UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TÍTULO DEL TRABAJO DE TITULACIÓN

"DESARROLLO DE UNA APLICACIÓN WEB CON MACHINE LEARNING PARA LA DETECCIÓN DE DEEPFAKES EN CASOS DE SUPLANTACIÓN DE IDENTIDAD DIGITAL"

AUTOR

Yagual Castillo José Manuel

PROYECTO DE INTEGRACIÓN CURRICULAR

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTOR

Ing. Mónica Karina Jaramillo Infante, Mgt.

Santa Elena, Ecuador

Año 2024

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

Ing. José Sánchez Aquino, Mgt.
DIRECTOR DE LA CARRERA

Ing. Mónica Jaramillo Infante, Mgt.
TUTOR

Ing. Carlos Castillo Yagual
DOCENTE ESPECIALISTA

Ing. Marjorie Coronel Suárez, Mgt.
DOCENTE GUÍA UIC

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **YAGUAL CASTILLO JOSÉ MANUEL**, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información, el mismo que obtuvo en el análisis de plagio el 2% de similitud.

La Libertad, a los 26 días del mes de noviembre del año 2024

TUTOR

Ing. Mónica Karina Jaramillo Infante Mgt.

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, Yagual Castillo José Manuel

DECLARO QUE:

El trabajo de Titulación, Desarrollo de una Aplicación web con Machine Learning para la detección de deepfakes en casos de suplantación de identidad digital, previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

La Libertad, a los 29 días del mes de noviembre del año 2024

EL AUTOR

Jose Yagual

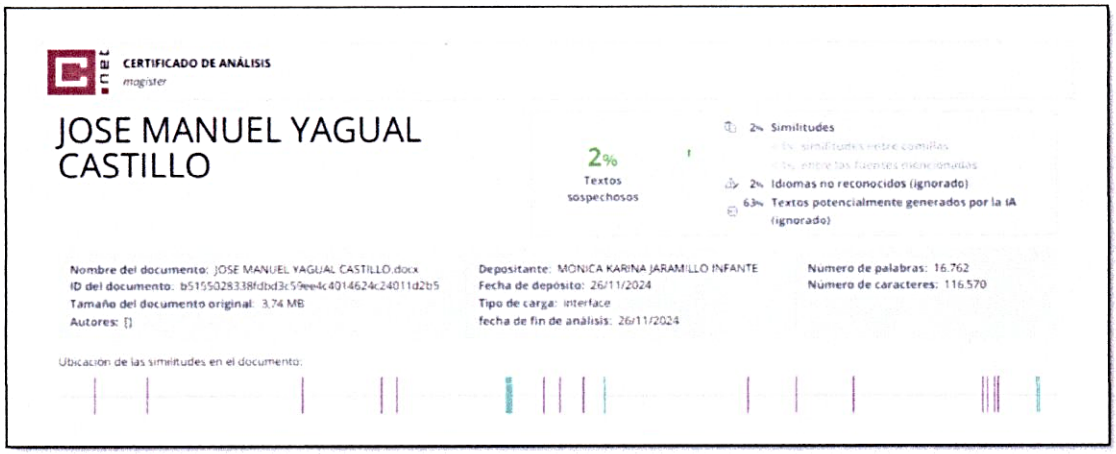
José Manuel Yagual Castillo

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Desarrollo de una Aplicación web con Machine Learning para la detección de deepfakes en casos de suplantación de identidad digital, presentado por el estudiante, Yagual Castillo José Manuel fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 2%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



CERTIFICADO DE ANÁLISIS
magister

JOSE MANUEL YAGUAL CASTILLO

2%
Textos sospechosos

2% Similitudes
0% similitudes entre comillas
0% entre las fuentes mencionadas
2% Idiomas no reconocidos (ignorado)
63% Textos potencialmente generados por la IA (ignorado)

Nombre del documento: JOSE MANUEL YAGUAL CASTILLO.docx
ID del documento: b515502838fdbd3c59ee4c4014624c24011d2b5
Tamaño del documento original: 3.74 MB
Autores: []

Depositante: MONICA KARINA JARAMILLO INFANTE
Fecha de depósito: 26/11/2024
Tipo de carga: interface
fecha de fin de análisis: 26/11/2024

Numero de palabras: 16.762
Numero de caracteres: 116.570

Ubicación de las similitudes en el documento:

TUTOR

Ing. Mónica Karina Jaramillo Infante, Mgt.

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Yagual Castillo José Manuel**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del presente trabajo de titulación con fines de difusión pública, dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor.

Santa Elena, a los 29 días del mes de noviembre del año 2024

EL AUTOR

Jose Yagual

José Manuel Yagual Castillo

AGRADECIMIENTO

Deseo expresar mi más sincero agradecimiento a todas las personas que hicieron posible la realización de este proyecto.

En primer lugar, agradezco a mi familia por su incondicional apoyo, comprensión y amor constante. Su aliento y motivación han sido fundamentales para mantenerme enfocado y decidido en este arduo proceso.

A mi tutora, Ing. Mónica Jaramillo, le estoy profundamente agradecido por su orientación experta y su constante apoyo. Sus conocimientos y valiosos consejos han sido clave para dar forma y dirección a mi investigación.

Este logro no habría sido posible sin la colaboración de cada uno de ustedes. Gracias por ser parte de este viaje académico.

Por último, quiero reconocer mis propios esfuerzos, las numerosas noches en las que me quedé trabajando hasta altas horas, dedicando tiempo a la elaboración de la documentación y a la programación del sistema. permitieron profundizar en los detalles de mi investigación y garantizar la calidad del proyecto.

José Manuel, Yagual Castillo

DEDICATORIA

Dedico este trabajo a todas las personas que han sido una fuente constante de inspiración y apoyo a lo largo de mi viaje académico.

A mi madre, Lorena Castillo Roca, cuyo amor incondicional y sacrificio han sentado las bases para que alcance mis metas. A ti, te dedico este logro con profunda gratitud y cariño.

A mi padrastro, Freddy Bazán, y a mis hermanos Justin, David, Eder y Nicole Yagual, por su aliento constante y su apoyo en cada paso que he dado. También quiero rendir homenaje a mi difunto hermano, Eithan Caleb Bazán, cuya memoria siempre será una luz en mi camino.

A todos ustedes, que de alguna manera contribuyeron a este proceso, les agradezco sinceramente por ser parte de este logro.

Este trabajo está dedicado a cada uno de ustedes como muestra de mi agradecimiento y afecto.

José Manuel, Yagual Castillo

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

Este trabajo se enfoca en el desarrollar una aplicación web para la detección de deepfakes en imágenes mediante técnicas avanzadas de aprendizaje automático. Utilizando redes neuronales convolucionales (CNN) implementadas con TensorFlow y Keras, la herramienta permitirá un análisis preciso de las imágenes para identificar manipulaciones digitales. Su objetivo es ofrecer una solución eficaz para detectar imágenes alteradas y prevenir fraudes y desinformación en el entorno digital. Para lograr esto, se aplicarán metodologías que incluyen la recopilación de datos, selección de tecnologías y diseño de algoritmos de detección. La implementación de esta aplicación mejorará la protección digital y la autenticidad en línea, elevando la efectividad en la identificación de deepfakes y fortaleciendo la seguridad del contenido visual en diversas plataformas digitales.

Palabras claves: Aprendizaje automático, detección de deepfakes, redes neuronales convolucionales.

ABSTRACT

This work focuses on developing a web application for detecting deepfakes in images using advanced machine learning techniques. By utilizing convolutional neural networks (CNN) implemented with TensorFlow and Keras, the tool will enable precise analysis of images to identify digital manipulations. Its objective is to provide an effective solution for detecting altered images and preventing fraud and misinformation in the digital environment. To achieve this, methodologies will be applied, including data collection, technology selection, and the design of detection algorithms. The implementation of this application will enhance digital protection and online authenticity, increasing effectiveness in identifying deepfakes and strengthening the security of visual content across various digital platforms.

Keywords: Machine learning, deepfake detection, convolutional neural networks**.**

INTRODUCCIÓN

En la actualidad, las tecnologías de inteligencia artificial generativa han alcanzado un grado de sofisticación tal que permite la creación de imágenes con un realismo tan avanzado que resulta cada vez más difícil para el ojo humano diferenciarlas de las originales. Este avance tecnológico representa un riesgo considerable para la sociedad, ya que facilita la suplantación de identidad y la difusión de desinformación, conocida comúnmente como 'fake news', cuyo propósito es manipular a las personas. Un elemento central de esta problemática son los 'deepfakes', técnicas que permiten alterar o reemplazar rostros en imágenes de manera extremadamente convincente, lo que ha incrementado su uso en actividades fraudulentas y engaños digitales, poniendo en duda la confianza en la autenticidad de los contenidos visuales que circulan en línea.

La proliferación de imágenes manipuladas, como los 'deepfakes', representa un desafío significativo para la ciberseguridad, facilitando tanto la suplantación de identidad como la difusión de desinformación. Para abordar esta problemática, es esencial implementar soluciones efectivas para la detección y análisis de contenido manipulado. Este proyecto tiene como objetivo desarrollar una aplicación web que empleará técnicas de machine learning, específicamente transfer learning con redes neuronales convolucionales (CNN), para identificar deepfakes. El sistema detectará patrones de manipulación en imágenes mediante técnicas de aumento de datos, logrando una alta precisión en la clasificación de contenido auténtico y manipulado.

El primer capítulo abordará los antecedentes de los deepfakes y la necesidad de un sistema eficaz para su detección, definiendo los objetivos del proyecto y justificando su relevancia. Se describirán las metodologías de investigación y desarrollo a seguir. El segundo capítulo presentará el marco teórico, con los conceptos clave sobre deepfakes y las técnicas de machine learning aplicadas a su detección. Además, se especificarán los requisitos técnicos de la aplicación web y se analizarán los resultados de las pruebas del sistema, demostrando su efectividad en el cumplimiento de los objetivos propuestos.

CAPÍTULO 1. FUNDAMENTACIÓN

1.1. Antecedentes

En la era digital, los deepfakes se han convertido en una amenaza grave para la seguridad informática y las investigaciones forenses. Deepfake se refiere a contenidos manipulados mediante inteligencia artificial (IA) que permiten el cambio de rostro de una persona, alterando su identidad y facciones [1]. La creciente perfección de estas técnicas dificulta aún más la detección de las falsificaciones, lo que facilita su uso para difundir desinformación, dañar reputaciones, realizar fraudes o extorsiones. Esto pone en riesgo la confianza en la información digital y genera inestabilidad tanto a nivel social como político.

Las redes generativas antagónicas (RGA)**,** también conocidas como Generative Adversarial Networks (GAN)**,** son fundamentales en la creación de deepfakes. Introducidas por Ian Goodfellow en 2014, esta tecnología emplea dos redes neuronales que trabajan en conjunto, pero con objetivos opuestos [2]. El generador crea contenido falso, mientras que el discriminador evalúa si el contenido generado es lo suficientemente convincente como para ser considerado real. A través de un proceso de competencia continua, ambas redes mejoran progresivamente, perfeccionando el contenido hasta que resulta extremadamente difícil de distinguir del original.

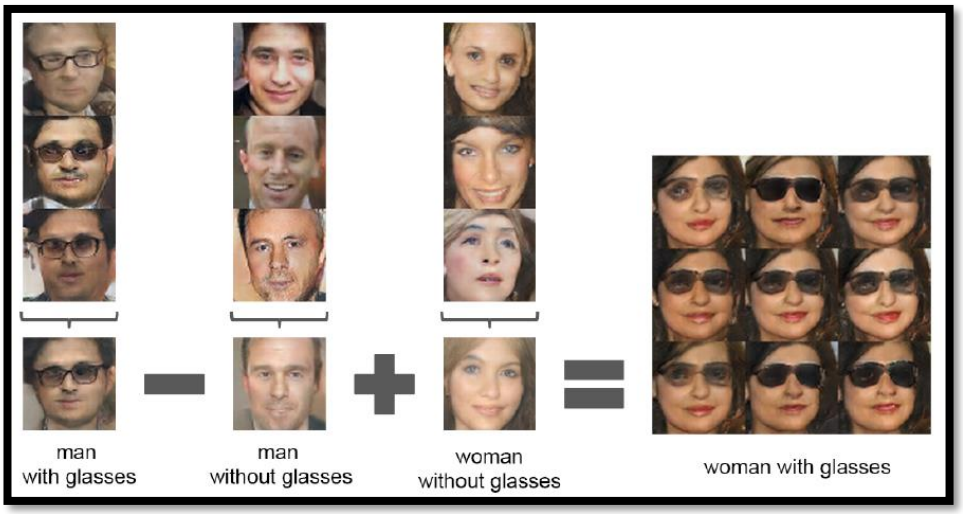


Figura 1. Aritmética vectorial para rostros generados por RGA.

Como se muestra en la **(Figura 1)**, una Red Generativa Antagónica (RGA) puede analizar miles de fotografías de una persona para crear un nuevo retrato que capture las características generales de esas imágenes sin replicar ninguna en particular. Este proceso da como resultado contenido original, ya sea una imagen, un video o un audio. En el futuro cercano, se espera que las RGA puedan funcionar con menos datos, lo que permitirá crear cambios más complejos, como intercambiar cabezas, cuerpos completos y voces de manera más precisa. Aunque actualmente se requiere una gran cantidad de imágenes para producir deepfakes realistas, los investigadores están desarrollando nuevas técnicas que podrían generar imágenes falsas a partir de una sola fotografía, como una selfie.

Una técnica relevante en los deepfakes es el Intercambio de Rostros (Faceswap), cuyo principal objetivo es reemplazar los rasgos faciales de un individuo por los de otro para generar imágenes sintéticas que parecen genuinas; este proceso se realiza mediante dos enfoques principales: uno basado en el objetivo, que transfiere la identidad facial de la imagen de origen a la imagen de destino, logrando una integración precisa y natural, y otro basado en la fuente, que también busca una integración realista. [3]. Los algoritmos de intercambio facial se utilizan a menudo para dañar la reputación de figuras públicas al crear situaciones en las que nunca estuvieron involucradas. También se emplean en la distribución de pornografía no consentida, infligiendo un daño considerable a la posición pública y al bienestar de las personas.

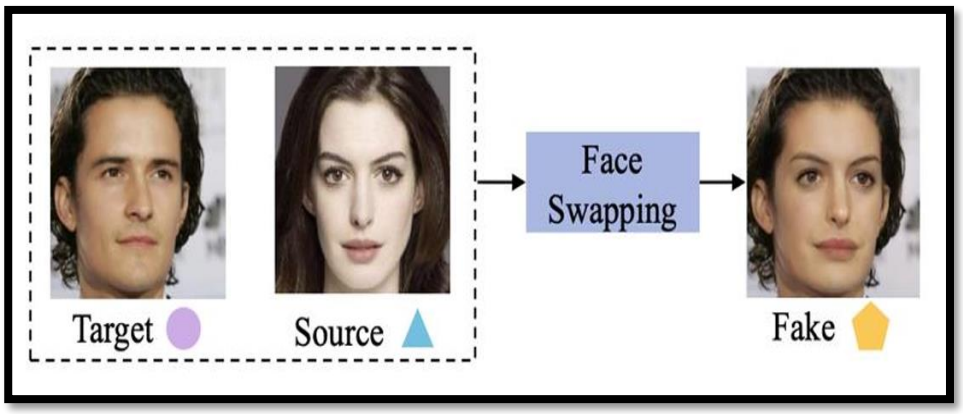


Figura 2. Proceso de Intercambio de Rostros (Faceswap).

La **(Figura 2)** ilustra el proceso de manipulación facial utilizado en los deepfakes. En este proceso, se fusiona la imagen de un individuo objetivo (target) con una imagen fuente (source) para crear una imagen final sintética (fake). En el enfoque basado en el objetivo, se transfiere la identidad facial de la imagen fuente a la imagen de destino, resultando en una integración precisa y natural de los rasgos faciales. Por otro lado, el enfoque basado en la fuente altera la imagen original utilizando atributos de la imagen de destino, lo que puede dar lugar a incoherencias debido a la falta de control sobre el contexto ambiental en la imagen resultante.

Además del intercambio de rostros, existen diversas herramientas que permiten generar imágenes de personas inexistentes, como el generador de retratos falsos disponible en "thispersondoesnotexist", que utiliza inteligencia artificial para crear rostros realistas en cuestión de segundos, empleando la red neuronal StyleGAN desarrollada por Nvidia. En 2020, este generador sorprendió al mostrar cómo la IA puede crear caras creíbles de personas inexistentes, resaltando el impacto que las redes neuronales tienen en nuestra vida diaria. [4].

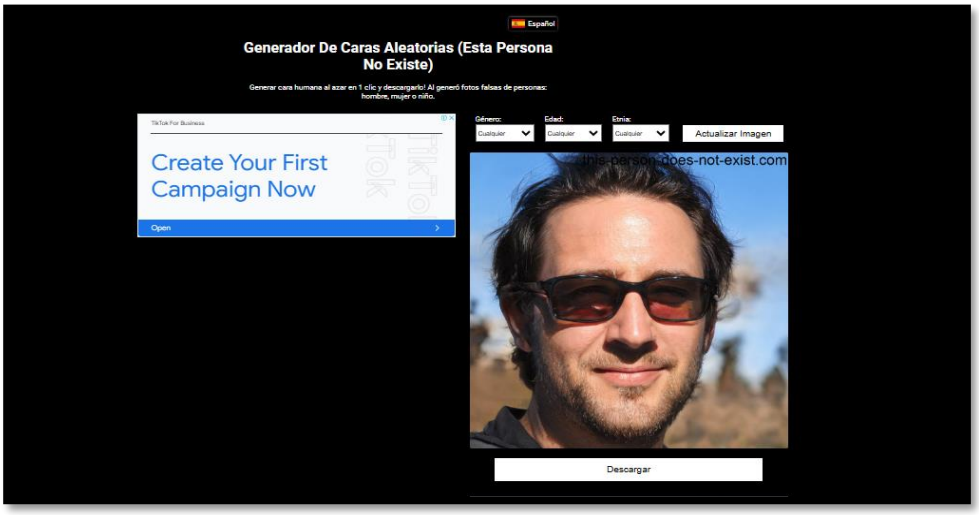


Figura 3. Retrato Falso Generado por el Sitio 'This Person Does Not Exist'.

En la **(Figura 3)**, se muestra un retrato generado por el sitio web mencionado. Este rostro no corresponde a ninguna persona real, lo que ilustra la capacidad de la inteligencia artificial para crear identidades completamente ficticias. Además, la facilidad de descargar estas imágenes es alarmante, ya que el sitio no requiere ningún tipo de registro ni información personal; simplemente permite la descarga directa de la imagen.

La capacidad de crear tales identidades ficticias plantea serias preocupaciones en términos de ética y seguridad, especialmente en contextos donde estas imágenes pueden ser utilizadas para la suplantación de identidad o fraudes. Es importante destacar que no solo existe este generador; hay múltiples herramientas en línea que permiten crear imágenes falsas con propósitos similares, lo que agrava aún más los desafíos relacionados con la desinformación y la manipulación.

El uso de deepfakes representan una grave amenaza para individuos, empresas e instituciones al permitir la creación de contenido falso con propósitos fraudulentos, de difamación o manipulación. En el ámbito legal pueden ser utilizados para estafas financieras, ciberacoso, manipulación política y obstaculizar la administración de justicia al dificultar la autenticación de pruebas digitales [5]. Además, representan riesgo significativo para la seguridad biométrica al comprometer sistemas de reconocimiento facial.

En el ámbito de las noticias, es muy probable que la industria del periodismo enfrente un desafío significativo en términos de confianza del consumidor debido al creciente uso de deepfakes, ya que estos contenidos falsos representan una amenaza superior a las noticias falsas tradicionales, siendo considerablemente más difíciles de detector. La tecnología de los deepfakes permite la creación de imágenes que parecen auténticas y pueden poner en peligro la reputación de los periodistas y los medios de comunicación al difundir contenido engañoso como si fuera real [2]. En un contexto competitivo, obtener imágenes exclusivas es crucial para los medios de comunicación; sin embargo, el riesgo aumenta si estas imágenes resultan ser falsificaciones. Además, los deepfakes pueden dificultar la alfabetización digital y la comprensión crítica del contenido, distorsionando la verdad.

El Diario “El Comercio” reportó la detención de tres presuntos ciberdelincuentes en Ecuador, tras 11 meses de investigación. Estos individuos operaban en Pichincha y Manabí, utilizaban imágenes deepfake para crear perfiles falsos de personas inexistentes que dejaban reseñas positivas sobre una tienda en línea falsa, lo que atraía a otros usuarios a realizar compras. Cuando las víctimas depositaban el dinero en las cuentas bancarias proporcionadas por los estafadores, estos cortaban toda comunicación, sin enviar nunca los productos prometidos [6].

Las estadísticas presentadas por el diario El País, los delitos informáticos que involucran el uso de deepfakes muestran una tendencia alarmante de crecimiento. Se estima que el número de estos delitos se duplica cada seis meses, lo que refleja un aumento significativo en su incidencia. Además, más del 90% de los deepfakes en circulación corresponden a pornografía no consentida, siendo aproximadamente el 95% dirigidos hacia mujeres. Estas cifras revelan un alto nivel de riesgo y vulnerabilidad, especialmente para las mujeres, quienes son las principales víctimas de este tipo de delitos cibernéticos [7].

Las repercusiones de los deepfakes van más allá de la desinformación, afectando tanto a nivel personal como social al dañar reputaciones con situaciones falsas, distorsionar percepciones públicas durante elecciones, e incluso incitar a la violencia con escenas manipuladas. Para hacer frente a estos desafíos, se plantea desarrollar una aplicación web que emplee machine learning para detectar deepfakes. Esta solución tecnológica reforzará la ciberseguridad en un entorno vulnerable a la manipulación, facilitando una identificación más precisa de contenidos falsificados y mitigando los riesgos relacionados con la desinformación y la manipulación social.

1.2. Descripción del Proyecto

El proyecto se centrará en desarrollar una aplicación web que utilice técnicas de Machine Learning para abordar el creciente problema de los deepfakes, que son contenidos sintéticos manipulados con el potencial de diseminar desinformación y comprometer la seguridad digital [5]. El foco principal radica en detectar y prevenir de manera efectiva la presencia de deepfakes en contenido multimedia, salvaguardando así la integridad y la reputación de individuos y entidades en entornos digitales.

Teniendo en cuenta los avances tecnológicos recientes, es fundamental destacar el papel de la inteligencia artificial (IA) en la detección de deepfakes. Actualmente, la IA, mediante el uso de redes neuronales convolucionales y recurrentes, ha transformado la forma en que se analiza y verifica el contenido digital. Estos algoritmos avanzados permiten identificar patrones complejos y sutiles en grandes volúmenes de datos, facilitando la distinción entre contenido auténtico y manipulado [8].

Al aplicar IA, no solo se mejora la precisión en la detección de deepfakes, sino que también se desarrollan estrategias preventivas más eficaces para proteger la integridad digital y combatir la desinformación. La tecnología mencionada jugará un papel crucial en este proyecto, por lo que se planifica la implementación de diversos módulos, que se integrarán de manera incremental en su estructura. Estos módulos permitirán desarrollar funcionalidades específicas, garantizando un enfoque ordenado y progresivo en el proceso de construcción del sistema. A continuación, se detallan los módulos previstos:

MÓDULO	DESCRIPCIÓN	VENTAJAS
MÓDULO 1: RECOPILACIÓN DE DATOS	Se emplearán herramientas web para generar imágenes falsas, junto con la recopilación de contenidos genuinos en redes sociales. Además, se utilizarán APIs que proporcionen imágenes falsas y verdaderas, lo que permitirá complementar el dataset necesario para el entrenamiento de modelos de Machine Learning.	Permite crear un conjunto de datos diverso, mejorando la calidad y la efectividad del modelo de detección.
MÓDULO 2: PROCESAMIENTO DE DATOS	Las imágenes serán preparadas para el análisis mediante técnicas de redimensionamiento, que ajusta el tamaño a un formato uniforme, normalización, que estandariza los valores de color, y extracción de características, que destaca los elementos relevantes. Se utilizarán métodos de análisis para identificar anomalías en imágenes, como inconsistencias faciales con redes neuronales convolucionales (CNNs) e irregularidades en la textura de la piel con filtros que indican manipulaciones.	Asegura que las imágenes estén en un formato uniforme y fácil de analizar, aumentando la precisión en la detección de deepfakes.

