

ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA-FEPADE



“PROTOTIPO DE SISTEMA DE VIDEOVIGILANCIA AUTOMATIZADO POR MEDIO
DE VISIÓN ARTIFICIAL PARA ITCA-FEPADE.”

TRABAJO DE INVESTIGACIÓN PARA OPTAR AL GRADO DE
INGENIERO EN DESARROLLO DE SOFTWARE

POR

Edgardo Antonio Argueta Bautista

Jefferson Francisco Pineda Quintanilla

Kevin Rodrigo Flores Acosta

Nicolás Gabriel Escobar Pérez

JUNIO 2023

SANTA TECLA, LA LIBERTAD, EL SALVADOR, C.A.

ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA-FEPADE

ESCUELA DE INGENIERÍA EN COMPUTACIÓN

RECTOR

CARLOS ALBERTO ARRIOLA MARTÍNEZ

VICERRECTOR ACADÉMICO

CHRISTIAN ANTONIO GUEVARA

DIRECTOR DE ESCUELA

MARTA CORINA QUIJANO DE GARCÍA

COORDINADOR DE ESCUELA

HÉCTOR EDMUNDO GONZÁLEZ MAGAÑA



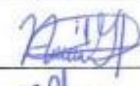

ASESOR DE TRABAJO DE INVESTIGACIÓN

GILBERT ALFREDO GARCÍA CÁCERES

**Escuela Especializada En Ingeniería ITCA-FEPADE
Escuela de Ingeniería de Computación.**

Ingeniería en Desarrollo de Software
Acta de Resolución de Evaluación de Informe Final

Este día **Miércoles 21 de Junio de 2023**, reunida la Comisión Evaluadora en el Campus de la Escuela Especializada en Ingeniería ITCA-FEPADE, para evaluar el proyecto de Trabajo de investigación titulado: **PROTOTIPO DE SISTEMA DE VIDEOVIGILANCIA AUTOMATIZADO POR MEDIO DE VISIÓN ARTIFICIAL PARA ITCA-FEPADE**, el cual ha sido presentado por los estudiantes:

	Nombre completo del estudiante	Firma
1	Edgardo Antonio Argueta Bautista	
2	Jefferson Francisco Pineda Quintanilla	
3	Nicolás Gabriel Escobar Pérez	
4	Kevin Rodrigo Flores Acosta	

Esta Comisión utilizando el instrumento para evaluación de Informe Final que la Escuela de Ingeniería en Computación de la Escuela Especializada en Ingeniería ITCA-FEPADE ha elaborado para tal fin ha asignado las notas y promedio que a continuación se detallan.

Nombre de los miembros de la Comisión Evaluadora	Calificación de Edgardo Antonio Argueta Bautista		Calificación de Jefferson Francisco Pineda Quintanilla		Calificación de Nicolás Gabriel Escobar Pérez		Calificación de Kevin Rodrigo Flores Acosta	
	Doc. Escrito	Pre. Oral	Doc. Escrito	Pre. Oral	Doc. Escrito	Pre. Oral	Doc. Escrito	Pre. Oral
Presidente	7.0	8.5	7.0	8.0	7.0	8.5	7.0	8.0
Secretario	7.6	8.45	7.6	7.95	7.6	8.45	7.6	7.7
Vocal	7.9	8.8	7.9	8.3	7.9	8.8	7.9	8.3
Promedio parcial	7.5	8.47	7.5	7.97	7.5	8.47	7.5	7.88
Promedio Global obtenido en número y letras	8.60 Ocho punto sesenta		8.10 Ocho punto diez		8.60 Ocho punto sesenta		8.00 Ocho punto Cero.	

Anexar los formularios llenos utilizados en la evaluación

Esta Comisión Evaluadora acuerda **Aprobar** y para constancia firmamos.

Nombre Presidente: **Inga. Yancy Steffany Ventura Aquilar.** Firma 

Nombre Secretario **Licda. Ana Maria de Lazo.** Firma 

Nombre Vocal **Inq. Gilbert Alfredo Garcia Cáceres.** Firma 

Escuela Especializada en Ingeniería ITCA-FEPADE

Sistema Bibliotecario



Carta de Autorización de Divulgación

Nosotros Edgardo Antonio Argueta Bautista, Con Documento Único de Identidad número (053390111), Jefferson Francisco Pineda Quintanilla, Con Documento Único de Identidad número (052105351), Kevin Rodrigo Flores Acosta, Con Documento Único de Identidad número (050420828), Nicolás Gabriel Escobar Pérez, Con Documento Único de Identidad número (055819080), estudiantes de la carrera ingeniería en Desarrollo de Software de la Escuela Especializada en Ingeniería ITCA-FEPADE

Manifestamos:

- 1) Que somos los autores del trabajo de graduación que lleva por título: "Prototipo de sistema de videovigilancia por medio de visión artificial para ITCA-FEPADE", y que en adelante denominaremos la obra, presentado como requisito de graduación de la carrera, anteriormente mencionada, el cual fue dirigido y asesorado por el ingeniero Gilbert Alfredo García Cáceres, quien se desempeña como docente del departamento de Computación en la institución.
- 2) Que la obra es una creación original y que no infringe los derechos de propiedad intelectual, ni los derechos de publicidad, comerciales, de propiedad industrial u otros, y que no constituye una difamación, ni una invasión de la privacidad o de la intimidad, ni cualquier injuria hacia terceros.
- 3) Nos responsabilizamos ante cualquier reclamo que se le haga a la Escuela Especializada en Ingeniería ITCA-FEPADE, en este sentido.
- 4) Que estamos debidamente legitimados para autorizar la divulgación de la obra mediante las condiciones de la licencia de Creative Commons. (marcar solo una)
 Reconocimiento (cc by)
 Reconocimiento - Compartir (cc by -sa)
 Reconocimiento - SinObraDerivada (cc by -nd)
 Reconocimiento - NoComercial (cc by-nc)
 Reconocimiento -NoComercial - CompartirIgual (cc by-nc-sa)
 Reconocimiento -NoComercial-SinObraDerivada (cc by-nc-nd)
De acuerdo con la legalidad vigente.
- 5) Que conocemos y aceptamos las condiciones de preservación y difusión de la Especializada en Ingeniería ITCA-FEPADE

En atención a lo antes expuesto solicitamos:

Que la obra quede depositada en las condiciones establecidas en la licencia de difusión anteriormente electa, y en consecuencia, tomando como base al artículo 7 de la Ley de Propiedad Intelectual, cedemos los derechos económicos de explotación necesarios para tal efecto. San Salvador, 12 de junio de 2023

AGRADECIMIENTOS

En primer lugar, quisiera agradecer al cuerpo docente de ITCA-FEPADÉ por proporcionarme los conocimientos y herramientas necesarias para llevar a cabo este proyecto. Todas sus enseñanzas y orientación fueron fundamentales para mi crecimiento académico y profesional.

No puedo dejar de mencionar mi profundo agradecimiento a mis compañeros de clase y amigos que me brindaron su apoyo moral y compartieron su conocimiento a lo largo de este proceso. Sus ideas y comentarios constructivos contribuyeron significativamente a la mejora de este proyecto.

Para finalizar quiero agradecer a mis seres queridos, mi familia y amigos, por todo el apoyo, aliento y comprensión durante este desafiante período. Sus palabras de aliento y su confianza en mis capacidades fueron el motor que me impulsó a seguir adelante y lograr mis metas, ya que, sin el respaldo y colaboración de todos ustedes, este proyecto no habría sido posible.

Su contribución ha dejado una huella significativa en mi vida y en mi desarrollo profesional.

Junio 2023.

Jefferson Francisco Pineda Quintanilla.

Estudiante de Ingeniería en Desarrollo de Software.

AGRADECIMIENTOS

Dios gracias por permitirme completar un objetivo más, por permitirme creer en ti, por la inteligencia y sabiduría que me has dado para poder superar cada una de las etapas, Gracias a mi familia por apoyarme en cada momento.

Agradezco a cada uno de los maestros por el conocimiento impartido en cada momento, he aprendido mucho como persona y profesional, a cada uno de los compañeros por los momentos que compartimos en cada módulo, por los desafíos que superamos y la sinergia que mantuvimos en cada uno de ellos.

Gracias a ITCA-FEPADE en general por la orientación y apoyo brindado en cada etapa.

Completar un objetivo más, es una satisfacción inmensa, es una etapa que será recordada y nos traerá recuerdos satisfactorios de una experiencia muy bonita.

Junio 2023

Nicolás Gabriel Escobar Pérez

Estudiante de Ingeniería en Desarrollo de Software

AGRADECIMIENTOS

Querida familia, compañeros de tesis e institución, quiero tomar un momento para agradecerles por el apoyo que me brindaron durante este proceso de tesis. El proyecto de visión artificial para la detección de movimientos sospechosos fue un desafío emocionante, y es gracias al apoyo y cooperación de todos que hemos logrado finalizar esta aventura con la frente en alto.

A mi querida familia, les agradezco por ser ese pilar indispensable para mi vida. Su apoyo incondicional y palabras de aliento han sido fundamentales para mantenerme firme en todo momento.

A mis compañeros de tesis, quiero expresarles mi gratitud por embarcarnos en esta aventura. Hemos podido experimentar el fascinante campo de la visión artificial y hemos superado el desafío que esto supone. Es la colaboración, dedicación y disposición de cada uno lo que nos ha llevado a alcanzar la meta y les estoy agradecido por haber recorrido este camino juntos.

Por último, quiero agradecer a la institución por permitirme ser parte de esta durante esta carrera. Su apoyo y recursos han sido valiosos para mi formación académica. Agradezco a los profesores y al personal por su dedicación y orientación a lo largo de mi carrera.

Junio 2023

Edgardo Antonio Argueta Bautista

Estudiante de Ingeniería en Desarrollo de Software

AGRADECIMIENTOS

Quiero tomar este espacio para poder expresar mis agradecimientos a todas las personas que me estuvieron apoyando en este largo proceso de formación académica, como Ingeniero, y en el proceso de elaboración de tesis.

Primeramente, a mi mamá y papá, los cuales siempre estuvieron acompañándome en todo el proceso, gracias por ese apoyo incondicional y fortaleza que me llevó hasta acá. De igual manera, a mi círculo de amigos y amigas más cercanos, los cuales siempre me impulsaron en momentos en los que uno se siente sin motivación. Gracias por esa amistad leal y siempre por estar ahí a mi lado, los llevo siempre en mi corazón.

A mis compañeros de carrera y tesis, les agradezco el hecho de compartir este proceso de 5 años muy largos. En donde intercambiamos conocimientos y momentos de colaboración entre todos, para poder lograr este gran desafío.

A la Escuela Especializada en Ingeniería, ITCA-FEPADE, le agradezco por proporcionar un entorno propicio para crecer académicamente. Y a la vez agradecido por la dedicación de mis profesores quienes han compartido su experiencia y conocimiento conmigo.

Sin el apoyo de cada uno de ustedes, este logro no habría sido posible. Mi más sincero agradecimiento.

Junio 2023

Kevin Rodrigo Flores Acosta

Estudiante de Ingeniería en Desarrollo de Software

ÍNDICE

INTRODUCCIÓN	3
CAPÍTULO I ESTADO ANTERIOR	5
DESCRIPCIÓN DEL PROBLEMA	6
JUSTIFICACIÓN	8
MARCO TEÓRICO	10
<i>¿Qué es la visión artificial?</i>	12
<i>Redes neuronales convolucionales</i>	14
<i>YOLO ("Solo miras una vez")</i>	16
<i>OpenCV (Biblioteca de visión artificial de código abierto)</i>	24
<i>MTCNN</i>	27
<i>Evento sospechoso</i>	29
<i>Estándar DORI</i>	30
<i>CUDA</i>	32
<i>PYTORCH</i>	34
MARCO LEGAL	35
<i>Ley de Protección de Datos Personales:</i>	35
<i>Estatutos locales:</i>	36
CAPÍTULO II DESARROLLO DE LA INNOVACIÓN.....	37
OBJETIVO GENERAL	38
OBJETIVOS ESPECÍFICOS	38
METODOLOGÍA Y ESTRATEGIA	39
MODELADO DE LA INNOVACIÓN	44
ARQUITECTURA DEL SISTEMA	46
BASE DE DATOS	48
<i>Diagrama de base de datos</i>	48
<i>Diccionario de datos</i>	49
CASOS DE USO	51
<i>Descripción de actores</i>	51
<i>Diagramas de Caso de Uso</i>	52

<i>Diccionarios de Caso de Uso</i>	53
INTERFAZ GRÁFICA DEL PROTOTIPO.	76
<i>Inicio de sesión</i>	76
<i>Gestión de usuarios</i>	77
<i>Gestión de eventos</i>	77
<i>Configuración</i>	78
<i>Iniciar sesión en el sistema</i>	79
<i>Registrar un nuevo usuario</i>	80
<i>Captura de eventos</i>	81
ORGANIZACIÓN, MONITOREO Y EVALUACIÓN PARA LA INNOVACIÓN	82
CRONOGRAMA Y PRESUPUESTO	84
<i>Costo del personal</i>	84
<i>Costo hardware y software</i>	85
<i>Requisitos operativos</i>	91
FASE DE OPERACIÓN	94
CAPACITACIÓN Y FORMACIÓN	95
CAPÍTULO III RESULTADOS OBTENIDOS DE LA INNOVACIÓN	97
CAMBIOS EN NECESIDADES ABORDADOS	98
PRUEBAS Y DEMOSTRACIONES DE LA EFICACIA, EFICIENCIA Y EFECTIVIDAD.....	100
<i>Resultados de las pruebas</i>	104
PERCEPCIONES Y EVALUACIÓN DEL USUARIO	109
CONCLUSIONES	110
RECOMENDACIONES Y PROPUESTAS	112
<i>Desarrollar una aplicación móvil</i>	112
<i>Integrar sistemas de detección de placas de vehículos</i> ...	112
<i>Explorar técnicas de seguimiento de objetos</i>	112
<i>Mejorar la eficiencia del sistema</i>	113
<i>Detección de colisiones o accidentes</i>	113
SOCIALIZACIÓN DE LOS RESULTADOS.....	114
<i>Plan de Socialización</i>	114

