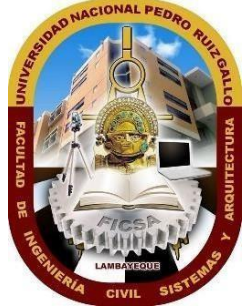




UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO



**FACULTAD DE INGENIERÍA CIVIL, SISTEMAS Y
ARQUITECTURA**

**ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS**

TESIS

**“SISTEMA INTELIGENTE BASADO EN DEEP
LEARNING PARA EL DIAGNÓSTICO DE CÁNCER DE
PRÓSTATA”.**

Para obtener el título profesional de:

INGENIERO DE SISTEMAS

AUTORES:

SANTAMARIA SANTISTEBAN, Jahir S.

SIESQUEN VALDIVIA, Luis Felipe

ASESOR:

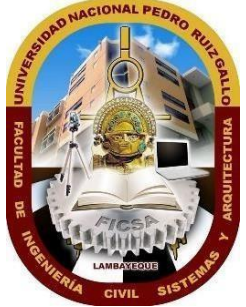
DR. ING. VILLEGAS CUBAS, Juan Elías

LAMBAYEQUE-PERÚ

2024



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO



FACULTAD DE INGENIERÍA CIVIL, SISTEMAS Y
ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS

TESIS

**“SISTEMA INTELIGENTE BASADO EN DEEP
LEARNING PARA EL DIAGNÓSTICO DE CÁNCER DE
PRÓSTATA”.**

Para obtener el título profesional de:

INGENIERO DE SISTEMAS

Aprobado por los miembros de Jurado:

DR. ING. CELI ARÉVALO, Ernesto Karlo
PRESIDENTE DEL JURADO

DR. ING. OTAKE OYAMA, Luis Alberto
SECRETARIO

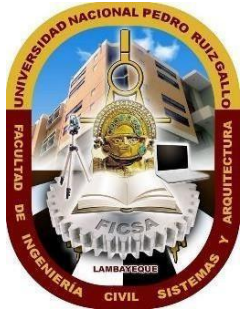
MG. ING. ARTEAGA LORA, Roberto C.
VOCAL

LAMBAYEQUE-PERÚ

2024



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO



FACULTAD DE INGENIERÍA CIVIL, SISTEMAS Y
ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE
SISTEMAS

TESIS

**“SISTEMA INTELIGENTE BASADO EN DEEP
LEARNING PARA EL DIAGNÓSTICO DE CÁNCER DE
PRÓSTATA”.**

Para obtener el título profesional de:

INGENIERO DE SISTEMAS

SANTAMARIA SANTISTEBAN, Jahir S.
AUTOR

SIESQUEN VALIDIVIA, Luis Felipe
AUTOR

DR. ING. VILLEGAS CUBAS, Juan Elías
ASESOR

LAMBAYEQUE-PERÚ

2024

AGRADECIMIENTO

A mis padres y hermanos por el apoyo incondicional, la fortaleza que me brindan y las ganas de superarme cada día.

Jahir Santos Santamaria Santisteban

AGRADECIMIENTO

Agradezco al ingeniero Juan Villegas, por su inestimable guía, su dedicación y compromiso han sido fundamentales para la realización de este trabajo.

Luis Felipe Siesquen Valdivia

DEDICATORIA

A mi padre José de los Santos por ser mi guía y acompañarme desde donde se encuentre. A mi madre Maria Andrea por ser siempre el motor y motivo de mi vida.

Jahir Santos Santamaria Santisteban

DEDICATORIA

A mis padres, por creer siempre en mí, incluso en los momentos más difíciles. Gracias por ser mi fuente de inspiración y mi fortaleza.

Luis Felipe Siesquen Valdivia

TABLA DE CONTENIDO

I.	RESUMEN	10
II.	INTRODUCCIÓN	12
III.	DISEÑO TEÓRICO.....	15
3.1.	Antecedentes	15
3.2.	Bases Teórica	19
3.2.1.	<i>Cáncer de Próstata</i>	19
3.2.2.	<i>Factores de riesgo del cáncer de próstata</i>	19
3.2.3.	<i>Diagnóstico del cáncer de próstata</i>	20
3.2.4.	<i>Agentes Inteligentes.</i>	20
3.2.5.	<i>Sistema inteligente.</i>	21
3.2.6.	<i>Deep Learning.</i>	21
3.2.7.	<i>Modelos de Deep Learning</i>	21
3.2.8.	<i>Modelo VGG-16</i>	22
3.2.9.	<i>Modelo VGG-19</i>	23
3.2.10.	<i>Modelo ResNet50</i>	24
IV.	DISEÑO METODOLÓGICO.....	26
4.1.	Tipificación de la investigación	26
4.2.	Tabla de operacionalización de variable	28
4.3.	Población y muestra	30
4.4.	Técnicas, instrumentos, equipos y materiales	30

V.	RESULTADOS.....	33
5.1.	Análisis del conjunto de Imágenes	33
5.1.1.	<i>Descripción del conjunto de datos SICAPv2</i>	33
5.2.	Procesamiento de Datos	35
5.3.	Seleccionar un modelo óptimo de Deep Learning	38
5.3.1.	Resultados de los modelos aplicados	38
5.3.2.	Elección del modelo.....	48
5.4.	Desarrollar el sistema inteligente.....	49
5.4.1.	<i>Requisitos y preparación</i>	49
5.4.2.	<i>Preparación del entorno de trabajo</i>	49
5.4.3.	<i>Desarrollo del sistema</i>	51
5.5.	Evaluación del rendimiento del sistema inteligente.....	58
VI.	DISCUSIÓN DE RESULTADOS	64
VII.	CONCLUSIONES	67
VIII.	RECOMENDACIONES	70
IX.	BIBLIOGRAFÍA	72

ÍNDICE DE TABLAS

Tabla 1: Tabla operacional según Indicadores.....	28
Tabla 2: Distribución de imágenes en dataset.....	33
Tabla 3: Resultados algoritmos aplicados.....	49
Tabla 4: Descripción de funciones.....	54

ÍNDICE DE FIGURAS

Figura 1: Arquitectura de VGG-16 separada por 16 capas convolucionales.....	22
Figura 2: Modelo lógico de Arquitectura VGG-16 con 13 capas convolucionales y 3 densas	23
Figura 3: Modelo de red VGG-19.....	23
Figura 4: Marco de entrenamiento del modelo de detección de uso de máscara.....	24
Figura 5: Arquitectura de ResNet50	25
Figura 6: Propiedad de imagen de dataset SICAPv2	34
Figura 7: Imágenes con patrones cribiformes, no contiene signos de cáncer de próstata	35
Figura 8: Imágenes con patrones cribiformes contiene signos de cáncer de próstata	35
Figura 9: Procesamiento de una imagen con cáncer	37
Figura 10: Procesamiento de una imagen sin cáncer	37
Figura 11: Resultado modelo VGG-16	38
Figura 12: Resultado modelo VGG-19	41
Figura 13: Resultado modelo ResNet50	44
Figura 14: Directorios de trabajo	50
Figura 15: Estructura Principal	50
Figura 16: Estructura de carpetas del backend	52
Figura 17: Cuenta enlazada al proyecto	53
Figura 18: Resultados enlace de cuenta	53
Figura 19: Repositorio de GitHub.....	54
Figura 20: Menú Login del sistema	55
Figura 21: Análisis de resultados	56
Figura 22: Resultados de la evaluación	56
Figura 23: Exportable en formato PDF.....	57
Figura 24: Resumen de análisis ejecutados en el sistema.....	58
Figura 25: Conversión de las probabilidades predichas	59
Figura 26: Decisión final de categoría	59
Figura 27: Etiquetas por lote.....	60
Figura 28: Almacenamiento de etiquetas reales y predictivas.....	61
Figura 29: Evaluación del desempeño del modelo	62

I. RESUMEN

La presente investigación aborda el problema del diagnóstico de cáncer de próstata, una de las principales causas de mortalidad en hombres a nivel mundial. El diagnóstico temprano de esta enfermedad es un desafío debido a la complejidad en la detección de signos tempranos y la falta de acceso a tecnologías avanzadas en muchas regiones. En respuesta a esta problemática, se desarrolló un sistema basado en Deep Learning que permite la identificación precisa de cáncer de próstata a través de imágenes histológicas.

Este proyecto se fundamenta en técnicas de Deep Learning, específicamente en el uso de redes neuronales convolucionales, que han mostrado un rendimiento superior en la clasificación de imágenes médicas. Los modelos utilizados incluyen arquitecturas avanzadas como VGG16, VGG19 y ResNet50, siendo el modelo VGG19 el seleccionado debido a su alta precisión en la detección de imágenes con cáncer y sin cáncer.

La metodología aplicada corresponde a una investigación aplicada con un enfoque cuantitativo. La población estuvo compuesta por un conjunto de imágenes de histología de próstata, y la muestra final utilizada para el entrenamiento del sistema fue de 13,827 imágenes, divididas en dos categorías: con cáncer y sin cáncer. Para la recolección de información se emplearon técnicas de análisis documental y observación.

Como resultado, el sistema inteligente logró una precisión del 94.11% utilizando el modelo VGG19, lo cual representa un avance significativo en la capacidad de detección de cáncer de próstata a través de herramientas automatizadas. Este sistema tiene el potencial de convertirse en un soporte fundamental para los profesionales de la salud, optimizando los procesos de diagnóstico y contribuyendo a una atención más efectiva y oportuna.

Palabras clave: Aprendizaje Profundo, cáncer de próstata, redes neuronales convolucionales, diagnóstico médico, sistema inteligente.

II. INTRODUCCIÓN

El cáncer representa una de las mayores preocupaciones de salud a nivel mundial, siendo responsable de millones de muertes cada año. Entre los tipos de cáncer más prevalentes se encuentra el cáncer de próstata, que afecta principalmente a la población masculina y puede tener un impacto significativo en la calidad de vida y la supervivencia de los pacientes. En el contexto peruano, el cáncer es la principal causa de mortalidad por grupo de enfermedad para el género masculino, lo que genera un alto impacto económico y social debido a su diagnóstico tardío y las limitaciones en el acceso a cuidados que se requieran.

Sin embargo, la detección temprana sigue siendo un desafío importante en el manejo de esta enfermedad, debido principalmente a la falta de acceso a pruebas diagnósticas oportunas, especialmente en regiones con recursos limitados, ya que los síntomas iniciales pueden ser inespecíficos o asintomáticos, lo cual lleva a muchos pacientes a consultar en etapas avanzadas. Estos factores resaltan la necesidad de implementar tecnologías que puedan mejorar la capacidad de detección precoz. En este contexto, la presente investigación busca desarrollar un sistema de inteligencia artificial basado en redes neuronales convolucionales, con el objetivo de anticipar de manera eficiente el diagnóstico de cáncer de próstata.

Los objetivos de esta investigación son los siguientes:

Objetivo general:

Desarrollar un sistema inteligente basado en Deep Learning para predecir el diagnóstico de cáncer de próstata.

Objetivos específicos:

Analizar y preprocesar imágenes de cáncer de próstata para crear un conjunto de datos representativo para el modelo.

Desarrollar y optimizar un modelo de Deep Learning adecuado para la clasificación de imágenes médicas que permita diferenciar entre muestras con cáncer y sin cáncer.

Implementar un sistema de predicción que integra el modelo entrenado en una interfaz de usuario accesible para profesionales de la salud.

Evaluar el rendimiento del sistema mediante métricas de precisión, sensibilidad y especificidad para validar su eficacia en la detección de cáncer de próstata.

El informe se organiza en los siguientes capítulos:

- Capítulo I introduce el problema central de la investigación, así como los objetivos y la relevancia del estudio.
- El Capítulo II presenta el diseño teórico, abordando antecedentes y bases teóricas fundamentales relacionadas con el cáncer de próstata y las técnicas de Deep Learning empleadas en diagnóstico médico.
- El Capítulo III describe la metodología empleada en el desarrollo del sistema inteligente, detallando los procedimientos de recolección y procesamiento de datos.
- El Capítulo IV expone los resultados obtenidos, incluyendo el análisis de desempeño de los modelos de Deep Learning aplicados y su precisión diagnóstica.
- El Capítulo V analiza y discute los resultados en el contexto de estudios previos y la literatura revisada.
- El Capítulo VI ofrece las conclusiones de la investigación y proporciona recomendaciones

para futuras investigaciones y posibles mejoras en la implementación clínica del sistema.

En resumen, esta investigación busca contribuir a mejorar la detección temprana del cáncer de próstata mediante la implementación de tecnologías de inteligencia artificial, lo que podría tener un impacto significativo en la atención y supervivencia de los pacientes afectados por esta enfermedad.

III. DISEÑO TEÓRICO

3.1. Antecedentes

Guajin y Kup-Sze (2018) desarrollaron un sistema para diagnosticar cáncer de próstata en una población china. Para ello, obtuvieron una base de datos de biopsias de próstata guiadas por ecografía transrectal (TRUS) en un hospital de Hong Kong, que incluía 1,625 registros de pacientes chinos. Se emplearon cuatro métodos de aprendizaje automático: Support Vector Machine (SVM), Least Squares Support Vector Machine (LS-SVM), Red Neuronal Artificial (ANN) y Random Forest (RF). ANN se destacó al alcanzar la mayor precisión con un 95.27%.

Faez (2018) utilizó datos recopilados del hospital Imam Reza (Teherán). Este conjunto de datos es simplemente acerca de quiénes dieron positivo los resultados de sus pruebas y si sufrían de cáncer de próstata. Aplicaron el uso de Deep Learning y tres ANN para mejorar el diagnóstico respectivo, Deep Learning logró una precisión de clasificación del 86,3 % siendo esta la más cercana al resultado deseado.

Lal et al. (2019) desarrollaron una aplicación que relaciona características morfológicas del cáncer de próstata con imágenes de resonancia magnética (MRI). Utilizaron redes bayesianas, donde nodos y arcos muestran una relación de causa y efecto a través de gráficos acíclicos dirigidos (DAG). Gracias a la asociación entre área y equidiámetro, la aplicación obtuvo una probabilidad de clasificación del 98.94%.

Forero Cuellar (2019) utilizó un algoritmo de aprendizaje en datos "Prueba y puntuación" del programa informático "Orange Data Mining Fruit and Fun" para realizar minería de datos y análisis predictivo. El proceso de diagnóstico es llevado a cabo por un médico experto que evalúa las variables apoyándose en imágenes de resonancia magnética y otras técnicas tradicionales, lo que a menudo resulta en diagnósticos difíciles y demorados. La novedad de este enfoque radica en

la capacidad de obtener un diagnóstico más rápido y preciso como complemento al diagnóstico clínico. Se encontró que la regresión logística y la red neuronal tienen un rendimiento superior al de los bosques aleatorios y los árboles de decisión para la clasificación de las categorías de evaluación PI-RADS, con una precisión del 83.1% para la regresión logística y del 79.5% para los árboles de decisión.

Yasser et al. (2019) buscaron predecir fármacos que interactúen con los receptores de adenosina, los cuales inhiben el crecimiento de células tumorales, incluido el carcinoma de próstata. Utilizaron algoritmos Random Forest (RF) y árboles de decisión (DT) con un conjunto de datos de medicamentos compilado por Saad et al. La combinación de datos permitió generar huellas dactilares moleculares (bits binarios), y RF obtuvo la mayor precisión en la clasificación de fármacos con un 75.09% en la interacción con el receptor de adenosina A2 a 75,09%.

Mohamed et al. (2020) aplicaron la técnica de regresión logística modificada (MLR) para identificar cáncer de próstata a partir de un conjunto de datos de Zhou W. Para la validación, se emplearon imágenes de resonancia magnética multiparamétrica (mpMRI), logrando un nivel de sensibilidad (Se) del 96% y especificidad (Sp) del 98%, mejorando la precisión del clasificador en un 3% comparado con métodos previos.

Carneiro Neto (2020) utilizó una arquitectura de red neuronal profunda, XmasNet, para realizar dos grandes experimentos sobre la clasificación de lesiones las imágenes se obtuvieron de un conjunto de datos IMPROD (ImprovedProstate Cancer Diagnosis - Combination of Magnetic Resonance Imaging and Biomarkers) se originó a partir de un ensayo clínico llevado a cabo por el Hospital Universitario de Turku y la Universidad de Turku. El ensayo incluyó a 175 pacientes entre 40 y 85 años con sospechas de cáncer de próstata respaldadas por resultados de detección. se obtuvo por resultado que la clasificación binaria del significado clínico de las lesiones y la detección

y segmentación de la próstata consiguen los resultados deseados 0,870 AUC y 0,915 dice score respectivamente.

Bese y Miró (2022) desarrollaron un sistema de aprendizaje profundo para la gradación del cáncer de próstata. Utilizaron anotaciones de personas no expertas en combinación con redes neuronales convolucionales (CNN) para clasificar imágenes médicas, logrando mejorar la precisión en la gradación del cáncer, especialmente útil en entornos con recursos limitados.

La combinación de anotaciones no expertas con la capacidad de las CNN para extraer características profundas de las imágenes permite crear un sistema eficiente y accesible para su implementación en entornos clínicos. El estudio muestra que, a través de este enfoque, es posible obtener resultados precisos y consistentes, destacando el potencial de las tecnologías de inteligencia artificial para apoyar y optimizar los procesos médicos, especialmente en áreas donde la disponibilidad de expertos es limitada.

