Automatización del diagnóstico de cáncer de próstata mediante aprendizaje supervisado.



Camilo Anibal Espinosa
Manuel Alejandro García Montero

Universidad Distrital Francisco José de Caldas
Tecnología en sistematización de datos
Facultad Tecnológica
Bogotá D.C.
Agosto de 2021

Automatización del diagnóstico de cáncer de próstata mediante aprendizaje supervisado.

Camilo Anibal Espinosa
Manuel Alejandro García Montero

Proyecto de grado para optar al título de Tecnólogo de sistematización de datos

Asesor: Hector Florez Fernandez

Universidad Distrital Francisco José de Caldas
Tecnología en sistematización de datos
Facultad Tecnológica
Bogotá D.C.
Agosto de 2021

Nota de aceptación:

_____

_____

_____

_____

_____

Asesor: Hector Florez Fernandez

_____

Jurado: Nelson Reynaldo Becerra Correa

_____

Jurado:

Bogotá, Agosto 28 de 2021

Agradecimientos

Agradecemos a nuestras familias que nos han apoyado durante el desarrollo del proyecto, a nuestro asesor quien nos ha orientado y a nuestros jurados que nos han ofrecido su pertinente retroalimentación

Resumen

Lo riguroso, invasivo, incómodo e ineficaz que pueden llegar a ser los procedimientos actuales del diagnóstico de cáncer de próstata sobre los hombres impide que este importante proceso pueda ser totalmente eficiente, ejemplificando diagnósticos como el PSA que su grado de precisión es bajo y el DRE que depende altamente de la experiencia y sensibilidad del médico que lo aplica. En consecuencia se presenta el proyecto de un sistema de información web por el cual utilizando imágenes de resonancia magnética junto con redes neuronales realiza el diagnóstico sobre el grado cáncer que tiene un paciente sobre su próstata, para ello se implementó el concepto de entrenamiento automatizado de inteligencia artificial entrenando las redes neuronales para que fueran capaces de predecir donde está la próstata en la imagen y decidir respecto al diagnóstico, mostrando el resultado en grado Gleason para que el médico pueda tomar decisiones respecto al paciente.

Abstract

The rigorous, invasive, uncomfortable and ineffective that the current procedures for diagnosing prostate cancer on men can be, prevents this important process from being fully efficient, exemplifying diagnoses such as PSA, which have a low degree of accuracy, and DRE, which is highly dependent on the experience and sensitivity of the doctor who performs it. As a result, we present the project of a web information system by which, using magnetic resonance imaging together with neural networks, diagnoses the degree of cancer that a patient has on his prostate. To this end, we implemented the concept of automated training of artificial intelligence training training training the neural networks so that they were able to predict where the prostate is the image and decide on the Diagnostic, showing the Gleason grade result so that the physician can make decisions about the patient.

Tabla de Contenido

Agradecimientos I

Resumen II

Abstract III

Introducción 1

1. Planeación 2
- 1.1. Planteamiento del Problema . . . 2
- 1.2. Justificación . . . 2
- 1.3. Objetivos . . . 3
  - 1.3.1. Objetivo General . . . 3
  - 1.3.2. Objetivos Específicos . . . 3
- 1.4. Alcances . . . 3
- 1.5. Limitaciones . . . 3
- 1.6. Marco de Referencia . . . 4
  - 1.6.1. Marco Histórico . . . 4
  - 1.6.2. Marco Teórico . . . 5
  - 1.6.3. Marco Metodológico . . . 6
- 1.7. Factibilidad . . . 9
  - 1.7.1. Factibilidad Técnica . . . 9
  - 1.7.2. Factibilidad Operativa . . . 10
  - 1.7.3. Factibilidad Económica . . . 11
  - 1.7.4. Factibilidad Legal . . . 12
- 1.8. Cronograma . . . 12

2. Análisis 14
- 2.1. Asignación de los roles del proyecto . . . 14
- 2.2. Product Backlog . . . 14
- 2.3. Análisis de Requerimientos . . . 15
  - 2.3.1. Requerimientos Funcionales . . . 20
  - 2.3.2. Requerimientos No Funcionales . . . 21
- 2.4. Definición de Actores . . . 22
- 2.5. Historias de usuario . . . 22

3. Diseño **27**
3.1. Diagrama de clases . . . 27
3.2. Diagramas de Componentes . . . 29
3.3. Modelo de red neuronal . . . 30
3.4. Modelo del algoritmo Edgerunner . . . 30
3.5. Modelo de segunda red Neuronal . . . 32

4. Implementación **33**
4.1. Modelo Relacional . . . 33
4.2. Diccionario de Datos . . . 34

5. Desarrollo **36**
5.1. Sprint 1 . . . 36
5.1.1. Desarrollo . . . 36
5.1.2. Resultado . . . 37
5.2. Sprint 2 . . . 37
5.2.1. Desarrollo . . . 38
5.2.2. Resultado . . . 40
5.3. Sprint 3 . . . 40
5.3.1. Desarrollo . . . 40
5.3.2. Resultado . . . 47
5.4. Sprint 4 . . . 48
5.4.1. Desarrollo . . . 48
5.4.2. Resultado . . . 48

6. Pruebas **50**
6.1. Sprint 1 . . . 50
6.2. Sprint 2 . . . 51
6.3. Sprint 3 . . . 51
6.4. Sprint 4 . . . 52

Conclusiones **54**

Bibliografía **55**

Lista de Figuras

1.1. Cronograma 13

2.1. Proceso para la subida de imágenes MRI 16
2.2. Proceso para la predicción de cáncer de próstata parte 1 17
2.3. Proceso para la predicción de cáncer de próstata parte 2 18
2.4. Proceso para la predicción de cáncer de próstata parte 3 19

3.1. Diagrama de Clases parte 1 28
3.2. Diagrama de Clases parte 2 29
3.3. Diagrama de componentes 29
3.4. Modelo de Red Neuronal Convolucional 30
3.5. Modelo del algoritmo EdgeRunner 31
3.6. Modelo de segunda Red Neuronal 32

4.1. Diagrama de Clases 33

5.1. Vista de la aplicación NBIA Data Retriever 38
5.2. Carpeta donde se aloja el set de datos convertido 39
5.3. Visualización de una capa del archivo Nifti cargado al sistema 39
5.4. Dimensiones del set de entrenamiento 40
5.5. Modelo de red neuronal de segmentación 41
5.6. Entrenamiento de red neuronal de segmentación 42
5.7. Predicción de red neuronal de segmentación 43
5.8. Zoom de imagen segmentada 44
5.9. Obtención de borde de imagen segmentada 44
5.10. Gráfica de puntos del centro al borde de la imagen segmentada 45
5.11. Grafica de función obtenida a partir de la gráfica de los puntos 45
5.12. Gráfica de Bins resultantes 46
5.13. Subida de resultados del edge runner 46
5.14. Modelo de la segunda red neuronal 47
5.15. Predicción de la segunda red neuronal 47
5.16. Prototipo de la interfaz gráfica encargada del inicio de sesión 48
5.17. Prototipo de la interfaz gráfica del paciente 49

6.1. visualización de contenido referencial de la investigación 50
6.2. visualización sobre la subida y manejo de imágenes en el entorno de desarrollo 51

6.3. visualización de la segmentación de la próstata 52
6.4. visualización de la clasificación de la imagen según el grado Gleason 52
6.5. Visualización de la interfaz principal desde el usuario paciente 53

Lista de Tablas

1.1. Actividades en Fases de SCRUM 7
1.2. Herramientas de desarrollo 10
1.4. Recursos físicos 10
1.6. Recursos Humanos 10
1.8. Costo de Integrantes del Proyecto 11
1.9. Costo de Recursos 11
1.10. factibilidad Económica Costo Total 12

2.1. Integrantes 14
2.2. Sprints 14
2.3. Requerimientos Funcionales 20
2.4. Requerimientos No Funcionales 21
2.5. Actores 22
2.6. Historias de Usuario 22

4.1. PACIENTE 34
4.2. MEDICO 34
4.3. CITA 35
4.4. TITULO 35
4.5. TITULO-HAS-MEDICO 35
4.6. HISTORIAL MEDICO 35
4.7. ADMINISTRADOR 35
4.8. LOG 35

5.1. Artículos preseleccionados 36
5.2. Artículos Seleccionados 37
5.3. Características de los artículos 37

Introducción

En la actualidad el cáncer de próstata representa una gran problemática a nivel global ya que tiene un alto nivel de mortalidad en los hombres mayores de 60 años, por lo tanto, es importante realizar las pruebas pertinentes para la detección temprana, que nos otorga con el análisis histológico un puntaje de Gleason (G-Score) y con los resultados se pues hacer el tratamiento adecuado para erradicar o disminuir el daño causado por el cáncer.

Las pruebas para la detección del cáncer de próstata en la actualidad constan de procedimientos bastantes invasivos entre las cuales se encuentran examen PSA (Antígeno Prostático especifico) que es la toma de una muestra de sangre la cual no es totalmente fiable por lo cual se hacen otros exámenes como el DRE (Examen transrectal digital) o TRUS (Ecografía Transrectal), exámenes que son evitados por parte de los pacientes precisamente por su incomodidad.

EL manejo de MRI (Imágenes de resonancia magnética) da una nueva alternativa para la detección de cáncer de próstata de una forma no invasiva con el análisis de las imágenes con la inteligencia artificial se pueden lograr algoritmos de los cuales obtener un resultado de cáncer en el individuo.

Capítulo 1

Planeación

1.1. Planteamiento del Problema

El cáncer de próstata se encuentra entre las principales causas de muertes relacionadas con el cáncer entre los hombres en el mundo occidental. Debido a una población cada vez más envejecida, se espera que el número de personas que mueren a causa de la enfermedad continúe aumentando. Pautas actuales para el diagnóstico en pruebas de sangre de antígeno prostático específico (PSA), aunque este método ha demostrado una baja especificidad. La densidad de PSA, obtenida al realizar una prueba de PSA estándar y normalizar el valor por el volumen de la próstata medido por técnicas de imagen, produce un rendimiento superior al PSA estándar.3 Además, los esfuerzos recientes se han centrado en el uso de IRM no invasiva dentro de la computadora asistidos por sistemas para detectar el cáncer de próstata. En particular, el uso de la RM ponderada en T2 ha aumentado a medida que los contornos de próstata, y las lesiones tumorales se distinguen en esta modalidad. Las lesiones de cáncer de próstata tienen más probabilidades de desarrollarse en regiones prostáticas específicas. Con 70-80 % de las lesiones originadas en la zona periférica, 10-20 % en la zona de transición, y solo alrededor del 5 % ubicadas en la zona central. Este hecho resalta la necesidad de un algoritmo de segmentación preciso con énfasis particular en las zonas periféricas que puedan identificar más lesiones que el estado del arte actual. También se requiere con urgencia, ayudando a los dispositivos que sustituyen el proceso de diagnóstico propenso a sesgos. Hoy en día, los médicos elaboran los veredictos utilizando información no determinista recopilada de muestras de sangre e imágenes. Los especialistas escalan casos complicados a comités donde los PIRads se emplean como instrumento de calificación. La automatización completa de la tubería de diagnóstico proporcionaría a los hospitales y clínicas un juez confiable e imparcial, acelerando el diagnóstico y evitando la incómoda y eventualmente perjudicial biopsia.

1.2. Justificación

El cáncer de próstata es uno de los más comunes en hombres de los Estados Unidos después del cáncer de pulmón con una tasa de 19 % en casos nuevos y una tasa de mortalidad de 9 % en 2018

[1]. Ahora, teniendo en cuenta el planteamiento del problema explicado anteriormente y la ventaja de que las imágenes de resonancia magnética brindan una clara visión del órgano a diagnosticar y es un método no invasivo, la solución a proponer consiste en la obtención de las imágenes de resonancia magnética enfocadas en la próstata sobre un paciente para así diagnosticar automáticamente si padece de cáncer o no junto con una medición que describe el grado de agresividad del cáncer.

1.3. Objetivos

1.3.1. Objetivo General

Diseñar un sistema de infracción web que, por medio de imágenes de resonancia magnética enfocadas en la próstata y aplicando redes neuronales, diagnostique automáticamente si tiene o no cáncer junto con su correspondiente grado Gleason.

1.3.2. Objetivos Específicos

- Diseñar un módulo para el ingreso de las imágenes médicas del paciente.
- Diseñar un módulo que permita visualizar el diagnostico resultante.
- Diseñar y entrenar usando las librerías Keras y Tensorflow una red neuronal convolucional que permita segmentar la próstata desde una imagen médica.
- Diseñar una interfaz gráfica que permita al usuario implementar el CAD diseñado.

1.4. Alcances

- Será desarrollado en Python.
- La red neuronal será probada con imágenes sacadas del "The Cancer Imaging Archive (TCIA)"
- Se integra el algoritmo Edgerunner descrito en [2].

1.5. Limitaciones

- El dispositivo debe ser un computador de escritorio.
- El diagnostico solo está ambientado al cáncer de próstata.
- La versión de Python deberá ser mayor de 3.0.

1.6. Marco de Referencia

1.6.1. Marco Histórico

[3] propuso una estimación automática del puntaje Gleason (puntaje G) usando la imagen ponderada MR T2 (T2W) y los mapas de coeficiente de difusión aparente (ADC) para TZ y PZ. Las imágenes requieren preprocesamiento y registro. En el esquema de Fehr, una máquina de vectores de soporte (SVM) utiliza características de textura extraídas de la intensidad en mpMRI y la matriz de coexistencia de nivel de gris (GLCM) que consiste en energía, entropía, correlación, homogeneidad y contraste con rendir una evaluación de puntuación G. Además, el sistema tiene una precisión del 87 %, una sensibilidad del 87 % y una especificidad del 84 %.
[4] implementan un algoritmo de bosque aleatorio (RF) para clasificar las imágenes de microscopía de interferencia de luz (SLIM) de la biopsia de próstata a producir un veredicto de puntuación G. El RF fue entrenado con un microarray de tejidos (TMA) compuesto por biopsias H&E y un histograma de vector de características para determinar qué píxel en el SLIM es un lumen, glándula o estroma con una precisión de 0.82 medida por la curva ROC con su AUC.
[5] introducen un método para explorar exclusivamente la zona periférica de la próstata (PZ). La estrategia utiliza imágenes T2w con CAD (detección asistida por computadora) para nominar las zonas cancerígenas candidatas. Además, el método de Giannini requiere el registro de imágenes para estandarizarlas. La segmentación de la próstata se describió identificando una región rectangular por cada segmento de la mp-MRI, luego el rectángulo se segmentó por la transformación de Hough. La extracción de características desarrolla la intensidad de los mapas ADC y las imágenes T2W y las características farmacocinéticas, un clasificador SVM, que se encarga de discriminar entre los vóxeles normales y tumorales y la reducción de falsos positivos donde se excluyen todas las áreas que tienen una piel cancerosa, el sistema tiene una precisión del 91 %.
[6] propuso un algoritmo de patrón binario lineal multiespectral (MLBP), este maneja imágenes multiespectrales, donde se obtienen las características de la imagen y se analiza la textura para obtener la intensidad, las imágenes se dividen en bloques, cada bloque tiene un vector de a características, cada conjunto de bloques forma un libro de códigos que se clasifican con SVM, este estudio tiene una precisión del 93,7 %
[7] desarrollaron un mecanismo de IA que imita el sistema de datos e informes de imágenes de próstata (PI-RADS v2). El procedimiento utiliza imágenes con ponderación T1 (T1WI), T2WI, imágenes con ponderación por difusión (DWI) e imágenes con contraste dinámico (DCE). Wang analiza las imágenes con una función de base radial (RBF) y ejerce veredictos con un clasificador SVM.
[8] se describe un estudio llamado Focalnet [8] que se enfoca en el diagnóstico y la detección de lesiones en una resonancia magnética pre-registrada para cáncer de próstata con una red neuronal convolucional (CNN) que utiliza el puntaje de Gleason como medida, de tal manera que caracteriza la agresividad de la lesión para ser clasificada. Focalnet utiliza una pérdida de hallazgo mutuo (MFL) que permite identificar las características óptimas en imágenes T2w y ADC para la fase de entrenamiento CNN, la imagen T2w se utilizó para su variación de intensidad. Por lo tanto, Focalnet tiene una precisión de 80.5 % y 79.2 % de sensibilidad.
En otro estudio, [9] implementan un sistema CAD para DW-MRI con el propósito de encontrar el tejido benigno y maligno en la próstata. Emplean la factorización de matriz no negativa (NMF) de la próstata a partir de DW-MRI. También se calculan los valores de ADC para la extracción de características, refinado por un Campo aleatorio de Gauss-Markov (GGMRF), por lo tanto, la función

de distribución acumulativa (CDF) globaliza las características benignas y malignas extraídas para entrenar el clasificador apilado sin codificador de restricción de negatividad (SNCAE), en general el CAD propuesto El sistema tiene una precisión del 100 % con un conjunto de datos53 casos.
Otro sistema CAD es propuesto por [10] que implementa imágenes mp-MR con una puntuación de Gleason para medir el diagnóstico. Las características fueron extraídas por las intensidades de la RM midiendo su concordancia por un coeficiente de correlación intraclase (ICC) y la curva característica de funcionamiento del receptor (ROC) con su área bajo curva (AUC). Utilizan dos algoritmos de regresión logística (LR) para clasificar los conjuntos de características, uno utilizado para clasificar PZ y el otro para detectar regiones cancerosas en TZ. Además, el CAD tiene una precisión del 73 % al 86 % demostrada con AUC.

