UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TITULO DEL TRABAJO DE TITULACIÓN

IMPLEMENTACIÓN DE UN PROTOTIPO PARA LA DETECCIÓN DE ARMAS DE FUEGO EN EL LABORATORIO DE INFORMÁTICA.

AUTOR

CEDEÑO VILLON RICHARD JOSÉ

EXAMEN COMPLEXIVO

TUTOR

Ing. CARLOS EFRAÍN SÁNCHEZ LEÓN

SANTA ELENA, ECUADOR

AÑO 2025

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

TRIBUNAL DE SUSTENTACIÓN

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

Ing. Carlos Sánchez León. Mgt
TUTOR

Ing. Mónica Jaramillo Infante. Mgt
DOCENTE ESPECIALISTA

Ing. Marjorie Coronel Suárez. Mgt.
DOCENTE GUÍA UIC

UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **Cedeño Villón Richard Jose**, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

La Libertad, a los 20 días del mes de junio del año 2025

TUTOR

Firmado electrónicamente por:
CARLOS EFRAIN
SANCHEZ LEON
Validar únicamente con FirmaEC

Ing. Carlos Efraín Sánchez León

UPSE

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Cedeño Villón Richard Jose**

DECLARO QUE:

El trabajo de Titulación, IMPLEMENTACIÓN DE UN PROTOTIPO PARA LA DETECCIÓN DE ARMAS DE FUEGO EN EL LABORATORIO DE INFORMÁTICA, previo a la obtención del título en Ingeniero en Tecnologías de la Información, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

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

La Libertad, a los 20 días del mes de junio del año 2025

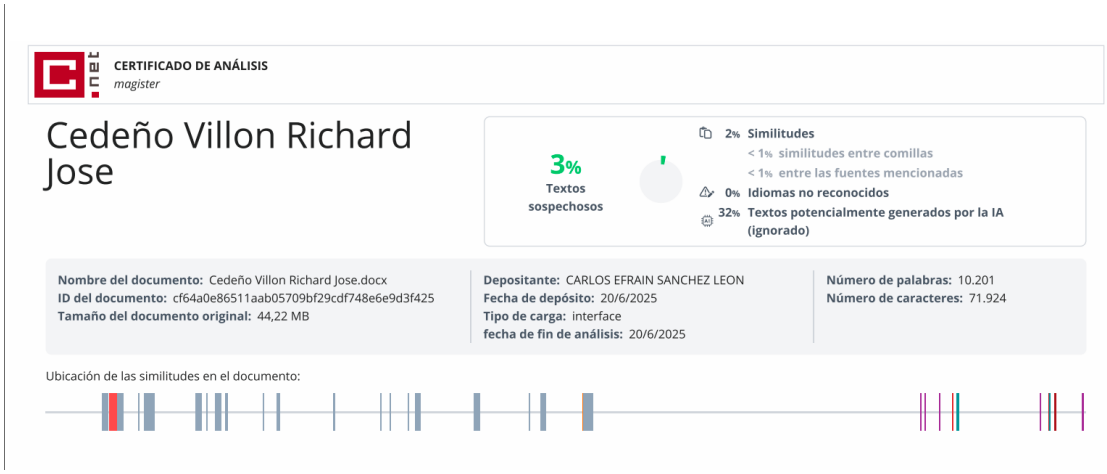
EL AUTOR

Richard Jose Cedeño Villón

UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado **IMPLEMENTACIÓN DE UN PROTOTIPO PARA LA DETECCIÓN DE ARMAS DE FUEGO EN EL LABORATORIO DE INFORMÁTICA**, presentado por el estudiante, Cedeño Villón Richard Jose fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 3%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



CERTIFICADO DE ANÁLISIS
magister

Cedeño Villon Richard Jose

3% Textos sospechosos

2% Similitudes
< 1% similitudes entre comillas
< 1% entre las fuentes mencionadas
0% Idiomas no reconocidos
32% Textos potencialmente generados por la IA (ignorado)

Nombre del documento: Cedeño Villon Richard Jose.docx
ID del documento: cf64a0e86511aab05709bf29cdf748e6e9d3f425
Tamaño del documento original: 44,22 MB

Depositante: CARLOS EFRAIN SANCHEZ LEON
Fecha de depósito: 20/6/2025
Tipo de carga: interface
fecha de fin de análisis: 20/6/2025

Número de palabras: 10.201
Número de caracteres: 71.924

Ubicación de las similitudes en el documento:

TUTOR

Firmado electrónicamente por:
CARLOS EFRAIN SANCHEZ LEON
Validar únicamente con FirmaEC

Ing. Carlos Efraín Sánchez León

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Cedeño Villón Richard Jose**

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

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

Santa Elena, a los 20 días del mes de junio del año 2025

EL AUTOR

Richard Jose Cedeño Villón

AGRADECIMIENTO

Expreso mi más sincero agradecimiento a todas las personas que han sido parte de mi vida durante el camino de mi formación académica y profesional.

En especial, agradezco profundamente a mi familia, cuyo amor, comprensión y apoyo incondicional fueron pilares fundamentales para mantenerme firme ante los desafíos y alcanzar esta meta tan importante.

Extiendo también mi gratitud al Ing. Carlos Sánchez León, quien, como tutor de este trabajo, me brindó su guía, conocimientos y experiencia con disposición y compromiso, permitiéndome desarrollar este proyecto con responsabilidad y claridad.

A todos quienes aportaron, directa o indirectamente, a la realización de esta etapa, muchas gracias.

Richard Jose, Cedeño Villón

DEDICATORIA

Dedico este proyecto a quienes han sido mi fuerza, mi guía y mi inspiración a lo largo de este camino.

A mis padres, *Janina* y *José*, por su amor incondicional, su ejemplo de esfuerzo y su apoyo constante en cada paso de mi formación.

A mis verdaderos amigos, que con su compañía y palabras de aliento me ayudaron a mantenerme firme incluso en los momentos más difíciles.

Y, sobre todo, dedico este logro a mi hija *Charlotte*, mi mayor motivo y motor de vida. Cada paso que doy es por ella y para ella, con la esperanza de construir un futuro del que pueda sentirse orgullosa.

A Dios, por darme la fortaleza, la salud y las oportunidades para llegar hasta aquí.

Richard Jose, Cedeño Villón

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente proyecto desarrolla un sistema inteligente para la detección de armas de fuego y reconocimiento facial en tiempo real, implementado en el laboratorio de informática de la Universidad Estatal Península de Santa Elena. El objetivo principal es fortalecer la seguridad mediante la identificación automática de personas y la alerta inmediata ante la presencia de armas. Se utilizó un enfoque experimental basado en visión por computador con los modelos YOLOv11 para detección de objetos y ArcFace para reconocimiento facial. El sistema fue programado en Python, con una interfaz gráfica desarrollada en customtkinter y conexión a una base de datos MySQL para registrar eventos. Los resultados muestran una alta efectividad en la detección y una correcta identificación de individuos registrados, además de la generación automática de notificaciones por Telegram. Se concluye que la solución propuesta es viable, funcional y contribuye significativamente a la vigilancia automatizada en espacios académicos.

Palabras clave: Seguridad, visión artificial, reconocimiento facial, detección de armas.

ABSTRACT

This project develops an intelligent system for the detection of firearms and facial recognition in real time, implemented in the computer laboratory of the Peninsula St. Helena State University. The main objective is to strengthen security through automatic identification of people and immediate alert to the presence of weapons. An experimental approach based on computer vision was used with YOLOv11 models for object detection and ArcFace for facial recognition. The system was programmed in Python, with a graphical interface developed in customtkinter and connection to a MySQL database to record events. The results show a high effectiveness in the detection and correct identification of registered individuals, in addition to the automatic generation of notifications via Telegram. It is concluded that the proposed solution is viable, functional and contributes significantly to automated surveillance in academic spaces.

Keywords: Security, artificial vision, facial recognition, weapons detection.

INTRODUCCIÓN

La seguridad en entornos educativos ha cobrado un papel protagónico en los últimos años debido al incremento de situaciones de riesgo que comprometen la integridad de estudiantes, docentes y personal administrativo. Incidentes como el ingreso de personas no autorizadas, robos, y en casos extremos, la presencia de armas de fuego, han generado una creciente necesidad de adoptar medidas tecnológicas que permitan una vigilancia más precisa, eficiente y en tiempo real. Frente a esta realidad, el uso de herramientas basadas en inteligencia artificial, específicamente visión por computador se ha consolidado como una alternativa efectiva y moderna para abordar los desafíos de seguridad institucional.

En este contexto, el presente proyecto tiene como objetivo principal el desarrollo de un sistema inteligente para la detección de armas de fuego y el reconocimiento facial en tiempo real, aplicado en el laboratorio de informática de la Universidad Estatal Península de Santa Elena. Esta solución tecnológica busca automatizar los procesos de vigilancia mediante la integración de modelos de aprendizaje profundo, interfaces gráficas intuitivas, bases de datos estructuradas y canales de notificación instantánea.

El sistema propuesto se fundamenta en una arquitectura modular ejecutada localmente. Se utilizan modelos de última generación como YOLOv11, entrenado para detectar armas de fuego en imágenes o secuencias de video, y ArcFace, modelo especializado en reconocimiento facial por medio de embeddings vectoriales. Ambos modelos son integrados dentro de una aplicación construida en Python, que cuenta con una interfaz gráfica desarrollada con la librería customtkinter, permitiendo al usuario visualizar el flujo de video en tiempo real, alternar entre fuentes de entrada y recibir resultados inmediatos.

Además, se implementa una base de datos relacional en MySQL, diseñada para almacenar información de las personas registradas, así como los eventos relevantes generados por el sistema (detecciones, alertas, identificaciones). Este almacenamiento estructurado permite consultar, auditar y gestionar los registros de manera eficiente. El sistema también cuenta con un componente de notificación remota, que envía alertas a través de

Telegram cada vez que se detecta un arma o una persona no registrada, contribuyendo así a una respuesta rápida y preventiva ante posibles amenazas.
El desarrollo de este sistema responde no solo a una necesidad técnica, sino también social. La propuesta se alinea con los principios de seguridad, prevención y vigilancia inteligente, y está pensada como una herramienta de bajo costo y fácil implementación en espacios académicos u organizacionales que carezcan de sistemas de seguridad avanzados. Su diseño escalable y adaptable permite que pueda ser replicado o mejorado en función de las necesidades de cada entorno.

En conclusión, este proyecto combina de forma efectiva los avances en visión artificial, el procesamiento en tiempo real y la gestión automatizada de información, aportando una solución práctica y funcional que contribuye significativamente a la construcción de entornos educativos más seguros, modernos y resilientes frente a los desafíos actuales.

CAPÍTULO 1. FUNDAMENTACIÓN

1.1. Antecedentes

Ecuador atraviesa un periodo complejo en materia de seguridad. Reportes oficiales indican un aumento en los hechos de violencia en los primeros meses del año, lo que ha generado preocupación en distintos sectores [1]. En respuesta, el Gobierno de Guillermo Lasso implementó medidas como la autorización para la tenencia y porte de armas con fines de defensa personal. Sin embargo, esta decisión ha sido cuestionada por la academia y organizaciones de derechos humanos. Al mismo tiempo, las autoridades han intensificado los operativos para incautar armas ilegales, buscando frenar esta problemática [2].

La Universidad Estatal Península de Santa Elena cuyo campus matriz está ubicado en la ciudad de La Libertad es la única Entidad de Educación Superior de la provincia y por lo tanto la concurrencia de estudiantes ha crecido constantemente desde su inauguración en el año 1998. La UPSE cuenta con 7 facultades y una oferta académica de 29 carreras que abarcan amplios campos laborales tales como tecnología, biología, educación, civil entre otros [3].

La misión es formar profesionales que aportan al desarrollo sostenible, contribuye a la solución de los problemas de la comunidad y promueve la cultura, y su visión es ser reconocida por su calidad académica, impacto de sus investigaciones y su aporte al desarrollo de la sociedad. Uno de sus principios es la igualdad de oportunidades, que consiste en garantizar a todos los actores del Sistema de Educación Superior las mismas posibilidades en el acceso, permanencia, movilidad y egreso del sistema, sin discriminación de género, credo, orientación sexual, etnia, cultura, preferencia política, condición socioeconómica o discapacidad [4].

La autorización del porte de armas de uso civil y los altos niveles de criminalidad ha obligado a las universidades del Ecuador a replantear sus acciones. Esto preocupa a las autoridades de las 71 universidades y escuelas politécnicas, públicas y privadas. No han faltado los comunicados de rechazo a la medida. Además,

ratificaron que está prohibido el ingreso de armas letales a sus campus universitarios. Sin embargo, es algo difícil de controlar en cada uno de los estudiantes ya que algunos de los rectores recalcan lo siguiente: ‘sabemos que existen ciertos estudiantes que, lamentablemente, andan en situaciones indebidas’ [5].

Diversos sectores han manifestado que el porte de armas podría incrementar el tráfico de estas, así como la violencia intrafamiliar, los femicidios y los suicidios [6]. En el ámbito universitario, se ha señalado la necesidad de contar con el apoyo de la Policía Nacional para fortalecer la seguridad ciudadana y el orden público. Además, se enfatiza la importancia de un trabajo conjunto orientado a mejorar la educación y la prevención de la delincuencia [7].

La fata de software que permitan la detección de armas en vídeos es una problemática que aumenta diariamente, ya que afecta a la seguridad de las personas. Si bien es un problema importante, no se encuentran muchos programas que se dediquen exclusivamente a ello. En la universidad Complutense de Madrid Se ha conseguido un programa que detecta pistolas en vídeos con una precisión del 70 %, una efectividad del 72 % y una exactitud del 71 %, mediante el uso de técnicas de inteligencia artificial para procesar información masiva, de esta forma se podría detectar un arma en una cámara y alertar a las autoridades de donde se halla el sospechoso, que tipo de arma porta, quien es, etc [8].

En la mayoría de los delitos se ha incrementado el uso de armas de fuego, que en algunos casos ha llegado a terminar con la vida de las víctimas. De acuerdo con los informes de la policía las armas más confiscadas son las armas de mano (revólveres y pistolas), ya que presentan una facilidad para portarlas y se pueden accionar con una sola mano. Los bajos índices de seguridad en la ciudad han hecho necesario la puesta en marcha de planes de choque y el uso herramientas que ayuden a mitigar las falencias por parte de las autoridades encargadas de la seguridad en la ciudad. En los últimos años, desde el campo de inteligencia artificial se han propuesto nuevas formas de abarcar las tareas de detección de objetos [9].

En Ecuador de acuerdo con el Instituto Nacional de Estadísticas y Censos, en el año 2018 la tendencia de robo a unidades económicas a crecido de manera considerable, esto dado a varios factores siendo uno de ellos la baja economía del país, dándose un total de aproximadamente 450 robos cada mes y un total de aproximadamente 5400 cada año esperándose un aumento para los próximos años, debido a que los sistemas de monitoreo en donde un personal observa son muy costosos y de baja eficiencia las entidades comerciales han optado por no utilizarlos dejando así más vulnerable la integridad del personal y de sus bienes [10].

Ante el incremento de actos delictivos reportados en la provincia de Santa Elena y la alta concurrencia de estudiantes y docentes en el Laboratorio de Informática de la Facultad de Sistemas y Telecomunicaciones de la UPSE, se identifica la urgente necesidad de un sistema de seguridad inteligente. En este contexto, el desarrollo de un prototipo de visión artificial mediante inteligencia artificial es una solución adecuada. El sistema propuesto permite la detección automática de armas en video, el reconocimiento facial de personas autorizadas (docentes y estudiantes) en tiempo real, el almacenamiento de eventos críticos en una base de datos y el envío inmediato de alertas al personal de seguridad. Así, se fortalece la protección de la comunidad universitaria mediante un monitoreo eficiente y automatizado.

1.2. Descripción del Proyecto

La función principal de este prototipo es identificar si las personas que ingresan al laboratorio de informática de la Universidad Estatal Península de Santa Elena portan armas de fuego y si sus rostros pertenecen al plantel, considerando la situación actual del país.

Para ello, el proyecto implementará técnicas de inteligencia artificial, específicamente visión por computadora y aprendizaje profundo, utilizando herramientas de código abierto.

Contará con el desarrollo de los siguientes módulos:

- Módulo de entrenamiento:
    - Crear el dataset.

    - División del dataset en dos folders:
        - Folder 1: 70% Para el entrenamiento.
        - Folder 2: 20% Para validación.
        - Folder 3: 10% Para test.
    - Entrenar la red neuronal las veces que sean necesarias para una mejor precisión.
- Módulo de detección:
    - Ingreso de personas al laboratorio.
    - Detección de objeto (arma) mediante visión por computadora
    - Captura mediante las cámaras de vigilancia.
    - Envío de datos al departamento encargado.
- Módulo de reconocimiento:
    - Detecta la imagen.
    - Localiza la imagen.
    - Clasifica la imagen.
- Módulo de notificación:
    - Notifica al departamento encargado.
    - Envía captura de imagen de la persona que porta arma de fuego.
    - Envía ubicación de la cámara.

Para la implementación del siguiente proyecto, se utilizarán las siguientes herramientas tecnológicas:

Plataforma	**Descripción**
Lenguaje de programación	**Python:** Es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos. La elegante sintaxis de Python y su tipado dinámico, junto a su naturaleza interpretada lo convierten en un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en muchas áreas, para la mayoría de las plataformas [11].

