REPOSITORIO
ACADÉMICO
USMP

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE COMPUTACIÓN Y SISTEMAS

SISTEMA DE PREVENCIÓN Y DETECCIÓN DE VIOLENCIA FÍSICA MEDIANTE REDES NEURONALES EN EL RESTAURANTE REAL PEZ EN EL AGUSTINO

PRESENTADA POR

GABRIELA DELLANIRE RODRIGUEZ BARDALES

WILLY ALEXANDER ESPINOZA DIAZ

ASESORES

AUGUSTO ERNESTO BERNUY ALVA

GENER VICTOR ZAMBRANO LOLI

TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE COMPUTACIÓN Y SISTEMAS

LIMA – PERÚ

2021

CC BY-NC-SA

Reconocimiento – No comercial – Compartir igual

El autor permite transformar (traducir, adaptar o compilar) a partir de esta obra con fines no comerciales, siempre y cuando se reconozca la autoría y las nuevas creaciones estén bajo una licencia con los mismos términos.

http://creativecommons.org/licenses/by-nc-sa/4.0/

USMP
UNIVERSIDAD DE SAN MARTIN DE PORRES

FACULTAD DE INGENIERÍA Y ARQUITECTURA

ESCUELA PROFESIONAL DE INGENIERÍA DE COMPUTACIÓN Y SISTEMAS

SISTEMA DE PREVENCIÓN Y DETECCIÓN DE VIOLENCIA FÍSICA MEDIANTE REDES NEURONALES EN EL RESTAURANTE REAL PEZ EN EL AGUSTINO

TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE COMPUTACIÓN Y SISTEMAS

PRESENTADA POR

RODRIGUEZ BARDALES, GABRIELA DELLANIRE

ESPINOZA DIAZ, WILLY ALEXANDER

LIMA – PERÚ

2021

Esta tesis está dedicada sobre todo a Dios, por darnos fuerza para continuar en este arduo trabajo; a nuestros padres, por su amor, trabajo y sacrificio en todos estos años y por inspirarnos y apoyarnos en todo lo que nos hemos propuesto.

Agradecemos a Dios por todas las oportunidades que nos brinda día a día, a nuestros padres por darnos todo su apoyo, motivarnos a crecer profesionalmente y dar lo mejor de nosotros siempre.

A los docentes de nuestra casa de estudio, la Universidad de San Martín de Porres, quienes en estos cinco años han compartido con nosotros su conocimiento y experiencia.

ÍNDICE

ÍNDICE DE TABLAS

Pág.

ÍNDICE DE FIGURAS

Pág.

RESUMEN

Esta tesis se titula Sistema de Prevención y Detección de Violencia Física Mediante Redes Neuronales en el Restaurante Real Pez en El Agustino y tuvo como objetivo crear un sistema que integre interfaces que permitan identificar posibles escenarios de violencia física dentro del restaurante.

Esta investigación utilizó las metodologías RUP y CRISP-DM, empleó el aprendizaje supervisado con redes neuronales y un conjunto de librerías, como Tensor Flow (keras) y OpenCV, Numpy, Face_recognition, etc., que permitieron realizar el procesamiento de imágenes a través de la extracción de fotogramas de videos para la identificación de personas requisitoriadas y/o acciones violentas.

Para corroborar el funcionamiento del sistema, se evaluaron dos algoritmos, en el primero se analizaron 50 casos basados en el reconocimiento de la acción y se obtuvo la identificación de 48 actos de violencia física en 14 días; en el segundo, se analizaron 40 casos basados en el reconocimiento del rostro en 14 días, y se logró la identificación de 36 casos registrados en la base de datos: “personas requisitoriadas”.

Los modelos propuestos según las métricas aplicadas tuvieron una mayor precisión, evidencia de ello es el 95% de asertividad para detectar acciones de violencia cuando la cámara está a 1.5 o 2 metros de distancia del objetivo; asimismo, la precisión en la identificación de puñetes o patadas alcanzó el 92%, lo que demuestra que el sistema permite que la empresa

brinde una probabilidad mayor de seguridad a sus clientes y trabajadores utilizando el procesamiento de imágenes para la identificación de personas y/o acciones violentas.

Palabras claves: Tensor Flow, OpenCV, redes neuronales, violencia física y aprendizaje supervisado.

ABSTRACT

The thesis "System for the Prevention and Detection of Physical Violence Using Neural Networks in the Real Pez Restaurant in El Agustino" aimed to create a system to integrate interfaces that allows identifying possible scenarios of physical violence in the restaurant.

This research used RUP and CRISP-DM methodologies, supervised learning with neural networks and a set of libraries, such as: Tensor Flow (keras) and OpenCV, Numpy, Face_recognition, among others, which allowed image processing through the extraction of video frames to identify violent actions and persons wanted by the police or with criminal records.

To verify the proper functioning of the system, two algorithms were evaluated. In the first, 50 cases based on action recognition were analyzed, and 48 acts of physical violence were identified in 14 days. In the second, 40 cases based on face recognition were analyzed, and 36 faces registered in the database called “people wanted by the police or with criminal records” were identified in 14 days.

The proposed models according to the applied metrics had greater precision, evidence of this is 95% assertiveness to detect violent actions when camera is 1.5 or 2 meters away from the target. Likewise, the precise identification of punches or kicks reached 92%, these percentages show that this system allows the restaurant to provide a greater probability of safety to its customers and workers using image processing to identify violent actions and people with a history of violence.

Keywords: Tensor Flow, OpenCV, neural networks, physical violence, supervised learning.

INTRODUCCIÓN

Los últimos cinco años incrementó notoriamente los casos de agresión de violencia física, de acuerdo con el Instituto Nacional de Estadística (INEI) en las encuesta realiza a personas de 15 a más años en el Perú en el año 2019 de 26 de cada 100 peruanos han sido afectos por algún acto de violencia en la ciudad de Lima, asimismo, otra encuestadora Lima como Vamos indica que en Lima existe la percepción de inseguridad al 65%, entre los distintos distritos más inseguros y peligrosos se tuvo a San Juan de Lurigancho, Villa María del Triunfo, Independencia, los Olivos y El Agustino que en el periodo 2019 han registrado índices de priorización del 1.64 de la violencia en Lima.

Para un efecto disuasorio entre los años 2017 y 2018 incrementó el número de cámaras de videovigilancia instaladas por los municipios para la seguridad ciudadana.

Para este proyecto se tomó en cuenta, la problemática de seguridad en la sucursal principal ubicada en la calle Riva Agüero- El Agustino de la empresa Real Pez SAC, el cual es un restaurante en el rubro de la elaboración de platos con pescados y mariscos que cuenta con 3 sucursales ubicados en distintos puntos en El Agustino. En los últimos años por la ubicación en la que se encuentra, la sucursal principal ha presentado distintos escenarios que han terminado en agresión física afectando la seguridad de los comensales, la sucursal cuenta con dos pisos, en el primer pise se ubica dos cámaras, uno

que enfoca directamente a la puerta y el segundo enfocado a las mesas de los comensales. Actualmente, por problemas de COVID-19 se tuvo que reducir personal y actualmente no se cuenta con una persona para un monitoreo diario de las cámaras de videovigilancia para la identificación y prevención precisa e inmediata de situaciones y escenarios que pueden darse antes de un acto de violencia física dentro y fuera del local que puedan afectar la seguridad de los comensales.

El interés de la investigación en el ámbito profesional radica en brindar alternativas tecnológicas que permitan llegar a nuevas soluciones en base a la problemática del restaurante, a nivel académico es aplicar el uso del Machine Learning por la importancia que tuvo actual y específicamente en los algoritmos del aprendizaje supervisado.

La organización de la tesis incluye seis capítulos, en el capítulo I se realiza el fundamento de la situación problemática en la que se formula el problema y los objetivos planteados para su solución, así como también se determina la justificación de la investigación, alcance y limitaciones. en el capítulo II, se presentan el marco teórico de la investigación con los antecedentes y bases teóricas. en el capítulo III, se plantea la metodología que se lleva en la investigación. en el capítulo IV se enfoca en el desarrollo del sistema, desde el diseño y análisis del modelo de red neuronal y el sistema web, diseño de la arquitectura de los modelos, integrar los modelos al sistema web y las pruebas del sistema. en el capítulo V se presentan los resultados obtenidos en la fase de pruebas del sistema. en el capítulo VI se expone la interpretación de la discusión, analizando el detallado de los resultados de la investigación. Finalmente, se proponen las conclusiones y recomendaciones del autor como resultado de la investigación.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1. Situación problemática

En un estudio Bonilla Alguera, G. (2020) “América Latina es el territorio en el planeta donde la violencia mortal aumentó entre el 2000 y el 2010, y las personas reconocieron que el crimen y la violencia son su fundamental preocupación” (p.64). Acorde con la información de 26 países, más del 50% de sus residentes se sienten indefensos caminando solos por la noche (Ver Anexo 1), entre los ciudadanos encuestados que ha solicitado un cambio de lugar por la inseguridad se encuentra Perú con el 16.3% (Ver Anexo 2).

La seguridad ciudadana es una de las principales demandas de la población peruana, según la agencia de análisis de Estados Unidos Gallup (2019) “Perú se encuentra dentro de los 10 países más inseguros de todo el mundo y con mayor violencia”. en los últimos 5 años, se han incrementado notoriamente los casos de violencia el 32% de los casos corresponden a la ciudad de Lima los cuales se atendieron en el periodo de enero a diciembre del 2019 (Ver Anexo 5) y de acuerdo con la INEI el 26% de la población fueron víctimas de algún hecho delictivo de violencia en la ciudad de Lima (Ver Anexo 6).

1.2. Definición del problema

En el informe del Comité Distrital de Seguridad Ciudadana por el periodo 2019 se han registrado más de mil denuncias por delincuencia en las comisarías en El Agustino como se puede observar en la figura 1:

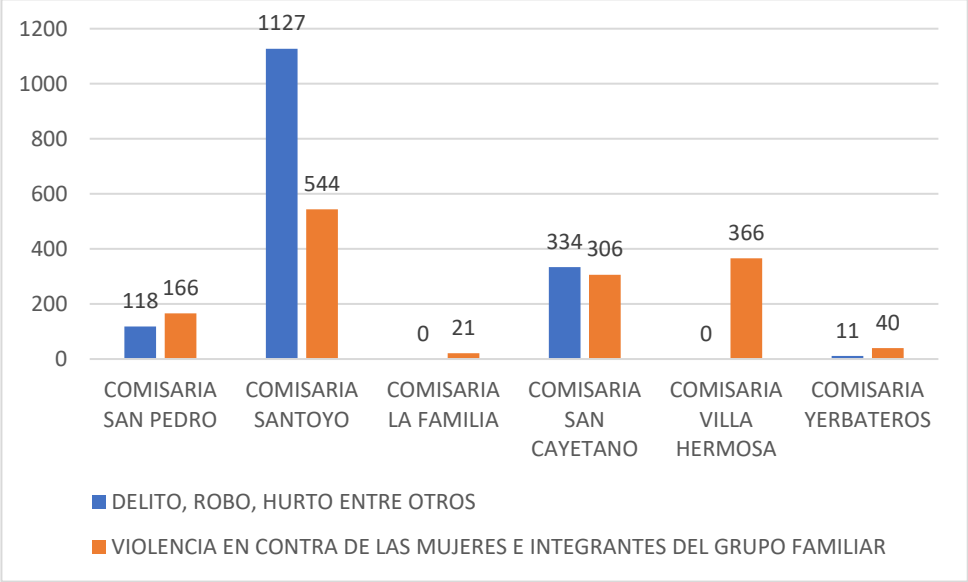


FIGURA *1* Gráfico de denuncias registradas por comisarías de El Agustino

Elaborado por: los autores

En la figura 2 se observa que solo por el primer trimestre del 2020 el total de denuncias registradas por violencia familiar incluido agresiones físicas son de 1 056 y por comisión de delitos en El Agustino es de 1 744.

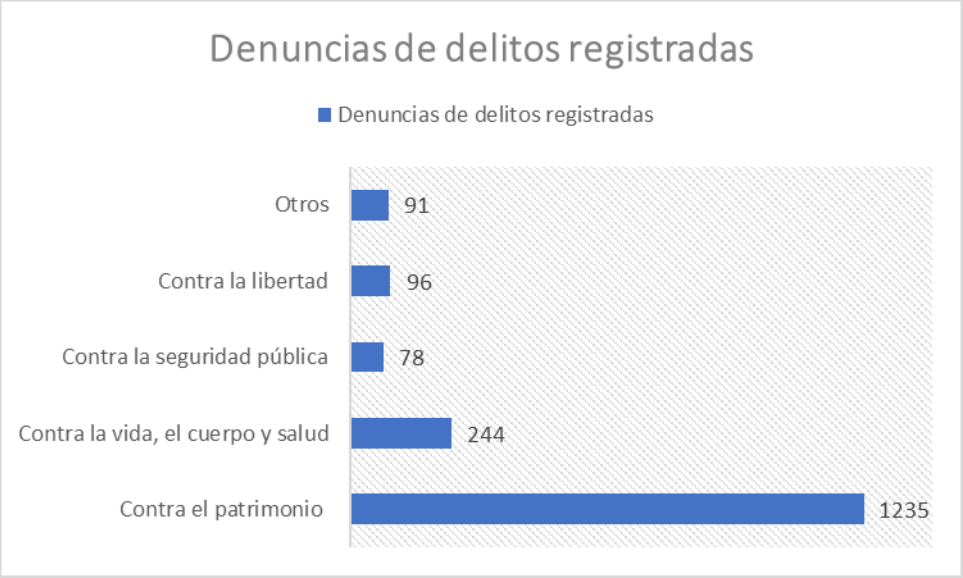


FIGURA 2 Gráfico de denuncias de delitos registradas por el primer trimestre

Elaborado por: los autores

Por otro lado, se realizó una encuesta por la plataforma Google Forms a más de 124 personas entre amistades, familiares y compañeros de estudios (Ver anexo 9) con la finalidad de analizar el comportamiento de la población del distrito y se obtuvo como resultado que más del 66% encuestados han sufrido violencia en el Agustino los cuales fueron realizado a través de puñetes y/o patadas para más detalle ver anexo 10 dando como resultado la figura 3 donde se observa 64 personas que han sufrido violencia física:

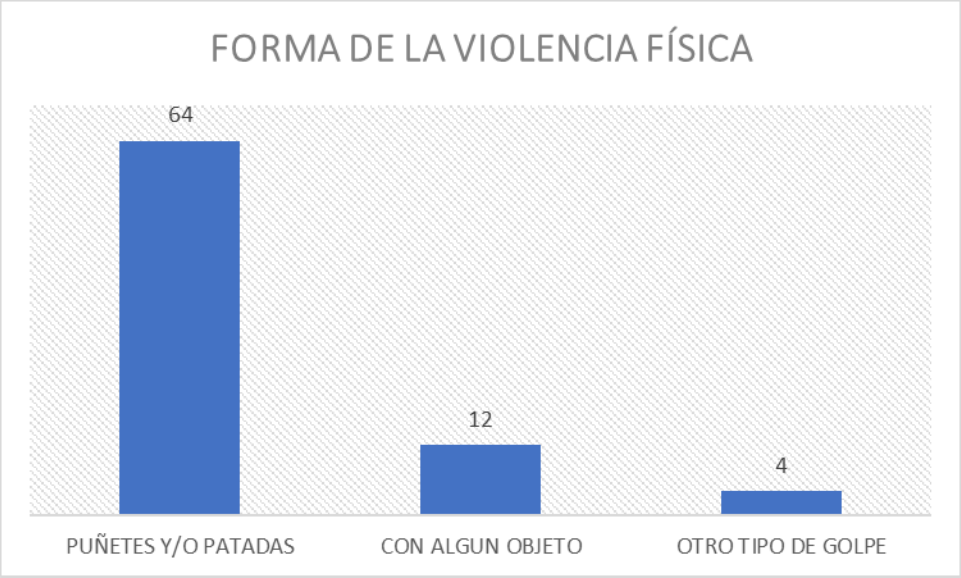


FIGURA 3 Gráfico de los resultados realizado en la pregunta 3 de la encuesta

Elaborado por: los autores

Entre el 2017 y el 2018 se incrementó en 44% el número de cámaras de videovigilancia instaladas por los municipios para la seguridad ciudadana, actualmente en el distrito de El Agustino se cuenta con más de 70 cámaras de seguridad instaladas perteneciente a la municipalidad, los locales y empresas privadas que se ubican en dicho distrito.

En la empresa Real Pez SAC se presentaron distintos escenarios, como se detallan en la tabla 1, que terminaron en agresión física afectando la seguridad de los comensales, para la detección de este tipo de agresiones el local cuenta con dos cámaras de videovigilancia y una computadora.

Tabla 1

Escenarios previos/ posteriores al acto de violencia

Escenarios previos al acto de violencia	***Escenarios posteriores al acto de violencia***
Se observó en los videos que los escenarios previos más frecuentes se inician con las pandillas, personas discutiendo, personas tomando alcohol, uso de celular en la vía pública y robos.	Se observó en los videos que los escenarios posteriores más frecuentes son la llegada del serenazgo, policía, personas corriendo, personas heridas y ambulancias.

Elaborado por: los autores

1.3. Formulación del problema

1.3.1 Problema Principal

¿Cómo identificar posibles escenarios que puedan provocar violencia física en la sucursal principal de la empresa Real Pez SAC ubicado en El Agustino?

1.3.2 Problemas Específicos

a) ¿Cómo identificar personas que generen posibles escenarios de violencia y detectar escenas de violencia física en la sucursal principal de la empresa Real Pez SAC?
b) ¿Cómo prevenir los escenarios posibles a un acto de violencia y como mitigar los actos de violencia en la sucursal principal de la empresa Real Pez SAC?

c) ¿De qué manera se puede articular la prevención y detección de violencia física?

1.4. Objetivos de la investigación

1.4.1 Objetivo General

Crear un sistema que integre interfaces de permitan identificar posibles escenarios violencia física en la sucursal principal de la empresa Real Pez SAC.

1.4.2 Objetivos Específicos

a) Identificar y entrenar el algoritmo que permita identificar personas asociados a posibles actos de violencia y acciones violentas usando las librerías OpenCV y Tensor Flow.
b) Diseñar y desarrollar interfaces que identifiquen personas asociadas a posibles escenarios de violencia para su prevención y detección de escenas de violencia física mediante reconocimiento de movimiento.
c) Desarrollar interfaces para emitir alertas y comunicar cuando se identifique posibles escenarios de violencia o detecte escenas de violencia física.

1.5. Justificación de la investigación

1.5.1 Importancia de la investigación

Debido a que en los últimos años la inseguridad ciudadana aumento y no solo basta contar con dos cámaras de videovigilancia, en la empresa Real Pez SAC en los últimos años han contado con un grado de delincuencia moderado (ver anexo 20) afectando a los ingresos mensuales que varían entre los S/. 5,000.00 a S/. 7,000.00 mensuales (ver anexo 19) ya que los comensales prefieren no ingresar al restaurante por miedo, y al realizar el análisis, el restaurante diariamente perdería un ingreso de 10 a 25 soles por persona dando como resultado una

pérdida anual mayor a S/. 3,000.00 (Ver anexo 21) con un porcentaje del 11% en el año 2018, un 15% en el 2019 y 11% en el periodo 2020.

Esta investigación adquiere importancia puesto que contribuye a la necesidad de procesar, identificar un acto de violencia y prevenir posibles escenarios de violencia física en una de las sucursales de la empresa Real Pez SAC con el uso de la tecnología a través de distintas librerías de Machine Learning para el monitoreo y reconocimiento de acciones violentas como puñetes y patadas de manera inmediata y precisa con el fin de obtener un mejor control de respuesta ante estos incidentes por parte de la municipalidad y/o policías, permitiendo la posibilidad de potenciar la seguridad de los comensales.

1.5.2 Aporte de la investigación

El aporte principal de este estudio es el desarrollo de un modelo que puede ser aplicado en un sistema web, creado con librerías (TensorFlow, OpenCV, keras y numpy) para identificar un acto de violencia y prevenir posibles escenarios de violencia beneficiando no solo a la seguridad de los comensales y del personal que trabaja en la empresa sino también emitió alertas que permitieron automáticamente comunicar al serenazgo y/o empresa de servicio de seguridad.

Asimismo, el uso de la tecnología en este caso permitió fortalecer deficiencias de seguridad para evitar incidentes relacionado a la violencia física en el restaurante debido a que anteriormente no se tenía una precisión adecuada en este proceso.

1.6. Viabilidad de la investigación

1.6.1 Viabilidad Operativa

Para esta investigación la perspectiva utilizada es de los procesos internos el cual se hizo referencia a mejorar la

eficiencia de prevención de seguridad y mejorar la respuesta ante un incidente de seguridad teniendo en cuenta la necesidad del negocio en aumentar sus ingresos mensuales (ver anexo 21).

Para las distintas fases que componen este proyecto se sustenta con la experiencia profesional del personal, resultando unas correctas condiciones operativas que permite tener la seguridad del cumplimiento de los objetivos planteado. (Ver tabla 2) por otro lado, el restaurante a necesidad del negocio realiza capacitaciones constantes a su personal, para este proyecto existe un costo bajo de mantenimiento y al contar con los recursos necesarios la compatibilidad del proyecto al restaurante Real Pez SAC es el más adecuado.

Tabla 2
Requerimientos viables de recursos humanos

ROL	RESPONSABLE	DESCRIPCIÓN
Dueño del producto	Miguel Angel Diaz Vasquez	
Administrador de la empresa	Cesar Diaz Vasquez	
Seguridad	Empresa de Seguridad Huayna	Empresa de servicio de protección de instalaciones tuvo como objetivo la prevención y custodia.
Analista	Espinoza, Willy Rodriguez, Gabriela	Encargado de investigar y analizar el caso de negocio.
Desarrollador	Espinoza, Willy	Encargado de realizar, diseñar las arquitecturas de software e infraestructuras del proyecto, así como del desarrollo.

Tester	Rodriguez, Gabriela	Se encargan de probar el software para detectar las fallas y errores.

Elaborado por: los autores

1.6.2 Viabilidad Técnica

El detalle de la tecnología de videovigilancia que cuenta actualmente el restaurante Real Pez SAC cuenta con 2 cámaras de red del modelo “DS-2CD2185FWD-I” que tuvo hasta 4 megapíxeles de alta resolución con una máxima resolución de1920 × 1080 con conexión a internet y la información es guardad en un disco duro 500gb con NVR que se guarda diariamente (ver tabla 3).

Por otro lado, el restaurante Real Pez SAC cuenta con el personal necesario y con el administrador de la empresa que utilizaron este sistema que cuenta con conocimientos técnicos requeridos para el sistema que se propone.

Tabla 3

Recursos Viables de Hardware

MEDIO	UND	MATERIAL	DESCRIPCIÓN
HARDWARE	2	**Cámaras**	Cámara IPs
	1	**NVR**	Dispositivo interactivo de grabación de televisión y video en formato digital
	2	**PC/Laptop**	Requisitos mínimos: Procesador Intel Core I5, Memoria RAM de 4GB y Disco Duro de 1 TB
	1	**Switch Interno**	Dispositivo de interconexión utilizado para conectar equipos en red formando lo que se conoce como una red de área local (LAN)

	1	**Router**	Es un dispositivo que permite interconectar computadoras que funcionan en el marco de una red.
SOFTWARE	1	**Dominio de red**	Agrupación lógica de servidores de red
	1	**Firewall**	es un dispositivo de seguridad de la red que monitorea el tráfico de red
	1	**Python 3.8**	Lenguaje de programación
	1	**C#**	Lenguaje de programación
	1	**Java Script**	Lenguaje de programación
	1	**HTML**	Lenguaje de etiquetas
	1	**Anaconda Navigator: Jupyter**	Interfaz gráfica de usuario (GUI) de escritorio
	1	**Sistema de Mensaje de Twilio**	Plataforma de comunicaciones, permite desarrollar aplicaciones elaboren funciones de comunicación y registro
	1	**Sistema Web de Recompensa**	Sistema Web con información de imágenes de personas con requisitoria impulsado por el Ministerio del Interior

Elaborado por: los autores

1.6.3 Viabilidad Económica

Para este punto se detalla la relación del costo total del desarrollo de la solución propuesta, en la cual se asume el modelo de una actividad sin fines de lucro, se toma en cuenta diversos factores tales como: cámaras, internet y el uso de repositorio de las imágenes, un servicio de almacenamiento gratuito. Hay que mencionar que las librerías de software utilizadas son open source al hardware el valor asignado es solo por el uso de estos en el proyecto, con una duración estimada de tres meses los costos asignados al proyecto se muestran en la tabla 6.

Tabla 4

Honorarios del Equipo técnico

EQUIPO	PRECIO X HORA	HORAS LABORALES	TOTAL
Analista	S/. 25	150	S/. 3,750.00
Desarrollador	S/. 20	300	S/. 6,000.00
Tester	S/: 15	100	S/. 1,500.00
TOTAL			**S/.11,250.00**

Elaborado por: los autores

Tabla 5

Gastos de los equipos y/o bienes

EQUIPO	TOTAL
Laptop/ Computadora	S/. 2,000.00
Cámara	S/. 1,000.00
Disco Duro	S/. 250.00
Host	S/. 200.00
TOTAL	**S/. 3,450.00**

Elaborado por: los autores

Tabla 6

Costo total para el desarrollo del prototipo

					Costos en Soles
Recursos	Unid	Tiempo (mes)	Valor	Mensual	Total
Hardware					
Computadora	1	3	2,000.00	50.00	150.00

Laptop	1	3	2,000.00	60.00	180.00
Cámara	2	3	2,000.00	60.00	180.00
NVR	1	3	400.00	20.00	60.00
Internet	1	3	80.00	24.00	72.00
Servicio Web	1	3	110.00	110.00	330.00
SUB TOTAL HARDWARE:					972.00
Software					
Open Source	6	3		Sin costos	
SUB TOTAL SOFTWARE:					-
Recursos Humanos					
Dueño de producto	1	3	6,000.00	1,800.00	5,400.00
Administrador de la empresa	1	1	4,000.00	1,000.00	1,000.00
Empresa de seguridad	1	3	1,500.00	500.00	1,500.00
Analista	1	3	5,500.00	1,250.00	3,750.00
Desarrollador	1	3	7,000.00	2,000.00	6,000.00
Tester	1	3	3,000.00	500.00	1,500.00
SUB TOTAL RECURSOS:					17,650.00
SUBTOTAL					18,532.00
10% Imprevistos					1,847.20
Costo total del proyecto					20,379.20

Elaborado por: los autores

Debido a que este es un sistema local y la empresa ya cuenta con los recursos, el proyecto no generó rentabilidad económica al restaurante Real Pez SAC, pero sí produjo beneficios en función a los objetivos estratégicos de la empresa. El financiamiento de este proyecto es autofinanciado en su totalidad por los tesistas, buscando que su desarrollo genere un aporte a la sociedad, mediante ventajas cualitativas, tales como:

— Identificar actos de violencia física
— Contar con un adecuado y preciso monitoreo de las cámaras de videovigilancia.
— Brindar de manera inmediata la comunicación de un acto de violencia física

1.7. Alcance y limitaciones

1.7.1 Alcance

a) Solo engloba el desarrollo de un sistema que utilizando dos modelos de Machine Learning permitan identificar personas con requisitoria en base a la base de datos creada y la detección de violencia física en solo dos acciones específicas como puñete y/o patada.

b) El objetivo de estudio solo es en la sucursal principal ubicado en la calle Av. Riva Agüero.

c) La base de datos utilizada para identificar personas con requisitoria es del programa de “Recompensas.pe”

d) El local solo cuenta con dos cámaras (Cámara número 1 ubicada en la puerta principal y cámara número 2 ubicada dentro del restaurante).

1.7.2 Limitaciones

Las principales limitaciones fueron:

- Para el entrenamiento de los modelos neuronales no se realizó con la data almacenada por la empresa Real Pez SAC.

- No se cuenta con una base de datos de información de la mayoría las personas con requisitoria en el Perú.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes de la investigación

En este capítulo se presenta los antecedentes de relevantes en estos últimos años, bases teóricas para el desarrollo del proyecto y la definición de términos básicos.

2.1.1 Antecedente Internacional

Ammar, S. M., Anjum, M., Rounak, T., Islam, M., & Islam, T. (2019), en su trabajo de investigación denominada: “Uso de algoritmos de aprendizaje profundo para detectar actividades violentas”. Este proyecto se basó en identificar actividades violentas en videos mediante una red neuronal profunda que permitió extraer características de nivel de cuadro de un video, utilizando una red neuronal convolucional con un modelo ImageNet previamente entrenado con una variante de memoria a corto plazo que usan capas completamente conectadas y unidades lineales rectificadas con fugas. Se utilizó la librería Tensor Flow para implementar y entrenar la red convolucional que permite reconocer un video como violento o no violento, pero en lugar de simplemente usar la información transmitida por un paso en

el pasado, aprenden a través de muchos pasos, lo que les permitió vincular causas y efectos que ocurren durante un período prolongado de tiempo.

Coşkun, S. (2019), su trabajo lleva por título: “Desarrollo de un sistema de reconocimiento facial para la seguridad en interiores”. Se trata de un proyecto especializado en un sistema de seguridad con reconocimiento facial para que el sistema de seguridad advierta al propietario de la casa, que compare con los rostros introducidos previamente, y si el rostro detectado no coincide con un rostro que pertenece a los propietarios, entonces advierte enviándoles una imagen del rostro no coincidente a la aplicación de Android a través de Internet, cuando se detecta una persona extraña con una cámara, también tuvo una aplicación de computadora que aprende ,detecta y envía imágenes y una aplicación de Android enciende / apaga la aplicación informática y recibe imágenes. Este trabajo se relacionó con la investigación aquí planteada utilizando OpenCV para la visión artificial de la cámara posicionada. la Red neuronal convolucional para generar incrustaciones que pertenecen a cada cara y guardarlas. Torch construye un clasificador usando algoritmos de redes neuronales en este caso se utilizó una red ya entrenada denominada face_recognition. Este estudio demostró los resultados de los métodos de detección y reconocimiento de rostros recopilado 900 imágenes, cada una de las cuales tuvo 2 caras, de un video que dura 30 segundos en un entorno iluminado para construir un conjunto de datos para el entorno iluminado dado como resultado la precisión de 0,71 y ambientes oscuros 0,38.

2.1.2 Antecedente Nacional

A. Lumba, I. Yahuarcani, C. Cortegano, et al. (2019), realizaron un proyecto que se basa en el reconocimiento de acciones básicas de violencia en video mediante la CNN YOLOv2, su objetivo fue el rastreo de golpes tipo puñete y golpes tipo patada. Su investigación se centró en el reconocimiento de violencia en vídeo y se utilizó la información de actividad humana enunciada por componentes visuales en vídeo. Este trabajo

se relacionó con la investigación aquí planteada es el uso de la segmentación espacial de la violencia en vídeo y el método planteado se estableció en las técnicas de aprendizaje no supervisado. la técnica utilizada es YOLOv2 ya que tuvo una mayor precisión de múltiples cuadros delimitadores por celda de cuadrícula y solo que un predictor de cuadro delimitador sea responsable de cada elemento. de acuerdo con sus resultados del método propuesto al utilizar el descriptor STIP con la probabilidad de acierto de 90.4 %y una segmentación espacial de violencia en el video con una probabilidad de acierto de 74 %.

Barreto Rodriguez, R. M., & Lizarraga Mendoza, D. J. (2019). la tesis “Modelo de sistema de reconocimiento facial para el control de la trata de personas” en la tesis describen sobre la forma para obtener las secuencias de video donde se hizo un previo análisis de un conjunto de fotogramas captados de un video y que al procesar cada fotograma extraído hace la detección de rostro hallados. la técnica dentro de la de red neuronal denominada “Feedforward” en el proceso se emite resultados generados por una red neuronal que hizo un análisis comparativo con características de imágenes de las personas comparando las distancias para saber cuál es el próximo más cercano, una vez encontrado su recepción en un mensaje se pudo encontrar a la persona extraviada. Las herramientas utilizadas por los investigadores son Python y Torch, usa redes neuronales profundas y FaceNet de visión artificial, que emplea el método de disminución de triplete de reconocimiento facial en la mayoría son cajas negras basados en su arquitectura. Además, utilizan el Servicio de Kairos que tuvo servidores de código licenciado que cuenta con la detección de caras, detección de rasgos faciales, agrupamiento de rostros.

Ramírez Ticona, J. T. (2017). propuso en su tesis una “Propuesta de un modelo para el reconocimiento de escenas violentas en video” a través de un modelo propuesto por los autores de detección de eventos violentos con videos en vivo segmentando clips de un segundo para extraer atributos a través del algoritmo STIP así mismo detectando espacio dentro del clip si contuvo violencia o no .El procedimiento de categorización se da con un clasificador denominado SVM que una vez

detectada la escena violenta detecta a la persona que actúen en el acto utilizando la técnica YOLO. Obtuvo como entradas los histogramas de palabras visuales de cada video y el modelo permite dar como resultado si el video contuvo violencia o no. para generar los vectores con propiedades en dividir los videos en clips, se utiliza una cuantificación vectorial a través del algoritmo K-mean que cumple una demostración con videos aleatorios, de los cuales son generados los puntos de interés y sus respectivos descriptores. por lo tanto, el algoritmo no siempre produce los mismos resultados con un mismo conjunto de datos presentando un buen resultado al generalizar en problemas de clasificación.

2.2 Bases teóricas

2.2.1 Violencia Física en la vía pública

Las ciudades son escenarios de distintas relaciones donde las personas interactúan y están en constante movimiento y roce; para París Albert (2005), “los conflictos son parte de la vida social, sus transacciones y a los intercambios que lleguen a expresar de diferentes formas”, y estos conflictos pueden convertirse en violencia física cuando se ejerce la fuerza física con finalidad de causar dolor, malestar, daño o sufrimiento físico a otra persona (Asociación Española de Periatría, 2018).

a) Factores que producen violencia física

Según las noticias, revistas, periódicos y páginas web que se difunden sabemos que los factores que ocasionan la violencia física en vía pública varían de distintas formas pueden ser de carácter social, familiar, comunitario e individual. (Asociación Española de Periatría, 2018). Entre los factores más comunes estuvieron:

- Por las condiciones de vida y de trabajo;
- La insuficiencia de educación ciudadana;
- El uso nocivo del alcohol y drogas;
- El mal uso del tiempo;

- La experiencia de violencia familiar;
- No tener unos correctos valores tradicionales;
- La marginalidad social;
- La conducta dominadora que mayormente es masculina hacia su pareja;
- El trastorno de personalidad antisocial.

b) Consecuencias que producen

La violencia física ocasiona graves problemas de salud física y mental largo plazo como a corto, afectando directamente a la familia. (Asociación Española de Periatría, 2018) Este tipo de consecuencias pueden ser:

- El homicidio o suicidio;
- Denuncias en la comisaría;
- Lesiones graves a terceros;
- Padecer de depresiones y/o problemas con el alcohol;
- Efectos en la salud física para las limitaciones de la movilidad;
- Estrés postraumático, trastornos de ansiedad, problemas con la salud mental.

2.2.2 Sistema de Video vigilancia sobre IP

Un sistema de vigilancia consiste en un conjunto de dispositivos electrónico e instalaciones destinadas a proveer y dominar los riesgos con el fin de facilitar seguridad a las personas y bienes materiales, así como ajustar de caída la pérdida ante incidentes como se detalla en la figura 4. (Barradas Arenas, Bárcenas Cortes, Sánchez Hernández, & Hernández Chan, 2017)

Los sistemas de videovigilancia se componen de un medio de trasmisión que puede ser con cable o sin cable, de un grabador, un disco duro donde guardar las grabaciones y las cámaras para supervisar el lugar las 24 horas.

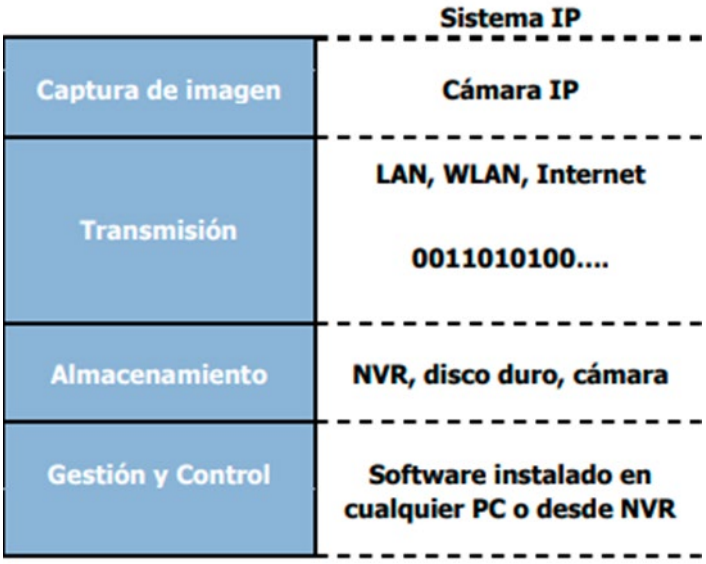


*FIGURA 4 C*omponentes del sistema de CCTV

Elaborado por: los autores

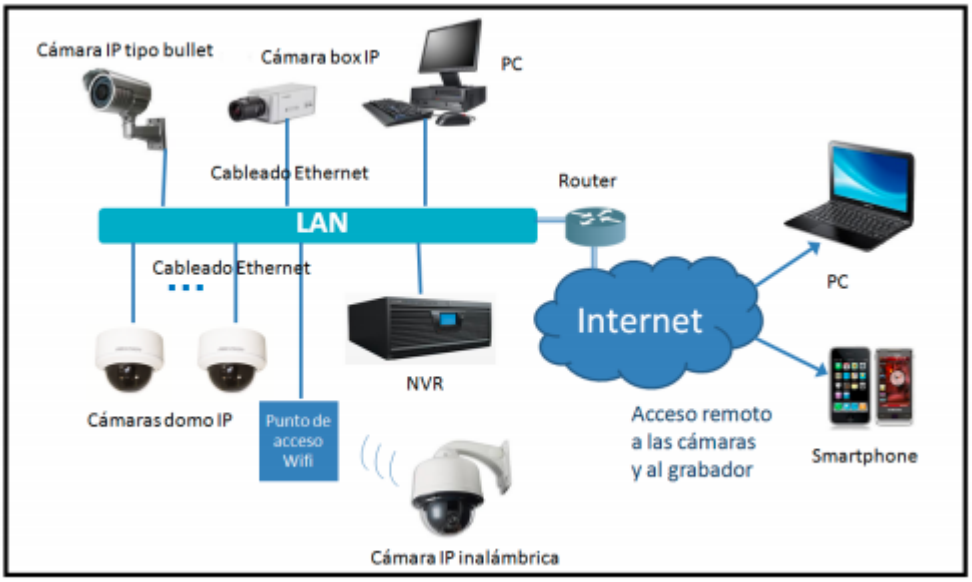


FIGURA 5 Infraestructura de CCTV sobre IP

Elaborado por: los autores

En la Figura 5 se observa una infraestructura básica de un CCTV sobre IP.” (Martí Martí, S, 2013). los elementos utilizados en un CCTV sobre IP son los descritos a continuación:

a) Cámara IP

Una cámara es definida por la Real Academia Española “Aparato que registra imágenes y sonidos en soporte electrónico, y los reproduce”.

Una cámara IP se denomina un tipo de cámara de vigilancia con conexión IP. la cámara está lista para conectarse a un router que se puede monitorear en todo momento lo que está pasando desde una computadora o un celular telefónico.

b) NVR

Un disco duro o HD (igual que el de los ordenadores, aunque de mayor resistencia). Se puede conectar al NVR un monitor TFT-LCD para visualizar las grabaciones, y un teclado especial para controlar el movimiento y/o zums desde el propio grabador. El NVR puede conectarse en cualquier parte de la LAN, lo que permite que comparta espacios con otros equipos de red equipados con climatización y sistemas de alimentación ininterrumpida (SAI). (Martí Martí, S, 2013)

El NVR (Network Video Recorder), a diferencia del DVR del caso analógico, puede no ser parte del sistema, ya que cualquier computadora en la intranet o en internet se pudo acceder directamente a las cámaras y almacenar las imágenes en su propio disco duro. (Martí Martí, S, 2013)

2.2.3 Reconocimiento Facial

El reconocimiento facial es estudio Biometría, donde se tuvo como fin desarrollar distintos métodos automáticos para identificar o validar el rostro de personas mediante características físicas ingresadas anteriormente. En estos últimos años, utilizar el reconocimiento facial se ha proyectado por todo distintas partes del planeta y aplicándolas en distintos modos como identificación de conductas, sistemas de monitoreo de acceso a equipos o ambientes, sistemas de seguridad, etc. y usadas de distintos modos, a esto se suma las mejoras notables de la tecnología donde no solo un sistema puede contar con esto sino hasta un celular o una laptop

puede contar con esta tecnología para ser desbloqueado. (Barraza, S. L., Thuillier, E., Will, A., & Rodriguez, S. A, 2013).

a) Reconocimiento Facial en video

Esto en los últimos años, se convirtió en unos de los campos del área de la biometría con gran desarrollo donde a nivel mundial distintas y grandes empresas han investido para tener más exactitud y precisión en el reconocimiento puesto que esta tecnología presenta grandes ventajas, Villalón de la Vega, D. E (2012) presenta lo siguiente:

> El video permite y brinda más datos para el estudio, ya que se cuanta con más cuadros para realizar la clasificación. Asimismo, presenta continuidad temporal, dando como resultado la información de clasificación obtenida de los cuadros de alta propiedad para procesar los cuadros de baja calidad. Por otro lado, el video permite realizar seguimiento de rostros, mediante los cambios de efectos y expresiones faciales se pueden detectar y tener una mayor precisión. (p.15)

2.2.4 Machine Learning

El aprendizaje automatizado o *Machine Learning*, en inglés, es una rama de la inteligencia artificial que permite que las computadoras puedan contar con la capacidad de aprender, sin tener que ser programadas explícitamente, centrándose desarrollar sistemas informáticos que pueden cambiarse cuando este se expone a nuevos datos (Gori, 2017). Dicho de otro modo, Norman, A. T (2018) menciona que en la figura 6 que “el aprendizaje automático permite a los ordenadores aprender desde la experiencia de esta misma.” (p.1)

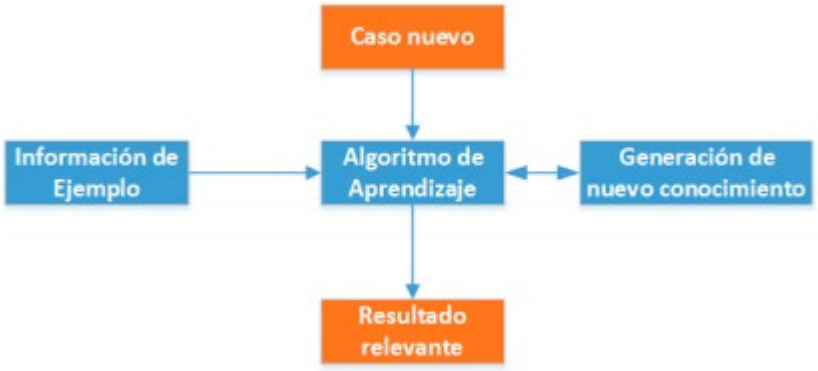


FIGURA 6 Muestra de aprendizaje automático

Fuente: Norman, 2018

Para continuar se debe esclarecer la diferencia entre el concepto de programación y del algoritmo de entrenamiento (Norman, A. T, 2018):

- La programación explícita cuando empieza que una persona ingresa las instrucciones para que un ordenador las siga, realizándolo de forma manual donde la persona es obligatoria que escriba las reglas sino el ordenador no sabría qué instrucción seguir.

- El algoritmo de entrenamiento permite que el ordenador pueda descubrir la respuesta de forma automática mediante una secuencia de retroalimentación que pueda corregir de forma automática la precisión luego de varias predicciones.

2.2.5 Aprendizaje Supervisado

Es un tipo de aprendizaje automatizado, en la figura 7 se inicia con un conjunto de información ya establecidos con una cierta comprensión de clasificación de esos datos, aprenden de problemas que ya han sido resueltos para que puedan detectar o predecir salidas futuras basadas en distintas características o comportamientos previamente en los datos ya guardados esto depende de datos que han sido etiquetados (Hurwitz y Kirsch, 2018).

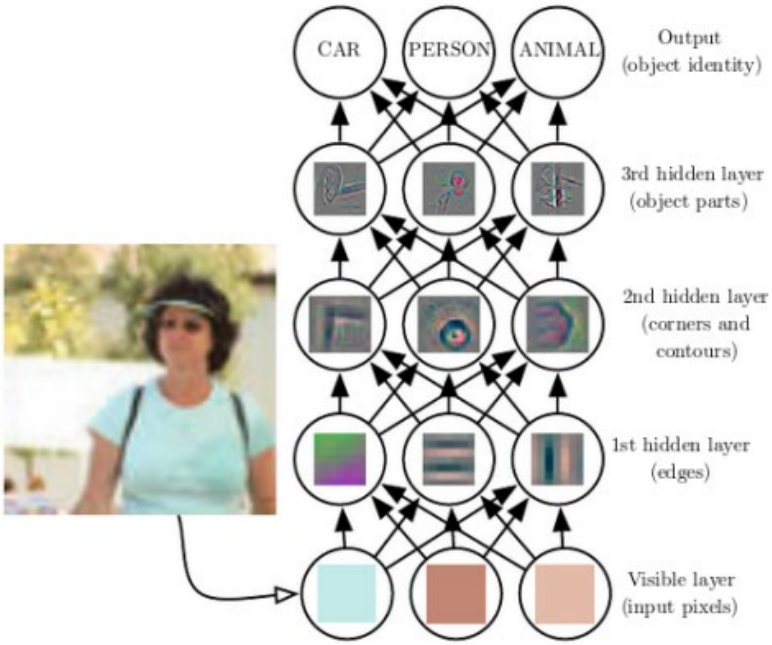


FIGURA 7 Modelo de aprendizaje profundo

Fuente: Hurwitz y Kirsch, 2018

Silva (2020) en su tesis menciona “los algoritmos de aprendizaje supervisado utilizan un conjunto de datos de entrenamiento, donde cada ejemplo está asociado con una etiqueta u objetivo, el cual se espera como resultado del ejemplo dado. los parámetros de la red se corrigen de acuerdo con la magnitud del error”.

2.2.6 Redes Neuronales

Fernández L. (2017) menciona “Las redes neuronales artificiales son un paradigma de programación de inspiración biológica que se permite a un algoritmo aprender a partir de un conjunto de datos observacionales” (p.11)

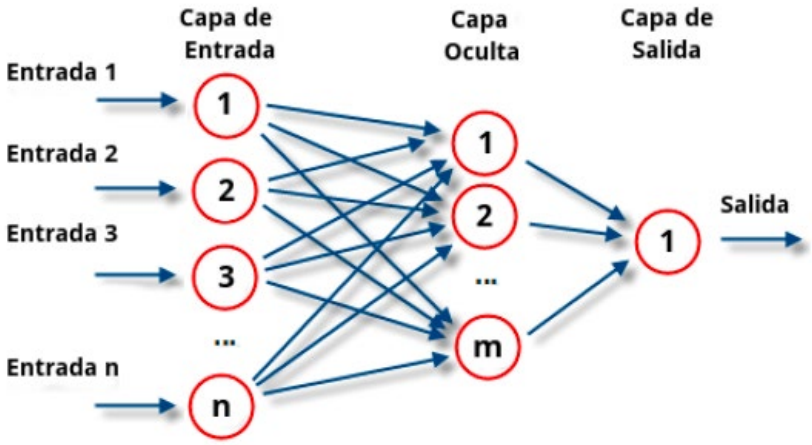


FIGURA 8 Función de red neuronal artificial

Fuente: Fernández Luis, 2017

En la figura 8 se observa cómo está constituida por neuronas interconectadas y arregladas en tres capas, los datos ingresan por medio de la “capa de entrada”, pasan a través de la “capa oculta” y salen por la “capa de salida”, en la capa oculta puede estar constituida por varias capas. (Macedo & Chavez, 2016, p.60)

a) Redes Neuronales Convolucionales

Esto son un tipo de redes neuronales, donde el desarrollo fue debido a la investigación neurobiológica de células nerviosas selectivas y localmente sensibles en la corteza visual (Fernández, 2017, p.11).

Esta red usualmente es utilizada en la mayoría de proyecto que se basan en el procesamiento de imágenes, debido a que la última capa oculta se cuenta con la función llamada Softmax, que permite realizar una conexión entre todas las neuronas maxpooling y las convoluciones que se está empleando en la red. (Arreola, 2019)

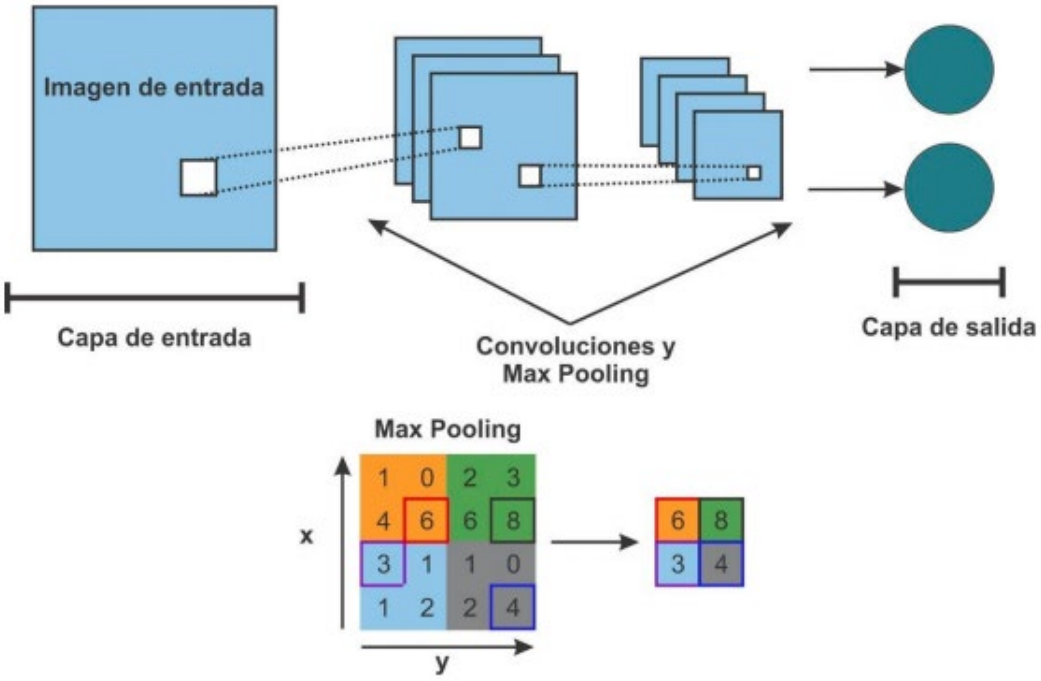


FIGURA 9 Modelo de redes neuronales convolucionales

Fuente: Arreola J., 2019

b) Capas de las redes neuronales

Para iniciar una red neuronal se realiza de dos maneras dependiendo de la secuencia de capas en el grafo de la relación de capas (ver figura 9). Donde en la red neuronal está compuesta de tres capas, capa de entrada, la capa oculta y la capa de salida en el cual:

- la capa de entrada se relaciona a las longitudes de onda obtenidas de variables independientes.
- la capa de salida se relaciona a la variable de salida, en la cual se desea predecir.
- la capa oculta se halla en las neuronas.

2.2.7 Programa Recompensa

En el 2017 el gobierno peruano a través del Ministerio del Interior decidió lanzar una campaña nacional que invita a la ciudadanía a colaborar con la captura de los delincuentes registrados en el Programa de Recompensas del Ministerio del Interior, ingresando al sitio www.recompensas.pe el público puede encontrar información general de la campaña y el programa, la lista de prófugos y capturados, la información de prensa al respecto y los vídeos con información de las personas con requisitorias más buscadas e incluso puede conocer los gustos y aficiones de estas personas.

A través de la plataforma digital única del Estado, el Programa de Recompensas del Ministerio del Interior (2019) se define como un programa anónimo y seguro, que funciona a través de llamadas gratuitas que ayudan a la justicia con información que permite la investigación, captura y entrega de criminales, presuntos delincuentes y terroristas a cambio de una recompensa económica.

En el Decreto Legislativo N.º 1180, se establecen las disposiciones generales sobre la ganancia de una recompensa para promover y lograr la captura de personas más buscadas. Esta información puede ser utilizada con la finalidad de promocionar la campaña (ver anexo 26).

