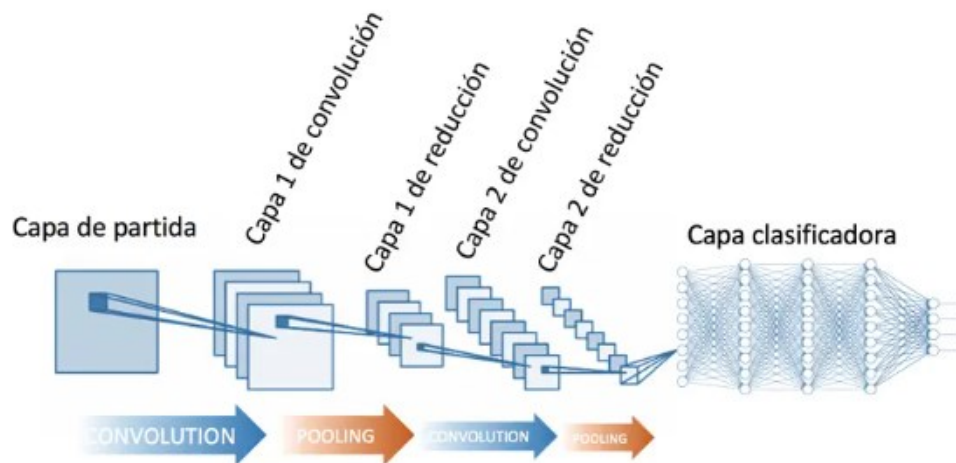
Modelo	Desempeño	Tiempo de entrenamiento
Simple_CNN	82.88 %	400 seg
MobileNetV1	88.60 %	72 seg
MobileNetV2	89.00 %	75 seg

Fuente: Elaboración propia

Tabla 3. Comparación de eficiencia de modelos en red neuronal convolucional

<i>Modelo</i>	<i>Operaciones por segundo (Millones)</i>	<i>Entradas (Millones)</i>	<i>Precisión</i>
<i>ResNet50</i>	3800	25.6	76 %
<i>InceptionV3</i>	5000	23.2	78.8 %
<i>DenseNet121</i>	2800	8	74 %
<i>SqueezeNet</i>	833	1.25	57.5 %
<i>MobileNetV2</i>	300	3.4	72 %
<i>MnasNet</i>	315	3.9	75.2 %

Fuente: C. Luo et al., "Comparison and benchmarking of AI models and frameworks on mobile devices," arXiv.org, <https://arxiv.org/abs/2005.05085> (accessed May 26, 2023).

Figura 4. Estructura de red neuronal convolucional

Fuente: <https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/>

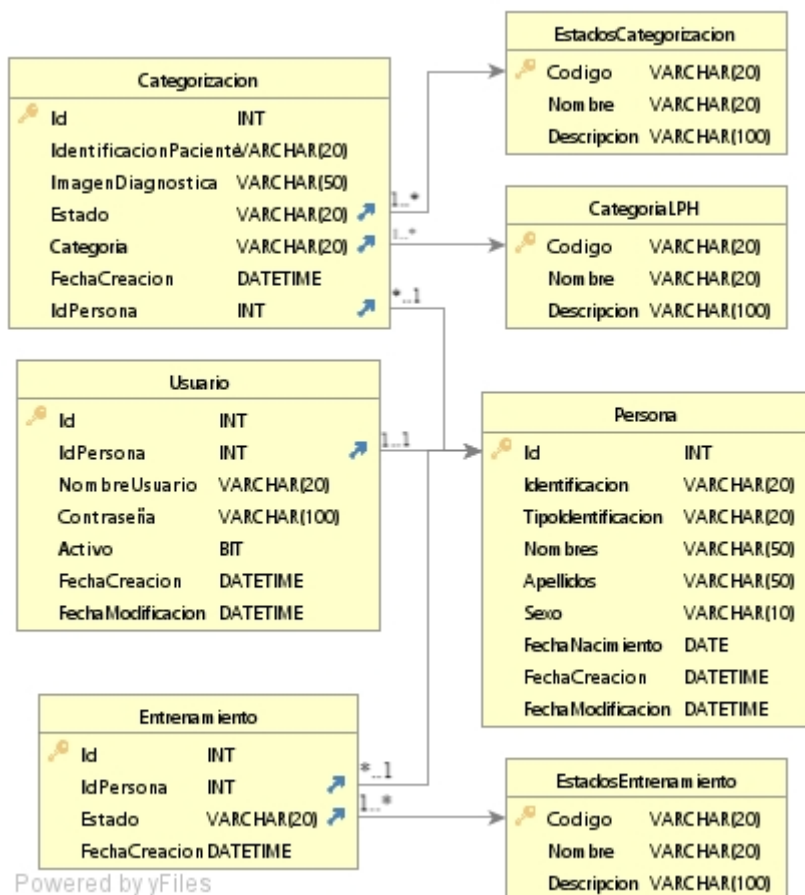
FASE DE DISEÑO

Esta fase comprende el diseño de la estructura de almacenamiento en una primera versión y así mismo la definición preliminar de la arquitectura. De igual manera, con base en el análisis previo e indagación de literatura, fueron seleccionadas las siguientes tecnologías:

- Lenguaje BackEnd: .Net
- Lenguaje FrontEnd: .Net
- Lenguaje del modelo de red neuronal: Python
- Motor de la base de datos: SQLServer
- Plataforma de alojamiento para los servicios: AWS EC2

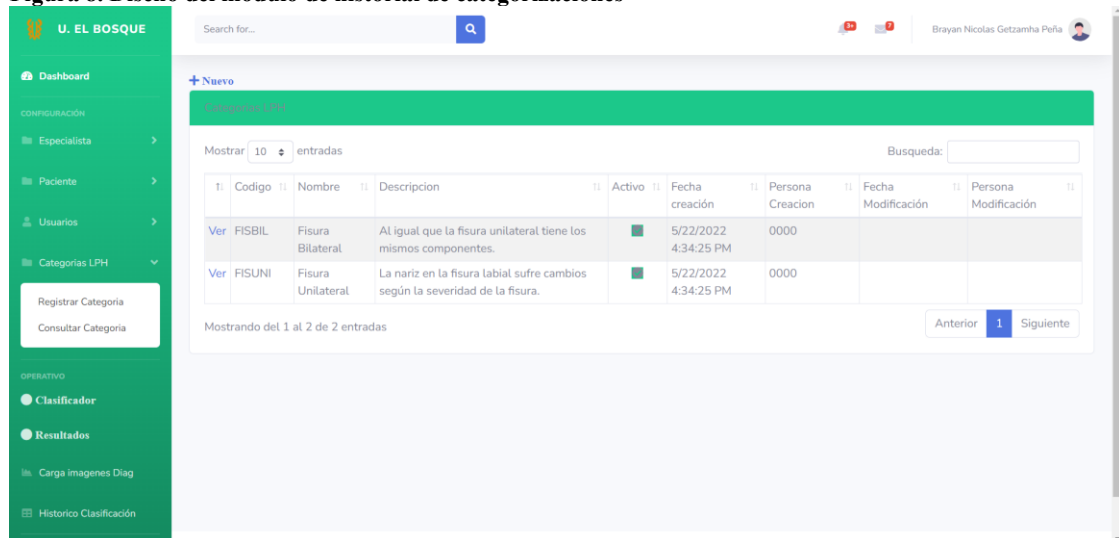
También se definió el diseño de la interfaz de usuario para el software de acuerdo a los requisitos presentados en la fase de análisis, en la siguiente página se presenta un ejemplo del módulo de “Inicio de sesión” y de “Gestión de parámetros” y además, un breve esquema de las tablas iniciales que conforman la base de datos junto a la arquitectura del sistema (Ver Figura 5, 6, 7 y 8)

Figura 5. Modelo relacional de la base de datos



Fuente: Elaboración propia

Figura 8. Diseño del módulo de historial de categorizaciones



Fuente: Elaboración propia

FASE DE CONSTRUCCIÓN

En la fase de construcción se llevó a cabo la construcción del software en cuanto a codificación del modelo de red neuronal, BackEnd de los módulos del software, construcción de la base de datos y construcción de la interfaz de usuario, en última instancia se realizó la implementación del modelo de red neuronal al negocio.