Herramientas de desarrollo	**Google Colab:** Es una herramienta en línea de Google que permite ejecutar código Python en la nube, sin necesidad de configuración local, con acceso a recursos como GPUs y TPUs, y es ampliamente utilizada en el desarrollo de proyectos de inteligencia artificial y aprendizaje automático [12].
Bibliotecas	**PyTorch:** Es una biblioteca de tensores optimizada para el aprendizaje profundo mediante GPU y CPU [13]. **Ultralytics:** Crea modelos de vanguardia y de última generación (SOTA) YOLO construidos sobre años de investigación fundacional en la visión por computadora y la IA. Actualizados constantemente para el rendimiento y la flexibilidad, nuestros modelos son rápidos, precisos y fáciles de usar. Exaltan en la detección de objetos, el seguimiento, la segmentación de la instancia, la clasificación de imágenes y las tareas de estimación de posean [14].
Modelos	**Yolov11:** YOLO11 es la última iteración de la serie Ultralytics YOLO de detectores de objetos en tiempo real, que redefine lo que es posible con una precisión, velocidad y eficacia de vanguardia. Basándose en los impresionantes avances de las versiones anteriores de YOLO, YOLO11 introduce mejoras significativas en la arquitectura y los métodos de entrenamiento, lo que lo convierte en una opción versátil para una amplia gama de tareas de visión por ordenador [15]. **ArcFace IR-SE-50:** Sujeta entrenamiento distribuido y escaso con múltiples ejemplos de entrenamiento distribuidos, incluyendo varias técnicas de ahorro de

	memoria, como entrenamiento de precisión mixta y control de gradiente [16].
Librería	**OpenCv:** OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión de computadora y aprendizaje automático de código abierto. OpenCV se construyó para proporcionar una infraestructura común para aplicaciones de visión por computadora y acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto con licencia Apache 2, OpenCV facilita a las empresas el uso y modifica el código [17].
Entorno de desarrollo	**Neovim:** Neovim es un editor de texto basado en Vim diseñado para ser extensible y fácil de usar, con el fin de fomentar nuevas aplicaciones y contribuciones [18].
Contenedores	**Docker:** Docker es una plataforma de código abierto diseñada para desarrollar, enviar y ejecutar aplicaciones dentro de contenedores. Los contenedores permiten empaquetar el software con todo lo necesario para su ejecución, incluyendo bibliotecas, dependencias y configuraciones, garantizando que se ejecute de manera consistente en diferentes entornos. Docker automatiza el despliegue de aplicaciones en entornos ligeros y portables, facilitando la integración continua y la entrega continua (CI/CD) en entornos de desarrollo modernos [19].

Table 1: Herramientas de software para la realización del proyecto

En el marco de los ajustes curriculares de la Carrera de Tecnología de la Información de la UPSE, aprobados mediante la Resolución RCS-SO-01-07-2025 de la sesión ordinaria N.º 01 del año 2025, este proyecto se enmarca dentro de la línea de investigación en “Desarrollo de Software”, específicamente en el uso de

tecnologías emergentes como visión artificial e inteligencia artificial aplicada para la seguridad en entornos educativos. Mediante la integración de protocolos de adquisición (cámaras, imágenes, videos), procesamiento de datos mediante redes neuronales (ArcFace, YOLOv11), almacenamiento en bases de datos y notificaciones inteligentes por Telegram, el sistema aporta al perfil formativo descrito en la resolución, que promueve la innovación tecnológica en aplicaciones reales con impacto institucional [20].

1.3. Objetivos del proyecto

1.3.1. Objetivo General

- Implementación de un prototipo mediante el uso de visión artificial, para la detección de armas de fuego en el laboratorio de informática.

1.3.2. Objetivos Específicos

- Integrar un modelo de reconocimiento facial basado en PyTorch para la detección en tiempo real, mediante el diseño de un dataset clasificado en estudiantes y docentes que permita extraer las características necesarias para realizar el reconocimiento facial.
- Diseñar un dataset con imágenes de armas de fuego para el entrenamiento de la red neuronal convolucional.
- Diseñar una red neuronal convolucional para el análisis y reconocimiento de armas de fuego.
- Evaluar el desempeño de efectividad de la detección de armas mediante pruebas de funcionamiento.

1.4. Justificación del Proyecto

La tarea de Detección de objetos en imágenes fue impulsora de mejora tanto en redes neuronales convolucionales como en la arquitectura general utilizada poniendo a prueba el valor real del deeplearning, entrelazando redes con funciones específicas. Los logros obtenidos son enormes, de gran aplicación y como vemos sigue siendo un campo en desarrollo, en donde grandes empresas como Google y Facebook siguen innovando con nuevas propuestas. Las aplicaciones que tiene la

detección de imágenes van desde seguridad, conducción de coches autónomos hasta salud [21].

Actualmente los sistemas de detección son de gran utilidad para la sociedad especialmente en temas de seguridad. Gracias a los avances en la inteligencia artificial ahora las cámaras de vigilancia pueden ver y reconocer diferentes objetos del entorno, entre estos, las armas de fuego, todo este proceso es realizado en tiempo real en cuestión de segundos [22].

El contexto de creciente inseguridad en la provincia de Santa Elena y la alta concurrencia de estudiantes y docentes en el Laboratorio de Informática de la Facultad de Sistemas y Telecomunicaciones de la UPSE exigen soluciones de vigilancia más avanzadas. La falta de software especializado para la detección automática de armas en video y reconocimiento facial en tiempo real agrava esta situación, exponiendo a la comunidad académica a potenciales riesgos. Frente a esta realidad, el empleo de técnicas avanzadas de visión artificial e inteligencia artificial se convierte en una herramienta indispensable para fortalecer la seguridad institucional y proteger el entorno educativo.

Mediante la implementación de esta solución tecnológica que incorpora detección automática de armas, reconocimiento facial en tiempo real, registro de eventos en base de datos y envío de alertas al personal de seguridad el proyecto fortalece los procesos institucionales de vigilancia y gestión de riesgos, incrementando la eficiencia operativa y respaldo tecnológico de la seguridad dentro del campus universitario. Este sistema no solo ofrece una solución reactiva frente a amenazas, sino también un mecanismo preventivo, al proveer monitoreo continuado y trazabilidad de eventos críticos, lo que permite una reacción inmediata y documentada en situaciones potencialmente peligrosas.

El presente proyecto se encuentra alineado con el Eje Social del Plan Nacional de Desarrollo para el Nuevo Ecuador 2024–2025, en particular con el Objetivo 3, que busca “Garantizar la seguridad integral, la paz ciudadana y transformar el sistema

de justicia respetando los derechos humanos" [23]. Además, se integra dentro del Eje Institucional, bajo el Objetivo 9, orientado a "propender la construcción de un Estado eficiente, transparente y orientado al bienestar social" [23].

1.5. Alcance del Proyecto

El desarrollo del proyecto está dirigido para el laboratorio de informática de la Universidad Estatal Península de Santa Elena, mediante la implementación del prototipo para la detección de armas de fuego y reconocimiento facial, para proteger la seguridad de los estudiantes y docentes que hacen uso del laboratorio por la inseguridad que se vive hoy en día en el País.

Se usarán los siguientes módulos:

- **Módulo de entrenamiento:** Se crea el dataset el cual estará divido en tres folders, el folder 1 se usará el 70% para el entrenamiento, el folder 2 el 20% para validación, y el folder 3 con el 10% restante, para el test. Entrenando la red neuronal las veces que sean necesarias para mejorar la precisión de la detección de armas de fuego y alcanzar el porcentaje deseado en precisión.
- **Módulo de detección:** Haciendo uso del modelo ya entrenado, para la detección de armas. El sistema verifica en tiempo real si existe un arma de fuego, en caso de que exista, de manera automática se enviará una notificación vía Telegram al departamento encargado, con los detalles correspondientes.
- **Módulo de reconocimiento:** Se realiza el reconocimiento y clasificación en tiempo real de la imagen, para verificar si la persona que ingresa al laboratorio es un estudiante, docente o no pertenece al plantel.
- **Módulo de notificación:** El encargado de seguridad del laboratorio será notificado mediante un mensaje en la aplicación de Telegram, este mensaje contendrá una captura de la imagen de la persona que porta el arma de fuego (en caso de poseer una), o si la persona que ingresa al laboratorio no pertenece al plantel.

CAPÍTULO 2. MARCO TEÓRICO Y METODLOGÍA DEL PROYECTO

2.1. Marco conceptual

2.2.1. Inteligencia artificial

La Inteligencia Artificial (IA) se refiere a sistemas computacionales diseñados para realizar tareas que normalmente requieren inteligencia humana, como el reconocimiento de patrones, la toma de decisiones y el aprendizaje. Estos sistemas pueden analizar datos complejos y adaptarse a nuevas entradas, mejorando su rendimiento con el tiempo. La IA se ha convertido en una herramienta esencial en diversas industrias, desde la medicina hasta la seguridad, debido a su capacidad para procesar grandes volúmenes de información de manera eficiente [24].

2.2.1.1. Aplicaciones en seguridad

La IA en seguridad cibernética es un concepto revolucionario que está cambiando la forma en que vemos y manejamos las amenazas cibernéticas. El uso de IA va más allá de la mera detección de amenazas. Permite respuestas automatizadas, anticipa amenazas futuras y simplifica los procesos de seguridad. A medida que las amenazas cibernéticas siguen evolucionando, la IA en seguridad cibernética proporciona un enfoque proactivo para garantizar la seguridad y la integridad de los activos digitales [25].

2.2.2. Visión por Computadora

La visión por computadora o visión computarizada hace referencia a un grupo de tecnologías o herramientas que permiten a los equipos captar imágenes del mundo real, procesarlas y generar información a través de ellas (análisis). Dicho de otra manera, la visión por computador es una propiedad de ciertas tecnologías que permiten a los equipos computarizados “ver”. Esto ha hecho posible el diseño de maquinaria industrial y colaborativa de gran flexibilidad, capaces de tomar decisiones inteligentes con base en su entorno, a niveles imposibles de igualar por el ojo humano [26].

2.2.2.1. Detección de objetos

La Detección de Objetos es una técnica de visión por computadora que identifica y localiza instancias de objetos dentro de una imagen o secuencia de video. A diferencia de la clasificación de imágenes, que solo determina la presencia de un objeto, la detección también especifica su ubicación mediante cajas delimitadoras. Esta técnica es esencial en aplicaciones como la conducción autónoma y la vigilancia, donde es crucial reconocer y localizar múltiples objetos simultáneamente [27].

2.2.2.2. Detección de Armas de Fuego mediante Visión por Computadora

La Detección de Armas de Fuego mediante visión por computadora implica el uso de algoritmos entrenados para identificar la presencia de armas en imágenes o videos. Estos sistemas analizan características visuales específicas de las armas y pueden generar alertas en tiempo real para prevenir incidentes violentos. Su implementación es especialmente relevante en entornos como aeropuertos, escuelas y eventos públicos, donde la seguridad es prioritaria [28].

2.2.2.3. Reconocimiento facial

El Reconocimiento Facial es una tecnología de visión por computadora que identifica o verifica la identidad de individuos analizando y comparando patrones basados en sus características faciales. Funciona detectando un rostro en una imagen o video, extrayendo sus características distintivas y comparándolas con una base de datos de rostros conocidos. Esta tecnología se utiliza en aplicaciones de seguridad, control de acceso y autenticación de identidad [29].

2.2.3. Aprendizaje Profundo (Deep Learning)

El Aprendizaje Profundo es una subárea del aprendizaje automático que utiliza redes neuronales artificiales con múltiples capas (profundas) para modelar y analizar patrones complejos en datos. Estas redes, conocidas como redes neuronales profundas, son capaces de aprender representaciones jerárquicas de los datos, lo

que las hace especialmente efectivas en tareas como el reconocimiento de voz, procesamiento de lenguaje natural y visión por computadora [30].

2.2.3.1. Redes Neuronales Convolucionales (CNN)

Las redes neuronales son un subconjunto de aprendizaje automático, y están en el corazón de algoritmos de aprendizaje profundo. Se componen de capas de nodos, que contienen una capa de entrada, una o más capas ocultas y una capa de salida. Cada nodo se conecta a otro y tiene un peso y un umbral asociados. Si la salida de cualquier nodo individual está por encima del valor umbral especificado, se activa ese nodo, enviando datos a la siguiente capa de la red. De lo contrario, no se transmiten datos a la siguiente capa de la red [31].

2.2.3.2 Transferencia de aprendizaje

El aprendizaje por transferencia (TL) es una técnica de machine learning (ML) en la que un modelo previamente entrenado para una tarea se refina con precisión para una nueva tarea relacionada. El entrenamiento de un nuevo modelo de ML es un proceso largo e intensivo que requiere una gran cantidad de datos, potencia informática y varias iteraciones antes de que esté listo para la producción. En su lugar, las organizaciones utilizan la TL para volver a entrenar a los modelos existentes en tareas relacionadas con datos nuevos. Por ejemplo, si un modelo de machine learning puede identificar imágenes de perros, puede entrenarse para identificar gatos mediante un conjunto de imágenes más pequeño que resalte las diferencias de características entre perros y gatos [32].

2.2.4 Herramientas de Código Abierto (Open Source)

El software open source es un código diseñado de manera que sea accesible al público: todos pueden ver, modificar y distribuir el código de la forma que consideren conveniente. El software open source se desarrolla de manera descentralizada y colaborativa, así que depende de la revisión entre compañeros y la producción de la comunidad. Además, suele ser más económico, flexible y duradero que sus alternativas propietarias, ya que las encargadas de su desarrollo son las comunidades y no un solo autor o una sola empresa [33].

2.2.5 PyTorch

PyTorch es un marco de aprendizaje profundo de código abierto basado en software que se emplea para crear redes neuronales, combinando la biblioteca de machine learning(ML) de Torch con una API de alto nivel basada en Python . Su flexibilidad y facilidad de uso, entre otros beneficios, lo convirtieron en el marco de ML líder para las comunidades académicas y de investigación [34].

2.2.5.1 Beneficios de PyThorch

El uso de PyTorch puede proporcionar los siguientes beneficios:

- Ofrece a los desarrolladores una estructura fácil de aprender y de codificar basada en Python.
- Permite una depuración sencilla con herramientas populares de Python.
- Ofrece escalabilidad y está bien soportado en las principales plataformas en la nube.
- Ofrece una pequeña comunidad centrada en el código abierto.
- Exporta modelos de aprendizaje al formato estándar Open Neural Network Exchange (ONNX).
- Ofrece una interfaz fácil de usar.
- Ofrece una opción de interfaz front-end en C++.
- Incluye un amplio conjunto de potentes API que amplían la biblioteca PyTorch [35].

2.2.6. ArcFace

ArcFace es una técnica de reconocimiento facial basada en aprendizaje profundo que mejora la discriminación entre identidades al aplicar una penalización angular entre los vectores de características y sus clases correspondientes. Fue propuesta para resolver las limitaciones de métodos anteriores, logrando una mayor precisión y robustez ante variaciones de iluminación, expresión facial y ángulo. ArcFace emplea una función de pérdida conocida como Additive Angular Margin Loss, la cual refuerza la separación entre diferentes identidades en el espacio de características [36].

2.2.6.1. Qué es y cómo funciona

ArcFace funciona generando vectores de características (embeddings) para cada rostro, los cuales son comparados mediante distancias angulares en un espacio métrico. Durante el entrenamiento, el modelo aprende a maximizar la distancia entre individuos distintos y a minimizarla entre imágenes de una misma persona. Esta técnica ha demostrado excelentes resultados en benchmarks públicos de reconocimiento facial y es especialmente útil en entornos donde se requiere alta precisión, como sistemas de vigilancia, control de accesos o autenticación biométrica [36].

2.2.6.2 Embeddings

Los embeddings son una técnica de procesamiento de lenguaje natural que convierte el lenguaje humano en vectores matemáticos. Estos vectores son una representación del significado subyacente de las palabras, lo que permite que las computadoras procesen el lenguaje de manera más efectiva. En otras palabras, los embeddings permiten que las palabras sean tratadas como datos y manipuladas matemáticamente. Esta técnica se utiliza ampliamente en la inteligencia artificial para tareas como el análisis de sentimiento, la clasificación de texto y la traducción automática [37].

2.2.6.3 Generación de embeddings

El proceso de creación de embeddings comienza con la construcción de un corpus, que es una colección de textos. A partir de este corpus, se crea un modelo de lenguaje que aprende a predecir palabras basándose en su contexto. Una vez que se ha entrenado el modelo, se utilizan las capas internas para generar los vectores de embeddings de las palabras. Los vectores generados por embeddings tienen varias propiedades útiles que los hacen especialmente efectivos en aplicaciones de procesamiento de lenguaje natural [37].

2.2.6.4 Estructura del dataset para entrenamiento

Para que ArcFace pueda entrenarse de forma efectiva, es necesario contar con un dataset bien estructurado. Este debe organizarse en carpetas, donde cada una represente una identidad única, y contenga varias imágenes del mismo rostro en distintas condiciones. Las imágenes deben estar etiquetadas correctamente y con buena calidad. Esta organización permite al modelo aprender a distinguir patrones relevantes de cada persona. Datasets populares como LFW, CASIA-WebFace o MS-Celeb-1M siguen este formato y son utilizados como referencia [38].

2.2.7. Telegram

Telegram es una aplicación de mensajería enfocada en la velocidad y seguridad, es súper rápida, simple y gratuita. Puedes usar Telegram en todos tus dispositivos al mismo tiempo. Tus mensajes se sincronizan a la perfección a través de cualquiera de tus teléfonos, tablets o computadoras. Telegram tiene más de 950 millones de usuarios activos mensuales y es una de las 5 apps más descargadas del mundo [39].

2.2.7.1. Telegram API (Bot API)

Esta API le permite conectar bots a nuestro sistema. Los bots de telegram son cuentas especiales que no requieren un número de teléfono adicional para configurar. Estas cuentas sirven como una interfaz para el código que se ejecuta en algún lugar de su servidor. Para usar esto, no necesitas saber nada sobre cómo funciona nuestro protocolo de cifrado MTProto. nuestro servidor intermediario manejará todo el cifrado y la comunicación con la API de Telegram para ti. Usted se comunica con este servidor a través de una interfaz HTTPS simple que ofrece una versión simplificada de la API de Telegram [40].