2.2.8 Técnicas y métodos a utilizar

Para el procesamiento e instalación de las librerías se optó por utilizar la suite anaconda que es una GUI de computadora de código abierto para el desarrollo del uso de la ciencia de datos en Python el cual facilita la administración de paquetes y entornos sin usar comandos de línea de comandos.

a) Librerías utilizadas

a.1) Numpy: Es una librería de código abierto que gracias a esta la computación numérica con el sistema de programación Python, se basa en el trabajo de las bibliotecas Numerical y Numarray. Numpy proporcionando una estructura de datos con la implementación de matrices multidimensionales, garantizando cálculos eficientes con matrices. (Numpy, 2020, p.1)

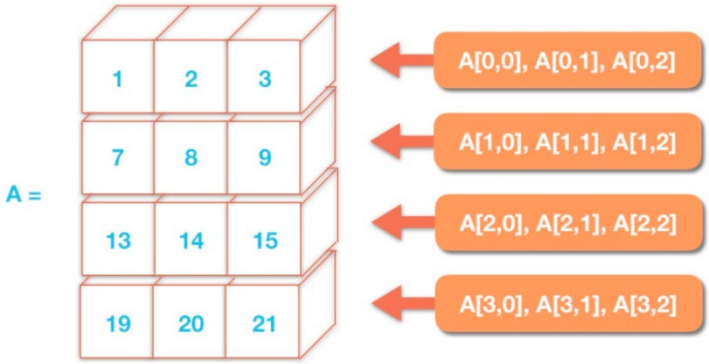


FIGURA 10 Matriz de estructura de datos

Fuente: Numpy. 2020

a.2) Keras: Es una API para modelos complejos de aprendizaje profundo desarrollada en Python, permite ejecutar sobre la plataforma de aprendizaje automático Tensor Flow que permite realizar una ejecución eficiente de operaciones de tensor de bajo nivel en CPU, GPU o TPU.

Las estructuras de datos centrales de Keras son capas y modelos, donde el tipo de modelo más simple es el modelo secuencial, que es una lineal de capas. (Keras, 2020)

a.3) Tensor Flow: Es una librería de código abierto para el aprendizaje automático donde se construye y entrena redes neuronales para descifrar, detectar y correlacionar análogos al aprendizaje y razonamiento usados por los humanos. (Tensor Flow, 2020)

a.4) Face Recognition: Garcés, A., & Jurado, M. (2016) menciona:" Es una librería para el procesamiento de imágenes, comparando las características faciales de la imagen ingresada con las imágenes almacenadas en el sistema por medio de un algoritmo de reconocimiento facial", es un vector proveniente de 128 dimensiones.

b) Capas en la red neuronal a utilizar

Asimismo, las capas que se utilizaron en este proyecto son:

b.1) Gaussian:

un modelo de mezcla gaussiana intenta encontrar una mezcla de distribuciones de probabilidad y gaussianas multidimensionales que modelen cualquier conjunto de datos de entrada

b.2) Conv2d:

Capa crea un núcleo de convolución que es viento con capas de entrada que ayuda a producir un tensor de salidas

b.3) Flatten:

Capa su función es de aplanar en un tensor cambiar la forma del tensor para que tenga que sea igual al número de elementos.

b.4) Dense:

la capa densa es la capa regular de la red neuronal profundamente conectada las capas pieza por pieza la que da flexibilidad a la API funcional.

c) Flask Enrutamiento: Conexión de Python con HTML.

Flask es auto denominado como un framework de Python para crear aplicaciones web, que permitieron enrutar las solicitudes. para ingresar a las distintas Irlos podemos utilizar 2 métodos, para este proyecto se utilizó con el método GET y POST, que son los métodos que se utilizó desde un navegador web donde:

c.1) GET: Se realiza una petición para obtener un recurso del servidor web. Es el método más utilizado.

c.2) POST: Aunque con el método GET también podemos mandar información al servidor (por medio de parámetros escritas en la URL), se utilizó el método POST para enviar información a una determinada URL. Se utilizó los formularios HTML para enviar información al servidor por medio del método POST.

por defecto las rutas indicadas en las funciones router sólo son accesibles utilizando el método GET. (Domingo, J. 2018)

d) AJAX

AJAX significa JavaScript asíncrono y XML (Asynchronous JavaScript and XML), javaScript es un lenguaje de programación muy conocido. Entre otras funciones, gestiona el contenido dinámico de un sitio web y permite la interacción dinámica del usuario. XML es otra variante de un lenguaje de marcado como HTML, como lo sugiere su nombre: extensible Markup Language. Mientras HTML está diseñado para mostrar datos, XML está diseñado para contener y transportar datos. (Gustavo B., 2019)

FIGURA 11 Uso de AJAX

Fuente: Krall Cesar, 2018

En la figura 11 como menciona Krall, C. (2018) se puede observar que el cliente tuvo una página web cargada (puede ser una página web completa, o sólo el esqueleto de una página web). El cliente sigue trabajando y en segundo plano (de ahí se dibujó con líneas punteadas las comunicaciones) le dice al servidor que le envíe un paquete de datos que le hacen falta. El

servidor procesa la petición. Ahora la respuesta es mucho más rápida: no tuvo que elaborar una página web completa, sino sólo preparar un paquete de datos. por tanto, el tiempo de respuesta es más rápido. El servidor envía el paquete de datos al cliente y el cliente los usa para cambiar los contenidos que se estaban mostrando en la página web.

e) ¿Cómo funciona las librerías a usar?

Como lo indica Geitgey, A. (2020) en su repositorio para el reconocimiento facial, se pueden utilizar las librerías OpenCV y dlib, para lo cual usando la función “face_recognition” permite procesar las imágenes dando un input al modelo para que se envíe un output (ver figura 12)

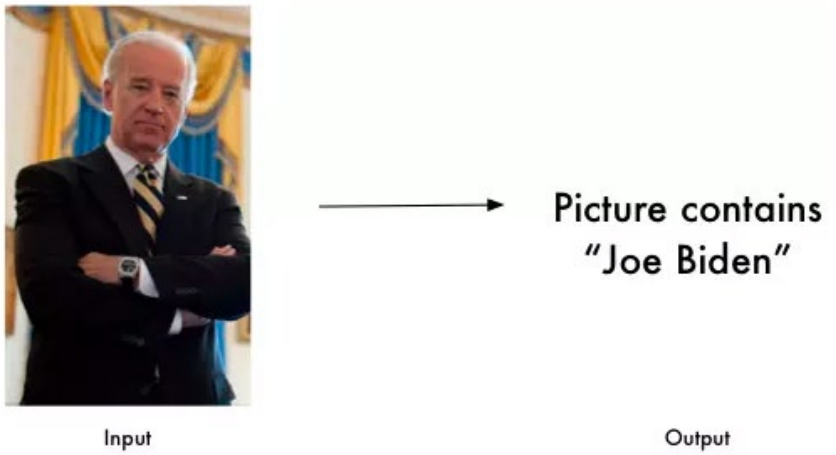


FIGURA 12 Procesamiento del input y output

Fuente: Geitgey, A. 2020

Como se observa en la figura 13, el reconocimiento facial a través del aprendizaje profundo y Python utilizando el método face_recognition genera un vector de características numéricas con valor real de 128d por cada rostro identificado, no se realizó un entrenamiento a la red, lo que se utilizó fue una red previamente entrenada para luego emplearla en construir inserciones de 128 dimensiones para cada una de las caras de la base de datos.

[-0.23, -0.54, ..., 0.27]

FIGURA 13 Reconocimiento con la función "face_recognition"

Fuente: Geitgey, A. 2020

Al contar una base de datos de las imágenes se detecta la cara fotograma por fotograma utilizando la función read() y “face_locations”, asimismo se requiere de la función “face_encoding”, que genera los vectores de 128 dimensiones como se detalla en la figura 14.

```
# encuentra y codifica todas las caras existentes en el fotograma
face_locations = face_recognition.face_locations(rgb_small_frame)

face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
```

FIGURA 14 Código utilizado en la detección de caras

Fuente: Romo Ricardo, 2019

La comparación de la cara entrenada se utilizó la función “compare_faces”. Donde se toma la cara nueva que está leyendo el modelo y la compara con las caras ya previamente entrenadas y el resultado es una orden que devuelve un valor como “True” por cada vez que haga match con una cara entrenada. Esta matriz realizada se pasa a otra función “face_distance” que devuelve la distancia en un espacio entre los vectores, de esta manera, las caras más parecidas son más cercanas. para presentar los resultados, se realiza a través de las librerías OpenCV, que permite pintar

los cuadros y texto sobre el fotograma y mostrarlo en una ventana de visualización.

Asimismo, la librería numpy realiza una manipulación básica de arrays, permitiendo leer las imágenes recibidas (ver figura 15)

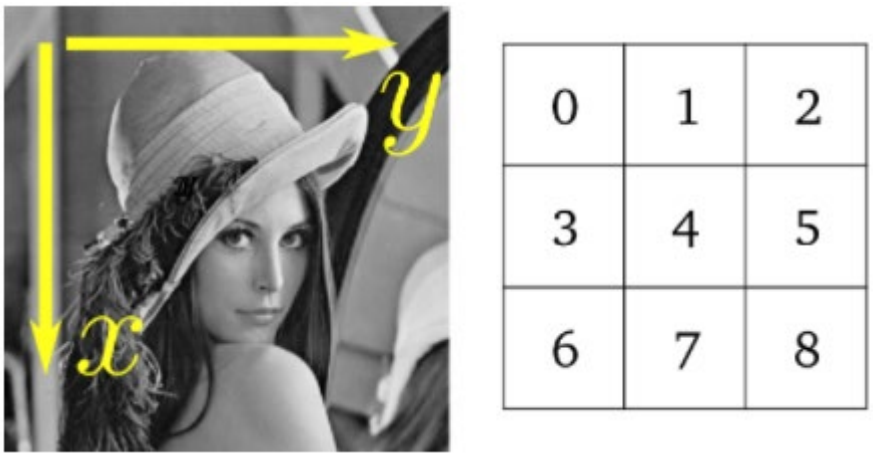


FIGURA 15 Imagenes usando numpy

Fuente: Numpy, 2020

Para la detección de acciones violenta se analizó las librerías keras que se encuentra incluida en el TensorFlow y el OpenCV, tomando de referencia el ejemplo de Bortolotti N. para Bortolotti N. (2017) en su blog, donde detecta un objeto al consumirlo dentro del video (pizzas, cakes) usando la librería TensorFlow, numpy y OpenCV.

El entrenamiento de este modelo inicia al leer el video y separarlo por frames, para posteriormente tratarlos como imágenes con el fin de analizar cada imagen en Tensor Flow, para segmentar el vídeo en fotogramas y redimensionarlo el video se utiliza OpenCV colocando denominadores para garantizar que, al detectar estas etiquetas dentro del video, cuando se encuentra una etiqueta de objeto prohibido con una probabilidad certera.

```
1  import sys
2  import os
3  import cv2
4  import numpy as np
5  import tensorflow as tf
6
7  sys.path.append("..")
8
9  from object_detection.utils import label_map_util
10
11 MODEL_NAME = 'ssd_mobilenet_v1_coco_11_06_2017'
12 MODEL_FILE = MODEL_NAME + '.tar.gz'
13 DOWNLOAD_BASE = 'http://download.tensorflow.org/models/object_detection/'
14 PATH_TO_CKPT = MODEL_NAME + '/frozen_inference_graph.pb'
15 PATH_TO_LABELS = os.path.join('data', 'mscoco_label_map.pbtxt')
16 NUM_CLASSES = 90
17
```

FIGURA 16 Programación de la rutina

Fuente: Bortolotti Nicolas, 2017

En la figura 16 en las primeras líneas se observa el importe de las librerías a usar, posteriormente en la línea 11 hacen el llamado del dataset donde se almacena los videos a procesar dando como resultado la identificación del tipo de alimento que tuvo la persona (Ver figura 17).

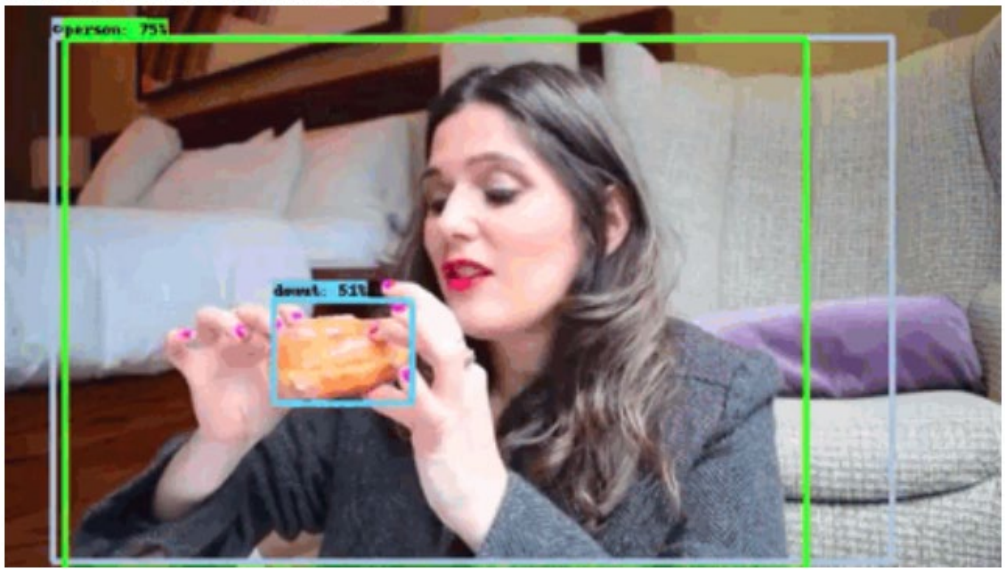


FIGURA 17 Extracto video ensamblado con detección visual

Fuente: Bortolotti Nicolas. 2017

f) Validación de métricas de calidad

El método para verificar las métricas de calidad, la cual permitió analizar la precisión y exactitud en función al objetivo específico 3, se utilizó la confusión o error Matrix es una tabla (ver figura 18) que describe la debilidad del modelo de aprendizaje supervisado con los datos de pruebas que validaremos en el capítulo IV, se llama “matriz de confusión” detectar dónde el sistema está confundiendo dos clases donde se muestra:

— TP (verdadero positivo) es el número de resultados verdaderos y la predicción del modelo es verdadera también.
— TN (verdadero negativo) es el número de resultados que fueron negativos y la predicción del modelo es negativa.
— FP (falso positivo) es el número de resultados que fueron negativos y la predicción del modelo es positivo.
— FN (falso negativo) es el número de resultados que fueron positivos y la predicción del modelo es negativo.

FIGURA 18 Error Matrix

Fuente: Fayrix, 2019

f.1) Exactitud:

La exactitud, es una métrica de calidad que obtuvo el total del número de las predicciones que son correctas hechas por el modelo, cuando los dos casos a evaluar tuvon la misma proporción de cantidad. en caso tenga un desbalance, la métrica no es fiable.

$$\text{Exactitud} = \frac{TP+TN}{TP+TN+FP+FN}$$

f.2) Precisión:

La métrica usualmente más utilizada para la identificación en imágenes es la precisión, la cual indica las predicciones positivas del modelo de un total de elementos identificados como positivos permitiendo, minimizando los falsos positivos. la métrica permite responder la pregunta: “¿Qué proporción de identificaciones positivas fue correcta?”

$$\text{Precisión} = \frac{TP}{TP+FP}$$

f.3) Recall:

La métrica recall, perite medir la proporción de casos positivos reales que se han identificado correctamente en el modelo, es decir, es la capacidad del clasificador para hallar todas las muestras positivas del número total de casos positivos en todo el conjunto de datos evaluados, que permite responder: “¿Qué proporción de positivos reales se identificó correctamente?”

$$\text{Recall} = \frac{TP}{TP+FN}$$

g) Metodologías a usar

g.1) Metodología RUP

Las siglas RUP en ingles significa Rational Unified Process (Proceso Unificado de Rational) es un producto del proceso de ingeniería de software que proporciona un enfoque disciplinado para asignar tareas y responsabilidades dentro de una organización del desarrollo. Su meta es asegurar la producción del software de alta calidad que resuelve las necesidades de los usuarios dentro de un presupuesto y tiempo establecidos. El ciclo de vida del software del RUP se

descompone en cuatro fases secuenciales Chacón, J. C. R. (2006) (ver figura 19).

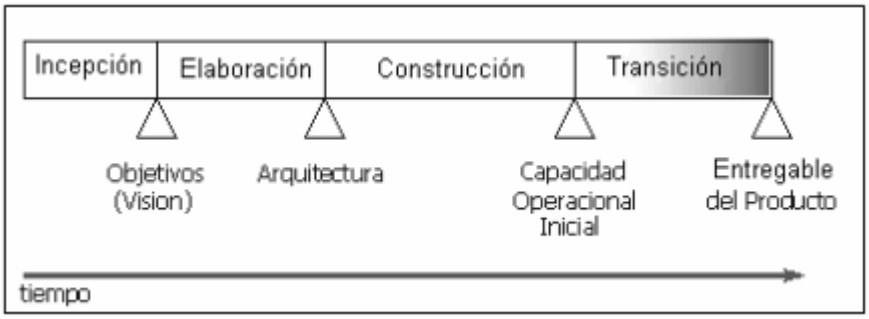


FIGURA 19 Fases del RUP

Fuente: Rueda Chacón, 2016

g.2) Metodología CRISP-DM

CRISP-DM (Cross Industry Standard Process for Data Mining) proporciona una descripción normalizada del ciclo de vida de un proyecto estándar de análisis de datos, de forma análoga a como se hace en la ingeniería del software con los modelos de ciclo de vida de desarrollo de software. El modelo CRISP-DM como se observa en la figura 20 que ocultar las fases de un proyecto y las relaciones entre estas actividades, es más completo porque tuvo en cuenta la aplicación al entorno del interés de los resultados, y por ello es la que más se acomodó popularmente. (Azevedo y Santos, 2008)

Crisp-DM es una constitución europea desarrollada por 3 grandes personas en proyectos de minería de datos que son SPSS, NCR Y Daimler Chrysler. la metodología es desarrollar los proyectos de datos bajo un desarrollo estandarizado y validación de tal forma que se desarrollen proyectos rebajando los costos que impacten en el negocio.

FIGURA 20 Fases de CRISP-DM

Fuente: Contreras & Sánchez, 2020

2.2.9 Reglas de Negocio

La empresa Real Pez SAC solo maneja tres reglas internas para tres procesos importantes que se maneja en el restaurante:

a) Venta de bebidas alcohólicas

Para el proceso de venta de bebidas alcohólicas, la primera regla de negocio se tuvo que solo la venta de bebidas alcohólicas presentando el DNI de la persona, posteriormente el mesero debe identificar la cantidad de personas que se encuentren en la mesa y consumieron la bebida, se procede a ejecutar la segunda regla si en la mesa hay más de 6 personas el máximo de venta de cervezas es de 12, en caso contrario la venta no puede exceder de 5 cervezas. Finalmente, si la persona puede provocar u ocasionar algún tipo de acto que atente contra los comensales y/o personal de la empresa se procede a ejecutar el proceso de comunicar algún hecho delictivo.

2.2.9.2 Vigilancia de cámaras de seguridad

El proceso de vigilancia de cámaras de seguridad la única regla de negocio que se tuvo es que al detectar un acto delictivo se proceda a llamar a las autoridades correspondientes ante cualquier caso de violencia o incidente fuera y dentro del local.

2.2.9.3 Comunicación de algún hecho delictivo

Para el proceso de comunicación de algún hecho delictivo, la única regla de negocio que al realizar la comunicación debe ser informado a las autoridades correspondientes como al dueño de la empresa, asimismo en caso se requiere se tuvo que brindar los videos de seguridad a las personas correspondientes.

2.3 Definición de términos básicos

Algoritmo: Definir los conceptos (conjuntos), relaciones y funciones necesarias para establecer las cláusulas que se ejecuten una acción o resolver un problema. (Rincón Chaparro, M. Á., 2017)

Aprendizaje: Palma-Orozco, R., García-Leyva, E., & Ruiz-Ledesma, E. F. (2020) dice que la educación se corrobora el hecho de que el aprendizaje es significativo cuando logra ser aplicado en una o más áreas del conocimiento."

Detección: Hace referencia al verbo descubrir, detectar o percibir cierto fenómeno o manifiesto de la existencia de una cosa.

Prototipo: Como lo menciona la Real Academia Española (2019), "es ejemplar original o primer molde en que se fabrica una figura u otra cosa" (párr.1).

Dataset: Como lo indica Gonzáles (2018) la define como la materia prima del sistema de predicción, que contuvo la data histórica que se usa para entrenar y probar al sistema que detecta los patrones.

CNN: Es una red neuronal convolucional donde su principal ventaja es que cada parte de la red se le entrena para realizar una tarea, reduciendo el número de capas ocultas para un entrenamiento más rápido. (Calvo, 2017)

Escenarios: Según la Real Academia Española (2020) es un conjunto de circunstancias que rodean a una persona o un suceso.

Procesamiento de imágenes: Mejorar el aspecto de las imágenes y hacer más evidentes en ellas ciertos detalles que se desean hacer notar, se puede en general hacer por medio de métodos ópticos, o bien por medio de métodos digitales, en una computadora. (Malacara,2015)

CAPÍTULO III

METODOLOGÍA

3.1 Diseño metodológico

Para lograr los objetivos planteados, se tuvo un enfoque metodológico donde se realizó distintas etapas en el proyecto, en el desarrollo del sistema web se utiliza el método RUP donde cada fase puede descomponerse en etapas dando como éxito una entrega de un resultado ejecutable, por otro lado, se utiliza el método CRISP-DM (Cross Industry Standard Process for Data Mining) debido a que contempla el proceso de análisis de datos, asimismo el seguimiento de las fases no es estática y permite una actividad hacia adelante y hacia atrás entre distintas fases que permitirán desarrollar los modelos.

3.1.1 Inicio

En la tabla 7 se detalla la primera etapa de inicio, se realiza el análisis de la situación en la cual se revisa estadísticas y gráficos descriptivos que permitan comprender los datos que se cuentan disponibles para tener un mejor entendimiento del problema. Asimismo, se define criterios, objetivos, delimitaciones y recursos disponibles a utilizar para la evaluación de la solución actual dando como resultado el primer capítulo de esta investigación.

Tabla 7

Metodología Fase Planificación

Fase	Actividad	Resultado	Herramienta
Inicio	Análisis Inicial	Situación problemática	El análisis del proyecto: Lluvia de ideas, encuestas, Gráficos
	Identificación del Problemática	-	Diagrama de causa y efecto del problema
	Definición de Objetivos	-	-
	Recursos Disponibles	Cámaras, y Disco Duro computadora.	Google Drive

Elaborado por: los autores

3.1.2 Elaboración

Tanto la operación como el dominio del problema se estudian en profundidad dando como resultado la arquitectura básica.

3.1.3 Construcción

Dentro de esta etapa el objetivo principal es diseñar y desarrollar el proyecto, para lograr lo antes mencionado se dividió en dos subetapas:

a) Analizar y diseñar el modelo de red neuronal

De acuerdo a los objetivos planteados anteriormente, se diseñó dos modelos: el primer modelo es para identificar acciones violentes donde se utilizó distintas librerías para la detección de dos acciones en particular y el segundo modelo para identificar personas asociados a posibles actos de violencia se utiliza el reconocimiento facial de personas con requisitoria del programa de Recompensa del Perú, esto debido a que las personas con requisitoria son involucradas en una relación de violencia física

constantemente ya que cuentan con antecedentes policiales o algún tipo de requisitoria con posibles escenarios de acciones violentas por tal motivo la empresa corre con el riesgo de que se pueda ocasionar un escenario de violencia física. Esta etapa contempla las siguientes fases:

a.1) Análisis del problema

Acorde a los objetivos para identificar las acciones violentas se utilizó un modelo de detección utilizando las librerías del TensorFlow (keras) permitió realizar el modelo de la red neuronal y OpenCV que ayuda a procesar las imágenes de los videos. El funcionamiento de algoritmo de detección de violencia empieza recibiendo la información por la cámara de videovigilancia, realizando análisis de video para seguridad del restaurante, las escenas de video se utilizaron para el procesamiento de este tipo de procedimiento se convierte frames generados desde los videos. Después de confirmar los frames que es posible detectar violencia física (puñete y patada) con técnicas CNN y de evaluar cuál es la conexión entre la dificultad del modelo CNN utilizado y el costo computacional, lo siguiente es entrenar o desarrollar un modelo de detección de violencia que puedas canalizar imágenes en el corto tiempo como para ser desarrollado en secuencias de video. Donde se realizó una lista de videos públicos de violencia y sin violencia y cuando se compila el modelo, el siguiente proceso ya con el modelo entrenado con data del pasado está listo para realizar predicciones y generar una alerta de aviso de violencia física.

Asimismo, para el segundo modelo “identificar personas asociados a posibles actos de violencia” se utiliza el Numpy, el cual permite realizar arreglos y concatenar imágenes y las funciones de la librería de Face Recognition para la identificación de rostros.

El funcionamiento del algoritmo empieza desde capturar una imagen entrante desde un dispositivo de forma bidimensional en función de las características del dispositivo, el video se puede considerar como

una secuencia de imágenes estáticas, por lo tanto, las métricas y los enfoques para evaluar la precisión de las imágenes también son aplicables al video, para esto es importante saber que el procesamiento del flujo de video cálculos se impone restricciones adicionales en todas las etapas del reconocimiento facial. Cuando se usa el video se deben realizar pruebas de rendimiento y estos comparan con una base de datos la información relevante de la señal de imagen entrante en tiempo real de vídeo, más fiable y segura que la información obtenida en una imagen estática; para ello no se necesita de una conexión a internet, dado que la base de datos se puede encontrar en la computadora de la empresa, pero para la actualización de la base de datos si se necesitara por el uso de internet por los servidores externos.

en esta comparación de rostros, se analiza matemáticamente y se verifica que los datos biométricos se corresponden con la persona que debe hacer uso del servicio al sistema. Dependiendo de si la persona a quien están buscando está en la base de datos para mostrar el recuadro de persona con requisitoria.

El lenguaje de programación a utilizar es Python, el cual es el más utilizado y popular de desarrollar tuvo iteraciones rápidas de datos que facilitan en el desarrollo de algoritmos por lo que lo hace ser más sencillo y consistente, además de caracterizarse por ser una de las principales claves para conectar y facilitar la creación de códigos entendible de rápida enseñanza como los que son necesarios en actividades de Machine Learning. Asimismo, la técnica más adecuada para realizar este proyecto es el aprendizaje supervisado, ya que como se comentó en el capítulo II, está técnica permite desarrollar algoritmos que “aprenden” de datos introducidos para que generen datos de salida esperados y las librerías usadas para la creación y entrenamiento de los modelos. Adicionalmente se utiliza anaconda, el cual incluye un administrador de ambiente y de paquetes.

a.2) Análisis de datos

En este paso se realiza la canalización de información en donde se inicia la exploración de estos, logrando comprender el resultado e identificando las primeras asociaciones y patrones. los datos utilizados para el entrenamiento de la red neuronal son datos referentes, debido a la limitación indicada en el punto 1.7.2 del capítulo I, por lo que se optó por realizar una propia base de datos para la detección de personas con requisitoria.

a.3) Preparar datos

En esta fase permitió construir el conjunto final de datos para el entrenamiento de modelo, donde a partir de la recolección de los datos se segmento la información necesaria para los modelos.

a.4) Modelado

En esta última fase, a través de las librerías y Python se crea los modelos para lo cual primero se debe realizar la instalación del entorno Python e importar las librerías a utilizar, asimismo, se descarga el SW Anaconda el cual permite ejecutar el proyecto.

b) Analizar y diseñar el sistema web

Para esta segunda etapa se basa en las disciplinas primarias más importante que se encuentran dentro de RUP, las cuales son básicas para desarrollar un proyecto de software, entre todas se tuvon: Modelado del Negocio, Requerimientos, Análisis y Diseño y Pruebas.

b.1) Modelado del Negocio

Esta disciplina permite describir las tareas de la empresa y los clientes para especificar cada CU del Negocio, además se utiliza los Diagramas de Actividad y de Clases. Asimismo, se utilizó el modelamiento AS IS y TO BE

b.2) Requerimientos

La finalidad es definir lo que el sistema debe disponer, precisar los términos del sistema y una interfaz de usuario.

b.3) Análisis y diseño

Determinar la arquitectura del sistema y tuvo como finalidad desplazar los requerimientos en las características de implementación, al decir análisis se refiere a cambiar Caso de Uso en clases, y al decir diseño se refiere a refinar el análisis para poder en práctica los diagramas de clases de análisis de cada Caso de Uso, los diagramas de participación de cada Caso de Uso, el de clases de diseño de cada Caso de Uso, el suceso de diseño de Caso de Uso, el de estados de las clases, el modelo de implementación de la arquitectura.

3.1.4 Pruebas

En la tabla 8 se detalla dentro de esta etapa, el objetivo principal es realizar las pruebas correspondientes a la página web con los modelos previamente ya entrenados.

a) Integrar modelo al sistema de prevención

Para esta fase se integró los dos modelos ya previamente entrenados al sistema, a través de la función “@approute” la cual permite conectar Python con HTML.

b) Realizar la recopilación de pruebas

El objetivo de la recopilación de pruebas es descubrir imperfecciones y errores que puedan tener en el uso del sistema, comprobar que el sistema cumple con los requerimientos concretos por el usuario y si tuvo una capacidad adecuada en el ambiente donde se encuentra instalado. Otra parte importante para evaluar son las características de seguridad relacionadas con la entrada no autorizada de usuarios, de manera que no puedan realizar modificaciones donde no sean permitidas.

Tabla 8

Metodología Fase Aplicación de la Propuesta

Fase	Actividad	Resultado	Herramienta
Aplicación de lo propuesto	Integrar el modelo al sistema de videovigilancia	Sistema de videovigilancia	Visual Code y python.
	Realizar la recopilación de las pruebas	Escenarios erróneos o exitosos	Google Chrome

Elaborado por: los autores

3.1.5 Resultado

en la tabla 9 se detalla en la última etapa del proyecto, se analiza los resultados dados en las pruebas para la elaboración de los resultados, discusiones de acuerdo con los objetivos planteados, asimismo, se elabora las conclusiones y recomendaciones.

Tabla 9

Metodología Resultado

Fase	Actividad	Resultado
Resultado	Analizar los Resultados Obtenidos	-
	Elaboración de los Resultados y Discusiones	-
	Elaboración de Conclusiones y Recomendaciones	-
	Cierre de Investigación	-

Elaborado por: los autores

Asimismo, en la tabla 10 se detalla en esta fase se valida las métricas de calidad donde se tuvo dos clase para cada uno de los dos modelos donde fue: “Si se identificó correctamente una persona con requisitoria”, fue la clase positiva (+) y “No se identificó correctamente una persona con requisitoria” fue la clase negativa (-) en el primer modelo de “detección de personas con requisitoria” (ver tabla 11) y para el otro modelo fue “Si se identificó violencia física”, fue la clase positiva (+) y “No se identificó violencia física” fue la categoría negativa (-) (ver tabla 12). Entonces, VP es la cantidad de objetos de la categoría positiva que el algoritmo predijo correctamente, FP es la cantidad de objetos de la clase positiva que el algoritmo predijo incorrectamente, VN son la cantidad de objetos de la categoría negativa que el algoritmo predijo con una porcentaje alto, FN es la cantidad de objetos de la categoría negativa que la red neuronal convolucional predijo incorrectamente.

Tabla 10

Matriz de Contingencia

		Inferencia por la red	
		Si Detecta Persona con requisitoria (Clase +)	No Detecta Persona con requisitoria (Clase -)
Situación Real	Se identificó correctamente una persona con requisitoria	VP	FN
	No se identificó correctamente una persona con requisitoria	FP	VN

Elaborado por: los autores

Tabla 11

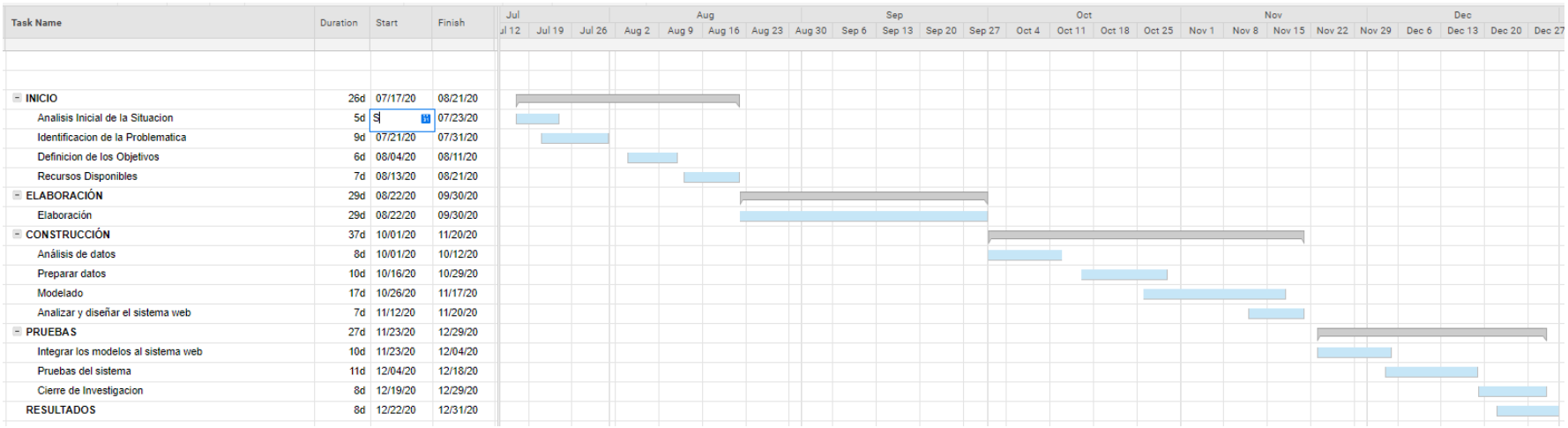
Matriz de Contingencia

	Inferencia por la red

		Si Detecta Violencia (Clase +)	No Detecta Violencia (Clase -)
Situación Real	Si se identificó violencia física	VP	FN
	No se identificó violencia física	FP	VN

Elaboración: Lumba, 2019.

3.2 Cronograma de proyecto



Task Name	Duration	Start	Finish
INICIO	26d	07/17/20	08/21/20
Analisis Inicial de la Situacion	5d	S	07/23/20
Identificacion de la Problematica	9d	07/21/20	07/31/20
Definicion de los Objetivos	6d	08/04/20	08/11/20
Recursos Disponibles	7d	08/13/20	08/21/20
ELABORACIÓN	29d	08/22/20	09/30/20
Elaboración	29d	08/22/20	09/30/20
CONSTRUCCIÓN	37d	10/01/20	11/20/20
Análisis de datos	8d	10/01/20	10/12/20
Preparar datos	10d	10/16/20	10/29/20
Modelado	17d	10/26/20	11/17/20
Analizar y diseñar el sistema web	7d	11/12/20	11/20/20
PRUEBAS	27d	11/23/20	12/29/20
Integrar los modelos al sistema web	10d	11/23/20	12/04/20
Pruebas del sistema	11d	12/04/20	12/18/20
Cierre de Investigacion	8d	12/19/20	12/29/20
RESULTADOS	8d	12/22/20	12/31/20

FIGURA 21 Cronograma del proyecto

Elaborado por: los autores

CAPÍTULO IV

DESARROLLO

4.1 Elaboración

4.1.1 Componentes de un Sistema de Video Vigilancia.

El sistema de video vigilancia actual cuenta con 2 cámaras de red utilizada es el modelo “DS-2CD2185FWD-I” que tuvo hasta 4 megapíxeles de alta resolución con una máxima resolución de1920 × 1080, la primera cámara se encuentra enfocando en la puerta principal (ver figura 22) y la segunda cámara enfocada a las mesas (ver figura 23) ubicadas en el primer piso del restaurante como se observa en la figura 24, el cual es un mapeo del local.

Asimismo, se cuenta con un grabador de vídeo digital (NVR) que se compone del hardware, que permite interactuar con las grabaciones de la televisión y los videos en formato digital.

FIGURA 22 Cámara N°1 del local

Elaborado por: los autores

FIGURA 23 Cámara N°2 del local

Elaborado por: los autores

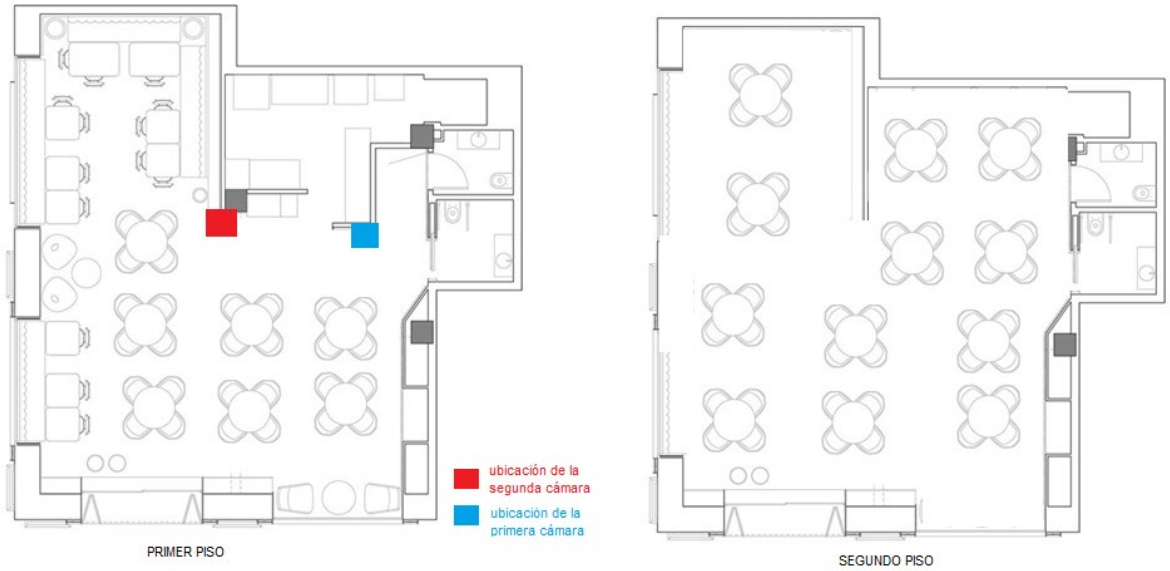


FIGURA 24 Mapeo del local

Elaborado por: los autores

4.1.2 Arquitectura de la empresa

Como se observa en la figura 25 la arquitectura actual de la empresa Real pez es independiente en cada sucursal, cada una de ella cuenta con dos cámaras IP, las cuales se conectan con un *switch* para posteriormente conectarse al router de internet. Asimismo, el *switch* permite la transmisión de la información en la PC y el NVR si es que los dispositivos están conectado en una misma red de área local.

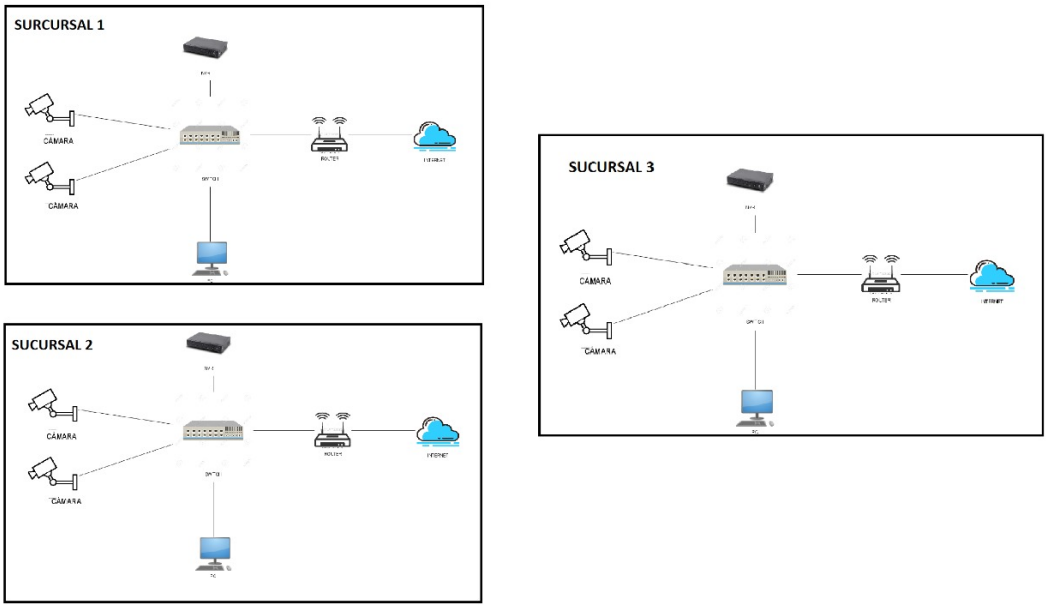


FIGURA 25 Arquitectura actual de la empresa

Elaborado por: los autores

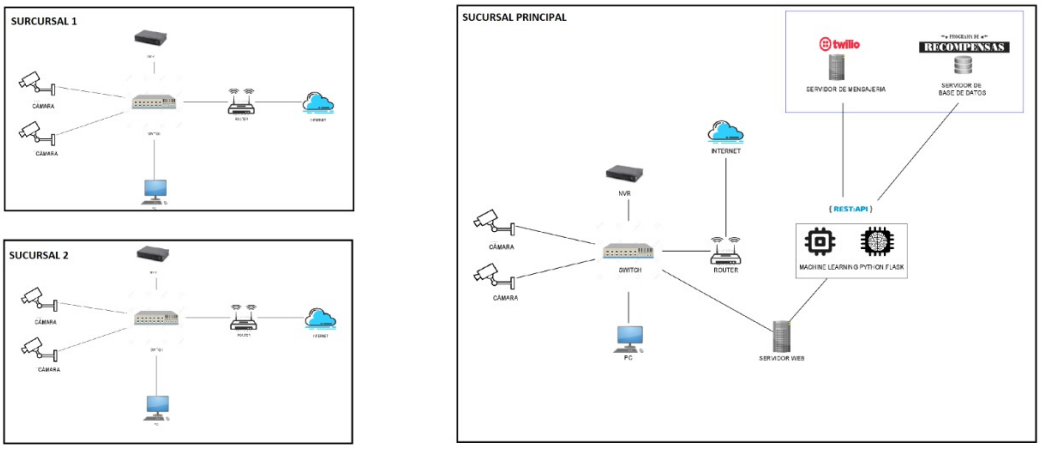


FIGURA 26 Arquitectura físico-propuesta del sistema para un local

Elaborado por: los autores

Como anteriormente se mencionó, la propuesta solución solo es para la primera sucursal, es por lo que en la figura 26 se observa que para la sucursal 1 y 2 su arquitectura sigue siendo la misma, sin embargo para la sucursal principal se basa cuando la información ingresa a través de dos cámaras IP, las cuales se conectan con un switch y permite la transmisión de la información en la pc y el NVR si es que los

dispositivos están conectado en una red de área local (LAN), para posteriormente conectarse al router de internet, el cual se conecta con el servidor web, en la que cuenta la página web que se conectó mediante un Rest Api que sirve como método de comunicación entre los modelos de Machine Learning con Python para luego conectarse con los servicios de base de datos del “Programa de Recompensas” para recolectar la información de personas con requisitoria y el servidor de mensajería de “Twilio” para el envío de la alerta (ver figura 27).

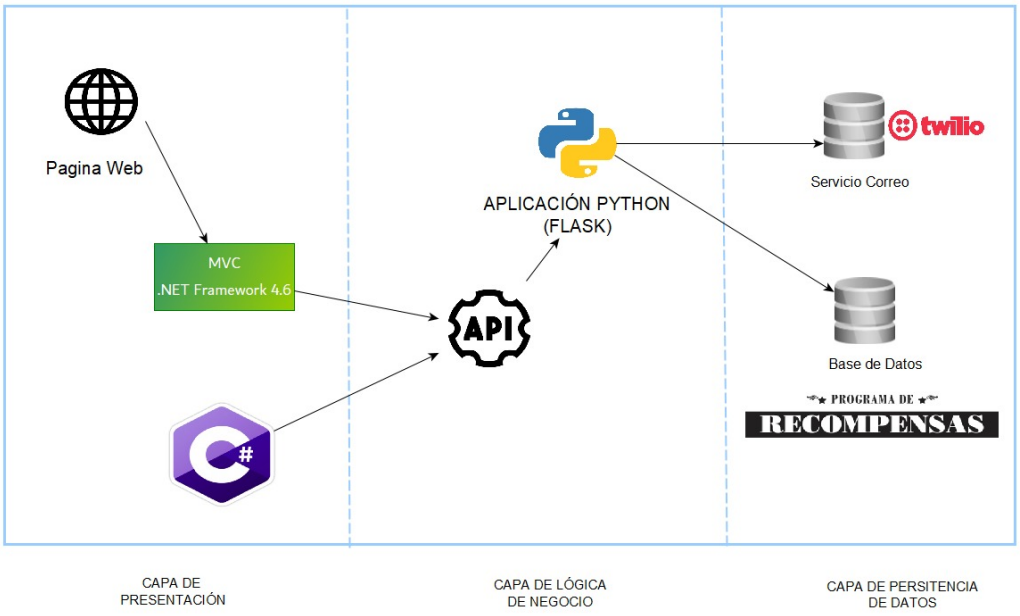


FIGURA 27 Arquitectura lógica propuesto del sistema para un local

Elaborado por: los autores

4.2 Construcción

4.2.1 Analizar y diseñar el modelo de red neuronal

a) Análisis de datos

a.1) Modelo para identificar acciones violentas

Para el análisis de datos, se realizó la recolección de datos sobre los videos de la página YouTube, donde se realizó una lista de videos públicos de violencia y sin violencia (ver figura 28) posteriormente, se utilizó un programa desarrollado por los tesistas (ver figura 29) el cual se ingresa las rutas de las páginas de los videos a descargar y automáticamente hace la descarga masiva descargándolos en una ruta especifica de la máquina (ver figura 30) dando como resultado el universo, el cual es de 600 videos con una duración muy variable (pueden ir desde segundos hasta horas), distintos escenarios, resolución, entre otros.

FIGURA 28 Lista de videos en YouTube

Elaborado por: los autores

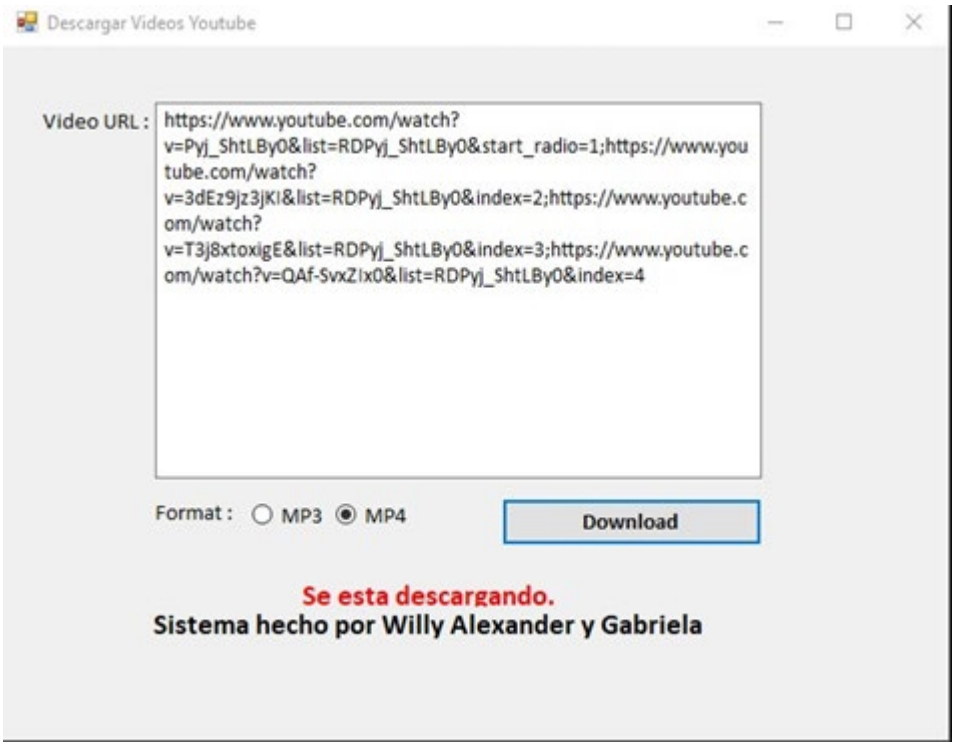


FIGURA 29 Programa para la descarga de videos

Elaborado por: los autores

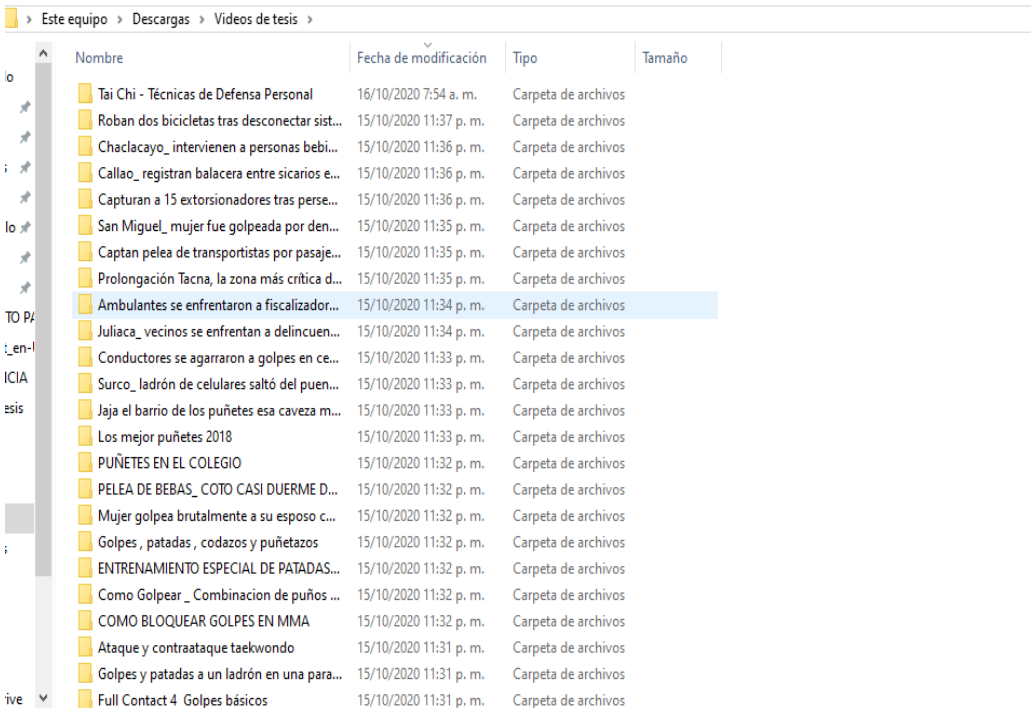


FIGURA 30 Videos recolectados

Elaborado por: los autores

a.2) Modelo para identificar personas asociados a posibles actos de violencia

Para la recolección de datos se utilizó un Servicio Web que permitieron obtener las imágenes de personas con requisitoria de la página web principal de recompensas dando como resultados las imágenes obtenidas (ver figura 31).

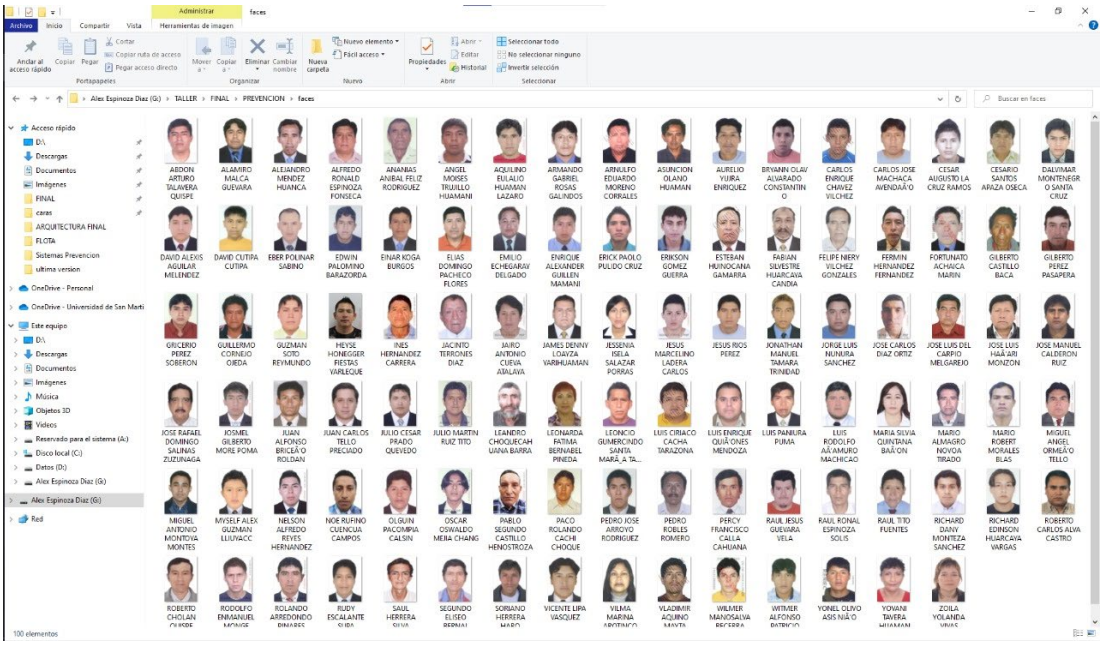


FIGURA 31 Base de datos de personas

Elaborado por: los autores

b) Preparar datos

b.1) Modelo para identificar acciones violentas

Una vez obtenido el universo de datos, se segmentaron los videos de violencia física que cumplan con las variables y atributos indicados en tabla 12, para los videos sin violencia la única condición es que no se observe alguna situación de agresión física, en total se obtuvieron 600

videos (300 videos de violencia y 300 videos sin violencia) los cuales fueron almacenado en la nube de Google Drive.

Se realizó de manera aleatoria las muestras en una distribución entre el conjunto de entrenamiento y pruebas, donde para la muestra de entrenamiento se utilizó un ratio del 80% de los videos, para la muestra de prueba fue el 20% y, por último, para la validación se realizó con videos en tiempo real.

