





**UNIVERSIDAD NACIONAL DE CHIMBORAZO**  
**FACULTAD DE INGENIERÍA**  
**INGENIERÍA EN TELECOMUNICACIONES**

Diseño e implementación de un sistema inteligente basado en visión artificial para el monitoreo y prevención de robos en tiempo real en la farmacia Pharmatodo de la ciudad de  
Riobamba

**TRABAJO DE TITULACIÓN PARA OBTENER EL TÍTULO DE INGENIERO EN  
TELECOMUNICACIONES**

**AUTOR:**

Villa Yuquilema Jimmy Andres

**TUTOR:**

Mag. Cuzco Silva Edgar Giovanni

**Riobamba, Ecuador.2024**

## **DERECHOS DE AUTORÍA**

Yo, Jimmy Andrés Villa Yuquilema, con cédula de ciudadanía 060438218-4, autor del trabajo de investigación titulado: diseño e implementación de un sistema inteligente basado en visión artificial para el monitoreo y prevención de robos en tiempo real en la farmacia Pharmatodo de la ciudad de Riobamba, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, el 12 de agosto del 2024.



---

Jimmy Andrés Villa Yuquilema  
0604382184

## **ACTA FAVORABLE-INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN**

En la Ciudad de Riobamba, a los 14 días del mes de octubre de 2024, luego de haber revisado el Informe Final del Trabajo de Investigación presentado por el estudiante **JIMMY ANDRÉS VILLA YUQUILEMA** con Ci: **0604382184**, de la carrera de **INGENIERÍA EN TELECOMUNICACIONES** y dando cumplimiento a los criterios metodológicos exigidos, se emite el **ACTA FAVORABLE DEL INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN** titulado **“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INTELIGENTE BASADO EN VISIÓN ARTIFICIAL PARA EL MONITOREO Y PREVENCIÓN DE ROBOS EN TIEMPO REAL EN LA FARMACIA PHARMATODO DE LA CIUDAD DE RIOBAMBA”**, por lo tanto se autoriza la presentación del mismo para los trámites pertinentes.



---

Mgs. Edgar Giovanni Cuzco Silva  
**TUTOR(A)**

## CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL

Quienes suscribimos, catedráticos designados Miembros del Tribunal de Grado para la evaluación del trabajo de investigación “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INTELIGENTE BASADO EN VISIÓN ARTIFICIAL PARA EL MONITOREO Y PREVENCIÓN DE ROBOS EN TIEMPO REAL EN LA FARMACIA PHARMATODO DE LA CIUDAD DE RIOBAMBA”, presentado por **JIMMY ANDRES VILLA YUQUILEMA**, con cédula de identidad número **060438218-4**, bajo la tutoría de Mgs. Edgar Giovanni Cuzco Silva, certificamos que recomendamos la **APROBACIÓN** de este con fines de titulación. Previamente se ha evaluado el trabajo de investigación y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en Riobamba 06 de diciembre de 2024.

Dr. Klever Torres  
**PRESIDENTE DEL TRIBUNAL DE GRADO**



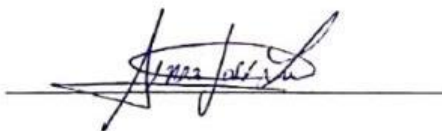
---

PhD. Yesenia Cevallos  
**MIEMBRO DEL TRIBUNAL DE GRADO**



---

Msc. José Jinez  
**MIEMBRO DEL TRIBUNAL DE GRADO**



---

## CERTIFICADO ANTI PLAGIO

# CERTIFICACIÓN

Que, **JIMMY ANDRÉS VILLA YAQUILEMA** con CC: **060438218-4**, estudiante de la Carrera de **TELECOMUNICACIONES**, Facultad de **INGENIERÍA**; ha trabajado bajo mi tutoría el trabajo de investigación titulado **"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA INTELIGENTE BASADO EN VISIÓN ARTIFICIAL PARA EL MONITOREO Y PREVENCIÓN DE ROBOS EN TIEMPO REAL EN LA FARMACIA PHARMATODO DE LA CIUDAD DE RIOBAMBA"**, cumple con el **9%**, de acuerdo al reporte del sistema Anti plagio **TURNITIN**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 29 de noviembre de 2024



Firmado electrónicamente por:  
**EDGAR GIOVANNY**  
**CUSCO SILVA**

---

Mgs. Edgar Giovanni Cusco Silva  
**TUTOR**

## **DEDICATORIA**

*El siguiente trabajo de investigación está dedicado a mis padres, por su amor incondicional, apoyo constante y sacrificios incansables que hicieron posible este logro. A mi director de tesis Mag. Edgar Giovanny Cuzco Silva, cuya guía experta me llevaron a alcanzar mis metas académicas. A mis amigos y seres queridos, por sus palabras de aliento y alegría compartida en cada etapa de este viaje.*

## **AGRADECIMIENTO**

*Quiero expresar mi sincero agradecimiento, en primer lugar, a mi madre Olga Yuquilema a mi padre Luis Villa a mis hermanos Fanny Villa y Francisco Villa y a mi adorable sobrina Saory Ramos por su apoyo incondicional y por estar siempre a mi lado durante este proceso. Agradezco profundamente a mi director de tesis, Mag. Edgar Giovanny Cuzco Silva, por su guía experta, apoyo constante y por brindarme la oportunidad de explorar este tema en profundidad. No puedo dejar de reconocer el apoyo incondicional de mis amigos, quienes estuvieron a mi lado brindándome ánimo. Finalmente, quiero expresar mi gratitud a todas las personas que participaron en este estudio, por su colaboración y disposición para que este proyecto de investigación se pueda culminar.*

*Su contribución fue fundamental para el desarrollo de esta investigación.*

*¡Muchas gracias a todos!*



## ÍNDICE GENERAL

**DERECHOS DE AUTORÍA**

**ACTA FAVORABLE-INFORME FINAL DEL TRABAJO DE INVESTIGACIÓN**

**CERTIFICADO DE LOS MIEMBROS DEL TRIBUNAL**

**CERTIFICADO ANTI PLAGIO**

**DEDICATORIA**

**AGRADECIMIENTO**

**ÍNDICE GENERAL**

**ÍNDICE DE TABLAS**

**ÍNDICE DE FIGURAS**

**RESUMEN**

**ABSTRAC**

<b>CAPÍTULO I. INTRODUCCIÓN .....</b>	<b>17</b>
1.1 PLANTEAMIENTO DEL PROBLEMA .....	17
1.2 JUSTIFICACIÓN .....	17
1.3 OBJETIVOS .....	18
1.3.1 GENERAL .....	18
1.3.2 ESPECÍFICOS .....	18
<b>CAPITULO II. MARCO CONCEPTUAL .....</b>	<b>19</b>
<b>2. ESTADO DEL ARTE.....</b>	<b>19</b>
2.1 MARCO CONCEPTUAL .....	21
2.1.1 INSEGURIDAD EN ECUADOR.....	21
2.1.2 IMPACTO PSICOLÓGICO EN EMPLEADOS .....	22
2.1.3 IMPACTO ECONÓMICO DE LA INSEGURIDAD.....	23
2.1.4 MÉTODOS DE PREVENCIÓN Y MITIGACIÓN DE ROBOS .....	23
2.2 VISIÓN ARTIFICIAL.....	25
2.2.1 ETAPAS DEL PROCESAMIENTO DE IMÁGENES CON VISIÓN ARTIFICIAL .....	25
2.2.2 ALGORITMOS DE VISIÓN ARTIFICIAL.....	26
2.2.3 ALGORITMOS DE DETECCIÓN DE OBJETOS .....	27

2.2.4 ALGORITMOS DE DETECCIÓN Y SEGUIMIENTO DE POSTURA HUMANA .....	28
<b>CAPITULO III. METODOLOGIA .....</b>	<b>29</b>
<b>3. TIPO DE INVESTIGACIÓN .....</b>	<b>29</b>
3.1 INVESTIGACIÓN EXPERIMENTAL.....	29
3.2 INVESTIGACIÓN DESCRIPTIVA .....	29
3.3 POBLACIÓN Y TAMAÑO DEMUESTRA.....	29
3.3.1 POBLACIÓN.....	29
3.3.2 MUESTRA.....	29
3.4 DISEÑO DE LA INVESTIGACIÓN .....	30
3.5 OPERACIÓN DE LAS VARIABLES .....	31
3.5.1 VARIABLES INDEPENDIENTES.....	31
3.5.2 VARIABLES DEPENDIENTES .....	31
3.6 DIAGRAMA MULTICAPA DEL PROCESO DE DESARROLLO DEL SISTEMA DE SEGURIDAD .....	32
3.6.1 FUNCIONAMIENTO DEL SISTEMA DE SEGURIDAD .....	33
3.6.2 FASE 1: SELECCIÓN DE MÉTODOS Y ALGORITMOS .....	33
3.6.3 FASE 2: CREACIÓN DE LA BASE DE DATOS Y ETIQUETADO.....	37
3.6.4 FASE 3: ENTRENAMIENTO YOLOV5.....	42
3.6.5 FASE 4: DESARROLLO DE LA CODIFICACIÓN EN PYTHON PARA LA IMPLEMENTACIÓN DE FUNCIONALIDADES DE ENVÍO DE INFORMACIÓN, CAPTURA DE IMÁGENES Y DETECCIÓN DE OBJETOS.....	47
3.6.6 FASE 5: DESARROLLO Y DESPLIEGUE DE LA APLICACIÓN CENTINELADEFENDER. ....	52
3.6.7 ANÁLISIS Y COMPROBACIÓN DE LA FUNCIONALIDAD DEL SISTEMA	
59	
<b>CAPITULO IV RESULTADOS.....</b>	<b>81</b>
<b>4. RESULTADOS DE LAS PRUEBAS EN AMBIENTES IDEAL, REAL Y DE ROBUSTEZ .....</b>	<b>81</b>
4.1 ANÁLISIS DE RESULTADOS ESTADÍSTICOS .....	87
<b>CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>109</b>
<b>5. CONCLUSIONES.....</b>	<b>109</b>
5.1 RECOMENDACIONES.....	110

<b>BIBLIOGRAFÍA .....</b>	<b>112</b>
<b>ANEXOS.....</b>	<b>114</b>

## ÍNDICE DE TABLAS

<b>Tabla 1</b> Variables Independientes .....	31
<b>Tabla 2</b> Variables Dependientes .....	31
<b>Tabla 3.</b> Ventajas y desventajas de YOLO .....	34
<b>Tabla 4.</b> Ventajas y desventajas de Haar Cascade Trairner .....	35
<b>Tabla 5.</b> Ventajas y desventajas red neuronal convolucional (CNN) .....	36
<b>Tabla 6</b> Tabla de evaluación en un ambiente ideal para un arma de fuego .....	63
<b>Tabla 7</b> Tabla de evaluación en un ambiente ideal para un arma blanca(cuchillo) .....	65
<b>Tabla 8</b> Tabla de evaluación en un ambiente real para un arma de fuego .....	70
<b>Tabla 9</b> Tabla de evaluación en un ambiente real para un arma blanca .....	76
<b>Tabla 10</b> Tabla de Evaluación en un Entorno de Robustez y Estrés (arma de fuego).....	78
<b>Tabla 11</b> Tabla de Evaluación en un Entorno de Robustez y Estrés (arma blanca) .....	81
<b>Tabla 12</b> Tabla De Evaluación Realizada En Un Ambiente Ideal.....	82
<b>Tabla 13</b> Tabla De Evaluación Realizada En Un Ambiente.....	83
<b>Tabla 14</b> Tabla De Evaluación Realizada En Un Ambiente Real Utilizando Un Arma Blanca .....	84
<b>Tabla 15</b> Tabla De Evaluación Realizada En Un Ambiente De Robustes Y Estrés Utilizando Un Arma Un Arma De Fuego Corta.....	85
<b>Tabla 16</b> Tabla De Evaluación Realizada En Un Ambiente De Robustes Y Estrés Utilizando Un Arma Blanca Cuchillo .....	86
<b>Tabla 17</b> Carta de aceptación.....	118

## ÍNDICE DE FIGURAS

<b>Figura 1</b> Diseño de la investigación .....	30
<b>Figura 2</b> Diagrama multicapa .....	32
<b>Figura 3</b> Arma mostrada completamente .....	38
<b>Figura 4</b> Arma mostrada parcialmente .....	38
<b>Figura 5</b> Arma apenas visible .....	38
<b>Figura 6</b> Usuario sosteniendo un arma de fuego en diferente posición.....	39
<b>Figura 7</b> Usuario sosteniendo un arma de fuego .....	39
<b>Figura 8</b> Arma de fuego con escenario diferente.....	39

<b>Figura 9</b> Arma de fuego con escenario diferente.....	39
<b>Figura 10</b> Make Sense .....	40
<b>Figura 11</b> Definimos la etiqueta .....	41
<b>Figura 12</b> Proceso de etiquetado.....	41
<b>Figura 13</b> Proceso de descarga de etiquetas .....	42
<b>Figura 14</b> Estructura de la base de datos .....	43
<b>Figura 15</b> Configuración del archivo YAML .....	44
<b>Figura 16</b> Configuración del Algoritmo Yolov5 .....	46
<b>Figura 17</b> Entrenamiento del modelo Yolo v5 .....	47
<b>Figura 18</b> Configuración de Firebase y carga del modelo de detección de objetos .....	48
<b>Figura 19</b> Control de envío de notificaciones.....	49
<b>Figura 20</b> Captura y envío de imágenes a Firebase Storage durante la detección en tiempo real .....	50
<b>Figura 21</b> Procesamiento de Video en Tiempo Real para Detección de Manos Levantadas y Registro en Firebase .....	51
<b>Figura 22</b> Interfaz principal .....	54
<b>Figura 23</b> Interfaz de pantalla de selección de contactos de emergencia .....	55
<b>Figura 24</b> A_1.....	61
<b>Figura 25</b> A_2.....	61
<b>Figura 26</b> A_4.....	61
<b>Figura 27</b> A_3.....	61
<b>Figura 28</b> A_1.....	62
<b>Figura 29</b> A_5.....	62
<b>Figura 30</b> C_2.....	63
<b>Figura 31</b> C_1.....	63
<b>Figura 32</b> C_4.....	64
<b>Figura 33</b> C_3.....	64
<b>Figura 34</b> C_5.....	64
<b>Figura 35</b> A2_4M.....	66
<b>Figura 36</b> A1_4M.....	66
<b>Figura 37</b> A6_6M.....	66
<b>Figura 38</b> A4_2M.....	66
<b>Figura 39</b> A4_2M.....	66

<b>Figura 40</b> A3_4M .....	66
<b>Figura 41</b> A8_6M .....	67
<b>Figura 42</b> A7_6M .....	67
<b>Figura 43</b> A10_2M .....	67
<b>Figura 44</b> A9_2M .....	67
<b>Figura 45</b> C5_4M .....	71
<b>Figura 46</b> C6_4M .....	71
<b>Figura 47</b> C2_2M .....	71
<b>Figura 48</b> C1_2M .....	71
<b>Figura 49</b> C4_2M .....	71
<b>Figura 50</b> C3_2M .....	71
<b>Figura 51</b> C12_6M .....	72
<b>Figura 52</b> C11_6M .....	72
<b>Figura 53</b> C8_4M .....	72
<b>Figura 54</b> C10_6M .....	72
<b>Figura 55</b> C9_4M .....	72
<b>Figura 56</b> C7_4M .....	72
<b>Figura 57</b> C14_6M .....	73
<b>Figura 58</b> C13_6M .....	73
<b>Figura 59</b> A2_ROBUSTES .....	76
<b>Figura 60</b> A1_ROBUSTES .....	76
<b>Figura 61</b> A4_ROBUSTES .....	76
<b>Figura 62</b> A3_ROBUSTES .....	76
<b>Figura 63</b> A5_ROBUSTES .....	77
<b>Figura 64</b> C1_ROBUSTES.....	79
<b>Figura 65</b> C2_ROBUSTES.....	79
<b>Figura 66</b> C5_ROBUSTES.....	79
<b>Figura 67</b> C4_ROBUSTES.....	79
<b>Figura 68</b> C3_ROBUSTES.....	79
<b>Figura 69</b> Tabla De Evaluación-Arma De Fuego.....	93
<b>Figura 70</b> Tabla cruzada Distancia*Verdaderos positivos .....	93
<b>Figura 71</b> Gráfico De Barras .....	94
<b>Figura 72</b> Tabla cruzada Cantidad de imágenes*Verdaderos positivos.....	95

<b>Figura 73</b> Gráfico De Barras .....	95
<b>Figura 74</b> Verdaderos Positivos.....	96
<b>Figura 75</b> Falsos Positivos.....	97
<b>Figura 76</b> Falsos Negativo.....	98
<b>Figura 77</b> Tabla De Evaluación Arma Blanca.....	103
<b>Figura 78</b> Tabla cruzada Distancia*Verdaderos positivos .....	103
<b>Figura 79</b> Gráfico De Barras .....	103
<b>Figura 80</b> Tabla cruzada Cantidad de imágenes*Verdaderos positivos.....	104
<b>Figura 81</b> Gráfica de Barras .....	105
<b>Figura 82</b> Verdaderos Positivos.....	106
<b>Figura 83</b> Falso Positivos .....	107
<b>Figura 84</b> Falsos Negativos .....	108
<b>Figura 85</b> Diseño y codificación de la aplicación CentinelaDefender .....	114
<b>Figura 86</b> Codificación en Python diseñado para la detección de armas .....	114
<b>Figura 87</b> Configuración de la base de datos Firebase .....	115
<b>Figura 88</b> Construcción de la base de datos .....	115
<b>Figura 89</b> Elementos utilizados para el desarrollo del sistema de seguridad: cámara, computador y trípode.....	116
<b>Figura 90</b> elementos utilizados para la detección.....	116
<b>Figura 91</b> Farmacia_2.....	117
<b>Figura 92</b> Farmacia_1 .....	117
<b>Figura 93</b> Farmacia_4.....	117
<b>Figura 94</b> Farmacia_3.....	117
<b>Figura 95</b> Farmacia_6.....	117
<b>Figura 96</b> Farmacia_5.....	117

## RESUMEN

El presente trabajo describe el diseño e implementación de un sistema inteligente basado en visión artificial para el monitoreo y prevención de robos en tiempo real en la farmacia PharmaTodo de la ciudad de Riobamba. La investigación se estructuró en varias fases: investigación y análisis de algoritmos de detección de objetos y gestos corporales, implementación de los algoritmos seleccionados, desarrollo de la aplicación para la interfaz de usuario, diseño de un sistema de comunicación para alertas de robo, y evaluación del rendimiento mediante pruebas prácticas en condiciones reales.

El sistema desarrollado utiliza principalmente el algoritmo YOLO v5 para la detección de armas de fuego cortas y armas blancas (cuchillos), complementado con el algoritmo de detección de postura de Mediapipe para identificar gestos corporales relevantes. Se diseñó y desarrolló una aplicación de interfaz intuitiva para los usuarios finales, facilitando la visualización de alertas y la interacción con el sistema de seguridad.

Se implementó un sistema de comunicación seguro basado en Firestore de Google Cloud para la transmisión eficiente de alertas en tiempo real a los propietarios de la farmacia. Los resultados obtenidos demostraron que el sistema es capaz de detectar con alta precisión tanto armas de fuego cortas como armas blancas, mejorando significativamente la seguridad y eficacia en la prevención de robos en la farmacia PharmaTodo. Este estudio subraya la importancia de la visión artificial como herramienta avanzada para la seguridad, proporcionando una respuesta rápida y efectiva ante situaciones de riesgo.

**Palabras clave:** Visión artificial, Detección de objetos, Algoritmo YOLO v5, Mediapipe pose, Firestore.



## ABSTRAC

The current work describes the design and implementation of an intelligent system based on an artificial vision for real-time monitoring and prevention of thefts at the PharmaTodo pharmacy in Riobamba. It was vital to develop the research in several phases: investigation and analysis of object and body gesture detection algorithms, implementation of the selected algorithms, development of the application for the user interface, design of a communication system for theft alerts, and performance evaluation through practical tests under natural conditions. The developed system mainly uses the YOLO v5 algorithm to detect handguns and knives, complemented by the Mediapipe pose detection algorithm to identify relevant body gestures. An intuitive user interface application was designed and developed for end-users, facilitating the visualization of alerts and interaction with the security system. Implementing a secure communication system based on Google Cloud's Firestore was essential to transmit real-time alerts to the pharmacy owners efficiently. The results demonstrated that the system can accurately detect handguns and knives, significantly improving security and efficiency in theft prevention at PharmaTodo pharmacy. This study highlights the importance of artificial vision as an advanced security tool, providing a quick and effective response to risky situations.

**Keywords:** Artificial vision, Object detection, YOLO v5 algorithm, Mediapipe pose, Firestore.



**Reviewed by:**

Mgs. Jessica María Guaranga Lema

**ENGLISH PROFESSOR**

C.C. 0606012607

## **CAPÍTULO I. INTRODUCCIÓN**

### **1.1 PLANTEAMIENTO DEL PROBLEMA**

América Latina es la segunda región más violenta del mundo, y sus ciudades muestran signos de aumentos dramáticos en el crimen diario. Además de estos evidentes impactos socioculturales, la violencia y la delincuencia también provocan graves pérdidas económicas [4].

Hoy en día resulta insuficiente los diferentes elementos tecnológicos que forman una red de seguridad como, por ejemplo: alarmas, cámaras de video vigilancia, vallas eléctricas o la utilización de sensores como métodos de protección y seguridad para locales comerciales u hogares [5].

Actualmente la sociedad busca sistemas que permitan prevenir e identificar los hechos delictivos apenas se han cometido, este tipo de tecnología debe ser dinámica y escalable (permitir incorpora más cantidad de equipos al sistema como por ejemplo incluir más cámaras), de cierta manera estar un paso delante de los acontecimientos delictivos.

Estos nuevos sistemas de seguridad deben adaptarse rápidamente, recibiendo y emitiendo datos, de esta manera brindar respuestas rápidas y tomar las precauciones debidas en tiempo real. La utilización de la visión artificial en los sistemas de seguridad nos permite responder esta problemática, como se mencionó anteriormente la visión artificial está en la capacidad de procesar datos y en base a este proceso tomar decisiones.

### **1.2 JUSTIFICACIÓN**

Hoy en día la delincuencia e inseguridad en el país se ha vuelto el principal problema para los ecuatorianos, provocando pérdidas económicas y afectando de manera negativa a quien sufre de este tipo de acontecimiento, la presente investigación se enfoca en el diseño e implementación de un sistema inteligente basado en visión artificial, técnicas de análisis de lenguaje corporal y análisis de imágenes para el monitoreo, prevención y alerta de robos. Esto con la finalidad de disuadir un posible robo y reducir sus

consecuencias, de esta manera se pretende aportar de manera positiva a la ciudadanía ofreciendo un sistema de seguridad inteligente enfocado a locales comerciales y hogares.

### **1.3 OBJETIVOS**

#### **1.3.1 GENERAL**

- Diseñar e implementar un sistema inteligente basado en visión artificial para el monitoreo y prevención de robos en tiempo real en la farmacia PharmaTodo de la ciudad de Riobamba.

#### **1.3.2 ESPECÍFICOS**

- Investigar y analizar algoritmos de detección de objetos(armas) y gestos corporales mediante visión artificial.
- Implementar algoritmos que permitan reconocer armas de fuego cortas, armas blancas y gestos corporales de personas en tiempo real, para mejorar la precisión y eficacia de la detección de robos.
- Diseñar e implementar un sistema de comunicación confiable y seguro que permita la transmisión de alertas de robo a las partes interesadas como los propietarios de la farmacia PharmaTodo, utilizando tecnologías de telecomunicaciones adecuadas.
- Evaluar el rendimiento del sistema desarrollado mediante pruebas prácticas en la farmacia PharmaTodo, comparando su precisión y sensibilidad, detección de objetos(armas).

## **CAPITULO II. MARCO CONCEPTUAL**

### **2. ESTADO DEL ARTE**

El concepto de visión artificial se concibió en la década de los años sesenta, y desde entonces ha experimentado una evolución significativa, ganando gran importancia hasta la actualidad. Se ha convertido en uno de los métodos más relevantes para la identificación y procesamiento de imágenes. En el área de seguridad y monitoreo, la capacidad de analizar y procesar imágenes de manera inteligente es de suma relevancia, ya que este proceso facilita la toma de decisiones.

El objetivo del siguiente texto es exponer los avances de la visión artificial en el área de monitoreo y destacar su contribución en diversos campos.

Para comenzar, es importante comprender el funcionamiento básico de la visión artificial, que se asemeja al proceso visual humano. La visión artificial toma la imagen captada por una cámara y la procesa mediante métodos o algoritmos específicos. Posteriormente, se realiza la toma de decisiones basada en este procesamiento.

La visión artificial enfocada en el monitoreo y la prevención ha sido objeto de investigación en los últimos años. En trabajos previos, diversos investigadores han explorado una amplia gama de aplicaciones y técnicas con el fin de mejorar la eficacia y precisión de estos sistemas.

Por ejemplo, S. Draghici llevo a cabo un estudio sobre un sistema de visión artificial basado en redes neuronales capaz de analizar la imagen de un automóvil proporcionada por una cámara, localizar la placa de matrícula y reconocer el número de registro del automóvil obteniendo como resultados reconocimiento de caracteres exitoso aproximadamente al 98% y reconocimiento exitoso de placas de registro completas aproximadamente al 80% [10].

De igual manera, J. A. Ludeña Chica desarrollo un sistema de seguridad para supervisión de niños entre 2 a 4 años usando Visión Artificial, sus resultados exponen en base a las pruebas realizadas que el sistema de seguridad propuesto es 15% más factible que los sistemas convencionales al implementarlo en los hogares para el cuidado de los niños de temprana

edad [11], de esta manera se evidencia que los sistemas de visión artificial aportan en gran medida al monitoreo y prevención.

Otro caso de estudio es del de G. D. Brito Moncayo quien desarrollo un Sistema de visión artificial con monitoreo web para conteo de pasajeros en buses de transporte público, el algoritmo BackgroundSubtractorMOG2 permite adaptación automática de la escena frente a perturbaciones de luz así como detecciones de sombras, junto a una interpretación más precisa de fondos dinámicos [12].

De la misma forma J. C. Broncano Torres desarrollo un sistema de seguridad y acceso vehicular, mediante visión artificial. Los resultados determinaron que la aplicación creada optimiza eficientemente el sistema, pues disminuye aproximadamente en un 87% el tiempo de congestión vehicular en el lugar de despliegue del sistema, para determinar la eficiencia del sistema se realizaron pruebas de efectividad de software, en las cuales se utilizaron 400 imágenes vehiculares, obteniendo como resultado un 78% de efectividad. [13].

En resumen, la visión artificial ha demostrado ser una herramienta poderosa en el ámbito del monitoreo y la seguridad, permitiendo la detección, análisis y toma de decisiones inteligentes basadas en imágenes capturadas. A lo largo de este texto, hemos explorado cómo esta tecnología ha evolucionado desde sus inicios en la década de 1960 hasta convertirse en un componente fundamental en una amplia gama de aplicaciones, desde la supervisión de niños hasta el control de acceso vehicular.

Sin embargo, a medida que avanzamos hacia el futuro, queda claro que aún hay mucho por descubrir y desarrollar en el campo de la visión artificial y el monitoreo. Las futuras investigaciones podrían centrarse en la mejora de la precisión y eficiencia de los algoritmos de procesamiento de imágenes, la integración de tecnologías emergentes como el aprendizaje profundo y la inteligencia artificial, y la exploración de nuevos enfoques para la detección y análisis de eventos en tiempo real.

## **2.1 MARCO CONCEPTUAL**

### **2.1.1 INSEGURIDAD EN ECUADOR**

La preocupación principal de la sociedad ecuatoriana en estos tiempos se centra en la inseguridad, un fenómeno que afecta directamente la tranquilidad de los ciudadanos. En los últimos años, diversas formas de delincuencia, como los asaltos, robos y fraudes, han aumentado, generando un clima de inseguridad generalizado. Este problema es alimentado por la actividad de asaltantes, bandas urbanas y grupos del crimen organizado, lo que representa un desafío significativo para la seguridad pública en el país.