1.6.2. Marco Teórico

UML

El lenguaje de modelado unificado *Unified Modeling Language UML* [11] consiste en una serie de gráficos, modelos y listas que tienen como objetivo expresar el sistema a desarrollar sobre un proyecto de software.

Como tal UML no es un lenguaje de programación sino un lenguaje que interpreta y muestra que es lo requerido junto con como se ejecuta ese requisito, por lo tanto UML se puede implementar en la mayoria de metodologias de desarrollo de software [11]. Teniendo lo anterior en cuenta UML contiene una gran variedad de diagramas que permiten especificar algo en concreto, siendo las dos bases principales definir la estructura del sistema a la vez que sus interacciones con los usarios como por ejemplo el diagrama de casos de uso, implementado a la hora de querer visualizar que requisitos necesita cada usario del sistema a desarrollar o el diagrama de clases en el caso de la estructuración del sistema utilizado cuando ha de realizarse el diseño de las aplicaciones y se implementa un paradigma orientado a objetos.

Framework

Un framework consiste de un conjunto de programas o componentes que al unirse a un lenguaje de programación base permiten la implementación y ejecución de algo en específico [12].

Aprendizaje Automático

Aprendizaje Automático es método de computación para realizar predicciones precisas,a partir de datos digitalizados de distintos tipos de funtes,el cual depende de la calidad y cantidad de datos utilizados para obtener la presición deseada [13].

Python

Python es un lenguaje de programación orientado a objetos multiplataforma creado en 1991 por Guido van Rossum. Debido a su código legible, su elegante sintaxis y que es código abierto Python es implementado por una gran comunidad de desarrolladores [14].

En el caso de este proyecto el lenguaje Python es implementado en diversas tareas siendo concretamente la programación del sistema de desarrollo web por medio del FrameWork Django y el entrenamiento de las redes neuronales implementadas por la libreria TensorFlow. Django permite la rápida ejecución del proyecto web, implementación de paquetes, librerías e incluso páginas ya establecidas en el Framework e implementar el patrón de arquitectura "Model Template View"(MTV) [15]. Ahora, tensorflow es una librería de código abierto desarrollada por el "Google Brain Team"que permite compilar e implementar el aprendizaje automático de una manera práctica y sencilla [16].

Cáncer de próstata

Teniendo en cuenta los datos de [1] el cáncer de próstata es encontrado en el 19 % de los hombres estadounidenses siendo el tipo de cáncer más frecuente en el país.

MySQL

Es un gestor de bases de datos de codigo abierto,creado en 1995 por MySQL AB y adquirido por Microsystems en 2008 el cual fue comprado por Oracle Coporation en 2010, utilizado para crear y administrar bases de datos basadas en modelos relacionales [17].

1.6.3. Marco Metodológico

La metodología SCRUM es una metodología de adaptación, iterativa, rápida, flexible y eficaz, diseñada para ofrecer un valor significativo de forma rápida en todo el proyecto. SCRUM garantiza transparencia en la comunicación y crea un ambiente de responsabilidad colectiva y de progreso continuo. El marco de SCRUM, tal como se define en [18], está estructurado de tal manera que es compatible con los productos y el desarrollo de servicios en todo tipo de industrias y en cualquier tipo de proyecto, independientemente de su complejidad. Esta metodología, como cualquier otra, tiene un flujo de trabajo que seguir para planificar y ejecutar el proyecto, siendo:

1. Caso de negocio del proyecto: Se define qué es lo que se quiere producir y el product owner es el encargado de hacérselo saber al equipo SCRUM con un lenguaje que entiendan.
2. Declaración de la visión del programa: Se especifica cuáles son los requerimientos del programa definido anteriormente de todo el modelo de negocio.
3. Lista priorizada de pendientes del producto: Se agrupan todos los requerimientos a cumplir por consideración del equipo scrum que puedan cumplir entre un rango de 1 a 6 semanas. Para

esto es necesario el análisis y el diseño de los requerimientos listados. Definiéndose ese grupo de requerimientos como un Sprint.

4. Cronograma de planificación del lanzamiento: Se crea un cronograma en donde se especifica el tiempo que durará cada fase de la metodología, dando un tiempo estimado de lanzamiento o culminación del producto.

5. Lista de pendientes del Sprint: Se realiza una lista de los sprint realizados en la fase 3, al tener ya los Sprint realizados se mandan a ejecutar.

6. Ejecución del Sprint y creación de entregables: Se ejecuta el sprint realizando reuniones diarias con el equipo scrum y el scrum master para ir verificando cómo va la ejecución del sprint. Al finalizar la ejecución del Sprint se crea un entregable para ser verificado por el product owner y para realizar pruebas en busca de errores. El product owner finalmente aprueba el entregable para ser registrado hasta cuando todos terminen.

7. Entregables Aceptados: Se traen los entregables dando por finalizado el producto aceptado por el product owner, aquí se terminaría la metodología.

La Tabla 1.1 presenta las diferentes actividades realizadas con base en las fases de la metodología SCRUM.

Tabla 1.1: Actividades en Fases de SCRUM

Fase	**Proceso**	**Descripción**
Inicio	Creación de la visión del proyecto	Generación de la visión del proyecto a partir del caso de negocio presentado por el propietario del proyecto
	Identificación del Scrum Master	Contratación del Scrum master
	Formación de equipos Scrum	Contratación del equipo desarrollador del proyecto.
	Desarrollo de épicas	Desarrollar la descripción de los requisitos funcionales por parte del propietario del proyecto descritas en una o dos frases.
	Realizar la planificación del lanzamiento	Se realiza el cronograma de actividades, dando una fecha estimada para la finalización del proyecto. También se determina la duración estimada de un Sprint.

Planificación y estimación	Creación de historias de usuario	Se describen detalladamente los requisitos funcionales por parte del dueño del producto junto con las especificaciones que permiten aprobar el cumplimiento de tal requisito.
	Aprobación y asignación de las historias de usuario	Al ser aprobadas las historias de usuarios por el dueño del producto pasan a ser analizadas por el Scrum Master y el Equipo Scrum para estimar su esfuerzo requerido terminando en la elaboración de un Sprint.
	Creación de tareas	Se planifican las tareas específicas de la historia de usuario para el equipo Scrum.
	Estimación de tareas	Se especifican el esfuerzo necesario para realizar cada tarea.
	Creación de la lista de pendientes del sprint	Realización de una lista donde contiene las tareas pendientes del Sprint.
Implementación	Creación de entregables	Se trabajan en las tareas de la lista de pendientes para realizar los entregables, normalmente se implementa un tablero Scrum para el seguimiento del trabajo.
	Realizar reunión diaria de pie	Reunión a una hora fija donde todo el equipo Scrum se actualizan de como van las tareas en desarrollo
	Realizar reunión diaria de pie	Reunión a una hora fija donde todo el equipo Scrum se actualizan de como van las tareas en desarrollo
	Mantenimiento de la lista priorizada de pendientes del producto	Actualización de la lista priorizada de pendientes del producto según los entregables realizados.
Revisión y retrospectiva	Convocar el Scrum de Scrums	Llegado el caso hayan varios grupos Scrum y sea necesario se convoca a un Scrum de Scrums para realizar seguimiento a todos los equipos.

	Demostración y validación del sprint	Se presenta el entregable ante el dueño del producto junto con sus asociados para su aprobación.
	Retrospectiva de Sprint	El Scrum Master junto con el equipo Scrum se reúnen para discutir lo aprendido en la elaboración del Sprint para documentarlo llegado el caso sea necesario en futuros Sprints.
Lanzamiento	Envío de entregables	Al ser aceptado los entregables estos son entregados a los socios correspondientes o al dueño del producto dando con ello la finalización del Sprint.
	Retrospectiva del proyecto	Se presenta el entregable ante el dueño del producto junto con sus asociados para su aprobación.
	Retrospectiva de Sprint	Ultimo paso para la finalización del proyecto, consiste en reunirse con todos los entes principales (Equipo Scrum, Scrum Master, etc.) para realizar una retrospectiva del proyecto junto con recomendaciones o lecciones aprendidas para tener en cuenta en futuros proyectos.

1.7. Factibilidad

1.7.1. Factibilidad Técnica

La aplicación se desarrolla para computadores de escritorio que deben contar con recursos suficientes para que la aplicación se pueda desarrollar, probar y ejecutar, cuando se refiere a recursos se trata de procesador, pantalla y memoria. En cuanto a la codificación del software, se hace necesarios computadores con suficiente memoria RAM y procesador para la fase de entrenamiento de la inteligencia artificial. En la tabla 1.2 se evidencian las herramientas de desarrollo para la ejecución del proyecto.

Tabla 1.2: Herramientas de desarrollo

Herramienta	**Descripción**
Jupyter Notebook	Herramienta basada en Python que sirve para desarrollar machine learning.
DCM2NII	Herramienta que permite la conversión de imágenes DICOM a imágenes NIFTI.
TCIA	Base de datos de donde se extraen las imágenes médicas referentes al cáncer.

En la tabla 1.4 se encuentran las especificaciones de los recursos físicos con los que cuentan los equipos para el desarrollo.

Tabla 1.4: Recursos físicos

Recurso	**Características**
Computador	Quad-core Intel o AMD, 3.0 GHz o mayor 16GB DDR4 de espacio Disco Duro solido: 500GB Tarjeta gráfica NVIDIA GTX 1660

En la tabla 1.6 se especifican los recursos humanos que se necesitaran para el desarrollo del proyecto.

Tabla 1.6: Recursos Humanos

Recurso humano	**Características**
Programador	Dos tecnólogos o estudiantes de sistematización de datos, en el caso de ser estudiantes deben estar cursando 5 o 6 semestre académico, con conocimientos en inteligencia artificial, concretamente redes neuronales. Manejo del lenguaje Python junto con el entorno jupyter notebook.
Tutor	Tutor cuya profesión debe estar relacionada en el campo de tecnología o información, mínimo 5 años de experiencia desempeñando el cargo afín. Debe contar con habilidades en lo referente a enseñanza y liderazgo.

1.7.2. Factibilidad Operativa

La aplicación a desarrollar será de gran ayuda para la comunidad médica, debido a que facilitaría la metodología del diagnostico del cáncer de próstata en eficiencia y eficacia tanto para el médico como el paciente. Por lo tanto, los hospitales o centros médicos basándose en el grado de precisión del proyecto lo implementarán. Agregando que la implementación del proyecto incita el constante diagnóstico al paciente, debido a que tendrá que someterse a un control rutinario para evitar un diagnostico tardío.

1.7.3. Factibilidad Económica

La factibilidad económica del proyecto es alta, ya que lo que necesitamos en términos financieros son mínimo dos equipos de trabajo, asesorías de los tutores del proyecto, acceso a Internet. En las tablas que se presentaran a continuación se describe la factibilidad económica, identificando los costos de papelería, hardware, software y recursos humanos necesarios para la realización del proyecto de investigación que se propone. Se dividió en tres aspectos, recursos humanos, recursos técnicos y otros recursos, la distinción de los recursos humanos se presenta en la Tabla 1.8 la Factibilidad de Recursos Humanos.

Tabla 1.8: Costo de Integrantes del Proyecto

Integrante	**descripción**	**Valor-Hora**	**Cantidad**	**Total**
Tutor 1	Asesorías para la realización del proyecto, referente al desarrollo del artículo.	$40.000	2 horas semanales	$2.240.000
Tutor 2	Asesorías para la realización del proyecto, referente al desarrollo práctico y teórico.	$40.000	2 horas semanales	$2.240.000
Desarrolladores	Dos programadores que realicen la implementación de la solución.	$30.000	16 horas semanales	$13.440.000
Total				**$17.920.000**

Aquí se presenta las asesorías que se tendrán y los gastos de los desarrolladores. A continuación, en la Tabla 1.9 se presentarán los gastos de los recursos que se ostentan en el desarrollo del proyecto.

Tabla 1.9: Costo de Recursos

Recurso	**Descripción**	**Valor Unitario**	**Cantidad**	**Total**
Computadores	Equipos de escritorio para el desarrollo y las pruebas del sistema.	$2.500.000	2	$5.000.000
Total				**$5.000.000**

Adicionalmente se muestran los gastos adicionales en la Tabla 1.10 que serán solventados por los desarrolladores del proyecto.

Tabla 1.10: factibilidad Económica Costo Total

Recurso	**Valor**
Total Recursos Humanos	$17.920.000
Total Recursos Técnicos	$5.000.000
Total Otros recursos	$50.000
Costos imprevistos (10 %)	$1.792.000
Total	**$24.762.000**

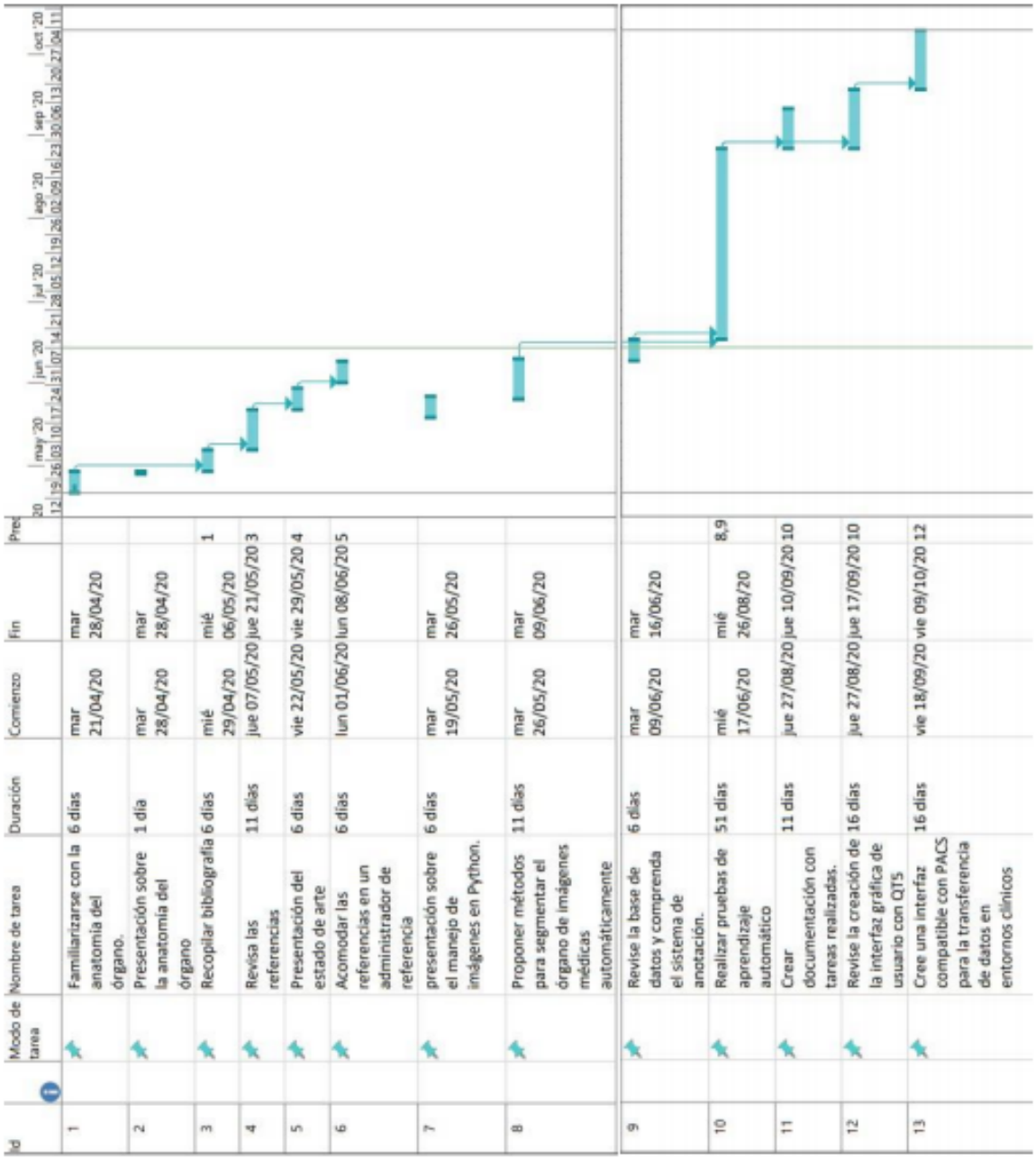
1.7.4. Factibilidad Legal

Las herramientas que utilizaremos a lo largo del proyecto son de software libre y las imágenes médicas extraídas del TCIA son de acceso público.

1.8. Cronograma

La Figura 1.1 presenta el cronograma.