MÓDULO 3: ALGORITMOS DE DETECCIÓN	Utilizando librerías como TensorFlow y Keras, se desarrollará y entrenará un modelo de Machine Learning que incluirá redes neuronales convolucionales y recurrentes para detectar deepfakes en contenidos multimedia. La precisión del modelo se mejorará ajustando hiperparámetros como el tamaño del lote, la tasa de aprendizaje y el rango de aumentos de datos, además de ampliar el conjunto de datos para reducir errores de detección y minimizar falsos positivos y negativos.	Mejora la capacidad del modelo para identificar deepfakes de manera más efectiva y precisa, lo que resulta en una mayor confiabilidad del sistema.
MÓDULO 4: GENERACIÓN DE RESULTADOS	La imagen se enviará al backend mediante el servicio Flask que conecta con el frontend para su análisis. El modelo entrenado clasificará las imágenes como "Real" o "Fake" según un porcentaje que indica la probabilidad de autenticidad, utilizando un umbral del 50%. Además, se extraerán metadatos relevantes y se marcarán áreas sospechosas. Se generará un informe detallado que documentará los resultados de la detección, incluyendo la clasificación del contenido como verdadero o falso y explicaciones sobre las características que respaldan cada decisión.	Proporciona resultados claros y fáciles de entender, lo que ayuda a los usuarios a confiar en las decisiones del sistema y a comprender los fundamentos de cada clasificación.
MÓDULO 5: APLICACIÓN WEB	La aplicación web será la plataforma principal para la comunicación entre el usuario y el sistema de detección de deepfakes. Esto permitirá una interacción eficiente con el modelo entrenado y facilitará la visualización de los resultados.	Facilita el acceso y uso del sistema, haciendo que la experiencia del usuario sea más amigable y efectiva.

Con base en los módulos descritos, se presentará la infraestructura general del proyecto en desarrollo, ilustrando el flujo de información entre los diferentes componentes del sistema.

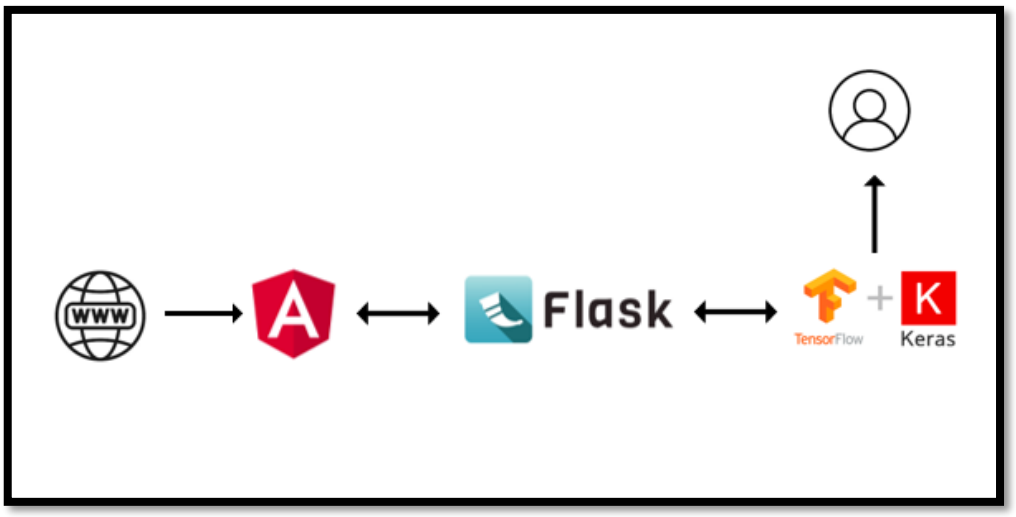


Figura 4. Infraestructura del proyecto.

Para llevar a cabo este proyecto, se hará uso de diversas tecnologías y herramientas de software:

CATEGORÍA	ELEMENTOS
Lenguajes de Programación	Python
Entornos de Ejecución	Jupyter Notebook
Frameworks	Flask, Angular, UI Bootstrap
Bibliotecas de Inteligencia Artificial	TensorFlow, Keras
Editor de Código	Visual Studio Code

De acuerdo con la Resolución RCF-FST-SO-09 No. 03-2021 del Consejo de la Facultad de Sistemas y Telecomunicaciones emitido en la Universidad Estatal Península de Santa Elena, el proyecto se encuentra alineado con la línea de investigación “Desarrollo de software (DSS)”, abordando específicamente el desarrollo de algoritmos y visión artificial aplicados en sistemas de detección de deepfakes [9].

1.3. Objetivos del Proyecto

1.3.1 Objetivo General

Desarrollar una aplicación web basada en técnicas de machine learning, utilizando redes neuronales convolucionales para la detección de deepfakes en imágenes, con el fin de mejorar la ciberseguridad y prevenir intentos de suplantación en entornos virtuales.

1.3.2 Objetivos Específicos

- Recopilar un dataset diverso de imágenes que serán utilizados para entrenar el modelo de detección de deepfakes.
- Implementar modelos de aprendizaje automático con TensorFlow y Keras para detectar con precisión deepfakes en imágenes.
- Desarrollar la interfaz del sistema utilizando el framework Angular para conectar el backend y proporcionar una experiencia de usuario fluida y eficiente.
- Evaluar la efectividad del modelo de detección de deepfakes mediante pruebas con conjuntos de datos específicos, validando la precisión del sistema en diferentes escenarios de manipulación digital.

1.4. Justificación del Proyecto

La inmersión en una cultura visual vasta, saturada de productos audiovisuales con diversos grados de fidelidad a la realidad, ha alterado la percepción de los espectadores sobre lo verosímil y lo auténtico. Los individuos enfrentan el desafío constante de distinguir entre lo real y lo artificial, y de discernir dónde empieza la manipulación y dónde concluye la realidad [10]. Esta transformación ha generado una creciente preocupación por la autenticidad de la información consumida, ya que la habilidad para crear contenidos visuales engañosos ha superado las capacidades tradicionales de verificación y análisis. Con el avance de las tecnologías de edición y generación de imágenes, la línea entre la realidad y la ficción se ha difuminado, exigiendo una mayor vigilancia y métodos sofisticados para garantizar la integridad de los contenidos.

La elección de investigar y desarrollar una aplicación web utilizando machine learning para la detección de deepfakes en imágenes responde a una necesidad urgente en el ámbito de la seguridad cibernética y la investigación forense digital. Los deepfakes tienen la capacidad de socavar la confianza, distorsionar la información y afectar áreas críticas como la política, las redes sociales y la economía, por lo que es crucial entender la naturaleza de estos contenidos falsificados, su posible impacto y desarrollar métodos efectivos para identificarlos y reducir sus efectos perjudiciales [3].

Este proyecto busca implementar una herramienta eficaz y adaptable que aproveche las capacidades de redes neuronales convolucionales, mediante el uso de tecnologías avanzadas como TensorFlow y Keras. A través de estos algoritmos, se pretende no solo detectar *deepfakes* con gran precisión, sino también mejorar continuamente el sistema frente a nuevas técnicas de manipulación, garantizando que la solución se mantenga actualizada y efectiva a largo plazo.

La suplantación de identidad mediante deepfakes no solo debilita la confianza en las instituciones y personas, sino que también constituye un delito tipificado en el Artículo 212 del Código Orgánico Integral Penal (COIP) de Ecuador. Este argumenta que “Quien, de cualquier manera, asuma la identidad de otra persona para obtener un beneficio propio o para un tercero, causando perjuicio a alguien, será castigado con una pena de prisión de uno a tres años.” [11]. La propuesta de una aplicación web para detectar deepfakes se alinea directamente con el objetivo de este artículo, al ofrecer una herramienta para prevenir y combatir este tipo de delitos cibernéticos.

En última instancia, esta aplicación contribuirá a la protección de la sociedad frente a los peligros que representan los deepfakes, preservando la integridad de la información y la confianza en los medios digitales. Al ofrecer un sistema de detección eficaz, se mitigarán los riesgos asociados con la manipulación digital, protegiendo a individuos, como a entidades y organizaciones de sus consecuencias negativas, en consonancia con lo estipulado en el Artículo 212 del COIP". Es importante destacar que, hasta la fecha, la mayoría de las soluciones para la detección de deepfakes se han centrado en el análisis teórico y la implementación de algoritmos en entornos controlados. Sin embargo, este proyecto se diferencia por su enfoque práctico y funcional.

1.5. Alcance del Proyecto

El propósito de este proyecto es desarrollar una aplicación web destinada a la detección automatizada de deepfakes en imágenes, abordando la creciente problemática de la manipulación digital que compromete la integridad de la información. La necesidad de una solución tecnológica en este ámbito es evidente, considerando el incremento de los deepfakes en el entorno digital, lo cual coloca a instituciones y usuarios en una posición vulnerable frente a la desinformación.

Mediante la implementación de técnicas avanzadas de Machine Learning, se diseñará un sistema que permita la identificación precisa de imágenes manipuladas, proporcionando a los usuarios una herramienta confiable para verificar la autenticidad de contenido visual. Además de detectar deepfakes, la aplicación incluirá funciones para gestionar imágenes analizadas, almacenar resultados y generar informes detallados. A continuación, se describen los módulos del proyecto:

Para la recopilación de datos, se utilizarán herramientas que permitan obtener un conjunto equilibrado de imágenes falsas y auténticas. Las imágenes falsas se generarán mediante plataformas en línea, mientras que las genuinas se extraerán de redes sociales. Además, se integrarán APIs que aporten contenido tanto manipulado como real, creando un dataset variado. Esto asegurará que el sistema disponga de ejemplos representativos para entrenar eficazmente los modelos de machine learning en la detección de deepfakes.

En el procesamiento de datos, se aplicarán técnicas esenciales para preparar las imágenes antes de su análisis, como el redimensionamiento, que ajustará el tamaño de todas las imágenes para eliminar variaciones de resolución que puedan afectar el rendimiento del modelo, y la normalización, que equilibrará los valores de color y brillo para garantizar uniformidad en todo el conjunto de datos.

Además, se procederá a la extracción de características clave, como texturas faciales y detalles visuales, lo que facilitará que el modelo reconozca patrones importantes para la detección de deepfakes. Este proceso es esencial para garantizar que las imágenes estén estructuradas adecuadamente, permitiendo al modelo identificar alteraciones o manipulaciones faciales con alta precisión.

En los algoritmos de detección, se desarrollarán modelos de aprendizaje automático utilizando librerías avanzadas como TensorFlow y Keras. Estos modelos incluirán redes neuronales convolucionales y recurrentes, que se entrenarán para identificar deepfakes en contenido multimedia. El rendimiento del modelo será optimizado mediante ajustes de parámetros como la tasa de aprendizaje y el tamaño de los lotes de datos. A través de la expansión del dataset y el refinamiento de los hiperparámetros, se busca reducir los falsos positivos y negativos, logrando así una mayor precisión en la detección de imágenes manipuladas.

En la generación de resultados, el análisis de las imágenes se llevará a cabo en el backend, conectado mediante el servicio Flask. El sistema clasificará las imágenes como "Real" o "Fake" basado en un umbral del 50% de autenticidad, además de extraer metadatos y marcar áreas sospechosas en las imágenes. Se generará un informe detallado que documentará los resultados del análisis, incluyendo explicaciones de por qué una imagen fue clasificada de cierta manera. Además, se incluirán visualizaciones como gráficos de pastel que mostrarán el porcentaje de autenticidad para facilitar la interpretación de los resultados.

La aplicación web ofrecerá una interfaz intuitiva donde los usuarios podrán cargar imágenes y recibir resultados del sistema de detección de deepfakes. Los informes generados incluirán la clasificación de las imágenes y explicaciones detalladas del análisis. Esta plataforma será el punto de interacción con el modelo entrenado, facilitando una experiencia accesible para todo tipo de usuarios.

El sistema será desarrollado con una arquitectura compatible con navegadores web, garantizando un acceso eficiente para los usuarios. La aplicación se enfocará en ofrecer una experiencia de usuario optimizada, priorizando tanto la velocidad como la precisión en la detección de deepfakes. En su fase inicial, el proyecto estará limitado a la detección de deepfakes en imágenes, con la posibilidad de extender estas capacidades a videos en futuras fases de desarrollo, sujeto a la disponibilidad de recursos y las necesidades detectadas durante su evolución.

1.6. Metodología del Proyecto

1.6.1. Metodología de Investigación

La investigación exploratoria se utiliza cuando el tema de estudio es relativamente novedoso o ha sido insuficientemente investigado, generando incertidumbre sobre su comprensión [12]. Esta metodología es esencial en contextos donde el conocimiento disponible puede ser limitado o disperso. En el marco de este proyecto, la investigación exploratoria servirá para llevar a cabo una revisión detallada de la literatura existente y analizar trabajos previos relevantes. El objetivo es identificar los principales retos y las prácticas efectivas en el desarrollo de aplicaciones web para la detección de deepfakes. Este análisis preliminar ofrecerá una base sólida para aplicar un enfoque más sistemático en la fase experimental.

La metodología cuantitativa se caracteriza por un enfoque sistemático para la recopilación y análisis de datos numéricos, diseñado para validar hipótesis o premisas mediante mediciones cuantitativas [13]. Este método permite identificar patrones de comportamiento y desarrollar teorías, destacando por su capacidad para cuantificar información y facilitar una medición precisa de los datos. Al centrarse en datos objetivamente medibles, asegura la exactitud de los resultados, excluyendo factores subjetivos que no pueden ser evaluados de manera efectiva. En el marco de la metodología cuantitativa se emplean herramientas como la matriz de confusión y el Area Under the ROC Curve (AUC-ROC) para evaluar el rendimiento de modelos de machine learning en la detección de deepfakes.

La matriz de confusión clasifica las predicciones en verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, lo que permite calcular métricas clave como precisión y sensibilidad. Por su parte, el AUC-ROC resume el rendimiento del modelo a través de distintos umbrales de decisión, siendo esencial para evaluar su capacidad para distinguir entre imágenes auténticas y manipuladas; un AUC elevado indica una buena clasificación. En conjunto, estas métricas aseguran un análisis riguroso y permiten realizar ajustes basados en datos objetivos para mejorar la detección de deepfakes.

Las pruebas se llevarán a cabo en un entorno controlado, donde el sistema será evaluado utilizando el conjunto de datos recolectado. Los resultados obtenidos permitirán verificar la capacidad del sistema para detectar deepfakes a través de métricas como la matriz de confusión y el área bajo la curva ROC (Receiver Operating Characteristic). Este análisis estadístico proporcionará una visión clara y objetiva sobre el rendimiento del sistema en la detección de deepfakes.

1.6.2. Beneficiarios del Proyecto

Entre los beneficiarios se incluyen:

- **Público general:** Personas que reciben o encuentran imágenes en plataformas digitales y desean asegurarse de que no han sido alteradas o manipuladas.

1.6.3. Variables

En el presente estudio, se identifican las siguientes variables:

- **Precisión en la detección de deepfakes**: Evalúa la capacidad del sistema para identificar correctamente las imágenes manipuladas, midiendo la exactitud de las detecciones realizadas. Esta variable se centra en la precisión del sistema en distinguir entre imágenes genuinas y manipuladas, asegurando que las detecciones sean correctas y confiables.

1.6.4. Análisis de recolección de datos

Una vez definida la metodología de investigación, se decidió utilizar imágenes como principal fuente de datos para este proyecto. Esta etapa se enfoca en la recolección, organización y análisis de un conjunto diverso de imágenes, tanto auténticas como manipuladas, que serán utilizadas para entrenar y evaluar los modelos de machine learning.

1.6.4.1 Recolección de Imágenes Auténticas

La recolección de imágenes auténticas es un paso fundamental en este análisis, ya que estas imágenes servirán como referencia para diferenciar contenido genuino de manipulado. Se obtendrán fotografías propias junto con imágenes de redes sociales, así como dataset disponibles en Kaggle. Este enfoque permitirá una variedad de contextos y condiciones, enriqueciendo así el conjunto de datos. Además, se prestará especial atención a la calidad y resolución de las imágenes recopiladas para asegurar que cumplan con los estándares necesarios en el desarrollo de un modelo efectivo para la detección de deepfakes.

1.6.4.2 Generación de Imágenes Manipuladas y Sintéticas (Deepfakes)

La siguiente fase del proyecto se centra en la generación de imágenes manipuladas y sintéticas. Esta etapa incluye dos enfoques distintos:

1. **Imágenes Manipuladas**: Se emplearán herramientas avanzadas de deep learning, siendo DeepFaceLab la principal por su efectividad en el intercambio de rostros y la generación de deepfakes realistas. Esta plataforma es reconocida por su eficacia en la manipulación de imágenes. Adicionalmente, Faceswap permitirá realizar alteraciones naturales en las imágenes al reemplazar rostros, complementando así el proceso de generación.
2. **Imágenes Sintéticas**: Se emplearán plataformas web especializadas para combinar distintas imágenes y crear nuevas representaciones visuales. Asimismo, se recurrirá a servicios impulsados por inteligencia artificial que generan retratos de personas inexistentes, asegurando así una amplia diversidad en las representaciones visuales obtenidas.

Cada imagen generada en ambos enfoques será evaluada para asegurar que cumpla con los estándares requeridos para el entrenamiento y la validación del modelo del modelo de machine learning. Esta evaluación incluirá la revisión de la autenticidad visual, la consistencia en los detalles faciales, y la variabilidad en las representaciones, asegurando que las imágenes sean adecuadas para mejorar la precisión del sistema.

1.6.4.3 Organización y Clasificación del Dataset

Una vez recopiladas y generadas las imágenes, se procederá a su organización y clasificación. Las imágenes se etiquetarán en dos categorías principales: "real" y "fake". Esta clasificación es fundamental para el entrenamiento de los modelos de machine learning, ya que permite distinguir claramente entre datos auténticos y manipulados. Se emplea un conjunto de datos organizado en tres particiones principales: train, validation y test, cada una de ellas con las categorías mencionadas. Esta división asegura un balance adecuado entre las clases para entrenar, validar y evaluar el rendimiento del sistema de detección de deepfakes. A continuación, se detalla la cantidad de imágenes en cada partición:

DATASET	REAL	FAKE	TOTAL
Train	5,000	5,000	10,000
Validation	1,000	1,000	2,000
Test	1,000	1,000	2,000

Tabla 1. Distribución del Dataset.

Como se muestra en la **(Tabla 1)**, la partición de “train” contiene un total de 10,000 imágenes, distribuidas equitativamente entre 5,000 imágenes reales y 5,000 falsas, representando el 71.43% del total del conjunto de datos. Las particiones de “validation” y “test” contienen 2,000 imágenes cada una, con un 14.29% del total para cada partición, manteniendo una distribución igualmente equilibrada entre imágenes reales y falsas. Este balance es esencial para garantizar un entrenamiento adecuado del modelo y evitar sesgos en las predicciones.

1.6.4.4 Evaluación del Rendimiento del Modelo

El modelo fue entrenado utilizando el conjunto de datos durante 20 Epoch. Una Epoch se refiere a un ciclo completo en el que el modelo revisa todo el conjunto de datos. Durante cada Epoch, se realizaron más de 10 iteraciones, que son pasos más pequeños en el proceso, lo que permite que el modelo ajuste sus parámetros. Los resultados obtenidos incluyen métricas clave que reflejan el desempeño del modelo, tales como la precisión, la pérdida y la tasa de aprendizaje en los conjuntos de entrenamiento y validación.

Precisión Entrenamiento	Pérdida Entrenamiento	Precisión Validación	Pérdida Validación	Tasa de Aprendizaje
98.77%	3.38%	90.30%	36.41%	0.0001

Tabla 2. Métricas de Rendimiento del Modelo.

Los resultados presentados en la **(Tabla 2)** indican que el modelo alcanzó una precisión de entrenamiento del 98.77% junto con una pérdida de 3.38%, lo que sugiere un ajuste efectivo al conjunto de datos utilizado durante el entrenamiento. Este alto nivel de precisión implica que el modelo es capaz de clasificar correctamente la mayoría de las imágenes reales y manipuladas que se le presentan durante el entrenamiento.

En cuanto al conjunto de validación, el modelo obtuvo una precisión del 90.30% y una pérdida de 36.41%. Estos resultados demuestran un rendimiento adecuado en datos no vistos, lo que es crucial para evaluar la capacidad de generalización del modelo. A pesar de que la precisión en la validación es ligeramente inferior a la del entrenamiento, sigue siendo un resultado satisfactorio que indica que el modelo mantiene su efectividad al clasificar imágenes que no formaron parte de la fase de entrenamiento.

Finalmente, la tasa de aprendizaje se mantuvo en 0.0001 durante el proceso de entrenamiento. Este valor relativamente bajo asegura un entrenamiento estable y gradual, permitiendo que el modelo ajuste sus parámetros de manera controlada y evitando saltos bruscos que podrían comprometer la convergencia.

1.6.4.5 Matriz de confusión

Se optó por utilizar la matriz de confusión como herramienta fundamental para evaluar el rendimiento del modelo de detección de deepfakes, esta métrica permite comparar las predicciones del modelo con las clasificaciones reales y facilita la identificación de su precisión y efectividad. Al analizar verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos, se pueden detectar áreas de mejora y optimizar la clasificación de contenido auténtico y manipulado.

		PREDICCIÓN	
		POSITIVOS	**NEGATIVOS**
OBSERVACIÓN	**POSITIVOS**	**Verdaderos Positivos (VP)**	**Falsos Negativos (FN)**
	NEGATIVOS	**Falsos Positivos (FP)**	**Verdaderos Negativos (VN)**

Tabla 3. Contingencia de Predicciones.

Descripciones de los términos:

- **VP:** Casos donde el modelo predijo correctamente que la imagen es real.
- **FN**: Casos donde el modelo predijo incorrectamente que la imagen es falsa cuando en realidad es real.
- **FP**: Casos donde el modelo predijo incorrectamente que la imagen es real cuando en realidad es falsa.
- **VN**: Casos donde el modelo predijo correctamente que la imagen es falsa.

Para evaluar el rendimiento del modelo, se llevó a cabo una implementación de la matriz de confusión en Python, utilizando los datos de la carpeta “Test”. Este enfoque permitió medir de manera precisa la efectividad de las predicciones generadas por el modelo al comparar los resultados obtenidos con las etiquetas reales del conjunto de datos de prueba.

La matriz de confusión proporciona una visión detallada de la clasificación de las imágenes, desglosando los verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. Este análisis no solo facilita la identificación de errores en las predicciones, sino que también brinda información valiosa que puede ser utilizada para ajustar y mejorar el modelo. Al entender mejor las áreas en las que el modelo puede necesitar optimización, se pueden realizar ajustes finos en sus parámetros, asegurando así un mejor rendimiento en futuras evaluaciones.