2.2.7.2. Envío de mensajes automatizados

El envío automatizado de mensajes a través de Telegram se realiza mediante bots, que son programas capaces de interactuar con usuarios y grupos. Estos bots pueden ser configurados para enviar alertas en tiempo real cuando se detecte un rostro no autorizado o un arma de fuego, incluyendo imágenes y hora de detección. Esta funcionalidad es esencial para responder de manera inmediata ante situaciones de riesgo, sin necesidad de una base de datos centralizada, lo que también reduce el costo y complejidad del sistema [41].

2.2. Marco teórico

2.2.1. Sistema de alarma automática de detección de armas de fuego en vídeos mediante aprendizaje profundo

A pesar de los avances tecnológicos, los sistemas actuales de vigilancia y control todavía requieren supervisión e intervención humana. Este trabajo propone un sistema automático innovador para la detección de armas de fuego en videos, pensado tanto para tareas de vigilancia como de control. En lugar de enfocarnos únicamente en detectar armas, planteamos el problema desde la perspectiva de reducir al máximo los falsos positivos. Para abordar este desafío, seguimos dos estrategias principales: i) construir un conjunto de datos clave guiado por los resultados de un clasificador profundo basado en redes neuronales convolucionales (CNN), y ii) comparar dos enfoques de detección: el de ventanas deslizantes y el basado en propuestas regionales [42].

Los mejores resultados se obtuvieron utilizando el modelo Faster R-CNN, entrenado con nuestra nueva base de datos. Este detector demostró un desempeño destacado, incluso en videos de baja calidad tomados de YouTube, funcionando de manera efectiva como sistema de alarma automática. En pruebas realizadas sobre 30 escenas, el sistema logró activar correctamente la alarma en 27 de ellas tras cinco detecciones positivas reales, y todo esto en menos de 0,2 segundos. Además, se propuso una nueva métrica llamada "Tiempo de Activación de Alarma por Intervalo" (AATpI), pensada específicamente para evaluar el rendimiento de modelos de detección en sistemas automáticos aplicados a video [42].

2.2.2. Cibernética Aplicada: Cómo el Prototipo Detecta y Responde a Amenazas

La cibernética, como ciencia del control y la comunicación en animales y máquinas, propuesta por Norbert Wiener, es fundamental en sistemas de vigilancia automática. Se basa en el principio de retroalimentación, donde el sistema ajusta su comportamiento según los datos recibidos. En el prototipo, los datos capturados por la cámara se procesan, y si se detecta una anomalía (como un arma), se genera una

acción de respuesta (alerta). Este ciclo de entrada-procesamiento-salida-retroalimentación es un claro ejemplo del modelo cibernético aplicado [43].

2.2.3. Desarrollo de un sistema de detección de armas de fuego cortas en el monitoreo de videos de cámaras de seguridad

Hoy en día, los sistemas de monitoreo con cámaras de seguridad se han vuelto una herramienta fundamental para garantizar la seguridad en negocios y otras entidades. Su funcionamiento depende en gran parte de la observación directa por parte de operadores humanos, quienes, al detectar algún incidente, lo reportan a las autoridades correspondientes para que tomen acción. Sin embargo, este tipo de sistemas todavía presentan varias limitaciones, especialmente en lo que respecta al tiempo de respuesta y a los errores derivados del cansancio o distracción del personal. Además, mantener estos sistemas puede resultar costoso, lo que los vuelve poco accesibles para muchas empresas [44].

En los últimos años, las redes neuronales convolucionales han transformado la visión por computadora, especialmente en tareas como la detección de objetos en tiempo real y la clasificación de imágenes. Estas tecnologías ya se aplican en distintos campos, como la conducción autónoma. En este proyecto se desarrolló un sistema capaz de detectar armas de fuego cortas en grabaciones de cámaras de seguridad, aprovechando el potencial de estas redes neuronales. Para ello, se construyó una base de datos con imágenes y fotogramas, tanto con armas como sin ellas, y se entrenó un modelo que luego fue evaluado con diversas métricas. El modelo se integró en un prototipo funcional de software, y los resultados fueron prometedores: obtuvo altos niveles de precisión y recall, y funcionó correctamente incluso en grabaciones complejas y nunca antes vistas por el sistema, con múltiples personas y objetos en escena [44].

2.2.3. Teoría del Reconocimiento de Patrones

El reconocimiento de patrones es una rama de la inteligencia computacional que busca identificar regularidades en los datos. Esta teoría es la base del reconocimiento facial y la detección de armas, ya que permite clasificar

información visual mediante características previamente aprendidas. Su importancia radica en su capacidad de generalización, es decir, identificar objetos o rostros no vistos antes, pero que comparten patrones similares con los entrenados. Esta teoría sustenta el uso de redes neuronales en tu sistema [45].

2.3. METODOLOGÍA DEL PROYECTO

2.3.1. Metodología de la investigación

Para la implementación del siguiente proyecto se utilizará la metodología exploratoria que se basa en la recopilación de la búsqueda de trabajos similares que se han realizado a nivel latinoamericano y mundial que ya han aplicado la inteligencia artificial para la detección de otros objetos en tiempo real con cámaras [45]. Sirviendo de ayuda para poder esclarecer semejanza y diferencias que permitan ser de utilidad en el desarrollo del proyecto propuesto ya que es un tema el cual ha sido muy poco estudiado a lo largo de este tiempo hasta la actualidad [46].

El estudio diagnostico nos ayudará a conocer el proceso que realizan los guardias del laboratorio de informática para precautelar la seguridad de los estudiantes y docentes que ingresan, permitiendo tener una perspectiva de las funciones que realizará el proyecto propuesto [47].

2.3.2. Técnicas e instrumentos de recolección de datos

Se utilizó la técnica de observación simple no estructurada en el laboratorio de informática de la Universidad Estatal Península de Santa Elena, donde se realizó el levantamiento de información que es de mucha importancia para el desarrollo del proyecto, con el propósito de comprender el proceso que realizan para salvaguardar la seguridad de quienes hacen uso del laboratorio y determinar las falencias (Anexo C).

Observación: La observación es una técnica que consiste en visualizar o captar mediante la vista, en forma sistemática, cualquier hecho, fenómeno o situación que se produzca en la naturaleza o en la sociedad, en función de unos objetivos de investigación preestablecidos [48].

a) Observación simple o no participante

Es la que se realiza cuando el investigador observa de manera neutral sin involucrarse en el medio o realidad en la que se realiza el estudio.

Usando la clasificación de observación libre o no estructurada, es la que se ejecuta en función de un objetivo, pero sin una guía prediseñada que especifique cada uno de los aspectos que deben ser observados [48].

2.3.3. Metodología de desarrollo del software

Metodología de Desarrollo Incremental

El modelo incremental es una metodología de desarrollo de software que permite construir sistemas grandes y complejos mediante la adición iterativa de nuevas funciones basadas en el análisis de los datos fuente. Se basa en la idea de añadir nuevas funciones, o "incrementos", a un sistema existente en lugar de construirlo todo desde cero. El modelo incremental consiste en que el equipo de desarrollo intenta completar cada compilación incremental lo más rápido posible. El objetivo es entregar un producto funcional poco a poco. El proceso incluye lanzamientos regulares, cada uno de los cuales representa un incremento en la funcionalidad y la calidad [49].

El modelo incremental es un proceso iterativo que ayuda a identificar y corregir defectos en las primeras etapas del desarrollo. Consta de cuatro fases principales:

Fase de requisitos y análisis

En esta fase se definieron los requerimientos funcionales y no funcionales del sistema. Se investigó sobre las técnicas y herramientas más adecuadas para realizar reconocimiento facial y detección de armas con visión artificial. También se determinó la necesidad de notificar al personal de seguridad mediante canales digitales (Telegram) y almacenar registros en una base de datos para llevar trazabilidad de los eventos.

Fase de diseño

Durante esta fase se definió la arquitectura general del sistema, tanto a nivel lógico como físico. Se planificó el uso de una red neuronal preentrenada (ArcFace con PyTorch) para extraer embeddings faciales y se estableció que dichos embeddings se usarían para entrenar un clasificador supervisado, capaz de reconocer si una persona pertenece al plantel.

También se diseñó una red neuronal convolucional con YOLOv11 entrenada específicamente para detectar armas de fuego. A nivel estructural, se definieron los siguientes componentes:

- Módulo de adquisición (entrada por cámara en vivo, imágenes o videos).
- Módulo de procesamiento (detección facial, extracción de embeddings, clasificación).
- Módulo de notificación (envío de mensajes por Telegram).
- Módulo de persistencia (registro de eventos en MySQL).
- Interfaz gráfica con customtkinter para operar el sistema de forma visual.

Fase de codificación e implementación

En esta etapa se implementaron todos los módulos del sistema. Se utilizó PyTorch para el reconocimiento facial, Ultralytics y YOLOv11 para la detección de armas, y MySQL para el almacenamiento de los registros.

Se creó un script para la extracción de embeddings faciales desde un dataset clasificado en estudiantes y docentes, y otro script para entrenar un clasificador (SVM o KNN). Además, se integraron los modelos dentro de una interfaz con customtkinter, donde se controlan las fuentes de entrada (cámara, imagen, video) y se muestran los resultados.

Se desarrolló una función para enviar notificaciones a Telegram automáticamente cuando se detecta un arma de fuego, y se implementaron funciones para insertar registros en la base de datos tanto de personas reconocidas como de alertas generadas.

Fase de prueba

En esta fase se realizaron pruebas unitarias y funcionales sobre cada módulo del sistema: reconocimiento facial, detección de armas, notificaciones, base de datos e interfaz. Se validó la capacidad del sistema para:

- Detectar rostros en diferentes condiciones de iluminación y ángulos.
- Reconocer personas previamente registradas en el dataset facial.
- Detectar armas con precisión razonable, minimizando falsos positivos.
- Enviar notificaciones en tiempo real mediante Telegram.
- Registrar cada evento en la base de datos correctamente.

También se realizaron pruebas con diferentes entradas (videos, imágenes, cámara en vivo) para asegurar la robustez del sistema ante distintos escenarios. Las observaciones de las pruebas sirvieron para realizar ajustes en los umbrales de similitud, mejorar la usabilidad de la interfaz y optimizar la eficiencia del sistema.

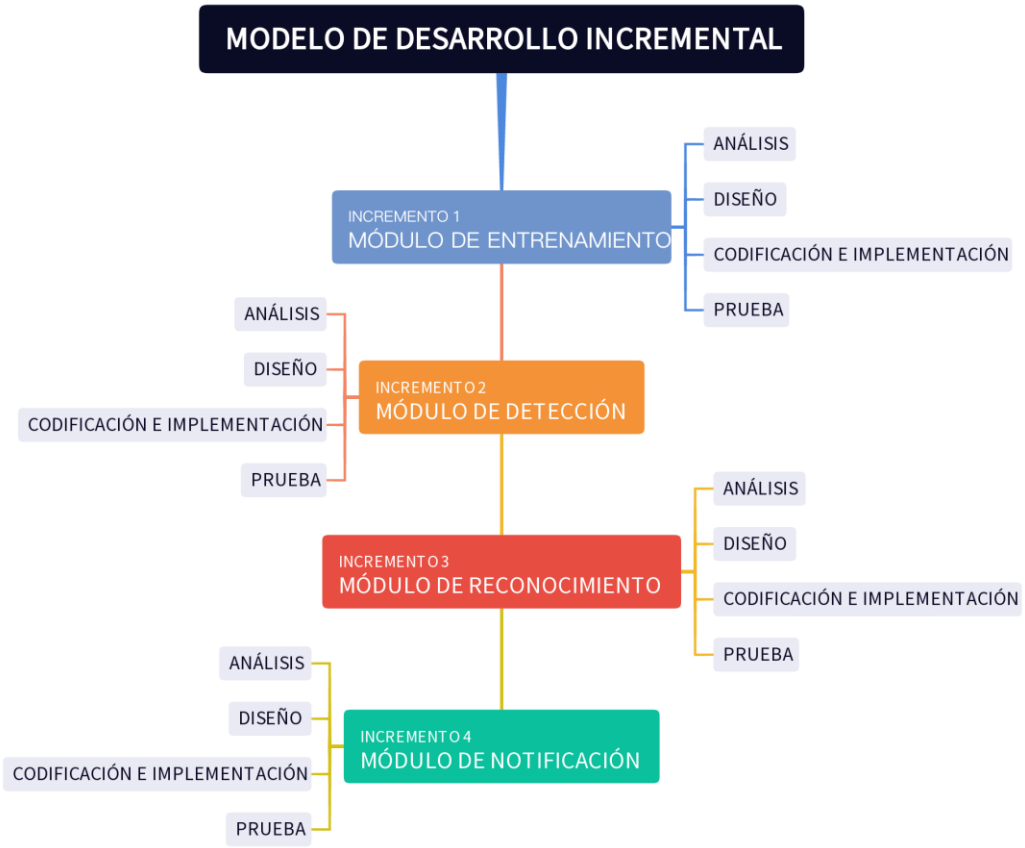


Figure 1: Modelo de desarrollo incremental

CAPÍTULO 3. PROPUESTA

3.1. Requerimientos

3.1.1. Requerimientos Funcionales

Código	Detalles de Requerimientos	Clase
RQF-01	El sistema detectará armas de fuego mediante una red neuronal entrenada.	Funcionalidad
RQF-02	El sistema generará embeddings faciales (vectores de características) a partir del dataset de personas.	Inteligencia Artificial
RQF-03	El sistema permitirá almacenar a las personas clasificadas en el dataset, a la base de datos.	Funcionalidad
RQF-04	El sistema comparará los embeddings con los del dataset, para verificar si la persona pertenece al plantel.	Inteligencia Artificial
RQF-05	El sistema clasificará a la persona identificada como, estudiante, docente.	Funcionalidad
RQF-06	El sistema permite detectar en tiempo real si existe un arma de fuego.	Visualización
RFQ-07	El sistema permite detectar en tiempo real si una persona pertenece al plantel o está registrada.	Visualización
RFQ-08	El sistema permite pasarle una imagen o video en el que podrá detectar si existe arma de fuego.	Visualización
RFQ-09	El sistema permite pasarle una imagen o video en el que detectará si una persona está o no registrada en la base de datos.	Visualización

RFQ-10	El sistema mostrará en la interfaz los datos de la persona una vez que le pasamos una imagen.	Visualización
RFQ-11	El sistema mostrará en la interfaz si la persona no pertenece o está registrada en la base de datos.	Visualización
RQF-12	El sistema se conectará mediante un bot de Telegram.	Comunicación
RQF-13	El sistema enviará una alerta automática a un grupo de Telegram, en caso de detección de arma.	Comunicación
RQF-14	En la alerta enviada a Telegram, se detalla donde fue la detección de arma.	Comunicación
RQF-15	El sistema almacena logs de detecciones en archivos locales (con fecha, hora, tipo de detección y usuario detectado).	Registro de actividad
RQF-16	El sistema permite almacenar las detecciones a una base de datos.	Registro de actividad
RQF-17	El sistema permite almacenar las alertas a una base de datos.	Registro de actividad
RQF-18	El sistema permitirá registrar nuevos rostros al dataset clasificándolos por categoría (estudiante, docente)	Gestión de usuarios
RQF-19	El sistema permitirá actualizar el dataset con nuevas imágenes o personal autorizado.	Mantenimiento.
RQF-20	El sistema validará si una persona ya ha sido registrada.	Validación.

Table 2: Requerimientos Funcionales

3.1.2. Requerimientos no Funcionales

Código	Detalles de Requerimientos	Clase
RQNF-01	El sistema debe funcionar correctamente en sistemas Linux con soporte para Pytorch y CUDA si está disponible.	Compatibilidad
RQNF-02	El código debe ser modular y permitir actualizaciones futuras del modelo o del dataset sin necesidad de reentrenar desde cero.	Mantenibilidad
RQNF-03	Las alertas deben enviarse de forma segura, sin exponer públicamente información sensible del personal.	Seguridad
RQNF-04	El bot de Telegram debe estar autenticado mediante token seguro y restringido solo al sistema.	Seguridad
RQNF-05	Las imágenes del dataset deben mantenerse organizadas y resguardadas en una estructura de carpetas jerárquica.	Organización de datos
RQNF-06	El sistema debe iniciar automáticamente al encender el equipo como un servicio del sistema.	Disponibilidad

Table 3: Requerimientos no Funcionales

3.2. Componente de la Propuesta

3.2.1. Arquitectura del Sistema

El sistema de detección de armas y reconocimiento facial fue desarrollado bajo una arquitectura modular de tipo monolítico, ejecutada localmente. Cada módulo (captura, preprocesamiento, detección, reconocimiento, interfaz gráfica) está separado lógicamente, pero opera de forma integrada dentro de una única aplicación. Adicionalmente, el sistema se comunica con servicios externos para la

gestión de notificaciones (API Telegram), lo que permite clasificarlo como una arquitectura híbrida local-remota [50].