Una de las consecuencias del incremento de la delincuencia son los robos a locales comerciales, que en los últimos años se han convertido en uno de los principales focos de preocupación. Afectando directamente a los comerciantes y empresarios, generando pérdidas económicas, trauma e impacto psicológico a quienes sufren el robo y un clima de inseguridad en el ámbito comercial.

En Ecuador los robos a locales comerciales y empresas se incrementaron un 12% el primer semestre del 2023, según la fiscalía general del estado. Entre enero y junio en Ecuador se registraron 2.920 robos a empresas y a locales comerciales, el pico más alto fue en marzo con 584 robos.[13]

De acuerdo con un estudio realizado por la fiscalía general del estado, se señala que la mayoría de los robos ocurren durante la mañana, destacando que los lunes son los días con mayor incidencia.[13] En el registro estadístico de empresas en Guayas, la provincia más poblada de Ecuador, están registradas más de 161.000 unidades económicas, lo que representa el 20% del total de empresas en el país. Guayas lidera en robos a negocios con aproximadamente 976 casos registrados. Le sigue Pichincha, donde están inscritas 216.000 negocios y empresas, con 642 casos de robo. Aunque los robos en el resto de provincias son evidentes, no alcanzan los niveles de Guayas y Pichincha. Por ejemplo, en El Oro se registraron 259 casos de robos y en Manabí, 204 casos, según datos obtenidos de la Fiscalía.[13]

Ante estos sucesos delictivos las empresas y pequeños negocios han empezado a invertir mas en seguridad, un ejemplo claro de este suceso es de una empresa del norte de Pichincha, que reforzo sus sistemas de seguridad, en 2022 la empresa tenía 5 cámaras de seguridad, en 2023 esa cifra aumento a 11 cámaras, además de instalarse un cerco eléctrico.[13]

Estas inversiones en seguridad por parte de las empresas son una clara manifestación de las consecuencias palpables de la delincuencia en la sociedad. El aumento de la criminalidad y los robos a locales comerciales ha obligado a las empresas y pequeños negocios a tomar medidas proactivas para proteger sus activos y garantizar la seguridad de su personal y clientes. Es un ejemplo vívido de cómo la inseguridad genera un impacto directo en la vida cotidiana y las actividades comerciales, llevando a la necesidad de adaptarse y fortalecer las medidas de seguridad.

### **2.1.2 IMPACTO PSICOLÓGICO EN EMPLEADOS**

De igual manera las consecuencias psicológicas pueden ser peores que a ver sufrido una agresión. La persona afectada puede quedar con secuelas como por ejemplo miedo, ansiedad y estrés postraumático. Un robo es un evento estresante que puede tener consecuencias psicológicas a largo plazo. La mayoría de las víctimas pueden sufrir una variedad de síntomas como por ejemplo depresión, miedo y culpa. También pueden estar estresados e irritables y poseer dificultad para dormir o concentrarse. [8]

El trauma psicológico puede ser muy difícil de superar y pueden pasar meses o incluso años antes de que la persona se sienta completamente segura y cómoda. Si bien es posible superar el trauma, se debe tener en cuenta que el proceso debe llevar tiempo y requerir ayuda profesional. [8]

A continuación, se menciona las secuelas más comunes:

- Trastorno Del Sueño
- Ansiedad Y Estrés
- Irritabilidad
- Depresión

- Miedo
- Cambios en el estilo de vida

### **2.1.3 IMPACTO ECONÓMICO DE LA INSEGURIDAD**

La inseguridad tiene repercusiones económicas significativas, ya que provoca la paralización de negocios, la interrupción de la producción y el comercio, y conlleva a una disminución en el empleo y los ingresos para los comerciantes y empresas. Sectores clave como el turismo, la alimentación, el hospedaje y el comercio sufren directamente las consecuencias de la delincuencia, lo que inevitablemente se traduce en una reducción en el consumo y menores ingresos económicos.[17]

Según el informe Top Risks 2024, Ecuador terminó el año 2023 como uno de los países más afectados por la inseguridad, ya desde junio de 2023 se evidenció una caída de ventas en locales comerciales entre -0.9% y -1.5% mensual durante el segundo trimestre del año. Actualmente en 2024 el comercio y el turismo proveen un descenso de ventas de hasta un 20% esto debido a los hechos delictivos se sufrió el país a inicios del año, como consecuencia se obtiene menos ingreso de dinero a los negocios y menos dinero al bolsillo de las personas.[18]

Un ejemplo evidente se puede observar en los resultados de una encuesta realizada a familias en Quito que gestionan pequeños negocios, quienes informaron que el consumo había disminuido. En cuanto a las empresas, incluso antes del inicio de la violencia en 2024, se percibía una perspectiva de una caída significativa.

### **2.1.4 MÉTODOS DE PREVENCIÓN Y MITIGACIÓN DE ROBOS**

Como se habló anteriormente, varias empresas y pequeños negocios optaron por invertir en métodos de seguridad como por ejemplo los sistemas de seguridad, los cuales permitirán el monitoreo y prevención de actos delictivos. Un sistema de seguridad se define como un conjunto de elementos y dispositivos interconectados en función del resguardo y protección de un espacio físico, hogar o negocio.[9]



Los sistemas de seguridad ofrecen capacidad de control y resistencia ante situaciones de riesgo y peligro. De ellos depende no solo evitar la intrusión o disuadirla, sino que además permita registrar el evento y activar un protocolo de seguridad ante este acontecimiento.

Entre los sistemas de seguridad más utilizados tenemos:

- **Sistemas de alarmas:** este tipo de sistema se caracteriza por emitir una alerta temprana, permitiendo una acción rápida de protección y disuasión. el sistema de alarmas esta constituido por interconexiones de equipos en lugares estratégicos de la zona a proteger. [9]

Los equipos que constituye este sistema de seguridad son contactos magnéticos, sensores de movimiento, detectores de humo, botón de pánico, etc. Su funcionamiento de basa en que cada uno de los equipos emite una señal cuando una zona o área protegida es vulnerada. [9]

- **Sistema de cámaras de seguridad:** este tipo de sistemas permite un análisis visual de la zona a proteger, de esta manera se determina la evaluación y documentación de elementos de peligro, daño o delitos de una zona bajo protección. [9]
- **Sistemas de control de acceso:** este tipo de sistema se caracteriza por permitir el ingreso y salida de personas en la zona bajo protección. [9]
- **Sistemas de prevención de incendios y humedad:** este tipo de sistemas son aquellos que intervine de manera automática, sin la intervención humana, funcionan a través de sensores los cuales miden una variable física y activan medida para su control o extinción. [9]
- **Cercos eléctricos:** este tipo de sistema de seguridad hace uso de una valla electrificada utilizada para la protección de la zona que se desea proteger. este sistema actúa frente a un posible intruso que desee ingresar a la zona protegida, causándole una sacudida ocasionada por la descarga eléctrica. [9]

## 2.2 VISIÓN ARTIFICIAL

La visión artificial es un área de estudio y aplicación dedicada al desarrollo de sistemas informáticos capaces de procesar, analizar y comprender imágenes o videos de manera análoga a la percepción humana. [19]

Su principal objetivo es dotar a las máquinas de la capacidad de observar y comprender su entorno visual. Esto implica la creación de algoritmos y técnicas para detectar y reconocer objetos, personas, formas, colores y otras características visuales en imágenes o videos. Estas funciones se llevan a cabo mediante el uso de cámaras o sensores que capturan imágenes, las cuales son luego procesadas y analizadas mediante algoritmos y modelos de aprendizaje automático.[19]

La visión artificial tiene diversas aplicaciones en numerosas industrias y campos, incluyendo la medicina, la robótica, la seguridad, la automoción, el control de calidad, la realidad aumentada, los videojuegos y la vigilancia, entre otros. Ejemplos de estas aplicaciones incluyen el reconocimiento facial, la detección de objetos en imágenes médicas, la clasificación de productos en líneas de producción y la navegación autónoma de vehículos. [19]

### 2.2.1 ETAPAS DEL PROCESAMIENTO DE IMÁGENES CON VISIÓN ARTIFICIAL

Las etapas que involucra el procesamiento de imágenes son las siguientes. Es importante tener en cuenta que estas son generalidades, ya que el método o algoritmo utilizado puede variar y cambiar los pasos expuestos.

1. **La Captura:** El objetivo de este proceso es proporcionar una o varias imágenes digitales al sistema a través de componentes externos como por ejemplo la utilización de cámaras, video-cameras, escáner, etc. [19]
2. **Pre\_procesamiento:** el objetivo de este proceso es el de realzar ciertos detalles o características de la imagen, aplicar técnicas de reducción de ruido y mejoramiento de contraste. [19]
3. **Segmentación:** la imagen es dividida en regiones o grupos de pixeles que sean de cierto interés de estudio, esta etapa es crucial para el funcionamiento del sistema de detección. [19]

4. **Descripción:** en esta etapa el sistema determina características relevantes para diferenciar un tipo de objeto de otro. Características como la forma, patrones de textura, color entre otros. [19]
5. **Reconocimiento:** este proceso clasifica los objetos según los descriptores del paso anterior , los objetos detectados que presentan descriptores similares se agrupan o selecciona en una misma categoría. [19]

## 2.2.2 ALGORITMOS DE VISIÓN ARTIFICIAL

En términos generales, los algoritmos de visión artificial se definen como un conjunto de instrucciones y procesos que permiten a las máquinas interpretar y comprender imágenes o videos, y tomar decisiones en función de esa interpretación.

El algoritmo le permitir a la maquina aprenda por si misma, en lugar de que alguien la programe para reconocer una imagen.

Los métodos clásicos de visión artificial se enfocan en identificar y resaltar elementos distintivos en una imagen, como los bordes. Estos bordes son áreas donde hay cambios significativos en la intensidad o el color de la imagen, lo que los hace destacar como puntos clave para el análisis visual.[20]

Han surgido numerosos algoritmos a lo largo del tiempo que posibilitan la extracción de una variedad de características. Por consiguiente, la eficacia de estas técnicas se ve influenciada por la calidad de las características que se logran extraer.[20]

A continuación, se mencionan algoritmos que cumplen con la función de obtener características únicas:

1. **Detección de contornos:** identifica los limites y bordes de los objetos.[20]
2. **Extracción de características:** identifica y describe rasgos distintivos.[20]
3. **Segmentación de imágenes:** divide una imagen en regiones.[20]
4. **Reconocimiento de texto:** extrae y reconoce texto de una imagen.[20]
5. **Análisis de movimiento:** detecta y analiza el movimiento en imágenes o videos.[20]

Sin embargo, el avance en el campo de la visión artificial ha dado lugar a los algoritmos de aprendizaje profundo y las redes neuronales. Estos algoritmos automatizan el proceso de extracción de características de las imágenes.[20] Permiten clasificar imágenes en distintas categorías, como diferenciar entre fotos de gatos y perros, o identificar personas y automóviles en una imagen. Además de clasificar las imágenes, estos algoritmos pueden también localizar varios objetos dentro de una imagen. Para desarrollar modelos de detección y clasificación de imágenes, es necesario disponer de un conjunto de datos de imágenes etiquetadas.[20]

### **2.2.3 ALGORITMOS DE DETECCIÓN DE OBJETOS**

- **YOLOV5 (YOU ONLY LOOK ONCE VERSIÓN 5)**

YOLOv5 es una versión de la arquitectura YOLO (You Only Look Once), que es un algoritmo de detección de objetos en imágenes en tiempo real. YOLO divide la imagen en una cuadrícula y realiza una sola pasada de la red neuronal convolucional (CNN) sobre toda la imagen. Luego, predice simultáneamente múltiples cajas delimitadoras y las probabilidades de clasificación para estas cajas. Esto permite una detección de objetos rápida y eficiente.[21]

YOLOv5 es una implementación de YOLO desarrollada por Ultralytics. Utiliza una arquitectura de red neuronal convolucional profunda y se ha optimizado para mejorar la precisión y la eficiencia computacional en la detección de objetos en imágenes. YOLOv5 ha ganado popularidad por su rendimiento y facilidad de uso en una variedad de aplicaciones de visión por computadora.[21]

- **RED NEURONAL CONVOLUCIONAL (CNN)**

Una red neuronal convolucional (CNN) es un tipo de red neuronal artificial que ha demostrado un gran éxito en el procesamiento de imágenes y en la visión por computadora. Las CNN están compuestas por capas de convolución, que aplican filtros a la imagen de entrada para extraer características relevantes, seguidas de capas de agrupación que reducen la dimensionalidad. Estas características extraídas se alimentan luego a capas totalmente conectadas para la clasificación o la regresión.[22]

Las redes neuronales convolucionales imitan el funcionamiento del sistema visual biológico y son altamente efectivas en la tarea de reconocimiento de patrones en imágenes. Han sido utilizadas con éxito en una amplia gama de aplicaciones, incluyendo reconocimiento de objetos, segmentación semántica, detección de rostros, entre otras.[22]

- **HAAR CASCADE TRAINER**

Haar Cascade Trainer es una herramienta utilizada para entrenar clasificadores de objetos basados en características de Haar. Las características de Haar son características simples que se utilizan para detectar objetos en imágenes. El entrenamiento implica proporcionar al algoritmo una serie de imágenes positivas que contienen el objeto de interés y una serie de imágenes negativas que no lo contienen. El algoritmo utiliza estas imágenes para aprender las características distintivas del objeto y luego puede ser utilizado para detectar ese objeto en nuevas imágenes.[23]

Haar Cascade Trainer es comúnmente utilizado en la detección de objetos en imágenes estáticas, como rostros, ojos, vehículos, entre otros. Utiliza un enfoque basado en características de Haar y puede ser eficaz para objetos que tienen características visuales distintivas y bien definidas. Sin embargo, puede no ser tan preciso o eficiente como los enfoques más avanzados basados en redes neuronales convolucionales, especialmente en situaciones donde los objetos pueden variar en apariencia o escala.[23]

#### **2.2.4 ALGORITMOS DE DETECCIÓN Y SEGUIMIENTO DE POSTURA HUMANA**

- **POSE DETECTION**

El algoritmo de detección de postura (pose detection) proporcionado por la biblioteca Mediapipe es una herramienta que permite detectar y estimar la postura humana en imágenes o videos en tiempo real.[24]

El algoritmo de detección de postura de Mediapipe utiliza una red neuronal convolucional (CNN) pre-entrenada para identificar y localizar puntos clave en el cuerpo humano, como las articulaciones y los extremos de los miembros. Estos puntos clave suelen incluir las muñecas, los codos, los hombros, las caderas, las rodillas y los tobillos, entre otros. Una vez

detectados, estos puntos clave se utilizan para estimar la posición y orientación de las diferentes partes del cuerpo, lo que permite reconstruir la postura global del individuo en la imagen o el video.[24]

## **CAPITULO III. METODOLOGIA**

### **3. TIPO DE INVESTIGACIÓN**

Para el desarrollo del siguiente proyecto de investigación es necesario estudiar y evaluar diferentes métodos de identificación de objetos y movimientos corporales mediante el uso de visión artificial. Para determinar el funcionamiento del sistema de forma estadística se realiza la comparación de los registros exitosos de detección de objetos(armas) y movimientos corporales en comparación a los datos ya registrados en el sistema.

#### **3.1 INVESTIGACIÓN EXPERIMENTAL**

El buen funcionamiento del sistema se basa en cuantas veces se realizó un registro exitoso de objetos (armas) y los movimientos corporales previamente ya registrados en el sistema. Este método implica la realización de experimentos cuidadosamente diseñados y controlados, en los que se manipulan variables independientes y se miden las respuestas o resultados correspondientes.

#### **3.2 INVESTIGACIÓN DESCRIPTIVA**

El método descriptivo se centra en la recopilación de datos a través de diferentes técnicas, en relación al tema de investigación se pretende utilizar técnicas de procesamiento de imágenes para lograr identificar objetos(armas) y movimientos corporales.

#### **3.3 POBLACIÓN Y TAMAÑO DEMUESTRA**

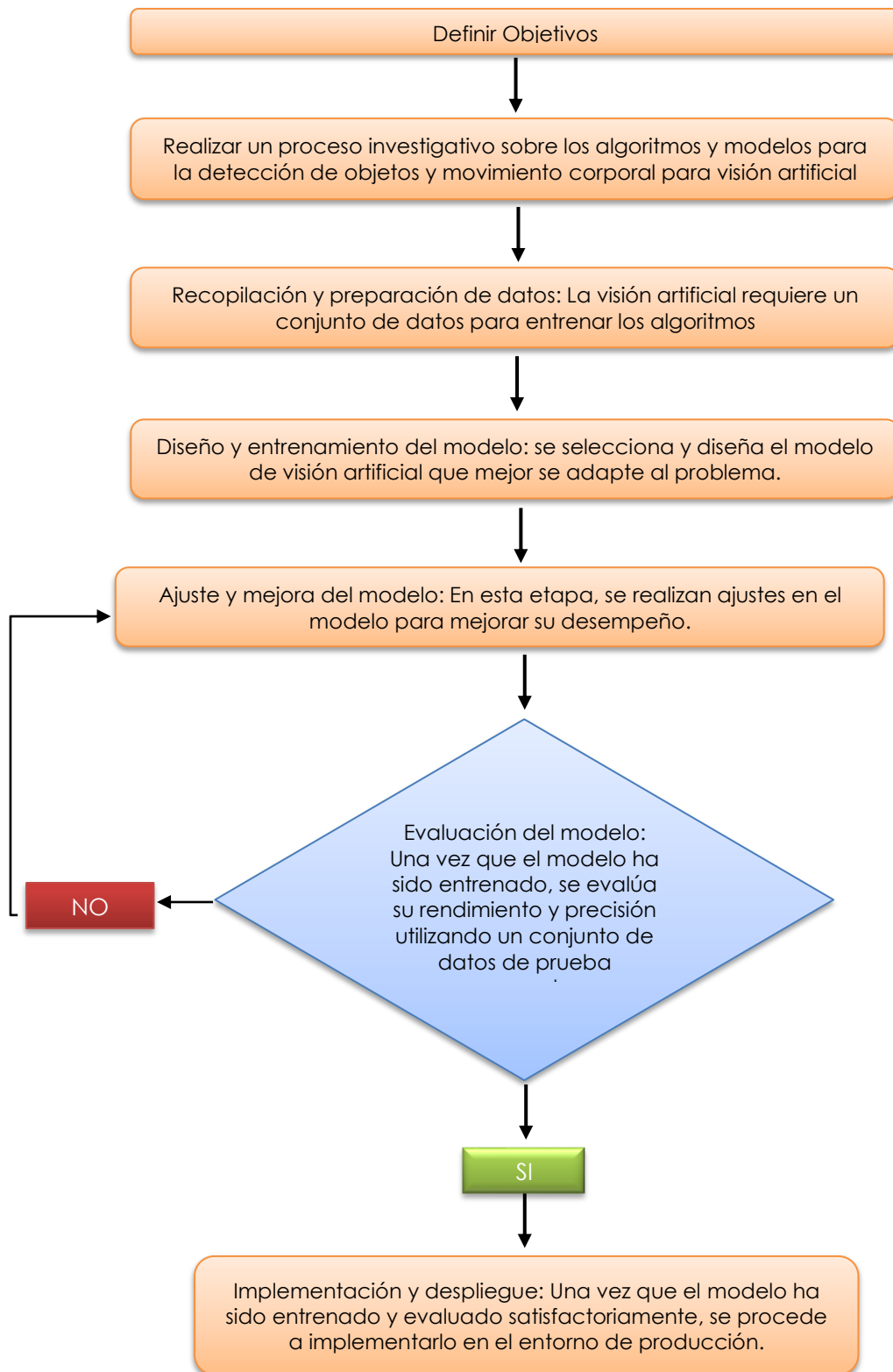
##### **3.3.1 POBLACIÓN**

- La población se define por todos los datos recogidos en las pruebas realizadas, donde las variables independientes variarían en cada prueba realizada.

##### **3.3.2 MUESTRA**

- Para determinar la muestra se seleccionará de manera aleatoria los datos obtenidos del experimento, los cuales serán analizados para decidir si estos influyen en el correcto funcionamiento del sistema de seguridad.

### 3.4 DISEÑO DE LA INVESTIGACIÓN



**Figura 1** *Diseño de la investigación*

### 3.5 OPERACIÓN DE LAS VARIABLES

#### 3.5.1 VARIABLES INDEPENDIENTES

Variable	Descripción	Unidad de medida
Distancia del objeto a la cámara	La distancia existe entre la cámara y el objeto puede afectar la capacidad de detección y reconocimiento de objetos.	<ul style="list-style-type: none"><li>• Metros</li></ul>
Cantidad de imágenes en la base de datos	La calidad, cantidad y diversidad de los datos utilizados en el entrenamiento pueden influir en la capacidad del modelo para generalizar y realizar predicciones precisas sobre nuevas imágenes. Un conjunto de datos insuficiente o sesgado puede afectar negativamente el rendimiento del sistema.	<ul style="list-style-type: none"><li>• Ninguno</li></ul>

**Tabla 1** Variables Independientes

#### 3.5.2 VARIABLES DEPENDIENTES

Variable	Descripción	Unidad de media
Precisión	Capacidad del sistema para realizar una detección adecuada de objetos (armas) y movimientos corporales.	<ul style="list-style-type: none"><li>• Porcentaje (%)</li></ul>
Sensibilidad	Evaluación de la capacidad del sistema para identificar correctamente los casos positivos, es decir, determinar si se detecta un robo basado en el reconocimiento de objetos y movimientos corporales.	<ul style="list-style-type: none"><li>• Porcentaje (%)</li></ul>

**Tabla 2** Variables Dependientes



### 3.6 DIAGRAMA MULTICAPA DEL PROCESO DE DESARROLLO DEL SISTEMA DE SEGURIDAD

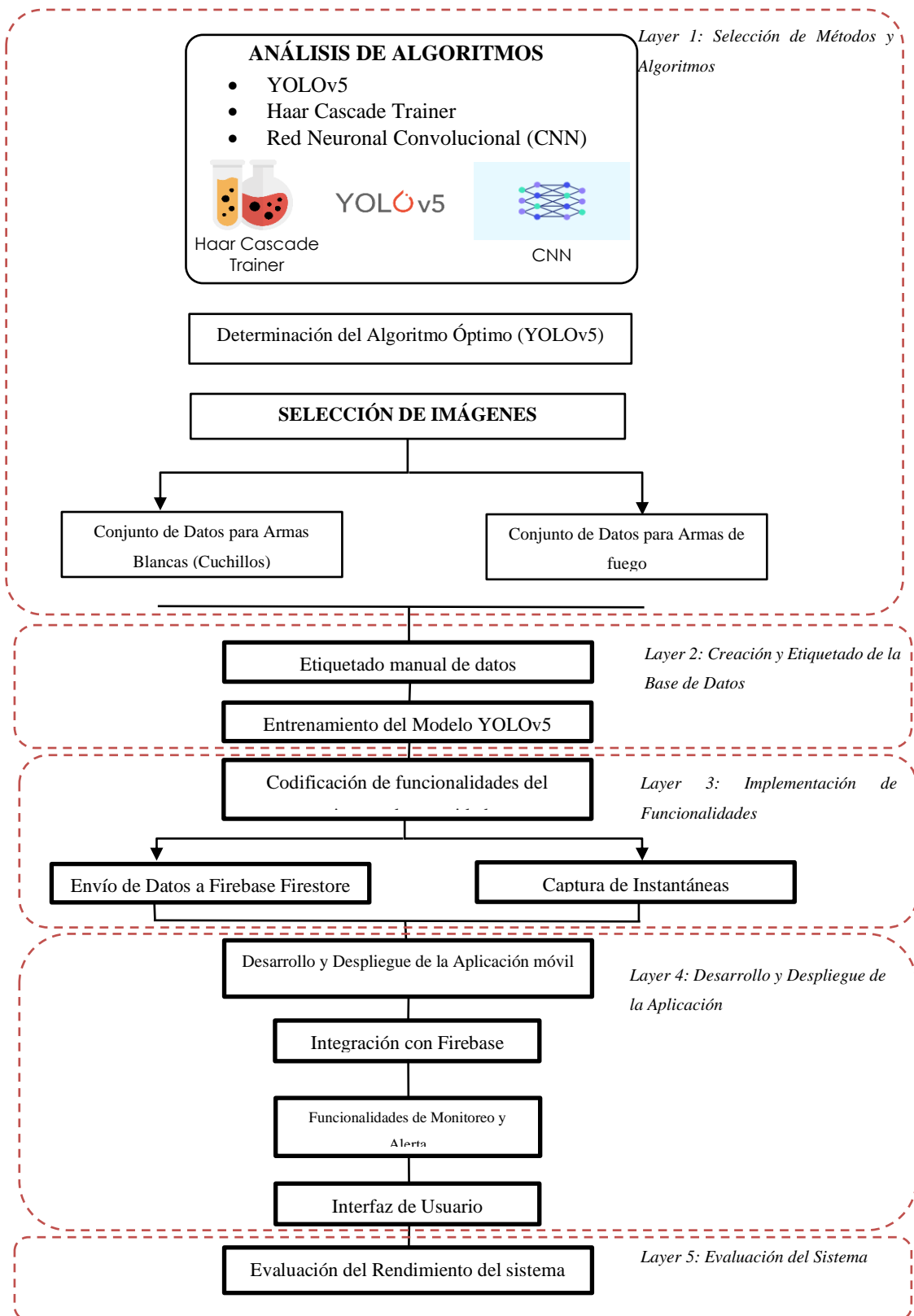


Figura 2 Diagrama multicapa

### **3.6.1 FUNCIONAMIENTO DEL SISTEMA DE SEGURIDAD**

El sistema de seguridad se estructura en varias fases esenciales y consecutivas. En la Fase 1, se realiza la selección de métodos y algoritmos adecuados. En esta etapa inicial, se analizaron dos algoritmos: YOLOv5 y Haar Cascade Trainer, así como un método basado en una red neuronal convolucional (CNN). A partir de este análisis, se determina cuál es el más óptimo para el proyecto. Tras una evaluación exhaustiva, se opta por implementar YOLOv5 debido a su eficacia en la detección de objetos. Posteriormente, se lleva a cabo una rigurosa selección de imágenes para crear una base de datos, un paso crucial para el entrenamiento del algoritmo seleccionado. Se generaron dos conjuntos de datos específicos: uno enfocado en la detección de armas de fuego cortas y otro destinado a identificar armas blancas, como cuchillos.

En la Fase 2, se crea y etiqueta la base de datos, procediendo luego con el minucioso proceso de etiquetado. Esto se complementa en la Fase 3, donde se entrena YOLOv5 con los datos etiquetados para optimizar su precisión y eficacia en la detección de objetos.

La Fase 4 se centra en el desarrollo de la codificación en Python para la implementación de funcionalidades. La primera función tiene como objetivo principal enviar datos relevantes sobre los objetos detectados, tales como su ubicación geográfica, fecha y hora de detección, así como la naturaleza del objeto identificado, a la base de datos Firebase Firestore. La segunda función se encarga de capturar una instantánea en el momento exacto en que se detecta un objeto de interés. Estas imágenes son almacenadas de forma segura en Firebase Storage, garantizando su disponibilidad cuando sea necesario.

En la Fase 5, se procede con el desarrollo y despliegue de la aplicación CentinelaDefender, asegurando su integración con la base de datos y los servicios de almacenamiento de Firebase. Finalmente, en la Fase 6, se evalúa el rendimiento y precisión del modelo para asegurar que cumple con los estándares requeridos.

### **3.6.2 FASE 1: SELECCIÓN DE MÉTODOS Y ALGORITMOS**

En primer lugar, es fundamental identificar el algoritmo o método más adecuado para nuestra necesidad de detectar objetos. Para determinar el más idóneo, se realizó una investigación y un análisis detallado de dos algoritmos: YOLO v5 y HAAR Cascade, así como de un método

basado en redes neuronales convolucionales (CNN). Esta investigación nos permitió examinar diferentes enfoques y evaluar su eficacia en función de nuestras necesidades específicas.

