



Facultad Regional Tucumán

Departamento de Ingeniería en

Sistemas de Información

SAURON

SISTEMA DE INTERPRETACIÓN DE IMÁGENES PARA LA DETECCIÓN DE ARMAS

Autores

Andreozzi, Stella María

andreozzi.stella@gmail.com

Ceriana, Emanuel

emanuelceriana@gmail.com

Nacchio, Gino Bernabé

ginonacchio@gmail.com

Osatinsky, Agustin

aosatinsky@gmail.com

Tutor

Dr. Ing. Diego Lizondo

lizondo.diego@gmail.com

AÑO 2021



San Miguel de Tucumán, viernes 05 de marzo de 2021

UNIVERSIDAD TECNOLÓGICA NACIONAL

FACULTAD REGIONAL TUCUMÁN

Director del Departamento de Sistemas

A.U.S. Augusto José Nasrallah

S / D

De mi mayor consideración:

En mi carácter de Tutor del presente trabajo correspondiente a la Tesis de Grado, denominada "SAURON: SISTEMA DE INTERPRETACIÓN DE IMÁGENES PARA LA DETECCIÓN DE ARMAS", perteneciente a/los alumnos **Andreozzi, Stella María, Legajo N° 43.589 - Ceriana, Emanuel, Legajo N° 43.622 - Nacchio, Gino Bernabé, Legajo N° 43.702 - Osatinsky, Agustín, Legajo N° 43.708.**

.....

Diego Lizondo

Dr. Ing. en Sistemas de Información

.....

Nombre Director de Proyecto de Investigación

(solo completar los proyectos de Investigación)

Dejo constancia que éste se desarrolló bajo mi dirección y presto mi total conformidad para ser presentado para su evaluación, considerando que se alcanzaron satisfactoriamente los objetivos planteados para su elaboración



Agradecimientos

Desde el equipo de Sauron queremos aprovechar este espacio para darle las gracias a todos aquellos que han aportado ayuda y apoyo en todo este proceso de formación educativa y personal.

Agradecer primeramente la contención y el soporte familiar. Nuestros padres y hermanos siempre están presentes desde el primer día, dándonos fuerzas en los momentos difíciles y felicitándonos por nuestros logros.

También, dar las gracias a nuestros compañeros y amigos, los cuales tuvieron mucho que ver con el crecimiento personal y facultativo, convirtiendo a la universidad no solo en una casa de estudios sino también en un lugar de reunión, un espacio en el cual compartimos grandes momentos que quedarán para siempre en nosotros.

No queremos dejar de lado a los profesores que de manera desinteresada solventar nuestras dudas una y otra vez por más inútiles que parecieran. Agradecer el tiempo dedicado a transmitirnos sus conocimientos y darnos contención a lo largo de toda la carrera.

Haciendo énfasis a la cátedra de Proyecto Final y en especial a nuestro tutor del proyecto Germán Correa quien nos acompañó en este largo trayecto de cursado de la materia. También hay que destacar a nuestro tutor de la Universidad Diego Lizondo por su gran aporte desinteresado de tiempo y conocimiento.

Finalmente reconocer especialmente al Decano Ing. Mg. Fabián Soria y al jefe del Departamento de Sistemas Augusto Nasrallah, quienes desde el principio y desde su posición nos brindaron apoyo absoluto y contribuyeron a que podamos llegar a esta instancia.

Agustín, Emanuel, Gino y Stella



Tabla de Contenidos

Agradecimientos	3
Índice de Tablas.....	8
1. Resumen Ejecutivo.....	9
2. Tema.....	9
3. Objetivos.....	11
3.1. Objetivo general.....	11
3.2. Objetivos Específicos.....	11
4. Estado del Arte del Tema.....	11
4.1. Contexto actual en Latinoamérica, Argentina y Tucumán.....	11
4.2. Innovación Tecnológica del Proyecto	12
4.3. Impacto Social del Proyecto	12
4.4. Necesidades y/o oportunidades que justifique el desarrollo del proyecto	13
4.5. Beneficios cuantitativos y cualitativos.....	14
4.6. Tecnologías similares.....	15
5. Gestión del proyecto.....	16
5.1. Fases del proyecto.....	16
5.2. Resultados Esperados e Indicadores de Evaluación	17
5.3. Metodología de desarrollo	18
5.4. Registro de interesados.....	22
5.5. Gestión de Riesgos del proyecto.....	23
5.5.1. Identificación y análisis cualitativo.....	23
5.5.2. Plan de respuesta.....	26
5.6. Plan de Gestión de Proyecto	34
6. Desarrollo del tema	35
6.1. Modelo de negocio	35
6.2. Desarrollo del producto.....	39
6.2.1. ANÁLISIS.....	39
6.2.1.1. Especificación de Requerimientos de Software.....	39
6.2.1.2. Requerimientos Funcionales.....	44
6.2.1.3. Requerimientos No Funcionales.....	47
6.2.1.4. Modelo de Casos de Uso.....	48
6.2.1.5. Diagramas de Colaboración del Sistema	56
6.2.1.6. Casos de Uso Reales:	60
6.2.2. DISEÑO.....	62



6.2.2.1.	Documento de Arquitectura del Software	62
6.2.2.2.	Mockup.....	66
6.2.2.3.	Diagrama Entidad Relación.....	70
6.2.3.	IMPLEMENTACIÓN.....	70
6.2.3.1.	Tecnologías de Implementación.....	70
6.2.3.2.	Entrenamiento del Modelo de Implementación	74
6.2.4	PRUEBAS.....	77
6.2.4.1	Modelo de Pruebas	77
6.2.4.2	Plan de Pruebas	79
7.	Resultados Obtenidos	90
8.	Manual de Usuario	92
8.1.	Pantalla Principal.....	92
8.2.	Sección cámaras	92
8.3.	Sección Historial.....	93
8.4.	Sección contactos	94
8.5.	Disparador de Alarma.....	95
8.6.	Notificacion enviada a correo a traves de SendGrid	96
9.	Manual de Instalación.....	97
10.	Incumbencias del Ingeniero en Sistemas de Información	99
11.	Conclusiones	100
12.	Bibliografía.....	101



Índice de Figuras:

Fig. N°1. FASES DEL PROYECTO – PARTE 1.....	16
Fig. N°2. DIAGRAMA DEL PROCESO DE DESARROLLO ITERATIVO.....	18
Fig. N°3. DIAGRAMA DE METODOLOGÍA SCRUM – Lado A	19
Fig. N°4. DIAGRAMA DE METODOLOGÍA SCRUM – LADO B.....	20
Fig. N°5. METODOLOGÍA PMI	21
Fig. N°6. MODELO DE NEGOCIO CANVA N°1 – SEGMENTO DE CLIENTE “DUEÑO DE LOCAL”	36
Fig. N°7. MODELO DE NEGOCIO CANVA N°2 – SEGMENTO DEL CLIENTE: CIUDADANO.....	38
Fig. N°8. DIAGRAMA DE CASOS DE USO SISTEMA SAURÓN.....	49
Fig. N°9. DIAGRAMA DE SECUENCIA DEL SISTEMA N°1 - CASO DE USO “INICIAR SESIÓN”	51
Fig. N°10. . DIAGRAMA DE SECUENCIA DEL SISTEMA N°2 - CASO DE USO “CONFIGURAR VISTAS DE CÁMARA”	52
Fig. N°11. DIAGRAMA DE SECUENCIA DEL SISTEMA N°3 - CASO DE USO “VER HISTORIAL DE EVENTOS”	53
Fig. N°12. DIAGRAMA DE SECUENCIA DEL SISTEMA N°4 - CASO DE USO “PROCESAR IMAGEN”	54
Fig. N°13. DIAGRAMA DE SECUENCIA DEL SISTEMA N°5 - CASO DE USO “ENVIAR NOTIFICACIÓN”.....	55
Fig. N°14. DIAGRAMA DE COLABORACIÓN N°1 - CASO DE USO “INICIAR SESIÓN”	56
Fig. N°15. DIAGRAMA DE COLABORACIÓN N°2 - CASO DE USO “CONFIGURAR VISTA DE CÁMARAS”.....	56
Fig. N°16. DIAGRAMA DE COLABORACIÓN N°3 - CASO DE USO “VER HISTORIAL DE EVENTOS”	57
Fig. N°17. DIAGRAMA DE COLABORACIÓN N°4 - CASO DE USO “PROCESAR IMAGEN”	57
Fig. N°18. DIAGRAMA DE COLABORACIÓN N°3 - CASO DE USO “ ENVIAR NOTIFICACIÓN”	58
Fig. N°19. MOCK UP INTERFAZ DE USUARIO: VISTA DE CÁMARAS.....	61
Fig. N°20. MOCK UP INTERFAZ DE USUARIO: VER HISTORIAL DE EVENTOS	61
Fig. N°21. ARQUITECTURA DE SOLUCIÓN DE SAURON DONDE SE VEN LAS HERRAMIENTAS UTILIZADAS Y LA INTERACCION ENTRE ELLAS.....	62
Fig. N°22. REPRESENTACIÓN GRÁFICA DE LA ARQUITECTURA LÓGICA DE RED NEURONAL: YOLO (A. Bochkovskiy, 2020).....	64
Fig. N°23. REPRESENTACIÓN GRÁFICA DE LA ARQUITECTURA EN CAPAS DEL SISTEMA	65
Fig. N°24. MOCK UP INTERFAZ DE USUARIO: VISTA DE CÁMARAS V1.0 - EL USUARIO PODRÁ VER LAS IMÁGENES EN TIEMPO REAL CAPTADAS POR EL SISTEMA DE CÁMARAS.	66
Fig. N°25. MOCK UP INTERFAZ DE USUARIO: HISTORIAL DE ALARMAS V1.0 - EN EL HISTORIAL DE EVENTOS, EL USUARIO PODRÁ VER UN SNAPSHOT (CAPTURA DE UN FRAME ESPECÍFICO DE UN VIDEO), CON FECHA Y HORA DE CUANDO LA INTELIGENCIA ARTIFICIAL DETECTE UN ARMA.....	67
Fig. N°26. MOCK UP INTERFAZ DE USUARIO: VISTA DE ALERTA POR ARMA DETECTADA V1.0 - SE MOSTRARÁ UN CARTEL LLAMATIVO Y CON COLORES DE ALARMA, PARA PREVENIR AL CLIENTE QUE SE DETECTÓ UN ARMA Y QUE YA SE LE INFORMÓ A LOS AGENTES DE SEGURIDAD.	67
Fig. N°27. MOCK UP INTERFAZ DE USUARIO: VISTA DE CÁMARAS V2.0 - EL USUARIO PODRÁ VER LAS IMÁGENES EN TIEMPO REAL CAPTADAS POR EL SISTEMA DE CÁMARAS.	68



Fig. N°28. MOCK UP INTERFAZ DE USUARIO: HISTORIAL DE ALARMAS V2.0 - EN EL HISTORIAL DE EVENTOS, EL USUARIO PODRÁ VER UN SNAPSHOT (CAPTURA DE UN FRAME ESPECÍFICO DE UN VIDEO), CON FECHA Y HORA DE CUANDO LA INTELIGENCIA ARTIFICIAL DETECTE UN ARMA.....	69
Fig. N°29. MOCK UP INTERFAZ DE USUARIO: VISTA DE ALERTA POR ARMA DETECTADA V1.0 - SE MOSTRARÁ UN CARTEL LLAMATIVO Y CON COLORES DE ALARMA, PARA PREVENIR AL CLIENTE QUE SE DETECTÓ UN ARMA Y QUE YA SE LE INFORMÓ A LOS AGENTES DE SEGURIDAD.	69
Fig. N°30. MODELO ENTIDAD RELACIÓN DEL SISTEMA.	70
Fig. N°31. COMPARACIÓN DE RESULTADOS OBTENIDOS EN LA DETECCIÓN DE OBJETOS UTILIZANDO YOLO, RETINADET, CSP, ENTRE OTRAS. (Xiang Long, 2020)	75
Fig. N° 32. Captura de Simulación demostrando Red en Funcionamiento	76
Fig. N° 33. Collage Demostrativo del Dataset Utilizado para el Entrenamiento de la Red Neuronal.....	76
Fig. N° 34. MEAN AVERAGE PRECISION. MUESTRA EL PROGRESO DE LA PRECISIÓN DEL MODELO DURANTE EL ENTRENAMIENTO.....	90
Fig. N°35. INTERFAZ DE USUARIO PANTALLA PRINCIPAL “SAURON” – VERSIÓN FINAL.....	92
Fig. N°36. INTERFAZ DE USUARIO “SECCIÓN CÁMARAS” – VERSIÓN FINAL.....	93
Fig. N°37. INTERFAZ DE USUARIO “SECCIÓN HISTORIAL DE ALARMAS” – VERSIÓN FINAL.....	93
Fig. N°38. INTERFAZ DE USUARIO “SECCIÓN HISTORIAL DE ALARMAS” EVENTO SELECCIONADO - VERSIÓN FINAL.....	94
Fig. N°39. INTERFAZ DE USUARIO “SECCIÓN CONTACTOS” - VERSIÓN FINAL.....	94
Fig. N°40. INTERFAZ DE USUARIO “AGREGAR NUEVO CONTACTO” – VERSIÓN FINAL.....	95
Fig. N°41. INTERFAZ DE USUARIO “SECCIÓN EVENTO DETECTADO - ALARMA” – VERSIÓN FINAL.....	95
Fig. N°42. INTERFAZ DE USUARIO “NOTIFICACIÓN ENVIDADA” – VERSIÓN FINAL.....	96



Índice de Tablas

Tabla 1 - COMPARACIÓN TECNOLOGÍAS SIMILARES	15
Tabla 2 - RESULTADOS ESPERADOS.....	17
Tabla 3 - LISTA DE INTERESADOS.....	22
Tabla 4 - TABLA DE RIESGOS IDENTIFICADOS.....	25
Tabla 5 - PLAN DE RESPUESTA RIESGO 1: INTERPRETACIÓN ERRÓNEA DE REQUERIMIENTOS.....	26
Tabla 6 - PLAN DE RESPUESTA RIESGO 2: Corrección de Documentos	27
Tabla 7 - PLAN DE RESPUESTA RIESGO 7: Dataset de entrenamiento.....	28
Tabla 8 - PLAN DE RESPUESTA RIESGO 12: Cámaras de baja resolución.....	29
Tabla 9 - PLAN DE RESPUESTA RIESGO 14: TRANSMISION DE IMAGEN LENTA	30
Tabla 10 - PLAN DE RESPUESTA RIESGO 16: Mal funcionamiento de Componentes.....	31
Tabla 11 - PLAN DE RESPUESTA RIESGO 18: MAL FUNCIONAMIENTO DE LA RED NEURONAL	32
Tabla 12 - PLAN DE RESPUESTA RIESGO 20: CORRECCIÓN DE DOCUMENTACIÓN	33
Tabla 13 - PLAN FINANCIERO ESTIMADO PARA DESARROLLAR SAURON	34
Tabla 14 - PERSONAL INVOLUCRADO N°1 – LÍDER DEL PROYECTO.....	40
Tabla 15 - PERSONAL INVOLUCRADO N°2 - DESARROLLADOR.....	40
Tabla 16 - PERSONAL INVOLUCRADO N°3 – DESARROLLADOR.....	41
Tabla 17 - PERSONAL INVOLUCRADO N°4 - ANALISTA	41
Tabla 18 - CARACTERÍSTICAS DE LOS USUARIOS QUE ADMINISTRARAN EL SISTEMA SAURON.....	43
Tabla 19 - REQUERIMIENTO FUNCIONAL N°1: Interpretación de Imagen.....	44
Tabla 20 - REQUERIMIENTO FUNCIONAL N°2: DETECCIÓN DE ARMAS POR RED NEURONAL.....	44
Tabla 21 - REQUERIMIENTO FUNCIONAL N°3: ACTIVACIÓN DE ALARMA.....	45
Tabla 22 - REQUERIMIENTO FUNCIONAL N°4: PROCESAMIENTO LOCAL DE LAS IMÁGENES.....	45
Tabla 23 - REQUERIMIENTO FUNCIONAL N°5: GENERACIÓN DE HISTORIAL DE EVENTOS.....	46
Tabla 24 - REQUERIMIENTO FUNCIONAL N°6: PRESENTACIÓN VISTA DE CÁMARAS.....	46
Tabla 25 - REQUERIMIENTO NO FUNCIONAL N°1: EXACTITUD DE DETECCIÓN DE ARMA	47
Tabla 26 - REQUERIMIENTO NO FUNCIONAL N°2: CALIDAD DE IMÁGENES TOMADAS POR LAS CÁMARAS	47
Tabla 27 - REQUERIMIENTO NO FUNCIONAL N°3: ENTRENAMIENTO DE RED NEURONAL.....	47
Tabla 28 - REQUERIMIENTO NO FUNCIONAL N°4: CAMARAS CON STREAMING.....	48
Tabla 29 - REQUERIMIENTO NO FUNCIONAL N°5: TIEMPO DE ENVÍO DE NOTIFICACIÓN.....	48



1. Resumen Ejecutivo

Nuestro proyecto consiste en la realización de una aplicación que permita mediante la utilización de técnicas de visión computacional y mecanismos de Inteligencia Artificial poder detectar situaciones de emergencia con armas de fuego en tiempo real y avisar a las autoridades pertinentes para una rápida resolución del incidente. Hoy en día los agentes de las fuerzas de seguridad son los encargados de observar los videos captados por las cámaras para poder prevenir y detectar situaciones de violencia.

Según estudios realizados, el ser humano pierde entre el 50% y el 90% de su percepción visual luego de 20 minutos de supervisión continua, debido a la sobrecarga visual. Nuestro sistema será un aporte considerable para los agentes de seguridad en estos aspectos.

El principal objetivo con este proyecto es contribuir a mejorar la seguridad de las personas que adquieran nuestro producto y a su vez colaborar con los agentes de seguridad a que puedan hacer frente de una manera más veloz y eficiente ante las distintas situaciones de emergencia que se puedan presentar. Además, transferir al entorno profesional el conocimiento aprendido en esta larga trayectoria.

2. Tema

Lo que se propone es el desarrollo de software con inteligencia artificial para el cuidado del ciudadano, ayudando a detectar de manera automática los robos a mano armada, alertando a las autoridades en caso de que ocurran. Así logramos un tiempo de respuesta menor y efectivo, disminuyendo los casos de violencia y de robo en la ciudad.

Actualmente sólo esta tecnología estará limitada a detectar armas de fuego, más precisamente con pistolas y revólveres. A mediano plazo se planea ampliar la detección para cualquier tipo de armas de fuego y armas blancas.

El sistema será capaz de interpretar imágenes captadas desde cámaras de seguridad mediante el uso de redes neuronales convolucionales con el fin de prevenir situaciones de emergencia tales como robos con armas. Una vez detectada el arma en dicha situación de violencia el sistema emitirá una alarma a la entidad de seguridad que utilice el sistema para así disminuir los tiempos de respuesta de la fuerza de seguridad.



Alcance del Producto

El sistema de interpretación de imágenes para la detección de armas SAURON es un sistema basado en una inteligencia artificial que permite la detección de armas en fuego de pequeño calibre (pistolas y revólveres) en imágenes tomadas en tiempo real por cámaras de seguridad. Si se detecta la presencia de algún arma anteriormente mencionada, el sistema emitirá una alarma visual y sonora.

Para el entrenamiento de la red neuronal convolucional se utilizará un Dataset de 19.000 imágenes. Las imágenes se encuentran a su vez etiquetadas en archivos TXT con el formato YOLO, lo que quiere decir que se señala con coordenadas (x,y) en qué posición de la imagen se encuentra el arma.

El sistema de procesamiento funcionará en su totalidad sin la necesidad de una conexión a internet, se hará utilizando un servidor de manera local. La conexión a internet solo será necesaria para emitir la alerta hacia puntos remotos.

El sistema será diseñado para trabajar con cámaras que se encuentren cerca del procesador principal respetando los principios de "Edge Computing" (W. Shi, 2016) para evitar el streaming de video. Así mismo, las cámaras deberán poseer una resolución igual o superior a 720p (Alta definición), para un buen reconocimiento de las imágenes.

Para este proyecto no utilizaremos un DVR, las posibles cámaras se conectarán directamente al procesador con Plug & Play, el procesamiento será en tiempo real y las imágenes no se guardarán. El procesamiento será de manera local utilizando el procesador NVIDIA Jetson Nano que está pensado para ser implementado en robots y otros dispositivos impulsados por Inteligencia Artificial (IA).

El sistema será implementado en un entorno de prueba en el laboratorio GITIA en la Universidad Tecnológica Nacional Facultad Regional Tucumán. Se realizarán distintas simulaciones con armas de juguete. Cuando las pruebas con simulaciones hayan pasado de manera satisfactoria, el sistema ya estará en condiciones de implementarse en un entorno real, para el cual se eligió la empresa Discar SC de venta de insumos tecnológicos ubicada la calle Córdoba 429 en la provincia de Tucumán.



3. Objetivos

3.1. Objetivo general

Mejorar la seguridad y el bienestar de la sociedad utilizando Edge-Computing en un sistema para la interpretación de imágenes capturadas con el fin de detectar situaciones de violencia con armas de fuego.

3.2. Objetivos Específicos

- Desarrollar un sistema basado en inteligencia artificial para la detección de situaciones de violencia con armas capaz de emitir alarmas en tiempo real cuando estas sucedan, hasta noviembre de 2019.
- Reducir los tiempos de respuesta de las entidades de seguridad al menos un 50%.
- Diseñar un modelo de redes neuronales artificiales mediante métodos de aprendizaje supervisado en los próximos 2 meses.
- Entrenar el modelo de redes neuronales artificiales diseñado para la detección de armas en los siguientes 4 meses.
- Presentar al menos un paper sobre avances del trabajo en congresos, compartir experiencias con más investigadores y transferir al medio lo aprendido, en el próximo año.
- Conseguir fondos para el desarrollo del proyecto antes de finalizar el año.

4. Estado del Arte del Tema

4.1. Contexto actual en Latinoamérica, Argentina y Tucumán

Se han realizado diversas investigaciones con el objetivo de abordar el problema de la detección de objetos e interpretación de imágenes. Kaiming He et al. (K. He, 2017) proponen utilizar un método llamado Mask Region-CNN, como extensión al método Fast R-CNN (Girshick, 2015) para la detección y posterior segmentación de objetos en una imagen. Tsung-Yi Lin et al. (T. Lin, 2017) proponen utilizar un método llamado “Feature Pyramid Networks” para la detección de objetos de diferentes escalas de tamaño. Georgia Gkioxari et al. (G. Gkioxari, 2017) crearon un modelo que llamaron “InteractNet” que a partir de la detección de un humano y un objeto busca inferir lo que está sucediendo en la imagen.



En China son pioneros en el tema, un documental realizado por la cadena de noticias BBC demuestra su sistema de seguridad llamado “SkyNet”, en donde con un conjunto de 20 millones de cámaras, permite identificar el sexo de la persona, edad, color de piel, características de la ropa y hasta rasgos únicos. Además, es capaz de reconocer vehículos por marca, modelo, color, así como tipo de vehículo y saber si está siendo conducido o aparcado. Con esto, sólo es necesario introducir una fotografía de un sospechoso y el sistema mostrará su ubicación en cuestión de segundos.

En una investigación publicada en JAMA en el año 2018 (I., 2018) llamada “Global Mortality From Firearms, 1990-2016” se estimó que hubo en 2016 entre 195,000 y 276,000 muertes ocasionadas por armas de fuego globalmente, en donde la mayoría fueron homicidios. En Argentina por ejemplo la tasa de homicidios con armas de fuego por cada 100,000 habitantes fue de un 3.3 mientras que la de Estados Unidos de un 4.0. En Brasil la cifra asciende a los 19.4 y en Venezuela, donde la crisis política, económica y social no da tregua, de un alarmante 38.7.

4.2. Innovación Tecnológica del Proyecto

Actualmente los agentes de las fuerzas de seguridad son los encargados de observar los videos captados por las cámaras para poder prevenir y detectar situaciones de violencia. Por lo tanto, se pierde efectividad ya que el ser humano pierde entre el 50% y el 90% de su percepción visual luego de 20 minutos de supervisión continua, debido a la sobrecarga visual (número de pantallas a monitorear, cantidad de actividad por pantalla), la vigilancia (capacidad de permanecer alerta y prestar atención durante un largo período de tiempo), interrupciones en las tareas (responder a una llamada telefónica) y ruido de fondo (conversación entre colegas). Otros problemas encontrados son las particularidades del ser humano como lo es su perfil, aburrimiento, estrés, desinterés.

Por lo tanto, existe la necesidad de soluciones automatizadas para gestionar el proceso de vigilancia. Estas soluciones reducirían estas ineficiencias y errores asociados con el monitoreo utilizando operadores humanos.

4.3. Impacto Social del Proyecto

En la actualidad, los agentes de las fuerzas de seguridad son los encargados de observar los videos captados por las cámaras de seguridad, para poder así prevenir y detectar situaciones de violencia. Es de público conocimiento que después de un largo periodo de tiempo observando un video aburrido o realizar tareas tediosas, el ser humano pierde la concentración que poseía en un comienzo.



Algunos de los factores que pueden distraer a la persona encargada de monitorear las cámaras de seguridad podrían ser:

- La cantidad de pantallas. (Para poder cubrir diversos ángulos y no dejar puntos ciegos, se suelen utilizar una gran cantidad de cámaras dispuestas de manera estratégica).
- Interrupciones en las tareas. (Responder a una llamada telefónica, satisfacer necesidades fisiológicas, ir al baño, comer, etc.).
- Ruido de fondo. (Una conversación entre colegas, gente entrando o saliendo del establecimiento, el tráfico en la calle, son eventos que pueden causar una disminución en la concentración de la persona encargada de la vigilancia).

Teniendo en cuenta los factores propuestos anteriormente y algunas particularidades ligadas directamente con las emociones, como lo pueden ser el aburrimiento, estrés y desinterés, queda demostrado que el ser humano no es el mejor recurso para controlar y supervisar la seguridad de un establecimiento.

A raíz de este problema y a la par de la creciente evolución de las nuevas tecnologías, como lo es la Inteligencia Artificial, surgen soluciones automatizadas que resultan ser muy eficientes en la gestión del proceso de vigilancia. Estas soluciones permiten reducir en gran medida, ineficiencias y errores en el monitoreo realizado por operadores humanos.

