UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO

FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y MECÁNICA

ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS

TESIS

IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN INTELIGENCIA ARTIFICIAL PARA LA DETECCIÓN DE INTRUSOS CON CÁMARAS DE SEGURIDAD CONVENCIONALES EN CONDOMINIOS

PRESENTADO POR:

Br. PATRICK ANTONIO INQUILTUPA CORTEZ

PARA OPTAR AL TÍTULO PROFESIONAL DE INGENIERO INFORMÁTICO Y DE SISTEMAS

ASESOR:

Dr. RONY VILLAFUERTE SERNA

Cusco - Perú
2024

INFORME DE ORIGINALIDAD

(Aprobado por Resolución Nro.CU-303-2020-UNSAAC)

El que suscribe, **Asesor** del trabajo de investigación titulado: **IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN INTELIGENCIA ARTIFICIAL PARA LA DETECCIÓN DE INTRUSOS CON CAMARAS DE SEGURIDAD CONVENCIONALES EN CONDOMINIOS**, presentado por: ***Br. Patrick Antonio Inquiltupa Cortez*** *con DNI Nº:* ***70576116***, para optar al Título Profesional de Ingeniero Informático y de Sistemas.

Informo que el trabajo de investigación ha sido sometido a revisión por **3ra** vez, mediante el software anti plagio, conforme al Artículo 6° del ***Reglamento para uso de Sistema Anti plagio de la UNSAAC*** y de la evaluación de originalidad se tiene un porcentaje de **1% (uno por ciento)**.

Evaluación y acciones del reporte de coincidencia para trabajos de investigación conducentes a grado académico o título profesional, tesis

PORCENTAJE	EVALUACIÓN Y ACCIONES	MARQUE CON UNA (X)
Del 1 al 10 %	No se considera plagio.	X
Del 11 al 30%	Devolver al usuario para las correcciones.	
Mayores a 31 %	El responsable de la revisión del documento emite un informe al inmediato jerárquico, quien a su vez eleva el informe a la autoridad académica para que tome las acciones correspondientes. Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a ley.	

Por tanto, en mi condición de asesor, firmo el presente informe en señal de conformidad y adjunto la primera hoja del reporte del software anti plagio.

Cusco, 13 de setiembre del 2024

Dr. Rony Villafuerte Serna
DNI: 23957778
ORCID: 0000-0003-4607-522X

Se adjunta:

1. Reporte generado por el sistema anti plagio.
2. Enlace del reporte generado por el sistema anti plagio: **OID: 27259:381098968**

NOMBRE DEL TRABAJO

IMPLEMENTACIÓN DE UN SISTEMA DE RECONOCIMIENTO FACIAL BASADO EN I NTELIGENCIA ARTIFICIAL PARA LA DET E

AUTOR

Patrick Antonio Inquiltupa Cortez

RECUENTO DE PALABRAS

31299 Words

RECUENTO DE CARACTERES

166266 Characters

RECUENTO DE PÁGINAS

108 Pages

TAMAÑO DEL ARCHIVO

16.1MB

FECHA DE ENTREGA

Sep 13, 2024 11:09 AM GMT-5

FECHA DEL INFORME

Sep 13, 2024 11:11 AM GMT-5

1% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 0% Base de datos de Internet
- 0% Base de datos de publicaciones
- Base de datos de Crossref
- Base de datos de contenido publicado de Crossref
- 0% Base de datos de trabajos entregados

Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado
- Coincidencia baja (menos de 20 palabras)
- Bloques de texto excluidos manualmente

Dedicatoria

A mi familia, mis padres y mi hermana por su apoyo incondicional e inmenso amor brindado, el cual me impulsa a seguir adelante y lograr todas las metas y logros propuestos.

Agradecimientos

Quiero expresar mi total agradecimiento a las personas las cuales me apoyaron e hicieron que se realice el presente proyecto.

Quiero expresar mi sincero agradecimiento a mi asesor por su orientación experta, paciencia y apoyo constante a lo largo de este proyecto. Su asesoramiento ha sido fundamental para el desarrollo y finalización de este proyecto.

A mis amistades, quienes estuvieron ahí para apoyarme y dándome ánimos en todo este proceso largo.

Y a todas las personas que formaron parte de este proceso, agradecer por su contribución el cual ha enrriquesido a este proyecto.

Resumen

El reconocimiento facial es una tecnología emergente que, además de su uso en la identificación de personas, puede extenderse al reconocimiento de animales y objetos. Este avance ha sido facilitado por el progreso continuo en el campo de la Inteligencia Artificial. Sin embargo, la implementación generalizada de estas tecnologías sigue siendo limitada por el alto costo de las cámaras de seguridad de última generación.

La seguridad ciudadana ha sido una preocupación persistente, y el reconocimiento facial ofrece una vía para mejorar tanto la seguridad física como emocional de los individuos. El presente proyecto de investigación busca superar las limitaciones actuales desarrollando e implementando un sistema interno capaz de integrarse con una amplia gama de cámaras que una persona pueda tener, incluyendo cámaras integradas en laptops, webcams, dispositivos móviles, videocámaras, cámaras de seguridad IP y cámaras de seguridad WiFi. El propósito es facilitar el reconocimiento facial en diversos dispositivos, haciéndolo más accesible.

El estudio también incluye una evaluación comparativa de varios algoritmos para la detección y el reconocimiento facial. Los resultados indicaron que el algoritmo Haar Cascades es altamente efectivo para la detección de rostros, mientras que el algoritmo LBPH Faces demostró una gran precisión en el reconocimiento facial. Ambos algoritmos mostraron un rendimiento sobresaliente en pruebas de transmisiones en vivo, destacando su eficacia y eficiencia.

Palabras clave: Reconocimiento Facial, Aprendizaje automático, Seguridad Ciudadana, Algoritmo Haar Cascades, Algoritmo LBPH Faces.

Abstract

Facial recognition is an emerging technology that, in addition to its use in the identification of people, can be extended to the recognition of animals and objects. This advance has been facilitated by continued progress in the field of Artificial Intelligence. However, the widespread implementation of these technologies remains limited by the high cost of state-of-the-art security cameras.

Citizen safety has been a persistent concern, and facial recognition offers an avenue to improve both the physical and emotional safety of individuals. This research project seeks to overcome the current limitations by developing and implementing an in-house system capable of integrating with a wide range of cameras that an individual may have, including cameras embedded in laptops, webcams, mobile devices, video cameras, IP security cameras and WiFi security cameras. The purpose is to facilitate facial recognition on a variety of devices, making it more accessible.

The study also includes a comparative evaluation of various algorithms for face detection and recognition. The results indicated that the Haar Cascades algorithm is highly effective for face detection, while the LBPH Faces algorithm demonstrated high accuracy in face recognition. Both algorithms showed outstanding performance in live streaming tests, highlighting their effectiveness and efficiency.

Keywords: Facial Recognition, Face Recognition, Machine Learning, Citizen Security, Haar Cascades Algorithm, LBPH Faces Algorithm.

Introducción

En la actualidad, el reconocimiento facial ha emergido como una herramienta indispensable en el ámbito de la seguridad, particularmente en la gestión de accesos a condominios y edificios residenciales. Esta tecnología permite un control más riguroso sobre quiénes ingresan a estas instalaciones, ofreciendo una capa adicional de protección para los residentes. Sin embargo, muchas de las cámaras de vigilancia disponibles en el mercado no cuentan con capacidades avanzadas de reconocimiento facial, lo que limita su eficacia en entornos donde la seguridad es una prioridad. Este proyecto de investigación surge como respuesta a estas limitaciones, con el objetivo de diseñar e implementar un sistema que integre reconocimiento facial de manera eficiente, abordando las carencias actuales. La presente documentación se estructura en cuatro capítulos que guiarán al lector a través de las distintas fases del proyecto.

En el primer capítulo, se realizará un análisis detallado del problema que se busca resolver, así como la formulación de los objetivos que orientan el proyecto. También se incluirá una justificación de la elección de la problemática, destacando su relevancia en el contexto actual de seguridad en espacios residenciales. Además, se definirán las limitaciones del estudio y el alcance que se espera lograr con la implementación del sistema propuesto.

El segundo capítulo estará dedicado a la revisión de antecedentes, tanto a nivel nacional como internacional, que han sido fundamentales para el desarrollo de este proyecto. Se abordarán conceptos clave relacionados con el reconocimiento facial y la tecnología de cámaras de seguridad, proporcionando una base teórica sólida. Asimismo, se explicarán los distintos modelos de detección de rostros, empleando librerías reconocidas como Mediapipe y OpenCV, y se explorarán los algoritmos de reconocimiento facial más relevantes, como Eigen Faces, Fisher Faces y LBPH Faces. Este capítulo también incluirá una descripción de las herramientas y tecnologías que se utilizarán en el desarrollo del proyecto.

El tercer capítulo se centrará en el desarrollo y la implementación del proyecto, dividido en cuatro secciones para una mejor comprensión. En la primera sección, se abordará la creación de un sistema interno para gestionar la información de los residentes de un condominio o edificio residencial. Este sistema se construirá utilizando un lenguaje de programación y un framework adecuado, junto con una base de datos que soporte las funcionalidades requeridas. Se desarrollarán interfaces de usuario y mecanismos de autenticación para asegurar la integridad del sistema. En la segunda sección, se presentará el desarrollo de tres modelos de detección de rostros, que serán implementados y evaluados para determinar cuál ofrece mejores resultados y eficiencia en el contexto del sistema propuesto. En la tercera sección, se procederá con la implementación de modelos de reconocimiento facial, siguiendo un enfoque similar al de la detección de rostros. Finalmente, en la cuarta sección, se describirá la integración de estos modelos en el sistema y se realizarán pruebas exhaustivas utilizando diversas cámaras conectadas a través de diferentes métodos, incluyendo cables, aplicaciones

móviles y el protocolo RTSP.

El cuarto capítulo estará dedicado a la evaluación de los resultados obtenidos durante el desarrollo del proyecto. Se realizará un análisis crítico de los datos recopilados, comparando los resultados con los objetivos inicialmente planteados y los antecedentes revisados. Este análisis permitirá identificar las fortalezas y debilidades del sistema implementado, y proporcionará una base para las conclusiones y recomendaciones finales del proyecto. Se presentarán sugerencias para mejorar el sistema en futuras investigaciones y se destacarán las contribuciones del estudio al campo de la seguridad mediante el uso de reconocimiento facial.

Listado de Abreviaturas

API Interfaz de Programación de Aplicaciones.

BD Base de Datos.

IP Protocolo de Internet.

ML Aprendizaje Automático.

DL Aprendizaje Profundo.

DNN Redes Neuronales Profundas.

CNN Redes Neuronales Convolucionales.

RNN Redes Neuronales Recurrentes.

GAN Redes Generativas Antagónicas.

SCRUM Gestión de Proyectos de Metología Ágil.

OpenCV Open Computer Visión Library.

TIR Tasa Interna de Retorno.

IA Inteligencia Artificial.

PCA Análisis de Componentes Principales.

LBPH Histogramas de Patrones Binarios Locales.

CCTV Circuito Cerrado de Televisión.

BCL Compensación de contraluz.

WDR Amplio rango dinámico.

KLT Kanade-Lucas-Tomas.

RTSP Protocolo de transmisión en tiempo real.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Aspectos Generales

1.1. Planteamiento del Problema

1.1.1. Descripción del problema

En el siglo XXI, la seguridad en los hogares ha disminuido considerablemente, como lo muestran los reportes en diversos medios de comunicación. Este incremento en la inseguridad afecta tanto la integridad física como emocional de las familias en nuestro país, generando un temor constante en los hogares peruanos. Las familias se preocupan no solo por la protección de sus pertenencias, sino también por su bienestar físico, que con frecuencia es amenazado o vulnerado por individuos dedicados al robo. Estos delincuentes no dudan en invadir propiedades privadas con el fin de apropiarse de objetos de valor, poniendo en riesgo la tranquilidad y seguridad de quienes habitan estos espacios.

La motivación detrás de estos actos delictivos es principalmente económica, una situación que se ha intensificado debido a la crisis socioeconómica generada por la pandemia de COVID-19. Esta situación ha propiciado un aumento en la frecuencia y variedad de modalidades de robo, muchas de las cuales ponen en grave peligro la vida de las personas. A esto se suma la falta de seguridad ciudadana por parte del Estado, y la inadecuada instalación de cámaras de vigilancia en muchas zonas del país. Como resultado, la actividad delictiva se ha convertido en un modo de vida para algunos, quienes no solo atentan contra la seguridad en espacios públicos, sino también en propiedades privadas como viviendas, condominios y establecimientos comerciales.

Los condominios, en particular, representan un blanco vulnerable para estos delincuentes. Aunque suelen contar con guardias de seguridad y cámaras de vigilancia, estos mecanismos no siempre son suficientes para prevenir o alertar sobre incidentes delictivos. Las cámaras de seguridad en estos lugares generalmente solo graban y almacenan video durante un tiempo limitado, lo que puede no ser efectivo en caso de un robo. Esta vulnerabilidad es aprovechada por los delincuentes, quienes encuentran diversas formas de infiltrarse en los condominios, ya sea haciéndose pasar por residentes o aprovechando distracciones de los guardias de seguridad, entre otras tácticas. Esta situación pone en riesgo tanto las propiedades como la seguridad de los residentes.

En el mercado, existe una amplia gama de cámaras de seguridad que varían en función de sus características y capacidades adicionales. Los usuarios eligen entre estos tipos de cámaras de acuerdo a sus necesidades específicas, considerando factores como la resolución del video, la capacidad de almacenamiento, la conectividad a redes, la detección de movimiento, o incluso el reconocimiento facial. Sin embargo, a medida que se incrementan las funcionalidades de una cámara, su costo también tiende a aumentar proporcionalmente, lo que limita el acceso de muchas personas a soluciones de seguridad más avanzadas.

1.1.2. Identificación del problema

En condominios y edificios, es común el uso de cámaras de vigilancia como una medida de seguridad. Sin embargo, estas cámaras no suelen ser las más avanzadas debido a su alto costo y a los gastos asociados con su mantenimiento continuo. En su lugar, se opta por cámaras de seguridad convencionales, cuya función principal es grabar y almacenar imágenes o videos, sin incorporar tecnología avanzada. Esta situación genera inseguridad entre los residentes, quienes no se sienten protegidos adecuadamente por estos dispositivos.

El problema radica en que el uso de cámaras de seguridad convencionales no garantiza la detección de intrusos ni ofrece una alerta temprana ante posibles amenazas. La falta de inteligencia artificial o recursos tecnológicos avanzados en estas cámaras limita su capacidad para identificar y prevenir incidentes de seguridad, dejando a los residentes expuestos a vulnerabilidades que no logran salvaguardar su bienestar personal y su propiedad.

1.2. Formulación del Problema

1.2.1. Problema General

La falta de software avanzado en cámaras de seguridad convencionales limita significativamente su capacidad para realizar reconocimiento facial y detectar intrusos, reduciendo la eficacia de la seguridad en condominios y edificios residenciales.

1.2.2. Problemas Específicos

- Ausencia de un sistema interno semi automático encargado del control de acceso en condominios y edificios residenciales.
- Dificultad en la selección del modelo más preciso y eficiente para la detección de rostros y reconocimiento facial.
- Ausencia de un sistema que permita la integración efectiva de diferentes cámaras convencionales para mejorar o agregar funciones modernas.

1.3. Objetivos

1.3.1. Objetivo General

Implementar un sistema de reconocimiento facial basado en inteligencia artificial para la detección de intrusos con cámaras convencionales en condominios y/o edificios de uso residencial.

1.3.2. Objetivos Específicos

- Implementar un sistema interno semi automático para controlar el ingreso de personas.
- Analizar, comparar y seleccionar el modelo de detección de rostros y reconocimiento facial más óptimo.
- Integrar las cámaras convencionales al sistema interno, para la detección de intrusos.

1.4. Justificación

1.4.1. Conveniencia

En la actualidad, muchos condominios y edificios residenciales carecen de un control eficiente sobre el acceso de personas, lo que facilita la entrada de individuos no autorizados o desconocidos (intrusos). Esta falta de seguridad expone las propiedades a riesgos de robos y otros incidentes. La mayoría de estas propiedades están equipadas únicamente con cámaras convencionales que no contribuyen significativamente a la protección de los residentes ni previenen o alertan sobre posibles amenazas.

Por lo tanto, es crucial desarrollar e implementar un sistema interno semi automático que compense esta falta de funcionalidades avanzadas en las cámaras convencionales. Esto garantizaría una mayor seguridad para las propiedades y sus habitantes. En este contexto, la integración de la inteligencia artificial (IA) se vuelve esencial, abordando aspectos como el aprendizaje supervisado y no supervisado para complementar y mejorar las capacidades de las cámaras convencionales.

1.4.2. Relevancia

El proyecto de investigación busca reforzar la seguridad en condominios y edificios residenciales mediante un control más efectivo del acceso de personas y la detección temprana de intrusos. Esto se propone como una medida frente a la creciente inseguridad ciudadana, un problema persistente en la sociedad actual. Además, el proyecto pretende aprovechar y modernizar algunas de las cámaras convencionales existentes, que se están quedando obsoletas ante el rápido avance tecnológico.

1.4.3. Implicancias Prácticas

- Se desarrollará un sistema semi atomático que se pueda integrar con cámaras convencionales, permitiendo su reutilización y actualización en lugar de reemplazarlas. Este sistema proporcionará funcionalidades adicionales que mejorarán la seguridad residencial.
- Se logrará un control más efectivo del acceso a las residencias, facilitando la realización de nuevos análisis y predicciones sobre el comportamiento de las personas.
- Al convertir el sistema interno semi atomático en una plataforma web, este podrá ser utilizado por múltiples condominios y edificios residenciales. Esto permitirá la conexión entre diferentes propiedades, compartiendo información sobre posibles intrusos y mejorando la vigilancia colaborativa para prevenir incidentes.

1.4.4. Valor Teórico

Este proyecto de investigación contribuirá al avance tecnológico al explorar y analizar diversas técnicas de reconocimiento facial disponibles en la actualidad. Además, abordará la adaptación de estas tecnologías a cámaras convencionales, que tienden a quedar obsoletas debido al constante desarrollo de nuevas tecnologías, prolongando así su vida útil y mejorando su funcionalidad.

1.4.5. Utilidad Metodológica

El sistema desarrollado en este proyecto de investigación podrá ser aprovechado en futuros trabajos, permitiendo la incorporación de más funcionalidades a cámaras convencionales. Entre estas funcionalidades se incluyen: el análisis y optimización de la calidad de imagen, el reconocimiento de placas vehiculares y la aplicación de técnicas para predecir comportamientos basados en los datos de acceso de los residentes. Además, la técnica de reconocimiento facial implementada servirá como fundamento para futuros estudios enfocados en la detección e identificación de personas, facilitando la integración de nuevas y avanzadas técnicas de inteligencia artificial.

1.5. Delimitación de estudio

1.5.1. Delimitación Espacial

Esta investigación se centra en la implementación de un sistema interno semi-automático, especialmente diseñado para su aplicación en condominios y edificios residenciales. El objetivo principal del sistema es reforzar la seguridad y optimizar el control de acceso en estos espacios, priorizando la protección y bienestar de los residentes. En este contexto, no se requiere una carta de consentimiento, ya que las pruebas se llevarán a cabo en un edificio que forma parte de una residencia familiar, lo que permite un entorno controlado y adecuado para la experimentación.

1.5.2. Delimitación Temporal

El desarrollo de este proyecto de investigación comenzó en noviembre de 2022, con la fase inicial dedicada a la recopilación de datos y al estudio del marco teórico, enfocándose en el tema central de la inteligencia artificial. Este proceso se extendió hasta enero de 2024, momento en el que se completó el desarrollo del sistema complementario para las cámaras convencionales.

1.6. Método

1.6.1. Alcance

El proyecto de investigación tendrá un alcance descriptivo, ya que se enfocará en detallar el funcionamiento del reconocimiento facial en un entorno controlado, como un condominio o edificio residencial. (Fernández Collado and Baptista Lucio, 2014)

Además, se llevará a cabo una descripción y análisis de diversos algoritmos de detección facial utilizando librerías como MediaPipe y OpenCV. Se examinarán algoritmos específicos para el reconocimiento facial, incluyendo Eigen Faces, Fisher Faces y LBPH Faces. El objetivo es proporcionar una comprensión exhaustiva de estos algoritmos, evaluando sus características y desempeño para seleccionar el más adecuado según las necesidades del proyecto.

1.6.2. Diseño

El reconocimiento facial, al estar inmerso en el campo de la inteligencia artificial (IA), presenta una característica distintiva: es un área que sigue evolucionando rápidamente. A pesar de los avances, el campo de la IA sigue siendo amplio y su conocimiento es relativo más que absoluto, debido al constante progreso y nuevas investigaciones. Este proyecto de investigación se centra en describir y analizar de manera detallada el funcionamiento del sistema de reconocimiento facial basado en IA, con el objetivo de evaluar su eficiencia y efectividad en un entorno específico, como condominios o edificios residenciales. Para lograr esto, se utilizará software especializado que permitirá identificar claramente a las personas que pertenecen o no a la propiedad, asegurando así la protección de la propiedad privada.

Además, el proyecto no se basa en una hipótesis específica ni en la recolección de datos numéricos o mediciones de variables. En su lugar, adopta una metodología descriptiva que se ajusta a la naturaleza del estudio y al enfoque del proyecto. Esto implica una descripción detallada del sistema y sus algoritmos sin la necesidad de datos cuantitativos.

Asimismo, el proyecto tendrá un enfoque experimental, ya que se desarrollará e implementará un sistema interno junto con algoritmos de IA. Estos serán probados y analizados para evaluar su rendimiento y efectividad, como se detallará en las siguientes secciones del proyecto. (Fernández Collado and Baptista Lucio, 2014)

1.6.3. Para el desarrollo de la parte informática

Se optará por la metodología SCRUM para el desarrollo del sistema a implementar, dado que esta metodología proporciona una gran flexibilidad y capacidad de adaptación a los requisitos que puedan surgir durante el proyecto. SCRUM facilita la entrega continua de avances mediante sprints, lo que permitirá ajustar y mejorar progresivamente el sistema vinculado a cámaras de seguridad convencionales, hasta alcanzar el objetivo final. (Atlassian, 2023)

En cuanto a la implementación de los modelos y algoritmos de inteligencia artificial para la detección y reconocimiento facial, se adoptará un enfoque iterativo e incremental. Este enfoque permitirá repetir y perfeccionar el proceso de desarrollo, desde la captura de imágenes y entrenamiento de modelos hasta la detección de rostros e identificación de intrusos. La metodología iterativa garantizará que cada ciclo de desarrollo acerque al sistema a una eficiencia y precisión óptimas. (nimble Humanize Work, 2023)

Capítulo 2

Marco Teórico

2.1. Antecedentes

2.1.1. Antecedentes Internacionales

Heredia Salazar C, Rea Rodriguez D, (2022), *"Diseño de un sistema de detección facial utilizando cámaras IP para el reconocimiento de individuos en la cercanía de residencias familiares"*, Uniersidad Politécnica Salesiana Sede Quito, Ecuador.

Conclusiones:

- Tras la experimentación con una Raspberry pi 4, se concluye que el procesamiento del algoritmo con la cámara IP se vuelve demasiado alto ya que llega a ocupar todos los recursos del dispositivo Raspberry, por lo tanto, para usos experimentales de casos de estudios no se recomienda, ya que se necesita de mayor procesamiento y nivel de respuesta para el reconocimiento en tiempo real.
- Se analizó que ha mayor cantidad de fotos registradas almacenadas, aumenta la lentitud del entrenamiento de rostros, por ello se convierte en un video entrecortado por lo cual se comprobó, que, al utilizar la propia cámara de la Raspberry Pi, obtuvo un mayor rendimiento y fluidez debido a que su conexión es directa a este dispositivo y su calidad soporta al procesamiento.
- Para un mayor nivel de precisión al momento del reconocimiento facial se debe tomar en cuenta que los rostros deben de alinearse de manera frontal, así se le proporcionará al algoritmo mayor exactitud en el momento de comparar los rasgos faciales.
- Al analizar las pruebas de luminosidad se determinó que el sistema funciona de tal manera: Con la luz natural, ya que obtuvo el 93 % de reconocimiento, al contrario de la luz nocturna que, debido al brillo que provoca la luz infrarroja de la cámara, así se obtuvo un descenso de reconocimiento del 10 % obteniendo el 83 % (determinado experimentalmente).
- En las pruebas de distancias del rostro, se comprobó que la resolución y calidad de la cámara es un aspecto importante para la detección, ya que a mayor distancia (2m)

menor fue el reconocimiento, y a menor distancia (1m) mayor el reconocimiento, teniendo en cuenta que con 1 metro de distancia la efectividad de reconocimiento fue del 93 % (determinado experimentalmente), así se comprobó que en esta distancia el sistema detecta mejor los puntos faciales para el reconocimiento.

Comentario: La inclusión de este antecedente, se debe a la metodología SCRUM que escogió para el desarrollo de su proyecto, así como el uso de las librerías de python como es OpenCV para realizar la detección de rostros, y por los modelos escogidos para realizar el reconocimiento facial que se ha usado para el desarrolo de dicho proyecto.

Vásquez Bohórquez D, Cortes Martínez C, (2021), *"Sistema de detección, extracción y reconocimiento de rostros en excenas de máximo 4 personas, para aplciaciones de videovigilancia residencial utilizando herramientas de software libre, en lugares cerrados"*, Universidad Distrital Francisco José de Caldas, Colombia.

Conclusiones:

- Por medio de técnicas de procesamiento digital de imágenes se logró realizar un sistema de videovigilancia residencial, con algoritmos de detección, extracción e identificación de rostros, para identificar las personas que ingresan a una zona residencial bajo condiciones específicas.
- La resolución de la cámara del sistema de video vigilancia es un factor importante para seleccionar, pues en caso de que la resolución sea demasiado baja, la distancia de detección es demasiado corta y si se desea detectar y reconocer 3 o 4 personas la resolución sobre los rostros no será la mínima requerida, por lo tanto, no detectará los rostros.
- Al realizar la validación final del sistema en un ambiente no residencial, como lo es un centro comercial, se evidencia que la iluminación presente en el recinto es uno de los factores más importantes a la hora de implementar estas técnicas de reconocimiento. La librería de OpenCV se ve muy afectada, a tal punto de no reconocer ninguno de los rostros presentes ya que la cámara estaba a contraluz. Por otra parte, la librería de Face-Recognition bajó su distancia máxima de detección a casi la mitad.
- Mediante la investigación realizada se pudo determinar cuáles son los algoritmos desarrollados en Python para aplicaciones relacionadas al reconocimiento de rostros, encontrando que los algoritmos más usados son los de las librerías Face-Recognition y OpenCV debido a su fácil implementación y alta efectividad en condiciones controladas.

Comentario: La inclusión de este antecedente, se debe a su desarrollo así como a las herramientas las cuales fueron usadas para desarrollar dicho proyecto de investigación, todos estos puntos como las herramientas, el lenguaje de programación y los códigos libre que ha usado en el desarrollo de dicho proyecto, me servirán de base para el presente proyecto de investigación que se está realizando.

Briones Gárate E, (2020), *"Sistema web de reconocimiento facial para control de acceso biométrico, utilizando inteligencia artificial"*, Escuela Superior Politécnica del Litoral, Colombia.

Conclusiones:

- Con el estudio realizado se puede determinar que no existe una técnica específica de reconocimiento facial que cumpla con todas las expectativas del caso, adicional se pudo conocer que ayuda en la lucha contra el crimen, pues en la investigación se puede ver que el 90 % de los encuestados afirman lo mencionado de manera que la lucha contra el 37 terrorísmo es un abrir y cerrar de ojos el reconocimiento facial cambia la idea de privacidad de las personas.
- El reconocimiento facial es un sistema computarizado que identifica automáticamente a una persona sobre la base de una imagen digital o una fuente de video que se contienen en una base de datos almacenada.
- La tecnología de reconocimiento facial llegó para quedarse. Si bien trae muchos beneficios, hay cuestiones evidentes que se deben resolver.

Comentario: La inclusión de este antecedente, nos servirá como ayuda para el desarrollo del proyecto en la sección del marco teórico, para recopilar información sobre la inteligencia artificial que usa para poder realizar el reconocimiento facial, así como la forma en que ha realizado el desarrollo de su proyecto de investigación..

2.1.2. Antecedentes Nacionales

Verdeguer Valderrama D, (2022), *"Diseño e implementación de un sistema de identificación de personas para la seguridad de los accesos a condominos, basado en el algoritmo de reconocimineto facial LBPH FACES"*, Universidad Privada del Norte, Perú.

Conclusiones:

- El sistema proporciona una mejor seguridad para el condominio san Antonio de Carabayllo, ya que el método LPBHFaces identifica mejor los rostros y para una mejor precisión, se debe de tomar la captura de las imágenes de cerca y de lejos para así evitar los falsos positivos.
- El sistema puede albergar el rostro de varias personas, pero al aumentar la cantidad de personas de confianza, el tiempo de entrenamiento también aumentará, es por ello por lo que el LPBHFaces es la mejor opción, ya que su tiempo de ejecución y entrenamiento es mucho menor.
- Al identificar a las personas que entran y salen del condominio y tener una buena precisión al analizar los rostros, se reduce el índice de robos al condominio, ya que, si se detecta a una persona que no esté registrada, se procederá a intervenirla de inmediato y validar su acceso; de esta manera se reforzará la seguridad del condominio.

Comentario: La inclusión de este antecedente, es por la importancia que tendrá en el presente proyecto de investigación, en virtud a su desarrollo y la forma en cuál obtiene las imágenes de los rostros de las personas, las cuales serán evaluadas y comparadas para realizar

el reconocimiento facial; además tendremos en cuenta los resultados que se obtuvieron de los 3 algoritmos realizados, entre los cuales se menciona el método de FisherFaces, LBPHFaces y EigenFaces; para de esta manera optar por un algoritmo, el cuál sea el más idóneo para el sistema interno a implementar.