Moral (2021) se centra en el desarrollo de un sistema automatizado para la gradación del cáncer de próstata. Este sistema utiliza el índice PI-RADS (Prostate Imaging Reporting and Data System) junto con técnicas avanzadas de aprendizaje profundo para analizar imágenes de resonancia magnética. El objetivo principal es mejorar la precisión y eficiencia en la gradación del cáncer de próstata, facilitando el diagnóstico y la toma de decisiones clínicas. El sistema desarrollado incorpora algoritmos de deep learning que permiten extraer y analizar características complejas de las imágenes médicas, ofreciendo una evaluación detallada y objetiva del estado de la enfermedad. Los resultados del estudio evidencian que el uso de técnicas de deep learning mejora significativamente la precisión en la gradación del cáncer de próstata, combinado con el índice PI-RADS, puede ofrecer una herramienta robusta y confiable para la gradación del cáncer de próstata. Esta aproximación tiene el potencial de optimizar el trabajo de los radiólogos y mejorar

los resultados clínicos mediante diagnósticos más precisos y rápidos.

Perez (2020) examina el desarrollo de un sistema innovador que integra anotaciones de personas no expertas con redes neuronales convolucionales (CNN) para la clasificación del cáncer de próstata. El objetivo principal del estudio es aumentar la precisión y eficiencia en la gradación del cáncer de próstata mediante el empleo de tecnologías de inteligencia artificial. Las anotaciones no expertas proporcionan un enfoque accesible y económico para la recopilación de datos, mientras que las CNN permiten una extracción y análisis detallado de las características de las imágenes médicas. Los resultados del estudio demuestran que este enfoque puede ofrecer una herramienta efectiva para apoyar el trabajo de los patólogos, proporcionando diagnósticos más rápidos y precisos. Además, la investigación resalta el potencial de las técnicas de deep learning para transformar los procesos médicos, especialmente en áreas con recursos limitados.

Mena (2023) tiene como objetivo principal diseñar y desarrollar un sistema de soporte a la decisión clínica (CDSS) que prediga el cáncer de próstata (CP) a partir de la expresión génica en tejidos de pacientes con CP y controles sanos. Se llevó a cabo una selección de genes relevantes para el CP utilizando datos de The Cancer Genome Atlas y técnicas de inteligencia artificial explicable para crear modelos comprensibles. Los modelos desarrollados mostraron un buen rendimiento en diversas métricas de calidad y fueron validados en poblaciones externas, logrando altos valores de sensibilidad y especificidad. Se obtuvieron explicaciones aditivas de Shapley del mejor modelo para ayudar a los profesionales clínicos a entender las decisiones del sistema. Estas explicaciones destacaron genes tanto conocidos como nuevos asociados con el cáncer de próstata (CP), ofreciendo una base sólida para su aplicación clínica. La validación adicional en muestras locales confirmó la viabilidad del CDSS para mejorar el diagnóstico y tratamiento del CP.

3.2. Bases Teórica

3.2.1. Cáncer de Próstata

El cáncer de próstata, en particular el adenocarcinoma, es relevante en este trabajo debido a su alta prevalencia en hombres mayores de 50 años y su desarrollo común en la zona periférica de la próstata. Los avances en las técnicas de diagnóstico, como las imágenes médicas, han mejorado la detección temprana, lo cual es fundamental para aumentar las tasas de supervivencia. Este proyecto se enfoca en el uso de técnicas de deep learning para asistir en la gradación de imágenes de resonancia magnética, contribuyendo a mejorar la precisión del diagnóstico en etapas tempranas del cáncer de próstata, lo que permite un tratamiento más oportuno y eficaz. Además, la edad, raza y genética son factores clave a tener en cuenta en la predicción y diagnóstico, que son abordados en el sistema desarrollado.

3.2.2. Factores de riesgo del cáncer de próstata

Según Yanes Chacón, Villalobos Campos y Cubas González (2023), entre los factores de riesgo que pueden provocar cáncer de próstata se incluyen la presencia de la enfermedad en estadios clínicamente avanzados, lo que lleva a un alto porcentaje de pacientes a consultar a un especialista. Esto representa un desafío tanto de salud como económico debido a los altos costos del tratamiento. Los factores pronósticos más utilizados son la cantidad y la ubicación de metástasis óseas, así como las metástasis viscerales, la escalada de Gleason, el antígeno prostático específico y la fosfatasa alcalina. Aunque no se dispone de métodos para prevenir el cáncer de próstata, se sugieren cambios en la dieta y el estilo de vida como medidas preventivas.

Otros factores de riesgo para el cáncer de próstata incluyen un alto consumo de grasas saturadas, una ingesta baja de carotenoides, la realización de una vasectomía y la frecuencia de

relaciones sexuales, entre otros; no obstante, los resultados entre estudios son contradictorios. (Báez Benavides & Armisen Yañez, 2012)

3.2.3. Diagnóstico del cáncer de próstata

Según Delgado (2016), el tacto rectal, la concentración de antígeno prostático específico (APE) y el ultrasonido transrectal son herramientas diagnósticas valiosas para detectar posibles indicios de cáncer de próstata; Sin embargo, el diagnóstico definitivo solo se puede confirmar con una biopsia. La mayoría de los tumores prostáticos se localizan en la zona periférica de la próstata y suelen ser detectables mediante tacto cuando tienen un diámetro superior a 0,2 cm. El APE, una serina proteasa relacionada con la calicreína producida en las células epiteliales de la próstata, tiene la función de licuar el semen. La medición de APE en sangre es crucial para el diagnóstico del cáncer de próstata. En casos donde el APE esté entre 4 ng/ml y 10 ng/ml, se utiliza la medición del APE libre y en complejos para decidir si se requiere una biopsia. El riesgo de cáncer es menor al 10% cuando el APE es superior a 25 ng/ml, pero puede aumentar hasta el 56% si el APE libre es menor al 10%.

3.2.4. Agentes Inteligentes.

Según Gamboa (2014), Estos sistemas están diseñados para percibir su entorno, analizarlo y actuar de manera que optimicen sus posibilidades de éxito. Este enfoque permite a los investigadores abordar problemas complejos y encontrar soluciones que sean tanto útiles como comprobables. La tecnología de agentes inteligentes constituye la base de una nueva generación de sistemas computacionales. Sus aplicaciones abarcan desde la búsqueda de información en Internet distribuida masivamente hasta sistemas móviles de información, sistemas de flujo de trabajo inteligentes e infraestructura de apoyo a operaciones y decisiones corporativas.

3.2.5. Sistema inteligente.

Según Gamboa (2014), Se denomina sistema inteligente a una entidad capaz de percibir, razonar, aprender, adaptarse, tomar decisiones y actuar racionalmente para satisfacer sus metas, en un entorno determinado. Son parte de las ciencias de la computación, que cubren una serie de temas teóricos-experimentales que sirven de base para la ingeniería de aplicaciones inteligentes.

3.2.6. Deep Learning.

Según IBM (2020), El Deep Learning es una subdisciplina del Machine Learning, el cual es una parte de la inteligencia artificial. En esta técnica, las redes neuronales—modelos computacionales que imitan el funcionamiento del cerebro humano—analizan grandes cantidades de datos para aprender y perfeccionarse. Los algoritmos de Deep Learning llevan a cabo tareas repetitivas que contribuyen a mejorar gradualmente los resultados mediante "capas profundas" o "deep Layers", facilitando así el aprendizaje progresivo. Este enfoque forma parte de un conjunto más amplio de métodos de Machine Learning basados en redes neuronales.

3.2.7. Modelos de Deep Learning

Según Sandoval (2021), los modelos lineales buscan identificar una línea que se ajuste de manera adecuada a un conjunto de datos representados en una nube de puntos. Entre estos modelos se incluyen técnicas bien establecidas como la regresión lineal, también conocida como regresión de mínimos cuadrados, y la regresión logística, que adapta la metodología lineal para abordar problemas de clasificación. Por otro lado, los modelos de árbol se destacan por su precisión, estabilidad y facilidad de interpretación, debido a que construyen reglas de decisión que se pueden visualizar en forma de un árbol, lo cual facilita su comprensión. Finalmente, las redes neuronales intentan emular el funcionamiento del cerebro humano, donde millones de neuronas están

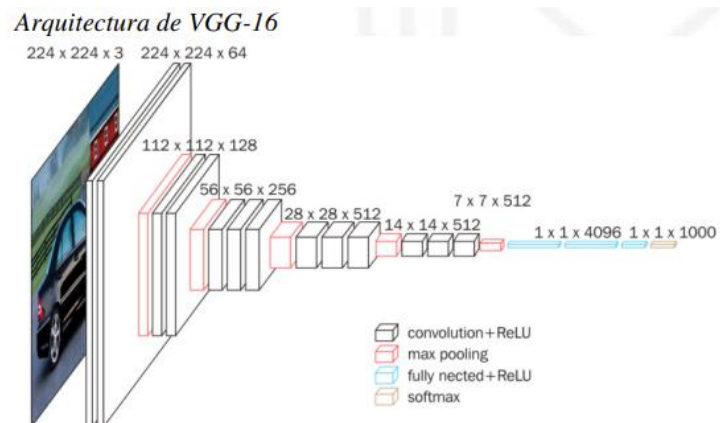
interconectadas en una red para intercambiar información entre sí.

Este enfoque se basa en el trabajo de Silva et al. (2020) sobre sistemas automáticos de clasificación y detección en histología de próstata, que utilizan la escala de puntuación de Gleason.

3.2.8. Modelo VGG-16

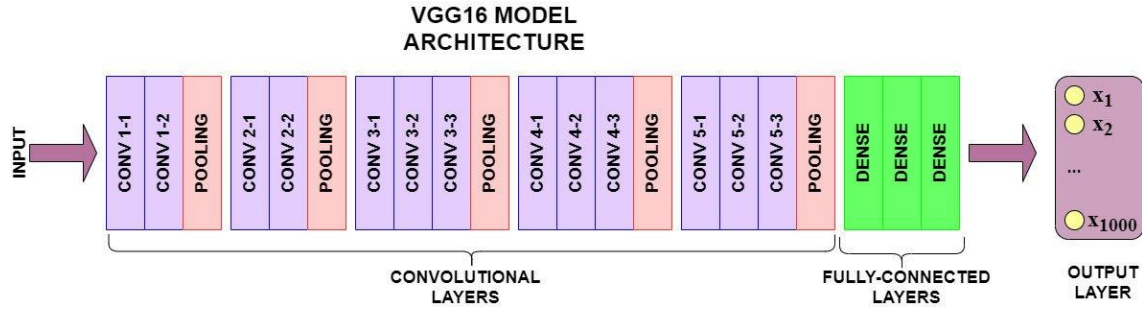
Quispe (2020), explica que esta arquitectura está compuesta por 13 capas convolucionales, cada una seguida de una capa de *max pooling*. Después de esto, hay 3 capas totalmente conectadas, lo que justifica su nombre (VGG-16), y finalmente se encuentra un clasificador *softmax*. Esta red se distingue por tener numerosos parámetros para entrenar, lo que puede prolongar el tiempo de entrenamiento en comparación con otras redes. Sin embargo, cuenta con la ventaja de disponer de pesos preentrenados del conjunto de datos *Imagenet*, los cuales pueden utilizarse para iniciar el entrenamiento de la red propuesta, como se menciona.

Figura 1: Arquitectura de VGG-16 separada por 16 capas convolucionales



Nota: Obtenido y modificado de Quispe (2020)

Figura 2: Modelo lógico de Arquitectura VGG-16 con 13 capas convolucionales y 3 densas

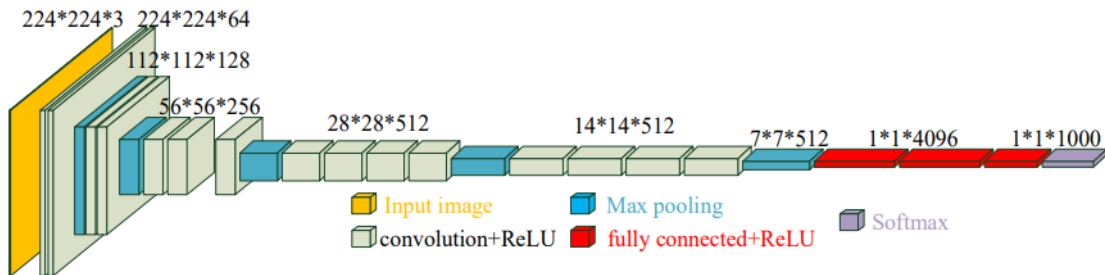


Nota: Obtenido y modificado de Quispe (2020)

3.2.9. Modelo VGG-19

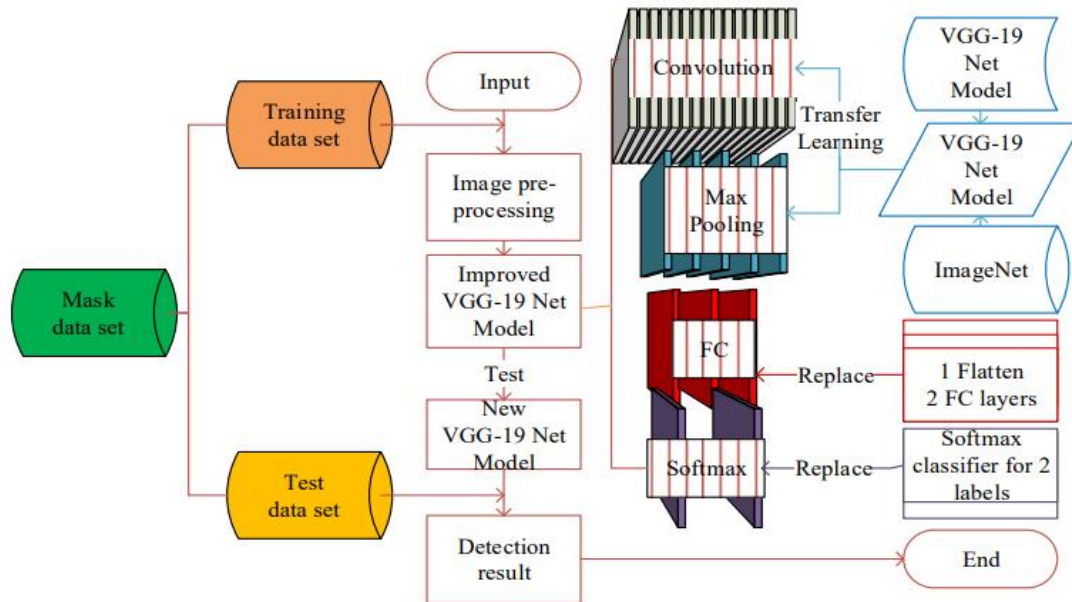
Según Jian, Jia, Shaozhong y Bilong (2020), este modelo tiene seis estructuras principales, cada una de las cuales se compone principalmente de múltiples capas convolucionales y capas completamente conectadas. El tamaño del núcleo convolucional es 3×3 , y el tamaño de entrada es $224 \times 224 \times 3$. El número de capas generalmente se concentra en 16 y 19.

Figura 3: Modelo de red VGG-19



Nota: Obtenido y modificado de Jian, Jia, Shaozhong y Bilong (2020)

Figura 4: Marco de entrenamiento del modelo de detección de uso de máscara



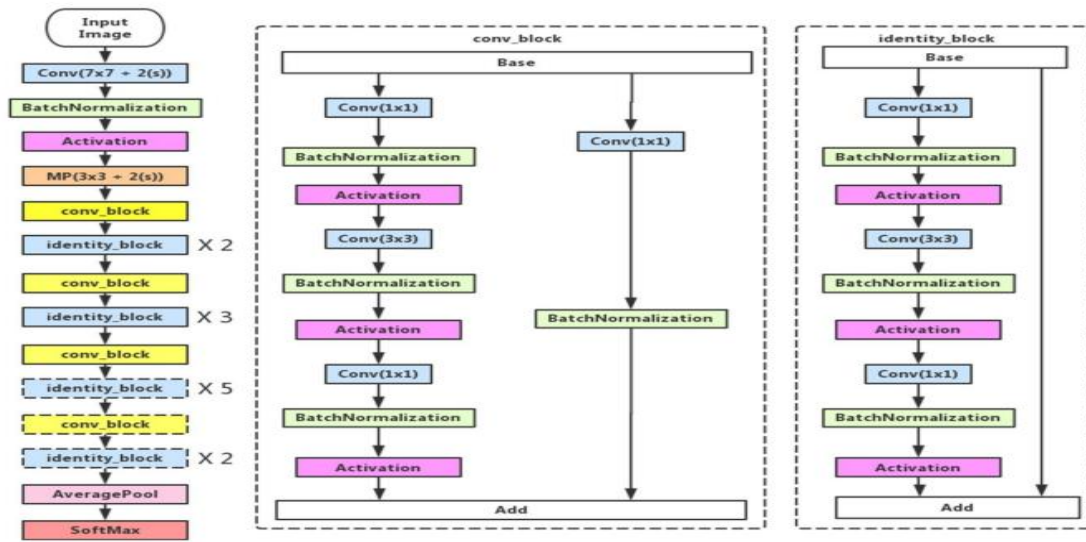
Nota: Obtenido y modificado de Jian, Jia, Shaozhong y Bilong (2020)

Utiliza una estructura alterna de múltiples capas convolucionales y capas de activación no lineales, lo cual es más efectivo que una sola convolución. Esta estructura de capas permite extraer mejores las características de la imagen, utilizando *Maxpooling* para reducir la resolución y la función de activación lineal (*ReLU*) para seleccionar el valor más grande en el área de la imagen como valor conjunto del área. La capa de reducción de resolución se emplea principalmente para aumentar la capacidad de la red para resistir distorsiones en la imagen, conservando las características esenciales de la muestra y disminuyendo el número de parámetros.

3.2.10. Modelo ResNet50

Según Nishant y Shrivastava (2021), ResNet-50 es un modelo preentrenado que ha sido entrenado en el conjunto de datos *ImageNet* para identificar diferentes imágenes de 1000 clases. El tamaño de la imagen de entrada para la red neuronal es de 224 x 224 píxeles.