4.4. Necesidades y/o oportunidades que justifique el desarrollo del proyecto

Las tasas de criminalidad causadas por las armas son muy preocupantes en muchos lugares del mundo, especialmente en países donde la posesión de armas es legal o conseguir una en el mercado negro es relativamente fácil. Un camino para reducir este tipo de hechos es mediante una detección temprana que permitiría a los agentes de seguridad actuar de una manera más rápida. Actualmente, la mayoría de los sistemas de vigilancia convencionales necesitan de presencia humana para la detección de un hecho particular. En un paper (J. Suss, 2015) realizado en Canadá se han identificado problemas y fenómenos relevantes relacionados a los operadores de CCTV como la sobrecarga visual (ej., número de pantallas a monitorear, cantidad de actividad por pantalla), la vigilancia (es decir, la capacidad de permanecer alerta y prestar atención durante un largo período de tiempo), interrupciones en las tareas (por ejemplo, responder a una llamada telefónica) y ruido de fondo (por ejemplo, conversación entre colegas). Otros problemas que han encontrado consisten en las particularidades del ser humano como lo es su perfil, aburrimiento, etc.



Por lo tanto, existe la necesidad de soluciones automatizadas para gestionar el proceso de vigilancia. Estas soluciones reducirían estas ineficiencias y errores asociados con el monitoreo utilizando operadores humanos.

En los últimos años el Deep Learning y las redes neuronales convolucionales (CNN) en particular han logrado resultados superiores a todos los métodos clásicos de aprendizaje automático en la clasificación, detección y segmentación de imágenes en varias aplicaciones. Es por eso por lo que se desea desarrollar un sistema capaz de interpretar independientemente imágenes captadas desde cámaras de seguridad mediante el uso de redes neuronales convolucionales con el fin de detectar situaciones de emergencia tales como robos para mejorar la seguridad y bienestar de la sociedad y disminuir los tiempos de respuesta de las autoridades.

Cabe destacar que el sistema podría implementarse con cámaras que ya se encuentran instaladas siendo de gran utilidad para los gobiernos (que cuentan con cámaras de seguridad en la vía pública) o un usuario final en su casa o empresa.

4.5. Beneficios cuantitativos y cualitativos

Beneficios Cualitativos:

- Motivan la realización de este trabajo los crecientes índices de inseguridad en nuestro país y en el mundo, representados por delitos comunes y terrorismo lo que genera nuestro deseo de desarrollar un instrumento que mejore la respuesta en tiempo y forma de los organismos responsables.
- Mejorar la seguridad y bienestar de la ciudadanía utilizando técnicas de inteligencia artificial.

Beneficios Cuantitativos:

- Ahorro de costes en horas humanas sin perder efectividad (el ser humano pierde entre el 50% y el 90% de su percepción visual luego de 20 minutos de supervisión continua), la precisión (detalles que se escapan al ojo humano, pueden ser detectados con un software con total exactitud) y la mayor capacidad de análisis: la inteligencia artificial permite una serie de funcionalidades de mayor alcance que los sistemas convencionales.
- Reducción de tiempos de respuestas de agentes de seguridad.



4.6. Tecnologías similares

Para el caso del servicio que ofrece nuestra empresa, no existe una competencia que se iguale en características y tecnologías utilizadas, pero hay varias otras que son líderes en seguridad privada, no automatizada, que ofrecen un servicio a un costo menor al nuestro, por lo cual, si el dinero que el cliente desea gastar es prioridad, las mismas pueden elegidas por sobre nosotros y convertirse en competencia.

Algunas de estas empresas a nivel local son:

Nombre	Status	SUAT	SAURON
Propuesta de Valor	Una empresa familiar dedicada a servicios de seguridad privada, caracterizada por cubrir las necesidades de sus clientes de manera personalizada, responsable y eficaz.	Empresa especializada en asuntos de seguridad que brinda servicios específicos del área las 24 horas, los 365 días del año; Asesoramiento en la prevención y solución de problemas empresariales y ejecución de acciones o técnicas de distinto tipo.	Nuestra empresa Sauron ofrece un sistema de cámaras de seguridad automatizado que permite la comunicación en tiempo real con las fuerzas de seguridad ante cualquier situación de peligro que comprometa la integridad de las personas que asistan a su local, empresa, organización.
Seguridad Física	SI	SI	NO
Monitoreo de Alarma	SI	SI	SI
Tiempo de Detección (en minutos)	<3	<2	Inmediata
Sistema Automatizado	NO	NO	SI
Dashboard para control de Historial y Eventos	NO	NO	SI

TABLA 1 - COMPARACIÓN TECNOLOGÍAS SIMILARES

5. Gestión del proyecto

5.1. Fases del proyecto

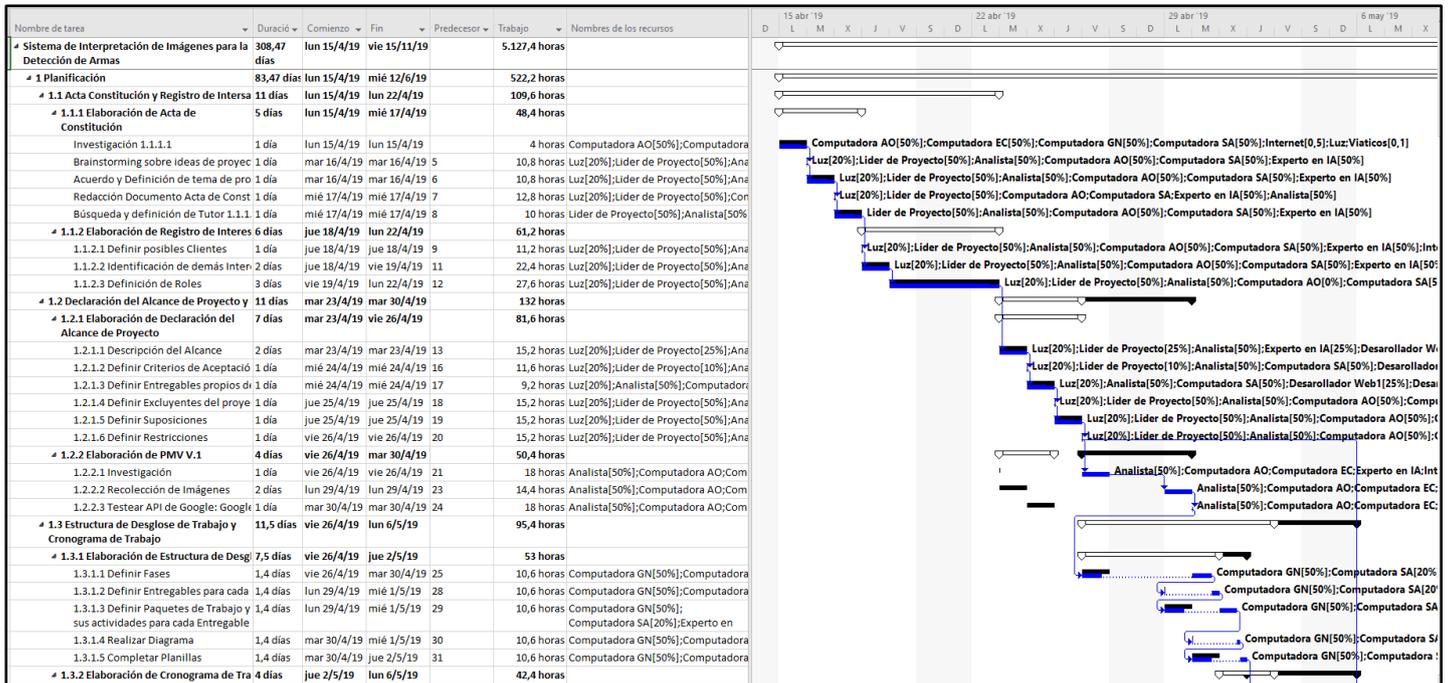


FIG. N°1. FASES DEL PROYECTO – PARTE 1



5.2. Resultados Esperados e Indicadores de Evaluación

OBJETIVOS DE PROYECTO (TIEMPO, COSTO, CALIDAD)		
Valor	Métrica	Atributo
Detectar un arma de fuego con exactitud del 80% en tiempo real	Porcentaje	80%
Reducir el tiempo de respuesta de las entidades de seguridad en al menos un 50%	Porcentaje	50%
Diseñar un modelo de redes neuronales artificiales mediante métodos de aprendizaje supervisado en los próximos 2 meses.	Tiempo	2 meses
Deberá activarse la alarma visual y sonora para notificar la detección del arma, cuando esta se produzca en al menos un 90%	Porcentaje	90%
Disminuir la cantidad de detecciones falsas positivas en un 80%	Porcentaje	80%
Reducir los costos operativos para el cliente en un 25%	Porcentaje	25%

TABLA 2 - RESULTADOS ESPERADOS

5.3. Metodología de desarrollo

Para la evolución de nuestro proyecto optamos por seguir una metodología ágil de desarrollo debido a la gran incertidumbre que se nos presentó y la posibilidad de realizar pequeñas iteraciones con entregables funcionales que nos garanticen que los requerimientos del producto se estén cumpliendo. La metodología colaborativa ágil divide los grandes proyectos en pequeños incrementos manejables o "sprints", por lo general de dos semanas de duración. Fomenta la experimentación y el uso de proyectos pequeños e iteraciones rápidas para facilitar la resolución rápida de problemas. En un proceso ágil, examina su progreso con frecuencia y puede cambiar de dirección rápidamente si un curso de acción elegido no está dando resultados. Trabaje rápidamente, demuestra valor y siga adelante.

El proceso de desarrollo iterativo se puede adaptar al desarrollo de la IA utilizando el marco que se describe en la figura siguiente.

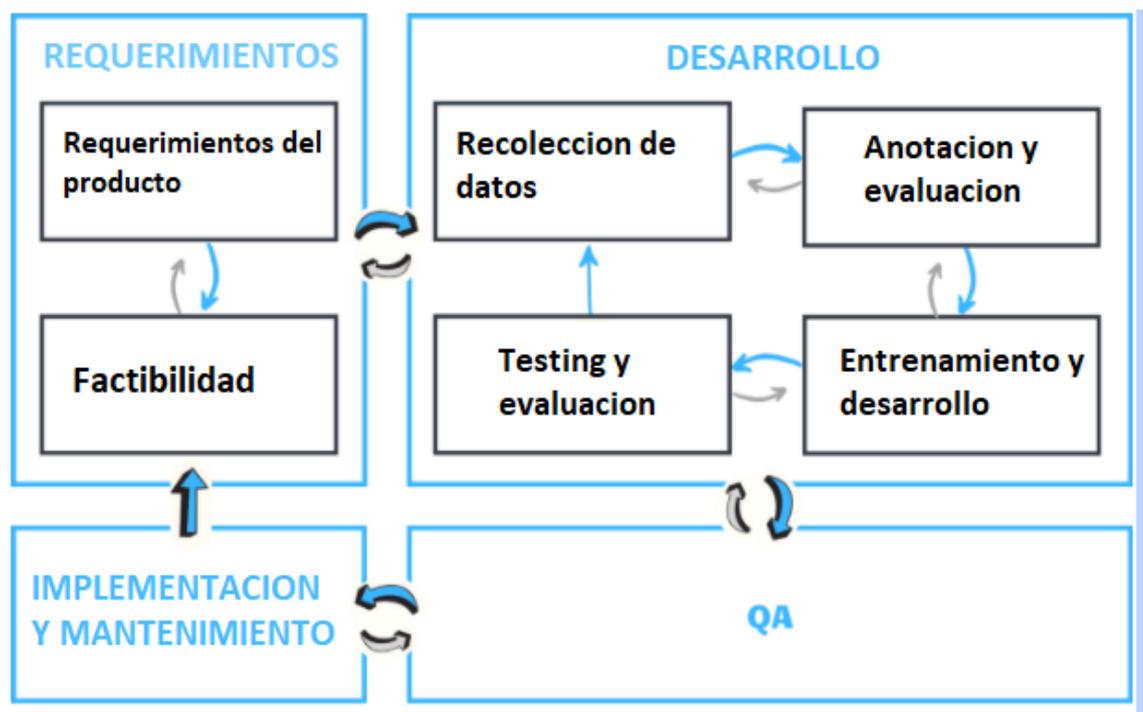


FIG. N°2. DIAGRAMA DEL PROCESO DE DESARROLLO ITERATIVO¹

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas

¹ <https://towardsdatascience.com/an-agile-framework-for-ai-projects-6b5a1bb41ce4?gi=5cc3a95aac6>

prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

En la siguiente grafica se observa un ciclo básico de desarrollo guiado bajo la metodología scrum.

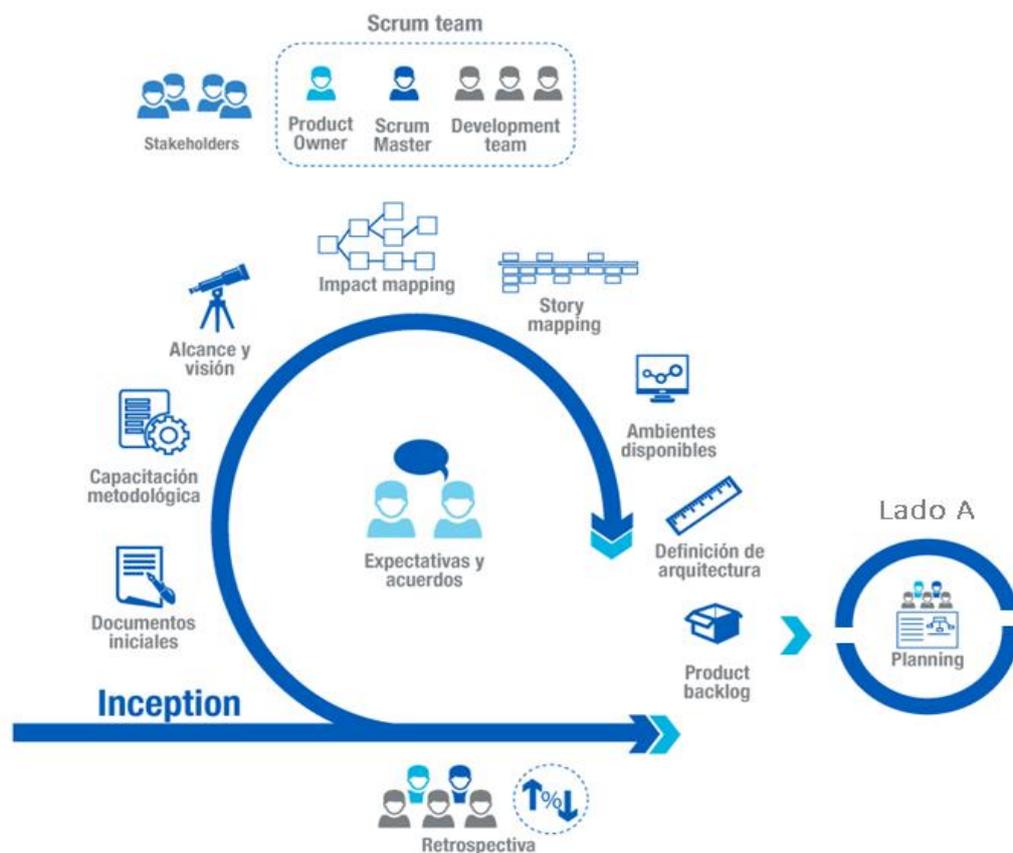


FIG. N°3. DIAGRAMA DE METODOLOGÍA SCRUM – LADO A

Además de SCRUM, seguimos las buenas prácticas presentes en la Guía del PMBOK. La misma define un proceso como "un conjunto de acciones y actividades interrelacionadas que se realizan para lograr un conjunto específico de productos, resultados o servicios", que es simplemente una forma de transformar una entrada en una salida utilizando herramientas y técnicas probadas que pueden ayudar a impulsar el progreso de principio a fin. Los procesos sirven como hoja de ruta para mantener el proyecto en la dirección correcta. Los buenos procesos se basan en principios sólidos y prácticas comprobadas que son extremadamente importantes para garantizar el éxito de un proyecto. Estos procesos pueden ayudar a minimizar la confusión y la incertidumbre entre el director del proyecto y las partes interesadas del proyecto.

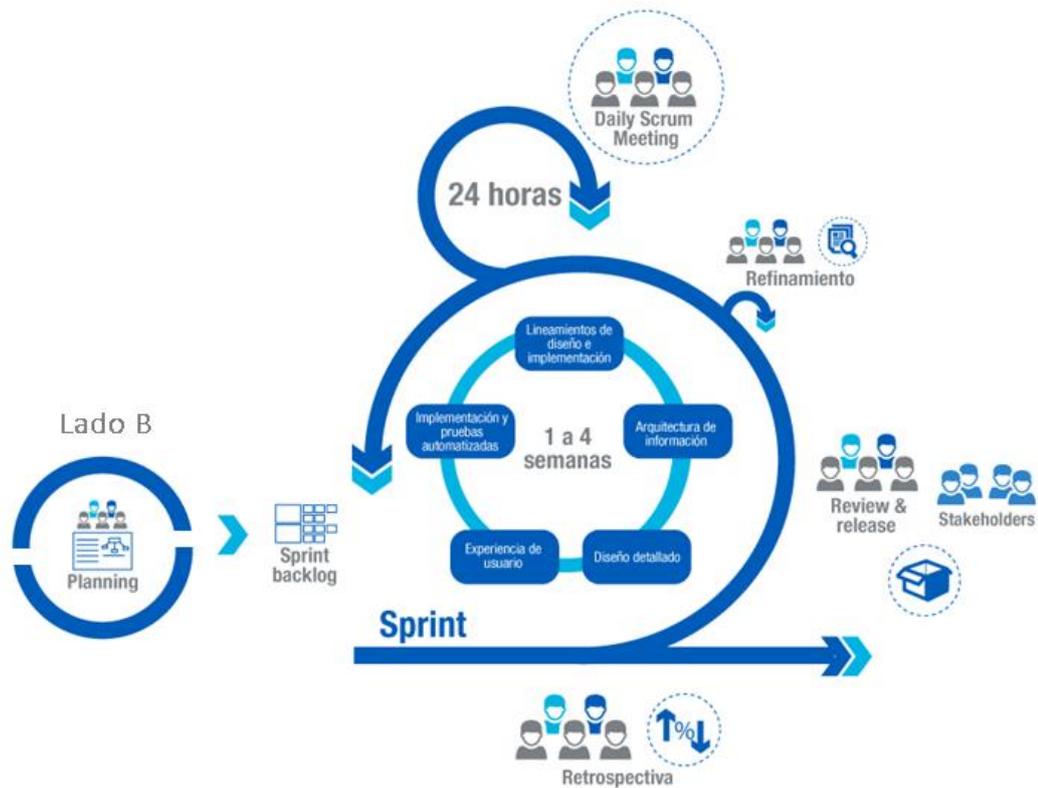


FIG. N°4. DIAGRAMA DE METODOLOGÍA SCRUM – LADO B



En la siguiente gráfica se observa un resumen de la metodología PMI.



PMI propone un modelo específico para la ejecución de los proyectos



Estos procesos de Dirección de Proyectos se aplican en todos los sectores industriales. Existe un acuerdo general, entre los Project Manager a nivel internacional, de que la aplicación de estos procesos aumenta las posibilidades del éxito en los Proyectos. La tabla de procesos de dirección de Proyectos de Pmbok PMI, en su 6ª versión, consta de 49 procesos. Todos los procesos constan de tres elementos para conseguir un resultado, un entregable, etc. Estos son: inputs (ingresos), tools & techniques (recursos que utilizaremos) y outputs (egresos).



Procesos Directivos vs Procesos Productivos

Los procesos productivos del Proyecto son procesos orientados a crear el producto, servicio o resultado del Proyecto. Los procesos directivos del Proyecto están orientados a coordinar todo el trabajo del proyecto, incluyendo los procesos productivos.



El Project Manager

No siempre deben aplicarse los mismo Procesos. Los procesos de dirección de Proyectos del PMBoK no deben aplicarse siempre en todos los Proyectos. El Director del Proyecto debe elegir los procesos de Dirección de Proyecto apropiados para su Proyecto en concreto. En el Master en Project Management hacemos especial énfasis en que un Project Manager no es un técnico, es un metodólogo y forma parte de sus funciones la toma continua de decisiones sobre su Proyecto



Procesos de inicio

Define y autoriza una fase del Proyecto, o define y autoriza el Proyecto en su globalidad.



Procesos de ejecución

Integra a las personas y a los recursos para llevar a cabo el plan de gestión del Proyecto.



Procesos de Cierre

El grupo de procesos de cierre del Proyecto. FORMALIZA la aceptación del producto, servicio o resultado del Proyecto por parte del cliente y a su entera satisfacción. Se ocupa de terminar ordenadamente el Proyecto o una fase del mismo.

Procesos de Planificación

Define y refina los objetivos del Proyecto. Planifica las acciones requeridas para lograr los objetivos y el alcance pretendido del Proyecto.

Procesos de Monitoreo y Control

Mide y supervisa el avance del Proyecto. Se busca identificar las variaciones respecto del plan de gestión del Proyecto.

Fig. N°5. METODOLOGÍA PMI²

² <https://www.studocu.com/pe/document/universidad-tecnologica-del-peru/estadistica-descriptiva-y-probabilidades/informe/la-direccion-de-proyectos/9319779/view>

5.4. Registro de interesados

Información de Identificación		Información de Evaluación	Tipo de Interesado	
Nombre y Apellido	Rol en el proyecto	Principales Expectativas	interno/externo	Defensor / Crítico / Neutral
Maley, Claudio	Stakeholder	Que sea implementado el sistema	Externo	Defensor
Nasrallah, Augusto José	Stakeholder	Que aprobemos la materia	Externo	Neutral
Lizondo, Diego	Tutor	Que realicemos el sistema propuesto	Interno	Crítico, Defensor
Andreozzi, Stella María	Documentador	Aprobar, realizar el sistema e implementarlo	Interno	Crítico
Ceriana, Emanuel	Desarrollador	Aprobar, realizar el sistema e implementarlo	Interno	Crítico
Nacchio, Gino Bernabé	Líder	Aprobar, realizar el sistema e implementarlo	Interno	Crítico
Osatinsky, Agustín	Desarrollador	Aprobar, realizar el sistema e implementarlo	Interno	Crítico
UTN - FRT	Stakeholder	Apoyar la ejecución del proyecto	Externo	Defensor
Organizaciones privadas	Stakeholder	Que sea implementado el sistema	Externo	Neutral
Profesores PF	Stakeholder	Finalización del proyecto en tiempo y forma	Externo	Crítico

TABLA 3 - LISTA DE INTERESADOS

5.5. Gestión de Riesgos del proyecto

5.5.1. Identificación y análisis cualitativo

EVALUACIÓN DE RIESGOS										
Identificación			Calificación	Cuantificación				Estado	Riesgo Residual	
ID	# WBS	Descripción Evento de Riesgo		Impacto	Consecuencias	Prob.	VME		Nueva Prob.	Nuevo VME
1	1.1.2.2	Mal Interpretación de los Requisitos	Inaceptable	5	Costo, Alcance, Tiempo y Calidad	40%	2	Finished		0,00
2	1.1.1.2	Mala decisión del Alcance	Inaceptable	5	Alcance y Tiempo	50%	3	In progress		
3	1.1.1.4	Mala Estimación de los Costos	Aceptable con atención	3	Costo y Tiempo	40%	1,20	In progress		0,00
4	1.1.1.4	Errores en Estimación de las Fechas de Presentación de Entregables	Aceptable con atención	3	Costo y Alcance	50%	2	In progress		0,00
5	1.1.2.1	Poca Predisposición del Cliente para Entrevistas	Aceptable con atención	3	Costo, Alcance, Tiempo y Calidad	50%	1,50	In progress		0,00



6	1.1.3.1	Falla en la Identificación de Riesgos Importantes	Aceptable con atención	4	Tiempo, Costo	20%	1	In progress	0,00
7	1.2.1.2 1.2.2.3 1.2.2.5 1.2.3.1	No Encontrar un Dataset de Imágenes en condiciones	Inaceptable	4	Tiempo, Calidad y Costo	70%	2,80	Finished	0,00
8	1.2.1.1	No encontrar información relevante para iniciar el desarrollo	Aceptable	2	Tiempo, Calidad	30%	1	Finished	0,00
9	1.2.2.4 1.2.2.6 1.2.3.2	El Algoritmo de entrenamiento no entrenó como se pensó	Aceptable con atención	3	Tiempo, Calidad, Costo	50%	1,50	In progress	0,00
10	1.2.2.5	Sobre entrenamiento de Red Neuronal Convolutiva	Aceptable con atención	3	Tiempo, Calidad, Costo	50%	2	Open	0,00
11	1.2.3.2	La red devuelve valores negativos al ser testeada	Aceptable con atención	2	Tiempo, Costo	70%	1,40	Open	0,00
12	1.2.4.1	Calidad de Imágenes de las Cámaras es inferior a la establecida (720p)	Inaceptable	5	Costo, Alcance, Tiempo y Calidad	30%	2	Open	0,00
13	1.2.2.6	El nivel de Interpretación de Imágenes es menor a lo requerido	Aceptable con atención	2	Tiempo, Costo y Calidad	70%	1,40	Open	0,00



14	1.2.4.3	La velocidad de transmisión de las imágenes es muy lenta (no en tiempo real)	Inaceptable	5	Costo, Alcance, Tiempo y Calidad	70%	4	Open	0,00
15	1.2.2.7	Interfaz de Usuario desaprobada por el Cliente	Aceptable	1	Tiempo y Costo	20%	0,20	Open	0,00
16	1.3.1.2 1.2.2.8 1.2.3.2 1.3.1.2	El hardware no funciona	Aceptable con atención	4	Tiempo, Costo, Calidad	40%	2	Open	0,00
17	1.3.2.1	Pruebas de Integración fallan con un número elevado de errores	Inaceptable	5	Tiempo, Costo, Calidad	5%	0,25	Open	0,00
18	1.3.2.3	Red Neuronal falla en la Simulación	Inaceptable	5	Tiempo, Costo, Calidad	40%	2,00	Open	0,00
19	1.3.3.1	Manual de Usuario incomprensible por el usuario	Aceptable	2	Tiempo, Calidad y Costo	30%	0,60	Open	0,00
20	1.3.3.3	Él informa final no es aprobado	Inaceptable	5	Costo, Alcance, Tiempo y Calidad	70%	3,50	Open	0,00

TABLA 4 - TABLA DE RIESGOS IDENTIFICADOS



5.5.2. Plan de respuesta

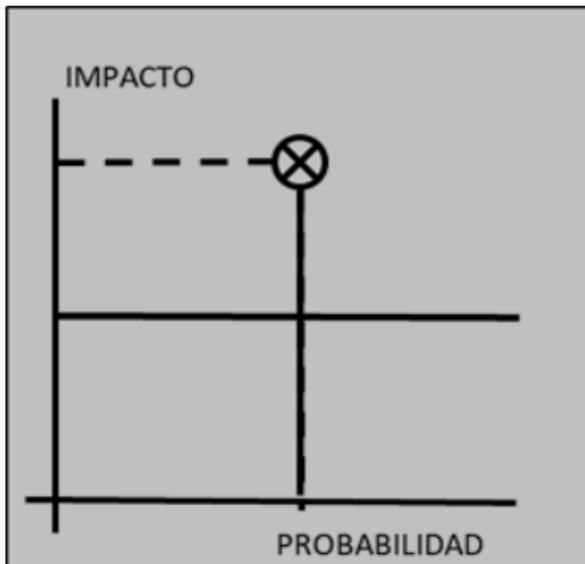
Riesgo # 1			
Descripción detallada del evento de riesgo			
<p>Al momento de las entrevistas, el cliente expresa sus deseos en cuanto al sistema, las necesidades y sus primeras inquietudes. Como equipo al analizar los requerimientos, la interpretación de los mismos es diferente a la del cliente.</p>			
Acciones para Evitar, Mitigar o Transferir	Evitar	Mitigar	Transferir
Realizaremos más entrevistas para verificar los requerimientos.	X		
Incluir preguntas abiertas en la entrevista, para dar más libertad al cliente.		X	
Incluir un experto en la materia al momento de las entrevistas			X
Acciones de Contingencia			
1. Planificar nuevas entrevistas y grabarlas para tener evidencias.			
2. Revisar las preguntas, para mejorar la estructura de entrevista.			