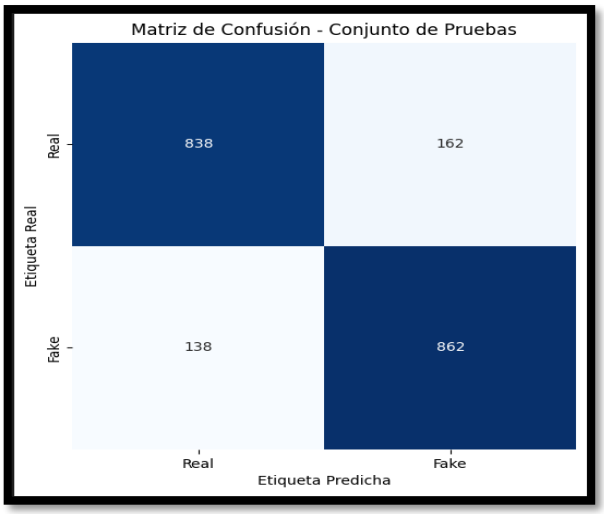


Figura 5. Resultados de Clasificación del Modelo.

El análisis de los resultados obtenidos del modelo de detección de deepfakes, presentado en la **(Figura 5)**, revela que, de un total de 2000 imágenes, se clasificaron correctamente 838 como reales (Verdaderos Positivos) y 862 como falsas (Verdaderos Negativos). Sin embargo, el modelo también mostró áreas de mejora, con 162 imágenes reales clasificadas erróneamente como falsas (Falsos Negativos) y 138 imágenes falsas etiquetadas incorrectamente como reales (Falsos Positivos). Estos errores resaltan la necesidad de optimizar el modelo para mejorar su capacidad de identificación, dado que los falsos negativos pueden permitir la difusión de contenido auténtico mal clasificado, mientras que los falsos positivos pueden validar contenido engañoso. Por lo tanto, es esencial buscar un equilibrio entre precisión y recuperación para fortalecer la fiabilidad del modelo en la clasificación de imágenes auténticas y manipuladas.

1.6.4.6 Área Bajo la Curva (AUC - ROC)

Se utilizó el área bajo la curva (AUC - ROC) como una métrica fundamental para evaluar el rendimiento del modelo de detección de deepfakes. Esta métrica proporciona una representación gráfica que ilustra de manera clara la relación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) a diferentes umbrales de decisión. Al analizar la curva ROC, se puede observar cómo varía el equilibrio entre la sensibilidad y la especificidad del modelo, lo que permite identificar el umbral óptimo para maximizar la precisión en la clasificación de imágenes auténticas y manipuladas.

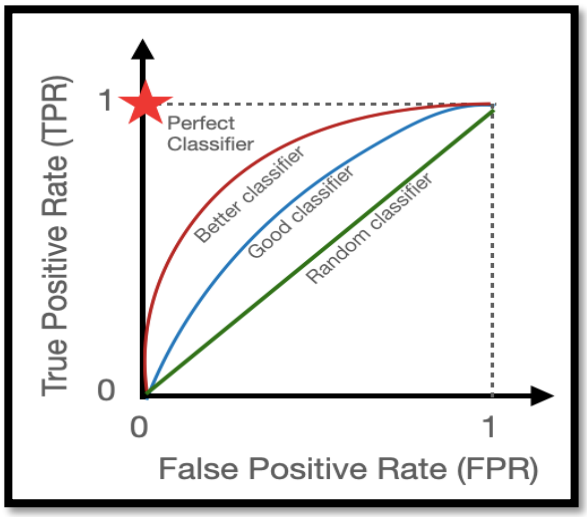


Figura 6. Gráfica del Área Bajo la Curva (AUC - ROC).

- **Valor de 1**: Indica un modelo perfecto que clasifica correctamente todos los casos positivos y negativos.
- **Valor entre 0.8 y 0.9**: Sugiere un modelo muy bueno, con alta capacidad de discriminación entre las clases.
- **Valor entre 0.7 y 0.8**: Indica un modelo aceptable, aunque podría beneficiarse de mejoras.
- **Valor entre 0.6 y 0.7**: Señala un modelo que tiene un rendimiento por encima del azar, pero con una capacidad de discriminación limitada.
- **Valor de 0.5**: Sugiere un rendimiento equivalente al azar, lo que significa que el modelo no tiene capacidad de discriminación.
- **Valor por debajo de 0.5**: Indica un modelo que discrimina de manera inversa, es decir, clasifica incorrectamente las instancias.

Para evaluar el rendimiento del modelo, se utilizó el área bajo la curva (AUC - ROC) a través de una implementación en Python utilizando los datos de prueba. Esta métrica proporcionó una visualización clara de la habilidad del modelo para diferenciar entre imágenes reales y deepfakes en distintos niveles de umbral. Al analizar las tasas de verdaderos positivos y falsos positivos, se generó un gráfico que facilita la identificación del umbral más efectivo, lo que es fundamental para mejorar la precisión del modelo en la detección de contenido manipulado.

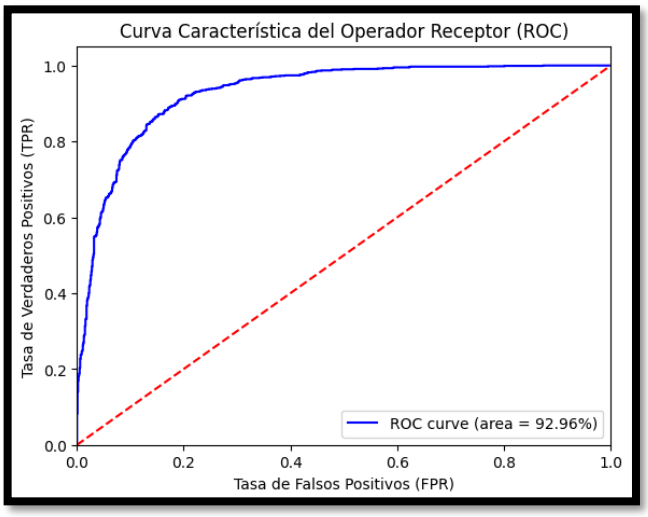


Figura 7. Resultados de la Curva ROC.

Mediante el análisis del AUC-ROC presentada en la **(Figura 7)**, se pueden examinar la sensibilidad y la especificidad del modelo, aspectos cruciales para mejorar su precisión en la clasificación de contenido auténtico y manipulado. En el caso del modelo de detección de deepfakes analizado, se obtuvo un AUC de 92.96%, lo que indica una excelente capacidad para discriminar entre imágenes reales y manipuladas. Esto sugiere que el modelo es altamente efectivo, aunque siempre hay oportunidades para seguir optimizando los umbrales de decisión y mejorar aún más su rendimiento.

1.7. Metodología de desarrollo

Este proyecto adoptará una metodología de desarrollo de software incremental que descompone el proceso en una serie de etapas sucesivas, donde cada etapa representa un avance significativo en la funcionalidad del sistema final, lo que facilita una gestión eficiente de los requisitos, así como del diseño, la implementación y las pruebas antes de avanzar al siguiente incremento.

El modelo incremental divide el proyecto en varios incrementos, cada uno de los cuales añade nuevas capacidades al producto final. Esta estructura permite un avance gradual y una validación continua del sistema a lo largo de su desarrollo, garantizando que cada funcionalidad se integre de manera efectiva y cumpla con los estándares [14].

La metodología incremental emplea un enfoque escalonado, donde el desarrollo del proyecto avanza a través de secuencias lineales. Cada etapa produce incrementos funcionales que se construyen de manera progresiva, lo que facilita la incorporación de retroalimentación y la adaptación a posibles cambios en los requisitos del usuario. Este proceso iterativo asegura que el sistema evolucione de acuerdo con las expectativas y necesidades de los usuarios finales, mejorando la calidad y la satisfacción del producto en cada fase de desarrollo.

- **Planeación**: Se establece un acuerdo detallado entre el cliente y el desarrollador, donde se define de manera precisa el alcance del proyecto, las limitaciones técnicas, los requisitos funcionales y no funcionales, así como los plazos y recursos necesarios. Este paso es fundamental para garantizar que todas las expectativas estén alineadas, y que la planificación sirva como guía estratégica para el desarrollo del sistema, incluyendo la identificación de riesgos y la asignación de responsabilidades.
- **Modelado**: En esta etapa, se lleva a cabo un análisis profundo de los procesos empresariales que pueden ser automatizados, utilizando herramientas como diagramas de flujo, mapas de procesos y estudios de caso. Esta fase permite visualizar cómo las soluciones tecnológicas impactarán en la estructura del negocio y ayuda a establecer un marco claro para el desarrollo del sistema.
- **Construcción**: Esta fase, se realiza el diseño detallado de las interfaces de usuario, seguido por el desarrollo del sistema y las pruebas correspondientes. Todo el proceso se basa en los requerimientos definidos en las etapas anteriores, asegurando que cada componente del sistema cumpla con las especificaciones acordadas. Las pruebas garantizan que el sistema funcione de manera correcta y eficiente antes de pasar a producción.
- **Despliegue**: Una vez completado y validado el desarrollo, se procede a la entrega del prototipo funcional, que ha pasado por un proceso riguroso de pruebas y ajustes. Este prototipo representa una versión parcial del sistema final, pero es lo suficientemente estable como para ser integrado con otros módulos en fases posteriores hasta culminar con el producto completo, garantizando una transición fluida a producción.

1.7.1 Implementación de la Metodología Incremental

Incremento 1: Módulo de recopilación de datos

- Uso de herramientas en línea para crear imágenes falsas.
- Obtención de imágenes auténticas de redes sociales.
- Integración de APIs para conseguir tanto imágenes reales como manipuladas.
- Creación de un conjunto de datos variado que ayudará en el entrenamiento de los modelos de detección.

Incremento 2: Módulo de procesamiento de datos

- Ajuste del tamaño de las imágenes para que todas sean del mismo formato.
- Normalización de colores para mantener una representación uniforme.
- Identificación de características importantes en las imágenes que ayuden en la detección.
- Aplicación de técnicas para encontrar anomalías, como diferencias en los rostros usando redes neuronales.

Incremento 3: Módulo de algoritmos de detección

- Desarrollo de un modelo de aprendizaje automático con herramientas como TensorFlow y Keras.
- Utilización de diferentes tipos de redes neuronales para identificar deepfakes.
- Mejora de la precisión del modelo ajustando parámetros como el tamaño de los grupos de datos y la velocidad de aprendizaje.
- Ampliación del conjunto de datos para reducir errores en la detección.

Incremento 4: Módulo de generación de resultados

- Clasificación de imágenes como "Real" o "Falsa" según un porcentaje de autenticidad.
- Creación de informes que expliquen los resultados de la detección.
- Inclusión de detalles sobre por qué se clasificó una imagen de cierta manera.

Incremento 5: Módulo de aplicación web

- Diseño de una interfaz fácil de usar que permita a los usuarios interactuar con el sistema de detección.
- Presentación clara de los resultados para que sean fáciles de entender.
- Establecimiento de una conexión efectiva entre el usuario y el modelo entrenado.

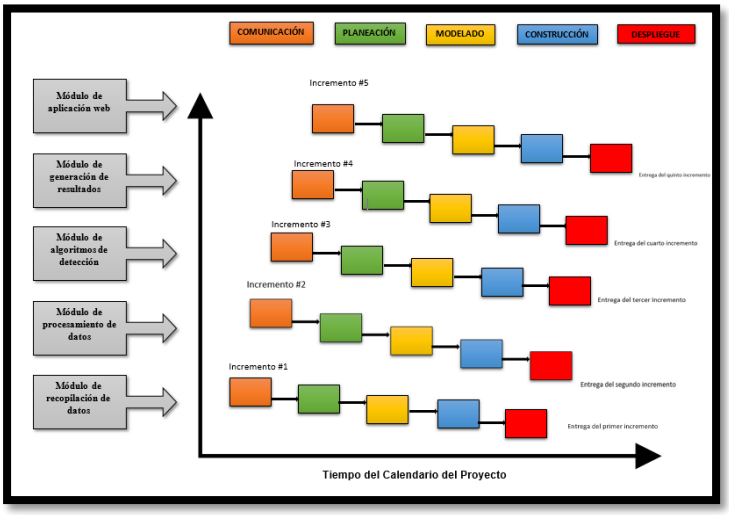


Figura 8. Modelo incremental del sistema.

CAPÍTULO 2. PROPUESTA

2.1 Marco Contextual

La expansión de las tecnologías de manipulación de imágenes, conocidas como deepfakes, impulsada por algoritmos avanzados de inteligencia artificial, ha suscitado serias preocupaciones en torno a la seguridad digital, la privacidad y la veracidad de la información. Estas técnicas permiten generar imágenes altamente realistas que dificultan su autenticación en entornos digitales, lo que facilita la suplantación de identidad y la distorsión de la información visual, afectando la confianza en el contenido compartido en línea.

En un mundo interconectado, donde la información se comparte y consume a un ritmo acelerado, las plataformas de redes sociales y otros canales digitales albergan una cantidad abrumadora de contenido visual. Esto hace urgente la detección y prevención de manipulaciones, convirtiendo el desarrollo de un sistema web basado en aprendizaje automático para detectar deepfakes en una necesidad esencial para proteger la integridad de la identidad digital y la confianza en la información.

La era digital ha transformado profundamente la manera en que consumimos y producimos contenido. Sin embargo, esta transformación también ha traído consigo nuevos desafíos, especialmente en lo que respecta a la veracidad de la información. Uno de los fenómenos más preocupantes es la aparición de los deepfakes, manipulaciones digitales que utilizan inteligencia artificial para crear imágenes, videos o audios engañosos que parecen reales. Según un estudio reciente, el 96% de los expertos en ciberseguridad considera que los deepfakes son una amenaza creciente para la privacidad y la seguridad (Franks, 2023).

Los deepfakes se basan en técnicas avanzadas de machine learning, especialmente las Redes Generativas Antagónicas (GANs). Estas redes utilizan dos modelos que compiten entre sí: un generador que crea contenido y un discriminador que evalúa su autenticidad (Karras, 2021). A medida que estas tecnologías evolucionan, la calidad de los deepfakes ha mejorado, lo que hace más difícil la detección.

El machine learning se ha consolidado como una herramienta clave en la lucha contra los deepfakes. Los modelos de aprendizaje profundo pueden analizar patrones y características específicas del contenido, detectando diferencias que pueden indicar manipulación (Mirsky, 2022). Herramientas como TensorFlow y PyTorch han facilitado este desarrollo, permitiendo la creación de modelos más robustos para la detección de contenido falso.

La creación de deepfakes plantea dilemas éticos significativos. Por un lado, pueden ser utilizados de manera creativa en cine y entretenimiento; sin embargo, su uso malintencionado puede contribuir a la desinformación y la manipulación (Harrison, 2023).

La facilidad para crear deepfakes ha facilitado la propagación de noticias falsas. Un estudio reciente indica que el 87% de los encuestados han visto contenido manipulado en redes sociales, lo que subraya la necesidad de herramientas efectivas de detección (Smith, 2023).

La suplantación de identidad digital se ha convertido en un problema serio. Las víctimas de deepfakes pueden sufrir daños a su reputación, estrés emocional y, en algunos casos, pérdidas económicas (Jones, D., & Smith, K., 2023). Esto destaca la necesidad de soluciones tecnológicas que protejan a los usuarios.

Esta propuesta se enmarca en el creciente interés académico y profesional por aplicar técnicas de machine learning en la detección de fraudes visuales. Investigaciones previas han demostrado la efectividad de modelos entrenados con conjuntos de datos diversos, destacando la importancia de contar con imágenes auténticas y manipuladas de alta calidad. A medida que las técnicas de creación de deepfakes evolucionan, los métodos de detección también deben adaptarse, lo que hace fundamental la investigación y el desarrollo en este campo.

Adicionalmente, la propuesta busca no solo detectar imágenes falsas, sino también crear un sistema que informe y eduque a los usuarios sobre las implicaciones de los deepfakes. Al empoderar a los individuos con herramientas para reconocer contenido manipulado, se espera contribuir a una mayor conciencia sobre la seguridad digital y la responsabilidad en el consumo de información.

2.2. Marco Conceptual

2.2.1 Entornos de Ejecución

Un entorno de ejecución (RTE) es un entorno de software que permite la ejecución de programas o aplicaciones, gestionando todos los recursos necesarios como la memoria, archivos y conexiones de red. Además de contener el código del programa, incluye bibliotecas y otros recursos esenciales para su funcionamiento [21]. En otras palabras, el RTE proporciona todo lo que una aplicación necesita para ejecutarse de manera efectiva, desde el manejo de recursos hasta la provisión de servicios esenciales, asegurando que el programa funcione de manera óptima en cualquier entorno.

2.2.1.1 Jupyter Notebook

Es una aplicación que permite la creación de cuadernos computacionales, documentos interactivos que combinan código, texto, datos y visualizaciones en un solo lugar. Forma parte del Project Jupyter, que se enfoca en ofrecer herramientas para la computación interactiva. Jupyter Notebook facilita la creación de prototipos, la explicación del código, la exploración de datos y el intercambio de ideas, proporcionando un entorno dinámico para la programación y la visualización de resultados [22]. En otras palabras, es una herramienta clave para desarrollar y compartir análisis de datos de manera eficiente, permitiendo a los usuarios experimentar, documentar y comunicar sus hallazgos de forma clara e intuitiva.

2.2.2 Frameworks

Un framework de programación, también conocido como marco de trabajo o estructura de software, es una infraestructura predefinida que proporciona un conjunto de herramientas, bibliotecas y patrones de diseño para facilitar el desarrollo de software o páginas web. Actúa como un esqueleto preconstruido que establece reglas, convenciones y directrices, permitiendo a los desarrolladores concentrarse en la lógica y funcionalidad específica de sus aplicaciones sin tener que crear todo desde cero [23]. Además, están diseñados para abordar tareas comunes en el desarrollo de software, como la gestión de solicitudes web, la interacción con bases de datos y la generación de interfaces de usuario.

Un framework puede cumplir varias funciones que agilizan el desarrollo de software, tales como:

- Acelerar el desarrollo de software.
- Mejorar la calidad del código.
- Facilitar el trabajo en equipo.
- Ampliar las funcionalidades.
- Fomentar la reutilización del Código.

2.2.2.1 Flask

Es un microframework web ligero y flexible escrito en Python, diseñado para facilitar la creación de aplicaciones web. Flask proporciona las herramientas esenciales para desarrollar, configurar y ejecutar aplicaciones de manera sencilla y eficiente [24]. Su enfoque minimalista permite a los desarrolladores construir desde aplicaciones simples hasta proyectos más complejos, manteniendo el control sobre las decisiones arquitectónicas y evitando imponer estructuras rígidas. Además, su amplia extensibilidad y compatibilidad con otras bibliotecas de Python lo convierten en una opción popular para desarrolladores que buscan flexibilidad y rapidez en el desarrollo web.

2.2.2.2 Angular

Es un framework y plataforma de desarrollo que permite crear aplicaciones de una sola página de manera eficiente y sofisticada. Ofrece una estructura modular que facilita el desarrollo de aplicaciones web dinámicas y escalables, brindando a los desarrolladores las herramientas necesarias para construir interfaces de usuario interactivas y responsivas [25].

Angular integra componentes clave para manejar enrutamiento, inyección de dependencias y gestión de datos, lo que optimiza el rendimiento y la organización del código. Además, su enfoque en el modularidad y la reutilización de código permite mantener aplicaciones complejas de forma ordenada y escalable, haciendo de Angular una opción ideal para proyectos que requieren un alto grado de interactividad y rendimiento en el entorno web.

2.2.2.3 UI Bootstrap

Es una biblioteca que ofrece directivas nativas para AngularJS basadas en el marcado y CSS de Bootstrap, sin necesidad de jQuery o Bootstrap JavaScript. Diseñada para AngularJS 1.4.x y versiones superiores, permite integrar fácilmente componentes de Bootstrap como modales y pestañas en aplicaciones AngularJS [26].

UI Bootstrap proporciona archivos tanto minificados como no minificados y permite personalizar la instalación según las necesidades del proyecto. La biblioteca facilita la creación de interfaces de usuario consistentes y estilizadas, siguiendo las prácticas recomendadas de AngularJS y Bootstrap.

2.2.3 Bibliotecas de Inteligencia Artificial

Son colecciones de código y funciones predefinidas que ofrecen a los desarrolladores herramientas y recursos para crear aplicaciones basadas en técnicas de inteligencia artificial [27]. Estas bibliotecas incluyen algoritmos, modelos de aprendizaje automático, funciones para el procesamiento de datos y otros componentes que facilitan el desarrollo de aplicaciones inteligentes.

Su propósito es simplificar la integración de capacidades avanzadas de IA en las aplicaciones, permitiendo a los desarrolladores centrarse en la implementación y personalización de soluciones en lugar de construir cada componente desde cero.

2.2.3.1 TensorFlow

Es una biblioteca de aprendizaje profundo de código abierto creada y mantenida por Google, que facilita la programación de flujo de datos para diversas tareas de aprendizaje automático. Está optimizada para ejecutarse en múltiples CPUs y GPUs, y también es compatible con sistemas operativos móviles, lo que la hace versátil para una amplia gama de aplicaciones [28].

TensorFlow incluye envolturas para varios lenguajes de programación, incluidos Python, C++ y Java, etc, permitiendo a los programadores seleccionar el lenguaje que mejor se adapte a sus necesidades y optimizar el rendimiento de sus aplicaciones de inteligencia artificial.

2.2.3.2 Keras

Es una biblioteca de redes neuronales de código abierto escrita en Python, que opera sobre plataformas como Theano o TensorFlow. Diseñada con un enfoque en la modularidad, rapidez y facilidad de uso, Keras simplifica el proceso de construcción y experimentación con algoritmos de aprendizaje profundo [29]. Su interfaz intuitiva y flexible permite a los desarrolladores e investigadores diseñar, entrenar y evaluar modelos complejos de manera eficiente, haciendo de Keras una herramienta valiosa tanto para principiantes como para expertos en el campo del aprendizaje automático.

2.2.3.3 Deep Learning

El Deep Learning (DL) es una subrama del aprendizaje automático (Machine Learning) que se inspira en la estructura y funcionamiento del cerebro humano, utilizando redes neuronales artificiales (RNA) para procesar datos de manera similar a cómo lo hacen las neuronas en el cerebro [30]. El DL se caracteriza por trabajar con modelos formados por múltiples capas de neuronas artificiales, de ahí el término "deep" (profundo), lo que le permite analizar y aprender patrones en grandes cantidades de datos de manera más eficiente y precisa.

1. **Capas de Entrada y Salida:**
    - Los datos ingresan a la red a través de la capa de entrada, donde las neuronas reciben la información.
    - Después de ser procesados a lo largo de varias capas ocultas, el resultado final se genera en la capa de salida.
2. **Capas Ocultas:**
    - Estas capas intermedias procesan los datos mediante cálculos matemáticos, buscando patrones y características específicas en la información de entrada. A medida que se añaden más capas a la red, esta se vuelve más "profunda", lo que le permite aprender y representar características más complejas y abstractas. Este aumento en la profundidad de la red contribuye a mejorar su capacidad para realizar tareas como el reconocimiento de imágenes y el procesamiento del lenguaje natural.

3. **Entrenamiento de la Red:**
    - La red ajusta sus pesos y sesgos en función de los errores cometidos durante el proceso de predicción, utilizando un método llamado retropropagación del error.
    - Minimiza el error entre la predicción y el valor real. Cuanto más se entrena la red, mejor será su capacidad de hacer predicciones precisas.
4. **Algoritmos de Optimización:**
    - Durante el entrenamiento, se emplean algoritmos como Gradiente Descendente para ajustar los pesos de las conexiones entre neuronas y mejorar el rendimiento de la red.
5. **Uso de Grandes Volúmenes de Datos:**
    - Es especialmente útil cuando se trabaja con grandes volúmenes de datos, ya que le permite aprender patrones complejos y hacer predicciones con alta precisión.