Se realizó la construcción del modelo en lenguaje Python utilizando la librería TensorFlow, se estableció un formato de imágenes en blanco y negro para que el modelo pueda ser entrenado con mayor facilidad y se implementó la estructura MobileNetV2 agregando dos capas con función de activación "relu" y "sigmoid" tomando las siguientes métricas de medición para evaluar el modelo (Ver Figura 9).

- *aaccuracy*: Exactitud del modelo.
- *precision*: Precisión del modelo.

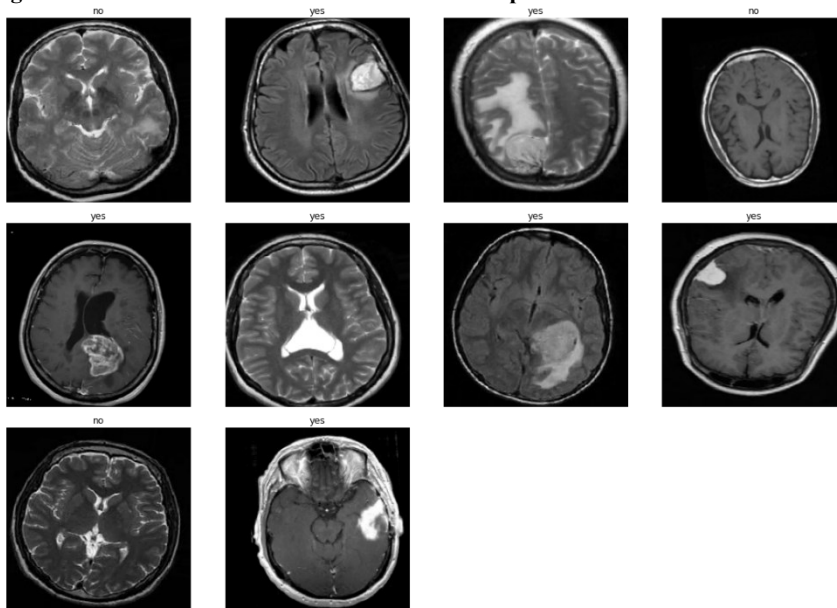
Figura 9. Codificación del modelo de red neuronal

```
[ ] 1 metrics = [
2     tf.keras.metrics.TruePositives(name='tp'), tf.keras.metrics.FalsePositives(name='fp'),
3     tf.keras.metrics.TrueNegatives(name='tn'), tf.keras.metrics.FalseNegatives(name='fn'),
4     tf.keras.metrics.BinaryAccuracy(name='accuracy'), tf.keras.metrics.Precision(name='precision'),
5     tf.keras.metrics.Recall(name='recall'), tf.keras.metrics.AUC(name='auc'),
6     tf.keras.metrics.AUC(name='prc', curve='PR'),
7 ]
8
9 base_model = tf.keras.applications.MobileNetV2(input_shape=(224,224,3),
10                                                include_top=False,
11                                                weights='imagenet')
12
13 base_model.trainable = False
14
15 model = tf.keras.Sequential([
16     base_model,
17     tf.keras.layers.GlobalAveragePooling2D(),
18     tf.keras.layers.Dense(1024, activation='relu'),
19     tf.keras.layers.Dropout(rate=0.2),
20     tf.keras.layers.Dense(1, activation='sigmoid')
21 ])
22
23 model.compile(optimizer=tf.optimizers.Adam(learning_rate=0.00008),
24               loss="binary_crossentropy",
25               metrics=metrics)
```

Fuente: Elaboración propia

Inicialmente, no se pudo contar con imágenes diagnósticas ofrecidas por el grupo de odontólogos para llevar a cabo algunas pruebas iniciales. Por tanto, se inició la implementación del modelo a partir de un set de datos abiertos para casos de tumores cerebrales obtenido de Kaggle, posteriormente se implementaron las imágenes reales del proyecto al modelo (Ver Figura 10).

Figura 10. Muestra de resultado de set de datos de prueba



Fuente: Elaboración propia

FASE DE PRUEBAS

Para las pruebas del modelo se utilizaron 64 imágenes diagnósticas, 34 imágenes de categoría LPHUB (Hendidura bilateral), 13 imágenes de categoría LPHUD (Hendidura derecha) y 21 imágenes de categoría LPHUI (Hendidura izquierda), con este set de imágenes se realizaron 4 configuraciones de “Epoch” de la siguiente manera (Ver Figura 11, 12, 13 y 14).

Figura 11. Prueba de entrenamiento de 5 Epoch

```
Administrador: C:\Windows\System32\cmd.exe - py "C:\inetpub\wwwroot\Red Neuronal\app.py"
9/12 [=====>.....] - ETA: 1s - loss: 1.1179 - tp: 32.0000 - fp: 51.0000 - tn: 31.0000 - fn: 9.0000 -
10/12 [=====>.....] - ETA: 0s - loss: 1.1226 - tp: 36.0000 - fp: 56.0000 - tn: 36.0000 - fn: 10.0000
11/12 [=====>.....] - ETA: 0s - loss: 1.1239 - tp: 40.0000 - fp: 61.0000 - tn: 41.0000 - fn: 11.0000
12/12 [=====>.....] - ETA: 0s - loss: 1.0937 - tp: 45.0000 - fp: 66.0000 - tn: 46.0000 - fn: 11.0000
12/12 [=====] - 6s 471ms/step - loss: 1.0937 - tp: 45.0000 - fp: 66.0000 - tn: 46.0000 - fn: 11
.0000 - accuracy: 0.5417 - precision: 0.4054 - recall: 0.8036 - auc: 0.6449 - prc: 0.5222 - val_loss: 1.0435 - val_tp: 1
0.0000 - val_fp: 14.0000 - val_tn: 10.0000 - val_fn: 2.0000 - val_accuracy: 0.5556 - val_precision: 0.4167 - val_recall:
0.8333 - val_auc: 0.7188 - val_prc: 0.5583
Epoch 5/5
1/12 [=>.....] - ETA: 0s - loss: 0.6368 - tp: 1.0000 - fp: 1.0000 - tn: 1.0000 - fn: 0.0000e+00
2/12 [====>.....] - ETA: 3s - loss: 1.5398 - tp: 2.0000 - fp: 7.0000 - tn: 5.0000 - fn: 4.0000 - ac
3/12 [====>.....] - ETA: 3s - loss: 1.2228 - tp: 7.0000 - fp: 11.0000 - tn: 11.0000 - fn: 4.0000 -
4/12 [====>.....] - ETA: 3s - loss: 1.3999 - tp: 8.0000 - fp: 20.0000 - tn: 12.0000 - fn: 8.0000 -
5/12 [====>.....] - ETA: 2s - loss: 1.3524 - tp: 11.0000 - fp: 27.0000 - tn: 15.0000 - fn: 10.0000
6/12 [====>.....] - ETA: 2s - loss: 1.3329 - tp: 15.0000 - fp: 33.0000 - tn: 19.0000 - fn: 11.0000
7/12 [====>.....] - ETA: 2s - loss: 1.2622 - tp: 19.0000 - fp: 37.0000 - tn: 25.0000 - fn: 12.0000
8/12 [====>.....] - ETA: 1s - loss: 1.2096 - tp: 24.0000 - fp: 42.0000 - tn: 30.0000 - fn: 12.0000
9/12 [====>.....] - ETA: 1s - loss: 1.1807 - tp: 29.0000 - fp: 47.0000 - tn: 35.0000 - fn: 12.0000
10/12 [====>.....] - ETA: 0s - loss: 1.1375 - tp: 34.0000 - fp: 54.0000 - tn: 38.0000 - fn: 12.0000
11/12 [====>.....] - ETA: 0s - loss: 1.1110 - tp: 39.0000 - fp: 59.0000 - tn: 43.0000 - fn: 12.0000
12/12 [====>.....] - ETA: 0s - loss: 1.0824 - tp: 44.0000 - fp: 66.0000 - tn: 46.0000 - fn: 12.0000
12/12 [=====] - 6s 500ms/step - loss: 1.0824 - tp: 44.0000 - fp: 66.0000 - tn: 46.0000 - fn: 12
.0000 - accuracy: 0.5357 - precision: 0.4000 - recall: 0.7857 - auc: 0.6850 - prc: 0.5812 - val_loss: 1.0534 - val_tp: 1
0.0000 - val_fp: 14.0000 - val_tn: 10.0000 - val_fn: 2.0000 - val_accuracy: 0.5556 - val_precision: 0.4167 - val_recall:
0.8333 - val_auc: 0.7274 - val_prc: 0.5333
loss : 1.054782509803772
tp : 10.0
fp : 14.0
tn : 10.0
fn : 2.0
accuracy : 0.5555555820465088
precision : 0.4166666567325592
recall : 0.8333333134651184
auc : 0.7100694179534912
prc : 0.520904541015625
127.0.0.1 - - [03/Oct/2022 02:28:08] "GET /entrenamiento HTTP/1.1" 200 -
```