<i>Identificación de la audiencia objetivo</i>	114
<i>Selección de estrategias de socialización</i>	114
<i>Cronograma de socialización</i>	115
<i>Evaluación y seguimiento</i>	115
ARTÍCULO PARA LA PUBLICACIÓN	115
RESUMEN	117
INTRODUCCIÓN	117
MARCO TEÓRICO	118
<i>¿Qué es la visión artificial?</i>	119
<i>Redes neuronales convolucionales</i>	119
<i>YOLO ("Solo miras una vez")</i>	121
<i>OpenCV (Biblioteca de visión artificial de código abierto)</i>	122
<i>MTCNN</i>	123
<i>Evento sospechoso</i>	123
<i>TWILIO</i>	124
<i>Estándar DORI</i>	125
METODOLOGÍA DE LA INVESTIGACIÓN	125
RESULTADOS Y ANÁLISIS.	128
CONCLUSIONES	129
REFERENCIAS	130
MANUAL DE PROGRAMADOR	132
MANUAL DE USUARIO	237
GLOSARIO	290
GLOSARIO	291
BIBLIOGRAFÍA	293
ANEXOS	297

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1 - ARQUITECTURA BASE DE YOLO	19
ILUSTRACIÓN 2 - COMPARACIÓN DE RENDIMIENTO	22
ILUSTRACIÓN 3 - VALORES DE ACUERDO CON EL MODELO	23
ILUSTRACIÓN 4 - ARQUITECTURA YOLOV8 DE ACUERDO RANGEKING	23
ILUSTRACIÓN 5 - DIAGRAMA DE REQUERIMIENTOS	39
ILUSTRACIÓN 6 - SELECCIÓN DE TECNOLOGÍAS	40
ILUSTRACIÓN 7 - ARQUITECTURA DEL SISTEMA	41
ILUSTRACIÓN 8 - PROCESO DE DETECCIÓN DE EVENTOS	42
ILUSTRACIÓN 9 - COMPARACIÓN DE ALGORITMOS DE DETECCIÓN	43
ILUSTRACIÓN 10 - DIAGRAMA DEL PROTOTIPO DE SISTEMA DE VIDEOVIGILANCIA AUTOMATIZADO POR MEDIO DE VISIÓN ARTIFICIAL PARA ITCA-FEPADE	47
ILUSTRACIÓN 11 - DIAGRAMA DE BASE DE DATOS	48
ILUSTRACIÓN 12 - DICCIONARIO DE TABLA ROLE	49
ILUSTRACIÓN 13 - DICCIONARIO DE TABLA USERS	49
ILUSTRACIÓN 14- DICCIONARIO DE TABLA CAPTURE_SETTING	49
ILUSTRACIÓN 15 - DICCIONARIO DE TABLA TIME_SETTING	49
ILUSTRACIÓN 16 - DICCIONARIO DE TABLA CLASSES	50
ILUSTRACIÓN 17 - DICCIONARIO DE TABLA CAPTURED_EVENTS	50
ILUSTRACIÓN 18 - DICCIONARIO DE TABLA EVENT_DETAILS	50
ILUSTRACIÓN 19 - DIAGRAMA DE CASOS DE USO	52
ILUSTRACIÓN 20 - INICIO DE SESIÓN SISTEMA WEB	76
ILUSTRACIÓN 21 - GESTIÓN DE USUARIOS SISTEMA WEB	77
ILUSTRACIÓN 22 - GESTIÓN DE EVENTOS SISTEMA WEB	78
ILUSTRACIÓN 23 - CONFIGURACIÓN DE PARA CAPTURA DE IMÁGENES Y TIEMPO PARA REGISTRO DE NUEVOS EVENTOS	79
ILUSTRACIÓN 24 - INICIO DE SESIÓN SISTEMA DE VISIÓN ARTIFICIAL	80
ILUSTRACIÓN 25 - CAPTURA DE FOTOGRAFÍA INICIO DE SESIÓN SISTEMA DE VISIÓN ARTIFICIAL	80
ILUSTRACIÓN 26 - REGISTRO EN SISTEMA DE VISIÓN ARTIFICIAL	81
ILUSTRACIÓN 27 - INICIAR MONITOREO	81

ILUSTRACIÓN 28 - CRONOGRAMA DE ACTIVIDADES	96
ILUSTRACIÓN 29 - RESULTADOS	100
ILUSTRACIÓN 30 - CLASES E INSTANCIAS	100
ILUSTRACIÓN 31 - MATRIZ DE CONFUSIÓN	101
ILUSTRACIÓN 32 - RESULTADOS ENTRENAMIENTO MODELO YOLOV8 PERSONALIZADO ...	102
ILUSTRACIÓN 33 - INTERSECCIÓN SOBRE UNIÓN	103
ILUSTRACIÓN 34 - ENTRENAMIENTO 1	104
ILUSTRACIÓN 35 - ENTRENAMIENTO 2	105
ILUSTRACIÓN 36 - VALIDACIÓN 1	106
ILUSTRACIÓN 37 - VALIDACIÓN 2	106
ILUSTRACIÓN 38 - PRUEBA	107
ILUSTRACIÓN 39 - PRUEBA	107
ILUSTRACIÓN 40 - PRUEBA	108
ILUSTRACIÓN 41 - PRUEBA	108
ILUSTRACIÓN 169 PRUEBAS Y CAPTURAS DE EVENTOS	297

ÍNDICE DE TABLAS

TABLA 1 - INICIO DE SESIÓN	54
TABLA 2 - REGISTRAR USUARIO	56
TABLA 3 - MODIFICAR DATOS DE USUARIO	57
TABLA 4 - ELIMINAR USUARIO	59
TABLA 5 - VER DETALLES DE USUARIO	60
TABLA 6 - CONFIGURAR AJUSTES DE CAPTURA.....	62
TABLA 7 - CONSULTAR AJUSTES DE CAPTURA	63
TABLA 8 - CONSULTAR EVENTOS	63
TABLA 9 - MODIFICAR ESTADO DE UN EVENTO.....	65
TABLA 10 - VER DETALLES DE UN EVENTO	66
TABLA 11 - CARGA DE CONFIGURACIONES	67
TABLA 12 - DETECCIÓN DE EVENTOS	69
TABLA 13 - REGISTRO DE EVENTOS	70
TABLA 14 - GENERACIÓN DE ALERTAS POR CORREO ELECTRÓNICO	72
TABLA 15 - GENERACIÓN DE ALERTAS POR SMS	74
TABLA 16 - ACTIVAR ALARMA	75
TABLA 17 - COSTO DE PERSONAL	85
TABLA 18 - COSTO DE HARDWARE Y SOFTWARE	86
TABLA 19 - DETALLE DE SERVICIO	89
TABLA 20 - PROYECCIÓN	90
TABLA 21 - REQUISITOS OPERATIVOS	91
TABLA 22 - EVALUACIÓN DE RECURSOS	92
TABLA 23 - SELECCIÓN DE EQUIPOS	92
TABLA 24 - INSTALACIÓN Y CONFIGURACIÓN DE EQUIPOS	93
TABLA 25 - FASE DE OPERACIÓN	94

Resumen

Carrera: Ingeniería en Desarrollo de Software.

Título de la Tesis: Prototipo de sistema de videovigilancia automatizado por medio de visión artificial para ITCA-FEPADE.

Autores:

Edgardo Antonio Argueta Bautista

Jefferson Francisco Pineda Quintanilla

Kevin Rodrigo Flores Acosta

Nicolás Gabriel Escobar Pérez

Resumen :

En los últimos años el índice de delincuencia en nuestro país ha ido en aumento. Por lo que garantizar la seguridad de las personas y a sus bienes se ha convertido en una labor primordial y cada vez más complicada.

La vigilancia es muy importante para mantener seguros lugares como los estacionamientos. Las personas requieren soluciones más avanzadas para prevenir situaciones peligrosas. Aquí es donde entra en juego la visión artificial ya que ayuda en gran manera a estar siempre pendiente de los lugares en donde se implementa, es ideal un software capaz de detectar acciones sospechosas en tiempo real, emitiendo alertas en el momento exacto de la acción.

El enfoque del proyecto es desarrollar un prototipo de sistema de visión artificial basado en redes neuronales para la videovigilancia de los estacionamientos del ITCA-FEPADE. Se

plantea la posibilidad de detectar acciones sospechosas y a su vez emitir alertas que nos indiquen que algo anómalo está sucediendo, dichas alertas deben llegar a las personas competentes mediante correo electrónico, mediante alarmas físicas y mediante MSM.

El proyecto brinda a ITCA-FEPADE una solución muy innovadora y a su vez útil para combatir la delincuencia que tanto asecha a la comunidad y a al país en general. Para dicho sistema se implementará el uso de tecnologías nuevas y librerías para poder gestionar y automatizar en gran manera los eventos.

Se ha implementado el uso de librerías tales como OpenCV Python, Redes neuronales convolucionales, TensorFlows entre otras tecnologías que ayudan en el desarrollo del sistema.

Introducción

En la actualidad, la vigilancia es importante para mantener seguros los lugares, como los estacionamientos. La gente está buscando cada vez más soluciones avanzadas para prevenir cualquier situación riesgosa. Ahí es donde entra la visión artificial, una tecnología increíble que puede analizar y procesar automáticamente imágenes y videos para identificar cosas y eventos importantes. Es súper prometedora y ayuda a mantener los espacios bajo control.

El enfoque del innovador proyecto es el desarrollo de un prototipo de sistema de visión artificial basado en redes neuronales para la videovigilancia de los estacionamientos de ITCA-FEPADE. El objetivo principal es brindar una solución innovadora y efectiva que no solo detecte movimientos sospechosos e identifique personas, sino que también advierta de eventos sospechosos en tiempo real. Esto mejorará significativamente la seguridad y el control del entorno del estacionamiento.

Como lenguaje de programación se ha utilizado Python ya que junto con la biblioteca OpenCV y las redes neuronales convolucionales (CNN) se podrá implementar sistemas de visión artificial. Se opto por trabajar con Python y OpenCV porque pueden procesar imágenes y videos de manera eficiente además por la amplia documentación de estos y las redes neuronales convolucionales se ha utilizado para entrenar el modelo que permitirá detectar y reconocer objetos en las imágenes capturadas por la cámara de videovigilancia.

Se implemento un sistema de alerta que es capaz de enviar notificaciones a través de correo electrónico, SMS y activar alertas físicas mediante una alarma programada con Arduino. Estas alertas se activan cuando se detectan eventos potencialmente sospechosos en el entorno de videovigilancia. Asimismo, se ha desarrollado una aplicación de vigilancia utilizando Django, un marco de desarrollo web basado en Python. Esta aplicación permite visualizar y gestionar los eventos registrados por el sistema de videovigilancia.

Cabe destacar la importancia de seleccionar cuidadosamente el equipo a utilizar. Para asegurar la selección anterior, se ha apegado al estándar DORI (Distancia, Observación, Reconocimiento e Identificación). Este estándar ayuda a definir los criterios y parámetros que permiten evaluar la calidad y capacidad de los sistemas de videovigilancia.

Se ha utilizado YOLO (You Only Look Once) una herramienta que proporciona un conjunto avanzado de algoritmos que brindan detección y clasificación de objetos de forma eficiente y precisa en tiempo real.

El objetivo de este innovador trabajo de desarrollar un prototipo de sistema de visión artificial basado en redes neuronales para la videovigilancia del estacionamiento de ITCA-FEPADE. Haciendo uso de algoritmos de detección y clasificación de objetos, el sistema permite la detección de movimientos sospechosos en tiempo real.

CAPÍTULO I

ESTADO ANTERIOR

Descripción del problema

En los últimos años el índice de delincuencia en el país ha ido en aumento. Según la fuente periodística Diario El Salvador Desde 2018 al 2021, hubo 2,037 salvadoreños acusados del delito de robo de vehículos automotores, según el ministerio público. Se ha pasado de 700 casos a 400 por año, en promedio¹.

En El Salvador ha sido recurrente que grupos delincuenciales hurten o roben vehículos con los que posteriormente cometen otros hechos delictivos y pese a que estos casos han disminuido desde 2019, la fiscalía general de la República (FGR) detalla que al menos 1 persona ha sido procesada cada día por dicho delito en los últimos cuatro años, de acuerdo con (Diario el salvador, Cecilia Fuentes (2022)).

Según los datos del ministerio público, San Salvador y La Libertad son donde se registra el mayor número de denuncias, con 1,359, es decir, el 60 % de los casos. Cabe mencionar que en ambos departamentos es donde hay más vehículos matriculados.

El problema que aborda este proyecto es la necesidad de mejorar el sistema de videovigilancia en el estacionamiento de la institución ITCA-FEPADE, ubicada en El Salvador. Actualmente, la vigilancia en el estacionamiento se basa en métodos convencionales que presentan limitaciones en términos de eficiencia, capacidad de análisis y respuesta en tiempo real ante eventos no deseados. El estacionamiento de la institución ITCA-

¹ Abigail Parada. (2022). Las colonias se han vuelto escenarios de delitos comunes. Diario El Salvador. <https://www.elsalvador.com/noticias/nacional/colonias-escenario-delitos-san-salvador/1054546/2023/>

FEPADE es un área crítica que requiere una atención especial en cuanto a la seguridad y el monitoreo.

La solución propuesta fue desarrollar un sistema de visión artificial para el estacionamiento de ITCA-FEPADE automatizado e inteligente con la finalidad de brindar seguridad en tiempo real y de manera eficiente, mejorando así la seguridad y la capacidad de respuesta en situaciones de emergencia.

Se ha usado Python, OpenCV y redes neuronales convolucionales (CNN) para desarrollar un prototipo de sistema que pueda detectar los eventos de manera efectiva.

Todas estas librerías o bibliotecas previamente mencionadas permiten a desarrolladores de software crear aplicativos que puedan procesar imágenes y vídeos en tiempo real, extrayendo información útil para poder ser utilizada con diferentes fines. En este caso, se ha enfocado principalmente en la detección de objetos y eventos.