- **ANÁLISIS DEL ALGORITMO YOLO V5**

Es un popular algoritmo de detección de objetos en tiempo real que se ha destacado por su capacidad para detectar objetos de manera rápida y precisa, de manera general su funcionamiento se basa en dividir la imagen en una cuadrícula y predecir las ubicaciones y las clases de objeto en cada celda.[25]

VENTAJAS	DESVENTAJAS
<p><b><i>Rapidez En Tiempo Real</i></b></p> <p>YOLOv5 es conocido por su capacidad para realizar detección de objetos en tiempo real. Esto lo hace adecuado para aplicaciones en las que se requiere baja latencia, como sistemas de asistencia al conductor y vigilancia.</p>	<p><b><i>Precisión En Objetos Pequeños</i></b></p> <p>YOLOv5 a veces tiene dificultades para detectar objetos muy pequeños, ya que las celdas de la cuadrícula pueden no capturar detalles finos.</p>
<p><b><i>Detección de múltiples objetos</i></b></p> <p>YOLOv5 puede detectar varios objetos en una sola imagen, incluso si se superponen</p>	<p><b><i>Localización Precisa De Objetos</i></b></p> <p>Aunque YOLOv5 es eficiente en la detección de objetos, puede no ser tan preciso en la localización exacta de objetos dentro de los cuadros delimitadores</p>
<p><b><i>Procesamiento de imagen en una sola pasada</i></b></p> <p>YOLOv5 procesa la imagen de una sola vez, a diferencia de los métodos que requieren múltiples pasadas o regiones de interés</p>	<p><b><i>Sensibilidad al tamaño de la cuadrícula</i></b></p> <p>La resolución de la cuadrícula utilizada en YOLOv5 afecta la precisión de la detección. Si la cuadrícula es muy grande, los objetos pequeños pueden pasarse por alto; si es muy pequeña, los objetos grandes pueden no detectarse correctamente</p>
<p><b><i>Versatilidad</i></b></p> <p>YOLOv5 es aplicable a una amplia gama de aplicaciones, desde la detección de objetos en imágenes fijas hasta el seguimiento de objetos en vídeo en tiempo real.</p>	<p><b><i>Necesidad de grandes conjuntos de datos de entrenamiento</i></b></p> <p>Como cualquier algoritmo de aprendizaje profundo, YOLOv5 requiere conjuntos de datos de entrenamiento grandes y variados para funcionar bien en diversas aplicaciones.</p>

**Tabla 3.** Ventajas y desventajas de YOLO

- **ANÁLISIS DEL ALGORITMO HAAR CAASADE TRAINER**

Haar Cascade Trainer es una herramienta fundamental en el campo de la visión por computadora, especialmente en la detección de objetos en imágenes. Su enfoque se basa en la creación de clasificadores de características Haar Cascade, los cuales son esenciales para identificar patrones específicos en las imágenes. Aunque este enfoque es menos complejo que los algoritmos de aprendizaje profundo como YOLO, tiene sus propias ventajas y desventajas.

**VENTAJAS Y DESVENTAJAS DEL ALGORITMO HAAR CAASADE TRAINER**

VENTAJAS	DESVENTAJAS
<p><b><i>Rapidez En Tiempo Real</i></b></p> <p>Haar Cascade Trainer es conocido por su rapidez en el procesamiento de imágenes, lo que lo hace adecuado para aplicaciones en tiempo real donde se requiere baja latencia.</p>	<p><b><i>Rapidez En Tiempo Real</i></b></p> <p>Aunque es rápido, puede no ser tan eficiente como algunos modelos de aprendizaje profundo en términos de velocidad y precisión en la detección de objetos en tiempo real.</p>
<p><b><i>Detección de múltiples objetos</i></b></p> <p>Haar Cascade Trainer es capaz de detectar múltiples objetos en una sola imagen, lo que lo hace útil para aplicaciones donde se requiere la detección de varios objetos simultáneamente.</p>	<p><b><i>Detección de múltiples objetos</i></b></p> <p>Puede tener limitaciones en la detección de objetos superpuestos o muy cercanos entre sí, lo que podría afectar su precisión en la identificación de cada objeto individualmente.</p>
<p><b><i>Procesamiento de imagen en una sola pasada</i></b></p> <p>El Haar Cascade Trainer procesa la imagen en una sola pasada, lo que lo hace más eficiente en términos de recursos computacionales en comparación con los métodos que requieren múltiples pasadas o regiones de interés.</p>	<p><b><i>Procesamiento de imagen en una sola pasada</i></b></p> <p>Esta eficiencia puede verse comprometida en escenarios donde se requiere una mayor precisión en la detección de objetos, ya que puede perder detalles finos debido a su enfoque de una sola pasada.</p>
<p><b><i>Versatilidad</i></b></p> <p>El Haar Cascade Trainer es aplicable a una amplia gama de aplicaciones, desde la detección de objetos en imágenes estáticas hasta el seguimiento de objetos en tiempo real en vídeo.</p>	<p><b><i>Versatilidad</i></b></p> <p>Aunque es versátil, puede tener limitaciones en la detección de objetos en ciertos escenarios o bajo ciertas condiciones de iluminación, lo que podría afectar su rendimiento en aplicaciones específicas.</p>

**Tabla 4.** Ventajas y desventajas de Haar Cascade Trainer

- **ANÁLISIS DEL METODO DE REDES NEURONALES CONVOLUCIONALES (CNN)**

Una red neuronal convolucional (CNN) es una arquitectura de red neuronal diseñada para procesar datos estructurados, como imágenes. Utiliza filtros convolucionales para extraer

características importantes de los datos y las combina en capas sucesivas para realizar tareas específicas, como la detección de objetos o el reconocimiento de patrones en imágenes. Aunque menos complejas que otros modelos de aprendizaje profundo, las CNN ofrecen alta precisión y versatilidad en aplicaciones de visión por computadora. Sin embargo, el entrenamiento y ajuste de parámetros pueden ser costosos en términos de datos y recursos computacionales.[28]

### VENTAJAS Y DESVENTAJAS DEL MÉTODO DE REDES NEURONALES CONVOLUCIONALES (CNN)

VENTAJAS	DESVENTAJAS
<p><b><i>Rapidez En Tiempo Real</i></b></p> <p>Las CNN pueden ser altamente eficientes en términos de velocidad y pueden procesar imágenes en tiempo real, lo que las hace adecuadas para aplicaciones que requieren baja latencia.</p>	<p><b><i>Rapidez En Tiempo Real</i></b></p> <p>En comparación con métodos más simples como el Haar Cascade, algunas CNN pueden ser más complejas y computacionalmente intensivas, lo que puede afectar su velocidad en algunos casos.</p>
<p><b><i>Detección de múltiples objetos</i></b></p> <p>Las CNN pueden detectar múltiples objetos en una sola imagen y son capaces de identificar objetos incluso en situaciones donde hay superposición o proximidad entre ellos.</p>	<p><b><i>Detección de múltiples objetos</i></b></p> <p>La detección de múltiples objetos puede ser más desafiante en términos de recursos computacionales y tiempo de procesamiento, especialmente en imágenes con una gran cantidad de objetos o en escenas complejas.</p>
<p><b><i>Procesamiento de imagen en una sola pasada</i></b></p> <p>Al igual que el Haar Cascade, algunas arquitecturas de CNN, como YOLO (You Only Look Once), procesan la imagen en una sola pasada, lo que puede mejorar la eficiencia en términos de recursos computacionales. .</p>	<p><b><i>Procesamiento de imagen en una sola pasada</i></b></p> <p>Aunque el procesamiento en una sola pasada puede ser eficiente, algunas arquitecturas de CNN pueden perder detalles finos en la imagen debido a su enfoque de paso único, lo que podría afectar la precisión en la detección de objetos.</p>
<p><b><i>Versatilidad</i></b></p> <p>Las CNN son altamente versátiles y aplicables a una amplia gama de aplicaciones, desde la detección de objetos en imágenes estáticas hasta el reconocimiento de objetos en vídeo en tiempo real.</p>	<p><b><i>Versatilidad</i></b></p> <p>Aunque son versátiles, algunas CNN pueden requerir conjuntos de datos de entrenamiento más grandes y variados para lograr un rendimiento óptimo en diferentes aplicaciones, lo que puede aumentar los requisitos de recursos y tiempo de entrenamiento</p>

**Tabla 5.** Ventajas y desventajas red neuronal convolucional (CNN)

### **3.6.2.1 JUSTIFICACIÓN TÉCNICA PARA LA SELECCIÓN DE YOLOV5 PARA EL DESARROLLO DEL SISTEMA DE SEGURIDAD**

En mi análisis comparativo de tres algoritmos de detección de objetos los cuales son YOLOv5, una Red Neuronal Convolutiva (CNN) y Haar Casca, opté por YOLOv5 debido a sus claras ventajas técnicas.

Primero, YOLOv5 destaca por su alta velocidad y eficiencia computacional. A diferencia de los algoritmos tradicionales como las CNN y Haar Cascade, que pueden ser computacionalmente intensivos y lentos, YOLOv5 utiliza una arquitectura de detección de objetos en una sola pasada. Esta arquitectura permite la identificación y clasificación de objetos en tiempo real, reduciendo significativamente el tiempo de inferencia al procesar la información de manera directa y sin necesidad de múltiples etapas de análisis.

Desde una perspectiva de flexibilidad y facilidad de entrenamiento, YOLOv5 también sobresale. Su arquitectura permite ajustar hiperparámetros y configurar modelos personalizados con relativa facilidad. Admite la transferencia de aprendizaje, lo que facilita la adaptación del modelo a nuevas clases de objetos con una cantidad reducida de datos etiquetados y menor tiempo de entrenamiento. En comparación, entrenar una CNN desde cero puede ser intensivo en recursos y tiempo, y Haar Cascade, aunque más simple de implementar, no ofrece la misma versatilidad para adaptarse a escenarios complejos o variados.

### **3.6.3 FASE 2: CREACIÓN DE LA BASE DE DATOS Y ETIQUETADO DESARROLLO DE BASE DE DATOS**

La base de datos consiste en un grupo de imágenes que muestran el objeto desde diversas perspectivas, ubicaciones y condiciones. Cuantas más imágenes variadas tengamos del objeto que queremos detectar, mejor será para el entrenamiento de YOLOv5. Además, es importante considerar que a medida que aumenta la cantidad de imágenes utilizadas en el entrenamiento, la precisión y eficacia de la detección también mejoran.

Dado que estamos buscando detectar dos tipos de objetos, específicamente armas de fuego cortas y armas blancas (cuchillos), se ha creado dos bases de datos separadas. Cada una de estas bases de datos contiene un total de 3000 imágenes. Como se mencionó previamente,

estas imágenes cubren una variedad de perspectivas para garantizar una detección eficiente y completa.

A continuación, se evidencia el tipo de imágenes que constituye la base de datos el cual se utilizara para el entrenamiento y la posterior detección del arma de fuego. La base de datos está compuesta por este tipo de imágenes, las cuales fueron tomadas en la Farmacia PharmaTodo. Este contexto proporciona una representación realista de cómo se verá el objeto en su entorno habitual, lo que facilita al algoritmo de detección adaptarse mejor a las condiciones del mundo real y al entorno que se ejecutará.



**Figura 4** Arma mostrada parcialmente



**Figura 3** Arma mostrada completamente



**Figura 5** Arma apenas visible

Aunque la base de datos ya contiene imágenes donde ya se evidencia el objeto a detectar, se llevó a cabo una recolección adicional de imágenes externas provenientes de diversas fuentes para mejorar su eficacia. A continuación, se muestra el tipo de imágenes adicionales que se han añadido.



**Figura 7** *Usuario sosteniendo un arma de fuego*



**Figura 6** *Usuario sosteniendo un arma de fuego en diferente posición*



**Figura 9** *Arma de fuego con escenario diferente*



**Figura 8** *Arma de fuego con escenario diferente*

El mismo procedimiento de toma de fotos en la farmacia PharmaTodo y la recopilación de imágenes de otras fuentes también fue realizada para la segunda base de datos, la cual está orientada a la detección de armas blancas (cuchillos). Todo esto con el objetivo de así asegurar una variedad de perspectivas y condiciones para mejorar la precisión y la generalización del modelo de detección.



## ETIQUETADO

El entrenamiento de Yolo v5 requiere que cada imagen que se utilizara debe estar etiquetada, el etiquetado proporcionar información sobre la ubicación y la clase de los objetos dentro de las imágenes. Estas etiquetas suelen estar en formato de coordenadas que delimitan los cuadros delimitadores (bounding boxes) alrededor de los objetos de interés, junto con la clase a la que pertenece cada objeto.

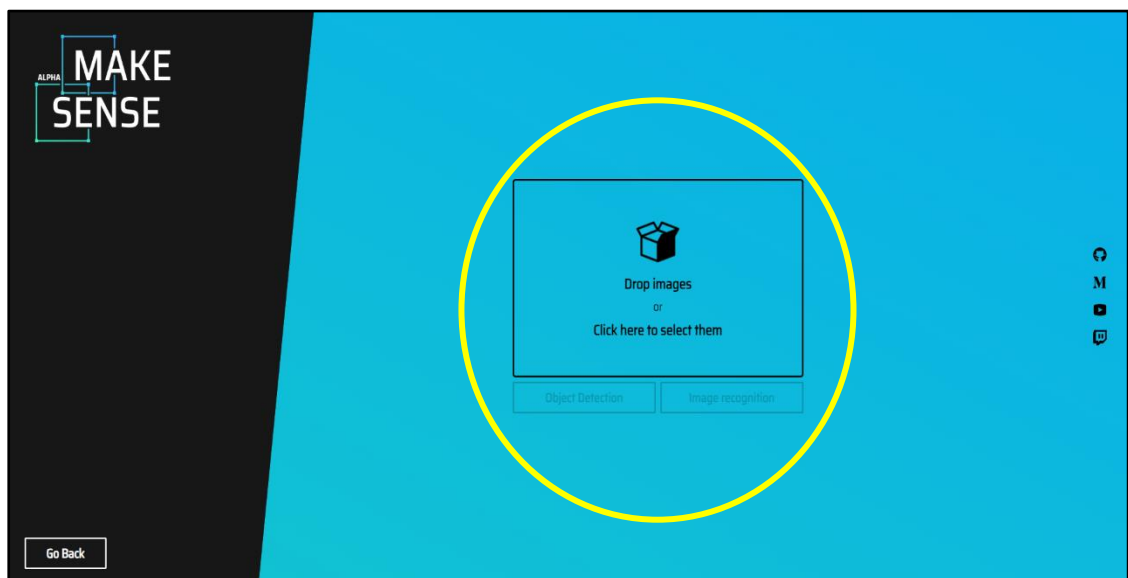
La información de las etiquetas es fundamental durante el entrenamiento del modelo, ya que permite que YOLOv5 aprenda a reconocer y localizar objetos dentro de las imágenes. Durante el proceso de entrenamiento, el algoritmo utiliza esta información para ajustar sus parámetros y mejorar su capacidad para detectar objetos con precisión en nuevas imágenes.

A continuación, se presenta el proceso de etiquetado que se utilizó:

- **Paso1: Cargar las imágenes de la base de datos a Makesense.Ai**

Makesense.AI esta es una plataforma en línea que ofrece herramientas para la creación, etiquetado y gestión de conjuntos de datos para tareas de visión artificial y aprendizaje automático, podemos acceder a su página web a través del siguiente link: <https://www.makesense.ai/>.

Una vez ingresado al link podemos observar la siguiente interfaz gráfica que personalmente es muy intuitiva de utilizar.

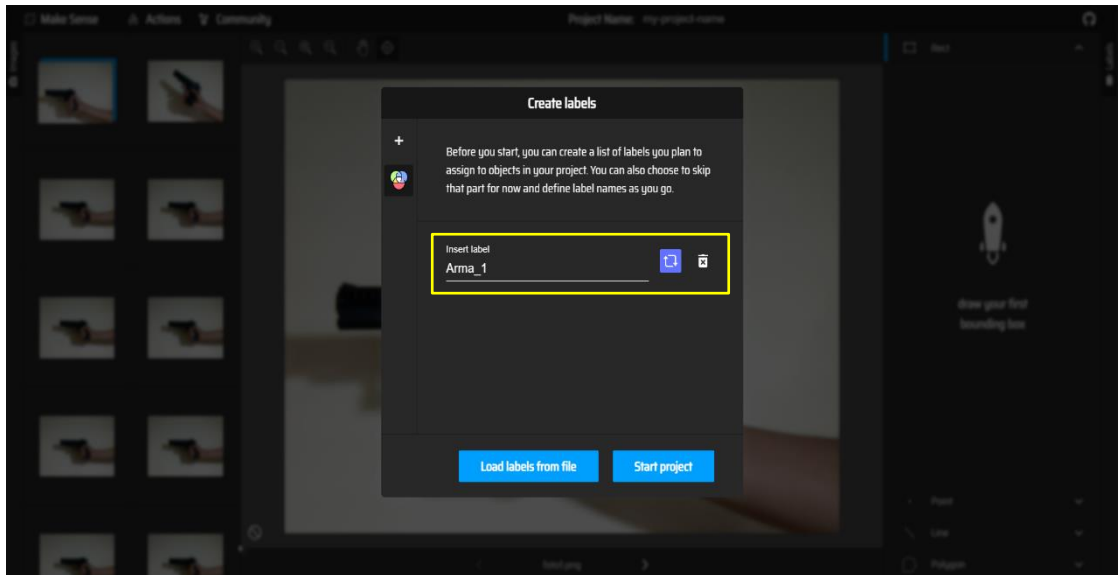


**Figura 10** Make Sense

Una vez dentro de la página debemos cargar o arrastrar nuestra base de datos al círculo amarillo, esto nos permitirá ingresar las imágenes y empezar el proceso de etiquetado.

- **Paso 2: Crear la etiqueta deseada**

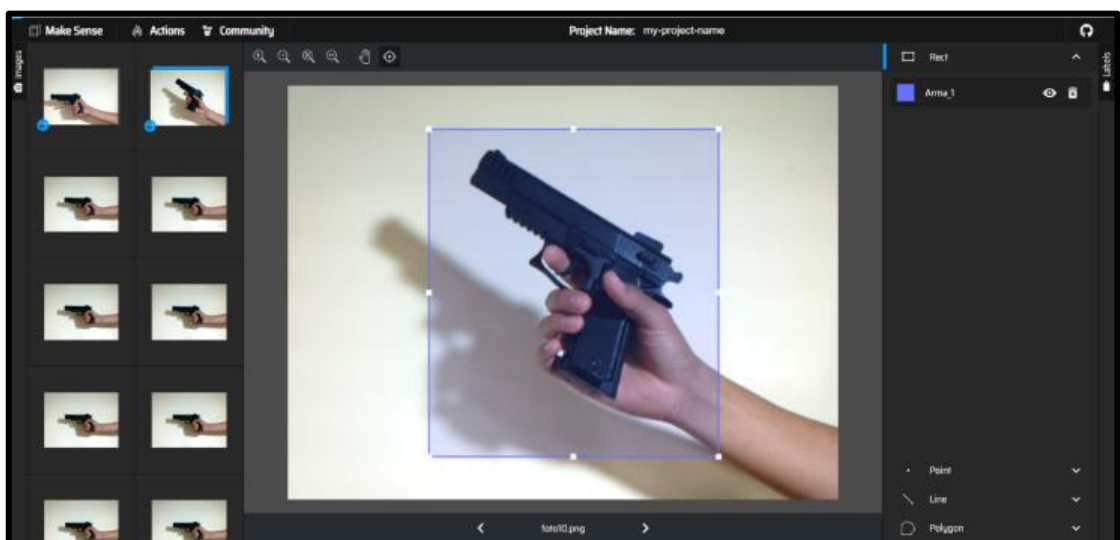
La etiqueta que usaremos será Arma\_1. Como se muestra en la imagen, hemos asignado este nombre a la etiqueta que rodea el recuadro amarillo.



**Figura 11** Definimos la etiqueta

- **Paso 3: Etiquetado**

Damos clic en Start Project y continuamos a la siguiente etapa del etiquetado, la cual es seleccionar el objeto a detectar, en este ejemplo en concreto seleccionaremos el arma.

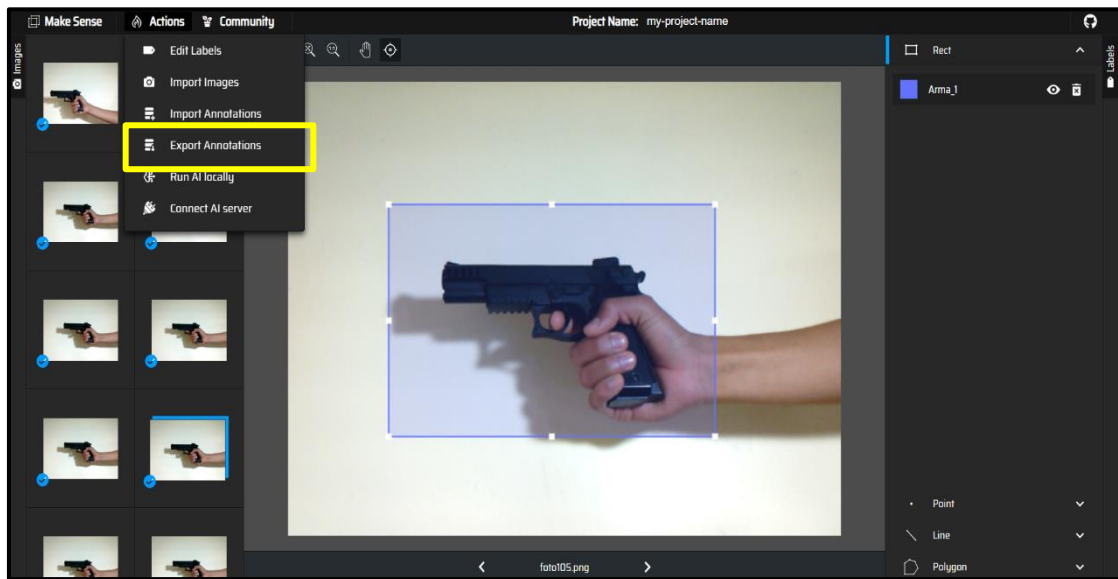


**Figura 12** Proceso de etiquetado

Como podemos observar solo seleccionamos el elemento a detectar, este proceso se debe realizar con todas las imágenes que se utilizaran para el proceso de entrenamiento.

- **Paso 3: Guardamos las etiquetas**

Por último, realizamos clic en la opción que se encuentra dentro del recuadro amarillo y descargamos nuestras etiquetas.



**Figura 13** Proceso de descarga de etiquetas

### 3.6.4 FASE 3: ENTRENAMIENTO YOLOV5

La fase de entrenamiento se constituye por varias partes de suma importancia que permiten que el entrenamiento sea posible, las cuales son las siguientes:

- **CLONACIÓN DEL REPOSITORIO DE YOLOV5**

La clonación del repositorio es esencial porque nos proporciona el código fuente y las herramientas necesarias para entrenar y trabajar con el modelo YOLOv5. Sin estos archivos, no podríamos ejecutar los scripts de entrenamiento, evaluar el modelo ni exportar los resultados.

- **ESTRUCTURA DE LA BASE DE DATOS**

Es esencial tener en cuenta la estructura básica de la base de datos para el entrenamiento con YOLOv5 es de suma importancia, esta estructura nos permitirá lograr un entrenamiento efectivo. Esta estructura proporciona los fundamentos necesarios para la detección de

objetos en imágenes y es fundamental comprenderla para aprovechar al máximo el proceso de entrenamiento, a continuación, se muestra la estructura adecuada que se debe tener

```
C:\Users\WinUser>cd C:\Users\WinUser\Documents\YOLO V55\data
C:\Users\WinUser\Documents\YOLO V55\data>tree
Listado de rutas de carpetas
El número de serie del volumen es 966B-5E81
C: .
├── images
│   ├── train
│   └── val
└── labels
    ├── train
    └── val
```

**Figura 14** Estructura de la base de datos

Como se puede observar se ha creado una carpeta llamada "data" que contiene dos subcarpetas: una llamada "imágenes" y otra llamada "labels". Dentro de la carpeta "imágenes" hay dos subcarpetas adicionales:

La subcarpeta "train" contiene las imágenes que el algoritmo utilizará para aprender y practicar. Durante este proceso, el modelo YOLO v5 analiza estas imágenes y hace predicciones sobre la ubicación y la identidad de los objetos en ellas.

La subcarpeta "val" contiene imágenes de validación, que funcionan como un examen para el modelo después de haber practicado con las imágenes de entrenamiento. Estas imágenes son distintas de las utilizadas en el entrenamiento, lo que representa situaciones nuevas y desconocidas para el modelo.

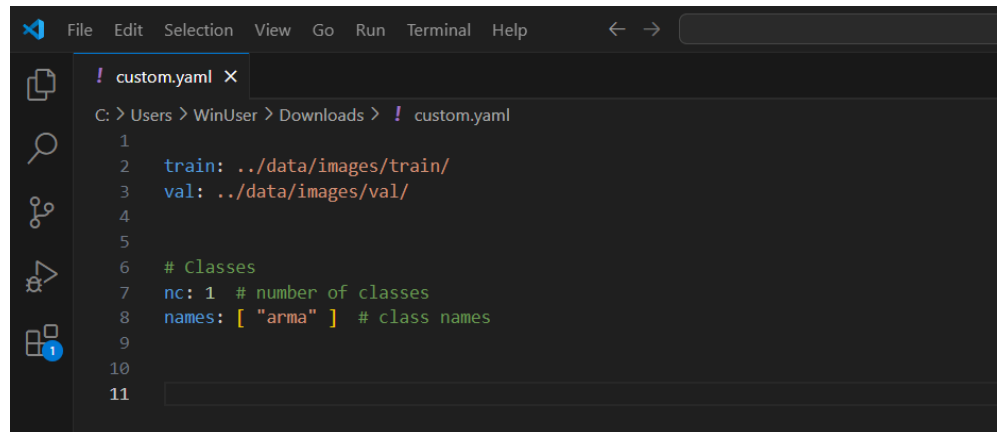
La carpeta "labels" contiene etiquetas para las imágenes tanto de las carpetas "train" como de "val". Estas etiquetas proporcionan información sobre la ubicación y la clase de los objetos detectados en las imágenes. Esto permite que el modelo YOLO v5 aprenda de manera supervisada, utilizando esta información durante el proceso de entrenamiento para mejorar su capacidad de detección de objetos.

- **CONFIGURACIÓN DEL DATASET**

Es imprescindible crear un archivo de configuración del dataset en formato YAML. Este archivo desempeña un papel fundamental al proporcionar al código de YOLOv5 información vital sobre la disposición de los datos de entrenamiento y validación, así como sobre las clases presentes en el conjunto de datos. Sin esta guía, el modelo estaría desorientado,

incapaz de localizar los datos o de reconocer la diversidad de clases presentes en nuestro conjunto de datos. Por lo tanto, la creación y correcta configuración de este archivo son pasos cruciales para asegurar un entrenamiento efectivo y exitoso del modelo YOLOv5.

A continuación, se expone la configuración realizada del archivo YAML:



```
File Edit Selection View Go Run Terminal Help
! custom.yaml x
C: > Users > WinUser > Downloads > ! custom.yaml
1
2 train: ../data/images/train/
3 val: ../data/images/val/
4
5
6 # Classes
7 nc: 1 # number of classes
8 names: [ "arma" ] # class names
9
10
11
```

**Figura 15** Configuración del archivo YAML

Esto incluye definir las rutas de las imágenes de entrenamiento y validación, así como el número y los nombres de las clases presentes en el conjunto de datos.

## **DESARROLLO DEL SCRIPT EN PYTHON PARA EL ENTRENAMIENTO DEL MODELO YOLOV5**

- **VERIFICACIÓN SI CUDA ESTA DISPONIBLE**

Después de establecer los fundamentos del script, procedemos a verificar la disponibilidad de CUDA en nuestra configuración. CUDA, una plataforma de computación paralela desarrollada por NVIDIA, constituye un elemento clave para potenciar el rendimiento en cálculos numéricos complejos al aprovechar el poder de procesamiento de las unidades de procesamiento gráfico (GPU). Es crucial abordar este paso inicialmente, ya que la omisión de esta verificación puede tener repercusiones significativas en el proceso de entrenamiento.