Fuente: Elaboración propia

Para el entrenamiento de 5 Epoch se obtuvo una precisión del 41% y una exactitud del 55%.

Figura 12. Prueba de entrenamiento de 15 Epoch

```
Administrador: C:\Windows\System32\cmd.exe - py "C:\inetpub\wwwroot\Red Neuronal\app.py"
9/12 [=====>.....] - ETA: 1s - loss: 1.0694 - tp: 39.0000 - fp: 45.0000 - tn: 45.0000 - fn: 6.0000 -
10/12 [=====>.....] - ETA: 0s - loss: 1.0760 - tp: 42.0000 - fp: 50.0000 - tn: 50.0000 - fn: 8.0000 -
11/12 [=====>.....] - ETA: 0s - loss: 1.0982 - tp: 44.0000 - fp: 57.0000 - tn: 53.0000 - fn: 11.0000
12/12 [=====>.....] - ETA: 0s - loss: 1.0887 - tp: 45.0000 - fp: 57.0000 - tn: 55.0000 - fn: 11.0000
12/12 [=====] - 6s 464ms/step - loss: 1.0887 - tp: 45.0000 - fp: 57.0000 - tn: 55.0000 - fn: 11
.0000 - accuracy: 0.5952 - precision: 0.4412 - recall: 0.8036 - auc: 0.6885 - prc: 0.5118 - val_loss: 1.0415 - val_tp: 7
.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0
.5833 - val_auc: 0.6979 - val_prc: 0.5613
Epoch 15/15
1/12 [=>.....] - ETA: 5s - loss: 1.2953 - tp: 4.0000 - fp: 7.0000 - tn: 3.0000 - fn: 1.0000 - ac
2/12 [====>.....] - ETA: 4s - loss: 1.1109 - tp: 7.0000 - fp: 13.0000 - tn: 7.0000 - fn: 3.0000 - a
3/12 [====>.....] - ETA: 2s - loss: 1.0660 - tp: 8.0000 - fp: 14.0000 - tn: 8.0000 - fn: 3.0000 - a
4/12 [====>.....] - ETA: 2s - loss: 1.1718 - tp: 11.0000 - fp: 21.0000 - tn: 11.0000 - fn: 5.0000 -
5/12 [====>.....] - ETA: 2s - loss: 1.1366 - tp: 15.0000 - fp: 26.0000 - tn: 16.0000 - fn: 6.0000 -
6/12 [====>.....] - ETA: 2s - loss: 1.1649 - tp: 17.0000 - fp: 31.0000 - tn: 21.0000 - fn: 9.0000 -
7/12 [====>.....] - ETA: 1s - loss: 1.1768 - tp: 21.0000 - fp: 37.0000 - tn: 25.0000 - fn: 10.0000
8/12 [====>.....] - ETA: 1s - loss: 1.1440 - tp: 25.0000 - fp: 41.0000 - tn: 31.0000 - fn: 11.0000
9/12 [====>.....] - ETA: 1s - loss: 1.1030 - tp: 30.0000 - fp: 48.0000 - tn: 34.0000 - fn: 11.0000
10/12 [====>.....] - ETA: 0s - loss: 1.0844 - tp: 34.0000 - fp: 52.0000 - tn: 40.0000 - fn: 12.0000
11/12 [====>.....] - ETA: 0s - loss: 1.1165 - tp: 36.0000 - fp: 58.0000 - tn: 44.0000 - fn: 15.0000
12/12 [====>.....] - ETA: 0s - loss: 1.0962 - tp: 41.0000 - fp: 63.0000 - tn: 49.0000 - fn: 15.0000
12/12 [=====] - 6s 476ms/step - loss: 1.0962 - tp: 41.0000 - fp: 63.0000 - tn: 49.0000 - fn: 15
.0000 - accuracy: 0.5357 - precision: 0.3942 - recall: 0.7321 - auc: 0.6351 - prc: 0.4966 - val_loss: 1.0344 - val_tp: 7
.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0
.5833 - val_auc: 0.7361 - val_prc: 0.4975
loss : 1.0572404861450195
tp : 7.0
fp : 8.0
tn : 16.0
fn : 5.0
accuracy : 0.638888895116272
precision : 0.46666666865348816
recall : 0.5833333134651184
auc : 0.7065972089767456
prc : 0.4528294801712036
127.0.0.1 - - [03/Oct/2022 02:37:56] "GET /entrenamiento HTTP/1.1" 200 -
```

Fuente: Elaboración propia

Para el entrenamiento de 15 Epoch se obtuvo una precisión del 46% y una exactitud del 63%.