Calderón Ñaccha G, Santillán Paredes J, Masias Donayre Y, (2021), *"Plan de negocios para implementar un sistema de detección y alertas proactiva de inseguridad ciudadana con inteligencia artificial"*, Universidad ESAN, Perú.

Conclusiones:

- El censo del año 2017 realizado por INEI, destaca un incremento considerable en la población que vive en edificios. Los resultados indican que el porcentaje de la población que vive en los condominios y/o edificios se ha duplicado. Es así que existe notoriamente una tendencia del crecimiento de las viviendas particulares verticales en el Perú, ya que del año 2007 al 2017 estas crecieron en un 116.6 % y esta tendencia viene desde años atrás ya que el crecimiento del 2007 con respecto al 1993 en este tipo de viviendas en el país es del 78 % (Instituto Nacional de Estadística e Informática, 2017).
- La referencia indicada, unido al alcance geográfico del presente plan de negocio, favorece la implementación de un servicio de seguridad para el beneficio de las personas cuyo objetivo es la identificación oportuna de personas con conductas delictivas, haciendo uso de la Inteligencia Artificial.
- En base a lo analizado en la investigación de mercado, el 54.1 % está de acuerdo a contratar un servicio de seguridad controlado por un sistema informático autónomo, que sea un complemento a la seguridad física y presencial (vigilante, portero, guardián) de su condominio o edificio de residencia. Un 34 % totalmente de acuerdo. Podemos concluir que las personas contrarían un servicio como complemento más que un reemplazo al 100 %.
- Existe un riesgo de que si la demanda baja un 30 % al escenario proyectado, el proyecto deja de ser rentable, para validar que este factor no sea critico se hizo una simulación de la variación de esta, obteniendo como resultado que la probabilidad de éxito del proyecto representada en el valor del TIR y del VAN eran altas, del 83 % y 73 %, lo que se concluye que el proyecto también tenía una alta probabilidad de tener éxito.

Comentario: La inclusión de este antecedente, es por la importancia que tendrá en el presente proyecto de investigación, en virtud a su marco teórico y toda la información recopilada para el desarrollo de su proyecto de investigación realizado.

Ataucusi Romero E., (2021), *"Influencia de un sistema con reconocimiento facial y medición de temperatura en el control de acceso de participacipantes del programa trabaja Perú en el Distrito de Talavera"*, Universidad Nacional José María Arguedas, Perú.

Conclusiones:

- El sistema RFMTC influye significativamente en el control de acceso de participantes del programa Trabaja Perú en el distrito de Talavera.

- En esta investigación se determinó que el sistema RFMTC influye significativamente en el tiempo promedio de registro de asistencia de participantes del programa Trabaja Perú en el distrito de Talavera.
- También se encontró que el sistema RFMTC influye significativamente en el tiempo promedio de medición de temperatura corporal de participantes del programa Trabaja Perú en el distrito de Talavera.

Comentario: La inclusión de este antecedente, es por la importancia que tendrá en el desarrollo del presente proyecto, al utilizar las librerías y modelos usados para el reconocimiento facial, como son la librería de OpenCV y los modelos de Eigen Faces y Fisher Faces.

Suarez Días L., (2021), *"Sistema Inteligente de seguriad biométrica usando la IoT para la alerta de robos en las residencias de Villa María del Triunfo"*, Universidad Nacional Tecnolófico De Lima, Perú.

Conclusiones:

- El sistema desarrollado es capaz de alertar al usuario si se produce alguna intrusión enviando un mensaje a la aplicación de Telegram.
- Al momento de realizar una simulación en el software Proteus a pesar de que se obtenga los resultados que se esperan pueden llegar a fallar cuando se implementa de manera física debido a que un entorno virtual no se tiene en cuenta los factores externos que puedan causar pérdida de información en una comunicación inalámbrica por lo que se tiene que realizar una modificación en la Lógica de la Programación.
- El proceso de Reconocimiento Facial se llega a obtener después de una serie de Programas que parten desde como encender la cámara, detectar el rostro de una persona encerrándolo en un marco azul, tomar captura de los perfiles de las personas como lo son sus gestos sus estados de ánimo para obtener la mayor cantidad de características posibles y luego hacer un entramiento de estas imágenes para poder compararlas con la imagen de la persona que esté enfocando al momento que se activen los sensores.

Comentario: La inclusión de este antecedente, es por la importancia que tendrá en el desarrollo del presente proyecto; es decir, gracias a esto podemos seguir el proceso por el cual tiene que pasar el reconocimiento facial para poder utilizarlo.

2.2. Bases Teóricas

2.2.1. Inteligencia Artificial

La inteligencia artificial o llamada también IA por sus siglas en inglés, es una de las disciplinas académicas, la cual está relacionada con la teoría de la computación, cuyo fin es el de emular y/o simular algunas de las facultades intelectuales que el ser humano posee, esto se realiza gracias a los sistemas artificiales. Por otro lado, desde un punto de vista coloquial, podemos definir a la inteligencia artificial como un conjunto de técnicas creadas por el ser humano, las cuales tienen como finalidad el de emular y/o simular la inteligencia biológica.

Esta es una de las tecnologías, que gracias a su avance y desarrollo constante hasta la actualidad, ha tenido y sigue teniendo un gran impacto a nivel mundial puesto que, esta tecnología tiene muchas ventajas y/o beneficios, dependiendo del campo en la que sea utilizada. La inteligencia artificial se realizó con el objetivo primordial que es el de disminuir el factor del error humano; además, de tener habilidades predictivas según sea el caso que se presente. Se denomina inteligencia artificial a todas las tecnologías que tratan de emular alguna de las facultades que poseen los seres humanos.

En resumen, la inteligencia artificial hace referencia a todos los sistemas y/o máquinas, las cuales imitan la inteligencia que posee el ser humano para poder realizar las diferentes tareas que se necesite y/o requiera; esto hace que la inteligencia artificial pueda aprender de forma iterativa a partir de la información recibida. La inteligencia artificial se puede manifestar de diversas formas, algunos de estos tipos son los que mencionaremos a continuación:

- Los chatbots son un sistema el cual hace uso de la IA, con la finalidad de comprender de forma más rapida, los problemas que tiene un cliente o usuario de dicho sistema, para que posteriormente se pueda proporcionar la respuesta más eficiente posible, según el caso que se presente en interacción con el usuario.
- Los asistentes inteligentes son sistemas que también hacen uso de la IA, estos sistemas pueden analizar la información relevante y/o crítica, las cuales son obtenidos de un conjunto de datos de texto libre que son usados con el fin de mejorar la programación.
- Los motores de recomendación, al hacer uso de la IA, puede llegar a proporcionar diversas respuestas, las cuales son recomendaciones automatizadas para programas de TV, estas respuestas se obtienen según los hábitos de visualización de los usuarios.

(Oracle, 2023)

Enfoques de la Inteligencia Artificial

Dentro del campo de la IA podemos hablar sobre 2 enfoques principales, entre los cuales tenemos: la inteligencia artificial supervisada y no supervisada.

Inteligencia Artificial Supervisada La IA supervisada es uno de los enfoques que se encuentran dentro del campo de la IA, lo cuál implica que este modelo sea entrenado utilizando uno o varios conjuntos de datos, los cuales están previamente etiquetados; es decir, que el modelo sabe cuál es la respuesta que se espera al analizar cada uno de los datos que se encuentran en nuestro dataset.

Durante este proceso, los datos que sirven como entrenamiento de dicho modelo, son ejemplos que están compuestos por datos de entrada y datos de salida, en donde sabes que para cada dato de entrada "X", se tendrá un dato de salida "Y", los cuales son las respuestas esperadas y/o correctas; de esta forma haces que el modelo del algoritmo busque similitudes, características en donde se busca que exista una relación entre el dato de entrada y el dato de salida. La finalidad de dicho proceso, es que el modelo llegue a aprender a través de estos ejemplos dados, para que después llegue a realizar predicciones y/o clasificaciones correctas con datos que ya no estén etiquetados, que no se tengan sus datos de salida.

Por último, podemos decir que, este enfoque se basa en el aprendizaje a partir de ejemplos conocidos, donde dicho modelo aprende a mapear las datos de entrada para que después den como respuesta los datos de salida esperados. Al utilizar el conjunto de datos etiquetados para el entrenamiento; la IA supervisada, permite a dicho modelo la acción de capturar patrones y relaciones complejas, permitiendo su aplicación en diversas tareas como clasificación, regresión, detección de anomalías, entre otras más. Mediante la supervisión y retroalimentación proporcionada por los datos etiquetados, se logra mejorar la precisión y la capacidad predictiva del modelo, permitiendo la toma de decisiones automatizada y la resolución de problemas complejos en diferentes dominios.

Inteligencia Artificial no Supervisada La IA no supervisada es otro de los enfoque que se encuentran dentro del campo de la IA, el cuál se centra en el descubrimiento de patrones, estructuras y relaciones sobre uno o varios conjuntos, cuyos datos no se encuentran etiquetados; es decir, no se sabe cuales son los resultados esperados.

Por el contrario, a diferencia del aprendizaje supervisado en donde se utilizan datos previamente etiquetados, la IA no supervisada trabaja con datos cuya información no tiene ninguna salida predefinida; es decir, este modelo empieza desde 0, teniendo un desconocimiento sobre la respuesta esperada. La finalidad de este proceso es el de explorar los datos con el objetivo de buscar y lograr encontrar agrupaciones, similitudes y/o tendencias. Este enfoque llega a permitir que se obtenga un análisis y comprensión más profunda sobre los datos de entrada ingresados.

Al emplear la IA no supervisada, los modelos aprenden a través de la identificación de patrones que emergen, y de estructuras inherentes a los datos ingresados; esto se logra mediante diferentes técnicas existentes, una de ellas es la de agrupamiento (clustering) y la reducción de la dimensionalidad, donde los datos son organizados en diferentes grupos o son representados de una manera más compacta. Al explorar los datos que no se encuentran etiquetados, se busca descubrir una información la cual no se tiene con exactitud, por el contrario es algo que se tiene que identificar; para ello se identifica algunas anomalías y/o también se puede detectar relaciones que puedan llegar a ser inesperadas.

Machine Learning

El Machine Learning es una de las técnicas de IA más usada cuando se trata del aprendizaje automático, ya que dicha técnica es la ciencia que hace y/o utilizan los ordenadores para que puedan aprender a partir de los datos que se han llegado a obtener; esto indica que en vez de programar algún sistema o software, el cual necesita estar implementado para cada caso que pueda suceder; es decir, es muy primordial la implementación de cada solución específica para cada necesidad planteada. Esta tecnología por el contrario se enfoca primordialmente en buscar los diferentes patrones que se tiene en los paquetes de datos obtenidos, a este algoritmo se le alimenta o suministra con diferentes datos de un mismo tipo, dichos datos deben estar con sus respectivas etiquetas para que asi el algoritmo sea capaz de aprender a identificar los datos nunca antes visto; esto hace que la tecnología sea mucho más flexible puesto que la construcción y/o implementación de un programa para la clasificación de imágenes de tipos de flores no tendrá cambios significativos comparado con otro programa de clasificación, como la clasificación de animales mediante imágenes. Todo lo mencionado con anterioridad son algunas de las técnicas, algoritmos y/o modelos basados en las redes neuronales o llamados también redes artificiales, que tienen como objetivo emular y/o simular el comportamiento de las neuronas cerebrales en los seres humanos, ya que existe una gran diferencia entre la forma en que funciona el cerebro humano que es totalmente distinto al funcionamiento de un computador, estas neuronas se juntan igual que en el cerebro del ser humano y llegan a formar redes. (Inc., 2023)

Deep Learning

Las redes neuronales artificiales se dividen en dos tipos, una de ellas son las redes superficiales y otras que son las redes profundas; por su parte las redes superficiales son usadas para poder realizar cualquier tarea simple que se tenga, los cuales no necesiten de mucho procesamiento de datos; por otro lado tenemos las redes profundas las cuales se denominan como otra tecnología llamada tambien aprendizaje profundo o Deep Learning. Una red neuronal profunda permite realizar de manera automática una extracción de las características de la información que se le suministra como entrada; dicha extracción se realiza de manera jerárquica, pasando primero por una capa de características de bajo nivel, luego encontrando las de nivel intermedio y así hasta llegar a las de más alto nivel. La gran diferencia entre el aprendizaje profundo y el aprendizaje automático es que en el aprendizaje automático se deben extraer manualmente las características de una imagen el cual logra permitir su clasificación, mientras que el aprendizaje profundo se encarga de extraer las características por medio de la red y es así como la misma red clasifica la imagen. (Inc., 2023)

2.2.2. Detección de rostros

La detección de rostros es una función la cual tiene como objetivo, buscar y encontrar uno o más rostros dentro de un material visual (imagen o video); estos rostros existentes deben ser reconocidos a pesar de las diferentes dificultades que se pueda hallar en el material visual; dicha detección tiene como resultado la obtencion de diferentes puntos dentro del cuál, en su área se encuentra un rostro, es así que la detección de rostros busca localizar los diversos "bounding box"(Sotaquirá, 2020) que se pueda encontrar en un material visual.

Además la deteccion de rostros es un paso o etapa previa para el análisis de rostros; es decir, la detección de rostros es esencial para realizar el reconocimiento facial.

La detección de rostros busca diferenciar otras partes del cuerpo y hasta objetos del rostro mismo de una persona, estos objetos pueden ser edificios, arboles, sillas, peluches, etc. A continuación mencionaremos algunas de las diversas dificultades que uno puede encontrar al momento de realizar la detección de rostros:

- Color de piel de un persona.
- Orientación de un persona.
- Oclusión del rostro.
- Cambios de iluminación.
- Los diferentes gestos que puede tener una persona.
- Las diferentes poses en las cuales se puede encontrar un persona.
- La distancia entre la cámara y el rostro, hace referencia al tamaño del rostro.

(RecFaces, 2023)

Figura 2.1: Dificultades al momento de realizar la detección de rostros.

(a) Dificultad con la pose de la persona

(b) Dificultad con el gesto de la persona

Fuente: (Sotaquirá, 2020)

Ventajas y desventajas de la detección de rostros

En la detección de rostros como en otros temas de interés, tenemos ciertas ventajas y desventajas como se pueden apreciar en la siguiente tabla **??**:

Tabla 2.1: Ventajas y desventajas de la detección de rostros

Ventajas	**Desventajas**
Brinda una mayor seguridad	Es vulnerable
Es fácil de usar	Tiene problemas de privacidad
Se puede automatizar	Requiere de almacenamiento y equipamiento

Fuente: (RecFaces, 2023)

Uso de la detección de rostros

En la actualidad, podemos apreciar diversos usos que se le da a estos algoritmos de detección de rostros, a continuación mencionaremos algunos de sus usos más frecuentes:

- En la captura de movimiento facial, esto se puede apreciar en diversos filtros de rostros que se tiene en aplicaciones como instagram, tiktok o snapchat, estas aplicaciones realizan el seguimiento del rostro para poder aplicar un filtro seleccionado por el usuario.
- En el reconocimeinto facial, esto se puede apreciar en las diversas cámaras de seguridad modernas las cuales poseen un software para poder realizar dicho reconocimiento facial, pero antes de ello necesitan detectar los rostros para luego analizarlos y compararlos con otros.
- En la fotografía, esto se puede apreciar en los diverentes celulares con alta tecnología, los cuales usan dicha detección de rostros para poder enfocar y visualizar mejor una imagen o video, según lo requiera el usuario.
- En la inferencia emocial, esto se da para poder analizar las diferentes emociones con la que se encuentra una persona, para ello se necesita realizar la detección de rostro sobre una persona, para seguir con su respectivo análisis.
- En el marketing, esto se puede apreciar en las diferentes redes sociales donde se usa la detección de rostros para poder etiquetar a una persona, lugar o negocio.
- En la lectura de labios, para poder generar los subtitulos de un material visual, se necesita el reconocimiento de rostros para poder identificar los labios de una persona para luego analizarlos.

(RecFaces, 2023)

Herramientas para la detección de rostros

OPENCV OpenCV es una biblioteca que fue construida en los lenguajes de C y C++; el cual fue diseñado para trabajar con algoritmos orientados a la visión computacional, como tambien al procesamiento y análisis de imágenes, hasta llegar a la detección de rostros. Entre sus aplicaciones se incluye la detección de rostros, uno de los algoritmos más conocidos en la actualidad sobre esta biblioteca, es el algoritmo denominado çascadas de Haar", el cuál fue introducida por Viola y Jonnes. (RecFaces, 2023)

MATLAB Matlab es otra de las tecnologías disponibles para realizar la detección de rostros, esto gracias al detector de objetos que este posee y que se realiza en cascadas, dicha tecnología se basa también en el algoritmo de Viola-Jones por una pequeña agregación y hasta combinación con el algoritmo de Kanade-Lucas-Tomasi (KLT); esta tecnología puede ser usada en diferentes materisuales visuales como son los videos, en donde podemos seguir a detalle un rostro detectado. (RecFaces, 2023)

TENSORFLOW TensorFlow es una librería que se encuentra en la categoría de aprendizaje automático; además de ser de código abierto, fue creada por Google. En dicha librería, puedes construir y entrenar tu propio modelo de detección rostros o de algún objeto en específico, ajustando los parámetros a uno de los modelos que ya se encuentran predefinidos en el denominado "Zoo de Modelos". Uno de estos algoritmos que se encuentran en la categoría de aprendizaje profundo, divide la imagen en un conjunto de cuadros delimitadores, extrayendo características visuales de cada cuadro y clasificándolas según el conjunto de reglas proporcionado. (RecFaces, 2023)

REDES NEURONALES Las redes neuronales tienen como finalidad poder imitar las capacidades y bondades que posee el cerebro humano, pero a diferencia de este que funciona a base de neuronas las redes neuronales trabajan mediante nodos, los cuales son usados para el procesamiento, y así reconocer y clasificar objetos, llegando al punto de predecir comportamientos y/o sucesos. Comúnmente se usan las redes neuronales para realizar diversas funciones, algunas de estas son la detección de rostros y el reconocimiento facial. (RecFaces, 2023)

2.2.3. Python

Python es un lenguaje de programación el cual es muy conocido a nivel mundial, esto se debe a la gran comunidad que tiene y a las diversas librerias que posee; debido a todo ello, python es considerado como una de las mejores alternativas al día de hoy para poder ser consideraba uno de los mejores lenguajes de programación así como uno de los más sencillos. (RecFaces, 2023)

Python es usado para muchas cosas hoy en día, es tanto su uso que se puede ejecutar en las diferentes plataformas que existen, además que se puede descargar de forma gratuita y sin pago, tambien se puede integrar con facilidad a todo tipo de sistema y el tiempo de desarrollo va en aumento.

OpenCV

OpenCV es una librería que también se encuentra en el lenguaje de programación de python y que se importa para poder usarse en la detección de rostros, esto gracias a un modelo entrenado, el cual está implementado con aprendizaje profundo.

OpenCV ya tiene algunos modelos listos para usarse, este es el caso de çascadas Haarçuyo modelo ya se encuentra entrenado y esta listo para usarse en la detección de rostros (Rosebrock, 2018). Por otro lado OpenCV posee un detector de rostros oculto, el cual se basa en el aprendizaje profundo; además de esto OpenCV te permite subir la arquitectura y los pesos de un modelo para poder realizar la detección de rostros, todo gracias a las redes neuronales profundas.

MediaPipe

MediaPipe es considerada como una tarea de detección de rostros , el cuál le permite detectar todos los rostros existentes de un material visual como puede ser un imagen o video; dicha tarea utiliza un modelo de ML, el cuál funciona con un flujo de imágenes.

Esta tarea de detección de rostros trae como resultado varios puntos claves de una imágen única y/o continua, para esto se necesita los siguientes puntos (Google, 2023):

- Ojo izquierdo.
- Ojo derecho.
- Punta de la nariz.
- Boca.
- Tragón del ojo izquierdo.
- Tragón del ojo derecho.

Al obtener todos estos puntos o coordenadas claves, se podrá obtener el área donde se encuentra ubicado el rostro de una persona, pueden haber varios puntos según a la cantidad de rostros que existan en una imagen; es decir, 6 puntos o coordenadas hacen referencia a un rostro.

2.2.4. Reconocimiento facial

Historia

Una de las formas más comunes de identificar a una persona es mediante su rostro, ya que este posee varios rasgos que son muy particulares para una persona; gracias a estos rasgos o características, se hace mucho más sencillo el reconocimiento facial o identificación de una persona por medio de la vista humana, ya que los seres humanos somos capaces de diferenciar a las diferentes personas con tan solo la descripción de ellos mismos. En los inicios de la llamada visión artificial, el reconocimiento facial ha llegado a ser estudiado y desarrollado debido a su importante práctica, la cual fue de interés teórico para los diferentes científicos que se inclinan por esa rama.

El reconocimiento facial automatizado es un concepto relativamente nuevo. Dichas investigaciones sobre este tipo de herramienta se inciarion en los años 60, cuando el matemático, informático y destacado educador estadounidense Woodrow Wilson Bledsoe alias Woody junto a su equipo de investigación lograron desarrollar en ese entonces los primeros sistemas de reconocimiento facial, estos sistemas eran semiautomáticos, ya que necesitaban de una persona la cual debia estar presente en todo momento para poder localizar todos los rasgos característicos de la persona en las fotografías capturadas y/u obtenidas.

En el año 1988, L. Sirobich y M. Kirby llegaron a aplicar un análisis el cual requeria de componentes principales o llamado también PCA por sus siglas en inglés, la cual es una

técnica estándar que va relacionado con el algebra lineal, para poder llegar a solucionar el problema del reconocimiento facial. En ese entonces, este análisis fue considerado como un hito al mostrar que se necesitaba menos de 100 componentes para poder lograr cifrar de forma correcta y/o acertada la imagen de un rostro, la cual este alineada como normalizada de forma conveniente. Al transcurrir un año, T. Kohonen introdujo este método de reconocimiento facial, el cual se fundamenta en la descripción de los rasgos faciales que posee el rostro de una persona, esto se da gracias a la extracción de los autovectores de una matriz de autocorrelación conocida mayormente como Eigen Faces. (Domínguez Pavón, 2017)

En 1991 Turk y Pentland lograron demostrar que el error porcentual en la precisión y exactitud del modelo el cual es utilizado al decodificar las Eigenfaces, pueden llegar a usarse como alternativa para detectar todos los rostros posibles dentro de las imágenes, un descubrimiento el cual llego a permitir el uso de estos en sistemas ya automatizados de reconocimiento facial, los cuales llegaron a realizar su respectivo análisis en tiempo real. Si bien la aproximación o porcentaje de certeza del resultado era un tanto forzada, esto fue de un gran interés significativo para los desarrolladores, lo que llevo a mejorarlos en posteriores desarrollos de estos sistemas de reconocimiento facial.

Esta tecnología que se basa en el reconocimiento facial, capturó la atención del público en general, tanto de personas naturales, como de empresas, fabricas, entre otros. A partir de esta reacción obtenida por la demanda que surgió de esta tecnología, el cual fue apreciada en una implementación de prueba que se dio en la Super Bowl de la NFL, el cual fue transmitido y compartido por los diferentes medios de comunicación en enero del 2001, la cual capturó varias imágenes de vigilancia para luego compararlas con una base de datos, la cual tenía almacenaba diversas fotos digitales. Esta demostración hizo que se iniciara un muy requerido debate sobre cómo usar esta tecnología, para satisfacer tanto las necesidades nacionales como el de las personales, en el transcurso del debate se tomaba en consideración las preocupaciones sociales y la privacidad de datos que existian para el público en general.

Por consiguiente luego de estas investigaciones, el interés sobre las técnicas que se usaba para realizar el reconocimiento facial aumentó considerablemente y, desde ese entonces se ha llegado a desarrollar e implementar numerosas técnicas que han sido desarrolladas en esta área, llegando a conseguir muy altos niveles de perfeccionamiento entre eficiencia y eficacia.

Fundamentos del reconocimiento facial

El rostro humano, como bien se indico con anterioridad, nos proporciona una gran cantidad de datos e información sobre una persona en específico, permitiéndonos así discernir e identificar a simple vista a diferentes personas según las informaciones recopiladas de ellos mismos. Las imágenes de rostros albergan y/o almacenan un gran conjunto de rasgos que se encuentran localizados en posiciones similares en toda la población; es por ello que un sistema de reconocimiento facial puede ayudar y lograr beneficiarse de esta característica en común de las personas. En general, los sistemas de reconocimiento facial que emplean algún tipo de técnica basado en modelos, son los que aprovechan estos rasgos particulares de las personas. Más adelante describiremos algunas de estas técnicas las cuales son usadas para el correcto funcionamiento del reconocimiento facial (Domínguez Pavón, 2017).

A continuación procederemos a describir algunos de estos ragos más significativos de las personas, las cuales se encuentran ubicadas en el rostro humano:

- **Orejas.** Habitualmente la variabilidad que presentan entre individuos es eminentemente geométrica, siendo el tamaño la característica que mejor las define. Las orejas al estar situadas en los laterales de la cara, pueden estar ocultas por el cabello, generando variaciones no deseadas. Es por ello por lo que en muchos sistemas de reconocimiento-facial, la región de la cara que se extrae las excluye para evitar esta variabilidad.

- **Cejas.** Están compuestas por vello situado en la parte superior de la cara justo encima de los ojos, estos ofrecen diferentes características a tomar en cuenta como son: el grosor, la forma, el espesor y el color del vello. Su localización puede estar modificada por la expresión aunque por lo general no existe mucha variación del resto de características frente a diferentes gestos.

- **Ojos.** Los ojos, dada su complejidad, son quizá unos de los rasgos más discriminativos del rostro humano. Situados en la mitad superior de la cara, están compuestos por pestañas, párpados y el globo ocular que a su vez se diferencia en córnea, iris y pupila. Ofrecen gran variabilidad entre sujetos puesto que, su geometría es diferente para cada uno y el iris, un rasgo biométrico por sí solo, dota a los ojos de gran información discriminativa. El inconveniente de los ojos es que en ocasiones los párpados ocluyen parcial o totalmente este rasgo, y además son bastante sensibles a cambios de expresión.

- **Nariz.** La nariz está situada aproximadamente en el centro de la cara. Su forma varía en gran medida entre los usuarios y la misma no suele ser afectada en los cambios de expresión. Los dos orificios nasales suelen ser un buen punto característico cuando se miden sus respectivas distancias con otro punto en específico.

- **Boca.** Por último, la boca está situada en la parte inferior de la cara, es otro rasgo característico que facilita información del individuo. Como característica particular, debido a la gran flexibilidad y diversidad de movimientos que puede realizar este rasgo, es posible encontrar gran variabilidad en un mismo sujeto dependiendo del gesto de la persona, como por ejemplo si este está sonriendo, si tiene la boca abierta, está sacando la lengua, entre otros. Los labios son el componente que siempre está visible y que suelen definir el aspecto de la boca.

Ventajas e inconvenientes

La herramienta de reconocimiento facial presenta ciertas características favorables para su uso, las cuales la convierten en una de las técnicas más viables para ser utilizado en ciertos ámbitos donde se requiera y/o necesite de estas técnicas biométricas. Algunos de estos aspectos más determinantes se mencionarán a continuación (Domínguez Pavón, 2017):

- Simetría. El rostro es uno de los rasgos que tiene una simetría bastante elevada, algo de lo podemos sacar provecho, ya que es un beneficio el cual puede obtenerse mediante las tareas de localización y/o extracción de distancias.

- El rostro es un rasgo muy particular de las personas, el cual por cierto es muy visible para otras personas, es por ello que facilita la tarea de recopilación y obtención de datos que en este caso son las características, lo que conlleva a que sea un poco intrusiva para las personas y no se sientan incómodas. La obtención de estos datos fueron gracias a una cooperación y vinculación que se da con las cámaras de seguridad, ya que estos equipos son capaces de capturar el rostro de una persona, el cual se obtendrá

con una calidad y resolución aceptable o mayor. Esto permite diseñar sistemas de reconocimiento facial, los cuales no serán percibidos por las personas, ni tendran la menor idea o no se percatarán de que están siendo identificados por esta herramienta que es el reconocimiento facial.

- Esta herramienta tiene un gran poder discriminante, ya que el rostro es uno de los rasgos con mayor poder discriminativo, por ser rasgos con una gran cantidad de información sobre las personas. La obtención y almacenamiento de una gran cantidad de imágenes de rostros en una base de datos, no llega a deteriorar en tal manera la eficacia y eficiencia de dicho sistema, puesto que aun así se logra obtener buenas tasas y probabilidades de reconocimiento facial.
- Tiene una disponibilidad extendida ya que el ser humano ha utilizado el rostro como un método de reconocimiento básico en una gran variedad de ámbitos, esto hizo que se creen y existan una gran variedad de bases de datos, el cual almacena estos rasgos obtenidos por la mayor parte de la población. Al ser este rasgo uno de los más utilizados, se han llevado a cabo una gran cantidad de investigaciones los cuales tienen como tema principal al reconocimiento facial, que se obtuvieron como resultado de la unión de diferentes técnicas y algoritmos de los cuales hoy en día nos podemos beneficiar.

Al igual que las ventajas descritas anteriormente, la herramienta de reconocimiento facial también cuenta con algunos puntos desfavorables o inconvenientes que serán mencionados a continuación:

- Variabilidad. Uno de los grandes inconvenientes que presenta esta técnica, es la plasticidad del rostro humano respecto a los gestos y a que es un rasgo que no presenta invariabilidad temporal, teniendo en cuenta varios rostros de un sujeto a lo largo de los años, los cuales son diferentes a simple vista.
- Entornos no controlados. Cuando el entorno no está controlado, los cambios en la pose, oclusiones y baja calidad de las imágenes tomadas, hacen que el sistema pierda su fiabilidad presentando inconvenientes a la hora de reconocer personas en un entorno real.
- Situaciones particulares. Por último, existen ciertas situaciones particulares en los que el reconocimiento facial no es viable, ya que debido al fácil acceso del rasgo, un impostor podría utilizar imágenes de rostros de otras personas sin mucha dificultad. En este sentido, en los últimos años se está prestando mucho interés estudiando y generando soluciones.