Se espera mejorar significativamente la seguridad y monitoreo de los estacionamientos del ITCA-FEPADE al brindar herramientas efectivas y confiables para prevenir y responder a situaciones imprevistas. Esto permite a la institución brindar un entorno más seguro y tranquilo para sus usuarios, ayudando a proteger sus activos y la tranquilidad de la comunidad académica.

Justificación

El desarrollo de un sistema de videovigilancia automatizado basado en visión artificial para el estacionamiento de ITCA-FEPADE en El Salvador se justifica por las siguientes razones:

Mejorar la seguridad: Integrar un sistema de videovigilancia permite incrementar exponencialmente los niveles de seguridad del estacionamiento de ITCA-FEPADE, ya que contará con detección temprana de eventos no deseados, entre los cuales se tienen robo o daño de vehículos lo cual a su vez permite una respuesta rápida y eficaz para poder controlar el riesgo y proteger los activos de los usuarios.

Prevenir la actividad delictiva: Contar un sistema de videovigilancia automatizado actúa como una medida disuasoria para actividades delictivas. Las cámaras de seguridad y la capacidad de detectar automáticamente eventos sospechosos disuaden a los posibles delincuentes y reducen el robo, el vandalismo.

Mejorar la detección de eventos sospechosos: Las herramientas que proporciona la visión artificial, ayudan a detectar eventos sospechosos en tiempo real dando la posibilidad de emitir alertas en el momento exacto en el que sucede la acción sospechosa. Estos beneficios son necesarios para aumentar la eficiencia de la toma de decisiones, para garantizar un control efectivo del estacionamiento y mejorar la seguridad.

Cumplir el estándar DORI: Al implementar el estándar DORI (Distancia, Observación, Reconocimiento e Identificación), se garantizará la calidad del equipo, lo que conlleva una buena detección de acciones o eventos sospechosos.

En resumen, la implementación de un sistema de videovigilancia automatizado basado en visión artificial para el estacionamiento de ITCA-FEPADE en El Salvador justifica la necesidad de mejorar la seguridad, disuadir actividades delictivas, optimizar los recursos humanos, aumentar la eficiencia en la detección de eventos y cumplir con los estándares de calidad.

Este proyecto ha sentado sus bases mediante el uso del lenguaje de programación Python, el cual es ampliamente utilizado para proyectos de inteligencia artificial, mediante un enfoque práctico aplicado en proyectos reales, se han definido las bases para la creación del prototipo de software que realmente contribuya a mejorar la seguridad en estacionamientos.

La visión artificial permite alertar en tiempo real mediante una alarma y notificación para evitar robos o hurtos de vehículos en estacionamiento del local o negocio, de esta forma alertar al personal de seguridad y poder tomar acciones tempranas para evitar este tipo de sucesos.

De igual modo, el uso del lenguaje Python, da acceso a un amplio catálogo de librerías y frameworks que pueden ser utilizados para una diversidad de soluciones, como el framework Django, que permite crear plataformas para la web, lo cual se implementara en el proyecto para crear el portal desde el que realizaran el monitoreo los usuarios del sistema.

Marco teórico

El proyecto de videovigilancia automatizada para el estacionamiento de ITCA-FEPADE en El Salvador se basa en un conjunto de tecnologías clave para su desarrollo. Se ha utilizado OpenCV, librería de visión artificial y procesamiento de imágenes, que proporciona una amplia gama de funciones y algoritmos para la detección de objetos, reconocimiento facial y manipulación de imágenes y videos, se han empleado redes neuronales convolucionales (CNN) para el reconocimiento de objetos en tiempo real, aprovechando la capacidad de estas redes para aprender y extraer características relevantes de las imágenes.²

Se utilizó Django, un framework de desarrollo web en Python que facilita la creación de interfaces interactivas, el manejo de formularios y la interacción con la base de datos. Esta combinación de tecnologías permite una implementación eficiente y precisa del sistema de videovigilancia, asegurando una detección efectiva de eventos, notificaciones por correo electrónico y SMS con una plataforma web intuitiva para el monitoreo y control de la seguridad en el estacionamiento.³

El proyecto de videovigilancia automatizada para el estacionamiento de ITCA-FEPADE en El Salvador se apoya en un conjunto de tecnologías avanzadas, Se utiliza YOLO (You Only Look Once), que implementa una red neuronal convolucional de detección

² OpenCV team. (2023). ABOUT. Open CV. <https://opencv.org/about/>

³ Django Software Foundation and individual contributors. (2023). Django. Django. <https://www.djangoproject.com/>

de objetos en tiempo real, con las que se puede identificar de forma precisa y eficiente eventos de interés en las imágenes capturadas por la cámara de videovigilancia.⁴

OpenCV Python es una poderosa librería de visión artificial que se integra con YOLO, facilitando el procesamiento de imágenes y videos, la extracción de características y el seguimiento de objetos.

El sistema utiliza como modelo principal YOLO v8 el cual permite entrenar un modelo personalizado que pueda predecir y emitir un valor de confianza estimado para enviar una alerta al usuario a través de correo electrónico y mediante SMS, de esta forma poder en pantalla indicar mediante un recuadro donde se detectó el objeto o la acción sospechosa, por cada acción se almacenara un registro en base de datos el cual puede visualizarse en un sistema web.

Se evaluaron soluciones de videovigilancia actuales para determinar que hardware es necesario adquirir para la implementación del proyecto, se debe tomar en cuenta el estándar DORI utilizado en sistemas CCTV, el cual permite clasificar el nivel de detalle que es posible obtener desde una cámara, el cual permite trazar objetivos alcanzables y realistas acorde al equipo que se tenga disponible.

Se hace uso de procesadores de NVIDIA utilizando CUDA, y tensores con Pytorch para lograr un mejor rendimiento usando la GPU del equipo.

⁴ Ultralytics 2023. (2023). Yolo v8. Ultralytics YOLOv8. <https://ultralytics.com/yolov8>

¿Qué es la visión artificial?

La visión artificial es un campo de la inteligencia artificial (IA) que permite a los ordenadores y sistemas extraer información significativa a partir de imágenes digitales, videos y otras entradas visuales, y tomar medidas o realizar recomendaciones en función de esa información. Si la IA permite a los ordenadores pensar, la visión artificial les permite ver, observar y comprender.⁵

La visión artificial funciona de manera muy similar a la visión humana, excepto que los humanos tienen una ventaja inicial. La vista humana tiene la ventaja de toda una vida de contexto para entrenar cómo distinguir los objetos, a qué distancia están, si se están moviendo y si hay algo mal en una imagen.⁶

La visión artificial entrena a las máquinas para realizar estas funciones, pero tiene que hacerlo en mucho menos tiempo con cámaras, datos y algoritmos en lugar de retinas, nervios ópticos y una corteza visual. Como un sistema entrenado para inspeccionar productos o ver un activo de producción puede analizar miles de productos o procesos por minuto, detectando defectos o problemas imperceptibles, puede superar rápidamente las capacidades humanas.⁷

⁵ Bernard Marr . (2019). 7 Amazing Examples Of Computer And Machine Vision In Practice. FORBES. <https://www.forbes.com/sites/bernardmarr/2019/04/08/7-amazing-examples-of-computer-and-machine-vision-in-practice/?sh=5915a2d81018>

⁶ Bernard Marr . (2019). 7 Amazing Examples Of Computer And Machine Vision In Practice. FORBES. <https://www.forbes.com/sites/bernardmarr/2019/04/08/7-amazing-examples-of-computer-and-machine-vision-in-practice/?sh=5915a2d81018>

⁷ Bernard Marr . (2019). 7 Amazing Examples Of Computer And Machine Vision In Practice. FORBES. <https://www.forbes.com/sites/bernardmarr/2019/04/08/7-amazing-examples-of-computer-and-machine-vision-in-practice/?sh=5915a2d81018>

La visión artificial es una disciplina que se centra en el desarrollo de sistemas y algoritmos que permiten a las máquinas obtener, procesar y comprender información visual del entorno de manera similar a la forma en que los seres humanos lo hacen. Se basa en la utilización de técnicas y algoritmos de procesamiento de imágenes para adquirir, analizar y extraer información útil de imágenes o secuencias de imágenes.⁸

La visión artificial busca simular la capacidad de percepción visual humana, permitiendo a las máquinas interpretar y comprender el contenido visual de una manera automatizada. Esto implica el desarrollo de algoritmos y métodos para realizar tareas como detección de objetos, reconocimiento facial, seguimiento de objetos en movimiento, segmentación de imágenes, medición de distancias, entre otras aplicaciones.⁹

Los sistemas de visión artificial suelen utilizar cámaras u otros dispositivos de adquisición de imágenes para capturar el entorno visual. Luego, a través del procesamiento de imágenes, se extraen características relevantes, se realizan análisis y se toman decisiones basadas en la información visual obtenida.¹⁰

La visión artificial tiene aplicaciones en diversas industrias y sectores, como la industria automotriz, la medicina, la seguridad, el control de calidad, la robótica y la realidad aumentada, entre otros. Su objetivo es mejorar y automatizar procesos que requieren de la interpretación y análisis de

⁸ IBM. (2023). ¿Qué es la visión artificial?. IBM. <https://www.ibm.com/mx-es/topics/computer-vision>

⁹ IBM. (2023). ¿Qué es la visión artificial?. IBM. <https://www.ibm.com/mx-es/topics/computer-vision>

¹⁰ IBM. (2023). ¿Qué es la visión artificial?. IBM. <https://www.ibm.com/mx-es/topics/computer-vision>

información visual, brindando soluciones eficientes y precisas.

11

Redes neuronales convolucionales

Una red neuronal convolucional (CNN) es un tipo de modelo de aprendizaje automático que se usa comúnmente en tareas de visión por computadora, también conocida como visión por computadora, reconocimiento de imágenes o procesamiento de imágenes.

Los ordenadores usan filtros o lentes para ver las diferentes características de las redes neuronales convolucionales: uno ve los bordes diagonales, otro los colores, etc. Funciona pasando los filtros por toda la imagen, escaneándola, para luego definirla y clasificarla.

Las CNN son una serie de redes que fueron creadas pensando en cómo funciona el cerebro, capaces de aprender en los diferentes niveles de abstracción: en la primera capa se diferencian formas simples, colores o bordes; en la siguiente se pueden distinguir combinaciones de bordes y colores; mientras que la última capa se fija en la forma con el fin de conseguir averiguar qué es exactamente.

Las CNN están diseñadas para procesar datos en forma de cuadrículas o matrices, como imágenes. A diferencia de las redes neuronales tradicionales, las CNN aprovechan la estructura y las

¹¹ IBM. (2023). ¿Qué es la visión artificial?. IBM. <https://www.ibm.com/mx-es/topics/computer-vision>

propiedades espaciales de las imágenes para extraer características relevantes.

Las principales características de las redes neuronales convolucionales son las siguientes¹²:

- **Capas convolucionales:** Utilizan operaciones de convolución para aplicar filtros o núcleos a la imagen de entrada, extrayendo características como bordes, texturas o patrones relevantes.
- **Capas de pooling:** Reducen la dimensionalidad de las características obtenidas, disminuyendo la cantidad de parámetros y manteniendo la información más relevante.
- **Capas totalmente conectadas:** En la etapa final de la red, se utilizan capas completamente conectadas para realizar la clasificación o la predicción final basada en las características extraídas.

La arquitectura de una CNN generalmente consta de varias capas convolucionales y de pooling, seguidas de una o varias capas completamente conectadas. A medida que los datos fluyen a través de la red, las capas convolucionales aprenden a reconocer características cada vez más complejas, lo que permite una representación jerárquica y progresiva de la imagen.

Una de las principales ventajas de las CNN es su capacidad para aprender de forma automática características relevantes de las imágenes, sin necesidad de una extracción manual de características. Además, las CNN pueden manejar eficientemente grandes conjuntos de datos y son robustas frente a las

¹² datascientest. (2023). Convolutional Neural Network : definición y funcionamiento. datascientest. <https://datascientest.com/es/convolutional-neural-network-es>

variaciones en la posición, tamaño y orientación de los objetos en las imágenes.

Las redes neuronales convolucionales han demostrado ser muy exitosas en tareas como la clasificación de imágenes, la detección de objetos, el reconocimiento facial, la segmentación de imágenes y el etiquetado automático, entre otras aplicaciones relacionadas con el procesamiento de imágenes y visión por computadora.¹³

YOLO ("Solo miras una vez")

YOLO (You Only Look Once) es un popular algoritmo de detección de objetos en tiempo real basado en redes neuronales convolucionales. A diferencia de los métodos tradicionales que requieren secuencias separadas de pasos para reconocer y clasificar objetos, YOLO aborda este problema de manera integral, realizando ambos procesos en un solo paso.¹⁴

En el campo de la detección de objetos en video e imágenes en tiempo real, existen otros enfoques que emplean técnicas como ventanas deslizantes o pirámides de imágenes para buscar objetos. YOLO es un algoritmo de detección de objetos de video e imágenes en tiempo real de código abierto. YOLO divide la imagen en cuadrículas y realiza la detección de objetos en cada cuadrícula.

¹³ NA8. (2018). ¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador. aprendemachinelearning. <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>

¹⁴ Albert Soto i Serrano. YOLO Object Detector for Onboard Driving Images. [Tesis doctoral o de maestría, UNIVERSITAT AUTONOMA DE BARCELONA]. https://ddd.uab.cat/pub/tfg/2017/tfg_71066/paper.pdf.

YOLO es capaz de localizar y clasificar objetos en una imagen de forma rápida y precisa. Utiliza una red neuronal convolucional que divide la imagen en una cuadrícula y asigna cajas delimitadoras (bounding boxes) a las regiones donde se encuentra un objeto. Cada caja delimitadora contiene información sobre las coordenadas (x, y, ancho y altura) y la confianza de que contiene un objeto.

En lugar de generar propuestas de regiones y aplicar clasificación por separado, YOLO aplica una sola red neuronal a la imagen completa y predice simultáneamente múltiples cajas delimitadoras y las probabilidades de clase asociadas. Permitiendo que sea extremadamente rápido y eficiente en la detección de objetos en tiempo real.