Figura 13. Prueba de entrenamiento de 30 Epoch

```
Administrador: C:\Windows\System32\cmd.exe - py "C:\inetpub\wwwroot\Red Neuronal\app.py"
9/12 [====>.....] - ETA: 1s - loss: 1.0886 - tp: 28.0000 - fp: 49.0000 - tn: 33.0000 - fn: 13.0000
10/12 [====>.....] - ETA: 0s - loss: 1.0529 - tp: 33.0000 - fp: 53.0000 - tn: 39.0000 - fn: 13.0000
11/12 [====>.....] - ETA: 0s - loss: 1.0597 - tp: 37.0000 - fp: 60.0000 - tn: 42.0000 - fn: 14.0000
12/12 [====>.....] - ETA: 0s - loss: 1.0790 - tp: 40.0000 - fp: 66.0000 - tn: 46.0000 - fn: 16.0000
12/12 [====>.....] - 5s 462ms/step - loss: 1.0790 - tp: 40.0000 - fp: 66.0000 - tn: 46.0000 - fn: 16
.0000 - accuracy: 0.5119 - precision: 0.3774 - recall: 0.7143 - auc: 0.6821 - prc: 0.5557 - val_loss: 1.0351 - val_tp: 7
.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0
.5833 - val_auc: 0.7222 - val_prc: 0.4996
Epoch 30/30
1/12 [=>.....] - ETA: 1s - loss: 0.6850 - tp: 1.0000 - fp: 1.0000 - tn: 1.0000 - fn: 0.0000e+00
2/12 [====>.....] - ETA: 3s - loss: 0.9688 - tp: 5.0000 - fp: 7.0000 - tn: 5.0000 - fn: 1.0000 - ac
3/12 [====>.....] - ETA: 3s - loss: 1.2663 - tp: 6.0000 - fp: 13.0000 - tn: 9.0000 - fn: 5.0000 - a
4/12 [====>.....] - ETA: 3s - loss: 1.0550 - tp: 11.0000 - fp: 16.0000 - tn: 16.0000 - fn: 5.0000 -
5/12 [====>.....] - ETA: 2s - loss: 1.0519 - tp: 16.0000 - fp: 23.0000 - tn: 19.0000 - fn: 5.0000 -
6/12 [====>.....] - ETA: 2s - loss: 1.0424 - tp: 19.0000 - fp: 27.0000 - tn: 25.0000 - fn: 7.0000 -
7/12 [====>.....] - ETA: 2s - loss: 1.0346 - tp: 24.0000 - fp: 32.0000 - tn: 30.0000 - fn: 7.0000 -
8/12 [====>.....] - ETA: 1s - loss: 1.0659 - tp: 29.0000 - fp: 38.0000 - tn: 34.0000 - fn: 7.0000 -
9/12 [====>.....] - ETA: 1s - loss: 1.0588 - tp: 33.0000 - fp: 42.0000 - tn: 40.0000 - fn: 8.0000 -
10/12 [====>.....] - ETA: 0s - loss: 1.0658 - tp: 38.0000 - fp: 48.0000 - tn: 44.0000 - fn: 8.0000 -
11/12 [====>.....] - ETA: 0s - loss: 1.0741 - tp: 42.0000 - fp: 53.0000 - tn: 49.0000 - fn: 9.0000 -
12/12 [====>.....] - ETA: 0s - loss: 1.0835 - tp: 45.0000 - fp: 59.0000 - tn: 53.0000 - fn: 11.0000
12/12 [====>.....] - 5s 489ms/step - loss: 1.0835 - tp: 45.0000 - fp: 59.0000 - tn: 53.0000 - fn: 11
.0000 - accuracy: 0.5833 - precision: 0.4327 - recall: 0.8036 - auc: 0.6949 - prc: 0.5450 - val_loss: 1.0286 - val_tp: 7
.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0
.5833 - val_auc: 0.7535 - val_prc: 0.6258
loss : 1.048167109489441
tp : 7.0
fp : 8.0
tn : 16.0
fn : 5.0
accuracy : 0.6388888955116272
precision : 0.46666666865348816
recall : 0.5833333134651184
auc : 0.6857638955116272
prc : 0.4972619414329529
127.0.0.1 - - [03/Oct/2022 02:54:11] "GET /entrenamiento HTTP/1.1" 200 -
```

Fuente: Elaboración propia

Para el entrenamiento de 30 Epoch se obtuvo una precisión del 46% y una exactitud del 63%.

Figura 14. Prueba de entrenamiento de 60 Epoch

```

Administrator: C:\Windows\System32\cmd.exe - py "C:\inetpub\wwwroot\Red Neuronal\app.py"
9/12 [=====>.....] - ETA: 1s - loss: 1.1148 - tp: 30.0000 - fp: 49.0000 - tn: 33.0000 - fn: 11.0000
10/12 [=====>.....] - ETA: 0s - loss: 1.1439 - tp: 34.0000 - fp: 55.0000 - tn: 37.0000 - fn: 12.0000
11/12 [=====>.....] - ETA: 0s - loss: 1.0891 - tp: 39.0000 - fp: 58.0000 - tn: 44.0000 - fn: 12.0000
12/12 [=====>.....] - ETA: 0s - loss: 1.0768 - tp: 42.0000 - fp: 64.0000 - tn: 48.0000 - fn: 14.0000
12/12 [=====>.....] - 5s 450ms/step - loss: 1.0768 - tp: 42.0000 - fp: 64.0000 - tn: 48.0000 - fn: 14.0000
- accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
Epoch 60/60
1/12 [=>.....] - ETA: 4s - loss: 1.0965 - tp: 3.0000 - fp: 5.0000 - tn: 5.0000 - fn: 2.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
2/12 [====>.....] - ETA: 4s - loss: 1.2612 - tp: 6.0000 - fp: 9.0000 - tn: 11.0000 - fn: 4.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
3/12 [=====>.....] - ETA: 3s - loss: 1.1057 - tp: 10.0000 - fp: 14.0000 - tn: 16.0000 - fn: 5.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
4/12 [=====>.....] - ETA: 3s - loss: 1.1836 - tp: 13.0000 - fp: 19.0000 - tn: 21.0000 - fn: 7.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
5/12 [=====>.....] - ETA: 2s - loss: 1.1634 - tp: 13.0000 - fp: 21.0000 - tn: 21.0000 - fn: 8.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
6/12 [=====>.....] - ETA: 2s - loss: 1.0770 - tp: 17.0000 - fp: 24.0000 - tn: 28.0000 - fn: 9.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
7/12 [=====>.....] - ETA: 1s - loss: 1.0802 - tp: 21.0000 - fp: 28.0000 - tn: 34.0000 - fn: 10.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
8/12 [=====>.....] - ETA: 1s - loss: 1.0227 - tp: 26.0000 - fp: 33.0000 - tn: 39.0000 - fn: 10.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
9/12 [=====>.....] - ETA: 1s - loss: 1.0033 - tp: 29.0000 - fp: 37.0000 - tn: 45.0000 - fn: 12.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
10/12 [=====>.....] - ETA: 0s - loss: 1.0153 - tp: 34.0000 - fp: 46.0000 - tn: 46.0000 - fn: 12.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
11/12 [=====>.....] - ETA: 0s - loss: 1.0479 - tp: 39.0000 - fp: 51.0000 - tn: 51.0000 - fn: 12.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
12/12 [=====>.....] - ETA: 0s - loss: 1.0709 - tp: 41.0000 - fp: 57.0000 - tn: 55.0000 - fn: 15.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
12/12 [=====>.....] - 5s 457ms/step - loss: 1.0709 - tp: 41.0000 - fp: 57.0000 - tn: 55.0000 - fn: 15.0000 - accuracy: 0.5357 - precision: 0.3962 - recall: 0.7500 - auc: 0.6842 - prc: 0.5823 - val_loss: 1.0370 - val_tp: 7.0000 - val_fp: 8.0000 - val_tn: 16.0000 - val_fn: 5.0000 - val_accuracy: 0.6389 - val_precision: 0.4667 - val_recall: 0.5833 - val_auc: 0.6962 - val_prc: 0.5348
loss : 1.0108808279037476
tp : 8.0
fp : 8.0
tn : 16.0
fn : 4.0
accuracy : 0.6666666865348816
precision : 0.5
recall : 0.6666666865348816
auc : 0.7378472089767456
prc : 0.5404316782951355
127.0.0.1 - - [03/Oct/2022 03:32:30] "GET /entrenamiento HTTP/1.1" 200 -

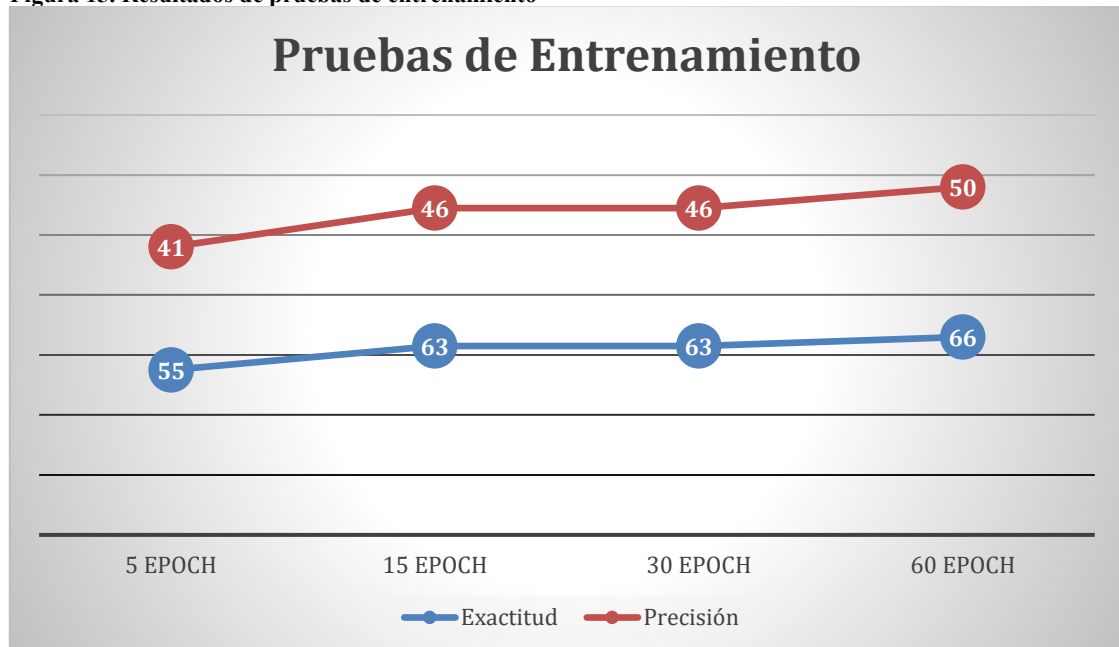
```