TABLA 5 - PLAN DE RESPUESTA RIESGO 1: INTERPRETACIÓN ERRÓNEA DE REQUERIMIENTOS

Riesgo # 2

Descripción detallada del evento de riesgo

Al redactar el documento de Alcance, no planteamos correctamente lo que nuestro sistema hará, las restricciones y suposiciones del mismo. Por lo tanto el entregable no es aprobado y necesitará revisión hasta que el tutor lo considere acertado.



Acciones para Evitar, Mitigar o Transferir	Evitar	Mitigar	Transferir
Asistir a consulta con nuestro tutor del proyecto.	x		
Tener en cuenta las correcciones del tutor de la catedra, al momento de rehacer el entregable.		x	

Acciones de Contingencia

1. Replantear el alcance con nuestro tutor.

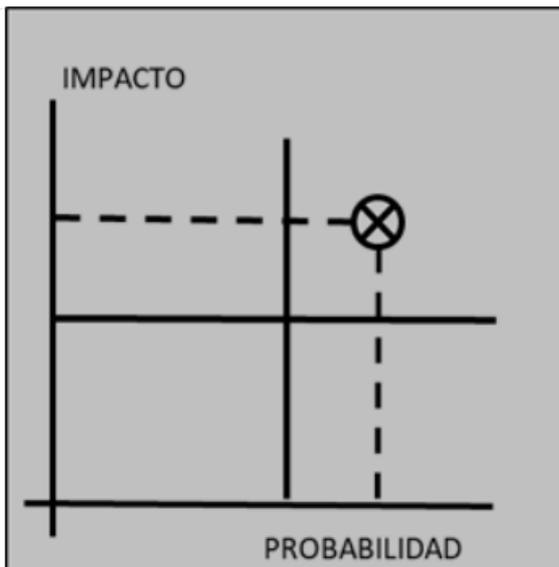
A43

TABLA 6 - PLAN DE RESPUESTA RIESGO 2: CORRECCIÓN DE DOCUMENTOS

Riesgo # 7

Descripción detallada del evento de riesgo

Buscar en internet un dataset de imágenes con armas gratuito para entrenar la red neuronal y con imágenes de calidad y variedad, pero no encontrarlo



Acciones para Evitar, Mitigar o Transferir	Evitar	Mitigar	Transferir
Realizar un dataset propio.		X	

Acciones de Contingencia

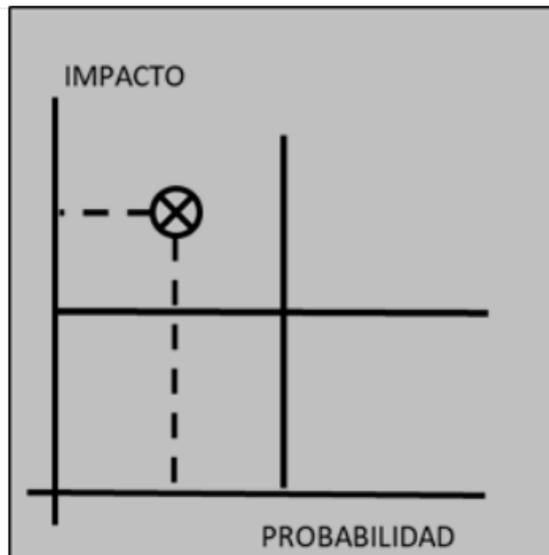
1. Buscar un dataset pago.
2. Recorrer la web para encontrar imágenes de armas.

TABLA 7 - PLAN DE RESPUESTA RIESGO 7: DATASET DE ENTRENAMIENTO

Riesgo # 12

Descripción detallada del evento de riesgo

Como definimos en nuestro alcance, las cámaras deben captar imágenes en alta definición para que puedan ser interpretadas por la red, por lo tanto, si las cámaras no poseen esta resolución, la red neuronal fallará al momento de detectar las armas.



Acciones para Evitar, Mitigar o Transferir	Evitar	Mitigar	Transferir
Chequear que las cámaras ya instaladas graben en la definición establecida.	X		

Acciones de Contingencia

1. Reinformar al cliente de la necesidad de poseer cámaras con resolución en alta defición para que funcione el sistema, para cambiarlas.
2. Utilizar un conversor en tiempo real que transforme de la imagen a una resolución de 720p

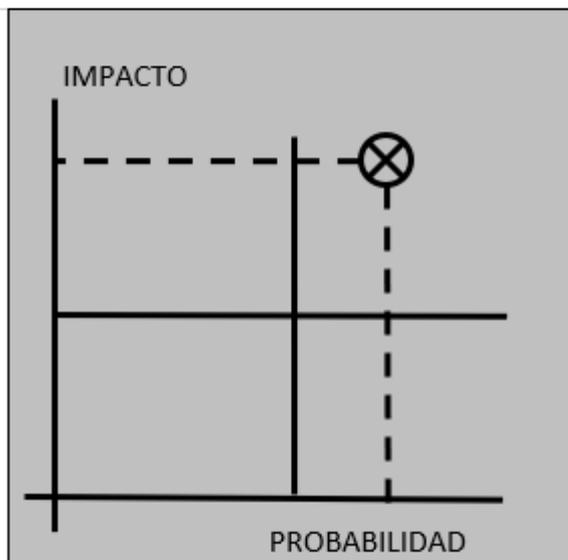
TABLA 8 - PLAN DE RESPUESTA RIESGO 12: CÁMARAS DE BAJA RESOLUCIÓN



Riesgo # 14

Descripción detallada del evento de riesgo

Al momento de funcionar el sistema, la transmisión de las imágenes no se envían en tiempo real debido al mal funcionamiento del cableado o las características del procesador de imágenes.



Acciones para Evitar, Mitigar o Transferir	Evitar	Mitigar	Transferir
Revisar las conexiones del cableado	X		
Realizar test de performance, para estresar el sistema.	X		
Acciones de Contingencia			
1. Comprar nuevos cables.			
1. Comprar nuevo procesador.			

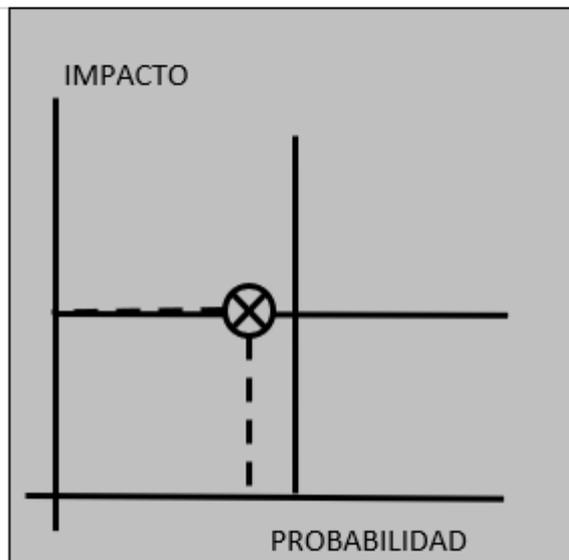
TABLA 9 - PLAN DE RESPUESTA RIESGO 14: TRANSMISION DE IMAGEN LENTA



Riesgo # 16

Descripción detallada del evento de riesgo

Al iniciar el sistema algún componente físico falla lo que provoca una imposibilidad de funcionamiento del mismo.



Acciones para Evitar, Mitigar o Transferir	Evitar	Mitigar	Transferir
Comprobar componentes de buena calidad o nuevos	X		
Realizar test de performance, para estresar el sistema.	X		
Acciones de Contingencia			
1. Reemplazar componente con falla			
1. Comprar nuevo procesador.			

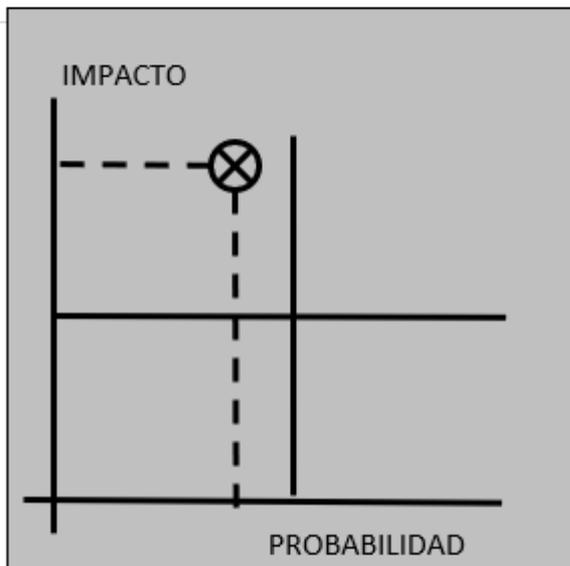
TABLA 10 - PLAN DE RESPUESTA RIESGO 16: MAL FUNCIONAMIENTO DE COMPONENTES



Riesgo # 18

Descripción detallada del evento de riesgo

Al realizar actuaciones de personas con armas de fuego en frente de las cámaras, las imágenes son enviadas en tiempo real, pero la red neuronal falla al momento de detectar las armas y generar la alarma, es decir, el sistema no se comporta como se espera.



Acciones para Evitar, Mitigar o Transferir	Evitar	Mitigar	Transferir
Evaluar en el momento de la simulación el comportamiento del sistema.		X	
Elegir las armas correctas para el momento de la actuación.	X		

Acciones de Contingencia

1. Pedir ayuda al tutor, experto en redes neuronales.
2. Reentrenar al red para mejorar su comportamiento.

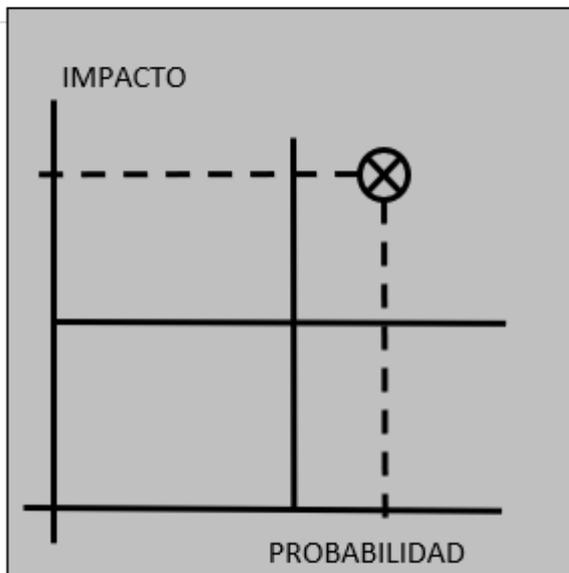
TABLA 11 - PLAN DE RESPUESTA RIESGO 18: MAL FUNCIONAMIENTO DE LA RED NEURONAL



Riesgo # 20

Descripción detallada del evento de riesgo

Una vez finalizado el cursado de la materia y presentado el informe final, de todo lo realizado durante el año sobre nuestro proyecto "Sauron", el mismo no es aprobado por nuestro tutor de cátedra.



Acciones para Evitar, Mitigar o Transferir	Evitar	Mitigar	Transferir
Asistir a todas las consultas que sean necesarias para mejorar el informe		X	
Elaborar el informe siguiendo los criterios de aceptación establecidos por la cátedra.	X		

Acciones de Contingencia

1. Rehacerlo.
2. Hablar con nuestro tutor para revisar lo que se hizo mal, y llegar a un acuerdo para mejorar el informe.

TABLA 12 - PLAN DE RESPUESTA RIESGO 20: CORRECCIÓN DE DOCUMENTACIÓN



5.6. Plan de Gestión de Proyecto

Plan Financiero del Proyecto

Línea de Base

Realizado el plan de referencia, el precio de venta del sistema es de \$1.100.000, donde ya se incluyen impuestos e instalación, quedando excluido el equipo necesario para poner en funcionamiento. El precio de este se encuentra sujeto a la inflación.

Aquél que solicite el servicio, tendrá la posibilidad de pagar en cuotas como indica la siguiente tabla:

N.º plan	Cantidad de Cuotas	Costo de Cuota Mensual	Interés Anual	Total a Pagar
1	12	\$119.167	%30	\$1.430.000
2	6	\$238.333	%15	\$1.265.000
3	1	\$990.000	%-10	\$990.000

TABLA 13 - PLAN FINANCIERO ESTIMADO PARA DESARROLLAR SAURON



6. Desarrollo del tema

6.1. Modelo de negocio

Nuestro modelo de negocio consta de proveer un sistema con inteligencia artificial para dar soporte a los agentes de seguridad, capaz de monitorear e interpretar de manera independiente las imágenes captadas desde cámaras de seguridad. Es así que es posible detectar situaciones de emergencia tales como siniestros viales, robos, incendios, personas buscadas, obteniendo una mejora de la seguridad, bienestar de la sociedad y disminución de los tiempos de respuesta por parte de las autoridades.

Para nuestra solución identificamos dos segmentos, tanto el comprador de nuestro producto (dueños de locales, gerentes) y el ciudadano que será el destinatario final, ya que se verá beneficiado por el uso del producto. Por ende, se desarrollaron dos Modelos Canvas, para estudiar y analizar la propuesta de valor para cada uno de ellos.

Para nuestra solución identificamos dos segmentos de cliente tanto el comprador de nuestro producto (dueños de locales, gerentes) y como también, el ciudadano que será el destinatario final, porque se verá beneficiado por la utilización del producto. Es por esto, que se desarrollaron dos Modelos Canvas, para estudiar y analizar la propuesta de valor para cada uno de ellos.

Como *Propuesta de Valor* para nuestros clientes: 'Ofrecemos un sistema de vigilancia automatizada para mejorar la seguridad de tu entorno, cuidando a tus clientes y empleados'. Como nombramos anteriormente, nuestros *clientes*, es decir, aquellos que comprarán nuestra solución (*Segmento de Cliente*), son aquellas personas que deseen aumentar la seguridad de su local, empresa, u hogar automatizando la vigilancia, para prevenir situaciones de violencia con armas de fuego. Es primordial que las necesidades del cliente sean satisfechas, por eso la atención será personalizada.

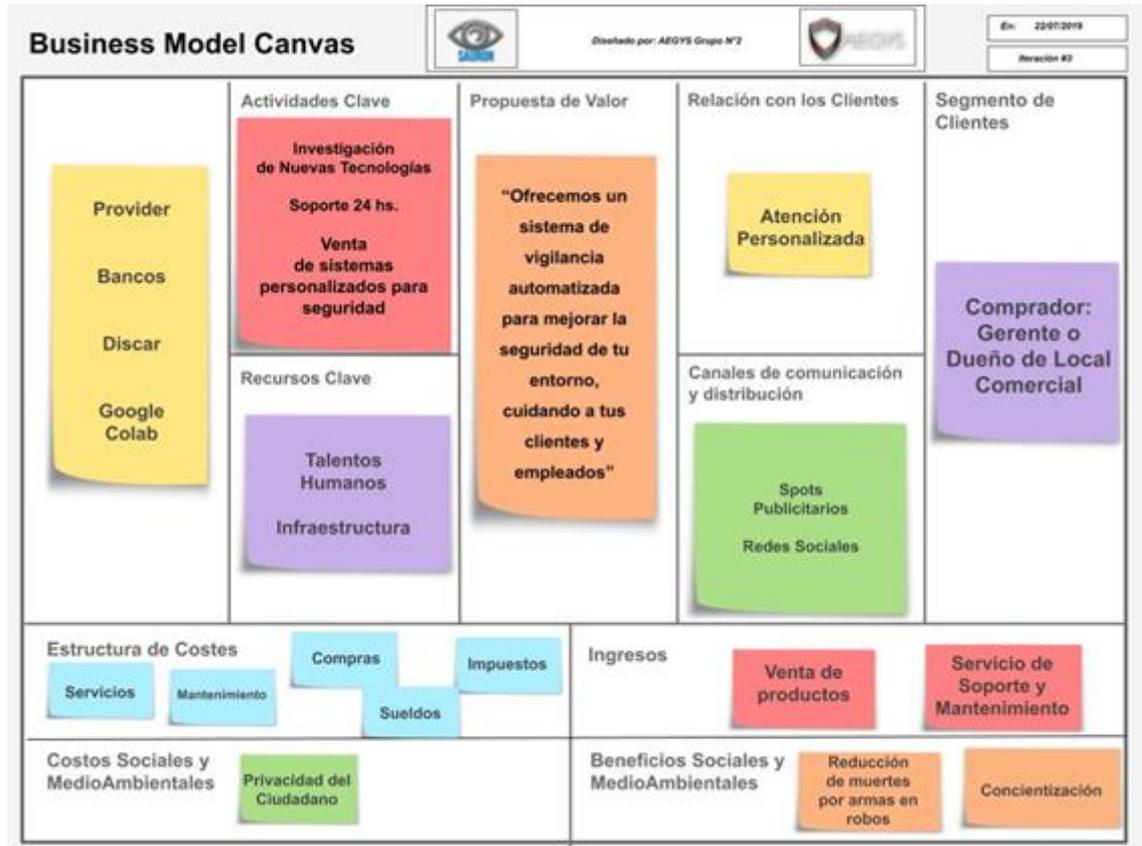


FIG. N°6. MODELO DE NEGOCIO CANVA N°1 – SEGMENTO DE CLIENTE “DUEÑO DE LOCAL”.

Nuestros *Proveedores Claves*, para lograr desarrollar nuestro producto y dar valor a través de él, son:

- Bancos: solicitaremos un préstamo inicial para solventar los gastos, o para eventualidades en caso de ser necesario.
- Provider: proveedor de servicio de internet con fibra óptica. Necesitamos un servicio de internet de calidad para poder tener un rápido acceso a la nube en donde se entrenará la Inteligencia Artificial, también es esencial un buen acceso a internet para el confort de nuestros desarrolladores.
- Discar: proveedor de insumos tecnológicos, como cámaras de seguridad, computadoras, cables, etc.
- Google Colab: herramienta clave para el entrenamiento de la red neuronal que dará vida a nuestro producto. Se la utilizará tantas veces como sea necesario para mejorar el rendimiento de esta.



Identificamos como *Actividades Claves*:

- Investigación de Nuevas Tecnologías: como empresa, queremos dar el mejor servicio, utilizando nuevas tecnologías que mejoren nuestro producto, por ello, como actividad clave, debemos mantenernos siempre informados.
- Soporte 24 hs: daremos soporte 24 hs, siempre y cuando el cliente lo necesite y acceda a este servicio.
- Venta de sistemas personalizados para seguridad: nos enfocaremos en desarrollar un producto de vigilancia automatizada para dar soporte a los agentes de seguridad. Decimos personalizados, porque nos adaptamos a los requisitos del cliente.

Para lograr estas actividades, nuestros *Recursos Claves* son:

- Talentos Humanos: consideramos fundamental contar con personas talentosas, proactivas, eficientes.
- Infraestructura: contamos con tecnologías de última generación, para maximizar el trabajo de nuestros empleados como así también el resultado de nuestro producto.

Como *Canales de Distribución*, planteamos publicitar nuestro producto a través de redes sociales, spots publicitarios en radios, televisión, etc.

Nuestra *Estructura de Costos*, está definida por:

- Sueldos: pago de sueldos a todo el capital humano involucrado en el desarrollo del producto.
- Compras: compras de infraestructura como ser, computadoras para desarrollo, servidores para entrenamiento de red, viáticos, etc.
- Servicios: luz, agua, gas, internet, alquiler del lugar donde se establecerá la empresa.
- Mantenimiento: costos de limpieza de establecimiento, mantenimiento de equipos, como ser reparaciones, repuestos, etc.
- Impuestos: pago de impuestos.

Y nuestras *Fuentes de Ingreso*, serán:

- Principalmente de las ventas del producto: este será el ingreso más fuerte, según el número de ventas podremos decir si el proyecto es sustentable a largo plazo.
- Servicios de soporte y mantenimiento: como segunda actividad clave, esperamos obtener ganancias con el servicio de soporte y mantenimiento al producto las 24 hs.

Nuestro producto tiene impacto social, es por esto por lo que utilizamos el Modelo Canvas para soluciones de este tipo. Este presenta dos componentes extras los cuales son:

Costo Sociales y Medioambientales: nuestro producto, afecta directamente la privacidad del ciudadano, debido a que las cámaras están filmando las 24 hs.

Beneficios Sociales y Medioambientales:

- Reducción de muertes y robos con armas de fuego involucradas: debido a que habrá una temprana respuesta de las fuerzas de seguridad.
- Concientización: a medida que nuestro producto obtenga reconocimiento en la sociedad, los delincuentes optarán por no usar armas.

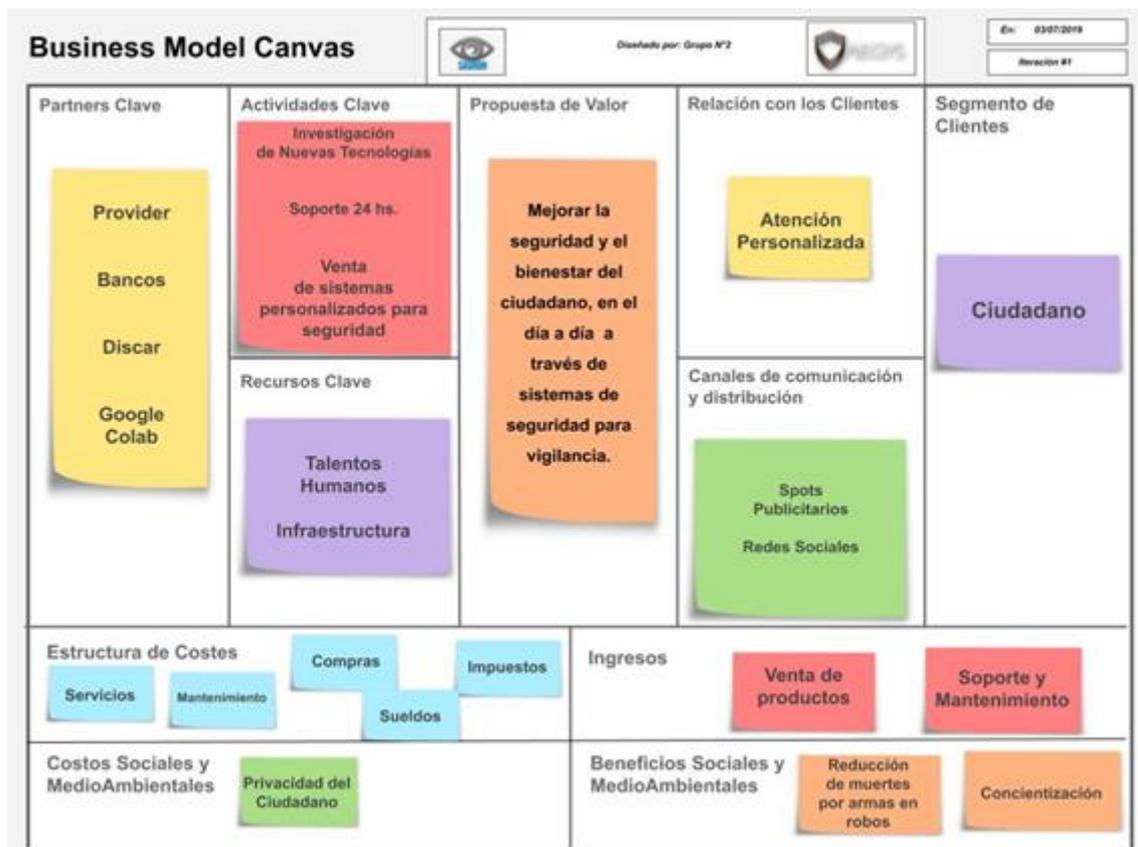


FIG. N°7. MODELO DE NEGOCIO CANVA N°2 – SEGMENTO DEL CLIENTE: CIUDADANO

Este Modelo Canvas (Fig. N°7) representa la propuesta de valor para el ciudadano, nuestro destinatario final.



Los campos que se diferencian con respecto al anterior son:

- *Segmento de Clientes*: Ciudadano, debido a que se beneficiará por el uso de nuestro producto instalado en los lugares que asista.
- *Propuesta de Valor*: 'Mejorar la seguridad y el bienestar del ciudadano, en el día a día a través de sistemas de seguridad para vigilancia'.

Los demás campos son idénticos modelo descrito anteriormente.

6.2. Desarrollo del producto

6.2.1. ANÁLISIS

6.2.1.1. Especificación de Requerimientos de Software

Introducción

Este documento es una Especificación de Requisitos Software (ERS) para el Sistema de información para la gestión de procesos y control de inventarios. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos de Software ANSI/IEEE 830, 1998.

Propósito

El propósito de este documento es definir cuáles fueron los requisitos tenidos en cuenta para satisfacer los objetivos de nuestro sistema y las necesidades del usuario.

Alcance

El alcance de este documento abarca desde quienes son los responsables y el personal involucrado y la definición de los requerimientos funcionales y no funcionales en formato ANSI/IEE830, 1998.

Personal Involucrado

Este punto indica quienes son las personas involucradas en el desarrollo de este documento.