Principios del reconocimiento facial

Hoy en día muchas de las personas se sienten preocupadas por el uso de este sistema, el cual hace uso del reconocimiento facial, especialmente para la video vigilancia, lo cual es muy entendible puesto que, dicho sistema puede tener fallas y lograr identificar de forma errónea a una persona, la cual no se tiene almacenada en la base de datos y es usada para poder realizar el respectivo análisis y comparación de imágenes, esta falla o identificación errónea puede llegar a suscitar la detención de una persona por motivos incorrectos, ya que este suceso ocurriria por una falla en el sistema, lo cual no deberia de pasar en ninguno de

los casos posibles. Otra interrogante que surge en las personas, es que sucede si un sistema llega a funcionar pero con una tasa menor de reconocimiento facial, esto puede ser debido a que los datos recopilados son muy pocos e insuficientes, logrando que las personas que pertenecen a estos grupos de reconocimiento, serán más propensas a ser víctimas de una identificación errónea.

En 2019, se actualizó las pautas sobre la privacidad de datos; teniendo en cuenta el crecimiento de la inteligencia artificial con el avance constante que se da en el día a día en la rama de la tecnología; gracias a estos avances se lograron crear y desarrollar los drones y también los sistemas de reconocimiento facial más sofisticados que podemos observar y encontrar en la actualidad. Estas pautas se siguen construyendo en base al trabajo que realizan las diferentes organizaciones en todos los ámbitos. En 2018, Microsoft realizó una publicación importante, el cual hablo sobre la importancia de regular el reconocimiento facial puesto que, es necesario para evitar que las empresas tecnológicas tengan que elegir entre responsabilidad social y éxito en el mercado (Thales, 2023). Dicha publicación se constituye con seis principios rectores, los cuales son los siguientes:

- **Justicia.** La tecnología de reconocimiento facial debe tratar a todas las personas de manera equitativa y justa.
- **Transparencia.** Las empresas tecnológicas deben documentar las capacidades y las limitaciones de la tecnología.
- **Responsabilidad.** Debe haber un nivel apropiado de control humano para los usos que puedan afectar a las personas de manera consecuente.
- **No discriminación.** Los términos de servicio deben prohibir la discriminación ilegal.
- **Notificación y consentimiento.** Las empresas deben proporcionar aviso y consentimiento seguro cuando implementan el reconocimiento facial.
- **Vigilancia legal.** Debe haber protección en las libertades democráticas de las personas en los escenarios de vigilancia policial.

Eigen Faces

Eigen Faces es una técnica que se apoya en el Análisis de Componentes Principales (PCA), una herramienta matemática, para llevar a cabo la compresión de imágenes y el desarrollo de sistemas de reconocimiento facial. Esta técnica es fundamental para comprender el funcionamiento de la tecnología del reconocimiento facial y constituye un componente esencial de métodos más avanzados. Su proceso consta de dos fases distintas: entrenamiento y clasificación.

En la fase de entrenamiento, se utiliza PCA para construir un espacio de facciones, también conocido como .^{ei}genspaces", a partir de imágenes de rostros faciales para realizar el entrenamiento. Este espacio se representa como una matriz de vectores propios (eigenvectors o eigenfaces), que encapsulan la variación de los valores de gris, en cada píxel del conjunto de imágenes empleadas durante el PCA.

La generación de eigenfaces implica normalizar un amplio conjunto de imágenes digitalizadas de rostros humanos, capturadas bajo condiciones de iluminación uniformes y

alineadas respecto a los ojos y la boca. Así, se obtiene un conjunto de ïmágenes normalizadas"mediante un análisis estadístico. Cualquier rostro humano puede expresarse como una combinación ponderada de estos estándares. Durante la fase de clasificación, una imagen de rostro facial desconocida se proyecta en el espacio de facciones, y mediante la distancia euclidiana, se busca la imagen proyectada más similar.

No obstante, la sensibilidad de esta técnica a las variaciones de iluminación constituye una limitación significativa. El control preciso de las condiciones de iluminación es crucial para su eficacia; además, cuando se desean agregar nuevas imágenes o sujetos que no formaron parte del entrenamiento original, es necesario repetir el PCA y volver a proyectar todas las imágenes, lo que puede ser un proceso intensivo dependiendo al numero de personas y al número de imágenes por persona. (Ottado, 2010)

Fisher Faces

La técnica de reconocimiento facial conocida como Fisher Faces, presenta diversas ventajas que la colocan en una posición destacada en comparación con otras técnicas disponibles, como es la técnica de Eigen Faces. Destaca por su velocidad, eficiencia y la capacidad de operar eficazmente en un gran número de rostros en un tiempo reducido. A diferencia de Eigen Faces, Fisher Faces resuelve problemas asociados con la iluminación; mientras que Eigen Faces requiere que las imágenes estén frontalmente alineadas y bajo condiciones de iluminación análogas, Fisher Faces muestra una menor sensibilidad a las variaciones en la iluminación y a los ángulos de las caras en las imágenes.

El enfoque de Fisherfaces se centra en maximizar la varianza entre clases (entre individuos) y minimizarla dentro de las muestras de la misma clase (de la misma persona). Esto resulta en un rendimiento superior cuando hay variaciones en la iluminación y expresión en comparación con las imágenes de entrenamiento. Para lograr esto, se requiere la captura de múltiples imágenes de cada sujeto en diversas condiciones de iluminación y pose, que sean representativas de las variaciones presentes en situaciones reales.

Esta técnica ha sido adaptada en diferentes tecnologías, así como en sistemas de reconocimiento facial debido a su bajo costo computacional, rapidez de implementación y su naturaleza de algoritmo libre, de uso general. Esto lo convierte en una herramienta versátil para clientes de empresas de diferentes tamaños.

Inicialmente desarrollada como software de seguridad, esta tecnología respondió a la necesidad de una empresa de seguros de contar con un sistema confiable de reconocimiento facial al momento de procesar reclamaciones de seguros de vida. El sistema puede realizar el reconocimiento facial y de microexpresiones, tanto en fotografías como en videos o mediante una transmisión en vivo. Esta solución contribuye a fortalecer la seguridad de las empresas, sus redes y propiedades, al validar la identidad de una persona y establecer claves de acceso basadas en expresiones difíciles de falsificar, como levantar cejas o guiñar un ojo. (Ottado, 2010)

LBPH Faces

El algoritmo de Patrones Binarios Locales (LBPH), concebido originalmente para la descripción de texturas, aborda la limitación de los descriptores de texturas convencionales al aplicarse a rostros. Estos descriptores, al basarse en promedios de información en regiones faciales, no preservan la relevancia de las relaciones espaciales cruciales para la descripción facial. Para lograr una descripción global, la imagen de rostro facial se segmenta en diversas regiones, a las cuales se les aplica un histograma que genera el operador LBPH, proporcionando información independiente por región. Estas descripciones locales se concatenarán para formar una descripción global cohesiva del rostro.

En el proceso de asignar etiquetas a cada píxel de la imagen, el LBPH toma en cuenta la disposición de los vecinos, siguiendo una serie de pasos específicos para el reconocimiento de imágenes. Una máscara de tamaño definido (8x8) recorre la imagen de manera iterativa, seleccionando un píxel central y sus vecinos en cada iteración. Este píxel central se compara con cada vecino de manera ordenada, asignándose un valor de 1 cuando el píxel central es menor que el vecino, y 0 en caso contrario. (Verdeguer Valderrama, 2022)

2.2.5. Cámaras de seguridad

Las cámaras de seguridad llamadas también cámaras de video vigilancia son un sistema, el cual consiste en brindar seguridad y vigilancia en un punto dado, lugar donde se encuentra instalada una cámara; dichas cámaras se encuentra ubicadas en lugares o puntos estratégicos, estos pueden estar dentro o fuera de una propiedad privada como es una vivienda, condominio, empresa, entre otros; también se encuentran en una propiedad pública, las cuales son áreas comunes donde las personas transitan libremente como son los parques, plazas, entre otros. Las cámaras de seguridad también son conocidos como circuito cerrado de televisión o por sus siglas CCTV que significa çloset circuit televition".

El circuito cerrado de televisión, se encuentra compuesto por una o varias cámaras de video vigilancia, las cuales están interconectadas a uno o varios monitores de video y/o televisores, los cuales son usados para reprodir las imágenes que se encuentran almacenadas y captadas por dichas cámaras. Esta tecnología de video vigilancia es usado especialmente con el fin de identificar intrusos o a cualquier otra persona que realice alguna actividad indebida o sospechosa, el cual ponga en riesgo la integridad de un lugar, objeto o individuo.

Estas cámaras de seguridad realizan vigilancia de forma remota de cualquier lugar, objeto y/o persona; esto se debe a los diferentes métodos y formas en las cuales se suscita un robo como podemos apreciar en la imagen 2.2; ante la posible sospecha de algo extraño se puede activar o no una alarma en caso sea necesario, todo ello desde el cuarto de control o habitación donde se encuentre la base central de dichas cámaras, desde el cual una persona puede configurar varias de las características de una cámara como son el enfoque, la dirección, la inclinación y hasta la vista panorámica (OVACEN, 2023).

Figura 2.2: Métodos más frecuentes de robo.

(OVACEN, 2023)

Las cámaras de video vigilancia, son las encargadas de capturar y/o grabar todo lo ocurrido en un determinado lugar donde se encuentren ubicadas e instaladas dichas cámaras como puede ser una casa, negocio, parque, entre otros. Contar con este sistema de seguridad como son las cámaras de video vigilancia, han llegado a proporcionar a una persona o grupo de personas, la sensación de seguridad y protección para sí mismos logrando un bienestar en ellos.

Hoy en día, se ha vuelto algo muy común y hasta casi necesaria el disponer de una cámara de video vigilancia, ya que este llega a ser una de las soluciones más factibles de las personas y/o familias, para que estas puedan mantenerse seguras y protegidas, ya que estas tecnologías son una de las soluciones muy óptimas para lograr mantener alejados a los ladrones y así proteger tanto el hogar como a sus habitantes. Gracias al desarrollo y el avance constante de la tecnología, se ha logrado obtener una gran variedad de equipos eficientes y con diversas funcionalidades incorporadas, las cuales ofrecen seguridad y tranquilidad a las personas, asi como otras funcionalidades y/o necesidades, los cuales se encuentran a precios muy competitivos y económicos. Esta amplia variedad de cámaras los veremos posteriormente y de forma más detallada.

Continuando con las cámaras de video vigilancia, podemos decir que esta tecnología es una de las herramientas que brindan un sentimiento mayor de seguridad a las personas ante intentos de robo u otro tipo de acontecimientos peligrosos que se susciten y lleguen a dañar la integridad de una persona; esto es porque las cámaras de seguridad tienen una mayor probabilidad de alertar y así detener un posible robo, las probabilidades en cuestión podemos apreciarlas en la siguiente imagen 2.3. Asi mismo, estas cámaras de video vigilancia sirven para que uno pueda tener pruebas de carácter legal, si se llega a tener un juicio, producto de un acontecimiento ocurrido; y es que de solo contar con este tipo de artefacto, hace que estemos mucho más seguros y protegidos.

Figura 2.3: Medidas de protección más utilizadas.

(OVACEN, 2023)

Ventajas de las cámaras de seguridad

- Existe una gran variedad y diferentes tipos de cámaras de seguridad que se adaptan a las necesidades que las personas necesiten; además, son muy accesibles puesto que, disponen de cámaras a precios económicos.
- Son muy útiles, ya que te permiten monitorear las actividades de las personas que visitan un lugar en específico como puede ser una oficina u hogar.
- Te ayudan a tomar buenas decisiones al momento de resolver algún problema o disputa.
- En el aspecto psicológico, las cámaras son muy importante, ya que debido a este sistema antirobo se proporciona una tranquilidad mental a las personas que habitan una vivienda, advirtiendo a cualquier persona que intente entrar a dicha vivienda.
- Desalienta un posible robo, dado que son realmente visibles.
- Recopila pruebas de los posibles delitos, robos que se practican en una propiedad.

(Acacio, 2019)

Desventajas de las cámaras de seguridad

- Las cámaras solo tienen la funcionalidad de grabar y vigilar, mas no podrá detener a un intruso si este irrumpe en una propiedad.
- Vulnera la privacidad de las personas, ya que muchas veces, los propietarios de un inmueble ponen cámaras en lugares destinados a la privacidad sin respetar los límites dados.
- Pueden ser engañadas. Por desgracia, algunas cámaras de seguridad no aseguran el 100 % de confiabilidad. Eso dependerá de la marca de cámara que se elija.

Cámaras de seguridad convencionales

Las cámaras de seguridad convencionales, también conocidas como cámaras analógicas, son dispositivos usados para la vigilancia y grabación en tiempo real de un área específica. Estas cámaras capturan imágenes que se transmiten a un sistema de grabación o monitoreo por medio de cables coaxiales. A pesar de que su calidad de imagen es generalmente menor en comparación con las cámaras digitales modernas, siguen siendo populares debido a su sencillez y bajo costo. (Engineering, 2023)

En cuanto a su funcionamiento, las cámaras de seguridad convencionales suelen estar conectadas a un DVR (grabador de video digital), el cual convierte la señal analógica en digital para su almacenamiento y visualización posterior. Aunque la tecnología ha avanzado hacia sistemas más complejos como las cámaras IP, las cámaras convencionales continúan siendo una opción viable para quienes buscan una solución de seguridad básica, especialmente en lugares donde no se requiere alta resolución de imagen o funciones avanzadas como análisis de video o acceso remoto vía internet.

Para el presente proyecto, se denominará como cámaras convencionales a todo tipo de cámaras que permita realizar la acción de grabar y almacenar un determinado lugar en un determinado momento, con la única condición que estas cámaras sean aquellas que no cuentan con funcionalidades avanzadas, como detección de movimiento, zoom, análisis de imágenes o grabación en condiciones de oscuridad mediante infrarrojos, entre otros.

Tipos de cámaras de seguridad

Con el constante avance que se da en el día a día en el campo de la tecnología, se desarrollo una gran variedad de diferentes cámaras de seguridad, las cuales se diferencian por las características que estos posean y algunas funcionalidades que se acoplan a estas tecnologías. A continuación mencionaremos los tipos de cámaras de seguridad los cuales podemos encontrar en el mercado de esta industria, así mismo describiremos en que se diferencian una de otras y que características poseen cada uno de estos tipo.

Cámara para interior. Este tipo de cámaras están diseñadas para los interiores de una propiedad, es así que estás cámaras tendrían como visión áreas internas como entradas, pasillos, comedor, entre otros. Estas cámaras son las más sencillas que podemos encontrar en el mercado, por consiguiente son una de las más baratas, dado que no poseen muchas características, ni protecciones a las cámaras en sí, como otros tipo de cámaras que si las poseen.

Figura 2.4: Cámara interior.

(OVACEN, 2023)

Cámara con movimiento y zoom. Este tipo de cámaras suelen usarse en espacios amplios o de grandes dimensiones, con una base central (centro de control) donde una persona o varias son las encargadas de monitorizar y vigilar dichas cámaras. Estas cámaras son robóticas, por consiguiente tienen la característica de movimiento, lo que les da mayor funcionalidad sobre su eje, giro, inclinación y hasta la capacidad de realizar zoom, es por ello que estas cámaras son más eficientes y flexibles.

Figura 2.5: Cámara con movimiento y zoom.

(OVACEN, 2023)

Cámara de infrarrojo o visión nocturna. Este tipo de cámaras son utilizadas en lugares o propiedades, donde la iluminación que se posee de los diferentes espacios o ambientes son parciales o casi nulas; es decir, que no se tiene buena iluminación en algunas partes de dicha propiedad. Estas cámaras normalmente graban todo el día por la condición de la luz solar, mientras que por la noche la condición cambiar ya que de forma automática se enciende su función de infrarrojo con una visión en escala de grises, es decir en blanco y negro. Este tipo de cámara son una de las más caras de esta industria, por la visión nocturna que posee mediante el uso de LED.

Figura 2.6: Cámara de infrarrojo o visión nocturna.

(OVACEN, 2023)

Cámara oculta. Este tipo de cámaras son llamadas tambien cámaras espía, los cuales permiten monitorear y vigilar una casa, oficina y hasta un negocio o empresa pasando desapercibidos para las personas, ya que no son visibles a simple vista por asi decirlo. Estas cámaras se pueden introducir dentro de algún objeto, ya que estas cámaras pueden llegar a ser las más pequeñas posibles, algunos de estos lugares donde se encuentran ubicado estas cámaras son los sensores de movimiento, detectores de humos, enchufes, maceteros, etc.

Figura 2.7: Cámara oculta.

(OVACEN, 2023)

Cámara IP. Este tipo de cámaras son aquellas las cuales se conectan de forma directa al Internet, mostrando asi la imagen que fue captada. Estas cámaras son las más utilizadas, vendidas y adquiridas por las personas, dado que actualmente estas cámaras tienen incorporadas la tecnología del wifi, esto hace que dichas cámaras establezcan una conexión sin necesidad de un cable físico, lo que convella a ser manejadas desde una corta o larga distancia, y también puedan llegar a visualizar la toma y/o captura de imágenes desde diferentes lugares desde el alcance de su ordenador, smartphone y/o tablet. La principal ventaja de una cámara IP, consiste en que este tipo de cámara es un dispositivo de vigilancia la cual te permite ver las imágenes en tiempo real, mientras una persona se encuentra lejos de dicho lugar o ubicación, esto sucede a través de la conexión con una dirección IP de Internet.

Figura 2.8: Cámara IP.

(OVACEN, 2023)

Cámara antivandálica. Este tipo de cámaras son utilizadas en zonas públicas, puesto que las cámaras de seguridad son vulnerables a robos y a posibles agresiones y/o golpes, es por ello que estas cámaras son diseñadas con materiables de características durables y fuertes, para que resistan tanto la manipulación e impactos que puedas ocurrir, ademas estas cámaras pueden contar con un sello de goma, los cuales tienen el fin de evitar que le entre polvo o agua a dichas cámaras, por otro lado estas cámaras están montadas sobre una carcasa fija y resistente, por consiguiente podemos indicar que este tipo de cámaras son usadas mayormente en almacenes, parking, discotecas, vías públicas y generalmente en cualquier otro espacio pública o donde usualmente se encuentre, se reunan o transiten un grupo de personas.

Figura 2.9: Cámara antivandálica.

(OVACEN, 2023)

Cámara wifi o inalámbrica. Este tipo de cámaras son aquellas las cuales no se encuentran conectadas de forma directa a un cable de conexión, una de las características más particulares es que posee normalmente una batería, el cual hace funcionar y también transmite la captura de imágenes por medio del WiFi. Estas cámaras son una de las más utilizadas en los hogares, como por ejemplo, cuando las personas quieren ver como se encuentran los niños o los bebes; con el constante avance de la tecnología, se ha llegado a incorporar a dichas cámaras, un software el cual es el encargado de transmitir y monitorear de forma directa, esto es posible gracias al internet.

Figura 2.10: Cámara wifi o inalámbrica.

(OVACEN, 2023)

Cámara exterior. Este tipo de cámaras son similares y/o parecidas a las cámaras antivandálicas, puesto que están diseñados para poder proteger a dichas cámaras. A comparación de las cámaras antivandálicas que resisten golpes o daños físicos, estas cámaras pueden resistir las distintas acciones climatológicas que pueda suceder. Estas cámaras fueron diseñadas para el exterior de una propiedad o edificio; entre estos tenemos una casa, un hotel o una vía pública, ya que pueden resistir las acciones climatologicas que se lleguen a suscitar, estos eventos puede ser la lluvia, el viento, entre otros; y son utilizadas e instalados en espacios al aire libre. El precio de estas cámaras llegan a ser más caras, por su particular característica que es su carcasa de protección.

Figura 2.11: Cámara exterior.

(OVACEN, 2023)

Cámara todo en uno. Con el constante avance que se viene dando en la tecnología, se ha llegado a producir e implementar una gran variedad de tipos de cámaras de seguridad, lo cual hace dificil para las personas al momento de elegir o inclinarse por uno de estos tipos, actualmente se ha llegado a incorporar diferentes características en las cámaras, los cuales hace que una cámara tenga más funcionalidades que otras, como bien se detallo en los otros tipos de cámara, una cámara de seguridad puede tener todas esas características y más, pero mientras mejores características tenga como la calidad de video, la resolución, resitencia, entre otros; y que posea varias funcionalidades como reconocimiento facial, detección de movimiento, entre otros; mayor será el precio de estas cámaras.

Figura 2.12: Cámara todo en uno.

(OVACEN, 2023)

Las cámaras de seguridad, también se pueden clasificar según su forma, a continuación mencionaremos cuales puedes ser, y estas son las siguientes:

- Cámaras cube, las cuales son usadas para el interior de un área.
- Cámaras de caja, las cuales son usadas para el interior de un área.
- Cámaras bullet, las cuales son usadas para el exterior de un área.
- Cámaras domo, las cuales son usadas tanto para el interior, como para el exterior de un área.

Figura 2.13: Clasificación de cámaras según su forma.

(a) Forma cube

(b) Forma caja

(c) Forma bullet

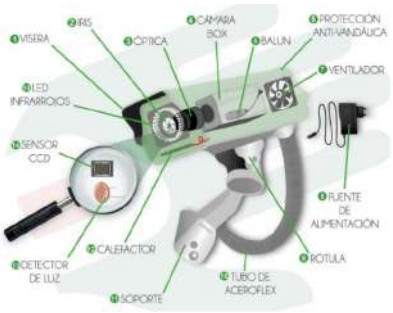
(d) Forma domo

Fuente: (OVACEN, 2023)

Partes de una cámara de seguridad

Las cámaras de seguridad están compuestas por varias partes o componente, las cuales les ofrecen mayor características o funciones a dichas cámaras. A continuación mencionare y describiré a detalle cuales son algunas de estas partes que componen una cámara video vigilancia; estas partes que se mencionarán, serán basadas a una cámara antivandálida exterior, es por ello que otras cámaras pueden tener estas partes u obviar alguna de ellas. Además ,explicaremos cual es la función que cumple cada uno de estos componentes dentro de la cámara.(OVACEN, 2023)

Figura 2.14: Partes de una cámara.



(OVACEN, 2023)

- **VISERA:** este componente ayuda de cierta manera a invertir y hasta anular el efecto de contraluz, tanto la luz solar como la luz lumínica intervienen en la captura de imágenes, haciendo que las imágenes salgan mal y con reflejo; es por ello que, con ayuda de este componente evitan una mala captura de imágenes.

- **IRIS:** este es un componente, el cual forma parte de otro componente más grande como es la óptica, cuya función tiene el de controlar la luz que entra o se emite en el sensor de la cámara. Dicho componente puede ser tanto manual como automático; en caso de ser automático, esto hace que se pueda adaptar a la luz de cualquier suceso o escenario posible, de la misma manera de como funciona el ojo humano, evitando de esta manera las imágenes blancas y/o completamente oscuras. Además, las nuevas cámaras de hoy en día, pueden contar con un sistema de compensación de contraluz BLC y WDR, el cual hace que se ilumine las zonas oscuras de la escena.

- **ÓPTICA:** este componente tiene una amplia variedad de tipos, el cual se adapta según las características que posea una cámara de seguridad, algunas de estas características son la apertura de iris, sensibilidad, sensor, zoom, distancia focal, entre otros; de esta forma se puede lograr optimizar el campo de visión que tiene una cámara de seguridad.

- **CÁMARA BOX:** este tipo de cámara de video vigilancia, como lo es una cámara antivandálica exterior, permite llegar a cambiar el tipo de lente, esto sucede según el ángulo de visión y zoom que necesita o requiere por parte del usuario; por otra parte, también podemos cambiar la cámara box por otras, las cuales tienen una mayor calidad o hasta tecnologías diferentes, esto permite que dicha cámara conserve el resto de componentes que posee.

- **BALUN:** este componente sirve para transformar las líneas de transmisión que son balanceadas a lineas que no lo son.

- **PROTECCIÓN ANTIVANDÁLICA:** este componente sirve para darle mayor protección a la cámara de vigilancia, de cualquier suceso inesperado por parte de la meteorología, estos sucesos pueden ser la nieve, lluvia, granizo y hasta de los daños que pueden ser ocacionados por otras personas, esto logra mantener fija dicha cámara mientras se realiza la captura de imágenes del lugar donde fue instalada. Este componente está usualmente hecha para tipos de cámaras que se usan para el exterior.

- **VENTILADOR:** este componente sirve para regular el calor dentro de las cámaras de seguridad, esto ayuda a no dañar los otros componentes de dicha cámara, impidiendo que no se calienten más de lo debido, esta función es similar a la de un ventilador de una PC.

- **FUENTE DE ALIMENTACIÓN:** este componente sirve para estabilizar la tensión protegiendo así todos los componentes que posee una cámara de seguridad, esto sucede cuando se da el caso de sobre alimentación o la tensión sube más de lo normal.

- **RÓTULA:** este componente permite a la cámara de seguridad brindarle una inclinación posible y/o necesaria para que dicha cámara pueda vigilar y capturar la toma de imágenes del espacio o lugar que se requiere visualizar y/o hasta proteger.

- **TUBO DE ACEROFLEX:** este componente brinda y ayuda de algunos daños a los cables de posee una cámara de seguridad, dicho componente ayuda a aislar y proteger los cables tanto del agua como de la humedad e impide que dichos cables puedan ser cortados o saboteados. Este componente tiene una característica peculiar ya que son herméticos, flexibles y muy resistentes.

- **SOPORTE:** este componente ayuda a que una cámara de seguridad pueda ser fijado firmemente en cualquier espacio y/o superficie como puede ser una pared, un techo, poste, entre otros.

- **CALEFACTOR:** este componente ayuda a que las cámaras de seguridad tengan siempre una buena nitidez en la toma y captura de imágenes; esto se da al momento que haya cambios de temperatura, haciendo que pueda empañarse el cristal de las cámaras; es por ello, que con este componente no sucederá estos problemas.
- **DETECTOR DE LUZ:** este componente ayuda detectar el nivel de luminosidad en las cámaras de seguridad, haciendo que cuando la luminosidad sea baja o hasta nula se activen los LED infrarrojos para que puedan ver de mejor manera.
- **SENSOR CCD:** este componente ayuda a que las cámaras de seguridad puedan transformar las señales luminosas en señales electrónicas, para que estas puedan ser transmitidas en formato digital o analógico, dependiendo que tipo de cámara sea la que se este usando.
- **LEDS INFRAROJOS:** este componente ayuda a que las cámaras de seguridad puedan tomar una mejor captura de imágenes en escenarios donde existe poca iluminación o hasta casi nula, permitiendo que dicha cámara grabe en la oscuridad pero en escala de grise; es decir, en blanco y negro. Estos LED's son encendidos de forma automática con ayuda del detector de luz que dicha cámara posea.

Cámaras con reconocimiento facial

La tecnología de reconocimiento facial es una herramienta la cual hoy en día, facilita y brinda una mayor sensación de seguridad y bienestar para las personas y/o familias. Esta tecnología es capaz de identificar a una o varias personas a través de un elemento audiovisual, cualquiera que se tenga; estos pueden ser imágenes, videos, entre otros.

Las cámaras que tienen incorporado esta función de reconocimiento facial, funcionan mediante un sistema de identificación el cual reconoce características biométricas de una persona; es decir, para que estas cámaras puedan verificar la identidad de una persona, se toman los rasgos particulares como las medidas de la cara y el mismo rostro de dicha persona.

Este tipo de cámaras logran recopilar toda la información biométrica de una persona, estos datos son llamados también perfiles faciales. Dichas cámaras realizan una captura de todas las expresiones faciales posibles del rostro, así como sus rasgos particulares; para ello debe tenerse una base de datos de rostros ya creadas, la cual servirá para identificar, verificar y/o autenticar a una o varias personas (Seguridad, 2022).

¿Cómo funcionan las cámaras con reconocimiento facial? Las cámaras que tienen incorporado la herramienta de reconocimiento facial, realizan la toma y captura de imágenes según las características que posea dicha cámara.

Con la imagen que se haya capturada, se realizará una comparación entre la base de datos, la cual ya se tiene implementada en la nube; este análisis se realizará en tiempo real por medio de un material audiovisual (imágen o video); de esta forma esta tecnología llega a ser mucho más fiable y segura que la información que se pueda adquirir de un imagen estática la cual puede ser manipulada por terceros.

Para poder hacer uso de esta tecnología es necesario acceder a una conexión de internet estable, ya que la base de datos estará alojada en los servidores y si la conexión no es estable fallará al momento de dar respuesta de dicho reconocimiento facial.

Al usar algún tipo de cámara con reconocimiento facial, se tiene que tener en cuenta que dicha cámara tomará capturas de todos los rasgos posibles de la persona, para que así mediante estas características el sistema de reconocimiento facial realice su respectiva comparación con una base de datos. A continuación, veremos una imagen en la cual podemos apreciar como se realiza su respectivo análisis de forma matemática, para que así se pueda verificar que los datos biométricos obtenidos por las cámaras de seguridad correspondan a la persona la cual se encuentre en dicha base de datos ya creada con anterioridad, dicha solicitud de acceso puede ser realizada mediante una aplicacion, un sistema o incluso para el ingreso de una vivienda o edificio.

Figura 2.15: Funcionamiento de una cámara con reconocimiento facial.

(Seguridad, 2022)

Utilidades de las cámaras con reconocimiento facial?