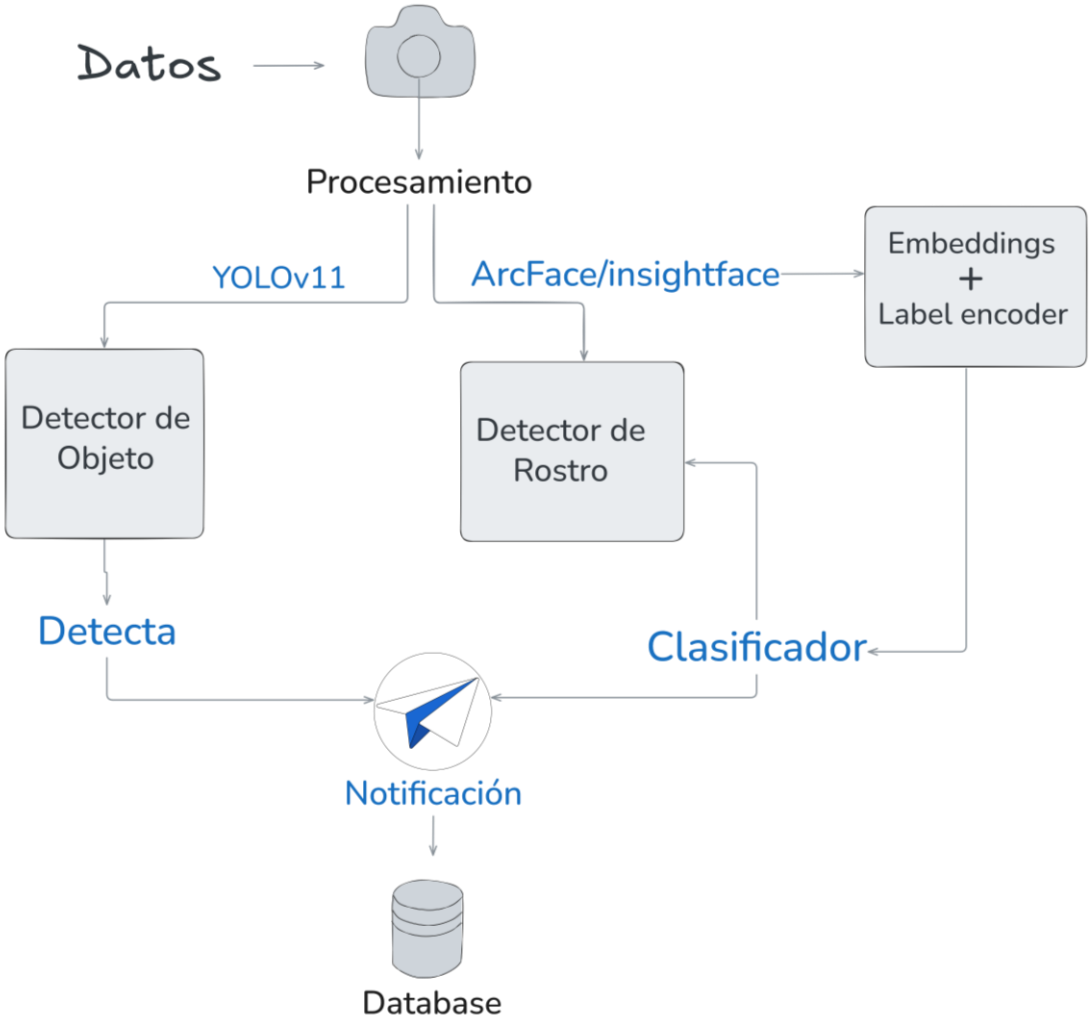


Figure 2: Arquitectura del sistema

3.2.2. Diagramas de casos de uso

Caso de uso: Detección de rostros.

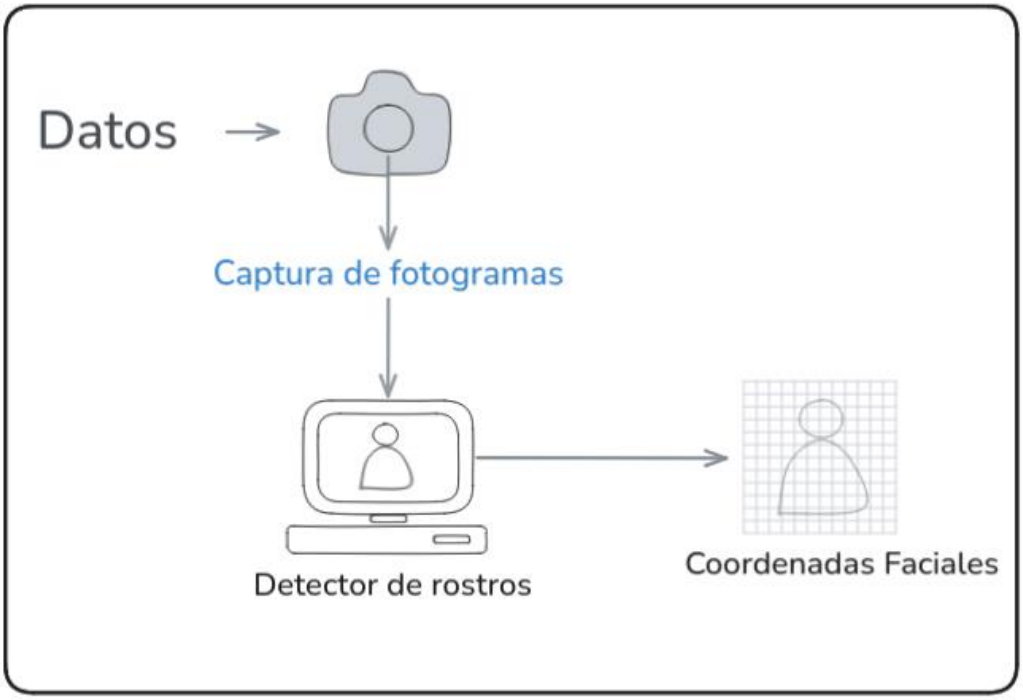


Figure 3: Caso de uso - Detectar rostro

CU-01 – Detectar rostro	
Actor	Sistema (script de vigilancia en Python)
Definición	Permite detectar rostros humanos en el área de vigilancia mediante una cámara conectada al sistema.
Acción	El sistema accede al stream de video en tiempo real y aplica técnicas de visión por computadora para detectar rostros.
Procedimiento	• Captura de fotograma desde la cámara. • Aplicación del modelo de detección facial (ArcFace) • Detección del rostro en la imagen.
Resultado	Se identifica la región de la imagen donde se encuentra el rostro, permitiendo su posterior análisis.

Table 4: Caso de uso - Detectar rostro

Caso de uso: Verificación de identidad.

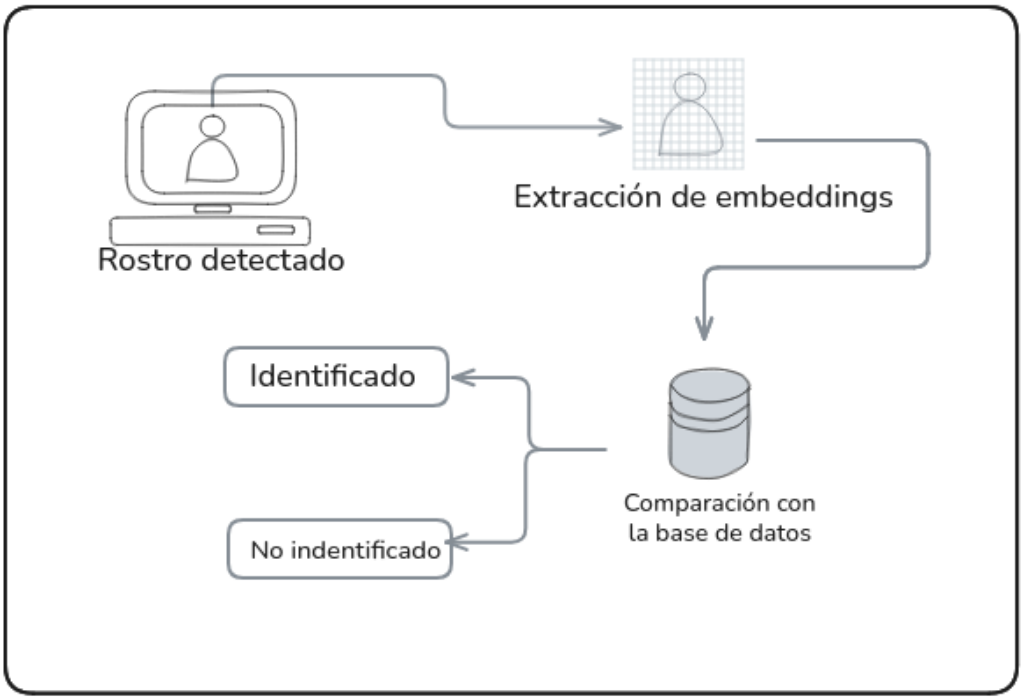


Figure 4: Caso de uso - Verificar identidad

CU-02 – Verificar identidad	
Actor	Sistema
Definición	Comprueba si el rostro detectado pertenece al personal autorizado (estudiantes, docentes).
Acción	El sistema genera un embedding facial y lo compara con los embeddings previamente almacenados.
Procedimiento	• Obtención del rostro detectado. • Generación del embedding con ArcFace. • Comparación con embeddings del dataset. • Evaluación de similitud
Resultado	Se determina si el rostro pertenece a alguien del plantel y se identifica su nombre y rol (docente, estudiante).

Table 5: Caso de uso - Verificar identidad

Caso de uso: Detección de armas de fuego.

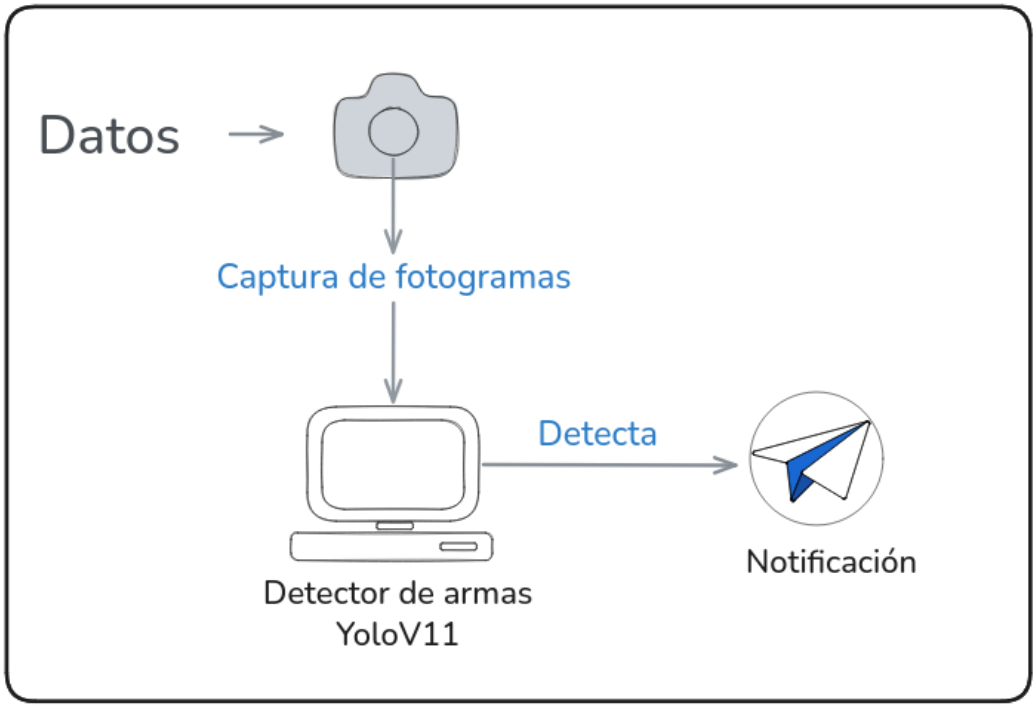


Figure 5: Caso de uso - Detectar arma de fuego

CU-03 – Detectar arma de fuego	
Actor	Sistema
Definición	Evalúa si una persona que ha ingresado al laboratorio porta un arma de fuego.
Acción	Utiliza un modelo de detección de objetos entrenado para reconocer armas.
Procedimiento	• Captura de imagen o video. • Aplicación del modelo (YOLOV11) para detectar objetos. • Filtrado de clases específicas (armas de fuego).
Resultado	Se genera una alerta si se detecta un arma en la imagen o video capturado.

Table 6: Caso de uso - Detectar arma de fuego

Caso de uso: Envío de alerta por Telegram.

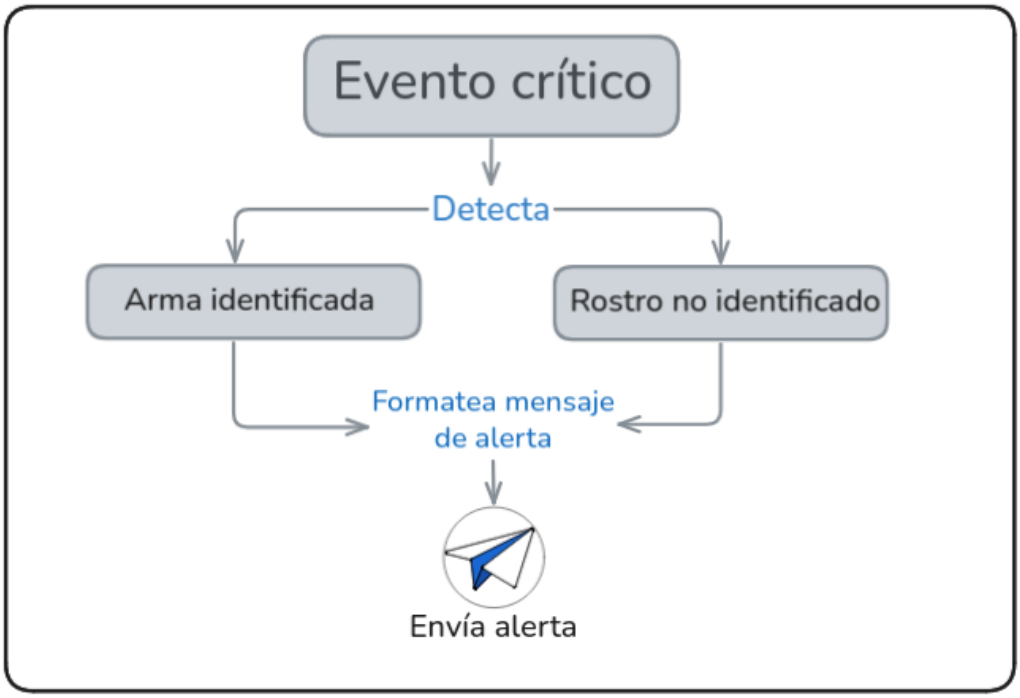


Figure 6: Caso de uso - Enviar alerta por Telegram

CU-04 – Enviar alerta por Telegram	
Actor	Sistema
Definición	Notifica a un grupo o canal de Telegram cuando se detecta una persona no autorizada o un arma.
Acción	El sistema envía un mensaje automáticamente a través del bot de Telegram configurado.
Procedimiento	• El sistema verifica si se ha detectado una anomalía (arma o persona no identificada). • Se formatea el mensaje con datos relevantes (nombre, hora, captura). • El mensaje se envía al canal o grupo mediante la API del bot.
Resultado	El grupo/canal recibe una alerta inmediata con la información crítica y la imagen de la detección.

Table 7: Caso de uso - Enviar alerta por Telegram

Caso de uso: Carga de embeddings desde dataset.

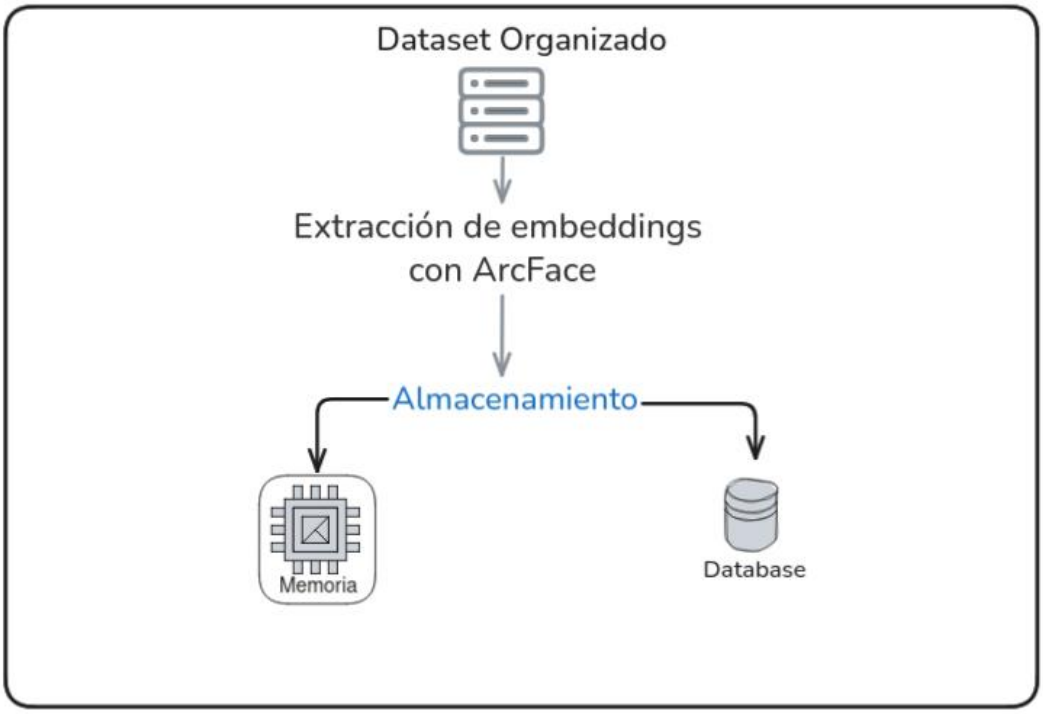


Figure 7: Caso de uso - Cargar embeddings desde dataset

CU-05 – Cargar embeddings desde dataset	
Actor	Usuario administrador
Definición	Permite cargar o actualizar la base de datos de embeddings para el reconocimiento facial desde el dataset estructurado.
Acción	Ejecuta un script que recorre las carpetas del dataset y genera embeddings para cada persona.
Procedimiento	• Leer imágenes del dataset organizado por carpetas (docentes, estudiantes). • Generar embeddings usando ArcFace. • Guardar los vectores generados en un archivo o base de datos local.
Resultado	El sistema queda preparado para identificar rostros con base en las nuevas imágenes ingresadas.

Table 8: Caso de uso - Cargar embeddings desde dataset

3.2.3. Modelado de Datos

El sistema utiliza una base de datos relacional implementada en MySQL para gestionar la información de las personas registradas y facilitar el proceso de reconocimiento facial. Esta base de datos cumple funciones tanto de almacenamiento persistente como de apoyo al sistema de identificación.

La estructura de datos está centrada en la tabla personas, que almacena la información relevante sobre cada individuo detectado y clasificado por el sistema. La elección de un enfoque relacional facilita la integridad de los datos, el uso de claves primarias y las consultas SQL para integración en el backend.