Figura 5: Arquitectura de ResNet50



Nota: Obtenido y modificado de Qingge, He, Yankui y Jie (2019)

Gracias a esta capa, el modelo puede extraer eficazmente patrones complejos y relevantes y resolver los problemas. problema de gradiente de desaparición. La salida del ResNet50 se pasa a través de una capa *softmax* para generar probabilidades para cada clase, permitiendo la clasificación final de las imágenes.

IV. DISEÑO METODOLÓGICO

4.1. Tipificación de la investigación

- **Enfoque: Cuantitativa**

La presente investigación es de enfoque **cuantitativo** porque se basa en la recolección y análisis de datos numéricos. El rendimiento del sistema de diagnóstico se evalúa mediante métricas cuantificables como la precisión, sensibilidad y especificidad. El uso de datos numéricos permite realizar mediciones objetivas del desempeño del modelo de Deep Learning, ofreciendo una base sólida para la interpretación de los resultados en el diagnóstico del cáncer de próstata.

- **Diseño: No experimental**

La investigación es **no experimental**, dado que no se manipulan directamente las variables bajo estudio. En este caso, las imágenes médicas y los datos utilizados para entrenar y evaluar el modelo de Deep Learning ya están disponibles y no se alteran de forma intencional. El estudio se enfoca en analizar el comportamiento del modelo frente a los datos preexistentes sin intervención directa en las condiciones de la muestra o en las variables involucradas. No se aplican tratamientos o modificaciones, sino que se observa y evalúa cómo el sistema se desempeña bajo condiciones naturales.

- **Alcance: Exploratorio**

El **alcance de la investigación** es **exploratorio**, ya que se busca investigar un área que aún no ha sido completamente abordada en la literatura, en este caso, la aplicación de técnicas de Deep Learning para la detección del cáncer de próstata.

El propósito es explorar y probar la viabilidad y efectividad de este enfoque en el diagnóstico médico. Aunque existen estudios previos sobre Deep Learning en medicina, la exploración de su aplicación específica para esta enfermedad con el uso del modelo VGG19 constituye un terreno novedoso, lo cual justifica este enfoque exploratorio.

- **Tipo: Aplicada**

La investigación es de carácter **aplicado**, ya que se enfoca en resolver un problema práctico en el campo de la salud: mejorar el diagnóstico temprano del cáncer de próstata mediante el desarrollo de un sistema inteligente basado en Deep Learning. A diferencia de la investigación básica, cuyo fin es generar nuevo conocimiento teórico, este estudio busca aplicar técnicas avanzadas de aprendizaje profundo a un contexto clínico específico, con la intención de tener un impacto tangible en la práctica médica y en la mejora de los servicios de salud.

Para validar la hipótesis planteada y medir el desempeño del sistema desarrollado, se utilizó un **método de análisis estadístico** basado en el cálculo de métricas clave: **precisión, sensibilidad, especificidad y exactitud**. Estas métricas permiten evaluar de manera objetiva cuán eficaz es el modelo de Deep Learning en la identificación del cáncer de próstata a partir de imágenes histológicas.

Las **matrices de confusión** se emplearon para visualizar el rendimiento del modelo, ya que proporcionan una representación detallada de las predicciones correctas e incorrectas para cada clase (imágenes con cáncer y sin cáncer). Esto, a su vez, permitió el cálculo de las métricas mencionadas, proporcionando una visión más clara del comportamiento del sistema en la detección

temprana del cáncer de próstata, tanto en términos generales como específicos para cada categoría. Este enfoque integral asegura una evaluación exhaustiva del modelo, facilitando la identificación de sus fortalezas y posibles áreas de mejora.

4.2. Tabla de operacionalización de variable

Tabla 1: Tabla operacional según Indicadores

Variable	Dimensión	Indicadores
Rendimiento	Eficiencia del sistema inteligente basado en Deep Learning	Exactitud
		Precisión
		Sensibilidad
		Especificidad

Nota: Tabla elaborada por los autores en base a los tipos de variables del modelo y sus Indicadores

- **Verdaderos Positivos (TP):** Se refiere a los casos en los que la verdadera categoría del dato es 1 (positivo) y el modelo también lo clasifica como 1 (positivo).
- **Verdaderos Negativos (TN):** Se refiere a los casos en los que la verdadera categoría del dato es 0 (negativo) y el modelo también lo clasifica como 0 (negativo).
- **Falsos Positivos (FP):** Ocurre cuando el dato tiene una verdadera categoría de 0 (negativo), pero el modelo lo clasifica incorrectamente como 1 (positivo).
- **Falsos Negativos (FN):** Se presenta cuando el dato tiene una verdadera categoría de 1 (positivo), pero el modelo lo clasifica incorrectamente como 0 (negativo).

Se definen los siguientes indicadores:

Exactitud. La exactitud, también conocida como *accuracy*, representa la proporción de predicciones correctas (incluyendo tanto los verdaderos positivos como los verdaderos negativos) en relación con el total de casos analizados. Este indicador se calcula

utilizando la siguiente fórmula.

$$Exactitud = \frac{VP + VN}{VP + FP + FN + VN}$$

Precisión: La precisión se determina como la proporción de verdaderos positivos en comparación con el total de resultados clasificados como positivos. Esta métrica se calcula mediante la siguiente fórmula.

$$Precisión = \frac{VP}{VP + FP}$$

La sensibilidad. La sensibilidad, también conocida como recall, evalúa la proporción de verdaderos positivos que son identificados correctamente por el modelo.

$$Sensibilidad = \frac{VP}{VP + FN}$$

La especificidad (Specificity). La especificidad refleja la capacidad de una prueba para identificar correctamente los resultados negativos. Evalúa la proporción de verdaderos negativos que han sido identificados con precisión.

$$Especificidad = \frac{VN}{VN + FP}$$

Los indicadores se fundamentan en el libro “Importancia del cálculo de la sensibilidad, la especificidad y otros parámetros estadísticos en el uso de las pruebas de diagnóstico clínico y de laboratorio” (Vizcaíno-Salazar, 2017). La evaluación de la usabilidad se basará en los criterios establecidos por la norma ISO25010. (iso25000, 2022).

4.3. Población y muestra

La muestra está compuesta por 13,827 imágenes agrupadas en cinco estratos, cada uno correspondiente a un grado de la clasificación de Gleason. La proporción de imágenes por estrato fue la siguiente: grado 1 (10%), grado 2 (15%), grado 3 (25%), grado 4 (25%) y grado 5 (25%). Esta distribución fue obtenida de la base de datos SICAPv2 (Rodríguez J. S., Mendeley Data, 2020).

Para el procesamiento de imágenes, todas las muestras pasaron por un preprocesamiento estándar que incluyó normalización y aumento de datos. Sin embargo, las imágenes de los grados más bajos (1 y 2) recibieron un tratamiento especial que incluyó técnicas de realce de bordes para facilitar la detección de sutiles diferencias en las células, lo que es crucial para la detección temprana.

Por otro lado, las imágenes de los grados más altos (3, 4 y 5) se trataron con técnicas de reducción de ruido para optimizar la identificación de patrones.

4.4. Técnicas, instrumentos, equipos y materiales

Para la recolección de datos en esta investigación se emplearon dos técnicas principales: **análisis documental** y **observación**. Cada técnica se aplicó en momentos específicos y estuvo acompañada de instrumentos que facilitaron el registro y análisis de la información.

Análisis Documental

Propósito: El análisis documental se utilizó para recopilar información teórica fundamental sobre modelos de Deep Learning aplicados al diagnóstico de cáncer de próstata.

Momento de Aplicación: Esta técnica fue empleada en la fase inicial del proyecto, cuando se construía el marco teórico y se fundamentaba la selección de modelos de Deep Learning.

Fuentes de Consulta: Se consultaron artículos científicos, bases de datos de imágenes médicas (como SICAPv2), y manuales técnicos que detallan la implementación de redes neuronales convolucionales.

Instrumento: Para organizar la información obtenida, se utilizó una **ficha de registro documental**, donde se anotaron aspectos clave de cada fuente, como datos sobre precisión y aplicaciones de los modelos en el diagnóstico de cáncer.

Observación

Propósito: La observación fue empleada para evaluar el rendimiento del sistema inteligente durante las pruebas de clasificación de imágenes de cáncer de próstata, permitiendo registrar los resultados del modelo.

Momento de Aplicación: Esta técnica se aplicó en la fase final del proyecto, cuando el modelo estaba listo para ser evaluado con imágenes de prueba.

Unidad de Análisis: La unidad de análisis consistió en las imágenes de histología de próstata usadas para validar el modelo y medir su eficacia.

Instrumento: Se utilizó una **ficha de observación** para registrar métricas clave como precisión, sensibilidad, especificidad y exactitud del modelo en cada prueba, documentando aciertos y errores en la clasificación de las imágenes.

Equipos y Materiales

Para el desarrollo y ejecución de la investigación, se utilizaron los siguientes equipos y materiales:

Equipos de Cómputo: Dos computadoras portátiles, una con un procesador Intel Core i7 3.20 GHz y 16 GB de RAM, y otra con un Intel Core i7 2.80 GHz y 16 GB de RAM. Ambas máquinas fueron utilizadas para entrenar el modelo de Deep Learning y desarrollar el sistema.

Software y Herramientas: Se emplearon Python como lenguaje de programación, el framework Django para el desarrollo del sistema web, MySQL para el almacenamiento de datos y Git/GitHub como sistema de control de versiones para gestionar el proyecto.

Los datos obtenidos a través de las técnicas de recolección fueron registrados y analizados progresivamente, de acuerdo con el avance del proyecto. Los resultados se guardaron en una base de datos MySQL para su posterior análisis y evaluación.

V. RESULTADOS

5.1. Análisis del conjunto de Imágenes

5.1.1. Descripción del conjunto de datos SICAPv2

El conjunto total de datos SICAPv2 (Silva-Rodríguez, 2020), contiene imágenes histológicas completas de próstata, con anotaciones de puntuaciones globales de Gleason y grados de Gleason a nivel de patrón. Está diseñado para estudios de clasificación y detección de patrones en histología prostática, específicamente para la detección de patrones cribiformes y la clasificación automática según la escala de Gleason.

La base de datos presentada en este artículo, SICAPv2, es, hasta donde saben los autores, la colección pública más grande de biopsias de próstata H&E con anotaciones a nivel local de grados de Gleason. SICAPv2 es una extensión de la base de datos introducida en [17] y estará disponible públicamente después de la publicación de este artículo.

Contiene 13,827 imágenes que son divididas entre los 2 estados, entre ellas tenemos: con cáncer de próstata y sin cáncer de próstata. El conjunto de imágenes está dividido en 2 carpetas llamadas train y val. Train contiene las imágenes para entrenamiento las cuales son un total de 10,957 imágenes y val contiene las imágenes para validación con un total de 2870 imágenes.

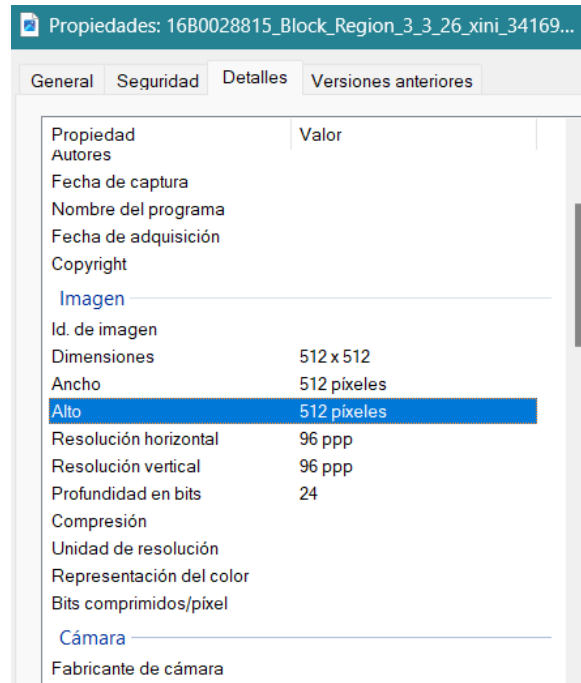
Tabla 2: Distribución de imágenes en dataset

Dataset	Entrenamiento(train)	Validación(val)	Total	Porcentaje
ConCancer	7,367	2,043	9,410	68%
SinCancer	3,590	827	4,417	32%

Nota: Tabla elaborada por los autores en base a la distribución de imágenes para el modelo

Las características principales de cada imagen del dataset son dimensiones de 512 x 512 píxeles, una resolución horizontal y vertical de 96 ppp (píxeles por pulgada), y una profundidad de color de 24 bits. Estos datos indican que la imagen tiene una resolución moderada y una profundidad de color estándar.

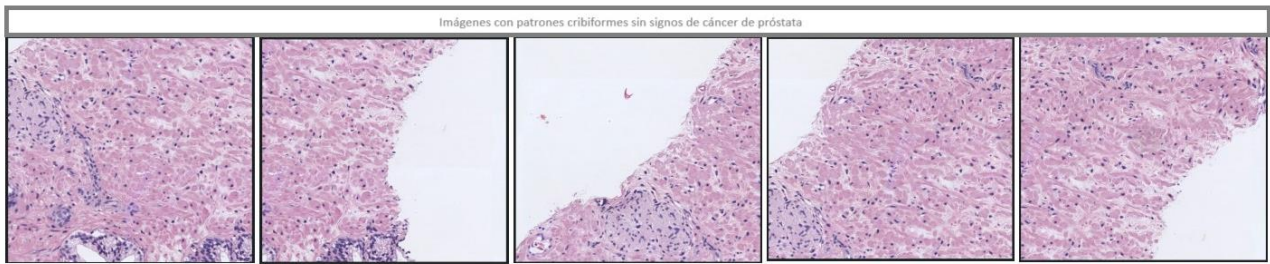
Figura 6: Propiedad de imagen de dataset SICAPv2



Nota: elaborado por los autores en base a las propiedades de una imagen del dataset

Se muestran 5 imágenes sin cáncer de próstata las cuales tienen una consistencia uniforme en las células, sin signos de crecimiento irregular, no se observan áreas de células agrupadas de manera anormal o formando patrones cribiformes.

Figura 7: Imágenes con patrones cribiformes, no contiene signos de cáncer de próstata



Nota: Obtenido y modificado de Rodríguez S. (2020)

Se presentan 5 imágenes con cáncer de próstata, en las cuales se observan características distintivas. Las células muestran un crecimiento desordenado y agrupaciones irregulares en forma de criba, el cual es indicativo de la enfermedad. Además, se observan núcleos agrandados y patrones cribiformes, señales claras de malignidad evidenciando la gravedad del cáncer.

Figura 8: Imágenes con patrones cribiformes contiene signos de cáncer de próstata



Nota: Obtenido y modificado de Rodríguez S. (2020)

5.2. Procesamiento de Datos

Los datos procesados incluyen una amplia gama de imágenes de estudios de cáncer de próstata para la detección de la enfermedad.

El conjunto de datos se divide en dos partes: entrenamiento y validación, ambas partes

contiene 2 subcarpetas las cuales están divididas en imágenes con cáncer y sin cáncer respectivamente.

Definición de variables y parámetros

Primero, se especifica la ubicación de las imágenes de entrenamiento y las categorías que se clasificarán: "ConCancer" y "SinCancer". También se determina el tamaño del lote (batch size) para el entrenamiento del modelo y las dimensiones a las que se redimensionarán las imágenes, que en este caso es de 512x512 píxeles, en nuestro caso las imágenes originales no se verán afectadas ya que se trabajará con su tamaño original.

Carga y procesamiento de imágenes

Se crea una lista vacía que almacenará las imágenes que han sido cargados para ser analizadas junto con sus etiquetas. La función se encarga de recorrer cada categoría y procesar todas las imágenes dentro de cada categoría. Para cada imagen, se realiza lo siguiente:

- Se lee la imagen desde el disco.
- Se redimensiona la imagen a las dimensiones especificadas (512x512 píxeles).
- Se almacena la imagen redimensionada y su etiqueta (0 para "ConCancer" y 1 para "SinCancer") en la lista.

Esta función se ejecuta para cargar y procesar todas las imágenes del conjunto de datos.

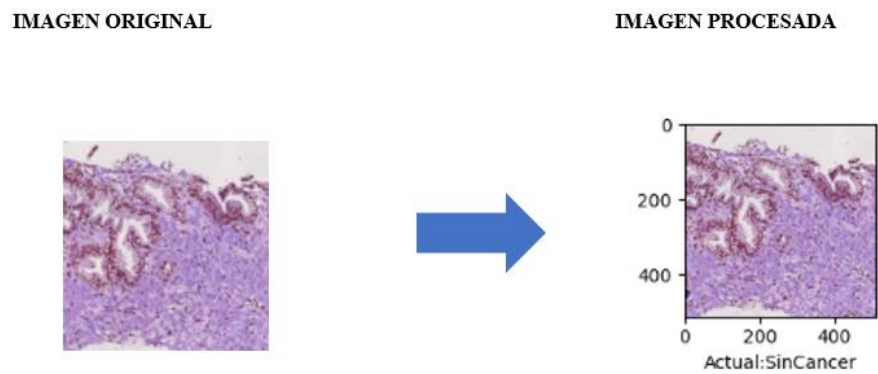
Luego de procesar las imágenes, se le asignaron las etiquetas correspondientes como se ve en las siguientes imágenes:

Figura 9: Procesamiento de una imagen con cáncer



Nota: elaborado por los autores en base al procesamiento de una imagen con cáncer

Figura 10: Procesamiento de una imagen sin cáncer



Nota: elaborado por los autores en base al procesamiento de una imagen sin cáncer

En el procesamiento la imagen original no es redimensionada ya que el parámetro inicial es de 512x512 y se posterior a ello se procede a clasificar las imágenes con una etiqueta ya sea “SinCancer” o “ConCancer”.