Nombre	Gino Bernabé, Nacchio
Rol	Líder del proyecto
Categoría Profesional	Estudiante
Responsabilidad	Gestionar todos los aspectos relacionados al proyecto y representar al resto del equipo.
Información de contacto	ginonacchio@gmail.com

TABLA 14 - PERSONAL INVOLUCRADO N°1 – LÍDER DEL PROYECTO

Nombre	Agustín, Osatinsky
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Encargado de la investigación y el desarrollo de redes neuronales.
Información de contacto	aosatinsky@gmail.com

TABLA 15 - PERSONAL INVOLUCRADO N°2 - DESARROLLADOR



Nombre	Emanuel, Ceriana
Rol	Desarrollador
Categoría Profesional	Estudiante
Responsabilidad	Encargado del desarrollo front-end, del diseño y la fachada del proyecto y apoyo a la investigación de redes neuronales.
Información de contacto	emanuelceriana@gmail.com

TABLA 16 - PERSONAL INVOLUCRADO N°3 – DESARROLLADOR

Nombre	Stella, Andreozzi
Rol	Analista
Categoría Profesional	Estudiante
Responsabilidad	Análisis de información, diseño y formulación de los documentos afines.
Información de contacto	andreozzi.stella@gmail.com

TABLA 17 - PERSONAL INVOLUCRADO N°4 - ANALISTA

Definiciones, acrónimos y abreviaturas.

Dataset: El término dataset en sí es un término extranjero, un anglicismo, que hemos incorporado a nuestra lengua como un término más en los países hispanohablantes. Su traducción a nuestra lengua sería conjunto de datos y es una colección de datos habitualmente tabulada.

Un conjunto de datos o dataset corresponde a los contenidos de una única tabla de base de datos o una única matriz de datos de estadística, donde cada columna de la tabla representa una variable en particular, y cada fila representa a un miembro determinado del conjunto de datos que estamos tratando. En un conjunto de datos o dataset tenemos todos los valores que puede



tener cada una de las variables, como por ejemplo la altura y el peso de un objeto, que corresponden a cada miembro del conjunto de datos. Cada uno de estos valores se conoce con el nombre de dato. El conjunto de datos puede incluir datos para uno o más miembros en función de su número de filas.

Streaming: El concepto de streaming se refiere a cualquier contenido de medios, ya sea en vivo o grabado, que se puede disfrutar en computadoras y aparatos móviles a través de Internet y en tiempo real. Los podcasts, webcasts, las películas, los programas de TV y los videos musicales son tipos comunes de contenido de streaming.

Descripción General

Perspectiva del producto.

Nuestro sistema está compuesto por dos secciones, la red neuronal que es la encargada de interpretar las imágenes recuperadas por las cámaras de vigilancia y disparar un evento de notificación a las entidades de seguridad en el caso de detectar algún arma de fuego, y el sistema frontal con el cual va a interactuar el cliente, eligiendo que cámara ver en el momento que desee.

Funcionalidad del producto.

La funcionalidad básica de nuestro sistema es desligar y disminuir la responsabilidad de los encargados de seguridad de los locales clientes al ofrecer una vigilancia pasiva y constante para la detección de armas de fuego y posterior el aviso a las autoridades pertinentes para así prevenir cualquier situación de inminente peligro.



Características de los usuarios.

Tipo de usuario	Actividades
Administrador	Encargado de administrar el sistema, asegurar el correcto funcionamiento, disponibilidad, performance del sistema. Responsable de la creación de usuarios.
Empleado de Seguridad	Encargado de controlar que los eventos se producen luego de la detección de la posible situación de violencia. Controlar que la alarma fue disparada.

TABLA 18 - CARACTERÍSTICAS DE LOS USUARIOS QUE ADMINISTRARÁN EL SISTEMA SAURON

Restricciones.

El tabú de las personas a las inteligencias artificiales y al sentirse vigilados todo el tiempo, puede llegar a restringir el potencial de nuestro sistema.

Suposiciones y dependencias.

Se presupone que:

- Las cámaras estarán previamente instaladas y funcionando correctamente.
- Las cámaras deberán tener como mínimo una definición de 720p.
- Las cámaras deberán soportar protocolos de streaming de video como RTSP.
- El módulo soportara el procesamiento de 6 cámaras.
- La aplicación NO funcionará si hay cortes de luz.
- La aplicación NO emitirá alertas remotas sin internet.
- La aplicación NO funcionará si el procesador falla.



6.2.1.2. Requerimientos Funcionales

Número de requisito	RF01
Nombre de requisito	Interpretación de imagen
Características	El sistema interpretará imágenes en tiempo real
Descripción del requerimiento	El sistema será capaz de analizar de manera automática las imágenes utilizando una inteligencia artificial.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

TABLA 19 - REQUERIMIENTO FUNCIONAL N°1: INTERPRETACIÓN DE IMAGEN

Número de requisito	RF02
Nombre de requisito	Detección de armas
Características	El sistema deberá detectar armas
Descripción del requerimiento	Dentro de las imágenes analizadas deberá detectar armas de fuego bajo calibre utilizando red neuronal convolucional
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

TABLA 20 - REQUERIMIENTO FUNCIONAL N°2: DETECCIÓN DE ARMAS POR RED NEURONAL



Número de requisito	RF03
Nombre de requisito	Activación de alarma
Características	Al momento de detectar un arma el sistema deberá mandar una notificación.
Descripción del requerimiento	El sistema deberá activar una alarma que notifique la detección del arma a los interesados.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

TABLA 21 - REQUERIMIENTO FUNCIONAL N°3: ACTIVACIÓN DE ALARMA

Número de requisito	RF04
Nombre de requisito	Procesamiento local
Características	El procesamiento se realizará de manera local
Descripción del requerimiento	El procesamiento será localmente en el microcontrolador el cual estará conectado a las cámaras, sin la necesidad de contar con una conexión a internet.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

TABLA 22 - REQUERIMIENTO FUNCIONAL N°4: PROCESAMIENTO LOCAL DE LAS IMÁGENES



Número de requisito	RF05
Nombre de requisito	Generación de Historial de Eventos
Características	Histórico de armas detectadas
Descripción del requerimiento	El sistema deberá generar un historial de las armas detectadas, contando con la captura, fecha y hora.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

TABLA 23 - REQUERIMIENTO FUNCIONAL N°5: GENERACIÓN DE HISTORIAL DE EVENTOS

Número de requisito	RF06
Nombre de requisito	Presentación de vistas de cámaras.
Características	Se podrá elegir cuántas y qué cámara visualizar.
Descripción del requerimiento	El sistema deberá permitir al usuario seleccionar cuántas y cuáles cámaras se presentarán en la interfaz, solo en caso de tener más de una.
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional

TABLA 24 - REQUERIMIENTO FUNCIONAL N°6: PRESENTACIÓN VISTA DE CÁMARAS

6.2.1.3. Requerimientos No Funcionales

Identificación del requerimiento:	RNF01
Nombre del requerimiento	Exactitud de detección
Descripción del requerimiento:	El sistema deberá detectar un arma de fuego con una exactitud de al menos 80% en tiempo real.
Prioridad del requerimiento:	Alta

TABLA 25 - REQUERIMIENTO NO FUNCIONAL N°1: EXACTITUD DE DETECCIÓN DE ARMA

Identificación del requerimiento:	RNF02
Nombre del requerimiento	Calidad de imagen
Descripción del requerimiento:	Las cámaras deberán tener como mínimo una definición de 720p.
Prioridad del requerimiento:	Alta

TABLA 26 - REQUERIMIENTO NO FUNCIONAL N°2: CALIDAD DE IMÁGENES TOMADAS POR LAS CÁMARAS

Identificación del requerimiento:	RNF03
Nombre del requerimiento	Entrenamiento de red
Descripción del requerimiento:	Entrenar la red con un dataset genérico con un mínimo de 3000 imágenes.
Prioridad del requerimiento:	Alta

TABLA 27 - REQUERIMIENTO NO FUNCIONAL N°3: ENTRENAMIENTO DE RED NEURONAL



Identificación del requerimiento:	RNF04
Nombre del requerimiento	Cámaras con streaming
Descripción del requerimiento:	Las cámaras deberán soportar protocolos de streaming de video.
Prioridad del requerimiento:	Alta

TABLA 28 - REQUERIMIENTO NO FUNCIONAL N°4: CAMARAS CON STREAMING

Identificación del requerimiento:	RNF05
Nombre del requerimiento	Tiempo de notificación
Descripción del requerimiento:	El tiempo de envío de la notificación de alarma por la detección del arma no debe superar el segundo.
Prioridad del requerimiento:	Media

TABLA 29 - REQUERIMIENTO NO FUNCIONAL N°5: TIEMPO DE ENVIÓ DE NOTIFICACIÓN

6.2.1.4. Modelo de Casos de Uso

El diagrama de casos de uso es una forma de diagrama de comportamiento en lenguaje de modelado unificado, con la que se representan procesos empresariales, así como sistemas y procesos de programación orientada a objetos. Por lo tanto, UML no es un lenguaje de programación, sino un lenguaje de modelado, es decir, un método estandarizado para representar sistemas planificados o ya existentes. En este diagrama, todos los objetos involucrados se estructuran y se relacionan entre sí. En cuanto al funcionamiento de nuestro sistema el administrador será capaz de iniciar sesión y a través de este validar sus credenciales. Además, a través de la configuración de las cámaras será capaz de obtener la imagen procesada por la inteligencia artificial y visualizar en tiempo real la detección de armas por parte del sistema. Cada vez que un evento es registrado el administrador podrá visualizar los

mismos y el sistema será el encargado de avisar a los contactos pertinentes la ocurrencia de este.

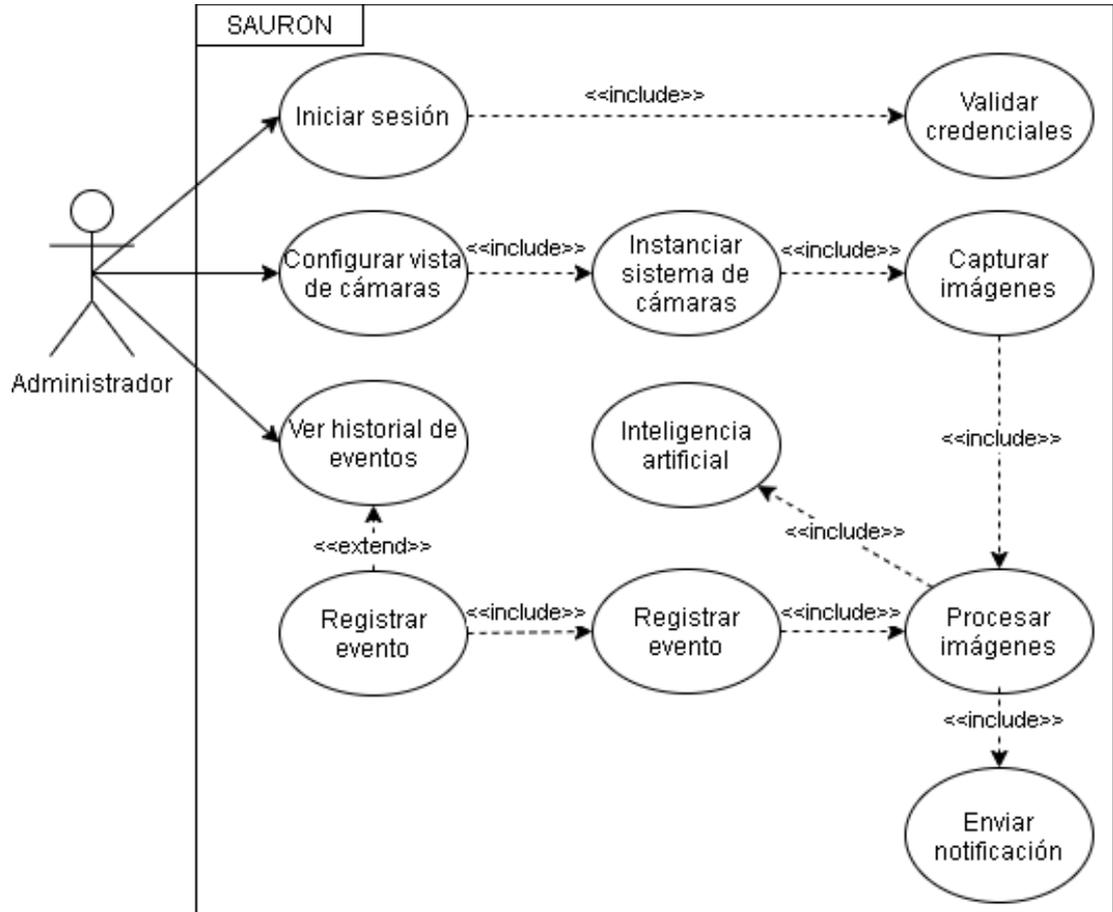


FIG. N°8. DIAGRAMA DE CASOS DE USO SISTEMA SAURÓN



Caso de Uso N1: Iniciar Sesión

Actor Principal: Administrador.

Personal Involucrado e Intereses:

- Administrador: acceder al sistema usando sus credenciales.

Precondiciones:

- Administrador dado de Alta.

Postcondiciones (Garantía de Éxito):

- El Administrador se identifica, auténtica y logra acceder al sistema.

Escenario principal de Éxito (o Flujo Básico):

Este caso de uso comienza cuando

1. El Administrador ingresa su nombre de usuario y clave en los campos correspondientes en la pantalla de Login del sistema.
2. El Administrador hace clic en el botón “Ingresar”.
3. El sistema valida que los datos ingresados sean correctos.
4. El Administrador accede al sistema y visualiza la página principal.

Flujo Alternativo

3. Los datos ingresados por el usuario son incorrectos o no se encuentran registrados en el sistema.
 - 3.1. El sistema no puede validar los datos ingresados.
 - 3.2. El sistema muestra un mensaje de error.

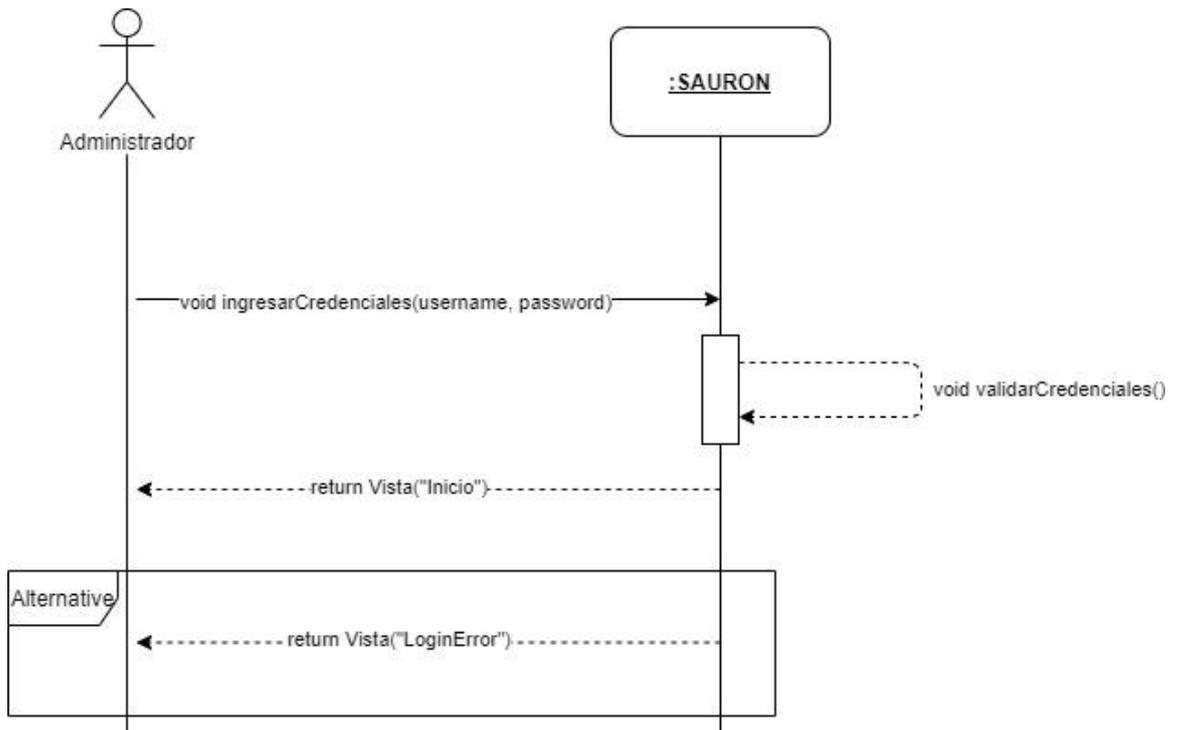


FIG. N°9. DIAGRAMA DE SECUENCIA DEL SISTEMA N°1 - CASO DE USO "INICIAR SESIÓN"

Caso de Uso N2: Configurar Vistas de Cámara

Actor Principal: Administrador.

Personal Involucrado e Intereses:

- Administrador: seleccionar el modo de vista y cantidad de cámaras a visualizar.

Precondiciones:

- El Administrador se identifica y autentica.

Postcondiciones (Garantía de Éxito):

- Interfaz de usuario presenta la cantidad de cámaras y modo de vista seleccionadas.

Escenario principal de Éxito (o Flujo Básico):

Este caso de uso comienza cuando

1. El Administrador selecciona la cantidad de cámaras a visualizar y el tipo de vista.
2. El sistema instancia el sistema controlador de cámaras.
3. El sistema presenta la interfaz de usuario con la cantidad de cámaras seleccionadas visualizando el video captado por las mismas.

Flujo Alternativo



1. La cantidad de cámaras supera el número de cámaras que utiliza el sistema.
- 1.1. El sistema muestra un mensaje de error.

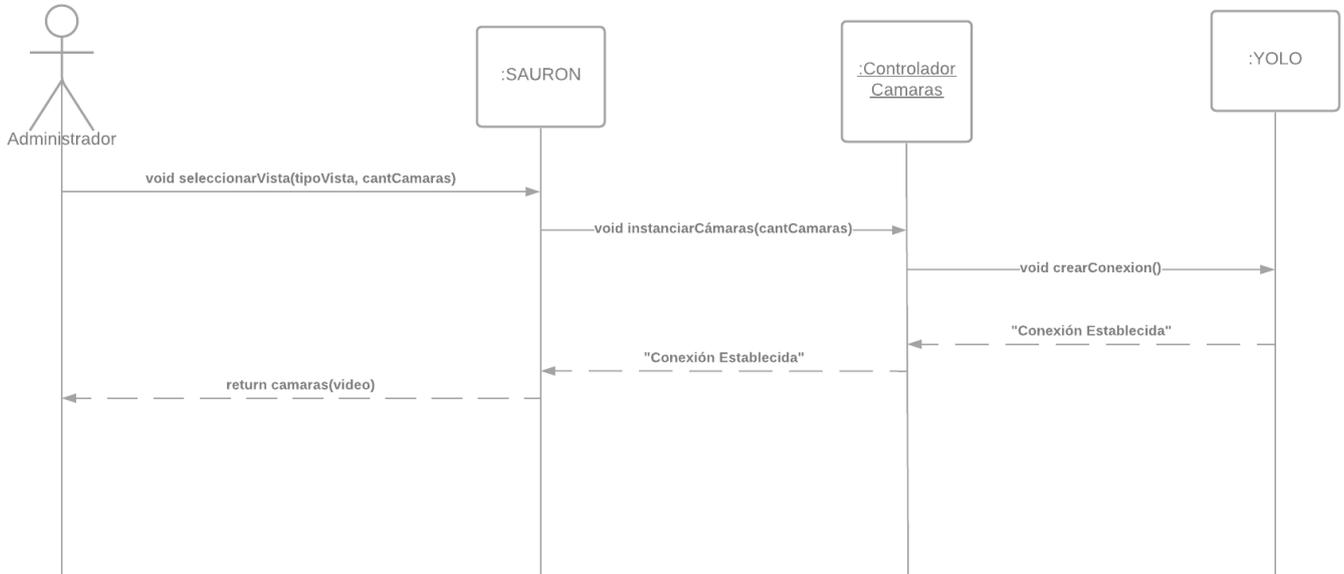


FIG. N°10. . DIAGRAMA DE SECUENCIA DEL SISTEMA N°2 - CASO DE USO “CONFIGURAR VISTAS DE CÁMARA”

Caso de Uso N3: Ver Historial de Eventos

Actor Principal: Administrador.

Personal Involucrado e Intereses:

- Administrador: ver historial de eventos en los que se detectó un arma de fuego.

Precondiciones:

- El Administrador se identifica y autentica.

Postcondiciones (Garantía de Éxito):

- Historial de eventos presentados.

Escenario principal de Éxito (o Flujo Básico):

Este caso de uso comienza cuando

1. El Administrador hace clic en Eventos, seleccione Ver Historial.
2. El Sistema presenta la interfaz de usuario de “Ver Historial”.
3. El Administrador ingresa un rango de fechas en el que desea ver los eventos.
4. El sistema presenta el historial de eventos según el rango de fechas seleccionado.

Flujo Alternativo

2. No existen eventos para mostrar en ese rango de fechas.

2.1. El sistema presenta un mensaje de error.

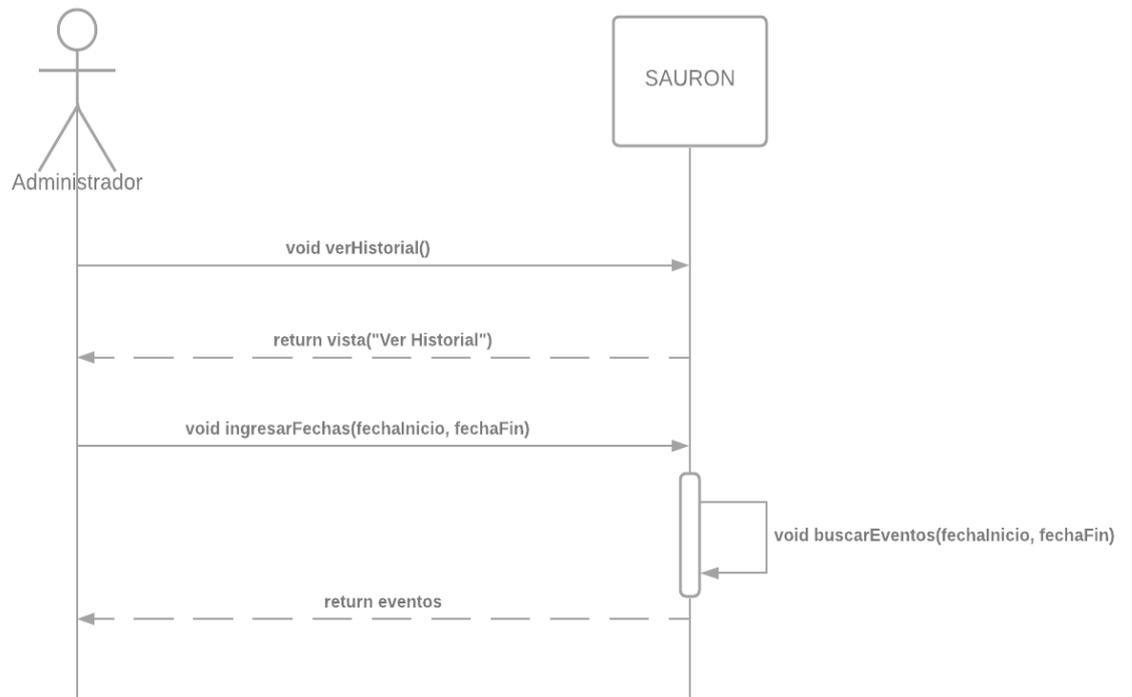


FIG. N°11. DIAGRAMA DE SECUENCIA DEL SISTEMA N°3 - CASO DE USO “VER HISTORIAL DE EVENTOS”



Caso de Uso N°4: Procesar Imagen

Actor Principal: Sistema Sauron.

Personal Involucrado e Intereses:

- Administrador: que el sistema detecte un arma fuego.

Precondiciones:

- Las cámaras están en correcto funcionamiento.
- La red está pre entrenada.

Postcondiciones (Garantía de Éxito):

- Imagen procesada por la red neuronal, con un porcentaje de precisión determinado.

Escenario principal de Éxito (o Flujo Básico):

Este caso de uso comienza cuando

1. El sistema recibe un fotograma de la cámara.
2. El Sistema envía el fotograma a la red neuronal YOLO.
3. La red neuronal infiere sobre el fotograma, devolviendo un porcentaje de precisión.

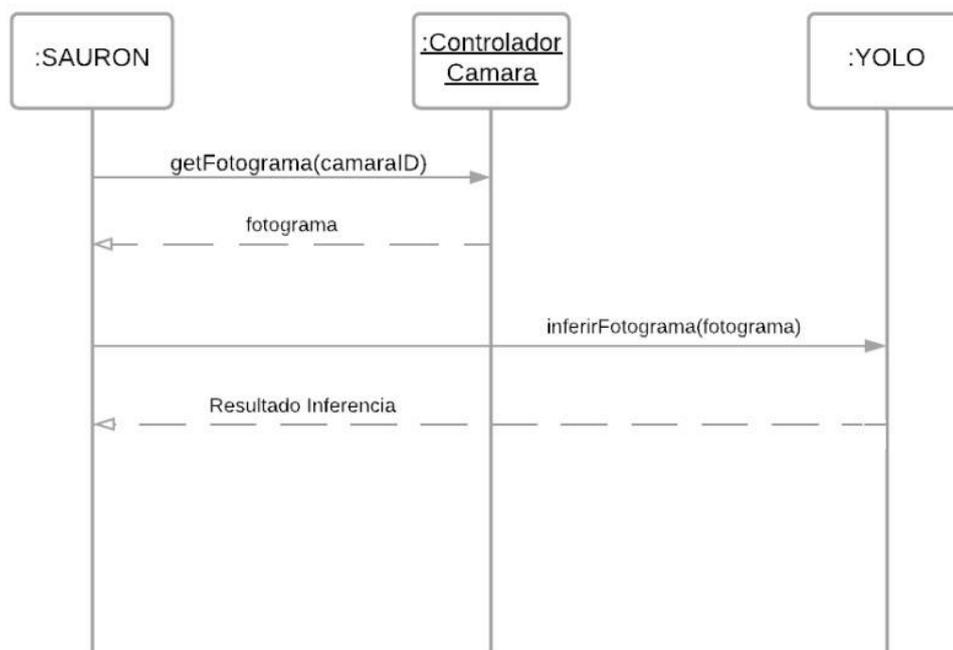


FIG. N°12. DIAGRAMA DE SECUENCIA DEL SISTEMA N°4 - CASO DE USO “PROCESAR IMAGEN”

Caso de Uso N°5: Enviar Notificación

Actor Principal: Sistema Sauron.