- **Control de acceso.** Brinda un acceso rápido y seguro a los empleados ya que la identificación suele tomar alrededor de pocos segundos. También para identificación de huéspedes y clientes en una hotelería, gimnasios u otras actividades.
- **Supervisión de asistencia.** Se utilizan a la entrada y salida del lugar de trabajo para verificar la asistencia laboral.
- **Reducción de delitos.** Gracias al reconocimiento facial podemos reconocer a una persona, la cual ha realizado un delito; y/o también podemos identificar a una persona que esta siendo buscada por algún delito en específico.
- **Prevención del COVID-19.** Los dispositivos detectan si las personas están utilizando la mascarilla. También pueden estar equipadas con un termómetro para verificar la temperatura corporal al ingreso al establecimiento.

- **Mejora de la experiencia cliente.** La cámara puede reconocer al cliente y recomendarle los productos en función de compras anteriores. El pago con reconocimiento facial elimina las colas frente a las cajas. En distintas industrias como viajes y hoteles se utiliza para personalizar los servicios a los clientes.
- **Publicidad.** Esta tecnología permite recopilar datos para juzgar cómo reaccionan las personas ante sus productos. En una etapa más avanzada, pueden activar anuncios dirigidos.

(Seguridad, 2022)

Análisis Costo-Beneficio de Cámaras de Seguridad

Habiendo culminado con el concepto y la descripción de los diversos tipos de cámaras de seguridad que existen hoy por hoy, realizaremos una comparación de costos y beneficios entre algunos de estos, para que así uno pueda saber cual es el precio de estas cámaras de seguridad y cuales son los beneficios que te trae.

Tabla 2.2: Costo y Beneficio de cámaras de seguridad

Tipo de cámara	**Costo aprox. (S/.)**	**Beneficios**	**Limitaciones**
Cámara Analógica	185 - 370	Económica y fácil de instalar; adecuada para áreas pequeñas con necesidades básicas de vigilancia.	Calidad de imagen limitada y funciones básicas.
Cámara IP (Digital)	370 - 1110	Ofrece alta calidad de imagen, acceso remoto, y funciones avanzadas como la detección de movimiento.	Requiere mayor ancho de banda y una configuración de red más compleja.
Cámara Térmica	5550 - 18500	Capaz de detectar movimiento en condiciones de poca luz o visibilidad nula, ideal para seguridad avanzada.	Precio elevado y resolución limitada para detalles específicos.
Cámara PTZ (Pan-Tilt-Zoom)	1850 - 7400	Proporciona una cobertura amplia con capacidad de control remoto para ajustar la posición y el zoom.	Más costosa y necesita un monitoreo activo para aprovechar todas sus funciones.
Cámara de Domo	295 - 740	Diseño discreto y resistente al vandalismo, ideal para áreas públicas.	Cobertura limitada comparada con cámaras PTZ.
Cámara de Bala	370 - 925	Adecuada para exteriores, ofrece un largo alcance de visión.	Más visible y propensa al vandalismo.

Fuente: (Amazon, 2024)

2.2.6. Condominio

La palabra condominio proviene del vocablo latino çondominium", el cual hace referencia a la potestad que se tiene de una propiedad entre dos o más personas. También es llamado como un bien compartido, de la cual las personas encargadas de dicha propiedad pueden o no tener un vínculo de sangre. (Pérez Porto and María, 2022)

Un condominio puede clasificarse en dos tipos. El primero, es llamado condominio ordinario, donde dos o más personas tienen una participación en la propiedad; es decir, que cada una posee una parte del condominio y, por ende, son los propietarios. El segundo tipo, es el condominio en mano común; que es lo opuesto al primer tipo (condominio ordinario). En este último caso, aunque varias personas viven o residen en el condominio, no necesariamente son los propietarios; es decir, simplemente lo habitan.

El condominio o llamado también conjunto habitacional, se puede regular según y mediante las decisiones que tomen los propietarios de dicho condominio, para así poder efectuar relaciones que sean un tipo de mancomunidad como también de solidaridad. Al hablar de condominio podemos hablar de algunas palabras más, como son:

- Propiedad compartida, es el área donde todas las personas que residen en dicho condominio pueden compartirlo, estas áreas son los: jardines, piscinas, estacionamiento, etc.
- Propiedad individual, son las viviendas donde reside cada persona o familia instalada en dicho condominio.
- Reglas y regulaciones, las cuales deben existir en todo condominio para que pueda prevalecer la armonia y la seguridad entre todos los que residen en dicho condominio.
- Mantenimiento, esto va referido a todas las áeas comunes o compartidas que existen en el condominio.
- Asociación de propietarios, las personas las cuales son los dueños de dicha propiedad, los cuales toman las decisiones con respecto al condominio.

Para que un lugar sea llamado un condominio, este tiene que tener la siguiente estructura o al menos una estructura similar, estos son:

- Cada residente o familia que reside en un lugar, debe de tener una propiedad propia, en la cual puedan habitar como támbien pueden ser un espacio comercial.
- Dicha propiedad debe de tener o constar de áreas comunes, las cuales sean compartidas por todos los residente de dicha propiedad, estos pueden ser áreas verdes, jardines, estacionamientos, etc
- Debe de existir una junta la cual regule y tome decisiones con respecto a dicha propiedad, estas son normal y reglas que todos los residentes deben de acatar.

Capítulo 3

Desarrollo del tema de tesis

3.1. Creación del sistema

Como se indico con anterioridad en el capítulo 1; haremos uso de la metodología SCRUM para el desarrollo e implementación del sistema interno propuesto. Esta metodología SCRUM se caracteriza por gestionar los trabajos por sprints, asignando un determinado número de tareas (backlog) para cada sprint; para esto cada sprint debe tener un tiempo de culminación y entrega del producto de dicho sprint; es decir, se hace un entregable de lo que se realizó en dicho sprint; de esta forma al final de cada sprint tendremos algo funcional, lo cual se ira incrementando con el paso del tiempo al finalizar cada sprint.

Por otro lado, los últimos sprints (10, 11 y 12) están más relacionados al desarrollo iterativo e incremental, esto se da porque estos últimos sprint se basan en modelos y algoritmos de IA, lo cuales necesitan de varios ciclos hasta encontrar el resultado más favorable para cada uno de estos modelos y/o algoritmos.

3.1.1. Historias de Usuario por Sprint

Historia de Usuario 1 - Sprint 1

- **Rol**: Desarrollador.
- **Objetivo**: Como desarrollador, quiero tener una base de datos implementada para almacenar la información esencial del sistema.
- **Beneficio**: Facilitar la organización y el almacenamiento de los datos esenciales, asegurando una estructura adecuada para el desarrollo del sistema.
- **Backlog del Sprint**:
  - Análisis de la estructura de la base de datos.
  - Implementación de tablas y relaciones necesarias.
- **Duración**: 2 semanas.

Historia de Usuario 2 - Sprint 2

- **Rol**: Administrador.
- **Objetivo**: Como administrador, quiero gestionar los roles dentro del sistema, permitiendo controlar los accesos y permisos.
- **Beneficio**: Asignar permisos adecuados a los usuarios del sistema, donde los administradores puedan crear, editar y eliminar roles, mientras que el personal de seguridad solo pueda visualizar los datos.
- **Backlog del Sprint**:
  - Creación de interfaces para el CRUD de roles: crear, listar, editar y eliminar.
  - Implementación de consultas y validaciones para roles.
  - Integración de las consultas en las interfaces.
  - Configuración de permisos para los roles.
- **Duración**: 3 semanas.

Historia de Usuario 3 - Sprint 3

- **Rol**: Administrador.
- **Objetivo**: Como administrador, quiero gestionar los usuarios del sistema para asegurar un control adecuado de los accesos.
- **Beneficio**: Facilitar la creación y administración de usuarios con sus respectivos permisos, garantizando el correcto funcionamiento del sistema de seguridad.
- **Backlog del Sprint**:
  - Creación de interfaces para el CRUD de usuarios: crear, listar, editar y eliminar.
  - Implementación de consultas para la gestión de usuarios.
  - Integración de las consultas y validaciones.
  - Configuración de roles para usuarios.
- **Duración**: 3 semanas.

Historia de Usuario 4 - Sprint 4

- **Rol**: Administrador y seguridad.
- **Objetivo**: Como administrador y seguridad, quiero gestionar los tipos de documentos en el sistema para garantizar una organización clara.
- **Beneficio**: Permitir la creación, activación/desactivación y edición de tipos de documentos, facilitando su control en el sistema.

- **Backlog del Sprint**:
  - Creación de interfaces para el CRUD de tipos de documentos: crear, listar, activar/desactivar, editar y eliminar.
  - Implementación e integración de consultas.
  - Configuración de permisos de acceso según roles.
- **Duración**: 3 semanas.

Historia de Usuario 5 - Sprint 5

- **Rol**: Administrador y seguridad.
- **Objetivo**: Como administrador, quiero gestionar la información de los pisos de los edificios para mejorar la administración de los residentes; como seguridad, quiero ver el listado de los pisos existentes.
- **Beneficio**: Facilitar el control y la organización de la información de los pisos, lo que permite una administración eficiente del sistema.
- **Backlog del Sprint**:
  - Creación de interfaces para el CRUD de pisos: crear, listar, editar y eliminar.
  - Implementación de consultas y validaciones para pisos.
  - Configuración de permisos de acceso según roles.
- **Duración**: 3 semanas.

Historia de Usuario 6 - Sprint 6

- **Rol**: Administrador y seguridad.
- **Objetivo**: Como administrador, quiero gestionar los apartamentos para asignar residentes y mantener una estructura organizada; como seguridad, quiero ver el listado de los apartamentos existentes.
- **Beneficio**: Garantizar una administración eficiente de los apartamentos dentro del sistema, mejorando el control de los residentes.
- **Backlog del Sprint**:
  - Creación de interfaces para el CRUD de apartamentos: crear, listar, editar y eliminar.
  - Implementación e integración de consultas.
  - Configuración de permisos según roles.
- **Duración**: 3 semanas.

Historia de Usuario 7 - Sprint 7

- **Rol**: Administrador y seguridad.
- **Objetivo**: Como administrador, quiero gestionar la información de los residentes para asegurar que los datos estén correctamente organizados y accesibles; como seguridad, quiero ver el listado de los residentes existentes.
- **Beneficio**: Facilitar el acceso y control de la información de los residentes, mejorando la seguridad y organización del sistema.
- **Backlog del Sprint**:
  - Creación de interfaces para el CRUD de residentes: crear, listar, editar y eliminar.
  - Implementación de consultas para la gestión de residentes.
  - Configuración de permisos de acceso según roles.
- **Duración**: 3 semanas.

Historia de Usuario 8 - Sprint 8

- **Rol**: Administrador y seguridad.
- **Objetivo**: Como administrador y seguridad, quiero configurar la barra de menú y las rutas para facilitar la navegación dentro del sistema.
- **Beneficio**: Proporcionar una navegación clara y sencilla dentro del sistema, asegurando que los usuarios puedan acceder fácilmente a las diferentes funcionalidades.
- **Backlog del Sprint**:
  - Integración y configuración de la barra de menú.
  - Configuración de vistas y rutas.
  - Pruebas de navegación para roles de administrador y seguridad.
- **Duración**: 2 semanas.

Historia de Usuario 9 - Sprint 9

- **Rol**: Administrador y seguridad.
- **Objetivo**: Como administrador y seguridad, quiero implementar una interfaz de login y proteger las rutas mediante middleware para asegurar que solo usuarios autorizados puedan acceder al sistema.
- **Beneficio**: Garantizar que solo usuarios autorizados puedan acceder al sistema, con roles claramente definidos para administrador y seguridad.
- **Backlog del Sprint**:
  - Implementación de la interfaz de login.
  - Integración de consultas de autenticación.

- Configuración de middleware para la seguridad de rutas.

- **Duración**: 3 semanas.

Historia de Usuario 10 - Sprint 10

- **Rol**: Administrador y seguridad.
- **Objetivo**: Como administrador y seguridad, quiero integrar algoritmos de detección de rostros al sistema para mejorar la seguridad en la detección de personas.
- **Beneficio**: Permitir la detección eficiente de rostros utilizando algoritmos de detección, incrementando la seguridad en el sistema.
- **Backlog del Sprint**:
  - Desarrollo de código para integrar algoritmos de detección de rostros.
  - Comparación de algoritmos para seleccionar el más eficiente.
- **Duración**: 1 mes.

Historia de Usuario 11 - Sprint 11

- **Rol**: Administrador y seguridad.
- **Objetivo**: Como administrador y seguridad, quiero implementar algoritmos de reconocimiento facial para identificar residentes y detectar intrusos.
- **Beneficio**: Incrementar la seguridad en el sistema mediante un reconocimiento facial preciso y eficiente.
- **Backlog del Sprint**:
  - Desarrollo de código para integrar algoritmos de reconocimiento facial.
  - Pruebas de rendimiento de los algoritmos.
- **Duración**: 1 mes.

Historia de Usuario 12 - Sprint 12

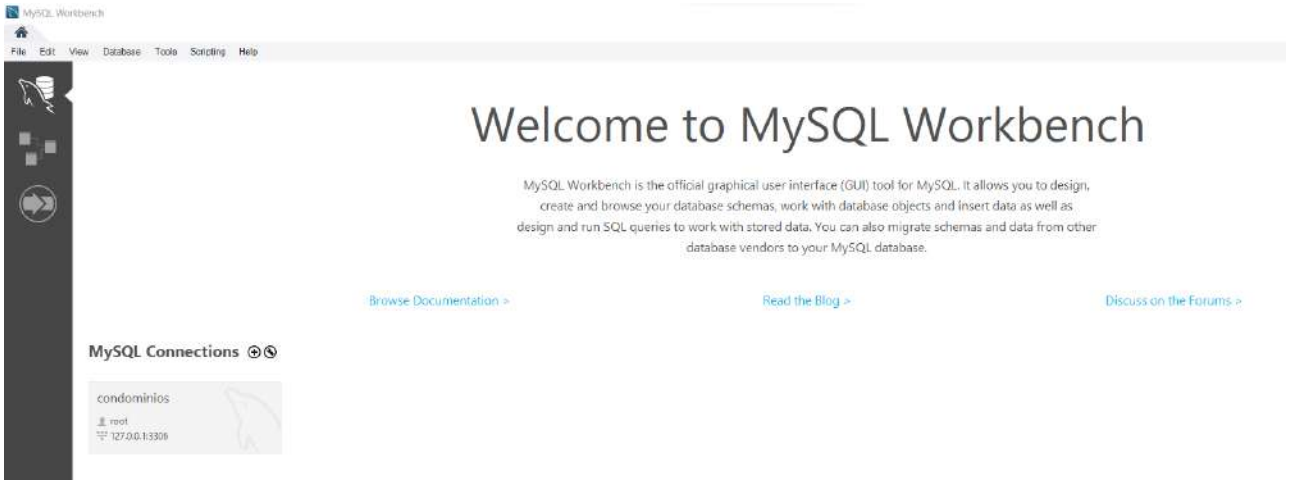
- **Objetivo**: Integrar cámaras convencionales al sistema interno.
- **Beneficio**: Expandir la funcionalidad del sistema, permitiendo el uso de cámaras convencionales para la detección de intrusos.
- **Backlog del Sprint**:
  - Integración de cámaras convencionales al sistema.
  - Pruebas de funcionamiento con distintos tipos de cámaras.
  - Configuración de compatibilidad para cámaras
- **Duración**: 1 mes.

3.1.2. Creación de la base de datos

Para poder desarrollar e implementar el sistema interno propuesto, primero se tiene que diseñar una BD capaz de soportarlo, para ello trabajaremos con la herramienta de Workbench que es un sistema integrado el cual trabaja con base de datos MySQL.

En esta herramienta crearemos una conexion local, a la cual nombraremos con el nombre de condominios, como vemos en la siguiente imagen 3.1, dicha conexión es creada automáticamente al momento de instalar el Workbench Mysql, cuyo software unicamente te pide como datos el nombre de la conexión y su contraseña en caso así se desee.

Figura 3.1: Creación de una conexión local.



Una vez instalado el software de Workbench MySQL, se realizó un análisis exhaustivo de las necesidades fundamentales del sistema, centrándose en la identificación de las tablas esenciales requeridas para establecer un PMV. Este enfoque se adoptó para garantizar la eficiencia en el desarrollo y la implementación del sistema, minimizando el tiempo y los recursos necesarios para la construcción de la base de datos.

Dicho proceso involucró la identificación de las entidades principales y sus relaciones, así como la definición de los atributos esenciales de cada tabla. Posteriormente, se llevó a cabo una revisión detallada para garantizar la coherencia y la integridad de la estructura de la base de datos, asegurando que cumpliera con los requisitos funcionales y operativos del sistema.

Dentro de dicha conexión podemos crear una BD, la cuál será la encargada del sistema interno a implementar; en dicha BD crearemos las tablas las cuales se identificaron al realizar el análisis exhaustivo para que así se tengan las funcionalidades básicas que va poseer nuestro sistema interno en cuestión; para ello enumeraremos las tablas las cuales necesitemos crear, así como sus respectivos atributos. Algunas de estas tablas son las siguientes:

- pisos
- apartamentos
- tipo de documentos

- residentes
- usuarios
- sesiones
- roles
- permisos
- migraciones

Además de la herramienta de Workbench MySQL, trabajaremos con el Framework Laravel, se usará la versión 10, para esto necesitamos tener instalado lo siguiente:

- Node v8.1 o superior.
- Composer v2.2.0 o superior.

Hoy en día existen diversos lenguajes de programación así como frameworks, los cuales podemos escoger al momento de crear e implementar un sistema interno como el que vamos a desarrollar en el presente proyecto de investigación.

Es por ello que escogimos realizar dicho sistema en php, en Laravel para ser más exactos, esto por el gran crecimiento que ha tenido últimamente y por su comunidad activa, el cuál brinda soporte constante, así como nuevas versiones anuales de esta.

Laravel posee una sintaxis expresiva y elegante puesto que, es un marco de aplicación web. Laravel tiene librerias de plantillas ya hechas, las cuales podemos reutilizar y configurar de acuerdo al uso que daremos al sistema interno a implementar. (Inc., 2024)

Así mismo, agregaremos un paquete a Laravel, el cual es AdminLTE versión 3; dicho paquete nos ayudará a usar una plantilla tanto para la vista del login como para la barra lateral del menú, gracias a este paquete podemos configurar y/o personalizar estas 2 vistas, las cuales son primordiales al momento de implementar el sistema interno en cuestión. Este paquete no tiene ninguna restricción o requisito previo para poder instalarse, ya que es completamente compatible con el framework de Laravel.

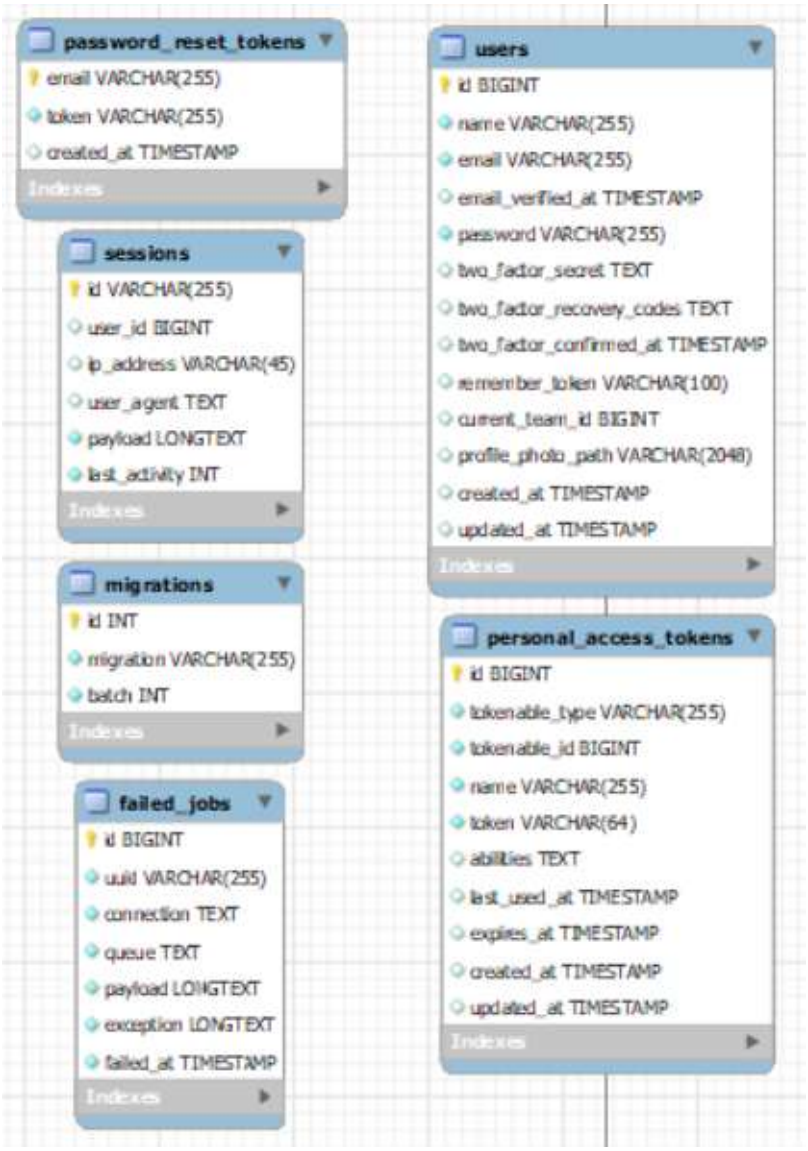
Podemos crear las tablas usando un script propio de mysql o también implementarlo desde el mismo Laravel, esto gracias a las migraciones que posee dicho framework; en este caso implementaremos la BD del sistema en migraciones, crearemos un archivo de tipo migración para cada tabla que necesitemos crear. Una vez terminado la implementación de las migraciones necesitaremos configurar el archivo ".env", que se encuentra en la raíz del proyecto creado, en dicho archivo necesitamos configurar la conexión con la BD creada o por crear, luego de ello ejecutaremos un comando el cuál será capaz de crear la BD así como sus tablas implementadas en los archivos de migraciones.

php artisan migrate:refresh –seed

Por otra parte, existen 2 paquetes de laravel, los cuales nos ayudarán en el proceso de la creación y diseño de la base de datos; puesto que, estos paquetes son primordiales y requeridos cuando hablamos de un sistema interno.

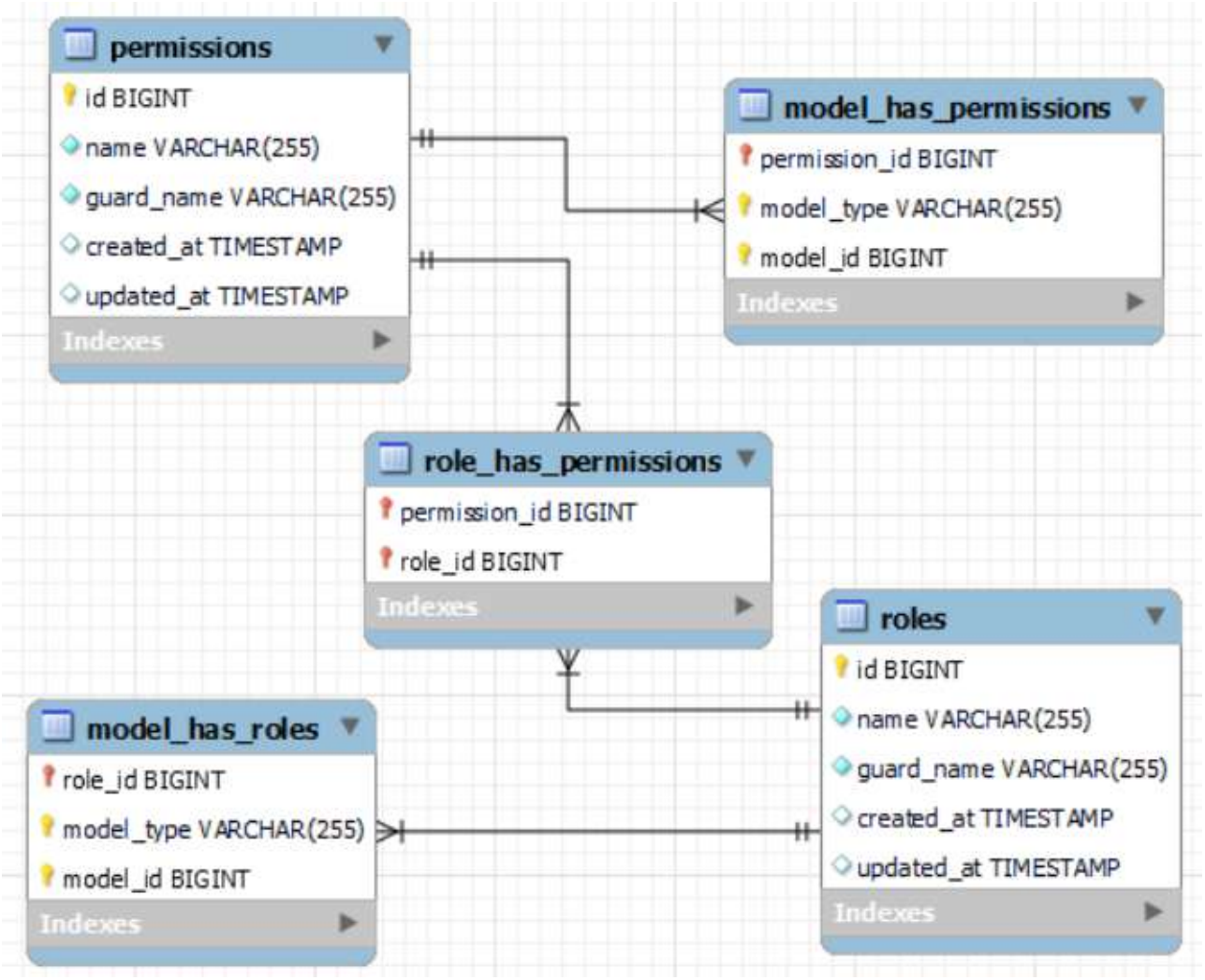
Como primer paquete tenemos a "laravel jetstream", que nos ayuda a identificar a un usuario que desee ingresar al sistema interno en cuestión; es decir, se encarga del login y las tablas que serán creadas gracias a este paquete, son las siguientes como se puede apreciar en la siguiente imagen 3.2.

Figura 3.2: Tablas del paquete Laravel Jetstream.



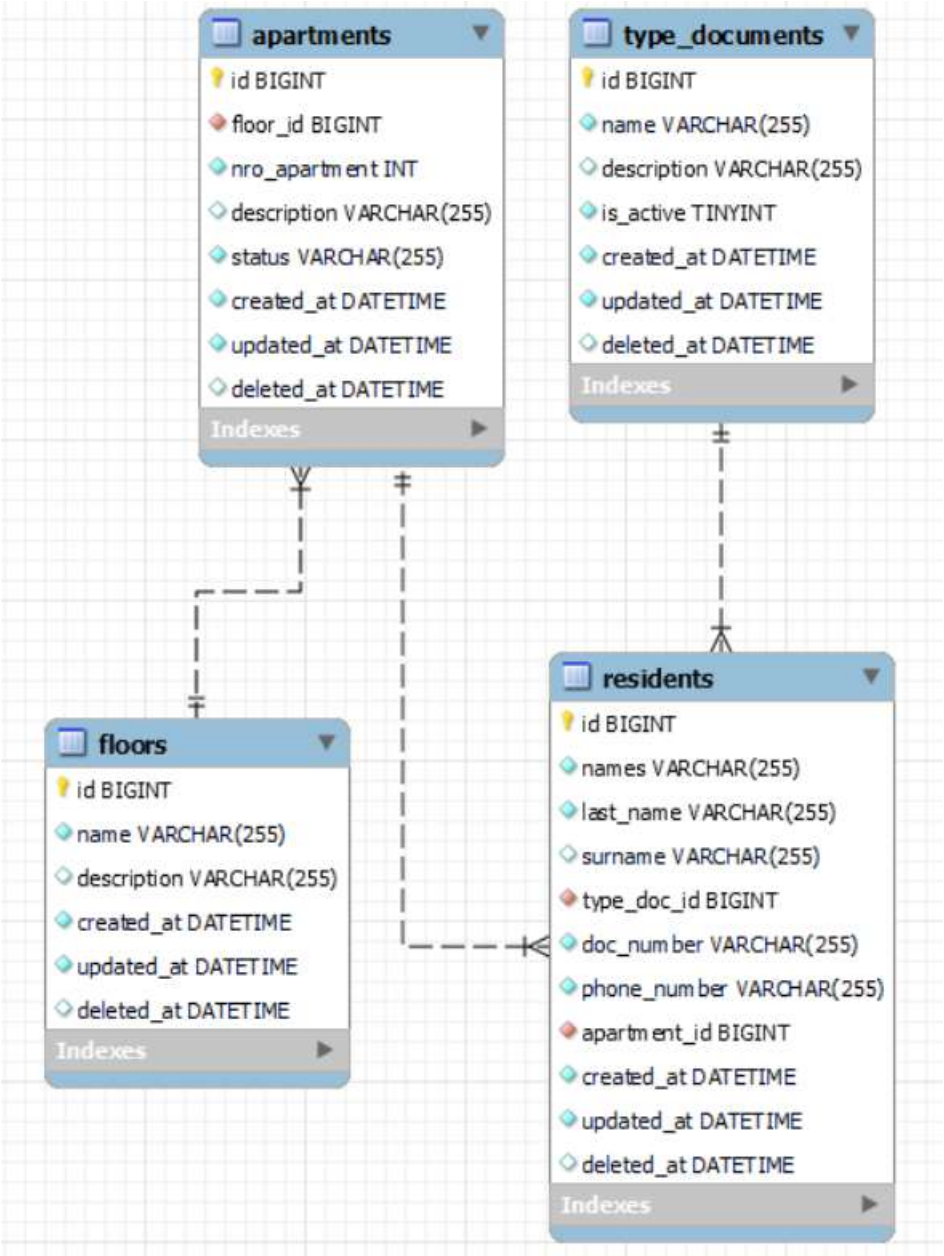
Como segundo paquete tenemos a "laravel permission", que nos ayuda a controlar los roles y permisos que tendrá un usuario dentro del sistema interno. De la misma forma que el paquete anterior, este paquete creara sus propias tablas para poder usarlas al momento de hacer consultas; gracias a ello tendremos las siguientes tabla como se muestra en la siguiente imagen 3.3.