Figura 1.1: Cronograma



Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Pred
1		Familiarizarse con la anatomía del órgano.	6 días	mar 21/04/20	mar 28/04/20	
2		Presentación sobre la anatomía del órgano	1 día	mar 28/04/20	mar 28/04/20	
3		Recopilar bibliografía	6 días	mié 29/04/20	mié 06/05/20	1
4		Revisa las referencias	11 días	jue 07/05/20	jue 21/05/20	3
5		Presentación del estado de arte	6 días	vie 22/05/20	vie 29/05/20	4
6		Acomodar las referencias en un administrador de referencia	6 días	lun 01/06/20	lun 08/06/20	5
7		presentación sobre el manejo de imágenes en Python.	6 días	mar 19/05/20	mar 26/05/20	
8		Proponer métodos para segmentar el órgano de imágenes médicas automáticamente	11 días	mar 26/05/20	mar 09/06/20	
9		Revise la base de datos y comprenda el sistema de anotación.	6 días	mar 09/06/20	mar 16/06/20	
10		Realizar pruebas de aprendizaje automático	51 días	mié 17/06/20	mié 26/08/20	8,9
11		Crear documentación con tareas realizadas.	11 días	jue 27/08/20	jue 10/09/20	10
12		Revise la creación de la interfaz gráfica de usuario con QTS	16 días	jue 27/08/20	jue 17/09/20	10
13		Cree una interfaz compatible con PACS para la transferencia de datos en entornos clínicos	16 días	vie 18/09/20	vie 09/10/20	12

Capítulo 2

Análisis

2.1. Asignación de los roles del proyecto

A continuación se describen los integrantes del proyecto junto con su rol espécifico, ver Tabla 2.1.

Tabla 2.1: Integrantes

Roles	**Descripción**
Scrum Master	Hector Arturo Florez Fernandez
Product Owner	Manuel Alejandro García Montero
Scrum Team	Camilo Anibal Espinosa

2.2. Product Backlog

El siguiente Product Backlog ver Tabla. 2.2 refleja cada elemento considerado en el desarrollo de este proyecto, mostrando cada elemento como una prioridad, con el responsable de su implementación y sus módulos asociados. Esto generará historias de usuarios coincidentes.

Tabla 2.2: Sprints

Sprint	**Nombre**	**Responsable**	**Modulo**	**Prioridad**
1	Recopilación de Información	Camilo	Planeación	Alta
2	Manejo de imágenes MRI	Manuel	Todas	Alta
3	Clasificación automatizada sobre las imágenes MRI	Camilo	Todas	Alta
4	Desarrollo de la interfaz	Manuel	Todas	Media

2.3. Análisis de Requerimientos

Este proyecto permite que tanto los pacientes como sus respectivos médicos de un determinado hospital o ente médico puedan obtener el diagnóstico sobre el índice de cancerosidad contenido en la próstata del paciente, permitiendo al médico tomar decisiones según el resultado.

La interacción de los usuarios con los sistemas construidos se presenta en las Figuras 2.1 y 2.4, las cuales describen los siguientes pasos. Para el proceso de subida de imágenes MRI:

1. El paciente inicia sesión en el sistema para confirmar su rol en la entidad.
2. El paciente selecciona el menú que corresponde al proceso de subir la imagen.
3. El sistema al obtener la imagen por parte del paciente se encarga de verificar si es válida la imagen para luego almacenarla en la base de datos.
4. El sistema notifica al paciente que la subida de su imagen fue exitosa.

Para el proceso de predicción de cáncer de próstata sobre la imagen:

1. El médico inicia sesión en el sistema para confirmar su rol en la entidad.
2. El sistema muestra el menú correspondiente.
3. El médico solicita al sistema la lista de sus pacientes, luego procede a buscar uno en específico.
4. El sistema al encontrar el paciente muestra sus datos junto con una lista de las imágenes MRI que ha subido el paciente para poder seleccionar alguna.
5. El médico selecciona una imagen para solicitar la predicción sobre ella.
6. El sistema realiza la predicción finalizando en el reporte que será guardado en la base de datos y mostrado al médico.

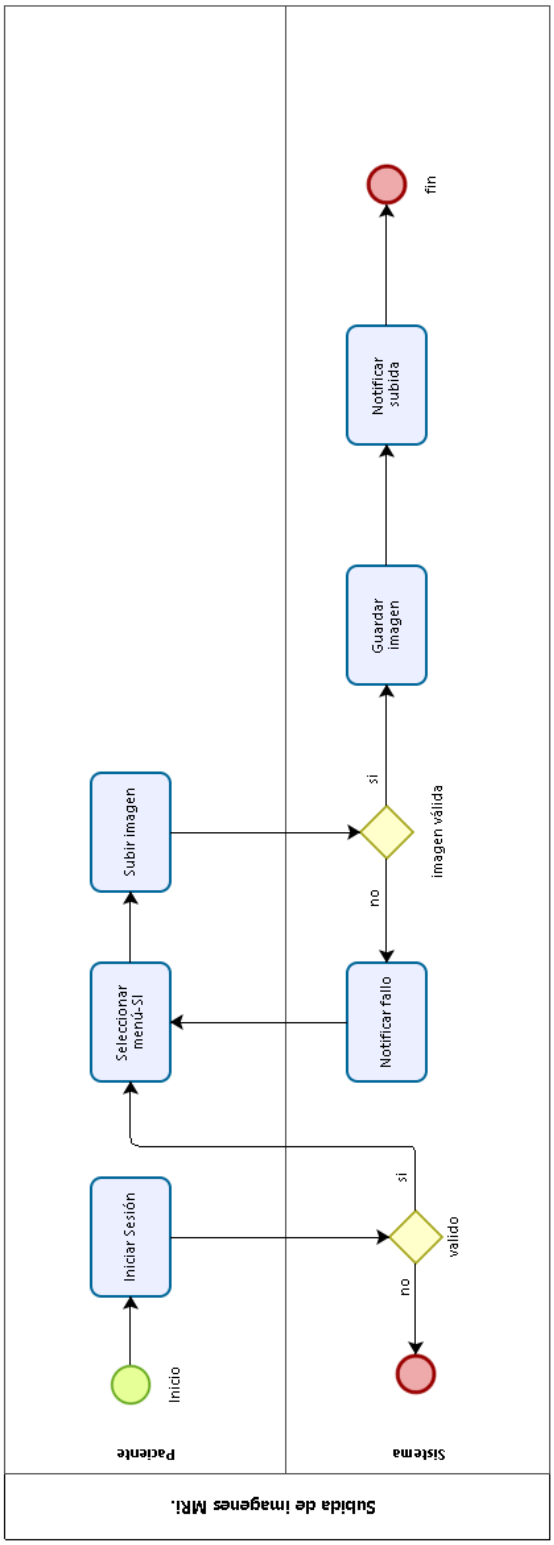


Figura 2.1: Proceso para la subida de imágenes MRI

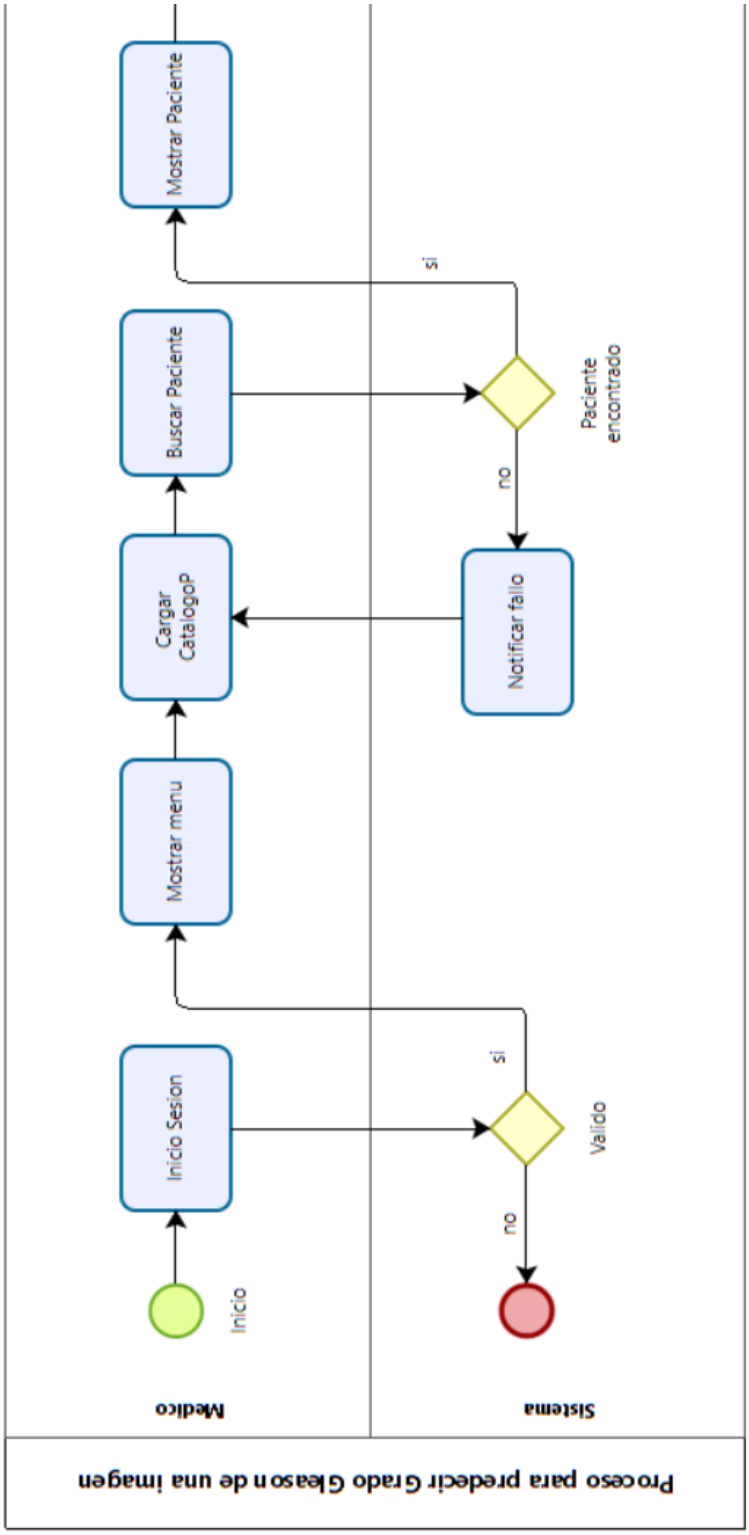


Figura 2.2: Proceso para la predicción de cáncer de próstata parte 1

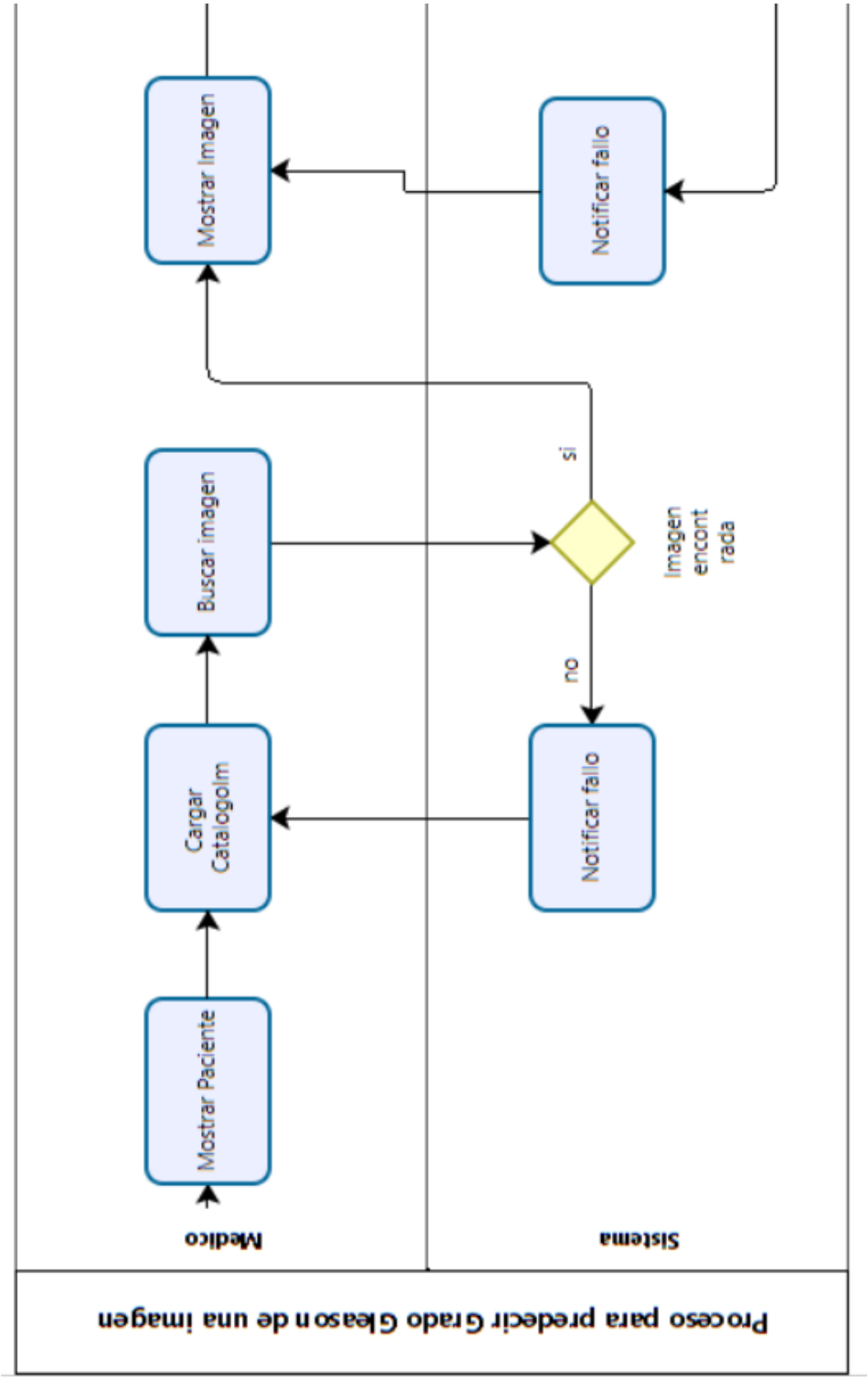


Figura 2.3: Proceso para la predicción de cáncer de próstata parte 2

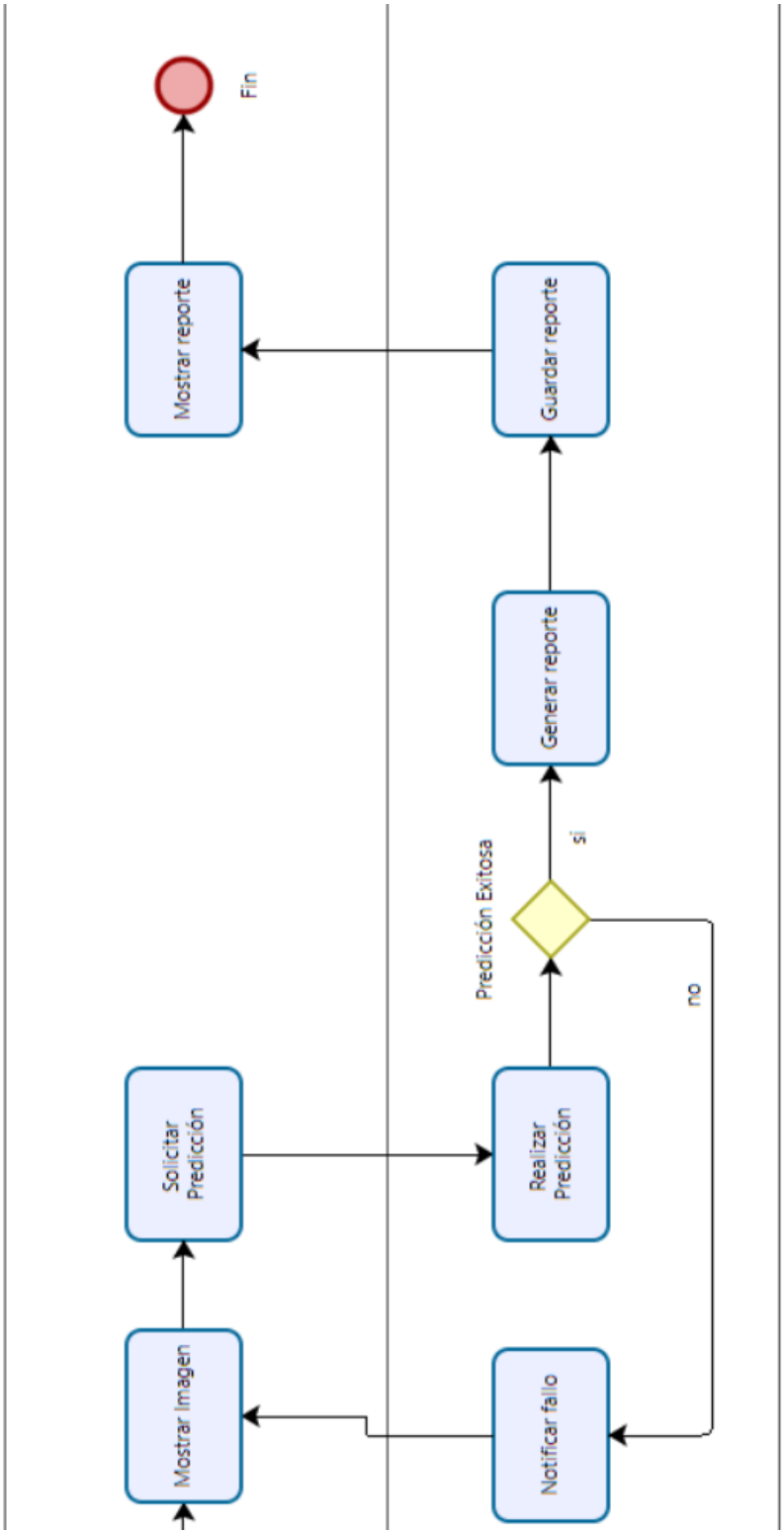


Figura 2.4: Proceso para la predicción de cáncer de próstata parte 3

2.3.1. Requerimientos Funcionales

Los requerimientos funcionales describen los servicios que debe proveer el sistema para determinada entidad hospitalaria. La Tabla 2.3 presenta los requerimientos funcionales del sistema.