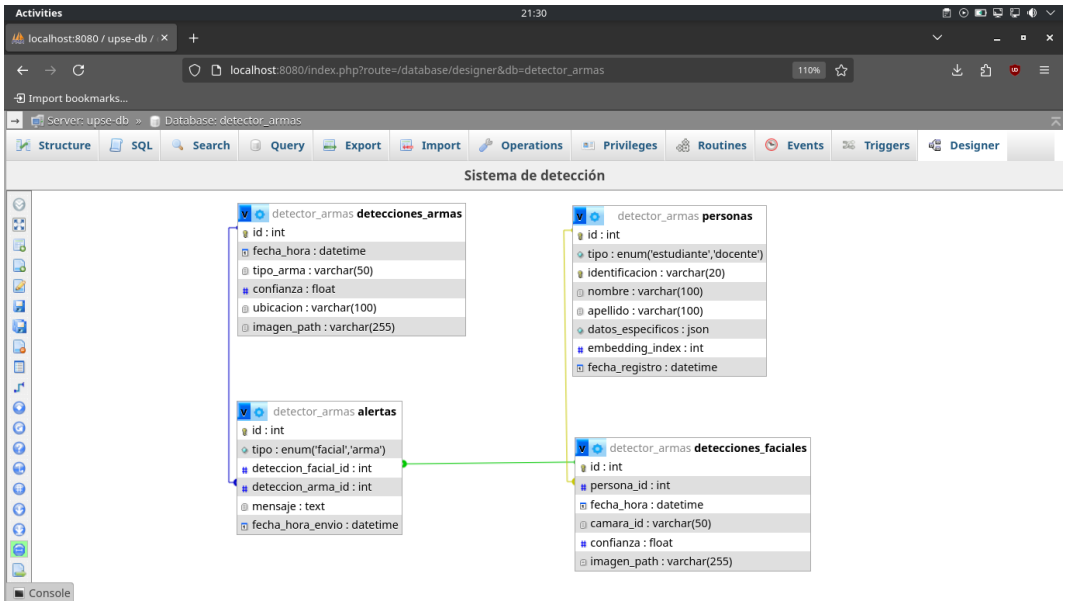


Figure 8: Modelado de datos

3.3. Diseño de Interfaces

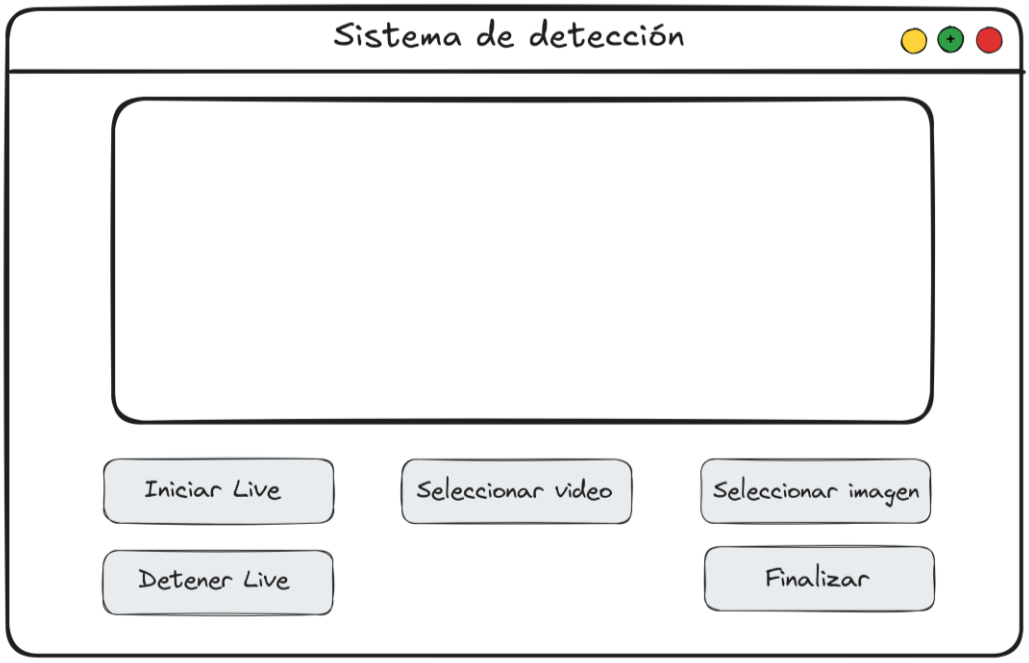


Figure 9: Interfaz - Inicio de sistema que permite la detección de armas y reconocimiento facial

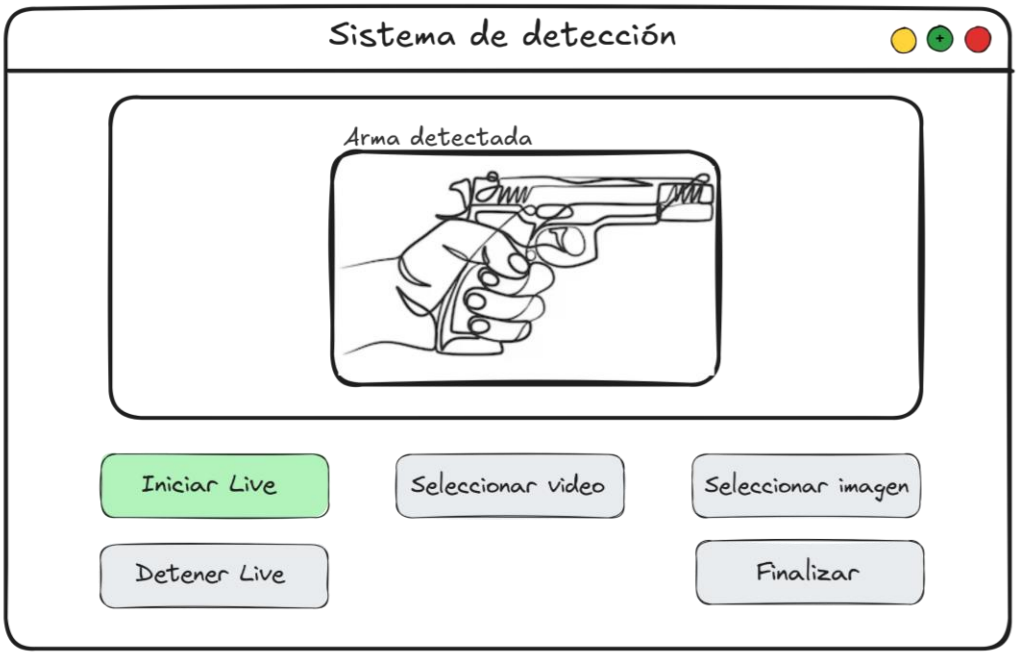


Figure 10: Interfaz - Detección de armas en tiempo real

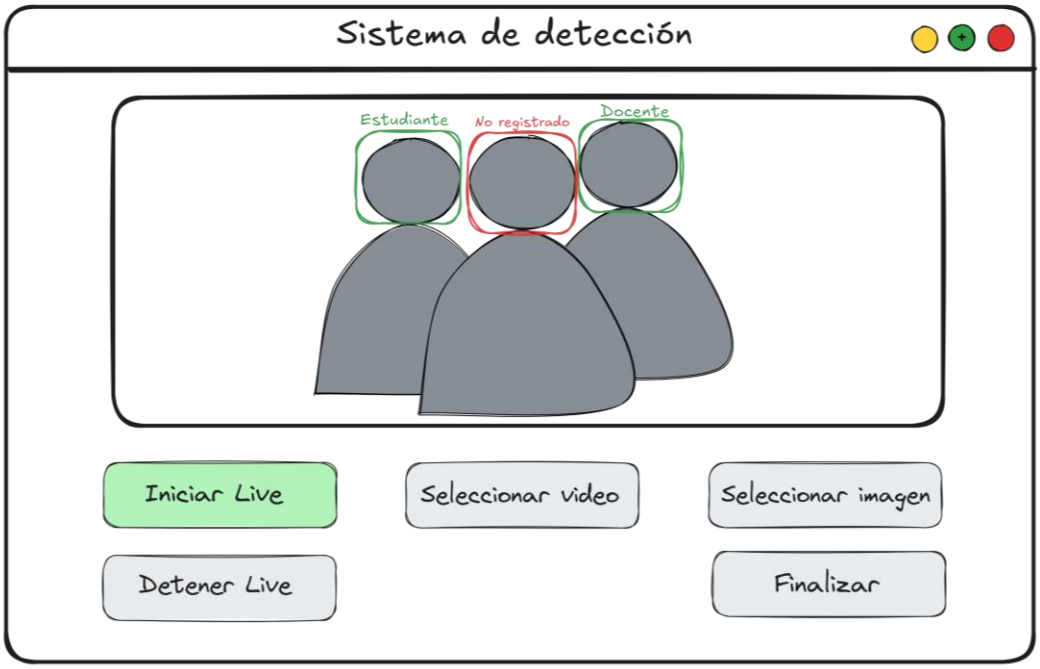


Figure 11: Interfaz - Detección facial en tiempo real

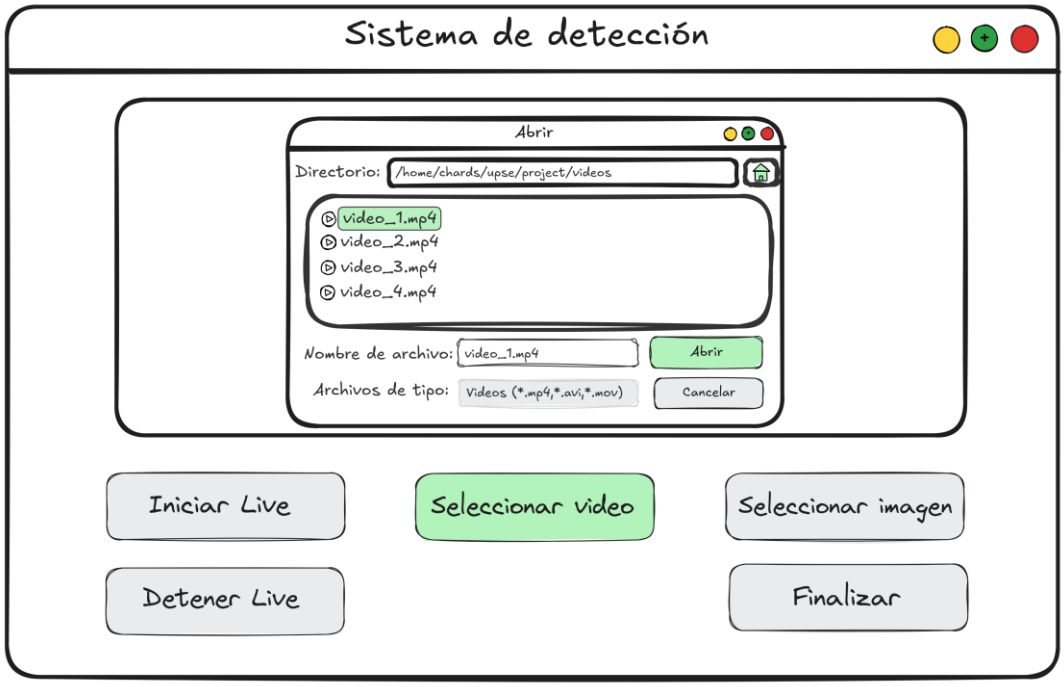


Figure 12: Interfaz - Selección de archivo de video, para la detección de armas de fuego y reconocimiento facial

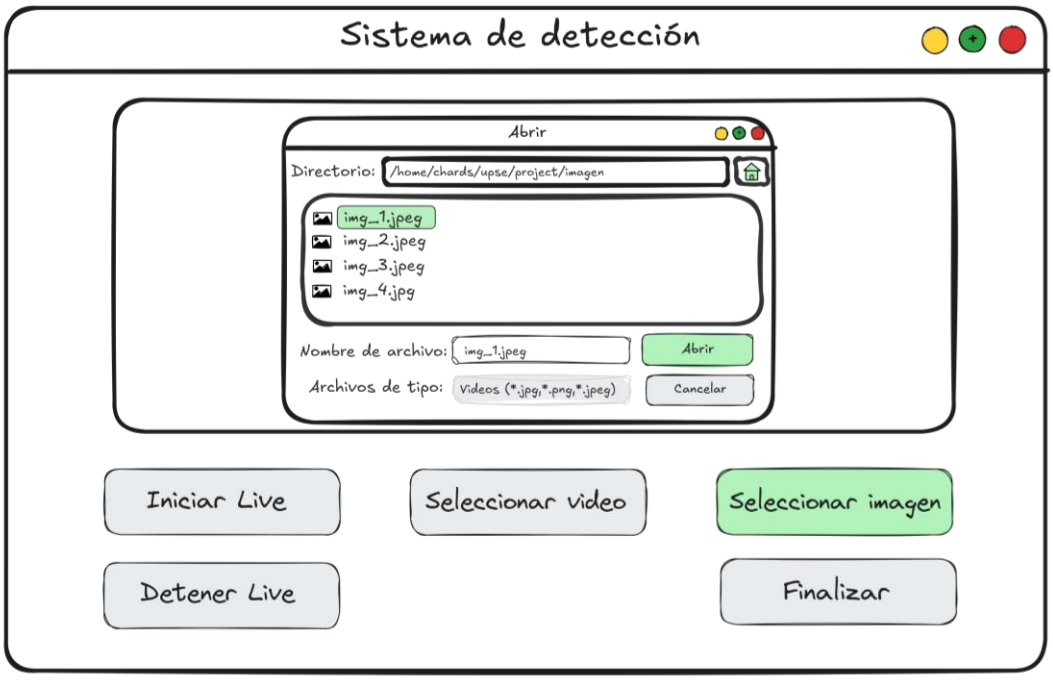


Figure 13: Interfaz - Selección de archivo de imagen, para la detección de armas de fuego y reconocimiento facial

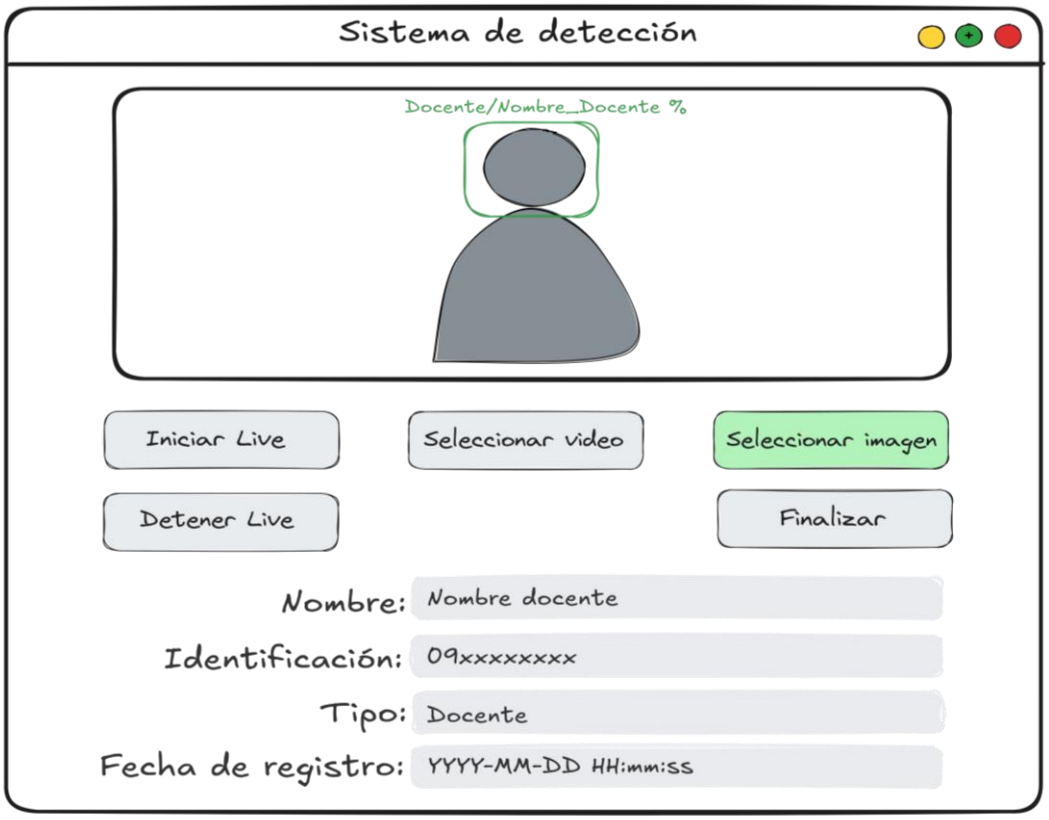


Figure 14: Interfaz - Visualización de los datos del docente, al pasarle una imagen de un docente registrado en la base de datos