Personal Involucrado e Intereses:

- Administrador: recibir notificación cuando un arma es detectada.

Precondiciones:

- Las cámaras están en correcto funcionamiento.
- La red está pre entrenada.

Postcondiciones (Garantía de Éxito):

- Notificación enviada.

Escenario principal de Éxito (o Flujo Básico):

Este caso de uso comienza cuando

1. La red neuronal infirió un 80% de precisión sobre un fotograma.
2. El Sistema envía una notificación a través del sistema de mensaje SendGrid.
3. El sistema registra un nuevo evento.

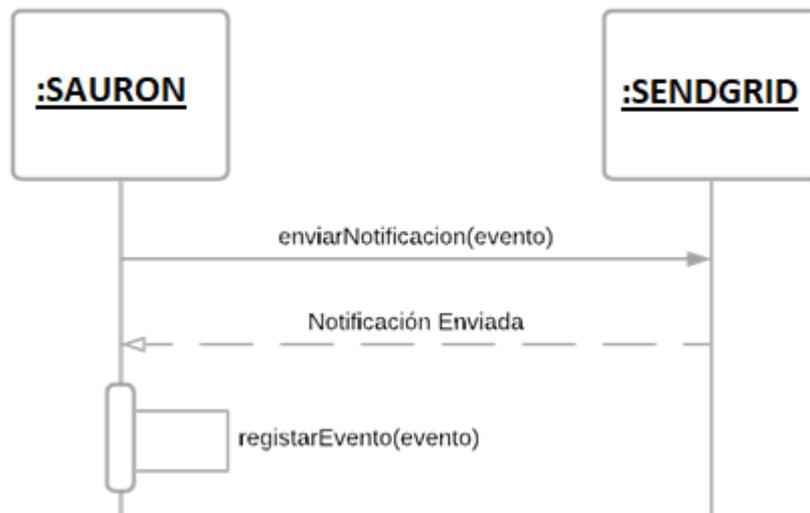


FIG. N°13. DIAGRAMA DE SECUENCIA DEL SISTEMA N°5 - CASO DE USO “ENVIAR NOTIFICACIÓN”

6.2.1.5. Diagramas de Colaboración del Sistema

En este apartado, presentamos los diagramas de colaboración que surgen a partir de los casos de usos y diagramas de secuencias.

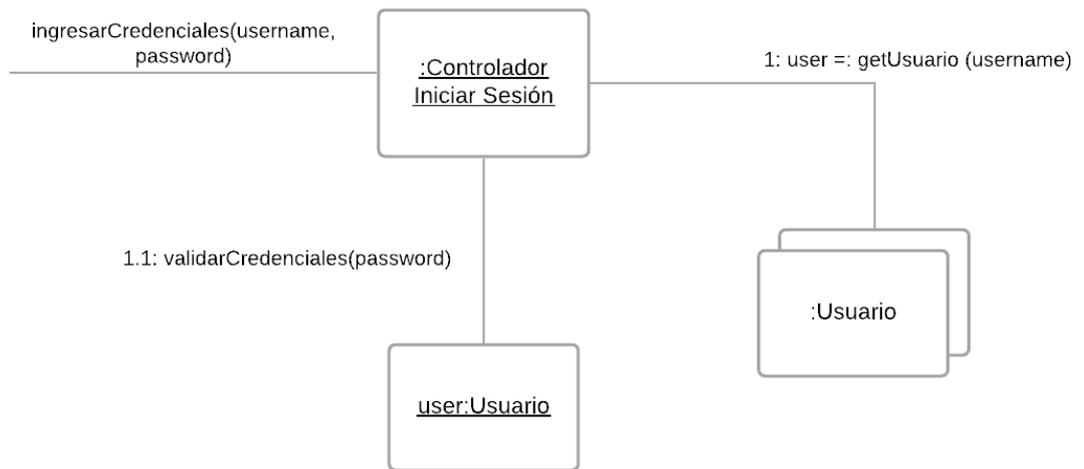


Fig. N°14. DIAGRAMA DE COLABORACIÓN N°1 - CASO DE USO "INICIAR SESIÓN"

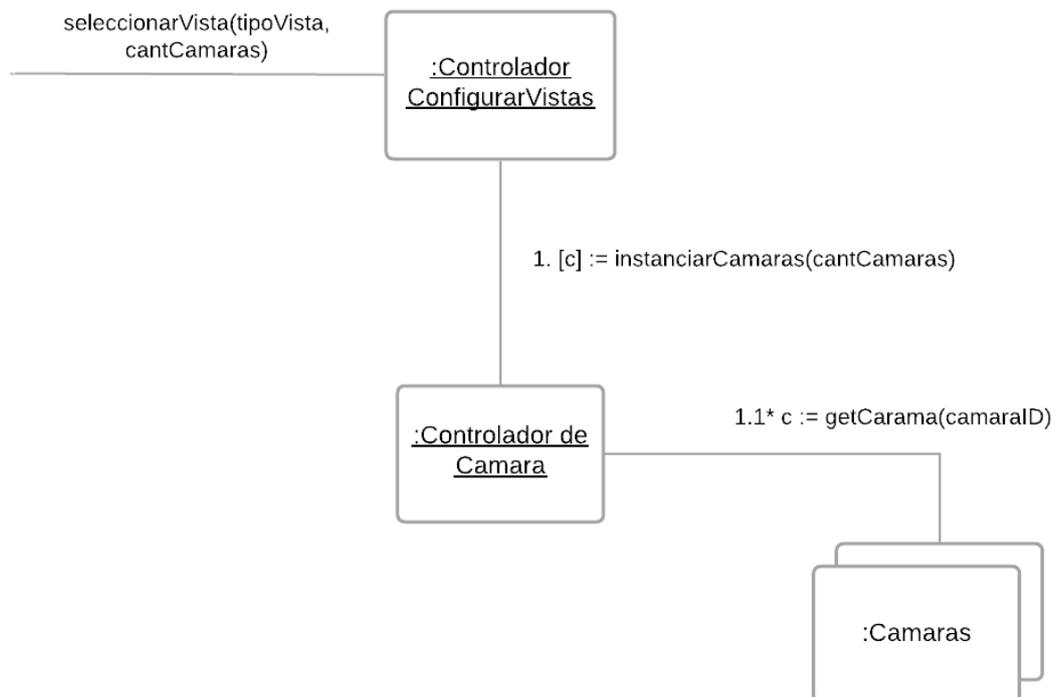


Fig. N°15. DIAGRAMA DE COLABORACIÓN N°2 - CASO DE USO "CONFIGURAR VISTA DE CÁMARAS"

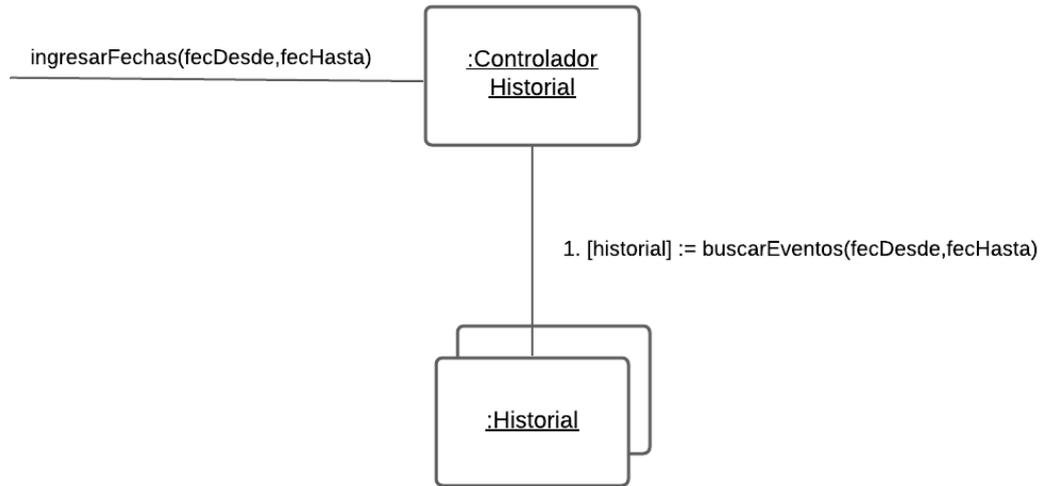


FIG. N°16. DIAGRAMA DE COLABORACIÓN N°3 - CASO DE USO “VER HISTORIAL DE EVENTOS”

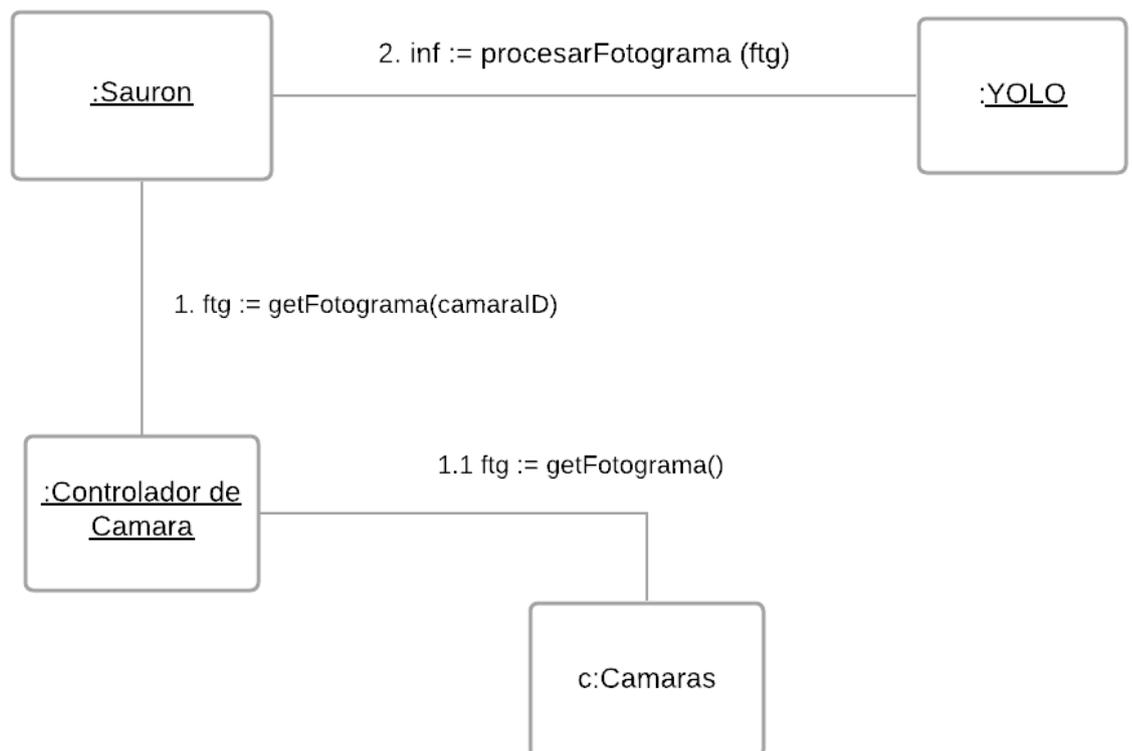


FIG. N°17. DIAGRAMA DE COLABORACIÓN N°4 - CASO DE USO “PROCESAR IMAGEN”

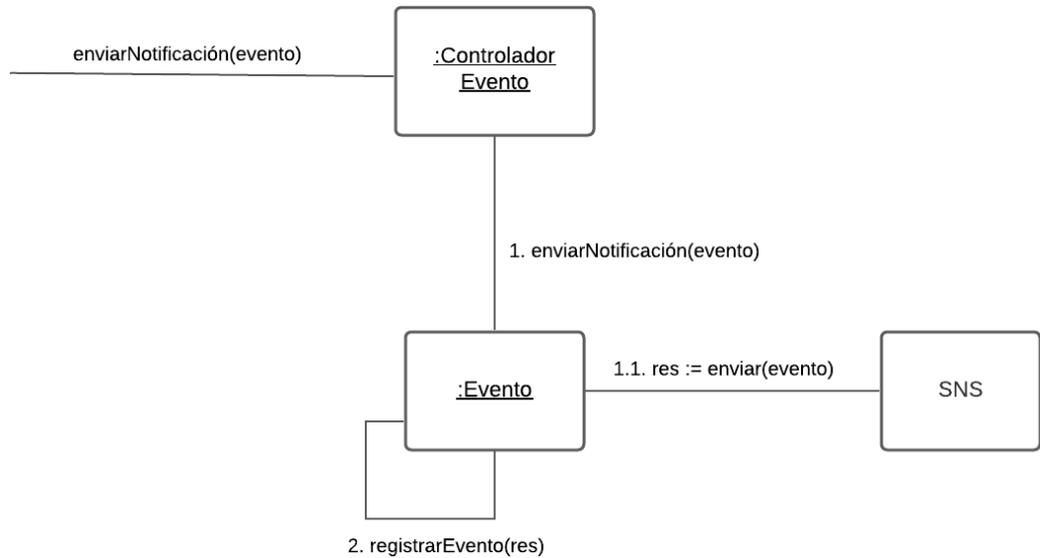


FIG. N°18. DIAGRAMA DE COLABORACIÓN N°3 - CASO DE USO “ ENVIAR NOTIFICACIÓN”

Para expandir y dar más detalle de los módulos de IA y de reconocimiento de armas, se presenta el siguiente conjunto de historias de usuario.

Historia de Usuario N°1	
Nombre de Historia: Detección de armas de fuego	
Prioridad en Negocio: Muy Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 5	
Descripción: Se deberán detectar armas de fuego por cada cámara instalada en tiempo real y con una gran exactitud.	
Validación: El criterio de aceptación es que se detecte el arma en menos de 1 segundo y con una exactitud superior al 0.80.	

TABLA N°30. HISTORIA DE USUARIOS N°1



Historia de Usuario N°2	
Nombre de Historia: Generación de Historial de Eventos	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Puntos estimados: 4	
Descripción: Se deberá tener un registro de todos los eventos captados por las diversas cámaras.	
Validación: El cliente podrá revisar un registro con todos los eventos captados y registrados por el sistema con fecha y hora de ocurrencia de este.	

TABLA N°31. HISTORIA DE USUARIOS N°2

Historia de Usuario N°3	
Nombre de Historia: Selección de Cámara	
Prioridad en Negocio: 2	Riesgo en Desarrollo: Media
Puntos estimados: 3	
Descripción: Se deberá poder elegir diferentes vistas de las cámaras.	
Validación: El cliente podrá elegir a qué cámara desea ver por pantalla y cuantas desea ver al mismo tiempo.	

TABLA N°32. HISTORIA DE USUARIOS N°3



6.2.1.6. Casos de Uso Reales:

Un caso de uso real describe el diseño concreto del caso de uso a partir de una tecnología particular de entrada y salida, así como de su implementación global. Por ejemplo, si interviene una interfaz gráfica para el usuario, el caso de uso real incluirá diagramas de las ventanas en cuestión y una explicación de la interacción de bajo nivel con los artefactos de la interfaz.

Los casos de uso tienen las siguientes características:

- Están expresados desde el punto de vista del actor.
- Se documentan con texto informal.
- Describen tanto lo que hace el actor como lo que hace el sistema cuando interactúa con él, aunque el énfasis está puesto en la interacción.
- Son iniciados por un único actor.

A continuación, presentamos los casos de uso real detallados para nuestro proyecto:

Caso de Uso Real: Presentación de Historial de Eventos

1. El usuario hace clic en A.
2. El sistema presenta una interfaz de usuario con el historial de eventos.

Caso de Uso Real: Selección de Vistas.

1. El usuario hace clic en B y presiona la tecla 'O'.
2. El sistema ocultará la vista de la cámara indicada.



FIG. N°19. MOCK UP INTERFAZ DE USUARIO: VISTA DE CÁMARAS

Caso de Uso Real: Activación de Alarma

En el caso de que el sistema detecte un arma, el sistema presentará una captura de la situación con el arma remarcada.

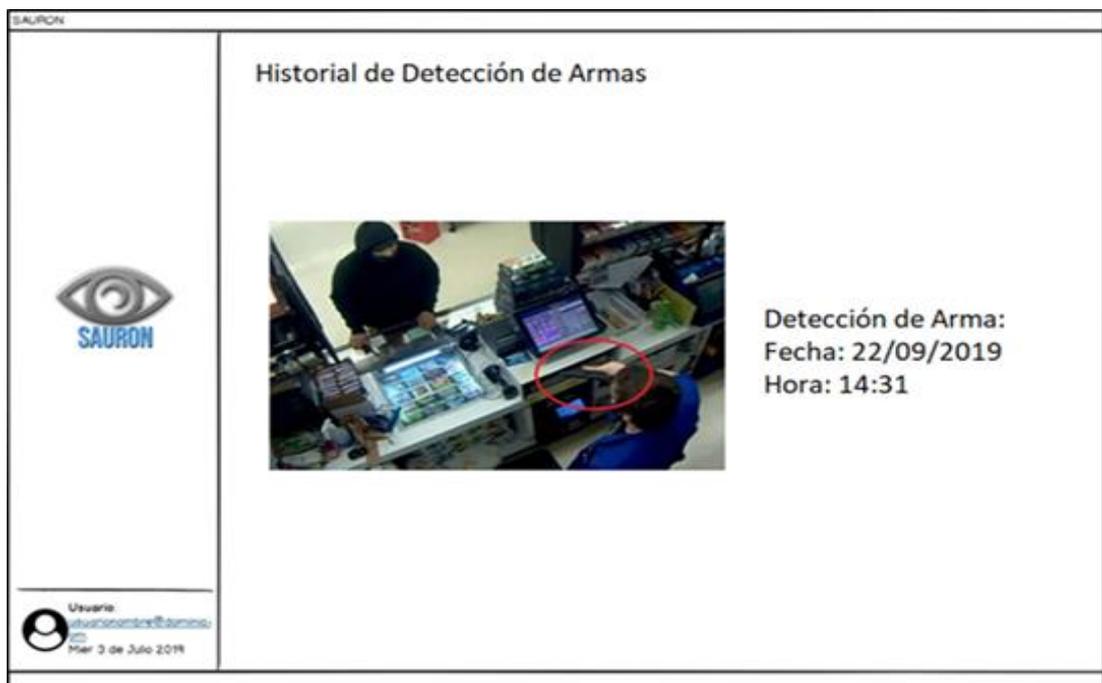


FIG. N°20. MOCK UP INTERFAZ DE USUARIO: VER HISTORIAL DE EVENTOS

6.2.2. DISEÑO

6.2.2.1. Documento de Arquitectura del Software

En esta sección esquematizamos y detallamos la arquitectura de la solución del sistema, es decir, el conjunto de todas las herramientas utilizadas y la relación entre estas. Gracias a estas herramientas logramos solucionar el problema planteado.

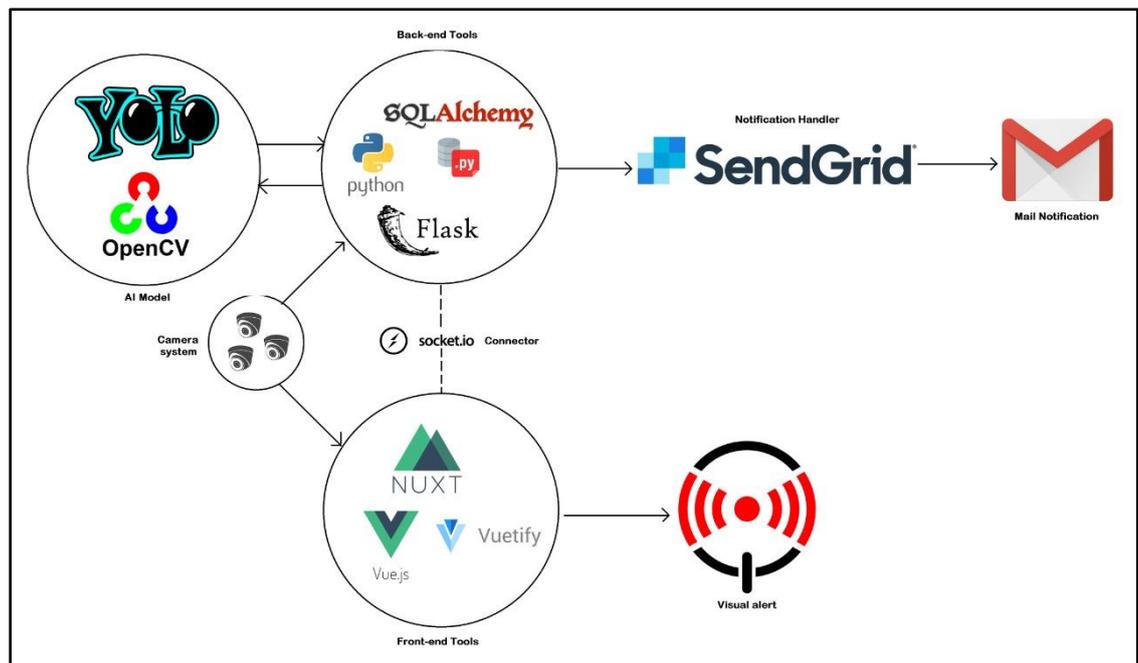


FIG. N°21. ARQUITECTURA DE SOLUCIÓN DE SAURON DONDE SE VEN LAS HERRAMIENTAS UTILIZADAS Y LA INTERACCION ENTRE ELLAS

Nos basaremos en un modelo conocido como Edge-computing, consiste en acercar el poder de procesamiento lo más cerca posible de donde los datos están siendo generados. Es decir, consiste en acercar la nube hasta el usuario, hasta el borde mismo (edge, en inglés) de la red. Lo que importa cuando hablamos del borde de la red es que todos esos elementos tienen una característica en común: procesadores. Todos tienen la capacidad de procesar y manejar datos. Gracias a ello, con el Edge Computing podemos virtualizar las capacidades del servidor y habilitamos que el poder de procesamiento ocurra en esos dispositivos del borde (como en nuestro caso, las cámaras de seguridad).

Eso permite mover capacidades que antes estaban “lejos”, en un servidor en la nube, muchísimo más cerca de los dispositivos. Es un cambio de paradigma que lo cambia todo. Las



funciones son similares, pero como el procesamiento sucede mucho más cerca, la velocidad se dispara, la latencia se reduce y las posibilidades se multiplican.

- PC: será el microprocesador encargado de correr bajo el mismo el servidor web Ngnix de Flask juntos con los scripts de inteligencia artificial desarrollados en Python junto con OpenCV que capturaron las imágenes de las cámaras, las procesaron y entregaron un resultado del análisis. De ser necesario ante un evento de emergencia, disparara una notificación de alarma a través del servicio SendGrid.
- Cámaras: encargadas de capturar las imágenes en tiempo real y enviarlas al servidor para su procesamiento y análisis.
- Socket.IO: permite comunicación basada en eventos, bidireccional y en tiempo real. Funciona en cualquier plataforma navegador y dispositivo, enfocándose de igual manera en velocidad y confiabilidad.³
- Base de datos SQL Lite: en esta base de datos relacional se llevará un registro de los sucesos de alarma que se detecten.
- SendGrid: este servicio será el encargado de enviar las notificaciones de alarma a los respectivos usuarios.

Arquitectura Lógica de la Red Neuronal: YOLO (You Only Look Once)

El algoritmo You Only Look Once (YOLO) (A. Bochkovskiy, 2020), es un sistema de código abierto del estado del arte para detección de objetos en tiempo real, el cual hace uso de una única red neuronal convolucional para detectar objetos en imágenes. Para su funcionamiento, la red neuronal divide la imagen en regiones, prediciendo cuadros de identificación y probabilidades por cada región; las cajas son ponderadas a partir de las probabilidades predichas. El algoritmo aprende representaciones generalizables de los objetos, permitiendo un bajo error de detección para entradas nuevas, diferentes al conjunto de datos de entrenamiento. El modelo se implementó como una red neuronal convolucional y fue evaluado en el set de datos para detección de PASCAL VOC. Las capas convolucionales iniciales de la red se encargan de la extracción de características de la imagen, mientras que las capas de conexión completa predicen la probabilidad de salida y las coordenadas del objeto.

³ <https://socket.io>

YOLOv4-tiny

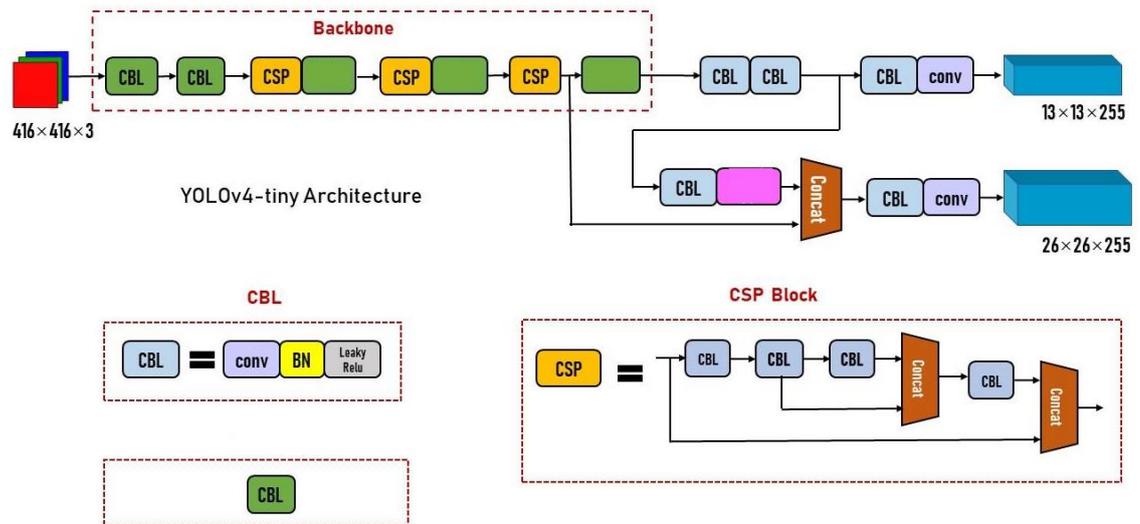


FIG. N°22. REPRESENTACIÓN GRÁFICA DE LA ARQUITECTURA LÓGICA DE RED NEURONAL: YOLO (A. BOCHKOVSKIY, 2020)

Tiny-YOLOv4 es la versión comprimida de YOLOv4. Se construye sobre la base de YOLOv4 para simplificar la estructura de la red y reducir los parámetros, de modo que resulte factible su desarrollo en dispositivos móviles e integrados como los propuestos en este proyecto.