Tabla 2.3: Requerimientos Funcionales

ID	**RF 01**
Nombre	Inicio de Sesión
Descripción	Cualquier usuario debe entrar al módulo que les permita entrar al sistema según su rol.
Actores	Médico, Paciente.
Observaciones	Se debe ingresar según su correo electrónico y contraseña.

ID	**RF 02**
Nombre	Subida de imágenes MRI
Descripción	El paciente debe ser capaz de subir su imagen MRi correspondiente al sistema. La imagen quedará guardada en un alojamiento respectivo y en la base de datos quedará registrada su dirección de alojamiento.
Actores	Paciente
Observaciones	El paciente deberá buscar el menú que corresponde a la subida de imagen. La imagen guardada debe quedar referenciada por el id del paciente que la subió. Se debe notificar al paciente que la imagen fue subida exitosamente. Se deberá avisar al paciente sobre asegurarse que la imagen a subir sea de su próstata.

ID	**RF 03**
Nombre	Selección de imagen MRI
Descripción	El médico debe poder seleccionar al paciente que desee para obtener su imagen MRI. El médico debe poder seleccionar cualquier imagen subida por el paciente seleccionado.
Actores	Médico
Observaciones	El médico solo podrá seleccionar pacientes que fueron asignados a él. Al seleccionar la imagen debe mostrarse un Slice de la imagen más los datos correspondientes: Fecha de subida y tipo de toma.

ID	**RF 04**
Nombre	Registro de usuarios
Descripción	El administrador registrará a un paciente o médico que solicite su entrada a la entidad.

Actores	Administrador
Observaciones	El administrador solo podrá registrar personas que la empresa haya solicitado. Los datos a ingresar del usuario nuevo serán los correspondientes a su rol y campos en la base de datos.

ID	**RF 05**
Nombre	Predicción de cáncer de próstata sobre imagen MRI
Descripción	El médico al tener la imagen seleccionada solicita al sistema que realice la predicción de cáncer de próstata. Al realizar la predicción se genera un reporte el cual es guardado y mostrado al médico.
Actores	Médico
Observaciones	El reporte debe ser guardado en la base de datos teniendo de referencia el id del médico, imagen y paciente correspondiente.

2.3.2. Requerimientos No Funcionales

Los requerimientos no funcionales describen diferentes características adicionales que el proyecto debe satisfacer para poder llevar a cabo los servicios que provee. La Tabla 2.4 presenta los requerimientos no funcionales.

Tabla 2.4: Requerimientos No Funcionales

ID	**RNF 01**
Nombre	Conexión a las redes neuronales
Descripción	El sistema debe estar conectado a un equipo remoto encargado de realizar las predicciones solicitadas por los médicos.
Observaciones	Servidor de 2 núcleos de CPU, 8GB RAM disponible, almacenamiento de 60GB SAN, ancho de banda de 2 TB, 2 IPs disponibles

ID	**RNF 02**
Nombre	Navegadores
Descripción	El sistema debe estar conectado a internet y los navegadores actualizados.
Observaciones	Navegador Mozilla Firefox versión 3.6 o superior, navegador Internet Explorer versión 8.0 o superior, navegador Chrome Version 36 o superior y velocidad de internet desde 2MB

2.4. Definición de Actores

Con base en el análisis de requerimientos realizado, se establecen los actores que se presentan en la Tabla 2.5.

Tabla 2.5: Actores

Nombre	**Médico**
Descripción	Actor encargado de solicitar pacientes, sus respectivas imágenes y realizar la predicción sobre tal imagen.
Atributos	Nombre, apellido, identificación, correo, contraseña, fecha de vinculación, títulos.

Nombre	**Administrador**
Descripción	Actor encargado del registro de pacientes o médicos al sistema.
Atributos	Nombre, apellido, identificación, correo, contraseña, fecha de vinculación

Nombre	**Paciente**
Descripción	Actor encargado de la subida de sus respectivas imágenes y visualización de su información.
Atributos	Nombre, apellido, identificación, correo, contraseña, fecha de vinculación, respectivas_imagenes.

2.5. Historias de usuario

A continuación se definen las historias de usuario para el desarrollo del proyecto, ver Tabla 2.6.

Tabla 2.6: Historias de Usuario

ID	**HU01**
Nombre	**Carga de imágenes médicas**
Prioridad	Alta
Riesgo	8

Descripción	Como paciente quiero subir las imágenes MRi de mi próstata.
Validación	1. Las imágenes deben guardarse en un sitio asequible para el sistema sin ser el servidor.
	2. Las imágenes deben estar ligadas al ID del paciente correspondiente.
	3. Las imágenes deben estar asociadas al paciente correspondiente dentro de la base de datos.

ID	**HU02**
Nombre	**Implementación de la predicción sobre la imagen**
Prioridad	Alta
Riesgo	10
Descripción	Como médico quiero realizar la predicción sobre si la próstata escogida tiene cáncer.
Validación	1. La predicción debe mostrarse sobre un reporte en pdf el cual contendrá los datos del paciente, el médico asignado, un slice de la imagen utilizada y el resultado de la predicción.
	2. Tal reporte debe guardarse en la base de datos.
	3. El pdf deberá nombrarse de la forma **ApellidoNombreDD/MM/AAAA** siendo la fecha correspondiente a la generación del reporte. Ej: **GarciaManuel15/09/2017.pdf**

ID	**HU03**
Nombre	**Visualización de los reportes médicos**
Prioridad	Media
Riesgo	7
Descripción	Como médico quiero visualizar los reportes generados hacia un paciente en específico asignado a mi.
Validación	1. El módulo debe generar un buscador donde se pueda seleccionar un paciente entre los asignados al médico.
	2. El médico no puede evidenciar pacientes que no fueron asignados a él.
	3. El módulo debe generar una lista para escoger entre los varios reportes sobre un paciente específico.
	4. Debe visualizarse el pdf seleccionado.

ID	**HU04**
Nombre	**Visualización de las imágenes subidas**
Prioridad	Media
Riesgo	7
Descripción	Como paciente quiero visualizar las imágenes que he subido al sistema.
Validación	1. El módulo debe generar un buscador donde se pueda seleccionar la imagen correspondiente
	2. Se deberán visualizar todos los slices de la imagen.

ID	**HU05**
Nombre	**Inicio de sesión**

Prioridad	Alta
Riesgo	8
Descripción	Como paciente o médico quiero una interfaz que me permita ingresar a mi página correspondiente.
Validación	1. El sistema pedirá correo y contraseña para validar su rol y pertenencia.
	2. Si se ingresan datos erróneos arrojar un mensaje de correo o contraseña inválidos.

ID	**HU06**
Nombre	**Diseño de la página del paciente**
Prioridad	Alta
Riesgo	10
Descripción	Como paciente quiero una interfaz donde pueda acceder a toda la información y servicios correspondientes.
Validación	1. La interfaz deberá tener un menú con el cual podrá acceder a sus servicios.
	2. Se deberán mostrar los logos de las organizaciones participantes en el proyecto.
	3. La interfaz deberá mostrar los datos del paciente siendo nombre, apellido, correo y fecha de vinculación.

ID	**HU07**
Nombre	**Diseño de la página del médico**
Prioridad	Alta
Riesgo	10
Descripción	Como médico quiero una interfaz donde pueda acceder a toda la información y servicios correspondientes.
Validación	1. La interfaz deberá tener un menú con el cual podrá acceder a sus servicios.
	2. Se deberán mostrar los logos de las organizaciones participantes en el proyecto.
	3. La interfaz deberá mostrar los datos del paciente siendo nombre, apellido, correo y fecha de vinculación.

ID	**HU08**
Nombre	**Diseño de la página del administrador**
Prioridad	Alta
Riesgo	10
Descripción	Como administrador quiero una interfaz donde pueda acceder a toda la información y servicios correspondientes.

Validación	1. La interfaz deberá tener un menú con el cual podrá acceder a sus servicios.
	2. Se deberán mostrar los logos de las organizaciones participantes en el proyecto.
	3. La interfaz deberá mostrar los datos del administrador siendo nombre, apellido, correo y fecha de vinculación.

ID	**HU09**
Nombre	**Diseño del menú paciente**
Prioridad	Alta
Riesgo	10
Descripción	Como paciente deseo tener un menú por el cual acceder a todos mis servicios.
Validación	1. Deberá incluir la redirección hacia todas las páginas que prestan los servicios requeridos del paciente.
	2. Deberá estar presente en todas las páginas del paciente como una barra de navegación.
	3. Deberá incluir una sección en la parte superior derecha donde pueda cerrar sesión.

ID	**HU10**
Nombre	**Diseño del menú médico**
Prioridad	Alta
Riesgo	10
Descripción	Como médico deseo tener un menú por el cual acceder a todos mis servicios.
Validación	1. Deberá incluir la redirección hacia todas las páginas que prestan los servicios requeridos del médico.
	2. Deberá estar presente en todas las páginas del paciente como una barra de navegación.
	3. Deberá incluir una sección en la parte superior derecha donde pueda cerrar sesión.

ID	**HU11**
Nombre	**Diseño del menú administrador**
Prioridad	Alta
Riesgo	10
Descripción	Como administrador deseo tener un menú por el cual acceder a todos mis servicios.
Validación	1. Deberá incluir la redirección hacia todas las páginas que prestan los servicios requeridos del administrador.
	2. Deberá estar presente en todas las páginas del administrador como una barra de navegación.
	3. Deberá incluir una sección en la parte superior derecha donde pueda cerrar sesión.

ID	**HU12**

Nombre	**Crear la base de datos**
Prioridad	Alta
Riesgo	10
Descripción	Como cualquier usuario quiero almacenar mis datos necesarios en el sistema.
Validación	1. Deberá ser creada en MySQL
	2. Deberá estar normalizada y definirse acorde a los servicios de los usuarios.

Capítulo 3

Diseño

3.1. Diagrama de clases

Teniendo en cuenta lo descrito anteriormente, el diagrama de clases se puede observar en la Fig. 3.1 y Fig. 3.2. A continuación describiremos cada una de las clases:

- Paciente: Contiene los atributos del usuario paciente así como los métodos para la manipulación de su información.
- Medico: Contiene los atributos del médico junto con los métodos para la manipulación de su información.
- Titulos: Contiene los atributos de los títulos profesionales de los médicos junto con los métodos para la manipulación de su información.
- Log: Contiene los atributos del registro de inicio de sesión de los usuarios junto con los métodos para la manipulación de su información.
- Historial_Medico: Contiene los atributos del historial médico de determinado paciente así como los métodos para la manipulación de su información.
- Cita: Contiene los atributos de la cita entre un paciente y su médico asignado, incluye a la vez métodos para la manipulación de información de la cita.
- Vista_paciente: Contiene los atributos de la cita junto con los datos del paciente y el médico para la implementación de los métodos ambientados al paciente.
- Vista_medico: Contiene los atributos de la cita junto con los datos del paciente y el médico para la implementación de los métodos ambientados al médico.

Los métodos implementados para las vistas son:

Para vista_paciente.

- homeP: Encargado de mostrar la interfaz principal junto con los servicios para el usuario 'paciente'.
- médico: Muestra al médico que fue asignado hacia el usuario actual.
- citas: Muestra las citas que tiene asignadas al médico correspondiente.
- historial: Encargado de mostrar y brindar las opciones de visualizar su propio historial médico e imágenes de próstata subidas, además de tener el servicio de poder subir y guardar las imágenes de próstata.

Para vista_medico.

- homeM: Encargado de mostrar la interfaz principal junto con los servicios para el usuario 'médico'.
- pacientes: Muestra al paciente que fue asignado hacia el usuario actual.
- consulta: Encargado de mostrar y brindar las opciones de visualizar y editar el historial médico sobre un paciente en específico, junto con la opción de poder generar un reporte de diagnóstico teniendo en cuenta la imagen subida por ese paciente.
- citas: Muestra las citas que tiene asignadas al paciente correspondiente.

Figura 3.1: Diagrama de Clases parte 1

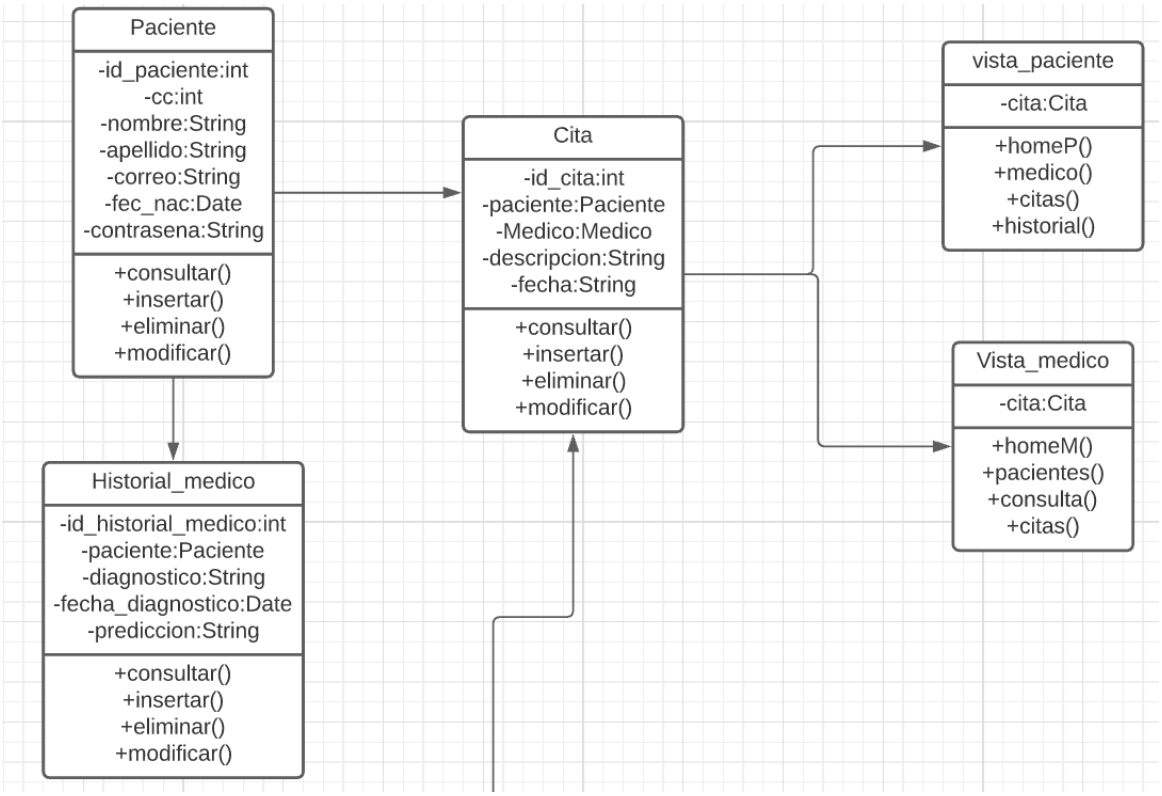
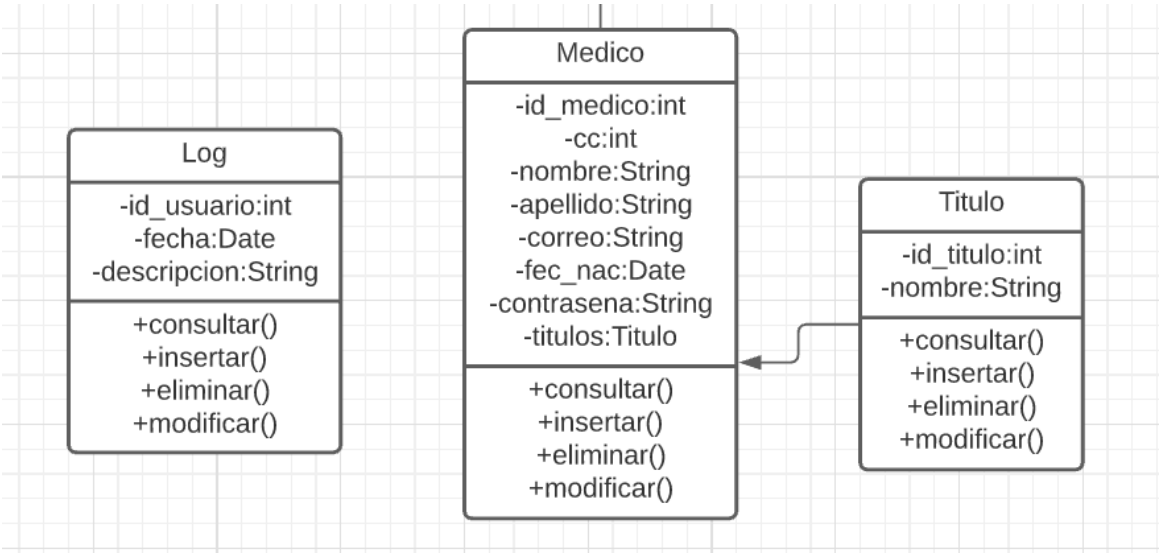