Fuente: Elaboración propia

Para el entrenamiento de 60 Epoch se obtuvo una precisión del 50% y una exactitud del 66%.

A continuación, se presenta una gráfica de desempeño de las pruebas de entrenamiento (Ver Figura 13).

Figura 15. Resultados de pruebas de entrenamiento

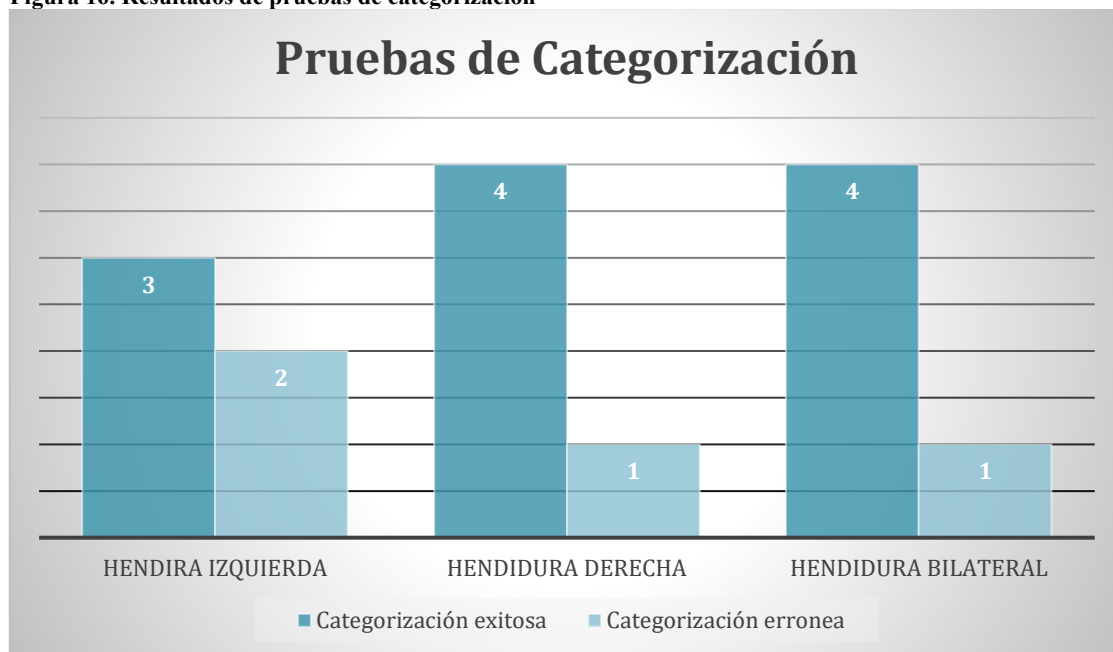


Fuente: Elaboración propia

Se esperaba durante esta fase contar con un mayor volumen de imágenes diagnósticas, pero por parte del grupo de odontólogos no fue posible su consecución. Además, no cuentan con dispositivos de alta gama para la toma de ellas por lo que algunas son tomadas con cámaras de baja resolución o desde un dispositivo móvil. Estas imágenes diagnósticas no ofrecen una calidad en su resolución y se tiene un nivel alto de riesgo de que los resultados no sean los esperados. Para el entrenamiento del modelo era necesario contar con una mayor cantidad de insumos.

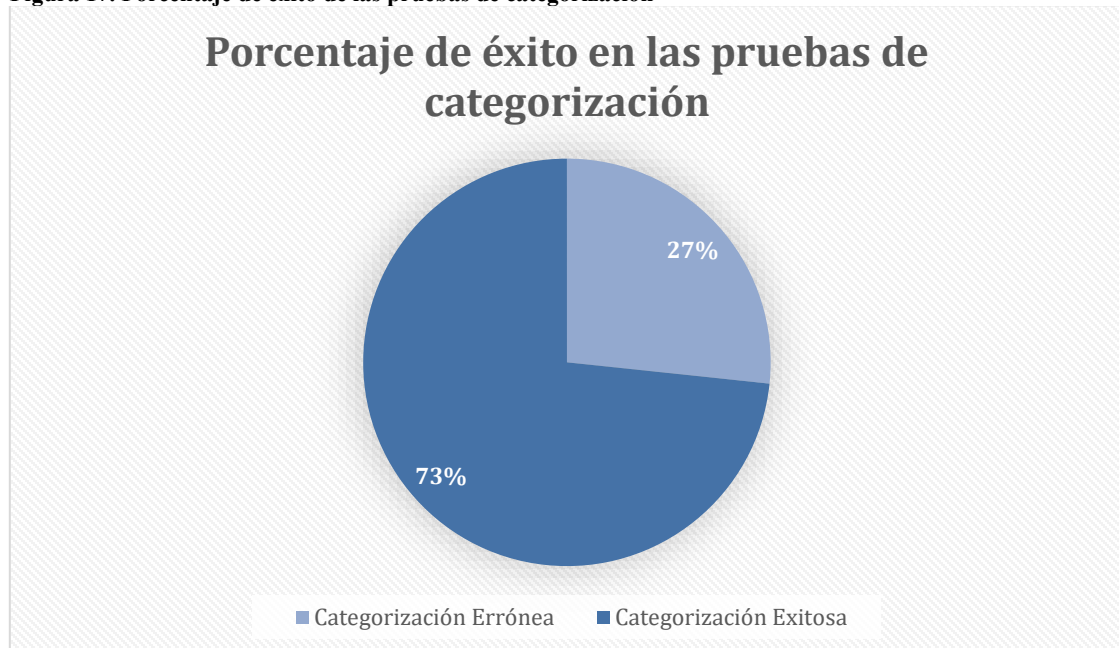
A continuación, se presenta la gráfica de resultados de las pruebas realizadas sobre las imágenes procesadas (Ver Figura 16 y 17).

Figura 16. Resultados de pruebas de categorización



Fuente: Elaboración propia

Figura 17. Porcentaje de éxito de las pruebas de categorización



Fuente: Elaboración propia

FASE DE DESPLIEGUE

En la fase de despliegue se lanzaron los servicios a una máquina virtual EC2 alojada en la plataforma de AWS, siendo esta una opción preliminar y temporal ya que es de pago, sin embargo, anteriormente se utilizó la plataforma de Azure y se realizó la migración a esta por temas de sostenibilidad.