Podemos utilizar Tiny-YOLOv4 para un entrenamiento del modelo y detección mucho más rápido ya que sólo tiene dos nodos YOLO frente a las tres de YOLOv4 lo que le quita precisión, pero aumenta considerablemente su velocidad de inferencia. La arquitectura del modelo espera como entrada imágenes 416×416 de 3 capas (RGB) y otorga como resultado las coordenadas de los posibles objetos encontrados.

Es por eso por lo que YOLOv4-tiny es la mejor opción en comparación con YOLOv4, ya que un tiempo de inferencia más rápido es más importante que la precisión o la exactitud cuando se trabaja con un entorno de detección de objetos en tiempo real.

Arquitectura en Capas

Arquitectura en Capas, es la vista conceptual de la estructura de la arquitectura de una aplicación. ⁴

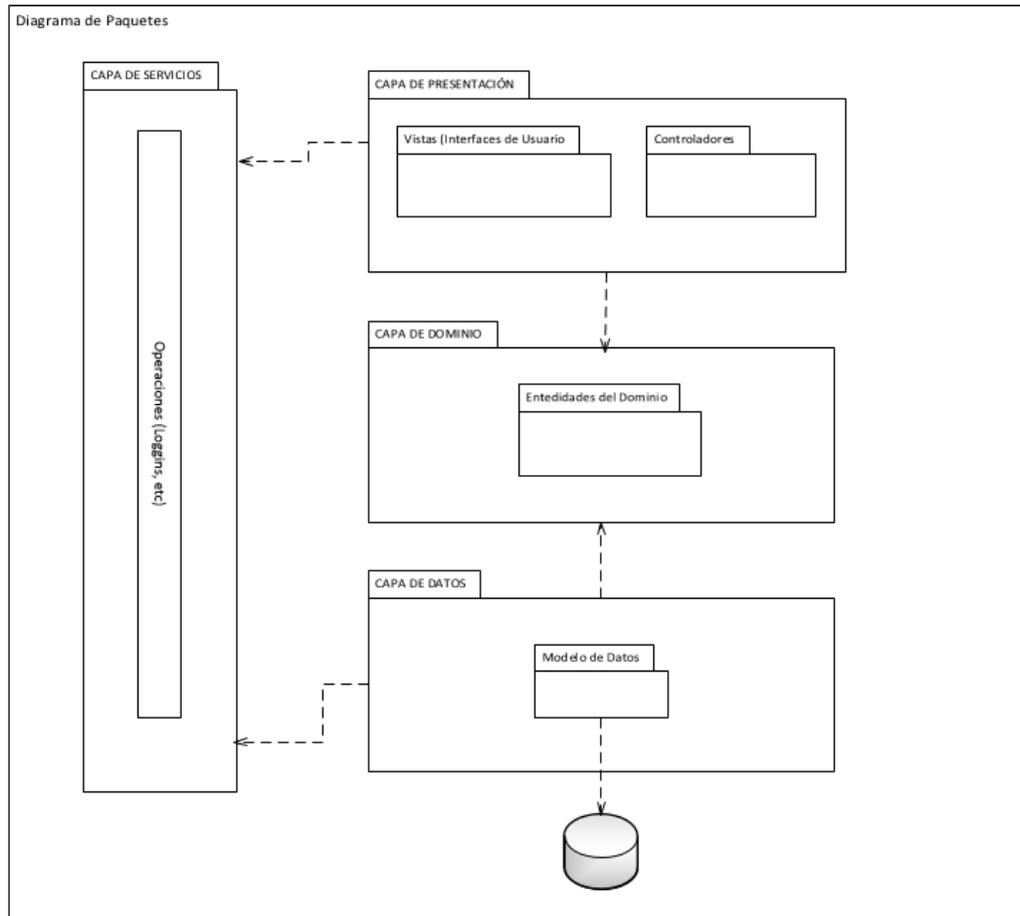


FIG. N°23. REPRESENTACIÓN GRÁFICA DE LA ARQUITECTURA EN CAPAS DEL SISTEMA

⁴ https://www.ecured.cu/Arquitectura_en_Capas

6.2.2.2. Mockup

En este apartado presentamos los mockups (modelo que se utiliza para exhibir o probar un diseño) según las iteraciones del desarrollo. En este caso hacemos referencia a la primera iteración. Un mockup básico de imágenes estáticas sin interacción.

Iteración N°1:



FIG. N°24. MOCK UP INTERFAZ DE USUARIO: VISTA DE CÁMARAS V1.0 - EL USUARIO PODRÁ VER LAS IMÁGENES EN TIEMPO REAL CAPTADAS POR EL SISTEMA DE CÁMARAS.

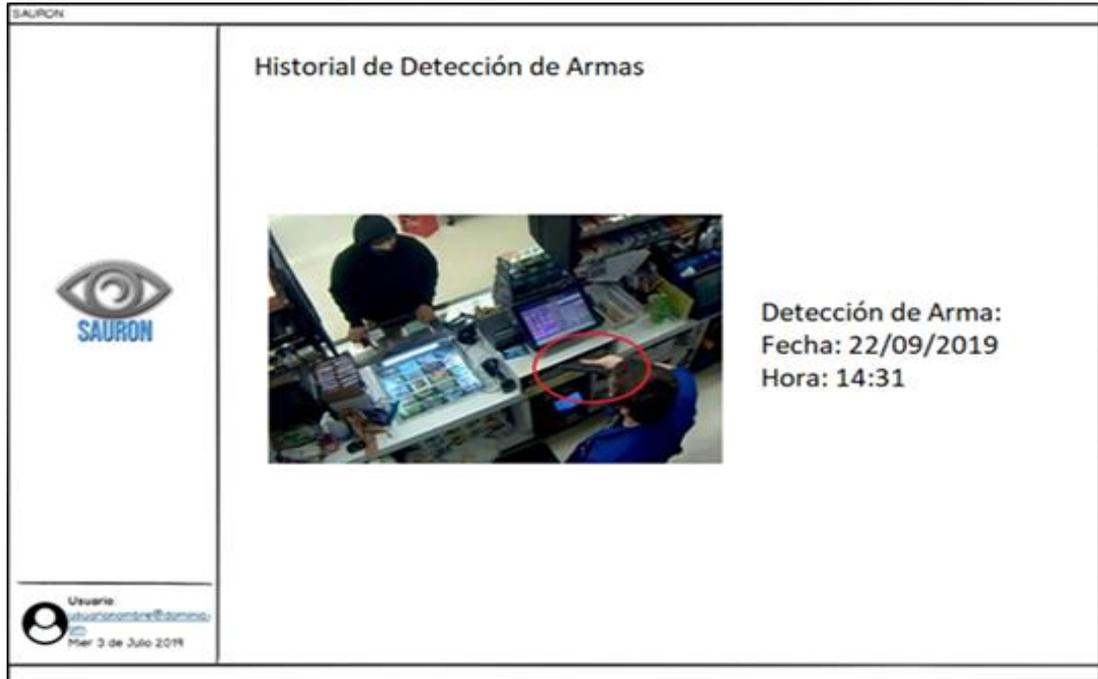


FIG. N°25. MOCK UP INTERFAZ DE USUARIO: HISTORIAL DE ALARMAS V1.0 - EN EL HISTORIAL DE EVENTOS, EL USUARIO PODRÁ VER UN SNAPSHOT (CAPTURA DE UN FRAME ESPECÍFICO DE UN VIDEO), CON FECHA Y HORA DE CUANDO LA INTELIGENCIA ARTIFICIAL DETECTE UN ARMA.



FIG. N°26. MOCK UP INTERFAZ DE USUARIO: VISTA DE ALERTA POR ARMA DETECTADA V1.0 - SE MOSTRARÁ UN CARTEL LLAMATIVO Y CON COLORES DE ALARMA, PARA PREVENIR AL CLIENTE QUE SE DETECTÓ UN ARMA Y QUE YA SE LE INFORMÓ A LOS AGENTES DE SEGURIDAD.

Iteración N°2: En esta segunda iteración se creó un mockup dinámico utilizando el framework **VueJs** y **Vuetify** para los estilos. La funcionalidad es la misma que la mostrada en la iteración número 1.

Vue (pronunciado /vju:/, como view) es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, Vue está diseñado desde cero para ser utilizado incrementalmente.⁵

Vuetify es un framework que combina la potencia del popular VueJs con la estética de Material Design. Permite acelerar el desarrollo de aplicaciones web complejas, incorporando una gran cantidad de componentes "listos para usar". <https://www.luisllamas.es/vuetify-estetica-material-design-para-tus-apps-en-vuejs/>

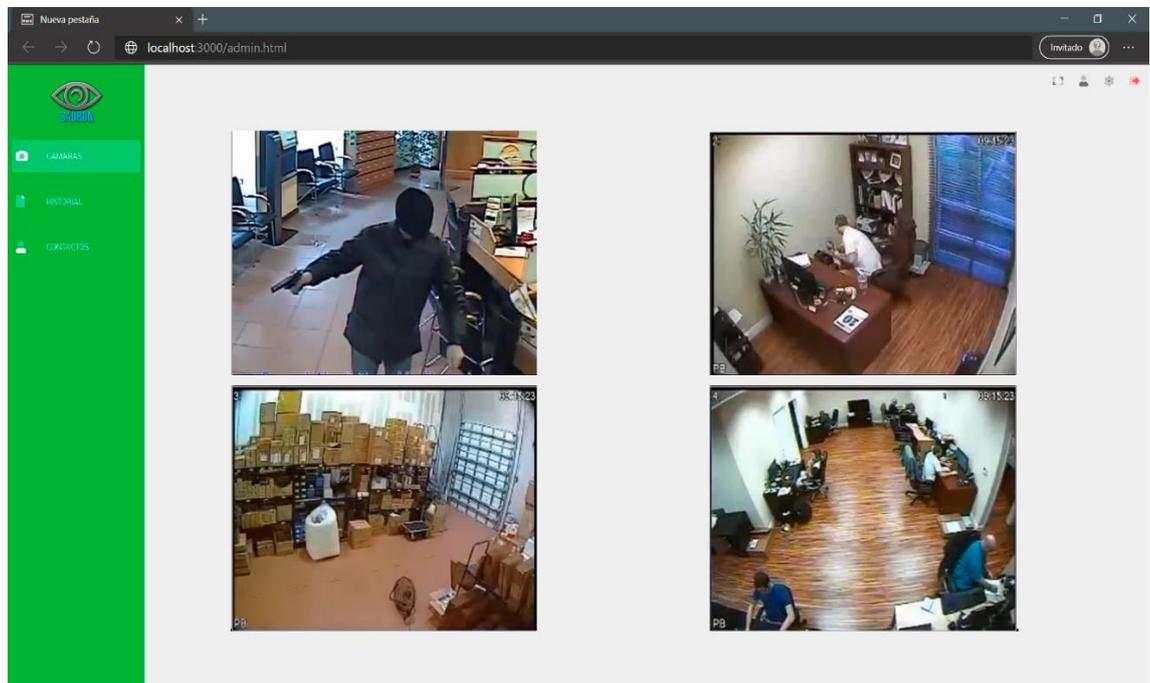


FIG. N°27. MOCK UP INTERFAZ DE USUARIO: VISTA DE CÁMARAS V2.0 - EL USUARIO PODRÁ VER LAS IMÁGENES EN TIEMPO REAL CAPTADAS POR EL SISTEMA DE CÁMARAS.

⁵ <https://es.vuejs.org/v2/guide/>

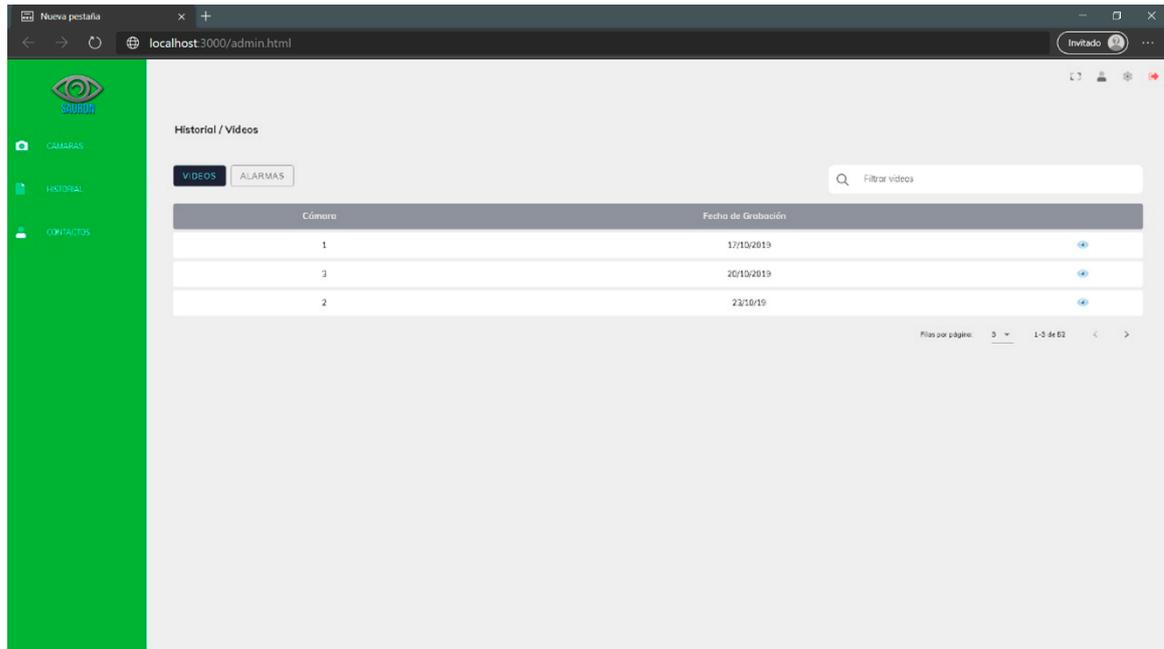


FIG. N°28. MOCK UP INTERFAZ DE USUARIO: HISTORIAL DE ALARMAS V2.0 - EN EL HISTORIAL DE EVENTOS, EL USUARIO PODRÁ VER UN SNAPSHOT (CAPTURA DE UN FRAME ESPECÍFICO DE UN VIDEO), CON FECHA Y HORA DE CUANDO LA INTELIGENCIA ARTIFICIAL DETECTE UN ARMA.

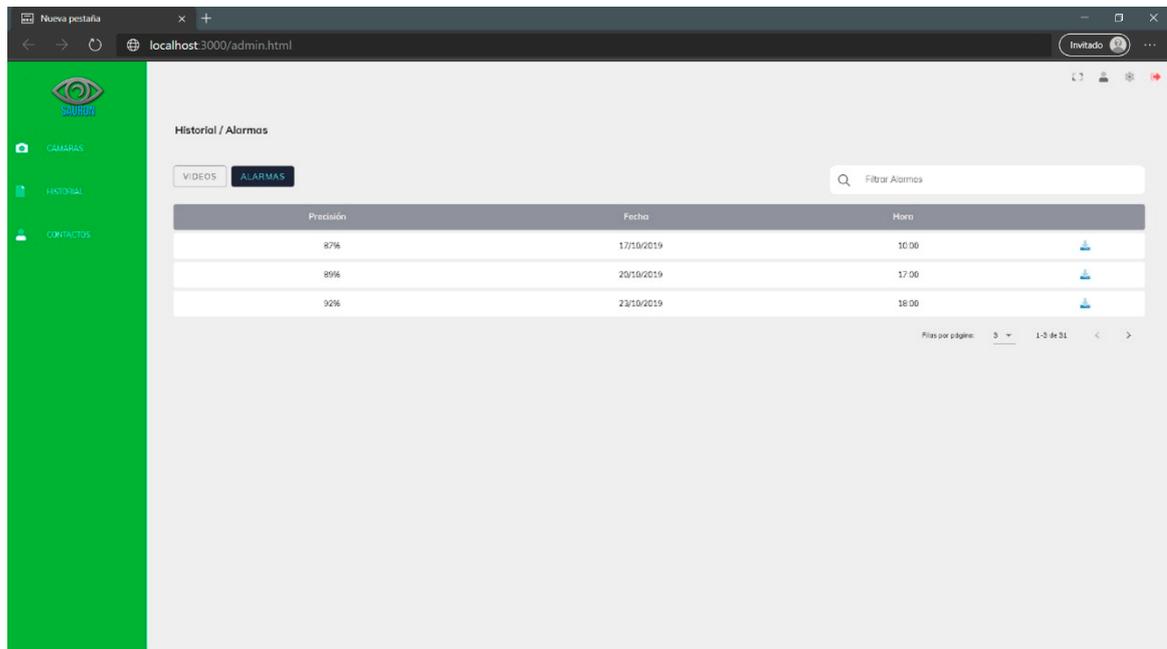


FIG. N°29. MOCK UP INTERFAZ DE USUARIO: VISTA DE ALERTA POR ARMA DETECTADA V1.0 - SE MOSTRARÁ UN CARTEL LLAMATIVO Y CON COLORES DE ALARMA, PARA PREVENIR AL CLIENTE QUE SE DETECTÓ UN ARMA Y QUE YA SE LE INFORMÓ A LOS AGENTES DE SEGURIDAD.

6.2.2.3. Diagrama Entidad Relación

Un modelo entidad-relación es una herramienta para el modelo de datos, la cual facilita la representación de entidades de una base de datos.

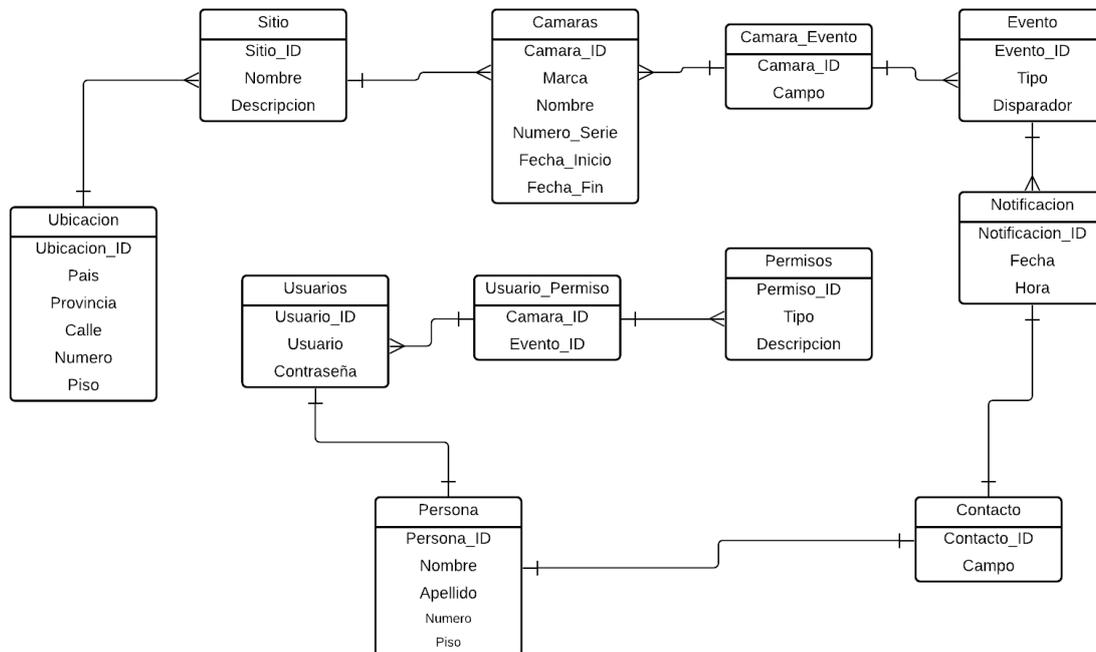


FIG. N°30. MODELO ENTIDAD RELACIÓN DEL SISTEMA.

6.2.3. IMPLEMENTACIÓN

6.2.3.1. Tecnologías de Implementación

Tecnologías de Back end, Front-End, scripting e IA:

Python: Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad.



Características del lenguaje:

- **Propósito general:** Se pueden crear todo tipo de programas. No es un lenguaje creado específicamente para la web, aunque entre sus posibilidades sí se encuentra el desarrollo de páginas.
- **Multiplataforma:** Hay versiones disponibles de Python en muchos sistemas informáticos distintos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.
- **Interpretado:** Quiere decir que no se debe compilar el código antes de su ejecución. En realidad, sí que se realiza una compilación, pero esta se realiza de manera transparente para el programador. En ciertos casos, cuando se ejecuta por primera vez un código, se producen unos bytecodes que se guardan en el sistema y que sirven para acelerar la compilación implícita que realiza el intérprete cada vez que se ejecuta el mismo código.
- **Interactivo:** Python dispone de un intérprete por la línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, que puede ayudarnos a entender mejor el lenguaje y probar los resultados de la ejecución de porciones de código rápidamente.
- **Orientado a Objetos:** La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
- **Funciones y librerías:** Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de strings, números, archivos, etc. Además, existen muchas librerías que podemos importar en los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.
- **Sintaxis clara:** hay que destacar que Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave begin y end. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores adopten unas mismas notaciones y que los programas de cualquier persona tenga un aspecto muy similar.

OpenCV: es una biblioteca libre de visión artificial originalmente desarrollada por Intel. OpenCV significa Open Computer Vision (Visión Artificial Abierta)⁶. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en una gran cantidad de aplicaciones, y hasta 2020

⁶ <https://opencv.org/>



se la sigue mencionando como la biblioteca más popular de visión artificial. Detección de movimiento, reconocimiento de objetos, reconstrucción 3D a partir de imágenes, son sólo algunos ejemplos de aplicaciones de OpenCV. Publicada bajo licencia BSD, permite que sea usada libremente para propósitos comerciales y de investigación. Además, es multiplataforma para los sistemas operativos GNU/Linux, Mac OS X, Windows y Android, y para diversas arquitecturas de hardware como x86, x64 (PC), ARM (celulares y Raspberry Pi).

Darknet: Darknet es el framework para entrenar las redes neuronales, es de código abierto y está escrito en C / CUDA y sirve como base para YOLO.

SendGrid: es una plataforma de comunicación con la cliente basada en Denver, Colorado para correo electrónico transaccional y de marketing. SendGrid proporciona un servicio basado en la nube que ayuda a las empresas con la entrega de correo electrónico. El servicio administra varios tipos de correo electrónico, incluidas notificaciones de envío, solicitudes de amistad, confirmaciones de registro y boletines informativos por correo electrónico. También maneja la supervisión del proveedor de servicios de Internet (ISP), las claves de dominio, el marco de políticas del remitente (SPF) y los bucles de retroalimentación. Además, la empresa proporciona seguimiento de enlaces e informes de tasa de apertura. También permite a las empresas realizar un seguimiento de las aperturas, las cancelaciones de suscripción, los rebotes y los informes de spam de los correos electrónicos. A partir de 2012, la empresa integró las capacidades de notificación por SMS, voz y push a su servicio a través de una asociación con Twilio.⁷

Flask: framework minimalista escrito en Python que permite crear aplicaciones web rápidamente y con un mínimo número de líneas de código. Está basado en la especificación WSGI de Werkzeug y el motor de templates Jinja2 y tiene una licencia BSD.⁸

HTML: definen la estructura semántica del sitio.

CSS: Los archivos CSS u hojas de estilo en cascada definen la apariencia de la página (los colores de fondo, tamaños y colores de fuente, ubicación de los elementos, cambios según el tamaño del dispositivo).

Javascript: Este es el lenguaje de programación que interpretan los navegadores, y por tanto, es el lenguaje que permite realizar interacciones con la página. JS permite escuchar eventos (clic, doble clic, cursor encima de, arrastrar) y realizar alguna acción como efecto de ello (modificar la

⁷ https://app.sendgrid.com/login?redirect_to=%2F

⁸ <https://codigofacilito.com/articulos/por-que-flask>



estructura HTML, o cambiar las propiedades de los elementos alterando sus estilos asignados inicialmente).

Vuetify: es un framework de interfaz de usuario completo construido sobre Vue.js. El objetivo del proyecto es proporcionar a los desarrolladores las herramientas que necesitan para crear experiencias de usuario enriquecedoras y atractivas.⁹

Base de Datos SQLite: SQLite es una biblioteca en proceso que implementa un autónomo , sin servidor , configuración cero , transaccional motor de base de datos SQL. El código de SQLite es de dominio público y, por lo tanto, es gratuito para cualquier propósito, comercial o privado. SQLite es la base de datos más implementada en el mundo con más aplicaciones de las que podemos contar, incluidos varios proyectos de alto perfil.

SQLite es un motor de base de datos SQL incorporado. A diferencia de la mayoría de las otras bases de datos SQL, SQLite no tiene un proceso de servidor separado. SQLite lee y escribe directamente en archivos de disco normales. Una base de datos SQL completa con múltiples tablas, índices, disparadores y vistas está contenida en un solo archivo de disco. El formato de archivo de la base de datos es multiplataforma: puede copiar libremente una base de datos entre sistemas de 32 bits y 64 bits o entre arquitecturas big-endian y little-endian . Estas características hacen de SQLite una opción popular como formato de archivo de aplicación . Los archivos de base de datos SQLite son un formato de almacenamiento recomendado por la Biblioteca del Congreso de EE. UU. Piense en SQLite no como un reemplazo de Oracle pero como reemplazo de fopen().¹⁰

Nginx: es un servidor web/proxy inverso ligero de alto rendimiento y un proxy para protocolos de correo electrónico (IMAP/POP3). Es software libre y de código abierto, licenciado bajo la Licencia BSD simplificada; Es multiplataforma, por lo que corre en sistemas tipo Unix (GNU/Linux, BSD, Solaris, Mac OS X, etc.) y Windows.¹¹

⁹ <https://vuetifyjs.com/en/>

¹⁰ <https://www.sqlite.org/about.html>

¹¹ <https://www.nginx.com/>



6.2.3.2. Entrenamiento del Modelo de Implementación

En esta etapa se entrenó el modelo de inteligencia artificial capaz de inferir sobre imágenes para detectar armas de fuego. En un primer acercamiento a las tecnologías de visión computacional optamos por probar un servicio brindado por Google llamado Google Visión capaz de reconocer distintos objetos en una imagen dada. A través de este servicio sacamos la conclusión de que era posible detectar armas usando cámaras de seguridad.

Debido a que planteamos utilizar un enfoque de Edge Computing, es decir, que el procesamiento de las imágenes se realice de manera local (donde fueron capturadas las mismas), continuar utilizando el servicio de Google visión no era factible. Es por eso que optamos por entrenar nuestra propia red neuronal basada en modelos ya existentes que probaron tener tasas de efectividad muy altas. Para lograr esto decidimos investigar sobre la creación de algoritmos de entrenamientos de CNN (redes neuronales convolucionales).