Si el script se ejecuta sin aprovechar CUDA, el procesamiento recae en la unidad central de procesamiento (CPU), lo que no solo ralentiza el proceso de manera considerable, sino que también puede ocasionar bloqueos del sistema debido al intenso consumo de recursos.

## **DESARROLLO DE ENTRENAMIENTO**

Una vez confirmada la capacidad del sistema para utilizar la GPU, procedemos a definir los parámetros necesarios para el entrenamiento respectivo. En primer lugar, se establece el archivo "train.py", el cual dirige el proceso de entrenamiento que YOLOv5 llevará a cabo con los datos disponibles. Es importante destacar que este archivo no necesita ser creado, ya que se toma del repositorio previamente clonado de YOLOv5.

A continuación, se configura el parámetro "--img 640", el cual determina que las imágenes de entrada durante el entrenamiento tendrán una resolución de 640 píxeles. Es relevante considerar que, si alguna imagen excede esta resolución, el modelo realizará automáticamente un ajuste para adaptarse al tamaño establecido.

Luego, se especifica el parámetro "--batch 16", el cual define el tamaño del lote, es decir, cuántas imágenes se procesarán simultáneamente en cada paso de entrenamiento. Dado que se cuenta con un conjunto de datos de 3000 imágenes y se ha establecido un tamaño de lote de 16, durante cada época de entrenamiento el modelo procesará el conjunto de datos en lotes de 16 imágenes.

Posteriormente, se define el parámetro "--epochs", que indica el número de épocas que se llevarán a cabo. En este caso, se han realizado 6 épocas de entrenamiento. Durante cada una de estas épocas, el modelo examina todas las imágenes de entrenamiento y ajusta sus pesos y parámetros para mejorar las predicciones. Entrenar durante múltiples épocas es esencial para permitir que el modelo practique lo suficiente y se adapte a los patrones presentes en los datos, similar a cómo se mejora una habilidad a través de la práctica repetida.

Finalmente, se definen los parámetros "--data custom.yaml" y "--weights yolov5s.pt". El primero especifica el archivo YAML que contiene la configuración de los datos de entrenamiento, mientras que el segundo indica el archivo de pesos preentrenados que se utilizarán como punto de partida para el entrenamiento, en este caso, "yolov5s.pt".

```
def train_yolov5():
    """
    Ejecuta el entrenamiento de YOLOv5.
    """
    print("Ejecutando el entrenamiento de YOLOv5...")
    subprocess.check_call([
        'python', 'train.py',
        '--img', '640',
        '--batch', '16',
        '--epochs', '3',
        '--data', 'coco128.yaml',
        '--weights', 'yolov5s.pt',
        '--cache'
    ])
```

**Figura 16** Configuración del Algoritmo YOLOv5

## ANÁLISIS DEL ENTRENAMIENTO Y EVALUACIÓN DE DESEMPEÑO DEL MODELO YOLOV5 PARA LA DETECCIÓN DE OBJETOS

Internamente, el modelo YOLOv5 realiza su entrenamiento de la siguiente manera: la imagen proporciona información crucial para evaluar su desempeño y progreso. En este proceso, se configura el entrenamiento a lo largo de seis épocas, utilizando imágenes de 640 píxeles y un batch size de 16 para optimizar el aprendizaje. Además, se aprovechan los pesos preentrenados (yolov5x.pt) y se almacenan en caché los datos para acelerar el proceso.

Al inicio del entrenamiento, se enfrentan a pérdidas iniciales significativas. En la primera época, la pérdida de la caja es de 0.09407, la del objeto es de 0.0283, y la de clase es cero. Estos valores reflejan la fase inicial de adaptación del modelo a los datos, donde aún no se ha capturado completamente la complejidad de las características.

Sin embargo, a medida que avanzan las épocas, se observa una mejora gradual en las pérdidas. Para la quinta época, la pérdida de la caja se reduce a 0.04415 y la del objeto a 0.02325. Esto indica que el modelo está refinando su capacidad para identificar y clasificar objetos con mayor precisión a medida que progresa el entrenamiento.

Este progreso también se evidencia en las métricas de evaluación. Inicialmente, la precisión es modesta, con un mAP50 de solo 0.0469 y un mAP50-95 de 0.00599. No obstante, hacia la última época, estas métricas muestran un crecimiento significativo. Se logra una precisión (P) del 95.3%, un recall (R) del 99.2%, y un mAP50 de 98.6%, lo que indica una mejora notable en la capacidad del modelo para detectar y clasificar objetos con precisión.

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	640: 100%	10/10	[00:15<00:00, 1.58s/it]
0/5	14.7G	0.09407	0.0283	0	39	640: 100%	100%	10/10	[00:15<00:00, 1.58s/it]
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	2/2	[00:05<00:00, 2.74s/it]
	all	41	41	0.00939	0.463	0.0469	0.00599		
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	640: 100%	10/10	[00:10<00:00, 1.01s/it]
1/5	15.3G	0.07319	0.031	0	35	640: 100%	100%	10/10	[00:10<00:00, 1.01s/it]
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	2/2	[00:08<00:00, 4.30s/it]
	all	41	41	0.786	0.927	0.951	0.443		
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	640: 100%	10/10	[00:09<00:00, 1.01it/s]
2/5	15.4G	0.06081	0.02925	0	46	640: 100%	100%	10/10	[00:09<00:00, 1.01it/s]
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	2/2	[00:08<00:00, 4.11s/it]
	all	41	41	0.892	0.951	0.951	0.433		
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	640: 100%	10/10	[00:10<00:00, 1.01s/it]
3/5	15.4G	0.05037	0.02692	0	39	640: 100%	100%	10/10	[00:10<00:00, 1.01s/it]
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	2/2	[00:01<00:00, 1.42it/s]
	all	41	41	0.403	0.856	0.418	0.159		
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	640: 100%	10/10	[00:10<00:00, 1.01s/it]
4/5	15.4G	0.04507	0.02485	0	34	640: 100%	100%	10/10	[00:10<00:00, 1.01s/it]
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	2/2	[00:01<00:00, 1.83it/s]
	all	41	41	0.474	0.927	0.626	0.308		
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	640: 100%	10/10	[00:10<00:00, 1.02s/it]
5/5	15.4G	0.04415	0.02325	0	30	640: 100%	100%	10/10	[00:10<00:00, 1.02s/it]
	Class	Images	Instances	P	R	mAP50	mAP50-95: 100%	2/2	[00:01<00:00, 1.86it/s]
	all	41	41	0.953	0.992	0.986	0.515		

**Figura 17** *Entrenamiento del modelo Yolo v5*

### **3.6.5 FASE 4: DESARROLLO DE LA CODIFICACIÓN EN PYTHON PARA LA IMPLEMENTACIÓN DE FUNCIONALIDADES DE ENVÍO DE INFORMACIÓN, CAPTURA DE IMÁGENES Y DETECCIÓN DE OBJETOS.**

Durante esta etapa, la programación en Python se destaca como un elemento clave. Esta elección facilita la integración del archivo `best.pt`, el cual, como se mencionó anteriormente, se genera a partir de entrenamientos previos. Este archivo es esencial para ejecutar el análisis de video en tiempo real y establecer el proceso de detección de objetos. Además, el código está cuidadosamente estructurado para transferir datos significativos a la base de datos. Asimismo, se configura para capturar imágenes en el momento exacto en que se detecta el objeto objetivo deseado.

#### **PRIMER CÓDIGO: IDENTIFICACIÓN DE ARMAS DE FUEGO, TRANSMISIÓN DE DATOS Y EVIDENCIA VISUAL.**

- **CONFIGURACIÓN DE ENTORNO Y CARGA DE DEPENDENCIAS**

En el desarrollo de la aplicación, se ha hecho uso de una variedad de bibliotecas y módulos en Python para diferentes propósitos. Para el procesamiento de imágenes y videos, así como la detección de objetos, hemos empleado la biblioteca OpenCV, conocida como `cv2`. Esta biblioteca nos brinda herramientas poderosas para realizar estas tareas de manera eficiente. Además, hemos utilizado la popular biblioteca NumPy, renombrada como `np`, para trabajar con matrices y realizar operaciones matemáticas, esencial para el procesamiento de imágenes que comúnmente se representan como matrices numéricas en la computación.

`firebase_admin`, junto con sus módulos `credentials`, `firestore` y `storage`, se han integrado para interactuar con los servicios de Firebase desde nuestra aplicación Python, permitiéndonos acceder y manipular datos en Firebase Storage, Firestore, entre otros. Para gestionar fechas y horas en nuestro programa, hemos empleado el módulo `Datetime`, mientras que `Pathlib` ha sido útil para trabajar con rutas de archivos y directorios de manera más intuitiva y orientada a objetos. Por otro lado, Torch nos ha proporcionado las herramientas necesarias para trabajar con PyTorch, un marco de aprendizaje profundo ampliamente utilizado, permitiéndonos cargar y utilizar modelos de aprendizaje profundo en nuestra aplicación.



Finalmente, el módulo `Os` nos ha facilitado la interacción con el sistema operativo subyacente en Python, especialmente para realizar operaciones relacionadas con archivos, como la eliminación de archivos locales después de cargarlos en Firebase Storage. En conjunto, estas bibliotecas y módulos han sido fundamentales para el desarrollo de nuestra aplicación, brindándonos las herramientas necesarias para llevar a cabo diversas tareas de manera eficiente y efectiva.

- **CONFIGURACIÓN DE FIREBASE Y CARGA DEL MODELO DE DETECCIÓN**

```
# Configuración de Firebase
cred =
credentials.Certificate("C:/Users/WinUser/Documents/1.visualcode_firebase_appand
roid/appvisualocode-c69c7a18db7f.json")
firebase_admin.initialize_app(cred, {
    'storageBucket': 'appvisualocode.appspot.com'
})

# Leemos el modelo de detección de objetos (YOLOv5)
model = torch.hub.load('ultralytics/yolov5', 'custom',
path='C:/Users/WinUser/Documents/YOLOV5/Preuba_1_ARMA_(FUNCIONAL)/arma1.pt')
```

**Figura 18** Configuración de Firebase y carga del modelo de detección de objetos

A continuación, se realizó una serie de pasos relacionados con Firebase para establecer una conexión efectiva con sus servicios. En primer lugar, hemos configurado la conexión con Firebase mediante un archivo de credenciales (`appvisualocode-c69c7a18db7f.json`), lo que nos permite autenticar nuestra aplicación Python con Firebase Admin y especificar el bucket de almacenamiento (`storageBucket`) que utilizaremos para almacenar recursos en la nube. Posteriormente, hemos establecido conexiones a Firestore y Firebase Storage utilizando los módulos correspondientes de Firebase Admin.

Esta configuración nos habilita para interactuar de manera directa y eficiente con la base de datos Firestore y el almacenamiento en la nube de Firebase desde nuestra aplicación Python. Además, hemos realizado la carga de un modelo de detección de objetos YOLOv5 utilizando PyTorch. Este modelo se ha cargado desde la ubicación especificada en la ruta `'C:/Users/WinUser/Documents/YOLOV5/Preuba_1_ARMA_(FUNCIONAL)/arma1.pt'`. Dicho modelo será empleado más adelante en el programa para detectar objetos en imágenes o videos, permitiendo así una funcionalidad integral de detección de objetos en nuestra aplicación.

- **CONTROL DE ENVÍO DE NOTIFICACIONES Y REGISTRO DE DETECCIONES**

```
# Función para enviar notificación y escribir en Firestore
def notificar_arma():
    global notificacion_enviada
    if not notificacion_enviada:
        # Obtenemos la hora actual
        hora_actual = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

        # Datos a escribir en Firestore
        datos_deteccion = {
            'fecha_y_hora': hora_actual,
            'ubicacion': 'Farmacia PharmaTodo',
            'objeto_detectado': 'Arma de fuego'
        }

        # Escribe los datos en Firestore
        db.collection('deteccion_arma').add(datos_deteccion)
        print('Arma detectada:', hora_actual, 'en Farmacia PharmaTodo')

    # Tomar captura y enviar a Firebase Storage
    tomar_y_enviar_captura()
```

**Figura 19** Control de envío de notificaciones

La función `notificar_arma` se encarga de enviar la notificación y registrar información relevante en Firestore. Antes de enviar la notificación, verifica el estado de la bandera `notificacion_enviada`. Si está en `false` (es decir, la notificación aún no se ha enviado), procede a enviarla y luego actualiza la bandera a `true` para indicar que la notificación ha sido enviada. Si la bandera ya es `true`, lo que indica que la notificación ya se ha enviado, la función no realiza ninguna acción adicional.

Dentro de la función `notificar_arma`, se obtiene la hora actual y se preparan los datos de la detección para ser escritos en Firestore. Estos datos incluyen la fecha y hora de la detección, la ubicación donde se detectó el arma y el objeto detectado.

Después de preparar los datos, la función escribe esta información en Firestore utilizando el cliente de Firestore y la colección específica. Además de enviar la notificación, la función también llama a otra función, `tomar_y_enviar_captura`, para capturar una imagen de la detección y enviarla a Firebase Storage.

- **PROCESAMIENTO Y VISUALIZACIÓN DE DETECCIONES EN TIEMPO REAL**

```
while True:
    # Lectura de frames
    ret, frame = cap.read()

    # Detección de objetos
    results = model(frame)

    # Obtención de información sobre las detecciones
    info = results.pandas().xyxy[0] # Predicciones de la primera imagen
    print(info)

    # Si se detecta un arma y no se ha enviado la notificación, enviar
    notificación y escribir en Firestore
    armas_detectadas = info[info['name'] == 'arma'] # Filtramos las
    detecciones de armas
    if notificación_enviada == False and not armas_detectadas.empty:
        # Enviar notificación y tomar captura
        notificar_arma()

    # Mostramos la detección en tiempo real
    cv2.imshow('Detector de Objetos', np.squeeze(results.render()))
```

**Figura 20** Captura y envío de imágenes a Firebase Storage durante la detección en tiempo real

En el proceso de análisis del video, se llevan a cabo una serie de pasos clave para la detección de objetos. En primer lugar, se define el video que se va a procesar, especificando el nombre del archivo como 'JL1.mp4'. Este video será la fuente de entrada para la detección de objetos. Luego, se inicia el bucle principal del código con `while True:`, lo que indica que el procesamiento del video se realizará de manera continua hasta que se detenga manualmente o se cumpla una condición específica.

Dentro de este bucle, se lee cada frame del video utilizando `cap.read()`, capturando el frame actual y almacenándolo en una variable. La variable `ret` indica si la operación de lectura fue exitosa o no. Utilizando un modelo pre-entrenado, se realiza la detección de objetos en el frame actual del video, almacenando el resultado en una variable. Posteriormente, se extrae información sobre las detecciones del primer frame del resultado de la detección, incluyendo detalles como coordenadas y etiquetas de los objetos detectados.

Si se detecta un arma en el frame actual y aún no se ha enviado una notificación, se llama a la función `notificar_arma()` para enviar la notificación correspondiente y tomar una captura de la detección. Para visualizar la detección en tiempo real, se muestra la detección de objetos en una ventana utilizando `cv2.imshow()`, permitiendo al usuario observar el proceso de detección mientras se procesa el video.

## SEGUNDO CÓDIGO: INTEGRACIÓN DE DETECCIÓN DE POSTURA EN TIEMPO REAL, ALMACENAMIENTO DE IMÁGENES DE MANOS LEVANTADAS Y REGISTRO EN FIREBASE

En esta etapa se importa la utilidad de dibujo de MediaPipe y el modelo de detección de pose para visualizar los resultados y detectar la pose humana en un video, respectivamente. Se configura Firebase proporcionando la ruta del archivo JSON de credenciales, que contiene la información necesaria para autenticar la aplicación con Firebase. Se inicializa la aplicación de Firebase utilizando estas credenciales y se configura el bucket de almacenamiento en la nube que se utilizará para guardar las capturas de pantalla.

- **PROCESAMIENTO DE VIDEO EN TIEMPO REAL PARA DETECCIÓN DE MANOS LEVANTADAS Y REGISTRO EN FIREBASE**

```
# Obtener las coordenadas y alturas de las muñecas izquierda y derecha
left_wrist_y = landmarks[mp_pose.PoseLandmark.LEFT_WRIST].y *
height
right_wrist_y = landmarks[mp_pose.PoseLandmark.RIGHT_WRIST].y *
height

# Obtener las alturas de los hombros
left_shoulder_y = landmarks[mp_pose.PoseLandmark.LEFT_SHOULDER].y
* height
right_shoulder_y =
landmarks[mp_pose.PoseLandmark.RIGHT_SHOULDER].y * height

# Verificar si ambas muñecas están levantadas
if left_wrist_y < left_shoulder_y and right_wrist_y <
right_shoulder_y and not deteccion_enviada:
    cv2.putText(frame, ";Manos levantadas!", (50, 50),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
```

**Figura 21** *Procesamiento de Video en Tiempo Real para Detección de Manos Levantadas y Registro en Firebase*

Cada cuadro del video se procesa para detectar la pose utilizando el modelo de detección de pose de MediaPipe, y los resultados se guardan en la variable results. Se verifica si se han detectado landmarks de la pose en la imagen del cuadro; si estos no son None, indica que se ha detectado la pose en la imagen. Luego, se calculan las alturas de las muñecas izquierda y derecha, así como las alturas de los hombros, en relación con la altura total de la imagen. Si ambas muñecas están por encima de los hombros y la detección de manos levantadas no se ha enviado previamente, se superpone un texto en la imagen indicando que las manos están levantadas, se obtiene la hora actual y se guarda la información de la detección en Firestore, incluyendo la fecha, hora y ubicación.

### **3.6.6 FASE 5: DESARROLLO Y DESPLIEGUE DE LA APLICACIÓN CENTINELADEFENDER.**

La culminación del proyecto se destaca por la creación y lanzamiento de una aplicación móvil, concebida para alertar de manera eficaz ante la detección de armas. Al desarrollar esta aplicación, el objetivo principal fue garantizar su accesibilidad y facilidad de uso para todos los usuarios. Por lo tanto, el compromiso primordial fue diseñar y codificar la aplicación utilizando las herramientas proporcionadas por Android Studio, asegurando así una experiencia fluida y sin complicaciones.

En primer lugar, la pantalla de inicio brinda una visión general y clara de las funciones principales de la aplicación, permitiendo a los usuarios acceder rápidamente a las herramientas necesarias en situaciones críticas. La disposición cuidadosamente organizada de los elementos en la pantalla de inicio facilita la comprensión instantánea de la aplicación y sus capacidades.

- **ASIGNACIÓN DE PERMISOS PARA LA GESTIÓN DE CONTACTOS, COMUNICACIÓN SMS y ACCESO A LA RED.**

Los permisos como `android.permission.READ_CONTACTS` posibilitan que la aplicación acceda a los contactos almacenados en el dispositivo del usuario, lo que resulta útil para funciones como la selección de destinatarios al enviar mensajes. Por otro lado, `android.permission.SEND_SMS` permite a la aplicación enviar mensajes de texto desde el dispositivo, facilitando la implementación de características como alertas o notificaciones automáticas. Además, el permiso `android.permission.INTERNET` es esencial para que la aplicación pueda acceder a internet, lo que le permite comunicarse con servidores remotos, cargar o descargar datos, y acceder a servicios en línea.

- **CONFIGURACIÓN DE DEPENDENCIAS DE FIREBASE EN PROYECTO ANDROID**

La línea `implementation(platform("com.google.firebase:firebase-bom:32.7.4"))` utiliza la plataforma de Firebase BOM (Bill of Materials) para administrar las dependencias de Firebase en el proyecto. Esto garantiza que todas las bibliotecas de Firebase utilizadas en el proyecto estén en la misma versión, en este caso, la versión 32.7.4. Por otro lado, la línea

implementation("com.google.firebase:firebase-firestore:23.0.0") asegura que la biblioteca de Firebase Firestore en la versión 23.0.0 esté disponible en el proyecto de Android. Esto permite utilizar las funcionalidades de Firebase Firestore, como la base de datos en tiempo real, en la aplicación.

- **ANÁLISIS DE LA INTERFAZ GRÁFICA DE LA APLICACIÓN**

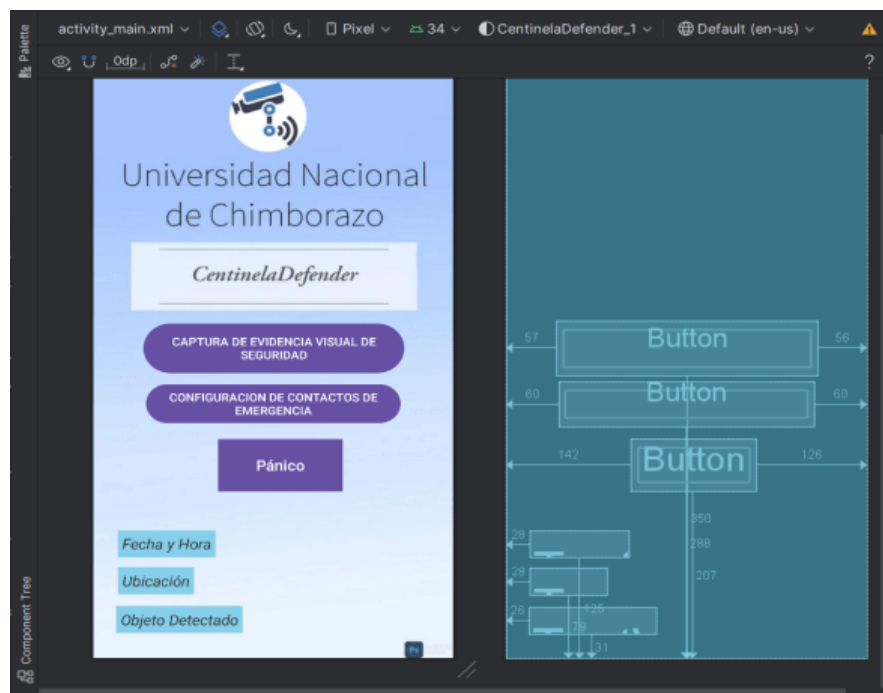
#### **ANÁLISIS DE LA INTERFAZ PRINCIPAL**

El botón de "Captura de evidencia visual de seguridad" ofrece acceso directo al servicio de almacenamiento de imágenes en Firebase. Esta funcionalidad es crucial para visualizar las instantáneas capturadas por el sistema ante la detección de armas de fuego o cuchillos, permitiendo una revisión detallada de los eventos registrados.

El botón de "Configuración de contactos de emergencia" permite al usuario seleccionar varios números de teléfono de su directorio de contactos. En situaciones de emergencia, estos contactos recibirán alertas automáticamente, asegurando una respuesta rápida y efectiva en momentos críticos. El botón de "Pánico" proporciona una forma rápida y eficiente de notificar a los contactos seleccionados en caso de emergencia. Al activarse, envía mensajes instantáneos para alertar sobre la situación, proporcionando una respuesta inmediata ante cualquier eventualidad.

Por último, la aplicación muestra datos cruciales como la fecha, hora, ubicación y objeto detectado. Estos detalles son esenciales para comprender y analizar los eventos capturados por el sistema de seguridad, ofreciendo una visión completa de la situación en tiempo real. Además de las funcionalidades mencionadas, la aplicación está configurada para emitir notificaciones que incluyen información previamente codificada en Python, como fecha y hora, ubicación y el objeto detectado.

Se ha ajustado para generar notificaciones sonoras con un tono prominente, similar al de una alarma, para asegurar una mayor atención y respuesta inmediata ante la detección de objetos.



**Figura 22** *Interfaz principal*

## **ANÁLISIS DE LA INTERFAZ SECUNDARIA - SELECCIÓN DE CONTACTOS DE EMERGENCIA**

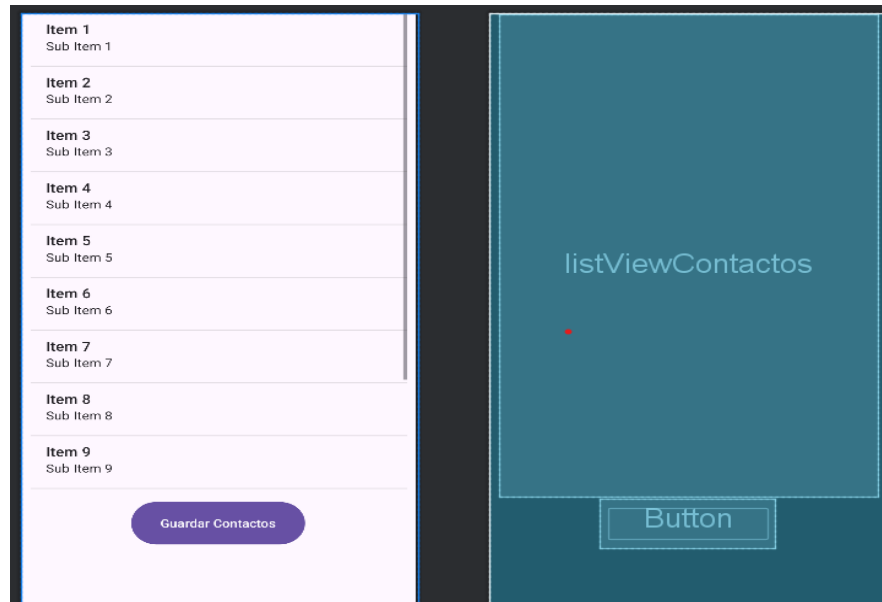
Como se puede observar, la interfaz gráfica está constituida por varios elementos clave, entre los cuales destaca un ListView. Este ListView es donde se cargan los contactos desde el directorio del usuario. Una vez cargados, el usuario puede seleccionar los contactos que desee, personalizando así la lista de destinatarios que recibirán las alertas de emergencia.

Después de seleccionar los contactos, el usuario puede guardar esta configuración utilizando el botón "Guardar Contactos". Al presionar este botón, los contactos seleccionados se almacenan y quedan listos para ser utilizados cuando sea necesario. Esta funcionalidad es crucial, ya que permite una rápida y eficiente comunicación en situaciones de emergencia.

En la pantalla principal de la aplicación, se encuentra el botón de "Pánico". Al activarse este botón, se envían mensajes de emergencia instantáneamente a los contactos previamente guardados. Este sistema garantiza que, en momentos críticos, las personas seleccionadas recibirán alertas de manera inmediata, facilitando una respuesta rápida y coordinada ante cualquier eventualidad.

La interfaz gráfica, con su diseño intuitivo, asegura que los usuarios puedan gestionar sus contactos de emergencia de forma sencilla y eficaz. Este proceso, desde la carga y selección

de contactos hasta el envío de alertas, está diseñado para maximizar la seguridad y la eficiencia en situaciones de emergencia.



**Figura 23** Interfaz de pantalla de selección de contactos de emergencia

## **DESGLOSE DE LA LÓGICA DEL MAINACTIVITY PRINCIPAL**

La estructura lógica de la aplicación se basa en seis etapas fundamentales, cada uno diseñado para desempeñar un papel crucial en el logro de los objetivos de la aplicación. Estos componentes forman el núcleo de la lógica del programa, proporcionando las funcionalidades necesarias para su correcto funcionamiento.

- **ETAPA 1: INICIALIZACIÓN Y CONFIGURACIÓN DE FIREBASE FIRESTORE**

La etapa de inicialización y configuración de Firebase Firestore constituye el fundamento de la comunicación efectiva entre la aplicación y la base de datos en la nube. Este proceso es esencial para establecer la conexión y permitir la sincronización de datos en tiempo real. Al crear una instancia de `FirebaseFirestore`, la aplicación establece una conexión activa con la base de datos de Firestore, lo que facilita la recuperación y actualización de información de forma dinámica. La implementación del método `addSnapshotListener` permite que la aplicación esté al tanto de cualquier cambio en las colecciones de datos específicas, lo que



garantiza una respuesta inmediata a actualizaciones, inserciones o eliminaciones de datos en la base de datos.

- **ETAPA 2: CREACIÓN DEL CANAL DE NOTIFICACIONES**

La creación del canal de notificación, se realiza a través de la elaboración del método `createNotificationChannel()`, es un paso esencial para garantizar una adecuada gestión de las notificaciones en la aplicación. Este proceso establece un canal de comunicación dedicado a través del cual se enviarán las notificaciones al usuario. Al llamar a este método durante la inicialización de la aplicación, se configuran atributos fundamentales del canal, tales como su identificador único, nombre descriptivo y una breve explicación de su propósito.