Tabla 12

Atributos a evaluar en la segmentación

	Videos con Violencia
Acciones contenidas	1. Una mujer metiendo un puñete a otra mujer o hombre o niño. 2. Una mujer tirando una patada a otra mujer o mujer o niño. 3. Una hombre metiendo un puñete a un hombre o mujer o niño. 4. Una hombre tirando una patada a un hombre o mujer o niño. 5. un niño metiendo un puñete a otro niño.
Fotograma por segundo	16
Duración (seg.)	1.5 a 3
Resolución	640 × 360
Formas de grabación	Videos de cámaras de seguridad, videos de celulares,
Calidad de los videos	Escenarios con poca/bastante iluminación, color, y posiciones distintas.

Elaborado por: los autores

b.2) Modelo para identificar personas asociados a posibles actos de violencia

Como anteriormente se mencionó se utiliza las funciones de la librería Face_Recognition, el cual es un modelo ya pre-entrenado con un acoplamiento de conjuntos de datos de reconocimiento facial por imágenes de rostros con 128 dimensiones y debido a que el universo de datos obtenido cuenta con las caras correctamente, sin interrumpir o taparse con algún objeto que no pueda identificarse correctamente no hubo necesidad de preparar la información.

c) Modelado

c.1) Modelo para identificar acciones violentas

Para la creación de modelo primero se realizó la instalación del entorno, donde la versión mínima de Python para poder ejecutar la librería Tensor Flow se usó la versión 3.5 a 3.8 (ver figura 32) Asimismo, se descarga el SW Anaconda el cual permitió ejecutar el proyecto.

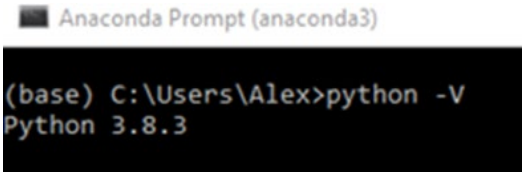


FIGURA 32 Versión de la verificación de Python

Elaborado por: los autores

Posteriormente, se instaló las librerías correspondientes dentro del entorno de desarrollo (ver figura 33) y se importó las librerías respectivas para la creación de la red neuronal. (ver figura 34)

FIGURA 33 Instalación de las librerías en el entorno de desarrollo

Elaborado por: los autores

```
#Imports
from glob import glob
from sklearn.model_selection import train_test_split
from os.path import exists
import subprocess
import cv2
from math import floor
import random
import numpy as np
from Params import *
```

FIGURA 34 Importación de las librerías en el entorno de desarrollo

Elaborado por: los autores

Por otro lado, se creó dos clases paraments.py, entrenamiento.py y ejecución.py las cuales permite entrenar el modelo y están establecidas de la siguiente manera:

Paraments.py: en esta clase se realiza la declaración de campos (ver figura 35) para el modelo, este se obtiene de campos de cualquier tipo y cada uno figura en una columna de datos que se quiere preservar. En este caso se creó capas dentro de la red, donde se establecen:

- Las rutas de las carpetas donde se encuentra la base de datos para la obtención de información;
- Las rutas de las carpetas donde se almacenan los frames;
- La altura y ancho de los videos a procesar;
- La cantidad y duración de videos a procesar;
- Transformación del frame a escala de grises;
- La sustracción de imágenes entre el frame seleccionado y la imagen patrón;

- La media aritmética de la imagen resultante de la sustracción de imágenes.

```
base_dir = "D:\\PROYECTO VIOLENCIA"

#Hyperparameters for frames generation
directory_videos_with_violence = ".\\CON VIOLENCIA"
directory_videos_without_violence = ".\\SIN VIOLENCIA"

max_videos = 15

#Hyperparameters for images generator
directory_videos_with_violence_for_training = base_dir + "\\CON VIOLENCIA\\training"
directory_videos_without_violence_for_training = base_dir + "\\SIN VIOLENCIA\\training"

directory_videos_with_violence_for_validation = base_dir + "\\CON VIOLENCIA\\validation"
directory_videos_without_violence_for_validation = base_dir + "\\SIN VIOLENCIA\\validation"

batch_size = 32

height = 32
width = 32

number_of_images_per_sample = 5

#Hyperparameters for Training
epochs = 10

checkpoint_path = base_dir + "\\model_trained"

number of epochs to stop = 5
```

FIGURA 35 Clase params.py

Elaborado por: los autores

Entrenamiento.py: En el caso de entrenamiento se utilizó la distribución del TensorFlow, para ello se realizó la configuración (ver figura 36)

```
#Tensorflow Configuration
import tensorflow as tf

from tensorflow.keras.models import Model

from tensorflow.keras.layers import Input, GaussianNoise, Conv2D, Flatten, Dense
from tensorflow.keras.layers import Input, Reshape, Permute, GaussianNoise, Conv3D, Flatten, Dense, LeakyReLU

from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping

from tensorflow.keras import backend as K

tf.random.set_seed(1)
```

FIGURA 36 Configuración del TensorFlow

Elaborado por: los autores

En este paso que se observa en la figura 37 se establecen las funciones para la validación del conjunto de frames del video que va a servir de entrada, para luego separarlo en un conjunto de imágenes y así reducimos los canales a un solo canal para el siguiente análisis.

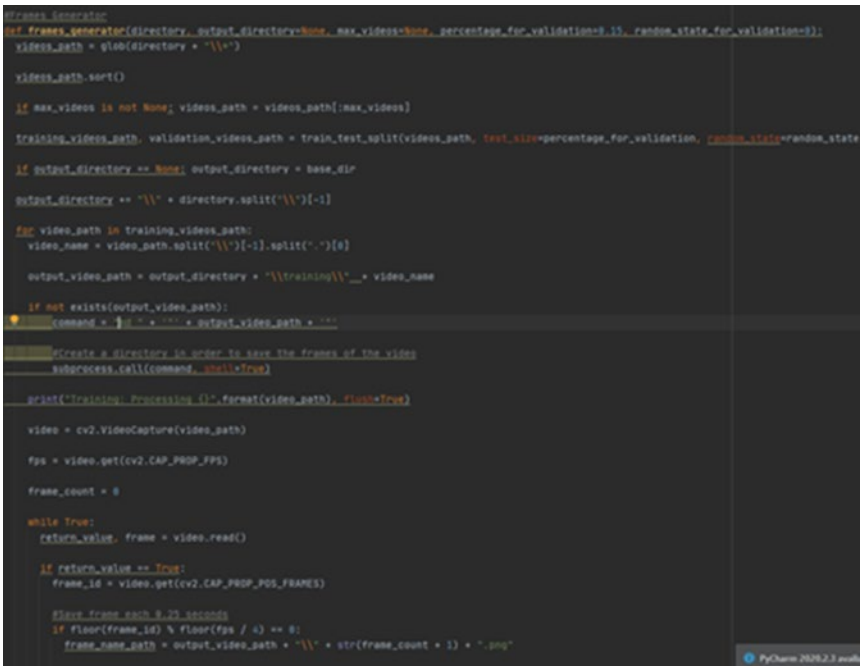


FIGURA 37 Frames generados

Elaborado por: los autores

Lo siguiente permite convertir frames generados por videos a formato “png” (ver figura 38), el almacenamiento de las imágenes se realiza asignándole un nombre formato por número ascendente de menor a mayor, en formato “png”. El conjunto de frames obtenidos se agrupan por bloques en una sola imagen para la detección de violencia.

FIGURA 38 Frames generados por videos convertido a Formato png

Elaborado por: los autores

En la figura 39 se muestra cómo se realiza el flujo para el reconocimiento automática de hechos humanas en secuencias de video para el apoyo de videovigilancia.

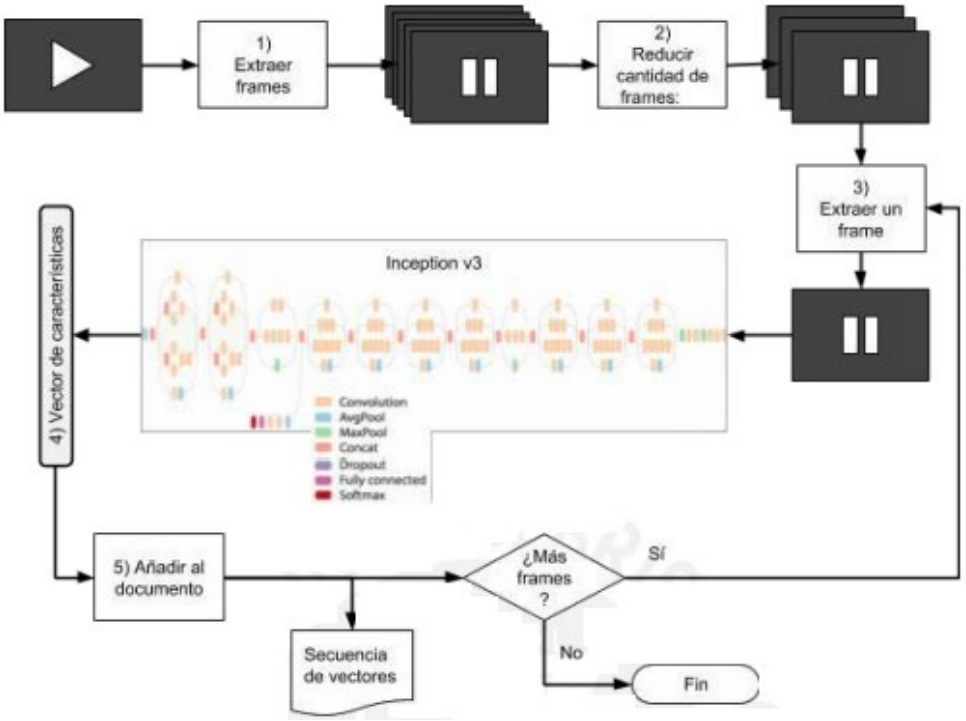


FIGURA 39 Extracción de frames, reducción a 30 frames y extracción de características

Fuente: Fernández Luis, 2017

Por otro lado, se estableció los bloques de construcción de una red neuronal que son las capas con un número de 4 campos, donde cada uno significa una columna de datos que se quiere obtener para que sean aprendidos durante el entrenamiento (ver figura 40). Finalmente, se establece la función de compilación del modelo (ver figura 41)

.

```
#Neural Network Model
K.clear_session()

input_layer = Input(shape=(height, width, 3 * number_of_images_per_sample))

gaussian_noise_layer = GaussianNoise(0.1)(input_layer)

convolution_layer = Conv2D(5, (3, 3), activation="relu", padding="same")(gaussian_noise_layer)

convolution_layer = Conv2D(3, (3, 3), activation="relu", padding="same")(convolution_layer)

convolution_layer = Conv2D(1, (3, 3), activation="relu", padding="same")(convolution_layer)

flatten_layer = Flatten()(convolution_layer)

dense_layer = Dense(150)(flatten_layer)

output_layer = Dense(2, activation="softmax")(dense_layer)

model = Model(input_layer, output_layer)

model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])
```

FIGURA 40 Creación de las capas de la red

Elaborado por: los autores

```
model = Model(input_layer, output_layer)

model.compile(optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"])
```

FIGURA 41 Compilación del modelo

Elaborado por: los autores

Ejecución.py: Se utiliza la función "*prediction* = model.predict(images)[0]" que permite realizar predicciones sobre imágenes. Asimismo, si la predicción es igual a "1" de etiqueta se tuvo "violencia" sino es "sin violencia" (ver figura 43)

```
#Process Video
def get_prediction(model, images):
  prediction = model.predict(images)[0]

  prediction = K.round(prediction).numpy()[0]#tensor el resultado round y numpy convertiendolo en vector

  prediction = int(prediction)

  return prediction

def process_video(video_path, model, height=96, width=96, number_of_images_per_sample=5, output_directory=None):
  video = cv2.VideoCapture(video_path)

  video_height = int(video.get(cv2.CAP_PROP_FRAME_HEIGHT))
  video_width = int(video.get(cv2.CAP_PROP_FRAME_WIDTH))

  fps = video.get(cv2.CAP_PROP_FPS)

  frames_queue = deque()

  color = (255, 255, 255) #BGR

  label = ""

  if output_directory == None: output_directory = base_dir

  video_with_label_name = output_directory + "\\" + video_path.rsplit("\\", 1)[-1].split(".")[0] + "_with_label" #aqui se guarda los videos dele resultado
  fourcc = VideoWriter_fourcc(*"MP42")
  video_with_label = VideoWriter(video_with_label_name + ".avi", fourcc, float(fps), (video_width, video_height))

  print("Processing: {}".format(video_path))
```

FIGURA 42 Clase ejecución

Elaborado por: los autores

```
prediction = get_prediction(model, images)

if prediction == 1:
  label = "CON VIOLENCIA"

  color = (0, 0, 255) #BGR
else:
  label = "SIN VIOLENCIA"

  color = (0, 255, 0) #BGR

frames_queue.popleft()
```

FIGURA 43 Predicción en el modelo para violencia y sin violencia

Elaborado por: los autores

Para el proceso de entrenamiento del modelo duró 8 días como figura en la figura 44, lo cual es una mejora en el tiempo de respuesta comparado desde el modelo inicial que fue de 8 días, 5 horas y 15 minutos, lo cual permitió tener resultados satisfactorios, y de esta

manera mejorar las herramientas del entrenamiento e ir mejorando el resultado al cual se debe llegar, después de 300 épocas se realizó una comparación de los resultados y se escogió las 150 épocas del conjunto total del entrenamiento.

Dando como resultado el entrenamiento de cada capa de una red neuronal, asimismo se concluyó con un análisis de caso. Asimismo, los intervalos de confianza y esto concede analizar la parte más primordial en la que, según el modelo generado determine el valor promedio de la variable que es el resultado final.

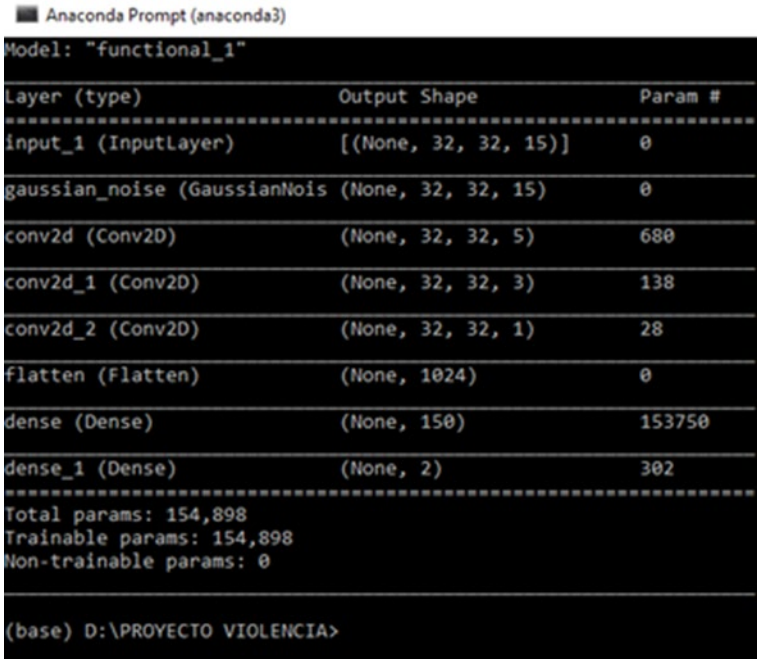


FIGURA 44 Resultado del Entrenamiento de cada capa de una red neuronal

Elaborado por: los autores

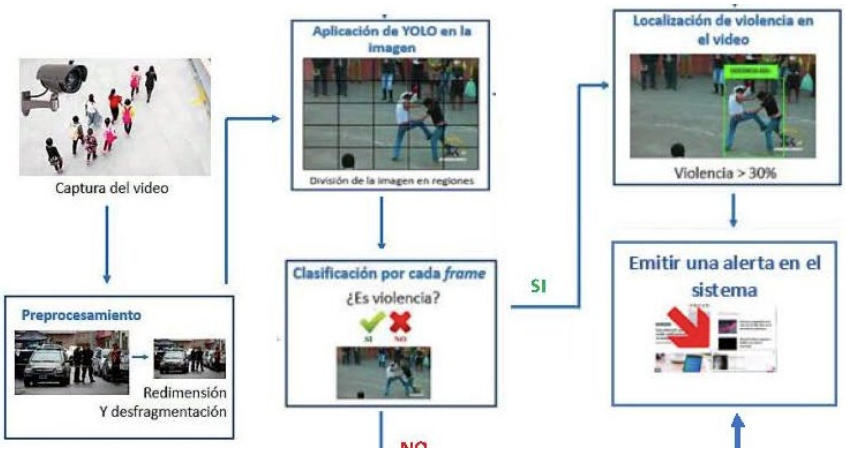


FIGURA 45 Funcionamiento del modelo

Elaborado por: los autores

En la figura 45 se observó el flujo del funcionamiento del modelo de violencia física.

c.2) Modelo para identificar personas asociados a posibles actos de violencia

Luego de contar con el entorno de desarrollo (ver figura 33) se instaló las bibliotecas PIP (ver figura 46) una vez completada la instalación, se importa las librerías a utilizar. (ver figura 47)

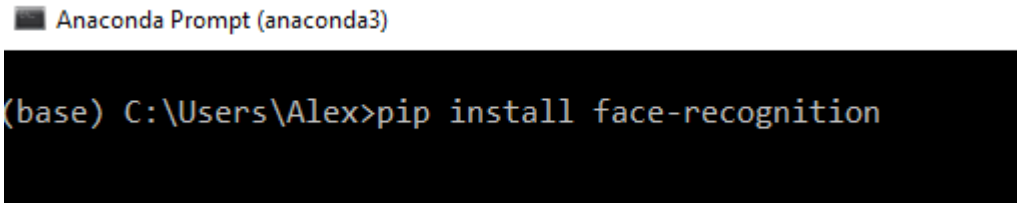


FIGURA 46 Instalación de la librería face-recognition

Elaborado por: los autores

```
import cv2
import numpy as np
import json
import os
```

FIGURA 47 importaciones de librerías de python

Elaborado por: los autores

Posteriormente, se crea una clase face_recognition.py en el cual se realiza la declaración de variables como face_locations, face_encodings y face_names (ver figura 48) asimismo, se declaró los campos donde se establecen:

- Todas las imágenes están en una carpeta llamada " faces".
- Los nombres de archivo se enumeran y asignan a la variable "nombres".
- Los tipos de archivo deben ser iguales con el formato "jpg" o “png”.

```
faces = get_encoded_faces()

# Initialize some variables
face_locations = []
face_encodings = []
face_names = []
process_this_frame = True
```

FIGURA 48 algoritmo de la clase face_recognition

Elaborado por: los autores

Para obtener los datos se creó un API Rest es una forma común de comunicación en .Net con la clase HttpWebRequest de WebClient() y pueda consumir con el url del “Programa de Recompensas” que información libre para cualquier persona (ver figura 49).

```
public class PersonasController : Controller
{
    // GET: Personas
    public ActionResult Index()
    {
        string result = "";
        for (int i = 1; i < 3; i++)
        {
            string url = "https://recompensas.pe/buscarlistadoreq/" + i.ToString();
            var json = new WebClient().DownloadString(url);
            var lista = "lista" + i.ToString();
            result += json.Replace("lista", lista);
        }
        ViewBag.Response = result;
        return View();
    }
}
```

FIGURA 49 Creación de API

Elaborado por: los autores

Luego escribimos la ubicación donde se quiere guardar las imágenes ya que las imágenes son tipo “data:image/jpg;base64” para su descarga (ver figura 50).

```
<script>
    $(document).ready(function () {
        //Ubicación
        var Ubicacion = "G:\TALLER\FINAL\PREVENCION\faces";
        for (var i = 1; i < 3; i++) {
            $("#lista" + i).children().each(function (i, j) {
                //img
                var position = this;
                position = $(position).children()[0];
                var img = $(position).children().children().children()[0];
                img = $(img).attr('src')

                //img nombre
                var position_inicial_text = this;
                position_text = $(position_inicial_text).children()[1];
                var text = $(position_text).children().children()[0];
                var nomb = $(text).text();

                //forceDownload(img, nomb)

                var xhr = new XMLHttpRequest();
                xhr.open("GET", img, true);
                xhr.responseType = "blob";
                xhr.onload = function () {
                    var urlCreator = window.URL || window.webkitURL;
                    var imageUrl = urlCreator.createObjectURL(this.response);
                    var tag = document.createElement('a');
                    tag.href = imageUrl;
                    tag.download = nomb + '.jpg';
                    document.body.appendChild(tag);
                    tag.click();
                    document.body.removeChild(tag);
                }
                xhr.send();
```

FIGURA 50 Código de la API

Elaborado por: los autores

Lo siguiente para detectar rostros en la secuencia en video y conocer la ubicación o coordenadas exactas de la cara para su posterior procesamiento, se utilizó las dos funciones mencionadas antes como se observa en la figura 50.

```
face_locations = fr.face_locations(rgb_small_frame)
face_encodings = fr.face_encodings(rgb_small_frame, face_locations)
```

FIGURA 51 algoritmo de la clase face_recognition

Elaborado por: los autores

Posteriormente, se extrae características de la cara tomando una imagen de la persona como entrada y genera un vector que representa las características más importantes de un rostro (ver figura 52). Este vector se denomina incrustación facial guardando está información en un archivo, el siguiente paso es reconocer una nueva imagen que no está en la base de datos, calculando la incrustación de caras para la imagen (ver figura 53 y 54).

```
for face_encoding in face_encodings:
    # See if the face is a match for the known face(s)
    matches = fr.compare_faces(known_face_encodings, face_encoding)
```

FIGURA 52 algoritmo de la clase face_recognition

Elaborado por: los autores

```
def get_encoded_faces():
    encoded = {}

    for dirpath, dnames, fnames in os.walk("./faces"):
        for f in fnames:
            if f.endswith(".jpg") or f.endswith(".png"):
                face = fr.load_image_file("faces/" + f)
                encoding = fr.face_encodings(face)[0]
                encoded[f.split(".")[0]] = encoding
    return encoded
```

FIGURA 53 algoritmo de la clase face_recognition

Elaborado por: los autores

```python
def get_camara(video_capture):
    detecction_prisoner = dict()
    process_this_frame = True
    while True:
        # Grab a single frame of video
        ret, frame = video_capture.read()

        known_face_encodings = list(faces.values())
        known_face_names = list(faces.keys())

        # Resize frame of video to 1/4 size for faster face recognition processing
        small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

        # Convert the image from BGR color (which OpenCV uses) to RGB color (which face_recognition uses)
        rgb_small_frame = small_frame[:, :, ::-1]

        # Only process every other frame of video to save time
        if process_this_frame:
            # Find all the faces and face encodings in the current frame of video
            face_locations = fr.face_locations(rgb_small_frame)
            face_encodings = fr.face_encodings(rgb_small_frame, face_locations)

            face_names = []
            for face_encoding in face_encodings:
                # See if the face is a match for the known face(s)
                matches = fr.compare_faces(known_face_encodings, face_encoding)
                name = ""
                names_ = ""
                # # If a match was found in known_face_encodings, just use the first one.
                # if True in matches:
                #     first_match_index = matches.index(True)
                #     name = known_face_names[first_match_index]

                # Or instead, use the known face with the smallest distance to the new face
                face_distances = fr.face_distance(known_face_encodings, face_encoding)
                best_match_index = np.argmin(face_distances)
                if matches[best_match_index]:
                    name = known_face_names[best_match_index]
                    names_ = "Con Requisitoria"
```

FIGURA 54 algoritmo de la clase face_recognition

Elaborado por: los autores

Como se mencionó anteriormente este es un modelo ya pre entrenado por lo que no requiere de un entrenamiento .

d) Analizar y diseñar el sistema web

d.1) Modelado del Negocio

En este punto se realizó las especificaciones del caso de uso del negocio:

- Caso de uso del negocio: Enviar alerta de detección de personas con requisitoria (ver tabla 13).

Tabla 13 Primer caso del uso del negocio

ID, Caso de Uso: CU001	**Nombre del CU:** Enviar alerta de detección de personas con requisitoria
Descripción: El usuario puede visualizar alertas cada que el sistema detecte personas con requisitoria.	
Actores: Cajera (Corresponde al actor principal)	
Precondiciones: para que el sistema envíe una alerta previamente mediante el modelo se debe identificar una persona con requisitoria	
Flujo de eventos: 1. El usuario ingresa al sistema por Chrome y Windows 10 2. El sistema detecta a una persona con requisitoria 3. El sistema automáticamente envía un mensaje de WhatsApp con ubicación del lugar.	

Elaborado por: los autores

- Caso de uso del negocio: Visualizar evidencia de acciones violentas detectadas (ver tabla 14).

Tabla 14 Segundo caso de uso del negocio

ID, Caso de Uso: CU002	**Nombre del CU:** Visualizar evidencia de acciones violentas detectadas
Descripción: El usuario puede visualizar una interfaz del historial de acciones violentas detectadas	

Actores: Cajera (Corresponde al actor principal)

Precondiciones: para que el sistema muestre las evidencia previamente debió existir acciones violentas detectadas.

Flujo de eventos:

1. El sistema muestra las dos cámaras existentes dentro del restaurante con sus respectivas fechas por cada cámara.
2. El sistema muestra el historial de acciones violentas detectadas de esa cámara por la fecha seleccionada.

Elaborado por: los autores

- Caso de uso del negocio: Visualizar un mapa de las automóviles de la empresa de seguridad y las comisarias más cercanas (ver tabla 15).

Tabla 15 Tercer caso de uso del negocio

ID, Caso de Uso: CU003	**Nombre del CU:** Visualizar un mapa de las automóviles de la empresa de seguridad y las comisarias más cercanas

Descripción: El usuario puede visualizar una interfaz que determine el tiempo, la distancia y la ubicación de las automóviles de la empresa de seguridad y las comisarias más cercanas.

Actores: Cajera (Corresponde al actor principal)

Precondiciones: para que el sistema muestre la información de los automóviles, comisarias previamente debe obtener su latitud y longitud.

Flujo de eventos:

1. El sistema muestra una interfaz ubicación de las automóviles de la empresa de seguridad y las comisarias.
2. El usuario puede visualizar una alerta que determine el tiempo, la distancia y el inicio y fin del recorrido.

Elaborado por: los autores

- Caso de uso del negocio: Enviar alerta de detección de violencia física (ver tabla 16).

Tabla 16 Cuarto caso de uso del negocio

ID. Caso de Uso: C004	**Nombre del CU:** Enviar alerta de detección una acción de puñete y/o patada

Descripción: El sistema envía automáticamente una alerta cuando detecte violencia física de las personas ingrese al restaurante.

Actores: Cajera (Corresponde al actor principal)

Precondiciones: para que el sistema envíe una alerta previamente mediante el modelo se debe identificar una acción de violencia física.

Flujo de eventos:

1. El sistema detecta a una acción de violencia física
2. El sistema automáticamente envía un mensaje de WhatsApp con ubicación del lugar.

Elaborado por: los autores

- Modelamiento AS IS: Se diagrama la forma actual de como los Stakeholders realizan sus operaciones. (Ver figura 55)

- Modelamiento TO BE: Se diagrama la forma futura de como los Stakeholders hizo sus operaciones. (Ver figura 56)

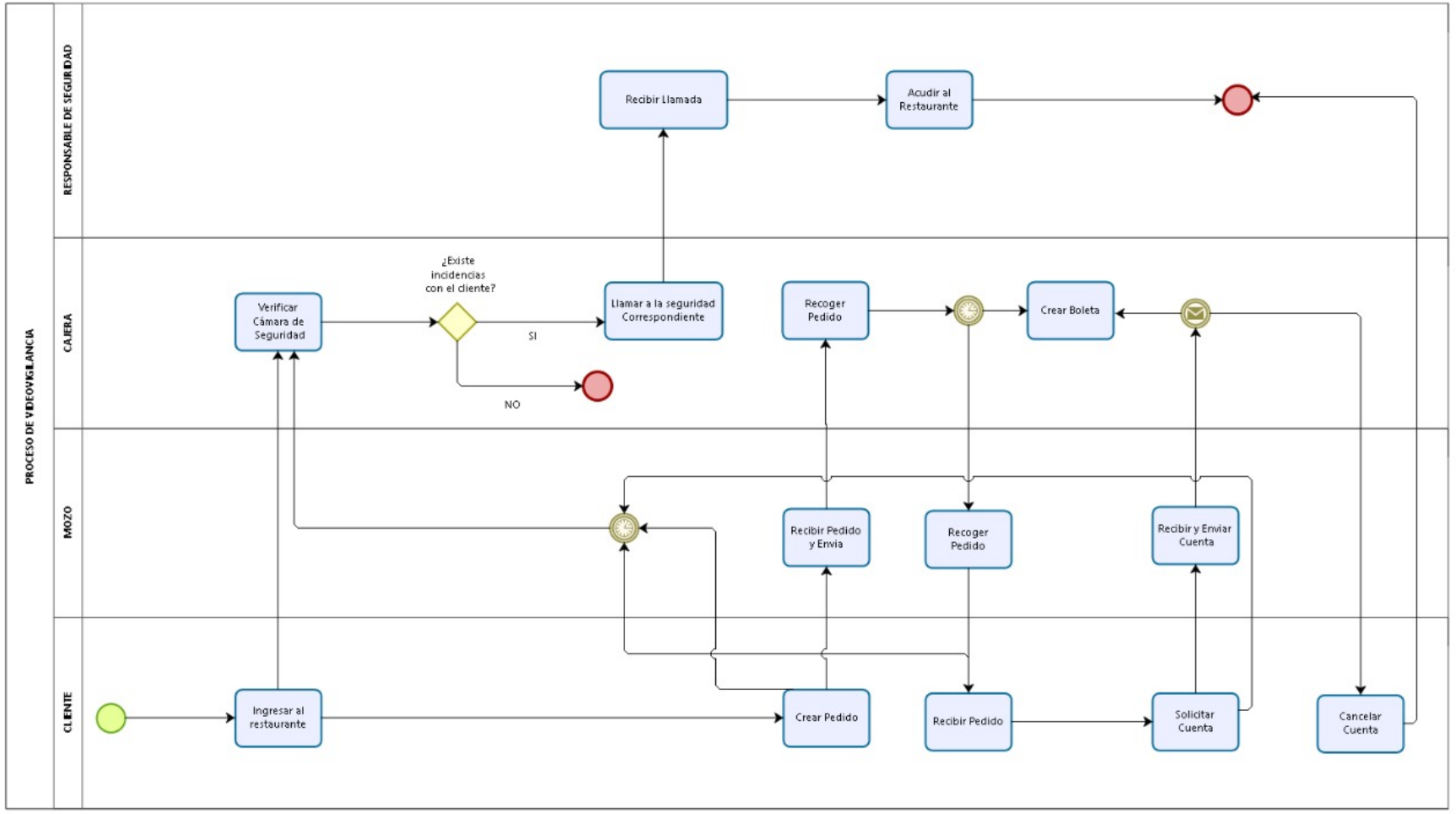


FIGURA 55 Diagrama AS-IS

Elaboración : El autor

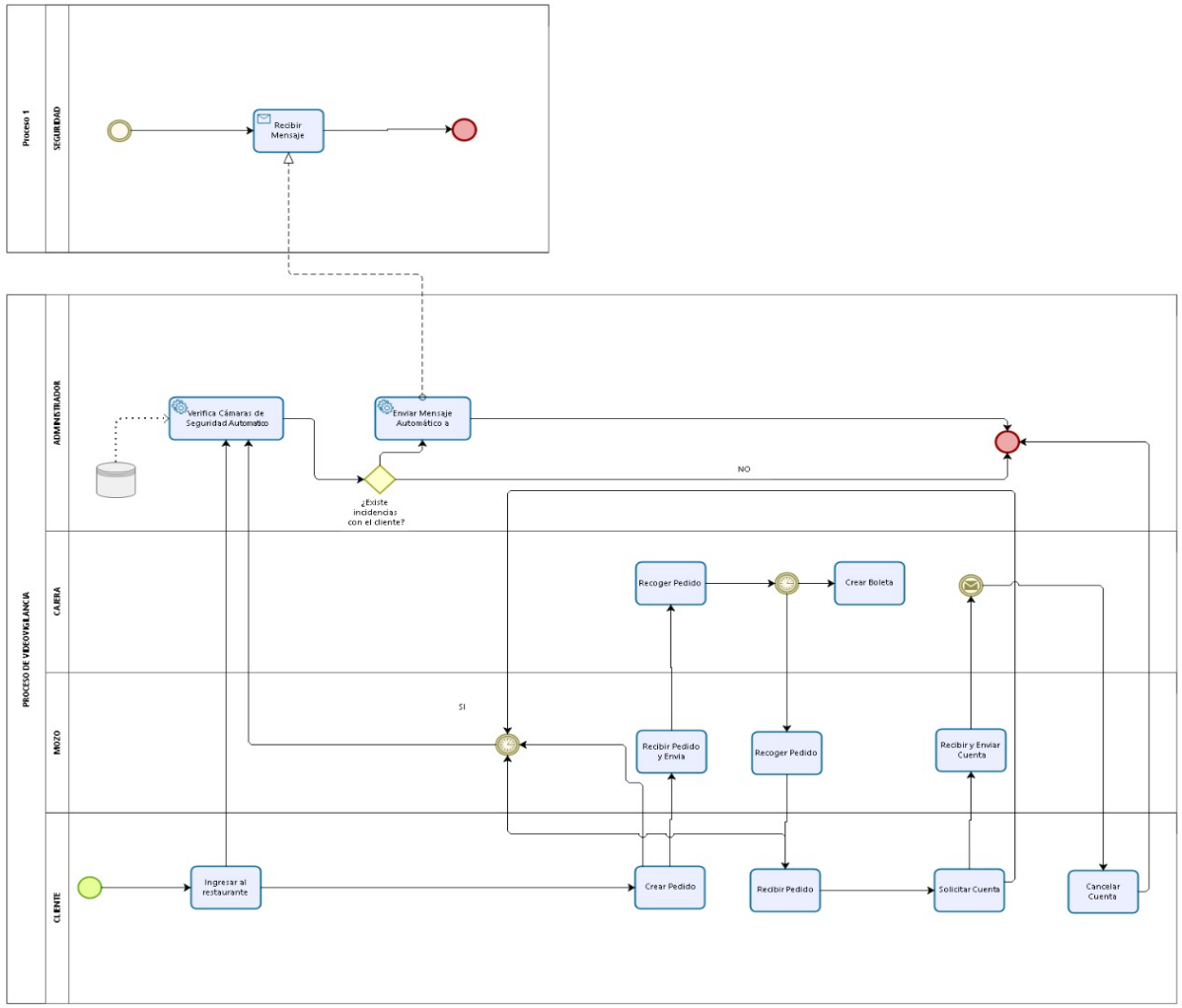


FIGURA 56 Diagrama To Be

Elaborado por: los autores

d.2) Requerimientos:

Tabla 17 Requerimientos funcionales y no funcionales

Requerimiento Funcionales	Requerimiento No Funcionales
RF1.- El usuario puede visualizar alertas cada que el sistema detecte personas con requisitoria.	RNF1.- Disponibilidad del sistema a través del explorador Google Chrome
RF2.- El usuario puede visualizar una interfaz del historial de acciones violentas detectadas.	RNF2.- El sistema debe contar con la protección contra robots dentro de las páginas webs.
RF3.- El usuario puede visualizar una interfaz que determine el tiempo, la distancia y la ubicación de las automóviles de la empresa de seguridad y las comisarias más cercanas.	RNF3.- El sistema debe correr en el sistema operativo Windows 10
RF4.- El usuario puede visualizar alertas cada que el sistema detecte una acción de puñete y/o patada.	RNF4.- El sistema puede permitir una cierta cantidad usuario simultáneamente.

Elaborado por: los autores

d.3) Análisis y diseño

- Diagrama de Contexto del Sistema

El siguiente diagrama permite visualizar los elementos externos como internos con los cuales el sistema va a interactuar, panorama básico

de todo el sistema; el siguiente diagrama va a facilitar al usuario externo y/o personas no técnicas a entender el sistema.

El administrador de la empresa realizar un monitoreo de la cámaras y un seguimiento sobre la comunicación de las alertas usando el Sistema de prevención y detección de violencia física el cual mediante el uso de las cámaras de video vigilancia identifica acciones violentas y personas asociados a posibles actos de violencia alimentado por el sistema web de recompensas. El Sistema de prevención y detección de violencia física para realizar la comunicación de alertas utiliza el sistema externo de Twilio que envía un mensaje de WhatsApp a las autoridades. El código de color en el diagrama indica los sistemas de software que ya existen (las cajas grises) y los que se van a construir (azules) (Ver Figura 57)

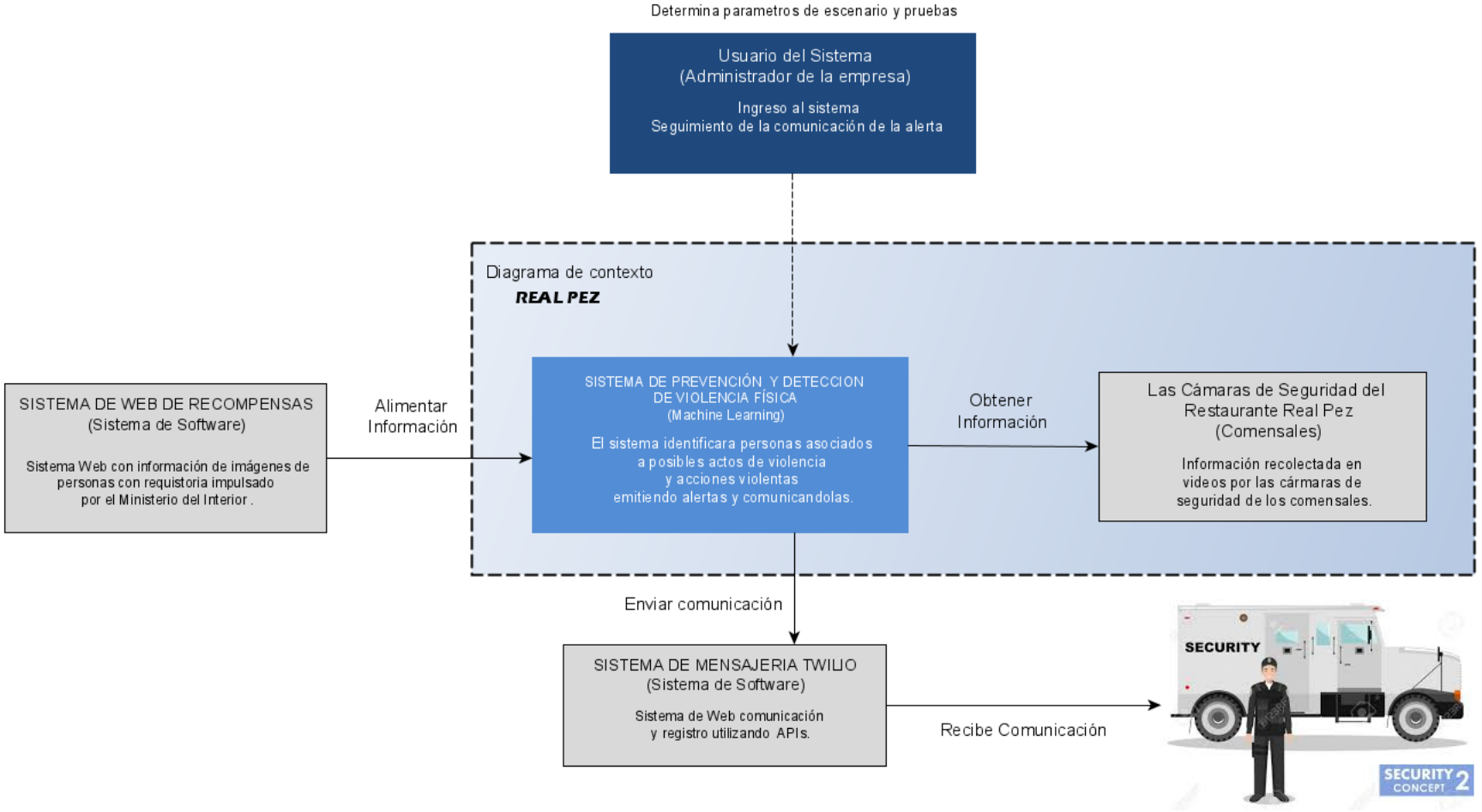


FIGURA 57 Diagrama de Contexto del Sistema propuesta del sistema

Elaborado por: los autores

- Diagrama de Contenedores

El contexto sistema tuvo como objetivo, visualizar el sistema desde un alto nivel, viendo lo que se desarrolló en el centro rodeado de todos los sistemas y usuarios que interactúan con él; por otra parte, también se observa la interacción con el agente externo una vez analizada la data. Este modelo de diagrama está orientado más para el entendimiento del personal técnico para el análisis de la solución de software.

El Sistema de prevención y detección de violencia física es un sistema C# que simplemente muestra contenido estático (HTML, CSS y JavaScript), utilizando una API HTTTPS.

El Sistema Web externo de Recompensa alimenta la lógica de negocio, el cual permite identificar los parámetros para detectar acciones violentas y personas asociados a posibles actos de violencia.

La aplicación en la API permite obtener información de la lógica de negocio teniendo en cuenta la interacción entre los modelos y la conexión de las cámaras de seguridad que obtuvo los datos de los comensales en videos en tiempo real, bajo el lenguaje Python. la aplicación de la API también utiliza el Sistema Externo de comunicación Twilio existente si en caso el sistema emita alguna alerta. (Ver Figura 58).

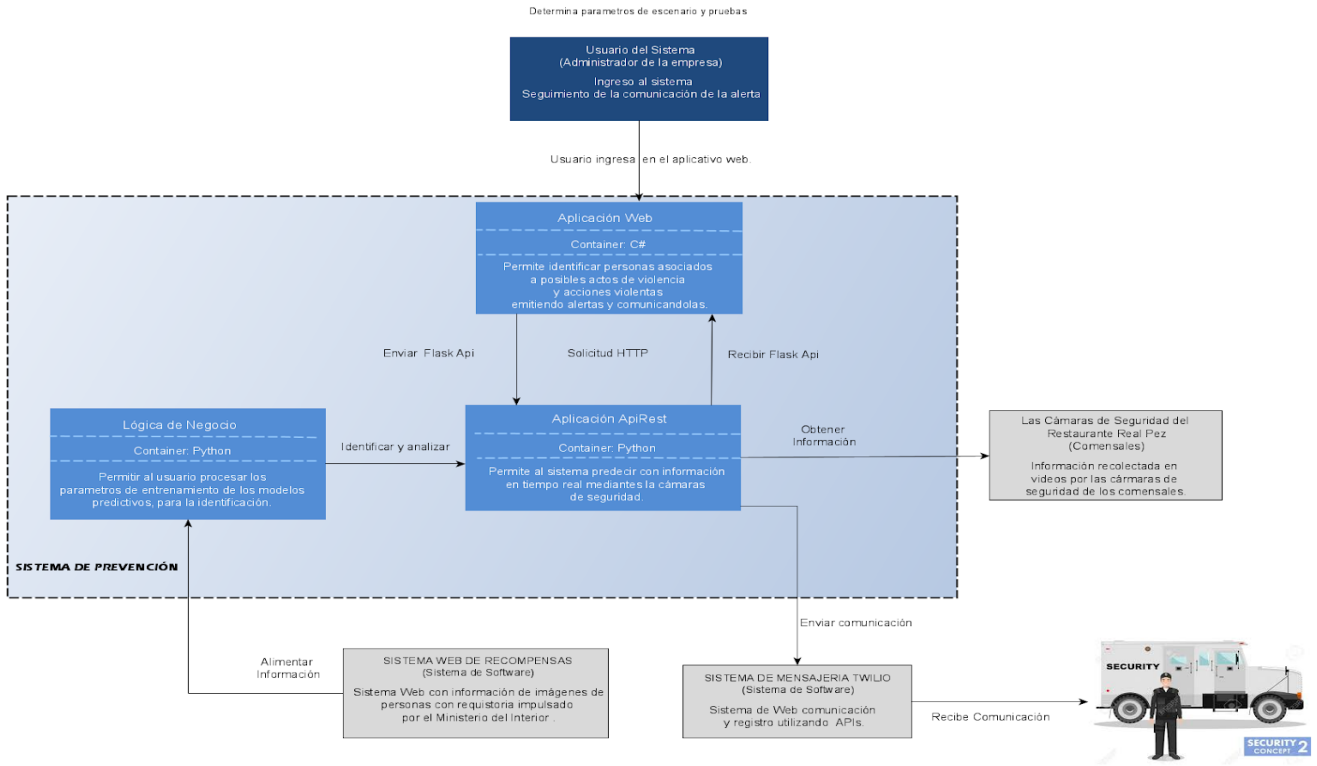


FIGURA 58 Diagrama de Contenedores del Sistema propuesta del sistema

Elaborado por: los autores

- Diagrama de componentes

Este diagrama permite observar de forma ilustre los bloques modulares a utilizar en varios puntos de una arquitectura, que realiza la interacción entre sistema backend y usuarios externos, y va a permitir tener una imagen más clara de las responsabilidades de los componentes de cada contenedor y sus dependencias entre ellos. (Ver Figura 59)

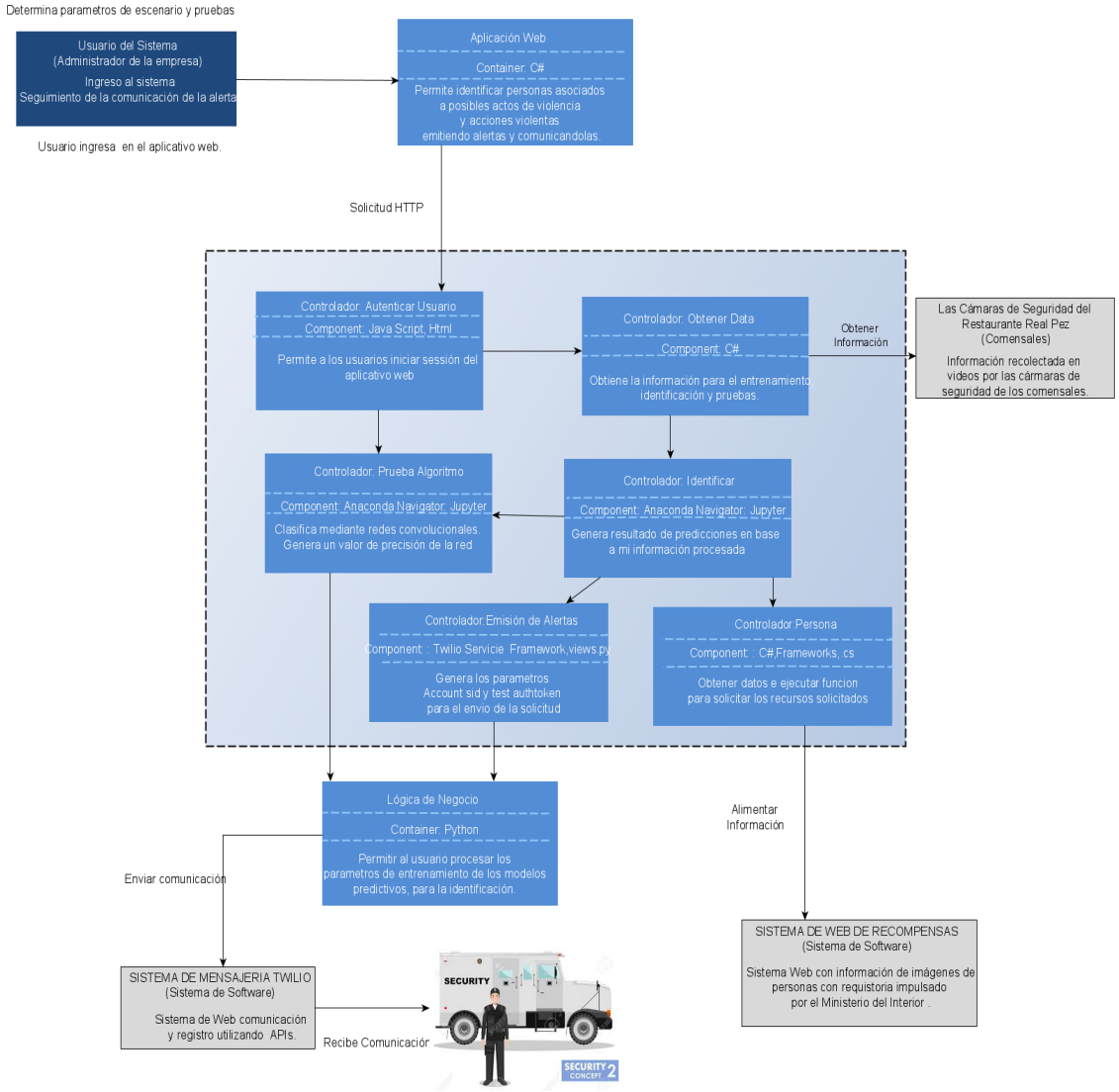


FIGURA 59 Diagrama de componentes del Sistema propuesta del sistema

Elaborado por: los autores

- Desarrollo

El sistema es un local host que tuvo tres interfaces que son desarrolladas con las clases de .CSS, C# y JavaScript.

Para la comunicación de las alertas se utilizó la conexión de Twilio con el sistema web, eso se basó en cuatro pasos:

Paso 1: Ingreso al Sistema Twilio para crear una cuenta

Paso 2: Obtener las credenciales correspondientes para el desarrollo siguiente (ver figura 60)

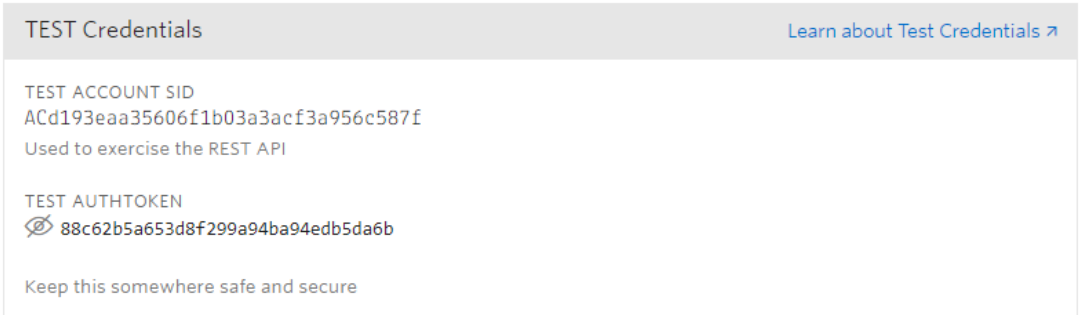


FIGURA 60 Credenciales de Twilio

Elaborado por: los autores

Paso 3: la instalación de la librería de Twilio con la consola de anaconda (ver figura 61).

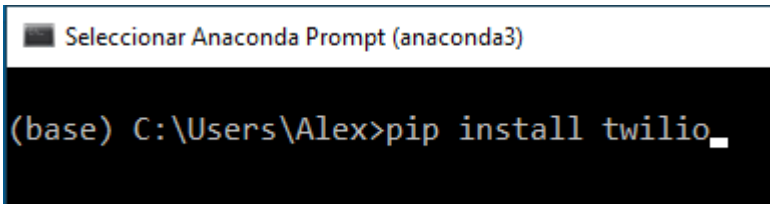


FIGURA 61 Instalación de Twilio

Elaborado por: los autores

Paso 4: la implementación del Servicio de Twilio en Python con las credenciales correspondientes (ver figura 62).

```
import os
from twilio.rest import Client


def EnviarMensaje(mensaje,url):
    account_sid = 'ACc0d8d810d64deacc3649cefc59a1efdf'
    auth_token = '88c62b5a653d8f299a94ba94edb5da6b'
    client = Client(account_sid, auth_token)
    message = client.messages.create(media_url=[url],
                                     from_='whatsapp:+14155238886',
                                     body='La persona indentificada como '+mensaje +' es requisitoriada',
                                     to='whatsapp:+51930317648'
                                     )
    message = client.messages.create(
                                     from_='whatsapp:+14155238886',
                                     body='La ubicación es',
                                     to='whatsapp:+51930317648',
                                     persistent_action='geo:-12.04937275773087,-77.00141457343213'
                                     )
```

FIGURA 62 Implementación de Twilio

Elaborado por: los autores

Como resultado se obtuvo las siguientes interfaces:

Primera Interfaz de inicio de sesión: la primera interfaz es un login, en cual al ingresar al sistema con el siguiente URL "https://Localhost:44334/" solicita identificarnos a través de las credenciales (ver figura 63).