2.2.3.3.1 Redes Neuronales Convolucionales (CNN)

Las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés) son un tipo de arquitectura en aprendizaje profundo (deep learning) especialmente diseñada para procesar datos con una estructura de grilla, como imágenes. A diferencia de las redes neuronales tradicionales, las CNN están optimizadas para manejar la alta dimensionalidad de las imágenes mediante la utilización de capas convolucionales que reducen la cantidad de parámetros necesarios para el aprendizaje. Su funcionamiento se basa en imitar el procesamiento visual humano al identificar características visuales clave, como bordes, texturas, formas, y patrones [31].

Estructura

1. **Capa de Convolución**: Aplica filtros que se desplazan sobre la imagen para extraer características relevantes.

2. **Capa de Reducción (Max Pooling)**: Reduce la dimensión de los mapas de características, conservando la información más significativa y mejorando la eficiencia del modelo.
3. **Capas Densas (Fully Connected)**: Integran las características extraídas para realizar la clasificación final.

Características

- **Extracción Automática de Características**: Aprenden a identificar patrones sin necesidad de intervención manual.
- **Invariancia Espacial**: Reconocen objetos independientemente de su posición en la imagen.
- **Eficiencia Computacional**: Manejan grandes volúmenes de datos de manera efectiva.

2.2.4 Lenguajes de Programación

Es una herramienta utilizada para controlar las acciones de una máquina, especialmente una computadora. Se compone de un conjunto de reglas que establecen tanto la forma en que deben escribirse las instrucciones (sintaxis) como el significado de cada una de ellas (semántica) [32]. Estas reglas permiten que los programadores estructuren y den sentido a sus códigos para que la máquina los interprete y ejecute correctamente.

2.2.4.1 Python

Es un lenguaje de programación robusto y fácil de aprender, que ofrece estructuras de datos de alto nivel eficientes y un sistema de programación orientado a objetos sencillo pero eficaz. Su sintaxis clara y su tipado dinámico, junto con su naturaleza interpretada, lo hacen ideal tanto para scripting como para el desarrollo rápido de aplicaciones en una amplia variedad de áreas y plataformas [33]. Su flexibilidad y capacidad de adaptarse a distintos entornos lo convierten en una opción preferida tanto para principiantes como para expertos, permitiendo la creación de soluciones complejas de manera ágil y eficiente.

2.2.5 Editor de Código

Son programas esenciales para gestionar el código fuente en proyectos, sobre todo cuando se trabaja con varios lenguajes como HTML, JavaScript, CSS o PHP [34]. Adicionalmente, ofrecen funciones avanzadas como autocompletado, resaltado de sintaxis, gestión de versiones y verificación automática de errores, lo que hace el trabajo mucho más ágil y eficiente

2.2.5.1 Visual Studio Code

Visual Studio Code es un editor de código ligero pero potente, que se instala localmente y es compatible con los sistemas operativos Windows, macOS y Linux. Ofrece soporte nativo para lenguajes como JavaScript, TypeScript y Node.js, y se distingue por su extensiva gama de extensiones que permiten trabajar con otros lenguajes y entornos de desarrollo, incluyendo C++, C#, Java, Python, PHP, Go y .NET [35]. Su capacidad para integrar múltiples lenguajes y herramientas, junto con características avanzadas como el autocompletado, el depurador incorporado y la gestión de versiones, lo convierte en una opción muy versátil y eficiente para desarrolladores

2.2.6 Postman

Es una herramienta versátil y potente diseñada para la creación, prueba, documentación y colaboración en APIs. Su amplio conjunto de características y su activa comunidad de usuarios la convierten en una solución preferida para el desarrollo de APIs [36]. Entre sus funcionalidades destacadas se incluyen:

- **Seguimiento de procesamiento**: Permite monitorear y analizar el flujo de las solicitudes y respuestas para asegurar que se procesen correctamente.
- **Envío de solicitudes HTTP**: Facilita el envío de solicitudes utilizando los métodos básicos como POST, PUT, DELETE y GET, así como otros métodos HTTP.
- **Verificación de conexiones**: Ofrece herramientas para comprobar la integridad y el estado de las conexiones API, asegurando que las interacciones entre cliente y servidor sean efectivas.

2.3. Marco Teórico

2.3.1. Imágenes falsas, efectos reales. Deepfakes como manifestaciones de la violencia política de género

En los últimos años, el uso de *deepfakes* en la sátira política ha ido en aumento, como se evidenció en el caso del "Equipo E- de España", donde los rostros de candidatos presidenciales fueron superpuestos en personajes de la serie *El Equipo A*. Sin embargo, el impacto de los *deepfakes* va más allá del humor. En 2018, Jordan Peele creó un video en el que Barack Obama parecía insultar a Donald Trump, utilizando esta tecnología como advertencia sobre los peligros de la desinformación. A pesar de su intención preventiva, los *deepfakes* también han sido empleados para ataques más dañinos, como el caso de Greta Thunberg, cuyo rostro fue insertado en un video pornográfico. Este tipo de manipulación no solo vulnera la privacidad de las víctimas, sino que también perpetúa la violencia de género en el entorno digital, afectando mayormente a mujeres y figuras públicas [37].

En este sentido los *deepfakes* pueden considerarse una forma de *gendertrolling*, un concepto introducido por Mantilla que describe un tipo de acoso dirigido específicamente a mujeres. Este tipo de troleo no se limita a simples burlas o agresiones momentáneas en línea, sino que, en muchos casos, se caracteriza por su naturaleza misógina, la participación masiva de usuarios coordinados y su prolongación en el tiempo. Las acciones de *gendertrolling* abarcan desde la creación de imágenes hipersexualizadas o memes machistas hasta amenazas graves como el acoso, la intimidación o incluso amenazas de violación o muerte, reflejando las mismas formas de violencia que existen en la vida offline [38].

2.3.2. Inteligencia artificial contra la desinformación: fundamentos, avances y retos

La Inteligencia Artificial (IA) es una tecnología emergente que desempeña un papel crucial en las sociedades contemporáneas y futuras. Su impacto en la lucha contra la desinformación es significativo, ofreciendo herramientas avanzadas para identificar y mitigar la propagación de contenidos falsos. Sin embargo, este campo de investigación también suscita un amplio debate ético y social.

Entre las principales preocupaciones se encuentran la potencial sustitución de empleos, la influencia desproporcionada en favor de sectores privilegiados, la invasión de la privacidad de los datos personales, y la posibilidad de que la IA supere en inteligencia y eficiencia a los seres humanos [39].

El aprendizaje automático (AA) es un componente esencial de la Inteligencia Artificial (IA) que se enfoca en construir sistemas inteligentes a partir de grandes volúmenes de datos. Se basa en el reconocimiento de patrones para hacer predicciones sobre eventos futuros (análisis predictivo) o descubrir relaciones entre datos (análisis descriptivo) [40].

Dentro del aprendizaje automático, existen dos categorías principales:

Aprendizaje Supervisado

En el aprendizaje supervisado, el modelo se entrena con un conjunto de datos que ya incluye etiquetas o resultados conocidos. Durante este proceso, el sistema aprende a asociar características específicas de las entradas con sus respectivas salidas, permitiéndole predecir o clasificar nuevos datos en función de lo que ha aprendido. Este enfoque abarca dos principales tipos de problemas: clasificación, que se ocupa de asignar categorías discretas a las entradas, y regresión, que se enfoca en predecir valores numéricos continuos. Entre las técnicas comunes están los árboles de decisión para clasificación y la regresión lineal para estimaciones numéricas [40].

Aprendizaje No Supervisado

Trabaja con datos sin etiquetas previas, buscando identificar patrones o estructuras intrínsecas dentro de los datos. El objetivo es agrupar o clasificar los datos en función de sus similitudes y diferencias, sin la guía de etiquetas externas. Un método destacado en este enfoque es el clustering (agrupamiento), que organiza los datos en clústeres o grupos basados en características comunes.

El clustering puede ser particional, como el algoritmo K-Means, que divide los datos en grupos separados, o jerárquico, que organiza los datos en una estructura de árbol con diferentes niveles de agrupación. Este tipo de aprendizaje permite descubrir la estructura subyacente en los datos y facilita su análisis al identificar grupos similares [40]

2.3.3. Análisis de Imágenes: El Papel de las CNNs en la Identificación de Deepfakes

El reconocimiento de deepfakes en imágenes se ha convertido en un aspecto crucial de la ciberseguridad, donde las tecnologías de inteligencia artificial juegan un papel fundamental. Los algoritmos especializados utilizados en este proceso pueden detectar artefactos visuales sutiles, como distorsiones geométricas, irregularidades en los bordes, o inconsistencias en la iluminación y la profundidad de campo, aspectos que a menudo pasan desapercibidos al ojo humano. Las CNNs, en particular, son capaces de analizar las imágenes a nivel de píxel, detectando patrones que los generadores de deepfakes no logran replicar, como ligeras variaciones en el tono de piel o la tasa de parpadeo. Estas herramientas resultan clave para identificar de manera precisa las imágenes manipuladas, ayudando a combatir fraudes visuales y la desinformación. [41].

El uso de redes neuronales convolucionales (CNNs) en entornos como Ubuntu ha demostrado ser una técnica efectiva para la detección de deepfakes. Las CNNs son especialmente adecuadas para el procesamiento de imágenes debido a su capacidad para analizar la estructura espacial de las mismas, identificando patrones y características a diferentes niveles de granularidad. Esta capacidad es esencial para detectar deepfakes, ya que estos pueden ocultar manipulaciones sutiles que las CNNs están diseñadas para identificar con alta precisión. El enfoque en redes neuronales convolucionales mejora significativamente la eficacia en la identificación de alteraciones visuales sofisticadas, permitiendo detectar manipulaciones que podrían pasar desapercibidas con otros métodos [42].

Para entrenar los modelos de detección, se utilizan conjuntos de datos disponibles a través de plataformas como GitHub, incluyendo recursos como FaceForensics e IMD8.wiki. En este proceso, se emplean técnicas de reconocimiento facial para enfocar la detección en el rostro del sujeto, dado que cualquier información fuera de esta área puede ser considerada ruido y complicar la identificación precisa de las manipulaciones. Con el fin de minimizar este ruido y mejorar la exactitud del reconocimiento, se utiliza OpenCV, una biblioteca de procesamiento de imágenes que permite filtrar y concentrarse en las características clave del rostro, facilitando así una detección más eficiente de alteraciones visuales [42].

2.4. Requerimientos

2.4.1. Requerimientos Funcionales

2.4.1.1. Módulo de Gestión de Archivos

Código	Especificación de requisitos
RF-01	Los usuarios podrán seleccionar imágenes desde su sistema local para su análisis.
RF-02	Solo se aceptarán los formatos de imagen: jpg, .png, .jpeg, .gif, .bmp, .tiff y. webp.
RF-03	La previsualización de la imagen seleccionada será mostrada antes de proceder con la carga.
RF-04	Notificaciones sobre el éxito o error al cargar una imagen serán proporcionadas al usuario, indicando el resultado de la operación.
RF-05	El sistema permitirá al usuario eliminar una imagen seleccionada antes de enviarla para análisis.
RF-06	La aplicación incluirá una opción para cancelar la selección de la imagen antes de proceder con el análisis.

Tabla 4. Requerimiento Funcional – Módulo de Gestión de Archivos.

2.4.1.2. Módulo de Procesamiento de Imágenes

Código	Especificación de requisitos
RF-07	El Sistema enviará la imagen seleccionada al backend a través de una API REST utilizando el servicio de Flask para su análisis.
RF-08	Todos los metadatos relevantes de la imagen cargada, que puedan influir en el análisis de autenticidad, serán extraídos.

RF-09	Se calculará el porcentaje de autenticidad de la imagen analizada, determinando si es real o fake.
RF-10	Áreas sospechosas en la imagen serán identificadas y marcadas para su visualización en la interfaz.
RF-11	Un indicador de carga (loader) será mostrado hasta que se complete el análisis y se reciban los resultados.
RF-12	El sistema realizará un análisis por capas de la imagen para detectar elementos no visibles que podrían ser indicadores de manipulación.
RF-13	Proporcionar una alerta de error si el análisis falla, permitiendo al usuario reintentar o cargar otra imagen para el análisis.

Tabla 5. Requerimiento Funcional – Módulo de Procesamiento de Imágenes.

2.4.1.3. Módulo de Presentación de Resultados

Código	Especificación de requisitos
RF-14	Presentar los resultados en una estructura de tres columnas en la interfaz.
RF-15	Los resultados del análisis serán presentados en una interfaz que incluirá: ➢ La imagen original. ➢ Un indicador de autenticidad (real o fake) y el porcentaje de autenticidad. ➢ Información sobre áreas sospechosas detectadas.
RF-16	Se mostrará una tabla con los metadatos extraídos de la imagen analizada.

Tabla 6. Requerimiento Funcional – Módulo de Presentación de Resultados.

2.4.1.4. Módulo de Generación de Informes

Código	Especificación de requisitos
RF-17	Al hacer clic en el botón "Generar Informe", se presentará un modal con dos opciones: visualizar el informe o descargar el informe en formato PDF.
RF-18	El contenido del informe será el siguiente: ➢ Un diagrama de pastel que represente el porcentaje de autenticidad. ➢ Una tabla con los metadatos extraídos de la imagen. ➢ Información sobre áreas sospechosas, en el caso de que la imagen sea fake, si la imagen es real, se incluirá un párrafo indicando que no se encontró nada sospechoso.
RF-19	Al seleccionar la opción de visualizar el informe en el modal, se abrirá una nueva pestaña en el navegador donde se presentará el informe completo.
RF-20	Incluir una nota explicativa en el informe que indique que no se encontraron áreas sospechosas en caso de que la imagen sea real

Tabla 7. Requerimiento Funcional – Módulo de Generación de Informes.

2.4.1.5. Módulo de Interacción con el Usuario

Código	Especificación de requisitos
RF-21	Se proporcionará una interfaz intuitiva que permita a los usuarios navegar entre las funcionalidades de manera clara y accesible.
RF-22	Se mostrarán mensajes de confirmación al usuario antes de realizar acciones que puedan alterar datos importantes, como la carga de imágenes o la generación de informes.

RF-23	Se manejarán entradas incorrectas, proporcionando mensajes de error específicos que informen al usuario sobre el problema encontrado.
RF-24	Implementar un diseño responsivo que permita el uso de la aplicación en dispositivos móviles y de escritorio.

Tabla 8. Requerimiento Funcional – Módulo de Interacción con el Usuario.

2.4.2. Requerimientos no Funcionales

Código	Especificación de requisitos
RNF-01	El sistema deberá ser accesible e utilizable desde cualquier navegador web.
RNF-02	La interfaz del sistema será responsiva, adaptándose a diferentes tamaños de pantalla y resoluciones para facilitar su uso en dispositivos móviles y de escritorio.
RNF-03	El tiempo de respuesta del sistema para cargar imágenes y procesar resultados no deberá superar los 10 segundos.
RNF-04	El sistema deberá permitir la detección de imágenes deepfakes, imágenes de personas inexistentes y aquellas con filtros aplicados.
RNF-05	La interfaz de usuario deberá ser intuitiva y fácil de usar, permitiendo a usuarios no técnicos navegar y utilizar todas las funciones del sistema sin dificultad.
RNF-06	El sistema deberá ser escalable, permitiendo la incorporación de nuevas funcionalidades y mejoras sin afectar el rendimiento existente.

Tabla 9. Requerimiento no Funcional del sistema.

2.5. Componente de la Propuesta

2.5.1. Arquitectura del Sistema

El modelo cliente-servidor es un tipo de arquitectura informática donde se distingue entre dos partes: el cliente, que solicita servicios, y el servidor, que provee dichos servicios. En este sistema, el cliente, generalmente un ordenador o una aplicación, envía solicitudes de recursos o información a través de una red, mientras que el servidor procesa esas solicitudes y responde con los datos o servicios requeridos [43].

Este proceso mejora la eficiencia porque el servidor centraliza el procesamiento y es capaz de gestionar varias solicitudes al mismo tiempo. Mientras tanto, el cliente no requiere de muchos recursos para cumplir su función, ya que solo se encarga de enviar peticiones y mostrar los resultados.

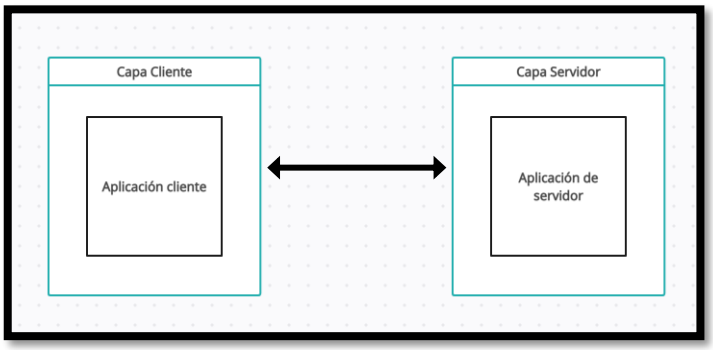


Figura 9. Arquitectura Cliente/Servidor en dos capas.

La arquitectura de dos capas se refiere a una estructura donde el cliente interactúa directamente con el servidor sin intermediarios [43]. En este modelo, el cliente envía las solicitudes al servidor, y este responde utilizando sus propios recursos y capacidades de procesamiento.

No se requieren componentes externos o sistemas adicionales para gestionar las solicitudes o procesar la información, lo que simplifica el flujo de comunicación. Esta arquitectura es especialmente adecuada para aplicaciones que manejan una carga moderada, donde las solicitudes pueden ser atendidas de manera eficiente y directa por el servidor.

2.5.2 Desarrollo de la aplicación web de detección de deepfakes

2.5.2.1 Creación del dataset

El primer paso en el desarrollo del sistema fue la creación del dataset necesario para entrenar y validar el modelo de machine learning. Este proceso se realizó siguiendo las siguientes etapas:

1. **Búsqueda y recolección de imágenes reales**: Las imágenes genuinas fueron obtenidas de diversas fuentes en la web, asegurando la diversidad en términos de calidad, resolución y contexto.

2. **Generación de deepfakes**: Para las imágenes falsas, se utilizaron aplicaciones web especializadas en la generación de deepfakes. Estas herramientas permitieron crear contenido falso a partir de las imágenes reales previamente recolectadas, garantizando que las falsificaciones fueran de alta calidad y difíciles de detectar a simple vista. Además, se exploraron sitios web dedicados a la generación y almacenamiento de deepfakes para aumentar la variedad del dataset.

2.5.2.2 Organización del dataset

El dataset se organizó en una estructura de carpetas clara para facilitar el proceso de entrenamiento y validación del modelo. Esta estructura es la siguiente:

- **Carpeta principal**: Contiene dos subcarpetas denominadas ¨train¨ y ¨validation¨, correspondientes a los conjuntos de datos de entrenamiento y validación, respectivamente.

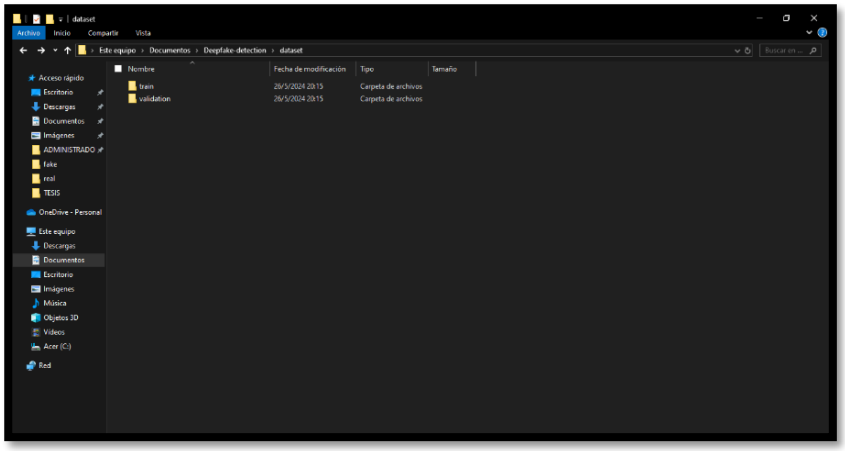


Figura 10. Distribución del Dataset en Carpetas.

Cada una de estas carpetas contiene 5000 imágenes seleccionadas cuidadosamente para asegurar una distribución equilibrada y representativa del conjunto de datos. Esta distribución equitativa es crucial para evitar sesgos durante el entrenamiento y para que el modelo pueda distinguir eficazmente entre imágenes reales y falsas.

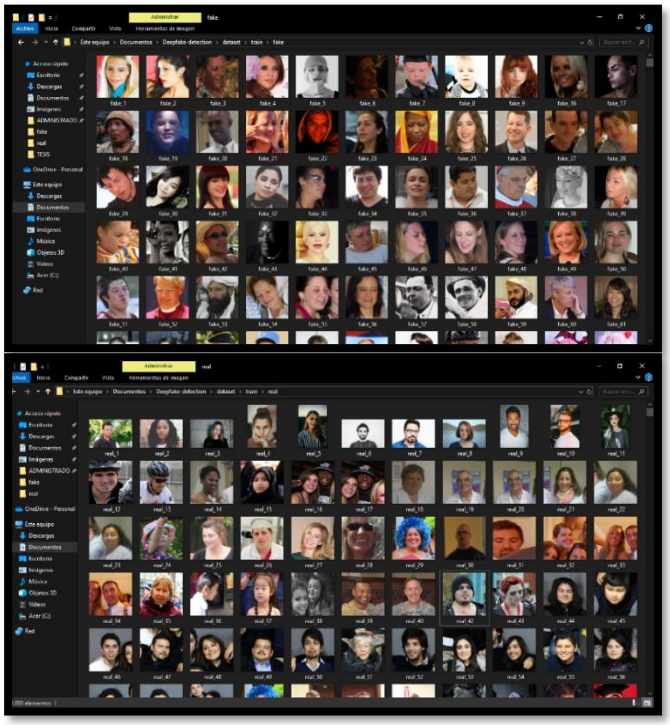


Figura 11. Organización Equilibrada de imágenes por Categorías.

En el desarrollo de esta investigación, se realizó un análisis comparativo entre tres arquitecturas de redes neuronales convolucionales (CNN): **MobileNetV2**, **EfficientNetB0**, y **ResNet50**, con el objetivo de determinar cuál modelo tiene un mejor desempeño en la detección de *deepfakes*. Estas arquitecturas fueron seleccionadas debido a su reconocimiento en el ámbito de la visión por computadora y su capacidad para adaptarse a diferentes aplicaciones.

Cada modelo fue entrenado utilizando el mismo conjunto de datos, configuraciones de hiperparámetros similares, y un esquema de entrenamiento dividido en una sola iteración con un enfoque experimental. Se registraron métricas clave como la precisión en el conjunto de entrenamiento, la precisión en el conjunto de validación, la pérdida en ambas fases, y métricas de clasificación en el conjunto de prueba.

Este análisis permitió identificar cuál de los modelos logra un balance óptimo entre precisión y generalización para la tarea de detección de *deepfakes*.

2.5.2.3 Resultados Comparativos

A continuación, se presenta una tabla comparativa de los resultados obtenidos por cada modelo tras el entrenamiento y evaluación:

Modelo	Precisión Entrenamiento	Pérdida Entrenamiento	Precisión Validación	Pérdida Validación	Precisión de Prueba	Recall	F1-Score	Tasa de Aprendizaje
MobileNetV2	99.83%	0.52%	87.95%	54.01%	85%	85%	85%	0.00001
EfficientNetB0	54.61%	46.40%	60.85%	44.34%	65%	65.56%	65%	0.00001
ResNet50	60.86%	39.15%	65.75%	45.78%	70%	70.75%	70%	0.00001

Tabla 10. Comparación de Desempeño de los Modelos de Aprendizaje Profundo.