Además, se define el nivel de importancia de las notificaciones utilizando la constante `NotificationManager.IMPORTANCE_DEFAULT`, lo que determina su impacto visual y sonoro en el dispositivo del usuario. Dentro de este método, se emplean los `AudioAttributes` para definir las características del sonido de notificación, permitiendo una experiencia auditiva coherente y personalizable para el usuario. Este aspecto incluye la capacidad de establecer la ruta del sonido de alerta.

Es importante destacar que estas notificaciones se generan y gestionan localmente en el dispositivo del usuario, lo que garantiza una respuesta rápida y una experiencia de usuario fluida. Finalmente, una vez configurado, el canal se registra en el sistema mediante `NotificationManager`, asegurando que las notificaciones se entreguen de manera oportuna y que el usuario pueda gestionarlas fácilmente desde la configuración del dispositivo.

- **ETAPA 3: INTERACCIÓN CON LA INTERFAZ GRÁFICA**

La ampliación está constituida por varios botones que facilitan la navegación y la interacción del usuario con la aplicación. Estos botones, definidos en el archivo de diseño XML correspondiente, se utilizan para acceder a diferentes secciones y funciones de la aplicación.

Uno de los botones, etiquetado como `btn`, dirige al usuario a la sección de "pantalla\_evidencia\_1". Al hacer clic en este botón, se activa la visualización de la evidencia relacionada con la detección, permitiendo al usuario examinar la información visual presentada en la página `firestore`.

El siguiente botón, conocido como btnContact, lleva al usuario a la sección de "emergenciacontactos". Al interactuar con este botón, el usuario puede gestionar los contactos de emergencia que recibirán alertas en situaciones críticas.

El tercer botón, identificado como btnPanic, tiene una función de seguridad importante. Cuando se activa este botón, se desencadena el envío de mensajes de emergencia a los contactos preseleccionados, proporcionando una forma rápida y efectiva de solicitar ayuda en momentos críticos.

- **ETAPA 4: MONITOREO Y PROCESAMIENTO DE CAMBIOS DE FIRESTORE**

Dentro del código, se implementa un sistema de observación activo en la colección Firestore "deteccion\_arma", mediante el método addSnapshotListener(). Este mecanismo se encarga de detectar y procesar de manera automática cualquier cambio en dicha colección, ya sea una actualización de datos o la inserción de nuevos documentos.

Cuando se produce un cambio en la colección, el sistema entra en acción, desencadenando el método onEvent() que procesa la información contenida en los documentos afectados. Esta información, que comprende datos cruciales como la fecha y hora de detección, la ubicación y el objeto detectado, se extrae y se utiliza para actualizar dinámicamente la interfaz de usuario y, en paralelo, generar notificaciones para informar al usuario sobre los cambios ocurridos.

Este proceso de procesamiento de cambios en Firestore asegura que la aplicación esté constantemente sincronizada con la base de datos en la nube, garantizando así que la información mostrada al usuario esté siempre actualizada y refleje fielmente el estado más reciente de los eventos detectados.

- **ETAPA 5: MANEJO DE NOTIFICACIONES**

El manejo de notificaciones en la aplicación se centraliza en el método mostrarNotificacion(), el cual se encarga de construir y mostrar notificaciones al usuario. Este método utiliza la clase NotificationCompat.Builder para crear una notificación con un estilo personalizado, en este caso, BigTextStyle, que permite mostrar un texto largo en la notificación. Se establecen varios atributos de la notificación, como el ícono pequeño, el

título y el contenido, utilizando los parámetros proporcionados al llamar al método. Además, se configura el nivel de prioridad de la notificación como predeterminado. Una vez configurada la notificación, se utiliza `NotificationManagerCompat` para obtener una instancia del administrador de notificaciones y se muestra la notificación llamando al método `notify()` con el identificador único de la notificación y el objeto de notificación construido.

Este enfoque de manejo de notificaciones proporciona una manera efectiva y estructurada de comunicar información importante al usuario, garantizando que las notificaciones sean mostradas de manera adecuada y cumplan con los estándares de diseño y usabilidad de Android. Además, al utilizar la clase `NotificationCompat` se asegura la compatibilidad con diferentes versiones de Android, lo que permite que la aplicación ofrezca una experiencia consistente independientemente del dispositivo utilizado.

- **ETAPA 6: ENVIÓ DE MENSAJES SMS**

Mediante la utilización de la función `enviarMensajeSms()` es posible enviar mensajes SMS desde la aplicación. Luego, recorre la lista de números de teléfono proporcionados como argumento y envía el mensaje especificado a cada uno de ellos utilizando el método `sendTextMessage()`. Este método permite enviar un mensaje de texto con el contenido especificado a un número de teléfono específico, sin necesidad de una aplicación de mensajería de terceros.

Este método de envío de mensajes SMS integrado en la aplicación brinda una forma directa y eficiente de comunicarse con los contactos seleccionados en situaciones de emergencia. Al no depender de aplicaciones externas, se garantiza la disponibilidad y la rapidez del envío de mensajes, lo que resulta fundamental en escenarios donde se requiere una respuesta inmediata ante una situación crítica.

## **DESGLOSE DE LA LÓGICA DEL MAINACTIVITY SECUNDARIO - SELECCIÓN DE CONTACTOS DE EMERGENCIA**

De manera similar a la sección anterior, el desglose que se expondrá a continuación está constituido por 2 etapas fundamentales, las cuales permiten a la aplicación cumplir con el propósito de cargar, seleccionar y guardar contactos de emergencia para su posterior uso.

- **ETAPA 1: CREACIÓN DE UN INTENT PARA DEVOLVER DATOS A LA ACTIVIDAD PRINCIPAL**

Para devolver los números de teléfono a la actividad principal en una aplicación Android, primero se crea un Intent para contener los datos. Se agrega una lista de números de teléfono seleccionados utilizando `putStringArrayListExtra` con la clave "contactosSeleccionados". Esto prepara los datos para ser enviados a la actividad principal.

Luego, se usa `setResult(RESULT_OK, intent)` para indicar que la operación se completó exitosamente y que los datos están listos para ser devueltos. Finalmente, se llama a `finish()` para cerrar la actividad actual y retornar el control a la actividad principal, permitiéndole recibir y procesar los números de teléfono devueltos.

- **ETAPA 2: OBTENCIÓN DE NÚMERO DE TELÉFONO DE UN CONTACTO EN ESPECÍFICO**

Para el desarrollo de esta etapa se crea un método llamado `obtenerNumeroDeTelefono`, el mismo se encarga de extraer el número de teléfono de un contacto específico. Primero, accede a la base de datos de contactos del dispositivo mediante `ContentResolver`. Luego, ejecuta una consulta que busca el número de teléfono asociado al nombre del contacto proporcionado.

Si la consulta arroja resultados y al menos un contacto coincide, el método obtiene el primer número de teléfono encontrado y lo asigna a la variable `número`. Finalmente, se devuelve este número de teléfono, esta devolución es crucial, ya que permite que otras partes de la aplicación accedan y utilicen esta información, o `null` si no se encontró ningún contacto que coincida.

### **3.6.7 ANÁLISIS Y COMPROBACIÓN DE LA FUNCIONALIDAD DEL SISTEMA**

Se procederá a la evaluación del sistema en tres escenarios distintos con el fin de obtener una comprensión holística de su rendimiento y adaptabilidad. Dichos escenarios comprenden un entorno ideal, uno real y condiciones extremas, con distancias de evaluación de 2, 4 y 6 metros entre la cámara y el objeto a detectar. Esta diversificación en la evaluación garantiza no solo la eficacia en situaciones controladas, sino también en contextos desafiantes.

Adicionalmente, se empleará una tabla de evaluación para el análisis de cada imagen, utilizando métricas como verdaderos positivos (TP), falsos positivos (FP), falsos negativos (FN) y verdaderos negativos (TN). Este análisis permitirá calcular indicadores cruciales

como precisión, sensibilidad, especificidad, tasa de error, tiempo de respuesta y consistencia del sistema. La implementación de estas métricas proporcionará una visión precisa sobre la eficiencia del modelo en los diferentes escenarios mencionados.

## DEFINICIÓN DE MÉTRICAS

- **True positive (TP):** Casos en el que el sistema detecto correctamente el arma.
- **False positives (FP):** Casos en los que el sistema detecta incorrectamente un objeto que no es un arma.
- **False negativos(FN):** Casos en las que el sistema no detecta un arma que está presente.
- **True negatives (TN):** Número de veces que el sistema correctamente identifica que no hay arma cuando efectivamente no hay arma en la imagen.

## CÁLCULO DE MÉTRICAS VASADO EN LOS VALORES YA MENCIONADOS

- **Precisión:** indica cuan confiable es el sistema cuando detecta un arma.

$$Precision = \frac{TP}{TP + FP}$$

- **Sensibilidad:** indica cuan bien el sistema detecta todas las armas.

$$Sensibilidad = \frac{TP}{TP + FN}$$

- **Especificidad:** indica cuan bien el sistema evita falsas alarmas.

$$Especificidad: \frac{TN}{TN + FP}$$

- **Tasa de error**

$$Tasa de error: \frac{FP + FN}{TOTAL DE CASOS}$$

## ESCENARIOS DE PRUEBAS

### A. AMBIENTE IDEAL

**Objetivo:** Verificar la funcionalidad básica sin interferencias.

**Cantidad de pruebas:** 10 destinadas par armas de fuego cortas y 10 para armas blancas (cuchillos).

**Condiciones:**

- Iluminación constante
- Fondo uniforme
- Posición y orientación variables

**DEMOSTRACIÓN DEL ENTRENAMIENTO EN CONDICIONES IDEALES  
ARMA DE FUEGO CORTA**

A continuación, se presentan 5 muestras representativas de las 10 pruebas realizadas en el escenario ideal. Esta selección se basa en la importancia de destacar la variabilidad en la posición y orientación del arma, lo que permite una comprensión más clara y concisa del rendimiento del sistema.



**Figura 24 A\_1**



**Figura 25 A\_2**



**Figura 27 A\_3**



**Figura 26 A\_4**



Figura 28 A\_1



Figura 29 A\_5

### Tabla de evaluación

PRUEBA	NOMBRE DE LA IMAGEN	TIPO DE ARMA	POSICIÓN	CONDICIÓN	DISTANCIA DEL OBJETO A LA CÁMARA	T P	F P	F N	T N
1	A_1	Arma de fuego corta	horizontal	Buena iluminación, fondo simple	50 cm	1	0	0	0
2	A_2	Arma de fuego corta	horizontal	Buena iluminación, fondo simple	50 cm	1	0	0	0
3	A_3	Arma de fuego corta	horizontal	Buena iluminación, fondo simple	50 cm	1	0	0	0
4	A_4	Arma de fuego corta	recostado	Buena iluminación, fondo simple	50 cm	1	0	0	0
5	A_5	Arma de fuego corta	recostado	Buena iluminación, fondo simple	50 cm	1	0	0	0
6	A_6	Arma de fuego corta	verticales	Buena iluminación, fondo simple	50 cm	1	0	0	0
7	A_7	Arma de fuego corta	verticales	Buena iluminación, fondo simple	50 cm	1	0	0	0

8	A_8	Arma de fuego corta	Diagonal	Buena iluminación, fondo simple	50 cm	1	0	0	0
9	A_9	Arma de fuego corta	Diagonal	Buena iluminación, fondo simple	50 cm	1	0	0	0
10	A_10	Arma de fuego corta	Frontal	Buena iluminación, fondo simple	50 cm	1	0	0	0
TOTAL						10	0	0	0

**Tabla 6** Tabla de evaluación en un ambiente ideal para un arma de fuego

### Análisis

El sistema de detección de armas de fuego ha mostrado un desempeño perfecto en las condiciones ideales analizadas, detectando con éxito todas las armas en diferentes posiciones y con buena iluminación y un fondo simple. No se observaron falsos positivos ni falsos negativos, lo que sugiere una precisión del 100% en este entorno controlado.

### DEMOSTRACIÓN DEL ENTRENAMIENTO EN CONDICIONES IDEALES ARMA BLANCA (CUCHILLO)

A continuación, se presentan 5 muestras representativas de las 10 pruebas realizadas en el escenario ideal con cuchillos. Esta selección destaca la variabilidad en la posición y orientación del cuchillo, permitiendo una comprensión más clara y concisa del rendimiento del sistema.



**Figura 31** C\_1



**Figura 30** C\_2





**Figura 33 C\_3**



**Figura 32 C\_4**



**Figura 34 C\_5**

### Análisis

Al igual que en las pruebas de detección de armas de fuego, estos resultados son alentadores. El sistema de detección de cuchillos ha demostrado un rendimiento excepcional en las condiciones ideales analizadas, logrando identificar con éxito todos los cuchillos en diferentes posiciones y bajo una iluminación adecuada con un fondo simple.

### Tabla de evaluación

PRUEBA	NOMBRE DE LA IMAGEN	TIPO DE ARMA	POSICIÓN	CONDICIÓN	DISTANCIA DEL OBJETO A LA CÁMARA	T P	F P	F N	T N
1	C_1	Arma blanca (cuchillo)	Horizontal	Buena iluminación, fondo simple	50 cm	1	0	0	0
2	C_2	Arma blanca (cuchillo)	Diagonal	Buena iluminación, fondo simple	50 cm	1	0	0	0

3	C_3	Arma blanca (cuchillo)	Frontal	Buena iluminación, fondo simple	50 cm	1	0	0	0
4	C_4	Arma blanca (cuchillo)	Vertical	Buena iluminación, fondo simple	50 cm	1	0	0	0
5	C_5	Arma blanca (cuchillo)	Trasera	Buena iluminación, fondo simple	50 cm	1	0	0	0
6	C_6	Arma blanca (cuchillo)	Vertical	Buena iluminación, fondo simple	50 cm	1	0	0	0
7	C_7	Arma blanca (cuchillo)	Vertical	Buena iluminación, fondo simple	50 cm	1	0	0	0
8	C_8	Arma blanca (cuchillo)	Horizontal	Buena iluminación, fondo simple	50 cm	1	0	0	0
9	C_9	Arma blanca (cuchillo)	Frontal	Buena iluminación, fondo simple	50 cm	1	0	0	0
10	C_10	Arma blanca (cuchillo)	Diagonal	Buena iluminación, fondo simple	50 cm	1	0	0	0
TOTAL						10	0	0	0

**Tabla 7** Tabla de evaluación en un ambiente ideal para un arma blanca(cuchillo)

## B. AMBIENTE REAL

**Objetivo:** Verificar la funcionalidad básica en condiciones reales de uso.

**Cantidad de pruebas:** 25 destinadas para armas de fuego cortas y 25 para armas blancas (cuchillos).

**Condiciones:**

- Variación de la iluminación.

- Fondo complejo y variado.
- Diferentes ángulos y posiciones.
- Distancia variable.

## DEMOSTRACIÓN DEL ENTRENAMIENTO EN CONDICIONES REALES ARMA DE FUEGO CORTA



**Figura 36** *A1\_4M*



**Figura 35** *A2\_4M*



**Figura 40** *A3\_4M*



**Figura 39** *A4\_2M*



**Figura 38** *A4\_2M*



**Figura 37** *A6\_6M*



**Figura 42** A7\_6M



**Figura 41** A8\_6M



**Figura 44** A9\_2M



**Figura 43** A10\_2M

## Análisis

Una vez procesadas las imágenes y sometidas al modelo entrenado de detección de armas de fuego cortas, se obtuvieron los siguientes resultados. Como primer punto importante, se menciona que al someter la imagen A1\_4M al sistema, el modelo entrenado no pudo reconocerla. Esto se debe a que el objeto, al ser de color negro, se confunde perfectamente con el entorno. Este mismo caso se presentó en la imagen A3\_4M.

Cabe resaltar que este fenómeno ocurrió únicamente en aquellas imágenes donde el objeto se ubicaba a 4 metros de distancia de la cámara. De las siete fotos tomadas a esta distancia, el modelo no pudo reconocer el arma en dos de ellas. Este resultado sugiere que la efectividad del modelo disminuye con la distancia, particularmente cuando el objeto y el fondo tienen colores similares, lo que puede causar problemas de contraste y visibilidad. Este hallazgo es crucial para futuras mejoras del modelo, indicando la necesidad de optimizar su capacidad para diferenciar objetos en condiciones de baja visibilidad o con fondos complejos.

Asimismo, se analizaron imágenes donde el arma se encontraba a dos metros de distancia de la cámara. En estos casos, el modelo logró detectar el arma sin ningún problema. Finalmente, se realizó el análisis correspondiente a una distancia de seis metros entre la cámara y el arma. En este análisis, el sistema no logró identificar el arma en un caso, debido a que esta se encontraba en una posición que dificultaba su reconocimiento.

### Tabla de evaluación

PRUEBA	NOMBRE DE LA IMAGEN	TIPO DE ARMA	POSICIÓN	CONDICIÓN	DISTANCIA DEL OBJETO A LA CÁMARA	TP	FP	FN	TN
1	A1_4M	Arma de fuego corta	Horizontal, a la altura de la cadera	Baja iluminación, fondo complejo	4 m	0	0	1	0
2	A2_4M	Arma de fuego corta	Horizontal, a la altura de los hombros	Baja iluminación, fondo complejo	4 m	1	0	0	0
3	A3_4M	Arma de fuego corta	Horizontal, a la altura del pecho	Baja iluminación, fondo complejo	4 m	0	0	1	0
4	A4_2M	Arma de fuego corta	Diagonal, a la altura del pecho	Iluminación natural, fondo complejo	2 m	1	0	0	0
5	A5_2M	Arma de fuego corta	Vertical, a la altura del pecho	Iluminación natural, fondo complejo	2 m	1	0	0	0
6	A6_6M	Arma de fuego corta	Horizontal, a la altura del pecho	Iluminación natural, fondo complejo	6 m	1	0	0	0
7	A7_6M	Arma de fuego corta	Vertical, a la altura del pecho	Iluminación natural, fondo complejo	6 m	1	0	0	0
8	A8_6M	Arma de fuego corta	Vertical, a la altura de la cadera	Iluminación natural, fondo complejo	6 m	0	0	1	0
9	A9_2M	Arma de fuego corta	Diagonal, a la altura de la cadera	Iluminación natural, fondo complejo	2 m	1	0	0	0

<b>10</b>	A10_2M	Arma de fuego corta	Horizontal, a la altura del pecho	Iluminación natural, fondo complejo	2 m	1	0	0	0
<b>11</b>	A11_6M	Arma de fuego corta	Diagonal, a la altura de la cadera	Iluminación natural, fondo complejo	6m	0	0	1	0
<b>12</b>	A12_6M	Arma de fuego corta	Horizontal, a la altura del pecho	Iluminación natural, fondo complejo	6m	1	0	0	0
<b>13</b>	A13_6M	Arma de fuego corta	Diagonal, por encima de la cabeza	Iluminación natural, fondo complejo	6m	1	0	0	0
<b>14</b>	A14_4M	Arma de fuego corta	Horizontal, a la altura del pecho	Baja iluminación, fondo complejo	4m	1	0	0	0
<b>15</b>	A15_4M	Arma de fuego corta	Diagonal a la altura del pecho	Baja iluminación, fondo complejo	4m	1	0	0	0
<b>16</b>	A16_4M	Arma de fuego corta	Diagonal a la altura de los hombros	Iluminación moderada, fondo complejo	4m	0	0	1	0
<b>17</b>	A17_4M	Arma de fuego corta	Horizontal, a la altura de la cadera	Baja iluminación, fondo complejo	4m	1	0	0	0
<b>18</b>	A18_2M	Arma de fuego corta	Horizontal, a la altura de la cadera	Baja iluminación, fondo complejo	2m	0	1	0	0
<b>19</b>	A19_2M	Arma de fuego corta	Vertical, a la altura del pecho	Baja iluminación, fondo complejo	2m	1	0	0	0
<b>20</b>	A20_2M	Arma de fuego corta	Diagonal, a la altura del pecho	Iluminación natural, fondo complejo	2 m	1	0	0	0
<b>21</b>	A21_2M	Arma de fuego corta	Vertical, a la altura del pecho	Iluminación natural, fondo complejo	2 m	1	0	0	0
<b>22</b>	A22_6M	Arma de fuego corta	Horizontal, a la altura del pecho	Iluminación natural, fondo complejo	6 m	1	0	0	0

<b>23</b>	A23_6M	Arma de fuego corta	Vertical, a la altura del pecho	Iluminación natural, fondo complejo	6 m	1	0	0	0
<b>24</b>	A24_6M	Arma de fuego corta	Vertical, a la altura de la cadera	Iluminación natural, fondo complejo	6 m	1	0	0	0
<b>25</b>	A25_2M	Arma de fuego corta	Diagonal, a la altura de la cadera	Iluminación natural, fondo complejo	2 m	1	0	0	0
	<b>TOTAL</b>					19	1	5	0

**Tabla 8** *Tabla de evaluación en un ambiente real para un arma de fuego*

## DEMOSTRACIÓN DEL ENTRENAMIENTO EN CONDICIONES REALES ARMA BLANCA CUCHILLO



Figura 48 C1\_2M



Figura 47 C2\_2M



Figura 50 C3\_2M



Figura 49 C4\_2M



Figura 45 C5\_4M

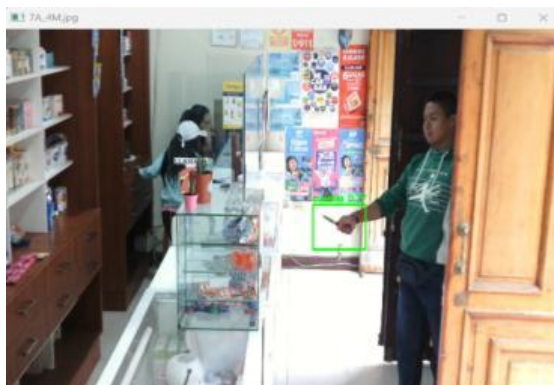


Figura 46 C6\_4M





**Figura 56** *C7\_4M*



**Figura 53** *C8\_4M*



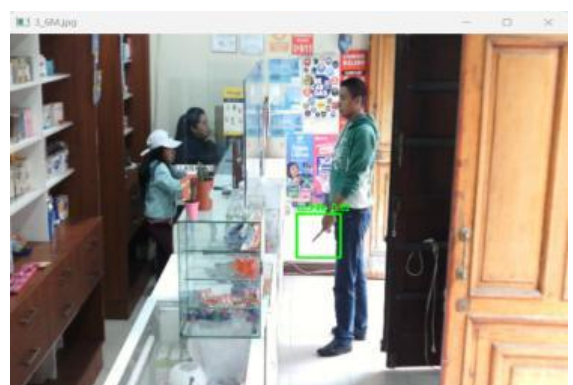
**Figura 55** *C9\_4M*



**Figura 54** *C10\_6M*



**Figura 52** *C11\_6M*



**Figura 51** *C12\_6M*



**Figura 58** C13\_6M



**Figura 57** C14\_6M

## **Análisis**

A continuación, se presenta una muestra con las imágenes más destacadas y en situaciones adversas a las que fue sometido el modelo de detección. Primero, es importante destacar las características del arma blanca (cuchillo) utilizado en las pruebas, que tiene una longitud de 20 cm de largo y un ancho de 4 cm. Este detalle es relevante porque, al ser un objeto pequeño y utilizado en un fondo complejo, aumenta la dificultad de detección para el modelo de entrenamiento.

En las pruebas realizadas con las imágenes C1\_2M y C2\_2M, se evaluó la capacidad del modelo para detectar el arma blanca en presencia de estas complicaciones. A pesar de las dificultades, el modelo de entrenamiento detectó correctamente el objetivo. Además, se llevó a cabo un análisis del arma blanca en un entorno despejado, donde se puede observar claramente el cuchillo, y el modelo también lo reconoció adecuadamente.

Es importante resaltar que hubo casos en los que el sistema no logró reconocer el arma blanca. Esto ocurrió en las imagen C9\_4M, donde el objeto se confundió con el color de la vestimenta de la persona que lo portaba, generándose además una única falsa predicción. También se realizaron pruebas en condiciones de baja iluminación, como se observa en las imágenes C7\_4M y C8\_4M, a pesar de esto el modelo supo detectar de manera correcta el objeto.

Asimismo, se evaluó el sistema a una distancia de seis metros, y el resultado fue favorable, ya que el modelo detectó el objeto en todas las ocasiones y posiciones en las que se colocó, gracias a que esta área estaba bien iluminada.

## **Tabla de evaluación**

<b>PRUEBA</b>	<b>NOMBRE DE LA IMAGEN</b>	<b>TIPO DE ARMA</b>	<b>POSICIÓN</b>	<b>CONDICIÓN</b>	<b>DISTANCIA DEL OBJETO A LA CÁMARA</b>	<b>T P</b>	<b>F P</b>	<b>F N</b>	<b>T N</b>
<b>1</b>	C1_2M	Arma blanca(cuchillo )	diagonal, a la altura de la cadera	Iluminación natural, fondo complejo, parcialmente oculto	2 m	1	0	0	0
<b>2</b>	C2_2M	Arma blanca(cuchillo )	Diagonal, a la altura de los hombros	Iluminación natural, fondo complejo, parcialmente oculto	2 m	1	0	0	0
<b>3</b>	C3_2M	Arma blanca(cuchillo )	Horizontal, a la altura de la cadera	Iluminación natural, fondo complejo	2 m	1	0	0	0
<b>4</b>	C4_2M	Arma blanca(cuchillo )	Horizontal, a la altura de la cadera	Iluminación natural, fondo complejo	2 m	1	0	0	0
<b>5</b>	C5_4M	Arma blanca(cuchillo )	diagonal, a la altura de la cadera	Iluminación baja, fondo complejo	4 m	0	0	1	0
<b>6</b>	C6_4M	Arma blanca(cuchillo )	diagonal, a la altura de la cadera	Iluminación baja, fondo complejo	4 m	1	0	0	0
<b>7</b>	C7_4M	Arma blanca(cuchillo )	Por encima de la cabeza	Iluminación baja, fondo complejo	4 m	1	0	0	0
<b>8</b>	C8_4M	Arma blanca(cuchillo )	Horizontal, por debajo de la cadera	Iluminación baja, fondo complejo	4 m	1	0	0	0
<b>9</b>	C9_4M	Arma blanca(cuchillo )	Diagonal, a la altura de la cadera	Iluminación baja, fondo complejo	4 m	0	0	1	0
<b>10</b>	C10_6M	Arma blanca(cuchillo )	Horizontal, por debajo de la cadera	Iluminación natural, fondo complejo	6 m	1	0	0	0
<b>11</b>	C11_6M	Arma blanca(cuchillo )	Diagonal, a la altura de la cadera	Iluminación natural, fondo complejo	6m	1	0	0	0
<b>12</b>	C12_6M	Arma blanca(cuchillo )	Horizontal, por debajo	Iluminación natural, fondo complejo	6m	1	0	0	0

			de la cintura						
<b>13</b>	C13_6M	Arma blanca(cuchillo )	Horizontal, a la altura de la cabeza	Iluminación natural, fondo complejo	6m	1	0	0	0
<b>14</b>	C14_6M	Arma blanca(cuchillo )	diagonal, por encima de la cabeza	Iluminación natural, fondo complejo	4m	1	0	0	0
<b>15</b>	C15_2M	Arma blanca(cuchillo )	diagonal, a la altura de la cadera	Iluminación baja, fondo complejo	4 m	0	0	1	0
<b>16</b>	C16_2M	Arma blanca(cuchillo )	Por encima de la cabeza	Iluminación baja, fondo complejo	4 m	0	0	1	0
<b>17</b>	C17_2M	Arma blanca(cuchillo )	Horizontal, por debajo de la cadera	Iluminación baja, fondo complejo	4 m	1	0	0	0
<b>18</b>	C18_2M	Arma blanca(cuchillo )	Horizontal, a la altura de la cadera	Baja iluminación, fondo complejo	2m	1	0	0	0
<b>19</b>	C19_4M	Arma blanca(cuchillo )	Vertical, a la altura del pecho	Baja iluminación, fondo complejo	2m	1	0	0	0
<b>20</b>	C20_4M	Arma blanca(cuchillo )	Diagonal, a la altura del pecho	Iluminación natural, fondo complejo	2 m	1	0	0	0
<b>21</b>	C21_4M	Arma blanca(cuchillo )	Vertical, a la altura del pecho	Iluminación natural, fondo complejo	2 m	1	0	0	0
<b>22</b>	C22_4M	Arma blanca(cuchillo )	Horizontal, a la altura del pecho	Iluminación natural, fondo complejo	6 m	1	0	0	0
<b>23</b>	C23_6M	Arma blanca(cuchillo )	Vertical, a la altura del pecho	Iluminación natural, fondo complejo	6 m	0	0	1	0
<b>24</b>	C24_6M	Arma blanca(cuchillo )	Vertical, a la altura de la cadera	Iluminación natural, fondo complejo	6 m	1	0	0	0
<b>25</b>	C25_6M	Arma blanca(cuchillo )	Diagonal, a la altura de la cadera	Iluminación natural, fondo complejo	2 m	1	0	0	0

	TOTAL	20	0	5	0
--	-------	----	---	---	---

**Tabla 9** *Tabla de evaluación en un ambiente real para un arma blanca*

### C. PRUEBAS DE ROBUSTES Y ESTRÉS

**Objetivo:** Evaluar la capacidad del sistema bajo condiciones extremas y situaciones no ideales.

**Cantidad de pruebas:** 10 destinadas par armas de fuego cortas y 10 para armas blancas (cuchillos).

**Condiciones:**

- Muy baja o muy alta iluminación
- Interferencias intencionales (objetos similares)
- Objetos parciamente ocultos

### DEMOSTRACIÓN DEL ENTRENAMIENTO EN CONDICIONES DE ROBUSTES Y ESTRÉS ARMA DE FUEGO



**Figura 60** *A1\_ROBUSTES*



**Figura 59** *A2\_ROBUSTES*



**Figura 62** *A3\_ROBUSTES*



**Figura 61** *A4\_ROBUSTES*



**Figura 63 A5\_ROBUSTES**

### Análisis

En esta fase, se utilizaron objetos que se asemejan a armas para evaluar el modelo. Por ejemplo, en la imagen A2\_ROBUSTES, se utilizó una pistola de agua de color rosa, y en la imagen A4\_ROBUSTES, se empleó una grapadora con la intención de confundir al sistema. En ambos casos, el modelo dedujo correctamente que no eran armas de fuego, lo cual es un resultado muy satisfactorio.