Figura 3.3: Tablas del paquete de Laravel Permission.



Por último y no menos importante, tendremos las tablas que creamos mediante los comandos de migraciones que nos ofrece Laravel, estás tablas fuero creadas luego de un análisis y fueron requeridos para poder tener un PMV en el sistema interno propuesto.

Figura 3.4: Tablas creadas mediante migraciones.



3.1.3. Implementación del sistema interno

Ahora continuaremos con la implementación del sistema interno propuesto; ya teniendo las tablas requeridas creadas, tenemos que crear los CRUD para estos, parte de los CRUD que implementaremos serán las funciones básicas que nos ayudarán en las funcionalidades del sistema, estas funcionalidades básicas son las siguientes:

- crear un nuevo registro (modal)
- editar un nuevo registro (modal)
- eliminar un registro (botón)
- listar los registros creados o activos (vista)

Para poder crear estas funcionalidades básicas, necesitaremos crear endpoints, los cuales serán consultas a la BD, de una o varias tablas según los datos que necesitemos; para realizar dichas consultas, necesitaremos crear modelos sobre cada tabla de la cual haremos uso y dentro de ellos debemos de definir los atributos que posee dicha tabla, ya que con estos modelos creados podremos realizar las consultas haciendo uso de ORM Eloquent, donde simularemos y crearemos las consultas que necesitemos.

Las consultas que implementaremos estaran ubicadas en una carpeta específica donde se almacenan los archivos de tipo controlador (controladores), estos archivos pueden crearse vacios para así personalizar dichas funciones o también crearlas por defecto con funciones como: index, create, store, show, edit, update, destroy. En estas funciones podemos realizar las consultas a la BD con ORM Eloquent, como hemos mencionado con anterioridad o también redireccionar a alguna vista que tengamos implementadas. Para que estas funciones del controlador funcionen correctamente, necesitamos definir las rutas de cada función que se encuentre en dicho controlador, estas rutas estarán ubicadas en el archivo /routes/web.php".

Continuaremos con la implementación de las vistas y modals que se requieran en dicho sistema interno propuesto. Como se mencionó con anterioridad, tendremos una vista para cada caso donde se obtenga un listado de datos; es así que, implementaremos el listado con registros de una determinada tabla y/o clase (conjunto de tablas) según corresponda. Además, implementaremos un buscador según la información que se visualice en la tabla (interfaz), así como la paginación de estos registros. En dicha interfaz, encontraremos 3 botones básicos: un botón que tendrá como función el de crear un nuevo registro, otro botón que funcionará para editar un registro, y por último un botón el cual elimine un registro; cabe resaltar que 2 de estos botones (crear y editar) abrirán un modal donde se deberá llenar los campos antes de concluir con la consulta y/o acción que este designado a de dicho botón; por el contrario el botón de eliminar realizará de forma directa la consulta para luego refrescar la vista.

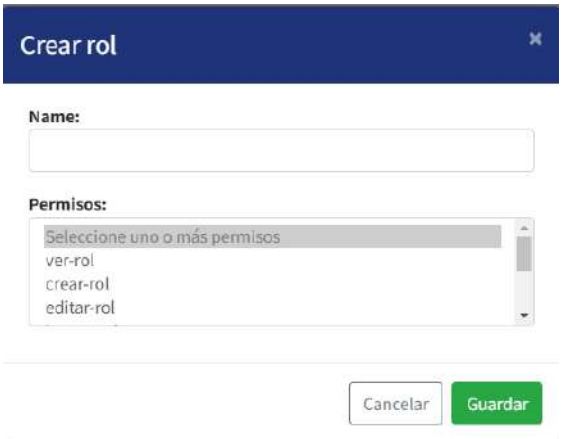
Habiendo mencionado algunas acciones en el listado de registros, continuaremos hablando sobre las vistas que habrán en nuestro sistema, así como los modals que existirán en cada una de ellas. Como primera sección que se tendrá en la barra de menú lateral y una de las más importantes en cualquier sistema son los roles, dicha sección constará de su vista en donde se tenga el listado de todos los roles que puedan existir dentro de un sistema, en nuestro caso dentro de un condominio y/o edificio de uso residencial, existirán solamente 2

roles: el rol de admin, el cuál pueda acceder a todas las vistas así como tambien pueda agregar, editar y/o eliminar cualquier tipo de dato; como segundo rol se tendrá el de seguridad el cuál solo podrá acceder a todas las vistas más no podrá crear, editar y/o eliminar algún tipo de dato, esto se da por el simple hecho que, debe haber jerarquías en todo sistema y que no toda persona puede tener las mismas acciones, funcionalidades o hasta el mismo acceso. Es por todo ello que necesitamos configurar y administrar los roles que van a existir dentro de un sistema cualquiera, en donde exista un nivel jerárquico entre sus usuarios.

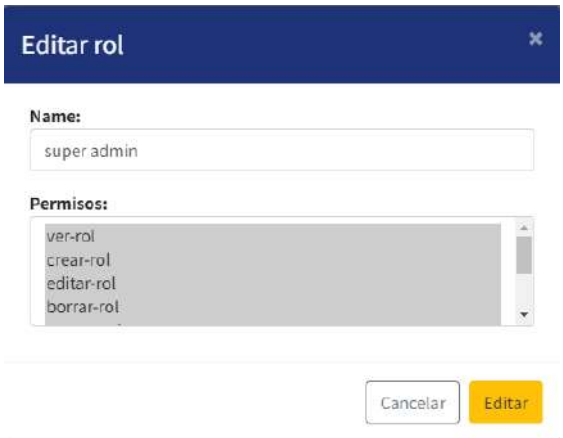
Como se puede apreciar en la siguiente imagen 3.5, los permisos serán cada acción posible que se pueda realizar dentro del sistema, como también el simple hecho de poder ver una sección, que en este caso conlleva a visualizar el listado de registros de un modelo en específico. Todos estos permisos son seleccionados para englobarlos y así poder crear un rol, de acuerdo a los permisos seleccionados, se le dará el acceso correspondiente en el sistema interno propuesto. Para que una usuario pueda crear, editar y/o eliminar un registro, es necesario tener el permiso de visualizar el listado ya que, cuando se tenga dicho permiso se visualizará como un ítem en la barra de menú lateral, caso contrario no se mostrará, lo que conlleva a que no se pueda acceder a dicha interfaz y a ninguna de sus acciones dentro de ella, aun teniendo el permiso correspondiente de hacerlo.

Figura 3.5: CRUD sobre roles.

(a) Listar los roles



(b) Crear un nuevo rol



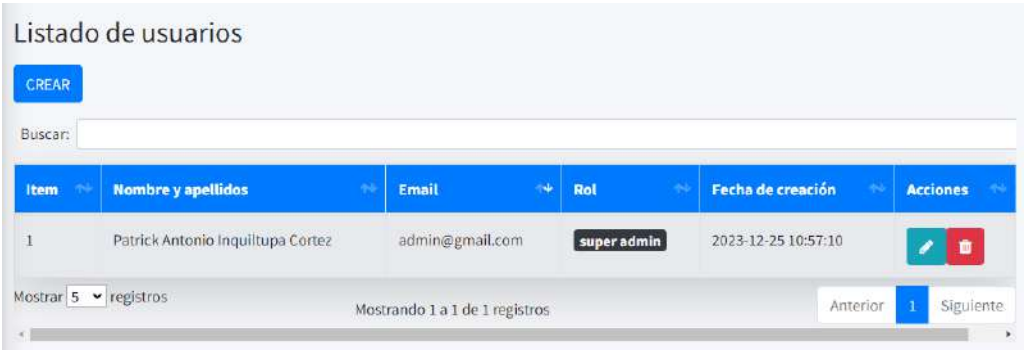
(c) Editar un rol

Otro de los CRUD a implementar es el de los usuarios, en esta vista se tendrá el listados de todos los usuarios que sean capaz de poder ingresar al sistema interno a implementar, de igual manera se podrá manipular estos datos, teniendo las acciones de agregar un nuevo usuario, editar un usuario ya creado, y el de poder eliminar un usuario existente. Cada usuario creado tendrá un rol en específico, esto para tener una jerarquía dentro del sistema y entre los usuarios, por ejemplo la persona propietaria del condominio y/o edificio tendrá el rango más alto, por lo tanto tendrá el rol de admin mientras que el personal de seguridad estará en un nivel menor a comparación del anterior, con un rol menor también como es el caso del rol de seguridad.

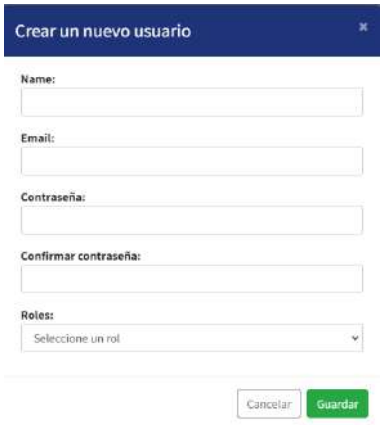
Al momento de crear un usuario se tendrá los siguientes campos en el formulario que existirá dentro del modal:

- nombre(s) y apellidos
- email
- contraseña
- repetir contraseña
- rol(es)

Figura 3.6: CRUD sobre usuarios.



(a) Listar los usuarios



(b) Crear un nuevo usuario

(c) Editar un usuario

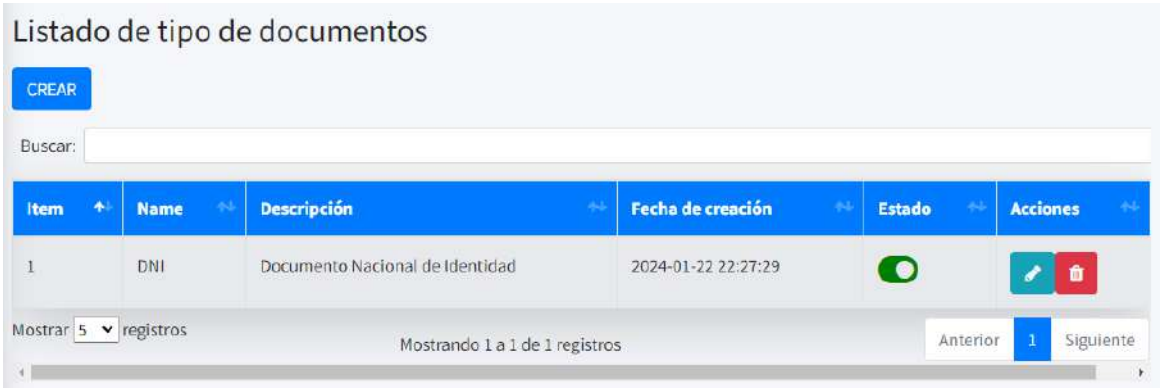
En el caso de editar un usuario, pudimos apreciar en la imagen anterior que, solo se podrá modificar tanto el nombre(s) y apellidos así como su rol(es) asignados, puesto que, la modificación del email y contraseña formarán parte del flujo de cambio de contraseña.

Como siguiente ítem de la barra de menú lateral, tendremos la interfaz en donde encontraremos el listado de los tipos de documento que pueda tener o poseer una persona, en este caso un residente puede identificarse con DNI en caso de ser peruano, pero en caso de que el residente sea extranjero no contará con dicho DNI, por el contrario contará con el carnet de extranjerá; es por ello que, vi por conveniente tener un ítem donde administremos todos los tipos de documentos con los cuales podamos identificar a un residente, existen varios tipos de documentos diferentes a los 2 ya mencionados con anterioridad.

Por otro lado, dicho listado constará de 4 botones y/o acciones como podemos apreciar en la siguiente imagen 3.7, entre estos botones tendremos las funciones de crear un tipo de documento, editar un tipo de documento, eliminar un tipo de documento y el de habilitar o deshabilitar un tipo de documento; estas 2 últimas acciones de habilitar/deshabilitar y eliminar son acciones de un solo click y realizarán su función respectiva, mientras tanto el de crear y editar abrirán un modal con los siguientes datos en el formulario:

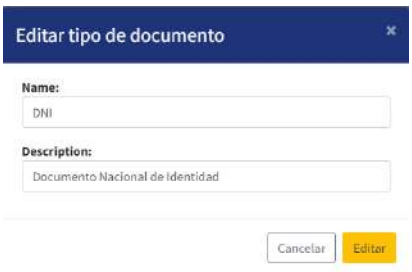
- nombre o abreviatura
- descripción
- estado (solo en el modal de crear)

Figura 3.7: CRUD sobre tipo de documentos.



(a) Listar los tipos de documentos

(b) Crear un nuevo tipo de documento



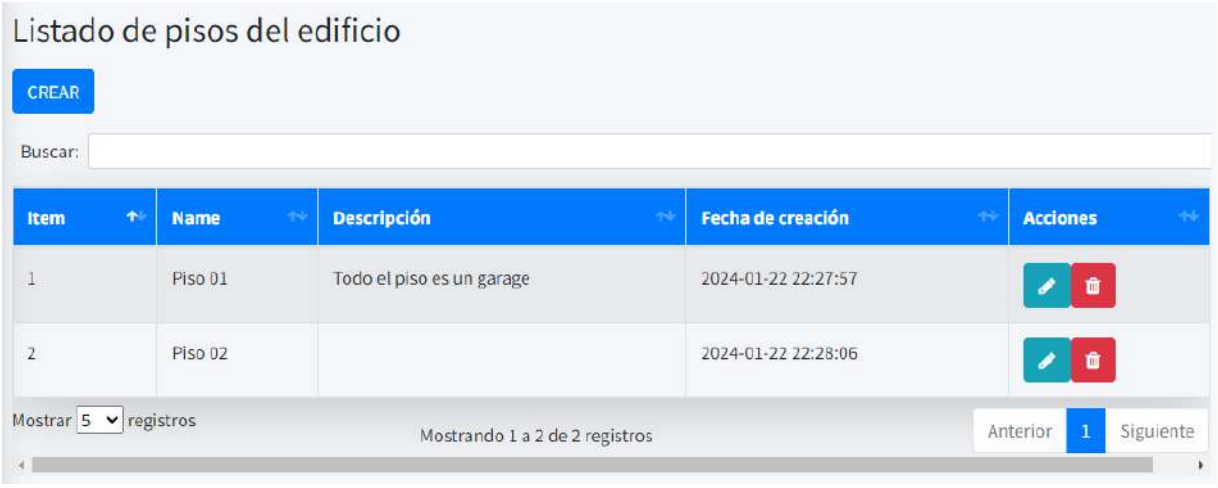
(c) Editar un tipo de documento

Como siguiente ítem, tendremos la interfaz de pisos, aquí tendremos el listado de los pisos que tenga un condominio y/o edificio de uso residencial; dicho ítem nos ayudará a administrar los pisos existentes, así como también para saber que pisos son usados como viviendas y cuales no.

Del mismo modo que las vistas anteriores, dicha vista poseerá de 3 botones los cuales son: crear un piso, editar un piso y eliminar un piso. El botón de eliminar tendrá la acción de click y ejecución como tal, mientras tanto los botones de crear y editar un piso abrirán un modal el cuál tendrá los siguientes campos dentro del formulario a completar:

- nombre
- descripción

Figura 3.8: CRUD sobre pisos.



(a) Listar los pisos

(b) Crear un nuevo piso

(c) Editar un piso

Como siguiente ítem en la barra de menu lateral, tenemos la vista de los apartamentos, esta interfaz será importante ya que con ella podremos administrar y contabilizar cuantos departamentos se tiene dentro de un condominio y/o edificio, también podremos clasificarlos o dividir los apartamentos, según el piso en el que se encuentren, además podremos definir que apartamentos o ambientes pueden ser habitadas y cuales no, ya que podrían ser ambientes destinados a depósitos, almacén, entre otros.

Por otro lado, también podremos definir el estado en el cuál se encuentra un apartamente, entre las opciones de estado tenemos los siguientes:

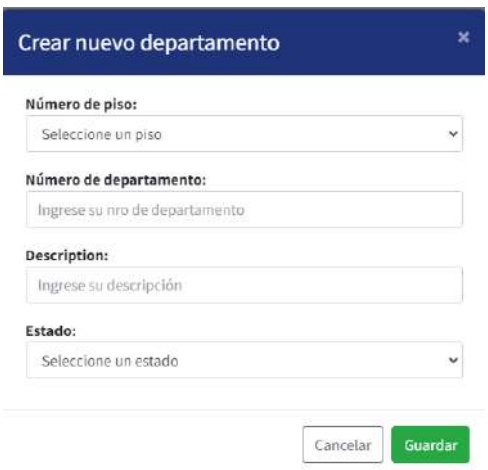
- disponible
- ocupado (compra del apartamento)
- alquilado
- en mantenimiento

La vista de apartamentos tendrán las mimas acciones y botones que se menciono con anterioridad, dentro de los modal se tendrá los siguientes campos dentro del formulario a completar:

- número o nombre de piso
- número de apartamento
- descripción
- estado

Figura 3.9: CRUD sobre apartamentos.

(a) Listar los apartamentos



(b) Crear un nuevo apartamento

(c) Editar un apartamento

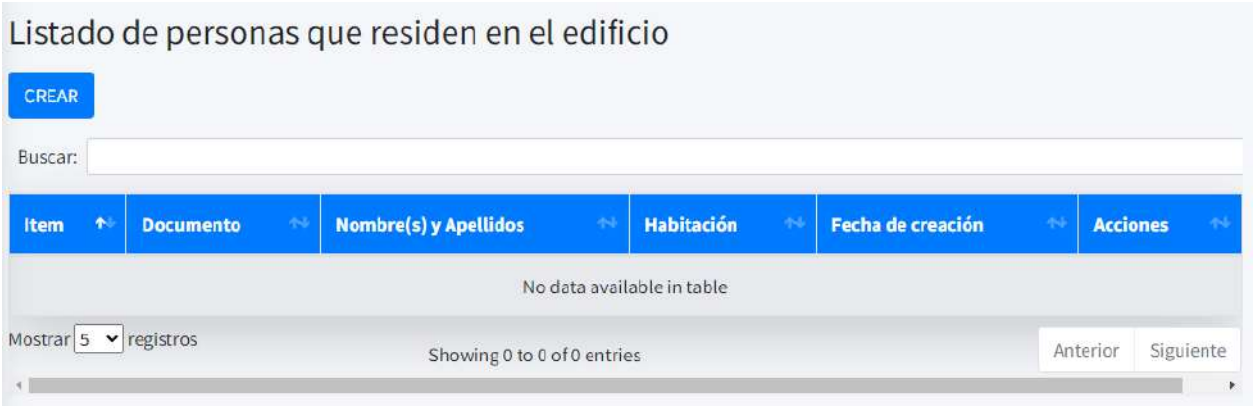
Otro ítem de la barra lateral de menú, es la vista de los residentes, donde podremos visualizar el listado de las personas, las cuales residen en dicho condominio y/o edificio de uso residencial. Dicha administración de residentes nos ayudará al momento de realizar el control de ingreso de personas al recinto; así como también, diferenciar a las distintas personas, residentes del edificio con otras personas ajenas y/o intrusos.

De igual forma que las vistas anteriores, esta interfaz constará de un botón el cuál elimine un residente, y de otros 2 botones que servirán para agregar y editar un residente, estos botones abrirán un modal en el cuál se encontrarán los siguientes campos del formulario a completar:

- tipo de documento
- número de documento
- nombre(s)
- apellido paterno
- apellido materno
- número de celular
- número de piso
- número de apartamento

Cabe reiterar que este no es el formulario finalizado para el modal con la acción de crear un nuevo residente, esto se debe a que falta un campo muy importante, que es el de detectar y almacenar rostros de una persona; este campo y funcionalidad adicional del formulario se vera posteriormente, al momento de seleccionar un algoritmo de detección de rostros.

Figura 3.10: CRUD sobre pisos.



(a) Listar los residentes

(b) Crear un nuevo residente

(c) Editar un residente

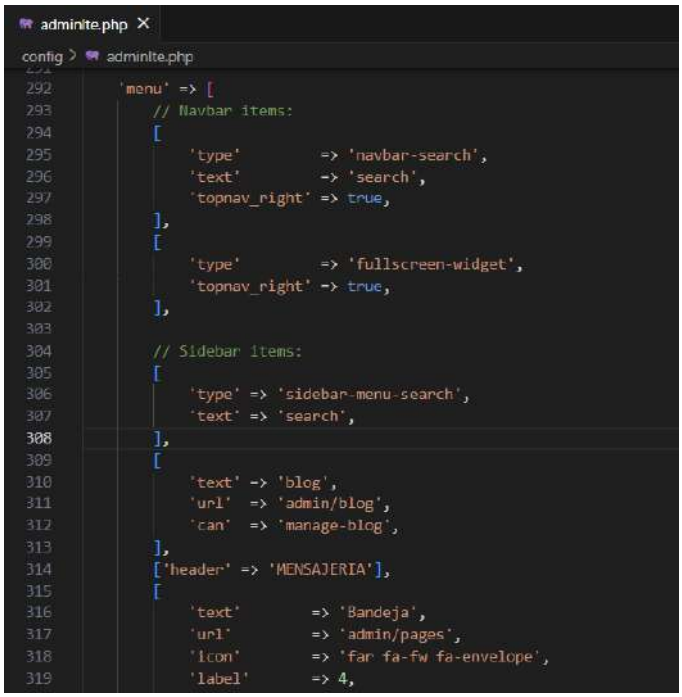
Al concluir con la implementación de las interfaces que existiran en nuestro sistema interno propuesto (vistas y controladores), necesitaremos la creación de un sidebar o barra de menú lateral, el cual pueda implementarse para poder interactuary vincular las diferentes vistas y/o CRUD que existan dentro del sistema interno propuesto, este sidebar es una barra de menú, el cuál se posicionara en el lado izquierdo del sistema interno, el sidebar estará compuesto por secciones que vincularán a las interfaces ya antes creadas. Para ello, haremos uso de un paquete llamado AdminLTE, este paquete ofrece varias plantillas las cuales podemos hacer uso y configurar diferentes aspectos de alguna de las plantillas que este paquete posea.

En la siguiente imagen 3.11, podemos apreciar el archivo de configuraciones de este paquete donde se define la barra lateral que queremos que se tenga, en esta parte mencionare

las vistas que existiran para que se pueda interactuar con todo el sistema, además de la configuración de la apariencia de este, usando estilos por defecto que te trae dicho paquete o agregando estilos personalizados.

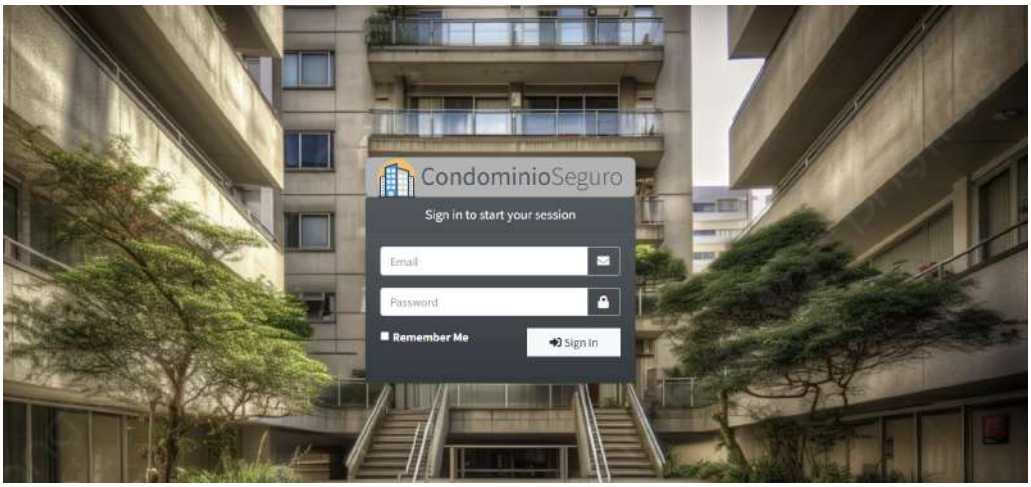
Además de las configuraciones ya mencionadas, se puede configurar el o los íconos que use el sistema como vista predeterminada de la viñeta del navegador, así como también el nombre que se desea que se vea en conjunto con el titulo de la vista.

Figura 3.11: Configuraciones del paquete AdminLTE.



Por último, realizaremos la implementación de la vista del login, esto en consecuencia a que dicho sistema necesitará una autentificación para poder acceder a las vistas y toda la funcionalidad que se implemente en dicho sistema interno, esto por un tema de seguridad ya que, solo el personal autorizado tendrá acceso a dicho sistema, entre estas personas tenemos al administrador o presidente de dicho condominio y/o edificio de uso residencial, y al personal de seguridad, quien es el encargado de velar por los residentes de dicho lugar, por el momento se tiene previsto que solo estas personas tengan el acceso al sistema interno implementado.

Figura 3.12: Vista del login del sistema.



3.2. Implementación de algoritmos de detección de rostros

Ya habiendo culminado con las interfaces y funcionalidades que tendrá nuestro sistema interno, pasaremos a realizar la implementación de scripts con modelos de IA para el reconocimiento facial; pero antes de ello, necesitamos de algo fundamental, el cual es requerido para hacer un buen reconocimiento facial; nos referimos a los modelos de detección de rostros.

A partir de esta sección empezaremos con el desarrollo de modelos de IA, como se indico con anterioridad, los scripts con los modelos y algoritmos de IA serán implementados basado en el desarrollo iterativo e incremental, lo que nos lleva a tener una constante mejora de los modelos que se vayan a utilizar. Otro punto muy importante es que los modelos que utilizaremos, son algoritmos ya implementados, los cuales integraremos a nuestro sistema interno.

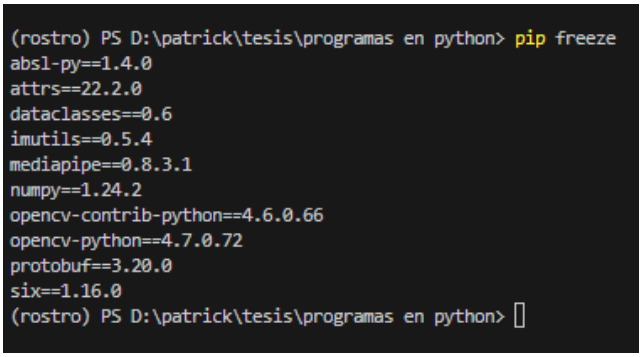
Al hablar sobre la detección de rostros, podemos apreciar que hay una gran variedad de métodos, modelos y/o algoritmos, los cuales nos van a ayudar a detectar uno o varios rostros que se encuentren dentro de un material visual (imagen, video). Para ello necesitamos escoger el algoritmo más idóneo; es decir, el modelo que más nos convenga según lo que vayamos a requerir. A continuación desarrollaremos los scripts con algunos de estos algoritmos que ya se encuentran implementados hoy en día, para luego proceder a realizar sus respectivas pruebas en diferentes escenarios, circunstancias y lugares; esto con la finalidad de poder compararlos y así, seleccionar el algoritmo mas eficiente y eficaz, según los resultados que obtengamos de cada uno de estos 3 algoritmos utilizados.

Gran parte de los modelos y/o algoritmos de detección de rostros, ya se encuentran implementados en python, por lo que será nuestro lenguaje de programación; además que pyhton tiene una comunidad muy grande y activa. Por otro lado, python cuenta con una gran cantidad de librerias, las cuales nos beneficiaran para este caso, como es la detección de rostros. Algunas de estas librerias son OpenCV y MediaPipe, los cuales ya fueron mencionados y descritos con anterioridad en el capítulo 2. Haremos uso de dichas librerias para los algoritmo que utilizaremos a continuación.

Por otro lado, cabe resaltar que para para los siguientes algoritmos utilizados se hizo uso de entornos virtuales, ya que no todos los algoritmos requerian la misma version de python y hasta la misma versión de librerías, de las cuales son requeridas y fundamentales para estos algoritmos. En un ordenador, uno puede instalar una versión específica, pero gracias a los entornos virtuales podemos tener diferentes versiones para cada algoritmo a utilizar, lo cual nos facilitará la implementación al momento de realizarlo de una forma más rápida y además que, no habra conflictos entre las distintas versiones que uno tenga.

Antes de desarrollar los diferentes algoritmos para la detección de rostros, iniciaremos con la creación de un entorno virtual que satisfazca todas las librerias y versiones de estas que se lleguen a utilizar, crearemos el entorno virtual rostro", el cual tendra las siguientes librerias como podemos visualizar en la siguiente imagen 3.13; es probable que en caso, al llegar hacer uso de diferentes librerias como es mediapipe y openCV, estas puedan coexistir en un mismo entorno virtual sin tener conflictos entre ellos.

Figura 3.13: Entorno virtual de python utilizado para los algoritmos de detección de rostros.



3.2.1. Detección de rostros por mediapipe con Face Detection

Como primer algoritmo a utilizar en nuestro script, tendremos el de la detección de rostros usando la librería de mediapipe, como se menciono en el capítulo 2, mediapipe es una librería que posee el método de "Face Detection", el cual está basado en BlazeFaces, que es un detector liviano de rostros. Antes de proceder con la implementación brindaremos las librerías utilizadas en este algoritmo:

- mediapipe v0.8.3.1
- opencv-python v4.7.0.72

Como primer paso de la implementación de nuestro script, que utilizará este modelo de detección de rostros, tendremos que definir en que modo se realizará la detección de rostros, estos pueden ser los diferentes materiales visuales que veremos a continuación:

- en una imagen
- en un video
- en una transmisión en vivo mediante una cámara web

Como siguiente paso tendremos que definir la confianza mínima del rostro, para saber cuando uno o varios de los objetos detectados, es con certeza un rostro o si es un posible caso de falso positivo, para ello tendremos que configurar y probar dicha confianza mínima para que pueda aceptar y/o detectar un rostro en diferentes escenarios posibles; el valor de esta variable se encuentra dentro del rango de 0 a 1, cuyo valor por defecto es de 0.5.