5.3. Seleccionar un modelo óptimo de Deep Learning

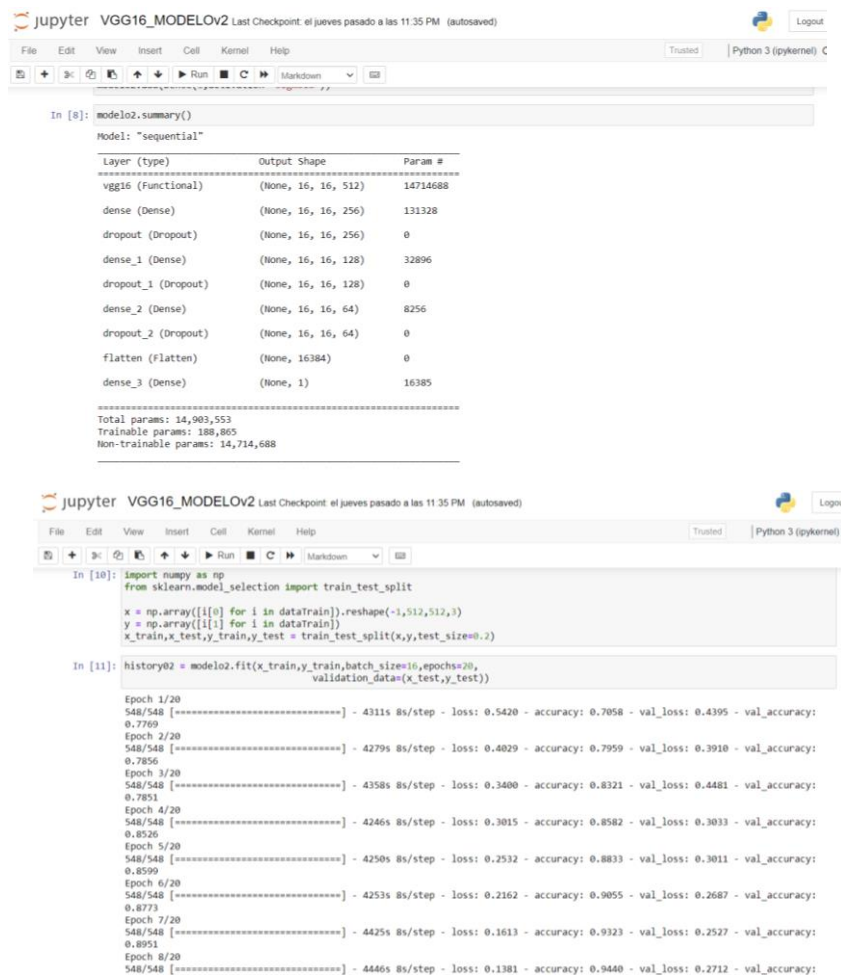
Se evaluaron 3 algoritmos de deep Learning que son VGG16, VGG19 y ResNet50 que se describen a continuación:

5.3.1. Resultados de los modelos aplicados

5.3.1.1. Resultado del modelo VGG16

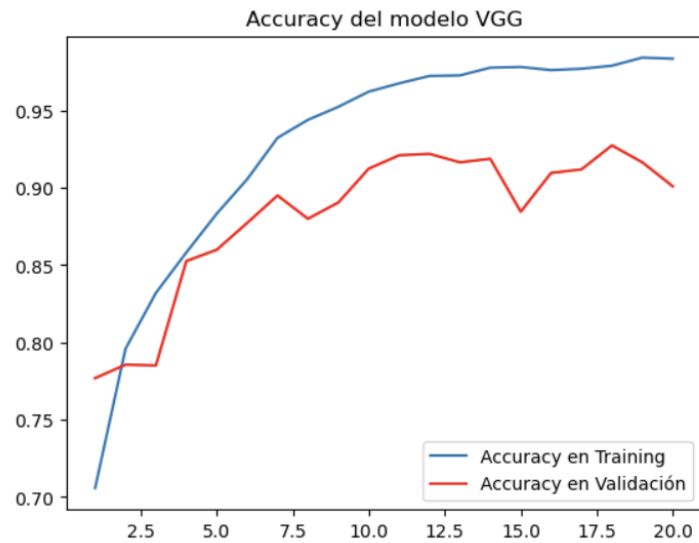
A continuación, se muestran los resultados del siguiente modelo considerando los resultados finales de loss y accuracy respectivamente:

Figura 11: Resultado modelo VGG-16



```
In [12]: loss, accuracy = modelo2.evaluate(x_test,y_test)
print("loss:",loss)
print("Accuracy:",accuracy)
```

```
69/69 [=====] - 845s 12s/step - loss: 0.3553 - accuracy: 0.9010
loss: 0.35527148842811584
Accuracy: 0.9010036587715149
```



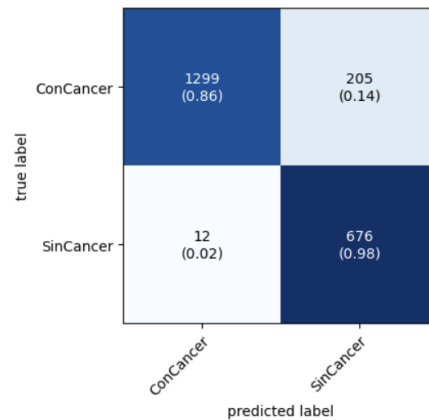
```
In [15]: accuracy_score(y_test,predictions)
```

```
Out[15]: 0.9010036496350365
```

```
In [16]: print(classification_report(y_test,predictions))
```

	precision	recall	f1-score	support
0	0.99	0.86	0.92	1504
1	0.77	0.98	0.86	688
accuracy			0.90	2192
macro avg	0.88	0.92	0.89	2192
weighted avg	0.92	0.90	0.90	2192

```
In [17]: from mlxtend.plotting import plot_confusion_matrix
matrix = confusion_matrix(y_test, predictions)
plot_confusion_matrix(conf_mat = matrix, figsize=(5,4), class_names = ['ConCancer', 'SinCancer'],
                      show_normed = True);
```

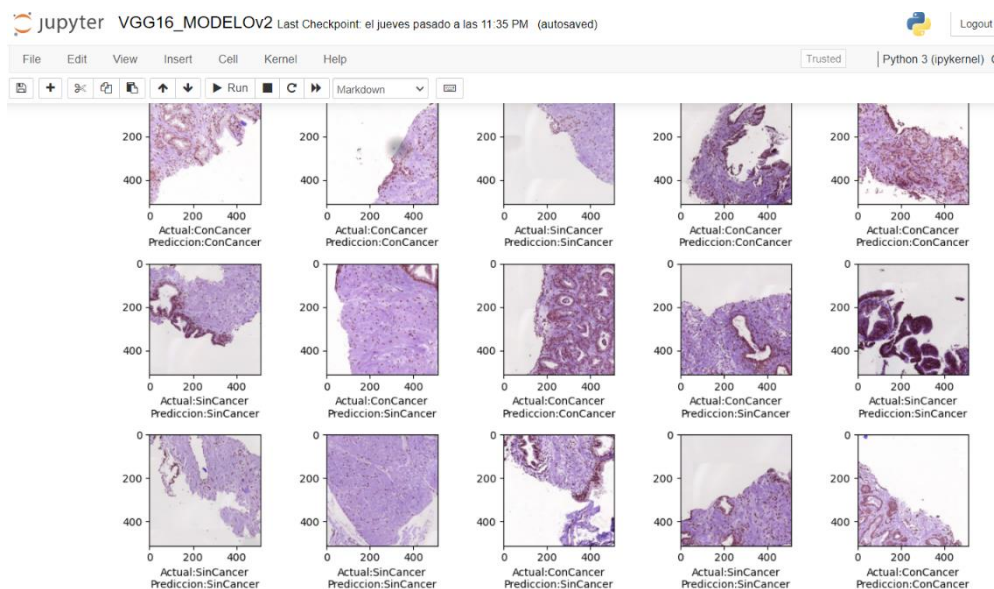


```
In [18]: import random
plt.figure(figsize=(12,7))
for i in range(15):
    sample = random.choice(range(len(x_test)))
    image = x_test[sample]
    category = y_test[sample]
    pred_category = predictions[sample]

    if category == 0:
        label = "ConCancer"
    else:
        label = "SinCancer"

    if pred_category == 0:
        pred_label = "ConCancer"
    else:
        pred_label = "SinCancer"

    plt.subplot(3,5,i+1)
    plt.imshow(image)
    plt.xlabel("Actual:{}\nPrediccion:{}".format(label, pred_label))
plt.tight_layout()
```



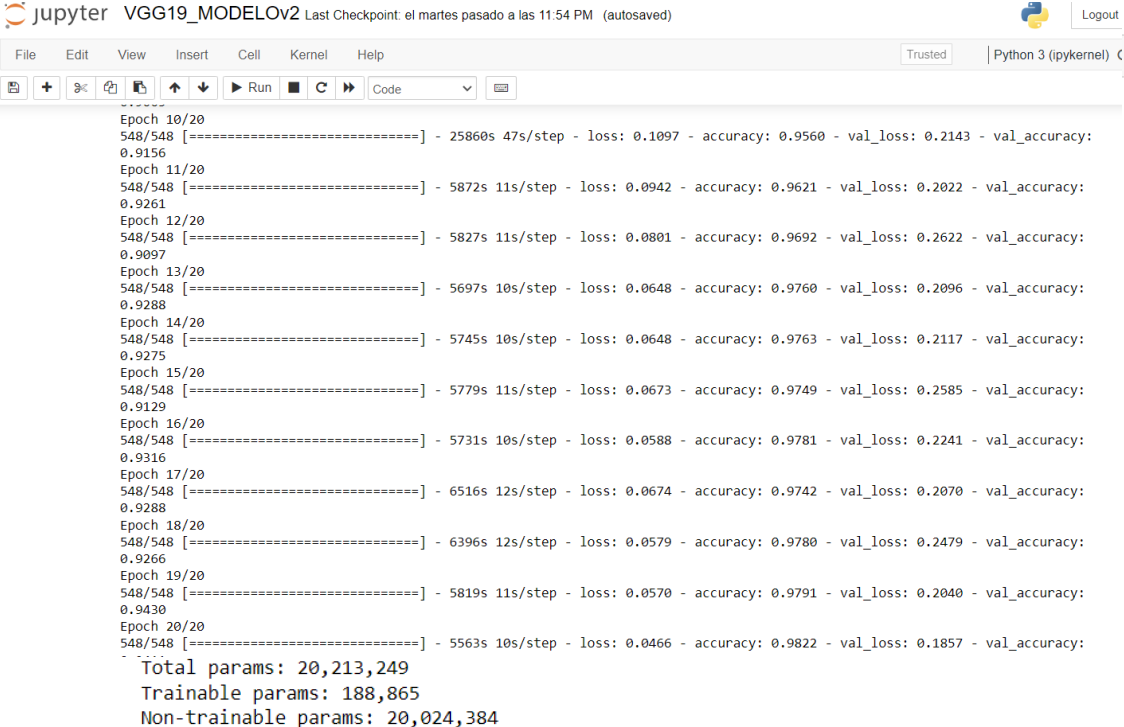
Nota: elaborado por los autores en base al Accuracy del modelo VGG-16 en validación y entrenamiento

Tras entrenar el modelo con el algoritmo VGG16, se observa que la precisión de los datos de entrenamiento se mantiene constante en las últimas épocas, alcanzando un resultado del 98.32%. Por otro lado, la precisión de los datos de validación también se mantiene estable, concluyendo con un resultado del 90.10%.

La función de pérdida para los datos de entrenamiento y los datos de validación en las últimas épocas muestra resultados constantes y poco significativos, por lo que se decidió finalizar la ejecución.

5.3.1.2.Resultado del modelo VGG19

Figura 12: Resultado modelo VGG-19



```
Epoch 10/20
548/548 [=====] - 25860s 47s/step - loss: 0.1097 - accuracy: 0.9560 - val_loss: 0.2143 - val_accuracy:
0.9156
Epoch 11/20
548/548 [=====] - 5872s 11s/step - loss: 0.0942 - accuracy: 0.9621 - val_loss: 0.2022 - val_accuracy:
0.9261
Epoch 12/20
548/548 [=====] - 5827s 11s/step - loss: 0.0801 - accuracy: 0.9692 - val_loss: 0.2622 - val_accuracy:
0.9097
Epoch 13/20
548/548 [=====] - 5697s 10s/step - loss: 0.0648 - accuracy: 0.9760 - val_loss: 0.2096 - val_accuracy:
0.9288
Epoch 14/20
548/548 [=====] - 5745s 10s/step - loss: 0.0648 - accuracy: 0.9763 - val_loss: 0.2117 - val_accuracy:
0.9275
Epoch 15/20
548/548 [=====] - 5779s 11s/step - loss: 0.0673 - accuracy: 0.9749 - val_loss: 0.2585 - val_accuracy:
0.9129
Epoch 16/20
548/548 [=====] - 5731s 10s/step - loss: 0.0588 - accuracy: 0.9781 - val_loss: 0.2241 - val_accuracy:
0.9316
Epoch 17/20
548/548 [=====] - 6516s 12s/step - loss: 0.0674 - accuracy: 0.9742 - val_loss: 0.2070 - val_accuracy:
0.9288
Epoch 18/20
548/548 [=====] - 6396s 12s/step - loss: 0.0579 - accuracy: 0.9780 - val_loss: 0.2479 - val_accuracy:
0.9266
Epoch 19/20
548/548 [=====] - 5819s 11s/step - loss: 0.0570 - accuracy: 0.9791 - val_loss: 0.2040 - val_accuracy:
0.9430
Epoch 20/20
548/548 [=====] - 5563s 10s/step - loss: 0.0466 - accuracy: 0.9822 - val_loss: 0.1857 - val_accuracy:
0.9430
Total params: 20,213,249
Trainable params: 188,865
Non-trainable params: 20,024,384
```

```
jupyter VGG19_MODELOv2 Last Checkpoint: el martes pasado a las 11:54 PM (autosaved) Logout
File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel) C
In [9]: modelo2.compile(optimizer="adam",loss="binary_crossentropy",metrics=["accuracy"])

In [10]: import numpy as np
from sklearn.model_selection import train_test_split

x = np.array([i[0] for i in dataTrain]).reshape(-1,512,512,3)
y = np.array([i[1] for i in dataTrain])
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)

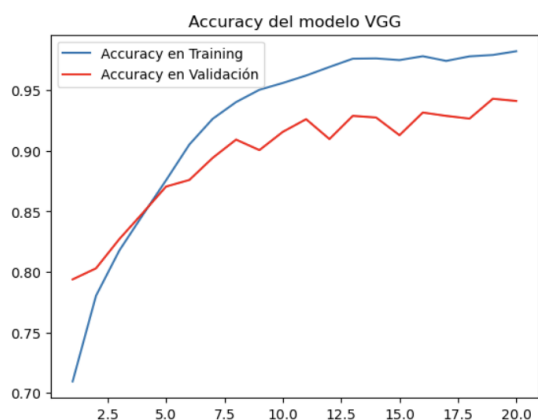
In [11]: history02 = modelo2.fit(x_train,y_train,batch_size=16,epochs=20,
                                validation_data=(x_test,y_test))

Epoch 1/20
548/548 [=====] - 6268s 11s/step - loss: 0.5385 - accuracy: 0.7095 - val_loss: 0.4065 - val_accuracy:
0.7938
Epoch 2/20
548/548 [=====] - 6441s 12s/step - loss: 0.4229 - accuracy: 0.7803 - val_loss: 0.3781 - val_accuracy:
0.8029
Epoch 3/20
548/548 [=====] - 6456s 12s/step - loss: 0.3687 - accuracy: 0.8177 - val_loss: 0.3409 - val_accuracy:
0.8271
Epoch 4/20
548/548 [=====] - 6430s 12s/step - loss: 0.3170 - accuracy: 0.8469 - val_loss: 0.3189 - val_accuracy:
0.8485
Epoch 5/20
548/548 [=====] - 6335s 12s/step - loss: 0.2748 - accuracy: 0.8756 - val_loss: 0.2852 - val_accuracy:
0.8704
Epoch 6/20
548/548 [=====] - 6430s 12s/step - loss: 0.2297 - accuracy: 0.9052 - val_loss: 0.2846 - val_accuracy:
0.8759
Epoch 7/20
548/548 [=====] - 6417s 12s/step - loss: 0.1787 - accuracy: 0.9264 - val_loss: 0.2496 - val_accuracy:
```

```
jupyter VGG19_MODELOv2 Last Checkpoint: el martes pasado a las 11:54 PM (autosaved) Logout
File Edit View Insert Cell Kernel Help Trusted Python 3 (ipykernel) C
In [12]: loss, accuracy = modelo2.evaluate(x_test,y_test)
print("loss:",loss)
print("Accuracy:",accuracy)

69/69 [=====] - 1405s 20s/step - loss: 0.1857 - accuracy: 0.9411
loss: 0.1857026219367981
Accuracy: 0.9411496520042419

In [13]: import matplotlib.pyplot as plt
acc2 = history02.history['accuracy']
val_acc2 = history02.history['val_accuracy']
epochs2 = range(1, len(acc2) + 1)
plt.plot(epochs2, acc2, label='Accuracy en Training')
plt.plot(epochs2, val_acc2, label='Accuracy en Validación', color='red')
plt.title('Accuracy del modelo VGG')
plt.legend()
plt.figure()
```



```
In [14]: from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
predictions = (modelo2.predict(x_test) > 0.5).astype("int32")

69/69 [=====] - 1514s 22s/step
```