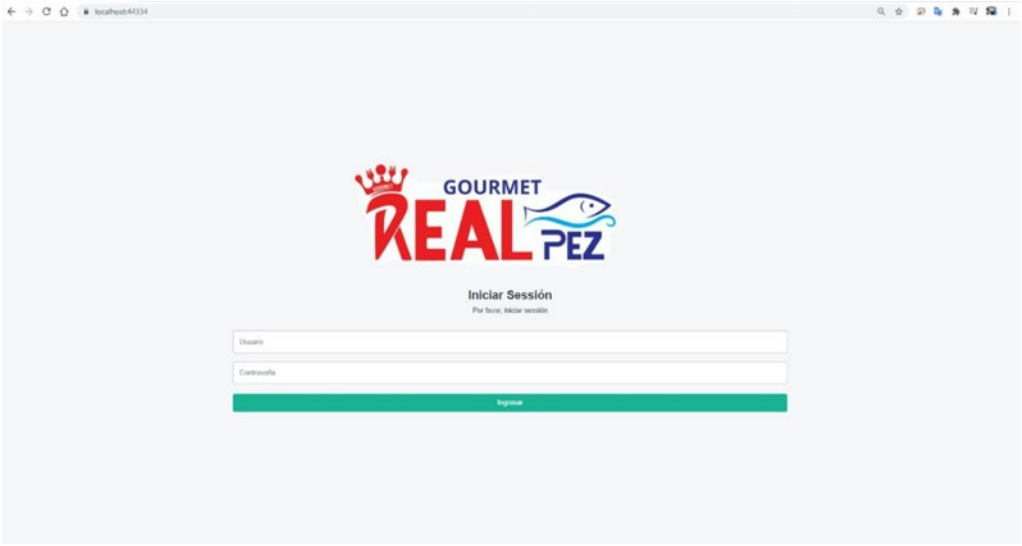


FIGURA 63 Login del sistema

Elaborado por: los autores

Segunda interfaz página de inicio: Esta interfaz permitió observar las dos cámaras en vivo conectadas, asimismo en lado izquierdo superior se encuentra la opción de menú (ver figura 64) por debajo se tuvo los

dos modelos de Machine Learning con dos botones respectivos que permitió observar el histórico de violencia así como también el historial de mensajes enviado por WhatsApp (ver figura 65).

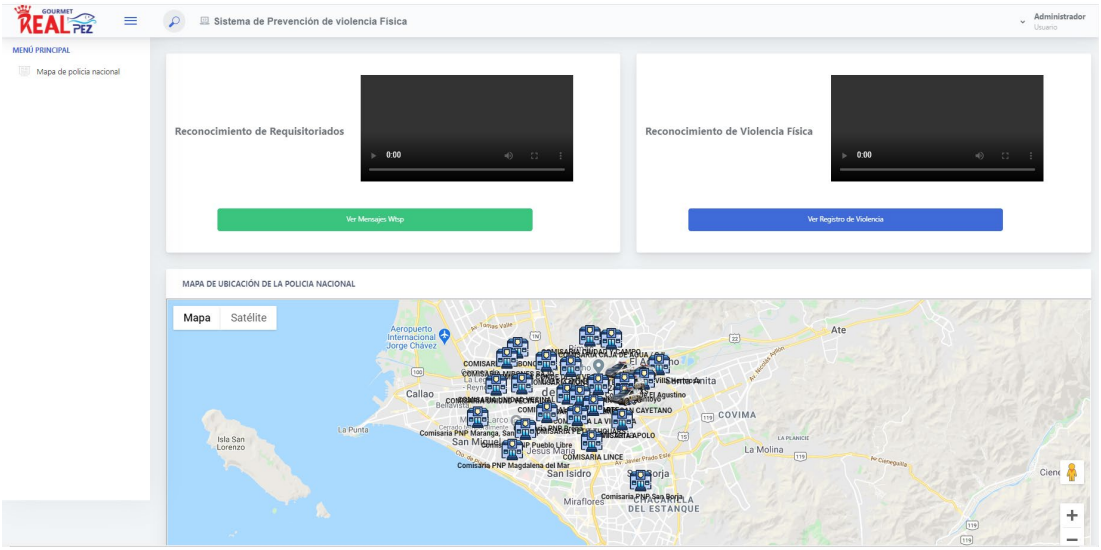


FIGURA 64 Pantalla principal

Elaborado por: los autores

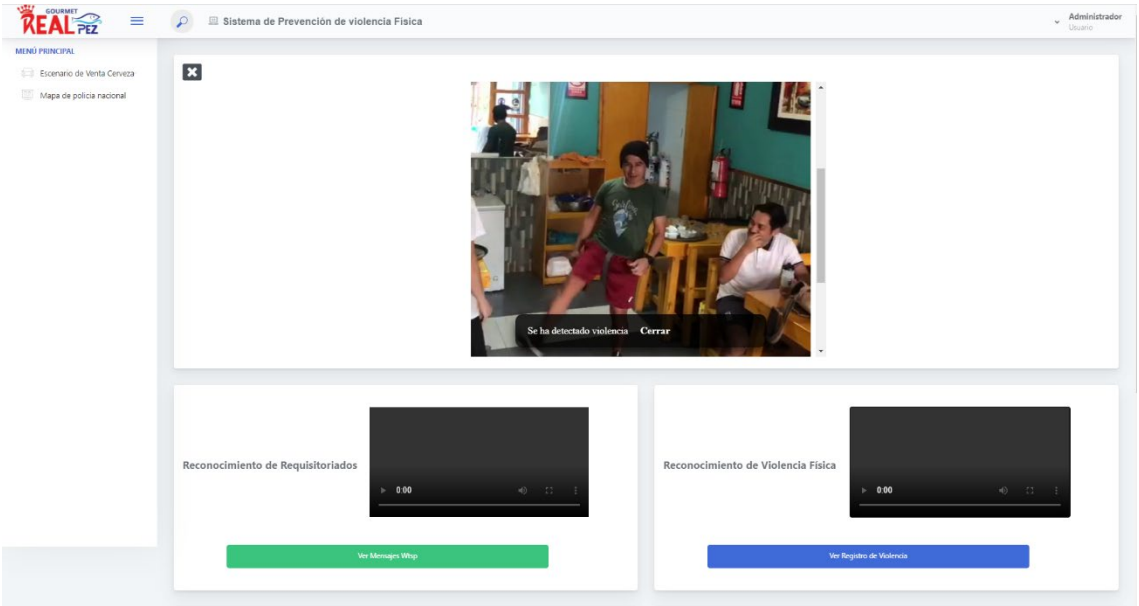


FIGURA 65 Pantalla Cámara-Reconocimiento de Violencia Física

Elaborado por: los autores

Tercera interfaz mapa en vivo de comisarias: Interfaz con la que se cuenta es un mapa que permite saber las patrullas y comisarias que se encuentren en Lima, que al darle click a una patrulla indica el tiempo y distancia en el que se encuentran del restaurante. (ver figura 66).

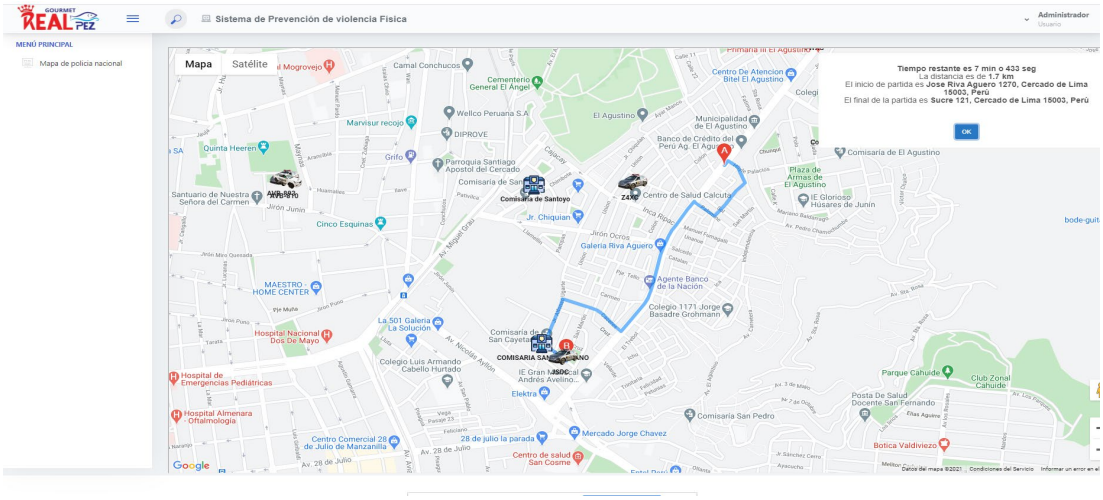


FIGURA 66 Pantalla Mapa de Ubicación Empresa de Seguridad

Elaborado por: los autores

Cuarta interfaz historial de registros de violencia física: Finalmente, la última interfaz con la que se encuentra el historial de violencia física en un archivo plano con la hora, fecha del incidente ocurrido. (ver figura 67).

FIGURA 67 Repositorio de Evidencia de Violencia física

Elaborado por: los autores

4.3 Pruebas

4.3.1 Integrar los modelos al sistema web

Para la integración de los modelos prevención se crearon dos clases params.py, server2.py las cuales están establecidas de la siguiente manera:

a) Params.py:

En esta clase se realizó la declaración de campos (ver figura 68) para la integración de los modelos, las funciones para el tiempo de demora de detención, el porcentaje de detención de una acción y se estableció el puerto del local host.

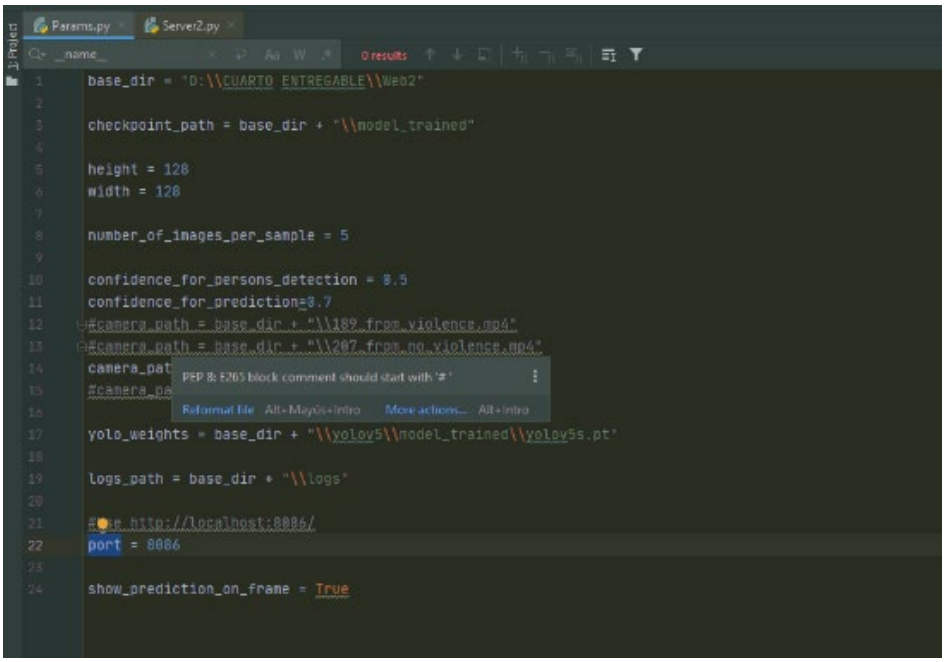


FIGURA 68 Clase Params para la integración

Elaborado por: los autores

b) Server2.py:

Esta última clase permitió importar y establecer la arquitectura del modelo, creando las variables (ver figura 69) para la prevención del escenario.

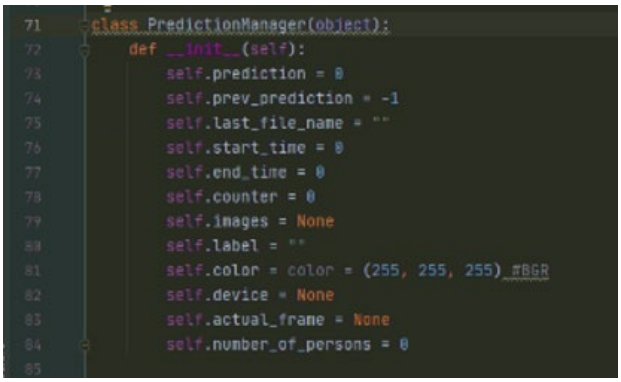


FIGURA 69 Variables de la arquitectura

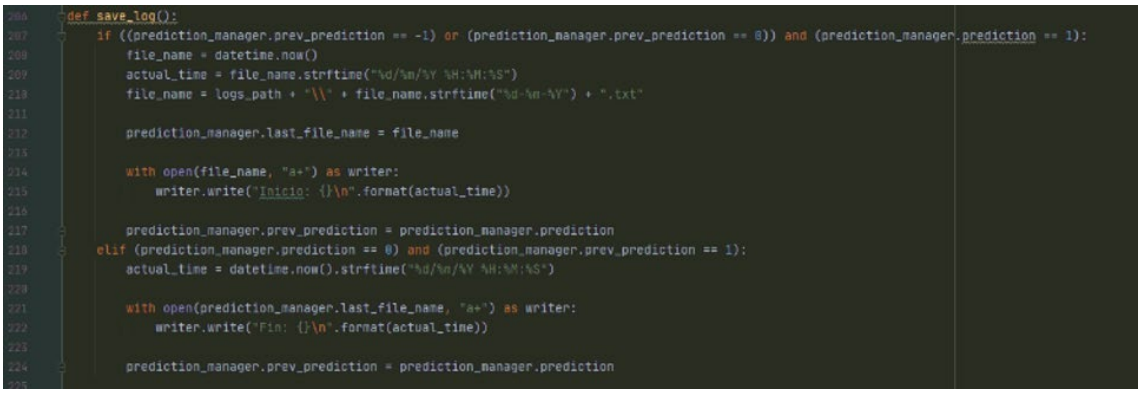
Elaborado por: los autores

Posteriormente, hace el llamado a la función “get_model” (ver figura 70) para obtener al modelo de detección de violencia física previamente creado (ver sección 4.1.4 del capítulo 4)

FIGURA 70 Función para obtener el modelo de detección

Elaborado por: los autores

Para guardar el registro de los resultados positivos de la detección de información se utiliza la función “def_save_log” (ver figura 71)



```
def save_log():
    if ((prediction_manager.prev_prediction == -1) or (prediction_manager.prev_prediction == 0)) and (prediction_manager.prediction == 1):
        file_name = datetime.now()
        actual_time = file_name.strftime("%d/%m/%Y %H:%M:%S")
        file_name = logs_path + "\\" + file_name.strftime("%d-%m-%Y") + ".txt"

        prediction_manager.last_file_name = file_name

        with open(file_name, "a+") as writer:
            writer.write("Inicio: {}\n".format(actual_time))

        prediction_manager.prev_prediction = prediction_manager.prediction
    elif (prediction_manager.prediction == 0) and (prediction_manager.prev_prediction == 1):
        actual_time = datetime.now().strftime("%d/%m/%Y %H:%M:%S")

        with open(prediction_manager.last_file_name, "a+") as writer:
            writer.write("Fin: {}\n".format(actual_time))

        prediction_manager.prev_prediction = prediction_manager.prediction
```

FIGURA 71 Función para el registro

Elaborado por: los autores

Finalmente, en la clase “server2.py” a través de la función “*@app_route*” se declara la variable *model_prediction* los cuales son los dos modelos (ver figura 72) y esto con el fin de que puedan ser llamados en la clase “index.html”.

```
@app.route("/streaming")
def streaming():
    return Response(get_frame(camera, frames_queue, prediction_manager), mimetype="multipart/x-mixed-replace; boundary=frame")

@app.route("/model_prediction")
def model_prediction():
    return jsonify({"prediction": prediction_manager.prediction})

@app.route("/")
def home():
    return render_template("index.html")
```

FIGURA 72 Función app_route

Elaborado por: los autores

4.3.2 Pruebas del sistema

Se realizó las pruebas del requerimiento funciona y no funcional con la finalidad de verificar la correcta función del código respetando y cumpliendo lo solicitado por el usuario.

a) Plan de Pruebas

La ejecución de esta tarea se realizó en base al Plan de Pruebas que se realizó en la fase de Planificación de la Tesis, con la finalidad de cumplir los requerimientos funcionales y no funcionales. Ver Anexo 26.

b) Aceptación Plan de Prueba

La aceptación del plan de pruebas se elaboró como un documento externo, se deja validado en el Anexo 27.

CAPÍTULO V

RESULTADOS

En este capítulo se presentan los resultados obtenidos de las pruebas realizadas tanto del sistema web como de los dos modelos:

5.1 Resultados primer objetivo

Identificar y entrenar el algoritmo que permita identificar personas asociados a posibles actos de violencia y acciones violentas usando las librerías OpenCV y TensorFlow.

En el anexo 18 se observa los resultados que obtuvimos de las pruebas por los 50 escenarios del primer modelo para identificar acciones violentes y en el anexo 17 los resultados por los 40 escenarios del segundo modelo para identificar personas asociados a posibles actos de violencia. A continuación, se describen solo 4 escenarios (verdadero positivo, verdadero negativo, falso positivo y falso negativo):

5.1.1 Resultados del primer modelo para identificar acciones violentas

El resultado de escenarios de verdadero positivo (VP) después de la inferencia del modelo donde la realidad es que existe violencia física y la predicción de la red neuronal es “detección de violencia física” fue de 46 casos identificados correctamente, como se puede observar en la figura 73.

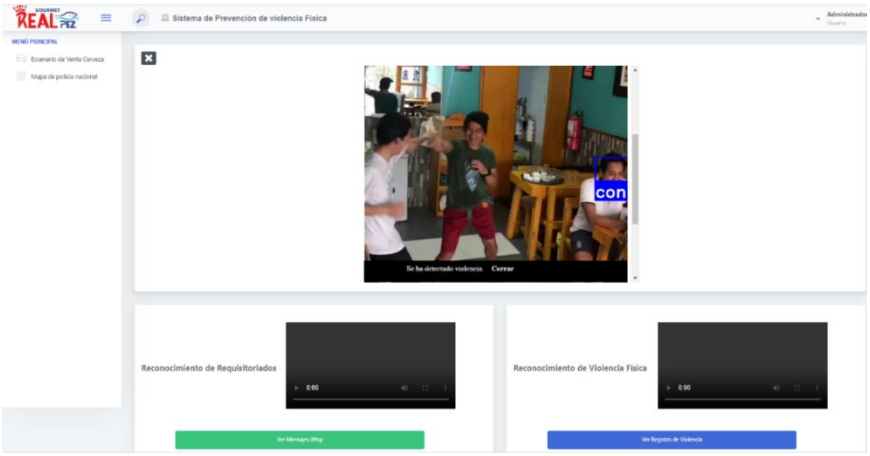


FIGURA 73 Resultado de un escenario de verdadero positivo con violencia

Elaborado por: los autores

El resultado de escenarios de verdadero negativo (VN) después de la inferencia del modelo donde la realidad es que no existe violencia física y la predicción de la red neuronal no identifico nada fue de 43 casos identificados correctamente, ver figura 74.

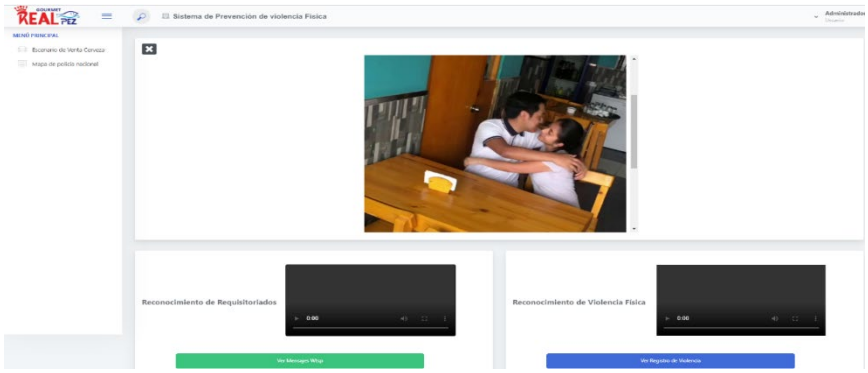


FIGURA 74 Resultado de verdadero negativo con violencia

Elaborado por: los autores

El resultado de escenarios de falso positivo (FP) después de la inferencia del modelo donde la realidad es que no hay violencia física y la predicción de la red neuronal es “detección de violencia

física” fue de 4 casos identificados erróneamente, como se puede observar en la figura 75.

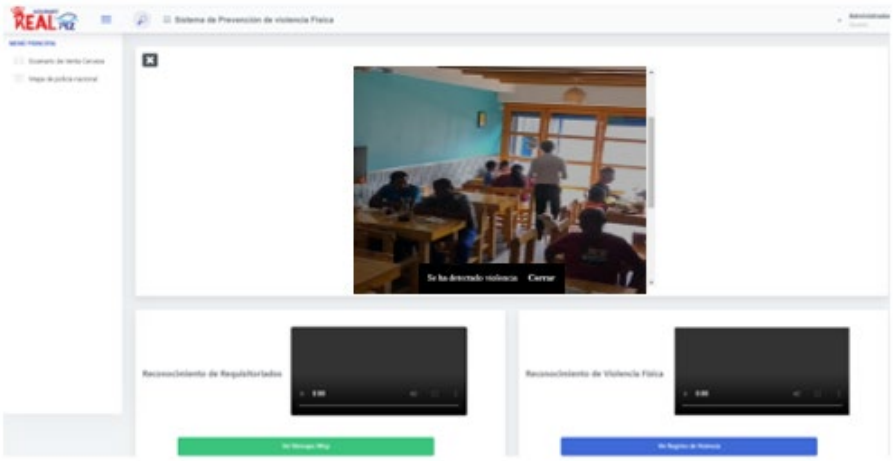


FIGURA 75 Resultado de un escenario falso positivo sin violencia

Elaborado por: los autores

El resultado de escenarios de falso negativo (FN) después de la inferencia del modelo donde la realidad es que existe un acto de violencia física y la predicción de la red neuronal no lo identificó fueron de 7 casos dando como resultado erróneo, como se puede observar en la figura 76.

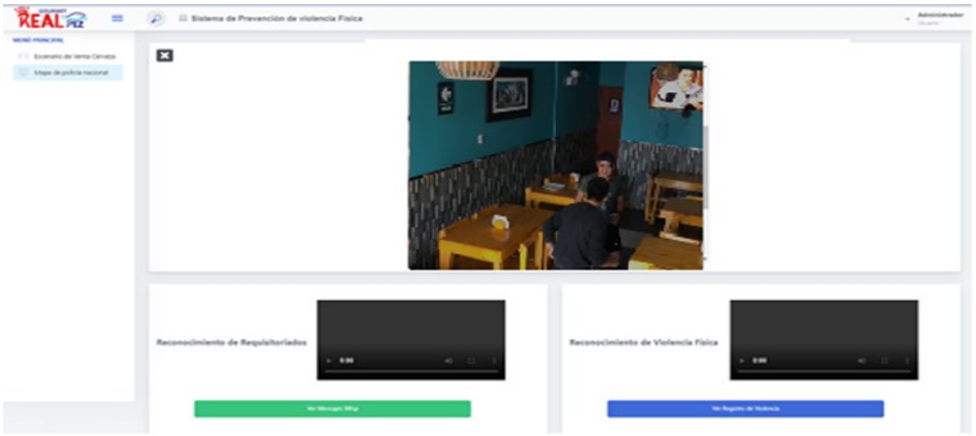


FIGURA 76 Resultado de un escenario falso negativo sin violencia

Elaborado por: los autores

La fórmula de la exactitud resultó de la siguiente manera:

$$\text{Exactitud} = (46+43) / (46+43+4+7)$$

$$\text{Exactitud} = 0.89$$

Mediante el uso de la fórmula se obtuvo un 89% de exactitud para este modelo donde de 50 escenas que no se encontraba un acto de violencia la red neuronal identificó correctamente 43 casos y en el caso donde existe violencia física la red predijo correctamente 46 casos. La fórmula de la Precisión resultó de la siguiente manera:

$$\text{Precisión} = 46/ (46+4)$$

$$\text{Precisión} = 0.92$$

Mediante el uso de la fórmula se obtuvo un resultado de precisión al 0.92, es decir, cuando el modelo está prediciendo un caso positivo está acertando el 92% de las veces. La fórmula de *Recall* resultó de la siguiente manera:

$$\textit{Recall} = 46/ (46+7)$$

$$\textit{Recall} = 0.89$$

Con la última se obtuvo una recuperación de 0.89, es decir, detecta correctamente el 89% de las escenas donde se encuentra un acto de violencia física.

5.1.2 Resultados del segundo modelo para identificar personas asociados a posibles actos de violencia

El resultado de escenarios de verdadero positivo (VP) después de la inferencia del modelo donde la realidad es que la persona está con Requisitoria y la predicción de la red neuronal es “Persona

con Requisitoria” fue de 36 casos identificados correctamente (ver figura 78), como se puede observar en la figura 67 el modelo detecto correctamente las características del rostro debido tomando en cuenta los puntos de intereses en base a la base de datos de personas con requisitoria (ver figura 77).

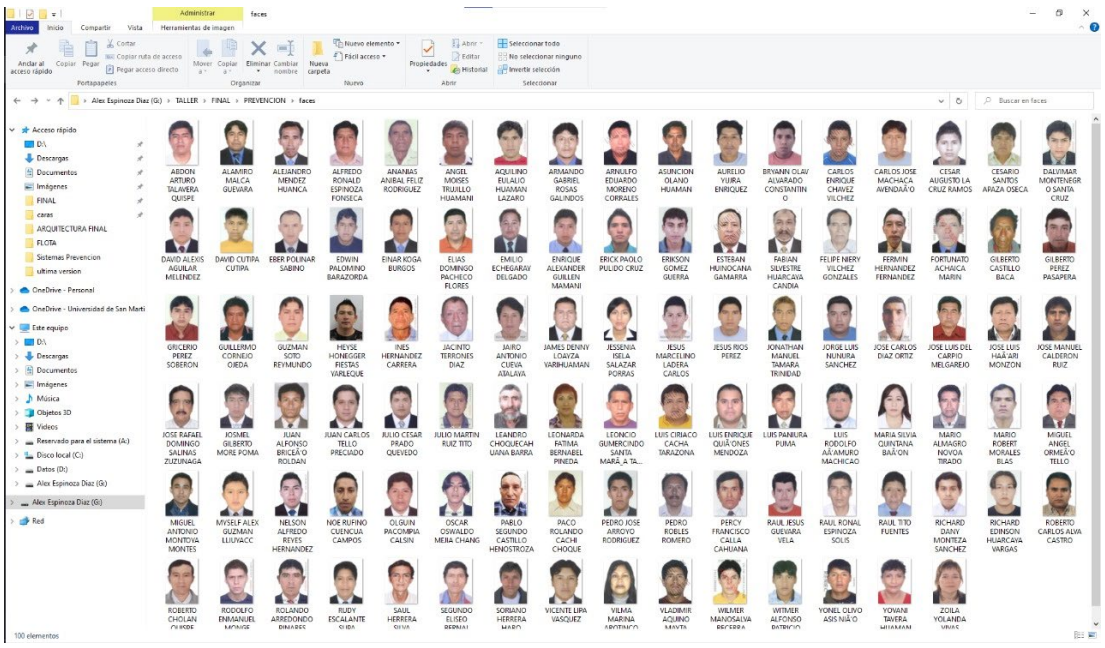


FIGURA 77 Base de requisitoria

Elaborado por: los autores

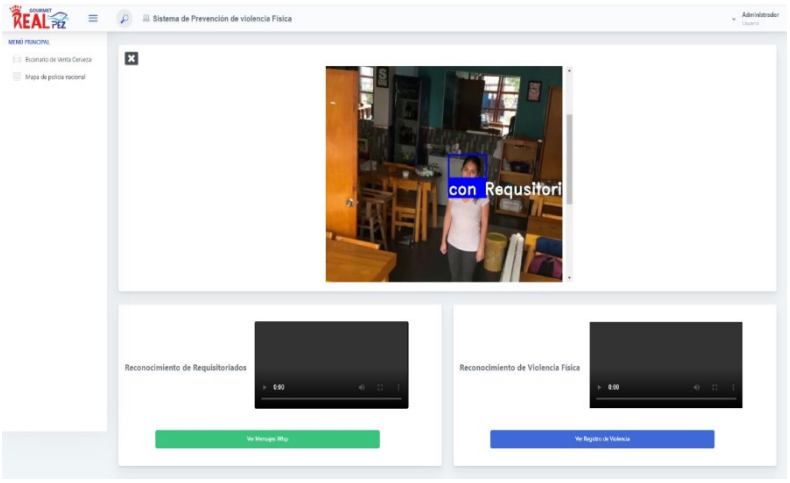


FIGURA 78 Resultado de un escenario con verdadero positivo

Elaborado por: los autores

El resultado de escenarios de verdadero negativo (VN) después de la inferencia del modelo donde la realidad es que la persona no se encuentra con requisitoria y la predicción de la red neuronal es “Persona apta” fue de 38 casos identificados correctamente, como se puede observar en la figura 79 el modelo detecto correctamente las características del rostro identificado como “persona apta”

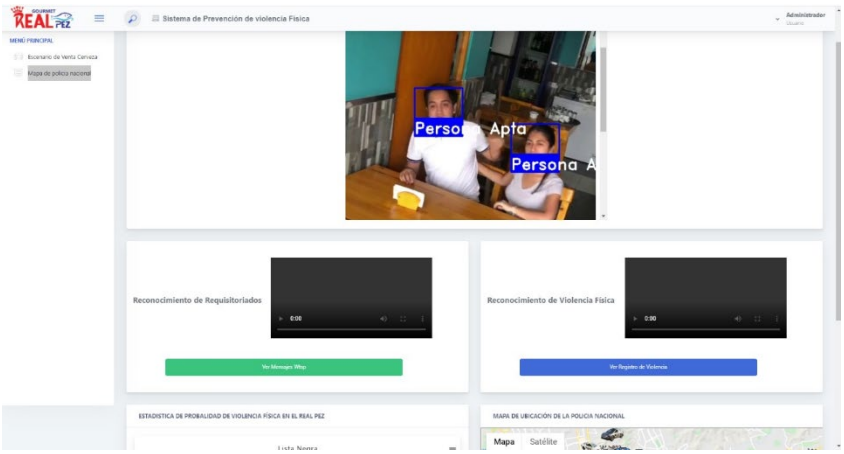


FIGURA 79 Resultado de un escenario de verdadero negativo

Elaborado por: los autores

El resultado de escenarios de falso positivo (FP) después de la inferencia del modelo donde la realidad es que la persona no está con Requisitoria y la predicción de la red neuronal es “Persona con Requisitoria” fue de 2 casos identificados erróneamente, como se puede observar en la figura 80 el modelo indica “persona con requisitoria”.

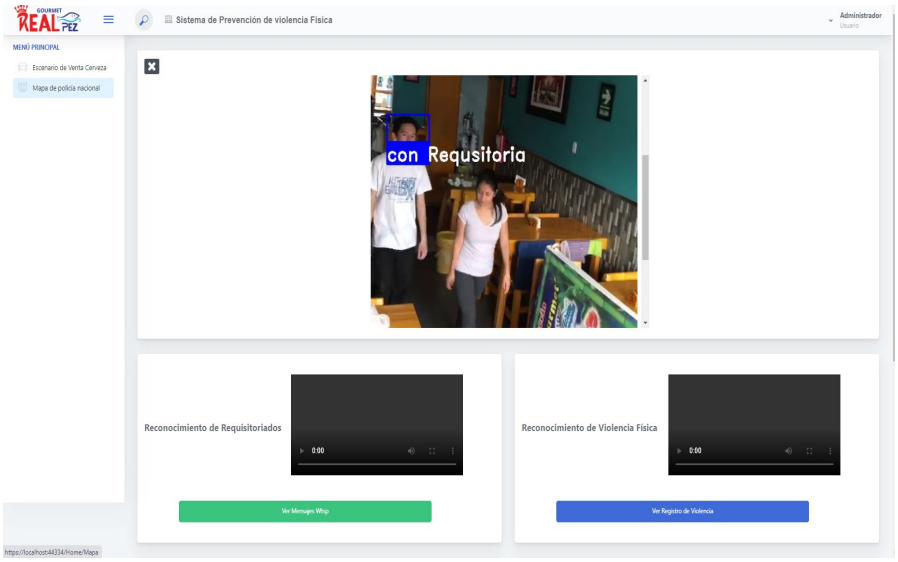


FIGURA 80 Resultado de un escenario falso positivo

Elaborado por: los autores

El resultado de escenarios de falso negativo (FN) después de la inferencia del modelo donde la realidad es que la persona si se encuentra con requisitoria y la predicción de la red neuronal no lo identificó fueron de 4 casos dando como resultado erróneo, como se puede observar en la figura 81.

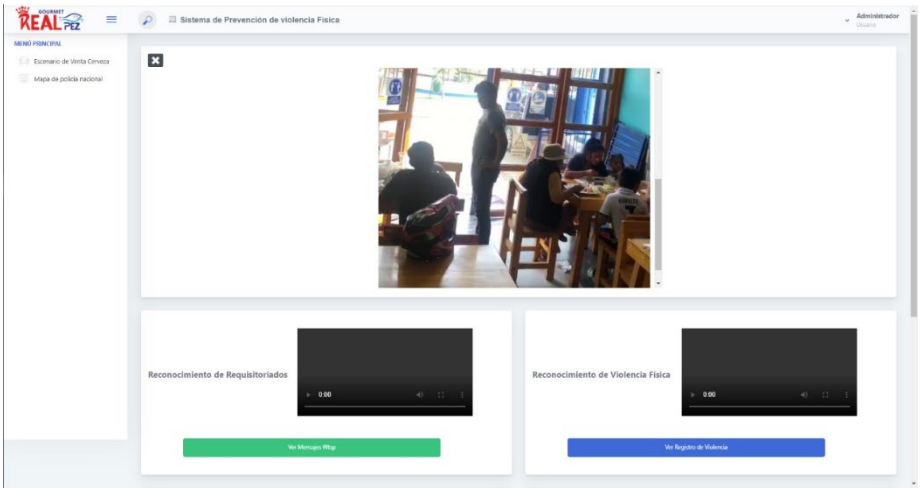


FIGURA 81 Resultado del escenario falso negativo

Elaborado por: los autores

La fórmula de la exactitud quedaría de la siguiente manera:

$$Exactitud= (36+38) / (36+38+2+4)$$

$$Exactitud=0.84$$

Mediante el uso de la fórmula se obtuvo el 84% de exactitud para este modelo donde de 40 escenas que no se encontraba “Persona con Requisitoria”, la red neuronal identificó correctamente 38 casos y en el caso de “Persona con Requisitoria” la red neuronal predijo correctamente 36 casos. La fórmula de la Precisión resultó de la siguiente manera:

$$Precisión= 36/ (36+2)$$

$$Precisión=0.95$$

Mediante el uso de la fórmula se obtuvo un resultado de precisión al 0.95, es decir, está acertando el 95% de las veces los casos positivos. La fórmula de Recall resultó de la siguiente manera:

$$Recall= 36/ (36+4)$$

$$Recall=0.9$$

Con la última se obtuvo una recuperación de 0.9, es decir, detecta correctamente el 90% de las escenas donde se encuentra una persona con Requisitoria.

5. 2 Resultados del segundo objetivo

Diseñar y desarrollar interfaces que identifiquen personas asociadas a posibles escenarios de violencia para su prevención y detección de escenas de violencia física mediante reconocimiento de movimiento.

En base a los dos modelos identificar acciones violentes e identificar personas asociados a posibles actos de violencia se observó que cuando el modelo detecta correctamente el sistema emite una alerta.

5.2.1 Resultados del primer modelo para identificar acciones violentas

Para este escenario, el flujo es similar al primer escenario en donde se observa en la figura 82, que el sistema detecto un acto de violencia física entonces automáticamente se envió una alerta.

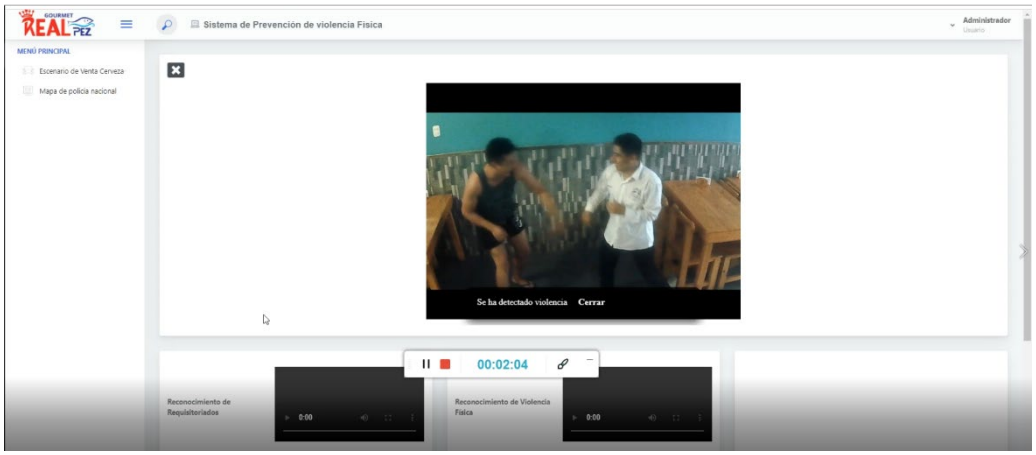


FIGURA 82 Detección de violencia física en el sistema

Elaborado por: los autores

5.2.2 Resultados del segundo modelo para identificar personas asociados a posibles actos de violencia

Como se observa en la figura 83, el sistema detecto una persona con requisitoria entonces automáticamente se envió una alerta de “con requisitoria”

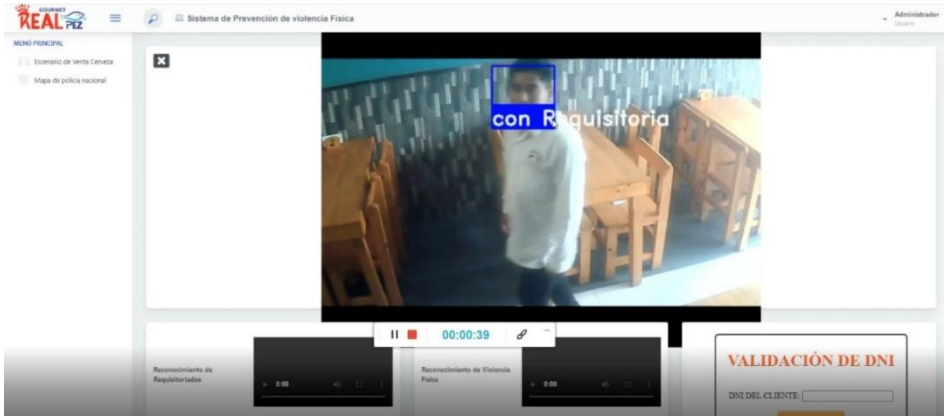


FIGURA 83 Detección de persona con requisitoria en el sistema

Elaborado por: los autores

5.3 Resultados del tercer objetivo

Desarrollar interfaces para emitir alertas y comunicar cuando se identifique posibles escenarios de violencia o detecte escenas de violencia física.

En base a los dos modelos identificar acciones violentes e identificar personas asociados a posibles actos de violencia se observó que cuando el modelo detecta correctamente comunica inmediatamente la alerta por medio de un mensaje a través de WhatsApp enviando la ubicación de esta misma.

5.3.1 Resultados del primer modelo para identificar acciones violentas

Para este escenario, el flujo es similar al primer escenario en donde se observa en la figura 84, el sistema detecto un acto de violencia física entonces envió un mensaje de WhatsApp y la ubicación del restaurante (ver figura 85). Asimismo, a través del módulo “mapa de ubicación” se puede observar y calcular a cuánto tiempo la persona que recibió el mensaje se encuentra desde la ubicación.

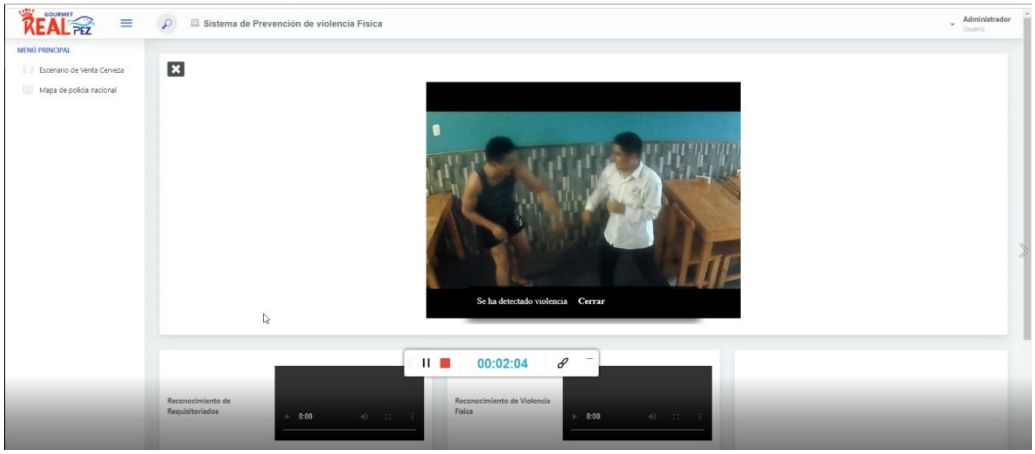


FIGURA 84 Detección de violencia física en el sistema

Elaborado por: los autores

FIGURA 85 Envió automáticamente mensaje de WhatsApp indicando que existe violencia en el Real Pez SAC y la ubicación del restaurante

Elaborado por: los autores

5.3.2 Resultados del segundo modelo para identificar personas asociados a posibles actos de violencia

Como se observa en la figura 86, el sistema detecto una persona con requisitoria entonces envió un mensaje de WhatsApp indicando los datos de la persona y la ubicación del restaurante (ver figura 87). Asimismo, a través del módulo “mapa de ubicación” se puede observar y calcular a cuánto tiempo la persona que recibió el mensaje se encuentra desde la ubicación.

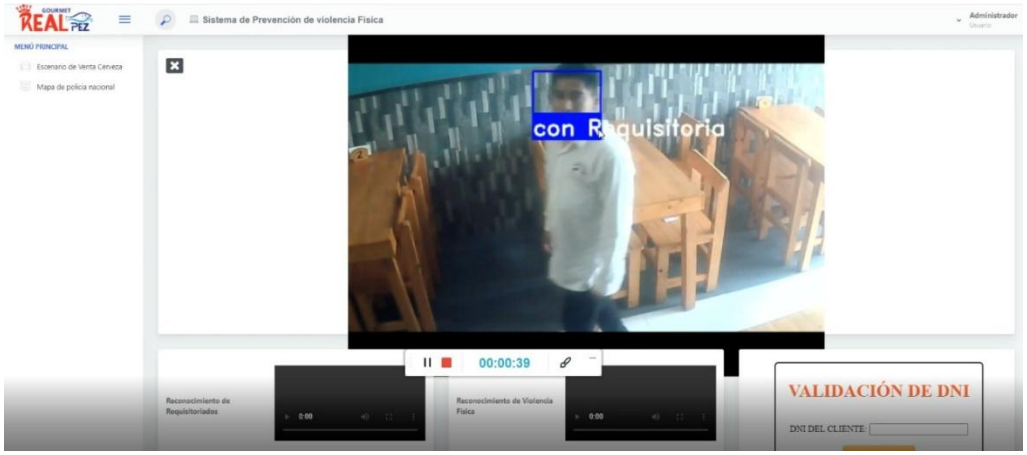


FIGURA 86 Detección de persona con requisitoria en el sistema

Elaborado por: los autores

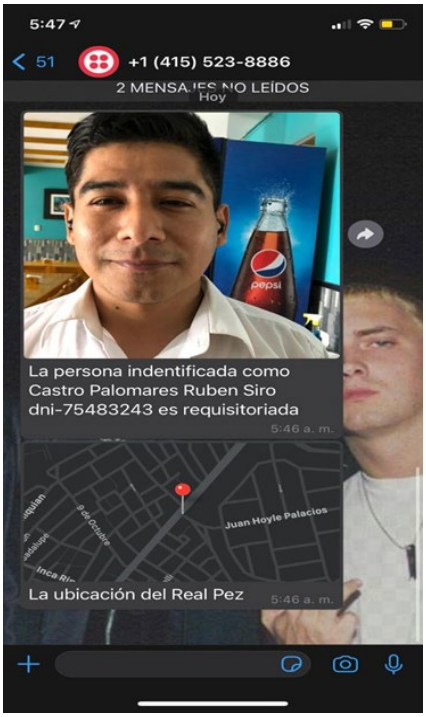


FIGURA 87 Envió automáticamente mensaje de WhatsApp indicando los datos de la persona y la ubicación del restaurante

Elaborado por: los autores

Asimismo, a través de la interfaz con el sistema Twilio se puede observar el estado en el que se encuentra el mensaje de WhatsApp “leído” y/ o “recibido” (ver figura 88)

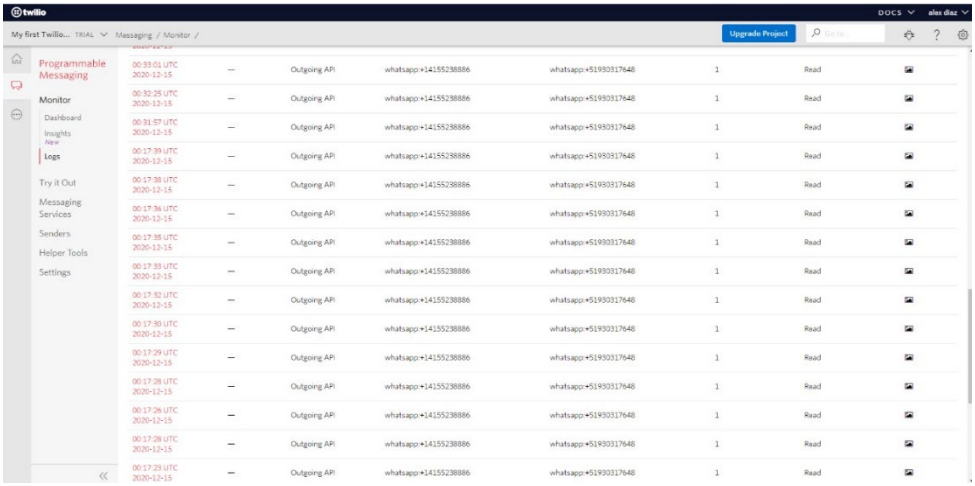


FIGURA 88 interfaz con el sistema Twilio se puede observar el estado en el que se encuentra el mensaje de WhatsApp

Elaborado por: los autores

CAPÍTULO VI

DISCUSIONES

Esta investigación tuvo como propósito crear un sistema que integre interfaces de permitan identificar posibles escenarios violencia física en la sucursal principal de la empresa Real Pez SAC., para ellos se utilizó algoritmos de aprendizaje supervisado de Machine Learning y como indica Silva (2020) en su tesis “se utilizó un conjunto de datos de entrenamiento, donde los diferentes ejemplos estaban asociado con una etiqueta u objetivo, el cual dieron un resultado del ejemplo dado”. para esto se tuvo un análisis experimental de un conjunto de datos, con el que se conocieron los escenarios previos a este tipo de actos delictivos. A continuación, se realizó una discusión con los hallazgos de esta investigación:

a) Con relación al primer objetivo planteado:

Identificar y entrenar el algoritmo que permita identificar personas asociados a posibles actos de violencia y acciones violentas usando las librerías OpenCV y TensorFlow.

Las fuentes de información utilizadas para este objetivo se pueden observar en la sección 2.2.7.3 y 2.2.7.4 del capítulo II donde:

Para identificar un escenario previo al acto de violencia física, se realizó un modelo de reconocimiento facial de personas con requisitoria para esto se utilizó la librería OpenCV que como indica Bustamante (2014) “esta librería a través de sus funciones permite extraer y procesar datos mediante imágenes

basándose en un algoritmo", de los resultados que obtuvimos se identificó y extraer imágenes de las caras a través de los videos de la cámara de seguridad en tiempo real. Asimismo, la función face_recognition el cual hace el llamado a un modelo pre entrenado que a través de 128 dimensiones realiza la detección de un rostro. para el modelo con una base previamente creada y cargada de personas con requisitoria se realiza el llamado de las funciones: compare_faces que permite compara comparar con las caras entrenadas del modelo, face_locations localiza las caras en las imágenes mediante un algoritmo ya entrenado, face_encodings que encuentra las codificaciones faciales del fotograma, previamente extraídas con la librería OpenCV y face_distance que mide la distancia entre caras para encontrar una mayor probabilidad de match, siendo óptimos al usar las dos librerías puesto que en la mayoría de los casos se obtuvo una correcta identificación y con una mayor precisión del 95% al encontrarse a 1.5 metros de la cámara o 2 metros de la cámara en comparación de los autores Barreto y Lizarraga (2019) en su tesis "Modelo de Sistema de Reconocimiento Facial para el Control de la Trata de Personas" tuvieron un resultado el 60% como mínimo y como máximo uno de 93%, concluyendo que el modelo es más efectivo al encontrarse hasta un metro de distancia y a una altura de 1.5 metros en un ambiente interior con mucha luz, que igualaron a las condiciones de la mañana y con una mínima resolución de 8 megapíxeles.

Por otro lado, el modelo tuvo un *feedback* automático que se conecta con la base de datos y puede ser actualizado y entrenar el modelo.

Para el segundo algoritmo que permita detectar un acto de violencia física se utiliza el keras que se encuentra dentro de la librería TensorFlow. Así, este es un API desarrollado para la detección de imágenes que permiten y facilitan la creación y entrenamiento de modelos como detección de objetos (TensorFlow, 2020). para tener un correcto modelo primero se debe crear el modelo ingresando datos para llevar a cabo a través de las cinco capas (Input, GaussianNoise, Conv2D, Flatten y Dense), pero en realidad no tuvo ningún valor, ya que en la segunda instancia se entrena el modelo con información de la base de datos. Asimismo, se llama a la función *prediction* para que obtenga la predicción de los frame enviados por el video y envíe un resultado.

Este modelo de detección de violencia física tuvo como resultado un 89% de exactitud al usar keras y TensorFlow a través de cinco capas.

b) Con relación al segundo objetivo planteado:

Diseñar y desarrollar interfaces que identifiquen personas asociadas a posibles escenarios de violencia para su prevención y detección de escenas de violencia física mediante reconocimiento de movimiento.

Para el segundo objetivo, para el diseño y desarrollo de la interfaz principal se utilizó AJAX que a través de diferentes técnicas de desarrollo web permitió prevenir demoras, las mismas de las peticiones y respuestas del servidor. Así como lo indica Krall (2018) “AJAX permite que se realicen posibles peticiones al servidor y de segundo plano tener una respuesta sin tener que recargar página web completa y para esto se usan esos datos y modifica los contenidos de la página” (p.1).

Por otro lado, para que el sistema pueda identificar personas asociadas a posibles escenarios de violencia y detectar escenas de violencia física, se integró los dos modelos previamente entrenados con el desarrollado de las interfaces a través del macro web Flask sin que los modelos entrenados se vean afectados. Como indica Domingo (2018) “Flask permite crear el puente de los modelos con la página web, a través de la función @app.route("/")” (p.1).

c) Con relación al tercer objetivo planteado:

Desarrollar interfaces para emitir alertas y comunicar cuando se identifiquen posibles escenarios de violencia o detecten escenas de violencia física.

Los resultados fueron exitosos cuando el sistema a través de los modelos detectó correctamente los escenarios porque realizó el envío automático de las alertas conteniendo la ubicación del lugar en el que ocurrieron los hechos. En el caso de la alerta en el escenario de detección de personas

con requisitoria, adicionalmente, envía la imagen de la persona requisitoriada para la comunicación de la prevención de escenas de violencia física, esto permitió a la empresa tener una mayor probabilidad de mantener seguros a sus comensales como a sus trabajadores, y no solo tener cámaras de seguridad como un efecto disuasorio, sino también permitió tener la exactitud de la distancia en la que se encuentra la persona que recibió el mensaje, cabe mencionar que la cajera o persona que utilizo el sistema en la empresa debió ingresar a la interfaz del sistema Twilio donde pudo observar el estado, ya sea “leído” o “recibido”, en el que se encontraba su mensaje.

CONCLUSIONES

1. Se utilizó el aprendizaje supervisado en redes neuronales incluyendo un conjunto de librerías con algoritmos que permitieron realizar el procesamiento de imágenes a través de la extracción de fotogramas de videos para la identificación de personas asociados a posibles actos de violencia y acciones violentas.

2. En base al objetivo específico 1, se concluyó que los modelos usando las librerías de OpenCV, TensorFlow (keras) para la identificación de personas con requisitoria y detección de violencia física tuvo un porcentaje alto de asertividad cuando la cámara enfoca directamente al rostro o la acción de violencia, pero los modelos son menos precisos cuando la cámara no enfoca en contraluz. los modelos propuestos según las métricas aplicadas tuvieron una mayor precisión, evidencia de ello es el 95% de asertividad para detectar acciones de violencia cuando la cámara se encuentra a 1.5 o 2 metros de distancia del objetivo; asimismo, la precisión en la identificación de puñetes o patadas alcanzó el 92%, lo que demuestra el éxito de la propuesta.

3. En base al objetivo específico 2 se logró diseñar y desarrollar interfaces en un sistema Web, estas fueron integradas con éxito a los dos modelos

entrenados para identificar personas asociadas a posibles escenarios de violencia y detectar escenas de violencia física.

4. En base al objetivo específico 3, al comunicar los hechos de prevención y detección de violencia de una forma inmediata y precisa, y considerando la exactitud de la distancia a la que se encuentra la empresa de seguridad o persona que recibió la comunicación, se determinó que el sistema sí permite que la empresa brinde una probabilidad mayor de seguridad para sus clientes y trabajadores.

RECOMENDACIONES

1. Para los futuros trabajo que tomen de referencia el modelo de detección de violencia física propuesto se recomienda buscar base de datos a gran escala que ofrece información de clips y fotogramas para el entrenamiento del modelo.

2. Se recomienda almacenar los videos del resultado de los modelos en una base de datos en la nube como evidencias de respaldo para cualquier incidente que ocurra dentro del restaurante.

3. Por el resultado del modelo de detección utilizando TensorFlow puede requerir mucha potencia informática. para acelerar se recomienda instancias de GPU AWS EC2 de la nube de Amazon para ejecutar varios modelos en paralelo.