2.5.2.4 Matrices de Confusión para el Conjunto de Entrenamiento

Los tres modelos entrenados (MobileNetV2, EfficientNetB0 y ResNet50) fueron puestos a prueba utilizando las matrices de confusión tanto en el conjunto de entrenamiento como en el conjunto de pruebas.

La matriz de confusión de MobileNetV2 destaca por su capacidad para diferenciar eficazmente entre imágenes reales y falsas, alcanzando un rendimiento notable en el conjunto de entrenamiento. Esto refleja un modelo robusto y bien entrenado.

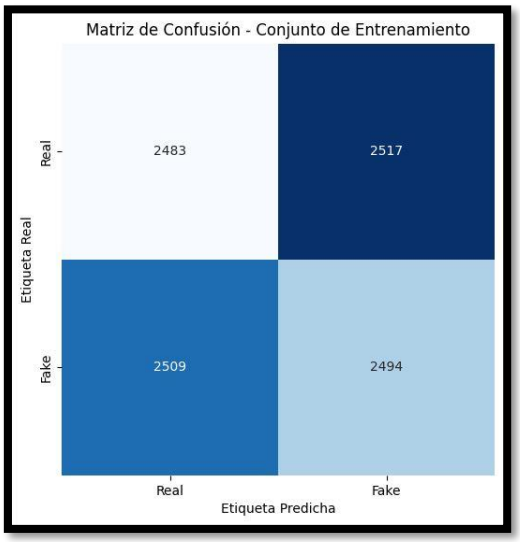


Figura 12. Análisis de desempeño de MobileNetV2 en el conjunto de entrenamiento.

En el caso de EfficientNetB0, la matriz de confusión muestra que el modelo presenta un rendimiento inferior, con un mayor número de errores en la clasificación durante el entrenamiento. Este desempeño evidencia que el modelo no logró aprender de manera óptima a distinguir entre imágenes reales y falsas.

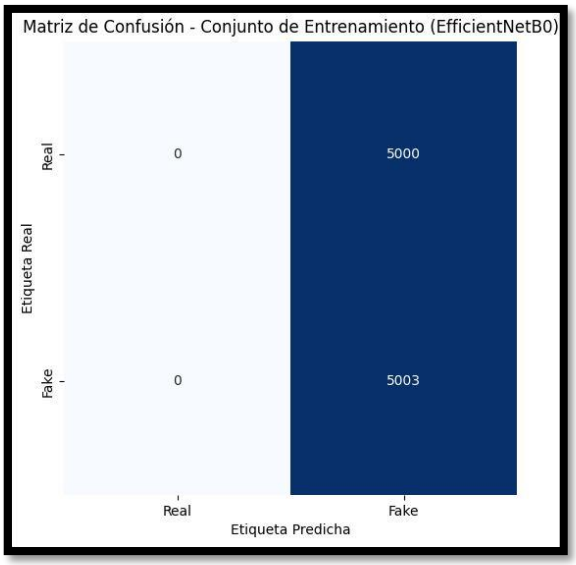


Figura 13. Análisis de desempeño de EfficientNetB0 en el conjunto de entrenamiento.

La matriz de confusión de ResNet50 refleja también un desempeño limitado durante el entrenamiento, con un nivel de errores superior al de MobileNetV2. Esto indica que el modelo podría beneficiarse de ajustes adicionales para mejorar su capacidad de clasificación.

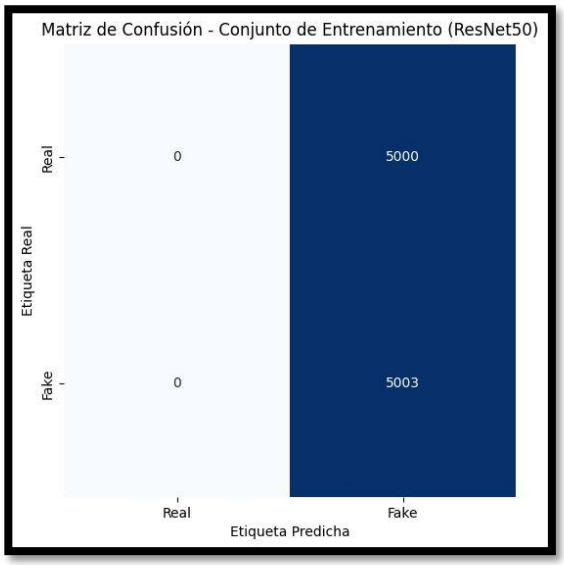


Figura 14. Análisis de desempeño de ResNet50 en el conjunto de entrenamiento.

2.5.2.5 Matrices de Confusión para el Conjunto de Pruebas

Los tres modelos también fueron sometidos a evaluación en el conjunto de pruebas mediante matrices de confusión. Este análisis es crucial para determinar la capacidad de generalización de cada modelo al trabajar con datos no vistos previamente. A continuación, se presentan las matrices de confusión obtenidas:

La matriz de confusión de MobileNetV2 resalta por su desempeño sobresaliente. Este modelo demuestra una alta capacidad para generalizar al clasificar correctamente la mayoría de las imágenes en el conjunto de pruebas, tanto reales como falsas. Esto sugiere que MobileNetV2 mantiene consistencia en su rendimiento, siendo una opción robusta para la tarea de detección de deepfakes.

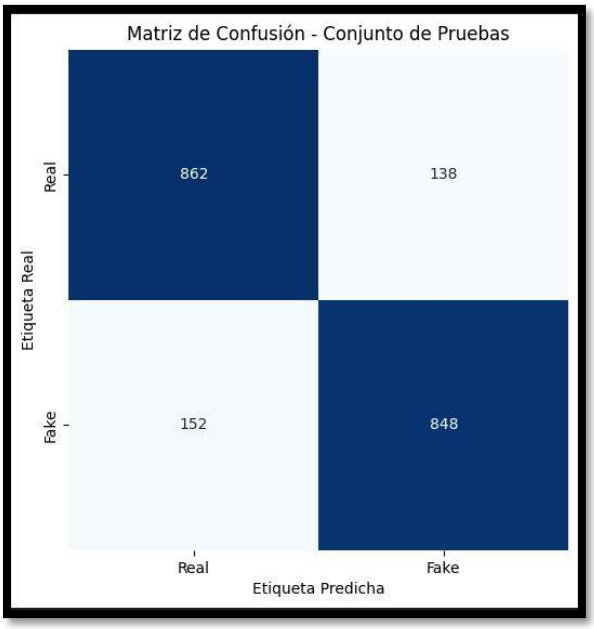


Figura 15. Evaluación del desempeño de MobileNetV2 en el conjunto de pruebas.

Aunque EfficientNetB0 mostró un rendimiento competitivo durante el entrenamiento, su matriz de confusión en el conjunto de pruebas refleja un nivel significativo de errores. Esto plantea dudas sobre su capacidad de generalización al trabajar con datos no vistos previamente, indicando que podría requerir ajustes adicionales en su arquitectura o hiperparámetros.

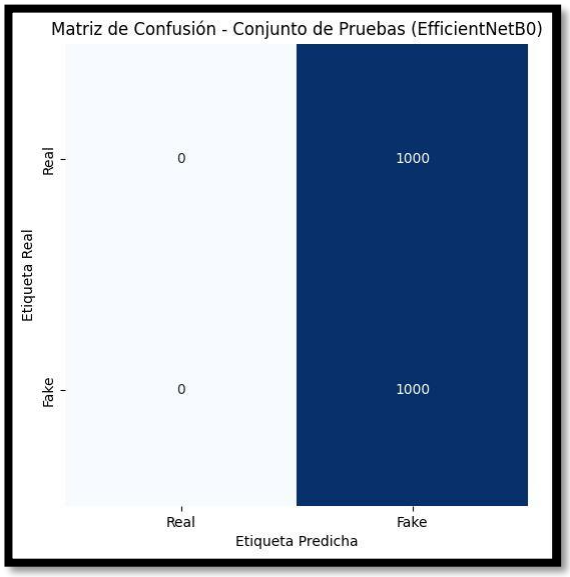


Figura 16. Evaluación del desempeño de EfficientNetB0 en el conjunto de pruebas.

Por su parte, ResNet50 muestra una matriz de confusión que refleja un comportamiento similar al observado en el conjunto de entrenamiento. Sin embargo, su capacidad de generalización es limitada, lo que resulta en un desempeño menos favorable al clasificar las imágenes del conjunto de pruebas. Este modelo parece enfrentar dificultades para adaptarse a variaciones presentes en datos nuevos.

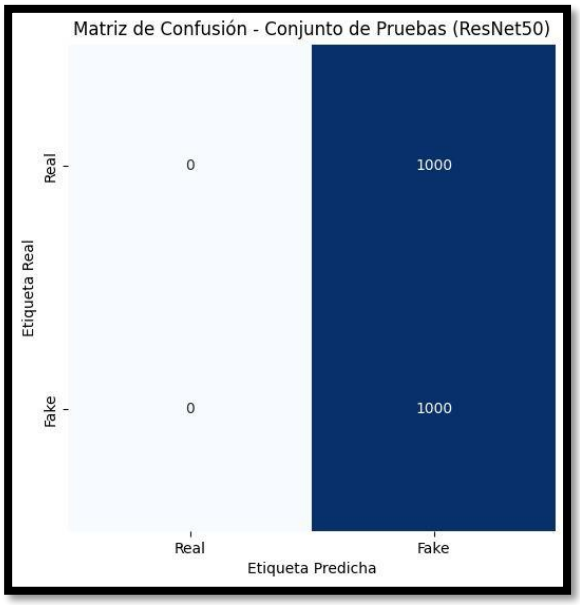


Figura 17. Evaluación del desempeño de ResNet50 en el conjunto de pruebas.

2.5.2.6 Análisis del AUC de los modelos

Se realizó el análisis del desempeño de tres modelos de redes neuronales convolucionales, MobileNetV2, EfficientNetB0 y ResNet50, utilizando la métrica del Área Bajo la Curva (AUC) en un conjunto de datos de prueba. El AUC es una métrica esencial para evaluar la capacidad de un modelo para distinguir entre las clases de imágenes reales y falsas, siendo especialmente útil en tareas de clasificación binaria, como la detección de deepfakes. A continuación, se presentan los resultados obtenidos para cada uno de los modelos, los cuales se ilustran mediante gráficas del área bajo la curva.

Modelo	AUC
Base (MobileNetV2)	0.9250615
EfficientNetB0	0.5021260000000001
ResNet50	0.544818

Figura 18. Resultados AUC de modelos

El modelo MobileNetV2 alcanzó un AUC de 92.51%, lo que indica una excelente capacidad de discriminación entre las clases de imágenes reales y falsas. Este resultado resalta la efectividad de MobileNetV2 en la tarea de detección de deepfakes, demostrando su robustez al clasificar correctamente tanto las imágenes reales como las generadas. La gráfica de AUC para MobileNetV2, presentada a continuación, refleja su alto rendimiento en el conjunto de pruebas.

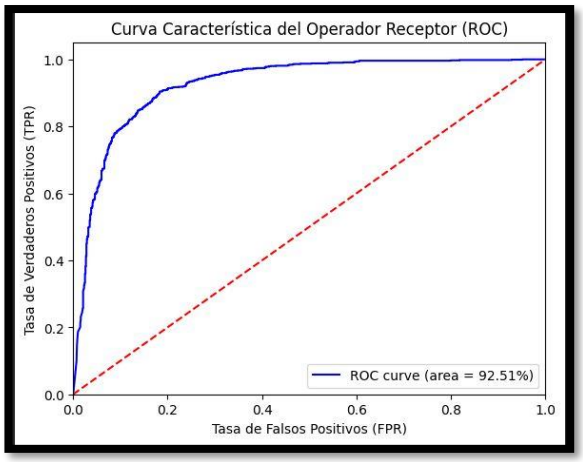


Figura 19. Curva ROC de MobileNetV2 en el Conjunto de Pruebas.

En contraste, el modelo EfficientNetB0 presentó un AUC significativamente más bajo, con un valor de 50.21%**.** Este resultado sugiere que EfficientNetB0 enfrenta dificultades para generalizar en esta tarea, lo que limita su capacidad para distinguir eficazmente entre imágenes reales y falsas. La gráfica del AUC de EfficientNetB0 a continuación muestra la reducción en el desempeño del modelo en comparación con MobileNetV2.

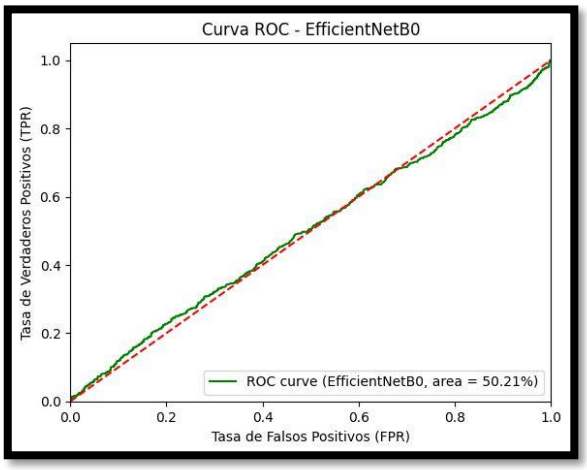


Figura 20. Curva ROC de EfficientNetB0 en el Conjunto de Pruebas.

De manera similar, ResNet50 obtuvo un AUC de 54.48%, lo que refleja una capacidad de discriminación limitada entre las clases de imágenes. Aunque el modelo muestra un desempeño superior al de EfficientNetB0, su AUC todavía es considerablemente bajo en comparación con MobileNetV2. La gráfica correspondiente a ResNet50 ilustra esta diferencia en el desempeño, destacando su limitación en la generalización al trabajar con imágenes de prueba

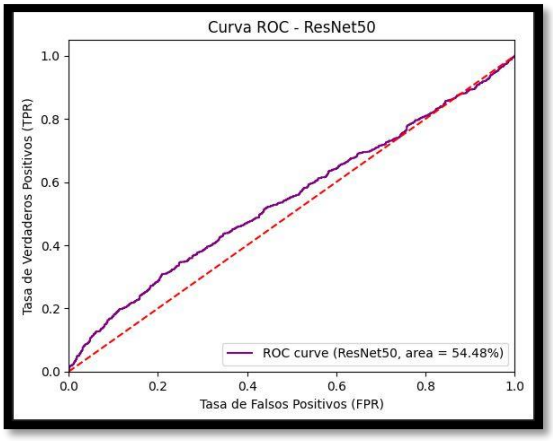


Figura 21. Curva ROC de ResNet50 en el Conjunto de Pruebas.

2.5.3. Diagramas de casos de uso

En el marco del modelado de lenguaje unificado, un diagrama de casos de uso es una representación visual que ilustra las interacciones entre los actores y el sistema. Estos actores, que pueden ser individuos o sistemas externos, se representan con figuras sencillas, conectadas a funciones específicas o casos de uso, que se muestran mediante elipses etiquetadas.

Este esquema permite identificar de manera precisa las interacciones posibles, ayudando a comprender los requerimientos funcionales del sistema y ofreciendo una visión clara de cómo se relacionan los actores con sus funcionalidades [44].

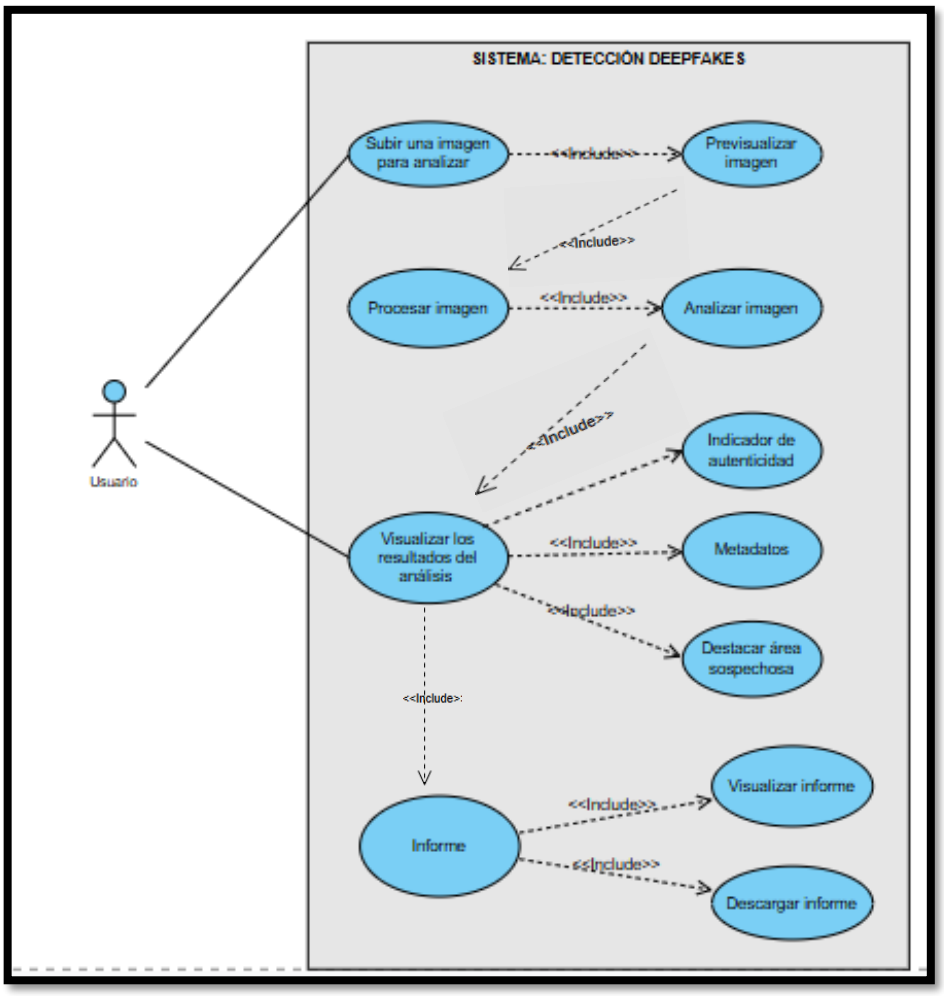


Figura 22. Caso de uso general del sistema.

Caso de Uso	01 - Procesamiento de Imagen para Detección de Deepfakes
Descripción	Este caso de uso describe cómo el usuario sube una imagen para que sea procesada y analizada con un modelo entrenado de detección de deepfakes.
Actor	Usuario.
Evento desencadenador	El usuario selecciona una imagen para subir desde su dispositivo y solicita su análisis.
Precondición	El usuario debe tener una imagen válida (jpg, .png, .jpeg, .gif, .bmp, .tiff) para subir.
Secuencia principal	1. El usuario accede a la funcionalidad de Subir Imagen. 2. El sistema solicita que el usuario seleccione una imagen. 3. El usuario selecciona la imagen y confirma la subida. 4. El sistema aplica filtros de validación a la imagen. 5. El sistema muestra una previsualización de la imagen. 6. El usuario confirma que desea analizar la imagen. 7. El sistema inicia el análisis con el modelo entrenado 8. El sistema presenta los resultados del análisis, incluyendo: • Porcentaje de realidad de la imagen. • Metadatos extraídos de la imagen. • Áreas sospechosas detectadas en la imagen. 9. El usuario tiene la opción de generar un informe PDF con los resultados del análisis.
Errores / Alternativas	ER–01 El formato de la imagen no es soportado. ER–02 El tamaño de la imagen excede el límite permitido. ER–03 Fallo al cargar la imagen por problemas de conectividad. ER–04 Error interno en el análisis del modelo.
Postcondición	Retorna HTTP 200 – OK y muestra los resultados del análisis.

Tabla 11. Caso de uso procesamiento de imagen.

2.5.4. Modelado de Datos

2.5.4.1. Diagrama de Flujo del proceso de detección de deepfake

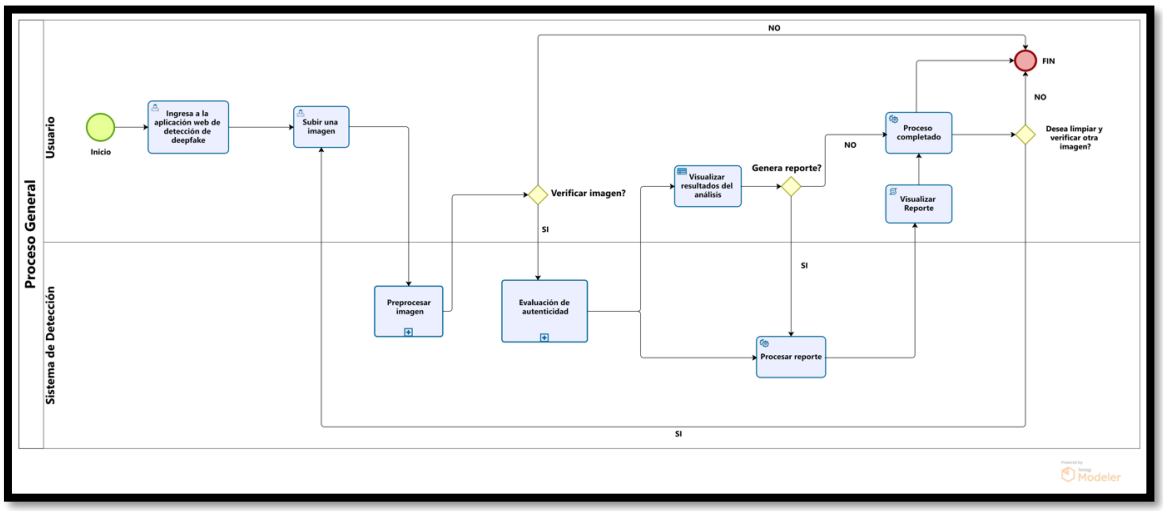


Figura 23. Diagrama del proceso general.

En la **(Figura 11)**, se describe la secuencia de interacción entre el usuario y el sistema de detección de deepfakes, iniciando cuando el usuario sube una imagen para su análisis, por consiguiente, verifica el tipo y formato del archivo, asegurándose de que corresponda a uno de los formatos permitidos como JPG, JPEG, PNG, GIF, BMP o TIFF; si el archivo no cumple con estos requisitos, el proceso se detiene y se genera un mensaje de error que se notifica al usuario. En caso de que el formato sea correcto, el sistema continúa con la fase de preprocesamiento de la imagen, donde se ajustan aspectos técnicos como la resolución y se aplican optimizaciones para que la imagen esté lista.

Una vez completado pasa a la evaluación de autenticidad, aplicando el modelo de machine learning que analiza la imagen en busca de manipulaciones y determina si es genuina o un deepfake. Durante este proceso, si se detecta que la imagen ha sido alterada, el sistema resalta visualmente las áreas sospechosas de manipulación. A su vez, se extraen los metadatos de la imagen, que proporcionan información adicional. Posteriormente, presenta los resultados de manera visual al usuario, mostrando tanto los porcentajes (real y fake) de que la imagen como las zonas específicas que han sido marcadas.

Finalmente, se genera un reporte detallado que documenta los resultados del análisis, incluyendo las métricas relevantes y cualquier información clave, y este reporte se pone a disposición del usuario junto con la visualización completa. Los subprocesos detallados, como la validación del formato de la imagen y la evaluación de autenticidad**,** se explican con mayor profundidad en los anexos correspondientes **(Véase Anexo 1 al 2)**.

2.6. Diseño de Interfaces

2.6.1. Interfaz principal – Menú

Figura 24. Interfaz principal de la aplicación web.

2.6.2. Interfaz para el análisis de imágenes

Figura 25. Interfaz para el análisis de imágenes de la aplicación web.