Al estar alojado el servicio en una máquina virtual con IP pública la plataforma presta la disponibilidad en cada momento para ser accedido por el cliente o por el equipo de desarrollo desde cualquier máquina.

Por parte de los beneficiarios de este proyecto, no se cuenta con una infraestructura tecnológica que permita llevar a cabo el despliegue de la aplicación. El despliegue realizado sobre plataformas de proveedores fue fundamental para demostrar el cumplimiento de los objetivos del proyecto.

6. RESULTADOS Y DISCUSIÓN

En primera instancia se ha completado el objetivo específico número 1 (uno) escogiendo el tipo de redes neuronales convolucionales con la estructura de MobileNetV2, estructura que se encuentra con los mejores valores para la clasificación de imágenes con menor tiempo y menor cantidad de datos, una vez definido el negocio se diseñó un software que consumiría el servicio con el modelo, dando fluidez al procesamiento entre tecnologías, el modelo respondió de manera satisfactoria en el caso de prueba con el set de datos obtenido de Kaggle, sin embargo esta cantidad de imágenes superaba las 200 (doscientos) imágenes para un caso de categorización binaria, una vez realizado el despliegue del modelo en Heroku su prueba no logró satisfacer el objetivo específico número 3 (tres) dado que solo obtuvo una precisión del 50% (cincuenta por ciento), no obstante la disponibilidad del servicio y su forma de trabajo es totalmente viable y accesible. Y esto se debe a diversos factores posiblemente, al tratarse de servicios en la nube con licencia open source, es factible que no se tenga la disponibilidad al ciento por ciento y se tienen demoras en las peticiones.

A continuación, se muestran los resultados obtenidos a partir del desarrollo realizado de acuerdo con los recursos disponibles (Ver Tabla 4).

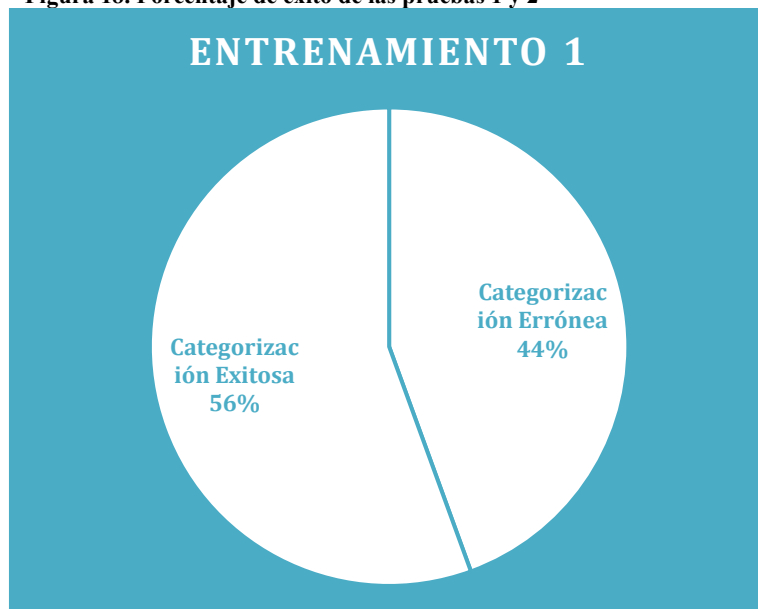
Tabla 4. Tabla de pruebas realizadas posterior a los entrenamientos

Tipo de hendidura Número de prueba	Hendidura izquierda		Hendidura derecha		Hendidura bilateral	
	Exitoso	Erróneo	Exitoso	Erróneo	Exitoso	Erróneo
1 – Entrenamiento 1	1	2	2	1	2	1
2 – Entrenamiento 1	2	1	1	2	2	1
3 – Entrenamiento 2	3	0	2	1	2	1
4 – Entrenamiento 2	2	1	1	2	2	1

Fuente: Elaboración propia

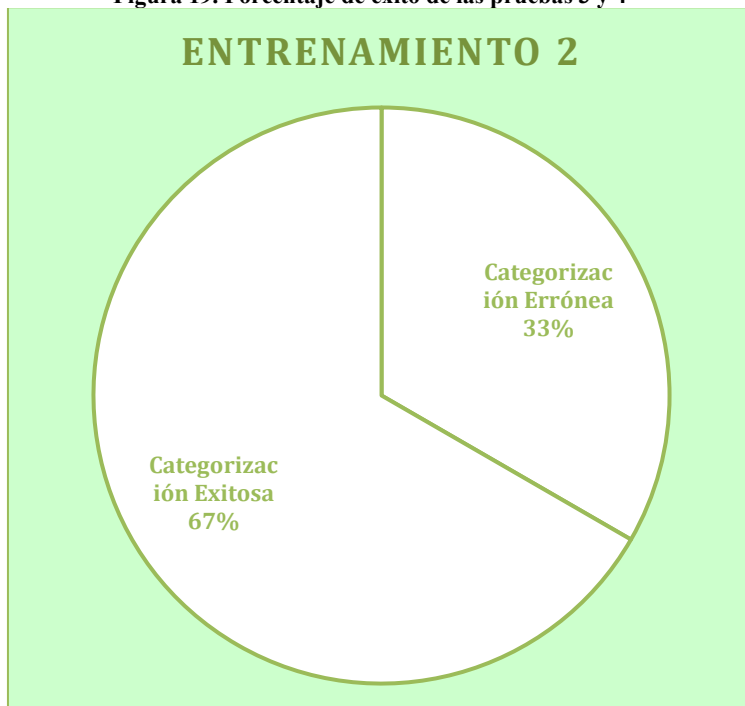
Se realizaron 4 (cuatro) pruebas donde las primeras 2 (dos) se realizaron con las 64 (sesenta y cuatro) imágenes dispuestas primeramente y las 2 (dos) últimas pruebas se realizaron con 10 imágenes más para el entrenamiento (Ver Figura 18 y 19).

Figura 18. Porcentaje de éxito de las pruebas 1 y 2



Fuente: Elaboración propia

Figura 19. Porcentaje de éxito de las pruebas 3 y 4



Fuente: Elaboración propia

La capacidad que tiene el software para clasificar depende de la cantidad de imágenes para su entrenamiento, se espera que en próximos entrenamientos la tasa de precisión aumente considerablemente, dado que se está usando un modelo de red neuronal pre establecido el cual fue modificado para recibir la capa de imágenes en escala de grises se presenta una ventaja a la hora de realizar un entrenamiento con imágenes en cualquier color dado que el modelo las convierte a escala de grises para su procesamiento.

Por otra parte, se tuvo la espera de contar con una mayor cantidad de imágenes, pero no fue posible por parte del equipo beneficiario. Sin embargo, se tuvo la oportunidad de llevar a cabo una serie de pruebas de manera conjunta en donde el personal beneficiario estuvo muy a gusto con los resultados obtenidos y conscientes de la situación desean colocar al servicio de la comunidad -y si es caso público- odontológica la utilización de este producto. Con ello se busca obtener una mayor cantidad de imágenes diagnósticas para obtener una capa de entrenamiento lo más robusta posible; es decir, contar con un mayor nivel de precisión y efectividad para el proceso de clasificación.

A partir de los resultados obtenidos se puede determinar que se podría conocer la categoría o clasificación del caso de LPH (Labio Paladar Hendido) incluso antes de ser remitidas a un radiólogo experimentado agilizando el proceso de análisis y toma de decisiones sobre el tratamiento a seguir. Estos modelos requieren gran variedad de imágenes dada visualización de las imágenes diagnósticas, es decir las similitudes entre los colores de la hendidura y la estructura ósea dificulta de reconocimiento haciendo que la cantidad de imágenes para validar la similitud de píxeles sea alta.