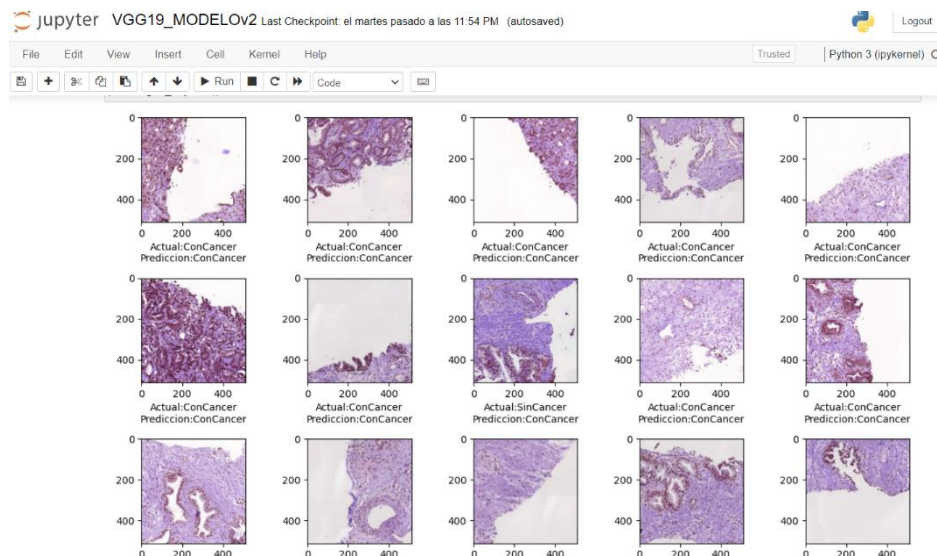
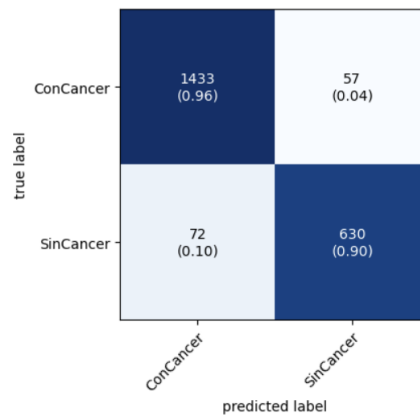
```
In [15]: accuracy_score(y_test, predictions)
```

```
Out[15]: 0.9411496350364964
```

```
In [16]: print(classification_report(y_test, predictions))
```

	precision	recall	f1-score	support
0	0.95	0.96	0.96	1490
1	0.92	0.90	0.91	702
accuracy			0.94	2192
macro avg	0.93	0.93	0.93	2192
weighted avg	0.94	0.94	0.94	2192

```
In [17]: from mlxtend.plotting import plot_confusion_matrix
matriz = confusion_matrix(y_test, predictions)
plot_confusion_matrix(conf_mat = matriz, figsize=(5,4), class_names = ['ConCancer', 'SinCancer'],
                      show_normed = True);
```



Nota: elaborado por los autores en base al Accuracy del modelo VGG-19 en validación y entrenamiento

Después de entrenar el modelo con el algoritmo VGG19, se observa que la precisión de los datos de entrenamiento se mantiene constante en las últimas épocas, alcanzando un resultado del 98.22%. En cuanto a la precisión de los datos de validación, también se mantiene estable y finaliza con un resultado del 94.11%.

La función de pérdida para los datos de entrenamiento disminuye de manera lineal hasta alcanzar un 0.18%, lo cual es un resultado positivo. Sin embargo, la función de pérdida para los datos de validación muestra resultados constantes, por lo que se decidió finalizar la ejecución.

5.3.1.3. Resultado del modelo ResNet50

Figura 13: Resultado modelo ResNet50

```
In [8]: modelo2.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
resnet50 (Functional)	(None, 16, 16, 2048)	23587712
dense (Dense)	(None, 16, 16, 256)	524544
dropout (Dropout)	(None, 16, 16, 256)	0
dense_1 (Dense)	(None, 16, 16, 128)	32896
dropout_1 (Dropout)	(None, 16, 16, 128)	0
dense_2 (Dense)	(None, 16, 16, 64)	8256
dropout_2 (Dropout)	(None, 16, 16, 64)	0
flatten (Flatten)	(None, 16384)	0
dense_3 (Dense)	(None, 1)	16385

=====
Total params: 24,169,793
Trainable params: 582,081
Non-trainable params: 23,587,712

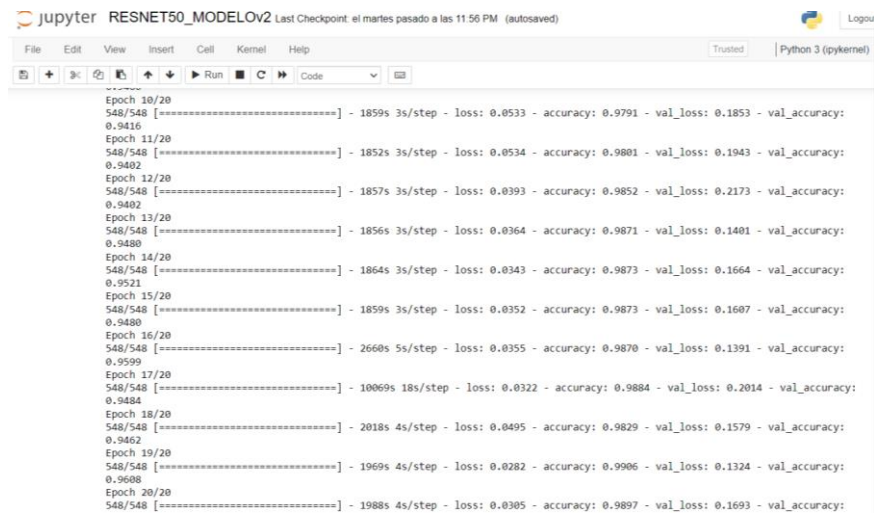
```
In [9]: modelo2.compile(optimizer="adam",loss="binary_crossentropy",metrics=["accuracy"])

In [10]: import numpy as np
from sklearn.model_selection import train_test_split

x = np.array([i[0] for i in dataTrain]).reshape(-1,512,512,3)
y = np.array([i[1] for i in dataTrain])
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)

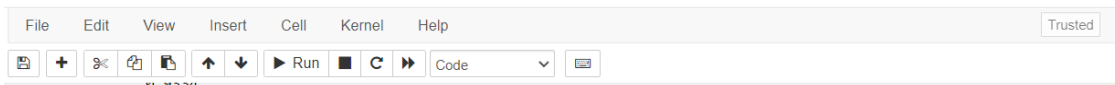
In [11]: history02 = modelo2.fit(x_train,y_train,batch_size=16,epochs=20,
                                validation_data=(x_test,y_test))

Epoch 1/20
548/548 [=====] - 1742s 3s/step - loss: 0.4635 - accuracy: 0.7544 - val_loss: 0.3366 - val_accuracy:
0.8203
Epoch 2/20
548/548 [=====] - 1717s 3s/step - loss: 0.3195 - accuracy: 0.8443 - val_loss: 0.2578 - val_accuracy:
0.8791
Epoch 3/20
548/548 [=====] - 1690s 3s/step - loss: 0.2506 - accuracy: 0.8849 - val_loss: 0.3553 - val_accuracy:
0.8312
Epoch 4/20
548/548 [=====] - 1772s 3s/step - loss: 0.2161 - accuracy: 0.9011 - val_loss: 0.2274 - val_accuracy:
0.9069
Epoch 5/20
548/548 [=====] - 2182s 4s/step - loss: 0.1444 - accuracy: 0.9366 - val_loss: 0.2297 - val_accuracy:
0.9092
Epoch 6/20
548/548 [=====] - 1687s 3s/step - loss: 0.1079 - accuracy: 0.9572 - val_loss: 0.1907 - val_accuracy:
0.9151
```



```
Epoch 10/20
548/548 [=====] - 1859s 3s/step - loss: 0.0533 - accuracy: 0.9791 - val_loss: 0.1853 - val_accuracy:
0.9416
Epoch 11/20
548/548 [=====] - 1852s 3s/step - loss: 0.0534 - accuracy: 0.9801 - val_loss: 0.1943 - val_accuracy:
0.9402
Epoch 12/20
548/548 [=====] - 1857s 3s/step - loss: 0.0393 - accuracy: 0.9852 - val_loss: 0.2173 - val_accuracy:
0.9402
Epoch 13/20
548/548 [=====] - 1856s 3s/step - loss: 0.0364 - accuracy: 0.9871 - val_loss: 0.1401 - val_accuracy:
0.9480
Epoch 14/20
548/548 [=====] - 1864s 3s/step - loss: 0.0343 - accuracy: 0.9873 - val_loss: 0.1664 - val_accuracy:
0.9521
Epoch 15/20
548/548 [=====] - 1859s 3s/step - loss: 0.0352 - accuracy: 0.9873 - val_loss: 0.1607 - val_accuracy:
0.9480
Epoch 16/20
548/548 [=====] - 2660s 5s/step - loss: 0.0355 - accuracy: 0.9870 - val_loss: 0.1391 - val_accuracy:
0.9599
Epoch 17/20
548/548 [=====] - 10069s 18s/step - loss: 0.0322 - accuracy: 0.9884 - val_loss: 0.2014 - val_accuracy:
0.9484
Epoch 18/20
548/548 [=====] - 2018s 4s/step - loss: 0.0495 - accuracy: 0.9829 - val_loss: 0.1579 - val_accuracy:
0.9462
Epoch 19/20
548/548 [=====] - 1969s 4s/step - loss: 0.0282 - accuracy: 0.9906 - val_loss: 0.1324 - val_accuracy:
0.9608
Epoch 20/20
548/548 [=====] - 1988s 4s/step - loss: 0.0305 - accuracy: 0.9897 - val_loss: 0.1693 - val_accuracy:
0.9334
```

jupyter RESNET50_MODEL0V2 Last Checkpoint: el martes pasado a las 11:56 PM (autosaved)

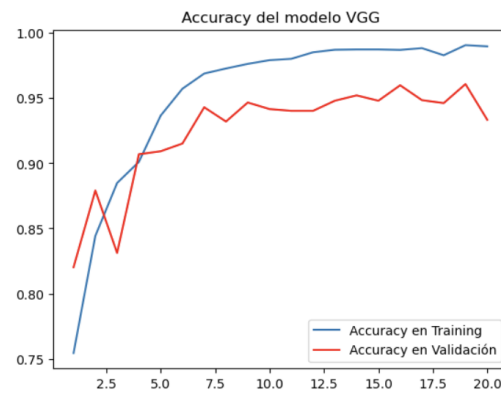


```
In [12]: loss, accuracy = modelo2.evaluate(x_test,y_test)
print("loss:",loss)
print("Accuracy:",accuracy)

69/69 [=====] - 393s 6s/step - loss: 0.1693 - accuracy: 0.9334
loss: 0.1693129539489746
Accuracy: 0.9333941340446472
```

```
In [13]: import matplotlib.pyplot as plt
acc2 = history02.history['accuracy']
val_acc2 = history02.history['val_accuracy']
epochs2 = range(1, len(acc2) + 1)
plt.plot(epochs2, acc2, label='Accuracy en Training')
plt.plot(epochs2, val_acc2, label='Accuracy en Validación', color='red')
plt.title('Accuracy del modelo VGG')
plt.legend()
plt.figure()
```

```
Out[13]: <Figure size 640x480 with 0 Axes>
```



<Figure size 640x480 with 0 Axes>

```
In [14]: from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
         predictions = (modelo2.predict(x_test) > 0.5).astype("int32")
```

```
In [14]: from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
         predictions = (modelo2.predict(x_test) > 0.5).astype("int32")
```

69/69 [=====] - 390s 6s/step

```
In [15]: accuracy_score(y_test, predictions)
```

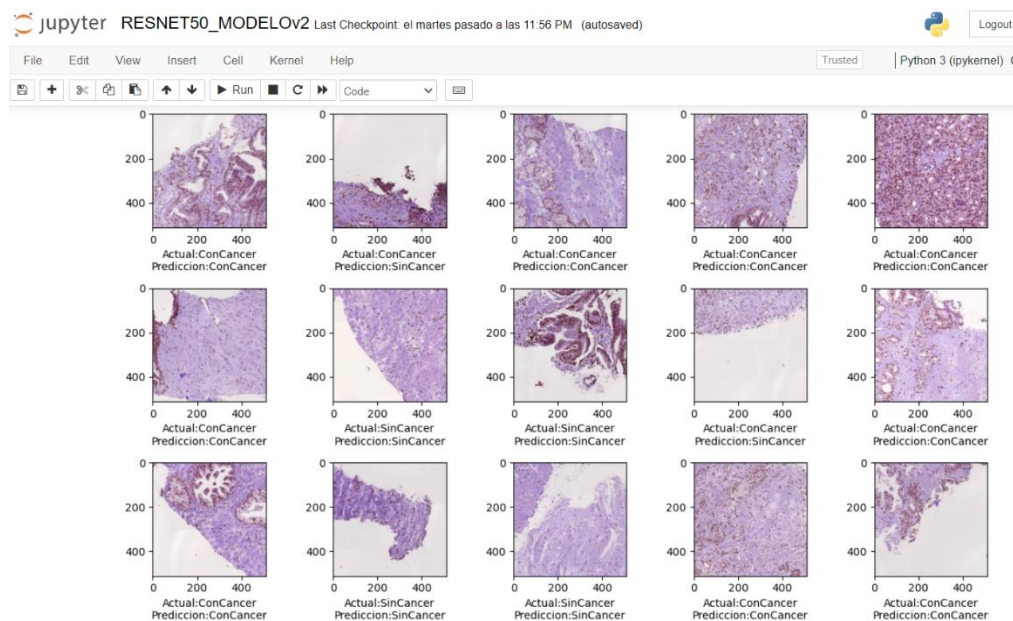
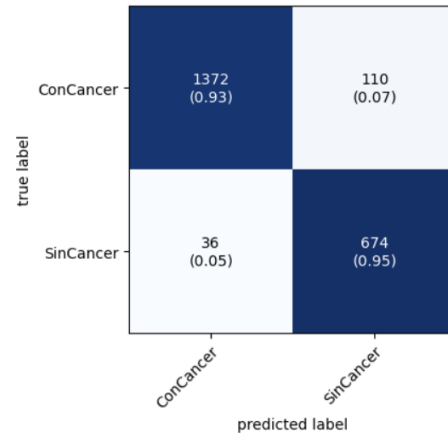
Out[15]: 0.9333941605839416

```
In [16]: print(classification_report(y_test, predictions))
```

	precision	recall	f1-score	support
0	0.97	0.93	0.95	1482
1	0.86	0.95	0.90	710
accuracy			0.93	2192
macro avg	0.92	0.94	0.93	2192
weighted avg	0.94	0.93	0.93	2192

```
In [17]: from mlxtend.plotting import plot_confusion_matrix
         matriz = confusion_matrix(y_test, predictions)
         plot_confusion_matrix(conf_mat = matriz, figsize=(5,4), class_names = ['ConCancer', 'SinCancer'],
                               show normed = True):
```