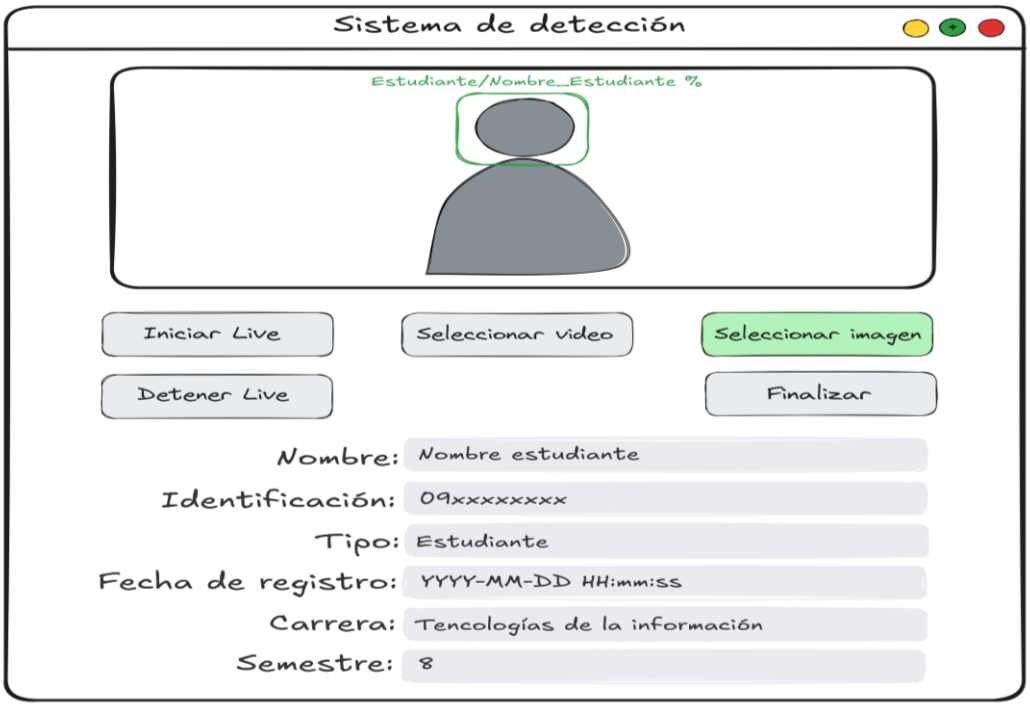


Figure 15: Interfaz - Visualización de los datos del estudiante, al pasarle una imagen de un estudiante registrado en la base de datos

Figure 16: Interfaz - Registro de personas a la base de datos

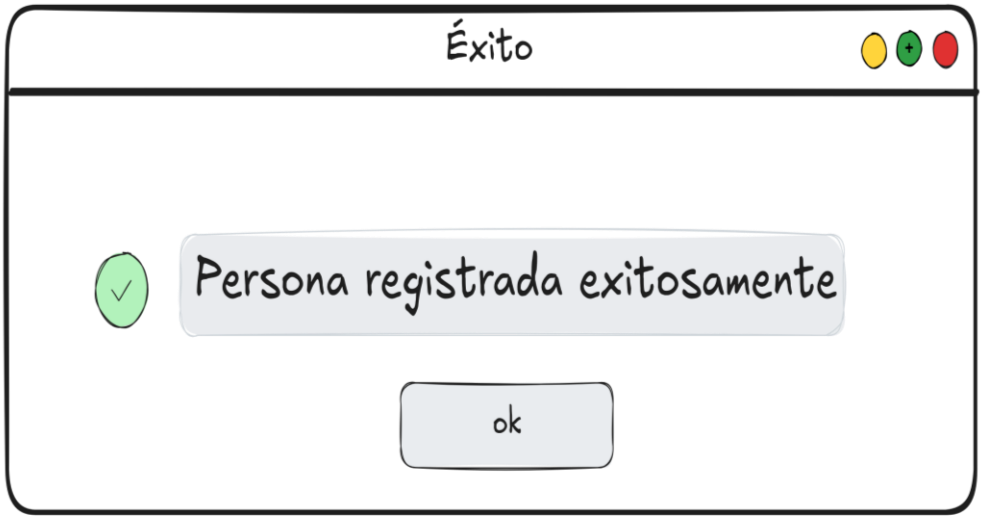


Figure 17: Interfaz - Persona registrada de manera exitosa

Figure 18: Interfaz - Error cuando se intenta registrar una persona que ya está registrada

3.4. Pruebas

Prueba	**No. 01**	**Detección de armas en tiempo real.**	
Usuario	Operador del sistema.		
Descripción	Verificar que el sistema detecte armas de fuego en tiempo real usando la cámara.		
Condiciones	Cámara conectada; modelo preentenado (model.pt) cargado.		
Pasos de la prueba	1. Iniciar la interfaz. 2. Seleccionar "cámara en vivo" 3. Apuntar con una réplica de arma de fuego al campo de visión.		
Resultados Obtenidos			
Resultado esperado		**Evaluación de la Prueba**	
El sistema debe detectar el arma, dibujar un recuadro y activar el evento de notificación.		✔	Exitoso.
			Fallido.

Table 9: Prueba No. 01 - Detección de armas en tiempo real

Prueba	**No. 02**	**Reconocimiento facial en tiempo real.**	
Usuario	Operador del sistema.		
Descripción	Verificar que el sistema reconozca a personas registradas frente a la cámara.		
Condiciones	Persona registrada previamente; modelo, embeddings cargados.		
Pasos de la prueba	1. Iniciar el sistema. 2. Acercar el rostro de un usuario conocido a la cámara.		
Resultados Obtenidos			
Resultado esperado		**Evaluación de la Prueba**	
Se debe mostrar el nombre de la persona reconocida sobre el rostro.		✔	Exitoso.
			Fallido.

Table 10: Prueba No. 02 - Reconocimiento facial en tiempo real

Prueba	**No. 03**	**Envío de notificación por detección de arma a Telegram.**	
Usuario	Operador.		
Descripción	Verificar que, al detectar un arma, el sistema envíe una alerta a Telegram.		
Condiciones	Bot de Telegram y chat_id configurados correctamente.		
Pasos de la prueba	1. Provocar una detección de arma. 2. Revisar la aplicación de Telegram.		
Resultados Obtenidos			
Resultado esperado		**Evaluación de la Prueba**	
Mensaje recibido en Telegram con alerta e imagen capturada.		✔	Exitoso.
			Fallido.

Table 11: Prueba No. 03 - Envío de notificación por detección de arma de fuego a Telegram

Prueba	**No. 04**	**Almacenamiento de notificación enviada.**	
Usuario	Sistema.		
Descripción	Verificar que cada notificación enviada también se registre en la base de datos.		
Condiciones	Base de datos MySQL conectada y funcional.		
Pasos de la prueba	1. Generar una detección de arma. 2. Revisar la tabla “alertas”		
Resultados Obtenidos			
Resultado esperado		**Evaluación de la Prueba**	
Nueva entrada con la fecha, hora y tipo de notificación.		✔	Exitoso.
			Fallido.

Table 12: Prueba No. 04 - Almacenamiento de notificación enviada

Prueba	**No. 05**	**Almacenamiento de detecciones.**	
Usuario	Sistema.		
Descripción	Verificar que cada reconocimiento facial o arma detectada se registre en la base de datos.		
Condiciones	Base de datos activa.		
Pasos de la prueba	1. Generar una detección de rostro o arma. 2. Consultar las tablas “detecciones_armas” y “detecciones_faciales”		
Resultados Obtenidos			
Resultado esperado		**Evaluación de la Prueba**	
Se debe registrar las detecciones a la tabla correspondiente, con sus detalles asociados.		✔	Exitoso.
			Fallido.

Table 13: Prueba No. 05 - Almacenamiento de detecciones

Prueba	**No. 06**	**Reconocimiento facial desde imagen.**	
Usuario	Operador.		
Descripción	Verificar que el sistema reconozca rostros desde una imagen cargada.		
Condiciones	Imagen que contenga un rostro conocido.		
Pasos de la prueba	1. Abrir la interfaz. 2. Cargar una imagen con un rostro.		
Resultados Obtenidos			
Resultado esperado		**Evaluación de la Prueba**	
Se muestran los detalles del usuario en la interfaz.		✔	Exitoso.
			Fallido.

Table 14: Prueba No. 06 - Reconocimiento facial desde imagen

Prueba	**No. 07**	**Detección de armas desde imagen.**	
Usuario	Operador.		
Descripción	Comprobar que se detecten armas en una imagen cargada.		
Condiciones	Imagen con arma visible.		
Pasos de la prueba	1. Abrir la interfaz. 2. Cargar imagen que contenga un arma.		
Resultados Obtenidos			
Resultado esperado		**Evaluación de la Prueba**	
Se dibuja el recuadro de arma y se genera la alerta.		✔	Exitoso.
			Fallido.

Table 15: Prueba No. 07 - Detección de armas desde imagen

Prueba	**No. 08**	**Reconocimiento facial desde video.**	
Usuario	Operador.		
Descripción	Verificar reconocimiento facial en un video cargado.		
Condiciones	Video con rostros conocidos.		
Pasos de la prueba	1. Cargar video desde la interfaz. 2. Esperar al análisis cuadro a cuadro.		
Resultados Obtenidos			
Resultado esperado		**Evaluación de la Prueba**	
Nombres sobre rostros cuando se detectan.		✔	Exitoso.
			Fallido.

Table 16: Prueba No. 08 - Reconocimiento facial desde video

Prueba	**No. 09**	**Detección de armas desde video.**
Usuario	Operador.	
Descripción	Verificar que se detecten armas en un video cargado.	
Condiciones	Video con presencia de armas.	

Pasos de la prueba	1. Cargar video desde la interfaz. 2. Observar detección.	
Resultados Obtenidos		
Resultado esperado	**Evaluación de la Prueba**	
Se detecta el arma y se registra.	✔	Exitoso.
		Fallido.

Table 17: Prueba No. 09 - Detección de armas desde video

Prueba	**No. 10**	**Registro de persona del dataset a la base de datos.**
Usuario	Administrador.	
Descripción	Probar que el usuario pueda registrar personas del dataset en la base de datos.	
Condiciones	Dataset cargado, base de datos activa.	
Pasos de la prueba	1. Abrir la interfaz. 2. Seleccionar persona del dataset. 3. Ingresar datos y guardar.	
Resultados Obtenidos		
Resultado esperado	Evaluación de la Prueba	
Persona registrada con éxito y sin duplicados.	✔	Exitoso.
		Fallido.

Table 18: Prueba No. 10 - Registro de personas del dataset a la base de datos

CONCLUSIONES

- La implementación de redes neuronales convolucionales y modelos de aprendizaje profundo para visión artificial permite desarrollar sistemas de seguridad inteligentes y autónomos. A través de la integración de modelos como ArcFace para reconocimiento facial y YOLOv11 para detección de armas, se logró un sistema funcional capaz de realizar análisis visual en tiempo real, respondiendo a amenazas con rapidez y precisión.
- El uso de PyTorch como biblioteca de desarrollo permitió un control detallado y preciso sobre el entrenamiento, evaluación y despliegue de modelos personalizados, favoreciendo la reutilización de redes preentrenadas y su adaptación a un entorno específico como la detección de armas en una institución educativa. Esta decisión tecnológica contribuyó a una mayor capacidad en el desarrollo e integración de los modelos dentro del sistema final.
- La clasificación de usuarios en categorías (docentes y estudiantes) dentro del dataset facial fue clave para el proceso de reconocimiento, ya que permitió no solo identificar rostros, sino también asociarlos a roles específicos. Esta integración de información contextual aporta un valor añadido al sistema de seguridad, al permitir una interpretación más precisa y significativa de los datos recolectados.
- El almacenamiento de registros en una base de datos MySQL y el envío automatizado de notificaciones por Telegram fortalecen la trazabilidad y monitoreo del sistema en tiempo real, permitiendo que personal de seguridad tome acciones inmediatas ante la detección de un arma o una persona no reconocida. Esta integración de inteligencia artificial, la mensajería y las bases de datos demuestra una solución completa y eficaz.
- El sistema se encuentra en un estado funcional y modular, lo que permite su adaptación a otras áreas de la institución o incluso su escalado a escenarios más complejos. La capacidad de aceptar entradas desde cámara en vivo, imágenes y videos, junto con su interfaz visual, lo convierte en una herramienta flexible para vigilancia y control de acceso inteligente.

RECOMENDACIONES

- Ampliar el dataset de entrenamiento tanto de armas de fuego como de rostros es fundamental para elevar la precisión del sistema. Se debe incluir variedad en condiciones de iluminación, ángulos, fondos y resoluciones. De igual forma, incorporar imágenes que representen posibles distracciones (objetos similares a armas, rostros parciales, uso de mascarillas) esto permitirá reducir falsos positivos y mejorar la robustez del modelo ante escenarios reales.
- Es fundamental establecer mecanismos de respaldo y seguridad en la base de datos que almacena las detecciones, alertas y registros. La creación de copias automáticas, uso de replicación o respaldos en la nube garantizarán la continuidad del sistema y la protección de la información ante fallos del sistema o accesos no autorizados.
- Se recomienda optimizar el rendimiento del sistema para su ejecución en dispositivos de bajo consumo energético, utilizando técnicas como la reducción de tamaño de los modelos, o el uso de versiones ligeras como YOLOv5n. Esto permitirá su implementación en dispositivos embebidos o cámaras inteligentes que cuenten con menor capacidad de procesamiento.
- El sistema ya envía alertas de forma automática a través de Telegram, sin embargo, se sugiere como mejora futura el desarrollo de una aplicación móvil dedicada para el personal de seguridad de la institución. Esta app permitiría a los guardias consultar en tiempo real la identidad y rol de cualquier persona del campus, detectada por el sistema, ver el historial de alertas, y recibir notificaciones desde la propia aplicación sin depender exclusivamente de Telegram. Además, podría incluir funcionalidades adicionales como escaneo de rostro desde el móvil, registros manuales, o verificación cruzada con la base de datos. Esta solución extendería la cobertura operativa del sistema, facilitando el trabajo del personal desde cualquier punto del campus universitario.

REFERENCIAS

[1] PRIMICIAS, "Zapata: 35 nuevas UPC deberían estar listas a finales de junio de 2023," *22 de marzo 2023*, la libertad, May 08, 2023. Accessed: May 09, 2023. [Online]. Available: https://www.primicias.ec/noticias/en-exclusiva/policia-incremento-muertes-violencia-juanzapata/

[2] CAROLINA MELLA, "Ecuador autoriza llevar armas para la defensa personal frente a la escalada del crimen," EL PAIS, Guayaquil, Apr. 05, 2023. Accessed: May 09, 2023. [Online]. Available: https://elpais.com/internacional/2023-04-05/ecuador-autoriza-llevar-armas-para-la-defensa-personal-frente-a-la-escalada-del-crimen.html

[3] UPSE, "Reseña Histórica de la Creación de la Universidad," UPSE. Accessed: May 09, 2023. [Online]. Available: https://www.upse.edu.ec/index.php?option=com_content&view=article&id=10&Itemid=188#

[4] UPSE, "Misión y Visión," UPSE. Accessed: Sep. 24, 2023. [Online]. Available: https://www.upse.edu.ec/index.php?option=com_content&view=article&id=12&Itemid=190

[5] Lineida Castillo and EL COMERCIO, "Universidades del Ecuador implementan tecnología y control de armas," Guayaquil, Apr. 07, 2023. Accessed: May 09, 2023. [Online]. Available: https://www.elcomercio.com/actualidad/universidades-del-ecuador-implementan-tecnologia-y-control-de-armas.html

[6] Ecuavisa, "Porte de armas: una decena de universidades de Ecuador rechazan la medida," Ecuavisa. Accessed: May 09, 2023. [Online]. Available: https://www.ecuavisa.com/noticias/seguridad/porte-de-armas-una-decena-de-universidades-de-ecuador-rechazan-la-medida-MK4826349

[7] el COMERCIO and LINEIDA CASTILLO, "Universidades del Ecuador rechazan el porte de armas," 03 de abril de 2023, Guayaquil, Apr. 03, 2023. Accessed: May 09, 2023. [Online]. Available:

https://www.elcomercio.com/tendencias/sociedad/universidades-del-ecuador-rechazan-el-porte-de-armas.html

[8] García Villalba Luis Javier and Sandoval Orozco Ana Lucila, “Detección de Armas en Vídeos Digitales Trabajo,” Madrid, 2019. Accessed: May 10, 2023. [Online]. Available: https://eprints.ucm.es/id/eprint/61320/1/1138495743-247939_PABLO_JOS%C3%89_ESTEVE_CALZADO_Memoria_Detector_Armas_en_V%C3%ADdeos_3940146_2086606904.pdf

[9] CRIOLLO LEAL BRAYAN ALEJANDRO and DÍAZ RONDÓN NICOLÁS, “MÉTODO DE DETECCIÓN AUTOMÁTICA DE ARMAS DE MANO EN VIDEO USANDO APRENDIZAJE PROFUNDO,” Bogota, 2019. Accessed: May 10, 2023. [Online]. Available: https://repository.ucatolica.edu.co/server/api/core/bitstreams/324b8eff-e939-461a-9d33-4d51097908f0/content

[10] OÑATE MIRANDA FERNANDO PATRICIO, “DISEÑO Y CONSTRUCCIÓN DE NODOS INTELIGENTES PARA DETECCIÓN DE ARMAS DENTRO DE UNA RED DE VIDEO- VIGILANCIA UTILIZANDO VISIÓN ARTIFICIAL,” Riobamba, 2020. Accessed: May 10, 2023. [Online]. Available: http://dspace.espoch.edu.ec/bitstream/123456789/13783/1/108T0323.pdf

[11] Python Software Foundation, “The Python Tutorial — Python 3.11.3 documentation,” Python Docs. Accessed: May 05, 2025. [Online]. Available: https://docs.python.org/es/3.13/tutorial/index.html

[12] Google, “Te damos la bienvenida a Colaboratory - Colaboratory,” Google Colab. Accessed: May 05, 2025. [Online]. Available: https://colab.research.google.com/

[13] PyTorch, “PyTorch documentation,” PyTorch Docs. Accessed: May 06, 2025. [Online]. Available: https://pytorch.org/docs/stable/index.html

[14] Ultralytics, “ultralytics: State-of-the-art deep learning models for object detection, segmentation, and classification,” GitHub. Accessed: May 06, 2025. [Online]. Available: https://github.com/ultralytics/ultralytics

[15] Ultralytics, “Ultralytics YOLO11,” Docs Ultralytics. Accessed: May 06, 2025. [Online]. Available: https://docs.ultralytics.com/es/models/yolo11/

[16] anxiangsir, “Distributed Arcface Training in Pytorch,” Github. Accessed: May 09, 2025. [Online]. Available: https://github.com/deepinsight/insightface/tree/master/recognition/arcface_torch

[17] OpenCV, “About - OpenCV,” OpenCV. Accessed: Jun. 05, 2023. [Online]. Available: https://opencv.org/about/

[18] Neovim Project, “¿Qué es Neovim?,” Neovim. Accessed: May 06, 2025. [Online]. Available: https://neovim.io/charter/

[19] Docker Inc., “What is a Container?,” Docker Documentation. Accessed: Apr. 12, 2025. [Online]. Available: https://www.docker.com/resources/what-container/

[20] Universidad Estatal Península de Santa Elena, “Resolución RCS-SO-01-07-2025 – Ajuste curricular no sustantivo de la Carrera de Software,” Santa Elena, Ecuador, Jan. 2025. Accessed: Jun. 12, 2025. [Online]. Available: https://www.upse.edu.ec/secretariageneral/images/archivospdfsecretaria/RESOLUCIONES/RESOLUCIONES_2025/RESOLUCION_SESIONES_ORDINARIA_2025/RESOLUCION_SESION_ORDINARIA_No._01-2025/RCS-SO-01-07-2025_Ajuste_curricular_no_sustantivo_de_la_Carrera_de_Software-signed-signed.pdf

[21] Na8, “Modelos de Detección de Objetos,” Aprende Machine Learning. Accessed: May 07, 2025. [Online]. Available: https://www.aprendemachinelearning.com/modelos-de-deteccion-de-objetos/

[22] Eileen Triana, “Detector de armas de fuego en sistemas de vigilancia,” Ebenezer Technologies. Accessed: May 01, 2025. [Online]. Available: https://ebenezertechs.com/detector-de-armas-de-fuego/

[23] Secretaría Nacional de Planificación, “Plan Nacional de Desarrollo para el Nuevo Ecuador 2024–2025,” Ecuador, 2024. Accessed: Jun. 13, 2025.