Con base en los resultados obtenidos durante diversas sesiones de trabajo en conjunto con el personal odontológico, se pudo determinar que el modelo de red neuronal implementando cumple en gran medida con lo solicitado en esta primera fase del proyecto: la clasificación de las imágenes. Cada profesional, pudo verificar el resultado obtenido mediante el modelo y los resultados previos obtenidos de manera "manual".

7. CONCLUSIONES

Las redes neuronales convolucionales demuestran ser bastante fuertes en el área de estudio sobre imágenes, específicamente en el tema de machine learning. Sin embargo, se contempló la idea de utilizar redes neuronales

transformers para este proyecto en un punto de revisión de mejora, esta mejora se descartó dado que este tipo de redes necesitan mayor cantidad de datos que las redes neuronales convolucionales, a largo plazo presentan mayor efectividad a la hora de predecir, es decir, acertar según su precisión y exactitud sobre la imagen a clasificar, pero para este proyecto no lo vimos conveniente dada cantidad de datos disponibles.

Para cargar el modelo y ser consumido por un servicio es recomendable realizar la instalación e inicialización del servicio en una máquina dado que se requiere un alto procesamiento de la máquina, aun mas si se habilita el módulo de tensorflow con gpu, al utilizar Heroku como primera opción se logró observar que un servicio en esta plataforma no funciona de manera prolongada por lo que se requiere realizar puntos de control al modelo, ya sea que se almacenen en un repositorio o en el servicio de almacenamiento de Heroku.

El tiempo de clasificación de una imagen se encuentra entre los 5 y 10 segundos de respuesta, es bastante rápido en comparación con la solicitud de un experto que puede tardar días.

De acuerdo al apartado económico se debe tener en cuenta el costo de los recursos a utilizar ya que este no solo afectará el consumo de energía, almacenamiento y equipos si no también el costo para los pacientes.

No se logró obtener un alto porcentaje de precisión en el modelo debido a la poca cantidad de imágenes.

8. LECCIONES APRENDIDAS Y TRABAJO FUTURO

Principalmente se debe establecer mayor compromiso en proyectos futuros, tanto en puntualidad, fechas e interés.

Se ha aprendido a desarrollar un software desde su inicio, es decir, desde su documentación hasta su mantenimiento, además de ser no solo un proyecto de conocimientos ya obtenidos por lo que se requiere un trabajo de investigación para solventar tanto un problema de contexto no habitual para uno como lo es el área odontológica y a nivel técnico el desarrollo de redes neuronales o machine learning dándonos la oportunidad de ser ingenieros íntegros en conocimientos aplicables a cualquier ámbito de la vida profesional.

Se debe realizar una investigación más profunda de las diferentes arquitecturas disponibles en el mercado para no incurrir en gastos adicionales a los previstos.

Como trabajos futuros se tiene que el objetivo a largo plazo del proyecto es que el modelo sea capaz de predecir el tratamiento a seguir para un paciente de LPH. Con la entrega realizada el proyecto podrá continuar evolucionando con base al sistema desarrollado.

Adicionalmente se busca publicar el sistema para que cualquier profesional de salud pueda incluir imágenes ya sea para la clasificación o para el entrenamiento.

9. REFERENCIAS

- [1] M. L. Worley, K. G. Patel, en L. A. Kilpatrick, "Cleft lip and palate", Clin. Perinatol., vol 45, no 4, bll 661-678, Des 2018.
- [2] C. Kuwada et al., "Detection and classification of unilateral cleft alveolus with and without cleft palate on panoramic radiographs using a deep learning system", Scientific Reports, vol 11, no 1, bll 1-10, 2021.
- [3] A. Heliövaara en J. Leikola, "Prediction of orthognathic surgery need in children with unilateral cleft lip palate: Dental arch relationships and 5-year-olds' index", Orthodontics & Craniofacial Research, 2021.
- [4] M. Morandini, T. A. Coleti, E. Oliveira Jr, και P. L. P. Corrêa, 'Considerations about the efficiency and sufficiency of the utilization of the Scrum methodology: A survey for analyzing results for development teams', Comput. Sci. Rev., τ. 39, τχ. 100314, ο. 100314, 2021.
- [5] G. Lin, P.-J. Kim, S.-H. Baek, H.-G. Kim, S.-W. Kim, και J.-H. Chung, 'Early prediction of the need for orthognathic surgery in patients with repaired unilateral cleft lip and palate using machine learning and longitudinal

lateral cephalometric analysis data', J. Craniofac. Surg., τ. 32, τχ. 2, 616–620, 2021.

[6] N. Shafi, F. Bukhari, W. Iqbal, K. M. Almustafa, M. Asif, και Z. Nawaz, 'Cleft prediction before birth using deep neural network', Health Informatics J., τ. 26, τχ. 4, 2568–2585, 2020.

[7] Kuwada, C., Arijji, Y., Kise, Y., Funakoshi, T., Fukuda, M., Kuwada, T., ... & Arijji, E. (2021). Detection and classification of unilateral cleft alveolus with and without cleft palate on panoramic radiographs using a deep learning system. Scientific Reports, 11(1), 1-10.

[8] R. Novak κ.ά., 'Neural Tangents: Fast and easy infinite neural networks in python', arXiv [stat.ML], 2019.

[9] Alvaro Artola Moreno, "Clasificación de imágenes usando redes neuronales convolucionales en Python", 2019.

[10] M. L. Worley, K. G. Patel, και L. A. Kilpatrick, 'Cleft lip and palate', Clin. Perinatol., τ. 45, τχ. 4, 661–678, 2018.

[11] R. Yamashita, M. Nishio, R. K. G. Do, και K. Togashi, 'Convolutional neural networks: an overview and application in radiology', Insights Imaging, τ. 9, τχ. 4, σσ. 611–629, Αυγούστου 2018.

[12] Masud, M., Hossain, M. S., Alhumyani, H., Alshamrani, S. S., Cheikhrouhou, O., Ibrahim, S., ... & Gupta, B. B. (2021). Pre-trained convolutional neural networks for breast cancer detection using ultrasound images. ACM Transactions on Internet Technology (TOIT), 21(4), 1-17.

[13] Zheng, L. et al. (2019) Intensified stiffness and photodynamic provocation in a collagen-based composite hydrogel drive chondrogenesis, Advanced science (Weinheim, Baden-Wurttemberg, Germany). U.S. National Library of Medicine.