Durante el análisis, se observó que, en la mayoría de los casos, el sistema no logró detectar armas de fuego. Sin embargo, hubo una excepción en la imagen A1\_ROBUSTES, donde el sistema detectó una posible arma debido a la simulación de sostener una pistola. Esta detección se debe a que en las imágenes de entrenamiento se incluyeron ejemplos de personas sosteniendo armas de manera similar.

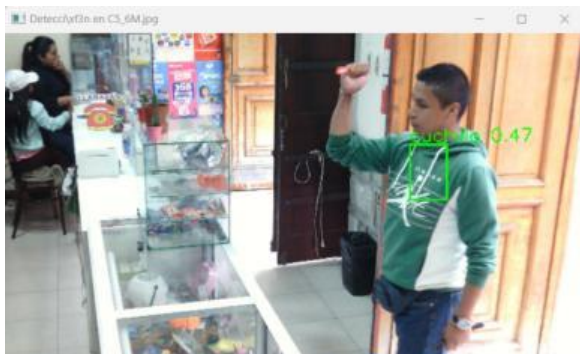
### Tabla de evaluación

PRUEBA	NOMBRE DE LA IMAGEN	TIPO DE OBJETO	POSICIÓN	CONDICIÓN	DISTANCIA DEL OBJETO A LA CÁMARA	T P	F P	F N	T N
1	A1_ROBUSTES	ninguna	ninguna	Buena iluminación, fondo complejo, simulación de sostener un arma	2 m	0	1	0	0
2	A2_ROBUSTES	Arma de juguete	Diagonal a la altura de la cabeza	Buena iluminación, fondo complejo	2 m	0	0	0	1

3	A3_ROBUSTES	Arma de juguete	Diagonal por encima de la cabeza	Buena iluminación, fondo complejo	2 m	0	0	0	1
4	A4_ROBUSTES	Grapadora	Horizontal a la altura de la cadera	Buena iluminación, fondo complejo	2 cm	0	0	0	1
5	A5_ROBUSTES	marcador	Diagonal por encima de la cabeza	Buena iluminación, fondo complejo	6 m	0	0	0	1
6	A6_ROBUSTES	marcador	Horizontal a la altura de la cadera	Buena iluminación, fondo complejo	6 m	0	0	0	1
7	A7_ROBUSTES	marcador	Horizontal a la altura de la cadera	Buena iluminación, fondo complejo	6 m	0	0	0	1
8	A8_ROBUSTES	Grapadora	Horizontal a la altura de la cadera	Buena iluminación, fondo complejo	6 cm	0	0	0	1
9	A9_ROBUSTES	Grapadora	Horizontal a la altura de la cadera	Buena iluminación, fondo complejo	2 cm	0	0	0	1
10	A10_ROBUSTES	Grapadora	Horizontal a la altura de la cadera	Buena iluminación, fondo complejo	2 cm	0	0	0	1
TOTAL						0	1	0	9

**Tabla 10** *Tabla de Evaluación en un Entorno de Robustez y Estrés (arma de fuego)*

## DEMOSTRACIÓN DEL ENTRENAMIENTO EN CONDICIONES DE ROBUSTES Y ESTRÉS ARMAS BLANCAS CUCHILLO.



**Figura 64 C1\_ROBUSTES**



**Figura 65 C2\_ROBUSTES**



**Figura 68 C3\_ROBUSTES**



**Figura 67 C4\_ROBUSTES**



**Figura 66 C5\_ROBUSTES**



## Análisis

Una vez sometidas las diez imágenes de robustez al modelo, cinco de ellas destacaron, arrojando resultados significativos: en la imagen c1\_robustes, el modelo detecta incorrectamente una posible arma blanca, debido a la notable similitud en color y forma con un cuchillo real. De manera similar, en la imagen C2\_ROBUSTES, se obtuvo una identificación errónea de un arma blanca, ya que el modelo no pudo diferenciar un cuchillo de juguete de uno real, debido a la forma de empuñar el cuchillo de juguete que es similar a las imágenes de entrenamiento utilizadas. Esta confusión denota la necesidad de mejorar la discriminación del modelo entre objetos reales y juguetes, especialmente cuando comparten características visuales clave.

Asimismo, los resultados obtenidos en las imágenes C3\_ROBUSTES, C4\_ROBUSTES y C5\_ROBUSTES muestran que el algoritmo logró identificar con precisión los objetos empuñados y descartar favorablemente que no se trataban de armas blancas(cuchillo). Este desempeño satisfactorio del algoritmo en estas tres imágenes destaca su capacidad para reconocer con éxito elementos no amenazantes y reducir falsos positivos. Estos resultados demuestran tanto los aciertos como las áreas de mejora para el modelo, subrayando la importancia de una continua optimización y ajuste fino en su entrenamiento.

### Tabla de evaluación

PRUEBA	NOMBRE DE LA IMAGEN	TIPO DE OBJETO	POSICIÓN	CONDICIÓN	DISTANCIA DEL OBJETO A LA CÁMARA	T P	F P	F N	T N
1	C1_ROBUSTES	Mango de cuchillo de juguete	Horizontal por encima de la cabeza	Buena iluminación, fondo complejo	2 m	0	1	0	0
2	C2_ROBUSTES	Cuchillo de juguete	Horizontal ala altura de la cadera	Buena iluminación, fondo complejo	2 m	0	1	0	0
3	C3_ROBUSTES	Peinilla	Horizontal a la altura del pecho	Buena iluminación, fondo complejo	6 m	0	0	0	1

4	C4_ROBUSTES	Tijera	Horizontal a la altura del pecho	Buena iluminación, fondo complejo	2 m	0	0	0	1
5	C5_ROBUSTES	Peinilla	Horizontal a la altura del pecho	Buena iluminación, fondo complejo	2 m	0	0	0	1
6	C6_ROBUSTES	peinilla	Horizontal a la altura de la cadera	Buena iluminación, fondo complejo	4 m	0	0	0	1
7	C7_ROBUSTES	Cuchillo de juguete	Horizontal a la altura de la cadera	Buena iluminación, fondo complejo	4 m	0	0	0	1
8	C8_ROBUSTES	Tijera	Diagonal, a la altura del pecho	Buena iluminación, fondo complejo	4 m	0	0	0	1
9	C9_ROBUSTES	Tijera	Horizontal a la altura del pecho	Buena iluminación, fondo complejo	6 m	0	0	0	1
10	C10_ROBUSTES	Peinilla	Diagonal, a la altura de la cadera	Buena iluminación, fondo complejo	4 m	0	0	0	1
TOTAL						0	2	0	8

**Tabla 11** *Tabla de Evaluación en un Entorno de Robustez y Estrés (arma blanca)*

## **CAPITULO IV RESULTADOS**

### **4. RESULTADOS DE LAS PRUEBAS EN AMBIENTES IDEAL, REAL Y DE ROBUSTEZ**

A continuación, se presentarán los resultados derivados de la aplicación de las fórmulas correspondientes a cada métrica y los datos recolectados en las tablas de evaluación realizadas. Este análisis tiene como objetivo proporcionar una comprensión más profunda del comportamiento del sistema frente a los diversos escenarios evaluados.

#### **RESULTADOS DE LA TABLA DE EVALUACIÓN REALIZADA EN UN AMBIENTE IDEAL.**

<b>métrica</b>	<b>Ambiente ideal</b>
<b>Precisión</b>	100%
<b>Sensibilidad</b>	100%
<b>Especificidad</b>	N/A
<b>Tasa de error</b>	0%
<b>Tiempo de respuesta</b>	100 MILISEGUNDOS (MS)

**Tabla 12** *Tabla De Evaluación Realizada En Un Ambiente Ideal.*

Los resultados obtenidos en las pruebas realizadas en el ambiente ideal revelan un rendimiento excepcional por parte del sistema de detección de armas de fuego cortas como de armas blancas. Estos hallazgos ofrecen una visión valiosa sobre la eficacia y la confiabilidad del sistema en condiciones óptimas.

- **PRECISIÓN Y SENSIBILIDAD OPTIMAS**

Los resultados de las pruebas destacan una precisión y sensibilidad sobresalientes del sistema de detección de armas. En cada instancia evaluada, el sistema logró identificar con precisión la presencia de armas, sin cometer errores de falsos positivos o falsos negativos. Esta precisión y sensibilidad del 100% reflejan la capacidad del sistema para realizar detecciones precisas y confiables en condiciones ideales.

- **AUSENCIA DE ERRORES SIGNIFICATIVOS**

Los datos recopilados revelan una ausencia total de errores significativos en las pruebas realizadas en el ambiente ideal. Con una tasa de error del 0%, el sistema demostró una capacidad excepcional para evitar errores de identificación, lo que subraya su fiabilidad y eficacia en la detección de armas.

- **CONSIDERACIÓN SOBRE LA ESPECIFICIDAD**

Aunque los resultados demuestran una precisión perfecta en la detección de armas, la especificidad no se puede calcular debido a la falta de datos sobre verdaderos negativos en las pruebas. Sin embargo, la ausencia de falsos positivos sugiere un alto nivel de especificidad en la identificación de armas por parte del sistema.

## RESULTADOS DE LA TABLA DE EVALUACIÓN REALIZADA EN UN AMBIENTE REAL UTILIZANDO UN ARMA DE FUEGO CORTA.

Métrica	Ambiente real
Precisión	95%
Sensibilidad	83.33%
Especificidad	100%
Tasa de error	16.67%
Tiempo de respuesta	100 MILISEGUNDOS (MS)

**Tabla 13** *Tabla De Evaluación Realizada En Un Ambiente*

- **PRECISIÓN**

La precisión del sistema de detección de armas es del 95%, lo que indica que de todas las clasificaciones realizadas, el 95% son verdaderos positivos, es decir, identificaciones correctas de armas. Esta métrica es crucial para evaluar la confiabilidad del sistema en la identificación precisa de objetos de interés. Una alta precisión demuestra la capacidad del sistema para minimizar los falsos positivos y maximizar la exactitud en la detección de armas, lo que resulta fundamental en aplicaciones para seguridad.

- **SENSIBILIDAD**

La sensibilidad del sistema, evaluada en un 83.33%, representa la capacidad del sistema para detectar correctamente el 83.33% de todas las instancias reales de armas presentes en las imágenes procesadas. Esta métrica es esencial para evaluar la capacidad de detección del sistema en la identificación de todas las armas presentes en el entorno. Aunque esta sensibilidad es aceptable, se indica que aún hay margen para mejorar la capacidad del sistema para detectar todas las armas presentes en el campo visual.

- **ESPECIFICIDAD**

La especificidad del sistema, que alcanza el 100%, indica que tiene una capacidad notable para identificar correctamente las instancias que no son armas, evitando así los falsos positivos. Esto significa que el sistema tiene una habilidad destacada para discernir entre

objetos inofensivos y armas reales, lo que resulta fundamental en entornos de seguridad donde la precisión y la eficacia son prioritarias.

- **TASA DE ERROR**

La tasa de error del sistema se sitúa en un 16.67%, lo que representa el porcentaje de predicciones incorrectas en relación con el total de predicciones realizadas. Aunque este valor indica una cierta proporción de predicciones incorrectas, se considera moderado y manejable. Sin embargo, es recomendable seguir trabajando para reducir aún más esta tasa, con el fin de mejorar la precisión general y la confiabilidad del sistema en una variedad de escenarios y condiciones.

**RESULTADOS DE LA TABLA DE EVALUACIÓN REALIZADA EN UN AMBIENTE REAL UTILIZANDO UN ARMA BLANCA(CUCHILLO).**

Métrica	Ambiente real
Precisión	100%
Sensibilidad	80.00%
Especificidad	100%
Tasa de error	20.00%
Tiempo de respuesta	100 MILISEGUNDOS (MS)

**Tabla 14** *Tabla De Evaluación Realizada En Un Ambiente Real Utilizando Un Arma Blanca*

- **PRECISIÓN**

Los resultados muestran una precisión del 100%, lo que indica que el sistema es altamente preciso en la detección de armas blancas en las condiciones evaluadas. Esto significa que todas las predicciones positivas realizadas por el sistema fueron correctas, sin ningún falso positivo, lo cual es crucial para aplicaciones donde la precisión es vital para la seguridad y la efectividad del sistema.

- **SENSIBILIDAD**

Con una sensibilidad del 80%, el sistema logra detectar la mayoría de las armas blancas presentes en las imágenes. Esto implica que, aunque el sistema es bastante efectivo, hay un 20% de las instancias de armas blancas que no fueron detectadas (falsos negativos).

- **ESPECIFICIDAD**

La especificidad del 100% indica que el sistema es capaz de identificar correctamente la ausencia de armas blancas en las imágenes, minimizando los falsos positivos.

- **TASA DE ERROR**

La tasa de error del 20% indica que hay margen de mejora, pero esta tasa se encuentra dentro de un rango aceptable para el tipo de aplicación y las condiciones de prueba.

**RESULTADOS DE LA TABLA DE EVALUACIÓN REALIZADA EN UN AMBIENTE DE ROBUSTES Y ESTRÉS UTILIZANDO UN ARMA UN ARMA DE FUEGO CORTA.**

Métrica	Ambiente real
Precisión	0%
Sensibilidad	N/A
Especificidad	90%
Tasa de error	10.00%
Tiempo de respuesta	100 MILISEGUNDOS (MS)

**Tabla 15** *Tabla De Evaluación Realizada En Un Ambiente De Robustes Y Estrés Utilizando Un Arma Un Arma De Fuego Corta*

- **PRECISIÓN**

La precisión del modelo es del 0.00%, lo que indica que no logró identificar correctamente ninguna instancia de arma falsa en el conjunto de imágenes evaluadas. Este resultado sugiere que el modelo cometió errores al clasificar las imágenes, identificando incorrectamente objetos que no son armas falsas como armas.

- **SENSIBILIDAD NO APLICABLE**

La sensibilidad no está aplicable en este contexto. Esto puede deberse a que no se evaluaron los casos positivos en el conjunto de datos, lo que limita nuestra comprensión de la capacidad del modelo para identificar correctamente las instancias positivas.

- **ESPECIFICIDAD**

La especificidad del 90% indica que el modelo puede identificar correctamente el 90% de las instancias negativas. Esto significa que el modelo tiene una capacidad sólida para evitar los falsos negativos, es decir, las instancias negativas que son incorrectamente clasificadas como positivas.

- **TASA DE ERROR**

Esta métrica indica que el modelo ha logrado una precisión del 90% en la clasificación de las instancias en el conjunto de datos evaluado. Un bajo porcentaje de error implica una capacidad satisfactoria del modelo para realizar predicciones precisas y consistentes.

**RESULTADOS DE LA TABLA DE EVALUACIÓN REALIZADA EN UN AMBIENTE DE ROBUSTES Y ESTRÉS UTILIZANDO UN ARMA BLANCA CUCHILLO.**

Métrica	Ambiente real
Precisión	0%
Sensibilidad	N/A
Especificidad	80%
Tasa de error	20.00%
Tiempo de respuesta	100 MILISEGUNDOS (MS)

**Tabla 16** *Tabla De Evaluación Realizada En Un Ambiente De Robustes Y Estrés Utilizando Un Arma Blanca Cuchillo*

- **PRECISIÓN**

La precisión del modelo es del 0.00%, lo que indica que no logró identificar correctamente ninguna instancia de arma blanca en el conjunto de imágenes evaluadas. Esto sugiere que el modelo cometió errores al clasificar las imágenes, identificando incorrectamente objetos que no son armas como armas.

- **SENSIBILIDAD NO APLICABLE**

Dado que no se detectaron verdaderos positivos en el conjunto de datos, la sensibilidad no se puede calcular y se muestra como no aplicable (N/A). Esto sugiere que el modelo no pudo identificar ninguna instancia de arma blanca entre las imágenes evaluadas.

- **ESPECIFICIDAD**

Aunque el modelo no logró detectar correctamente las armas blancas, la especificidad del 80.00% indica que pudo identificar correctamente la ausencia de armas en la mayoría de las imágenes. Esto significa que el modelo tiende a evitar los falsos positivos, es decir, clasificar incorrectamente objetos no peligrosos como armas blancas.

- **TASA DE ERROR**

La tasa de error del 20.00% indica que el modelo cometió errores en la clasificación de las imágenes.

#### **4.1 ANÁLISIS DE RESULTADOS ESTADÍSTICOS**

El proceso estadístico se llevó a cabo basándose en un estudio detallado de 100 datos, los cuales fueron sometidos a rigurosas pruebas de detección, teniendo en cuenta tanto el conjunto de datos que conforma el modelo de entrenamiento como la distancia del objeto a la cámara. Este análisis meticuloso se realizó con el objetivo de determinar la precisión y sensibilidad del sistema.

Para realizar este análisis exhaustivo de los datos, se utilizó el programa SPSS, empleando técnicas de tablas cruzadas y análisis de frecuencias. Los parámetros evaluados incluyen TP (verdaderos positivos), FP (falsos positivos), FN (falsos negativos) y TN (verdaderos negativos). Estos indicadores clave de rendimiento nos permitieron evaluar con detalle las imágenes y proporcionar una medida precisa de la exactitud y sensibilidad del sistema.

#### **ESTRUCTURA DE LAS TABLAS DE EVALUACIÓN**

A continuación, se presenta un análisis estructurado del sistema utilizado para la evaluación de detección de objetos a diferentes distancias utilizando modelos entrenados con diferentes bases de datos:

Modelo inicial con base de datos de 500 imágenes:



- El sistema fue evaluado inicialmente utilizando un modelo entrenado con una base de datos compuesta por 500 imágenes.
- Estas imágenes fueron capturadas a distancias específicas de 2, 4 y 6 metros entre el objeto a detectar y la cámara.

Modelo con base de datos ampliada a 700 imágenes:

- Posteriormente, se realizó un análisis utilizando un modelo entrenado con una base de datos expandida a 700 imágenes.
- Al igual que en el caso anterior, las imágenes fueron tomadas a distancias de 2, 4 y 6 metros respecto al objeto y la cámara.

Modelo final con base de datos de 3000 imágenes:

- Finalmente, se procedió a evaluar el sistema utilizando un modelo entrenado con una base de datos adicional que consta de 3000 imágenes.
- Estas imágenes también fueron distribuidas en las mismas distancias de 2, 4 y 6 metros entre el objeto y la cámara.

Este enfoque secuencial permitió una evaluación progresiva del sistema de detección de objetos, escalando tanto el tamaño de la base de datos como la variedad de distancias evaluadas. Esta metodología proporcionó una comprensión más completa del rendimiento del modelo en diversas condiciones de captura, contribuyendo así a su refinamiento y optimización para aplicaciones prácticas.

### TABLA DE EVALUACIÓN-ARMA DE FUEGO

ID DE PRUEBA	CANTIDAD DE IMÁGENES QUE COMPONE LA BASE DE DATOS	DISTANCIA (METROS)	TP	FP	FN	TN	PRESIÓN	SENSIBILIDAD
1	500	2	1	0	0	0	63.64%	58.33%
2	500	2	0	1	1	0		
3	500	2	1	0	0	0		
4	500	2	1	0	0	0		
5	500	2	1	0	0	0		

6	500	2	0	1	1	0			
7	500	2	1	0	0	0			
8	500	2	0	1	1	0			
9	500	2	1	0	0	0			
10	500	2	1	0	0	0			
11	500	2	0	1	1	0			
TOTAL			7	4	5	0			
ID DE PRUEBA	CANTIDAD DE IMÁGENES QUE COMPONE LA BASE DE DATOS	DISTANCIA (METROS)	TP	FP	FN	TN	PRECISIÓN	SENSIBILIDAD	
12	500	4	0	0	1	0	57.14%	33.33%	
13	500	4	0	1	0	0			
14	500	4	0	0	1	0			
15	500	4	1	0	0	0			
16	500	4	1	0	0	0			
17	500	4	0	0	1	0			
18	500	4	0	0	1	0			
19	500	4	0	0	1	0			
20	500	4	1	0	0	0			
21	500	4	0	0	1	0			
22	500	4	0	1	1	0			
TOTAL			4	3	8	0			
ID DE PRUEBA	CANTIDAD DE IMÁGENES QUE COMPONE LA BASE DE DATOS	DISTANCIA (METROS)	TP	FP	FN	TN	PRECISIÓN	SENSIBILIDAD	
23	500	6	0	1	1	0	50%	36.36%	
24	500	6	0	1	1	0			

25	500	6	0	0	1	0		
26	500	6	1	0	0	0		
27	500	6	0	1	1	0		
28	500	6	0	0	1	0		
29	500	6	1	0	0	0		
30	500	6	1	0	0	0		
31	500	6	0	1	1	0		
32	500	6	1	0	0	0		
33	500	6	0	0	1	0		
TOTAL			4	4	7	0		
ID DE PRUEBA	CANTIDAD DE IMÁGENES QUE COMPONE LA BASE DE DATOS	DISTANCIA (METROS)	TP	FP	FN	TN	PRECISIÓN	SENSIBILIDAD
34	700	2	1	0	0	0	77.78%	58.33%
35	700	2	1	0	0	0		
36	700	2	1	0	0	0		
37	700	2	1	0	0	0		
38	700	2	1	0	0	0		
39	700	2	0	1	1	0		
40	700	2	1	0	0	0		
41	700	2	1	0	0	0		
42	700	2	0	0	1	0		
43	700	2	0	1	1	0		
44	700	2	0	0	1	0		
TOTAL			7	2	5	0		
ID DE PRUEBA	CANTIDAD DE IMÁGENES QUE COMPONE	DISTANCIA (METROS)	TP	FP	FN	TN	PRECISIÓN	SENSIBILIDAD

	LA BASE DE DATOS									
45	700	4	1	0	0	0	70%	53.85%		
46	700	4	1	0	0	0				
47	700	4	0	1	1	0				
48	700	4	0	1	1	0				
49	700	4	0	0	1	0				
50	700	4	0	1	1	0				
51	700	4	1	0	1	0				
52	700	4	1	0	0	0				
53	700	4	1	0	0	0				
54	700	4	1	0	0	0				
55	700	4	1	0	0	0				
TOTAL			7	3	6	0				
ID DE PRUEBA	CANTIDAD DE IMÁGENES QUE COMPONE LA BASE DE DATOS	DISTANCIA (METROS)	TP	FP	FN	TN	PRECISIÓN	SENSIBILIDAD		
56	700	6	1	0	0	0	72.73%	72.73%		
57	700	6	0	1	1	0				
58	700	6	1	0	0	0				
59	700	6	1	0	0	0				
60	700	6	1	0	0	0				
61	700	6	1	0	0	0				
62	700	6	1	0	0	0				
63	700	6	0	1	1	0				
64	700	6	1	0	0	0				
65	700	6	1	0	0	0				
66	700	6	0	1	1	0				
TOTAL			8	3	3	0				
ID DE PRUEBA	CANTIDAD DE	DISTANCIA (METROS)	TP	FP	FN	TN	PRECISIÓN	SENSIBILIDAD		

	IMÁGENES QUE COMPONE LA BASE DE DATOS								
67	3000	2	1	0	0	0	100%	90.91%	
68	3000	2	1	0	0	0			
69	3000	2	1	0	0	0			
70	3000	2	1	0	0	0			
71	3000	2	1	0	0	0			
72	3000	2	1	0	0	0			
73	3000	2	1	0	0	0			
74	3000	2	0	0	1	0			
75	3000	2	1	0	0	0			
76	3000	2	1	0	0	0			
77	3000	2	1	0	0	0			
TOTAL			10	0	1	0			
ID DE PRUEBA	CANTIDAD DE IMÁGENES QUE COMPONE LA BASE DE DATOS	DISTANCIA (METROS)	TP	FP	FN	TN	PRECISIÓN	SENSIBILIDAD	
78	3000	4	0	1	0	0	90.91%	90.91%	
79	3000	4	1	0	0	0			
80	3000	4	1	0	0	0			
81	3000	4	1	0	0	0			
82	3000	4	1	0	0	0			
83	3000	4	1	0	0	0			
84	3000	4	0	0	1	0			
85	3000	4	1	0	0	0			
86	3000	4	1	0	0	0			
87	3000	4	1	0	0	0			
88	3000	4	1	0	0	0			
TOTAL			10	1	1	0			

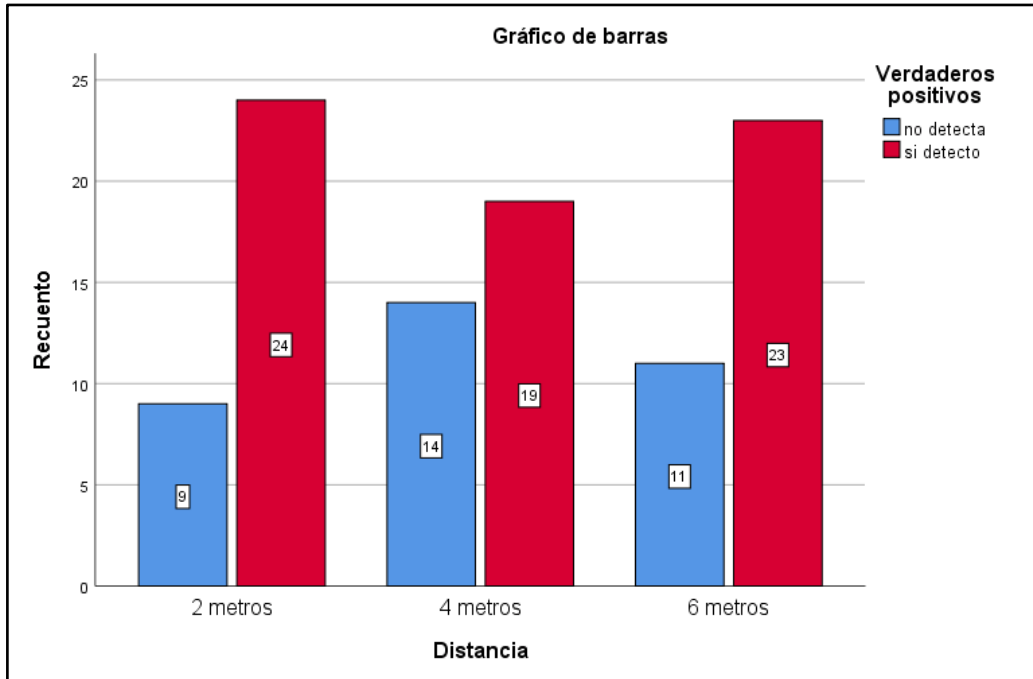
ID DE PRUEBA	CANTIDAD DE IMÁGENES QUE COMPONE LA BASE DE DATOS	DISTANCIA (METROS)	TP	FP	FN	TN	PRECISIÓN	SENSIBILIDAD
89	3000	6	1	0	0	0	91.67%	91.67%
90	3000	6	1	0	0	0		
91	3000	6	1	0	0	0		
92	3000	6	0	1	1	0		
93	3000	6	1	0	0	0		
94	3000	6	1	0	0	0		
95	3000	6	1	0	0	0		
96	3000	6	1	0	0	0		
97	3000	6	1	0	0	0		
98	3000	6	1	0	0	0		
99	3000	6	1	0	0	0		
100	3000	6	1	0	0	0		
TOTAL			11	1	1	0		

**Figura 69** *Tabla De Evaluación-Arma De Fuego*

## ANÁLISIS DESCRIPTIVO DE LAS VARIABLES

<b>Tabla cruzada Distancia*Verdaderos positivos</b>				
Recuento				
		Verdaderos positivos		Total
		no detecta	si detecto	
Distancia	2 metros	9	24	33
	4 metros	14	19	33
	6 metros	11	23	34
Total		34	66	100

**Figura 70** *Tabla cruzada Distancia\*Verdaderos positivos*



**Figura 71** Gráfico De Barras

## ANÁLISIS

La tabla cruzada muestra el recuento de verdaderos positivos detectados a diferentes distancias (2, 4 y 6 metros). A 2 metros, se detectaron 24 verdaderos positivos de un total de 33 casos. Esta distancia demuestra una alta efectividad en la detección, lo que la hace ideal para escenarios donde se requiere precisión y confiabilidad en la detección cercana, como en espacios reducidos donde la proximidad es crítica.