YOLO utiliza una red neuronal convolucional para predecir las coordenadas del cuadro delimitador, la clase del objeto y la probabilidad de detección de cada objeto en la imagen. Esto se hace en una sola pasada de la red neuronal, lo que permite que YOLO sea muy rápido y eficiente en términos de recursos computacionales.

Debido a su velocidad y eficiencia, YOLO se ha convertido en una de las tecnologías de detección de objetos en tiempo real más populares y se utiliza en diversas aplicaciones, como vigilancia y seguridad, conducción autónoma, robótica y realidad aumentada.

Con YOLO se ha entrenado un modelo personalizado que permite predecir eventos futuros utilizando un dataset de imágenes recopiladas dentro de ITCA-FEPADE el cual se puede entrenar y validar utilizando una muestra de estas.

Ultralytics es una librería de Python que permite realizar la implementación de YOLO a través del modelo YOLOv8, el cual admite varios modos que se pueden usar para realizar diferentes tareas.¹⁵ Estos modos son:

- **Entrenar:** para entrenar un modelo YOLOv8 en un conjunto de datos personalizado.
- **Val:** para validar un modelo YOLOv8 después de haberlo entrenado.
- **Predecir:** Para hacer predicciones usando un modelo YOLOv8 entrenado en nuevas imágenes o videos.
- **Exportar:** para exportar un modelo YOLOv8 a un formato que se puede usar para la implementación.
- **Track:** Para rastrear objetos en tiempo real usando un modelo YOLOv8.
- **Benchmark:** para comparar la velocidad y la precisión de las exportaciones de YOLOv8 (ONNX, TensorRT, etc.).

YOLO ofrece modelos preentrenados listos para utilizarse, bajo la licencia **AGPL-3.0** sin embargo, el proyecto utilizara un modelo personalizado para definir acciones permitidas entrenadas usando conda, Pytorch y CUDA de NVIDIA para el procesamiento de datos empleando la GPU de la computadora.

¿Como trabaja YOLO?

YOLO nació trabajando sobre una red neuronal convolucional CNN, hasta la fecha de creación del proyecto se encuentra en la

¹⁵ Albert Soto i Serrano. YOLO Object Detector for Onboard Driving Images. [Tesis doctoral o de maestría, UNIVERSITAT AUTONOMA DE BARCELONA]. https://ddd.uab.cat/pub/tfg/2017/tfg_71066/paper.pdf.

versión 8 con ultralytics como desarrollador la primera versión funcionaba sobre el siguiente modelo¹⁶.

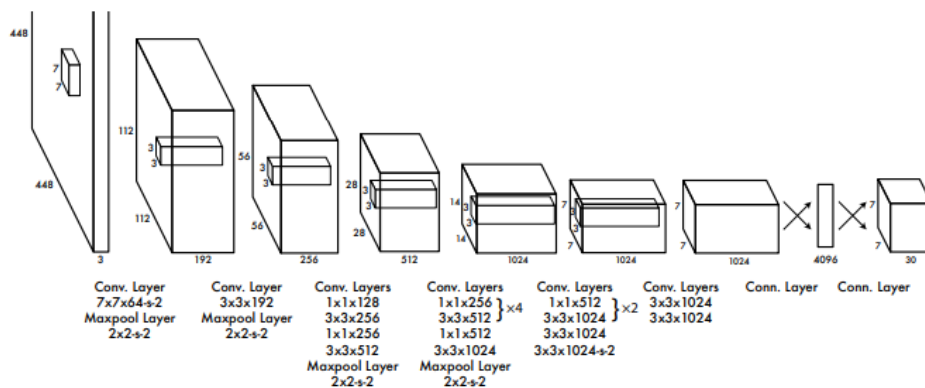


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

Ilustración 1 - Arquitectura base de YOLO

Este modelo usa Imagenet para el entrenamiento de las primeras 20 capas de convolución y el agrupamiento temporal promedio y compa completamente conectada, Este modelo preentrenado se convierte para realizar la detección. La última capa conectada predice las probabilidades de clase y las coordenadas del cuadro delimitador.

YOLO divide una imagen de entrada en una cuadrícula $S \times S$. Si el centro de un objeto cae en una celda de cuadrícula, esa celda de cuadrícula es responsable de detectar ese objeto. Cada celda de la cuadrícula predice cuadros delimitadores B y puntuaciones de confianza para esos cuadros. Estos puntajes de confianza reflejan qué tan seguro está el modelo de que la caja contiene un objeto y qué tan precisa cree que es la caja predicha.

¹⁶ Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

YOLO predice múltiples cuadros delimitadores por celda de cuadrícula. En el momento del entrenamiento, solo queremos que un predictor de cuadro delimitador sea responsable de cada objeto. YOLO asigna un predictor para que sea "responsable" de predecir un objeto en función de qué predicción tiene el IOU actual más alto con la verdad del terreno. Esto conduce a la especialización entre los predictores de cuadro delimitador. Cada predictora mejora en la previsión de ciertos tamaños, relaciones de aspecto o clases de objetos, lo que mejora la puntuación general de recuperación.

YOLO ha sido nutrido por la comunidad de visión artificial desde su primer lanzamiento en 2015 por Joseph Redmond. En los primeros días (versiones 1-4), YOLO se mantuvo en código C en un marco de aprendizaje profundo personalizado escrito por Redmond llamado Darknet.

YOLO fue lanzado el 2015 por Joseph Redmon, Fue desarrollado en código C en un marco de aprendizaje profundo escrito por Redmond llamado Darknet.

El autor de YOLOv8, Glenn Jocher de Ultralytics, siguió el repositorio de YOLOv3 en PyTorch (un marco de aprendizaje profundo de Facebook). A medida que mejoraba el entrenamiento en el repositorio oculto, Ultralytics finalmente lanzó su propio modelo: YOLOv5.

YOLOv5 se convirtió rápidamente en el repositorio SOTA del mundo dada su estructura Pythonic flexible. Esta estructura permitió a la comunidad inventar nuevas mejoras de modelado y compartirlas rápidamente en el repositorio con métodos PyTorch similares.

Junto con los sólidos fundamentos del modelo, los mantenedores de YOLOv5 se han comprometido a respaldar un ecosistema de software saludable en torno al modelo. Ellos solucionan problemas activamente y aumentan las capacidades del repositorio según lo exige la comunidad.

En los últimos dos años, varios modelos se derivaron del repositorio YOLOv5 PyTorch, incluidos Scaled-YOLOv4, YOLOR y YOLOv7. Otros modelos surgieron en todo el mundo a partir de sus propias implementaciones basadas en PyTorch, como YOLOX y YOLOv6. En el camino, cada modelo YOLO ha traído nuevas técnicas SOTA que continúan impulsando la precisión y eficiencia del modelo.

Durante los últimos seis meses, Ultralytics trabajó en la investigación de la versión SOTA más reciente de YOLO, YOLOv8. YOLOv8 se lanzó el 10 de enero de 2023 de acuerdo con Jacob Solawetz, Franceso¹⁷.

Arquitectura YOLOv8

YOLOv8 aún no tiene un artículo publicado, por lo que carecemos de información directa sobre la metodología de investigación directa y los estudios realizados durante su creación

Es una modelo basada en las últimas versiones YOLOv5, YOLOv6, YOLOv7 e introduce nuevas características.

¹⁷ Solawetz, J. (2023, enero 11). What is YOLOv8? The Ultimate Guide. Roboflow Blog. <https://blog.roboflow.com/whats-new-in-yolov8/>

En la actualidad YOLO trabaja con una arquitectura de red CNN EfficientNet y sus diferentes variantes¹⁸.

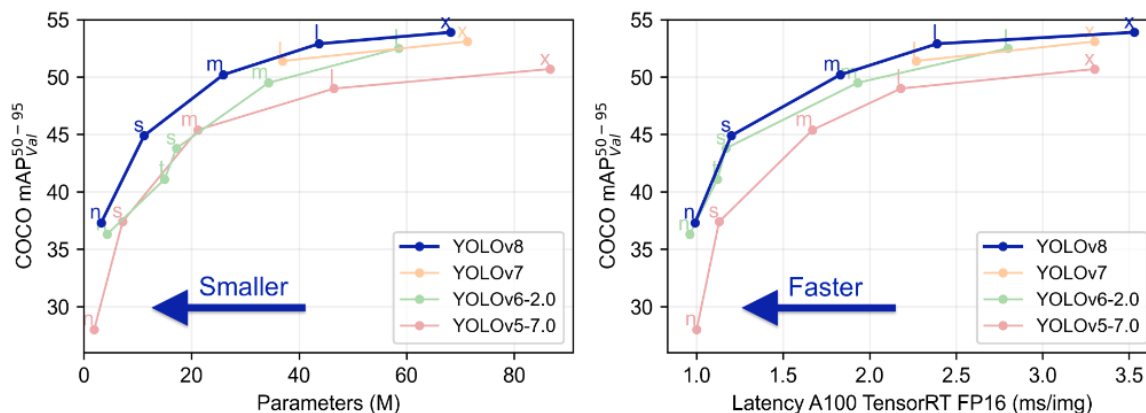


Ilustración 2 - Comparación de rendimiento

Al comparar el rendimiento de las diferentes versiones de YOLO y tamaños de modelos en el conjunto de datos de COCO, queremos comparar diferentes métricas.

Rendimiento: Precisión media media (mAP)

Velocidad: Velocidad de la inferencia (En fps)

Calcular (costo): el tamaño del modelo en FLOP y parámetros

Para la comparación de detección de objetos de los 5 tamaños de modelo, el modelo YOLOv8m logró un mAP del 50,2 % en el conjunto de datos COCO, mientras que el modelo más grande, YOLOv8x, logró un 53,9 % con más del doble de parámetros.

¹⁸ Xie, Q., Luong, M.-T., Hovy, E., & Le, Q. V. (2020). Self-training with noisy student improves ImageNet classification. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Model	size (pixels)	mAP ^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

- mAP^{val} values are for single-model single-scale on [COCO val2017](#) dataset. Reproduce by `yolo val detect data=coco.yaml device=0`
- Speed averaged over COCO val images using an [Amazon EC2 P4d](#) instance. Reproduce by `yolo val detect data=coco128.yaml batch=1 device=0|cpu`

Ilustración 3 - Valores de acuerdo con el modelo

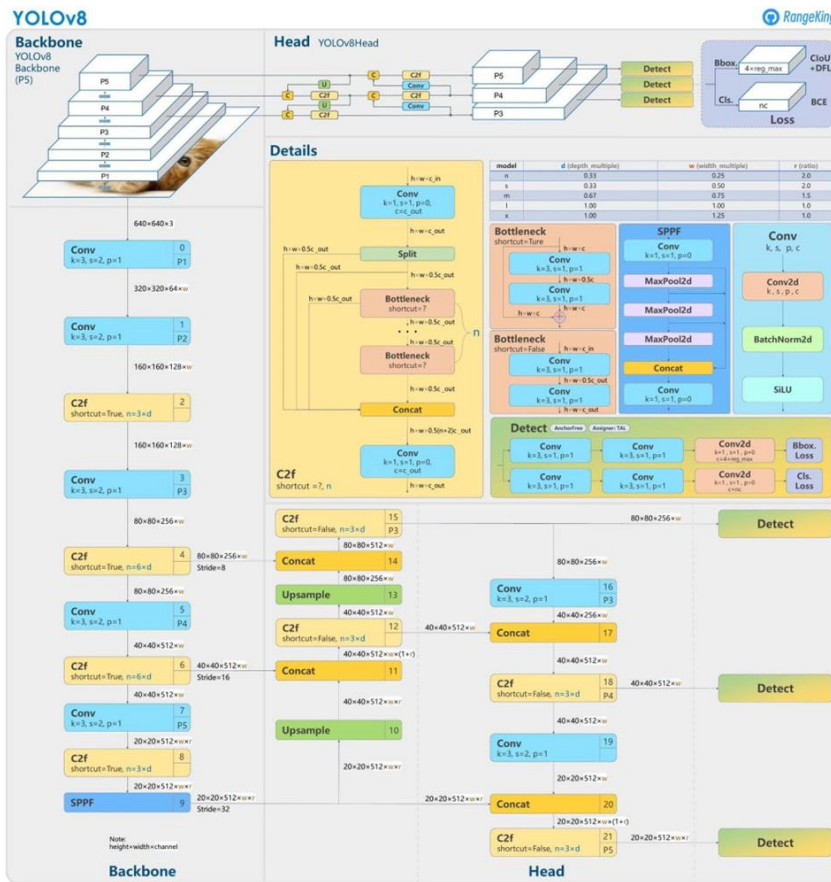


Ilustración 4 - Arquitectura YOLOv8 de acuerdo RangeKing

Los cambios más significativos respecto a otras versiones:

1. Reemplace el C3 módulo con el C2f módulo
2. Reemplace el primero 6x6 Conv con 3x3 Conv en el Backbone
3. Eliminar dos Convs (No.10 y No.14 en la configuración de YOLOv5)
4. Reemplace el primero 1x1 Conv con 3x3 Conv en el Bottleneck
5. Use la cabeza desacoplada y elimine la objectness rama, De acuerdo con RangeKing ¹⁹.

Utiliza detección sin anclaje y nuevas capas convolucionales para hacer predicciones más precisas²⁰.