```
In [17]: from mlxtend.plotting import plot_confusion_matrix
matriz = confusion_matrix(y_test,predictions)
plot_confusion_matrix(conf_mat = matriz,figsize=(5,4),class_names = ['ConCancer', 'SinCancer'],
show_normed = True);
```



Nota: elaborado por los autores en base al Accuracy del modelo ResNet50 en validación y entrenamiento

Después de entrenar el modelo con el algoritmo ResNet90, se observa que la precisión de los datos de entrenamiento se mantiene constante en las últimas épocas, alcanzando un resultado del 98.97%. Por otro lado, la precisión de los datos de validación también se mantiene estable,

concluyendo con un resultado del 93.33%.

La función de pérdida para los datos de entrenamiento disminuye de manera lineal hasta alcanzar un 0.16%, lo que es un resultado positivo. En cambio, la función de pérdida para los datos de validación muestra resultados constantes, por lo que se decidió finalizar la ejecución.

5.3.2. Elección del modelo

Tras entrenar el modelo con el algoritmo VGG19, se observa que la precisión de los datos de entrenamiento se mantiene alta en las últimas épocas, alcanzando un 98.22%. La precisión de validación también se mantiene estable, concluyendo con un 94.11%. La función de pérdida de los datos de entrenamiento disminuye de manera lineal hasta alcanzar un 0.18%, lo que es un resultado positivo. Sin embargo, la pérdida de validación muestra resultados constantes, por lo que se consideró conveniente finalizar la ejecución.

Probamos tres modelos diferentes y, después de evaluar sus desempeños, determinamos que el VGG19 era el mejor. La precisión y la pérdida de los otros modelos no lograron alcanzar ni mantener los niveles obtenidos con el VGG19. En particular, el VGG19 demostró una mayor estabilidad en las métricas de precisión y pérdida tanto en el conjunto de entrenamiento como en el de validación.

Comparado con los otros modelos, el VGG19 demostró una superior capacidad de generalización y adaptación a nuevos datos. Por estas razones, el VGG19 fue seleccionado como el mejor modelo para nuestro estudio, destacando por su alta precisión y baja pérdida, así como por su consistencia durante el proceso de entrenamiento y validación. Estos resultados indican que el VGG19 es más efectivo y fiable para la tarea de clasificación de imágenes en este contexto.

Como conclusión considerando la siguiente tabla adjunta:

Tabla 3: Resultados algoritmos aplicados

Algoritmo	Cantidad de épocas	Accuracy	Loss
VGG16	20	90.10%.	0.34%
VGG19	20	94,11%	0.18%
ResNet50	20	93,33%	0.16%

Nota: Tabla elaborado por los autores en base a los resultados Accuracy de los modelos vs la pérdida de precisión obtenida

Podemos indicar que el modelo con mayor precisión y menor valor de pérdida se obtuvo aplicando el algoritmo VGG19 con una precisión de 84,11% dentro del rango de aceptación en la predicción para esta investigación.

5.4. Desarrollar el sistema inteligente

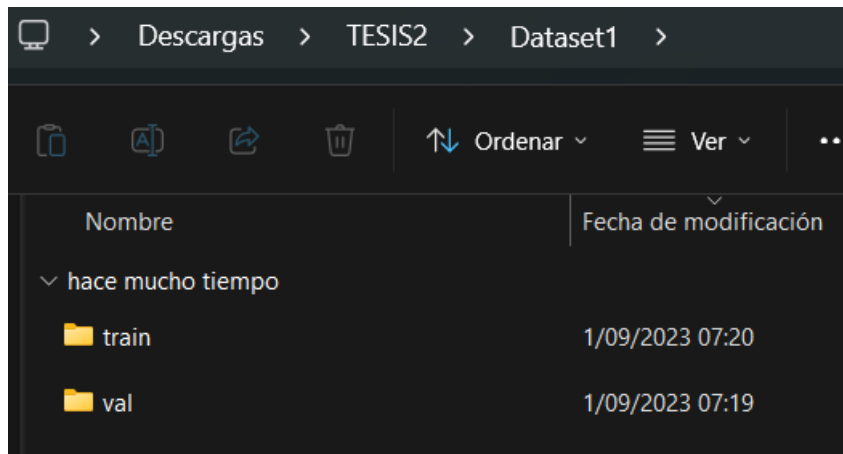
5.4.1. *Requisitos y preparación*

La investigación se llevó a cabo en la plataforma Anaconda utilizando Python 3.10 y TensorFlow 2.10. Para entrenar el modelo, se utilizó una laptop equipada con un procesador Ryzen 7 de 13^a generación y una tarjeta gráfica Nvidia GeForce RTX Studio para el procesamiento de imágenes.

5.4.2. *Preparación del entorno de trabajo*

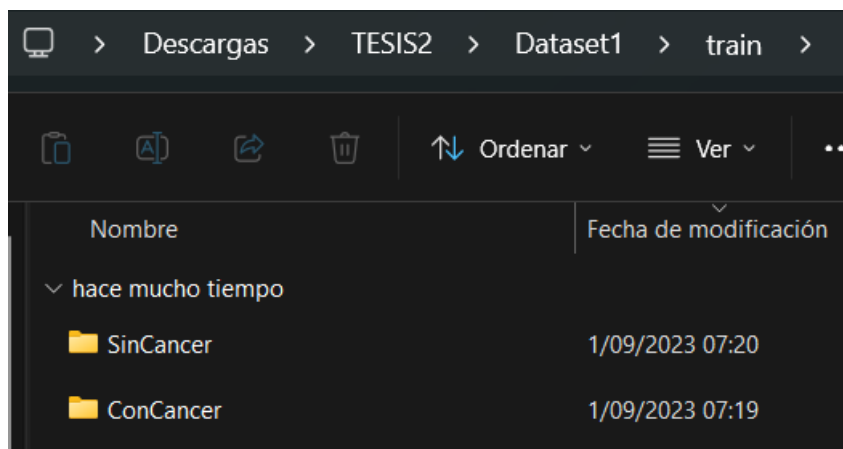
Para el desarrollo de esta investigación, se empleó el lenguaje de programación Python y la plataforma de desarrollo Anaconda, utilizando una estructura específica en nuestro dataset almacenado localmente en el explorador de archivos.

Figura 14: Directorios de trabajo



Nota: elaborado por los autores en base a la ubicación y distribución del dataset para entrenamiento y validación

Figura 15: Estructura Principal



Nota: elaborado por los autores en base a la ubicación y distribución del dataset en sus 2 categorías

El directorio de datos del proyecto se subdividió en dos subdirectorios los cuales son train (entrenamiento) el val (validación), tienen data de la clasificación de Gleason que determina si el paciente tiene cáncer de próstata, siendo subdivididas en carpetas que tienen los siguientes nombres: ConCancer y SinCancer.

5.4.3. Desarrollo del sistema

Se creó un sistema inteligente utilizando Python 3.11.5 y el framework Django, diseñado para asistir en el ámbito médico con la detección temprana y precisa del cáncer de próstata.

Para almacenar y gestionar los datos de manera eficiente, se ha elegido MySQL como sistema de gestión de bases de datos. Con MySQL, se asegura la integridad y la seguridad de los datos médicos sensibles.

El desarrollo consiste en la capacidad de realizar predicciones precisas utilizando un modelo de aprendizaje automático previamente entrenado. Este modelo ha sido entrenado con un amplio conjunto de datos de imágenes médicas de próstatas.

Cuando un usuario carga una imagen médica en el sistema, los datos son transmitidos al backend, donde se aplican técnicas de preprocesamiento para preparar la imagen para su análisis. Luego el modelo de aprendizaje automático entra en acción, realizando una evaluación detallada de la imagen y generando una predicción sobre la presencia o ausencia de cáncer de próstata.

Los resultados de la predicción se muestran al usuario a través de la interfaz y también pueden ser descargados en un informe en formato PDF.

A continuación, se dará información más detallada acerca de la creación del sistema inteligente:

5.4.3.1. Modelo de estructura para la base de datos

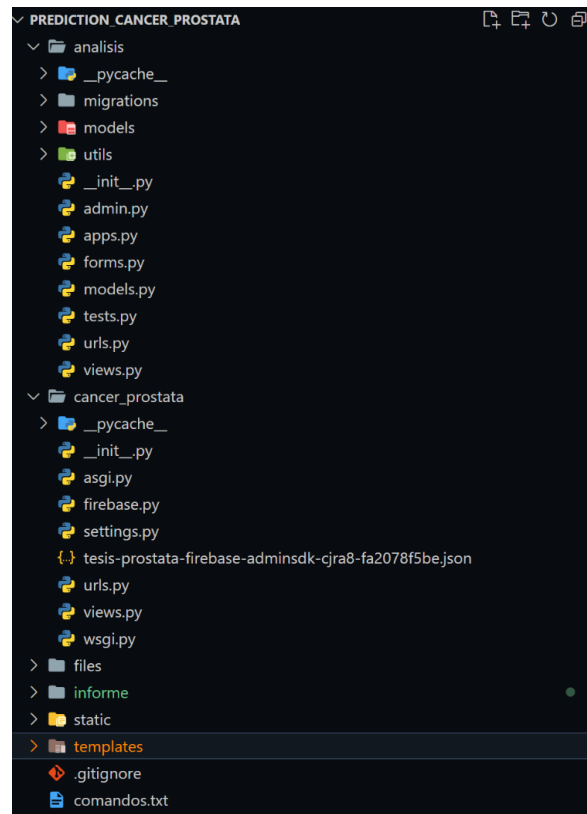
La base de datos utilizada fue Mysql, para crear nuestras tablas lo hicimos desde Django con las migraciones al definir nuestros modelos se autogeneran como tablas en nuestra base de datos y esto hace que la creación no sea un proceso manual si no automático.

5.4.3.2. Desarrollo para el backend para el software

Para el desarrollo del backend se utilizó Python en la versión 3.11.5 usando el framework Django en la versión 4.0.4.

La implementación consistió principalmente en conectar nuestro backend con nuestro modelo de predicción que ha sido entrenado anteriormente además de conectar con el frontend para recibir los datos enviados. La utilizada fue la de Modelo - Vista - Controlador.

Figura 16: Estructura de carpetas del backend

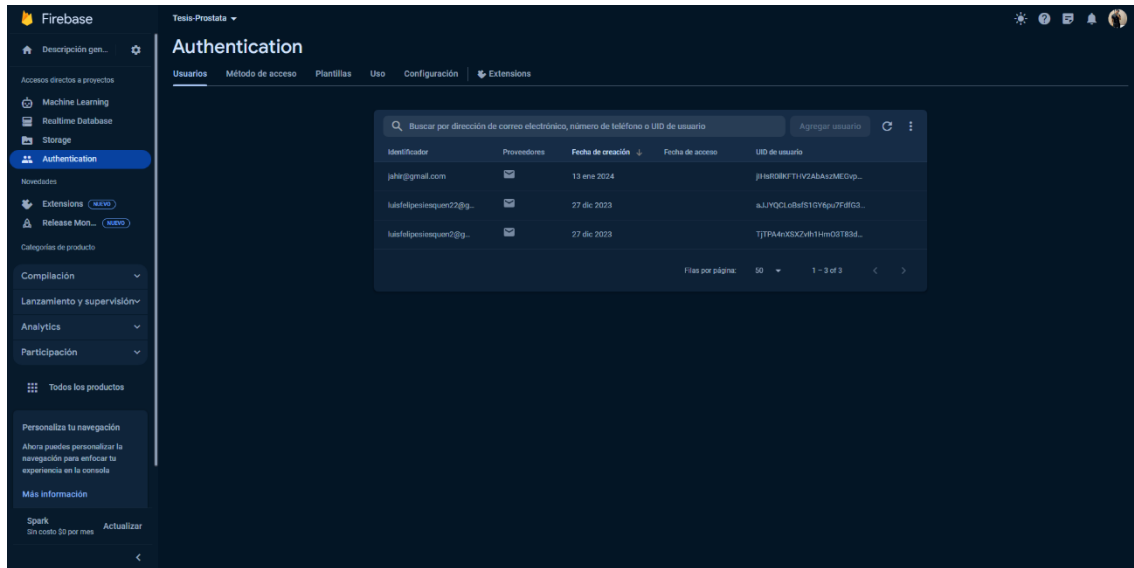


Nota: elaborado por los autores en base a la estructura backend del sistema

Esta estructura ayuda a mantener un código limpio y organizado, facilita la reutilización de código y promueve la escalabilidad de las aplicaciones.

Utilizamos Firebase Authentication que nos proporcionó una solución completa y segura para manejar el proceso de autenticación de usuarios en nuestra aplicación.

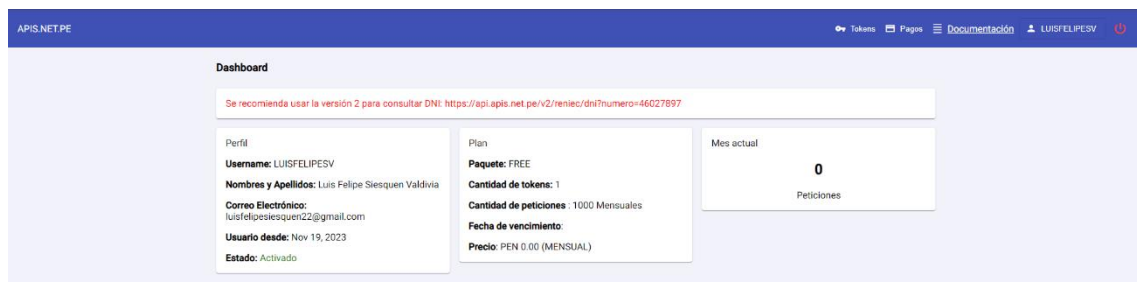
Figura 17: Cuenta enlazada al proyecto



Nota: elaborado por los autores en base al módulo de autenticación con la cuenta enlazada al sistema

Finalmente se utilizó el API de la RENIEC para obtener los datos de un usuario a través de su Dni.

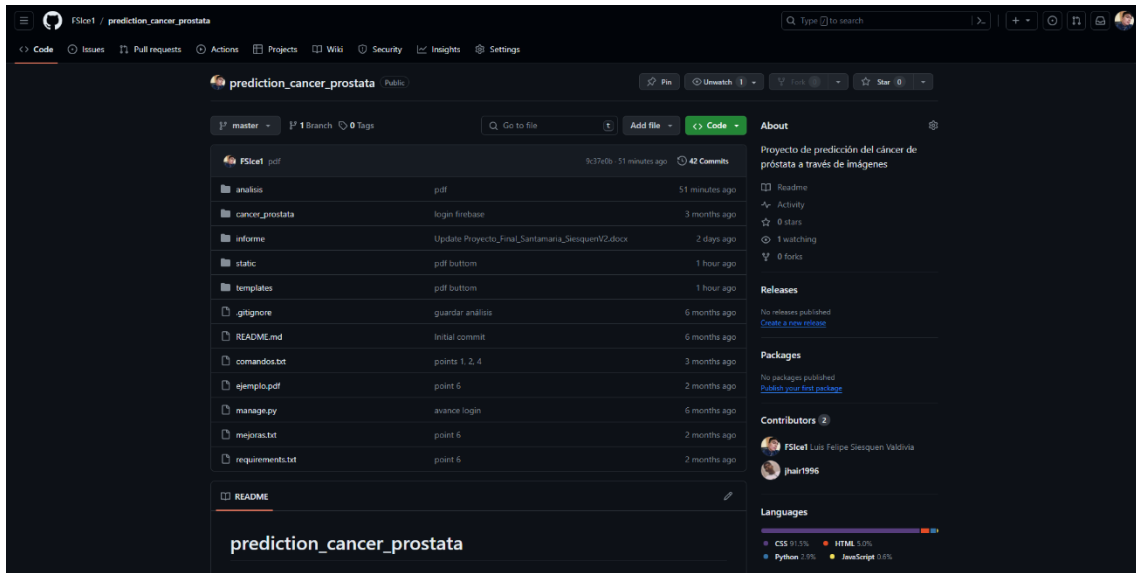
Figura 18: Resultados enlace de cuenta



Nota: elaborado por los autores en base a los resultados de la cuenta enlazada al sistema.

El código del sistema fue almacena en GitHub y el enlace para acceder al repositorio es https://github.com/FSIce1/prediction_cancer_prostata, este es un repositorio público así que cualquiera puede acceder al código.

Figura 19: Repositorio de GitHub



Nota: elaborado por los autores en base al repositorio de GitHub que contiene el proyecto

5.4.3.2.1. Funciones

Se detalla mediante una tabla las funciones elaboradas con una descripción explicando el funcionamiento de cada punto elaborado.

Tabla 4: Descripción de funciones

Función	Descripción
crear_usuario	Nos sirve para registrar el usuario para poder loguearse posteriormente.
Inicio	Redirige al sistema en caso de que el usuario se haya autenticado.
analisis_imagen	Realiza la predicción a través del modelo.
cerrar_sesion	Cierra la sesión actual en el sistema y redirige a la pestaña del login.

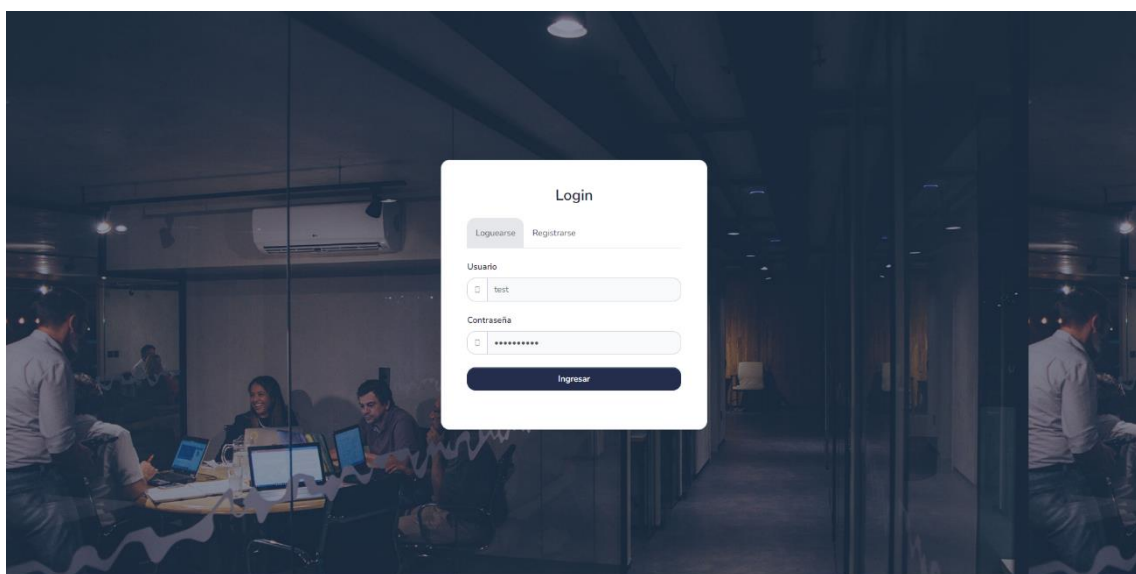
buscar_por_dni	Consulta el api de la RENIEC y nos devuelve los datos del usuario que se consultó con su dni.
resultado_imagen	Nos lleva a la vista final que es mostrar el resultado.
realizar_analisis	En esta función se consume el modelo de predicción.
historial_analisis	Historial de todos los análisis realizados en el sistema.
pacientes	Registra los pacientes que fueron consultados por la RENIEC
generar_pdf	Reporte final de cada análisis realizado

Nota: elaborado por los autores en base al detalle de funciones del sistema

5.4.3.3. Desarrollo para el frontend para el software

Para desarrollar el frontend de la aplicación utilizamos HTML, Css y JavaScript para integrarlo con el motor de plantillas de Django, tenemos las vistas de Login, Registro, Análisis, Resultado, Historial y Pacientes.

Figura 20: Menú Login del sistema



Nota: elaborado por los autores en base al login de inicio del sistema

Para la siguiente prueba se adjuntó una imagen extraída al azar del dataset en la opción “seleccionar archivo”, posterior a ello se digita el número de DNI de la persona que está realizando la consulta, la cual genera una búsqueda a través del API de reniec los datos personales del usuario:

Figura 21: Análisis de resultados

Nota: elaborado por los autores en base al análisis de resultados consulta RENIEC

A continuación, el sistema te arroja los siguientes resultados:

Figura 22: Resultados de la evaluación

Título: PRUEBA 1

Descripción: Descripción 1

Nombre de la imagen: imagenes/1680027040_Block_Region_12_5_5_xini_23652_yini_11960_PfjSfWR.jpg

Tiempo de análisis: 2.7399 segundos

Resultado Final: Sin Cáncer

Porcentaje: 99.99909996986389

PDF

Nota: elaborado por los autores en base al resultado de predicción de una imagen

Adicional al resultado obtenido, el sistema nos permite exportarlo con un formato PDF para emitirle físicamente el resultado al paciente evaluado.

Figura 23: Exportable en formato PDF

REPORTE FINAL

Información del paciente

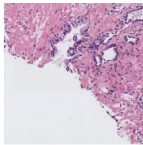
DNI: 72314652
Nombres: LUIS FELIPE
Apellido Paterno: SIESQUEN
Apellido Materno: VALDIVIA

Información del resultado

Título: PRUEBA 1
Descripción: Descripción 1

Resumen Final

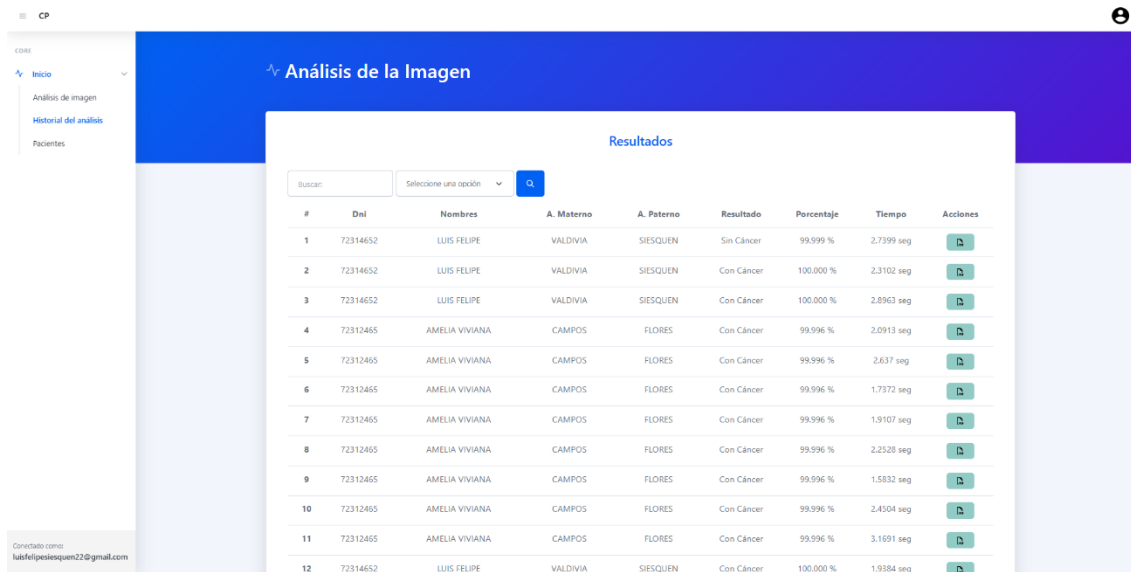
Tiempo de análisis: 2.7399 seg
Resultado Final: Sin Cáncer
Porcentaje: 99.999



Nota: elaborado por los autores en base al reporte final de la prueba realizada a una imagen de validación

El sistema también nos permite visualizar un reporte de todos los pacientes que realizaron una evaluación en el sistema con su resultado obtenido considerando parámetros como Dni, Nombre y Apellidos, Resultado, Porcentaje de precisión y el tiempo que demoró en realizarse la consulta.

Figura 24: Resumen de análisis ejecutados en el sistema



The screenshot shows a web application interface for 'Análisis de la Imagen'. On the left is a sidebar with navigation links: 'Inicio', 'Análisis de imagen', 'Historial del análisis', and 'Pacientes'. The main content area has a blue header with the title 'Análisis de la Imagen' and a 'Resultados' section. Below this is a search bar with a 'Buscar:' input, a 'Seleccione una opción' dropdown, and a search button. The results are displayed in a table with 12 rows. Each row contains an ID, DNI, Name, Mother's Name, Father's Name, Result, Percentage, Time, and Actions. The results show two groups of patients: one with 'Sin Cáncer' (rows 1-3) and another with 'Con Cáncer' (rows 4-12). The 'Con Cáncer' group shows a 99.996% accuracy rate across 10 different patients, while the 'Sin Cáncer' group shows a 100.000% accuracy rate for three patients.

#	Dni	Nombres	A. Materno	A. Paterno	Resultado	Porcentaje	Tiempo	Acciones
1	72314652	LUIS FELIPE	VALDIVIA	SIESQUEN	Sin Cáncer	99.999 %	2.7399 seg	[Icon]
2	72314652	LUIS FELIPE	VALDIVIA	SIESQUEN	Con Cáncer	100.000 %	2.3102 seg	[Icon]
3	72314652	LUIS FELIPE	VALDIVIA	SIESQUEN	Con Cáncer	100.000 %	2.8963 seg	[Icon]
4	72312465	AMELIA VIVIANA	CAMPOS	FLORES	Con Cáncer	99.996 %	2.0913 seg	[Icon]
5	72312465	AMELIA VIVIANA	CAMPOS	FLORES	Con Cáncer	99.996 %	2.637 seg	[Icon]
6	72312465	AMELIA VIVIANA	CAMPOS	FLORES	Con Cáncer	99.996 %	1.7372 seg	[Icon]
7	72312465	AMELIA VIVIANA	CAMPOS	FLORES	Con Cáncer	99.996 %	1.9107 seg	[Icon]
8	72312465	AMELIA VIVIANA	CAMPOS	FLORES	Con Cáncer	99.996 %	2.2528 seg	[Icon]
9	72312465	AMELIA VIVIANA	CAMPOS	FLORES	Con Cáncer	99.996 %	1.5832 seg	[Icon]
10	72312465	AMELIA VIVIANA	CAMPOS	FLORES	Con Cáncer	99.996 %	2.4504 seg	[Icon]
11	72312465	AMELIA VIVIANA	CAMPOS	FLORES	Con Cáncer	99.996 %	3.1691 seg	[Icon]
12	72314652	LUIS FELIPE	VALDIVIA	SIESQUEN	Con Cáncer	100.000 %	1.9384 seg	[Icon]

Nota: elaborado por los autores en base al módulo de resumen de todos los análisis realizados en el sistema

5.5. Evaluación del rendimiento del sistema inteligente

Primero, cargamos el modelo que había sido previamente entrenado. Luego, se preparó los datos de prueba, que son imágenes organizadas en carpetas según sus categorías (Sin Cáncer y Con Cáncer). Se utilizó la herramienta TensorFlow para cargar estas imágenes y agruparlas en lotes, lo que facilita su procesamiento.

El modelo realizó predicciones sobre estos datos de prueba. Cada predicción es una suposición del modelo sobre a qué categoría pertenece cada imagen. Se convirtió estas predicciones a etiquetas que representan dichas categorías.

Luego, se comparó las predicciones del modelo con las etiquetas reales de las imágenes para evaluar su precisión. Se utilizó una matriz de confusión para ver cuántas predicciones fueron correctas y cuántas no.

También se generó un informe detallado que muestra métricas como precisión, recall y F1-

score para cada categoría. Estas métricas nos ayudan a entender el funcionamiento el modelo en general y en cada categoría específica.

Por último, se identificaron las clases de las imágenes (Sin Cáncer y Con Cáncer), estas se utilizaron para analizar los resultados y hacer conclusiones sobre el rendimiento del modelo.

Explicación del modelo de rendimiento

Figura 25: Conversión de las probabilidades predichas