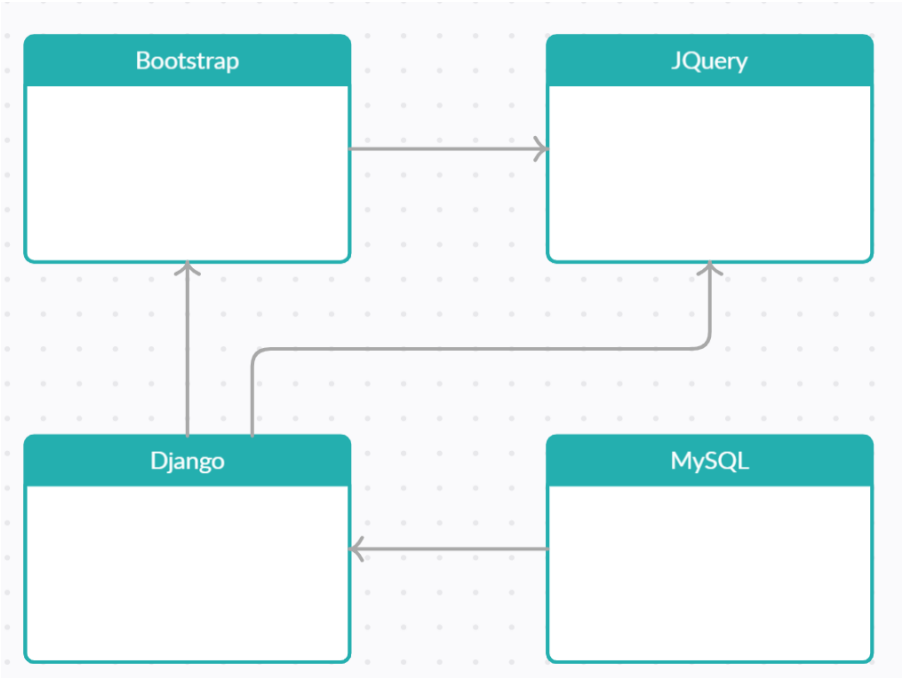
Figura 3.2: Diagrama de Clases parte 2



3.2. Diagramas de Componentes

Las dependencias entre los distintos componentes desarrollados, los frameworks y bases de datos utilizadas se observan en el diagrama de componentes de la figura 3.3.

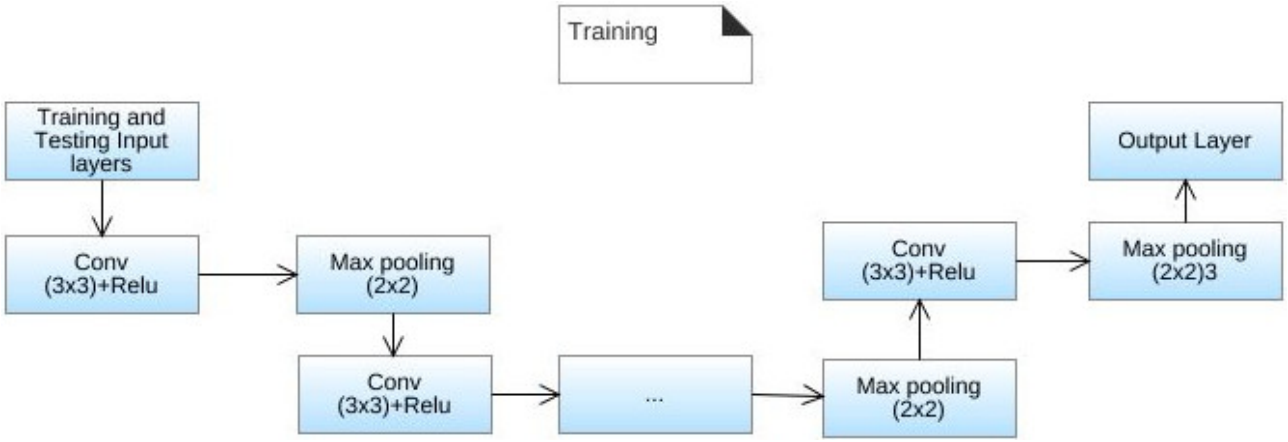
Figura 3.3: Diagrama de componentes



3.3. Modelo de red neuronal

Este proceso es con una CNN como la de Cao [8], que funciona modificando los datos obtenidos, como se ve en la Fig. 3.4, Tomando la MRI original y su correspondiente imagen segmentada. El siguiente paso es una convolución que consiste en un operador matemático entre una matriz de filtro determinada (en este caso una matriz de 3x3) y la imagen, obteniendo una nueva matriz para la extracción de características. A continuación, un agrupamiento máximo que consiste en una ventana de muestra con un valor específico (en este caso 2x2) que recorre toda la imagen donde se recupera el valor del píxel superior en la ventana de selección y lo coloca en una nueva matriz de dimensiones reducidas . La convolución y la agrupación máxima se pueden repetir tantas veces como sea necesario, siendo el último paso la capa de salida para dar la segmentación estimada de la próstata.

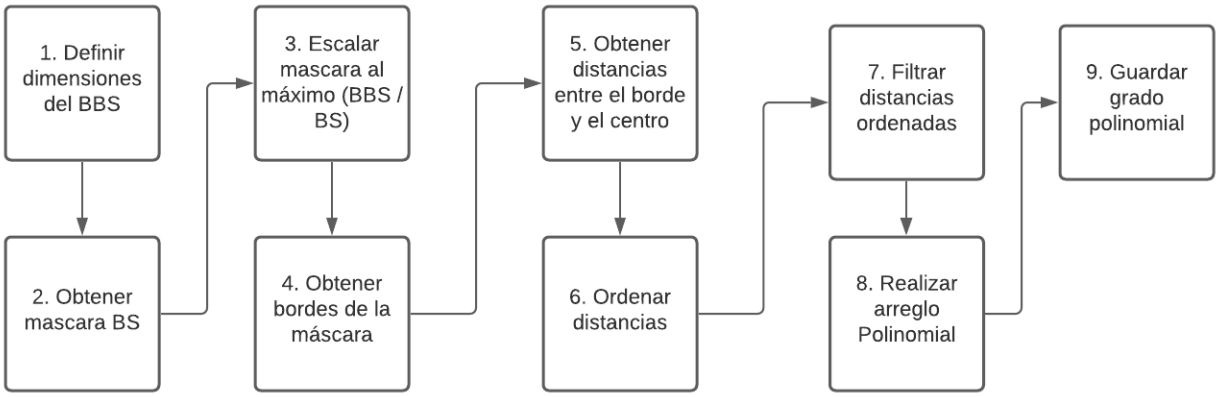
Figura 3.4: Modelo de Red Neuronal Convolucional



3.4. Modelo del algoritmo Edgerunner

A continuación en Fig. 3.5 se evidencia la linea de trabajo que ejecuta el algoritmo EdgeRunner propuesto por [2], este es implementado con el objetivo de generar la información necesaria para la ejecución de la segunda red neuronal generando un valor del Grupo de Grado Gleason para definir el grado de cancerosidad del paciente.

Figura 3.5: Modelo del algoritmo EdgeRunner



La Fig. 3.5 contiene un diagrama de 9 pasos los cuales son:

1. Definir dimensiones del BBS: Siendo BBS (Big Bounding Square) la máscara más grande entre todas las que definen exclusivamente la próstata de un paciente se deben encontrar y parametrizar las dimensiones.

2. Obtener máscara BS: Todas las máscaras serán situadas como en el paso 1 pero sin tener el cuenta que tan grande es, definiendo el "Bounding Square"(BS).

3. Escalar máscara al máximo: Todas las máscaras definidas dentro de su BS correspondiente serán escaladas hasta el máximo BS posible siendo este el BBS.

4. Obtener bordes de la máscara: Todas las máscaras pasan por un filtro en donde se obtendrá el borde que las define y eliminará todo lo demás.

5. Obtener distancias entre el borde y el centro: Todas las máscaras tendrán como punto de referencia el centro de ellas (ambas dimensiones divididas en 2) y se calcula la distancia euclidiana entre el centro y cada uno de los puntos definidos por el borde generando un arreglo desorganizado de puntos (ADP).

6. Ordenar distancias: El ADP procede a organizarse teniendo de punto inicial el encontrado directamente hacia arriba del punto central y siguiendo de punto en punto del borde en contra de las manecillas del reloj, en consecuencia se genera el arreglo ordenado de puntos (AOP).

7. Filtrar distancias ordenadas: Debido a que la AOP tiene componentes de alta frecuencia por el proceso del paso 3, procedemos a eliminar este factor por medio de un filtro el cual consiste en truncar las frecuencias que pasen el 80 % del espectro original.

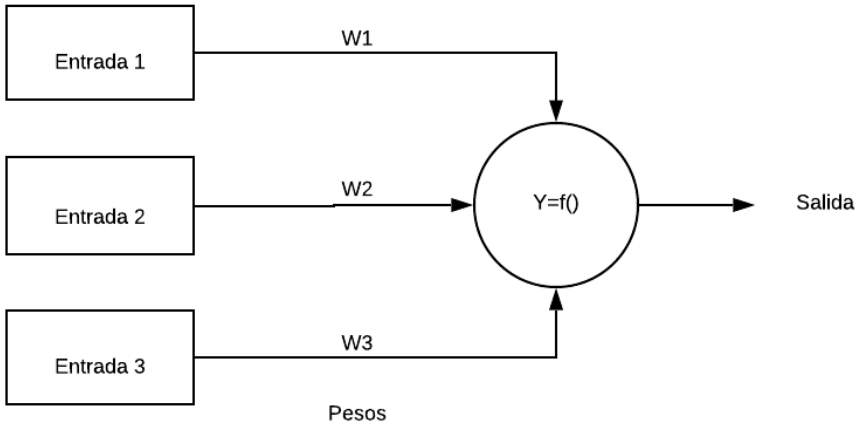
8. Realizar arreglo polinomial: Teniendo en cuenta el AOP filtrado se realiza un arreglo polinomial con un valor de correlación del 0.98 manteniendo la heterogeneidad de los bordes.

9. Guardar grado polinomial: Cada máscara obtendrá un valor de correlación siguiendo los pasos anteriores, se guardará y describirá por medio de un histograma.

3.5. Modelo de segunda red Neuronal

El modelo de la segunda red neuronal consta de el ingreso de 3 valores lo cuales son correspondientes a los bins obtenidos en el algoritmo de Edge runner y nos da como salida un valor de 1 a 5 como se muestra en Fig. 3.6, correspondiente con una puntuación de G-score.

Figura 3.6: Modelo de segunda Red Neuronal

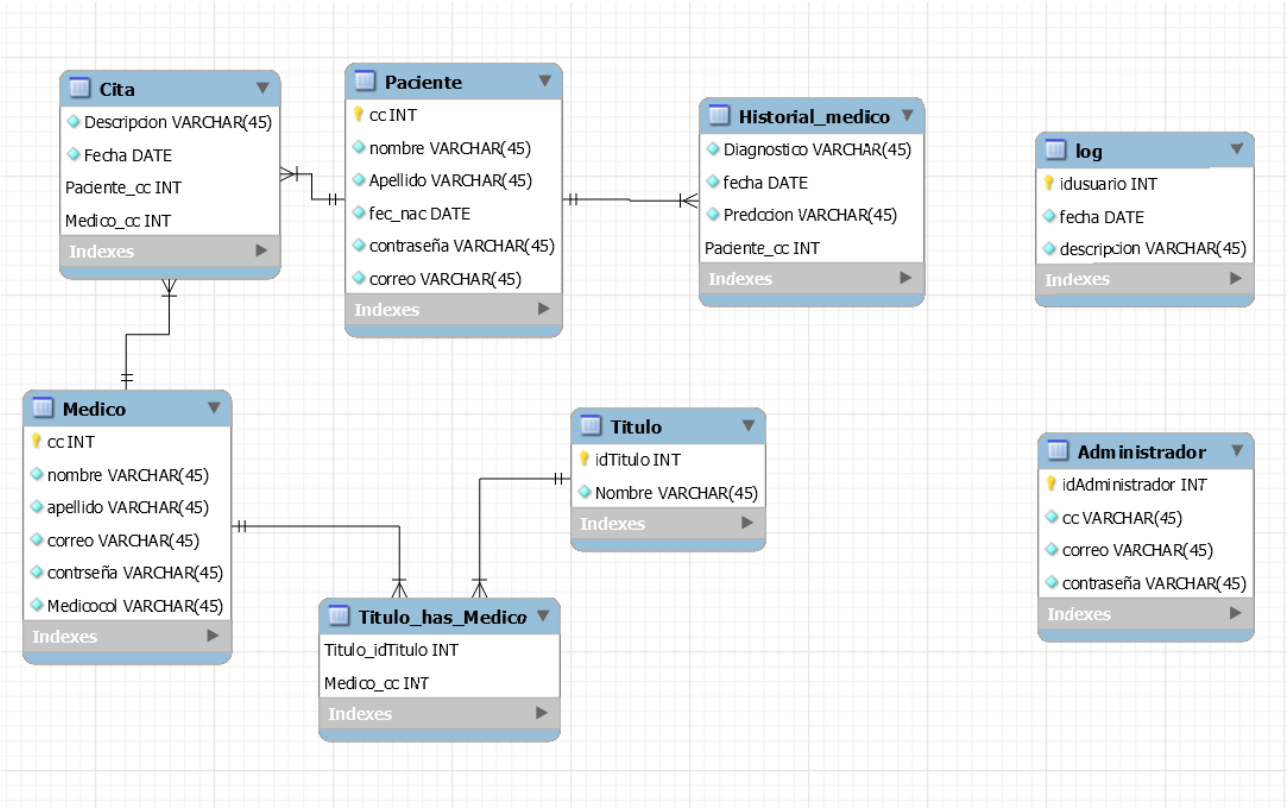


Capítulo 4

Implementación

4.1. Modelo Relacional

Figura 4.1: Diagrama de Clases



Con base en los casos de uso y su documentación se crea el diagrama de clases que se presenta en la Figura 4.1. El relacional cuenta con las siguientes clases:

- `Medico`; Está compuesta por los atributos del médico y la aplicación de los servicios adquiridos a través del análisis de requerimientos.
- `Paciente`; Está compuesta por los atributos del Paciente y la aplicación de los servicios adquiridos a través del análisis de requerimientos.
- `Log`; Está compuesta por los atributos del Log y la aplicación de los servicios adquiridos a través del análisis de requerimientos.
- `Cita`; Está compuesta por los atributos del Cita y la aplicación de los servicios adquiridos a través del análisis de requerimientos.
- `Titulo`; Está compuesta por los atributos del Titulo y la aplicación de los servicios adquiridos a través del análisis de requerimientos.
- `Historial médico`; Está compuesta por los atributos del Historial médico y la aplicación de los servicios adquiridos a través del análisis de requerimientos.
- `Administrador`; Está compuesta por los atributos del Administrador y la aplicación de los servicios adquiridos a través del análisis de requerimientos.