OpenCV (Biblioteca de visión artificial de código abierto)

OpenCV (Biblioteca de visión por computadora de código abierto) es una biblioteca de procesamiento de imágenes y visión por computadora de código abierto que proporciona una amplia gama de funciones y algoritmos para manipular imágenes y videos. Originalmente desarrollado por Intel y ahora mantenido por la comunidad de código abierto.²¹

Esta es una biblioteca de código abierto que proporciona herramientas y algoritmos para la visión artificial y el procesamiento de imágenes. Desarrollado originalmente por Intel en 1999, desde entonces ha sido desarrollado por una gran

¹⁹ Ultralytics. (s. f.). Brief summary of YOLOv8 model structure · Issue #189 · ultralytics/ultralytics. GitHub.
<https://github.com/ultralytics/ultralytics/issues/189>

²⁰ Ultralytics: NEW - YOLOv8 🚀 in PyTorch > ONNX > CoreML > TFLite. (s/f)

²¹ OpenCV team. (2023). ABOUT. Open CV. <https://opencv.org/about/>

comunidad de desarrolladores. OpenCV está diseñado para proporcionar una infraestructura común para aplicaciones de visión artificial y acelerar la implementación de la visión artificial en productos comerciales. Un producto con licencia de Apache 2, OpenCV facilita a las empresas el uso y la modificación de su código.²²

La biblioteca contiene más de 2500 algoritmos de optimización, incluido un conjunto completo de algoritmos de visión artificial y aprendizaje automático clásicos y de última generación. Estos algoritmos se utilizan para la detección y el reconocimiento de rostros, el reconocimiento de objetos, la clasificación de acciones humanas en videos, el seguimiento del movimiento de la cámara, el seguimiento de objetos en movimiento, la extracción de modelos 3D de objetos, la extracción de nubes de puntos 3D de cámaras estéreo. Une imágenes para producir alta resolución. Obtenga imágenes de escenas completas, busque imágenes similares en bibliotecas de imágenes, use el flash para eliminar los ojos rojos en las imágenes, rastree el movimiento de los ojos, reconozca paisajes, establezca marcadores para superponer con realidad aumentada y mucho más.

OpenCV tiene más de 47 mil personas de usuario comunidad y número estimado de descargas que superan los 18 millones. Se utiliza ampliamente en empresas, grupos de investigación y organismos gubernamentales.

OpenCV está escrito en C++ y cuenta con interfaces y soporte para los lenguajes de programación más populares, como Python y

²² OpenCV team. (2023). ABOUT. Open CV. <https://opencv.org/about/>

Java, lo que permite su utilización en diferentes entornos de desarrollo. Es ampliamente utilizado en aplicaciones de visión por computadora, robótica, realidad aumentada, seguimiento de objetos, reconocimiento de patrones, entre otros.

Algunas características y funcionalidades clave de OpenCV incluyen²³:

- **Procesamiento de imágenes:** OpenCV ofrece una amplia gama de funciones para cargar, manipular y guardar imágenes en diferentes formatos. Puede realizar operaciones como ajuste de brillo y contraste, cambio de tamaño, recorte, filtrado, conversión de color y corrección de distorsiones.
- **Detección de objetos:** La biblioteca proporciona algoritmos para la detección de objetos, incluyendo la detección de rostros, detección de características específicas, detección de bordes y líneas, y detección de formas y contornos.
- **Seguimiento de objetos:** OpenCV ofrece algoritmos de seguimiento de objetos para rastrear y seguir el movimiento de objetos en secuencias de imágenes o videos.
- **Reconocimiento de patrones:** La biblioteca incluye algoritmos para el reconocimiento y clasificación de patrones, como reconocimiento facial, reconocimiento de escritura a mano, reconocimiento de objetos y reconocimiento de gestos.

²³ Rubén Crespo Cano. (2017). CURSO DE INTRODUCCIÓN A OPENCV Y PYTHON. [Tesis doctoral o de maestría, UNIVERSIDAD DE ALICANTE]. <https://github.com/rcrespocano/opencv-python/blob/master/opencv-and-python.ipynb>.

- **Calibración de cámaras:** OpenCV permite calibrar cámaras para corregir distorsiones y calcular parámetros intrínsecos y extrínsecos de la cámara.
- **Visión estéreo:** OpenCV proporciona herramientas para la visión estéreo, que permite la reconstrucción tridimensional a partir de imágenes estéreo y el cálculo de disparidad y mapas de profundidad.
- **Interoperabilidad:** OpenCV se puede integrar con otras bibliotecas y herramientas de procesamiento de imágenes y aprendizaje automático, como NumPy, scikit-learn y TensorFlow, lo que permite aprovechar las capacidades de estas bibliotecas en conjunto.

OpenCV es utilizado debido a las funciones que proporciona a los programadores, es muy potente y debido a eso su uso en el área de la visión artificial y procesamiento de imágenes está ganando mucho terreno.

MTCNN

MTCNN es una biblioteca de python (pip) escrita por el usuario de Github ipacz , que implementa el artículo de Zhang, Kaipeng et al. "Detección y alineación de caras conjuntas mediante redes convolucionales en cascada multitarea". *Cartas de procesamiento de señales IEEE* 23.10 (2016): 1499-1503. Ref. cruzada web.²⁴

²⁴ Paz Centeno, I. (s/f). mtcnn: MTCNN face detection implementation for TensorFlow, as a PIP package.

Las redes neuronales convolucionales en cascada multitarea, o MTCNN, han demostrado su eficacia en la detección de rostros en una variedad de puntos de referencia. También identifican puntos de referencia faciales como los ojos, la nariz y la boca.

El modelo está basado en hacer uso del aprendizaje multi-task para así dividir las tareas de detección de rostros y facciones, ya que al hacer esto en la forma multi-task se mejora la generalización al aprovechar la información específica del dominio contenido en las señales de entrenamiento de las tareas relacionadas, y esto lo hace entrenando las tareas en paralelo mientras usa una representación compartida de la información.²⁵

El primer paso antes de empezar las etapas es tomar la imagen y cambiar su tamaño a diferentes escalas para construir una pirámide de imágenes, que sea la entrada de la primera parte de la red.

La primera etapa consta de tener una red tipo fully convolutional network (FCN) se usa está en vez de una CNN porque no tiene una capa densa en su arquitectura. Esta red propuesta se utiliza para obtener los rostros y su respectiva bounding box. La salida de esta red da todos los posibles candidatos de los rostros.

En la segunda etapa de la red, se utiliza la salida de la primera etapa como entrada de una red CNN y a esta se le llamada "The Refine Network (R-Net)", esta reduce el número de posibles

²⁵ Vega, M. (2021, mayo 12). Aplicación de reconocimiento facial en tiempo real para identificar a alumnos de la licenciatura en ciencias de la computación de la Universidad de Sonora usando Pytorch y redes neuronales convolucionales en cascada y en multitarea (MTCNN). LCC-Unison. <https://medium.com/lcc-unison/aplicaci%C3%B3n-de-reconocimiento-facial-en-tiempo-real-para-identificar-a-alumnos-de-la-licenciatura-en-2e2b53870995>

candidatos, calibra las bounding box y se hace una supresión no máxima (NMS) para unir nuestros posibles candidatos. Como salida dice si la entrada es una cara o no.

Por ultima la tercer es muy similar a la R-Net, esta da como salida la posición de los ojos nariz y boca de los candidatos.

Esta librería permite en conjunto con open cv integrar un sistema de login y registro.

Evento sospechoso

Un evento sospechoso es cualquier acto o situación cuya naturaleza inusual o peligrosa despierte sospecha o temor. Un evento sospechoso en un sistema de videovigilancia es cualquier actividad o comportamiento que pueda indicar un accidente, una brecha de seguridad o una actividad delictiva.

Algunos ejemplos de eventos sospechosos pueden incluir:

- **Movimiento inusual:** Detección de movimientos que no siguen los patrones normales, como personas merodeando en áreas restringidas o comportamientos erráticos.
- **Comportamiento agresivo o violento:** Detección de acciones o gestos agresivos, como peleas, intimidación o actos de vandalismo.
- **Cambios repentinos en la actividad:** Identificación de situaciones en las que la actividad normal se ve interrumpida o alterada significativamente, como evacuaciones de emergencia, incendios o accidentes.

La definición de un evento sospechoso puede variar según el contexto y las políticas de seguridad de cada lugar. Para un sistema de videovigilancia automatizado, se pueden establecer

criterios y reglas específicas para definir qué se considera un evento sospechoso, y con base en esos criterios se generan las alertas correspondientes para tomar las acciones necesarias.

Para el proyecto de visión artificial se denomina "Evento sospechoso" a toda acción la cual genere incertidumbre sobre las intenciones de un individuo al acercarse a un vehículo que pueda generar un potencial robo o hurto. nos enfocaremos en los siguientes eventos:

La detección de un sujeto el cual se encuentre alrededor de un vehículo sin entrar a este o alejarse durante un tiempo determinado.

Reconocer que un sujeto que se aproxime a las puertas o ventanas de un vehículo lleve en sus manos alguna herramienta como algún destornillador, cuchillo, varilla, arma de fuego o incluso una roca.

Cuando un sujeto, posicione su cabeza o manos sobre alguna de las ventanas de un vehículo.

Estándar DORI

El estándar DORI (Distancia, Observación, Reconocimiento, Identificación) es una métrica utilizada en videovigilancia para evaluar y establecer los requisitos de calidad y rendimiento de los sistemas de cámaras de seguridad. Proporciona una guía para determinar la idoneidad de una cámara o sistema de videovigilancia en función de la distancia a la que se encuentra

el objeto de interés y los detalles que se pueden capturar en diferentes escenarios.²⁶

Cada una de las letras en DORI representa un aspecto específico:

- **Distancia (Distance):** Se refiere a la distancia entre la cámara de videovigilancia y el objeto o área de interés. La capacidad de capturar detalles precisos disminuye a medida que aumenta la distancia. El estándar DORI establece rangos de distancia para los cuales se espera que una cámara o sistema de videovigilancia proporcione niveles aceptables de detalle y reconocimiento.
- **Observación (Observation):** Se refiere a la capacidad de la cámara para observar y detectar objetos en movimiento. Esto implica la capacidad de detectar eventos o actividades de interés, como movimiento de personas, vehículos u otros objetos relevantes.
- **Reconocimiento (Recognition):** Se refiere a la capacidad de la cámara para reconocer y distinguir características básicas de los objetos, como la identificación de un rostro o la lectura de una matrícula de un automóvil.
- **Identificación (Identification):** Se refiere a la capacidad de la cámara para capturar suficientes detalles y características específicas de un objeto para permitir su identificación precisa. Esto implica la capacidad de leer letras o números pequeños, reconocer rasgos faciales distintivos u otros detalles que permitan una identificación clara.

²⁶ Argos.red. (2018). DORI EN CÁMARAS CCTV. argos. <https://argos.red/dori-en-camaras-cctv/>

El estándar DORI ayuda a los profesionales de la seguridad y a los instaladores de sistemas de videovigilancia a evaluar y seleccionar las cámaras adecuadas para diferentes escenarios y necesidades. Al considerar los requisitos de distancia, observación, reconocimiento e identificación, se puede determinar qué cámaras o sistemas son más apropiados para capturar los detalles necesarios en cada situación específica²⁷.

CUDA

CUDA son las siglas de Compute Unified Device Architecture (Arquitectura Unificada de Dispositivos de Cómputo) que hace referencia a una plataforma de computación en paralelo que incluye un compilador y un conjunto de herramientas de desarrollo creadas por Nvidia que permiten a los programadores usar una variación del lenguaje de programación C (CUDA C) para codificar algoritmos en GPU de Nvidia.

CUDA ® es una plataforma informática paralela y un modelo de programación inventado por NVIDIA. Permite aumentos drásticos en el rendimiento informático al aprovechar la potencia de la unidad de procesamiento de gráficos (GPU).

CUDA se desarrolló con varios objetivos de diseño en mente:

²⁷ Argos.red. (2018). DORI EN CÁMARAS CCTV. argos. <https://argos.red/dori-en-camaras-cctv/>

Proporcione un pequeño conjunto de extensiones para lenguajes de programación estándar, como C, que permiten una implementación sencilla de algoritmos paralelos. Con CUDA C/C++, los programadores pueden concentrarse en la tarea de paralelizar los algoritmos en lugar de dedicar tiempo a su implementación.

Admite computación heterogénea donde las aplicaciones usan tanto la CPU como la GPU. Las porciones en serie de las aplicaciones se ejecutan en la CPU y las porciones paralelas se descargan en la GPU. Como tal, CUDA se puede aplicar de forma incremental a las aplicaciones existentes. La CPU y la GPU se tratan como dispositivos separados que tienen sus propios espacios de memoria. Esta configuración también permite el cálculo simultáneo en la CPU y la GPU sin disputa por los recursos de memoria.

Las GPU compatibles con CUDA tienen cientos de núcleos que pueden ejecutar colectivamente miles de subprocesos informáticos. Estos núcleos tienen recursos compartidos que incluyen un archivo de registro y una memoria compartida. La memoria compartida en el chip permite que se ejecuten tareas paralelas en estos núcleos para compartir datos sin enviarlos por el bus de memoria del sistema. ²⁸

²⁸ NVIDIA.DOCS. <https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/index.html>

PYTORCH

PyTorch es una biblioteca de aprendizaje automático de código abierto basada en la biblioteca de Torch, utilizado para aplicaciones como visión artificial y procesamiento de lenguajes naturales, principalmente desarrollado por el Laboratorio de Investigación de Inteligencia Artificial de Facebook (FAIR). Es un software libre y de código abierto liberado bajo la Licencia Modificada de BSD. A pesar de que la interfaz de Python está más pulida y es el foco principal del desarrollo, PyTorch también tiene una interfaz en C++.

Un número de las piezas de software de Aprendizaje Profundo están construidas utilizando PyTorch, incluyendo Tesla Autopilot, Uber's Pyro, HuggingFace's Transformers, PyTorch Lightning, y Catalyst.

PyTorch proporciona dos características de alto nivel

Computación de tensores (como NumPy) con una aceleración fuerte a través de unidades de procesamiento gráfico (GPU).