```
[2] predicciones = modelo_vgg.predict(test_ds)
predicciones

... 343/343 [=====] - 5794s 17s/step

... array([[2.5467061e-08],
          [3.5322708e-01],
          [1.0498173e-01],
          ...,
          [2.0911165e-04],
          [0.0000000e+00],
          [2.1779264e-04]], dtype=float32)
```

Nota: elaborado por los autores en base a la conversión de las probabilidades en matriz

Se carga el modelo previamente entrenado (modelo vgg19) y se aplica sobre un conjunto de datos de prueba llamado “test_ds” el cual contiene imágenes organizadas para las categorías “Sin Cáncer” y “Con Cáncer”

Figura 26: Decisión final de categoría

```
[3] import numpy as np
predicciones = np.argmax(predicciones, axis = 1)
predicciones

... array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

Nota: elaborado por los autores en base a la decisión de categoría con librería de python Numpy

Luego de que el modelo realiza las predicciones sobre las imágenes del conjunto de prueba, obtenemos un conjunto de probabilidades para cada imagen, indicando la confianza del modelo en cada categoría posible. Sin embargo, necesitamos convertir estas probabilidades en decisiones finales sobre la categoría de cada imagen. Para hacer esto, usamos `np.argmax`, que selecciona la categoría con la mayor probabilidad para cada imagen. Esto nos da las etiquetas de clase predichas, que podemos comparar con las etiquetas reales para evaluar el rendimiento del modelo.

Figura 27: Etiquetas por lote

```
for img_batch, y_reales in test_ds:
    print(y_reales)
```

[4]

Python

```
... tf.Tensor([1 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0 1 1 0 0 1], shape=(32,), dtype=int32)
tf.Tensor([1 0 1 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 1 0 0 1 0 0 1 1 0 0 1 0 0 1], shape=(32,), dtype=int32)
tf.Tensor([0 0 0 0 0 1 0 0 0 1 0 1 1 1 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 1], shape=(32,), dtype=int32)
tf.Tensor([0 0 0 0 1 0 1 0 1 0 0 0 1 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0], shape=(32,), dtype=int32)
tf.Tensor([0 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 1 0 0 1 1 1 1 1 1 1 1 0 0 1 0 1], shape=(32,), dtype=int32)
tf.Tensor([1 1 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0], shape=(32,), dtype=int32)
tf.Tensor([0 1 0 1 0 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1 0 0 0 0 1 0 1 0 0 0 1 0], shape=(32,), dtype=int32)
tf.Tensor([0 0 1 1 1 0 0 0 1 0 1 0 0 1 0 0 1 1 0 0 0 0 0 0 1 1 0 1 1 0 0 0], shape=(32,), dtype=int32)
tf.Tensor([0 0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 0 1], shape=(32,), dtype=int32)
tf.Tensor([0 0 0 0 0 1 0 0 0 1 1 0 1 0 1 0 0 1 1 1 0 1 1 0 0 0 0 0 0 0 1], shape=(32,), dtype=int32)
tf.Tensor([0 0 0 0 0 0 0 0 1 1 1 0 0 0 1 1 1 0 0 1 0 1 0 0 0 0 1 0 0 0 0], shape=(32,), dtype=int32)
tf.Tensor([0 0 0 0 1 0 0 0 1 0 0 1 1 1 0 0 1 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0], shape=(32,), dtype=int32)
tf.Tensor([0 1 0 0 0 1 0 1 0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 1 1 1 0 0 0 1 0 0], shape=(32,), dtype=int32)
tf.Tensor([0 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0 1 1 1 0 1 0 1 0 1 0 1 0 0 0 1 1 0], shape=(32,), dtype=int32)
tf.Tensor([0 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 0 0 1 1 0 0 1 0 0 1 0 0 1 0 0], shape=(32,), dtype=int32)
tf.Tensor([1 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 1 0], shape=(32,), dtype=int32)
tf.Tensor([0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 1 0], shape=(32,), dtype=int32)
tf.Tensor([1 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 1 1 1], shape=(32,), dtype=int32)
tf.Tensor([1 1 1 1 0 0 0 0 0 1 1 0 0 0 1 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0], shape=(32,), dtype=int32)
tf.Tensor([1 1 0 1 1 1 0 0 0 0 0 1 1 1 0 1 0 0 1 0 1 1 0 1 0 0 0 1 0 1 0 0], shape=(32,), dtype=int32)
tf.Tensor([0 1 1 0 0 1 0 0 0 1 0 1 1 1 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 1], shape=(32,), dtype=int32)
tf.Tensor([0 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 1 0 0 1 1 0 0], shape=(32,), dtype=int32)
tf.Tensor([1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1], shape=(32,), dtype=int32)
tf.Tensor([0 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 1 1 1 0 0 0 1 0 0 0 0 1 0 0], shape=(32,), dtype=int32)
tf.Tensor([0 1 0 1 0 0 0 1 1 1 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 1 0 0 0 1], shape=(32,), dtype=int32)
```

Nota: elaborado por los autores en base a todas las etiquetas por lote de las clases predichas

Aquí se itera todo el conjunto de datos de prueba “`test_ds`”, extrayendo lotes de imágenes y sus correspondientes etiquetas reales. La instrucción `print(y_reales)` dentro del bucle imprime las etiquetas reales de cada lote de imágenes.

Esto significa que cada vez que el bucle itera, `y_reales` contiene las etiquetas reales de un lote de imágenes del conjunto de datos de prueba. Este tipo de iteración es útil para verificar qué etiquetas se están utilizando como datos de prueba y para asegurarse de que las etiquetas reales se

correspondan correctamente con las imágenes que el modelo está evaluando.

Figura 28: Almacenamiento de etiquetas reales y predictivas

```
y_pred = []
y_real = []

for img_batch, y_batch in test_ds:
    y_real.append(y_batch)
    predicciones = modelo_vgg.predict(img_batch)
    y_pred.append(np.argmax(predicciones, axis = 1))

y_reales = tf.concat([item for item in y_real], axis = 0)
y_predicciones = tf.concat([item for item in y_pred], axis = 0)
```

[5] Python

Nota: elaborado por los autores en base a la etiquetas reales y predictivas y su lugar de almacenamiento

En este segmento de código, se inicializan dos listas vacías, `y_pred` y `y_real`, donde la primera almacenará las predicciones del modelo y la segunda almacenará las etiquetas reales de los datos de prueba. Luego, se inicia un bucle que itera sobre el conjunto de datos de prueba `test_ds`, obteniendo en cada iteración un lote de imágenes (`img_batch`) y sus correspondientes etiquetas reales (`y_batch`). Las etiquetas reales de cada lote se añaden a la lista `y_real`, y el modelo `modelo_vgg` se utiliza para predecir las clases de las imágenes en el lote, almacenando las predicciones resultantes en la variable `predicciones`. Para cada lote de predicciones, se determina la clase con la mayor probabilidad usando `np.argmax(predicciones, axis=1)`, lo que convierte las probabilidades de salida en etiquetas de clase que se añaden a la lista `y_pred`. Después de completar el bucle y haber procesado todos los lotes de datos, se concatenan las listas `y_real` y `y_pred` en tensores únicos utilizando `tf.concat`, donde `y_reales` contiene todas las etiquetas reales concatenadas y `y_predicciones` contiene todas las etiquetas predichas concatenadas.

Figura 29: Evaluación del desempeño del modelo

```
from sklearn.metrics import confusion_matrix
from mlxtend.plotting import plot_confusion_matrix

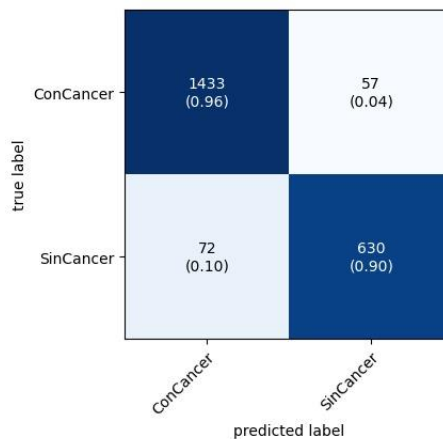
matriz = confusion_matrix(y_reales,y_predicciones)
plot_confusion_matrix(conf_mat = matriz,figsize=(5,4),
                      class_names = categorias,
                      show_normed = True);
```

[9] Python

Nota: elaborado por los autores en base a la creación de la matriz de confusión para evaluar el desempeño del modelo

Se definen los datos de etiquetas reales, predicciones y nombres de categorías. Genera una matriz de confusión que compara las etiquetas reales con las predichas, y finalmente, dibuja un gráfico de la matriz de confusión mostrando las proporciones de clasificaciones correctas e incorrectas, permitiéndote evaluar el desempeño de tu modelo de clasificación.

Figura 30: Matriz de confusión



Nota: elaborado por los autores en base a los resultados de la matriz de confusión

La matriz de confusión de las dos clases: 'ConCancer' y 'SinCancer'. La interpretación de los valores en la matriz es:

Clase 'ConCancer' (etiqueta real):

Verdaderos Positivos (TP): 1433 (0.96)

Falsos Negativos (FN): 72 (0.10)

Clase 'SinCancer' (etiqueta real):

Falsos Positivos (FP): 57 (0.04)

Verdaderos Negativos (TN): 630 (0.90)

Los valores entre paréntesis representan la proporción de las predicciones respecto al total de predicciones para cada clase. Por ejemplo, 0.96 indica que el 96% de las predicciones para 'ConCancer' son correctas.

VI. DISCUSIÓN DE RESULTADOS

Nuestro modelo implementado en esta investigación, VGG-19, superó a los otros dos modelos evaluados (VGG16 y ResNet50), alcanzando una precisión del 94.11% y una pérdida mínima, dentro del rango de aceptación para esta investigación. Esta precisión es comparable o incluso superior a la obtenida en otros estudios similares, lo que sugiere la efectividad de la arquitectura VGG-19 en el diagnóstico del cáncer de próstata.