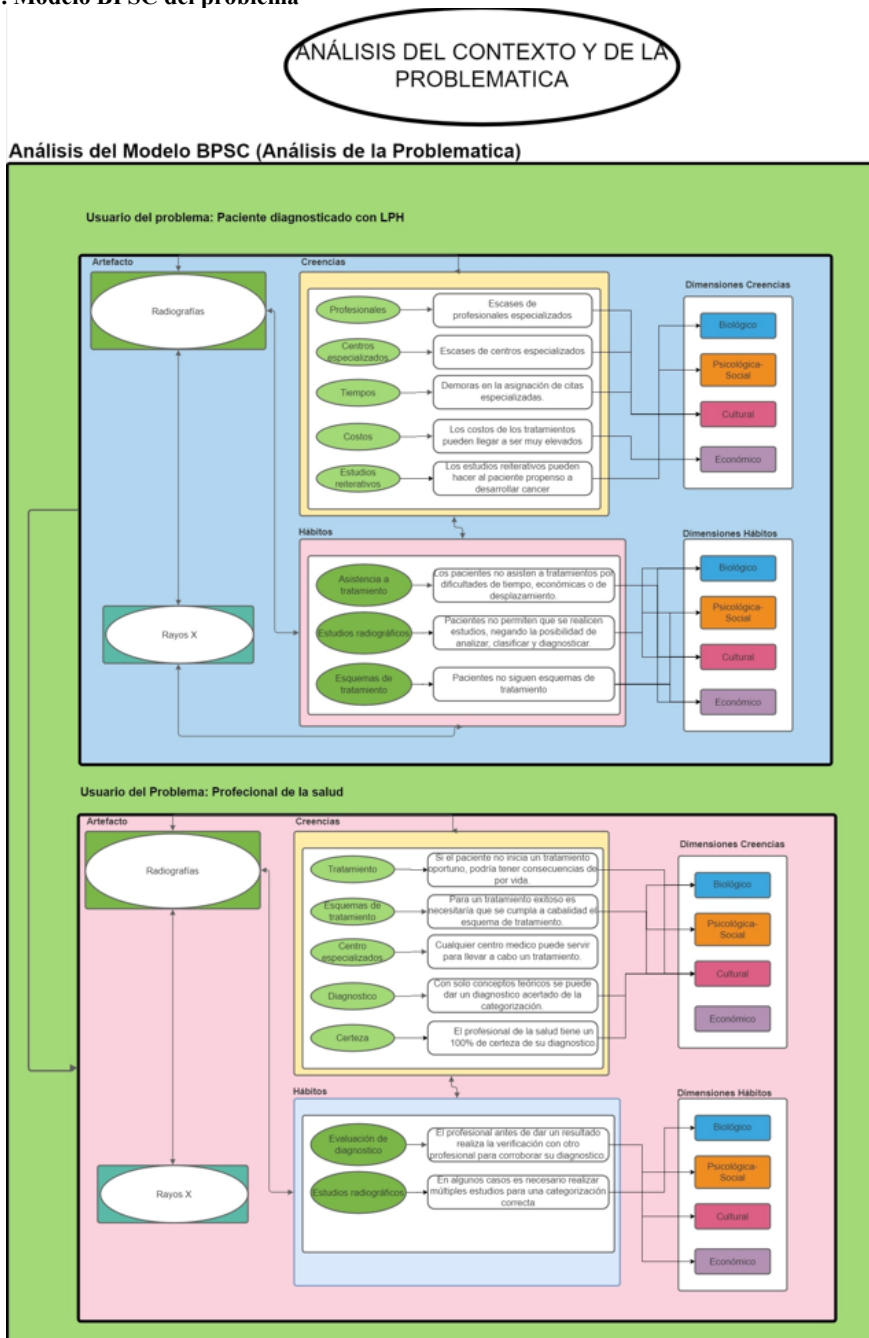
[14] J., Z.Y.Z.F.L.Z.Y. (no date) Current landscape and future directions of biomarkers for predicting responses to immune checkpoint inhibitors, Cancer management and research. U.S. National Library of Medicine.

[15] Cai, G. et al. (2019) A preliminary study showing no association between methylation levels of C3 gene promoter and the risk of CAD, Lipids in health and disease. U.S. National Library of Medicine.

[16] Buckley, T., Cox, C. and Doron, I.(I. (2020) Considering LTSS through the lenses of social construction of target populations: Theory and neoliberalism, Innovation in Aging. Oxford University Press.

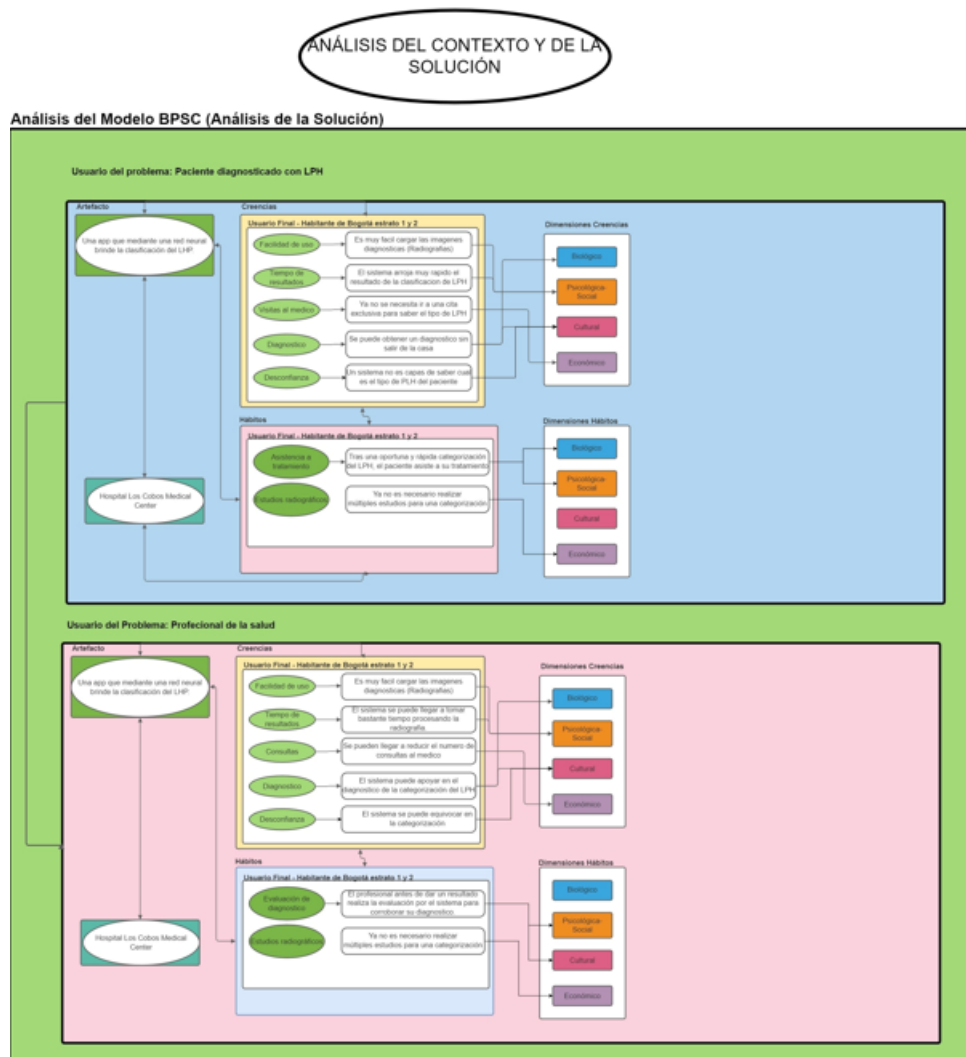
MODELO BPSC DEL PROBLEMA

Figura 20. Modelo BPSC del problema



MODELO BPSC DE LA SOLUCIÓN

Figura 21. Modelo BPSC de la solución



CRONOGRAMA

Diríjase a [Cronograma](#)

REQUERIMIENTOS FUNCIONALES

Diríjase a [Requerimientos funcionales](#)

REQUERIMIENTOS NO FUNCIONALES

Diríjase a [Requerimientos no funcionales](#)

DIAGRAMA DE CASOS DE USO

Diríjase a [Diagrama de casos de uso](#)

HISTORIAS DE USUARIO

Diríjase a [Historias de Usuario](#)

PRODUCT BACKLOG

Diríjase a [Product Backlog](#)

MODELO RELACIONAL DE LA BASE DE DATOS

Diríjase a [Modelo relacional](#)

TABLERO DE CONTROL KANBAN

Diríjase a [Tablero de control](#)

DIAGRAMA DE ARQUITECTURA

Diríjase a [Diagrama de arquitectura](#)

DICCIONARIO DE DATOS

Diríjase a [Diagrama de datos](#)

MANUAL DE USUARIO

Diríjase a [Manual de usuario](#)

MANUAL TÉCNICO

Diríjase a [Manual técnico](#)