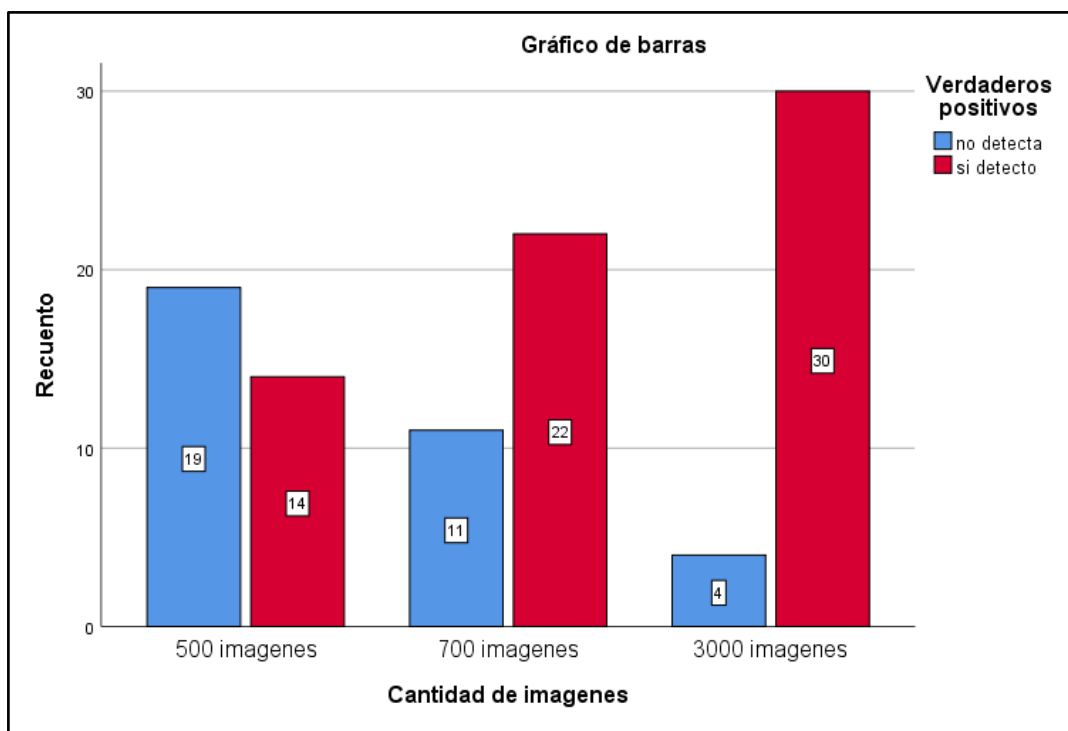
A 4 metros, se detectaron 19 verdaderos positivos de un total de 33 casos. Aunque la efectividad es menor en comparación con 2 metros, esta distancia permite una detección efectiva a una distancia intermedia. La disminución en la detección a 4 metros puede atribuirse a la baja iluminación en esta zona, lo que provocó que el arma a detectar se confundiera con el entorno. No obstante, esta distancia sigue siendo útil para aplicaciones donde los sujetos no están ni demasiado cerca ni demasiado lejos, como en entornos con un rango moderado de espacio que requieren una cobertura de detección razonable.

A 6 metros, se detectaron 23 verdaderos positivos de un total de 34 casos. Esta distancia es eficaz para la detección a mayor distancia, lo que es beneficioso en situaciones donde los sujetos están más alejados. Es ideal para áreas grandes o espacios abiertos donde captar sujetos a mayor distancia es importante.

## CANTIDAD DE IMÁGENES-VERDADEROS POSITIVOS (TP)

Tabla cruzada Cantidad de imágenes*Verdaderos positivos				
Recuento				
		Verdaderos positivos		Total
		no detecta	sí detecto	
Cantidad de imágenes	500 imágenes	19	14	33
	700 imágenes	11	22	33
	3000 imágenes	4	30	34
Total		34	66	100

**Figura 72** Tabla cruzada Cantidad de imágenes\*Verdaderos positivos



**Figura 73** Gráfico De Barras

## ANÁLISIS

La tabla cruzada muestra cómo varía la detección de verdaderos positivos según el número de imágenes utilizadas para entrenar los modelos (500, 700 y 3000 imágenes). El modelo entrenado con 500 imágenes logró detectar 14 verdaderos positivos de un total de 33 casos evaluados. Aunque la cantidad de datos de entrenamiento es menor, este modelo demostró una capacidad adecuada para la detección, indicando que con una menor base de datos se



puede alcanzar una efectividad considerable, especialmente en aplicaciones donde la eficiencia computacional es crucial.

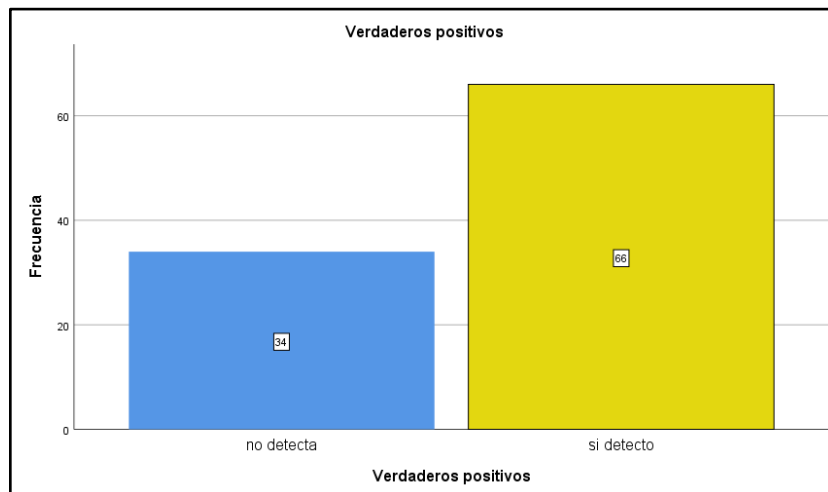
En contraste, el modelo entrenado con 700 imágenes detectó 22 verdaderos positivos de los mismos 33 casos. Este incremento en la cantidad de imágenes resultó en una mejora significativa en la capacidad de detección. La precisión y la robustez del modelo aumentaron, demostrando que una base de datos más extensa puede mejorar notablemente la capacidad del sistema para identificar correctamente los positivos.

El modelo entrenado con 3000 imágenes alcanzó la detección de 30 verdaderos positivos de 34 casos. Este modelo mostró la mayor efectividad, beneficiándose de una base de datos amplia que permitió capturar con alta precisión los detalles y variaciones presentes en los datos de prueba. Esta capacidad es esencial en escenarios donde la precisión es crítica y se dispone de recursos adecuados para procesar grandes volúmenes de datos.

En resumen, la cantidad de imágenes utilizadas en el entrenamiento influye significativamente en la capacidad de detección de los modelos. Modelos con menos datos pueden ser efectivos en aplicaciones con limitaciones de recursos, mientras que modelos entrenados con bases de datos más extensas ofrecen una mayor precisión y confiabilidad en la detección de verdaderos positivos. La elección del tamaño de la base de datos debe equilibrarse con las necesidades específicas del sistema y los recursos disponibles para maximizar la efectividad del modelo.

## TABLA DE FRECUENCIAS DE LOS PARÁMETROS ANALIZADOS

- **VERDADEROS POSITIVOS**



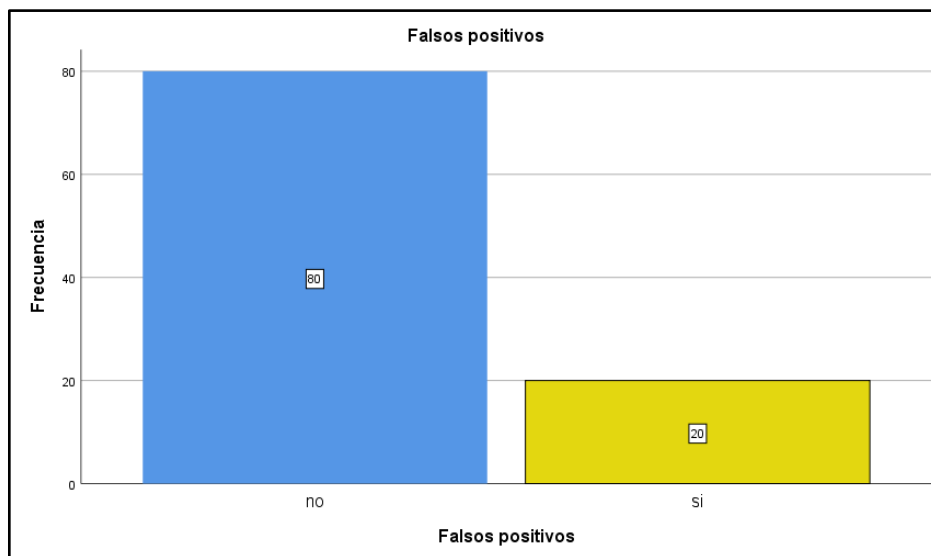
**Figura 74** Verdaderos Positivos

La tabla muestra la eficacia de un sistema para detectar verdaderos positivos en 100 casos. De estos, 34 casos no fueron detectados (34.0%), mientras que 66 casos fueron correctamente detectados como verdaderos positivos (66.0%).

El análisis es completo, ya que todos los casos evaluados están incluidos, sumando al 100.0%. Esta información proporciona una visión clara de la efectividad del sistema y destaca su capacidad para identificar correctamente la mayoría de los verdaderos positivos.

En resumen, el sistema tiene una tasa de detección efectiva del 66.0%, lo que refleja una buena capacidad para identificar verdaderos positivos en la muestra analizada.

- **FALSOS POSITIVOS**



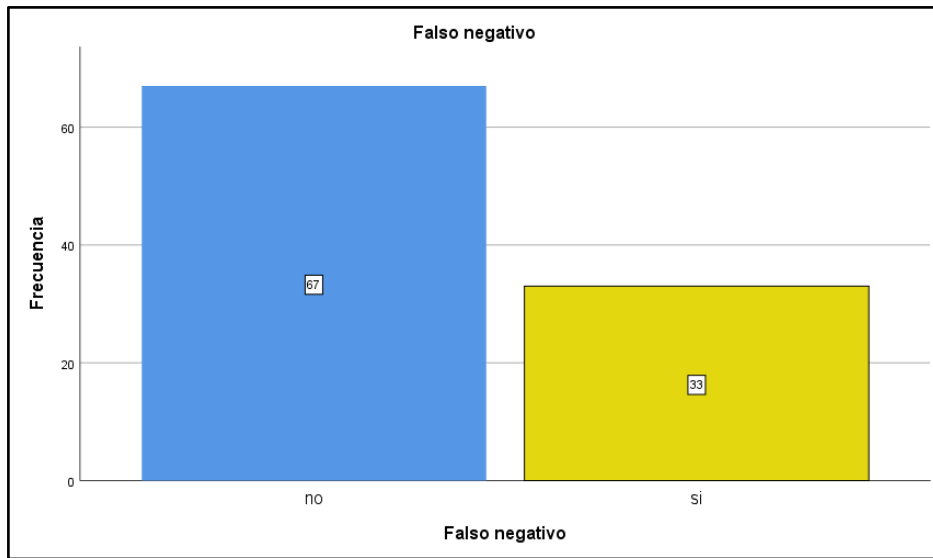
**Figura 75** Falsos Positivos

La tabla de falsos positivos proporciona una visión crítica sobre la precisión del sistema en la detección de eventos no deseados. En el 80.0% de los casos evaluados, no se observaron falsos positivos, indicando una alta especificidad del sistema para evitar alertas erróneas. Esta capacidad es fundamental en aplicaciones donde la minimización de falsas alarmas es crucial para mantener la eficiencia operativa y la confianza del usuario.

Por otro lado, el 20.0% restante de los casos registró falsos positivos. Este resultado subraya la necesidad de mejorar la capacidad del sistema para discriminar de manera más precisa entre eventos reales y falsas alarmas. La reducción de los falsos positivos es esencial no solo

para mejorar la precisión general del sistema, sino también para reducir costos asociados con intervenciones innecesarias y recursos mal dirigidos.

- **FALSOS NEGATIVO**



**Figura 76 Falsos Negativo**

La tabla de falsos negativos proporciona una evaluación crítica de la sensibilidad del sistema en la detección de casos positivos dentro de una muestra de 100 casos. En el 67.0% de los casos evaluados, no se observaron falsos negativos, indicando una alta capacidad del sistema para identificar correctamente los casos positivos.

Por el contrario, en el 33.0% restante de los casos, se registraron falsos negativos. Esta proporción subraya la necesidad de mejorar la capacidad del sistema para capturar todos los casos positivos de manera efectiva.

**TABLA DE EVALUACIÓN ARMA BLANCA**

ID de Prueba	Cantidad de Imágenes que compone la base de datos	Distancia (metros)	TP	FP	FN	TN	Presión	Sensibilidad
1	500	2	0	1	1	0	40%	33.33%
2	500	2	0	1	1	0		

3	500	2	0	0	1	0			
4	500	2	0	1	1	0			
5	500	2	0	0	1	0			
6	500	2	0	1	1	0			
7	500	2	1	0	0	0			
8	500	2	0	1	1	0			
9	500	2	1	0	0	0			
10	500	2	1	0	0	0			
11	500	2	1	0	0	0			
TOTAL			4	6	8	0			
ID de Prueba	Cantidad de Imágenes que compone la base de datos	Distancia (metros)	TP	FP	FN	TN	Precisión	Sensibilidad	
							33.33%	20%	
12	500	4	0	0	1	0			
13	500	4	1	0	0	0			
14	500	4	0	0	1	0			
15	500	4	0	1	1	0			
16	500	4	0	1	1	0			
17	500	4	0	0	1	0			
18	500	4	0	0	1	0			
19	500	4	0	1	1	0			
20	500	4	1	0	0	0			
21	500	4	0	0	1	0			
22	500	4	0	1	1	0			
TOTAL			2	4	8	0			
ID de Prueba	Cantidad de Imágenes que compone la base de datos	Distancia (metros)	TP	FP	FN	TN	Precisión	Sensibilidad	
23	500	6	0	1	0	0	37.5%	33.33%	
24	500	6	0	0	1	0			
25	500	6	0	0	1	0			
26	500	6	1	0	0	0			

27	500	6	0	1	0	0		
28	500	6	1	0	0	0		
29	500	6	0	0	1	0		
30	500	6	0	1	1	0		
31	500	6	1	0	0	0		
32	500	6	0	1	1	0		
33	500	6	0	1	1	0		
TOTAL			3	5	6	0		
ID de Prueba	Cantidad de Imágenes que compone la base de datos	Distancia (metros)	TP	FP	FN	TN	precisión	Sensibilidad
34	700	2	1	0	0	0	75%	54.55%
35	700	2	1	0	0	0		
36	700	2	1	0	0	0		
37	700	2	0	1	1	0		
38	700	2	0	0	1	0		
39	700	2	1	1	0	0		
40	700	2	1	0	0	0		
41	700	2	1	0	0	0		
42	700	2	0	0	1	0		
43	700	2	0	0	1	0		
44	700	2	0	0	1	0		
TOTAL			6	2	5	0		
ID de Prueba	Cantidad de Imágenes que compone la base de datos	Distancia (metros)	TP	FP	FN	TN	Precisión	sensibilidad
45	700	4	1	0	0	0	62.5%	45.45%
46	700	4	0	0	1	0		
47	700	4	0	1	1	0		
48	700	4	0	0	1	0		
49	700	4	0	1	0	0		
50	700	4	0	0	1	0		

51	700	4	0	1	1	0		
52	700	4	1	0	0	0		
53	700	4	1	0	0	0		
54	700	4	1	0	1	0		
55	700	4	1	0	0	0		
TOTAL			5	3	6	0		
ID de Prueba	Cantidad de Imágenes que compone la base de datos	Distancia (metros)	TP	FP	FN	TN	Precisión	Sensibilidad
56	700	6	1	0	0	0	66.67%	60%
57	700	6	0	0	1	0		
58	700	6	0	1	1	0		
59	700	6	1	0	0	0		
60	700	6	1	0	0	0		
61	700	6	0	1	1	0		
62	700	6	1	0	0	0		
63	700	6	1	0	0	0		
64	700	6	0	1	0	0		
65	700	6	1	0	0	0		
66	700	6	0	0	1	0		
TOTAL			6	3	4	0		
ID de Prueba	Cantidad de Imágenes que compone la base de datos	Distancia (metros)	TP	FP	FN	TN	precisión	Sensibilidad
67	3000	2	1	0	0	0	100%	90.91%
68	3000	2	1	0	0	0		
69	3000	2	1	0	0	0		
70	3000	2	1	0	0	0		
71	3000	2	1	0	0	0		
72	3000	2	1	0	0	0		
73	3000	2	1	0	0	0		
74	3000	2	0	0	1	0		

75	3000	2	1	0	0	0				
76	3000	2	1	0	0	0				
77	3000	2	1	0	0	0				
TOTAL			10	0	1	0				
ID de Prueba	Cantidad de Imágenes que compone la base de datos	Distancia (metros)	TP	FP	FN	TN	precisión	Sensibilidad		
78	3000	4	1	0	0	0	81.82%	81.82%		
79	3000	4	1	0	0	0				
80	3000	4	1	0	0	0				
81	3000	4	1	0	0	0				
82	3000	4	0	1	1	0				
83	3000	4	1	0	0	0				
84	3000	4	1	0	0	0				
85	3000	4	1	0	0	0				
86	3000	4	1	0	0	0				
87	3000	4	1	0	0	0				
88	3000	4	1	0	0	0				
TOTAL			10	1	1	0				
ID de Prueba	Cantidad de Imágenes que compone la base de datos	Distancia (metros)	TP	FP	FN	TN	Precisión	Sensibilidad		
89	3000	6	1	0	0	0	91.67%	91,67%		
90	3000	6	1	0	0	0				
91	3000	6	1	0	0	0				
92	3000	6	1	0	0	0				
93	3000	6	1	0	0	0				
94	3000	6	0	1	1	0				
95	3000	6	1	0	0	0				
96	3000	6	1	0	0	0				
97	3000	6	1	0	0	0				
98	3000	6	1	0	0	0				

99	3000	6	1	0	0	0		
100	3000	6	1	0	0	0		
TOTAL			11	1	1	0		

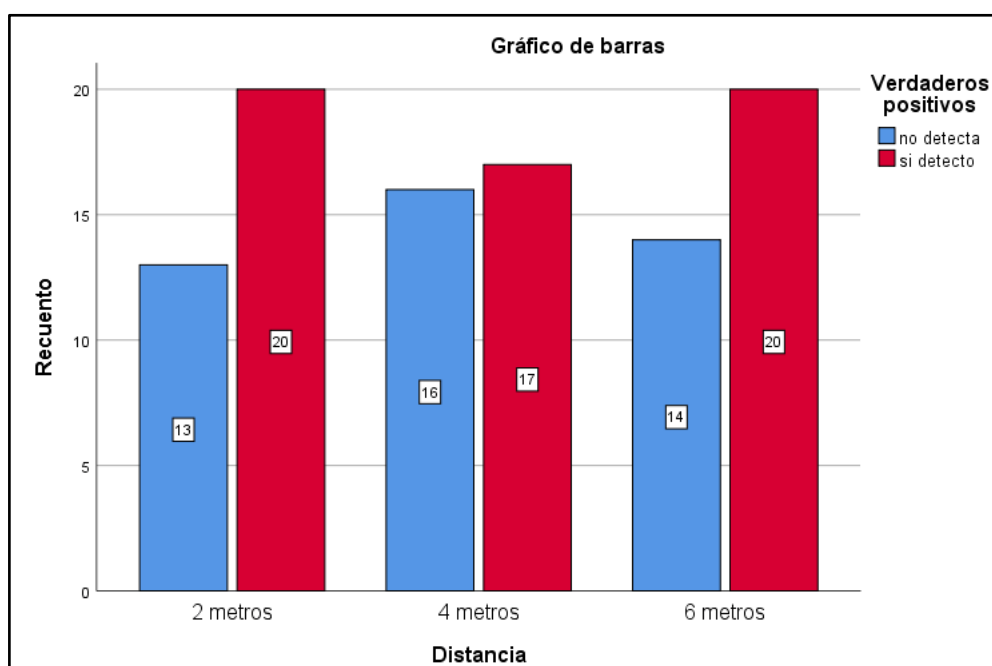
**Figura 77** Tabla De Evaluación Arma Blanca

## ANÁLISIS DESCRIPTIVO DE LAS VARIABLES

### DISTANCIA – VERDADEROS POSITIVOS (PT)

Tabla cruzada Distancia*Verdaderos positivos				
Recuento				
		Verdaderos positivos		Total
		no detecta	si detecto	
Distancia	2 metros	13	20	33
	4 metros	16	17	33
	6 metros	14	20	34
Total		43	57	100

**Figura 78** Tabla cruzada Distancia\*Verdaderos positivos



**Figura 79** Gráfico De Barras

## ANÁLISIS

La tabla cruzada de Distancia versus Verdaderos Positivos revela cómo la efectividad de la detección varía según la distancia en una muestra de 100 casos evaluados. En general, se



observa que a medida que aumenta la distancia de detección, también aumenta la proporción de casos donde se logra detectar verdaderos positivos. Por ejemplo, a 6 metros se detectaron 20 verdaderos positivos de un total de 34 casos evaluados, mientras que a 2 metros se detectaron 20 de 33 casos. Esto sugiere una relación positiva entre la distancia y la capacidad del sistema para identificar correctamente los casos positivos.

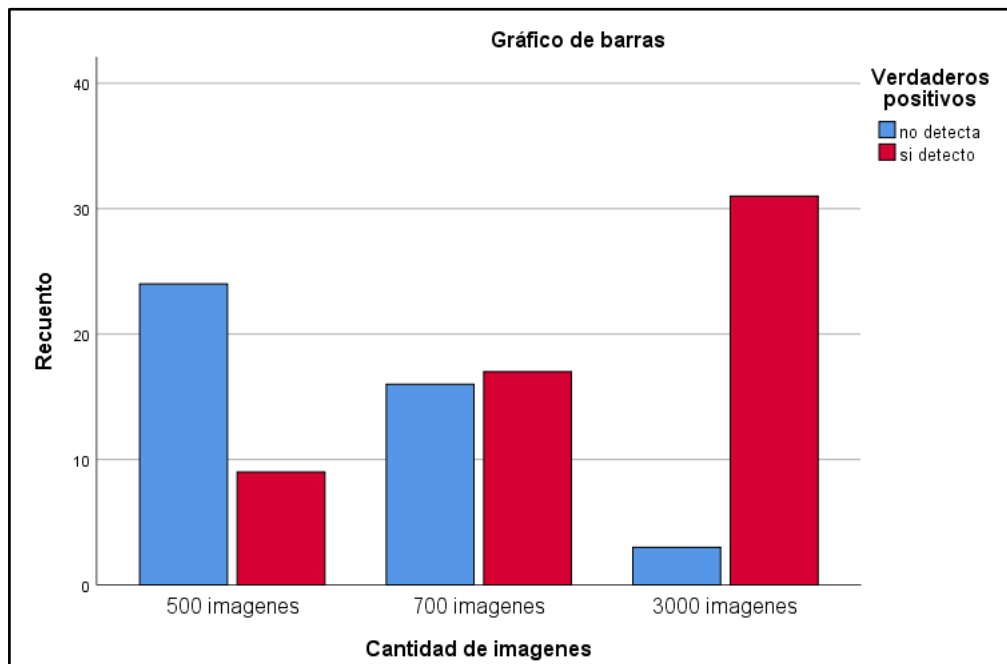
En detalle, a 2 metros se registraron 13 casos donde no se detectaron verdaderos positivos, lo que representa un área donde el sistema puede necesitar ajustes para mejorar la sensibilidad. A 4 metros, aunque se detectaron menos casos en comparación con 6 metros, la distribución fue más equilibrada con 16 casos no detectados y 17 verdaderos positivos detectados. Esta distancia intermedia muestra una efectividad moderada de detección.

Por otro lado, a 6 metros se alcanzó la mayor cantidad de verdaderos positivos detectados, con solo 14 casos no detectados de un total de 34. Este análisis subraya la importancia de considerar la distancia como un factor crítico en el diseño y la optimización de sistemas de detección.

### CANTIDAD DE IMÁGENES - VERDADEROS POSITIVOS (PT)

Tabla cruzada Cantidad de imágenes*Verdaderos positivos				
Recuento				
		Verdaderos positivos		Total
		no detecta	si detecto	
Cantidad de imágenes	500 imágenes	24	9	33
	700 imágenes	16	17	33
	3000 imágenes	3	31	34
Total		43	57	100

**Figura 80** Tabla cruzada Cantidad de imágenes\*Verdaderos positivos



**Figura 81** Gráfica de Barras

## ANÁLISIS

La tabla cruzada de Cantidad de Imágenes versus Verdaderos Positivos ofrece una evaluación detallada de cómo la cantidad de imágenes en la base de datos impacta la capacidad del sistema para detectar correctamente casos positivos en una muestra de 100 casos evaluados. A 500 imágenes, se detectaron solo 9 verdaderos positivos de un total de 33 casos evaluados, con 24 casos donde no se logró la detección. Esto sugiere una limitación en la capacidad del sistema para capturar casos positivos cuando se dispone de una base de datos más reducida.