Además de la variable de confianza mínima existe otra variable configurable, que es la del umbral mínimo de supresión, cuya función reside en la superposición de un rostro y de igual manera aceptar a uno o varios objetos obtenidos y decidir si es un rostro o no; el valor de la variable tambien se encuentra dentro del rango de 0 a 1, cuyo valor por defecto es de 0.3.

Con estos 3 primeros pasos, los cuales nos ayudaron a definir el modo de detección y las variables de aceptación mínima, debemos obtener y/o ubicar los 6 puntos claves o puntos de referencia como usualmente se conoce, entre estos puntos tenemos los siguientes:

- centro del ojo derecho
- centro del ojo izquierdo
- punta de la nariz
- centro de la boca
- trago de la oreja derecha
- trago de la oreja izquierda

Figura 3.14: Puntos de refecrencia haciendo uso de mediapipe.

```
# Ojo derecho
x_RE = int(detection.location_data.relative_keypoints[0].x * width)
y_RE = int(detection.location_data.relative_keypoints[0].y * height)
cv2.circle(image, (x_RE, y_RE), 3, (0, 0, 255), 25)

# Ojo izquierdo
x_LE = int(detection.location_data.relative_keypoints[1].x * width)
y_LE = int(detection.location_data.relative_keypoints[1].y * height)
cv2.circle(image, (x_LE, y_LE), 3, (255, 0, 255), 25)

# Punta de la nariz
x_NT = int(detection.location_data.relative_keypoints[2].x * width)
y_NT = int(detection.location_data.relative_keypoints[2].y * height)
cv2.circle(image, (x_NT, y_NT), 3, (255, 0, 0), 25)

# Centro de la boca
x_MC = int(mp_face_detection.get_key_point(detection, mp_face_detection.FaceKeyPoint.MOUTH_CENTER).x * width)
y_MC = int(mp_face_detection.get_key_point(detection, mp_face_detection.FaceKeyPoint.MOUTH_CENTER).y * height)
cv2.circle(image, (x_MC, y_MC), 3, (0, 255, 0), 25)

# Trago de la oreja derecha
x_RET = int(mp_face_detection.get_key_point(detection, mp_face_detection.FaceKeyPoint.RIGHT_EAR_TRAGION).x * width)
y_RET = int(mp_face_detection.get_key_point(detection, mp_face_detection.FaceKeyPoint.RIGHT_EAR_TRAGION).y * height)
cv2.circle(image, (x_RET, y_RET), 3, (0, 255, 255), 25)

# Trago de la oreja izquierda
x_LET = int(mp_face_detection.get_key_point(detection, mp_face_detection.FaceKeyPoint.LEFT_EAR_TRAGION).x * width)
y_LET = int(mp_face_detection.get_key_point(detection, mp_face_detection.FaceKeyPoint.LEFT_EAR_TRAGION).y * height)
cv2.circle(image, (x_LET, y_LET), 3, (255, 255, 0), 25)
```

Una vez obtenidos estos 6 puntos de referencia, procederemos a dibujar un cuadro, en donde el área contenga estos 6 puntos, de esta forma se hara referencia a que dicho cuadro pertenece a la imagen de un rostro; con esto obtendremos todos los posibles rostros obtenidos ya sea en una imagen, video o transmisión en vivo. Al visualizar la ventana emergente de python, según el modo y tipo de material visual escogido, se realizará el análisis en donde obtendremos el recuadro dibujado donde se albergue el rostro de una o varias personas; además de dibujar los 6 puntos de referencia, la librería de mediapipe nos permite configurar estos dibujos o trazos modificandolos en su color, grosor del borde y el radio en caso de los puntos de referencia.

Figura 3.15: Algoritmo de detección de rostros usando mediapipe.

detector_in_photo_mp.py | detector_in_video_mp.py

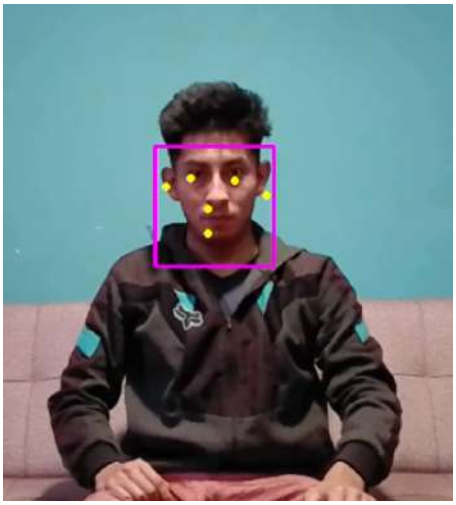
detector de rostros > detector_in_video_mp.py > ...

```python
import cv2
import mediapipe as mp
import imutils
mp_face_detection = mp.solutions.face_detection
mp_drawing = mp.solutions.drawing_utils

cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)
#cap = cv2.VideoCapture("Videos/video_001.mp4")
#cap = cv2.imread("Images/imagen_0002.jpg")

with mp_face_detection.FaceDetection(
    min_detection_confidence=0.75) as face_detection:

    while True:
        ret, frame = cap.read()
        if ret == False:
            break
        frame = imutils.resize(frame, width=720)
        frame = cv2.flip(frame, 1)
        frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

        results = face_detection.process(frame_rgb)

        if results.detections is not None:
            for detection in results.detections:
                mp_drawing.draw_detection(frame, detection,
                    mp_drawing.DrawingSpec(color=(0, 255, 255), circle_radius=2),
                    mp_drawing.DrawingSpec(color=(255, 0, 255)))

        cv2.imshow("Frame", frame)
        k = cv2.waitKey(1) & 0xFF
        if k == 27:
            break
cap.release()
cv2.destroyAllWindows()
```

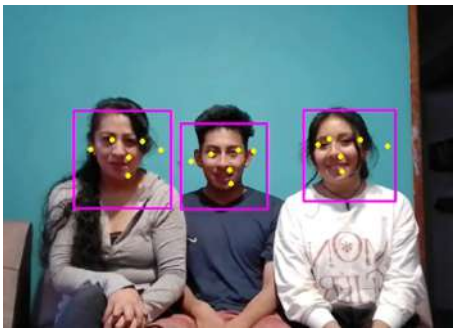
Luego de haber configurado estos valores para la obtención de rostros, adaptaremos el algoritmo utilizado para realizar las respectivas pruebas en los diferentes materiales visuales que se obtenga, entre estos tendremos: imágenes, videos y video cámara.

Ya habiendo terminado con la implementación del algoritmo que usaremos, procederemos a realizar pruebas en estos 3 casos de material visual, donde en cada caso se realizarán pruebas para detectar un rostro y otras para detectar varios rostros, además de diferentes posibles escenarios donde existan riesgos como variación de la iluminación, enfoque del rostro, posición del rostro, gestos y/o muecas. Algunas de estas pruebas realizadas son como las siguientes imágenes 3.16, en dichas pruebas se estuvo cambiando la variable de confianza de modo que se pueda precisar con mayor exactitud el o los rostros encontrados en una instancia de imagen.

Figura 3.16: Detección de rostros usando mediapipe.



(a) Detección de una sola persona



(b) Detección de varias personas

Mediantes las pruebas realizadas con la detección de rostros usando la librería de mediapipe, pudimos apreciar que se detecta los rostros de forma exitosa y reconoce los 6 puntos de referencia todo esto desde una distancia muy corta entre la cámara y el rostro de la persona, a medida que la distancia aumenta, la detección de rostros sigue siendo eficiente, pero los 6 puntos de referencia no lo son tanto ya que, varian o indican el lugar de uno de estos puntos de referencia erroneamente; además, la precisión del algoritmo es dependiente al ángulo en el cual se encuentra ubicada la cámara, al igual que la luminosidad que exista en el lugar donde se realiza dicha detección de rostros. Cabe recalcar que este algoritmo pierde la precisión al momento de ampliar y/o aumentar la distancia entre la cámara y la persona a detectar, pero si la detección de rostros se realiza a una distancia corta el rendimiento aumenta, haciendo de esta la más óptima posible.

3.2.2. Detección de rostros por OpenCV con DNN

Como segundo algoritmo a utilizar, tenemos la detección de rostros basado en el módulo DNN que nos brinda la librería de OpenCV, en el lenguaje de python, antes de proceder con la implementación brindaremos las librerías utilizadas para este segundo algoritmo:

- numpy v1.24.2
- opencv-contrib-python v4.6.0.66

Como única librería a usar tenemos la de opencv-contrib-python, pero al momento de instalar dicha libreria en python se nos instala también la librería de numpy. Además de las librerias, necesitamos obtener 2 archivos los cuales son sumamente importantes:

- La arquitectura de la red, que es la forma de como esta diseñado dicha red, que datos de entrada espera, cual es el desarrollo que se realiza entre las redes profundas y que datos de salida devuelve. El archivo obtenido, tiene como nombre deploy.prototxt

- Los pesos de la red, estos datos son complementarios a la arquitectura para definir cuando un objeto detectado es un rostro y cuando no lo es, para obtener estos pesos se tiene que haber realizado el entrenamiento con un dataset de rostros, existen varios framework para realizar el entrenamiento, en este caso se uso el framework de caffe. El archivo obtenido tiene como nombre res10 300x300 ssd iter 140000.caffemodel

Estos archivos son creados y entrenados por terceros, para luego ser publicados en la página oficial de cada libreria o plataformas digitales como GitHub, en donde se tiene una comunidad muy activa y se encuentra en constante mantenimiento por posibles bugs o refactorizaciones que se lleguen a dar según este lo requiera.

Una vez ya definido la librería junto a su versión requerida y de haber obtenido los archivos de la arquitectura y pesos de la red, procederemos con la implementación del algoritmo a utilizar. Como primer paso, necesitamos definir la ruta de los archivos obtenidos tanto de la arquitectura como de los pesos de la red, habiendolos definido los llamaremos para cargar el modelo, esto gracias a una función del modulo DNN de openCV llamado - eadNetFromCaffe", este método es usado particularmente ya que, el entrenamiento se realizó en el framework caffe.

Habiendo cargado el modelo con la arquitectura y los pesos de la red, procederemos a definir el modo en que se realizará la detección de rostros, del mismo modo que el anterior algoritmo, se tiene 3 modos posibles como son la detección en una imagen, un video y/o una transmisión en vivo. Una vez definido el modo de detección de rostros, pasamos a obtener el tamaño de la imagen y/o grabación, esto para luego redimensionarlo en un tamaño, el cual requiera la arquitectura de red.

Luego de haber concluido dichas especificaciones procederemos a usar otra de las funciones del módulo DNN de openCV llamado blobFromImage, el cual obtiene la imagen después de haber pasado por su normalización, cambio de canales y resta media, dicha función te pide parámetros para poder ejecutarse, entre estos parámetros tenemos los siguientes:

- la imagen
- el factor de escala, que es una redimensión de la imagen, cuyo valor por defecto es de 1.0
- tamaño de la imagen esperado
- la resta media, esta puede ser un valor global o general para todos o un valor para cada canal usado
- cambio de los canales de la red

Teniendo configurado e implementado todo lo anterior, veremos que obtenidos de la función anterior, nos damos cuentas que son los datos de entrada de la arquitectura de la red, esto quiere decir que los aceptará; pasamos todos estos valores obtenido al modelo cargado con anterioridad y luego de esto podremos obtener una serie de arreglos donde nos brindarán datos sobre las coordenadas donde se encuentra un posible rostro , esto junto a su confianza.

Para ello tenemos que definir cual es la confianza mínima, en nuestro caso, para que pueda aceptar un objeto como rostro; luego de esto se necesita dibujar un recuadro con

Figura 3.17: Algoritmo de detección de rostros usando openCV con DNN.

detector_in_photo_cv.py | detector_in_video_cv.py

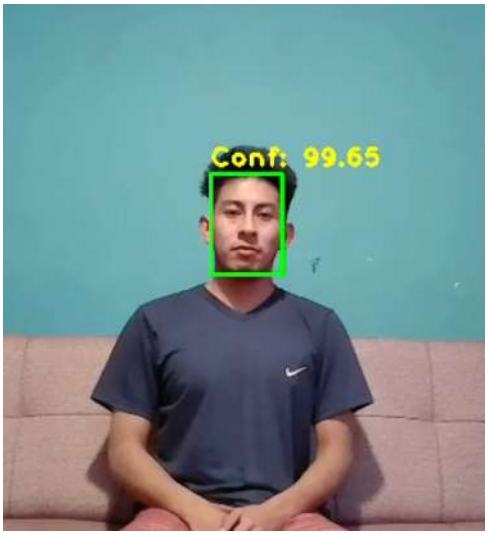
detector de rostros > detector_in_video_cv.py > ...

```python
import cv2

# ------------ READ DNN MODEL ------------
# Model architecture
prototxt = "model/deploy.prototxt"
# Weights
model = "model/res10_300x300_ssd_iter_140000.caffemodel"

# Load the model
net = cv2.dnn.readNetFromCaffe(prototxt, model)

# ------- READ THE IMAGE AND PREPROCESSING -------
cap = cv2.VideoCapture("Videos/video_001.mp4")

while True:
    ret, frame = cap.read()
    if ret == False:
        break

    height, width, _ = frame.shape
    frame_resized = cv2.resize(frame, (300, 300))

    # Create a blob
    blob = cv2.dnn.blobFromImage(frame_resized, 1.0, (300, 300), (104, 117, 123))

    # ------- DETECTIONS AND PREDICTIONS ----------
    net.setInput(blob)
    detections = net.forward()
    #print("detections.shape:", detections.shape)

    for detection in detections[0][0]:
        #print("detection:", detection)
        if detection[2] > 0.75:
            box = detection[3:7] * [width, height, width, height]
            x_start, y_start, x_end, y_end = int(box[0]), int(box[1]), int(box[2]), int(box[3])
            cv2.rectangle(frame, (x_start, y_start), (x_end, y_end), (0, 255, 0), 2)
            cv2.putText(frame, "Conf: {:.2f}".format(detection[2] * 100), (x_start, y_start - 5), 1, 1.2, (0, 255, 255), 2)

    cv2.imshow("Frame", frame)
    if cv2.waitKey(1) & 0xFF == 27:
        break
cap.release()
cv2.destroyAllWindows()
```

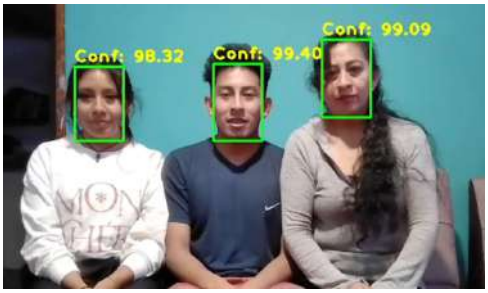
las coordenadas obtenidas de la red como se puede apreciar en el algoritmo de la siguiente imagen 3.17, realizaremos algunas pruebas para saber cual será la confianza mínima para este algoritmo, por otro lado podemos configurar el dibujo del recuadro que se haga como el grosor del borde, el color, entre otros.

Una vez configurado todos estos valores que nos ofrece DNN en openCV, procederemos a realizar las pruebas en los diferentes materiales visuales y en diferentes circusntancias y/o contextos, para poder así comparar y analizar posteriormente estos algoritmos basados en la detección de rostros, algunas de estas pruebas se realizaron de la siguiente manera como se puede apreciar en la imagen 3.18, en dichas imágenes podemos apreciar la detección de rostros cuando se enfoca en una sola persona así como cuando existen varias personas en el enfoque realizado; otra de las mediciones para poder medir la eficiencia de dicho algoritmo fue realizar varias pruebas a distancias diferentes, a modo de hallar la distancia máxima a la cual el algoritmo pueda seguir siendo eficiente.

Figura 3.18: Detección de rostros usando openCV con DNN.



(a) Detección de una sola persona



(b) Detección de varias personas

Del mismo modo que el algoritmo anterior, se estuvo modificando la variable de confianza para hacer que dicho algoritmo sea lo mas eficiente posible, también se realizaron pruebas en diferentes tonalidades de luminidad para ver como actuaba el algoritmo y que tan efciente era en esas situaciones, también probamos en diferentes ángulos posibles de la cámara y vimos que tuvo una gran probabilidad de éxito.

A diferencia del anterior algoritmo, este algoritmo de detección de rostros usando la librería de openCV juntos con las DNN, es relativamente más eficiente que el anterior algoritmo puesto que, dicho algoritmo tiene mayor rango en cuanto a la distancia se refiere.

3.2.3. Detección por OpenCV con Haar Cascades

A continuación realizaremos la implementación del algoritmo de Haar Cascades para la detección de rostros, este algoritmos es una de los más comunes que suelen usar, además viene integrada en la librería de openCV de python, antes de proseguir indicaremos la librería y su versión que se está utilizando para este algoritmo:

- opencv-contrib-python v4.2.0.32

Una vez instalada la librería y su versión requerida, continuaremos con la implementación del script con el algoritmo a utilizar. Comenzaremos importando la librería requerida, para luego definir el modo en el que se realizará la detección de rostros, así como en los algoritmos anteriores existen 3 modos: detección en una imagen, en un video y/o en una transmisión en vivo.

Continuaremos cargando el clasificador de Haar Cascades, este clasificador ya viene al momento de instalar dicha librería o también puede descargarse desde su plataforma oficial de GitHub, existen varios clasificadores de dicho algoritmo, en este caso utilizaremos el clasificador por defecto.

Ahora continuamos convirtiendo la imagen que está en colores a una imagen en escala de grises, para ello utilizaremos la función çvtColor"de la librería de openCV; una vez obtenida la imagen en escala de grises utilizaremos otra función llamada "detectMultiScale"del clasificador cargado de Haar Cascades, dicha función acepta los siguientes parámetros:

- la imagen, esta puede ser a color o en escala de grises, según como se esta usando en nuestro algoritmo.
- el factor de escala, el cual especifica que la imagen puede ser reducida, este parámetro tiene como valor por defecto 1.0.
- el mínimo de vecinos, el cual indica cuantos objetos mínimos se necesita que compartan el mismo área o región para que dicho objeto pueda ser detectado como rostro.
- tamaño mínimo, esto indica cual es el tamaño mínimo para que un objeto pueda ser detectado como rostro.
- tamaño máximo, esto indica cual es el tamaño máximo para que un objeto pueda ser detectado como rostro.

Como datos de salida de la función de detectMultiScale, obtendremos 4 coordenadas que pertenece a la región donde se encuentra almacenada el rostro detectado, con estas coordenadas procedemos a dibujar un cuadro que lo enmarque, dicho dibujo se puede configurar para cambiar el color del borde asi como el grosor de dicho borde.

Teniendo todo la implementación el código debe ser como el que se aprecia en la siguiente imagen 3.19.

Figura 3.19: Algoritmo de detección de rostros usando openCV con Haar Cascades.

detector_in_video_cascades.py | detector_in_photo_cascades.py

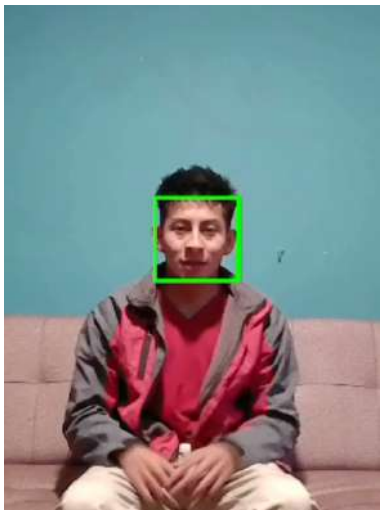
detector de rostros > detector_in_video_cascades.py > ...

```
1  import cv2
2
3  cap = cv2.VideoCapture(0)
4  faceClassif = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
5
6  while True:
7    ret,frame = cap.read()
8    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
9
10   faces = faceClassif.detectMultiScale(gray, 1.3, 5)
11
12   for (x,y,w,h) in faces:
13     cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
14
15   cv2.imshow('frame',frame)
16
17   if cv2.waitKey(1) & 0xFF == ord('q'):
18     break
19 cap.release()
20 cv2.destroyAllWindows()
```

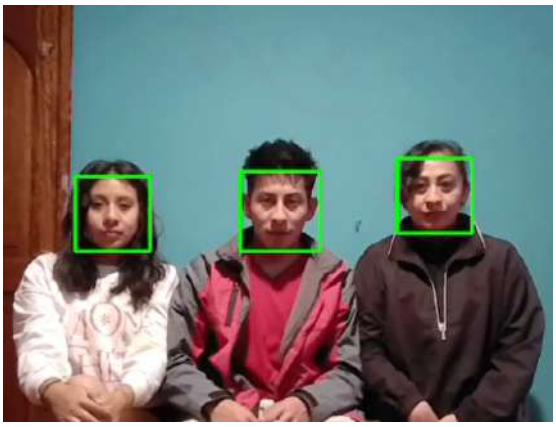
Una vez culminado la implementeción del script con el algoritmo y/o modelo q utilicemos. Continuaremos con la realización de las pruebas correspondientes para realizar un análisis sobre dicho algoritmo, para ello realizamos algunas pruebas para la detección de rostros de una y/o varias personas como se muestra en las siguientes imágenes 3.20.

Además, realizamos las pruebas a diferentes distancias entre la cámara y la(s) persona(s), para ver el rango máximo de distancia que pueda alcanzar dicho algoritmo, por otro lado también realizamos las pruebas en ambientes con diferente luminosidad para ver el comportamiento y la eficacia del algoritmo, al igual que los anteriores algoritmos realizamos las pruebas en diferentes angulos de la cámara y/o ángulos de la persona.

Figura 3.20: Detección de rostros usando openCV con Haar Cascades.



(a) Detección de una sola persona



(b) Detección de varias personas

Analizando este algoritmo pudimos apreciar que la detección en diferentes ambientes así como en diferentes circunstancias no bajan demasiado en la eficacia de la detección de rostros, por otro lado los valores obtenidos por este algoritmo es un tanto superior a los otros 2 previamente vistos y analizados de igual forma.

3.2.4. Integración de la detección de rostros

Luego de haber realizado la implementación de los scripts, con los 3 algoritmos vistos anteriormente y haber realizado pruebas para cada uno de estos, comparamos los resultados de estos, teniendo como conclusión que hay ciertas diferencias según las circunstancias y el ambiente en donde se realice la detección de rostros, esto se puede apreciar en los 3 algoritmos utilizados, si bien la eficiencia de cada uno de estos varia de forma diferente, existe una disminución en su eficiencia ante las diferentes circunstancias en las cuales se realizaron las pruebas.

Al haber analizado cada uno de estos algoritmos, pudimos apreciar que la detección de rostros usando la librería de OpenCV con el modelo de Haar Cascades es el algoritmo más óptimo entre los 3 mencionados con anterioridad, ya que dicho algoritmo tiene la eficiencia más alta en cuanto se refiere a las diferentes pruebas que se han realizado en cada uno de estos 3 algoritmos implementados; por otro lado, cabe resaltar que, el lugar donde se realizarán las grabaciones será un ambiente controlado por el personal de seguridad.

Tabla 3.1: Comparación entre los modelos de detección de rostros

Modelo	**Puntos de referencia clave**	**Distancia máxima (metros)**	**Intensidad luminosa**	**Precisión**
Librería mediapipe	Ojos y nariz	2	Luz natural	96.85 %
			Iluminación máxima	96.85 %
			Iluminación moderada	95.14 %
			Iluminación media	88.57 %
Librería openCV con DNN	Ojos	4	Luz natural	98.14 %
			Iluminación máxima	98.14 %
			Iluminación moderada	97.14 %
			Iluminación media	95.57 %
Librería openCV con Haar Cascades	Ojos	4	Luz natural	98.71 %
			Iluminación máxima	98.71 %
			Iluminación moderada	98.57 %
			Iluminación media	97.42 %

Luego de haber realizado el análisis y comparación entre estos 3 algoritmos utilizados, llegamos a la conclusión de seleccionar el algoritmo de Haar Cascades para la detección de rostros; puesto que, es el algoritmo con mejores resultados. Ahora, procedemos a integrar el algoritmo de Haar Cascades al sistema que venimos implementando.

Para la integración tenemos que analizar en que parte del sistema debemos de integrar dicho algoritmo, para ello viendo los distintos CRUD's ya implementados y sabiendo que en una primera fase tenemos que realizar la detección y recopilación de rostros de todas las personas que viven en un condominio y/o edificio de uso residencial; se optó que en el CRUD de residentes, al momento de crear un nuevo residente vincularemos el algoritmo de Haar Cascades para la detección de rostro y asi tendremos los datos con los cuales se realizará el entrenamiento y comparativa del reconocimiento facial de lo cual se hablará posteriormente.

En el sistema interno emplementado, existirán 2 formas de realizar la detección de rostros, una opción es el de subir un video para realizar la obtención de rostros de una persona, en este caso de un residente, dicho video debe ser solo de la persona que vamos a agregar y debe estar bien enfocado, puesto que si en el video aparece más de una persona, se detectará todos los rostros para luego guardarlos y decir que los rostros detectados pertenecen a una persona a la cuál se está agregando como residente, es ahí donde los rostros obtenidos serán datos inválidos o engañosos puesto que, se tendra rostros de varias personas.

Otra de las opciones disponibles que habrá, será la de la obtención de rostros desde una grabación en vivo; es decir, debemos tener vinculado una cámara al ordenador donde se use el sistema en cuestión, con ello podremos realizar la detección de rostros en vivo (en tiempo real), esta fimación se hará según las indicaciones que de la persona autorizada al momento de agregar un nuevo residente en el sistema, algunas de estas indicaciones puede ser que se ponga de perfil, que mire a la cámara, que realice algunos gestos en el rostro, que simule algunos estados de ánimo, entre otros. Este último tipo de obtención de rostros de una persona mediante la transmisión en vivo, es la más idónea para que el algoritmo utilizado funcione de la forma más óptima posible.

Por otro lado, se debe agregar como un nuevo residente a una o varias personas que tengan algún tipo de parentezco con el propietario o residente titular del apartamento, mas no a otra persona como algún amigo(a) de dicho residente puesto que, un familiar es más cercano que una amistad misma; además que, es más probable que un familiar llegue a habitar el adartamento, por el contrario una amistad solo lo frecuentará por un determinado tiempo que puede ser algo efímero.

Con la integración de dicho algoritmo, podremos obtener una cantidad aproximada de 700 imágenes de rostros por residente, lo cual nos ayudará posteriormente al momento de realizar la comparación para efectuar el reconocimiento facial, estas imágenes obtenidas serán almacenadas en el proyecto más no en la BD puesto que, esto lo haría demasiado lento al tener una cantidad elevada de imágenes; ademas que la BD tendrá un peso excesivo, es por ello que para minimizar todo esto almacenaremos las imágenes en carpetas del mismo proyecto, donde cada carpeta tendra el nombre y id de la persona residente de dicho condomino y/o edificio de uso residencial.

Figura 3.21: Integración del algoritmo Haar Cascades para la detección de rostros.

3.3. Implementación de algoritmos de reconocimiento facial

Luego de haber implementado los algoritmos para la detección de rostros y haberlo vinculado al sistema interno implementado, procederemos a la implementación de scripts con modelos ya existentes de reconocimiento facial, al igual que los modelos de detección de rostros, existe una gran variedad de modelos para realizar el reconocimiento facial, en esta sección hablaremos sobre alguno de estos, asi como su respectiva implementación usando estos algoritmos ya existentes, para proceder a realizar las pruebas y el análisis de cada uno de estos algoritmos.

Para realizar el reconocimiento facial necesitamos también la detección de rostros, para ello usaremos el modelo escogido con anterioridad para realizar dicha detección; al realizar la detección de rostros almacenaremos alrededor de 700 rostros por persona, como se indico con anterioridad estos rostros estarán almacenados en carpetas, una carpeta por residente, todos estos rostros almacenados serán nuestros datos de prueba con los cuales entrenaremos los siguientes algoritmos. Del mismo modo para estas implementaciones haremos uso de los entornos virtuales de python para no tener el problema de versiones tanto en la versión del lenguaje de programación como en las librerias que vayamos a requerir.

Como pruebas realizaremos el reconocimiento facial en videos previamente ya grabados, tendremos videos simulando en los lugares donde se encuentran instaladas las cámaras de seguridad, las cuales se encuentran en los puntos de entrada hacia el condominio y/o edificio de uso residencial. Los videos serán grabados en distintas horas del día para poder simular los cambios de luminosidad en dicha área, que se encuentra cubierta por las cámaras.

3.3.1. Reconocimiento facial con Eigen Faces

Como primer algoritmo a reutilizar, tenemos el reconocimiento facial mediante Eigen Faces, el cual se basa en obtener y almacenar los componentes principales de una persona.

Como primer paso tenemos que realizar el entrenamiento del algoritmo antes de ponerlo en práctica, para ello haremos uso de los rostros obtenidos previamente; leeremos todas las imágenes almacenadas y les pondremos una etiqueta por persona, esto para tener en cuenta que las 700 imágenes que contiene cada carpeta pertenecen a una persona diferente. Luego de haber etiquetado todos los rostros, haremos uso del módulo face de la librería de OpenCV, dicho módulo tiene un método llamado EigenFaceRecognizerCreate el cuál va a crear un reconocedor de rostros.

Habiendo creado ya el reconocedor de rostros de Eigen Faces, realizaremos su respectivo entrenamiento, para esto haremos uso del método "trainçuyos parámetros son los siguientes:

- El arreglo donde se almacenen todas las imágenes obtenidas en la detección de rostros de cada uno de los residentes.
- El arreglo donde se tengan todas las etiquetas de los diferentes rostros obtenidos, estos deben estar almacenados en un arreglo de la librería numpy.