4.2. Diccionario de Datos

Tabla 4.1: PACIENTE

Columna	Tipo	Atri	Nulo	Predeter	Extra	enlaces
cc	int(11)		NO			
nombre	varchar(45)		NO			
Apellido	varchar(45)		NO			
fec-nac	Date		NO			
contraseña	varchar(45)		NO			
correo	varchar(45)		NO			

Tabla 4.2: MEDICO

Columna	Tipo	Atri	Nulo	Predeter	Extra	enlaces
cc	int(11)		NO			
nombre	varchar(45)		NO			
apellido	varchar(45)		NO			
correo	varchar(45)		NO			
contraseña	varchar(45)		NO			
Medicocol	varchar(45)		NO			

Tabla 4.3: CITA

Columna	**Tipo**	**Atri**	**Nulo**	**Predeter**	**Extra**	**enlaces**
Descripcion	int(11)		NO			
fecha	varchar(45)		NO			
Paciente-cc	int(11)		NO			Paciente
Medico-cc	int(11)		NO			Medico

Tabla 4.4: TITULO

Columna	**Tipo**	**Atri**	**Nulo**	**Predeter**	**Extra**	**enlaces**
idTitulo	int(11)		NO		autoIncrement	
Nombre	varchar(45)		NO			

Tabla 4.5: TITULO-HAS-MEDICO

Columna	**Tipo**	**Atri**	**Nulo**	**Predeter**	**Extra**	**enlaces**
Titulo-idTitulo	int(11)		NO		Titulo	
Medico-cc	int(11)		NO			Medico

Tabla 4.6: HISTORIAL MEDICO

Columna	**Tipo**	**Atri**	**Nulo**	**Predeter**	**Extra**	**enlaces**
Diagnostico	varchar(45)		NO			
fecha	varchar(45)		NO			
Prediccion	varchar(45)		NO			
Paciente-cc	int(11)		NO			Paciente

Tabla 4.7: ADMINISTRADOR

Columna	**Tipo**	**Atri**	**Nulo**	**Predeter**	**Extra**	**enlaces**
idAdministrador	int(11)		NO		autoIncrement	
cc	varchar(45)		NO			
correo	varchat(45)		NO			
contraseña	varchat(45)		NO			

Tabla 4.8: LOG

Columna	**Tipo**	**Atri**	**Nulo**	**Predeter**	**Extra**	**enlaces**
idusuario	int(11)		NO		autoIncrement	
fecha	Date		NO			
descripcion	varchar(45)		NO			

Capítulo 5

Desarrollo

5.1. Sprint 1

En esta sección se desea recopilar información de artículos de revistas o conferencias reconocidas, los cuales presentan distintos métodos o técnicas para el desarrollo de diagnostico basado en imágenes MR, esta investigación permitirá obtener los conocimientos necesarios para el desarrollo del proyecto.

5.1.1. Desarrollo

La estrategia de búsqueda implementada en esta recopilación de información fue la estrategia de población, intervención y resultado sugerida por Kitchenham [19], que nos permite generar una cadena de búsqueda adecuada.

Se hizo una recolección de artículos con trabajos relacionados con la manipulación de MRI y cáncer de próstata de bases de datos como scopus, IEEE Xplore, SpringerLink y ScienceDirect. Donde se clasificaron de tal manera para tener como preseleccionados los mas relevantes con la investigación tal como se muestra en la tabla 5.1.

Tabla 5.1: Artículos preseleccionados

Base de Datos	**Encontrado**	**Duplicados**	**Pre-seleccionado**
Scopus	12	2	6
IEEE Xplore	10	1	7
SpringerLink	7	0	2
ScienceDirect	5	1	3

Obtenidos los artículos preseleccionados se reviso y analizo uno por uno las técnicas o métodos usados para encontrar cuales eran mas afines a la presente investigación con el propósito de obtener los artículos seleccionados definitivamente como se muestra en la tabla 5.2.

Tabla 5.2: Artículos Seleccionados

Base de Datos	**Preseleccionados**	**Seleccionados**
Scopus	6	3
IEEE Xplore	7	3
SpringerLink	2	1
ScienceDirect	3	1

Tabla 5.3: Características de los artículos

Autores	**Tipo de Aprendizaje automático**	**Tipo de Puntaje de diagnóstico**	**Tipo de Imagenes**
Fehr *et al*	SVM	G-score	T2W
Nguyen *et al*	RF	G-score	SLIM
Giannini *et al*	SVM	NA	T2W
Peyret *et al*	SVM	NA	Multispectral
Wang *et al*	SVM	PI-RADS v2	mp-MRI
Cao *et al*	CNN	NA	T2W
Reda *et al*	SNCAE	NA	DWI
Ginsburg *et al*	LR	G-Score	mp-MRI

5.1.2. Resultado

Los artículos obtenidos se hizo una clasificación como se muestra en la tabla 5.3 con los aspectos más importantes como: tipo de Aprendizaje donde encontramos SVM, RF, SNCAE y LR, Tipo de puntaje de diagnóstico donde resalta G-score y pi-rads v2 y Tipo de imágenes donde se evidencia que todas las imágenes usadas se encuentra entre las mp-MRI.

El resultado de la investigación otorgó la suficiente información para impulsar el desarrollo del proyecto actual así dando una guía a seguir con las metodologías más óptimas y precisas.

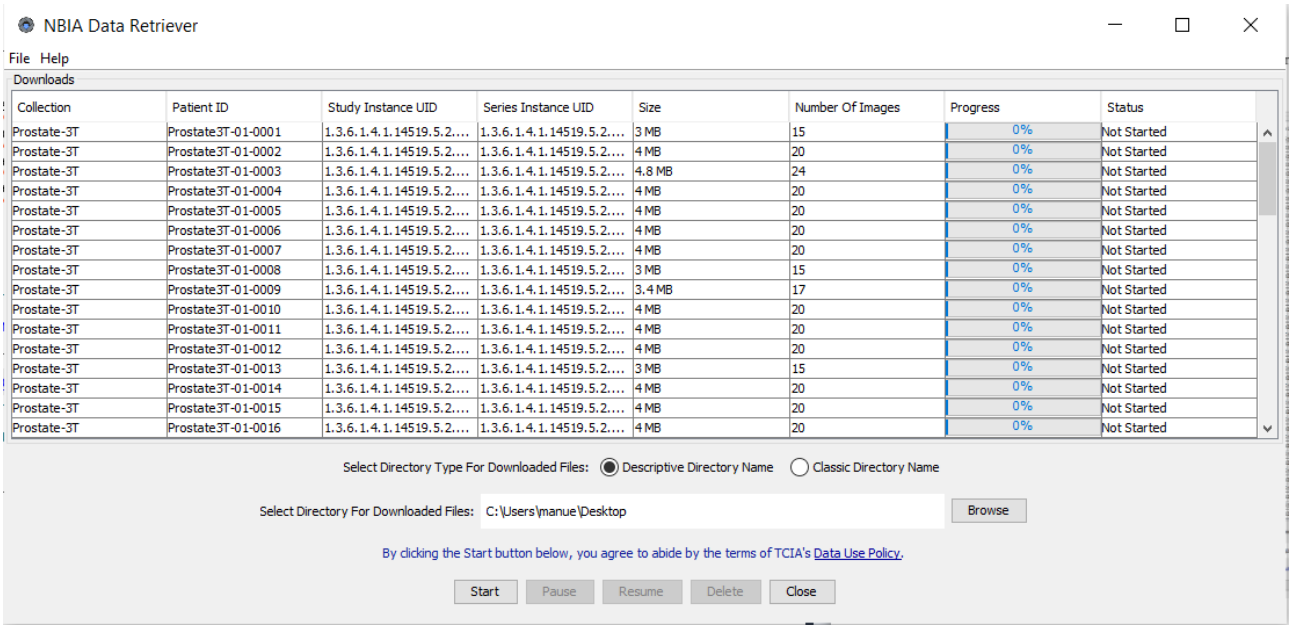
5.2. Sprint 2

En esta sección, se desea desarrollar la implementación de la subida, carga y manipulación de imágenes médicas dentro del sistema para su implementación tanto en el entrenamiento de las redes neuronales como base para el desarrollo del servicio de subida de imágenes médicas para el paciente.

5.2.1. Desarrollo

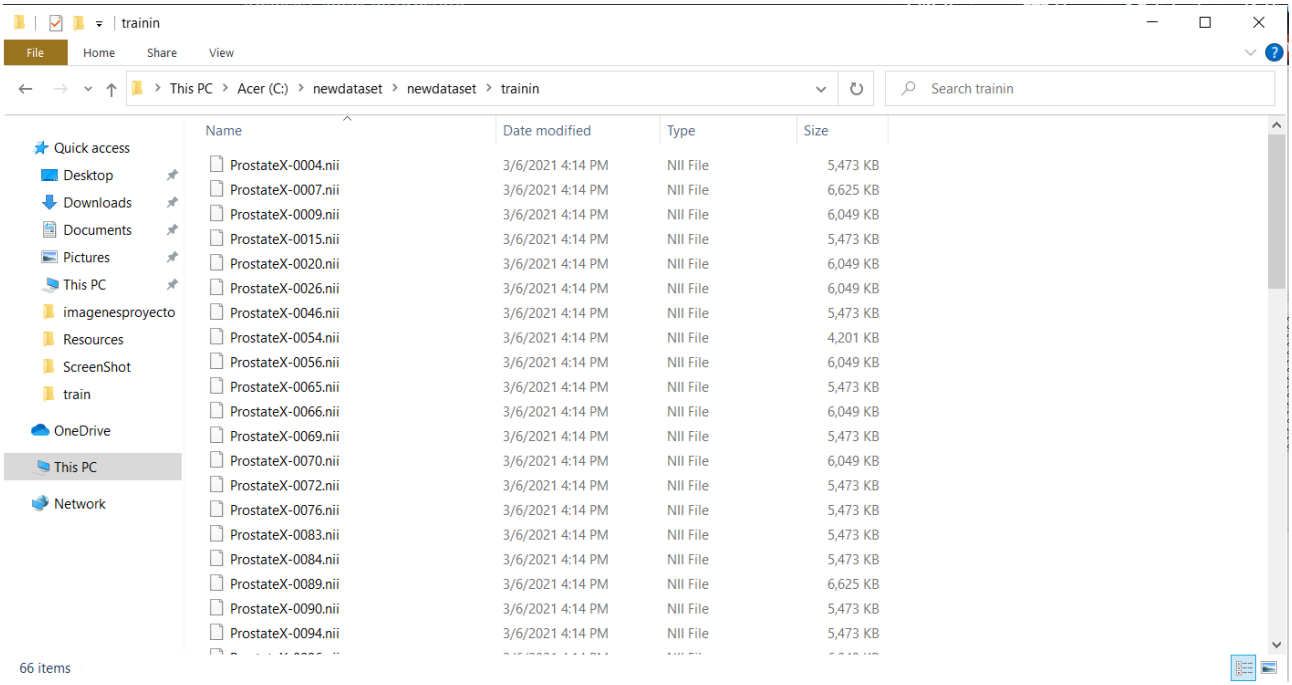
Para ello utilizamos un set de datos público que aloja imágenes MRI enfocadas a la próstata desde diferentes ángulos, este set fue encontrado en la página "The Cancer Imaging Archive"donde está compuesto por imágenes MRI como de imágenes de la próstata segmentada esto con el fin de entrenar las redes neuronales. En consecuencia se implementa el programa de "NBIA Data Retrieverçomo se evidencia en la Fig. 5.1 donde cada fila en la tabla corresponde a un conjunto de imágenes DICOM que definen la próstata de un paciente.

Figura 5.1: Vista de la aplicación NBIA Data Retriever



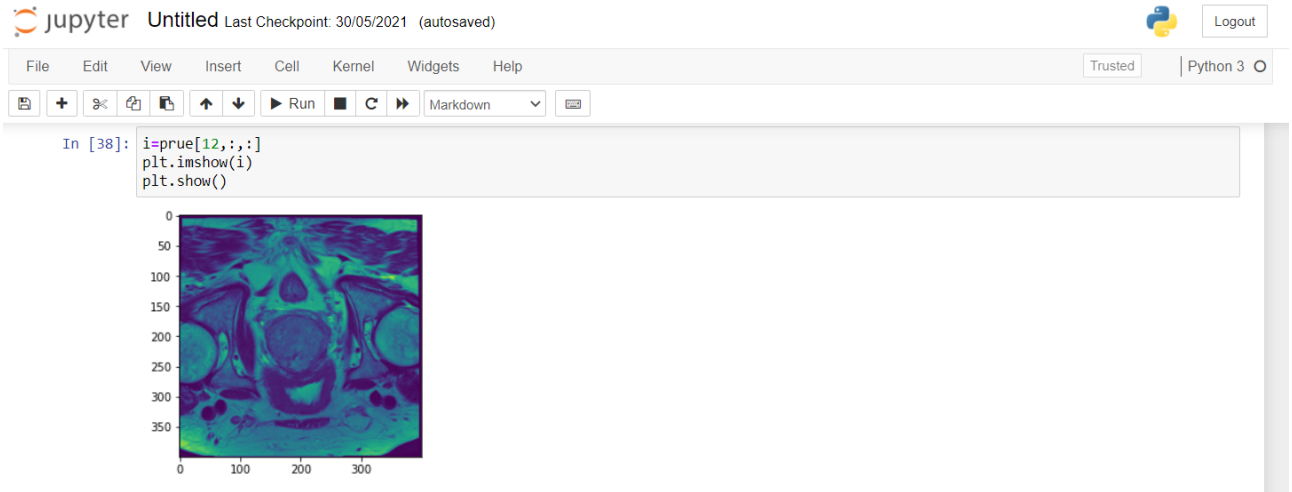
Prosiguiendo con lo anterior, por cuestiones de facilidad para el sistema las imágenes ya sean del set de datos o de un próximo paciente que vaya a subir su imagen debe ser formato Nifti (.nii) debido a lo práctico y flexible que puede llegar a ser en el sistema a comparación del formato DICOM, por lo que se optó por implementar el programa "dcm2niix.[el] cual permite convertir un conjunto de imágenes DICOM que define solo una próstata en un solo archivo Nifti, por ende en Fig. 5.2 se puede evidenciar como el set de datos descargado en formato DICOM visto en Fig. 5.1 es convertido en archivos Nifti.

Figura 5.2: Carpeta donde se aloja el set de datos convertido



Teniendo las imágenes Nifti listas para su uso procedemos a cargarlas en el sistema gracias a librerías en python capaces de cargar, mostrar y manipular archivos nifti, dicho esto en la Fig 5.3 se demuestra la carga de un archivo nifti aleatorio del set de datos dentro del entorno de desarrollo.

Figura 5.3: Visualización de una capa del archivo Nifti cargado al sistema

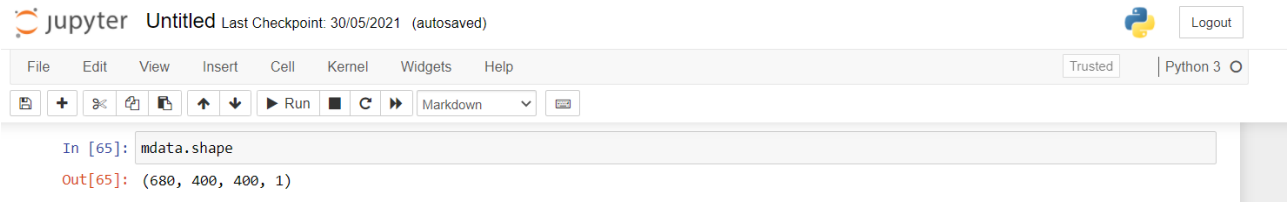


Con lo anterior en cuenta, el paso final consiste en normalizar la imagen, con normalizar hacemos

referencia a generalizar varios parámetros de la imagen debido a que la red neuronal necesita una entrada de datos específica. De este modo se implementó la normalización de imágenes en el set de datos para el entrenamiento de la red como para el paciente que desee subir su imagen al sistema, esta normalización consistió en definir las dimensiones de la imagen en (Nx400x400x1) donde N son la cantidad de slices o capas que contiene la imagen, los 400 son el alto junto al ancho de la imagen y el 1 es la dimensión que demuestra la implementación del formato de escala de grises sobre la imagen. Finalizando, el tipo de dato empleado sobre las imágenes fue float32.

5.2.2. Resultado

Figura 5.4: Dimensiones del set de entrenamiento



En Fig. 5.4 se puede apreciar la variable "mdata"la cual es un numpy array que alberga todas las imágenes del set de entrenamiento para la red neuronal. Al ejecutar el método "shape"que permite ver las dimensiones del array evidenciamos la dimensión que define las capas, la cual tiene un valor de 680 además de la implementación del formato de escala de grises debido al número que aparece como última dimensión siendo un 1.

5.3. Sprint 3

En esta sección se desea desarrollar la implementación de las redes neuronales y el algoritmo Edge runner para la obtención de un puntaje G-score como diagnostico para mostrar dentro de la interfaz tanto al paciente como al médico.

5.3.1. Desarrollo

Inicialmente para el desarrollo de la red neuronal convolucional se plantea el modelo basado en la figura 3.4, el que se usó tiene como canales 16, 32, 64 y 128 teniendo como 4 la cantidad de convoluciones que realiza esta red neuronal como se muestra en la figura 5.5.