[Online]. Available: https://www.planificacion.gob.ec/wp-content/uploads/2024/02/PND2024-2025.pdf

[24] Coursera, “What Is Artificial Intelligence? Definition, Uses, and Types,” Coursera. Accessed: May 09, 2025. [Online]. Available: https://www.coursera.org/articles/what-is-artificial-intelligence

[25] Amit sheps, “What Is AI in Cyber Security? ,” Aqua Security. Accessed: May 09, 2025. [Online]. Available: https://www.aquasec.com/cloud-native-academy/application-security/ai-in-cyber-security/

[26] EDS Robotics, “Visión por computador: qué es, objetivos y aplicaciones,” EDS Robotics . Accessed: May 09, 2025. [Online]. Available: https://www.edsrobotics.com/blog/vision-computador-que-es/

[27] Murel Jacob and Kavlakoglu Eda, “What is object detection? ,” IBM. Accessed: May 09, 2025. [Online]. Available: https://www.ibm.com/think/topics/object-detection

[28] viso.ai, “Weapon Detection,” viso.ai. Accessed: May 09, 2025. [Online]. Available: https://viso.ai/application/weapon-detection/

[29] Amazon Web Services (AWS), “¿Qué es el reconocimiento facial?,” Amazon Web Services (AWS). Accessed: May 09, 2025. [Online]. Available: https://aws.amazon.com/what-is/facial-recognition/

[30] Brownlee Jason, “ A Gentle Introduction to Transfer Learning for Deep Learning,” Machine Learning Mastery . Accessed: May 09, 2025. [Online]. Available: https://machinelearningmastery.com/transfer-learning-for-deep-learning/

[31] IBM, “What are convolutional neural networks?,” IBM. Accessed: May 09, 2025. [Online]. Available: https://www.ibm.com/think/topics/convolutional-neural-networks

[32] Amazon Web Services (AWS), “¿Qué es el aprendizaje por transferencia?,” Amazon Web Services (AWS). Accessed: May 09, 2025. [Online]. Available: https://aws.amazon.com/es/what-is/transfer-learning/

[33] Red Hat, “¿Qué es el open source?,” Red Hat. Accessed: May 09, 2025. [Online]. Available: https://www.redhat.com/es/topics/open-source/what-is-open-source

[34] Bergmann Dave and Stryker Cole, "¿Qué es PyTorch? ," IBM. Accessed: May 09, 2025. [Online]. Available: https://www.ibm.com/mx-es/think/topics/pytorch

[35] Yasar Kinza and Lewis Sarah, " PyTorch," TechTarget. Accessed: May 09, 2025. [Online]. Available: https://www-techtarget-com.translate.goog/searchenterpriseai/definition/PyTorch?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc

[36] Jiankang Deng, Guo Jia, Xue Niannan, and Zafeiriou Stefanos, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," Long Beach, California, Jun. 2019. Accessed: May 09, 2025. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2019/html/Deng_ArcFace_Additive_Angular_Margin_Loss_for_Deep_Face_Recognition_CVPR_2019_paper.html

[37] Espíndola Gustavo, "¿Qué son los embeddings y cómo se utilizan en la inteligencia artificial con python?," Medium. Accessed: May 09, 2025. [Online]. Available: https://gustavo-espindola.medium.com/qu%C3%A9-son-los-embeddings-y-c%C3%B3mo-se-utilizan-en-la-inteligencia-artificial-con-python-45b751ed86a5

[38] Cao Qiong, Shen Li, Xie Weidi, M. Parkhi Omkar, and Zisserman Andrew, "VGGFace2: A dataset for recognising faces across pose and age," May 2018. Accessed: May 10, 2025. [Online]. Available: https://www.robots.ox.ac.uk/~vgg/publications/

[39] Telegram, "¿Que es Telegram?," Telegram. Accessed: May 10, 2025. [Online]. Available: https://telegram.org/faq#p-que-es-telegram-que-puedo-hacer-aqui

[40] Telegram, "Bot API," Telegram. Accessed: May 10, 2025. [Online]. Available: https://core.telegram.org/api

[41] Telegram, "Telegram Bot API," Telegram. Accessed: May 10, 2025. [Online]. Available: https://core.telegram.org/bots/api

[42] R. Olmos, S. Tabik, and F. Herrera, "Automatic handgun detection alarm in videos using deep learning," Neurocomputing, vol. 275, pp. 66–72, Jan. 2018, doi: 10.1016/J.NEUCOM.2017.05.012.

[43] Norbert Wiener, Cybernetics or Control and Communication in the Animal and the Machine, Segunda Edición. Cambridge, Massachusetts: The MIT Press, 2019. Accessed: Apr. 09, 2025. [Online]. Available: https://doi.org/10.7551/mitpress/11810.001.0001

[44] ROMERO MOGROVEJO DAVID ORLANDO, "Desarrollo de un sistema de detección de armas de fuego cortas en el monitoreo de videos de cámaras de seguridad," Cuenca, 2018. Accessed: Apr. 09, 2025. [Online]. Available: https://dspace.ups.edu.ec/handle/123456789/16793

[45] S. Theodoridis, A. Pikrakis, K. Koutroumbas, and D. Cavouras, Introduction to Pattern Recognition: A Matlab Approach. 2010. doi: 10.1016/C2009-0-18558-6.

[45] Salinas Meruane Paulina and Cárdenas Castro Manuel, "Métodos de investigación social," FLACSO, vol. Segunda Edición, pp. 60–61, 2009, Accessed: May 10, 2025. [Online]. Available: www.flacsoandes.edu.ec

[46] Universidad Latinoamericana, "HRM558 | Investigación Exploratoria INVESTIGACIÓN EXPLORATORIA: Fundamentos básicos," 2017. Accessed: May 10, 2025. [Online]. Available: https://practicasprofesionales.ula.edu.mx/documentos/ULAONLINE/Maestria/MAN/HRM558/Publicaci%C3%B3n/Semana_3/Estudiante/HRM558_S3_E_Inv_explo.pdf

[47] Solidaridad, "INVESTIGACIÓN Y METODOLOGÍA DIAGNÓSTICA. | solidaridad2010," Solidaridad. Accessed: May 10, 2025. [Online]. Available: https://solidaridad2010.blogia.com/2011/020304-investigaci-n-y-metodolog-a-diagn-stica..php

[48] Fidias G. Arias, El proyecto de investigación Introducción a la metodología científica, 6a Edición., vol. 6ta Edición. Caracas: Editorial Episteme, C.A., 2012. Accessed: May 10, 2025. [Online]. Available: https://abacoenred.org/wp-content/uploads/2019/02/El-proyecto-de-investigaci%C3%B3n-F.G.-Arias-2012-pdf-1.pdf

[49] Plutora, "Incremental Model: Efficient Development Strategies," Plutora. Accessed: Apr. 09, 2025. [Online]. Available:

https://www.plutora.com/blog/incremental-model-what-and-how-to-implement-it

[50] Qasemjaber Zahraa, “Design and Implementation of Real Time Face Recognition System (RTFRS),” ResearchGate, May 2014, doi: 10.5120/16395-6014.

[51] Ministerio de la Mujer y Derechos Humanos, “Validación, consolidación y actualización de cifras de homicidios intencionales de mujeres y femicidios corte al 09 de abril de 2023,” Apr. 2023. Accessed: Jun. 14, 2025. [Online]. Available: https://www.igualdadgenero.gob.ec/wp-content/uploads/downloads/2023/04/15.numerico_HI_femicidios_09042023-17.Abr_..23.pdf

ANEXOS

A. Manual de Usuario

Manual de Usuario

Sistema de Detección de Armas y Reconocimiento Facial

Autor: Richard Jose Cedeño Villón

Carrera: Tecnologías de la Información – UPSE

Tutor: Ing. Carlos Efraín Sánchez León

Fecha: [14/06/2025]

Introducción

Bienvenido al sistema de Detección de Armas y Reconocimiento Facial. Este sistema está diseñado para permitir a guardias, docentes o personal no técnico:

- Detectar objetos de tipo arma en video o cámara en tiempo real.
- Reconocer rostros registrados (docentes o estudiantes).
- Recibir alertas automáticas vía Telegram cuando se detecta un arma.
- Guardar registros de forma automática.

Requisitos

- Cámara web USB o integrada.
- PC o laptop con Windows/Linux.
- Conexión a internet para Telegram.
- Al menos 1 GB libres para modelos y base de datos.
- Python 3.8 o superior instalado.
- Git instalado (para clonar repositorio).
- Instalación de dependencias mediante pip

Instalación paso a paso

A. Clonar el repositorio

1. Abrir un emulador de terminal

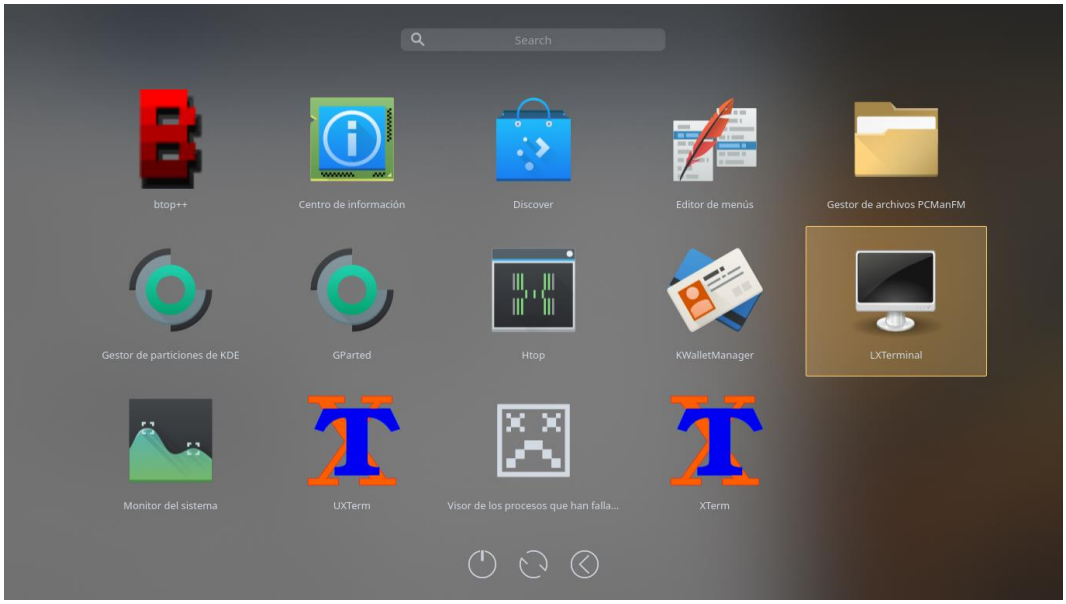


Figure 19: Manual de usuario - Abrir una terminal

2. Nos ubicamos en el directorio home del usuario actual, ingresando el comando “cd ~” o en el directorio de preferencia, con el comando “cd /ruta/al/directorio/”
3. Clonamos el repositorio del sistema de detección con el siguiente comando “git clone https://github.com/richardscedeno/detection-system.git”
4. Verificamos que se ha clonado el repositorio con el comando “ls”
5. Accedemos al directorio clonado con el comando “cd detection-system”

Figure 20: Manual de usuario - Clonando repositorio del sistema

B. Crear entorno virtual

1. Dentro del directorio creamos un entorno virtual con Python, ingresando el comando “python3 –m venv nombre_de_entorno” en este caso le llamaré “detection” quedando el comando de la siguiente manera “python3 –m venv detection”
2. Activamos el entorno con el comando para linux o mac “source detection/bin/activate” para activarlo desde windows ingresar el comando “venv\Scripts\activate“

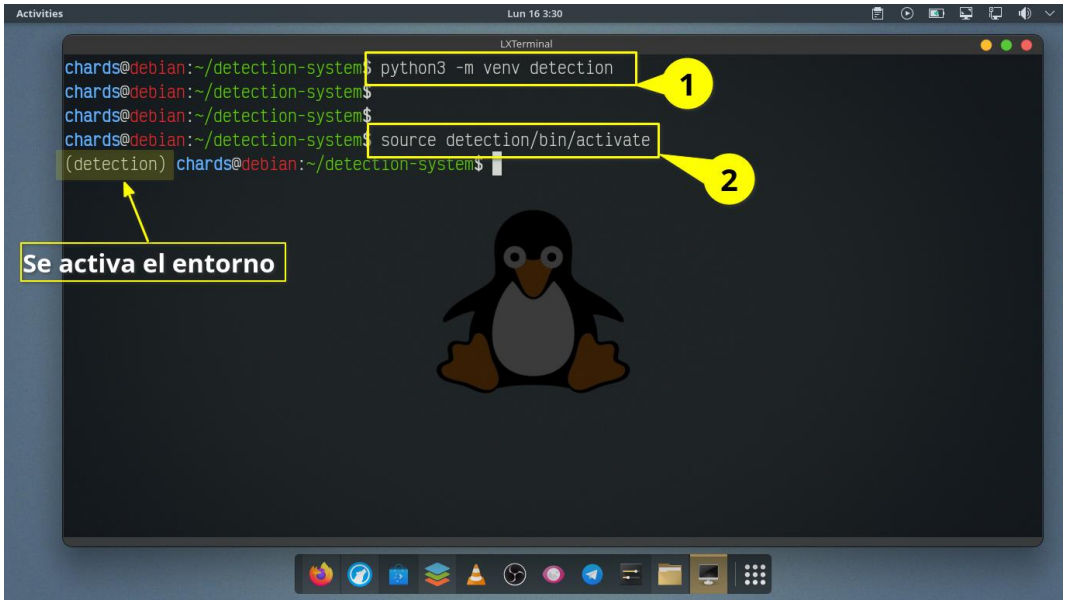


Figure 21: Manual de usuario - Creación y activación del entorno virtual

C. Instalar dependencias

1. Verificamos que este el archivo “requirements.txt” dentro del directorio, con el comando “ls”
2. Procedemos a instalar las dependencias con el comando “pip install -r requirements.txt”

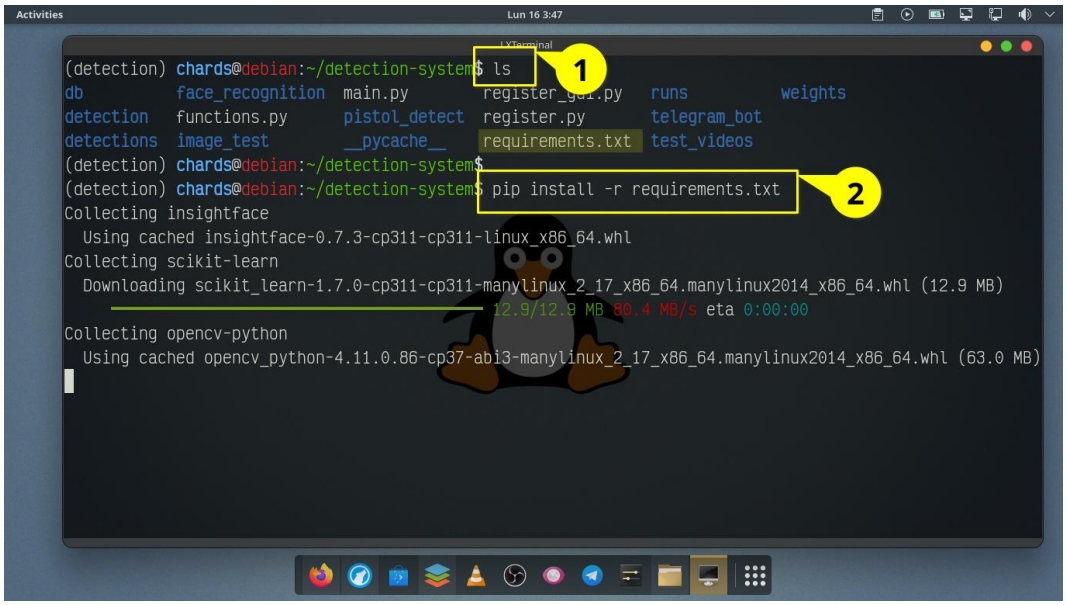


Figure 22: Manual de usuario - Instalación de dependencias

D. Ejecutar el Sistema

1. Verificar que exista el archivo “main.py” con el comando “ls”
2. Iniciamos el sistema con el comando “python main.py” y se abrirá la interfaz gráfica.