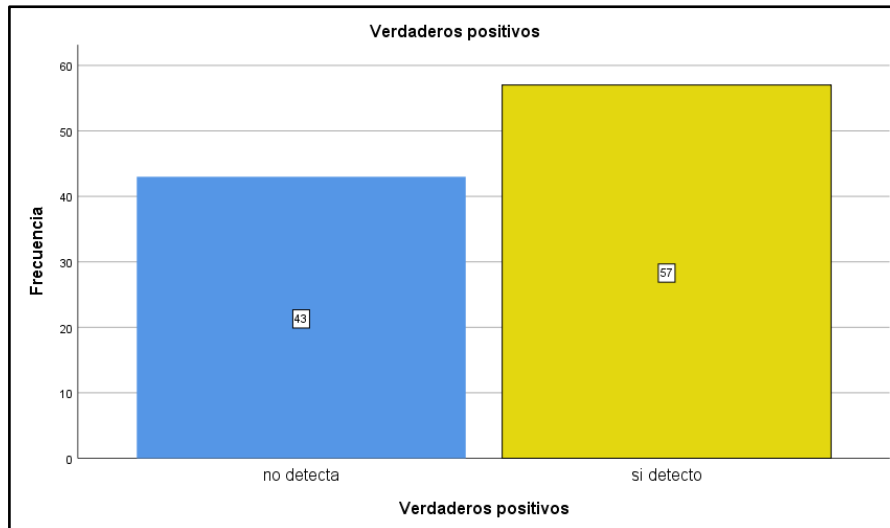
En contraste, con 700 imágenes, se mejoró la efectividad de detección, detectándose 17 verdaderos positivos de 33 casos evaluados, mientras que, con 3000 imágenes, se detectaron 31 verdaderos positivos de 34 casos evaluados, demostrando una notable mejora en la capacidad del sistema para identificar correctamente los casos positivos a medida que aumenta la cantidad de imágenes disponibles.

Este análisis destaca la importancia crítica de la cantidad de imágenes en la base de datos como factor determinante en la efectividad del sistema de detección. Aumentar la cantidad de imágenes no solo mejora la capacidad del sistema para identificar verdaderos positivos,

sino que también optimiza su rendimiento en aplicaciones donde la precisión y la confiabilidad son fundamentales, como en sistemas de diagnóstico médico o de seguridad.

### TABLA DE FRECUENCIAS DE LOS PARÁMETROS ANALIZADOS

- VERDADEROS POSITIVOS

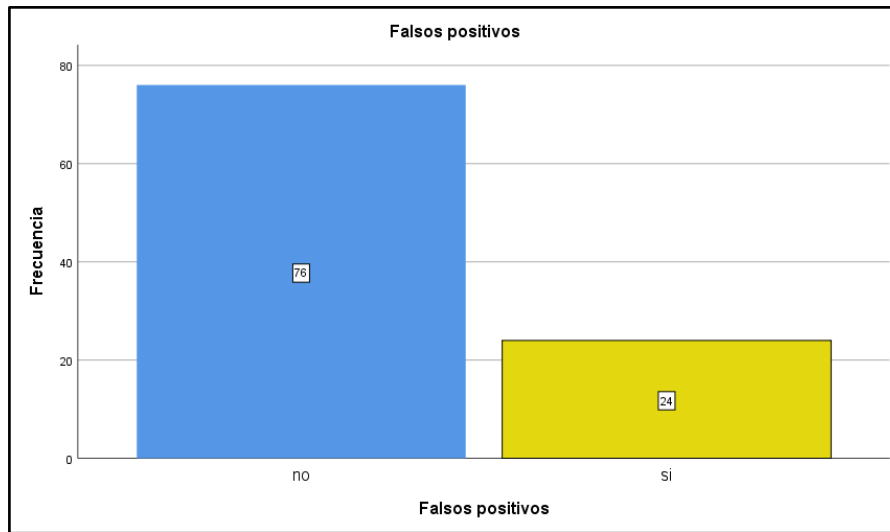


**Figura 82** Verdaderos Positivos

La tabla de Verdaderos Positivos proporciona una evaluación detallada de la capacidad del sistema para detectar correctamente eventos o condiciones de interés en una muestra de 100 casos evaluados. De los casos analizados, se observa que el 57.0% fueron detectados como verdaderos positivos, lo cual indica una capacidad efectiva del sistema para identificar adecuadamente eventos relevantes.

Por otro lado, el 43.0% de los casos no fueron detectados como verdaderos positivos, lo que subraya áreas potenciales para mejorar la sensibilidad del sistema. Este análisis enfatiza la importancia de equilibrar la sensibilidad del sistema (capacidad para detectar verdaderos positivos) con la especificidad (capacidad para evitar falsos positivos y negativos).

- **FALSO POSITIVOS**

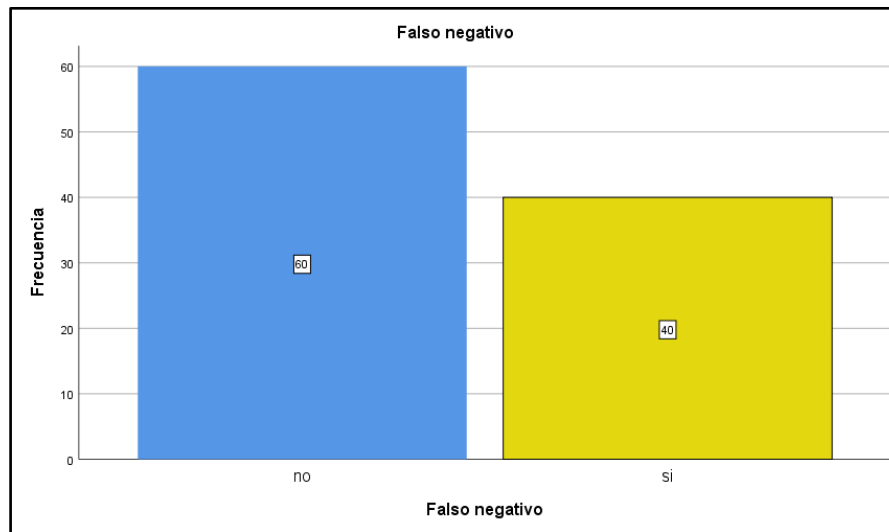


**Figura 83** *Falso Positivos*

La tabla de Falsos Positivos presenta una evaluación detallada de cómo el sistema ha manejado la identificación errónea de eventos como positivos en una muestra de 100 casos evaluados. Según los datos, el 76.0% de los casos fueron correctamente identificados como negativos, lo que indica una alta precisión del sistema en evitar etiquetar incorrectamente eventos no relevantes como positivos.

Sin embargo, se observó que el 24.0% de los casos fueron identificados erróneamente como falsos positivos. Estos casos resaltan áreas potenciales para mejorar la especificidad del sistema, como ajustes en los criterios de detección o refinamientos en los algoritmos de análisis de datos.

- **FALSOS NEGATIVOS**



**Figura 84** *Falsos Negativos*

La tabla de Falso Negativo revela cómo el sistema ha enfrentado la situación donde eventos positivos no fueron detectados en una muestra de 100 casos evaluados. Según los resultados, el 60.0% de los casos fueron correctamente identificados como negativos, indicando una efectividad en la detección cuando no había eventos positivos presentes. Esta capacidad es esencial en contextos donde la precisión es crucial.

Por otro lado, el 40.0% de los casos fueron mal clasificados como falsos negativos. Estos resultados resaltan la necesidad de mejorar la sensibilidad del sistema, lo que podría lograrse mediante ajustes en los parámetros de detección o mejorando la calidad de los datos de entrada.

## CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

### 5. CONCLUSIONES

- Después de un extenso proceso de investigación y desarrollo se a logrado diseñar e implementar un sistema de seguridad inteligente basado en visión artificial para la farmacia Pharmatodo de la Ciudad de Riobamba. Este sistema ha demostrar ser capaz de detectar eficazmente armas de fuego cortas y armas blancas(cuchillos). la implementación de este sistema requirió un enfoque gradual, superando varios desafíos técnicos y ajustando continuamente los parámetros para optimizar su rendimiento. Como resultado, se a conseguido un sistema eficiente y adecuado que mejora significativamente la seguridad y contribuye a la prevención de robos en tiempo real en la farmacia.
- Tras una exhaustiva investigación, se evaluaron varios tipos de algoritmos y modelos para la detección de armas y gestos corporales mediante visión artificial. Se determinaron las ventajas y desventajas de cada uno, realizando descartes progresivos que nos condujeron a enfocarnos en dos algoritmos principales: YOLO v5 y Haar Cascade, junto con un modelo de red neuronal convolucional. Cada opción fue meticulosamente evaluada en términos de su capacidad para reconocer armas de fuego cortas, armas blancas y gestos corporales en tiempo real. Finalmente, se optó por implementar YOLO v5 debido a su rendimiento superior en tiempo real y su capacidad para satisfacer los requisitos específicos del proyecto. Para la detección de postura, se integró el algoritmo de detección de postura de la biblioteca Mediapipe, que complementó eficazmente la funcionalidad del sistema al identificar gestos corporales relevantes para la prevención de robos.
- El diseño del sistema de comunicación confiable para la transmisión de alertas de robo se fundamentó en la implementación de Firestore, una base de datos NoSQL de Google Cloud, conocida por su robustez y seguridad. Firestore facilita una comunicación segura entre el sistema de seguridad y sus servidores, donde se almacenan de manera eficiente todas las alertas generadas. La arquitectura del sistema permite que la aplicación receptora de notificaciones mantenga una conexión persistente con Firestore, garantizando una respuesta inmediata a cualquier cambio en la base de datos. Esta integración no solo asegura la entrega confiable de alertas en tiempo real a los propietarios de PharmaTodo, sino que también establece una

infraestructura escalable y adaptable para futuras mejoras en la gestión y análisis de datos de seguridad.

- Para asegurar el correcto funcionamiento del sistema desarrollado, se llevaron a cabo tres tipos de pruebas distintas. En primer lugar, se realizaron pruebas en un entorno ideal con iluminación constante y distancias controladas para evaluar la sensibilidad y precisión en condiciones óptimas. Los resultados mostraron que el sistema demostró una sensibilidad y precisión adecuadas en estas condiciones controladas. Posteriormente, se evaluó el sistema en situaciones reales dentro de la farmacia PharmaTodo, enfrentándolo a escenarios más complejos que incluyeron variaciones en la iluminación, distancia y contextos adversos. A pesar de estos desafíos adicionales, el sistema mantuvo niveles satisfactorios de sensibilidad y precisión en la detección de objetos. Además, se llevaron a cabo pruebas de robustez diseñadas para simular situaciones donde el sistema podría enfrentar intentos de confundirlo, como la simulación de la presencia de armas y gestos ambiguos. Estas pruebas permitieron validar la capacidad del sistema para mantener su precisión y fiabilidad bajo condiciones desafiantes.

## **5.1 RECOMENDACIONES**

- Si bien el sistema de seguridad demostró un funcionamiento favorable en general, se observó mediante análisis estadístico una mínima detección en el área de análisis a 4 metros de la cámara. Este hallazgo se atribuyó a la baja iluminación presente en esta zona, lo que afecta la eficiencia de detección del sistema. Se recomienda aumentar la iluminación en áreas de baja luminosidad para mejorar la fiabilidad y precisión del sistema de seguridad, asegurando así una cobertura efectiva y continua en todas las condiciones ambientales.

## **BIBLIOGRAFÍA**

- [1] Primicias.” Seis de cada diez robos en Ecuador se producen a plena luz del día” obtenido de : <https://acortar.link/AjZC5z>. (2022).
- [2] Lechner, M. “Tecnologías aplicadas a la seguridad ciudadana: desafíos para la justicia transicional ante nuevos mecanismos de control social”. *Divulgatio*, 1(1). Disponible en RIDAA-UNQ Repositorio Institucional Digital de Acceso Abierto de la Universidad Nacional de Quilmes <http://ridaa.unq.edu.ar/handle/20.500.11807/264>. (2016).
- [3] Carlos, J. “Visión artificial aplicada en la identificación de objetos y su parametrización geométrica” Universidad Carlos III de Madrid, (2019).
- [4] Zúñiga, L. “Seguridad y violencia: desafíos para la ciudadanía”. Facultad Latinoamericana de Ciencias, (2007).
- [5] Jaime, A. “La tecnología y la seguridad ciudadana ” obtenido de: <https://acortar.link/rzxWSS>, (2023).”
- [6] Alvear-Puertas, V. et al “Internet de las Cosas y Visión Artificial, Funcionamiento y Aplicaciones”. *Enfoque UTE*, 244-256. (2017).
- [7] Fanny V. “Robo a comercios tiene más incidencia en siete provincias y afecta la imagen de Ecuador,”2023. <https://www.lahora.com.ec/pais/robo-comercios-inseguridad-turismo-ecuador/>. (2023).
- [8] Psicoactivacom, “Secuelas psicológicas tras un robo o intrusión en casa,” *PsicoActiva.com: Psicología, test y ocio Inteligente*, Feb. 16, 2023. <https://www.psicoactiva.com/blog/secuelas-psicologicas-sufrir-robo-intrusion/>, (2023).
- [9] Plussegur, “Sistemas de Seguridad» Guía Informativa Actualizada **[2022]** ,” Plussegur, Apr. 05, 2019. <https://plussegur.com/>. (2023).
- [10] S. Draghici, "A Neural Network Based Artificial Vision System for Licence Plate Recognition," in *Vision and Neural Networks Laboratory*, Dept. of Computer Science, Wayne State University, 431 State Hall, Detroit, MI, USA.(1997).
- [11] J. A. Ludeña Chica, "Implementación de un sistema de seguridad para supervisión de niños entre 2 a 4 años usando Visión Artificial."(2019).
- [12] G. D. Brito Moncayo, "Sistema de visión artificial con monitoreo web para conteo de pasajeros en buses de transporte público."(2018).
- [13] J. C. Broncano Torres and M. S. T., "Control de Seguridad y Acceso Vehicular Mediante Visión Artificial”(2020).



- [14] Primicias. “Los robos a negocios y empresas es otra cara de la inseguridad, según la Fiscalía”. Obtenido de : <https://www.primicias.ec/noticias/economia/robos-negocios-inseguridad-empresas/#:~:text=Ministerio%20de%20Gobierno>. (2024).
- [15] Ecuador en Cifras. “Directorio de Empresas”. Obtenido de <https://www.ecuadorencifras.gob.ec/directoriodeempresas/>. (2024).
- [16] Fiscalía General del Estado de Ecuador. (s.f.). “Consulta de denuncias en línea”. Obtenido de <https://www.fiscalia.gob.ec/consulta-de-denuncias-en-linea-tramite-1/>.(2024).
- [17] Mideros Mora, A. “Los desafíos económicos frente a la crisis de inseguridad”. Obtenido de: <https://n9.cl/u16747>.(2024).
- [18] La Hora. “Economía: entorno para hacer negocios, principales damnificados de la creciente inseguridad”. Recuperado de <https://www.lahora.com.ec/pais/economia-entorno-hacer-negocios-principales-damnificados-creciente-inseguridad/>.(2024)
- [19] García, I., & Caranqui, V. “La visión artificial y los campos de aplicación.”.(2014)
- [20] Atria Innovation, "Guía sobre visión artificial para principiantes," Obtenido de: <https://n9.cl/18gzif>. (2022). Control de Miopía, "Algoritmos en Visión Artificial," Obtenido de: <https://controldemiopia.net/articulos/algoritmos-en-vision-artificial/>.(2022).
- [21] G. Jocher, "YoloV5" Obtenido de: <https://docs.ultralytics.com/es/models/yolov5/>. (2024).
- [22] C. Bonilla Carrión, "Redes Convolucionales," (2020).
- [23] A. Ahmadi, "CASCADE TRAINER GUI," Obtenido de: <https://amin-ahmadi.com/cascade-trainer-gui/>.
- [24] ML-KIT, "Pose Detection," Obtenido de: <https://developers.google.com/ml-kit/vision/pose-detection?hl=es-419>.
- [25] G. Jocher, "YOLOv5," obtenido de: <https://docs.ultralytics.com/es/models/yolov5/>
- [26] T.-Y. Lin et al., "Focal Loss for Dense Object Detection,"(2018).
- [27] Analytics Vidhya, "YOLOv4 vs YOLOv4 Tiny," Obtenido de: <https://n9.cl/y7sqpk>.(2021).
- [28] IBM. “Convolutional Neural Networks.” Obtenido de <https://www.ibm.com/es-es/topics/convolutional-neural-networks>.

## ANEXOS

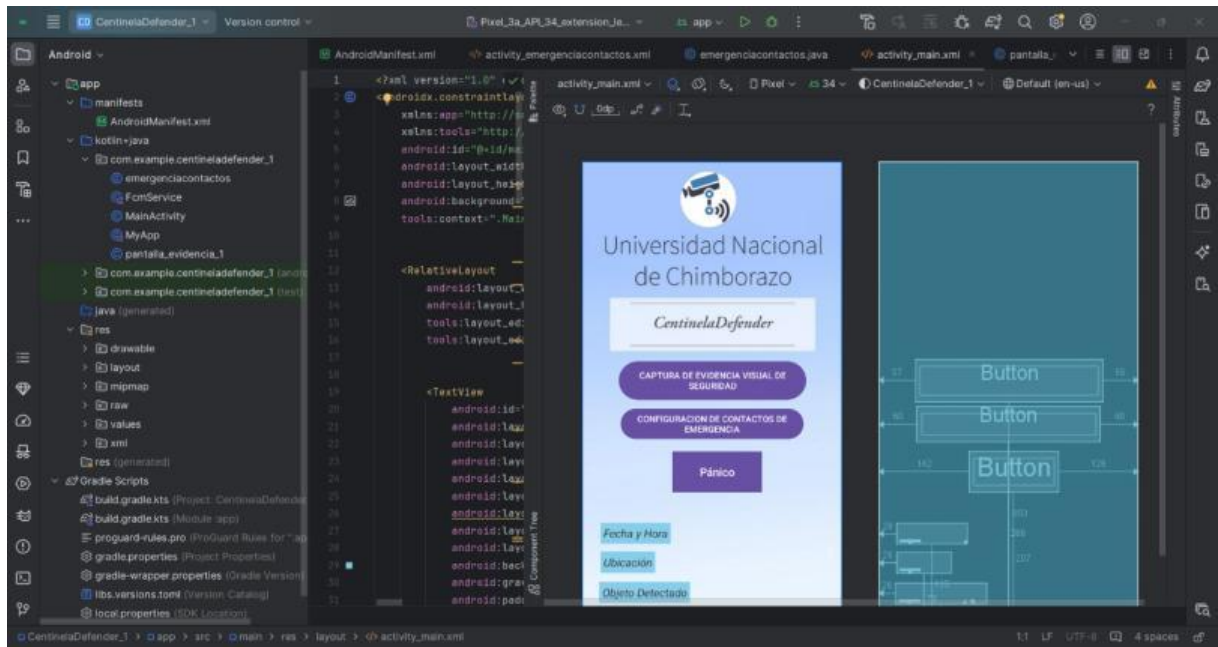


Figura 85 Diseño y codificación de la aplicación CentinelaDefender

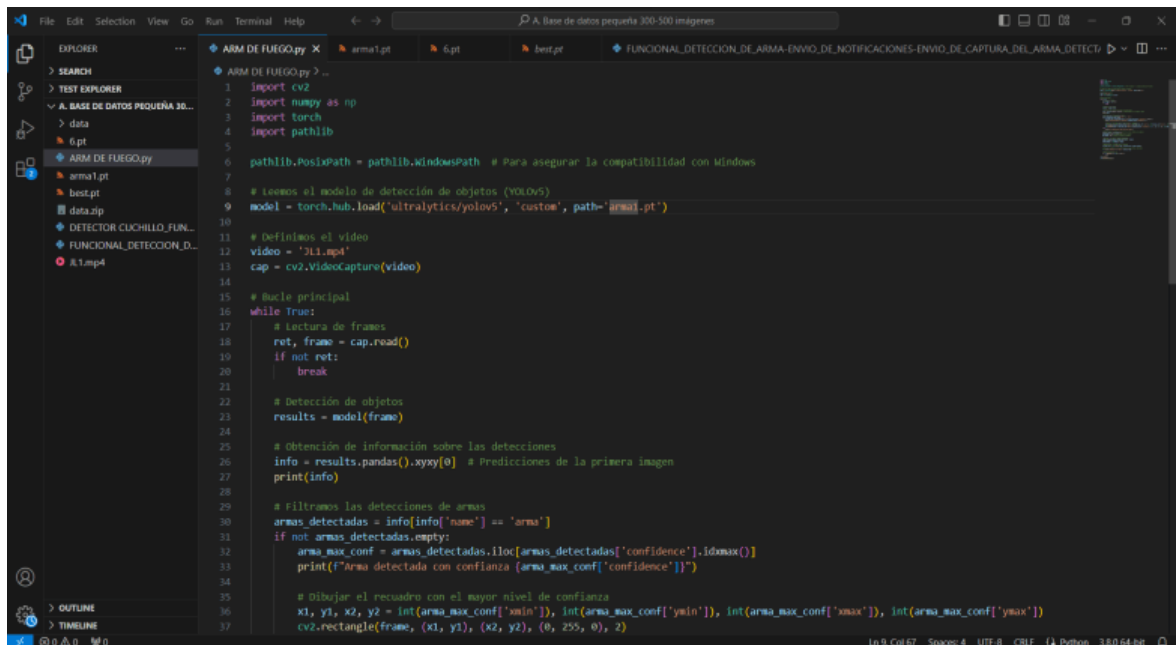
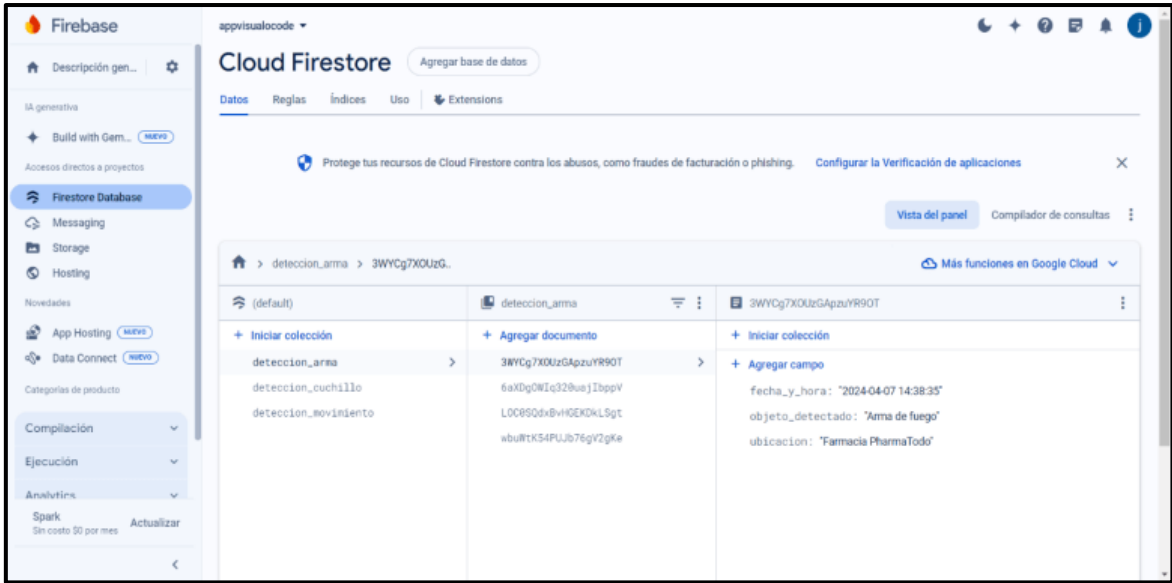
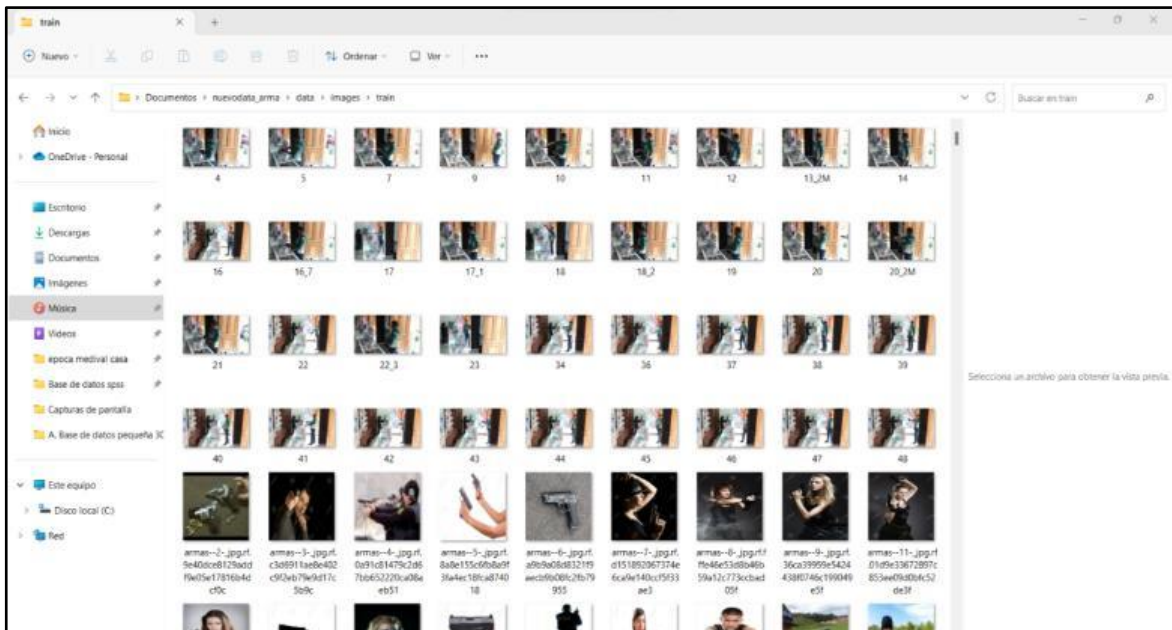


Figura 86 Codificación en Python diseñado para la detección de armas



**Figura 87** Configuración de la base de datos Firebase



**Figura 88** Construcción de la base de datos



**Figura 89** Elementos utilizados para el desarrollo del sistema de seguridad: cámara, computador y trípode



**Figura 90** elementos utilizados para la detección

## RECOLECCIÓN DE IMÁGENES EN LA FARMACIA FORMATO



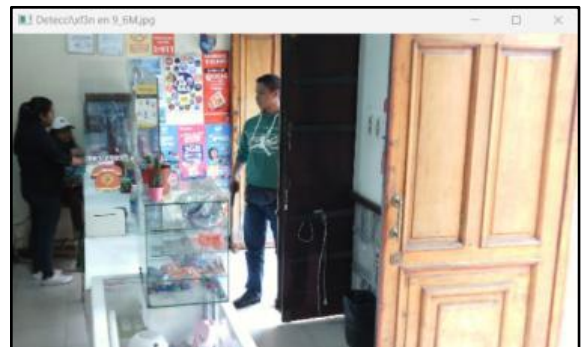
**Figura 92** *Farmacia\_1*



**Figura 91** *Farmacia\_2*



**Figura 94** *Farmacia\_3*



**Figura 93** *Farmacia\_4*



**Figura 96** *Farmacia\_5*



**Figura 95** *Farmacia\_6*

## CARTA DE ACEPTACIÓN

### CARTA DE ACEPTACIÓN DE PROYECTO DE INVESTIGACIÓN DE LA FARMACIA PHARMATODO

Riobamba, 21 de junio del 2023

Cordial saludo.

Por este medio, **Ivette Patricia Inca Cando**, nos complace notificarle la aceptación del proyecto "Diseño e implementación de un sistema inteligente basado en visión artificial para el monitoreo y prevención de robos en tiempo real en la Farmacia PharmaTodo de la ciudad de Riobamba" a llevarse a cabo por Jimmy Andres Villa Yuquilema, a partir del próximo 23 de junio del 2023 hasta 31 de octubre del 2023.

Como se indicó en la presentación del proyecto, Jimmy Andres Villa Yuquilema estará a cargo del diseño, implementación y fase de pruebas del sistema inteligente basado en visión artificial para el monitoreo y prevención de robos en tiempo real en la farmacia PharmaTodo. Por su lado, PharmaTodo será responsable de prestar todas las facilidades para la realización del proyecto de tesis.

Estamos seguros de que la culminación de este proyecto se llevará a cabo bajo las condiciones y características estipuladas en la convocatoria.

Atentamente,

  
**Ivette Patricia Inca Cando**  
Propietaria  
Farmacia Pharmatodo  
Cel:0989061821

FARMACIA  
**PHARMA  
Todo** CEL  
0989061821  
RUC 2100413273001

*Tabla 17 Carta de aceptación*