Buscamos librerías Open Source (libres y de uso gratuito), creamos un repositorio central, desarrollamos el código de entrenamiento y finalmente con el mismo conjunto de imágenes utilizadas en las pruebas anteriores entrenamos nuestro modelo de IA y obtuvimos resultados. Dichos resultados en un principio no cumplían con los requerimientos planteados ya que presentaban una alta tasa de falsos positivos, lo que representaría un problema en la solución final. Por este motivo decidimos reentrenar el modelo con una mayor cantidad de imágenes y cambiarlo por uno más eficiente.

Cambiamos de RetinaNet a Yolo V4 obteniendo unos resultados muchos más favorables y acordes a nuestro planteamiento inicial. Como es posible observar en la siguiente imagen el modelo de YoloV4 demostró ser más eficiente y preciso que RetinaNet usando el dataset de prueba MS-COCO. Podemos observar que a medida que aumentan los FPS (cuadros por segundo) el modelo de Yolo varía levemente en mAP (precisión media), mientras que RetinaNet sufre un cambio muy brusco a los 40 FPS.

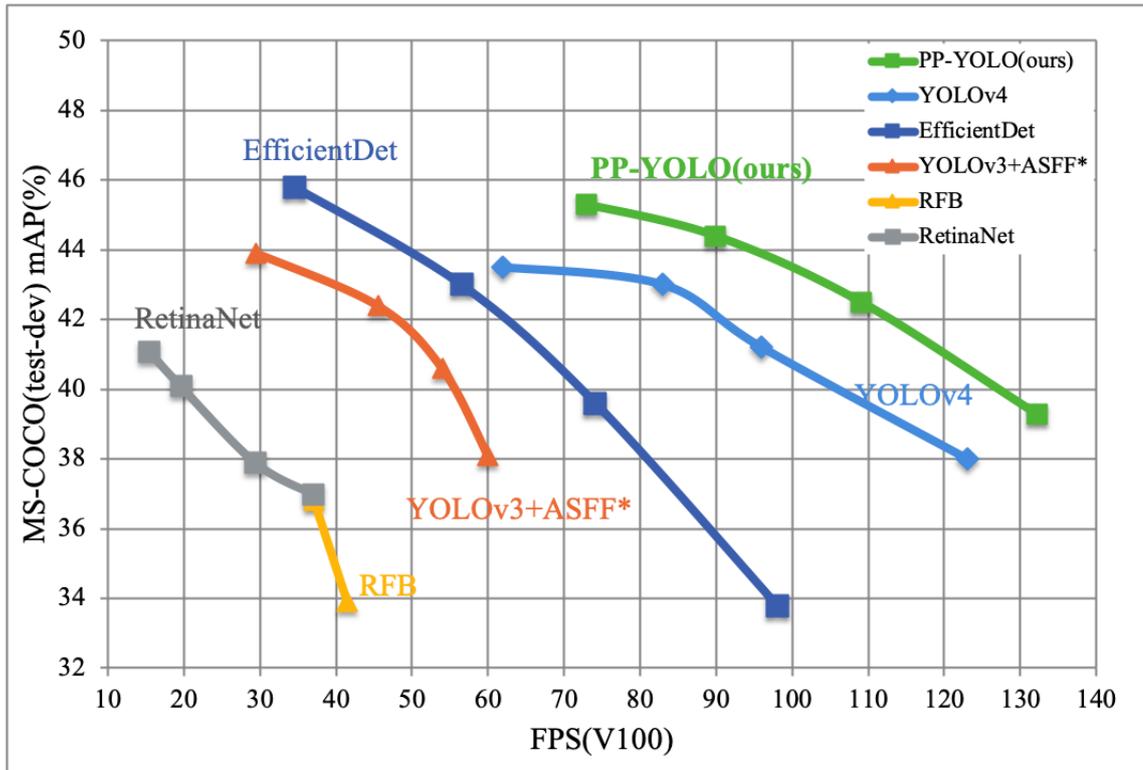


FIG. N°31. COMPARACIÓN DE RESULTADOS OBTENIDOS EN LA DETECCIÓN DE OBJETOS UTILIZANDO YOLO, RETINADET, CSP, ENTRE OTRAS. (XIANG LONG, 2020)

El servicio utilizado para el entrenamiento del modelo fue Google Colab que es un entorno de máquinas virtuales basado en Jupyter Notebooks. Es posible elegir correr el notebook en una CPU, GPU o en una TPU de forma gratuita durante 12 horas.

A continuación mostramos un collage de algunas de las imágenes utilizadas en el dataset para el entrenamiento de nuestro modelo de visión computacional y otra imagen donde se demuestra la red en funcionamiento en una simulación con el equipo de desarrollo como actores.



FIG. N° 32. CAPTURA DE SIMULACIÓN DEMOSTRANDO RED EN FUNCIONAMIENTO



FIG. N° 33. COLLAGE DEMOSTRATIVO DEL DATASET UTILIZADO PARA EL ENTRENAMIENTO DE LA RED NEURONAL



6.2.4 PRUEBAS

6.2.4.1 Modelo de Pruebas

Basándonos en las recientes experiencias en la industria de desarrollo de aplicaciones con Inteligencia Artificial, probar AI software ha traído nuevos problemas y desafíos debido a sus características especiales, nombradas a continuación.

- Desarrollo de software basado en ciencia en vez de ingeniería: a diferencia del software que se desarrolla generalmente, los basados en AI se desarrollan a partir de enfoques científicos.
- Data limitada para entrenamiento y validación: AI software es construido basado en modelos y técnicas de machine learning, y entrenados y validados con datos de entrada limitados.
- Incertidumbre sobre las salidas, respuesta y toma de decisiones: debido a que los modelos de AI que existen dependen de algoritmos estáticos, esto trae incertidumbre sobre las salidas que podría dar el sistema.

Estas características únicas y especiales de software basados en AI presentan nuevas dificultades pero también desafíos al momento de testear y validar la calidad del sistema. Y es por eso por lo que nos encontramos con las siguientes preguntas:

- ¿Cómo realizar control de calidad para grandes cantidades de datos que puede ser utilizada como datos de entrenamiento o datos de prueba para el algoritmo?
- ¿Cómo realizar validación de calidad para el servicio de la aplicación? Es decir, ¿Cuál es la precisión del servicio?

Es por ello que para realizar las pruebas de calidad y validación de nuestro sistema, es que utilizaremos la técnica de “Testing AI Software” (Testing and Quality Validation for AI Software— Perspectives, Issues, and Practices CHUANQI TAO^{1,2,3}, JERRY GAO⁴, AND TIEXIN WANG^{1,2}).

Testing AI Software: se refiere a diversas actividades de prueba para Software/sistemas basados en IA. Modelos de validación, métodos y técnicas de calidad bien definidas deben ser desarrolladas y aplicadas para software basado en AI, para facilitar las actividades de pruebas para alcanzar requerimientos de prueba correctamente definidos cumpliendo criterios de prueba adecuados y estándares de garantía de calidad.



Se establecen las siguientes actividades para lograr encontrar errores y verificar la performance de nuestro sistema.

1. Determinar el alcance del plan de testing, y donde se hará foco.
2. Validar la calidad de la información de entrada.
3. Determinar la calidad de los parámetros de evaluación como radio de reconocimiento, calidad de la imagen, exactitud (accuracy).
4. Realizar pruebas a las diferentes funcionalidades del sistema.



6.2.4.2 Plan de Pruebas

PLAN DE TESTING – “SAURON”

SISTEMA DE INTERPRETACIÓN DE IMÁGENES PARA LA DETECCIÓN DE ARMAS

Identificador del Plan de Testing

PT-SIIDA-2019-01

Introducción

Quienes suscriben deseamos desarrollar un sistema para la detección de armas mediante la interpretación de imágenes.

Las siguientes secciones describen la naturaleza del sistema bajo prueba, los objetivos de la prueba y la amplitud. También se provee una lista de documentos relacionados.

Naturaleza del Proyecto

Este sistema permitirá al cliente que instale la aplicación detectar armas de fuego blancas a través de la interpretación de imágenes captadas por cámaras de seguridad en tiempo real.

Objetivos y Amplitud de la Prueba del Sistema

Este plan de pruebas describe el entorno, actividades, tareas, herramientas y cronogramas para el testing del sistema “SAURON, SISTEMA DE INTERPRETACIÓN DE IMÁGENES PARA LA DETECCIÓN DE ARMAS”, el cual es desarrollado por el área de desarrollo.

Nuestro objetivo para el testing del sistema es asegurar que:

- todas las funcionalidades requeridas están presentes y operan correctamente de acuerdo con los requerimientos;
- el software alcanza sus requerimientos de performance;
- el software es confiable y robusto, se recupera adecuadamente de fallas;
- el software es fácil de usar y mantener;
- se relaciona correctamente con el software y hardware externo requerido.



Un conjunto completo de pruebas será realizado, incluyendo:

- Prueba Funcional: con esta prueba se pretende verificar que el sistema integrado de hardware y software cumple los requisitos especificados. El objetivo perseguido es determinar cuándo está preparado el sistema para llevar a cabo la prueba de aceptación que se realiza en el entorno del usuario. En la prueba de sistema se verifican los requisitos del sistema y las funciones que realiza. Se utilizará la estrategia de Caja Negra, también denominada “conducida por los datos” o “conducida por la entrada/salida”. Esta estrategia considera al software como una caja negra, es decir, no considera el detalle procedimental de los programas. Se centra en buscar situaciones donde el programa no se ajusta a su especificación, utilizando precisamente ésta como entrada para derivar los casos de prueba. Los casos de prueba consistirán en conjuntos de datos de entrada que deberán generar una salida en concordancia con la especificación.
- Pruebas de Seguridad (login y permisos de acceso a módulos y funcionalidades).

Documentos Relacionados

Los siguientes documentos son usados como fuentes de información para el desarrollo de este Plan de Testing:

Documento	Descripción
DefinicióndeAlcance.docx	Define el alcance que tendrá el proyecto.
EDT.docx	Estructura de Desglose de Trabajo. Consiste en la descomposición jerárquica del trabajo a realizar.
PlandeAnálisisyDiseño.docx	Define el plan de análisis y diseño.
PlanficiaciónFinanciera.exe	Estimación de costos y ganancias esperada.
SeguimientoCronograma.mpj	Diagrama de Gantt de las tareas planificadas cronológicamente.



Ítems Por Probar

Todos los ítems que constituyen el sistema serán testeados durante la prueba del sistema para asegurar que funcionan en conjunto, e implementan los requerimientos del cliente. Estos ítems están listados a continuación con su identificador y número de versión.

Nombre	Identificador	Número de Versión
Modulo Red Neuronal	MCNN	1.0.0

Características para Probar

La característica para probar será: “Identificación de armas a través de imagen captada por cámara web”.

Aproximación

Las pruebas las llevarán a cabo el personal del área de Testing y Calidad, cuya misión es “Asegurar la calidad de los productos desarrollados”.

Fuentes de Datos de Ejemplos del Dominio

Para realizar las pruebas con datos del dominio del sistema, se requerirá contar con arma de juguete.

Personal

Dos analistas de Testing y Calidad serán responsables del diseño de los casos de prueba, la ejecución de las pruebas y la registración de los resultados. Serán supervisados por el líder de proyecto de Testing y Calidad.

Registro

Todos los eventos observados durante la ejecución serán guardados en un log, que estará asociado con cada prueba, los archivos relativos a las pruebas serán ubicados en un repositorio de datos (TC-SIIDA) en GITHUB. Todas las fallas tendrán asignadas un nivel de severidad y serán registradas en un reporte de incidente/problema. Todos los defectos encontrados durante la prueba del sistema serán registrados en un repositorio de defectos. La productividad del tester será monitoreada usando una medida de número de



casos de prueba construidos por semana y el número de casos de prueba ejecutados por semana.

Estado de la Prueba e Indicadores

La prueba del sistema será seguida con reuniones semanales entre el personal del área de Testing y Calidad, para revisar el estado de esta. El número de requisitos cubiertos por semana contra el número total de requisitos será uno de los índices utilizados para seguir progreso de la prueba. Otros índices que se obtendrán incluirán el número de defectos de un nivel específico de severidad encontrados por semana, defectos encontrados en cada ciclo de prueba, cantidad de ciclos de pruebas realizados, cantidad de casos de prueba elaborados por semana, cantidad de casos de prueba ejecutados por semana.

Herramientas de Testing

Al realizarse pruebas funcionales manuales, no se utilizarán herramientas específicas de Testing, solo se usarán utilitarios para documentar los casos de prueba, sus resultados y el registro de defectos. Se utilizará el GITHUB para versionar la documentación utilizada.

Criterio de Terminación de la Prueba

La decisión de terminar la prueba del sistema estará basada en (i) la cobertura de todos los requerimientos, (iii) tiempo de respuesta, (iv) porcentaje de precisión mayor a 85% y (v) el número de reportes de incidentes o problemas abiertos. Consideraremos que la prueba del sistema estará completada cuando todos los requerimientos hayan sido cubiertos con al menos un caso de prueba, y no haya reportes de problemas o incidentes asociados con defectos de nivel 1-3 (impacto crítico-medio). Aceptaremos el riesgo que el software aún contendrá defectos de bajo nivel de severidad que tendrán un efecto mínimo en los usuarios.

Criterio de Aprobación

Los niveles de severidad para los defectos se distribuyen en una escala de niveles, la cual se extiende de valores de 1 a 4, donde 1 es una falta que tiene un efecto crítico en el sistema/usuario, y 4 indica un efecto bajo sobre el sistema/usuario. Para este sistema en cuestión, una prueba será considerada como aprobada si la falta observada es clasificada en un nivel de 3 o 4. Eso significa que esa prueba puede continuar; sin embargo, todas las faltas y defectos asociados deben ser registrados y ser tratados. Los reportes de incidentes



de la prueba y los reportes de problema/defecto deben ser terminados para todas las fallas observadas. Todas las fallas se deben remitir a desarrollo y priorizarlas para su subsecuente reparación, seguido por la prueba de regresión realizada por el grupo de prueba.

Criterio de Suspensión / Recomienzo

Normalmente la prueba se suspenderá al final del día. Todos los documentos relacionados con la prueba deben ser enviados a la carpeta <\\files2\ARCHIVOS\sistemas\tyc>. Las pruebas se recomenzarán al comienzo del día siguiente. Además, las pruebas se suspenderán en los siguientes casos:

- una falla de nivel 1 o 2 de severidad es observada.
- ocurre una falla en el entorno.
- ocurre una falla de hardware.

Cuando el defecto causante de la falla del software es reparado, se ejecutará una prueba de regresión a la nueva versión del software. Si la nueva versión pasa la prueba de regresión, entonces la prueba normal puede ser recomenzada.

Si durante las pruebas del sistema hay una falla de hardware o de entorno, el tester notificará al área correspondiente y recomenzará las pruebas cuando las reparaciones hayan sido hechas, arrancando las pruebas desde el principio.

Entregables de la Prueba

Los siguientes ítems serán generados por el área de Testing para este proyecto.

- Plan de Testing del sistema: documento que describe el alcance, acercamiento, recursos y cronograma de las actividades de la prueba prevista.
- Especificaciones de casos de prueba del sistema: documento que muestra entradas, resultados previstos y un conjunto de condiciones de ejecución para un ítem de la prueba.
- Especificaciones de procesos de prueba del sistema: documento que exhibe una secuencia de acciones para la ejecución de una prueba.
- Registro de las pruebas del sistema: listado cronológico de detalles relevantes sobre la ejecución de pruebas.
- Reporte de incidentes de pruebas del sistema: documento que reporta cualquier acontecimiento que ocurra durante el proceso de la prueba que requiera una investigación.



- Reporte general de pruebas del sistema: documento que sintetiza las actividades y los resultados de la prueba.

Tareas de Testing

A continuación, se muestra un listado de tareas a efectuar por el personal del área de Testing y Calidad: Listado de Tareas de Testing:

- Preparación de Ambiente de Testing
- Preparación de las especificaciones de los casos de prueba
- Diseño de casos de prueba
- Revisión de Casos de Prueba
- Ejecución de casos de prueba y registro de los resultados
- Comunicación a las áreas interesadas los resultados de las pruebas

Entorno de la Prueba

El sistema en cuestión está basado en una arquitectura de tecnología Python bajo las librerías que componen el paquete “Anaconda”. El IDE utilizado para las pruebas es Spyder.

Cronogramas

A continuación, se detalla el cronograma de la prueba

Actividad/Semana	1	2	3	4	5	6	7
Planificación de pruebas	■						
Estudio preliminar y preparación de ambiente		■	■				
Diseño de casos de prueba			■	■			
Ejecución de casos de prueba ciclo 1				■	■		
Ejecución de casos de prueba ciclo 2						■	
Ejecución de casos de prueba ciclo 3							■
Pruebas de aceptación							■

TABLA N°33. CRONOGRAMA EN SEMANAS PARA LA ETAPA DE PRUEBAS DEL SISTEMA.



Casos de Prueba:

Id caso de prueba: TC-SIIDA-001-01	Tester: Osatinsky, Agustín
Fecha: 28/10/2019	En tomo: testing
Sistema: SIIDA	Base de datos:
Versión: 1.0.0	Ciclo: 1
Revisión:	Pantalla/Modulo/Caso de Uso: Módulo de interpretación de imagen
Tipo de prueba: Unidad Integridad Sistema Aceptación	
Objetivos de la prueba: Interpretar imagen de un arma. Válido.	
<p>Prerrequisitos de la prueba:</p> <ol style="list-style-type: none"> 1. La red debe estar entrenada con imágenes de armas. 2. Cámara conectada correctamente a la computadora. 	
<p>Procedimiento:</p> <ol style="list-style-type: none"> 1. Se acerca el arma dentro del rango de visión de la cámara. 2. Se deberá apretar la tecla espacio para tomar una fotografía. 	
<p>Resultados esperados:</p> <ol style="list-style-type: none"> 1. Se detecta el arma dentro del rango de la imagen mostrada. 2. El contorno del arma es marcado para indicar la ubicación de esta dentro de la imagen. 	
Resultados obtenidos: El arma fue satisfactoriamente detectada.	
Observaciones:	
<p>Resultados de la prueba: Aprobado</p> <p>No aprobado</p>	

TABLA N°34. CASO DE PRUEBA DEL SISTEMA N°1



Id caso de prueba: TC-SIIDA-001-02	Tester: Ceriana, Emanuel
Fecha: 30/10/2019	En tomo: testing
Sistema: SIIDA	Base de datos:
Versión: 1.0.0	Ciclo: 1
Revisión:	Pantalla/Modulo/Caso de Uso: Módulo de interpretación de imagen
Tipo de prueba: Unidad Integridad Sistema Aceptación	
Objetivos de la prueba: Interpretar imagen de un arma con un porcentaje precisión mayor o igual a 85%. Válido.	
Prerrequisitos de la prueba:	
<ol style="list-style-type: none"> 1. La red debe estar entrenada con imágenes de armas. 2. Cámara conectada correctamente a la computadora. 	
Procedimiento:	
<ol style="list-style-type: none"> 1. Se acerca el arma dentro del rango de visión de la cámara. 2. Se deberá apretar la tecla espacio para tomar una fotografía. 	
Resultados esperados:	
<ol style="list-style-type: none"> 1. Se detecta el arma dentro del rango de la imagen mostrada. 2. El contorno del arma es marcado para indicar la ubicación de esta dentro de la imagen. 	
Resultados obtenidos: El arma fue satisfactoriamente detectada pero no respeta el porcentaje de precisión porcentaje de precisión el cual es igual a 81%.	
Observaciones Se deberá revisar el módulo para ajustar los porcentajes y alcanzar el nivel de precisión deseado.	
Resultados de la prueba: Aprobado	
No aprobado	

TABLA N°35. CASO DE PRUEBA DEL SISTEMA N°2



Id caso de prueba: TC-SIIDA-001-03	Tester: Osatinsky, Agustín
Fecha: 31/10/2019	En tomo: testing
Sistema: SIIDA	Base de datos:
Versión: 1.0.0	Ciclo: 1
Revisión:	Pantalla/Modulo/Caso de Uso: Interfaz llamada a IA.
Tipo de prueba: Unidad Integridad Sistema Aceptación	
Objetivos de la prueba: Validar la llamada a la IA.	
Prerrequisitos de la prueba:	
<ol style="list-style-type: none"> 1. La interfaz de llamada a la IA debe estar creada. callIA(imagen) 2. La IA deberá estar corriendo al momento de la llamada a la interfaz. 	
Procedimiento:	
<ol style="list-style-type: none"> 1. Se elige la imagen deseada. 2. Se realiza la llamada a la interfaz con dicha imagen. 	
Resultados esperados:	
<ol style="list-style-type: none"> 1. Llamada realizada con éxito a través de un mensaje. 	
Resultados obtenidos: La llamada se realiza con éxito y el mensaje es el esperado.	
Observaciones:	
Resultados de la prueba: Aprobado	
No aprobado	

TABLA N°36. CASO DE PRUEBA DEL SISTEMA N°3



12. Defectos Encontrados:

Id defecto: DM-SIIDA-001	Tester: Ceriana, Emanuel												
Fecha: 02/11/2019	Defecto o mejora:												
Sistema: SIIDA	Título del problema: error en TC-SIIDA-001-02												
Versión: 1.0.0	Caso de prueba: TC-SIIDA-001-02												
Revisión:	Entorno: testing												
Área del defecto: Sistemas	Otros entornos:												
Descripción del problema: El porcentaje de precisión es menor del 85%													
<p>Tipos de defecto:</p> <table> <tr> <td>Arquitectónico</td> <td>GUI</td> </tr> <tr> <td>Conectividad</td> <td>Performance</td> </tr> <tr> <td>Consistencia</td> <td>Seguridad y controles</td> </tr> <tr> <td>Integridad de la BD</td> <td>Estándares convencionales</td> </tr> <tr> <td>Documentación</td> <td>Usabilidad</td> </tr> <tr> <td>Funcionalidad</td> <td>Otros</td> </tr> </table>		Arquitectónico	GUI	Conectividad	Performance	Consistencia	Seguridad y controles	Integridad de la BD	Estándares convencionales	Documentación	Usabilidad	Funcionalidad	Otros
Arquitectónico	GUI												
Conectividad	Performance												
Consistencia	Seguridad y controles												
Integridad de la BD	Estándares convencionales												
Documentación	Usabilidad												
Funcionalidad	Otros												
<p>Quien detecto:</p> <table> <tr> <td>Usuario interno</td> <td>Desarrollo</td> </tr> <tr> <td>Usuario externo</td> <td>Testing y calidad</td> </tr> </table>		Usuario interno	Desarrollo	Usuario externo	Testing y calidad								
Usuario interno	Desarrollo												
Usuario externo	Testing y calidad												



Como fue detectado:

Revisión	JAD
Recorrido	Prueba

Asignado a:

Prioridad:

Critico	Medio
Alto	Bajo

Severidad:

Critico	Medio
Alto	Bajo

Estado:

Abierto	Cerrado (Testing y calidad)
Siendo revisado por desarrollo	Devuelto por (Testing y calidad)
Devuelto por desarrollo	Diferido para la siguiente versión
Listo para probar en la siguiente versión	

Descripción del estado: Se arregló el error, la nueva prueba dio como resultado 87%.

Arreglado por: Osatinsky, Agustín

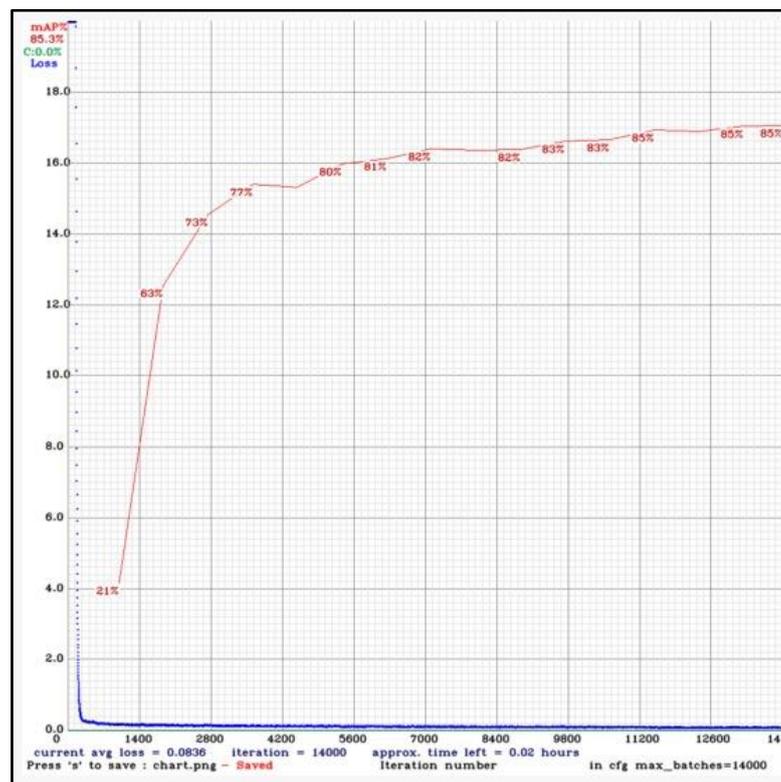
Arreglo previsto para la versión:

TABLA N°37. DEFECTO DETECTADO DEL SISTEMA N°1

7. Resultados Obtenidos

A lo largo del cursado de la materia Proyecto Final, hemos aplicado los conocimientos adquiridos en la carrera, lo que nos permitió crecer tanto personal como académicamente.

Con respecto a la realización del sistema planteado en este documento, logramos obtener resultados muy prometedores y satisfactorios en base a las funcionalidades y requerimientos planteados en un comienzo. Pudimos comprobar que el modelo de inteligencia artificial Tiny-YOLO V4 cumplió con el comportamiento esperado obteniendo una precisión media del 85% sobre el Dataset de prueba cómo es posible comprobar en la siguiente imagen. Además, podemos suponer que el uso de SAURON mejorará el tiempo de respuesta entre la detección de un arma y la notificación a las autoridades pertinentes.



8.

FIG. N° 34. MEAN AVERAGE PRECISION. MUESTRA EL PROGRESO DE LA PRECISIÓN DEL MODELO DURANTE EL ENTRENAMIENTO.

Si bien estamos satisfechos con los resultados de nuestra investigación, creemos que podemos mejorar el rendimiento y velocidad de detección de nuestro modelo de inteligencia artificial entrenando el mismo con un banco de imágenes más amplio y empleando técnicas de procesamiento más optimizadas.