En relación con la hipótesis planteada que la implementación de un sistema inteligente basado en Deep Learning mejora significativamente la capacidad de diagnóstico de cáncer de próstata, los resultados de este estudio la respaldan firmemente. La precisión y fiabilidad demostradas por VGG-19 evidencian que este modelo optimiza la capacidad de clasificación en comparación con métodos tradicionales y otras arquitecturas, validando su aplicabilidad en el ámbito clínico para la detección temprana de cáncer de próstata.

Aunque la optimización del tiempo de diagnóstico no fue el enfoque principal de este estudio, es importante destacar que la rapidez y eficiencia computacional de los modelos de Deep Learning son factores que deben explorarse en futuras investigaciones para garantizar su viabilidad clínica. La velocidad de procesamiento del modelo y su precisión mejorada indican que estos sistemas tienen el potencial de integrarse en flujos de trabajo médicos sin comprometer la calidad del diagnóstico.

Comparando estos resultados con la literatura previa, nuestros hallazgos son consistentes con estudios que han demostrado la eficacia de las redes neuronales convolucionales en el diagnóstico de enfermedades a partir de imágenes médicas. La alta precisión alcanzada en este estudio respalda la hipótesis inicial de que un sistema de inteligencia artificial basado en Deep Learning puede mejorar significativamente el diagnóstico del cáncer de próstata. Esto refuerza la

viabilidad de utilizar herramientas avanzadas como el VGG-19 en un entorno clínico, proporcionando un diagnóstico preciso y confiable.

Es relevante señalar que, aunque nuestros resultados son prometedores, aún existen desafíos a considerar. Uno de ellos es la generalización del modelo a diferentes conjuntos de datos, ya que la variabilidad en las imágenes médicas puede afectar su rendimiento. Además, es importante desarrollar mecanismos que permitan a los profesionales de la salud interpretar de manera clara las decisiones del modelo, un aspecto esencial para su adopción en la práctica clínica. Estos desafíos deberán ser abordados en futuras investigaciones para garantizar la aplicabilidad clínica y el impacto real en la atención médica.

En comparación con estudios previos, como el de Faez (2018), quien utilizó datos del Hospital Imam Reza en Teherán y aplicó técnicas de Deep Learning y redes neuronales artificiales (ANN) para optimizar el diagnóstico de cáncer de próstata, nuestros resultados son notablemente superiores. Faez alcanzó una precisión de clasificación del 86.3%, mientras que nuestro estudio alcanzó un 94.11% con el modelo VGG-19. Esta comparación resalta la efectividad de emplear arquitecturas más avanzadas de Deep Learning, lo que sugiere que el uso de VGG-19 ofrece un rendimiento superior en la detección de esta enfermedad.

Forero Cuellar (2019) empleó el software "Orange Data Mining Fruit and Fun" para llevar a cabo minería de datos y análisis predictivo en el diagnóstico de cáncer de próstata. En su enfoque, el diagnóstico se realiza mediante la evaluación de un médico experto, quien analiza variables usando imágenes de resonancia magnética y otras técnicas tradicionales, logrando una precisión del 83.1% con regresión logística y del 79.5% con árboles de decisión. Nuestro estudio, en contraste, demuestra que los modelos de Deep Learning, particularmente VGG-19, superan considerablemente estos resultados al lograr una precisión del 94.11%, subrayando las ventajas de

utilizar arquitecturas avanzadas de Deep Learning en el diagnóstico de enfermedades.

La comparación con el estudio de Forero Cuellar destaca el potencial de las técnicas de Deep Learning para superar las limitaciones de los métodos tradicionales y los algoritmos de minería de datos en la clasificación y diagnóstico del cáncer de próstata. Esto subraya la importancia de la investigación continua y la implementación de modelos más sofisticados para optimizar los resultados clínicos, mejorando la precisión y eficiencia en la detección de esta enfermedad.

En resumen, los resultados obtenidos apoyan la hipótesis de que un sistema de Deep Learning, basado en la arquitectura VGG-19, mejora significativamente el diagnóstico de cáncer de próstata a partir de imágenes médicas. Este sistema ofrece una alternativa efectiva y precisa que podría integrarse en el flujo de trabajo médico, contribuyendo a diagnósticos más rápidos y efectivos, y con el potencial de mejorar significativamente la atención y el pronóstico de los pacientes.

VII. CONCLUSIONES

Este estudio confirma que un sistema inteligente basado en Deep Learning, utilizando el modelo VGG19, mejora significativamente la precisión en el diagnóstico de cáncer de próstata. Con una precisión del 94.11%, el modelo VGG19 demuestra ser una herramienta prometedora que puede asistir eficazmente a los profesionales de la salud en la identificación y clasificación de esta enfermedad, optimizando los procesos de diagnóstico.

La utilización del conjunto de datos SICAPv2, compuesto por 13,827 imágenes segmentadas en dos categorías (con cáncer y sin cáncer), facilitó un entrenamiento exhaustivo y una evaluación robusta de los modelos de Deep Learning. Esta base de datos permitió asegurar la representatividad y calidad de los resultados, validando la efectividad del sistema con una precisión de diagnóstico del 94,11%.

La evaluación y comparación de los modelos VGG16, VGG19 y ResNet50 confirman la importancia de seleccionar una arquitectura de red neuronal adecuada para obtener resultados óptimos. Aunque los modelos VGG16 y ResNet50 lograron precisiones del 90.10% y 93.33% respectivamente, el modelo VGG19 alcanzó un 94.11% de precisión, lo que lo convierte en la opción más eficiente y precisa para esta tarea de diagnóstico médico.

La implementación del sistema inteligente desarrollado en esta investigación ha demostrado ser una herramienta valiosa para los profesionales de la salud, mejorando notablemente la precisión y eficiencia en el diagnóstico del cáncer de próstata. Con una precisión validada del 94,11%, el sistema proporciona diagnósticos confiables, aumentando su relevancia para el uso clínico.

1. **Análisis y preprocesamiento de imágenes:** El proceso de análisis y preprocesamiento de las imágenes médicas permitió construir un conjunto de datos

representativo y balanceado, asegurando que el modelo fuera entrenado con datos de alta calidad y consistencia. Este paso fue fundamental para optimizar el rendimiento del sistema, mejorando la precisión de las predicciones.

2. **Desarrollo y optimización del modelo:** Se desarrolló y optimizó un modelo de Deep Learning basado en la arquitectura VGG19, en especial ajustado para diferenciar eficazmente entre imágenes de muestras con y sin cáncer de próstata. Los resultados demuestran que el modelo es altamente preciso y sensible en la detección de cáncer, validando así su efectividad para abordar este desafío médico crítico.
3. **Implementación del sistema de predicción:** El sistema entrenado se integró en una interfaz de usuario intuitiva y accesible, facilitando su adopción por profesionales de la salud. Esta herramienta ha demostrado ser efectiva en entornos clínicos, proporcionando una interacción rápida y optimizando el flujo de trabajo médico, lo que contribuye a una mejor atención al paciente.
4. **Evaluación del rendimiento del sistema:** El sistema fue evaluado rigurosamente mediante métricas como precisión, sensibilidad y especificidad, las cuales confirman su eficacia en la detección del cáncer de próstata. Estas métricas validan que el sistema no solo es preciso, sino también confiable para su aplicación clínica, brindando una solución robusta que puede integrarse efectivamente en la práctica médica diaria.

En conclusión, este estudio confirma que el uso de modelos avanzados de Deep Learning, como VGG19, puede transformar el diagnóstico del cáncer de próstata, proporcionando una herramienta con una precisión del 94,11% en la detección temprana. La adopción de este sistema

en hospitales y centros de salud tiene el potencial de mejorar notablemente la calidad de vida de los pacientes y optimizar los recursos del sistema de salud. No obstante, se recomienda seguir validando el sistema en pruebas piloto y entornos hospitalarios reales para garantizar su aplicabilidad y confiabilidad a mayor escala.

VIII. RECOMENDACIONES

En este estudio, hemos evaluado diversos modelos de aprendizaje profundo, como VGG16, VGG19 y ResNet, con el objetivo de identificar el modelo más eficiente para la clasificación de imágenes médicas. Los resultados mostraron que VGG19 se destacó por su precisión y consistencia, posicionándose como el modelo más adecuado para nuestro conjunto de datos.

Recomendaciones para futuros estudios:

1. **Integración de datos clínicos adicionales:** Para mejorar la precisión y capacidad predictiva de los modelos de diagnóstico, se recomienda llevar a cabo estudios que no se limiten únicamente al análisis de imágenes médicas, sino que también incorporen variables clínicas adicionales. Entre ellas se incluyen la edad del paciente, antecedentes familiares, y marcadores biológicos relevantes. La combinación de estas variables con los datos de imágenes podría proporcionar un diagnóstico más robusto y personalizado.
2. **Protocolos estandarizados de captura de imágenes:** Para garantizar la efectividad y reproducibilidad de los modelos, es fundamental que las imágenes médicas sean capturadas bajo un protocolo estandarizado. Este debe asegurar la correcta visualización de la región de interés y mantener una calidad y consistencia óptimas en todas las imágenes. El uso de estos protocolos evitaría variaciones que puedan afectar el entrenamiento y la evaluación de los modelos, aumentando así la fiabilidad de los resultados.

Es recomendable establecer protocolos estandarizados para la obtención de imágenes médicas en investigaciones futuras. Esto permitiría mantener la

consistencia y calidad de las imágenes, lo que facilitaría un entrenamiento y evaluación más precisos de los modelos, garantizando resultados más fiables en el diagnóstico.

3. **Exploración de nuevas arquitecturas y enfoques:** Aunque VGG19 demostró un rendimiento superior en este estudio, se sugiere explorar otras arquitecturas de redes neuronales, como modelos más recientes o enfoques híbridos que combinen técnicas de deep learning con algoritmos tradicionales. Esto podría abrir nuevas oportunidades para mejorar la precisión diagnóstica en futuras investigaciones.

La integración del sistema con el modelo VGG19 en el sistema de salud pública contribuiría a una atención más rápida y eficiente, mejorando los resultados clínicos y la satisfacción del paciente. Para asegurar el éxito de esta implementación, es fundamental proporcionar la capacitación adecuada al personal médico y realizar pruebas piloto para ajustar el sistema a las necesidades específicas de cada hospital.

Se sugiere implementar el modelo VGG19 en hospitales y centros de salud públicos como una herramienta para agilizar el diagnóstico a partir de imágenes médicas. Esta integración podría no solo reducir los tiempos de espera y mejorar la calidad de atención, sino también optimizar la gestión de recursos en el sistema de salud. Para asegurar una implementación exitosa, es fundamental capacitar adecuadamente al personal médico y realizar pruebas piloto controladas antes de su despliegue a gran escala.

Además, considerando la superioridad demostrada por VGG19 en esta investigación, se

recomienda desarrollar un sistema específico basado en esta arquitectura para la detección del cáncer de próstata, utilizando un conjunto de datos adecuado. La capacidad de VGG19 para extraer características profundas y detalladas de las imágenes lo hace especialmente efectivo en aplicaciones donde la precisión diagnóstica es esencial. La incorporación de este modelo en futuros estudios podría mejorar considerablemente el diagnóstico temprano y la identificación de diversas condiciones médicas, con un impacto significativo tanto en la práctica clínica como en la investigación biomédica.

IX. BIBLIOGRAFÍA

- Forero Cuellar, O. (2019). Clasificadores supervisados del cáncer de próstata a partir de imágenes de resonancia magnética en secuencias T2.
- Alvaro Daniel Quispe Carbonel. (2020). *Predicción de edad ósea con red basada en VGG-16 y Transfer Learning*. Lima.
- Báez Benavides, P., & Armisen Yañez, R. (2012). Actualización en cáncer de próstata: generalidades y diagnóstico. *MEDICINA & LABORATORIO*, 333-353.
- Bese Robles, M., & Miró Padilla, A. (2022). *Diseño de un sistema de clasificación semiautomático para la gradación del cáncer de próstata a partir de anotaciones no expertas y redes neuronales convolucionales*.
- Carneiro Neto, P. (05 de 06 de 2020). Análisis basado en aprendizaje profundo del cáncer de próstata de MP-MRI.
- Culig, Z. B. (2020). Clinical and biological relevance of the prostate-specific antigen. *European Journal of Cancer Prevention*, 3.
- Delgado, D. D. (2016). CÁNCER DE PRÓSTATA: ETIOLOGÍA, DIAGNÓSTICO Y TRATAMIENTO. *REVISTA MEDICA DE COSTA RICA Y CENTROAMERICA LXXIII*, 707-710.
- Faez, K. (2018). Mejorar el diagnóstico temprano del cáncer de próstata mediante el uso de Redes Neuronales Artificiales y Aprendizaje Profundo. *IEEE Xplore*, 4.
- Gamboa, H. A. (2014). *Inteligencia Artificial - Principios y Aplicaciones*.
- Guajin, W., Jeremy Yuen-Chun, T., & Kup-Sze, C. (2018). Diagnóstico de cáncer de próstata en una población china mediante el uso de métodos de aprendizaje automático. *IEEE Xplore*, 4.
- Hinton, G. (2018). Deep learning—a technology with the potential to transform health care. *JAMA*, págs. 1103-1104.
- IBM. (2020). Obtenido de <https://www.ibm.com/co-es/cloud/deep-learning>
- iso25000. (3 de 10 de 2022). Obtenido de <https://iso25000.com/index.php/normas-iso-25000/iso-25010/23-usabilidad>
- Jian, X., Jia, W., Shaozhong, C., & Bilong, L. (2020). Application of a Novel and Improved VGG-19. *Conference Series*, 7.
- Lal, H., Amjad, A., Saima, R., Sharjl, S., Adnan, I., Muhammad Usama, U., . . . Doug Young, S. (2019). Aplicación del enfoque de red bayesiana a determinar la asociación entre características morfológicas extraídas de imágenes de cáncer de próstata. 16.
- Mazurowski, M. A. (2019). Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MR. *Journal of Magnetic Resonance Imaging*, págs. 939-954.
- Mena, A. R. (2023). *Clasificación del cáncer de próstata por medio de inteligencia artificial Explicable a partir de datos de expresión génica*. Granada.
- Minsa. (2021). *El plan nacional de cuidados integrales de Cáncer*. Lima: Ministerio de Salud.
- Mohamed , I., Mansoor , A., Mansour , T., Hari Kiran, V., & P., R. (2020). Una técnica de clasificación de aprendizaje automático para predicción del cáncer de próstata.
- Moral, J. D. (2021). *Diseño y desarrollo de un sistema de gradación de imágenes cancerosas de resonancia magnética de próstata utilizando algoritmos de aprendizaje profundo multimodalidad*. Valencia.
- Nishant, B., & Shrivastava, M. (2021). ResNet50-Based Effective Model for Breast Cancer Classification Using Histopathology Images. *Computer Modeling in*, 17.
- Organización Mundial de la Salud. (01 de 02 de 2022). págs. <https://www.who.int/es/news-room/fact-sheets/detail/cancer>.
- Perez, N. A. (2020). *Modelo de clasificación automática de imágenes de resonancia magnética para el diagnóstico del cáncer de próstata*. Bogotá.
- Pineau, J. V.-L. (2021). Improving reproducibility in machine learning research (A report from the NeurIPS 2019 reproducibility program). *Journal of Machine Learning Research*, págs. 1-20.
- Qingge, J., He, W., Yankui, S., & Jie, H. (2019). Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images. *MDPI*, 12.
- Rodríguez, J. S. (10 de 2020). *Mendeley Data*. Obtenido de Prostate Whole Slide Images with Gleason

- Grades Annotations: <https://data.mendeley.com/datasets/9xxm58dvs3/2>
- Rodríguez, J. S. (2020). *SICAPv2 - Prostate Whole Slide Images with Gleason Grades Annotations*. (M. Data, Editor) Obtenido de <https://paperswithcode.com/dataset/sicapv2>
- Rodríguez, S. (2020). *Going deeper through the Gleason scoring scale : An automatic end-to-end system for histology prostate grading and cribriform pattern detection*. Obtenido de <https://paperswithcode.com/dataset/sicapv2>
- Silva, J., Colomer, A., Salec, M., Molina, R., & Naranjo, V. (2020). Profundizando en la escala de puntuación de Gleason: Un sistema automático de extremo a extremo para histología de próstata, clasificación y detección de patrones cribiformes.
- Silva-Rodríguez, J. (20 de 07 de 2020). *SICAPv2 - Prostate Whole Slide Images with Gleason Grades Annotations*. Obtenido de <https://data.mendeley.com/datasets/9xxm58dvs3/1>
- Society, A. C. (2021). *American Joint Committee on Cancer*.
- Vázquez, J. R. (2020). Epidemiología del cáncer de próstata. *OF NEGATIVE & NO POSITIVE RESULTS*, 12.
- Vizcaíno-Salazar, G. J. (2017). Importancia del cálculo de la sensibilidad, la especificidad y otros parámetros estadísticos en el uso de las pruebas del diagnóstico clínico y de laboratorio. En G. J. Vizcaíno-Salazar. Obtenido de <https://docs.bvsalud.org/biblioref/2018/05/883697/importancia-calculo-sensibilidad-y-especificidad.pdf>
- Yanes Chacón, A., Villalobos Campos, N., & Cubas González, S. (2023). Cáncer de próstata: una perspectiva global.
- Yasser Mk, O., Fahima A., M., & Abdelrahman I., S. (09 de 10 de 2019). Predicción de la interacción farmacológica con adenosina Receptores que utilizan técnicas de aprendizaje automático y SMOTE. *IEEAccess*.