A continuación, se describen los componentes principales de la interfaz de la aplicación web para el análisis de imágenes:

Interfaz - Menú

- **Encabezado estático**: Permanece visible al cambiar de pantalla
- **Fondo de imagen**: Proporciona una estética atractiva y moderna.
- **Contenedor**:
    - **Imagen interactiva**: Al hacer clic en esta imagen, el usuario es redirigido a la interfaz para el análisis de imágenes.

Interfaz - Análisis de imágenes

- Área principal donde el usuario puede cargar la imagen a analizar.
- **Botones funcionales**:
    - **Verificar**: Inicia el proceso de análisis de la imagen.
    - **Limpiar**: Permite borrar la imagen cargada.
    - **Generar PDF**: Facilita la creación de un informe en formato PDF con los resultados del análisis realizado.

2.7. Pruebas

Es fundamental garantizar el correcto funcionamiento de una aplicación web de detección de deepfakes, ya que está orientado a analizar imágenes cargadas por los usuarios y proporcionar resultados precisos. Asegurar que el sistema funcione de manera óptima es clave para ofrecer un servicio confiable y eficiente. A continuación, se detallan las pruebas realizadas para asegurar la eficacia del sistema:

- Validar que el formato de la imagen cargada por el usuario sea uno de los permitidos (JPG, JPEG, PNG, GIF, BMP, TIFF).
- Verificar que el sistema realice el preprocesamiento de la imagen correctamente antes de iniciar el análisis de deepfakes.

- Comprobar que el análisis de deepfakes proporcione resultados claros y precisos, indicando si la imagen es genuina o ha sido manipulada.
- Validar que, en caso de detectar un deepfake, el sistema marque las áreas manipuladas en la imagen para una mejor visualización.
- Confirmar que los metadatos de la imagen se extraigan y se muestren correctamente al usuario.
- Verificar que se genere un reporte en PDF con los resultados del análisis.
- Comprobar el correcto funcionamiento de los botones "Verificar", "Limpiar" y "Generar PDF".
- Validar que se presenten mensajes de error al usuario en caso de cargar una imagen no válida o si se produce algún fallo durante el proceso.

<table>
<tr><th colspan="3">Información del caso de prueba</th></tr>
<tr><td>Caso de prueba N°</td><td colspan="2">01</td></tr>
<tr><td>Caso de uso</td><td colspan="2">Verificar que acepte formatos de imagen permitidos</td></tr>
<tr><td>Objetivos de la prueba</td><td colspan="2">Comprobar que el sistema solo se acepten archivos en los formatos JPG, JPEG, PNG, GIF, BMP o TIFF.</td></tr>
<tr><td>Rol</td><td colspan="2">Usuario</td></tr>
<tr><th colspan="3">Pasos de la prueba</th></tr>
<tr><td colspan="3">1. El usuario selecciona un archivo en formato incorrecto (por ejemplo, PDF).
2. El usuario intenta cargar el archivo en el sistema.
3. El sistema muestra un mensaje de error indicando que el formato no es válido.</td></tr>
<tr><th colspan="3">Resultados de la prueba</th></tr>
<tr><th>Resultados esperados</th><th colspan="2">Evaluación</th></tr>
<tr><td>El sistema debe rechazar el archivo e indicar el formato incorrecto.</td><td>Exitoso: ✓</td><td>Fallido:</td></tr>
</table>

Tabla 12. Prueba de funcionalidad - Verificación de formato de imagen.

Información del caso de prueba		
Caso de prueba N°	02	
Caso de uso	Verificar el análisis de deepfake	
Objetivos de la prueba	Comprobar que el sistema detecte si una imagen es un deepfake o real.	
Rol	Usuario	
Pasos de la prueba		
1. El usuario carga una imagen deepfake. 2. El sistema analiza la imagen y determina su autenticidad. 3. El sistema muestra el resultado al usuario, indicando si es genuina o un deepfake.		
Resultados de la prueba		
Resultados esperados	**Evaluación**	
El sistema debe identificar correctamente la autenticidad de la imagen.	**Exitoso: ✓**	**Fallido:**

Tabla 13. Prueba de funcionalidad - Análisis de deepfake.

Información del caso de prueba	
Caso de prueba N°	03
Caso de uso	Generar reporte en PDF con los resultados del análisis.
Objetivos de la prueba	Comprobar que el sistema pueda generar un reporte en PDF con los resultados del análisis de la imagen.
Rol	Usuario
Pasos de la prueba	

1. El usuario realiza el análisis de una imagen. 2. El sistema permite generar un reporte en PDF con los resultados. 3. El usuario hace clic en "Generar PDF".		
Resultados de la prueba		
Resultados esperados	**Evaluación**	
El sistema debe generar el PDF con los resultados del análisis, incluyendo detalles como el nivel de confianza y los metadatos de la imagen.	**Exitoso: ✓**	**Fallido:**

Tabla 14. Prueba de funcionalidad - Generación de reporte en PDF.

Información del caso de prueba	
Caso de prueba N°	04
Caso de uso	Restablecimiento de la interfaz de usuario.
Objetivos de la prueba	Comprobar que el botón de "Limpiar" restablezca correctamente todos los campos de la interfaz.
Rol	Usuario
Pasos de la prueba	
1. El usuario carga una imagen válida. 2. El usuario hace clic en el botón "Limpiar". 3. El sistema debe limpiar todos los campos y eliminar la imagen seleccionada.	
Resultados de la prueba	
Resultados esperados	**Evaluación**

El sistema debe restablecer la interfaz a su estado inicial, sin imágenes cargadas.	**Exitoso: ✓**	**Fallido:**

Tabla 15. Prueba de funcionalidad - Limpieza de campos de la interfaz.

Información del caso de prueba		
Caso de prueba N°	05	
Caso de uso	Verificar el manejo de errores del servidor.	
Objetivos de la prueba	Comprobar que el sistema maneje correctamente los errores del servidor durante el análisis de deepfakes.	
Rol	Usuario	
Pasos de la prueba		
1. El usuario carga una imagen válida. 2. El servidor presenta una falla durante el análisis. 3. El sistema muestra un mensaje de error informando al usuario sobre el problema del servidor.		
Resultados de la prueba		
Resultados esperados	**Evaluación**	
El sistema debe notificar al usuario que ha ocurrido un error en el servidor, cancelar automáticamente el análisis	**Exitoso: ✓**	**Fallido:**

Tabla 16. Prueba de funcionalidad - Manejo de errores del servidor.

Se llevaron a cabo cinco pruebas para evaluar diversas funcionalidades del sistema en situaciones distintas. Estas incluyeron la validación de formatos de imagen permitidos, la detección de deepfakes, la generación de reportes en PDF, el restablecimiento de la interfaz de usuario y la gestión de errores del servidor. Cada prueba fue diseñada para asegurar el correcto desempeño del sistema y optimizar la experiencia del usuario. Para más detalles, **(Véase Anexo 3 al 8)**.

2.8. Resultados

2.8.1 Análisis de resultados

Se llevó a cabo la recolección de datos sobre sobre la identificación de deepfakes, seguida de un estudio para identificar las dificultades que enfrentan los usuarios al cargar imágenes para su verificación. Esta investigación condujo a la definición de requisitos clave para el desarrollo de una aplicación web destinada a automatizar el análisis de contenido falso, en el marco de este proyecto de titulación.

Se realizaron pruebas de funcionalidad para evaluar el rendimiento de la aplicación, lo que permitió medir su efectividad y eficiencia en el procesamiento de imágenes. A partir de estas evaluaciones, se creó un cuadro comparativo que resalta las diferencias entre el análisis manual de imágenes y la solución automatizada

Análisis de pruebas realizadas	
Dificultades halladas	**Solución aplicada**
Se presentó el desafío de implementar un sistema que permitiera cargar imágenes en diversos formatos.	Se desarrolló un sistema de carga que acepta todos los formatos de imagen (JPG, JPEG, PNG, GIF, BMP y TIFF). Las imágenes se convierten al formato RGB y se redimensionan a un tamaño estándar antes de su procesamiento. Además, se incorporaron mensajes de error claros para notificar sobre archivos no válidos, mejorando así la experiencia del usuario.

Durante el desarrollo, se identificó que los largos tiempos de espera para el análisis de autenticidad de las imágenes afectaban el rendimiento de la aplicación, dificultando una interacción fluida.	Se optimizó el proceso de análisis mediante la automatización y la implementación de algoritmos avanzados de detección de deepfakes. Esto permitió reducir el tiempo de respuesta a menos de 10 segundos.
Se constató que la presentación de los resultados del análisis carecía de claridad, lo que dificultaba la identificación de las secciones alteradas en las imágenes.	Se rediseñó la interfaz de la aplicación para presentar los resultados de manera clara y accesible, incorporando visualizaciones que destacan las áreas afectadas y ofreciendo explicaciones detalladas sobre el análisis realizado.

Tabla 17. Resultados - Análisis de pruebas.

2.8.2. Comparación entre Herramientas de Detección de Deepfakes: AI Image Detector vs. Sistema Propuesto

Para comprender mejor las capacidades del sistema de detección de deepfakes propuesto, resulta útil compararlo con la herramienta AI Image Detector. Aunque AI Image Detector proporciona una verificación básica de autenticidad de imágenes, su alcance y precisión son limitados en comparación con las capacidades avanzadas de análisis y presentación que ofrece nuestro sistema.

La aplicación desarrollada en este proyecto incorpora un enfoque detallado y estructurado que abarca desde la gestión y análisis de imágenes hasta la generación de informes. Estos componentes no solo optimizan la detección de deepfakes, sino que también aseguran que los usuarios obtengan información comprensible y detallada para tomar decisiones informadas.

CRITERIO	AI IMAGE DETECTOR	SISTEMA DE DETECCIÓN DE DEEPFAKES
Propósito	Herramienta diseñada para verificar la autenticidad de imágenes generadas por inteligencia artificial.	Aplicación destinada a detectar deepfakes y analizar la autenticidad de imágenes, incluyendo imágenes de personas inexistentes.
Carga de Imágenes	Permite a los usuarios subir imágenes, pero su funcionalidad es básica, sin soporte para múltiples formatos o previsualización.	❖ Permite seleccionar imágenes desde el sistema local. ❖ Soporta múltiples formatos de imagen (jpg, png, jpeg, gif, bmp, tiff, webp). ❖ Previsualización de la imagen antes de la carga. ❖ Notificaciones sobre éxito o error en la carga.
Análisis de Imágenes	Realiza un análisis básico de la autenticidad sin extraer metadatos específicos ni identificar áreas sospechosas.	❖ Extracción de metadatos relevantes. ❖ Cálculo del porcentaje de autenticidad. ❖ Identificación de áreas sospechosas marcadas en la imagen.
Presentación de Resultados	Presenta los resultados de manera simple, pero carece de un formato estructurado.	❖ Resultados presentados en una interfaz estructurada de tres columnas. ❖ Incluye la imagen original, un indicador de autenticidad y metadatos extraídos. ❖ Información sobre áreas sospechosas detectadas.

Generación de Informes	No proporciona funcionalidades para generar informes detallados sobre los resultados del análisis.	❖ Opción de visualizar o descargar el informe para análisis posterior ❖ Generación de informes en PDF. ❖ Incluye diagramas que representan el porcentaje de autenticidad. ❖ Tablas con metadatos extraídos e información sobre áreas sospechosas.
Interacción con el Usuario	Interfaz sencilla y fácil de usar, pero carece de opciones avanzadas de interacción.	❖ Interfaz intuitiva y clara. ❖ Mensajes de confirmación para acciones críticas (carga de imágenes, generación de informes) ❖ Manejo de entradas incorrectas con mensajes de error específicos ❖ Diseño responsivo que permite el uso en dispositivos móviles y de escritorio
Uso Educativo	No está orientado a la educación, enfocado más en la verificación técnica.	Potencial para incluir una sección educativa que explique los resultados y el impacto de los deepfakes, lo que aumenta la confianza del usuario en la herramienta.

Tabla 18. Comparativa de Capacidades: AI Image Detector vs. Sistema detección de deepfakes

2.8.2 Resultados finales

El desarrollo de una aplicación web para la detección de deepfakes ha permitido automatizar el análisis de imágenes, lo que mejora considerablemente el proceso de verificación de autenticidad. Los resultados muestran la precisión y eficacia del sistema en la identificación de contenido manipulado, destacando su potencial para enfrentar los retos técnicos y de seguridad vinculados a la detección de información alterada. Las estrategias de recolección de información se centraron en usuarios interesados en la validación de imágenes. Esto posibilitó la identificación de requisitos tanto funcionales como no funcionales para la aplicación.

Se utilizó una arquitectura cliente-servidor de dos capas, utilizando Python como lenguaje de programación y Flask como framework para el desarrollo del servidor. Las interfaces gráficas fueron implementadas con Angular y UI Bootstrap, garantizando una experiencia de usuario intuitiva. Para el análisis de imágenes y la detección de deepfakes, se emplearon bibliotecas de inteligencia artificial como TensorFlow y Keras. Todo el trabajo se realizó en Visual Studio Code, lo que optimizó el proceso de desarrollo.

La automatización en el proceso de detección ha mejorado de forma notable la autenticación de imágenes en plataformas digitales, reduciendo de forma considerable los tiempos de respuesta en las consultas y aliviando la carga de trabajo de los profesionales dedicados a la verificación digital.

Finalmente, se llevaron a cabo pruebas de funcionalidad para asegurar el correcto funcionamiento de la aplicación. Estas pruebas, fundamentadas en casos de uso previamente establecidos, confirmaron un rendimiento óptimo, reflejando resultados altamente satisfactorios.

CONCLUSIONES

- La recolección para construir un dataset diverso que integra imágenes reales y falsas a través de APIs y redes sociales fue crucial para el desarrollo del proyecto, ya que proporcionó un amplio espectro de ejemplos para entrenar el modelo, lo que es esencial para garantizar su precisión y eficacia en la detección de deepfakes.
- La obtención de información sobre las necesidades y expectativas de los usuarios, así como el análisis de casos de uso y estándares, fueron elementos fundamentales para el éxito del proyecto. Este proceso permitió definir los requisitos funcionales, tales como las funcionalidades específicas de la aplicación, y los requisitos no funcionales, que incluyen la usabilidad, rendimiento y seguridad.
- Se desarrolló una aplicación web destinada a la detección de deepfakes, utilizando técnicas avanzadas de inteligencia artificial basadas en TensorFlow y Keras. Esta aplicación permite la carga de imágenes en varios formatos, como JPG, JPEG, PNG, GIF, BMP y TIFF, y automatiza el proceso de análisis para verificar su autenticidad. La implementación de esta tecnología facilita la identificación de deepfakes de manera precisa y eficiente, optimizando los procesos de seguridad digital.
- La aplicación web siguió una arquitectura cliente-servidor de dos capas. La capa de presentación fue construida con Angular, mientras que el backend fue desarrollado utilizando Flask. El uso de Python en conjunto con entornos como Jupyter Notebook facilitó la integración de algoritmos de aprendizaje profundo para la detección automatizada de deepfakes. Esta estructura modular permitió una interacción fluida entre los componentes, asegurando la escalabilidad y el rendimiento del sistema.
- Tras realizar pruebas de funcionamiento, se validó que la aplicación responde adecuadamente a los requisitos planteados, mostrando un rendimiento en todas sus funcionalidades. Sin embargo, es importante considerar que las pruebas se llevaron a cabo en un entorno controlado. Por lo tanto, es probable que, al implementarse en un entorno real, surjan pequeños inconvenientes o errores que requerirán ajustes menores para garantizar una experiencia óptima.

RECOMENDACIONES

- Se recomienda ampliar el sistema para que, además de imágenes, también pueda analizar videos en la detección de deepfakes. Esto puede lograrse mediante la integración de tecnologías de procesamiento de video y el uso de modelos más avanzados, como redes neuronales recurrentes (RNN), para asegurar mayor precisión.
- Es importante actualizar constantemente los modelos de machine learning utilizados en el sistema, ya que las técnicas de creación de deepfakes evolucionan rápidamente. Se sugiere automatizar este proceso, reentrenando los modelos de forma periódica con nuevas bases de datos que contengan imágenes y videos manipulados.
- Para garantizar el rendimiento del sistema en situaciones de alta demanda, se recomienda migrar a una infraestructura en la nube, como AWS o Google Cloud. Estas plataformas ofrecen escalabilidad automática y balanceo de carga, lo que mejorará el rendimiento y la disponibilidad del sistema sin afectar su estabilidad.
- Se sugiere agregar explicaciones más detalladas en los resultados del análisis para que los usuarios puedan entender mejor los datos generados por el sistema. Además, se recomienda incluir una sección educativa que explique cómo interpretar los resultados, lo que mejorará la confianza del usuario en la herramienta.
- Es esencial seguir realizando pruebas de accesibilidad y compatibilidad en diferentes dispositivos y navegadores para asegurar que la aplicación funcione correctamente en todos los entornos posibles. Esto garantizará una experiencia de usuario óptima, independientemente del dispositivo o navegador utilizado.
- Para aumentar la credibilidad del sistema en entornos profesionales, se recomienda buscar colaboraciones con expertos en seguridad digital. Esto permitirá validar la efectividad del sistema en análisis forenses reales y obtener retroalimentación que ayude a mejorar sus capacidades.

BIBLIOGRAFÍA

[1] Jan Kietzmann, Linda W. Lee, Ian P. McCarthy Tim C. Kietzmann, "Deepfakes: Trick or treat?.Business Horizons 2020. 63, 135e146," Communication Systems(ICESC) (pp. 1110 1113).

[2] F. J. GARCÍA-ULL, Deepfakes: el próximo reto en la detección de noticias falsas, Anàlisi: Quaderns de Comunicació i Cultura, 64, 103-120. DOI, 2021.

[3] C. M. B. Valeria, "Estudio metodológico de las herramientas tecnológicas actuales para detectar y reconocer DeepFakes, abordando la creciente amenaza de la manipulación de contenidos multimedia generados por inteligencia artificial", Sep. 08, 2023. http://repositorio.ucsg.edu.ec/handle/3317/22086.

[4] ThisPersonDoesNotExist, Random AI Generado Fotos de Personas Falsas, 2021. https://this-person-does-not-exist.com/es#:~:text=La%20%EE%80%80gente%EE%80%81%20tiende%20a%20no (accessed Sep. 29, 2024)..

[5] LISA Institute, ""Deepfakes: Tipos, Consejos, Riesgos, Amenazas"".

[6] C. Pazan, "3 183 delitos informáticos se han registrado en el Ecuador, desde el 2020," El Comercio.

[7] R. Limon, "“‘Deepfakes’: la amenaza con millones de visualizaciones que se ceba con las mujeres y desafía la seguridad y la democracia | Tecnología |," El Pais.

[8] M. Lavanda, "Deepfake: Cuando la inteligencia artificial amenaza el Derecho y la Democracia", ResearchGate, Jul. 2022, [Online]. Available: https://www.researchgate.net/publication/368330820_Deepfake_Cuando_la_inteligencia_artificial_amenaza_el_Derecho_y_la_Democracia.

[9] F. De Sistemas, Y. Telecomunicaciones, O. Moreir, and J. Antonio, 55, ""UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA"," Accessed: Aug. 29, 2024. [Online]. Available: https://repositorio.upse.edu.ec/bitstream/46000/8680/1/UPSE-TTI-2022-0038.pdf.

[10] V. D. Gandasegui, "Espectadores de falsos documentales. Los falsos documentales en la sociedad de la información", Dialnet, 2012. https://dialnet.unirioja.es/servlet/articulo?codigo=4153454.

[11] M. d. D. Nacional, "CÓDIGO ORGÁNICO INTEGRAL PENAL, COIP", Available: https://www.defensa.gob.ec/wp-content/uploads/downloads/2021/03/COIP_act_feb-2021.pdf.

[12] R. H. Sampieri, C. F. Collado, and P. B. Lucio, "Metodología de la investigación", Dialnet, 2014. https://dialnet.unirioja.es/servlet/libro?codigo=775008.

[13] Eugenia, "Te explicamos qué es la investigación cuantitativa y cómo usarla en tus proyectos universitarios", Tesis y Másters, Sep. 30, 2022. https://tesisymasters.es/investigacion-cuantitativa/.

[14] G. Mancuzo, ""¨Qué es el modelo incremental,¨Blog - ComparaSoftware.," https://blog.comparasoftware.com/que-es-el-modelo-incremental/#:~:text=Qu%C3%A9%20es%20modelo%20incremental%201%20Para%20qu%C3%A9%20sirve , 2021.

[15] R. J. L. &. P. A. Franks, ""Understanding the Impact of Deepfakes on Privacy and Security."," *Journal of Cybersecurity Research*., 2023.

[16] T. L. S. &. A. T. Karras, " "Analyzing and Improving the Image Quality of StyleGAN."," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[17] Y. K. J. &. W. P. Mirsky, "Deepfake Detection: Advances and Future Directions.," *ACM Computing Surveys*, 2022.

[18] T. Harrison, "Ethical Implications of Deepfake Technology in Modern Media.," **Media Ethics Journal*.*, 2023.

[19] A. J. C. &. W. R. Smith, "Public Perception of Deepfakes: Survey Results and Implications.," *Journal of Digital Media*, 2023.

[20] Jones, D., & Smith, K., "Victimology of Deepfake Crimes: A Comprehensive Review.," **Cybercrime Studies*.*, 2023.

[21] Techopedia, "Entorno de ejecución", Techopedia.com, https://www.techopedia.com/es/definicion/entorno-ejecucion (accedido el 3 de septiembre de 2024)..

[22] "Jupyter Notebook Documentation", "Jupyter Notebook 7.2.1 documentation," Jupyter Notebook Documentation — Jupyter Notebook 7.2.1 documentation. Accedido el 4 de julio de 2024. [En línea]. Disponible: https://jupyter-notebook.readthedocs.io/en/stable/.

[23] Apinemark, "Qué es FRAMEWORK en programación? FUNCION más EJEMPLOS!!", ApInEm Marketing Digital, May 01, 2024. https://www.apinem.com/que-es-framework-en-programacion/.

[24] "Bienvenido a Flask" -, "Documentacion de Flask (3.0.x).," Welcome to Flask — Flask Documentation (3.0.x). Accedido el 4 de julio de 2024. [En línea]. Disponible: https://flask.palletsprojects.com/es/latest/.

[25] "Angular"., "Angular," Accedido el 4 de julio de 2024. [En línea]. Disponible: https://docs.angular.lat/docs.

[26] "Angular directives for Bootstrap", https://angular-ui.github.io/bootstrap/ (accessed Sep. 03, 2024)..

[27] R. Martínez de Madariaga and B. Jorge Juan, "Bibliotecas inteligentes? Comentarios sobre inteligencia artificial aplicada a las bibliotecas,", Enredadera Rev. la Red Bibl. y Arch. del CSIC, no. 39, pp. 91–99, Jun. 2023, doi: 10.20350/DIGITALCSIC/15390..

[28] "TenserFlow vs Kera", "Key Difference Between Them.," Guru99. Accedido el 4 de julio de 2024. [En línea]. Disponible: https://www.guru99.com/tensorflow-vs-keras.html.

[29] Kera |, "TenserFlow Core," TensorFlow. Accedido el 4 de julio de 2024. [En línea]. Disponible: https://www.tensorflow.org/guide/keras?hl=es-419.

[30] Reiner Hernández Ávila, E. Pérez-Perdomo, D. Orozco, & M. Hidalgo, "Deep Learning. Una revisión", ResearchGate, Mar. 19, 2018. https://www.researchgate.net/publication/323858502_Deep_Learning_Una_revision (accessed Oct. 02, 2024).

[31] F. Lubinus Badillo, C. A. Ruefa Hernández, B. MArcoini Narváez, & Y. E. Arias Trillos, "Redes neuronales convolucionales: un modelo de DeepLearning en imágenes diagnósicas. Revisión de tema", Revista colombiana de radiología, vol. 32, no. 3, pp. 5591–5599, Sep. 2021, doi: https://doi.org/10.53903/01212095.161..