FUENTES DE INFORMACIÓN

Ammar, S.R., Anjum, M.R., & Islam, M.T. (2019). Using deep learning algorithms to detect violent activities. Recuperado de http://dspace.bracu.ac.bd/xmlui/bitstream/handle/10361/12270/15101026,16301140,15301133_CSE.pdf?isAllowed=y&sequence=1

Arreola, J. D. J. V. (2019). Identificación de peatones en imágenes aéreas con redes neuronales explicativas y fusión de sensores. Recuperado https://207.249.117.40/jspui/bitstream/1009/1787/1/VelazquezAJJ.pdf

Asociación Española de Periatría. (2018). Tipos de Violencia. Tipos de Violencia. España. Recuperado el 24 de Noviembre de 2019, de https://www.aeped.es/una-vision-global-violencia-contra-ninos/definiciones

Barradas Arenas, U., Bárcenas Cortes, A., Sánchez Hernández, M., & Hernández Chan, G. (06 de 2017). Implementación de sistema de video cámaras IP como medio de seguridad para el Tecnológico de Álbaro obregón. Revista Académica de la Facultad de Ingeniería, Universidad Autónoma de Yucatán, 21(2), 11.

Barraza, S. L., Thuillier, E., Will, A., & Rodriguez, S. A. (2013). Primeros pasos para una aplicación móvil offline de reconocimiento facial. Recuperado de: www.conaiisi.unsl.edu.ar/2013/205-529-1-DR. pdf.

Barreto Rodriguez, R. M., & Lizarraga Mendoza, D. J. (2019). Modelo de sistema de reconocimiento facial para el control de la trata de personas. (tesis de pregrado) Universidad Tecnológica del Perú. Recuperado de https://revistas.uandina.edu.pe/index.php/mastariy/article/download/171/138/

Bonilla Alguera, G. (2020). Enrique Desmond Arias, Criminal Enterprises and Governance in Latin America and the Caribbean, Cambridge, Cambridge University Press, 2017, 302 pp. Gestión y política pública, 29(1), 262-266.

Bortolotti N. (2017) Tensorflow Object Detection en videos, basta de pizza, donuts. Recuperado de https://nbortolotti.blogspot.com/2017/07/tensor-flow-object-detection-en-videos.html

Brahmbhatt, S. (2013). Introduction to computer vision and opencv. In Practical OpenCV (pp. 3-5). Apress, Berkeley, CA.

Cáceres Mariño, E. L. (2018). Aplicación móvil de reconocimiento facial en personas con antecedentes de abuso sexual en la provincia de Andahuaylas, Apurímac-2018.Recuperado de https://repositorio.unajma.edu.pe/bitstream/handle/123456789/358/Ervin_Lewis_Tesis_Bachiller_2018.pdf?sequence=1&isAllowed=y

Calvo D. (2017) Red Neuronal Convolucional CNN. Recuperado de: https://www.diegocalvo.es/red-neuronal-convolucional/

Carrero D., B. Ruíz, L. Puente y M.J. Poza (2009). Prestaciones de la Normalización del Rostro en el Reconocimiento Facial. Universidad Carlos III de Madrid.

Chacón, J. C. R. (2006). Aplicación de la metodología RUP para el desarrollo rápido de aplicaciones basado en el estándar J2EE. Guatemala:(tesis de grado) para obtener el título de ingeniería en ciencias y sistemas-Universidad de San Carlos de Guatemala. Recuperado de http://biblioteca.usac.edu.gt/tesis/08/08_0308_CS.pdf

Cheng, G., Wan, Y., Saudagar, A. N., Namuduri, K., & Buckles, B. P. (2015). Advances inhuman action recognition: A survey. Recuperado de http://arxiv.org/abs/1501.05964

Contreras Arteaga, A. I., & Sánchez Cotrina, F. W. (2020). Analítica predictiva para conocer el patrón de consumo de los clientes en la Empresa Cienpharma SAC utilizando IBM SPSS Modeler y la metodología CRISP-DM.

Collobert, R., Kavukcuoglu, K., & Farabet, C. (2011). Torch7: A matlab-like environment for machine learning. In BigLearn, NIPS workshop (No. CONF). Recuperado de https://infoscience.epfl.ch/record/192376/files/Collobert_NIPSWORKSHOP_2011.pdf

Coşkun, S. (2019). Developing a Face Recognition System for Indoor Security (Doctoral dissertation, İzmir Institute of Technology).

Datta, A., Shah, M., and Lobo, N. D. V. (2002). Person-on-person violence detection in video data. In Pattern Recognition, 2002. Proceedings. 16th International Conference on, volume 1, pages 433–438. IEEE

Domingo Jose (2018.) Flask: Enrutamiento (2ª parte). Recuperado de: https://www.josedomingo.org/pledin/2018/03/flask-enrutamiento-2a-parte/#:~:text=Enrutamiento%3A%20rutas,que%20se%20hace%20la%20petici%C3%B3n.

Fernández Caraballo, E., & Gómez Franco, Y. (2018). Metodología para el análisis de la violencia en el departamento de Bolívar mediante técnicas de machine learning. Recuperado de https://repositorio.utb.edu.co/bitstream/handle/20.500.12585/1118/0074561.pdf?sequence=1&isAllowed=y

Fernández, E & Gómez, Y (2018). Metodología para el análisis de la violencia en el departamento de Bolívar mediante técnicas de machine learning [Tesis de pregado, Universidad Tecnológica de Bolivar] Recuperado de https://biblioteca.utb.edu.co/notas/tesis/0074561.pdf

Fernández Martínez, L. C. (2017). Identificación automática de acciones humanas en secuencias de video para soporte de videovigilancia. Recuperado de http://tesis.pucp.edu.pe/repositorio/bitstream/handle/20.500.12404/13049/FERN%c3%81NDEZ_MART%c3%8dNEZ_LUIS_CHRISTIAN.pdf?sequence=1&isAllowed=y

Garcés, A., & Jurado, M. (2016). Sistema de reconocimiento facial con visión artificial para apoyar al ECU 911 con la identificación de personas en la lista de los más buscados.

Geitgey, A. (2020). Face Recognition. GitHub. Recuperado de: https://github.com/ageitgey/face_recognition.

Gestión (2019) Estos son los 120 distritos del Perú con mayor delincuencia y violencia del país, según la PNP. Recuperado de https://gestion.pe/peru/policia-detecta-120-distritos-crimenes-violencia-269349-noticia/?ref=gesr

Gonzales, A. (2016). ¿Qué es Machine Learning? Cleverdata. Recuperado de: https://cleverdata.io/que-es-machine-learning-big-data/

Gonzáles, A. (2018). Basic machine learning concepts. Cleverdata. Recuperado de: https://cleverdata.io/basic-machine-learningconcepts/

Gonzales Ojeda, M. (2017). Violencia contra la mujer en el distrito de Santiago de Surco.

Gori, M. (2017). Machine Learning: A Constraint-Based Approach. Morgan Kaufmann.

Grapheverywhere. (2019). Machine Learning en Python. Recuperado de: https://www.grapheverywhere.com/machine-learning-en-python/

Guillén-Gámez, F. D., García-Magariño, I., & Palacios-Navarro, G. (2018, March). Comparative Analysis Between Different Facial Authentication Tools for Assessing Their Integration in m-Health Mobile Applications. In World Conference on Information Systems and Technologies (pp. 1153-1161). Springer, Cham.

Gustavo B. (2019) ¿Qué es AJAX y cómo funciona? Recuperado de: https://www.hostinger.es/tutoriales/que-es-ajax#:~:text=JavaScript%20es%20un%20lenguaje%20de,su%20nombre%3A%20eXtensible%20Markup%20Language.

Huaman, Z (2020). Implementación de un sistema de gestión de seguridad electrónica con Machine Learning dirigido a Prosegur Perú para gestión de seguridad en viviendas de Lima Metropolitana [Tesis de

pregrado, Universidad Tecnológica del Perú] Recuperado de: http://repositorio.utp.edu.pe/handle/UTP/2842

Hurwitz, J., y Kirsch, D. (2018). Machine Learning For Dummies. John Wiley y Sons.Recuperado de https://mscdss.ds.unipi.gr/wp-content/uploads/2018/02/Untitled-attachment-00056-2-1.pdf

King, D. E. (2009). Dlib-ml: A machine learning toolkit. The Journal of Machine Learning Research, 10, 1755-1758.

Krall Cesar. (2018). ¿Qué es y para qué sirve Ajax? Ventajas e inconvenientes. JavaScript asíncrono, XML y JSON. Recuperado de https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=882:ique-es-y-para-que-sirve-ajax-ventajas-e-inconvenientes-javascript-asincrono-xml-y-json-cu01193e&catid=78&Itemid=206

Lumba, A. P. L., Yahuarcani, I. O., Cortegano, C. A. G., Satalaya, A. M. N., Llaja, L. A. S., Nuñez, R. R. R., ... & Gómez, E. G. Solución Informática para el Reconocimiento de Acciones Básicas de Violencia en Tiempo Real, a partir del uso de Redes Neuronales Convolucionales, Secuencias de Video y Computación de Alto Desempeño. Economía, 21, 40.

Macedo, L. & Chavez, G (2016) Aplicación de redes neuronales artificiales sobre la violencia de la mujer por su pareja según la encuesta demográfica y de salud familiar, ENDES.

Malacara, D. (2015). Procesamiento de Imágenes. Recuperado de: http://bibliotecadigital.ilce.edu.mx/sites/ciencia/volumen2/ciencia3/084/htm/sec_9.htm

Martí Martí, S. (2013). Diseño de un sistema de televigilancia sobre IP para el edificio CRAI de la Escuela Politécnica Superior de Gandía (Doctoral dissertation).

Ministerio del Interior del Perú (2019). Programa de recompensas – la lista de los más buscados. Lima, Perú. Recuperado de: https://www.recompensas.pe

Norman, A. T. (2019). Aprendizaje Automático en acción: Un Libro Para El Lego, Guía Paso A Paso Para Los Novatos. España: Tektime.

Numpy (2005) About us - Some information about the NumPy project and community. Estados Unidos. Recuperado de: https://numpy.org/about/

OMS, I. M. Consideraciones en torno al «Informe mundial sobre la violencia y la salud» de la Organización Mundial de la Salud. Recuperado de: https://www.who.int/violence_injury_prevention/violence/world_report/en/abstract_es.pdf

Organización Mundial de la Salud (2012) Violencia. Recuperado de: https://www.who.int/topics/violence/es/

Palma-Orozco, R., García-Leyva, E., & Ruiz-Ledesma, E. F. (2020). Aprendizaje significativo: El caso de la computación, la matemática y la música. Recuperado de: http://www.iiisci.org/journal/CV$/risci/pdfs/CB125BY20.pdf.

Parmar, D. N., & Mehta, B. B. (2014). Face recognition methods & applications. arXiv preprint arXiv:1403.0485. Recuperado de; https://arxiv.org/abs/1403.0485.

Precup, D. (2017) Introduction to Machine Learning. McGill University. Montreal. Recuperado de: https://escholarship.mcgill.ca/concern/theses/5q47rt28f

QuestionPro. (2018). Tipos de análisis de datos. Recuperado de: https://www.questionpro.com/es/analisis-de-datos.html

Ramírez Ticona, J. T. (2017). Propuesta de un modelo para el reconocimiento de escenas violentas en video. Recuperado de: http://repositorio.unsa.edu.pe/bitstream/handle/UNSA/4507/ISratijt.pdf?sequence=1&isAllowed=y

Ramzan, Abid, Ullah, Mahmood, Ismail, Ahmed, Ilyas y Mahmood, (2017 "A Review on state-of-the-art Violence Detection Techniques" Recuperado de: https://www.researchgate.net/publication/334778564_A_Review_on_state-of-the-art_Violence_Detection_Techniques

Real Academia Española (2020) Escenario. Recuperado de: https://dle.rae.es/escenario

Real Academia Española (2019) Prototipo. Recuperado de: https://dle.rae.es/prototipo

Rincón Chaparro, M. Á. (2017). Propuesta de una Metodología de Programación de Operaciones Detallada para Órdenes de Trabajo en Operaciones Secuenciales a Través de Algoritmos Heurísticos Basados en Lógica Proposicional Articulados Sobre una Serie de Autómatas Programables.

.

Salvador, S (2020) Cuáles son los distritos con más robos en Lima. Concepto. Recuperado de: https://elconsejosalvador.com/cuales-son-los-distritos-con-mas-robos-en-lima/

Sánchez Morales (s.f). Máquinas de aprendizaje extremo multicapa: Estudio y evaluación de resultados en la segmentación automática de carótidas en imágenes ecográficas (Proyecto). Universidad Politécnica de Cartagena

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).

Scikit-Learn, herramienta básica para el Data Science en Python. (2019). Máster en Data Science. Recuperado de: https://www.master-data-scientist.com/scikit-learn-data-science/

Silva Guzmán. (2020). Entrenamiento de la Red Neuronal Convolucional YOLO para objetos propios. Recuperado de: https://biorobotics.fi-p.unam.mx/wp-content/uploads/Courses/reconocimiento_de_patrones/tutoriales/YOLO-Introducci%C3%B3n-e-implementaci%C3%B3n-.pdf

Scikit-Learn, herramienta básica para el Data Science en Python. (2019). Máster en Data Science. Recuperado de: https://www.master-data-scientist.com/scikit-learn-data-science/

Statistical Analysis System (2016 . The difference between artificial intelligence and machine learning. Recuperado de: https://www.sas.com/en_us/insights/analytics/machine-learning.html

Sucar, L., & Gómez, G. (2015). Visión Computacional. Puebla - México. https://www.sas.com/es_pe/insights/analytics/machine-learning.html

Sucar, L., & Gómez, G. (2015). Visión Computacional. Puebla - México. Recuperado de: https://www.sas.com/es_pe/insights/analytics/machine-learning.html

Tensorflow.org. (2019) Por qué TensorFlow - Compilación sencilla de modelos. Recuperado de: https://www.tensorflow.org

Torres, C (2005) Jóvenes y Violencia. Recuperado de: https://rieoei.org/historico/documentos/rie37a03.html

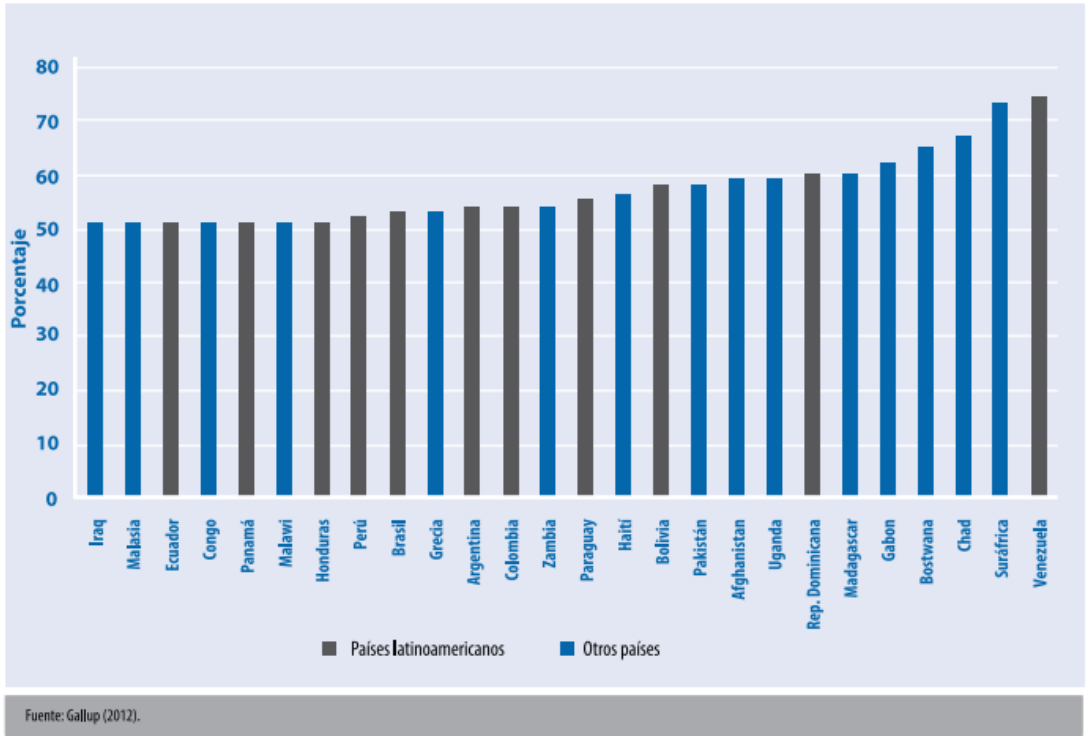
Tutor de programación (2017) Face Landmarks Detector con Dlib y OpenCV. Recuperado de: http://acodigo.blogspot.com/2017/11/face-landmarksdetector-con-dlib-y.html. [Accedido: 25- abr-2019]

Villalón De La Vega, D. E. (2012). Diseño e implementación de una plataforma de software para reconocimiento facial en video.

ÍNDICE DE ANEXOS

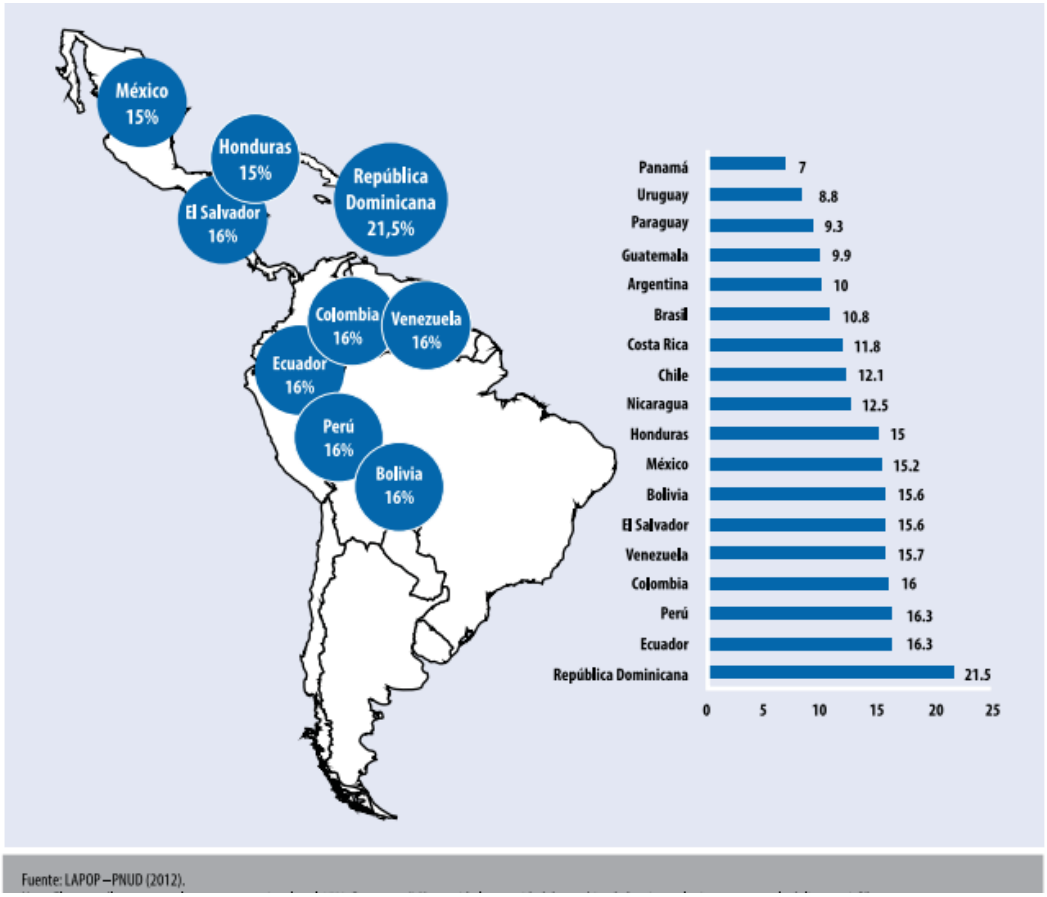
Pág.

ANEXO

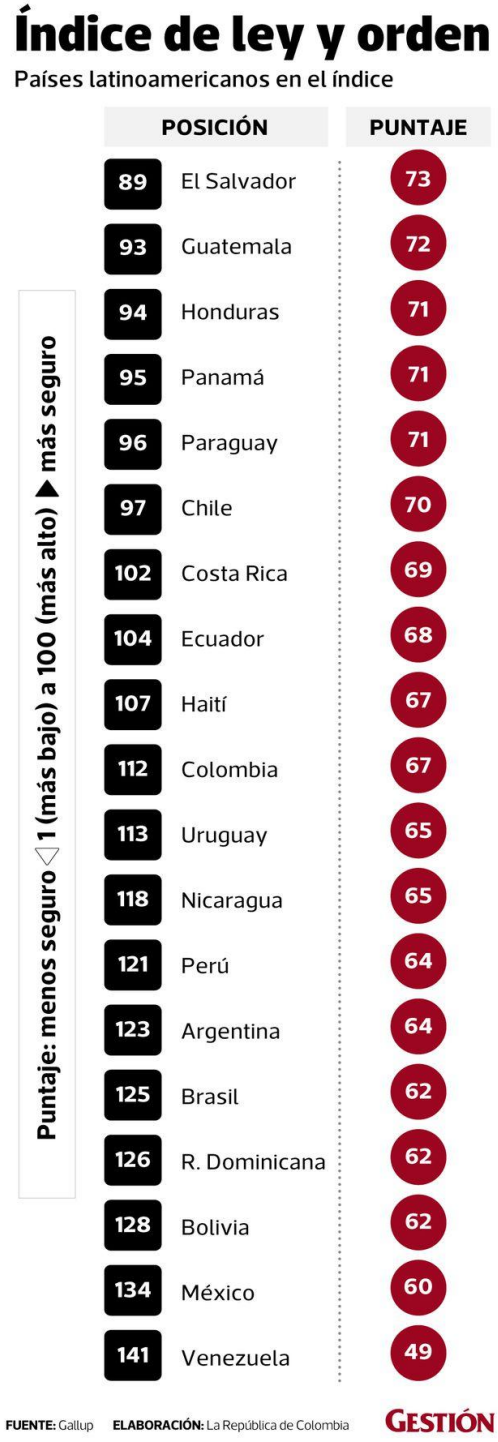


Fuente: Gallup (2012).

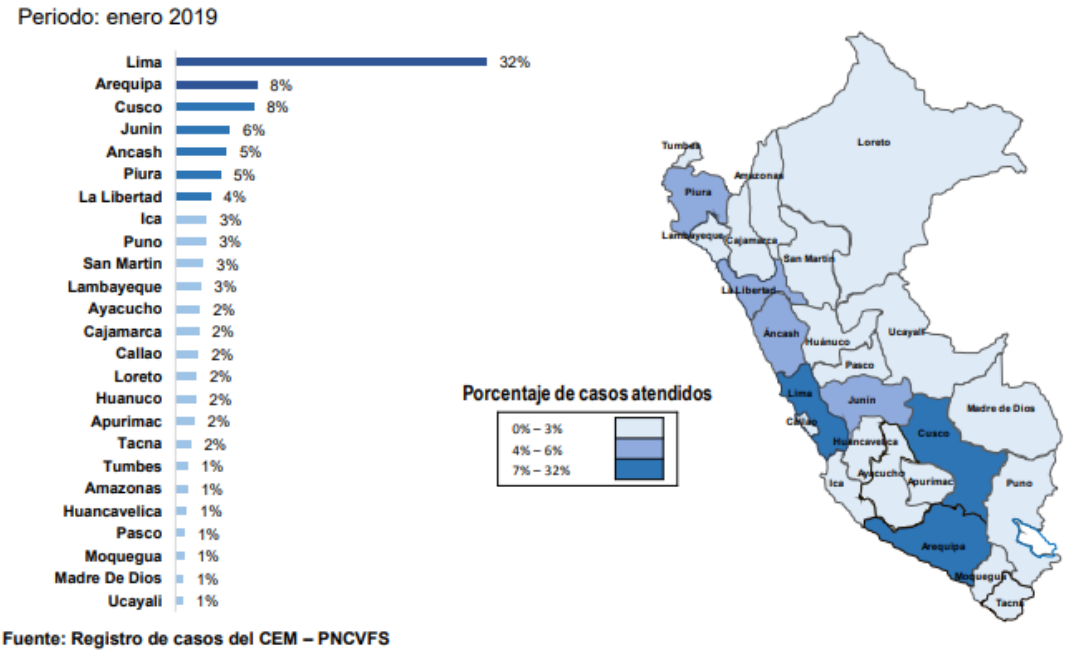
Anexo 1 Porcentajes mayor a 50% de respuestas afirmativas sobre la inseguridad al caminar solos en la noche



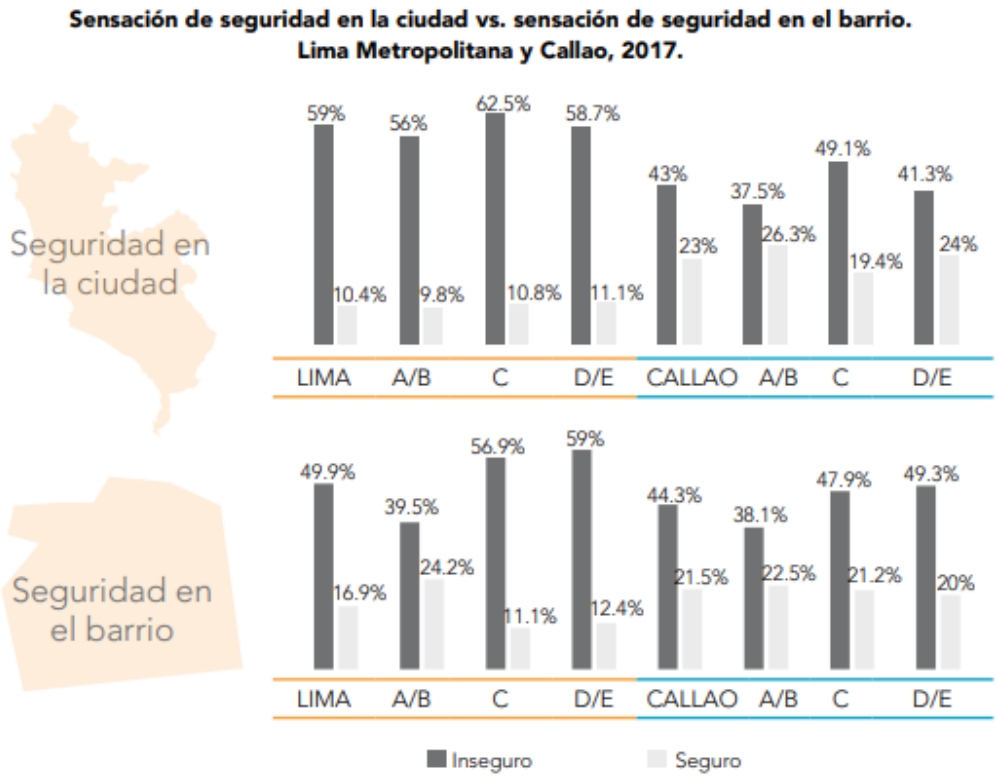
Anexo 2 Porcentaje de personas que han sentido la necesidad de cambiarse de lugar por la inseguridad



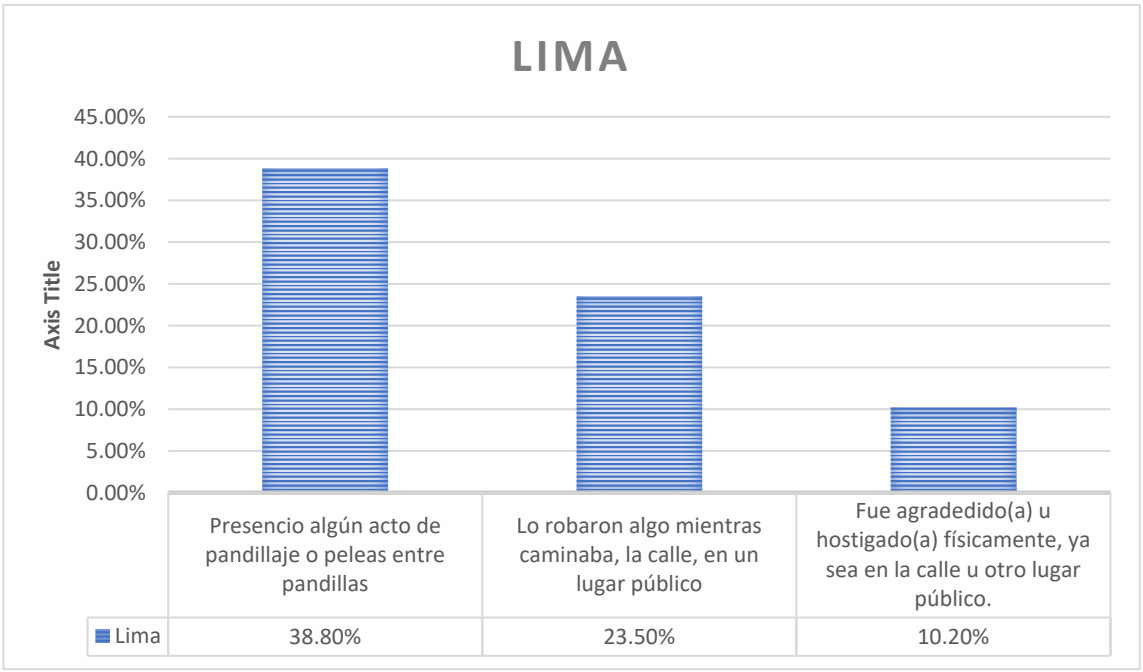
Anexo 3 Países con mayor índice de inseguridad



Anexo 4 Porcentaje de Violencia por departamento



Anexo 5 Porcentaje de percepción de seguridad ciudadana en Lima



Anexo 6 Porcentaje de hechos delictivos en la ciudadana de Lima

Violencia Física en El Agustino

Encuesta realizada para tesis de titulación relacionada a la violencia física en el distrito del Agustino

*Obligatorio

¿Cuál es tu edad? *

Tu respuesta

Alguna vez has sufrido de violencia física en la vía pública del distrito El Agustino? *

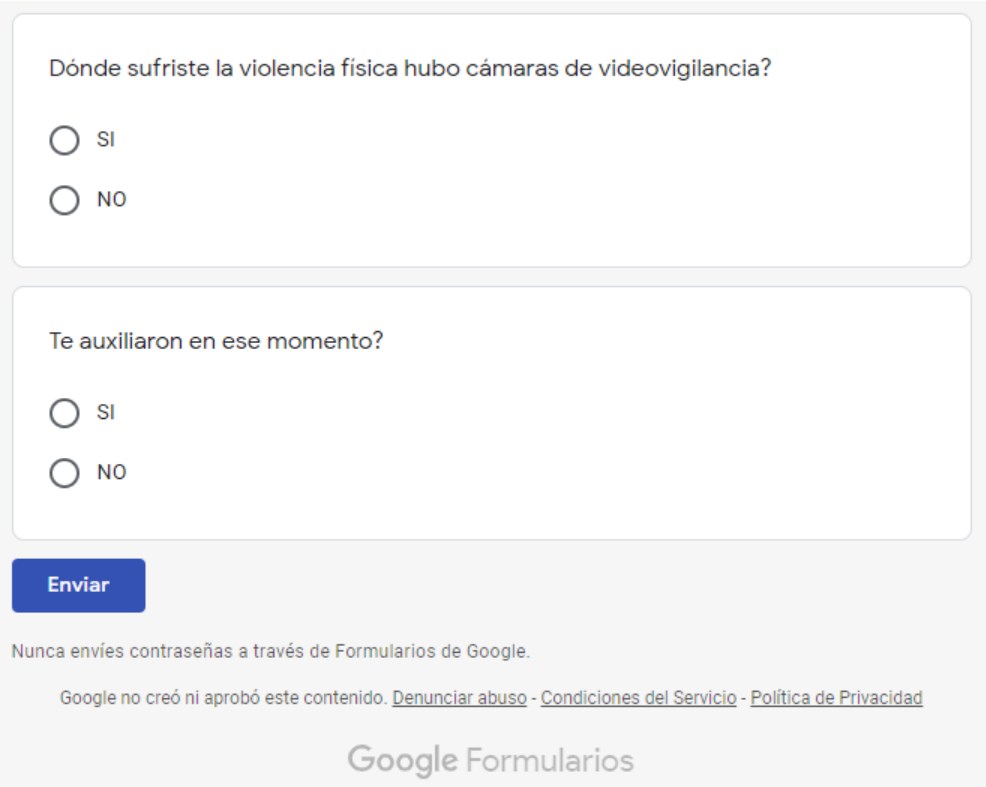
- ◯ SI
- ◯ NO

En caso sea sí, en qué momento sufriste la violencia?

- ◯ En un intento de robo
- ◯ Por parte de tu pareja y/o familiar
- ◯ Por parte de las autoridades
- ◯ Otros: _____

Cómo fue la violencia física?

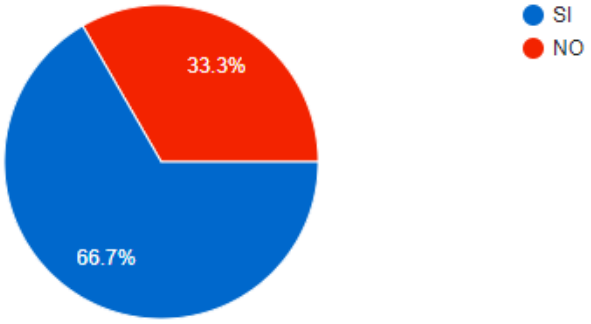
- ◯ Con puñetes y/o patadas
- ◯ Con algún objeto (azote, vara, cinturón, arma, etc)
- ◯ Otro tipo de golpes (cachetada, empujar, sacudir, etc)
- ◯ Otros: _____



Anexo 7 Preguntas realizas en el cuestionario

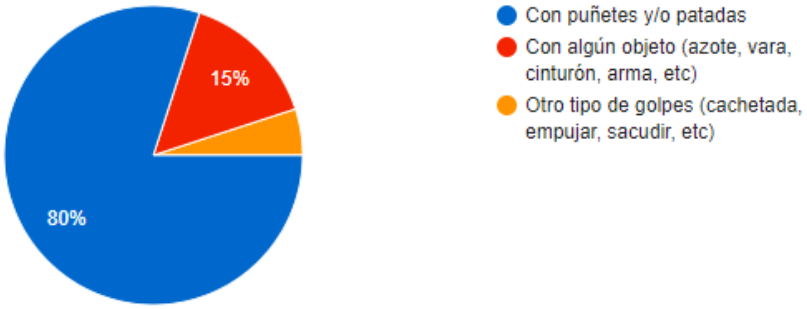
Alguna vez has sufrido de violencia física en la vía pública del distrito El Agustino?

124 respuestas



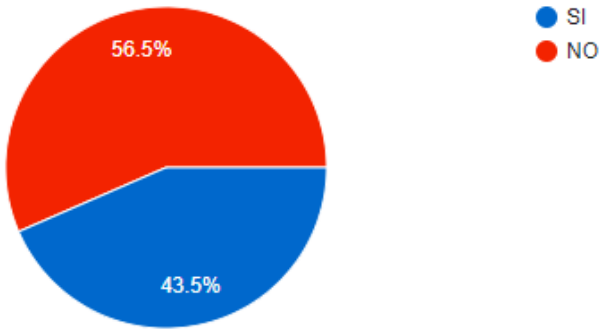
Cómo fue la violencia física?

80 respuestas

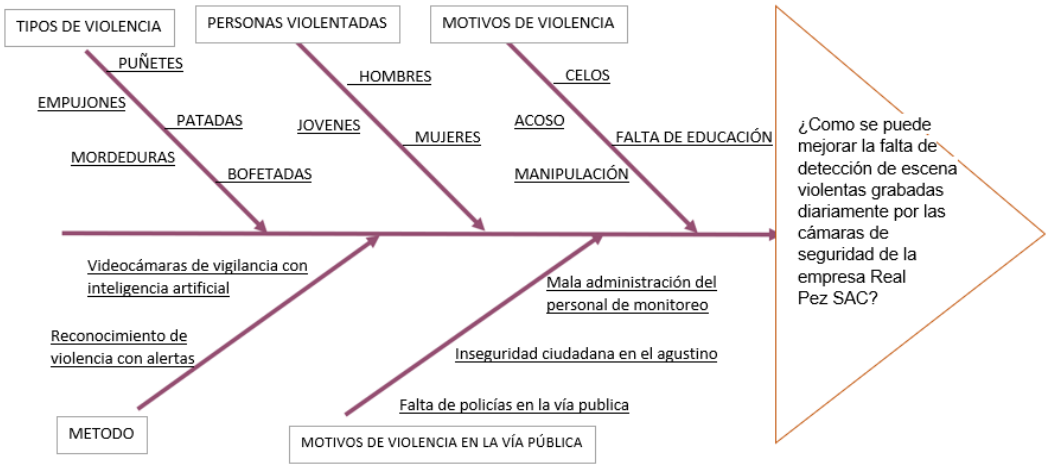


Dónde sufriste la violencia física hubo cámaras de videovigilancia?

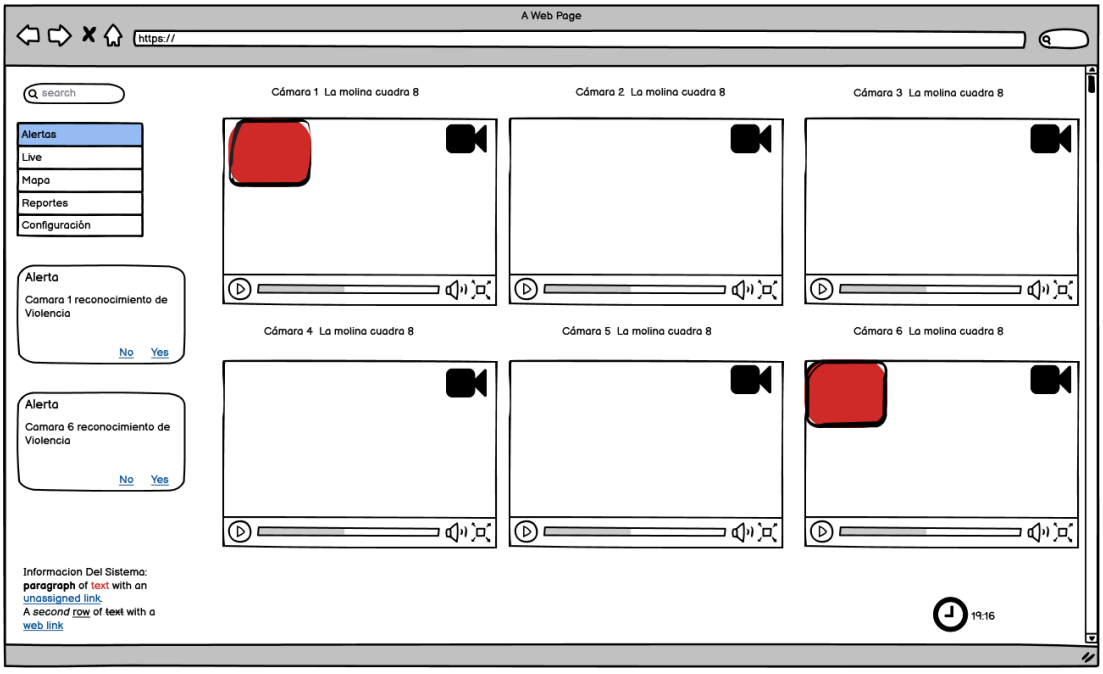
80 respuestas



Anexo 8 Respuestas principales del cuestionario

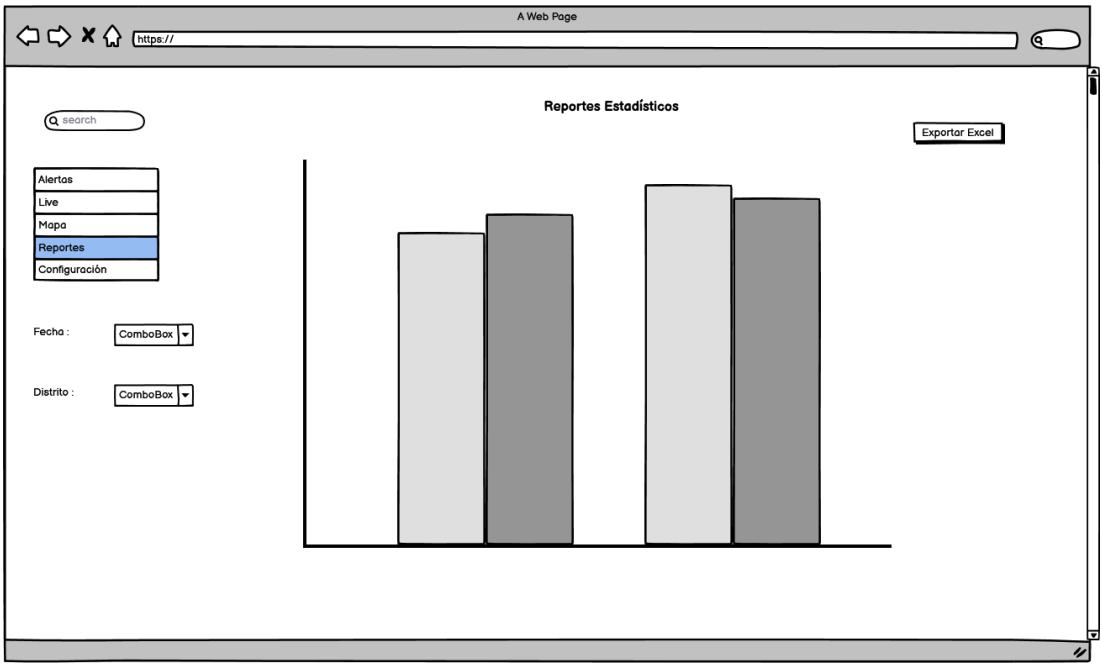


Anexo 9 Diagrama de causa y efecto del problema

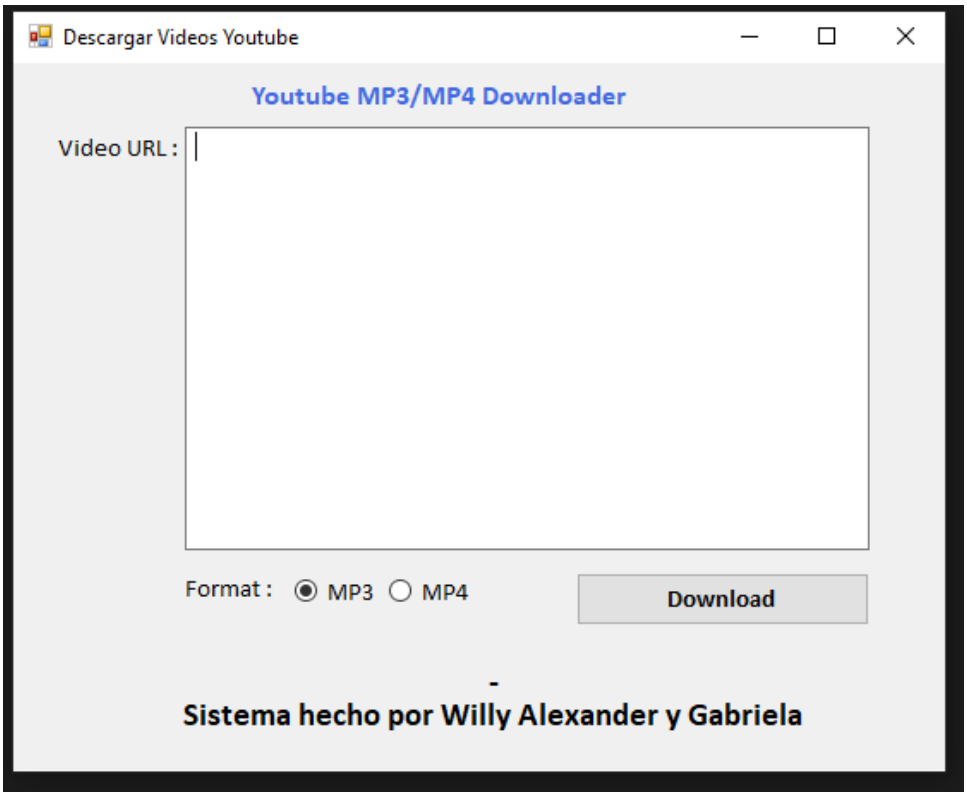


Anexo 10 Prototipo Alertas y Live

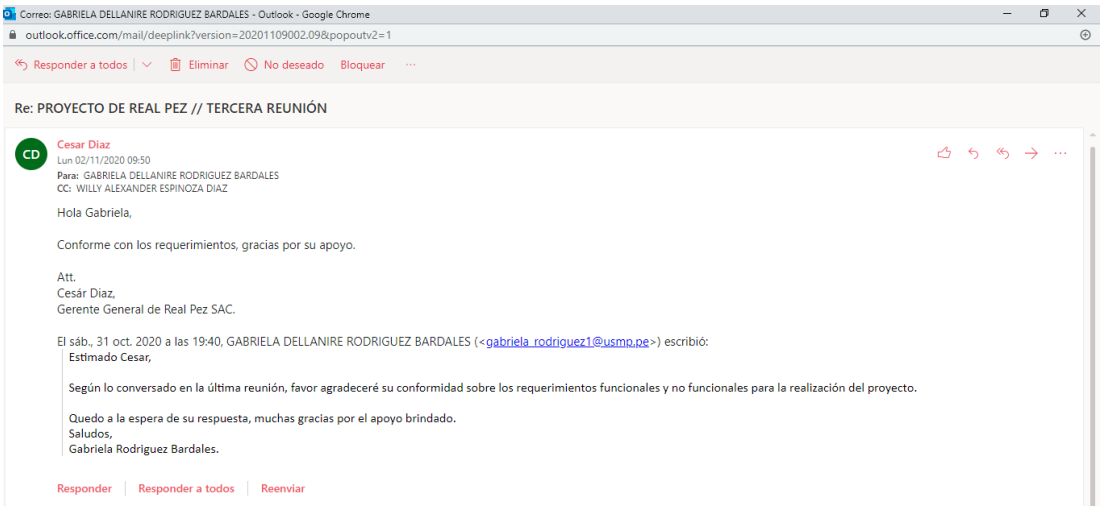
Anexo 11 Prototipo Alertas y Live



Anexo 12 Prototipo Reporte Estadístico



Anexo 13 Batch Descargar Videos Youtube

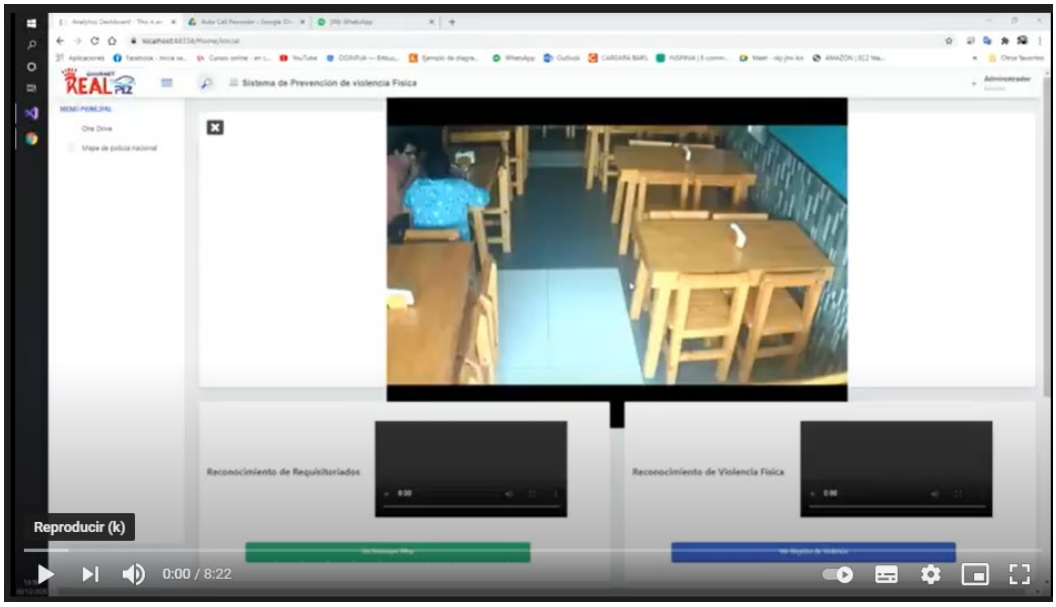


Anexo 14 Conformidad de la reunión

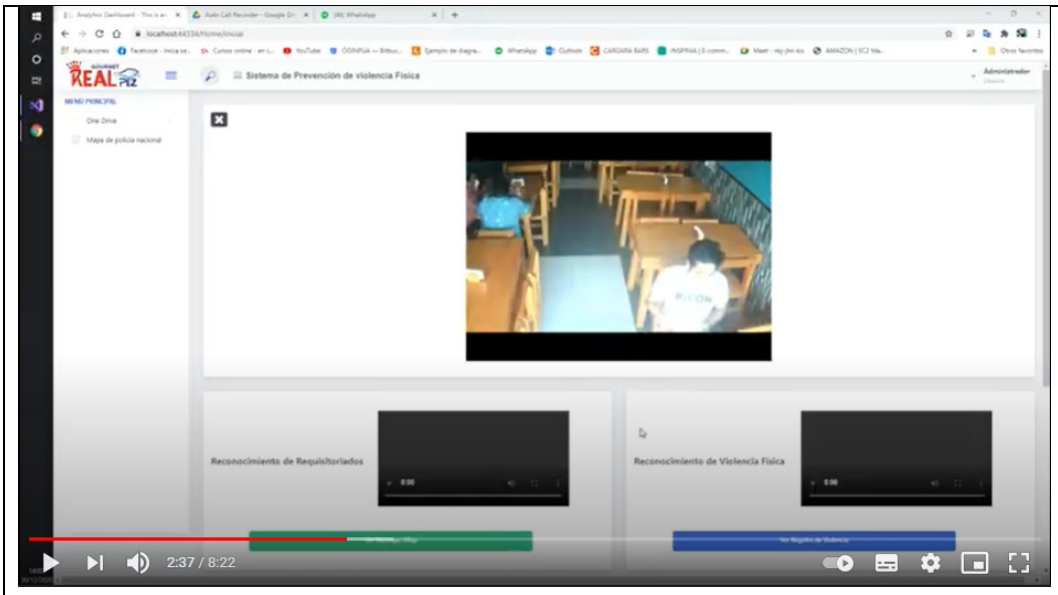
RESULTADOS DEL MODELO DE MACHINE LEARNING:

“NO DETECTO PERSONA CON REQUISITORIA”

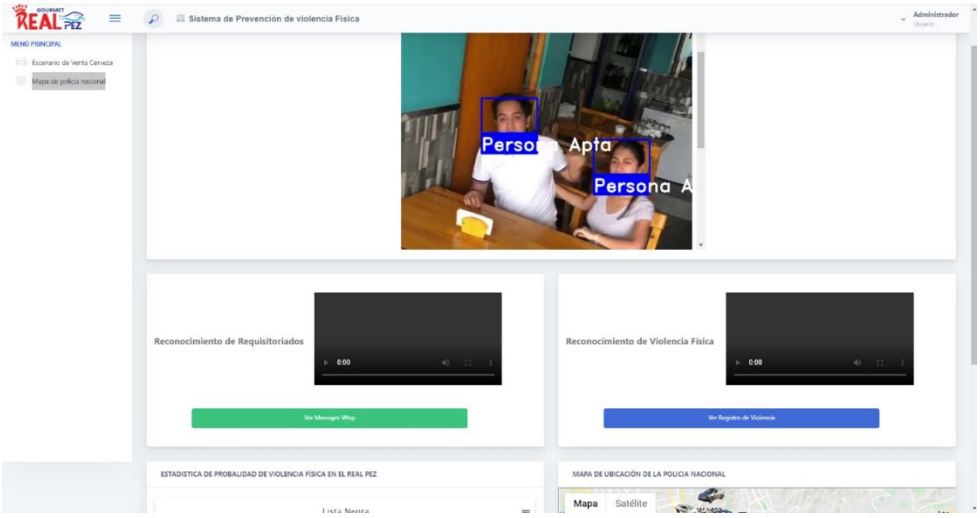
1. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



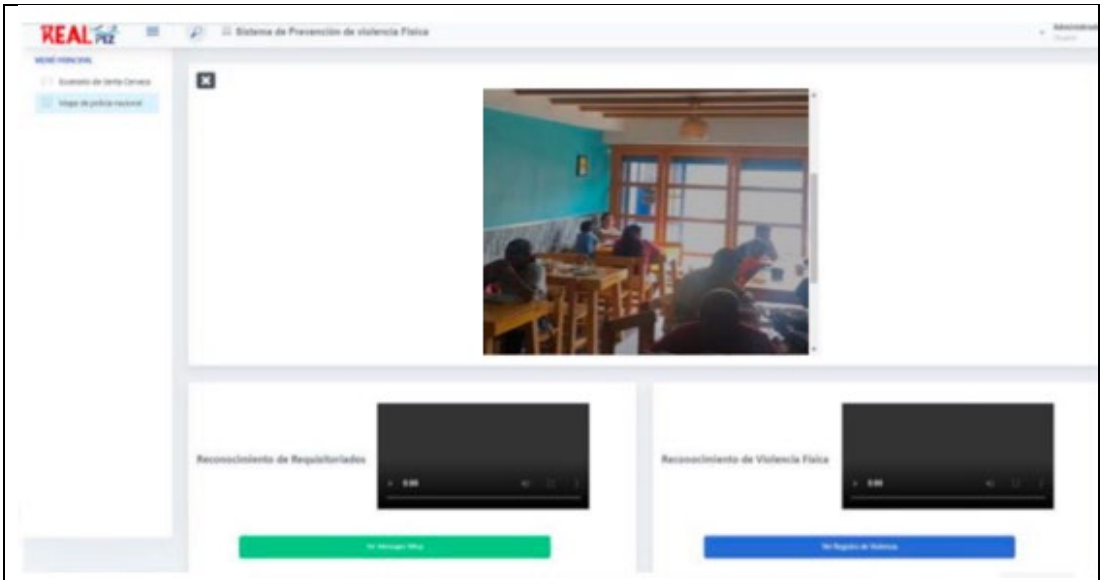
2. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