Redes neuronales profundas construidas en un sistema de diferenciación automática de bases de datos.²⁹

²⁹ Adam Paszke, Wikipedia, <https://es.wikipedia.org/wiki/PyTorch>

Marco legal

El desarrollo e implementación del sistema de videovigilancia automatizada en el estacionamiento de ITCA-FEPADE debe cumplir con el marco legal y regulatorio vigente. A continuación, se detallan las principales normativas para tener en cuenta:

Ley de Protección de Datos Personales:

Art. 5 del Decreto Legislativo N°133.- "El tratamiento de los datos personales que precisen los prestadores de servicios de certificación y los prestadores de servicio de almacenamiento de documentos electrónicos se sujetarán a lo siguiente: 1) tendrán la prohibición de ceder los datos personales de los usuarios a terceros; 2) los titulares podrán solicitar la rectificación o cancelación de los datos personales, cuando éstos fueren inexactos o incompletos; y, 3) por lo tanto, las empresas tendrán la obligación de mantener la confidencialidad de los mismos y al deber de guardarlos. Obligaciones que en los términos jurídicos subsistirán aún después de finalizar las relaciones con el responsable del registro de datos y las empresas; observando, la falta de un componente que determine hasta donde pueden llegar la transferencia de datos personales, esto no imposibilita que estos proveedores estén sujetos a sanciones, multas y obligaciones a pesar de que sus servicios se encuentren

en la nube o con alguna codificación electrónica fuera del territorio nacional".³⁰

Esta ley regula todos los datos personales tratados en el territorio salvadoreño, tanto en el sector público y privado. Se exceptúa los datos crediticios, aquellos utilizados para el ejercicio del periodismo, datos de seguridad nacional, así como los que hayan sido disociados o anonimizados. Además, establece los principios y requisitos para el tratamiento adecuado de la información personal. se deben tomar medidas para garantizar la seguridad de los datos y respetar el derecho a la privacidad de las personas.

Estatutos locales:

Es importante revisar y seguir cualquier otra regulación a nivel local o institucional. Esto puede incluir políticas internas de ITCA-FEPADE relacionadas con la seguridad y privacidad de la información, así como requisitos específicos municipales o regionales para la instalación y operación de sistemas de videovigilancia.

³⁰ (S/f-c). Gob.sv. Ley de Protección de Datos Personales, de <https://www.transparencia.gob.sv/institutions/rree/documents/408850/download>

CAPÍTULO II

DESARROLLO DE LA INNOVACIÓN

Objetivo General

1. Desarrollar un prototipo de sistema de visión artificial basado en redes neuronales para la videovigilancia de estacionamientos de vehículos en el entorno de ITCA-FEPADE, con detección de movimientos sospechosos y reconocimiento de personas en tiempo real, que permita el almacenamiento y gestión de eventos capturados, el envío de alertas por correo electrónico, SMS y la integración de un sistema de alarmas.

Objetivos específicos

1. Desarrollar un sistema de detección de movimientos sospechosos basado en redes neuronales y el enfoque YOLO, que permita identificar de manera precisa y en tiempo real situaciones de riesgo en el estacionamiento de ITCA-FEPADE.
2. Crear un sistema de almacenamiento y gestión de eventos capturados, que permita el registro y la recuperación eficiente de los datos recopilados por el sistema de videovigilancia, facilitando la revisión y el análisis posterior de los eventos relevantes.
3. Emitir alertas por correo electrónico y SMS de forma automática a los encargados de seguridad o administradores, informando sobre eventos sospechosos detectados en tiempo real, para una rápida respuesta y toma de decisiones.

Metodología y estrategia

Se da a conocer la metodología y estrategia utilizada para el desarrollo del proyecto de videovigilancia en estacionamientos. Se describen las técnicas y métodos empleados, así como las tecnologías y herramientas seleccionadas. Se explica cómo estos enfoques fueron apropiados para alcanzar los objetivos establecidos, garantizando un sistema eficiente y efectivo de videovigilancia. También se destaca que el proyecto ha sido desarrollado exitosamente, manteniendo la confidencialidad de su generación mediante inteligencia artificial.

1. **Definición de requerimientos:** Como resultado a un análisis detallado de los requerimientos del sistema se logró concluir que los requerimientos principales para el sistema de visión artificial de ITCA-FEPADE son:
 - a. Detección de eventos sospechosos,
 - b. Detección de personas,
 - c. Detección de objetos,
 - d. Almacenamiento y gestión de eventos sospechosos,
 - e. Generación de alertas mediante correo electrónico, SMS y alertas físicas.

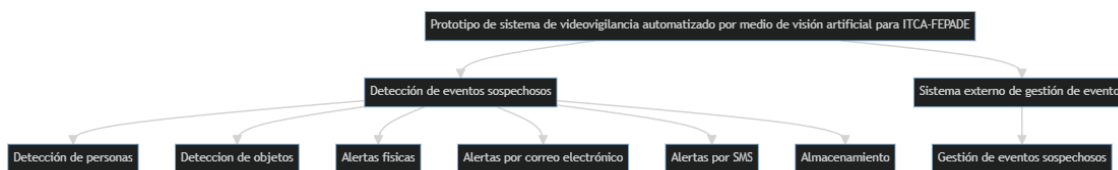


Ilustración 5 - Diagrama de requerimientos

2. **Investigación y selección de tecnologías:** Se llevo a cabo una investigación exhaustiva para definir las librerías seleccionadas para dicho proyecto, teniendo en cuenta que tienen que ser las más precisas y las que mejor se adapten a nuestro entorno de trabajo. Entre las seleccionadas tenemos, OpenCv Python, Python, TensorFlow, Pytorch que ayudan a lograr la funcionabilidad requerida con una mayor estabilidad y exactitud. Con estas librerías se accede a técnicas de procesamiento de imágenes que son ideales para los objetivos del sistema.

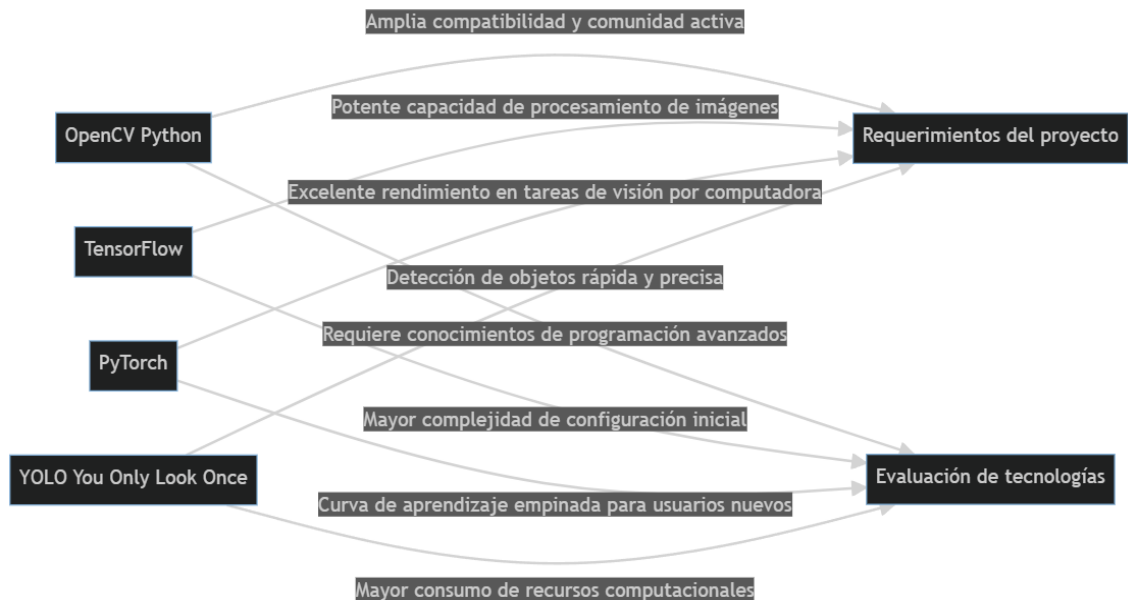


Ilustración 6 - Selección de tecnologías

3. **Diseño del sistema:** Se identifico los componentes necesarios, como cámaras de vigilancia, equipos de procesamiento, software de análisis de imágenes e infraestructura de red. La arquitectura del sistema se define mediante la integración de las tecnologías seleccionadas y la creación de interfaces de usuario para

la gestión de eventos, así como para la visualización de estos.

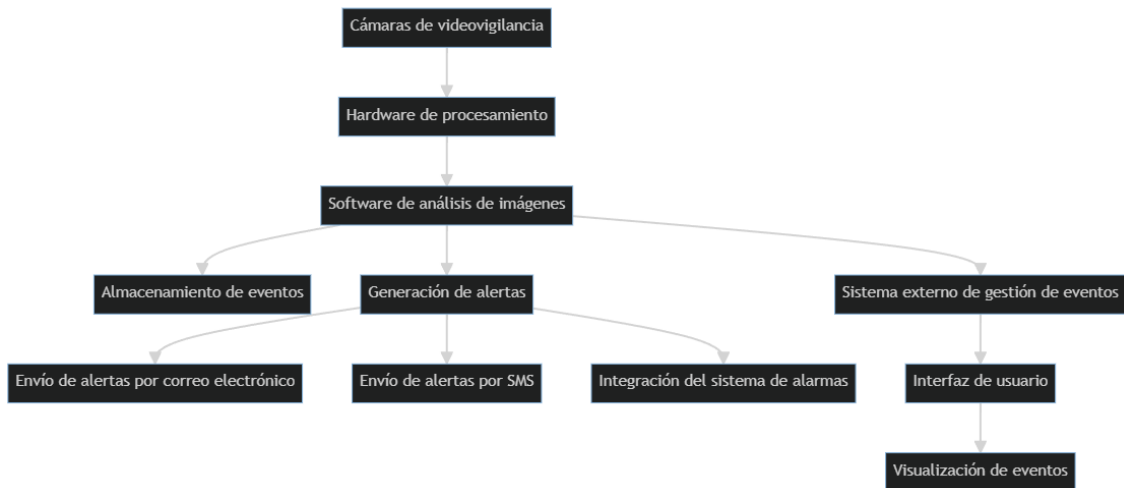


Ilustración 7 - Arquitectura del sistema

4. **Desarrollo del prototipo:** Se procedió con el desarrollo de prototipo donde se crearon los componentes principales entre los cuales se mencionan la detección del comportamiento sospechoso, detección de personas y objetos, almacenamiento de eventos, gestión de eventos, generación de alertas por correo electrónico, SMS y mediante alarmas físicas. Se utilizó un enfoque iterativo para la implementación que prueba y ajusta a medida que avanza el desarrollo.



Ilustración 8 - Proceso de detección de eventos

5. **Evaluación y validación del prototipo:** Se realizaron pruebas exhaustivas para evaluar el funcionamiento y rendimiento del prototipo. Se recopilaron datos de prueba y se analizaron los resultados obtenidos. Se compararon los

resultados con los objetivos esperados y se realizaron ajustes en los parámetros y algoritmos según corresponda. Para este punto se escogió trabajar con YOLOv8 debido a su alta precisión ya que se concluyó que la precisión es primordial en el desarrollo e implementación del prototipo.



Ilustración 9 - Comparación de algoritmos de detección

Los métodos y estrategias utilizados en el prototipo de videovigilancia automatizado se basan en un enfoque sistemático y estructurado para lograr objetivos definidos. Se tomaron varias medidas durante el desarrollo del proyecto. Primero se definieron los requisitos de su sistema e identifique las capacidades clave, como la detección de eventos sospechosos y la generación de alertas. Luego se paso al diseño del sistema, estableciendo la arquitectura y eligiendo la mejor tecnología. Posteriormente se desarrolló un prototipo, se implementó la funcionalidad requerida y se probó su correcta funcionalidad.

Modelado de la innovación

El principal objetivo de este innovador proyecto fue desarrollar un sistema de videovigilancia para el estacionamiento de ITCA-FEPADE mediante tecnología de visión artificial. El objetivo es mejorar la seguridad del estacionamiento identificando eventos sospechosos con anticipación, generando alertas y respuesta rápida.

El diseño preliminar del prototipo de sistema de videovigilancia automatizada para el estacionamiento de ITCA-FEPADE se compone de los siguientes elementos:

- **Hardware:** Cámara de videovigilancia: Selecciona una cámara de alta resolución y calidad de imagen que pueda cubrir eficientemente el área del estacionamiento. La cámara debe ser capaz de capturar videos en tiempo real con claridad y nitidez, incluso en condiciones de iluminación variables.
- **Software:** Lenguaje de programación: Utiliza Python como lenguaje principal para el desarrollo del sistema, debido a su amplia disponibilidad de librerías y su facilidad de uso.
- **Bibliotecas y frameworks:**
 - **OpenCV:** Es una biblioteca popular de procesamiento de imágenes y videos que proporciona diversas funciones para la captura de video, detección de objetos, seguimiento de movimiento y análisis de eventos. OpenCV será fundamental para el procesamiento y análisis de las imágenes capturadas por la cámara de videovigilancia.

- **YOLO (You Only Look Once)**: Es un modelo de detección de objetos en tiempo real que se basa en redes neuronales convolucionales. YOLO permite la detección de múltiples objetos en una sola pasada del modelo, lo que lo hace rápido y eficiente para aplicaciones de videovigilancia en tiempo real.
- **Django**: Es un framework de desarrollo web en Python que facilita la creación de aplicaciones web robustas y escalables. Django se utilizó para desarrollar un aplicativo de monitoreo web, donde los usuarios autorizados podrán acceder al video en tiempo real, recibir alertas y revisar registros de eventos anteriores.
- **Base de datos**: Se utilizó MySQL ya que es una base de datos relacional y nos servirá para almacenar y gestionar la información relevante del sistema, como registros de eventos, usuarios y configuraciones.

Arquitectura del sistema

- **Captura de video:** Una cámara de videovigilancia captura un flujo de video en tiempo real del estacionamiento.
- **Procesamiento de video:** El video se procesa mediante OpenCV y otros algoritmos de detección de objetos basados en YOLO con los que podemos obtener la captura de eventos sospechosos y objetos potencialmente peligrosos como pueden ser armas de fuego.
- **Generación de alertas:** Al momento de capturar eventos sospechosos el aplicativo emite alertas por medio de correo electrónico, SMS y mediante alertas físicas que se manejan mediante Arduino.
- **Aplicativo de monitoreo:** A través de la aplicación web desarrollada en Django, los usuarios autorizados pueden recibir alertas en tiempo real y consultar registros de eventos anteriores.
- **Almacenamiento de datos:** Se maneja una base de datos relacional donde se puede almacenar los eventos registrados, con la idea de tener centralizada la información y en un futuro poder generar reportes y poder acceder a la información de manera sencilla.