[32] P. D. Cumba Armijos and B. A. Barreno Pilco, "Análisis de PYTHON con Django frente a Ruby on Ralls para desarrollo ágil de aplicaciones web. Caso práctico: DECH", [Online]. Available: http://dspace.espoch.edu.ec/handle/123456789/2553.

[33] ¨El tutorial de Python¨, "Python documentation," Accedido el 4 de julio de 2024. [En línea]. Disponible: https://docs.python.org/es/3/tutorial/.

[34] J. Soler-Adillon, "Laboratorio de programación creativa Dosier - Editores de código", Feb. 07, 2017. http://multimedia.uoc.edu/blogs/labpc/es/2017/02/07/dossier-editors-de-codi/.

[35] Microsoft, "Documentation for Visual Studio Code," Visual Studio Code - Code Editing Redefined Accedido el 4 de junio de 2024 Disponible: https://code.visualstudio.com/docs.

[36] "What is Postman?", "Postman API Platform," Postman API Platform. Accedido el 4 de julio de 2024. [En línea]. Disponible: https://www.postman.com/product/what-is-postman/.

[37] A. Barrientos-Báez, M. T. P. Otero, & D. P. Renó, "Imágenes falsas, efectos reales. Deepfakes como manifestaciones de la violencia política de género", 2024. https://www.semanticscholar.org/paper/Im%C3%A1genes-falsas%2C-efectos-reales.-Deepfakes-como-de-Barrientos-B%C3%A1ez-Otero/675ff2ac4f59b039430daf402f2f8fe10f65fc5d#:~:text=Introducci%C3%B3n:%20El%20estudio%20aborda%20la%20problem%C3%A1tica.

[38] Piñeiro-Otero, T. & Martínez-Rolán. X. (2021), Eso no me lo dices en la calle. Análisis del discurso del odio mujeres, en https://doi.org/10.3145/epi.2021.sep.12 Twitter. Profesional de la Información, 30(5).

[39] S. P. Remeseiro, "Inteligencia Artificial: un estudio de su impacto en la sociedad", RUC. Accedido el 19 de septiembre de 2024. [En línea]. Disponible: https://ruc.udc.es/dspace/bitstream/handle/2183/28479/PardinasRemeseiro_Sofia_TFG_2020.pdf?sequence.

[40] A. Montoro-Montarroso, "Fighting disinformation with artificial intelligence: fundamentals, advances and challenges,", Profesional de la información, vol. 32, no. 3, Jun. 2023, doi: https://doi.org/10.3145/epi.2023.may.22..

[41] I. Chile, "Detección avanzada: La función de la IA para reconocer deepfakes - IAB Chile", May 16, 2024. https://www.iab.cl/iab-new/deteccion-avanzada-la-funcion-de-la-ia-para-reconocer-deepfakes/.

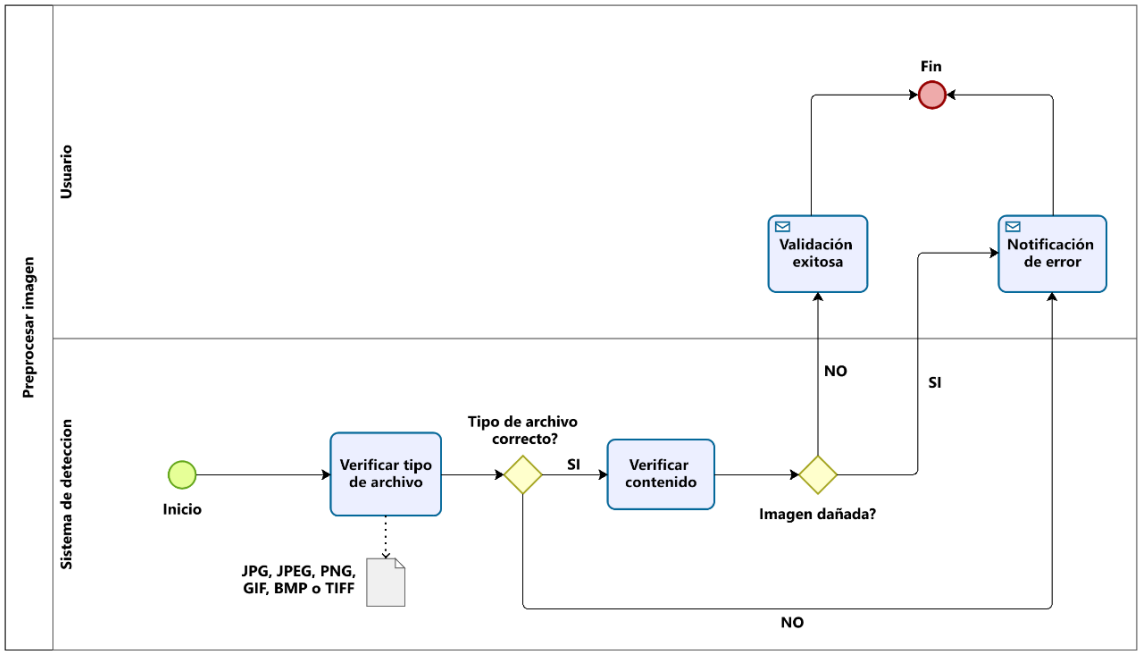
[42] P. A. Fernández, "TRABAJO FIN DE GRADO 'Reconocimiento de DeepFakes.'", Accessed: Sep. 19, 2024. [Online]. Available: https://digibuo.uniovi.es/dspace/bitstream/handle/10651/74464/TFG_PabloArgalleroFernandez.pdf?sequence=5.

[43] A. Schiaffarino, "Modelo cliente-servidor", Infranetworking, Mar. 12, 2019. https://blog.infranetworking.com/modelo-cliente-servidor/.

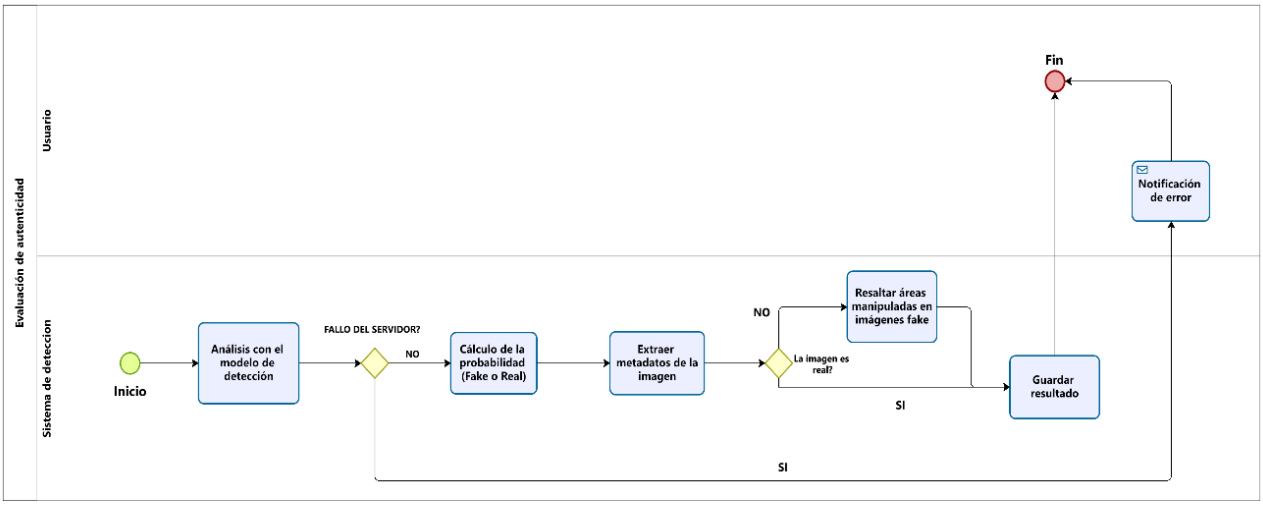
[44] S. Ian, INGENIERÍA DE SOFTWARE, https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf (accessed Oct. 07, 2024).

ANEXOS

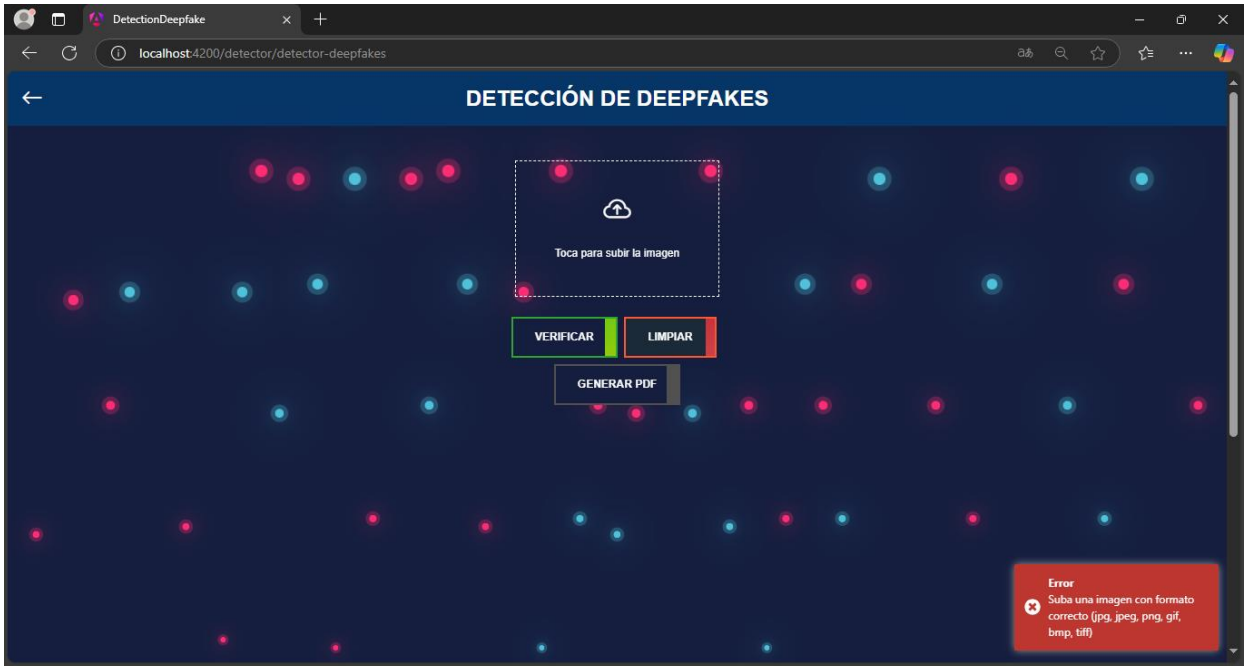
Anexo 1. Diagrama de validación del formato de la imagen



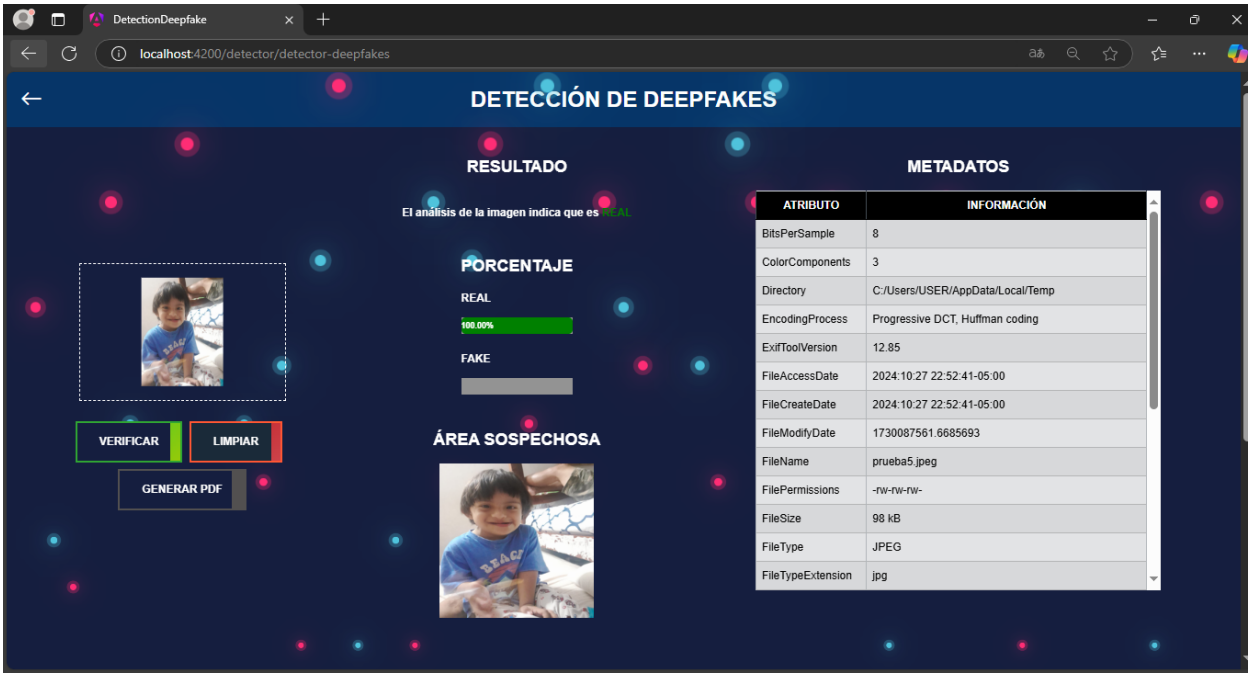
Anexo 2. Diagrama de la evaluación de autenticidad



Anexo 3. Caso de Prueba N° 01 - Verificación de formatos de imagen permitidos

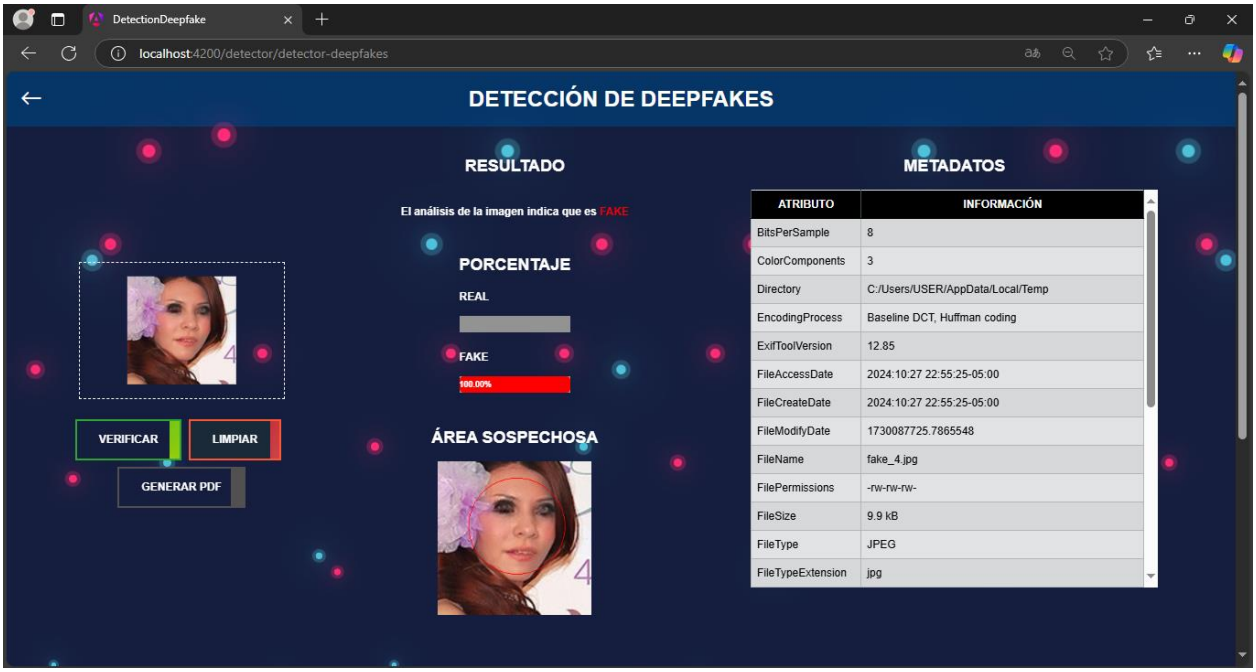


Anexo 4. Caso de Prueba N° 02 - Análisis de autenticidad de una imagen genuina.

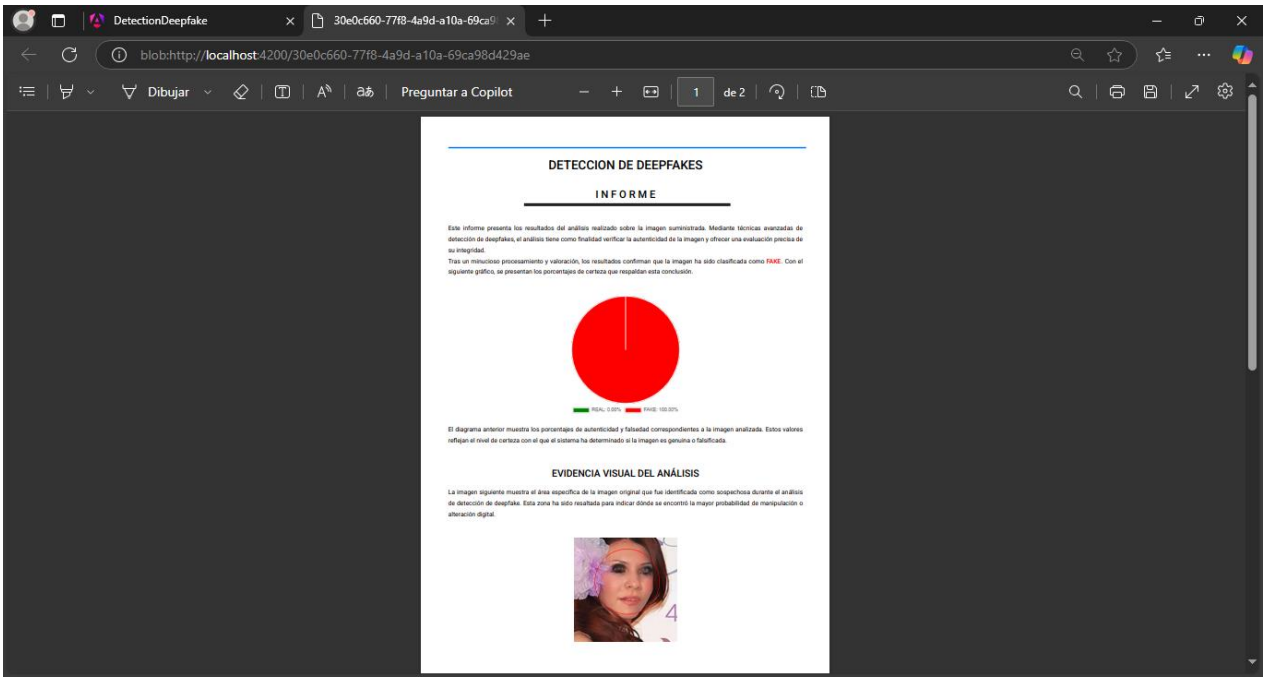


ATRIBUTO	INFORMACIÓN
BitsPerSample	8
ColorComponents	3
Directory	C:/Users/USER/AppData/Local/Temp
EncodingProcess	Progressive DCT, Huffman coding
ExifToolVersion	12.85
FileAccessDate	2024:10:27 22:52:41-05:00
FileCreateDate	2024:10:27 22:52:41-05:00
FileModifyDate	1730087561.6685693
FileName	prueba5.jpeg
FilePermissions	-rw-rw-rw-
FileSize	98 kB
FileType	JPEG
FileTypeExtension	jpg

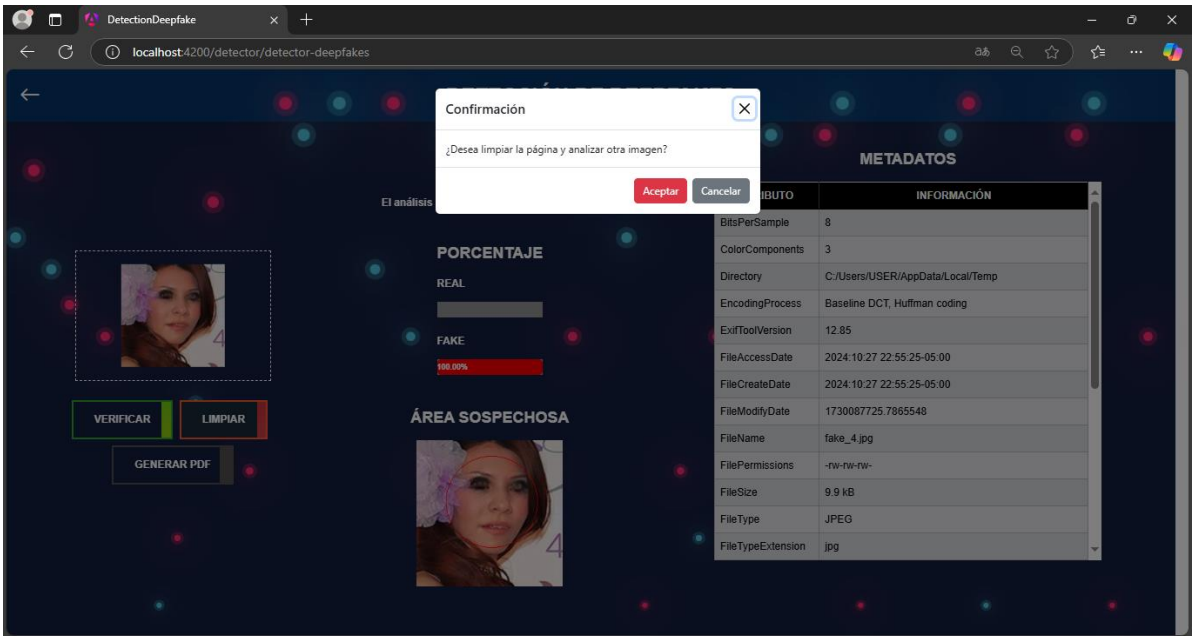
Anexo 5. Caso de Prueba N° 02 - Análisis de autenticidad de una imagen deepfake.



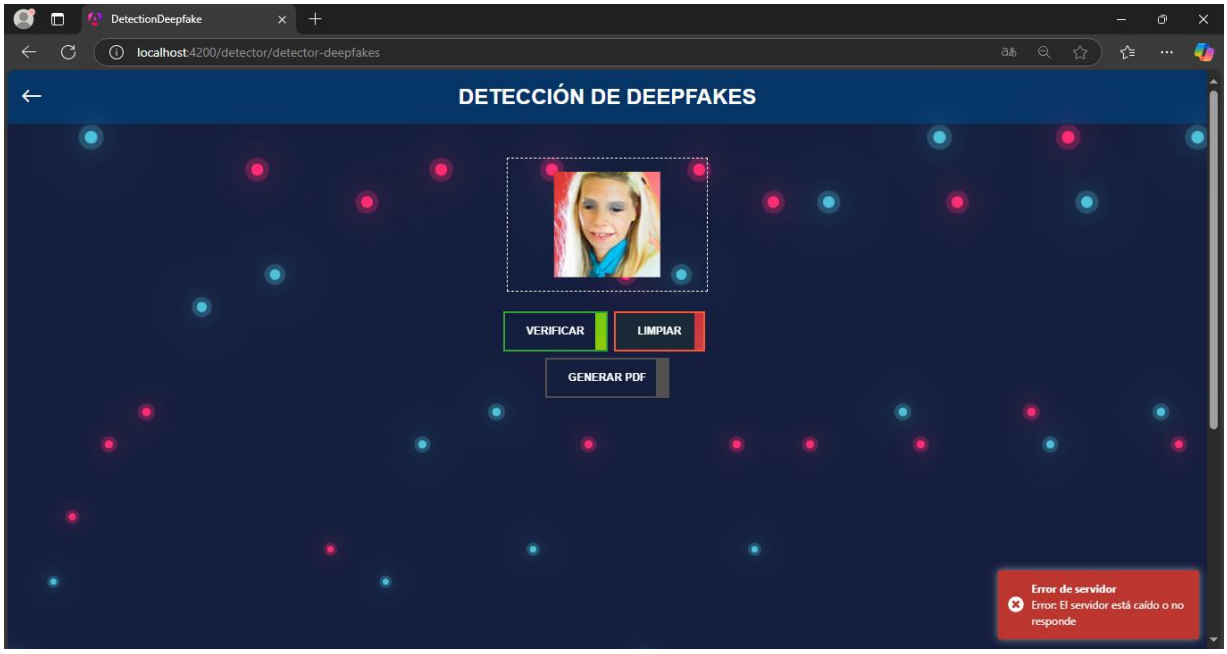
Anexo 6. Caso de Prueba N° 03 - Generación de reporte en PDF.