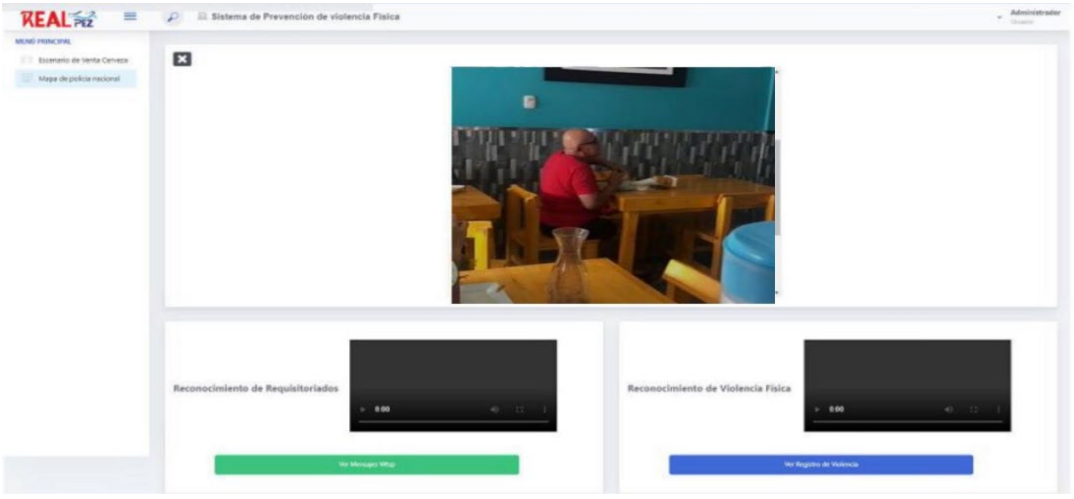
3. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



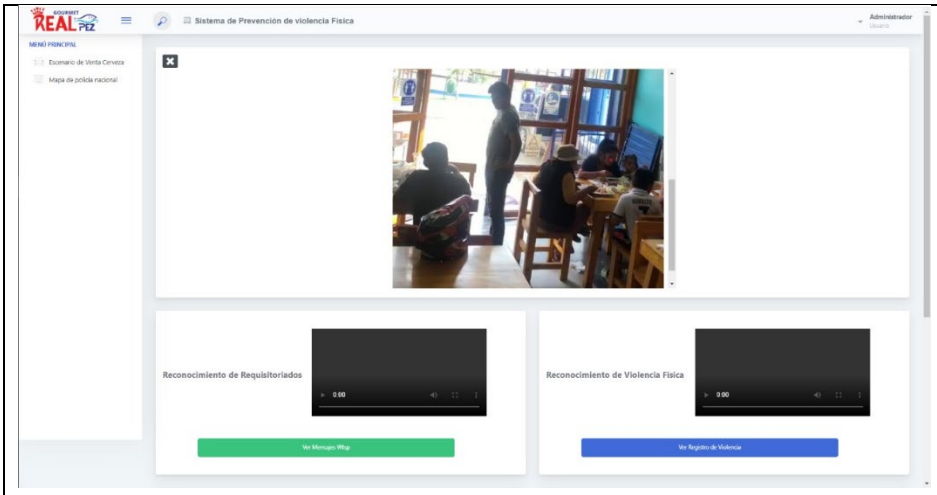
4. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



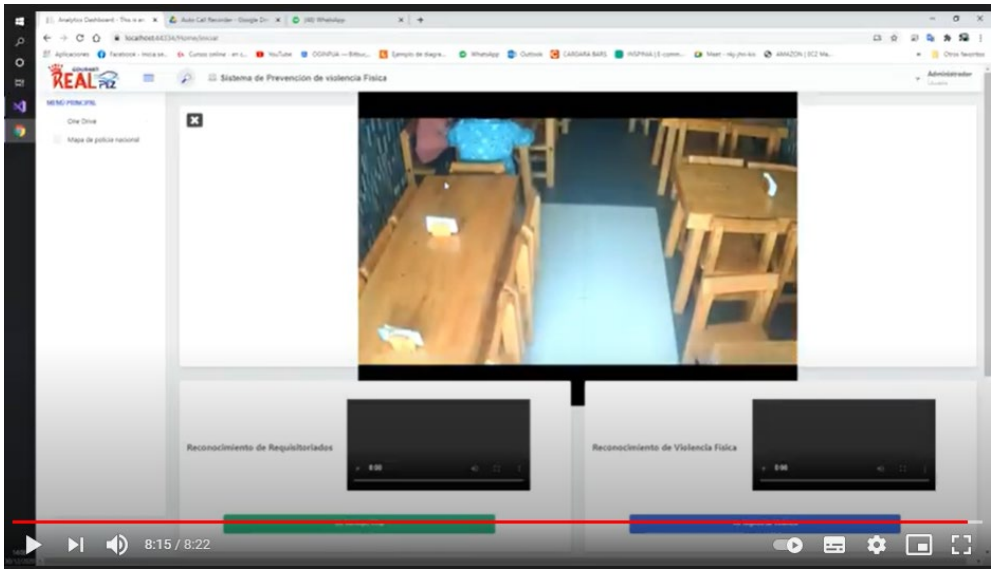
5. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



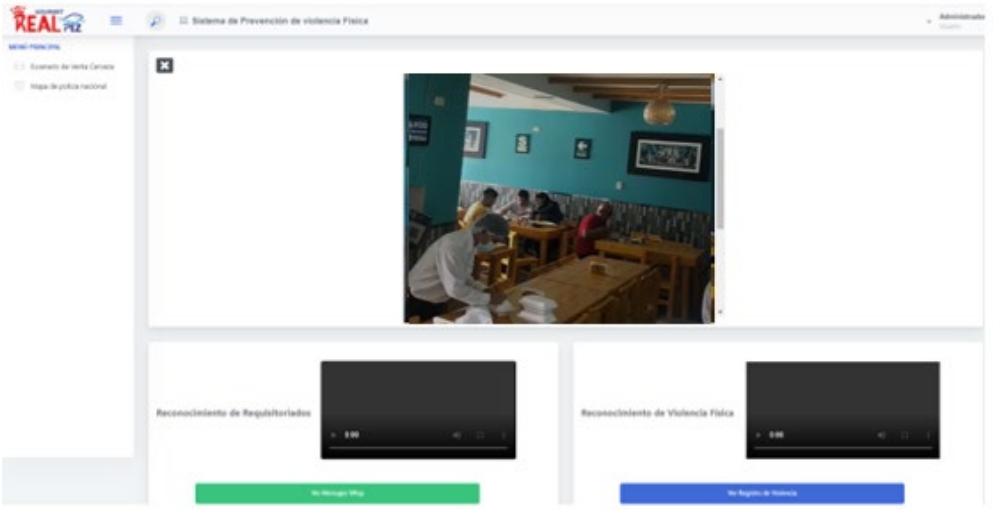
9. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



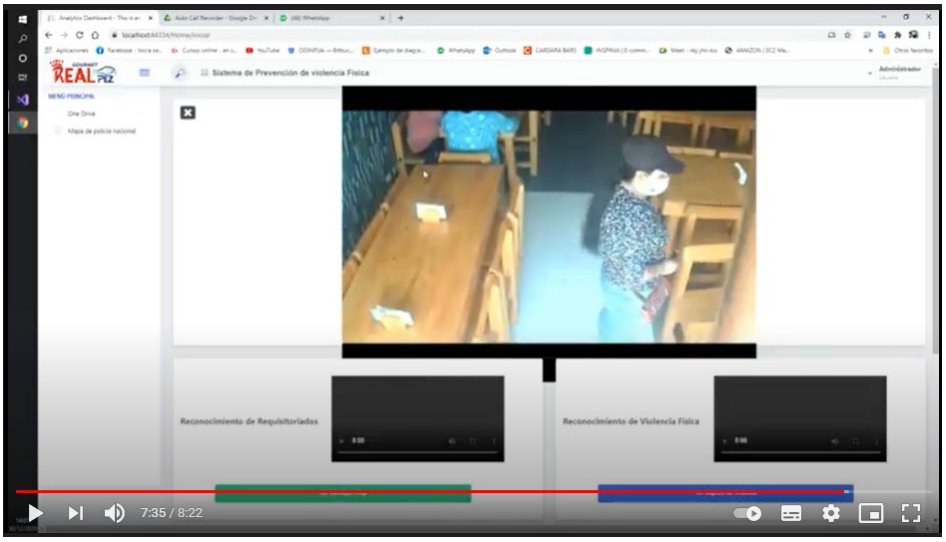
10. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



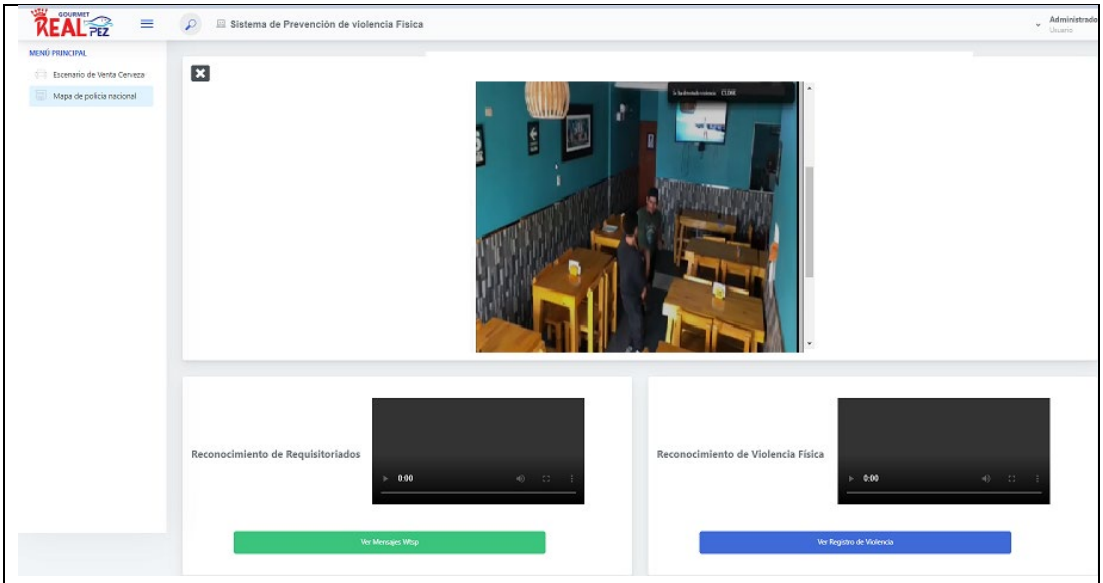
11. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



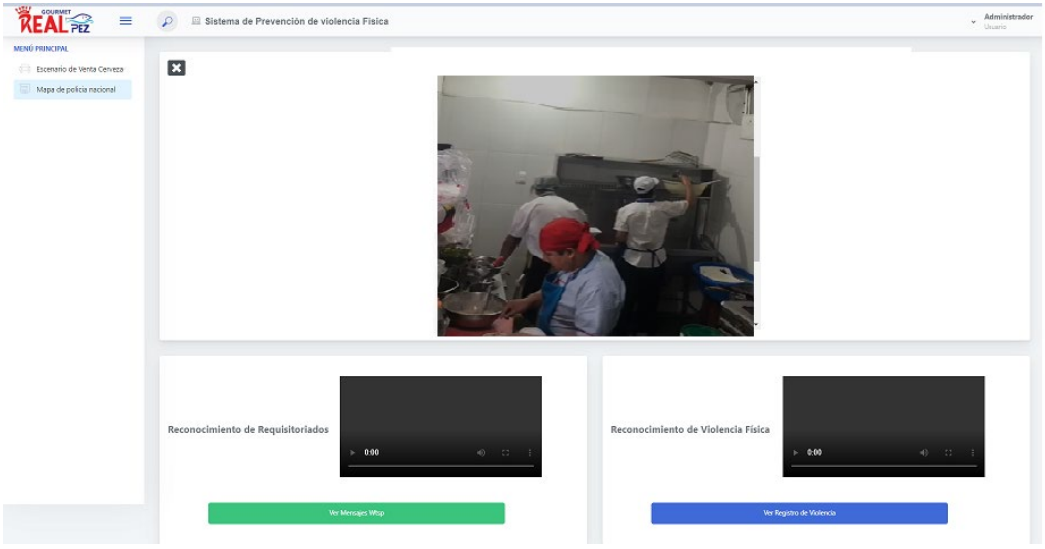
12. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



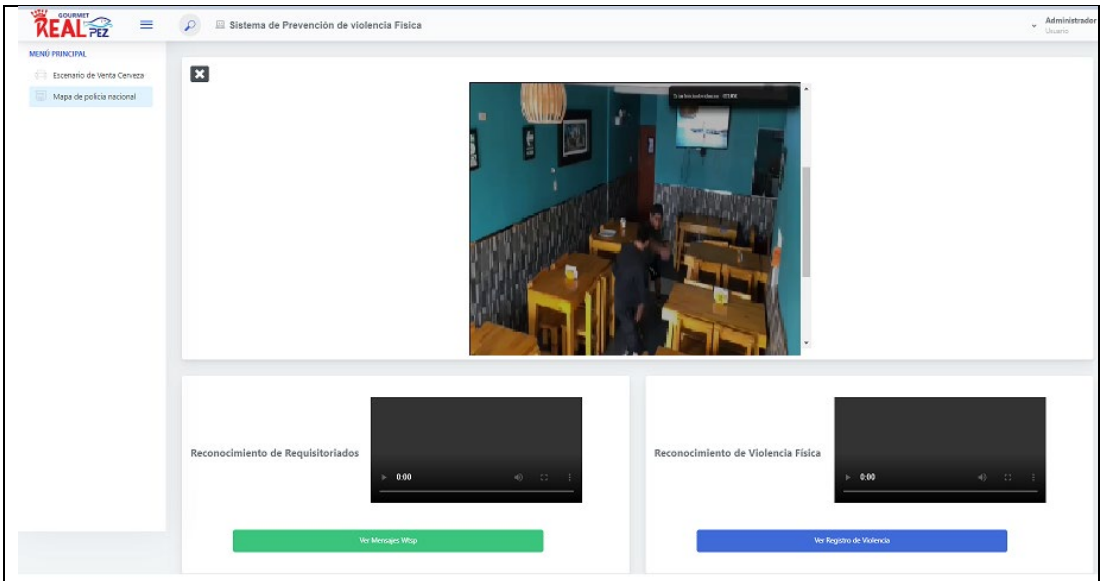
14. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



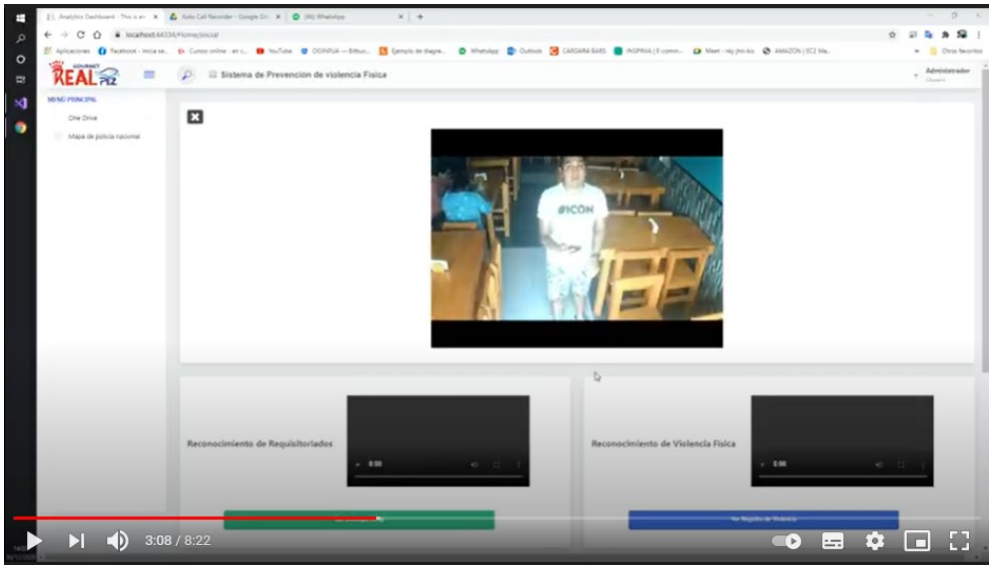
15. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



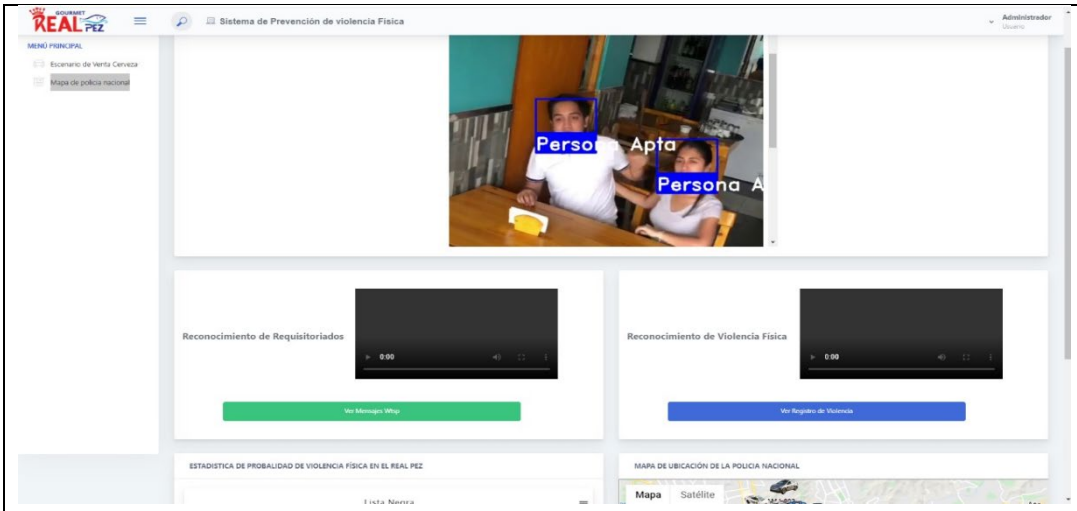
17. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



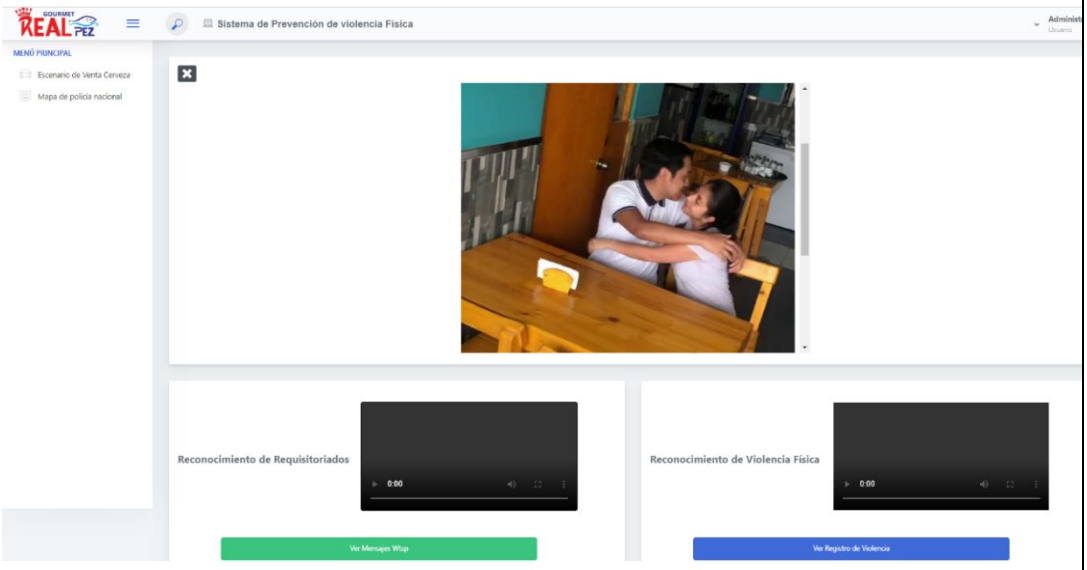
18. EXITOSO NO DETECTO PERSONA CON REQUISITORIA



19. EXITOSO NO DETECTO PERSONA CON REQUISITORIA

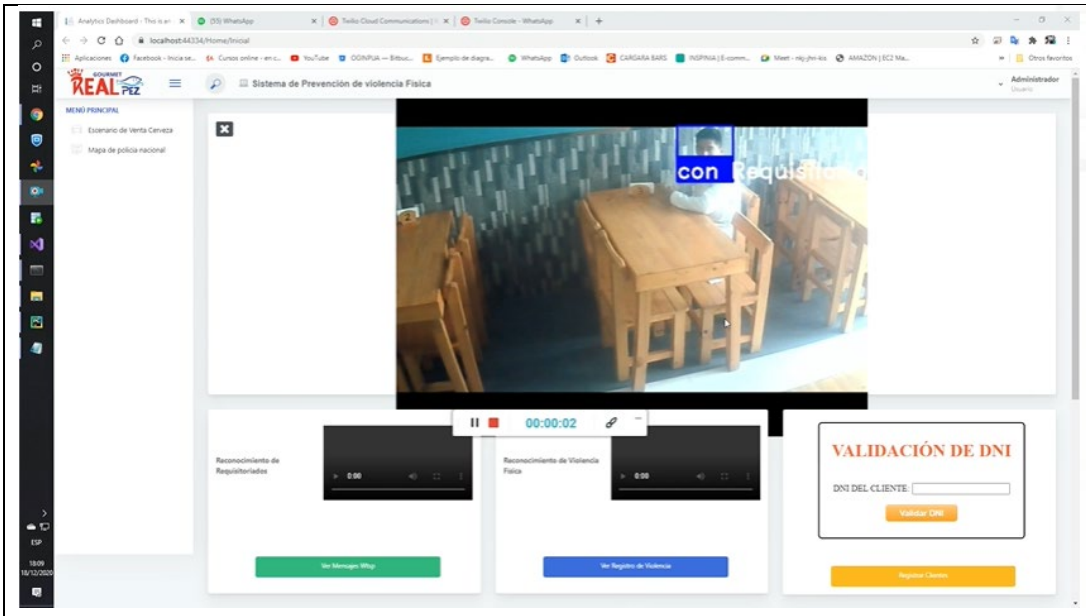


20. EXITOSO NO DETECTO PERSONA CON REQUISITORIA

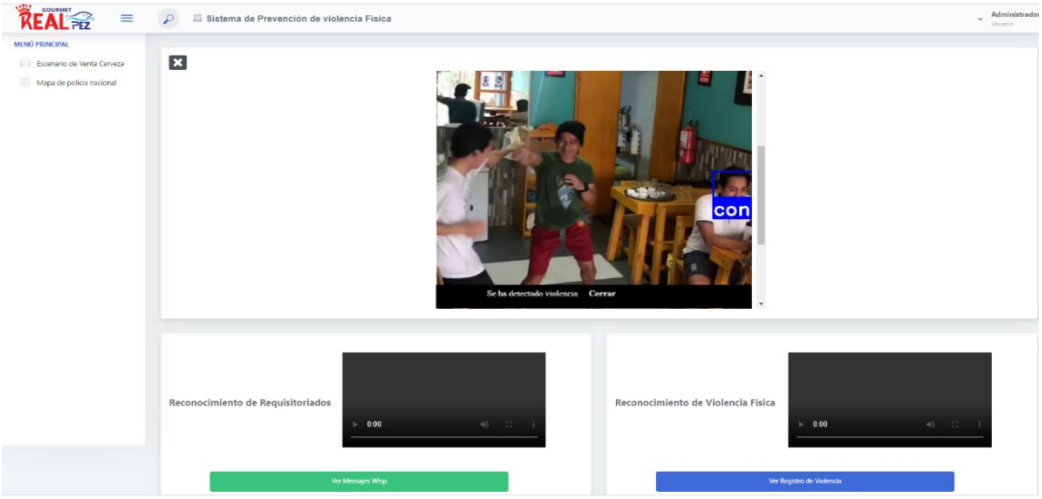


RESULTADOS DE ESCENARIOS: "SI DETECTO PERSONA CON REQUISITORIA"

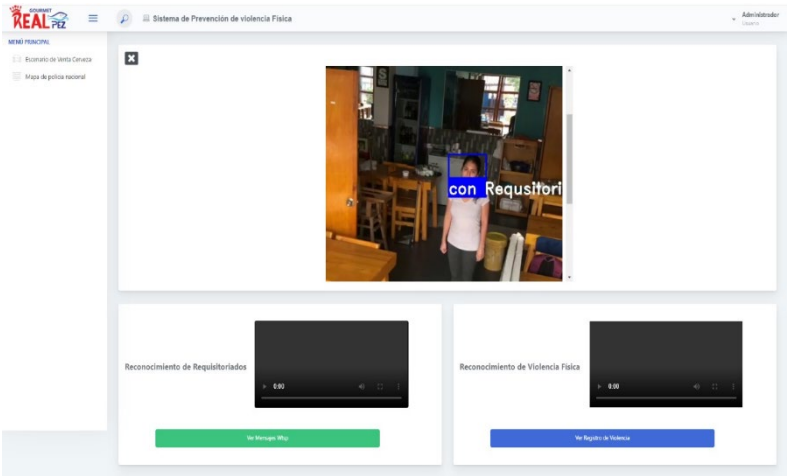
21. EXITOSO SI DETECTO PERSONA CON REQUISITORIA



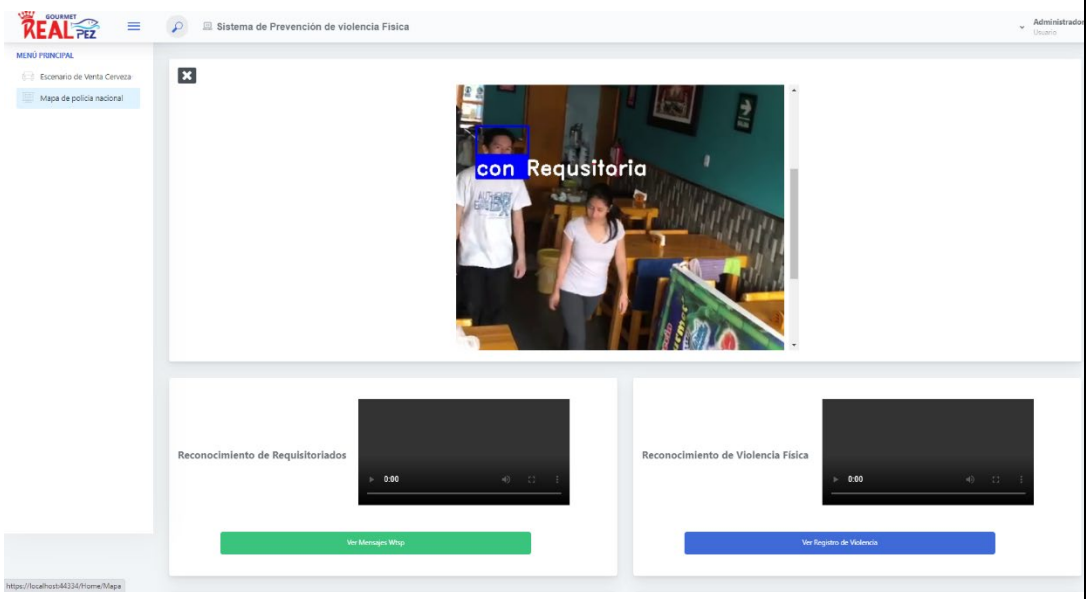
22. EXITOSO SI DETECTO PERSONA CON REQUISITORIA



23. EXITOSO SI DETECTO PERSONA CON REQUISITORIA

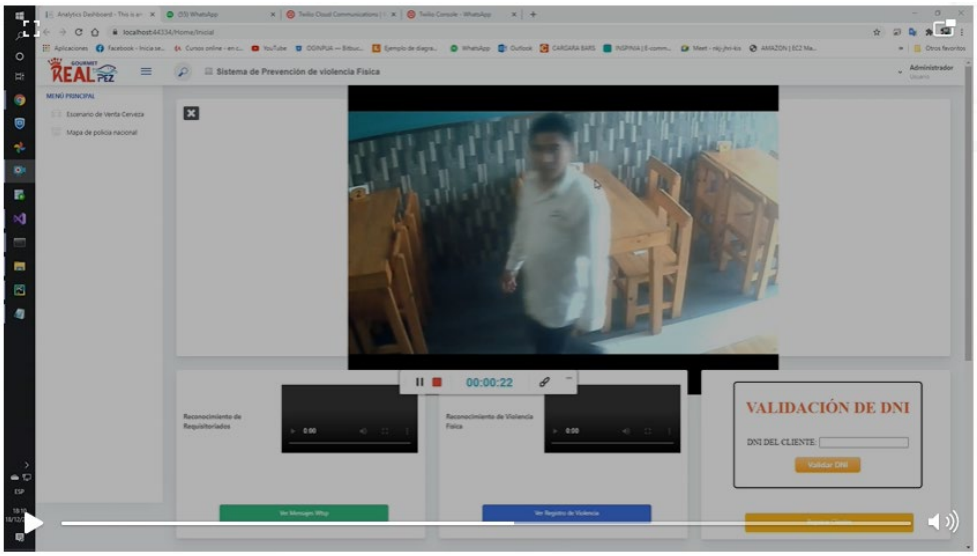


24. EXITOSO SI DETECTO PERSONA CON REQUISITORIA

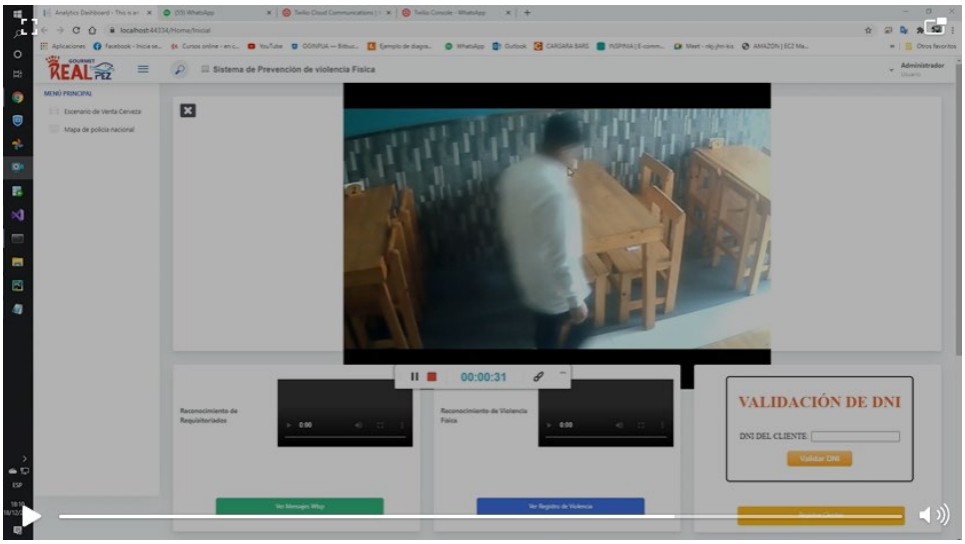


RESULTADOS DE ESCENARIOS: "NO DETECTO PERSONA CON REQUISITORIA Y SI ESTABA EN LA BASE"

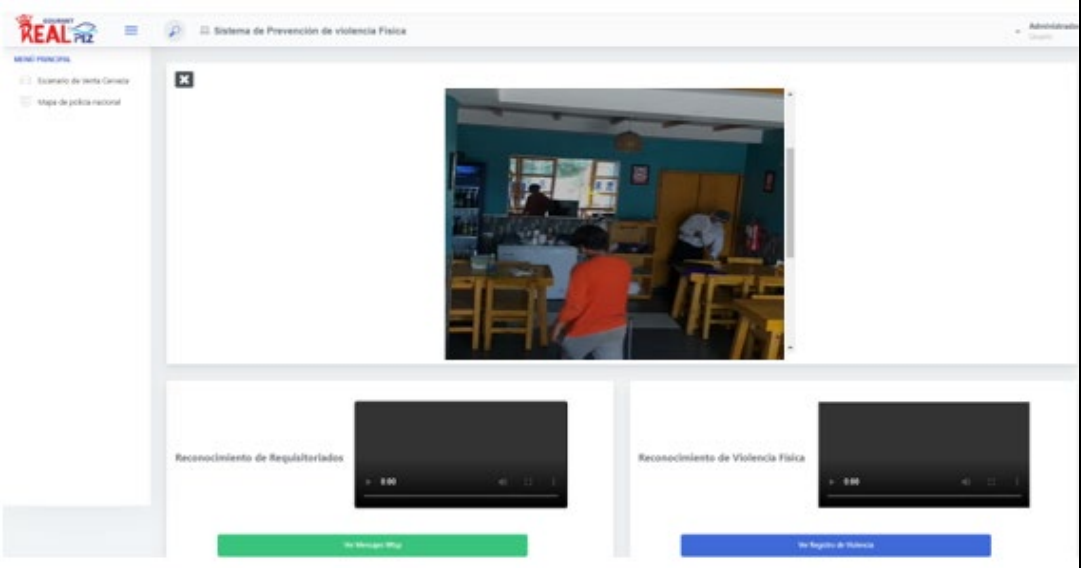
25. FALLO NO DETECTO PERSONA CON REQUISITORIA CUANDO SI ESTABA EN LA BASE



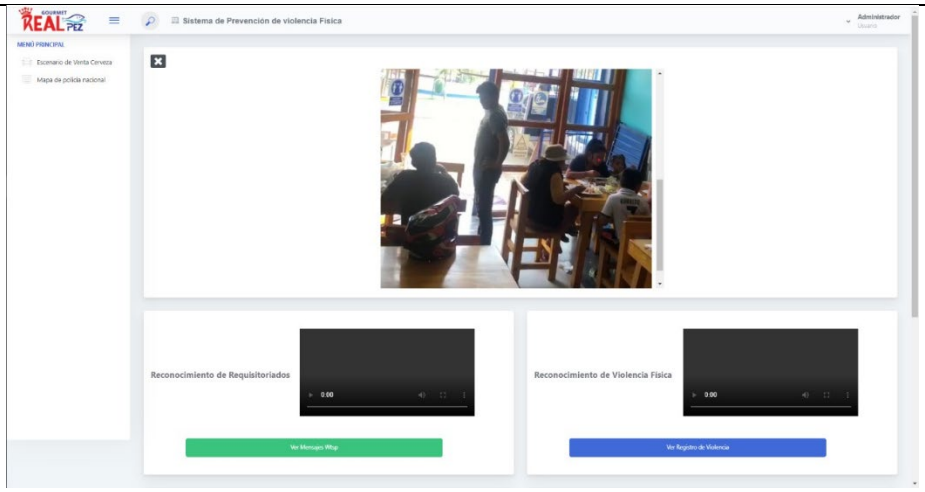
26. FALLO NO DETECTO PERSONA CON REQUISITORIA CUANDO SI ESTABA EN LA BASE



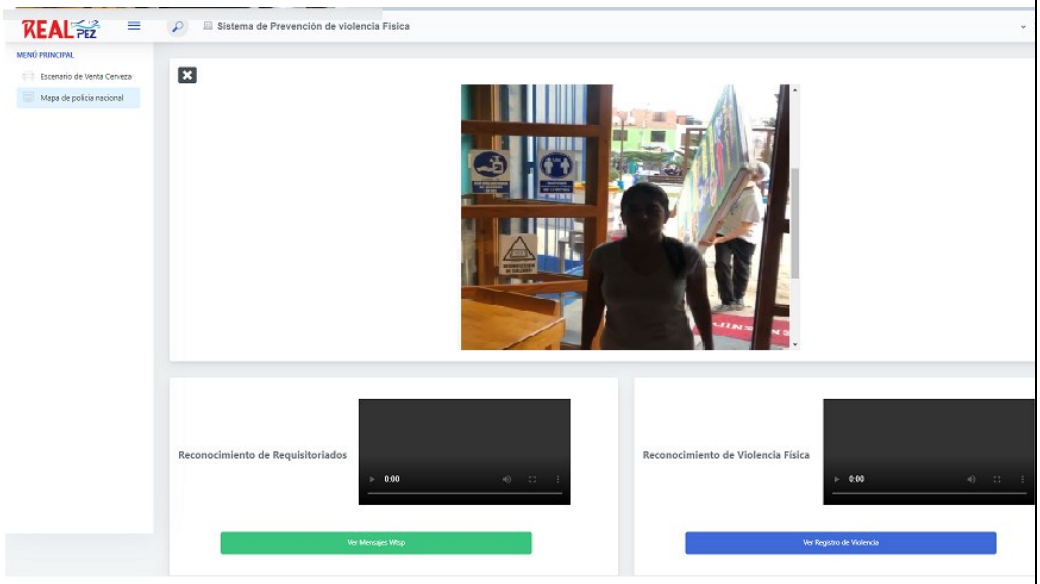
27. FALLO NO DETECTO PERSONA CON REQUISITORIA CUANDO SI ESTABA EN LA BASE



28. FALLO NO DETECTO PERSONA CON REQUISITORIA CUANDO SI ESTABA EN LA BASE



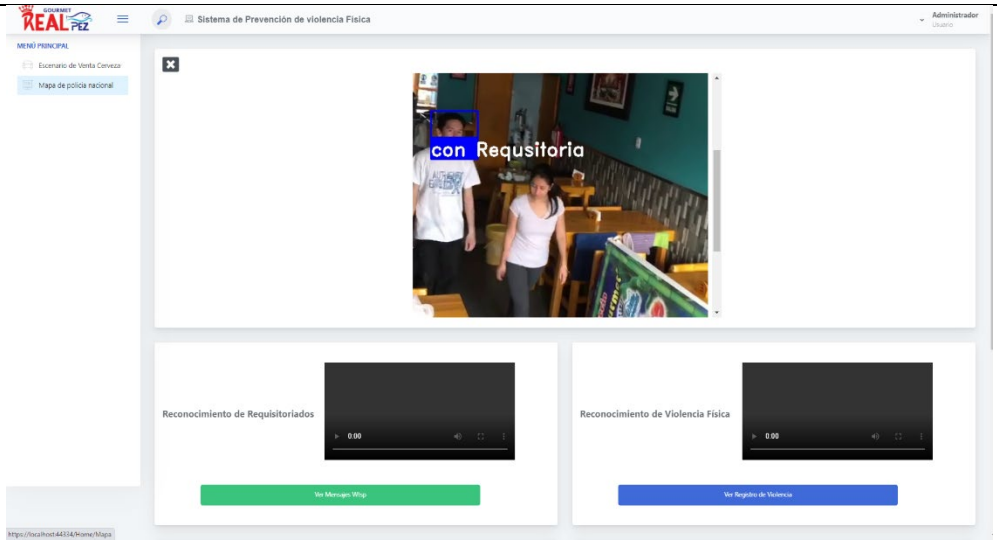
29. FALLO NO DETECTO PERSONA CON REQUISITORIA CUANDO SI ESTABA EN LA BASE



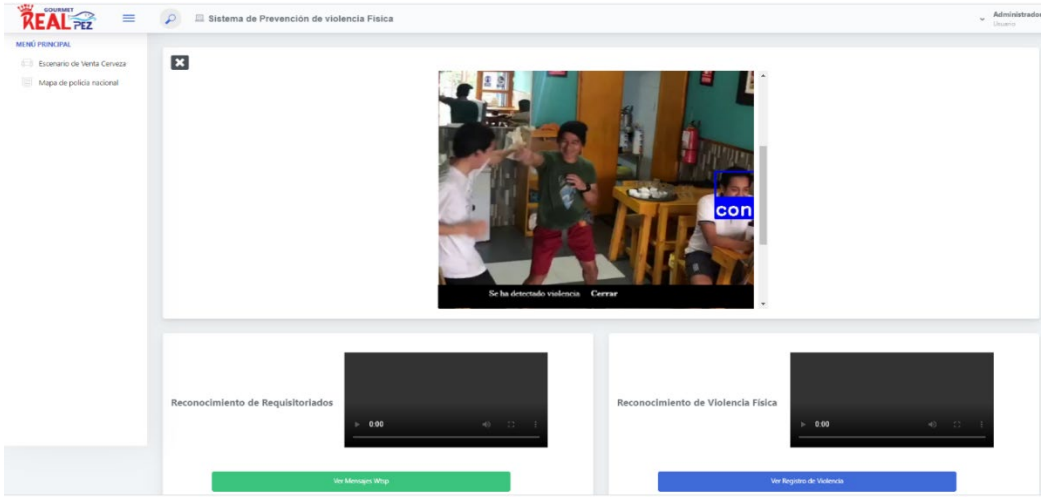
RESULTADOS DEL MODELO DE MACHINE LEARNING:

“DETECTO PERSONA CON REQUISITORIA Y NO ESTABA EN LA BASE”

30. Dos personas caminando, se observa que identificó a una persona con requisitoria cuando no se encontraba en nuestra base de requisitoria.



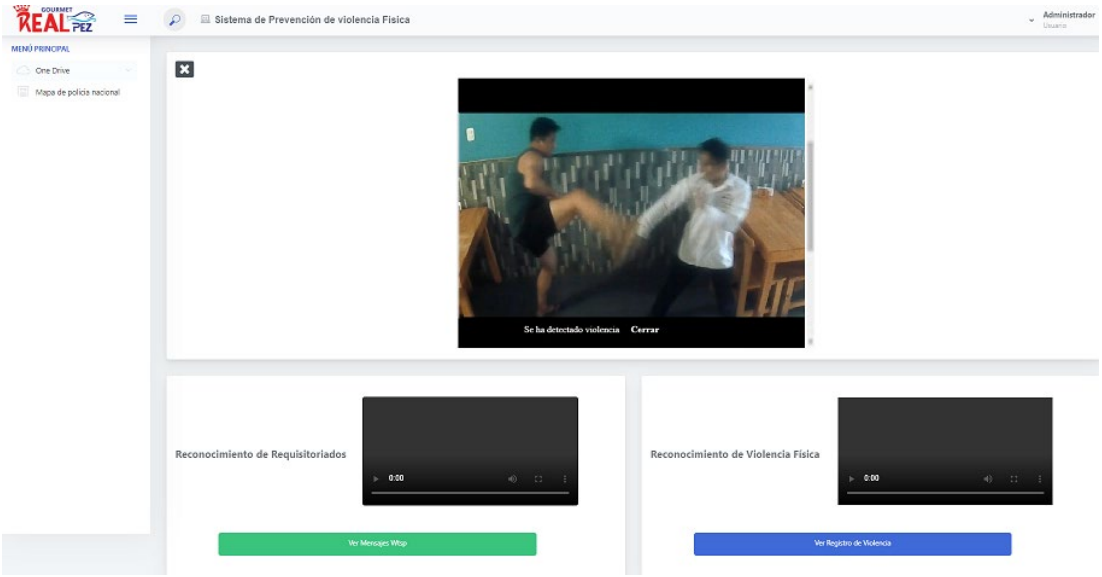
31. Resultado de escenarios donde se percibe más de dos personas, se observa que si se identifica a una persona con requisitoria cuando no estaba en nuestra base.



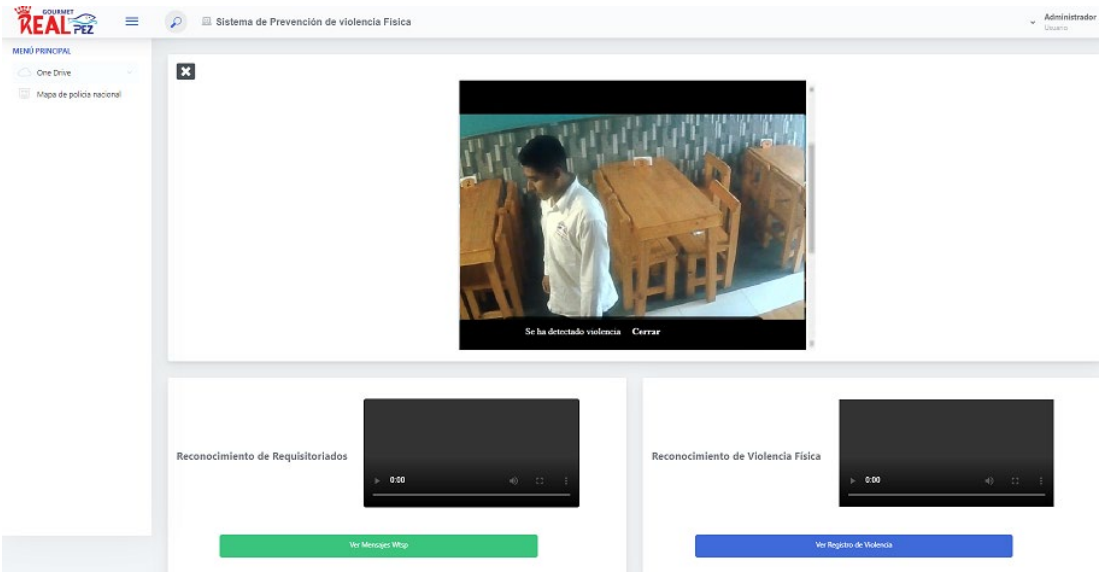
Anexo 17: Resultados de los 40 escenarios del primer modelo

RESULTADOS DE ESCENARIOS: “DETECCIÓN VIOLENCIA FÍSICA”