Figura 5.5: Modelo de red neuronal de segmentación

```
In [25]: def conv_block(x, num_filters):
             x = keras.layers.Conv2D(num_filters, (3, 3), padding="same")(x)
             x = keras.layers.BatchNormalization()(x)
             x = keras.layers.Activation("relu")(x)

             x = keras.layers.Conv2D(num_filters, (3, 3), padding="same")(x)
             x = keras.layers.BatchNormalization()(x)
             x = keras.layers.Activation("relu")(x)

             return x

         def build_model():
             size = 256
             num_filters = [16, 32,64,128]
             inputs = keras.layers.Input((400, 400, 1))

             skip_x = []
             x = inputs
             ## Encoder
             for f in num_filters:
                 x = conv_block(x, f)
                 skip_x.append(x)
                 x = keras.layers.MaxPool2D((2, 2))(x)

             ## Bridge
             x = conv_block(x, num_filters[-1])

             num_filters.reverse()
             skip_x.reverse()
             ## Decoder
             for i, f in enumerate(num_filters):
                 x = keras.layers.UpSampling2D((2, 2))(x)
                 xs = skip_x[i]
                 x = keras.layers.Concatenate()([x, xs])
                 x = conv_block(x, f)

             ## Output
             x = keras.layers.Conv2D(1, (1, 1), padding="same")(x)
             x = keras.layers.Activation("sigmoid")(x)

             return keras.models.Model(inputs, x)


         if __name__ == "__main__":
             model = build_model()
             model.summary()
conv2d_35 (Conv2D)              (None, 400, 400, 16) 6928        concatenate_7[0][0]
_____
batch_normalization_34 (BatchNo (None, 400, 400, 16) 64          conv2d_35[0][0]
_____
activation_35 (Activation)      (None, 400, 400, 16) 0           batch_normalization_34[0][0]
_____
conv2d_36 (Conv2D)              (None, 400, 400, 16) 2320        activation_35[0][0]
_____
batch_normalization_35 (BatchNo (None, 400, 400, 16) 64          conv2d_36[0][0]
_____
activation_36 (Activation)      (None, 400, 400, 16) 0           batch_normalization_35[0][0]
_____
conv2d_37 (Conv2D)              (None, 400, 400, 1)  17          activation_36[0][0]
_____
activation_37 (Activation)      (None, 400, 400, 1)  0           conv2d_37[0][0]
==========
Total params: 1,229,665
Trainable params: 1,227,233
Non-trainable params: 2,432
_____
```

Ya con el modelo de la red neuronal establecido se compila y entrena en este caso se usó como optimizador el RMSprop con un valor de lr de 0.001, función de perdida binary crossentropy y función de precisión accuracy. El entrenamiento se hizo con los siguientes parámetros: 10 epoch, 17 pasos por epoch y verbose con el valor de 1, como se muestra en la figura 5.6.

Figura 5.6: Entrenamiento de red neuronal de segmentación

```
In [19]:
from tensorflow.keras.optimizers import RMSprop
model.compile(optimizer=RMSprop(lr=0.001), loss="binary_crossentropy"
              , metrics=['accuracy'])

In [27]:
history = model.fit(
      dataset,
      epochs=10,
      steps_per_epoch=17,
      verbose=1)

Epoch 1/10
17/17 [==============================] - 862s 49s/step - loss: 0.7301 - accuracy: 0.5687
Epoch 2/10
17/17 [==============================] - 885s 52s/step - loss: 0.4664 - accuracy: 0.9718
Epoch 3/10
17/17 [==============================] - 886s 52s/step - loss: 0.4082 - accuracy: 0.9779
Epoch 4/10
17/17 [==============================] - 872s 51s/step - loss: 0.3967 - accuracy: 0.9761
Epoch 5/10
17/17 [==============================] - 866s 51s/step - loss: 0.3510 - accuracy: 0.9776
Epoch 6/10
17/17 [==============================] - 866s 51s/step - loss: 0.3168 - accuracy: 0.9782
Epoch 7/10
17/17 [==============================] - 859s 50s/step - loss: 0.2985 - accuracy: 0.9758
Epoch 8/10
17/17 [==============================] - 870s 51s/step - loss: 0.2719 - accuracy: 0.9784
Epoch 9/10
17/17 [==============================] - 874s 51s/step - loss: 0.2424 - accuracy: 0.9789
Epoch 10/10
17/17 [==============================] - 883s 52s/step - loss: 0.2186 - accuracy: 0.9795
```

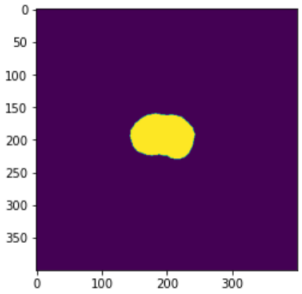
Por ultimo como se muestra en la figura 5.7, como se realiza una predicción con la red neuronal ya entrenada con un valor de batch de 17 de una imagen nifti normalizada con los requerimientos nombrados en el sprint 2.

Figura 5.7: Predicción de red neuronal de segmentación

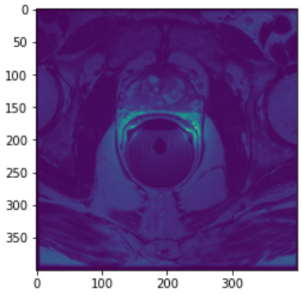
```
In [29]: classes = model.predict(prue, batch_size=17)
         classes.shape

Out[29]: (40, 400, 400, 1)

In [17]: i=classes[17,:,:,0]
         plt.imshow(i)
         plt.show()
```



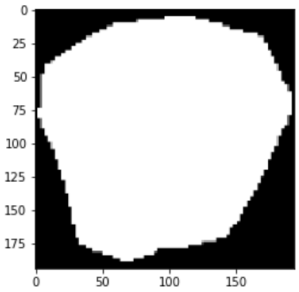

```
In [16]: i=prue[17,:,:,0]
         plt.imshow(i)
         plt.show()
```



A continuación se empieza a trabajar con la imagen segmentada el primer paso que se realiza es cortar la imagen encuadrando la próstata y luego aumentarla como se muestra en la figura 5.8, para facilitar la manipulación de la imagen para obtener los datos.

Figura 5.8: Zoom de imagen segmentada

```
In [17]: plt.imshow(bbs[12,:,:], cmap="gray")
         #plt.imshow(bbs[15,:,:])
         plt.show()
         bb.shape
```

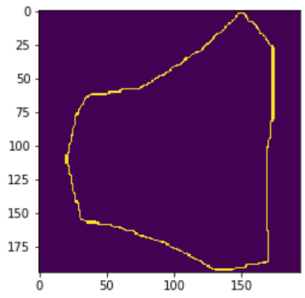



```
Out[17]: (194, 194, 1)
```

Con la imagen aumentada lo que se hace a continuación es recorrer la imagen punto a punto tomando tan solo los puntos externos para obtener con precisión el borde de la próstata como se muestra en la figura 5.9.

Figura 5.9: Obtención de borde de imagen segmentada

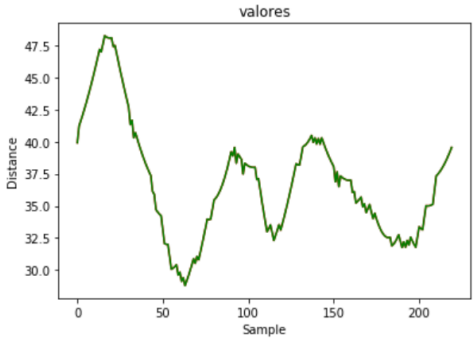
```
In [26]: img=edges[14,:,:]
         plt.imshow(img)
         plt.show()
```



A partir del borde obtenido lo que procede es tomar el punto central de la imagen y desde este punto se hace un recorrido en línea recta hasta una sección de borde y en orden de las manecillas del reloj se va recorriendo el borde punto a punto obteniendo el valor del recorrido del centro a cada punto hasta llegar al punto de inicio esto con el objetivo de general una lista de medidas las cuales se puede ver gráficamente en la figura 5.10.

Figura 5.10: Gráfica de puntos del centro al borde de la imagen segmentada

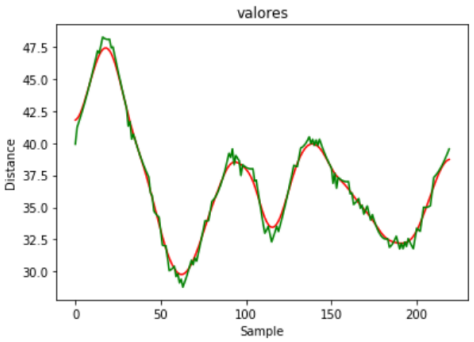
```
In [275]: cs.createPlotCompare1(variable[:,1],variable[:,1],"valores")
```



La gráfica generada con los puntos se modifica de tal manera que podamos obtener de ella una gráfica senoidal como se muestra en la figura 5.11, con la cual se puede que generar una función matemática.

Figura 5.11: Grafica de función obtenida a partir de la gráfica de los puntos

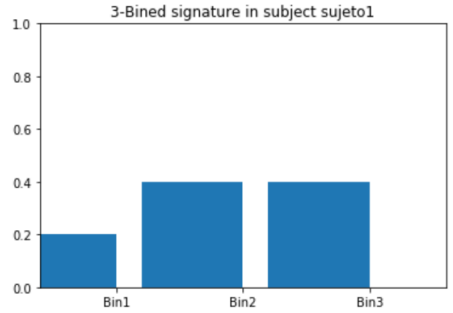
```
In [279]: cs.createPlotCompare1(osf,variable[:,1],"valores")
```



Finalmente con la función matemática obtenida anteriormente se define la complejidad que esta tiene y según sea se trafican 3 bins como se muestra en la figura 5.12 donde si el bin de mayor valor es el primero tiene una complejidad baja y si el ultimo bin es el del mayor valor la complejidad es alta, a partir de estos valores se genera el dataset de la segunda red neuronal.

Figura 5.12: Gráfica de Bins resultantes

```
In [9]: cs.plotHistograms(signatures, rb,"C:/imagenes.nii/arrays/")
```



A continuación en la figura 5.13 se muestra la subida de archivos que resultaron del algoritmo de edge runner con una cantidad de 60 sujetos para el entrenamiento de la segunda red neuronal.

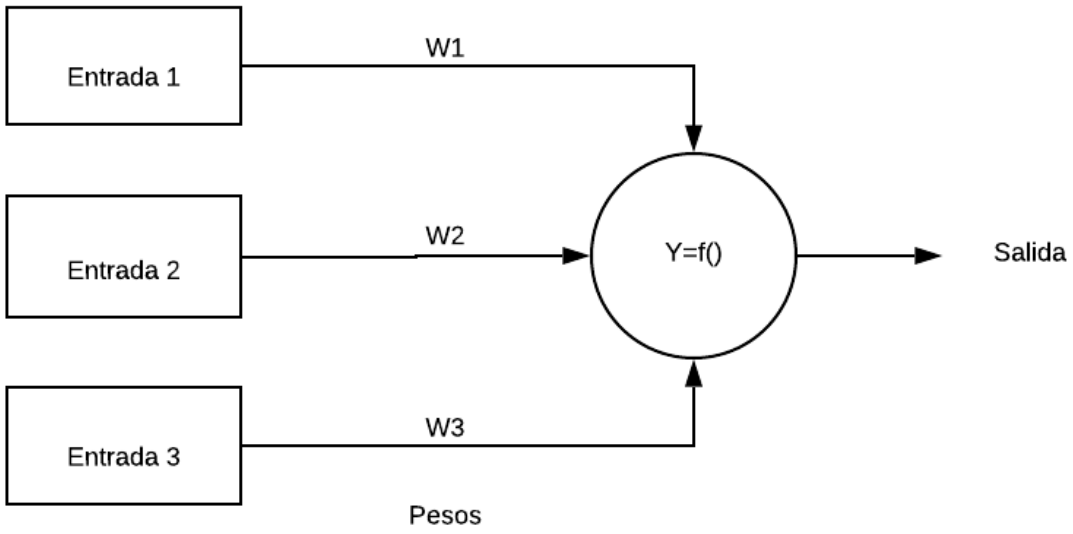
Figura 5.13: Subida de resultados del edge runner

```
In [3]: mdata=[]
        mtest=[]
        numeros= []
        while(len(numeros)<0):
            num = random.randint(0, 20)
            if (num not in numeros):
                numeros.append(num)
        for i in range(0,len(contenido2)):
            aux=mdata
            aux2=mtest
            array1 = np.load("C:/datos2red/"+contenido2[i]+".npy",allow_pickle=True)
            sig=array1[0]
            sig=np.array(sig[0])
            sig=np.delete(sig,0)
            data=sig[0]
            data1=sig[1]
            while(data.shape!=(3,)):
                data=np.insert(data,  np.array((data.shape)), 0)
        #         data1=np.insert(data1,  np.array((data1.shape)), 0)
        #     data=np.array((np.array(data, dtype='float32'),np.array(data1, dtype='float32')))
            if(i in numeros):
                mtest.append(data)
            else:
                mdata.append(data)
        mtest=np.array(mtest)
        mdata=np.array(mdata)
        print(mdata.shape)
        print(mtest.shape)

        (59, 3)
        (0,)
```

El modelo que contempla el ejercicio de la clasificación del puntaje diagnostico es de una red neuronal básica como se muestra en la figura 5.14 el cual recibe 3 valores y retorna un valor entre 1 y 5 que corresponden al puntaje respectivo de G-score.

Figura 5.14: Modelo de la segunda red neuronal



Finalmente se hace predicción con datos externos donde el resultado que retorna la red neuronal está en un rango de 1 a 5 como se muestra en la figura 5.15.

Figura 5.15: Predicción de la segunda red neuronal

```
In [10]: data=np.expand_dims(imgprueba,axis=0)
         data.shape
         classes=model.predict(data)

In [11]: classes

Out[11]: array([[1.]], dtype=float32)
```

5.3.2. Resultado

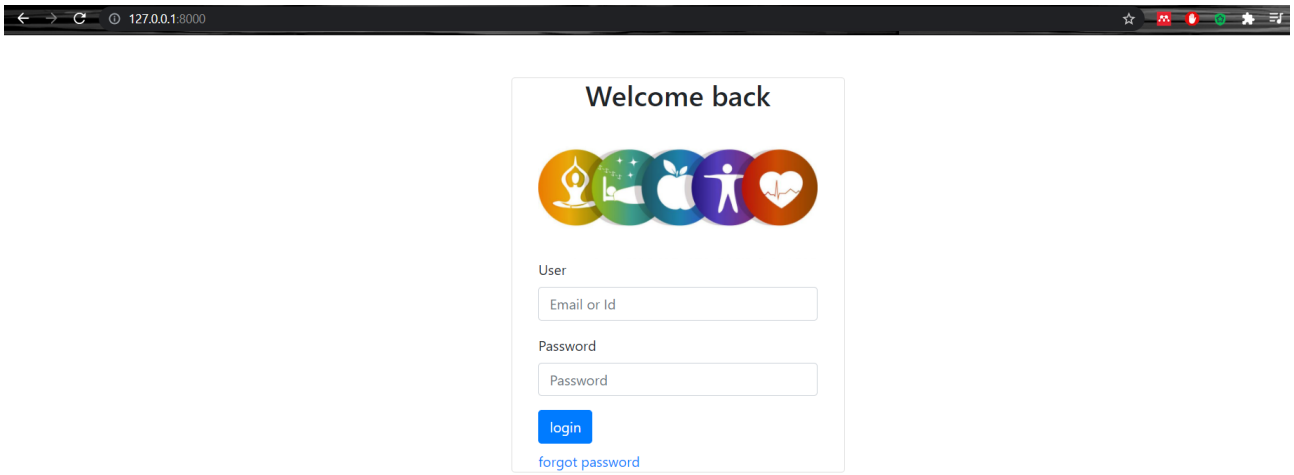
De los resultados obtenidos en el desarrollo de esta fase podemos encontrar todo el análisis de las MRI donde concluyeron con el resultado de un puntaje de diagnostico que se puede encontrar en la figura 5.15. Con el cual se reproducirá tanto la ejecución como el resultado dentro de la interfaz gráfica del sprint 4.

5.4. Sprint 4

5.4.1. Desarrollo

En esta sección se implementa el desarrollo de la interfaz gráfica encargada de mostrar los servicios propuestos anteriormente a los usuarios correspondientes, cada usuario deberá iniciar sesión para ser direccionado a su página correspondiente como se ve en Fig. 5.16.

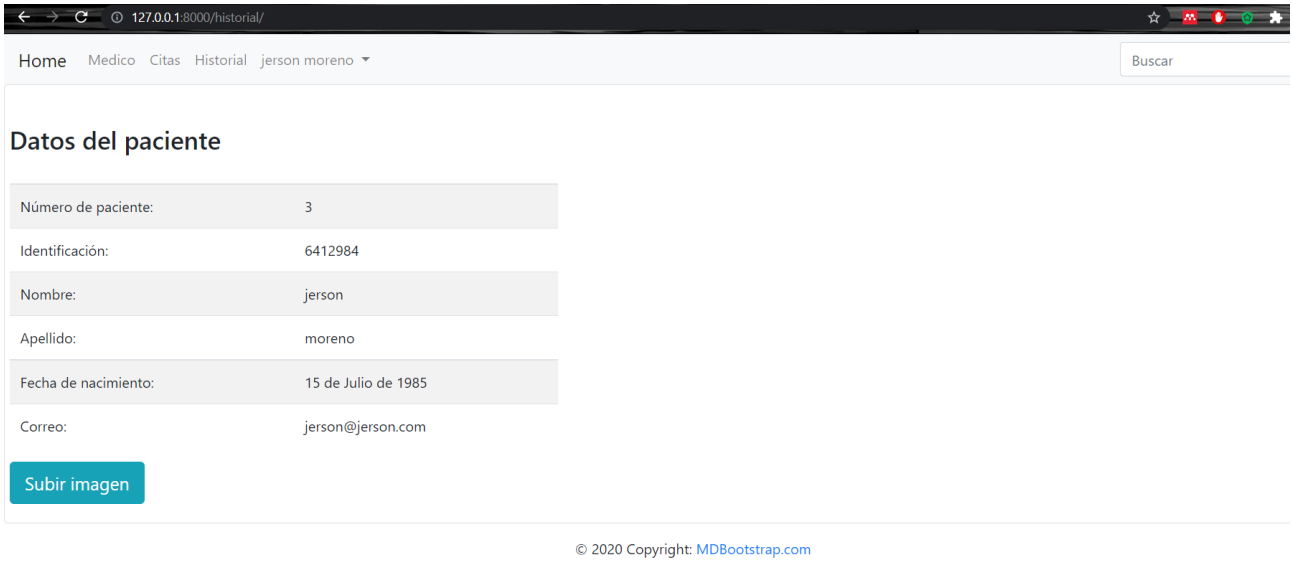
Figura 5.16: Prototipo de la interfaz gráfica encargada del inicio de sesión



5.4.2. Resultado

En este caso como se ve en Fig. 5.17 se evidencia el prototipo de la página del paciente en donde tiene se encuentra en la sección para subir su imagen médica al sistema.