El diseño preliminar combina el hardware y el software necesario para capturar, procesar, analizar y gestionar eficazmente la información del estacionamiento de ITCA-FEPADE. La integración de tecnologías como Python, OpenCV, YOLO, Django y bases de datos relacionales permite desarrollar un sistema de videovigilancia automatizada robusto y escalable, mejorando la

seguridad y el monitoreo de eventos sospechosos en el estacionamiento.

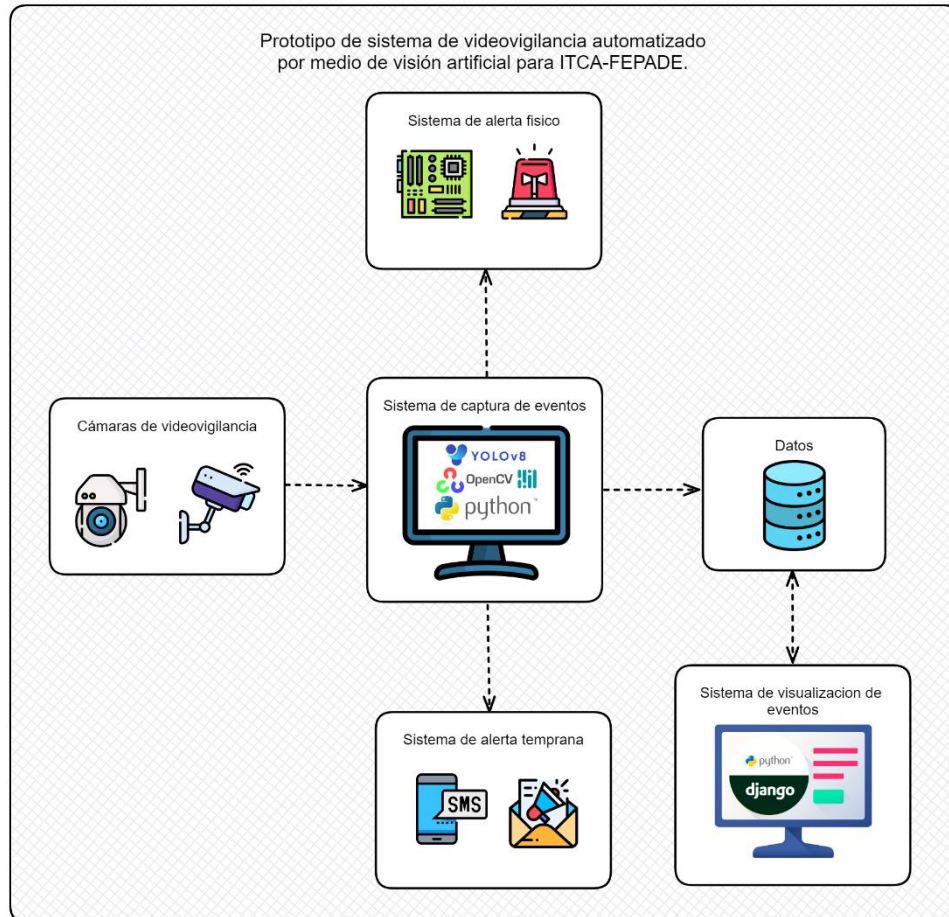


Ilustración 10 - Diagrama del prototipo de sistema de videovigilancia automatizado por medio de visión artificial para ITCA-FEPADE

Base de datos

Base de datos que guarda información sobre eventos capturados en tiempo real y predicciones usando Python a través de un modelo entrenado con YOLO v8, se usa un dataset personalizado de imágenes, para el dataset de entrenamiento se usaron imágenes capturadas en el estacionamiento de ITCA-FEPADE.

Diagrama de base de datos

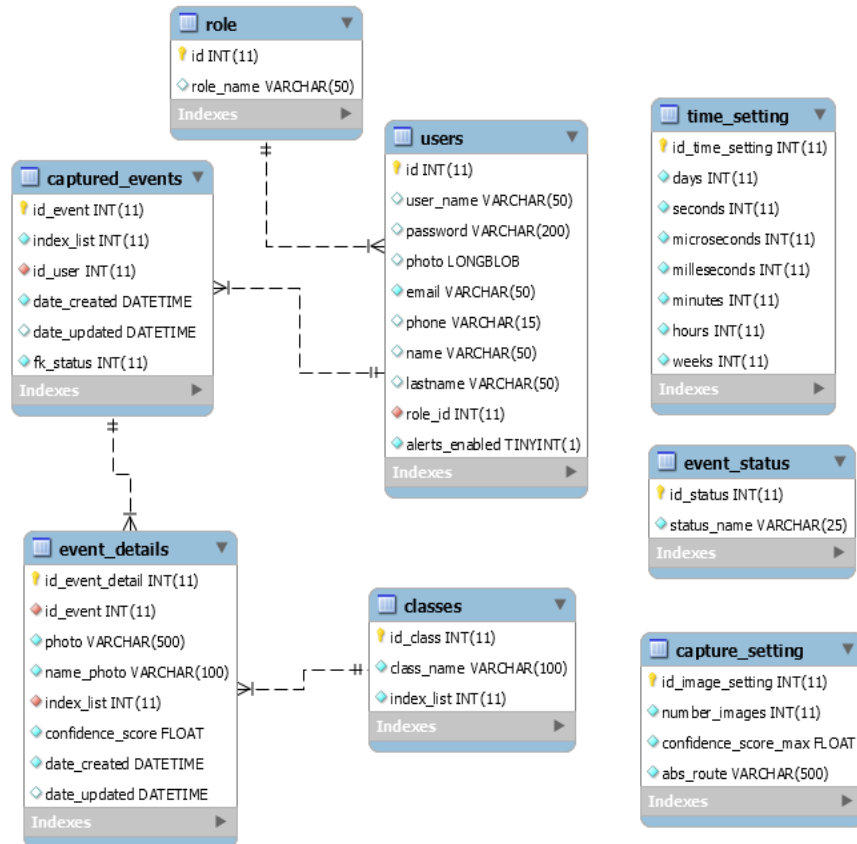


Ilustración 11 - Diagrama de base de datos

Diccionario de datos

role									
COLUMN	PRIMARY KEY	DATA TYPE	MAX LENGHT	IS NULL	AUTO INCREMENT	FOREIGN KEY	REFERENCE	UNIQUE	DESCRIPTION
id	SI	INT	11	NO	SI	-	-	-	Identificador del registro role
role_name	-	VARCHAR	50	NO	-	-	-	-	Identifica el nombre del rol de un usuario

Ilustración 12 - Diccionario de tabla role

users									
COLUMN	PRIMARY KEY	DATA TYPE	MAX LENGHT	IS NULL	AUTO INCREMENT	FOREIGN KEY	REFERENCE	UNIQUE	DESCRIPTION
id	SI	INT	11	NO	SI	-	-	-	Identificador del registro users
user_name	-	VARCHAR	50	NO	-	-	-	SI	identificacion del usuario
password	-	VARCHAR	200	NO	-	-	-	-	contraseña del usuario
photo	-	LONGBLOB		NO	-	-	-	-	imagen del rostro del usuario
phone	-	VARCHAR	15	NO	-	-	-	-	telefono del usuario
email	-	VARCHAR	50	NO	-	-	-	SI	correo electronico del usuario
name	-	VARCHAR	50	NO	-	-	-	-	nombre del usuario
lastname	-	VARCHAR	50	NO	-	-	-	-	apellidos del usuario
alerts_enabled	-	BOOLEAN		NO	-	-	-	-	identifica si el usuario recibira las alertas por correo o no
role_id	-	INT	11	NO	-	SI	role	-	Identifica el rol al que pertenece el usuario

Ilustración 13 - Diccionario de tabla users

capture_setting									
COLUMN	PRIMARY KEY	DATA TYPE	MAX LENGHT	IS NULL	AUTO INCREMENT	FOREIGN KEY	REFERENCE	UNIQUE	DESCRIPTION
id_image_setting	SI	INT	11	NO	SI	-	-	-	Identificador del registro capture setting
number_images	-	INT	11	NO	-	-	-	-	Número de imágenes que se almacenaran en base de datos por evento registrado
confidence_score	-	FLOAT		NO	-	-	-	-	Puntuación de confianza minimo permitido para registrar un evento y emitir alerta
abs_route	-	VARCHAR	500	NO	-	-	-	-	Ruta absoluta donde se almacenaran las imágenes capturadas por el sistema de visión artificial y sera la ruta que usara el sistema web para cargar las imágenes capturas por evento

Ilustración 14- Diccionario de tabla capture_setting

time_setting									
COLUMN	PRIMARY KEY	DATA TYPE	MAX LENGHT	IS NULL	AUTO INCREMENT	FOREIGN KEY	REFERENCE	UNIQUE	DESCRIPTION
id_time_setting	SI	INT	11	NO	SI	-	-	-	Identificador del registro time setting
days	-	INT	11	NO	-	-	-	-	Indica los días de espera para registrar nuevos eventos
seconds	-	INT	11	NO	-	-	-	-	Indica los segundos de espera para registrar nuevos eventos
microseconds	-	INT	11	NO	-	-	-	-	Indica los microsegundos de espera para registrar nuevos eventos
milliseconds	-	INT	11	NO	-	-	-	-	Indica los milisegundos de espera para registrar nuevos eventos
minutes	-	INT	11	NO	-	-	-	-	Indica los minutos de espera para registrar nuevos eventos
hours	-	INT	11	NO	-	-	-	-	Indica las horas de espera para registrar nuevos eventos
weeks	-	INT	11	NO	-	-	-	-	Indica las semanas de espera para registrar nuevos eventos

Ilustración 15 - Diccionario de tabla time_setting

classes									
COLUMN	PRIMARY KEY	DATA TYPE	MAX LENGHT	IS NULL	AUTO INCREMENT	FOREIGN KEY	REFERENCE	UNIQUE	DESCRIPTION
id_class	SI	INT	11	NO	SI	-	-	-	Identificador del registro classes
class_name	-	VARCHAR	100	NO	-	-	-	-	Nombre de la clase a capturar según el listado usado para entrenar el modelo en YOLO v8
index_list	-	INT	11	NO	-	-	-	SI	Posición de la clase según el listado usado para entrenar el modelo en YOLO v8

Ilustración 16 - Diccionario de tabla classes

captured_events									
COLUMN	PRIMARY KEY	DATA TYPE	MAX LENGHT	IS NULL	AUTO INCREMENT	FOREIGN KEY	REFERENCE	UNIQUE	DESCRIPTION
id_event	SI	INT	11	NO	SI	-	-	-	Identificador del registro captured events
index_list	-	INT	11	NO	-	SI	classes	-	Posición de la clase según el listado usado para entrenar el modelo en YOLO v8
id_user	-	INT	11	NO	-	SI	users	-	Identificación del usuario que ingreso al sistema al momento capturar eventos
date_created	-	DATETIME	-	NO	-	-	-	-	Fecha de captura del evento
date_updated	-	DATETIME	-	NO	-	-	-	-	Fecha de modificación del evento

Ilustración 17 - Diccionario de tabla captured_events

event_details									
COLUMN	PRIMARY KEY	DATA TYPE	MAX LENGHT	IS NULL	AUTO INCREMENT	FOREIGN KEY	REFERENCE	UNIQUE	DESCRIPTION
id_event_detail	SI	INT	11	NO	SI	-	-	-	Identificador del registro captured events
id_event	-	INT	11	NO	-	SI	captured_event	-	Identificador del evento al que pertenece el detalle
photo	-	VARCHAR	500	NO	-	-	-	-	Url del directorio donde se encuentra la imagen capturada
name_photo	-	VARCHAR	100	NO	-	-	-	-	Nombre relacionado con la imagen
index_list	-	INT	11	NO	-	SI	classes	-	Identificador de la clase capturada a la que pertenece el evento
confidence_score	-	FLOAT	-	NO	-	-	-	-	Puntuación de confianza detectada con el modelo preentrenado en YOLO v8
date_created	-	DATETIME	-	NO	-	-	-	-	Fecha de captura del detalle del evento
date_updated	-	DATETIME	-	NO	-	-	-	-	fecha de modificación del detalle del evento

Ilustración 18 - Diccionario de tabla event_details

Casos de uso

En esta sección se presenta en primer lugar, los principales actores, luego se muestran los diagramas de casos de uso identificados para el prototipo de videovigilancia empleando visión e inteligencia artificial.

Descripción de actores

- Administrador: persona encargada de gestionar las funciones del sistema; tales como:

Sistema web.

- Gestionar usuarios.
- Gestionar eventos.
- Gestionar ajustes.

Sistema de visión artificial.

- Registrar usuarios.

- Visión artificial: sistema que utiliza inteligencia artificial, el cual se encargara de gestionar las funciones del sistema de visión artificial; tales como:

- Capturar eventos.
- Registrar eventos en base de datos.
- Capturar objetos.
- Emitir alerta a través de una notificación.
- Activar alerta temprana.

- Operador: persona encargada de monitorear las cámaras de seguridad, se encarga de iniciar monitoreo.

Diagramas de Caso de Uso

En esta sección se presentan los diagramas de casos de uso para cada uno de los actores.