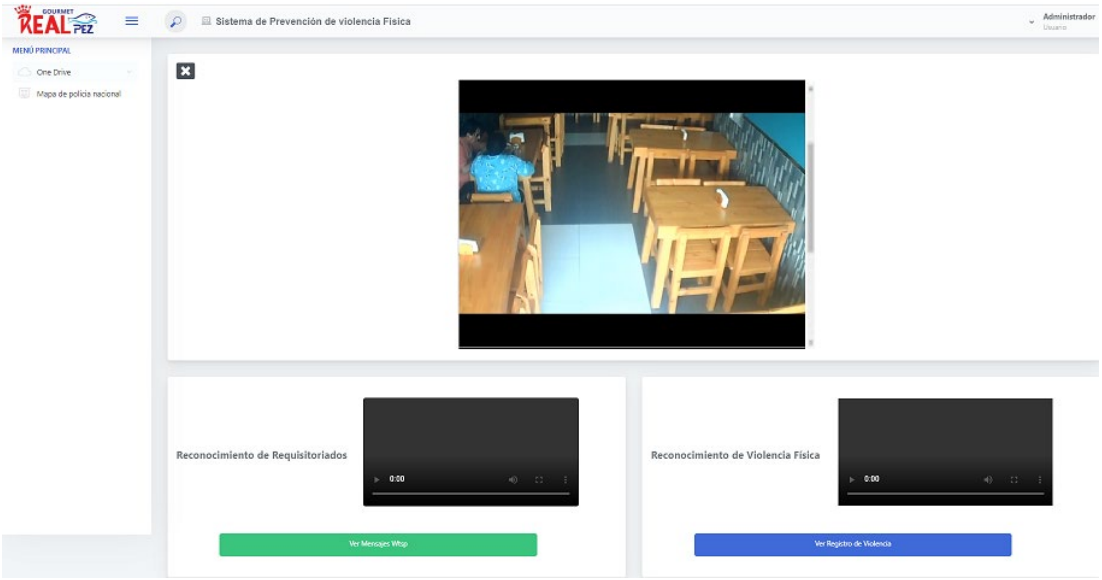
1.- EXITOSO NO DETECTO VIOLENCIA



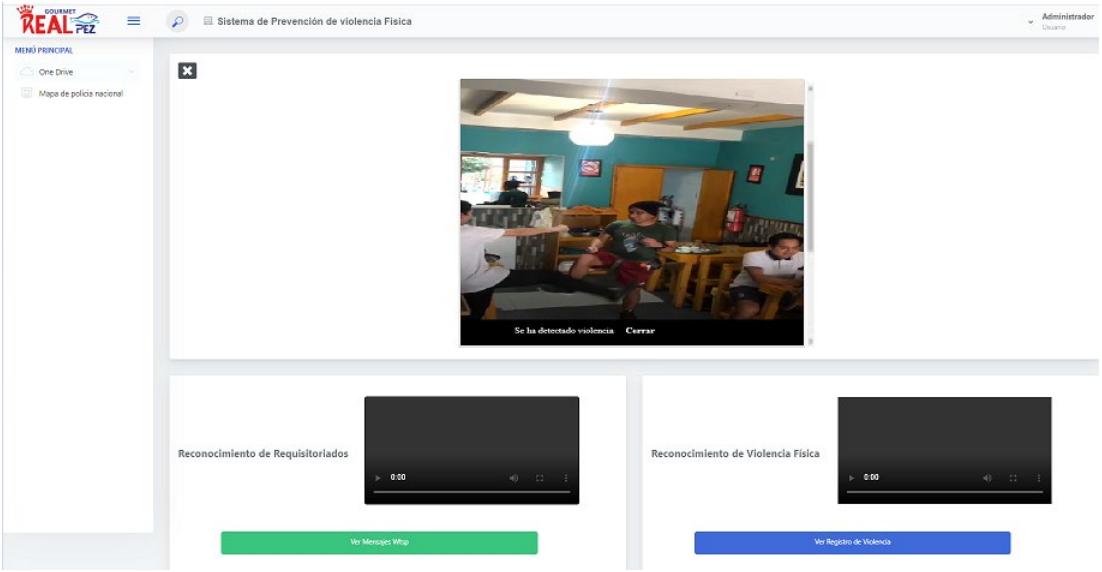
2.- EXITOSO DETECTO VIOLENCIA



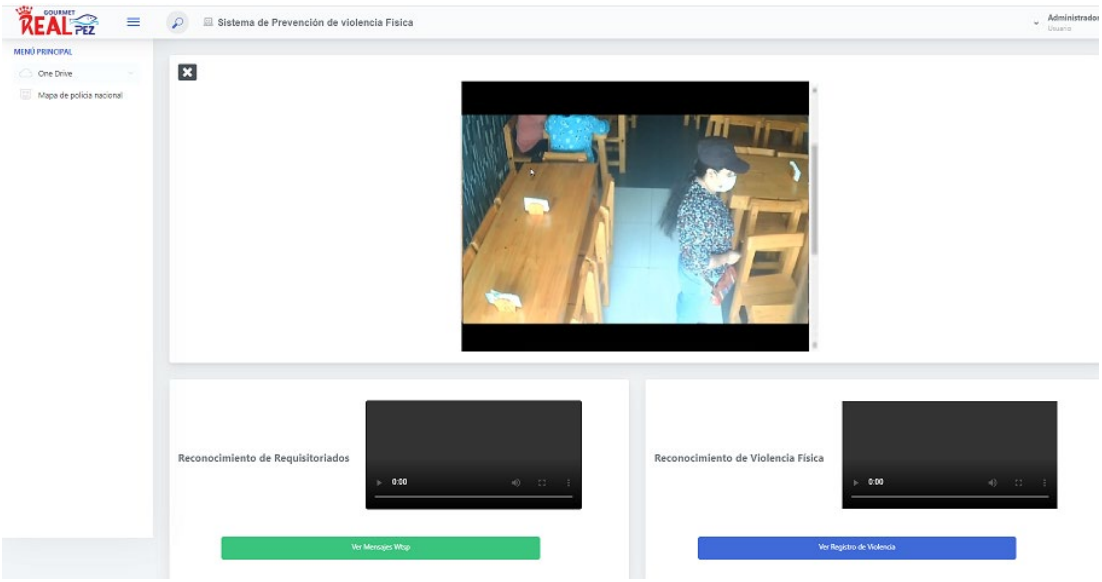
3.- EXITOSO NO DETECTO VIOLENCIA



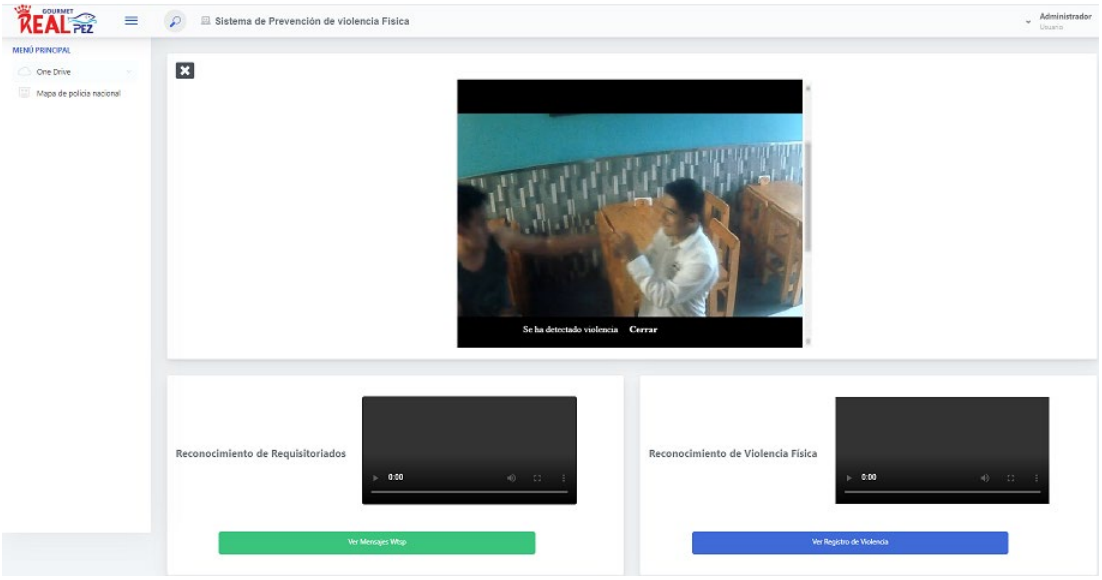
4.- EXITOSO DETECTO VIOLENCIA



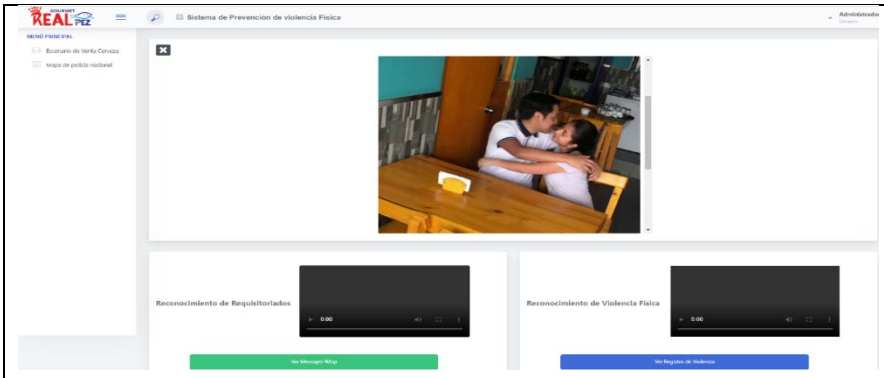
5.- EXITOSO NO DETECTO VIOLENCIA



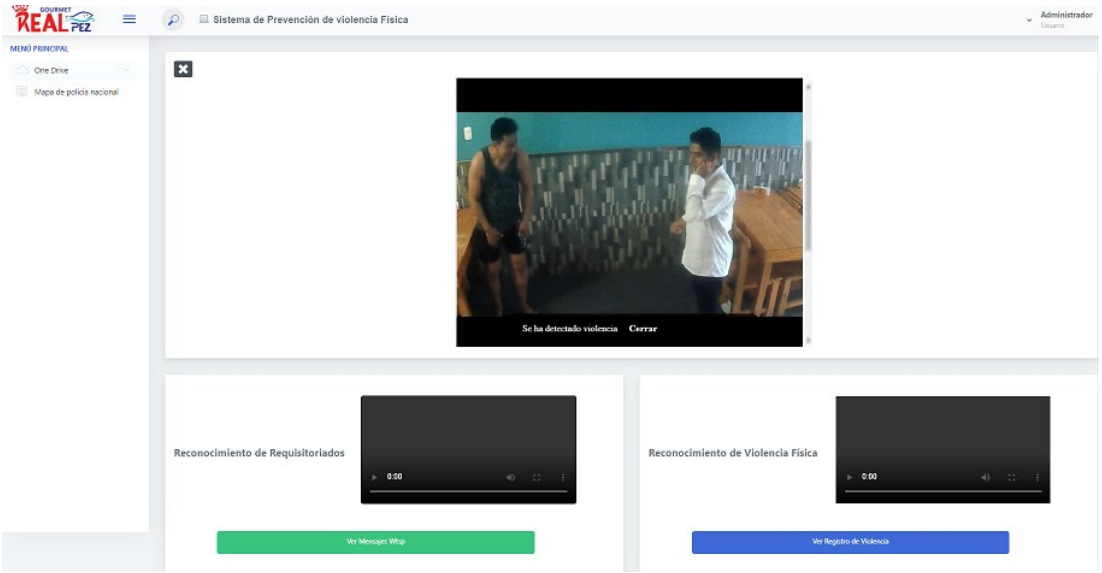
6.- EXITOSO DETECTO VIOLENCIA



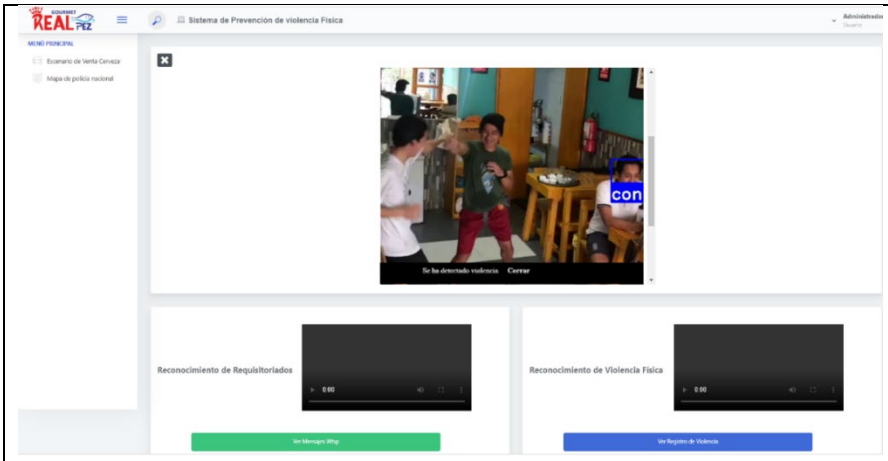
7.- EXITOSO NO DETECTO VIOLENCIA



8.- EXITOSO DETECTO VIOLENCIA

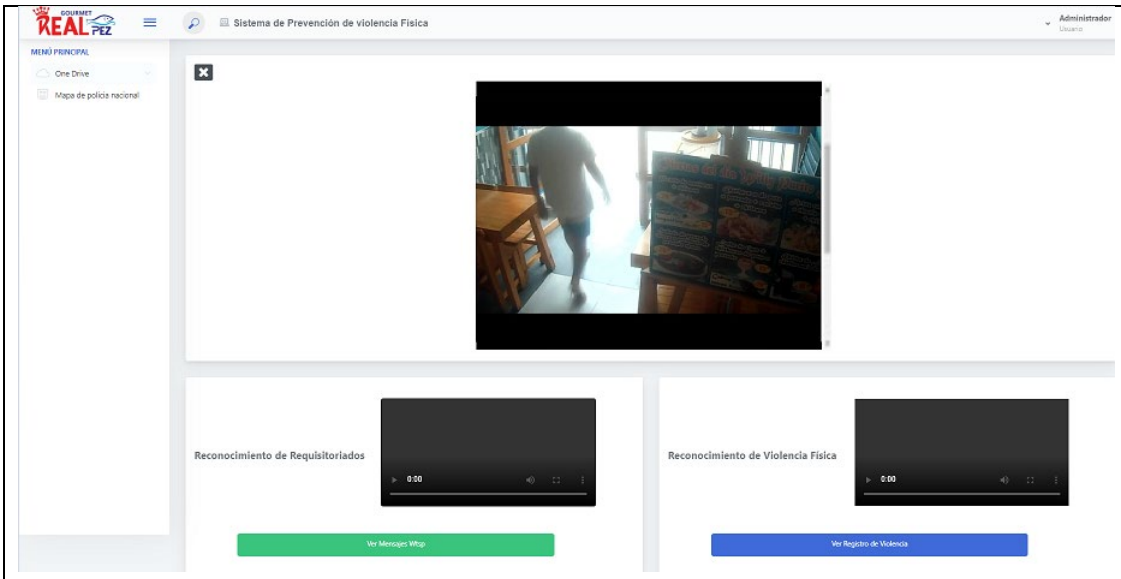


9.- EXITOSO NO DETECTO VIOLENCIA

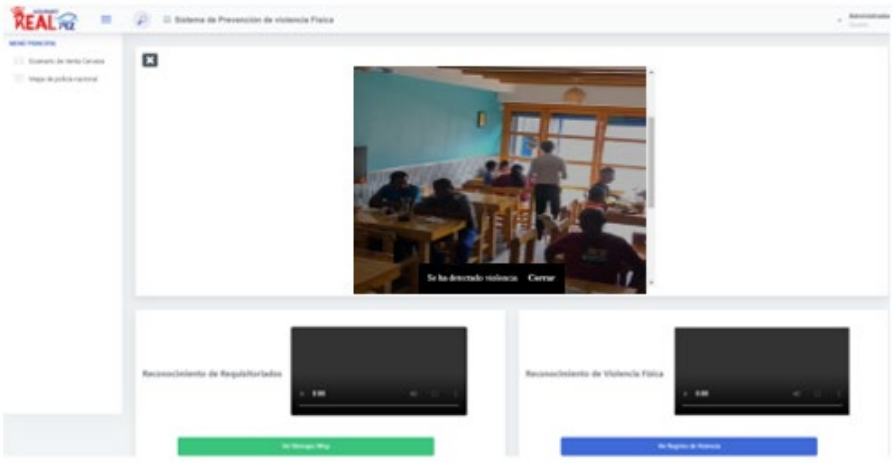


10.- NO EXITOSO DETECTO VIOLENCIA

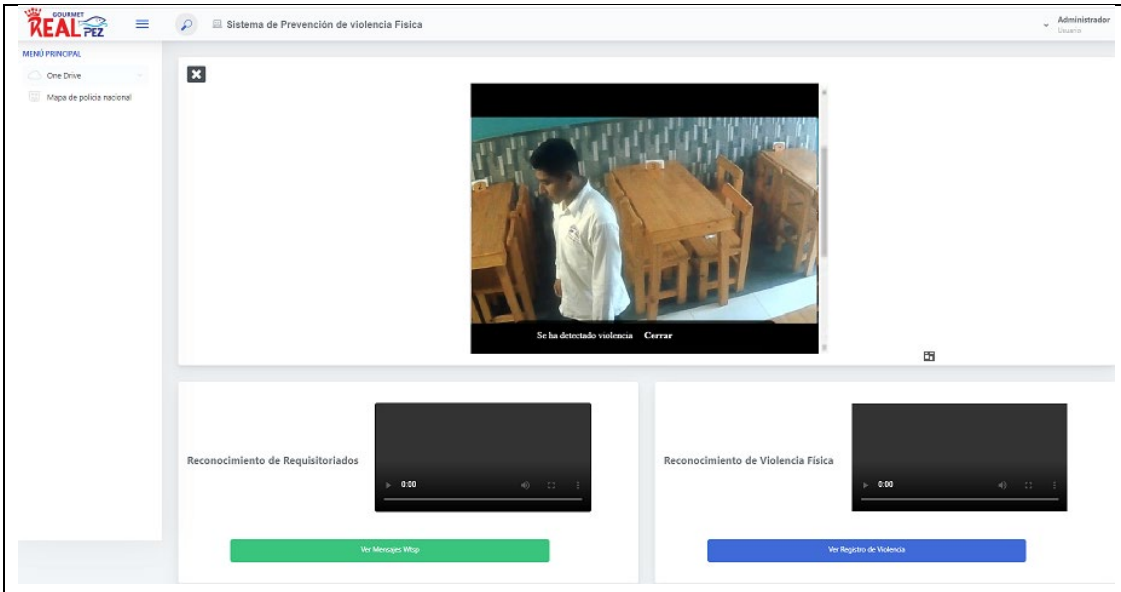
11.- EXITOSO NO DETECTO VIOLENCIA



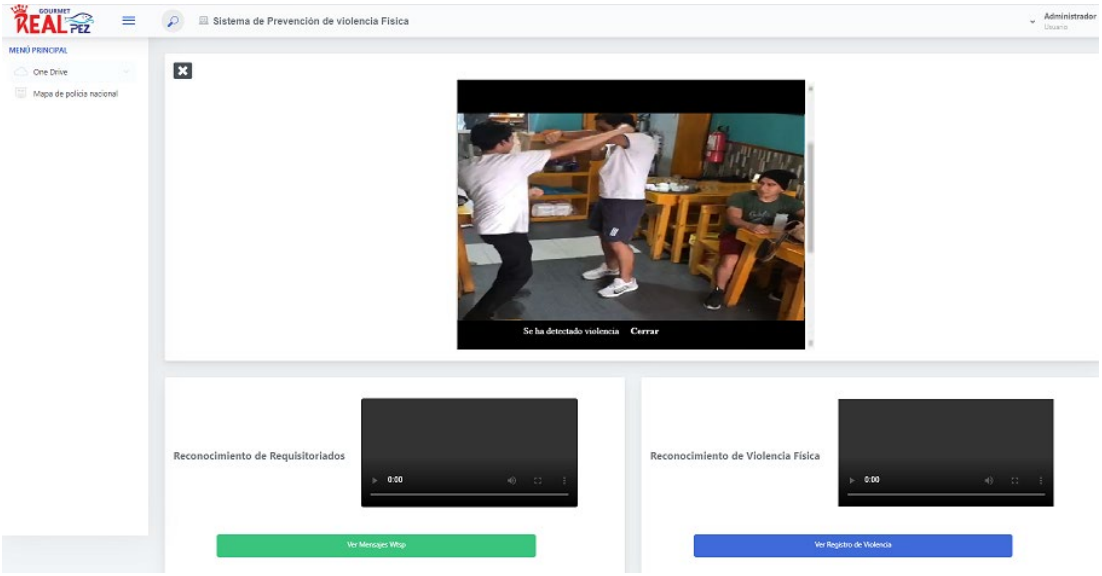
12.- NO EXITOSO DETECTO VIOLENCIA



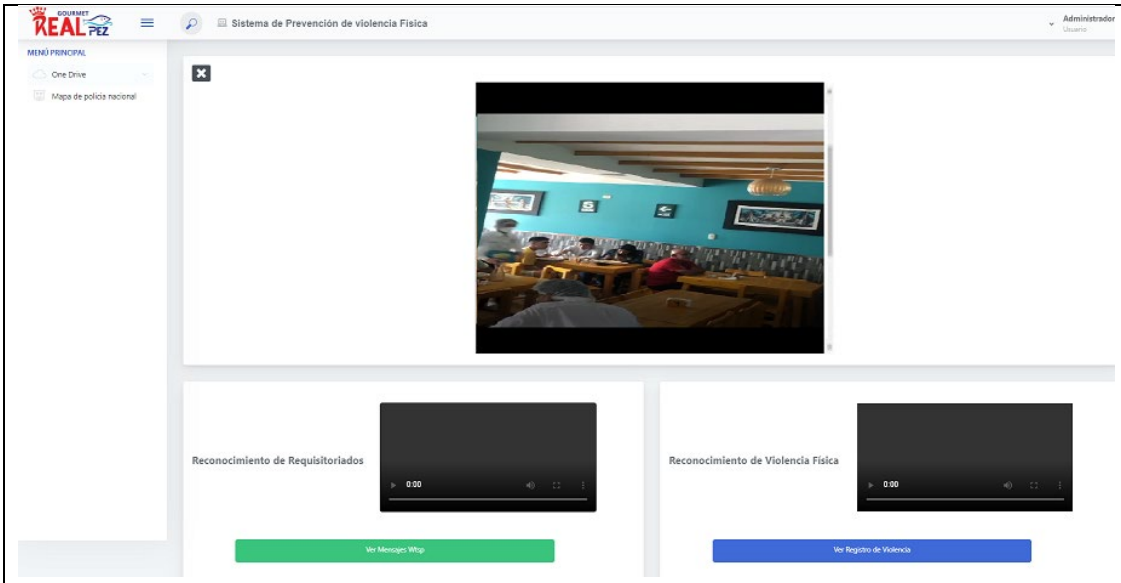
13.- EXITOSO NO DETECTO VIOLENCIA



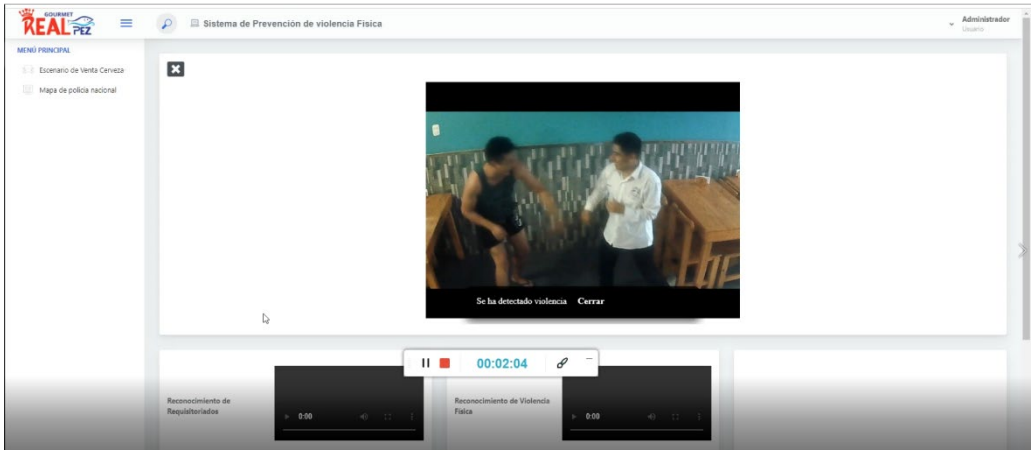
14.- EXITOSO DETECTO VIOLENCIA



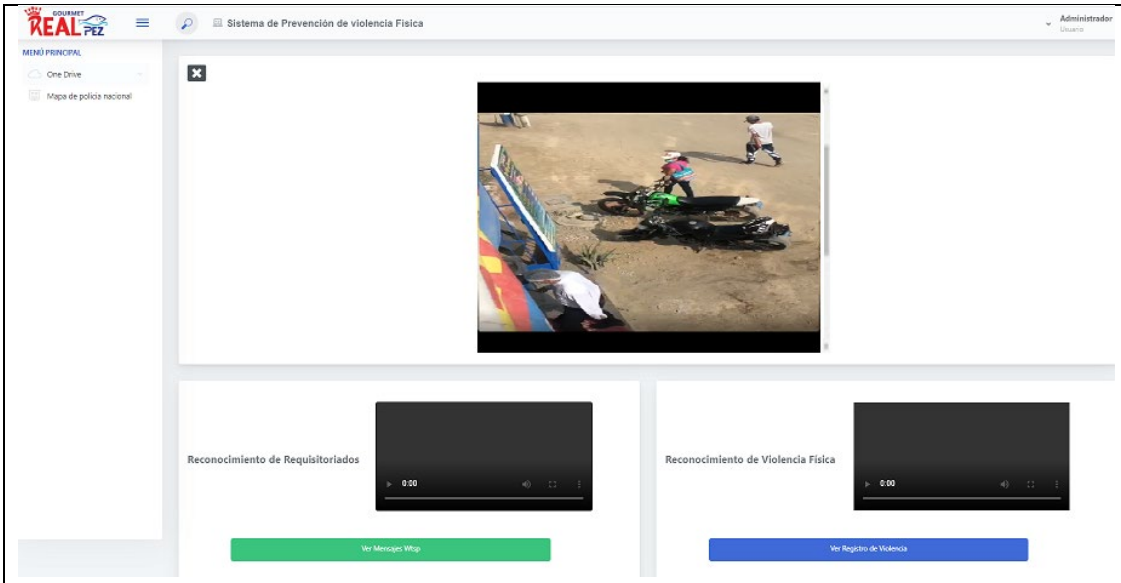
15.- EXITOSO NO DETECTO VIOLENCIA



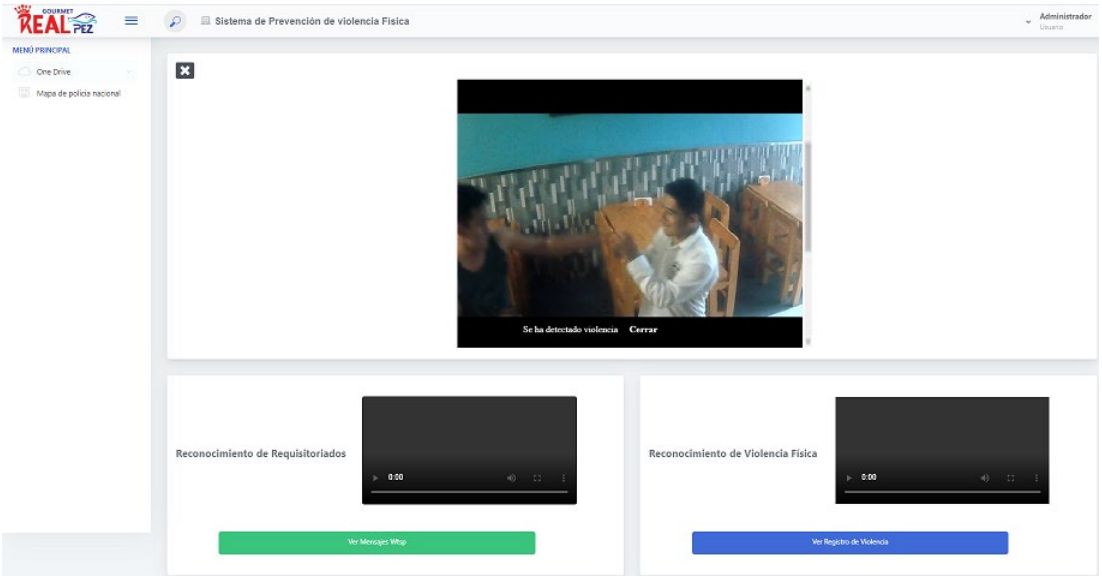
16.- EXITOSO DETECTO VIOLENCIA



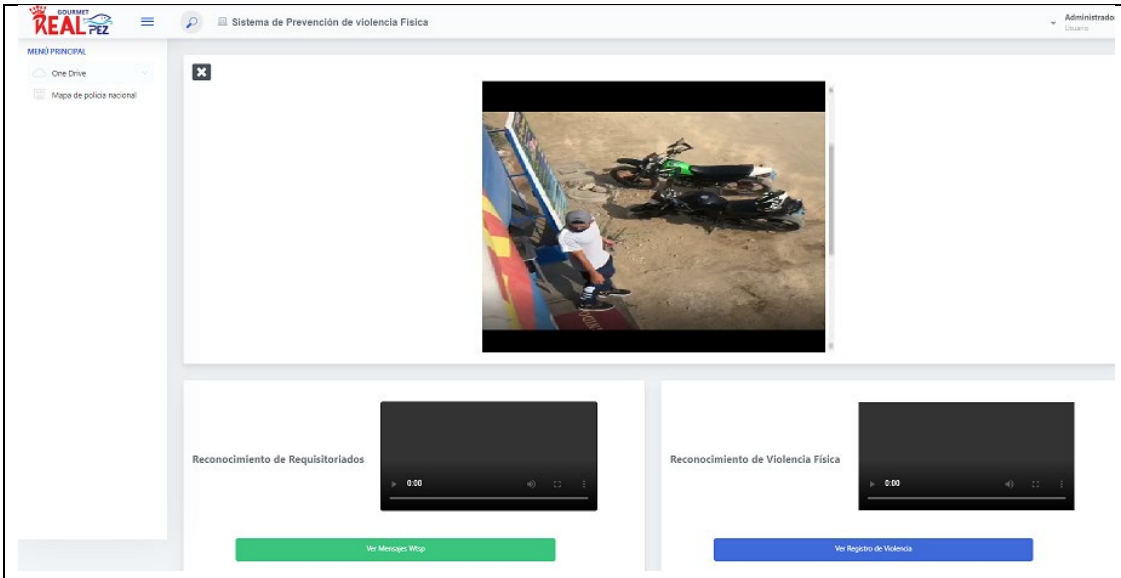
17.- EXITOSO NO DETECTO VIOLENCIA



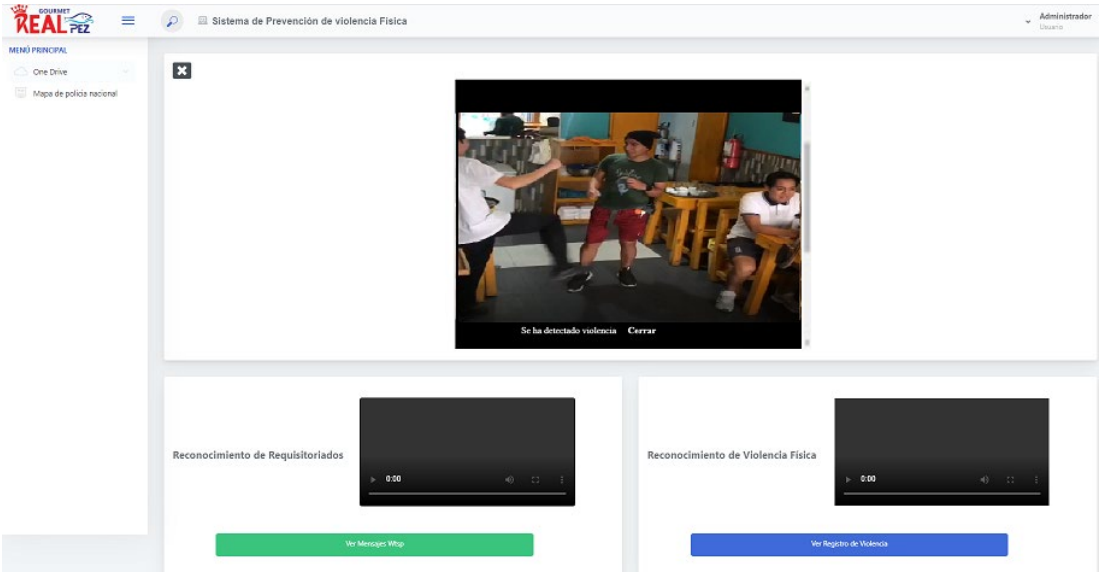
18.- EXITOSO DETECTO VIOLENCIA

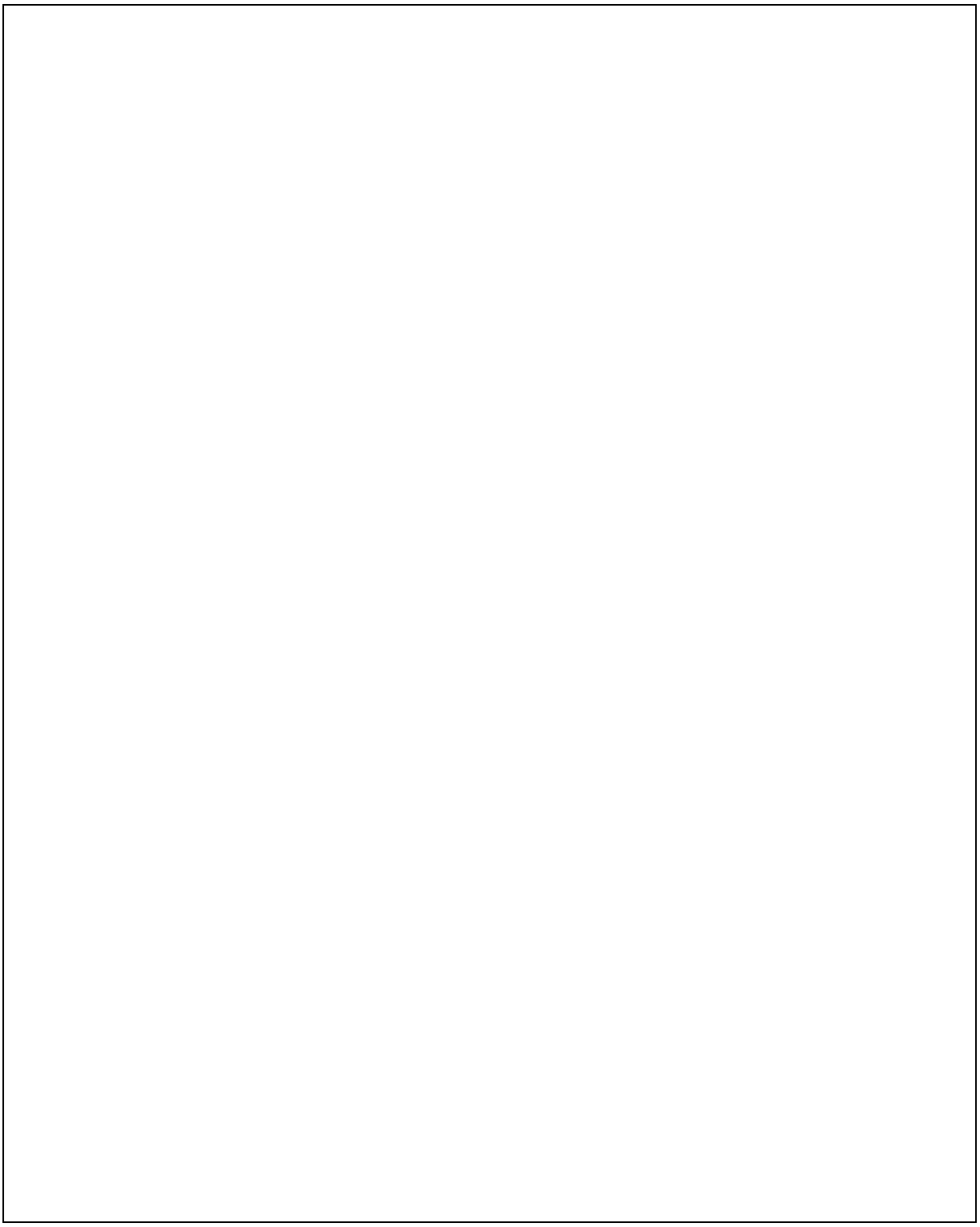


19.- EXITOSO NO DETECTO VIOLENCIA

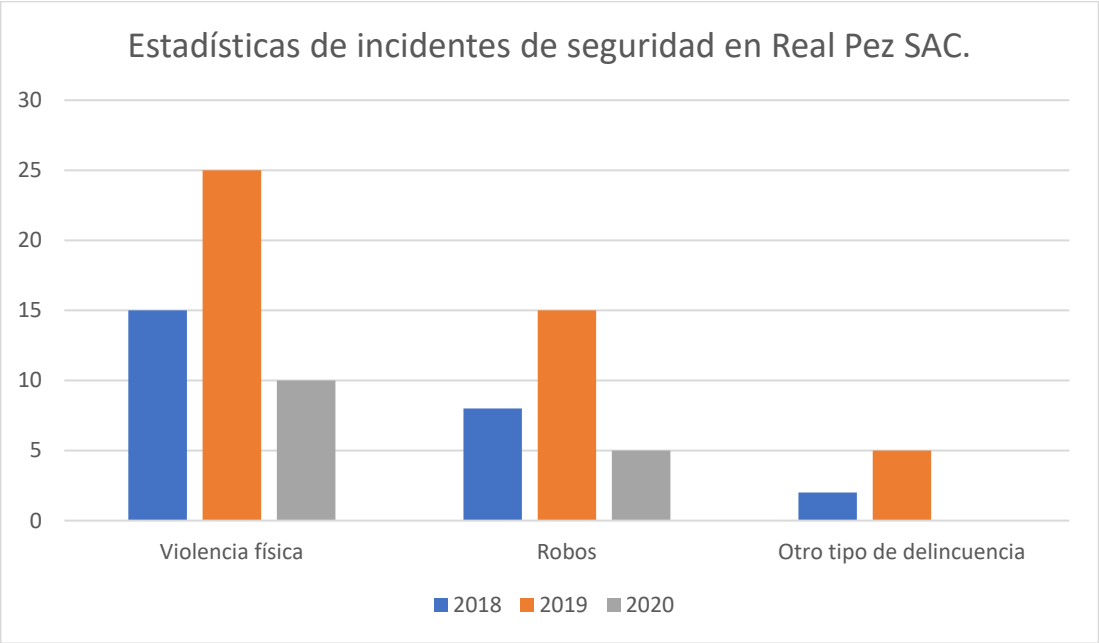


20.- EXITOSO DETECTO VIOLENCIA

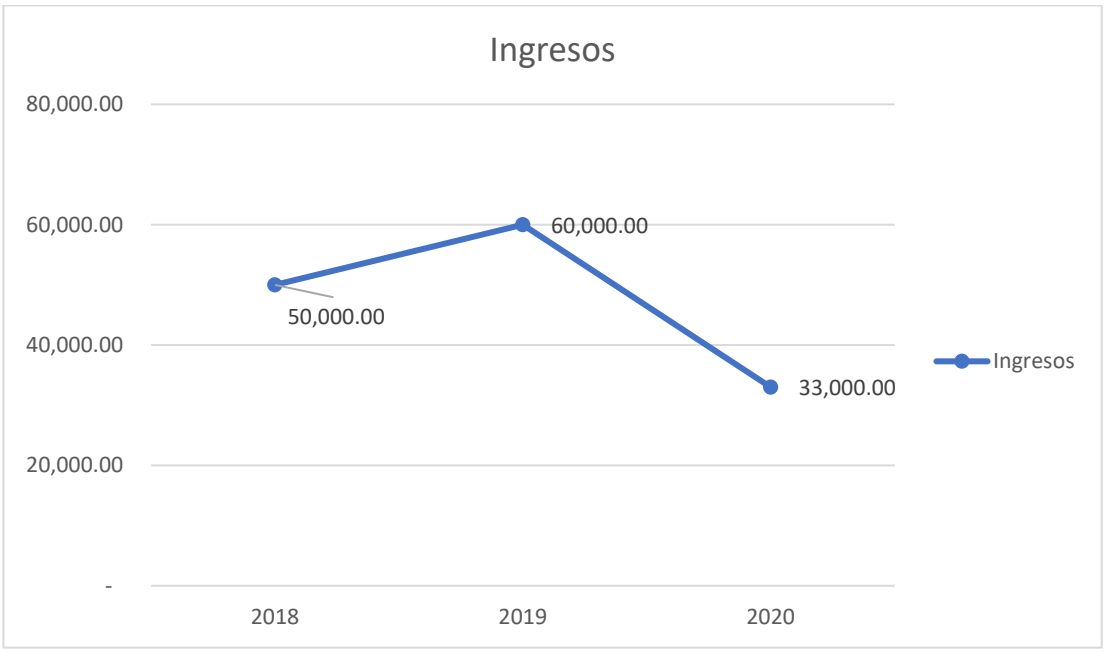




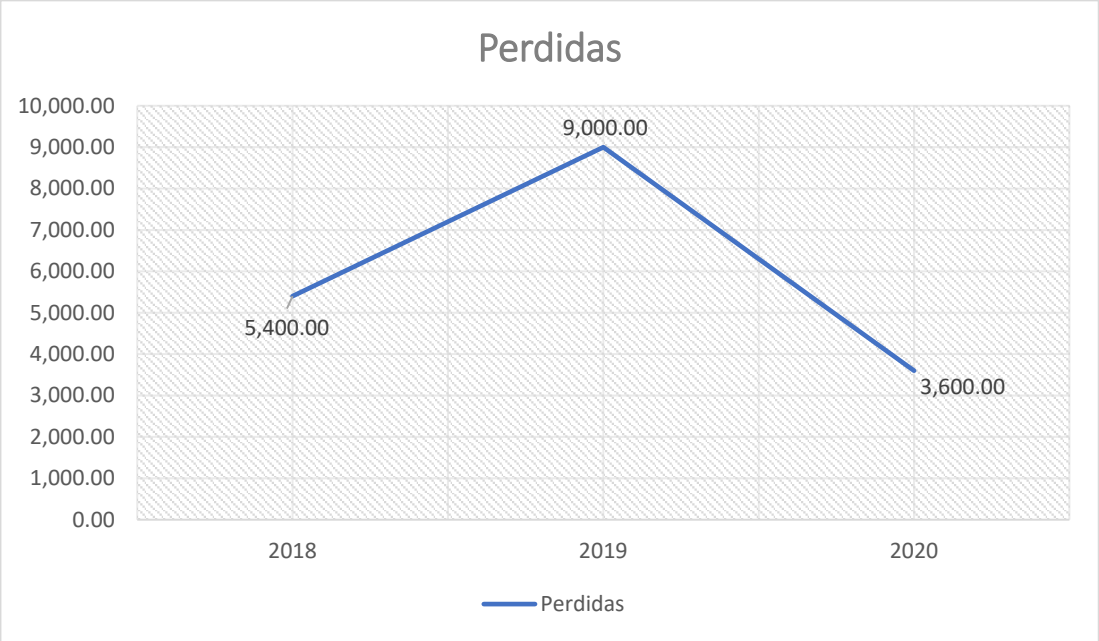
Anexo 15 Resultados de los 50 escenarios del segundo modelo



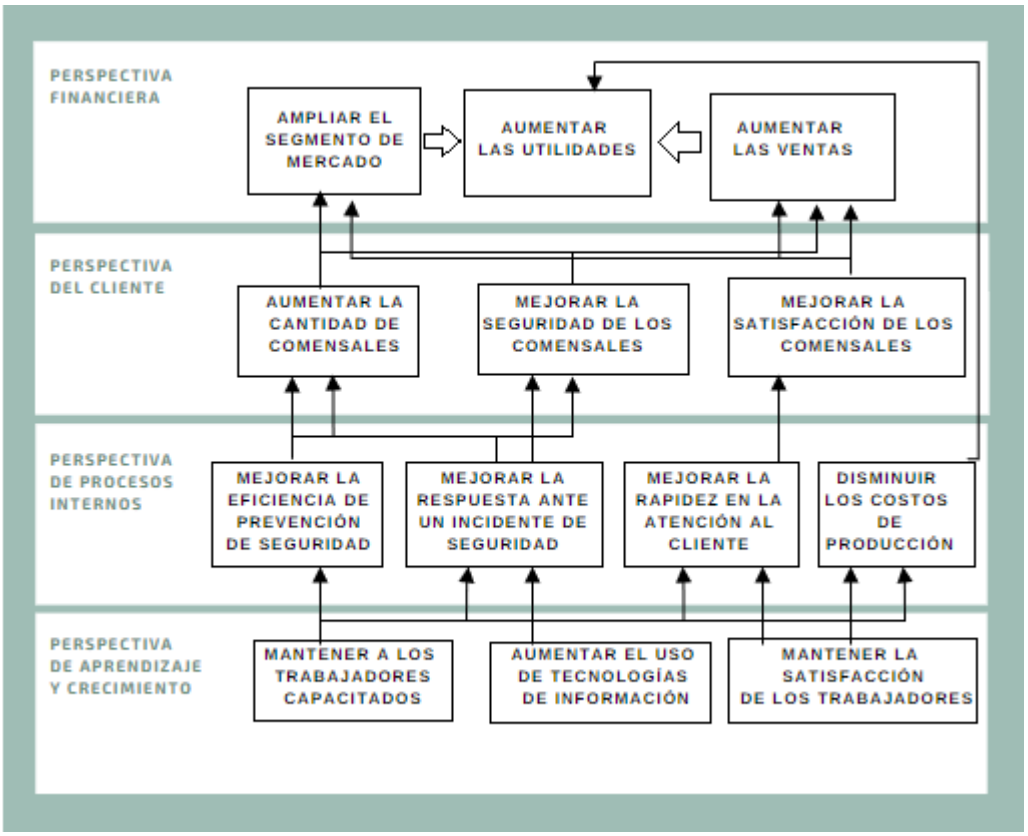
Anexo 16 Incidentes de seguridad en el Real Pez SAC



Anexo 17 Ingresos en el Real Pez SAC



Anexo 18 Perdidas en el Real Pez SAC



Anexo 19 Mapa estratégico

Acuerdo de Confidencialidad

Las partes exponen que como parte de la elaboración de la tesis "SISTEMA DE PREVENCIÓN DE VIOLENCIA FÍSICA MEDIANTE REDES NEURONALES EN EL RESTAURANTE REAL PEZ UBICADO EN EL DISTRITO DEL AGUSTINO" para optar el titulo profesional de ingeniero de Computación y Sistemas en la Facultad de ingeniería y Arquitectura de la Universidad San Martin de Porres, que la información recopilada como parte del desarrollo de la investigación del Restaurante Real Pez el titular de la información, en calidad de Jefe del Restaurante Real Pez con el DNI 44979563 y de nombres Miguel Angel Diaz Vázquez han involucrado o involucraran divulgación escrita o verbal y comunicación al Receptor por parte del Divulgador de información propia, la que puede incluir, pero no se limita a información del negocio, planes de negocio, información personal, dibujos ejemplos y prototipos de artefactos, demostraciones, secretos comerciales, información técnica, escrita, relacionada con la investigación, ya sea dicha comunicación se produzca verbalmente, visualmente o mediante demostraciones o cualquier otro medio, tanto en forma de dibujos, modelos, documentos impresos y/o formato de archivos electrónicos o de cualquier otra manera, en adelante en información.

El Receptor podrá utilizar la información confidencial con el propósito de adquirir conocimientos o para fines académicos de la investigación en mención, para lo cual otorgan el acuerdo del que da cuenta este instrumento.

La información confidencial, y todos los derechos ala misma que ha sido o será divulgadas al Receptor, permanecerán como propiedad del Divulgador. El Receptor no adquirirá derecho alguno, de ningún tipo, obre la información, ni tampoco ningún derecho de utilizarla, excepto para el objetivo del presente acuerdo. La divulgación de la información confidencial no implica el licenciamiento de derecho de patentes o derecho de autor o ningún otro derecho por parte del Divulgador, que no sean los establecidos aquí.

Suscrito por duplicado en Lima, a los 20 días del mes de enero de 2021, firmar como acto de conformidad.

GABRIELA RODRIGUEZ BARDALES

DNI 76400162

WILLY ESPINOZA DIAZ

DNI 75486273

Firma

Miguel Angel Diaz Vázquez

+51 924 344 968

Anexo 20 Acuerdo de confidencialidad

<REAL PEZ SAC>

***Plan de Pruebas* SISTEMA DE PREVENCIÓN Y DETECCION DE VIOLENCIA FÍSICA MEDIANTE REDES NEURONALES EN EL RESTAURANTE REAL PEZ EN EL GUSTINO**
Versión: 2.0

Historial de Revisiones

Versión	Fecha	Autor	Descripción
2.0	*20/12/20*	*Gabriela Rodriguez* Willy Espinoza	

PLAN DE PRUEBAS

1 Alcance

El principal propósito de la evaluación es encontrar errores y defectos que puedan existir en el uso del sistema a fin de corregirlos. Verificar que los validadores de datos funcionen y limiten el ingreso de información, para que no se puedan ingresar datos que no estén permitidos (sólo números en campos numéricos, por ejemplo). Se quiere comprobar además que el sistema cumple con los requerimientos establecidos por el usuario, tiene un rendimiento adecuado en el ambiente donde se encuentra instalado. Otro aspecto importante para evaluar son las características de seguridad relacionadas con el ingreso no autorizado de usuarios, de manera que no puedan realizar modificaciones donde no sean permitidas.

2 Plan de pruebas del proyecto

N°	Casos de Prueba	ID Caso de Uso	Descripción	ID Caso de Prueba
1	Enviar alerta de detección de personas con requisitoria	**CU001**	El usuario podrá visualizar alertas cada que el sistema detecte personas con requisitoria.	**CP001**
2	Visualizar historial de acciones violentas detectadas	**CU002**	El usuario podrá visualizar una interfaz del historial de acciones violentas detectadas	**CP002**
3	Visualizar un mapa de las automóviles de la policía y las comisarias más cercanas	**CU003**	El usuario podrá visualizar una interfaz que determine el tiempo, la distancia y la ubicación de las automóviles de la policía y las comisarias más cercanas	**CU003**
4	Enviar alerta de detección una acción de puñete y/o patada	**CU004**	El usuario podrá visualizar alertas cada que el sistema detecte una acción de puñete y/o patada	**CP004**

3. Requisitos del entorno para las pruebas

Para el proceso de pruebas del proyecto se requiere de la disponibilidad de los siguientes entornos y requerimientos:

Requerimiento	Versión	Fabricante	Implementación de la configuración física
Sistema Operativo	XP Profesional Versión 11.04 Versión 6.0	Windows 10 Pro	Software básico de una computadora que provee una interfaz entre el resto de los programas de los dispositivos hardware y el usuario.
Servidor Web IIS HTTP Sever	Versión 7.5	Microsoft Windows	Presenta entre otras características altamente configurables, bases de datos de autenticación y negociadode contenido.

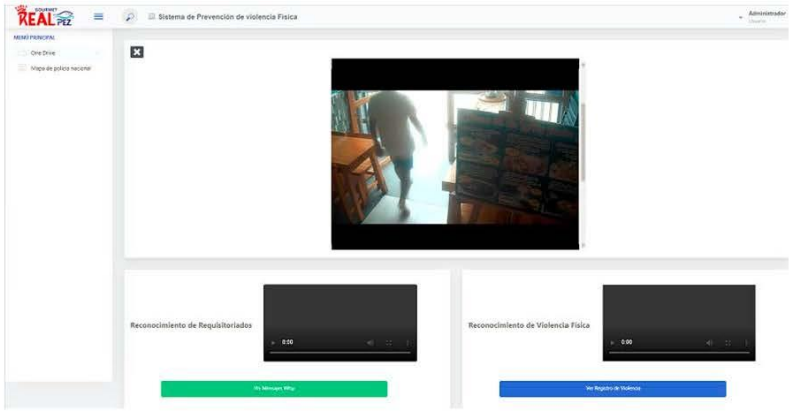
Recurso	Cantidad	Descripción
Disco Duro	1	1024 GB
Memoria Virtual	1	60 GB
Tarjeta de Video	2	Tarjeta gráfica NVIDIA GeForce GTX 950M
Memoria RAM	2	8 GB
Computador	1	Intel Core i7
Microprocesador	1	2.50 GHz
Puerto USB	4	N° 4
Monitor	3	LCD 17" Syncmaster 732n Plus

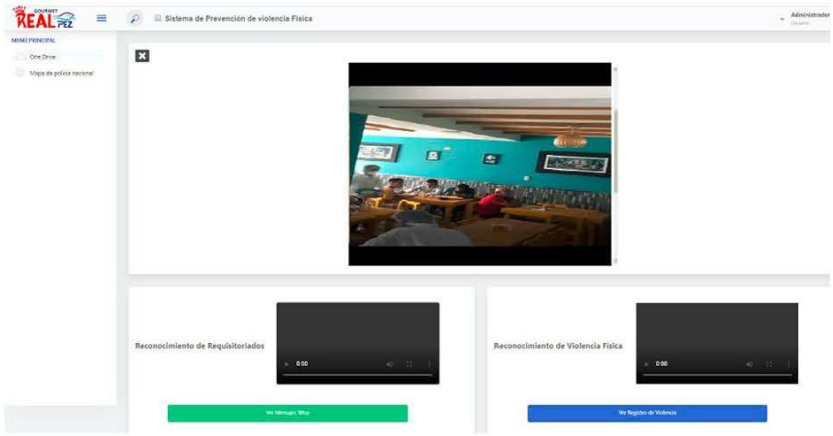
4. Pruebas de los escenarios

A. Pruebas en el Modelo para identificar acciones violentas

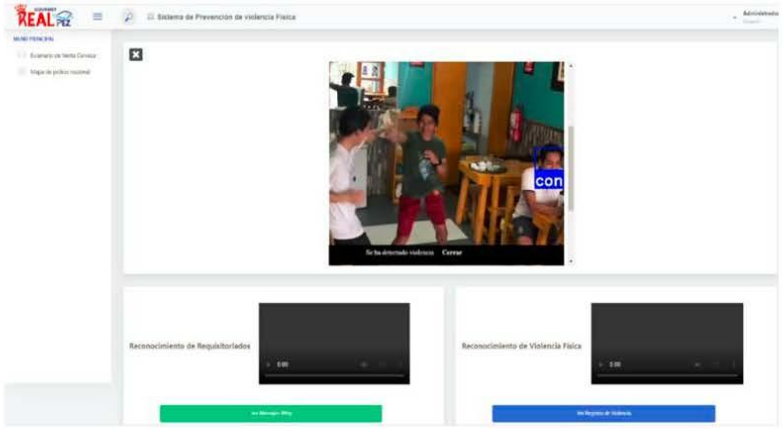
A continuación, se observa las pruebas que se realizaron se realizaron con diferentes escenarios donde se observa violencia física y no violencia física.