Una vez que se haya terminado de entrenar el reconocedor de rostros con los datos brindados, vamos a generar y guardar el modelo de Eigen Faces obtenido del entrenamiento realizado, esto gracias al método write del reconocedor de rostros cuyo único parámetro es el nombre con el cuál se guardara el modelo Eigen Faces, dicho archivo tiene que almacenarse en una extensión ya sea XML o YAML, así habremos terminado de implementar el código para el entrenamiento del modelo de Eigen Faces como se ve en la siguiente imagen 3.23.

Figura 3.22: Entrenamiento del modelo de Eigen Faces.

entrenandoRF.py

reconocimiento facial > entrenandoRF.py > ...

```python
import cv2
import os
import numpy as np

dataPath = 'D:/patrick/tesis/programas en python/reconocimiento facial/data'
peopleList = os.listdir(dataPath)
print('Lista de personas: ', peopleList)

labels = []
facesData = []
label = 0

for nameDir in peopleList:
    personPath = dataPath + '/' + nameDir
    print('Leyendo las imágenes')

    for fileName in os.listdir(personPath):
        labels.append(label)
        facesData.append(cv2.imread(personPath+'/'+fileName,0))
        image = cv2.imread(personPath+'/'+fileName,0)
    label = label + 1

# Métodos para entrenar el reconocedor
face_recognizer = cv2.face.EigenFaceRecognizer_create()

# Entrenando el reconocedor de rostros
print("Entrenando modelo...")
face_recognizer.train(facesData, np.array(labels))

# Almacenando el modelo obtenido
face_recognizer.write('modeloEigenFace.xml')
print("Modelo almacenado...")
```

Ya habiendo obtenido el modelo de Eigen Faces, podremos empezar con la implementación del reconocimiento facial. Para ello empezamos creando nuevamente el reconocedor de rostros de Eigen Faces, leemos el modelo generado luego de realizar el entrenamiento, esto gracias al método read.

A partir de ellos definimos algunas características ya antes mencionadas, una de estas es definir el modo para el reconocimiento facial, estos pueden ser video o transmisión en vivo. También debemos implementar la detección de rostros, esto reutilizaremos del algoritmo seleccionado con anterioridad el cual es el algoritmo de Haar Cascades.

Al momento de analizar cada imagen dentro del video o transmisión en vivo, haremos uso del método "predict"del modelo de Eigen Faces, el cual nos ayudará a predecir la etiqueta a la cual pertenece dicha imagen junto a su confianza asociada. Del mismo modo que en la detección de rostros debemos de controlar la confianza para saber a partir de que confianza se debe detectar una imagen como intruso o persona desconocida.

Como último paso procederemos a dibujar un recuadro del rostro que se detecte en dicha grabación y pondremos el nombre de la persona y/o residente en caso se detecte similitud con los rostros almacenados de cada persona que reside en dicho condominio y/o edificio de uso residencial, caso contrario se pondrá un texto que diga persona desconocida, de esta manera el código será similar a la siguiente imagen 3.23.

Figura 3.23: Código del reconocimiento facial con el modelo de Eigen Faces.

entrenandoRF.py | ReconocimientoFacial.py

reconocimiento facial > ReconocimientoFacial.py > ...

```python
import cv2
import os

dataPath = 'D:/patrick/tesis/programas en python/reconocimiento facial/Data'
imagePaths = os.listdir(dataPath)
print('imagePaths=',imagePaths)

face_recognizer = cv2.face.EigenFaceRecognizer_create()

# Leyendo el modelo
face_recognizer.read('modeloEigenFace.xml')

cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)

faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')

while True:
    ret,frame = cap.read()
    if ret == False: break
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    auxFrame = gray.copy()

    faces = faceClassif.detectMultiScale(gray,1.3,5)

    for (x,y,w,h) in faces:
        rostro = auxFrame[y:y+h,x:x+w]
        rostro = cv2.resize(rostro,(150,150),interpolation= cv2.INTER_CUBIC)
        result = face_recognizer.predict(rostro)

        cv2.putText(frame,'{}'.format(result),(x,y-5),1,1.3,(255,255,0),1,cv2.LINE_AA)
        # EigenFaces
        if result[1] < 5700:
            cv2.putText(frame,'{}'.format(imagePaths[result[0]]),(x,y-25),2,1.1,(0,255,0),1,cv2.LINE_AA)
            cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
        else:
            cv2.putText(frame,'Desconocido',(x,y-20),2,0.8,(0,0,255),1,cv2.LINE_AA)
            cv2.rectangle(frame, (x,y),(x+w,y+h),(0,0,255),2)

    cv2.imshow('frame',frame)
    k = cv2.waitKey(1)
    if k == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

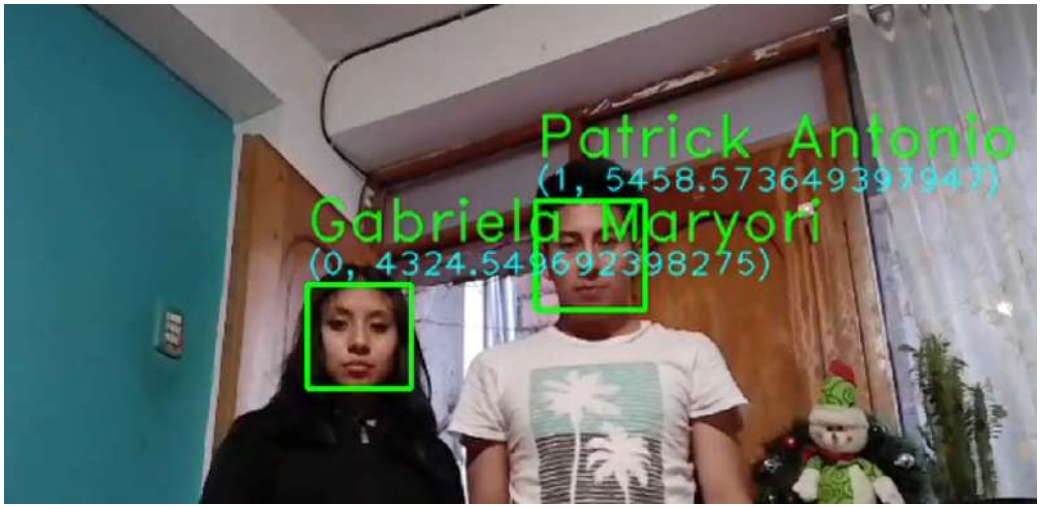
Una vez que terminemos de implementar tanto la creación y entrenamiento del modelo como el reconocimiento facial, empezaremos a realizar las diferentes pruebas posibles, de modo que podamos analizar la eficiencia y eficacia de dicho algoritmo.

Como se menciono con anterioridad, para realizar las pruebas debemos de haber realizado con anticipación la detección y el registro respectivo de los rostros de las personas a las cuales tiene que reconocer el algoritmo de Eigen Faces; además con dicha data de los rostros vamos a realizar el entrenamiento del modelo para que con ello podamos realizar

con éxito el respectivo reconocimiento facial usando Eigen Faces. Las pruebas se realizarán en entornos controlados así como en posibles lugares donde tenga o se llegue a instalar las cámaras de seguridad, algunos de estos lugares son las entradas a dichas viviendas.

Al realizar las pruebas nos dimos cuenta que dicho algoritmo de Eigen Faces trabaja muy bien, ya que su eficiencia y eficacia son muy altas, pero por otro lado hay momentos en los cuales no reconoce del todo a una persona a la cual se le conoce, esto se da por diversas razones como la variación con respecto a la iluminación en el entorno donde se realice el reconocimiento facial, no tener imágenes de rostro con todas las expresiones faciales de uno, al igual que no tener diferentes posiciones del rostro de una persona.

Figura 3.24: Reconocimiento facial con el modelo de Eigen Faces.



3.3.2. Reconocimiento facial con Fisher Faces

Como siguiente algoritmo tenemos el reconocimiento facial mediante el algoritmo de Fisher Faces.

En este caso la implementación del script para utilizar el algoritmo de Fisher Faces, es muy similar al algoritmo de Eigen Faces, exceptuando las siguientes partes de toda la implementación del algoritmo en código:

- Para este algoritmo se usará el método FisherFaceRecognizerCreate para crear el reconocedor de rostros.
- Debemos de entrenar dicho algoritmo con los datos obtenidos en la detección de rostros.
- Debemos de generar y almacenar el modelo obtenido luego de haber realizado el entramiento del modelo.
- Debemos de leer el modelo de Fisher Faces.
- Debemos de controlar y validar la confianza asociada a la imagen para definir cuando un rostro se detecta como intruso.

A continuación mostraremos las partes del código en donde se modifica y/o se cambia el método utilizado, eso en comparación del modelo de Eigen Faces, primero visualizaremos el entrenamiento del modelo como se ve en la siguiente imagen 3.25, después continuaremos con el reconocimiento facial utilizando el modelo generado luego del entrenamiento con Fisher Faces, dicha modificación será como en la imagen 3.26.

Figura 3.25: Entrenamiento del modelo de Fisher Faces.

```
22
23  # Métodos para entrenar el reconocedor
24  face_recognizer = cv2.face.FisherFaceRecognizer_create()
25
26  # Entrenando el reconocedor de rostros
27  print("Entrenando...")
28  face_recognizer.train(facesData, np.array(labels))
29
30  # Almacenando el modelo obtenido
31  face_recognizer.write('modeloFisherFace.xml')
32  print("Modelo almacenado...")
```

Figura 3.26: Código del reconocimiento facial con el modelo de Fisher Faces.

```
10  # Leyendo el modelo
11  face_recognizer.read('modeloFisherFace.xml')
12
13  cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
14  #cap = cv2.VideoCapture('Video.mp4')
15
16  faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
17
18  while True:
19      ret,frame = cap.read()
20      if ret == False: break
21      gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
22      auxFrame = gray.copy()
23
24      faces = faceClassif.detectMultiScale(gray,1.3,5)
25
26      for (x,y,w,h) in faces:
27          rostro = auxFrame[y:y+h,x:x+w]
28          rostro = cv2.resize(rostro,(150,150),interpolation= cv2.INTER_CUBIC)
29          result = face_recognizer.predict(rostro)
30
31          cv2.putText(frame,'{}'.format(result),(x,y-5),1,1.3,(255,255,0),1,cv2.LINE_AA)
32          # FisherFace
33          if result[1] < 500:
34              cv2.putText(frame,'{}'.format(imagePaths[result[0]]),(x,y-25),2,1.1,(0,255,0),1,cv2.LINE_AA)
35              cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
36          else:
37              cv2.putText(frame,'Desconocido',(x,y-20),2,0.8,(0,0,255),1,cv2.LINE_AA)
38              cv2.rectangle(frame, (x,y),(x+w,y+h),(0,0,255),2)
39
40      cv2.imshow('frame',frame)
41      k = cv2.waitKey(1)
42      if k == 27:
43          break
44
45  cap.release()
46  cv2.destroyAllWindows()
```

Después de haber realizado las modificaciones en el código anterior, para que al momento de realizar el entrenamiento ya no se realice con el algoritmo de Eigen Faces sino con el de Fisher Faces, de igual forma al momento de realizar el reconocimiento facial se tuvo que adecuar el código implementado para que funcione con este nuevo algoritmo de Fisher Faces, con ello procederemos a realizar las diferentes pruebas para analizar este algoritmo y ver su rendimiento tanto en su eficiencia como su eficacia.

Las pruebas que realizarán serán las mismas ya antes hechas en el reconocimiento facial con el algoritmo de Eigen Faces, esto con el fin de realizar un comparación entre estos algoritmo y ver cuál de ellos es el mejor en los diferentes ambientes y contextos en donde se realicen dichas pruebas.

Al realizar las pruebas con el algoritmo de Fisher Faces nos dimos cuenta que dicho algoritmo funciona bien y es muy eficiente, pero no tanto en comparación al algoritmo de Eigen Faces que se mencionó con anterioridad.

Figura 3.27: Reconocimiento facial con el modelo de Fisher Faces.

3.3.3. Reconocimiento facial con LBPH Faces

Como siguiente algoritmo tenemos el reconocimiento facial mediante el algoritmo LBPH Faces.

En este caso la implementación del algoritmo es muy similar exceptuando las siguientes partes de toda la implementación del algoritmo en código:

- Para este algoritmo se usará el método LBPHFaceRecognizerCreate para crear el reconocedor de rostros.
- Debemos de entrenar dicho reconocedor con los datos obtenidos en la detección de rostros.
- Debemos de generar y almacenar el modelo obtenido luego de haber realizado el entramiento del modelo.
- Debemos de leer el modelo de LBPH Faces.
- Debemos de controlar y validar la confianza asociada a la imagen para definir cuando un rostro se detecta como intruso.

A continuación mostraremos las partes del código en donde se cambia el método a utilizar en comparación a los modelos anteriormente mencionados e implementados, primero

visualizaremos el entrenamiento del modelo como se ve en la siguiente imagen 3.28, después continuaremos con el reconocimiento facial utilizando el modelo generado luego del entrenamiento con LBPH Faces, dicha modificación será como en la siguiente imagen 3.29.

Figura 3.28: Entrenamiento del modelo de LBPH Faces.

```
22
23  # Métodos para entrenar el reconocedor
24  face_recognizer = cv2.face.LBPHFaceRecognizer_create()
25
26  # Entrenando el reconocedor de rostros
27  print("Entrenando...")
28  face_recognizer.train(facesData, np.array(labels))
29
30  # Almacenando el modelo obtenido
31  face_recognizer.write('modeloLBPHFace.xml')
32  print("Modelo almacenado...")
```

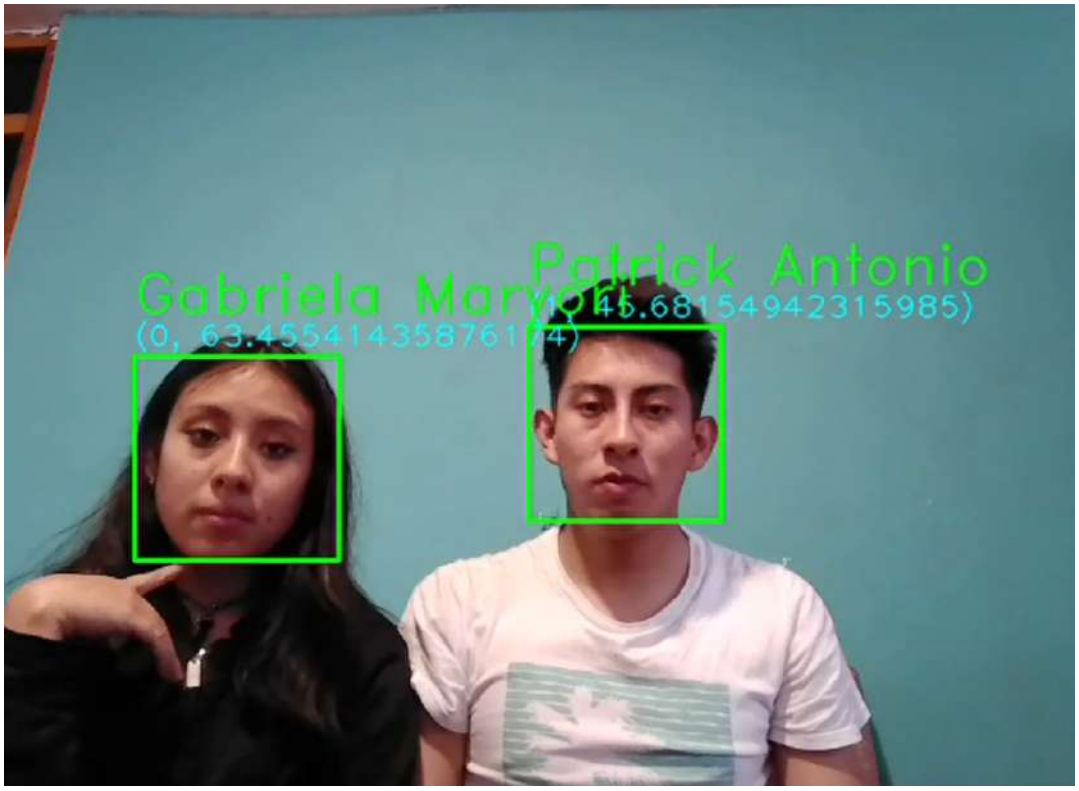
Figura 3.29: Código del reconocimiento facial con el modelo de LBPH Faces.