Anexo 7. Caso de Prueba N° 04 - Restablecimiento de la interfaz de Usuario.



Anexo 8. Caso de Prueba N° 05 - Manejo de errores del servidor



2024

Manual de Usuario

Manual de usuario

ÍNDICE

Responsable de la aplicación

Desarrollador	José Manuel Yagual Castillo
Fecha de creación del documento.	23/ 11/ 2024
Fecha de actualización.	23/ 11/ 2024
Nombre del Sistema.	Aplicación web para la detección de deepfakes

Acerca del manual

Propósito

El presente manual tiene como finalidad ser una guía práctica para el uso de la aplicación web de detección de deepfakes, proporcionando al usuario los conocimientos necesarios para un manejo adecuado del sistema. Además, sirve como herramienta de consulta rápida para resolver dudas sobre su funcionamiento en cualquier momento.

Se ofrece una visión clara de las capacidades y beneficios de la aplicación, que utiliza técnicas avanzadas de machine learning para analizar imágenes y detectar manipulaciones.

Conocimientos necesarios

- Familiaridad básica con el uso de navegadores web y aplicaciones en línea.
- Conocimiento general sobre los deepfakes y los riesgos asociados.
- Capacidad para gestionar archivos, como cargar imágenes en plataformas web.

Introducción

La aplicación web para la detección de *deepfakes* fue diseñada como una herramienta especializada para analizar imágenes y verificar su autenticidad mediante técnicas avanzadas de *machine learning*. A través de un proceso interactivo, la aplicación guía al usuario desde la selección y carga de una imagen hasta la obtención de resultados detallados sobre su posible manipulación.

El sistema valida la imagen, muestra una previsualización y permite iniciar un análisis con un modelo entrenado. Los resultados incluyen el porcentaje de realidad, metadatos y áreas sospechosas. Además, se puede generar un informe PDF con los resultados, útil para investigaciones o reportes. Esta herramienta es ideal para detectar contenido manipulado, ofreciendo una solución eficiente, precisa y accesible para enfrentar los desafíos de las imágenes falsas en entornos digitales.

Visión global

Conceptos generales

Aplicación web

Las aplicaciones web son programas diseñados para ser ejecutados en un navegador, permitiendo a los usuarios acceder a diversas funcionalidades y servicios sin necesidad de instalar software adicional. Estas aplicaciones pueden realizar tareas como la gestión de contenido, análisis de datos o interacción con sistemas de seguridad, entre otras. A diferencia de las aplicaciones móviles, las aplicaciones web están optimizadas para su uso en dispositivos con acceso a Internet, ofreciendo accesibilidad y flexibilidad en diversos entornos.

Deepfakes

Los *deepfakes* son contenido digital manipulado mediante técnicas de inteligencia artificial y *machine learning*, particularmente en imágenes y videos. Estos contenidos falsificados pueden ser extremadamente realistas, lo que representa un riesgo significativo para la seguridad digital y la integridad de la información. Las técnicas utilizadas en los *deepfakes* permiten modificar rostros, voces y otros elementos visuales de manera casi indistinguible de la realidad.

La detección de *deepfakes* utiliza algoritmos avanzados para identificar las manipulaciones en las imágenes, analizando patrones y características que los humanos no pueden percibir a simple vista. Al aplicar este tipo de tecnología, se busca prevenir el fraude digital y proteger la identidad de los usuarios en el entorno digital.

Machine Learning (Aprendizaje Automático)

El *machine learning* es una rama de la inteligencia artificial que se enfoca en el desarrollo de algoritmos que permiten a las máquinas aprender patrones y tomar decisiones basadas en datos. En el contexto de la detección de *deepfakes*, el *machine learning* se utiliza para entrenar modelos que pueden identificar manipulación en imágenes y videos mediante el análisis de características visuales y patrones de datos.

Redes Neuronales Convolucionales (CNN)

Las redes neuronales convolucionales son un tipo de red neuronal utilizada principalmente para el procesamiento de imágenes. Este tipo de red es muy eficaz para detectar patrones visuales y características dentro de imágenes, lo que las hace ideales para aplicaciones de visión por computadora, como la detección de *deepfakes*.

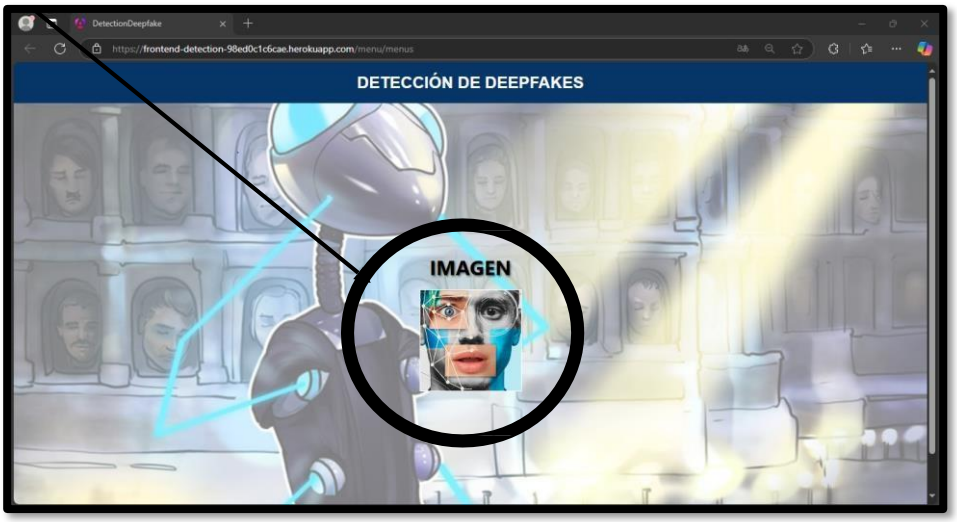
Acceso a la Aplicación Web

Para utilizar la aplicación de detección de *deepfakes*, primero debe accederse a la aplicación a través de un navegador web. Para ello, ingrese el siguiente enlace en la barra de direcciones de su navegador:

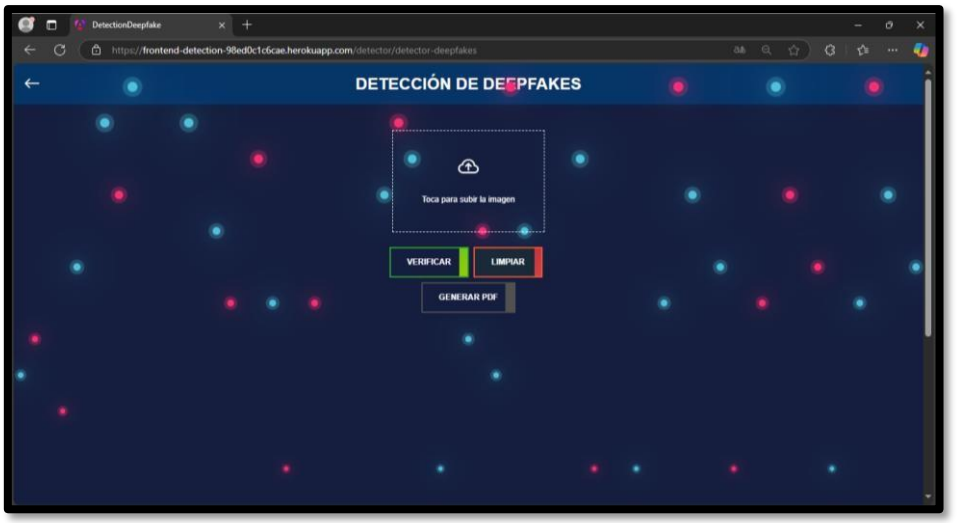
- https://frontend-detection-98ed0c1c6cae.herokuapp.com/

Pantalla Principal

Una vez cargue la página, se mostrará la pantalla principal de la aplicación, similar a la imagen a continuación. En esta interfaz, se encuentra un espacio donde podrá interactuar con la aplicación.



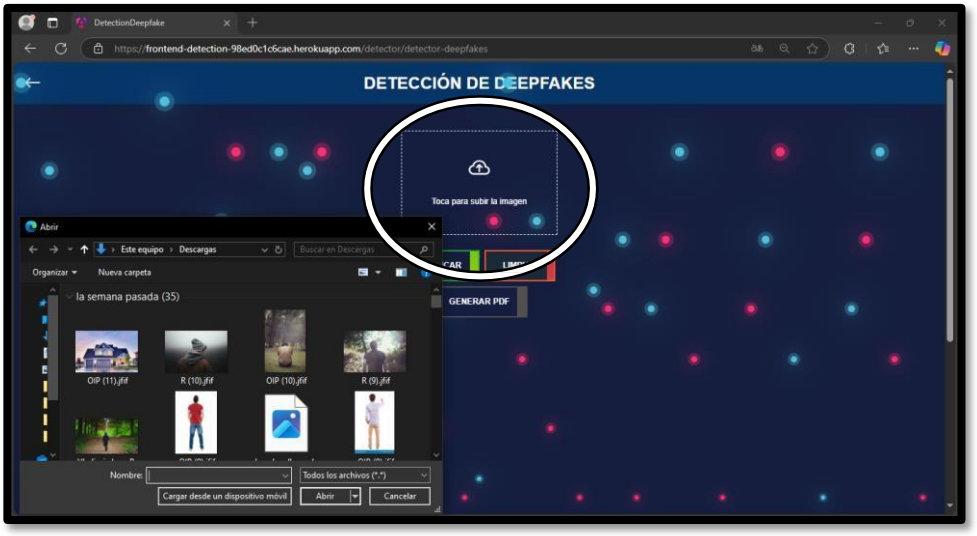
Después de hacer clic en la imagen central de la pantalla principal, se redirigirá automáticamente a la interfaz donde podrás subir una imagen para su análisis. La interfaz es la siguiente:



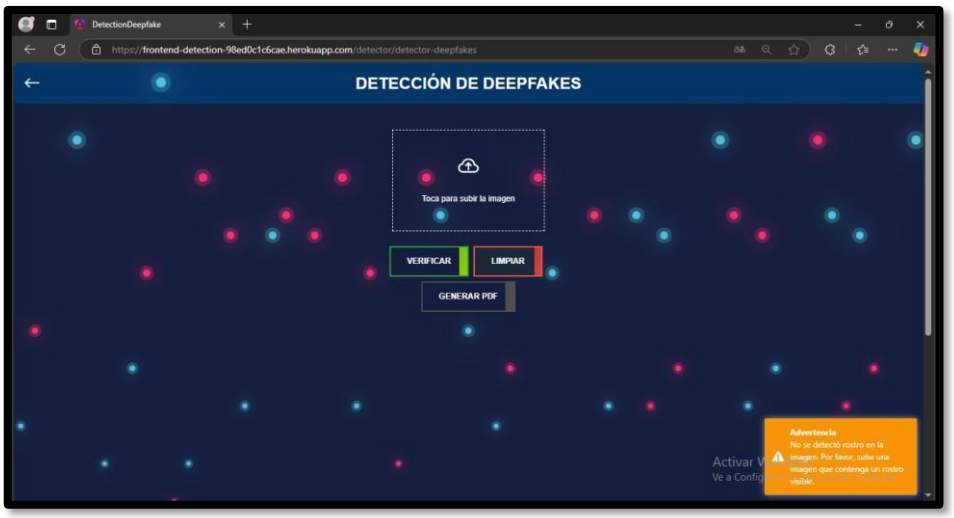
Menú de opciones

Cargar una imagen

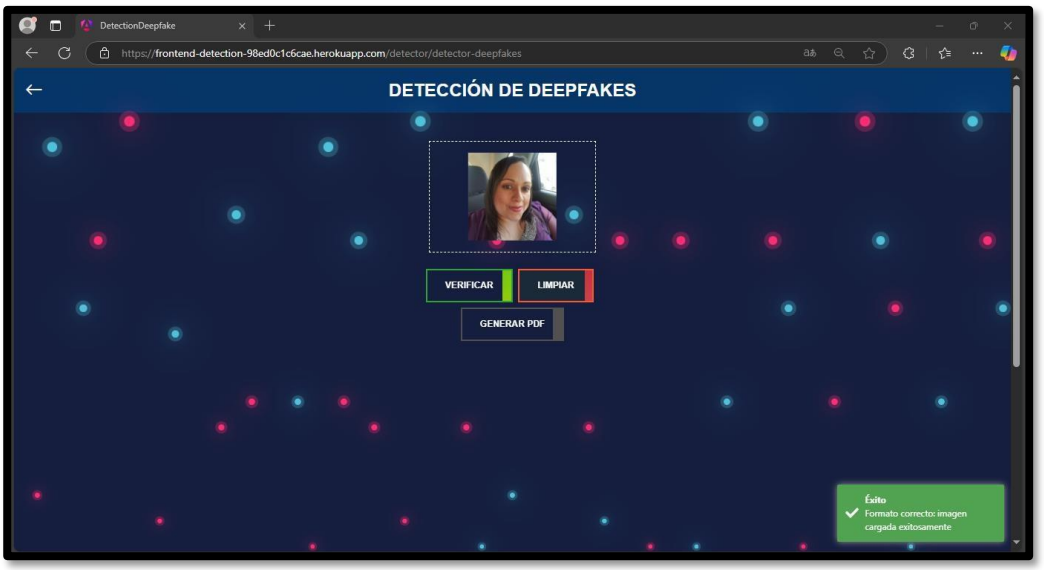
Haciendo clic en el área destacada, que dice "Toca para subir la imagen", podrás seleccionar una imagen desde tu dispositivo. La imagen será procesada para detectar si es un *deepfake*.



Nota: Si la imagen que mandas a verificar no contiene un rostro detectable, el sistema emitirá un mensaje de advertencia indicando que no se ha encontrado un rostro en la imagen. Tras este mensaje, la interfaz se restablecerá automáticamente para permitirte cargar una nueva imagen que contenga un rostro, ya que el análisis requiere la presencia de uno para ser efectivo.

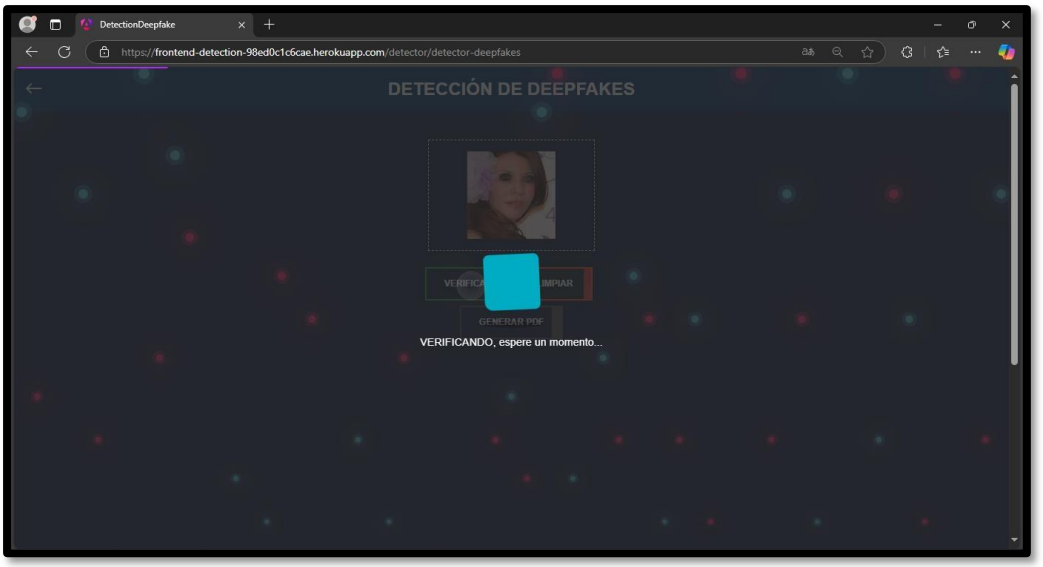


Al momento de cargar la imagen correctamente, podrás visualizarla en el área designada de la interfaz, permitiéndote revisar qué imagen has seleccionado antes de proceder con el análisis.



Iniciar Análisis de Imagen

Después de cargar una imagen, selecciona el botón **VERIFICAR** (con el color verde). Este botón activará el proceso de verificación que escaneará la imagen para detectar deepfakes o si es real.



Resultados del Análisis

Una vez que se haya completado el análisis de la imagen, los resultados se mostrarán en tres columnas:

Imagen Original:

- Se mostrará la imagen que se cargó para su verificación.

Indicador de Autenticidad:

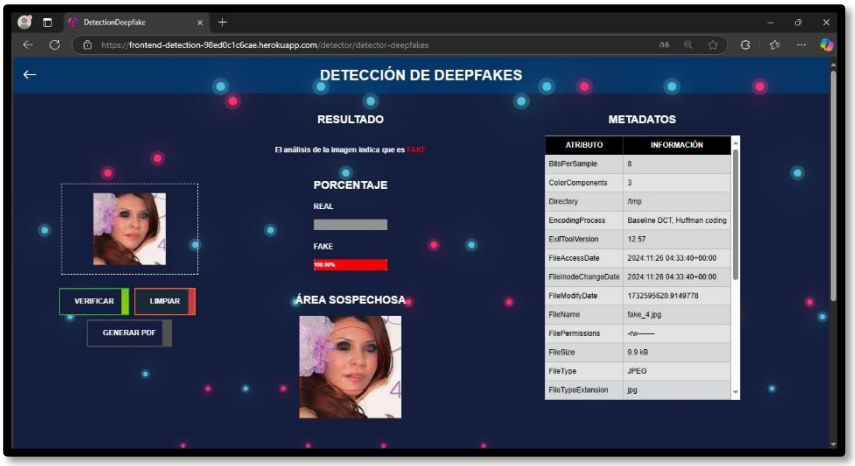
- Se mostrará si la imagen es real o falsa, acompañado de un porcentaje que indica el nivel de certeza.

Áreas Sospechosas:

- Se destacarán las **áreas sospechosas** encontradas en la imagen.

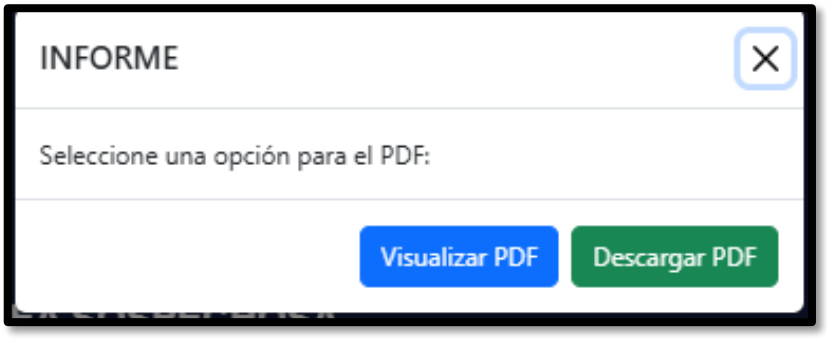
Metadatos de la Imagen:

- Se presentará una tabla con los **metadatos** relacionados con la imagen, como la fecha de creación, resolución y otros detalles técnicos.

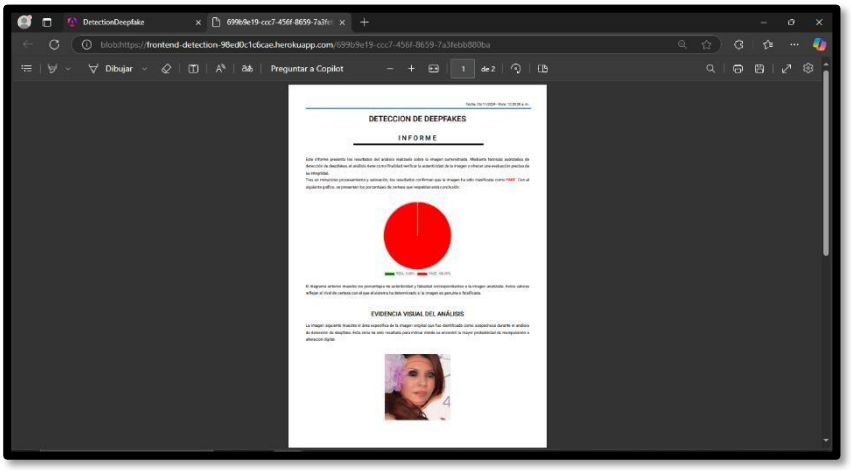


Generar Reporte en PDF

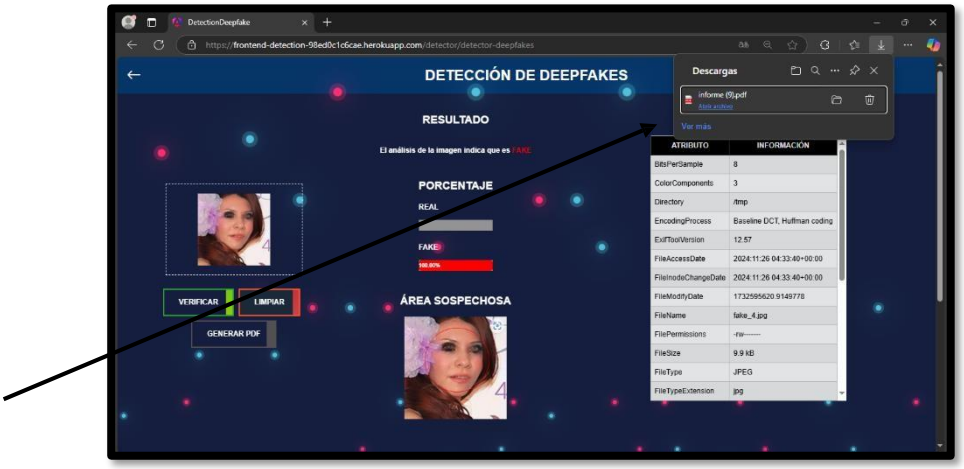
Una vez que se haya completado el análisis, puedes hacer clic en el botón GENERAR REPORTE. el usuario tendrá dos opciones:



- Visualizar Informe: Mostrará una pestaña adicional donde estará el informe detallado en pantalla con los resultados del análisis.



- Descargar Informe: Permite descargar un archivo PDF con los resultados completos, incluyendo un reporte sobre si se detectaron manipulaciones en la imagen.



Borrar imagen y restablecer

Si deseas eliminar la imagen cargada y reiniciar el proceso, haz clic en el botón **LIMPIAR** (con el color rojo).

Antes de proceder, se mostrará un mensaje de confirmación preguntando: **"¿Deseas limpiar la página y analizar otra imagen?"**

Tendrás dos opciones:

- Aceptar: Para confirmar que deseas borrar la imagen y comenzar de nuevo.
- Cancelar: Para mantener la imagen actual y continuar con el análisis.