Finalmente, basándonos en los resultados de este trabajo, podemos decir que es posible implementar esta solución para videos obtenidos a partir de cámaras de seguridad en tiempo real, para detectar situaciones de violencia. Creemos que el uso de estas herramientas es fundamental para mejorar calidad de vida de las personas y contribuir así al desarrollo y crecimiento de las Smart Cities.

8. Manual de Usuario

8.1. Pantalla Principal

Para poder tener acceso al sistema se deberá ingresar a través de un navegador web (Chrome, Firefox, Safari), en donde colocaremos la URL www.projectsauron.com en la barra de navegación.

Una vez dentro de la página web, podrá ver la siguiente pantalla principal:

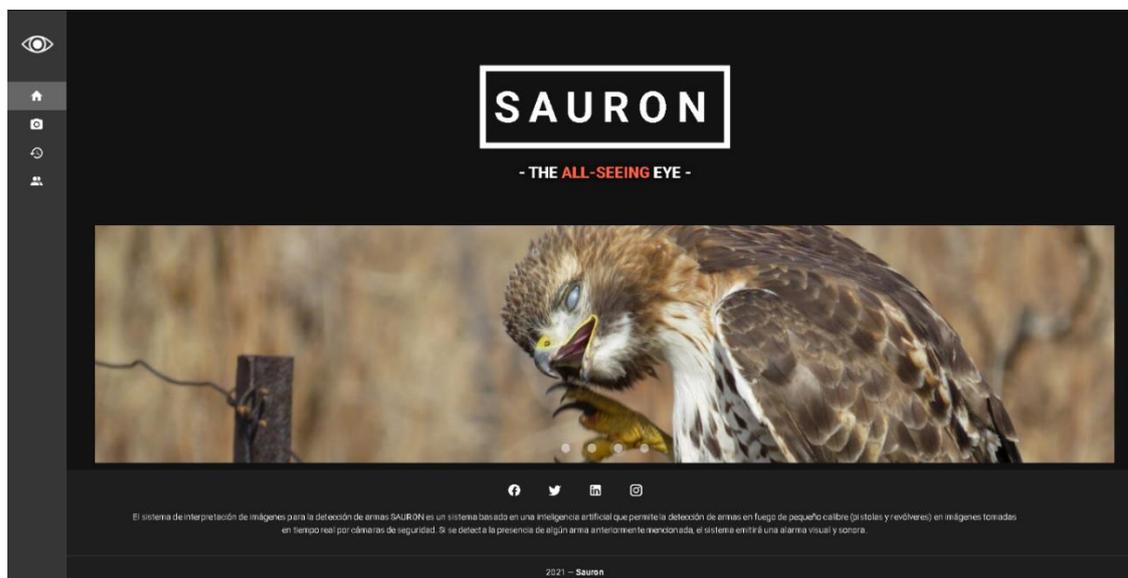


FIG. N°35. INTERFAZ DE USUARIO PANTALLA PRINCIPAL “SAURON” – VERSIÓN FINAL.

8.2. Sección cámaras

Esta sección de la aplicación permitirá al usuario configurar y visualizar de manera rápida hasta 4 cámaras en tiempo real. En caso de detectarse algún evento sospechoso por el sistema de inteligencia artificial, se notifica al usuario a través de una alerta visual (POP-UP).

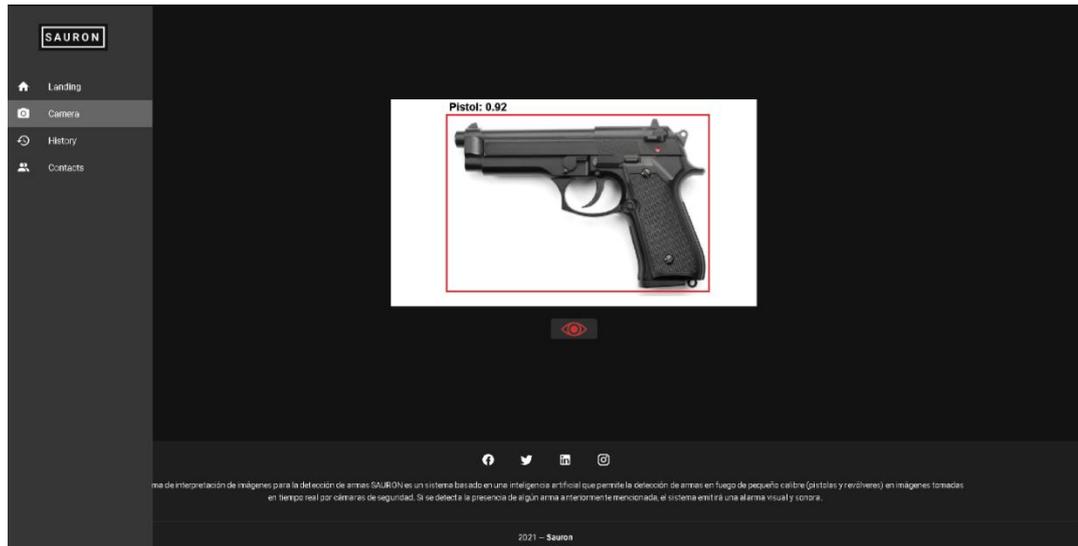


FIG. N°36. INTERFAZ DE USUARIO “SECCIÓN CÁMARAS” – VERSIÓN FINAL.

8.3. Sección Historial

En esta sección de la aplicación encontraremos el Historial de Alertas, donde el usuario podrá visualizar el historial de alertas emitidas por el sistema.

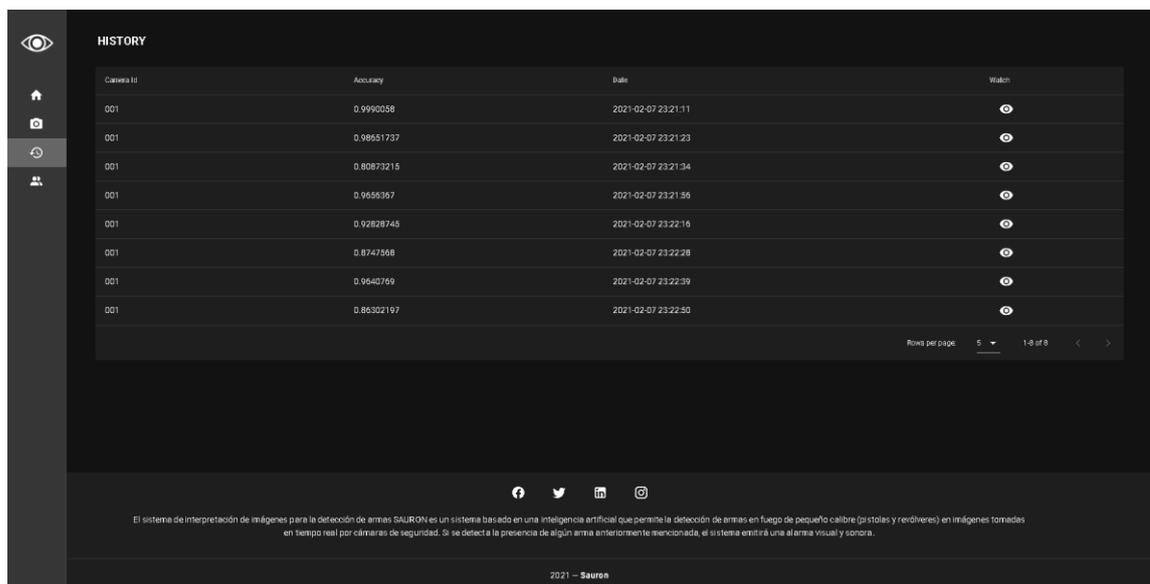


FIG. N°37. INTERFAZ DE USUARIO “SECCIÓN HISTORIAL DE ALARMAS” – VERSIÓN FINAL

El usuario podrá seleccionar un evento del historial para visualizar lo detectado. El sistema mostrará una captura del objeto detectado.

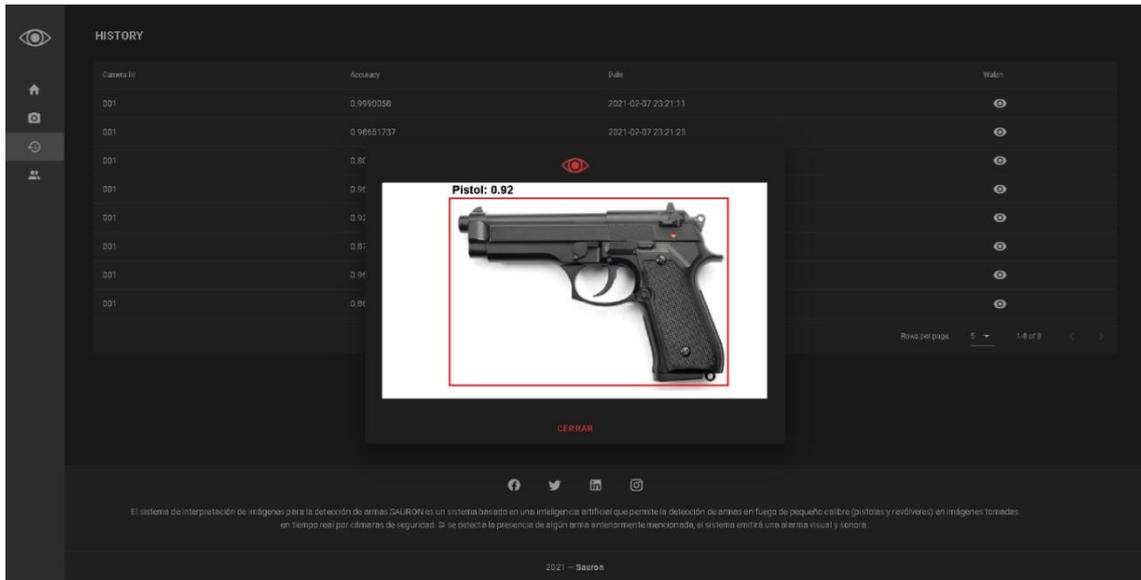


FIG. N°38. INTERFAZ DE USUARIO “SECCIÓN HISTORIAL DE ALARMAS” EVENTO SELECCIONADO - VERSIÓN FINAL.

8.4. Sección contactos

En esta sección de la aplicación es posible realizar la administración de contactos que serán notificación ante la detección de una situación de emergencia utilizando el servicio de SendGrid a sus respectivos emails.

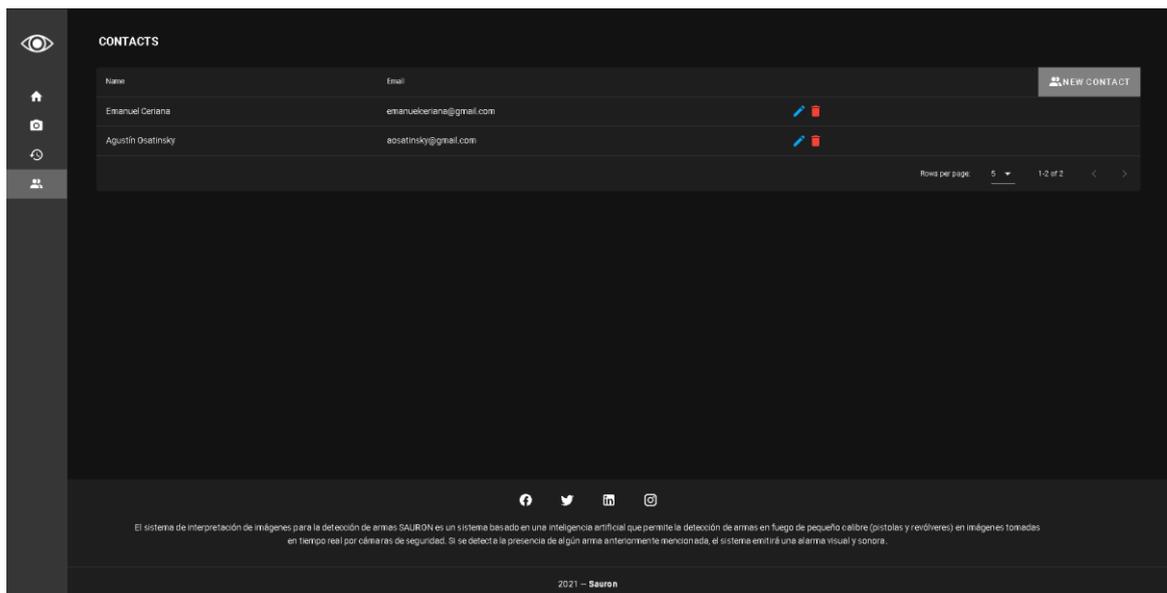


FIG. N°39. INTERFAZ DE USUARIO “SECCIÓN CONTACTOS” - VERSIÓN FINAL.

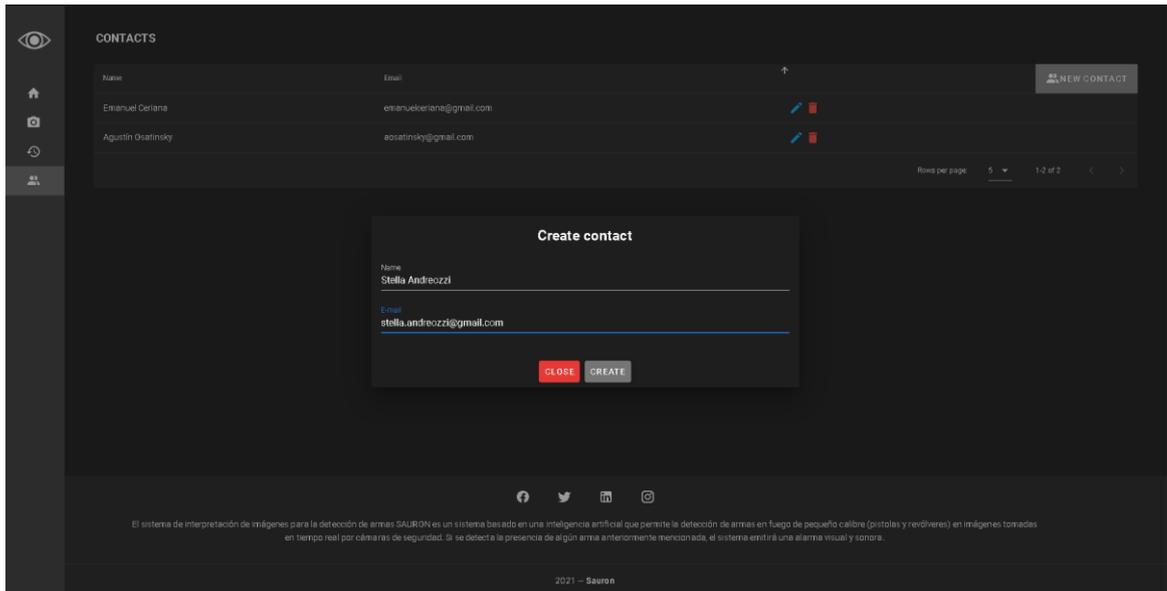


FIG. N°40. INTERFAZ DE USUARIO “AGREGAR NUEVO CONTACTO” – VERSIÓN FINAL.

8.5. Disparador de Alarma

Cuando el sistema detecta un arma se abre un dialogo pop up que indica que cámara fue la que detectó el arma. Al mismo tiempo se emitirá un sonido de alerta.

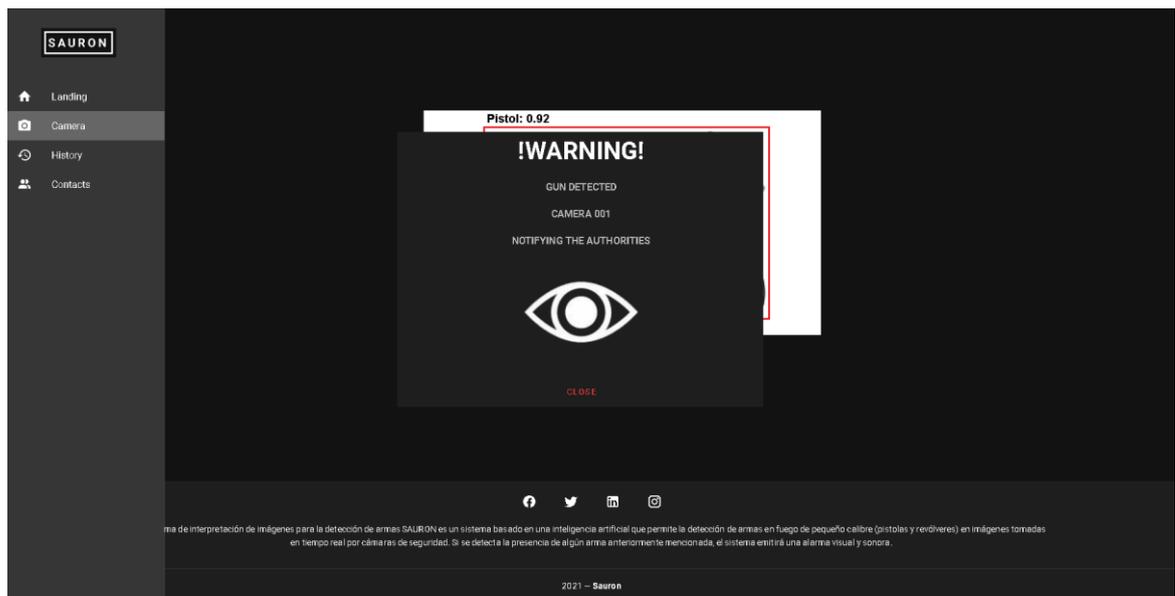


FIG. N°41. INTERFAZ DE USUARIO “SECCIÓN EVENTO DETECTADO - ALARMA” – VERSIÓN FINAL.

8.6. Notificación enviada a correo a través de SendGrid

Al detectar un evento el sistema genera una alarma, que será enviada por mail a la lista de contactos existente.

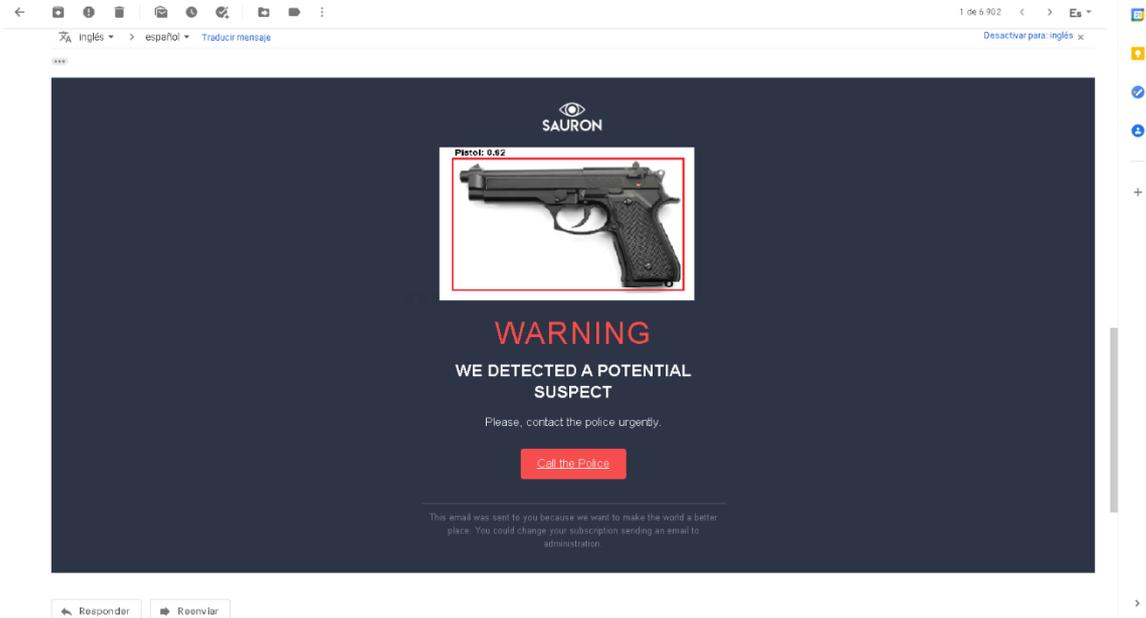


FIG. N°42. INTERFAZ DE USUARIO “NOTIFICACIÓN ENVIDADA” – VERSIÓN FINAL.



9. Manual de Instalación

Lo primero a realizar es la implementación de la infraestructura cloud/hardware necesario. En SendGrid es necesario dar de alta el servicio para permitir las notificaciones en tiempo real.

Una vez realizada la instalación física de la PC y de las cámaras será necesaria la configuración del servidor, para ello se debe seguir los siguientes pasos:

1. Instalar el servidor web NGIX: Dado a que Nginx está disponible en los repositorios predeterminados, es posible instalarlo usando el sistema de empaquetado apt.

- a. Ejecutar las siguientes instrucciones en la línea de comando:

```
sudo apt update
```

```
sudo apt install nginx
```

2. Instalar Python 3.8: es necesario correr las siguientes instrucciones a través de la línea de comando:

```
sudo apt-get update
```

```
sudo apt-get install python3.8
```

3. Una vez instalado Python debemos correr el siguiente comando para instalar las dependencias necesarias

```
pip install -r requirements.txt
```

Por último, resta por descargar los paquetes correspondientes al proyecto. Una vez ejecutado los scripts de configuración. El sistema será accesible a través de la siguiente dirección web:

<http://localhost:3000>

Además, es necesario configurar las siguientes variables de entorno:

- A. SENDGRID_API_KEY: API KEY del servicio SendGrid encargado de las notificaciones en tiempo real.
- B. CAM_URL_1: URL para la cámara número 1. Debe cumplir con el protocolo RTSP.
- C. CAM_URL_2: URL para la cámara número 2. Debe cumplir con el protocolo RTSP.
- D. CAM_URL_3: URL para la cámara número 3. Debe cumplir con el protocolo RTSP.
- E. CAM_URL_4: URL para la cámara número 4. Debe cumplir con el protocolo RTSP.
- F. MASTER_USER: nombre de usuario principal del sistema. Necesario para ingresar.



- G. MASTER_PASSWORD: contraseña del usuario principal del sistema. Necesario para ingresar.
- H. SQLITE_STRING: cadena de conexión de la base de datos SQLite.



10. Incumbencias del Ingeniero en Sistemas de Información

1. Especificar, proyectar y desarrollar sistemas de información, sistemas de comunicación de datos y software cuya utilización pueda afectar la seguridad, salud, bienes o derechos.
2. Proyectar y dirigir lo referido a seguridad informática.
3. Establecer métricas y normas de calidad de software.
4. Certificar el funcionamiento, condición de uso o estado de lo mencionado anteriormente.
5. Dirigir y controlar la implementación, operación y mantenimiento de lo anteriormente mencionado.



11. Conclusiones

Es muy probable que la inteligencia artificial transforme radicalmente nuestro pensamiento y nuestros enfoques prácticos hacia la fuerza armada y la defensa. Este trabajo, presenta un novedoso sistema de detección automática de emergencias en videos apropiados para propósitos de vigilancia y control. Se han obtenido resultados prometedores con nuestro modelo de IA basado en tiny-YOLO V4, reentrenado con nuestro conjunto de datos que ofrece un alto potencial incluso en videos de YouTube de baja calidad y proporciona resultados muy satisfactorios como sistema de alarma automático. Este novedoso sistema fue diseñado principalmente para utilizarse en la detección en tiempo real de armas en lugares controlados por cámaras, pero podría servir también para controlar, por ejemplo, si los videos subidos a las redes sociales contienen escenas con armas.

Se ha demostrado que el uso de operadores humanos de CCTV para monitorear cámaras de video tiene ciertas limitaciones. Los diferentes autores revisados coinciden en que existe la necesidad de soluciones de vigilancia automatizadas para mitigar estos desafíos. Como aporte de esta investigación se busca mejorar drásticamente el tiempo de respuesta de las autoridades ya que uno de los principales objetivos de este proyecto es el tratamiento y auxilio de las personas que se encuentren en una situación que atente contra su integridad.

El campo de la visión artificial está en constante crecimiento y este proyecto aportaría una gran base de conocimiento aplicada a la seguridad y el bienestar del ciudadano, utilizando técnicas de inteligencia artificial. Cabe destacar que el sistema podría implementarse con cámaras que ya se encuentran instaladas siendo de gran utilidad para los gobiernos (que cuentan con cámaras de seguridad en la vía pública) o un usuario final en su casa o empresa.



12. Bibliografía

- ¿Por qué Flask? (s.f.). Obtenido de <https://codigofacilito.com/articulos/por-que-flask>
- A. Bochkovskiy, C. W. (2020). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. Corner University.
- Agile Framework for AI Projects*. (s.f.). Obtenido de <https://towardsdatascience.com/an-agile-framework-for-ai-projects-6b5a1bb41ce4?gi=5cc3a95aaac6>
- ecured.cu*. (s.f.). Obtenido de https://www.ecured.cu/Arquitectura_en_Capas
- es.vuejs.org*. (s.f.). Obtenido de <https://es.vuejs.org/v2/guide/>
- FrontEnd y BackEnd*. (s.f.). Obtenido de programacionymas.com:
<https://programacionymas.com/frontend-vs-backend>
- G. Gkioxari, R. G. (2017). Detecting and Recognizing Human-Object Interactions. 1704.
- Girshick, R. (2015). Fast R-CNN. *IEEE International Conference on Computer Vision (ICCV)*, 1440-1448.
- I., T. G. (2018). Global Mortality From Firearms. *JAMA*, 792-814.
- Introduction to YOLO V4*. (s.f.). Obtenido de <https://medium.com/analytics-vidhya/introduction-to-yolov4-object-detection-fcba8bb72449>
- J. Suss, F. V. (2015). Don't Overlook the Human! Applying the Principles of Cognitive Systems Engineering to the Design of Intelligent Video Surveillance Systems. *IEEE International Conference on Advanced Video and Signal*.
- K. He, G. G. (2017). Mask R-CNN. *IEEE International Conference on Computer Vision (ICCV)*, 2980-2988.
- PMI. (s.f.). *Dirección de Proyectos PMI*. Obtenido de <https://www.studocu.com/pe/document/universidad-tecnologica-del-peru/estadistica-descriptiva-y-probabilidades/informe/la-direccion-de-proyectos/9319779/view>
- SendGrid. (s.f.). *SendGrid*. Obtenido de https://app.sendgrid.com/login?redirect_to=%2F
- SQLite*. (s.f.). Obtenido de <https://www.sqlite.org/about.html>
- T. Lin, P. D. (2017). Feature Pyramid Networks for Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 936-944.
- W. Shi, J. C. (2016). "Edge Computing: Vision and Challenges, ". *IEEE Internet of Things Journal*.