```
10  # Leyendo el modelo
11  face_recognizer.read('modeloLBPHFace.xml')
12
13  cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
14  #cap = cv2.VideoCapture('Video.mp4')
15
16  faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
17
18  while True:
19      ret,frame = cap.read()
20      if ret == False: break
21      gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
22      auxFrame = gray.copy()
23
24      faces = faceClassif.detectMultiScale(gray,1.3,5)
25
26      for (x,y,w,h) in faces:
27          rostro = auxFrame[y:y+h,x:x+w]
28          rostro = cv2.resize(rostro,(150,150),interpolation= cv2.INTER_CUBIC)
29          result = face_recognizer.predict(rostro)
30
31          cv2.putText(frame,'{}'.format(result),(x,y-5),1,1.3,(255,255,0),1,cv2.LINE_AA)
32          # LBPHFace
33          if result[1] < 70:
34              cv2.putText(frame,'{}'.format(imagePaths[result[0]]),(x,y-25),2,1.1,(0,255,0),1,cv2.LINE_AA)
35              cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
36          else:
37              cv2.putText(frame,'Desconocido',(x,y-20),2,0.8,(0,0,255),1,cv2.LINE_AA)
38              cv2.rectangle(frame, (x,y),(x+w,y+h),(0,0,255),2)
39
40      cv2.imshow('frame',frame)
41      k = cv2.waitKey(1)
42      if k == 27:
43          break
44
45  cap.release()
46  cv2.destroyAllWindows()
```

Al igual que los algoritmo anteriores, luego de realizar las respectivas modificaciones para que el algoritmo trabaje con LBPH Faces, procederemos a realizar las diferentes pruebas usando las mismas grabaciones y en los entornos donde se procedio a realizar las pruebas de los anteriores algoritmos, esto para poder analizarlos y hacer una comparación entre todos ellos.

En estas pruebas dicho reconocimiento facial tuvo una alta eficiencia, del mismo modo que su eficacia fue muy buena, pero como en el primer algoritmo implementado hubo momentos en los cuales no reconoce a una persona por el tema de cambios en la iluminación de dicho entorno, y la falta de más imágenes de rostros con diferentes expresiones faciales y posiciones del mismo.

Figura 3.30: Reconocimiento facial con el modelo de LBPH Faces.



3.3.4. Integración del reconocimiento facial

Luego de haber realizado la implementación de los scripts, reutilizando 3 de los algoritmos para realizar la función de reconocimiento facial, procedemos a realizar una comparativa entre ellos para seleccionar el algoritmo más óptimo, para luego integrarlo al sistema interno implementado.

Entre estos algoritmo tenemos Eigen Faces, Fisher Faces y LBPH Faces, algunas de estas pruebas realizadas fueron tanto dentro de un ambiente como es el departamente de una persona y fuera de dicho ambiente, en este caso nos referimos al ambiente de las escaleras y al ambiente del garage.

Entre estas pruebas realizadas obtuvimos resultados favorables para los algoritmos de Eigen Faces y LBPH Faces que tuvieron una alta eficiencia dentro de sus resultados, por otro lado tuvimos un resultado aceptado pero no tan alto del algoritmo de Fisher Faces, obviamente estos resultados varían según al tipo de luminidad, distancia y gestos de la persona en donde se encuentra ubicada la cámara.

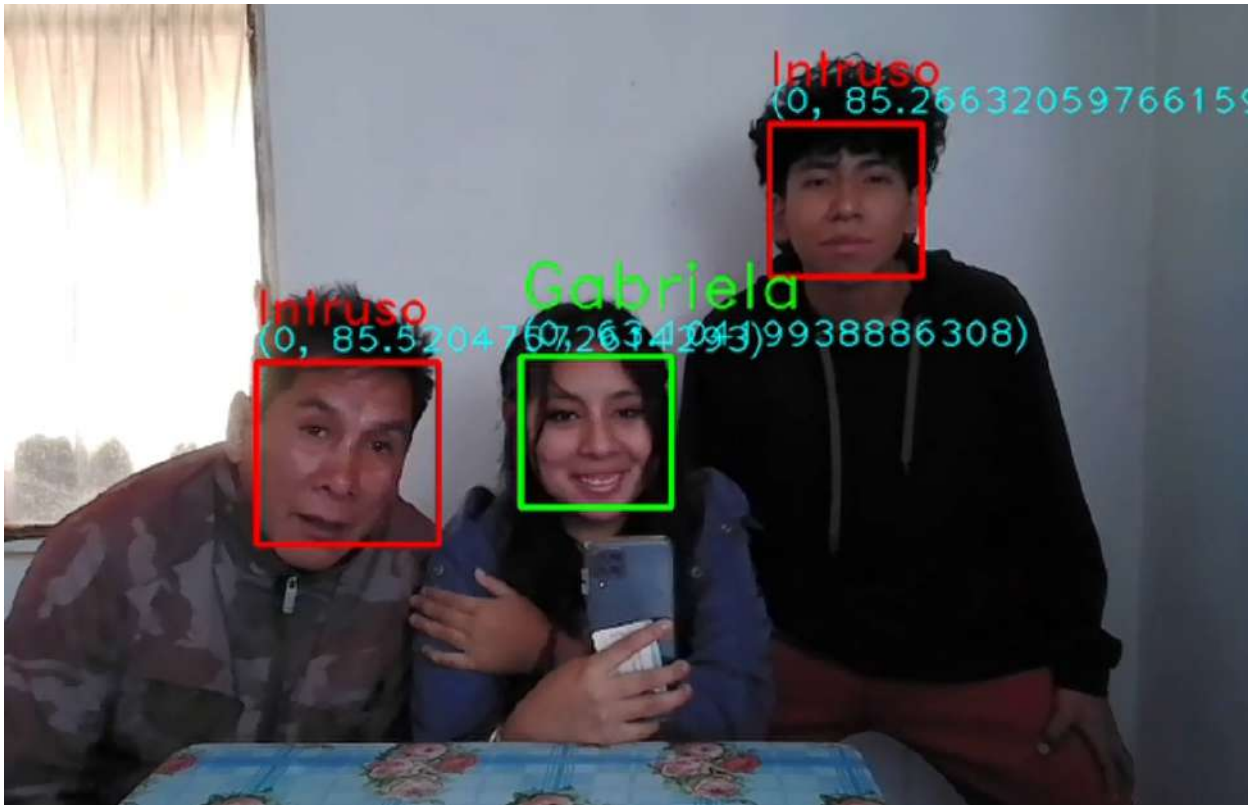
Es así que se dio como ganador al algoritmo de LBPH Faces por obtener un resultado mayor que los otros 2 algoritmos de reconocimiento facial, dando como ganador a este, para luego proceder a integrarlo al sistema interno implementado, para que el reconocimiento facial se aplique con este algoritmo.

Tabla 3.2: Comparación entre los algoritmo de reconocimiento facial con 3500 imágenes de rostros

Algoritmo	**Tiempo de Entrenamiento**	**Compilación FPS**	**Iluminación**	**Precisión**
Eigen Faces	1h 54m 2s	16.46	Luz natural	95.60 %
			Iluminación máxima	95.60 %
			Iluminación moderada	94.00 %
			Iluminación promedia	91.60 %
Fisher Faces	1h 44m 55s	18.12	Luz natural	94.40 %
			Iluminación máxima	94.40 %
			Iluminación moderada	92.80 %
			Iluminación promedia	90.00 %
LBPH Faces	16s	15.56	Luz natural	96.40 %
			Iluminación máxima	96.40 %
			Iluminación moderada	94.80 %
			Iluminación promedia	93.20 %

Este algoritmo detectará a todas las personas conocidas, con los cuales se hizo el entrenamiento; en caso no detecte ningún parecido con las imágenes de entrenamiento, detectará a dicha persona como un intruso a dicho condominio y/o residencia de uso residencial, de esta forma sabremos quien es un residente y quien no.

Figura 3.31: Detección de instrusos con el reconocimiento facial.



3.4. Integración de cámaras al sistema

En esta sección daremos a conocer las distintas cámaras con las cuales se realizaron las pruebas de reconocimiento facial, cada una de estas cámaras tienen diferentes aspectos y características una de otra, de esta forma podremos saber que tipos de cámaras pueden vincularse al sistema interno implementado y poder efectuar la función y análisis de la transmisión en vivo para que finalmente se pueda proceder al reconocimiento facial en el espacio donde se encuentran ubicadas estas cámaras.

Como primera cámara de prueba usamos la cámara que se encuentran integradas en las laptops, estás cámaras ya tienen la funcionalidad propia de una webcam; es por ello que, es mucho más fácil de obtener la transmisión en vivo de estas, ya que el driver de la cámara se encuentra instalado en la laptop con los datos de fábrica, obviamente estos driver pueden ser actualizadas de forma periódica cada vez que el modelo suba una nueva actualización de estas. A continuación mostraremos en la imagen 3.32 la información de la laptop con la cuál se realizo la prueba de reconocimiento facial con su cámara que tiene integrada.

Figura 3.32: Información de la laptop en la cuál se uso su cámara integrada.

Como segunda cámara de prueba, usamos una cámara que funciona como webcam el cuál se conecta mediante un puerto USB a la laptop y/o computadora que se tenga, esta cámara al tener un cable de conexión como es el puerto USB, basta con conectar al equipo y automáticamente ya se tiene acceso a dicha cámara, en caso se tenga más de una cámara conectada solo se configura para que la cámara que uno desee, sea la cámara principal y de uso general en caso así se desee. Estas cámaras son usualmente usadas para la transmisión en vivo de diferentes plataformas como meet, zoom, stream, entre otros; es así que podemos decir que este tipo de cámaras son las más usadas por personas que tienen como fin reuniones virtuales en las diferentes plataformas web que existen. La cámara que usamos es como la que se aprecia en la siguiente imagen:

Figura 3.33: Cámara webcam usada.

Como tercera cámara de prueba, usamos una que cualquier persona puede llegar a tener; nos referimos a las cámaras que nos ofrecen nuestros equipos celulares. Hoy en día los celulares que podemos ver en el mercado, en su gran mayoria son de gama media, gracias a ello podemos decir que, las cámaras que tienen los equipos celulares de hoy en día son bastante buenas e incluso mejores que las cámaras de prueba que se uso y se mencionó con anterioridad, de igual manera tenemos las cámaras que se encuentran en las tablets, ya que estos equipos son también considerados dispositivos móviles.

Para poder vincular estas cámaras con el sistema interno implementado, tenemos que tener un aplicativo instalado en dicho dispositivo móvil, este aplicativo es llamado "DroidCam", el cuál podemos obtener desde la tienda de playStore en caso de que sea un equipo Android como se visualiza en la imagen 3.34 o en la tienda de Apple en caso de ser un equipo IOS. Este aplicativo nos permite vincular la cámara del dispositivo móvil al equipo en donde se trabaje, existen 2 tipos de vinculación:

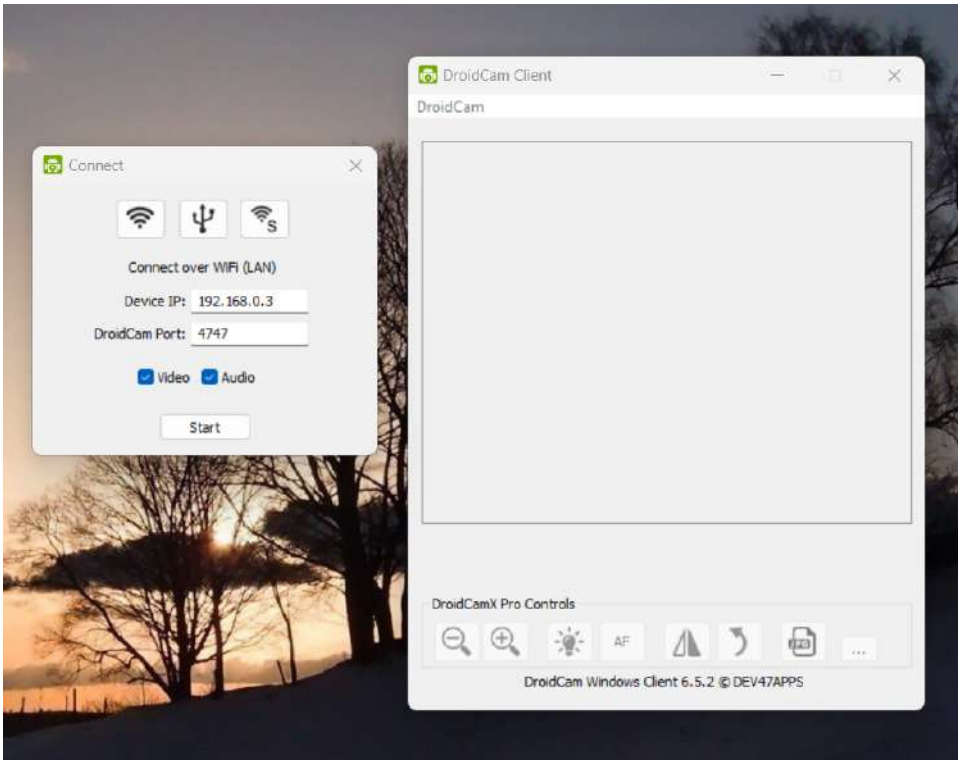
- Por cable, esta manera es una de las más fáciles y básicas que existe, puesto que al realizar una conexión por cable es una conexión directa.
- Por WiFi, para ello tanto el dispositivo móvil como el equipo de trabajo tienen que estar conectadas a la misma red, ya que de esta manera la red a la cuál se encuentran conectadas cumple la función de un mediador o conductor entre estas.

Dicho aplicativo tiene que estar instalado tanto en el dispositivo móvil como en el equipo de trabajo, de esta forma se trabajará como un emisor en el caso del dispositivo móvil a usar y un receptor en el caso del computador, al momento de abrir el aplicativo en el ordenador tenemos que escoger una opción el cual es referente al tipo de conexión que se realizará, si es por cable o por wifi; en caso de ser una conexión por cable se vinculará automaticamente, por otro lado si es una conexión por WiFi está te pedirá la dirección IP de tu dispositivo móvil el cuál se obtendra al abrir el aplicativo en dicho dispositivo móvil.

Figura 3.34: Aplicativo DroidCam para conectar cámara del dispositivo móvil al ordenador.

Al terminar con la configuración del aplicativo solo se da click al botón de start y la cámara del dispositivo móvil ya se encuentra vinculado al ordenador en cuestión, de esta forma podemos tener acceso a la cámara del dispositivo móvil desde una computadora y/o laptop ya que esta lo detectará como si fuese una WebCam propia.

Figura 3.35: Aplicativo DroidCam en la laptop.



A continuación mostraremos la información del dispositivo móvil usado para esta prueba, en donde se dispondrá de sus respectivas cámaras como una WebCam dentro de la computadora y/o laptop que se utilice:

Figura 3.36: Dispositivo móvil de prueba donde se uso su cámara.

Como siguiente cámara de prueba, tenemos a las cámaras analógicas, en este caso una cámara marca Sony Cyber-shot el cuál pudimos vincular con nuestro equipo de trabajo mediante su cable USB como en la imagen 3.37 y un programa llamado "Imaging Edge Desktopçomo se ve en la imagen 3.37; dicha aplicación nos ayuda a vincular dicha cámara como si fuese una webcam, lo cual hace más facil la integración y/o la transmisión en vivo de dicha cámara.

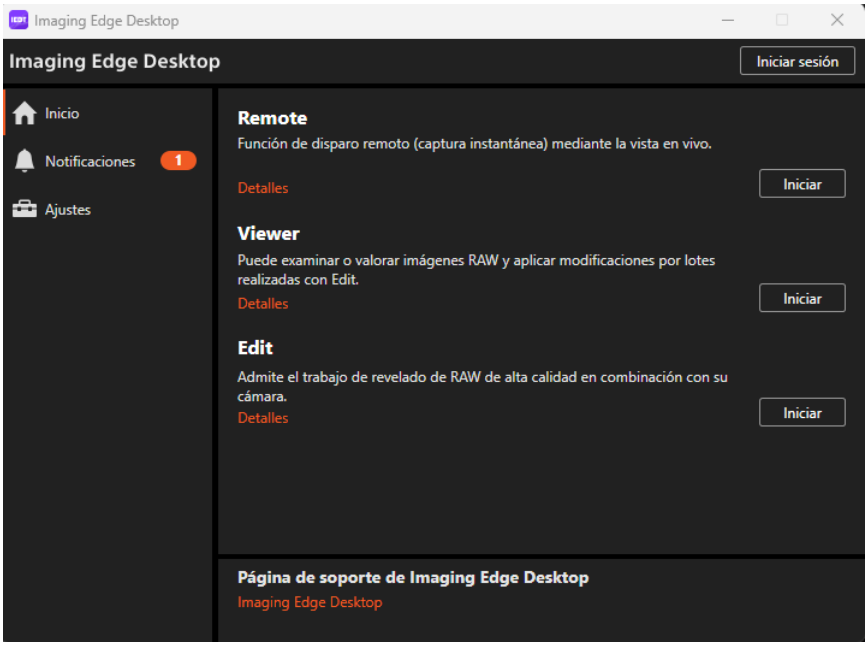
La mayoría de cámaras análogas se puede obtener su transmisión en vivo siempre y cuando tenga un puerto de salida de video, a este puerto de salida se le puedo agregar un capturador de video lo cual al conectarse se conecta y vincula como si fuese una webcam, lo cual lo hace más fácil aun, es así que este tipo de cámaras pueden ser vinculadas a un equipo y/u ordenador.

Figura 3.37: Cámara análoga Sony.

(a) Imagen frontal de la cámara análoga

(b) Imagen trasera de la cámara análoga

Figura 3.38: Aplicación de integración para la cámara análoga Sony.



Otra de las cámaras con la cuál se hicieron pruebas, es una cámara IP marca Sitecom cuyo modelo es el WL-405, este tipo de cámaras son usadas para interiores las cuales se conectan a una red mediante un cable Ethernet.

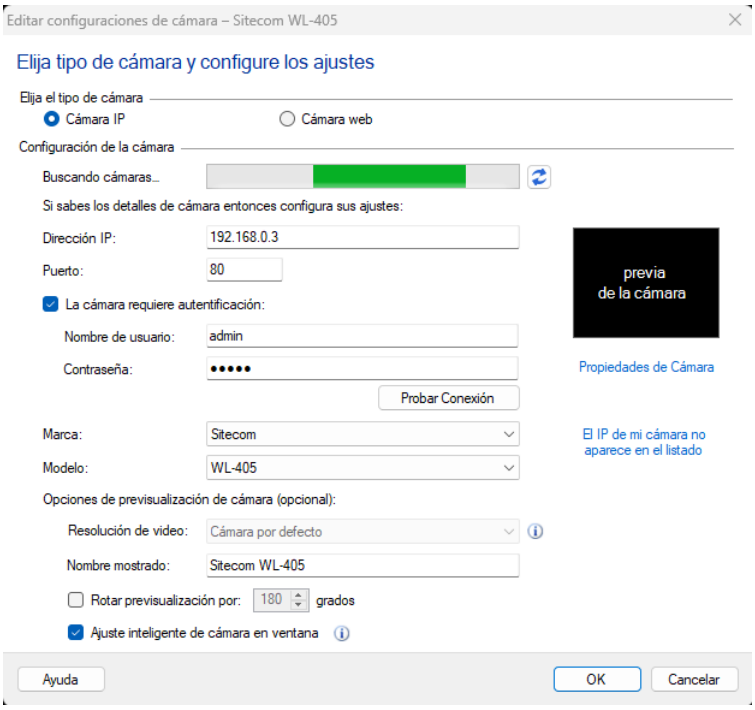
Figura 3.39: Cámara IP - Sitecom.

(a) Imagen frontal de la cámara IP

(b) Imagen trasera de la cámara IP

Al conectar dicho cable podemos hacer uso de distintas aplicaciones para obtener la dirección IP de dicha cámara conectada, en este caso usaremos el aplicativo ÏP Camera Viewer 4çomo vemos en la imagen 3.40, en donde configuramos según la cámara de seguridad que tengamos. Una vez terminado de configurar el aplicativo obtendremos la visualización de la transmisión en vivo de dicha cámara en el aplicativo usado, para poder obtener dicha transmisión en vivo y poder analizarlo haremos uso del protocolo RTSP con el cuál tendremos acceso a la transmisión en vivo a través de una URL, para este caso, la URL en cuestion es "rtsp://admin:admin@192.168.0.3:554/ipcam.sdp".

Figura 3.40: Aplicación IP Camera VIewer 4.



Con este protocolo RTSP podemos vincular la mayoría de las cámaras de seguridad, ya sean cámaras IP, cámaras web y/o cámaras que estén gestionadas por un NVR.

Otra de las cámaras con las que se realizó las pruebas sin mucho éxito fue la que se visualiza en la siguiente imagen 3.41; esto se debe a que esta marca de cámaras no tienen configurado el acceso al protocolo RTSP, lo que hace que su única forma de recuperar la transmisión en vivo sea por su aplicativo mismo.

Figura 3.41: Cámara WiFi Ezviz modelo C2C.

Como última cámara de prueba se uso una cámara de marca HikVision las cuales están vinculadas a un DVR de la misma marca, esto nos ayuda a vincular estas cámaras análogas a este DVR que transmite en una determinada IP, si bien de este DVR se puede obtener la transmisión en vivo mediante el protocolo RTSP uno de los inconvenientes visto en este tipo de cámaras es su resolución ya que es mala y se encuentra pixeleada, por lo tanto no es apta para poder realizar el reconocimiento facial.

Figura 3.42: Cámara y DVR HikVision.

(a) Cámara HikVision usada

(b) DVR HikVision usada

3.5. Pruebas realizadas al sistema

En este punto ya se tiene implementado el sistema interno propuesto, así como la integración de un par de cámaras convencionales que se posee en dicho edificio de uso residencial y también la detección de intrusos aplicado en la transmisión en vivo de dichas cámaras.

Figura 3.43: Sistema interno de un usuario administrador

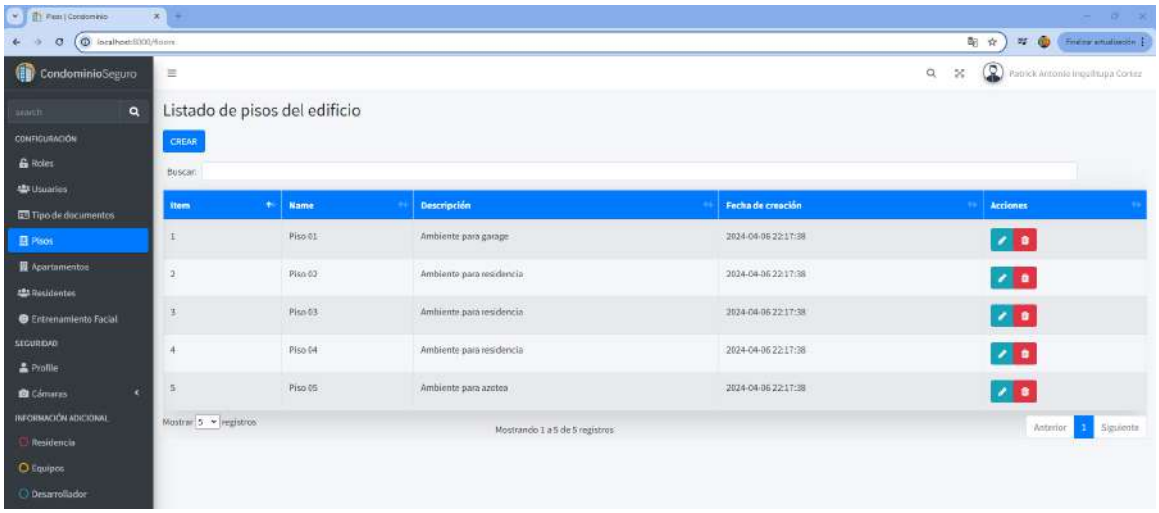
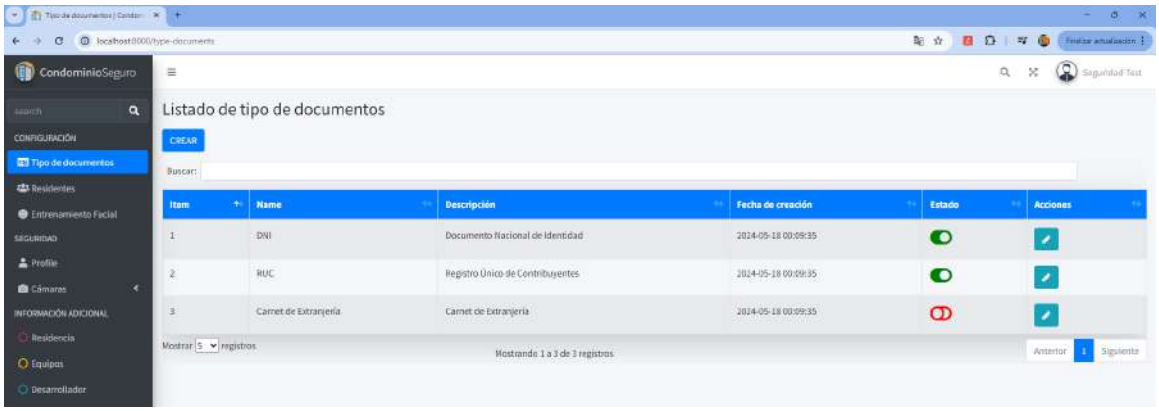


Figura 3.44: Sistema interno de un usuario de seguridad



Como se hablo con anterioridad se tiene 2 tipos de usuario:

- Usuario Administrador, este usuario tendrá acceso general al sistema interno y será usado por el gerente y/o propietario de un condominio o edificio de uso residencial.
- Usuario de Seguridad, este usario tendrá acceso a vistas de su interes como el de su información personal, residentes y a las cámaras de videovigilancia, entre estas vistas podrá realizar la acción de crear, listar y editar más no la acción de eliminar, dicho usuario será usado por el personal de seguridad o cuidante que este encargado con el ingreso de personal a dicha propiedad.

Para cada uno de estos usuarios se realizarón las diferentes pruebas que veremos a continuación, dichas pruebas fueron según al acceso y permisos que tiene cada uno de estos usuarios.

Empezaremos con las pruebas realizadas para un usuario administrador:

- Prueba de login y logout.
- Editar su perfil.
- Cambiar su contraseña.
- Visualizar el listado de roles.
- Crear un nuevo rol.
- Editar un rol existente.
- Eliminar un rol existente.
- Visualizar el listado de usuarios.
- Crear un nuevo usuario.
- Editar un usuario existente.
- Eliminar un usuario existente.
- Visualizar el listado de tipo de documentos.
- Crear un nuevo tipo de documento.
- Editar un tipo de documento existente.
- Eliminar un tipo de documento existente.
- Habilitar o deshabilitar un tipo de documento existente.
- Visualizar el listado de pisos.
- Crear un nuevo piso.
- Editar un piso existente.
- Eliminar un piso existente.
- Visualizar el listado de apartamentos.
- Crear un nuevo apartamento.
- Editar un apartamento existente.
- Eliminar un apartamento existente.
- Visualizar el listado de residentes.
- Crear un nuevo residente almacenando los rostros de dicha persona.

- Editar un residente existente.
- Aumentar más capturas de rostros de un residente existente.
- Eliminar un residente existente.
- Realizar el entrenamiento del modelo de reconocimiento facial.
- Visualizar la transmisión en vivo de las cámaras con la detección de intrusos integrada.

Continuaremos con las pruebas realizadas para un usuario de seguridad:

- Prueba de login y logout.
- Editar su perfil.
- Cambiar su contraseña.
- Visualizar el listado de tipo de documentos.
- Crear un nuevo tipo de documento.
- Editar un tipo de documento existente.
- Habilitar o deshabilitar un tipo de documento existente.
- Visualizar el listado de residentes.
- Editar un residente existente.
- Aumentar más capturas de rostros de un residente existente.
- Realizar el entrenamiento del modelo de reconocimiento facial.
- Visualizar la transmisión en vivo de las cámaras con la detección de intrusos integrada.

Con respecto al número de cámaras que se llego a integrar al sistema interno, se obtuvo la transmisión en vivo de 2 cámaras en simultáneo con las cuales se realizó las pruebas de la detección de intrusos al momento de ingresar a dicha propiedad; estas cámaras se encontraron ubicados en el interior de dicha propiedad, con dirección a la puerta de ingreso de este.

A continuación indicaremos las características de las pruebas que se realizaron del sistema interno implementado con la funcionalidad del reconocimiento facial.

Tabla 3.3: Pruebas del sistema interno implementado

Propiedad	**Cámara(s) utilizada(s)**	**Ubicación de la cámara (Altura - metros)**	**Inclinación de la cámara**	**Iluminación**	**Precisión**
Edificio 1	Cámara EzViz y móvil	2.36m	45°	Luz natural	96.40 %
				Iluminación máxima	96.40 %
Edificio 2	Cámara móvil	1.95m	30°	Luz natural	96.40 %
				Iluminación máxima	96.40 %
Edificio 3	Cámara móvil y webcam	2.16m	45°	Luz natural	95.70 %
				Iluminación moderada	94.80 %
Edificio 4	Cámara análoga	1.87m	30°	Luz natural	95.20 %
				Iluminación promedia	93.20 %

Capítulo 4

Análisis y discusión de resultados

4.1. Análisis de resultados respecto a los objetivos

- La base de datos creada, fue realizada según las funcionalidades que se desea tener en el sistema interno implementado, en una primera fase (MVP); esto puede llegar a evolucionar en caso se quiera aumentar funciones nuevas o potenciar las ya existentes en dicho sistema.
- Al realizar la detección de rostros se procedio a guardar las imágenes de rostros en el proyecto puesto que, si se almacena en la BD, haría que este incremente su tamaño, donde el rendimiento sea el menos óptimo ya que, las imágenes de rostro por persona que se obtendrá, pueden crecer de acuerdo a lo que uno desee; es así que, habra problemas de escalabilidad en caso se guarden las imágenes de rostros en una BD.
- Se obtuvo buenos resultados en la detección de rostros con los diferentes algoritmos que reutilizamos, un modelo fue hecho usando la librería de mediapipe, los otros 2 fueron implementados con la librería de OpenCV, con la diferencia que, una de estas fue cargando la arquitectura y pesos de un modelo y el otro fue con el modelo de Haar Cascades; de esta manera se dio como ganador al modelo de Haar Cascades por su eficiencia en la detección de rostros y además porque es un modelo ligero el cuál basta para realizar dicha detección de rostros en un ambiente controlado.
- Se analizó 3 algoritmos de reconocimiento facial como Eigen Faces, Fisher Faces y LBPH Faces, obteniendo muy buenos porcentajes de eficiencia en estos 3 algoritmos, superando los 90 %, pero entre estos 3 algoritmos hubo uno que sobresalio del resto, dando así como ganador al algoritmo de LBPH Faces para realizar el reconocimiento facial, para luego integrarlo al sistema.
- La detección de rostros y almacenado de una persona, tienen que ser con características similares a las cámaras que se usarán como video vigilancia del ambiente en donde se instale, para que cuando se realice el entrenamiento, la eficiencia sea el más óptimo posible.
- Se realizarón diversas pruebas con diferentes tipos de cámaras, dando como resultado que, la mayoria de las cámaras convencionales que una persona usa o posee, puede ser utilizado como una cámara de seguridad para su entorno y así controlar el ingreso de personas a su ambiente de uso residencial; cabe resaltar que para tener una buena

eficiencia del reconocimiento facial, se debe tener una cámara de gama media, con esto nos referimos que su resolución tiene que ser como mínima de 1080px y de 2MP.

4.2. Discusión de resultados respecto a los antecedentes

- En el trabajo presentado por (Heredia Salazar and Rea Rodríguez, 2022), se llegó a tener algunas coincidencias tales como: cuanto mayor sea la cantidad de rostros por persona y el número de personas en si, el entrenamiento del algoritmo demorará más, así mismo mientras el reconocimiento facial se aplique a mayor distancia la efectividad de este se reducirá de forma proporcional hasta que dicha efectividad sea nula, esto se da a partir de los 4 metros de distancia. Por otra parte, la efectividad a 1 metro de distancia es la más óptima puesto que, se llego al 96.40 % de efectividad, como se puede apreciar en la tabla 3.1.
- En el trabajo presentado por (Córtes Martínez and Vásquez Bohórquez, 2021), se pudo apreciar que igual que en el presente proyecto de investigación el reconocimiento facial detecta a más de una persona según este en el panorama de la cámara, para el caso de 1 metro de distancia que es el mejor caso se llega a detectar y analizar a 2 y/o hasta 3 personas en una misma captura, puesto que, el ambiente de pruebas se hizo en un ambiente controlado, nos basto con aplicar uno de los algoritmos que nos ofrece la librería de openCV en python, en este caso el algoritmo usado es el de LBPH Faces.
- En el trabajo presentado por (Briones Gárate, 2020) nos menciona que el reconocimiento facial ayuda a combatir el crimen, en el presente proyecto de investigación concluimos que con la integración del reconocimiento facial podremos mejorar el control de una o varias personas en un condominio y/o edificio de uso residencial, haciendo que de esta manera el personal de seguridad o el encargado de dicho sistema, pueda actuar según la situación lo amerite al momento de detectar un intruso.
- En el trabajo presentado por (Verdeguer Valderrama, 2022) se llego a una misma conclusión al verificar que el algoritmo de LBPH Faces es superior en cuanto al tiempo de entrenamiento ya que demoro tan solo 16s a comparación de Eigen Faces y Fisher Faces que demoraron 1 hora con 54 minutos y 2 segundos y 1 hora con 44 minutos y 55 segundos respectivamente; del mismo modo la eficiencia de dicho algoritmo fue superior dando como resultado el 96.40 % de efectividad contra 95.60 % y 94.40 % de efectividad respectivamente, como se puede apreciar en la tabla 3.2.
- En el trabajo presentado por (E., 2021) se concluyó de forma similar ya que, al implementar el sistema interno propuesto no se pasa mucho tiempo pidiendo que cada persona que ingrese al condominio y/o edificio de uso residencial tenga que identificarse puesto que, con el reconocimiento facial ya sabremos quien es la persona que ingrese a dicho resinto. Por otro lado, el desarrollo en la detección de rostros es el mismo ya que se uso el modelo de Haar Cascades.
- En el trabajo presentado por (L., 2021) se llegó a una misma conclusión ya que, al detectar el ingreso de un intruso a un condominio y/o edificio de uso residencial, este alertará al personal de seguridad o cuidante que está haciendo uso de este sistema; por otro lado, se planteo el envio de notificaciones por correo al usuario (solo planteamiento no se implemento).

Conclusiones

1. Se ha desarrollado y puesto en marcha un sistema interno semi-automático que aborda la carencia de software avanzado en cámaras convencionales. Este sistema optimiza la capacidad de las cámaras convencionales al incorporar funciones de reconocimiento facial y detección de intrusos. La integración con una base de datos eficiente asegura un rendimiento óptimo, mientras que el diseño escalable del sistema permite su evolución y adaptación futura según sea necesario.

2. Se diseñó e implementó una base de datos que gestiona de manera eficaz la información crítica del sistema. Para evitar la sobrecarga de la base de datos, los rostros detectados se almacenan directamente en el sistema, optimizando así su rendimiento. Este enfoque no solo mejora la funcionalidad actual del sistema, sino que también facilita su escalabilidad y adaptación a nuevas características en el futuro.

3. Se implementaron y evaluaron tres algoritmos de reconocimiento facial: Eigen Faces, Fisher Faces y LBPH Faces. Tras comparar su desempeño, se determinó que el algoritmo LBPH Faces es el más preciso, alcanzando una precisión del 96.40 %, superando a Eigen Faces y Fisher Faces, que lograron 95.60 % y 94.40 % respectivamente, como se puede apreciar en la tabla 3.1. Del mismo modo se realizo con tres algoritmos de detección de rostros: librería OpenCV con DNN, librería OpenCV con modelo de Haar Cascades y la librería de MediaPipe con Face Detection. Al finalizar la comparación entre la eficiencia de estos 3 algoritmos se vio que una de estas resaltó por encima de las otras, dando así a la librería de OpenCV con el modelo de Haar Cascades, como el algoritmo más preciso, alcanzando una precisión del 98.71 %, superando a la librería de OpenCV con DNN y a la librería de MediaPipe con Face Detection, que lograron 98.14 % y 96.85 % respectivamente, como se puede apreciar en la tabla 3.2.

4. Se logró integrar una variedad de cámaras convencionales al sistema, incluidas cámaras integradas en laptops, webcams, cámaras analógicas conectadas por USB y cámaras de dispositivos móviles mediante aplicaciones como DroidCam. Además, se incorporaron cámaras IP y WiFi a través del protocolo RTSP. La integración exitosa con estos diferentes tipos de cámaras demuestra la compatibilidad del sistema con una amplia gama de equipos disponibles en el mercado, aunque algunas marcas como EzViz no fueron compatibles. El sistema ha demostrado ser eficaz en la detección de intrusos y en el almacenamiento eficiente de información relevante de los residentes.

Recomendaciones

1. Seleccionar conjuntos de datos amplios y representativos que incluyan una variedad de condiciones y escenarios típicos de detección de intrusos en condominios. Esto asegurará que el sistema de reconocimiento facial sea lo suficientemente robusto para manejar diversas situaciones de manera efectiva.

2. Integrar múltiples capas de seguridad, combinando el reconocimiento facial con tecnologías como el reconocimiento de voz y la detección de movimiento. Esta integración permitirá desarrollar un sistema de seguridad más completo y sólido para condominios y otros entornos residenciales.

3. Llevar a cabo estudios a largo plazo que, evalúen la efectividad sostenida del sistema de reconocimiento facial en la identificación y prevención de intrusos en condominios y/o edificios de uso residencial. Esto permitirá detectar posibles variaciones en el rendimiento del sistema y aplicar mejoras continuas.

4. Abordar de manera adecuada las cuestiones éticas y de privacidad relacionadas con la implementación de sistemas de reconocimiento facial en entornos residenciales. Es necesario establecer políticas claras y transparentes sobre el uso y almacenamiento de datos biométricos, además de obtener el consentimiento informado de los residentes.

5. Desarrollar algoritmos de alerta temprana que detecten personas con rostros parcialmente o completamente cubiertos por objetos, al intentar ingresar a un condominio o edificio residencial. Estos algoritmos ayudarán a identificar posibles amenazas y mejorar la seguridad general.

6. Desarrollar el sistema interno como una plataforma web, lo que permitirá que múltiples propiedades o condominios puedan utilizarlo de manera conjunta. Esto facilitará la compartición de datos de posibles sospechosos entre distintas propiedades, mejorando la seguridad a través de la colaboración entre los diferentes usuarios del sistema.

Bibliografía

Acacio, M. G. (2019). CÁmaras de vigilancia: CaracterÍsticas y ventajas. url: https://www.acacioseguridad.com/camaras-de-vigilancia/.

Amazon (2024). Amazon.com, inc. o sus afiliados. url: https://www.amazon.com/-/es/ref=nav-logo.

Atlassian (2023). Qué es scrum y cómo empezar. url: https://www.atlassian.com/es/agile/scrum.

Briones Gárate, E. A. (2020). Sistema web de reconocimiento facial para control de acceso biométrico, utilizando inteligencia artificial. url: http://www.dspace.espol.edu.ec/handle/123456789/50333.

Córtes Martínez, C. S. and Vásquez Bohórquez, D. F. (2021). Sistema de detección, extracción y reconocimiento de rostros en escenas de máximo 4 personas, para aplicaciones de videovigilancia residencial utilizando herramientas de software libre, en lugares cerrados. url: http://hdl.handle.net/11349/26732.

Domínguez Pavón, S. (2017). Reconocimiento facial mediante el análisis de componentes principales (pca). url: https://idus.us.es/handle/11441/66514.

E., A. R. (2021). Influencia de un sistema con reconocimiento facial y mediciÓn de temperatura en el control de acceso de participantes del programa trabaja perÚ en el distrito de talavera. url: http://repositorio.unajma.edu.pe/handle/20.500.14168/666.

Engineering, T. (2023). ¿qué tipos de cámaras de seguridad y videovigilancia existen? url: http://www.tipengineer.com/categoria-blog/tecnologia/que-tipos-de-camaras-de-seguridad-y-videovigilancia-existen/.

Fernández Collado, C. and Baptista Lucio, P. (2014). *Metodología de la Investigación*. México D.F.

Google (2023). Face detection guide. url: https://developers.google.com/mediapipe/solutions-/vision/face-detector/.

Heredia Salazar, C. V. and Rea Rodríguez, D. A. (2022). Diseño de un sistema de detección facial utilizando cámaras ip para el reconocimiento de individuos en la cercanía de residencias familiares. url: http://dspace.ups.edu.ec/handle/123456789/23050.

Inc., L. H. (2024). The php framework for web artisans. url: https://laravel.com/.

Inc., S. I. (2023). Inteligencia artificial: Qué es y por qué importa. url: https://www.sas.com/espe/insights/analytics/what-is-artificial-intelligence.html.

L., S. D. (2021). Sistema inteligente de seguriad biométrica usando la iot para la alerta de robos en las residencias de villa maría del triunfo. url: http://repositorio.untels.edu.pe/jspui/handle/123456789/779.

nimble Humanize Work (2023). Desarrollo iterativo e incremental. url: https://www.nimblework.com/es/agile/desarrollo-iterativo-e-incremental/.

Oracle (2023). ¿qué es la ia? conoce la inteligencia artificial. url: https://www.oracle.com/pe/artificial-intelligence/what-is-ai/.

Ottado, G. (2010). Reconocimiento de caras: Eigenfaces y fisherfaces. url: https://eva.fing.edu.uy/file.php/514/ARCHIVO/2010/TrabajosFinales2010/informe-final-ottado.pdf.

OVACEN (2023). Cámaras de seguridad: Tipos, consejos y cuál comprar para casa. url: https://ovacen.com/camaras-de-seguridad/.

Pérez Porto, J. and María, M. (2022). Condominio - qué es, en el derecho, definición y concepto. url: https://definicion.de/condominio/. [Actualizado el 9 de junio de 2022].

RecFaces (2023). Detección de la cara: ¿qué es y cómo funciona esto tech. url: https://recfaces.com/es/articles/deteccion-de-la-cara-que-es-y-como-funciona-esto-tech.

Rosebrock, A. (2018). Detección de rostros con opencv y aprendizaje profundo. url: https://pyimagesearch.com/2018/02/26/face-detection-with-opencv-and-deep-learning/.

Seguridad, P. (2022). Cámaras con reconocimiento facial: ¿por qué deberías implementarlas en tu negocio? url: https://www.protek.com.py/novedades/camaras-con-reconocimiento-facial/.

Sotaquirá, M. (2020). Detección de rostros con machine learning. url: https://www.codificandobits.com/blog/deteccion-de-rostros-machine-learning/.

Thales (2023). 6 principios rectores para el reconocimiento facial. url: https://www.thalesgroup.com/es/countries/americas/latin-america/dis/gobierno/inspiracion/reconocimiento-facial-reglamentacion.

Verdeguer Valderrama, D. C. (2022). Diseño e implementación de un sistema de identificación de personas para la seguridad de los accesos a condominios, basado en el algoritmo de reconocimiento facial lbph faces. url: https://hdl.handle.net/11537/30258.