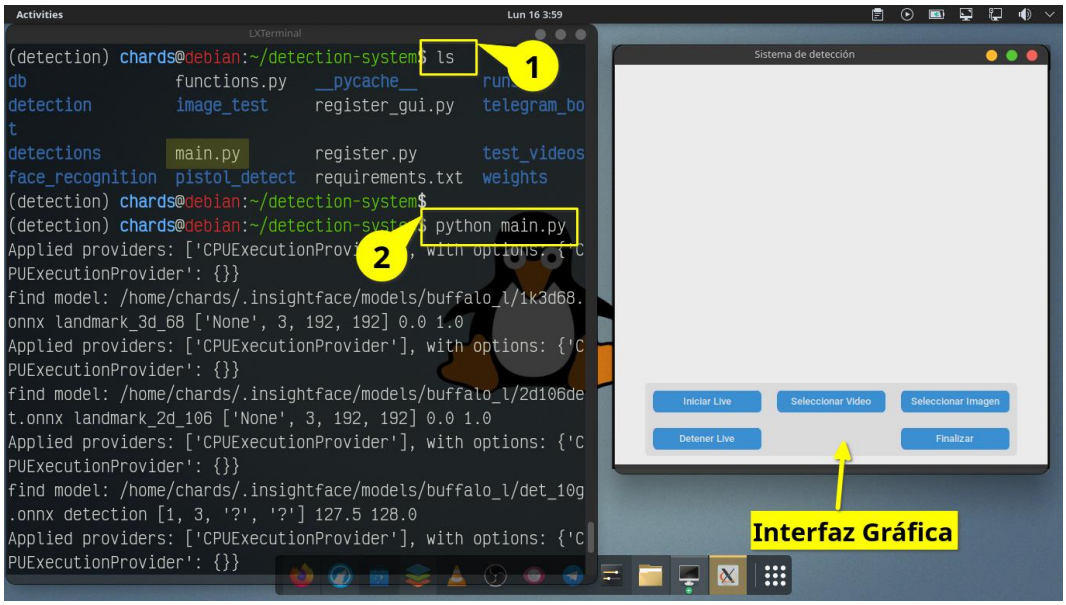


Figure 23: Manual de usuario - Ejecución del sistema

E. Uso del sistema

Modo Cámara.

1. Seleccionar el botón “Iniciar Live”.
2. Observará recuadros alrededor de armas (en caso de existir una) y rostros.
3. Si detecta un arma o rostros desconocidos: enviará de manera automática una notificación a un grupo de Telegram previamente configurado.

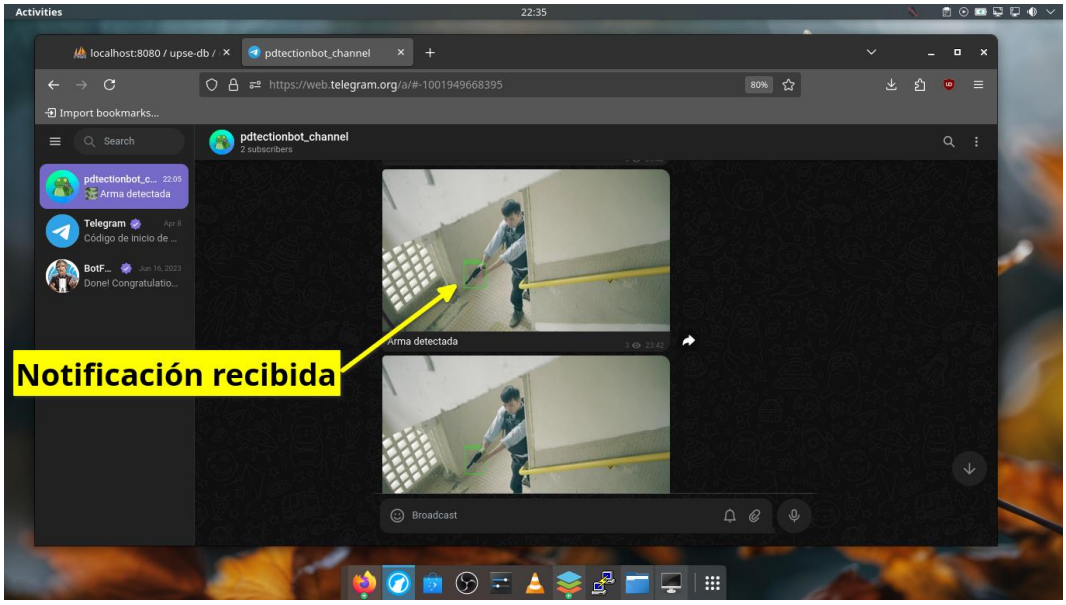


Figure 24: Manual de usuario - Notificación de detección de arma recibida

Modo “Video”

1. Selecciona el botón “Seleccionar Video”.
2. Haz clic en Cargar vídeo, selecciona archivo MP4.
3. El video se ejecuta y procesa fotograma a fotograma, detectando armas y rostros.

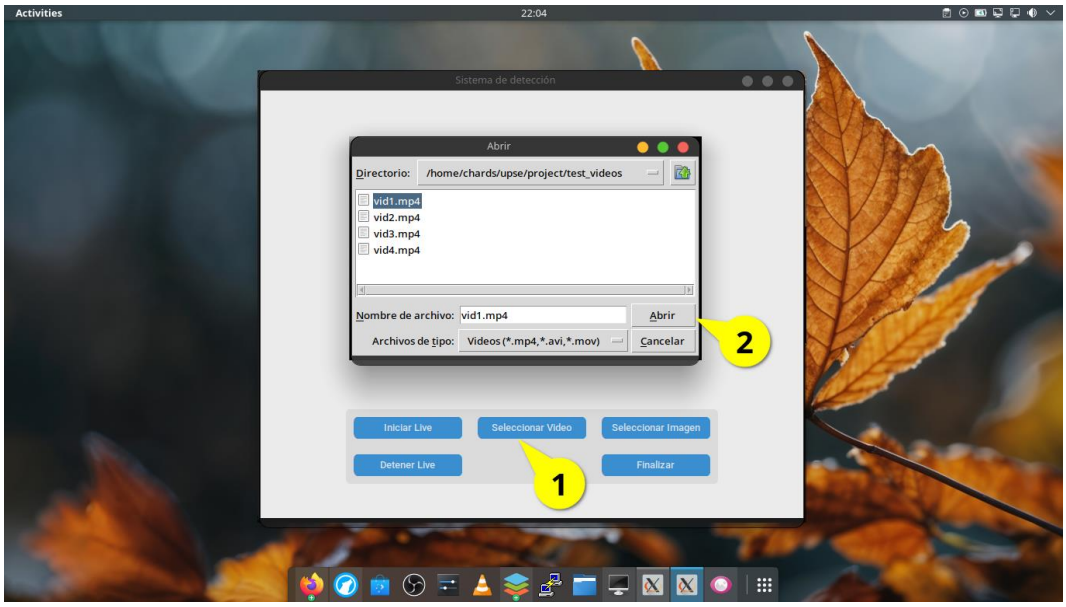


Figure 25: Manual de usuario - Selección de video para verificar detecciones

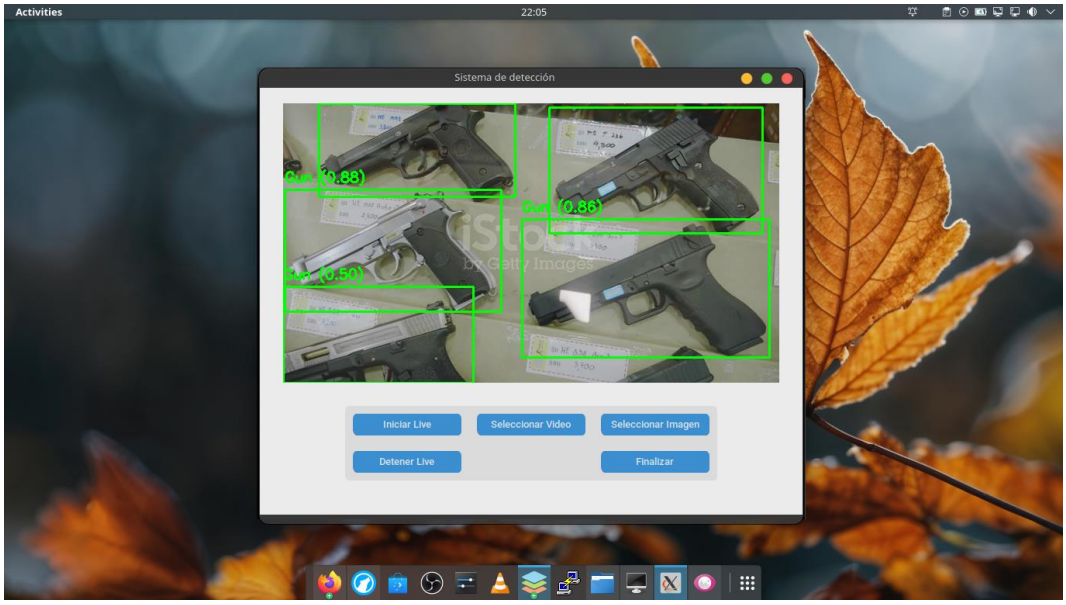


Figure 26: Manual de usuario - Detección de arma en video seleccionado

Modo “Imagen”

1. Selecciona el botón “Seleccionar Imagen”.
2. Haz clic en Cargar imagen, selecciona una JPEG o PNG.
3. Se mostrará la imagen procesada con detecciones y etiquetas, también mostrará en la parte inferior, los datos de un estudiante, docente o si es desconocido.

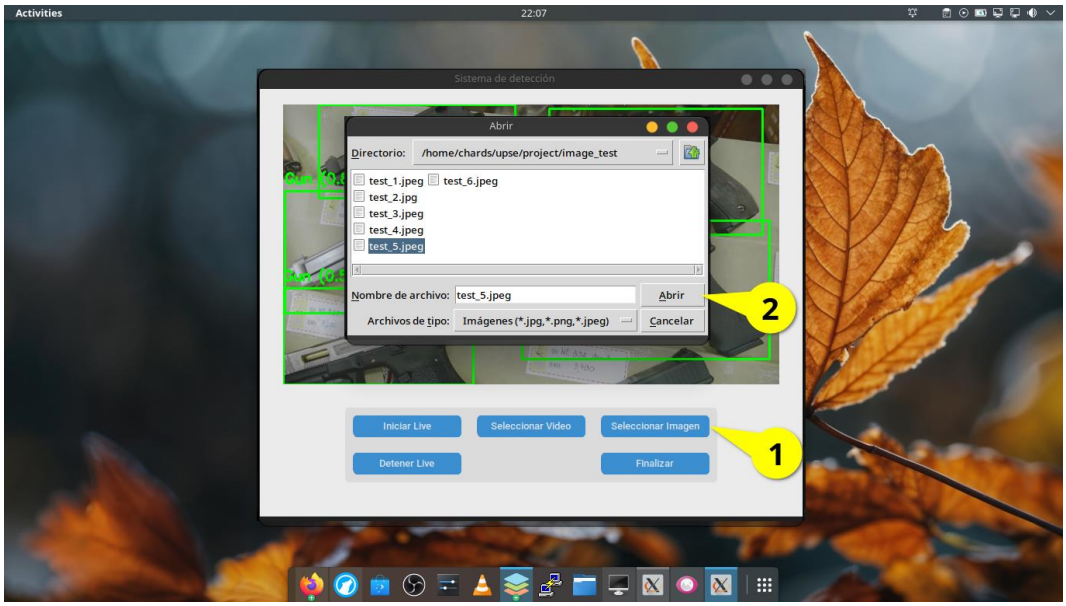


Figure 27: Manual de usuario - Selección de imagen para verificar detecciones

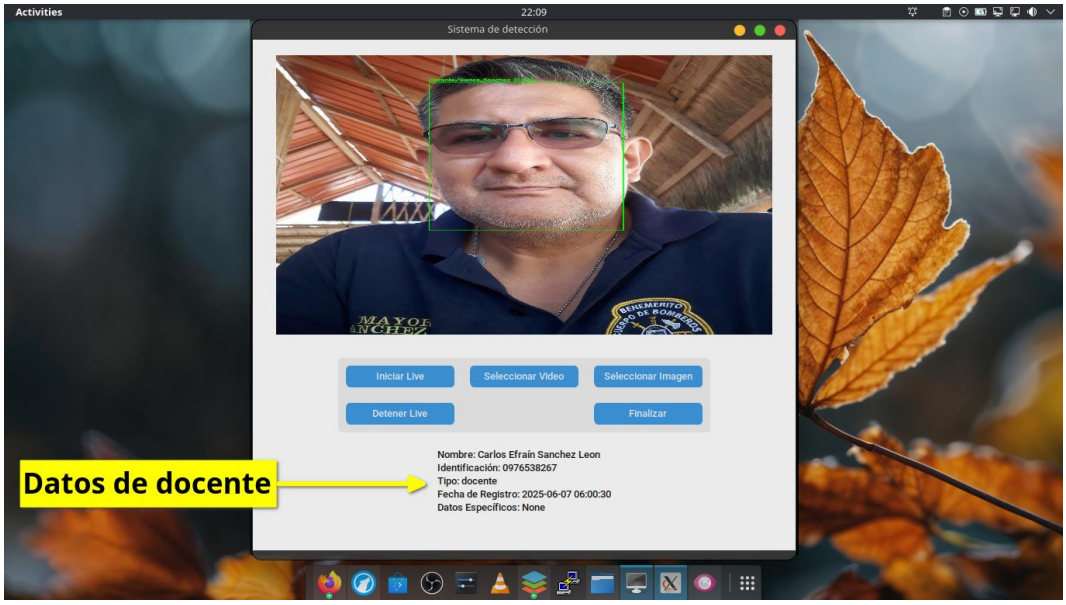


Figure 28: Manual de usuario - Visualización de datos del docente registrado en el sistema

F. Interpretación de resultados.

Recuadro de color verde: Cuando detecta un arma de fuego, en la pantalla se observará un recuadro de ese color.

Recuadro de color verde + nombre de persona + porcentaje: Esto se muestra cuando hace el reconocimiento facial, de una persona que está registrada en el sistema.

Recuadro de color rojo: Se mostrará cuando no reconoce a la persona que está en foco ya sea con la cámara en tiempo real o pasándole una imagen.

G. Preguntas frecuentes

¿Qué hacer si no detecta mi cámara?

- Verifica que esté conectada y tenga permisos.

No llegan alertas de Telegram

- Revisa token y chat_id en el siguiente archivo /detection-system/telegram_bot/.telegram_keys

H. Soporte técnico

Para soporte, escribe a: richard.cedenovillon4588@upse.edu.ec

B. Muertes violentas en el ecuador del 10 de agosto del año 2014 al 09 de abril del año 2023

Año infracción	Sexo	Mes de infracción												Total
		Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sept	Oct	Nov	Dic	
2014	Hombres	-	-	-	-	-	-	-	70	78	67	94	95	404
	Mujeres	-	1	-	-	-	-	-	11	12	15	12	18	69
2015	Hombres	87	73	83	74	83	82	60	63	67	75	69	62	878
	Mujeres	11	9	23	11	12	18	15	15	15	14	13	16	172
2016	Hombres	83	54	67	66	68	65	73	60	65	46	72	62	781
	Mujeres	16	19	11	11	21	18	14	12	17	10	17	11	177
	No determinado	0	0	0	0	0	1	0	0	0	0	0	0	1
2017	Hombres	69	68	77	70	75	62	58	51	59	60	56	69	774
	Mujeres	26	22	16	16	15	18	17	17	12	9	9	19	196
2018	Hombres	62	77	73	85	82	66	64	65	76	65	74	67	856
	Mujeres	11	9	16	14	16	10	7	11	13	11	8	13	139
2019	Hombres	73	78	69	104	92	87	78	84	81	100	93	98	1.037
	Mujeres	9	9	14	13	10	14	12	5	14	15	18	16	149
	No determinado	1	0	0	0	0	0	0	0	0	0	0	0	1
2020	Hombres	101	104	71	74	82	96	84	109	93	129	125	139	1.207
	Mujeres	12	12	9	14	14	19	10	13	12	9	23	18	165
2021	Hombres	113	213	147	151	150	155	180	186	300	203	262	208	2.268
	Mujeres	9	13	26	25	22	16	18	16	25	22	20	15	227
2022	Hombres	293	286	309	332	397	322	362	353	436	427	389	472	4.378
	Mujeres	22	26	33	34	29	34	39	34	39	41	29	52	412
	No determinado	0	0	2	0	0	1	2	0	2	1	2	0	10
2023	Hombres	484	439	569	179									1.671
	Mujeres	32	45	59	13									149
	No determinado	0	4	2	0									6

Figure 29: Cuadro estadístico de muertes violentas del 2014 al 2023 [51].

C. Ficha de Observación (no estructurada)

Objetivo: Determinar mediante la técnica de observación cuáles son las problemáticas que se dan al no existir un adecuado control de acceso del laboratorio de informática.

<table>
<tr><td colspan="2">UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA
Facultad de Sistemas y Telecomunicaciones
Carrera de Tecnologías de la Información y Comunicación
2024-1</td></tr>
<tr><td colspan="2">Lugar: Laboratorio de informática de la facultad de sistemas y telecomunicaciones de la UPSE</td></tr>
<tr><td>Fecha: 10/06/2024</td><td>Responsable: Richard José Cedeño Villon</td></tr>
<tr><td colspan="2">Hechos observados:

No existe un control de las personas que ingresan al laboratorio de informática, y podría ocasionar un problema o accidente a los estudiantes, docentes y personal que se encuentran en el laboratorio debido a la inseguridad que vive el país hoy en día.</td></tr>
<tr><td colspan="2">Recomendaciones: Se recomienda implementar un prototipo de detección de armas de fuego y reconocimiento facial con inteligencia artificial, para disminuir los riesgos de posibles accidentes.</td></tr>
</table>