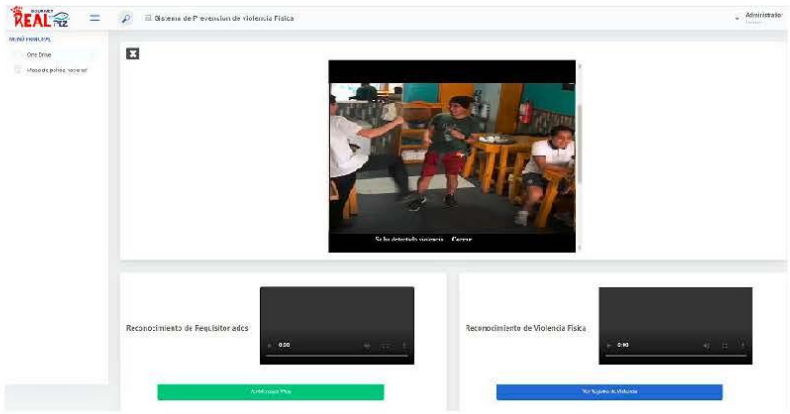
Escenarios sin violencia:

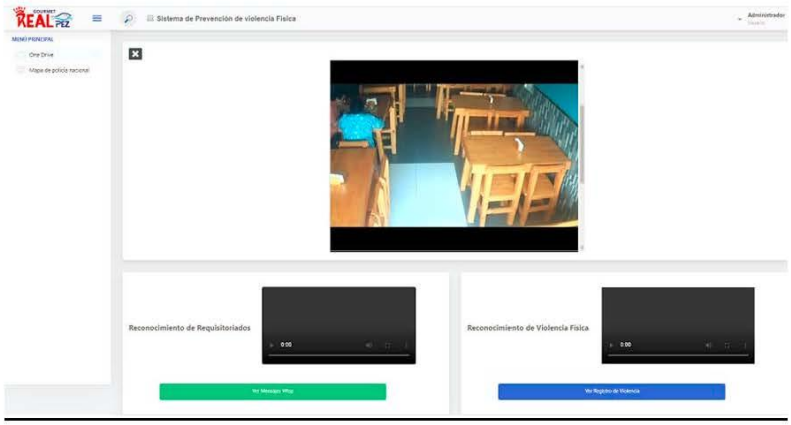


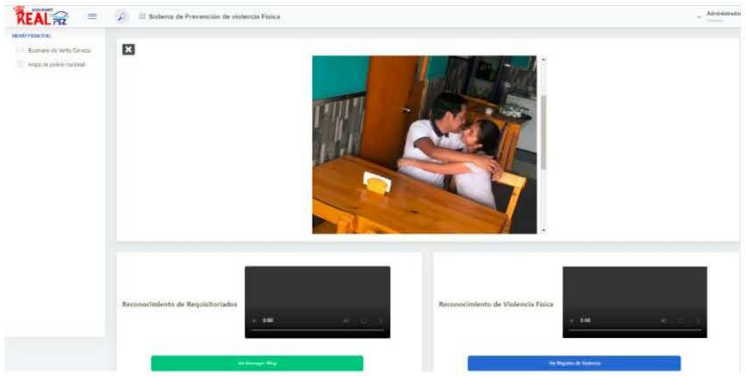


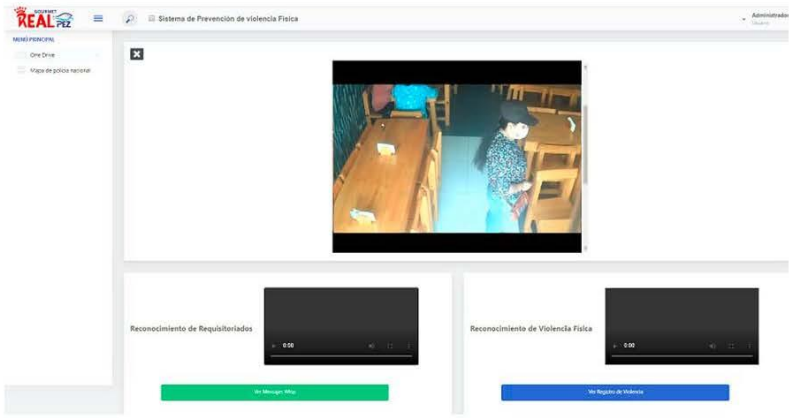
Escenarios con violencia:

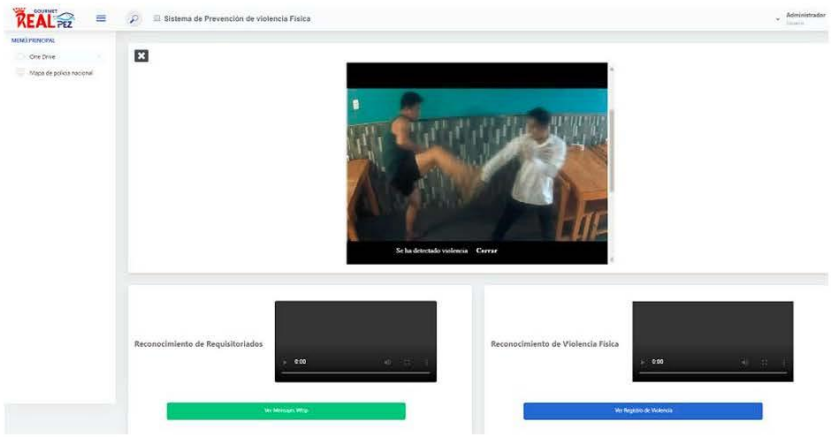

Reconocimiento de Requisitorlados
Reconocimiento de Violencia Física

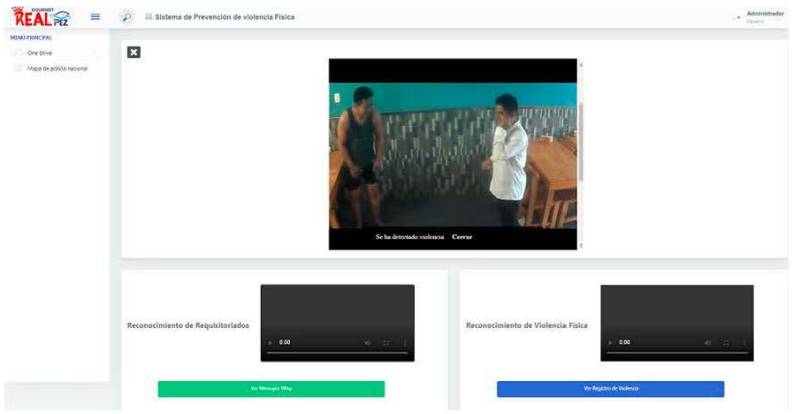

Reconocimiento de Feqlisiter ados
Reconocimiento de Violencia Física

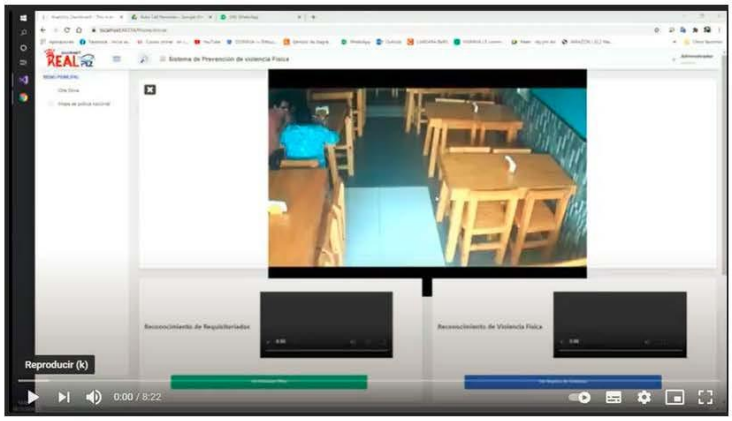

Sistema de Prevención de violencia Física
Reconocimiento de Requisitoriados
Reconocimiento de Violencia Física

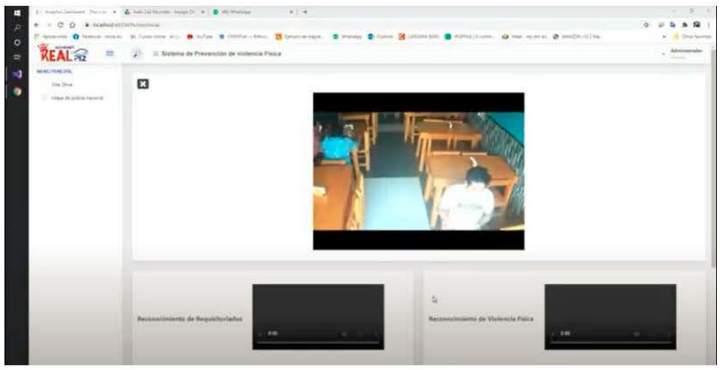

Sistema de Prevención de violencia Física
Reconocimiento de Requisitoriados
Reconocimiento de Violencia Física

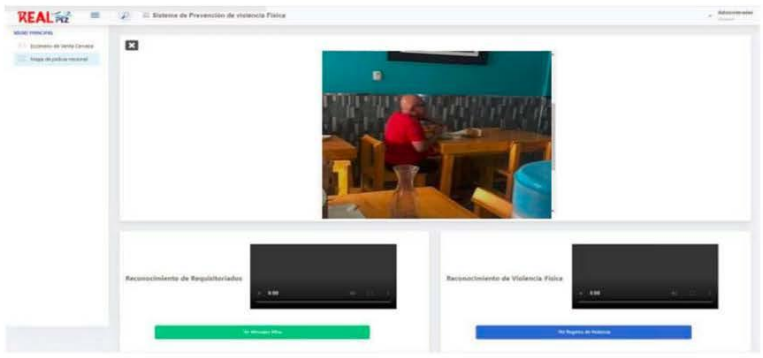

Sistema de Prevención de violencia Física
Reconocimiento de Requisitoriados
Reconocimiento de Violencia Física

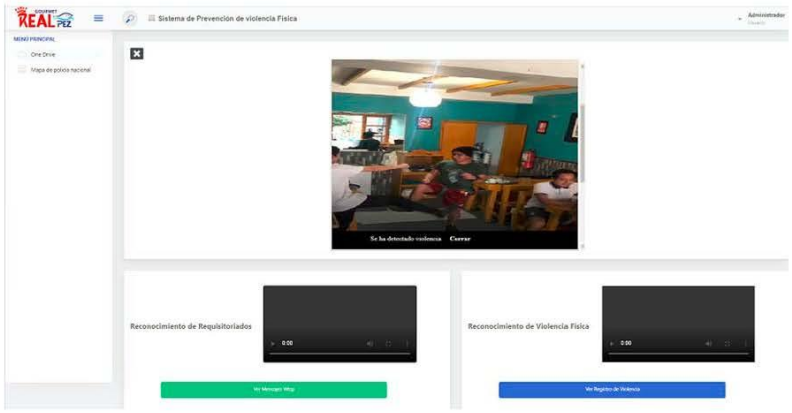

Sistema de Prevención de violencia Física
Se ha detectado violencia Cerrar
Reconocimiento de Requisitoriados
Reconocimiento de Violencia Física

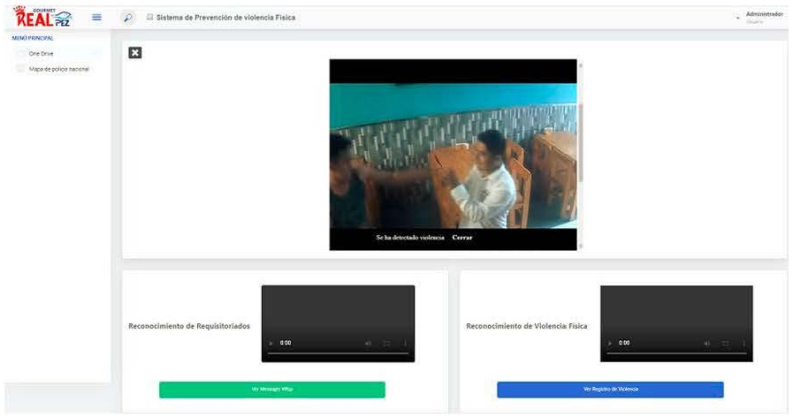

Sistema de Prevención de violencia Física
Se ha detectado violencia Cerrar
Reconocimiento de Requisitoriados
Reconocimiento de Violencia Física


Reproducir (k)
0:00 / 8:22





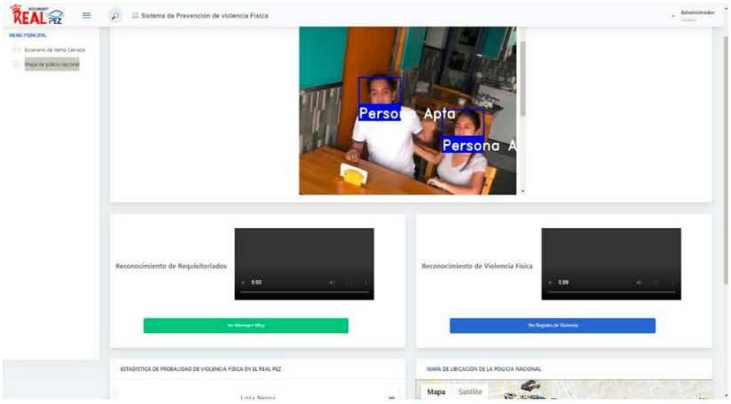


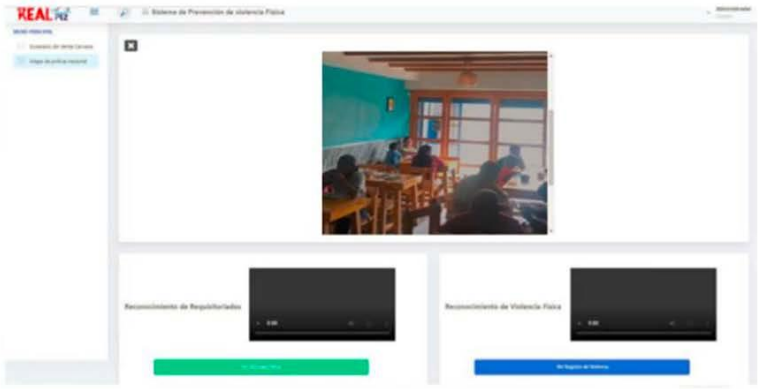


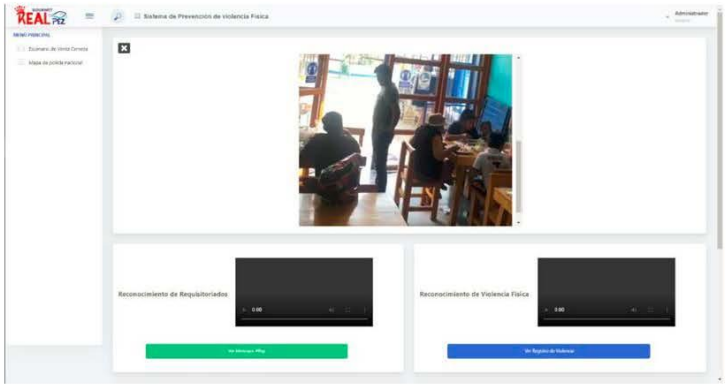
B. Pruebas en el Modelo para identificar escenarios asociados a posibles actos de violencia

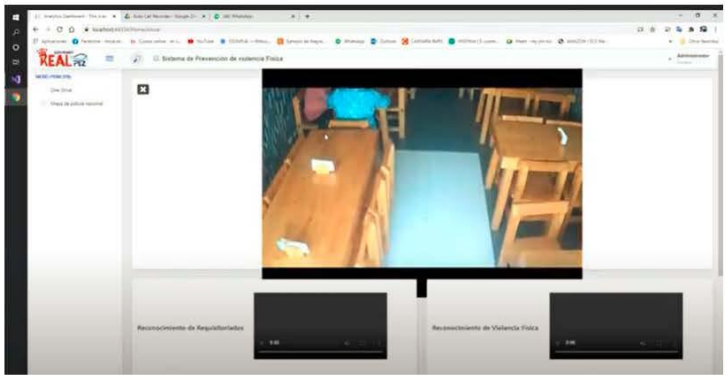
A continuación, se observa las pruebas que se realizaron se realizaron con requisitoria y personas sin requisitoria.

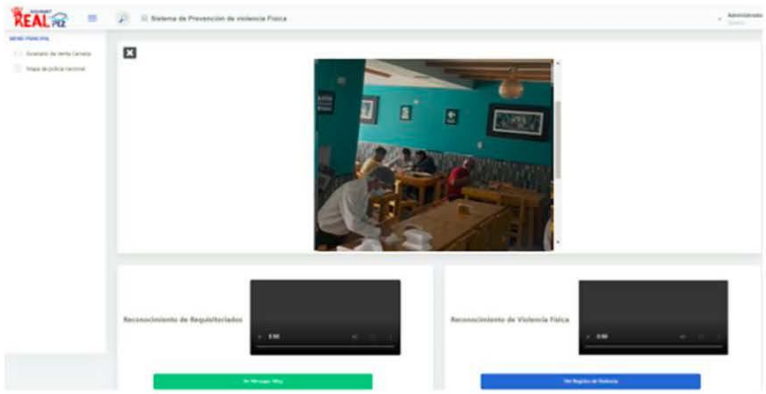
Escenarios de personas sin requisitoria:

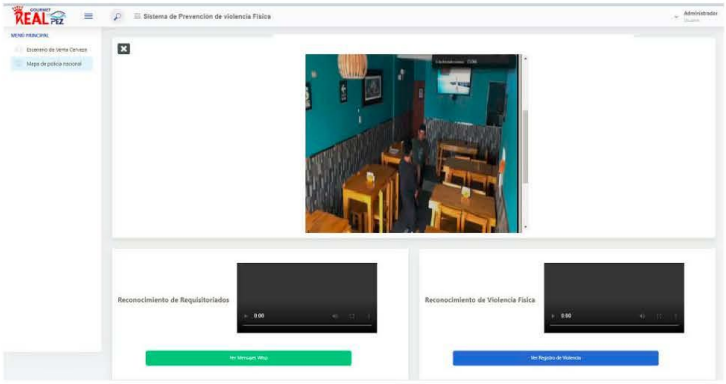

Sistema de Prevención de violencia Física
Apta
Reconocimiento de Requisitoriados
Reconocimiento de Violencia Física
Mapa
Satélite

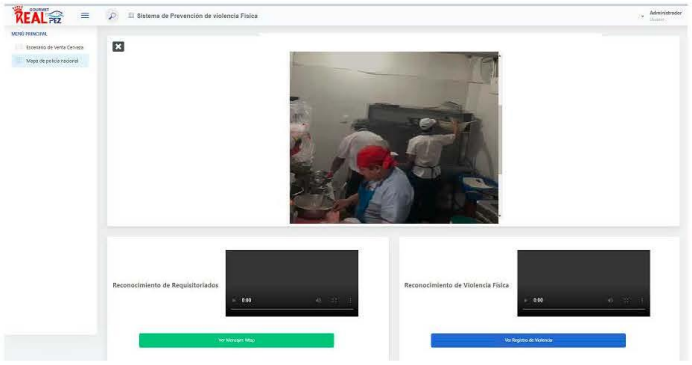




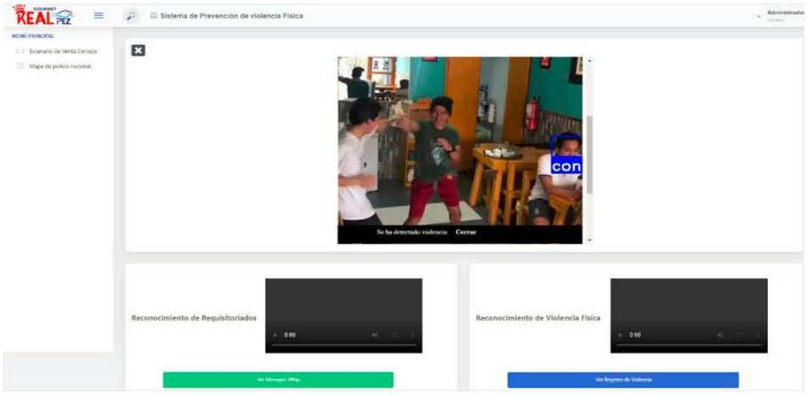


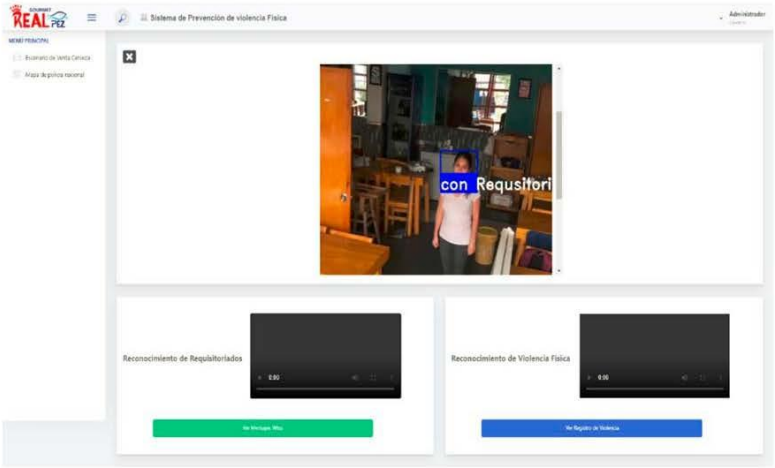


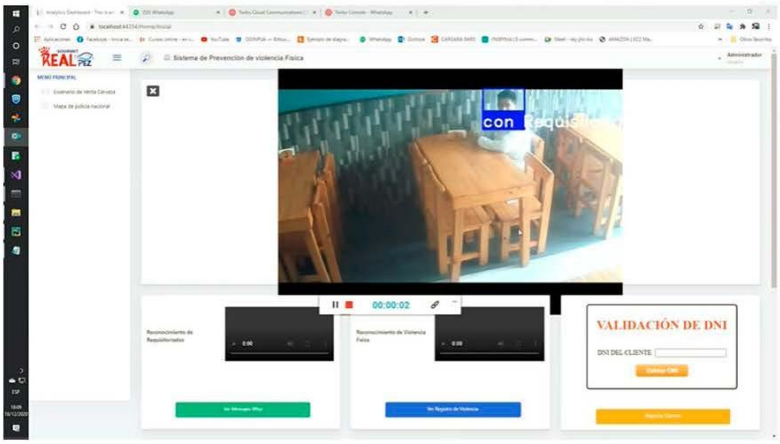


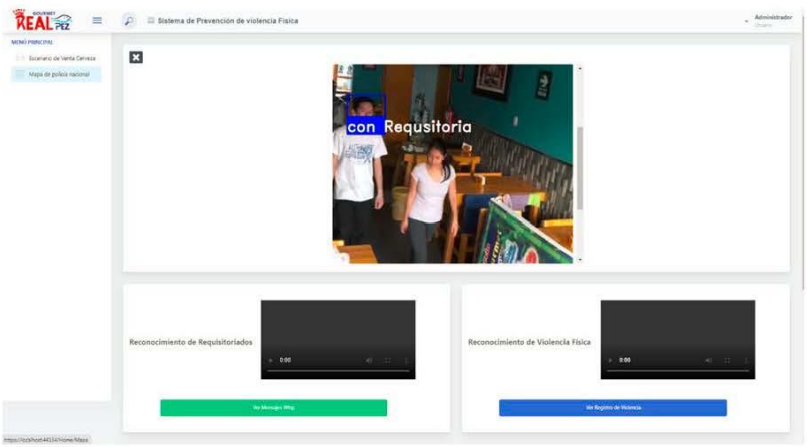


Personas con requisitoria:


Sistema de Prevención de violencia Física
con
Reconocimiento de Requisitoriados
Reconocimiento de Violencia Física


Sistema de Prevención de violencia Física
con Requisitori
Reconocimiento de Requisitoriados
Reconocimiento de Violencia Física





5. Pruebas en el sistema web

Se describirá a detalle las pruebas realizadas referente a los casos de uso del sistema

ID. Caso de Uso: CP001	**Nombre Caso de Prueba:** Enviar alerta de detección de personas con requisitoria

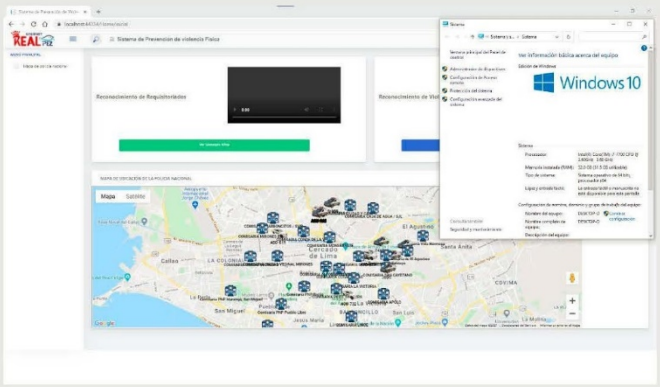
Descripción del caso de uso: El usuario podrá visualizar alertas cada que el sistema detecte personas con requisitoria.	**Autor del Caso de Prueba:** Alexander Espinoza y Gabriela Rodriguez

Condición: Para que el sistema envíe una alerta previamente mediante el modelo se debe identificar una persona con requisitoria

Nro.	Descripción del paso	Resultado Esperado
1	El usuario ingresara al sistema por Chrome y Windows 10	El sistema será ingresado por Google Chrome para poder ejecutarse dentro del Servidor IIS
2	El sistema detecta a una persona con requisitoria	El sistema detecta el rosto e indica un mensaje "Persona con requisitoria"
3	El sistema automáticamente envía un mensaje de WhatsApp con ubicación del lugar.	El sistema automáticamente envía un WhatsApp con ubicación del lugar e indica "Se ha detectado a una persona con requisitoria"

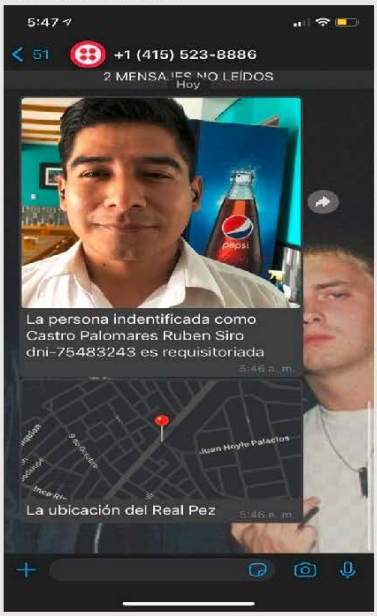
Evidencia de las Pruebas realizadas:

1. El usuario ingresara al sistema por Chrome y Windows 10



2. El sistema detecta el rosto e indica un mensaje "Persona con requisitoria"

3. El sistema automáticamente envía un WhatsApp con ubicación del lugar e indica "Se ha detectado a una persona con requisitoria"



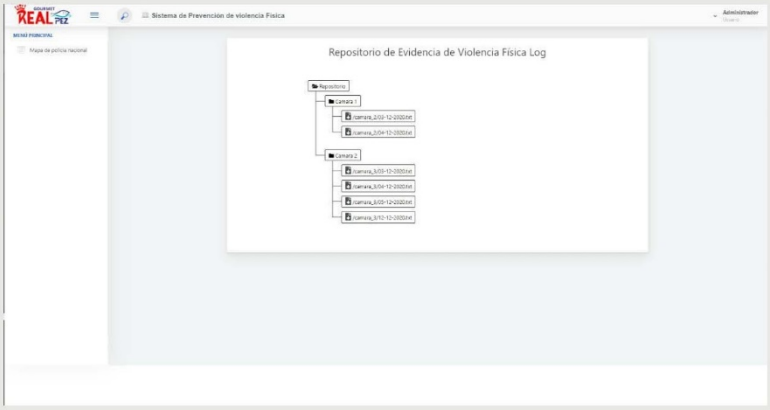
Decisión de Aprobación del Caso de Prueba:	Aprobó: _x_ Fallo: ___
Nombre y firma del Probador	Alexander Espinoza
Fecha de Aprobación del Caso de Prueba: 17/12/2020	

ID. Caso de Uso: CP002	**Nombre Caso de Prueba:** Visualizar evidencia de acciones violentas detectadas
Descripción del caso de uso: Visualizar historial de acciones violentas detectadas	**Autor del Caso de Prueba:** Alexander Espinoza y Gabriela Rodriguez

Condición: Para que el sistema muestre las evidencia previamente debió existir acciones violentas detectadas.

Nro.	Descripción del paso	Resultado Esperado
1	El sistema mostrará las dos cámaras existentes dentro del restaurante con sus respectivas fechas por cada cámara.	El sistema listara las cámaras del local y mostrara la evidencia por cada día registrado con la fecha de la evidencia como nombre.
2	El sistema mostrará el historial de acciones violentas detectadas de esa cámara por la fecha seleccionada.	El sistema listara el log de acciones violentas detectadas por esa cámara con la fecha seleccionada del restaurante.
Evidencia de las Pruebas realizadas:		

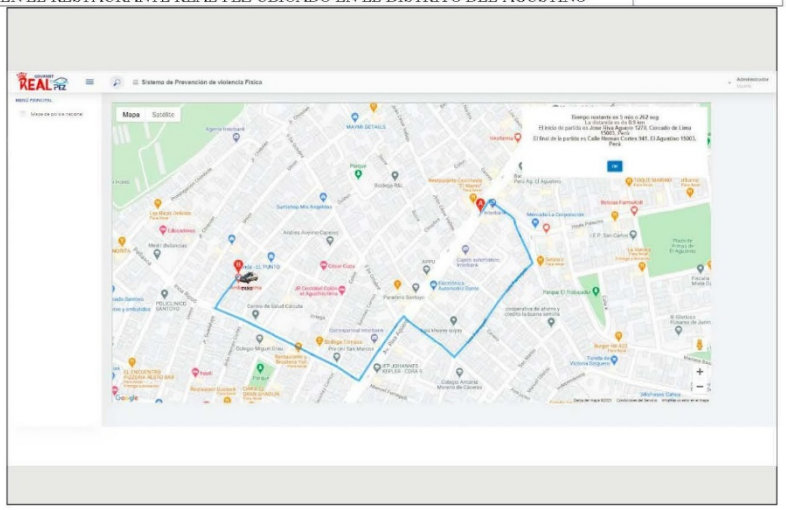
1. El sistema mostrara las dos cámaras existentes dentro del restaurante con sus respectivas fechas por cada cámara.



2. El sistema mostrará el historial de acciones violentas detectadas de esa cámara por la fecha seleccionada.

Decisión de Aprobación del Caso de Prueba:	Aprobó: _x_ Fallo: ___
Nombre y firma del Probador	Alexander Espinoza
Fecha de Aprobación del Caso de Prueba: 17/12/2020	

ID. Caso de Uso: CP003	**Nombre Caso de Prueba:** Visualizar un mapa de las automóviles de la policía y las comisarias más cercanas
Descripción del caso de uso: El usuario podrá visualizar una interfaz que determine el tiempo, la distancia y la ubicación de las automóviles de la policía y las comisarias más cercanas	**Autor del Caso de Prueba:** Alexander Espinoza y Gabriela Rodriguez



Decisión de Aprobación del Caso de Prueba: Aprobó: x Fallo:	
Nombre y firma del Probador	Alexander Espinoza
Fecha de Aprobación del Caso de Prueba: 17/12/2020	

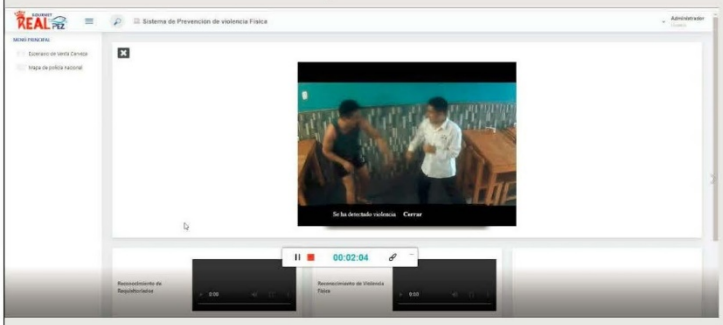
ID. Caso de Uso: CP004	**Nombre Caso de Prueba:** Enviar alerta de detección una acción de puñete y/o patada
Descripción del caso de uso: El usuario podrá visualizar alertas cada que el sistema detecte una acción de puñete y/o patada	**Autor del Caso de Prueba:** Alexander Espinoza y Gabriela Rodriguez

Condición: Para que el sistema envíe una alerta previamente mediante el modelo se debe identificar una acción de violencia física

Nro.	Descripción del paso	Resultado Esperado
1	El sistema detecta a una acción de violencia física	El sistema emite una alerta "Violencia detectada"
2	El sistema automáticamente envía un mensaje de WhatsApp con ubicación del lugar.	El sistema automáticamente envía un WhatsApp con ubicación del lugar e indica "Existe violencia en el Real Pez SAC"

Evidencia de las Pruebas realizadas:

3. El sistema detecta a una acción de violencia física



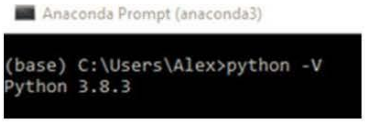
4. El sistema automáticamente envía un mensaje de WhatsApp con ubicación del lugar.

Decisión de Aprobación del Caso de Prueba: Aprobó: x Fallo:	
Nombre y firma del Probador	Alexander Espinoza
Fecha de Aprobación del Caso de Prueba: 17/12/2020	

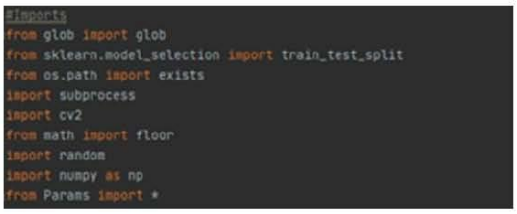
5. Pruebas de instalación

Verificar que el sistema al instalar no presenta anomalías.

A. Instalación del entorno, donde la versión mínima de Python para poder ejecutar la librería a utilizar en el proyecto es Tensor Flow debe ser de 3.5 a 3.8. Asimismo, se descarga el SW Anaconda el cual nos permitirá ejecutar el proyecto.



B. Posteriormente, se instala las librerías correspondientes dentro del entorno de desarrollo y se importa las librerías respectivas para la creación de la red neuronal.



```
#Imports
from glob import glob
from sklearn.model_selection import train_test_split
from os.path import exists
import subprocess
import cv2
from math import floor
import random
import numpy as np
from Params import *
```

C. Finalmente, se observa que el modelo ha sido levantado con éxito.

6. Pruebas de caja blanca:

Se da a entender el enfoque interno del sistema, en este proceso de pruebas se examina el software desde el interior, no es obligatorio solo si se entrega la funcionalidad sino como es analizada la información del sistema para ofrecer dicho proceso.

Por otro lado, se ha utilizado Coverage.py es una técnica que procesa el programa, analiza fracciones del código se han ejecutado, luego analiza la fuente para reconocer el código que podría haberse compilado y otro que no.

Para la instalación ejecutamos el siguiente comando “pip install coverage” y se descargara automáticamente

El siguiente paso es ejecutar dentro de nuestro servidor Anaconda el siguiente comando

Para poder validar con las pruebas deberá probarse con varios casos de prueba: Determinar posibles salidas distintas.

- Cada caso deberá acatar en un caso y en otro no.
- Puede no llegar al 100%
- Código que nunca se compilara

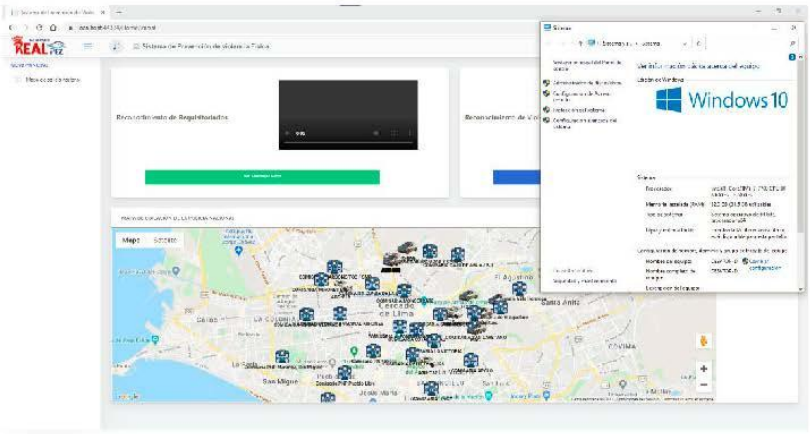
Para informar sobre los resultados siguiente ejecutados la siguiente línea de comando “coverage report -m” y si queremos ver en html

```
(base) G:\TALLER\FINAL\PREVENCION>coverage report
Name              Stmts   Miss  Cover
-----------------------------------
Mensaje_wtsp.py       8      5    38%
face_rec.py          98     59    40%
-----------------------------------
TOTAL               106     64    40%
```

El resultado de la prueba del 40%, ya que nunca se ejecutaría lo que está dentro del ciclo de pruebas y comprobamos el total determina las posibles salidas de la compilación del código.

7. Pruebas Requerimientos no Funcionales

A. Disponibilidad del sistema a través del explorador Google Chrome



B. El sistema deberá contar con la protección contra robots dentro de las páginas webs.

La evaluación de la seguridad de un aplicativo, en todas o alguna de sus capas, buscando vulnerabilidades y malas prácticas de codificación o configuración que puedan derivar en riesgos de intrusión.

- Protección Contra Robots:

  La primera prueba es la protección contra robots que es lo más común dentro de las páginas webs

  La forma de incluir dicho meta-tag es página de inicio, para impedir su ingreso.

  <meta name="robots" content="index, follow">

  Indica que la página puede ser indexada y sus enlaces seguidos

  <meta name="robots" content="index, nofollow">

  Indica que la página puede ser indexada, pero sus enlaces no pueden ser seguidos

  <meta name="robots" content="noindex, follow">

  Indica que la página no puede ser indexada, pero sus enlaces pueden ser seguidos

  <meta name="robots" content="noindex,nofollow">

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta http-equiv="Content-Language" content="en">
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Sistema de Prevención de Violencia Física</title>
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no, shrink-to-fit=n
    <meta name="msapplication-tap-highlight" content="no">
    @Scripts.Render("~/bundles/modernizr")
    <link href="https://demo.dashboardpack.com/architectui-html-free/main.css" rel="stylesheet">
    <meta name="googlebot" content="noindex">
    <meta name="robots" content="noindex" />
</head>
```

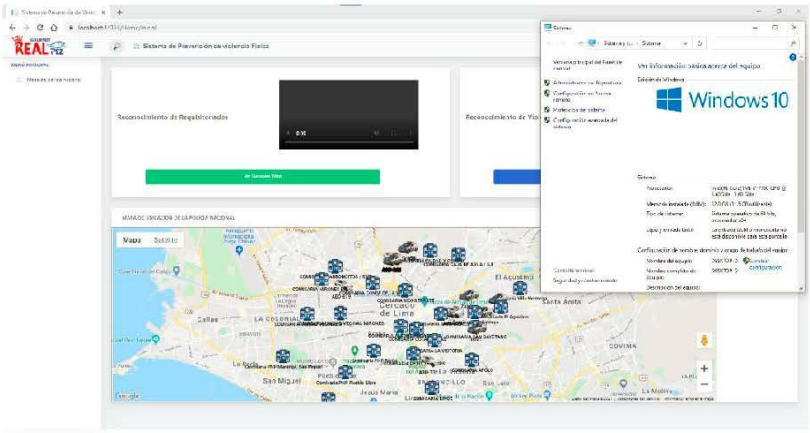
- Peticiones Get y Post

Proteger los códigos y programas internos del servidor web para evitar la transferencia de parámetros o información a través de la dirección de acceso a las páginas al usar el método GET para la entrega de parámetros, los

cuales son mecanismos frecuentes de hackeo o robo de información. Y se puede bloquear configurando dentro de nuestro Web.config en la etiqueta como partura "<system.webServer>" delimitar el requestFiltering

```
<system.webServer>
  <handlers>
    <remove name="BlockViewHandler"/>
    <add name="BlockViewHandler" path="*" verb="*" preCondition="integratedMode" type="System.Web.HttpNotFoundHandler" />
  </handlers>
  <security>
    <requestFiltering>
      <verbs allowUnlisted="true">
        <add verb="POST" allowed="false" />
        <add verb="GET" allowed="false" />
      </verbs>
    </requestFiltering>
  </security>
</system.webServer>
```

C. El sistema deberá correr en el sistema operativo Windows 10



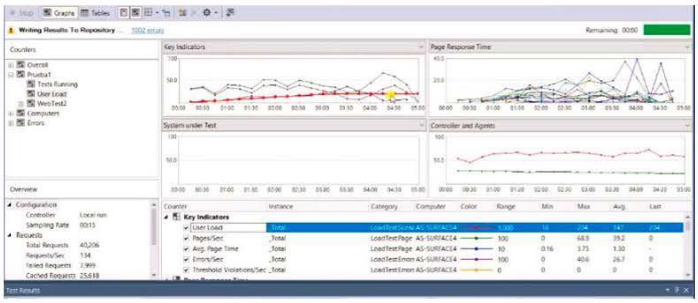
D. El sistema podrá permitir una cierta cantidad usuario simultáneamente.

Dentro de c# existe un componente llamado "Herramientas de rendimiento web y pruebas de carga y Seleccionaremos la plantilla Proyecto de prueba de carga y rendimiento web.

El siguiente paso es la prueba de carga local y luego escoger las pruebas de carga y configuramos nuestra página local y escogemos el escenario de prueba a cargar.

La prueba de carga comienza a ejecutarse aparece la prueba que está

en curso, Una vez finalizada la prueba, se muestra los Gráficos, obtener información de distinto tipo de detalles sobre los resultados de la prueba de carga.



Anexo 21 Plan de Pruebas

Resultados de la prueba

Se obtuvieron resultados positivos, no hubo errores en las pruebas del sistema web, instalación, caja blanca ni pruebas no funcionales.

Nombre del Probador:	Alexander Espinoza Gabriela Rodriguez
Fecha de Aprobación de la prueba: 27/12/2019	
Firma del aprobador (dueño de la empresa)  Firma **Miguel Angel Diaz Vázquez** **+51 924 344 968**	

Anexo 22 Aceptación de Plan de Pruebas

PODER EJECUTIVO

PRESIDENCIA DEL CONSEJO DE MINISTROS

Decreto Supremo que aplica el beneficio de recompensas del Decreto Legislativo Nº 1180 y su Reglamento, para promover y lograr la captura de responsables de los delitos de corrupción grave cometidos por funcionarios públicos o particulares

DECRETO SUPREMO
Nº 027-2017-PCM

EL PRESIDENTE DE LA REPÚBLICA

CONSIDERANDO:

Que, mediante Decreto Legislativo Nº 1180, se establece el beneficio de recompensas para promover y lograr la captura de miembros de las organizaciones criminales, organizaciones terroristas y responsables de delitos de alta lesividad, el cual regula el establecimiento y el otorgamiento del beneficio de recompensa a favor de ciudadanos colaboradores que brinden información oportuna e idónea que permita la búsqueda, captura y/o entrega de miembros de una organización criminal, personas vinculadas a ella o que actúen por encargo de la misma, organizaciones terroristas, así como autores o presuntos autores y partícipes de uno o más delitos, con la finalidad de reducir los índices de criminalidad que afectan el orden interno y la seguridad ciudadana;

Que, de acuerdo a lo establecido en el Decreto Legislativo Nº 1180, se considera ciudadano colaborador a cualquier persona que brinde información oportuna e idónea que permita la búsqueda, captura y/o entrega de miembros de una organización criminal, personas vinculadas a ella o que actúen por encargo de la misma, organizaciones terroristas, así como autores o presuntos autores y partícipes de uno o más delitos, siempre que no se encuentre dentro de las causales de exclusión previstas en el artículo 2 del mencionado Decreto Legislativo.

Que, estando a la Única Disposición Complementaria Final del mencionado Decreto Legislativo, mediante Decreto Supremo N° 011-2016-PCM, se aprobó el Reglamento del Decreto Legislativo Nº 1180, que establece el beneficio de recompensas para promover y lograr la captura de miembros de las organizaciones criminales, organizaciones terroristas y responsables de delitos de alta lesividad, en adelante el Reglamento;

Que, en el marco de la Ley N° 30506, Ley que delega en el Poder Ejecutivo la facultad de legislar en materia de reactivación económica y formalización, seguridad ciudadana, lucha contra la corrupción, agua y saneamiento y reorganización de Petroperú S.A., se han emitido normas con rango de ley que fortalecen la lucha contra la corrupción en todos los niveles de gobierno, que hace necesaria la adecuación de la normatividad vigente a estos parámetros de lucha contra la corrupción en los que se prioriza los delitos de corrupción de mayor incidencia y con mayor grado de complejidad en su ejecución como en sus efectos.

Que, el Reglamento establece que pueden ser objeto del Decreto Legislativo N° 1180, todos los delitos contra la administración pública, no obstante, existen delitos de corrupción que afectan gravemente la función pública y el tesoro público, por lo que deben ser priorizados e individualizados, a fin que los beneficios de recompensa cumplan su fin de modo eficiente y efectivo.

Que, el Reglamento y su Directiva aprobada por Resolución Ministerial N°250-2016-IN, establecen el procedimiento para el otorgamiento del beneficio de recompensa mediante: 1) el procedimiento regular que se inicia con el contacto del ciudadano colaborador con la Policía Nacional del Perú en materia de delitos de terrorismo y delitos señalados en el artículo 9 del Reglamento; y, 2) el procedimiento de oficio, de competencia de la Policía Nacional del Perú y la Comisión Evaluadora de Recompensas contra la Criminalidad de acuerdo al artículo 22 del Reglamento para incluir a personas con requisitoria vigente.

Que, en este sentido, es necesario incorporar los delitos de corrupción grave dentro de los alcances del Decreto Legislativo Nº 1180, que establece el beneficio de recompensas para promover y lograr la captura de miembros de las organizaciones criminales, organizaciones terroristas y responsables de delitos de alta lesividad y su Reglamento aprobado mediante el Decreto Supremo N° 011-2016-PCM, a fin de establecer precisiones normativas que aporten a la lucha contra la corrupción de funcionarios.

Que, adicionalmente, mediante el Decreto Legislativo N° 1244 que fortalece la lucha contra el crimen organizado y la tenencia ilegal de armas se ha definido el tipo penal de Organización Criminal, siendo necesaria la incorporación de esta definición al Reglamento por ser la definición legal que rige la normatividad penal vigente en esta materia en concordancia con la establecida en la Ley N° 30077, Ley Contra el Crimen Organizado.

De conformidad con el numeral 8 del artículo 118 de la Constitución Política del Perú, la Ley Nº 29158, Ley Orgánica del Poder Ejecutivo y el Decreto Legislativo 1132;

DECRETA:

Artículo 1.- Aplicación del beneficio de recompensas para casos de corrupción grave.

Aplíquese el beneficio de recompensas establecido en el Decreto Legislativo Nº 1180 y su Reglamento aprobado mediante el Decreto Supremo N° 011-2016-PCM, para promover y lograr la captura de responsables de los delitos de corrupción grave cometidos por funcionarios públicos o particulares, que atentan o pueden atentar contra el correcto funcionamiento de la administración pública y/o el tesoro público, ejecutados a través del aprovechamiento indebido de la función encomendada o de los procedimientos administrativos de las entidades públicas.

Artículo 2.- Incorporación.

De conformidad con el artículo 4 del Decreto Legislativo Nº 1180, incorpórese los delitos que se encuentran previstos en el Código Penal en los artículos 382 (Concusión), 384 (Colusión simple y agravada), 387(Peculado), 389 (Malversación), 393 (Cohecho pasivo propio), 393-A (Soborno internacional pasivo), 394 (Cohecho pasivo impropio), 395 (Cohecho pasivo específico), 396 (Corrupción pasiva de auxiliares jurisdiccionales), 397 (Cohecho activo genérico), 397-A (Cohecho activo transnacional), 398 (Cohecho activo específico), 399 (Negociación incompatible o aprovechamiento indebido del cargo), 400 (Tráfico de Influencias) y 401 (Enriquecimiento Ilícito), como delitos que son materia de evaluación por la Comisión de Evaluación de Recompensas contra la Criminalidad.

Artículo 3.- Modificación

Modifíquese el artículo 4 y el inciso p) del artículo 9 del Reglamento del Decreto Legislativo Nº 1180, que establece el beneficio de recompensas para promover y lograr la captura de miembros de las organizaciones criminales, organizaciones terroristas y responsables de delitos de alta lesividad, quedando redactado de la siguiente manera:

"(…)

Artículo 4.- Definición de términos

a) Nivel.- Ubicación alcanzada por los integrantes o miembros de las organizaciones terroristas o criminales de acuerdo al grado de lesividad delictiva o peligrosidad del denunciado que permite a la Comisión respectiva evaluar los montos que se fijan por la captura de los denunciados, según reporte de la Dirección Nacional de Operaciones Policiales a través de las Unidades de la Policía Nacional del Perú especializadas.

Para efectos del presente Reglamento, los niveles son:

a.1. En caso de terrorismo: dirigentes, secretarios generales, cabecillas, jefe u otro equivalente, integrantes o miembros de grupos armados y/o de apoyo organizado.

a.2. En caso de crimen organizado: jefes, dirigentes o cabecillas e integrantes.

a.3. En caso de delitos de alta lesividad: autores, presuntos autores o partícipes de delitos de alta lesividad.

a.4. En caso de delitos de corrupción grave: autores, presuntos autores o participes de los delitos de corrupción grave.

b) Cabecilla Terrorista.- Dentro de las organizaciones terroristas es el jefe operativo responsable de la ejecución de acciones terroristas en determinadas circunscripciones conocido como:

b.1 "Mando Político": a cargo de la planificación, aprobación y/o ejecución de las acciones terroristas, así como del adoctrinamiento ideológico del grupo y/o población.

b.2 "Mando Militar": a cargo de grupos armados para aniquilamiento físico de personas.

b.3 "Mando Logístico": responsable del aprovisionamiento, almacenamiento, seguridad y resguardo de armamentos, explosivos, pertrechos y demás efectos para perpetrar el delito de terrorismo.

c) Dirigentes, Secretarios Generales, Jefes u otro equivalente en organizaciones terroristas.- Integrantes de la cúpula, comités u órganos de dirección de las organizaciones terroristas que operan en nuestro país y que poseen dominio nacional y/o regional sobre las actividades terroristas que perpetran contra el Estado y sus ciudadanos cuyas modalidades están establecidas en el Decreto Ley Nº 25475, Ley que establece la penalidad para los delitos de terrorismo y los procedimientos para la investigación, la instrucción y el juicio.

d) Integrantes o miembros de Grupos Armados y Apoyo Organizado.- Dentro de las organizaciones terroristas son aquellos que siguen las órdenes de los dirigentes, secretarios y/o cabecillas ejecutando las conductas previstas de acuerdo a la normatividad de la materia, integrando grupos o pelotones armados. Los integrantes de apoyo organizado realizan actividades de manutención y aprovisionamiento de los grupos armados, vigilancia, inteligencia, alojamiento, cobertura radial y/o celular, enlaces personales, canalización de cobro de cupos, transporte fluvial y/o terrestre, guías, reporte de la ubicación de las fuerzas del orden y toda gestión que facilite el accionar del terrorismo e impida la captura de sus integrantes.

e) Organización Criminal.- Agrupación de tres o más personas con carácter estable, permanente o por tiempo indefinido, que de manera organizada, concertada o coordinada, se repartan diversas tareas o funciones, y está destinada a cometer delitos graves establecidos en la Ley N° 30077, Ley contra el crimen organizado.

f) Jefe, Dirigente o Cabecilla en criminalidad organizada.- Persona que dirige una organización criminal o que es seguida por otras que se someten a su voluntad, tienen un papel principal o superior en la organización, interviene en la creación de la asociación delictiva, pudiendo tener el financiamiento de las operaciones delictivas. Definen, programan, supervisan, dirigen y distribuyen con autoridad propia, las funciones de quienes están a su cargo. El jefe, dirigente o cabecilla es el que preside y desempeña el papel principal de la organización.

g) Integrante o miembro de Organización Criminal.- Son aquellas personas que pertenecen a una organización criminal, están vinculadas a ella o actúan por encargo de la misma con un rol determinado previamente por el jefe, dirigente o cabecilla.

h) Delitos de Alta Lesividad.- Son delitos que por su grado de ejecución, motivación, empleo de medios o la nocividad de las consecuencias del accionar delictivo sobre los bienes jurídicos protegidos por el ordenamiento legal peruano al producir alarma, zozobra o impacto en las condiciones de convivencia armoniosa y pacífica. Estos delitos generan repercusión nacional o internacional.

h.1 Delitos de repercusión nacional: se entiende como delito de repercusión nacional cuando la acción o sus efectos generan: i) lesión o puesta en peligro de bienes jurídicos que comprometen el interés de la colectividad, generando grave alarma social; ii) grave afectación a la seguridad y/o economía nacional o a la administración de justicia o su obstaculización; o iii) cuando la actividad criminal se desarrolla simultáneamente en diferentes áreas geográficas.

h.2 Delitos de repercusión internacional: un delito tiene repercusión internacional, siempre que: i) se comete, además del territorio nacional, en otro o más Estados; ii) se comete dentro de un solo Estado, pero una parte sustancial de su perpetración, planificación, dirección o control se realiza en otro Estado; iii) se comete dentro de un solo Estado, pero entraña la participación de un grupo delictivo organizado que realiza actividades delictivas en más de un Estado; o iv) se comete en un solo Estado, pero tiene efectos sustanciales en otro Estado.

i) Delitos de Corrupción Grave.- Son aquellos delitos cometidos por funcionarios públicos o particulares, que atentan o pueden atentar contra el correcto funcionamiento de la administración pública y/o el tesoro público, ejecutados a través del aprovechamiento indebido de la función encomendada o de los procedimientos administrativos de las entidades públicas, siendo estos delitos los regulados por el Código Penal en los artículos 382 (Concusión), 384 (Colusión simple y agravada), 387(Peculado), 389 (Malversación), 393 (Cohecho pasivo propio), 393-A (Soborno internacional pasivo), 394 (Cohecho pasivo impropio), 395 (Cohecho pasivo específico), 396 (Corrupción pasiva de auxiliares jurisdiccionales), 397 (Cohecho activo genérico), 397-A (Cohecho activo transnacional), 398 (Cohecho activo específico), 399 (Negociación incompatible o aprovechamiento indebido del cargo), 400 (Tráfico de Influencias) y 401 (Enriquecimiento Ilícito).

Para efectos de lo dispuesto en el artículo 5 del Decreto Legislativo Nº 1180, se considerarán Delitos de Alta Lesividad los delitos mencionados en el párrafo anterior, aun cuando no tengan repercusión nacional o internacional.

j) Ciudadano Colaborador.- Es la persona natural que proporciona información acerca de la identidad y ubicación de un delincuente buscado por las autoridades policiales u operadores de justicia, así como de autores o presuntos autores y partícipes de uno o más delitos, a fin de posibilitar su búsqueda, captura y/o entrega, recibiendo a cambio un pago de recompensa. Dicha persona no debe estar incursa en los impedimentos establecidos en el artículo 2 del Decreto Legislativo.

En caso el ciudadano colaborador fallezca antes de ser beneficiado de la recompensa, su cónyuge, descendientes o ascendientes pueden percibir la recompensa, previa declaratoria de herederos.

k) Denunciado.- Para efectos del presente Reglamento, se entiende al jefe, dirigente, cabecilla, secretario general, cabecilla terrorista, integrante o partícipe de grupo armado o apoyo organizado de una organización terrorista o criminal, o al autor o presunto autor de delito de alta lesividad, identificado o por identificar sin capturar; cuya detención debe reunir los presupuestos legales exigidos por la normatividad nacional vigente.

l) Recompensa.- Suma de dinero que se otorga como beneficio a los ciudadanos colaboradores por parte de las Comisiones Evaluadoras, con cargo a los presupuestos institucionales de los Ministerios de Defensa e Interior o con cargo a los recursos del Fondo Especial para la Seguridad Ciudadana, regulado mediante Decreto de Urgencia N° 052-2011.

(...)

Artículo 9.- Comisión Evaluadora de Recompensas contra la Criminalidad

(...)
p) Delitos de Corrupción Grave.
(...)"

Artículo 4.- Refrendo.
El presente Decreto Supremo será refrendado por el Presidente del Consejo de Ministros, el Ministro del Interior, la Ministra de Justicia y Derechos Humanos y el Ministro de Defensa.

Dado en la Casa de Gobierno, en Lima, a los quince días del mes de marzo del año dos mil diecisiete.

PEDRO PABLO KUCZYNSKI GODARD
Presidente de la República

FERNANDO ZAVALA LOMBARDI
Presidente del Consejo de Ministros

JORGE NIETO MONTESINOS
Ministro de Defensa

CARLOS BASOMBRIO IGLESIAS
Ministro del Interior

MARÍA SOLEDAD PÉREZ TELLO
Ministra de Justicia y Derechos Humanos

1497714-1

Disponen la publicación del proyecto de Decreto Supremo que modifica el Reglamento del Libro de Reclamaciones del Código de Protección y Defensa del Consumidor, en el portal institucional del INDECOPI

RESOLUCIÓN MINISTERIAL
N° 061-2017-PCM

Lima, 13 de marzo de 2017

VISTA: La Carta N° 191-2017/PRE-INDECOPI del Presidente del Consejo Directivo del Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual – INDECOPI; y,

CONSIDERANDO:

Que, el artículo 150 de la Ley Nº 29571, Código de Protección y Defensa del Consumidor, dispone que los establecimientos comerciales deben contar con un Libro de Reclamaciones, en forma física o virtual y que el Reglamento establece las condiciones, los supuestos y las demás especificaciones para el cumplimiento de la obligación señalada en el mencionado artículo;

Que, mediante Decreto Supremo Nº 011-2011-PCM se aprobó el Reglamento del Libro de Reclamaciones del Código de Protección y Defensa del Consumidor;

Que, conforme al artículo 14 del Reglamento que establece disposiciones relativas a la publicidad, publicación de Proyectos Normativos y difusión de Normas Legales de Carácter General, aprobado por Decreto Supremo N° 001-2009-JUS, las entidades públicas dispondrán la publicación de los proyectos de normas de carácter general que sean de su competencia en el Diario Oficial El Peruano, en sus Portales Electrónicos o mediante cualquier otro medio, en un plazo no menor de treinta (30) días antes de la fecha prevista para su entrada en vigencia, salvo casos excepcionales; permitiendo que las personas interesadas formulen comentarios sobre las medidas propuestas;

Que, a través del documento de vista, el Instituto Nacional de Defensa de la Competencia y de la Protección de la Propiedad Intelectual - INDECOPI, en su calidad de Autoridad Nacional de Protección del Consumidor, ha elaborado el proyecto de Decreto

Anexo 23 Decreto Supremo que aplica el beneficio de recompensas del Decreto Legislativo Nº 1180 y su Reglamento