Figura 5.17: Prototipo de la interfaz gráfica del paciente



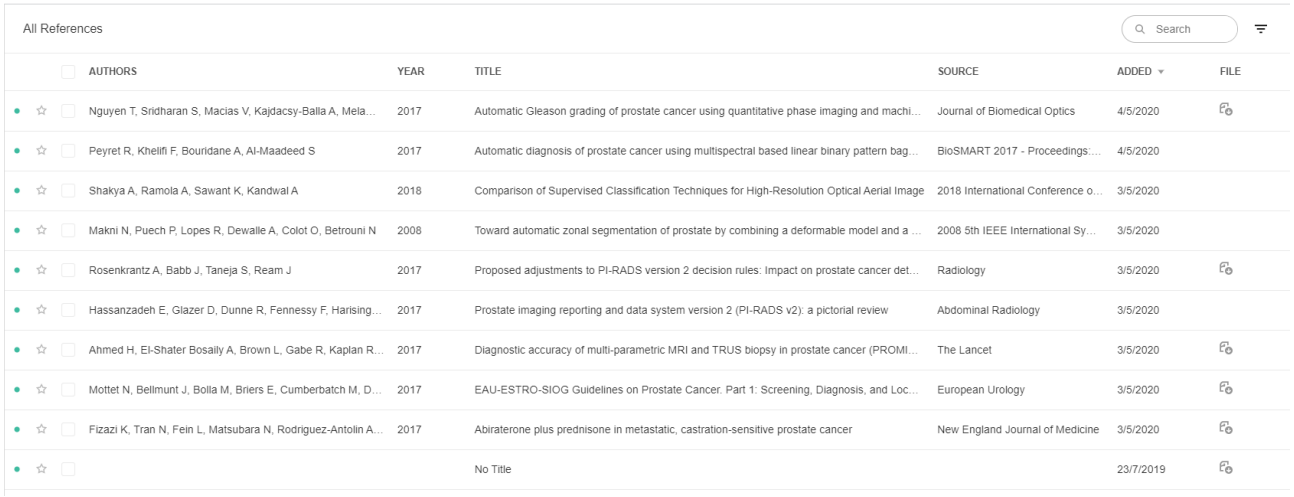
Capítulo 6

Pruebas

6.1. Sprint 1

En esta fase se puede evidenciar en Figura6.1 una recopilación de artículos de los cuales se obtuvo información sobre la prostata y estadísticas globales sobre el cáncer donde se puede enfatizar las distintas técnicas para segmentación de la próstata para su clasificación y correspondiente relación con el nivel de cáncer.

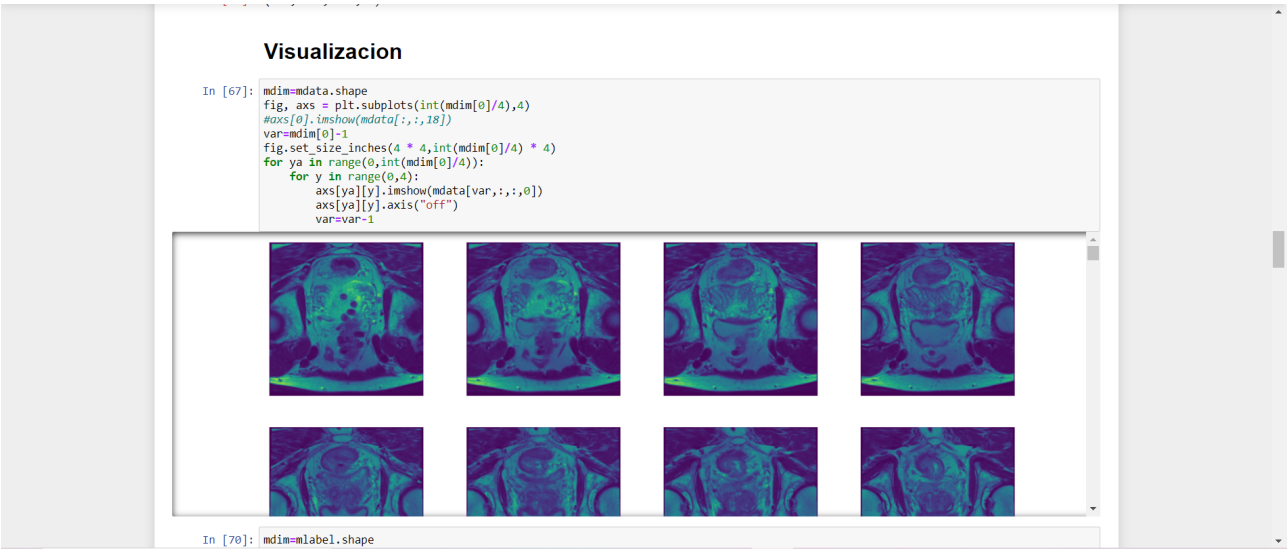
Figura 6.1: visualización de contenido referencial de la investigación



6.2. Sprint 2

En esta fase se puede evidenciar en Figura 6.2 un conjunto de imágenes correspondiente a las imagenMRI de pacientes aleatorios subidas al sistema y normalizadas por el mismo. Debido a que las imágenes pueden variar en cuestión de dimensiones, cantidad de slices e incluso la definición de los bits dentro de ella se recurre a normalizar las dimensiones (alto y ancho respectivamente) de todas las que entren al sistema a (300x300) y definir el tipo de bits dentro de ellas en Float32 antes de emplearlas en la predicción.

Figura 6.2: visualización sobre la subida y manejo de imágenes en el entorno de desarrollo



6.3. Sprint 3

En esta fase se puede evidenciar en Figura 6.3 el resultado de el aprendizaje supervisado con la librería tensorflow en python, donde se muestra la segmentación de la próstata en MRI. Teniendo eso en cuenta la imagen segmentada pasa por el algoritmo edgeRunner donde se obtiene una función polinomial 5.11 con la cual finalmente ingresa a la segunda red neuronal para la última predicción como se evidencia en Figura 6.4 donde el resultado final donde la próstata segmentada siendo imgprueba pasa por el algoritmo edgeRunner.

Figura 6.3: visualización de la segmentación de la próstata

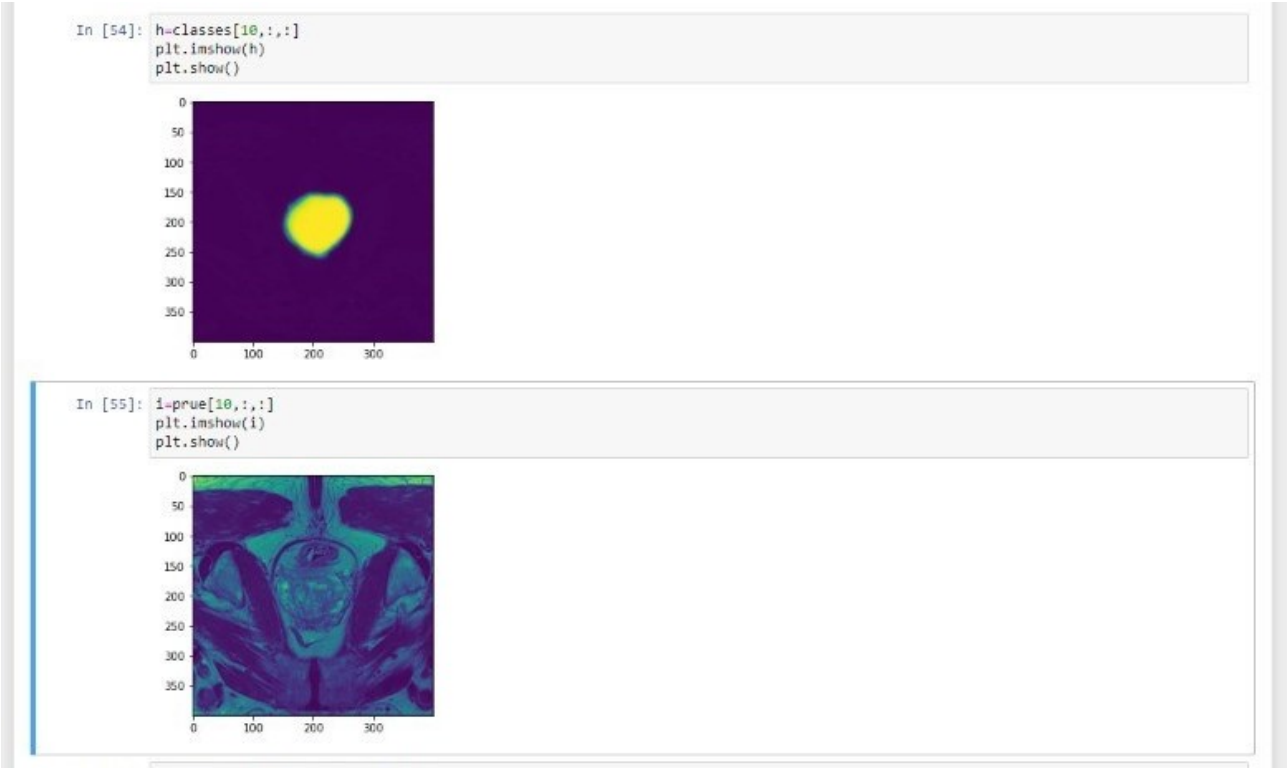


Figura 6.4: visualización de la clasificación de la imagen según el grado Gleason



6.4. Sprint 4

En esta fase se puede evidenciar en Figura 6.5 el prototipo de la interfaz del sistema que proporcionará los servicios correspondientes a sus respectivos usuarios, en la parte superior de la página está alojada una barra de tareas por la cual el usuario puede acceder a todos sus servicios.

Figura 6.5: Visualización de la interfaz principal desde el usuario paciente



Conclusiones

Concluyendo, el proyecto presentado en este documento puede ser aplicado en el ámbito práctico de un hospital debido a que el sistema web simplifica el procedimiento a seguir para cualquier paciente y médico que desee o se le ha impuesto un examen de próstata. Agregando la posibilidad de que este sistema pueda ser agregado como módulo sobre un sistema más grande o por el contrario sea el núcleo para un sistema de información web más elaborado en el ámbito médico.

Sumado a lo anterior, se espera que este proyecto evolucione con la implementación de las PACS con el objetivo de poder archivar y comunicar las imágenes de varios entes hospitalarios con tal de implementar este sistema entre una red hospitalaria.

Respecto al diseño de los módulos, se obtuvo el resultado esperado de la interfaz gráfica junto con las redes neuronales entrenadas, esto se debe a lo intuitivo que llegan a ser los procesos tanto para el paciente como el médico, además de ofrecer una gran reducción de tiempo en el proceso del diagnóstico junto con un grado de precisión considerable para tener en cuenta a la hora de tomar decisiones. En consecuencia se evita la necesidad de que la imagen sea analizada manualmente o se requieran de métodos invasivos e incómodos para el paciente como puede ser el PSA y el médico tendrá la opción de generar un documento para la toma de decisiones sobre ese paciente sin necesidad de otro profesional como un radiólogo.

Finalizando, los inconvenientes presentados en el sistema son la generación de una alta dependencia entre médico y usuario, como no hay un ente mediador entre ellos los dos interactúan directamente, de este modo si uno no cumple con su deber el otro no podrá continuar el proceso. Otro inconveniente se presenta según la cantidad de uso que se le de al sistema, si llegan grandes cantidades de solicitudes la eficacia del sistema puede decaer en grandes cantidades e incluso colgarse pues la predicción de las redes neuronales requiere una cantidad considerable de procesamiento.

Bibliografía

[1] R. L. Siegel and K. D. Miller, "(2018) cancer statistics," *CA - A Cancer Journal for Clinicians*, vol. 68, no. 1, pp. 7–30, 2018.

[2] F. Yepes-Calderon, M. D. Nelson, and J. G. McComb, "Automatically measuring brain ventricular volume within pacs using artificial intelligence," *PLOS ONE*, vol. 13, pp. 1–14, 03 2018.

[3] D. Fehr, H. Veeraraghavan, A. Wibmer, T. Gondo, K. Matsumoto, H. A. Vargas, E. Sala, H. Hricak, and J. O. Deasy, "Automatic classification of prostate cancer gleason scores from multiparametric magnetic resonance images," *Proceedings of the National Academy of Sciences*, vol. 112, no. 46, pp. E6265–E6273, 2015.

[4] T. H. Nguyen, S. Sridharan, V. Macias, A. Kajdacsy-Balla, J. Melamed, M. N. Do, and G. Popescu, "Automatic gleason grading of prostate cancer using quantitative phase imaging and machine learning," *Journal of biomedical optics*, vol. 22, no. 3, p. 036015, 2017.

[5] V. Giannini, S. Mazzetti, A. Vignati, F. Russo, E. Bollito, F. Porpiglia, M. Stasi, and D. Regge, "A fully automatic computer aided diagnosis system for peripheral zone prostate cancer detection using multi-parametric magnetic resonance imaging," *Computerized Medical Imaging and Graphics*, vol. 46, pp. 219–226, 2015.

[6] R. Peyret, F. Khelifi, A. Bouridane, and S. Al-Maadeed, "Automatic diagnosis of prostate cancer using multispectral based linear binary pattern bagged codebooks," in *2017 2nd International Conference on Bio-engineering for Smart Technologies (BioSMART)*, pp. 1–4, IEEE, 2017.

[7] J. Wang, C.-J. Wu, M.-L. Bao, J. Zhang, X.-N. Wang, and Y.-D. Zhang, "Machine learning-based analysis of mr radiomics can help to improve the diagnostic performance of pi-rads v2 in clinically relevant prostate cancer," *European Radiology*, vol. 27, no. 10, pp. 4082–4090, 2017.

[8] R. Cao, A. M. Bajgiran, S. A. Mirak, S. Shakeri, X. Zhong, D. Enzmann, S. Raman, and K. Sung, "Joint prostate cancer detection and gleason score prediction in mp-mri via focalnet," *IEEE transactions on medical imaging*, vol. 38, no. 11, pp. 2496–2506, 2019.

[9] I. Reda, A. Shalaby, F. Khalifa, M. Elmogy, A. Aboulfotouh, M. A. El-Ghar, E. Hosseini-Asl, N. Werghi, R. Keynton, and A. El-Baz, "Computer-aided diagnostic tool for early detection of prostate cancer," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 2668–2672, IEEE, 2016.

[10] S. B. Ginsburg, A. Algohary, S. Pahwa, V. Gulani, L. Ponsky, H. J. Aronen, P. J. Boström, M. Böhm, A.-M. Haynes, P. Brenner, *et al.*, "Radiomic features for prostate cancer detection on mri differ between the transition and peripheral zones: preliminary findings from a multi-institutional study," *Journal of Magnetic Resonance Imaging*, vol. 46, no. 1, pp. 184–193, 2017.

[11] J. Rumbaugh, I. Jacobson, and G. Booch, "The unified modeling language," *Reference manual*, 1999.

[12] J. J. Gutiérrez, "¿ qué es un framework web," *Available in: http://www. lsi. us. es/˜ javierj/investigacion_ficheros/Framework. pdf Accessed May*, vol. 12, 2014.

[13] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2018.

[14] R. González Duque, "Python para todos," 2014.

[15] D. S. Foundation, "Django overview." `https://www.djangoproject.com/start/overview/`, 2018.

[16] G. B. Team, "¿por qué tensorflow?." `https://www.tensorflow.org/about?hl=es-419`, 2015.

[17] A. MySQL, "Mysql," 2001.

[18] T. Satpathy, "Una guía para el conocimiento de scrum (guía sbok™)," *Phoenix, Arizona: SCRUMstudy*, 2016.

[19] B. Kitchenham, "Procedures for undertaking systematic reviews: Joint technical report." http://www.inf.ufsc.br/ aldo.vw/kitchenham.pdf, 2004.