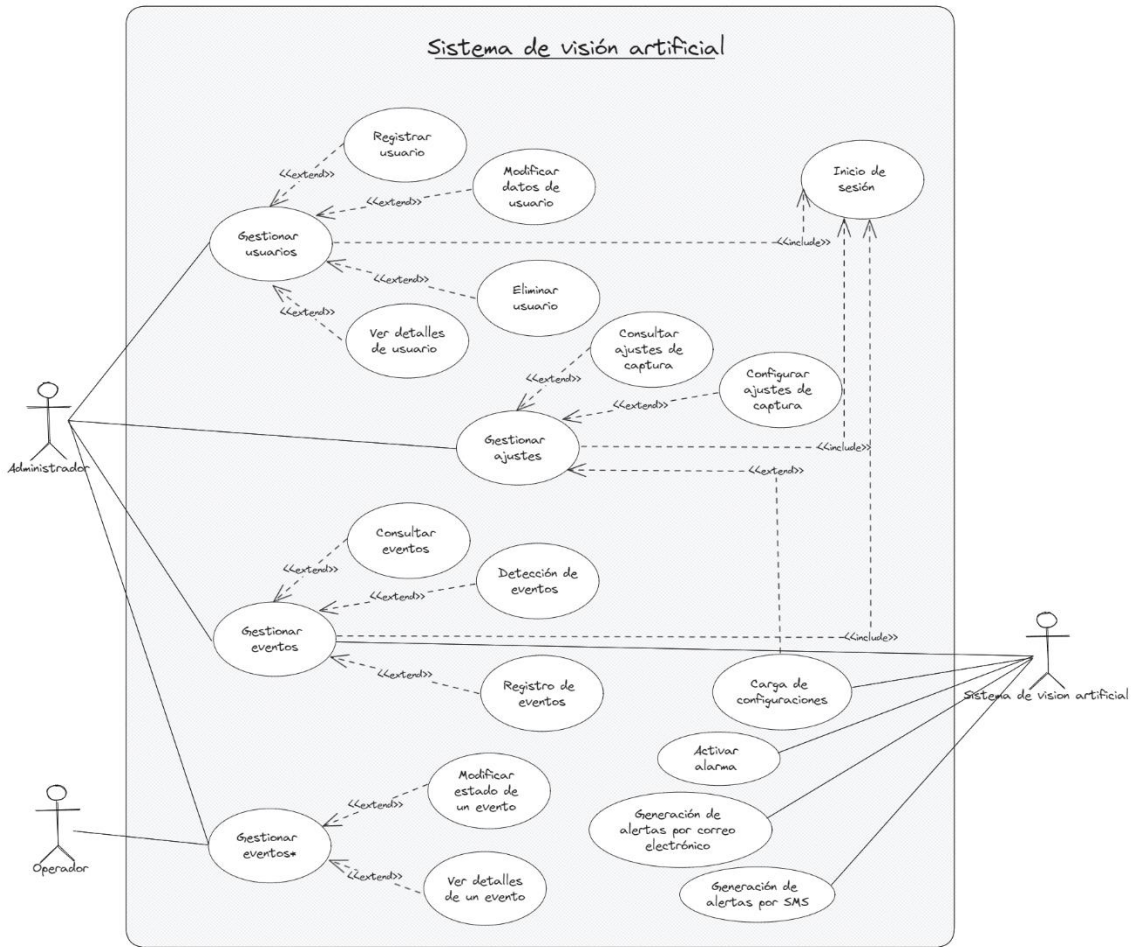


Ilustración 19 - Diagrama de casos de uso

Diccionarios de Caso de Uso

En esta sección se presentan los diccionarios de casos de uso del prototipo de visión artificial.

Caso de uso #: 001	
Nombre:	Inicio de sesión
Actores Principales:	Usuario del sistema
Flujo de eventos:	<ol style="list-style-type: none">1. El usuario ingresa su nombre de usuario y contraseña en el formulario de inicio de sesión.2. El sistema valida las credenciales ingresadas.3. Si las credenciales son válidas, el sistema autentica al usuario y le permite acceder al sistema.4. Si las credenciales son inválidas, el sistema muestra un mensaje de error y solicita al usuario que ingrese nuevamente sus credenciales.
Precondiciones:	Para que el usuario puede acceder al sistema mediante nombre y contraseña, el usuario debe estar registrado en la base de datos.

Caso de uso #: 001	
Escenarios:	<p>Escenario 1: Las credenciales ingresadas son válidas.</p> <ul style="list-style-type: none"> • El sistema autentica al usuario y le permite acceder a las funcionalidades del sistema. <p>Escenario 2: Las credenciales ingresadas son inválidas.</p> <ul style="list-style-type: none"> • El sistema muestra un mensaje de error indicando que las credenciales son incorrectas y solicita al usuario que ingrese nuevamente sus credenciales.

Tabla 1 - Inicio de sesión

Caso de uso #: 002	
Nombre:	Registrar usuario
Actores Principales:	Administrador del sistema
Flujo de eventos:	<ol style="list-style-type: none"> 1. El administrador selecciona la opción "Registrar usuario". 2. El sistema muestra un formulario para ingresar los datos del usuario, como nombre de usuario, contraseña, correo electrónico, número de teléfono, nombre, apellido, habilitación de correos electrónicos y selección de rol. 3. El administrador completa los campos del formulario. 4. El sistema valida los datos ingresados. 5. El sistema crea un nuevo registro de usuario en la base de datos con la información proporcionada.
Precondiciones:	El administrador ha iniciado sesión en el sistema.
Escenarios:	Escenario 1: El formulario se completa correctamente.

Caso de uso #: 002	
	<ul style="list-style-type: none"> • El sistema muestra un mensaje de confirmación de registro exitoso. <p>Escenario 2: El formulario contiene errores o campos faltantes.</p> <ul style="list-style-type: none"> • El sistema muestra mensajes de error indicando los campos que deben corregirse.

Tabla 2 - Registrar usuario

Caso de uso #: 003	
Nombre:	Modificar datos de usuario
Actores Principales:	Administrador del sistema
Flujo de eventos:	<ol style="list-style-type: none"> 1. El administrador selecciona la opción "Modificar datos de usuario". 2. El sistema muestra una lista de usuarios registrados con sus respectivos datos. 3. El administrador selecciona el usuario que desea modificar. 4. El sistema muestra un formulario con los datos del

Caso de uso #: 003	
	<p>usuario seleccionado, permitiendo al administrador editarlos.</p> <p>5. El administrador realiza las modificaciones necesarias en el formulario.</p> <p>6. El sistema valida los datos modificados.</p> <p>7. El sistema actualiza el registro del usuario en la base de datos con la información modificada.</p>
Precondiciones:	El administrador ha iniciado sesión en el sistema y ha consultado la lista de usuarios.
Escenarios:	<p>Escenario 1: El formulario se completa correctamente.</p> <ul style="list-style-type: none"> • El sistema muestra un mensaje de confirmación de modificación exitosa. <p>Escenario 2: El formulario contiene errores o campos faltantes.</p> <ul style="list-style-type: none"> • El sistema muestra mensajes de error indicando los campos que deben corregirse.

Tabla 3 - Modificar datos de usuario

Caso de uso #: 004	
Nombre:	Eliminar usuario
Actores Principales:	Administrador del sistema
Flujo de eventos:	<ol style="list-style-type: none"> 1. El administrador selecciona la opción "Eliminar usuario". 2. El sistema muestra una lista de usuarios registrados con sus respectivos datos. 3. El administrador selecciona el usuario que desea eliminar. 4. El sistema muestra una confirmación de eliminación para el usuario seleccionado. 5. El administrador confirma la eliminación del usuario. 6. El sistema elimina el registro del usuario de la base de datos.
Precondiciones:	El administrador ha iniciado sesión en el sistema y ha consultado la lista de usuarios.
Escenarios:	Escenario 1: El administrador confirma la eliminación del usuario.

Caso de uso #: 004	
	<ul style="list-style-type: none"> • El sistema muestra un mensaje de confirmación de eliminación exitosa. <p>Escenario 2: El administrador cancela la eliminación del usuario.</p> <ul style="list-style-type: none"> • El sistema muestra un mensaje indicando que la eliminación ha sido cancelada.

Tabla 4 - Eliminar usuario

Caso de uso #: 005	
Nombre:	Ver detalles de usuario
Actores Principales:	Administrador del sistema
Flujo de eventos:	<ol style="list-style-type: none"> 1. El administrador selecciona la opción "Ver detalles de usuario". 2. El sistema muestra una lista de usuarios registrados con sus respectivos datos. 3. El administrador selecciona el usuario del cual desea ver los detalles.

Caso de uso #: 005	
	<p>4. El sistema muestra los detalles completos del usuario seleccionado, incluyendo su nombre de usuario, correo electrónico, teléfono, nombre, apellido y rol asignado.</p>
Precondiciones:	<p>El administrador ha iniciado sesión en el sistema y ha consultado la lista de usuarios.</p>
Escenarios:	<p>Escenario 1: El usuario seleccionado existe en el sistema.</p> <ul style="list-style-type: none"> • El sistema muestra los detalles completos del usuario seleccionado. <p>Escenario 2: El usuario seleccionado no existe en el sistema.</p> <ul style="list-style-type: none"> • El sistema muestra un mensaje indicando que el usuario seleccionado no está disponible.

Tabla 5 - Ver detalles de usuario

Caso de uso #: 006	
Nombre:	Configurar ajustes de captura
Actores Principales:	Administrador del sistema
Flujo de eventos:	<ol style="list-style-type: none"> 1. El administrador accede a la sección de configuración de ajustes de captura. 2. El sistema muestra los ajustes de captura actuales, como el número de imágenes y la puntuación mínima de confianza. 3. El administrador modifica los ajustes según los requisitos del sistema. 4. El sistema valida los valores ingresados y actualiza la configuración en la base de datos.
Precondiciones:	El administrador ha iniciado sesión en el sistema.
Escenarios:	<p>Escenario 1: Los ajustes se modifican correctamente.</p> <ul style="list-style-type: none"> • El sistema muestra un mensaje de confirmación de modificación exitosa.

Caso de uso #: 006	
	<p>Escenario 2: Los ajustes contienen valores inválidos o no cumplen con los requisitos del sistema.</p> <ul style="list-style-type: none"> • El sistema muestra mensajes de error indicando los ajustes que deben corregirse.

Tabla 6 - Configurar ajustes de captura

Caso de uso #: 007	
Nombre:	Consultar ajustes de captura
Actores Principales:	Usuarios del sistema
Flujo de eventos:	<ol style="list-style-type: none"> 1. El usuario accede a la sección de configuración de ajustes de captura. 2. El sistema muestra los ajustes de captura actuales, como el número de imágenes y la puntuación mínima de confianza.
Precondiciones:	El usuario ha iniciado sesión en el sistema.
Escenarios:	Escenario 1: Los ajustes se muestran correctamente.

Caso de uso #: 007	
	<ul style="list-style-type: none"> • El usuario puede visualizar los ajustes de captura actuales.

Tabla 7 - Consultar ajustes de captura

Caso de uso #: 008	
Nombre:	Consultar eventos
Actores Principales:	Usuarios del sistema
Flujo de eventos:	<ol style="list-style-type: none"> 1. El usuario accede a la sección de eventos capturados. 2. El sistema muestra la lista de eventos capturados, incluyendo el identificador, la posición en el listado de YOLO v8, el estado actual, el usuario que capturó el evento y las fechas de creación y actualización.
Precondiciones:	El usuario ha iniciado sesión en el sistema.
Escenarios:	<p>Escenario 1: Los eventos se muestran correctamente.</p> <ul style="list-style-type: none"> • El usuario puede visualizar la lista de eventos capturados.

Tabla 8 - Consultar eventos

Caso de uso #: 009	
Nombre:	Modificar estado de un evento
Actores Principales:	Usuarios del sistema
Flujo de eventos:	<ol style="list-style-type: none"> 1. El usuario accede a la sección de eventos capturados. 2. El sistema muestra la lista de eventos capturados. 3. El usuario selecciona el evento que desea modificar. 4. El sistema muestra los detalles del evento, incluyendo el estado actual. 5. El usuario modifica el estado del evento. 6. El sistema valida los cambios y actualiza el estado del evento en la base de datos.
Precondiciones:	El usuario ha iniciado sesión en el sistema.
Escenarios:	<p>Escenario 1: El estado del evento se modifica correctamente.</p> <ul style="list-style-type: none"> • El sistema muestra un mensaje de confirmación de la modificación exitosa.

Caso de uso #: 009	
	<p>Escenario 2: Ocurrió un error durante la modificación del estado del evento.</p> <ul style="list-style-type: none"> • El sistema muestra mensajes de error indicando los problemas encontrados.

Tabla 9 - Modificar estado de un evento

Caso de uso #: 010	
Nombre:	Ver detalles de un evento
Actores Principales:	Usuarios del sistema
Flujo de eventos:	<ol style="list-style-type: none"> 1. El usuario accede a la sección de eventos capturados. 2. El sistema muestra la lista de eventos capturados. 3. El usuario selecciona un evento de la lista. 4. El sistema muestra los detalles del evento, incluyendo las fotos asociadas, los nombres de las fotos, las clases capturadas, las puntuaciones de confianza y las fechas de creación y actualización de los detalles.

Caso de uso #: 010	
Precondiciones:	El usuario ha iniciado sesión en el sistema y existen eventos capturados con detalles.
Escenarios:	<p>Escenario 1: Los detalles del evento se muestran correctamente.</p> <ul style="list-style-type: none"> • El usuario puede ver las fotos asociadas, los nombres de las fotos, las clases capturadas, las puntuaciones de confianza y las fechas de creación y actualización de los detalles del evento.

Tabla 10 - Ver detalles de un evento

Caso de uso #: 011	
Nombre:	Carga de configuraciones
Actores Principales:	Sistema de visión artificial
Flujo de eventos:	<ol style="list-style-type: none"> 1. Consultar configuraciones de captura de eventos y tiempos de eventos a la base de datos. 2. Asignar valores a variables globales.

Caso de uso #: 011	
	3. Analizar imágenes con los parámetros configurados.
Precondiciones:	El usuario ha iniciado sesión en el sistema y existen configuraciones en las tablas de la base de datos.
Escenarios:	<p>Escenario 1: El sistema analiza las imágenes satisfactoriamente con los parámetros establecidos.</p> <p>Escenario 2: El sistema experimenta dificultades en la conexión con la base de datos y trabaja con parámetros por defecto en cero.</p>

Tabla 11 - Carga de configuraciones

Caso de uso #: 012	
Nombre:	Detección de eventos
Actores Principales:	Sistema de visión artificial
Flujo de eventos:	1. El sistema de visión artificial recibe la entrada de video de la cámara de vigilancia.

Caso de uso #: 012	
	<p>2. Utilizando algoritmos de detección basados en redes neuronales convolucionales, el sistema analiza el flujo de video en tiempo real.</p> <p>3. El sistema identifica objetos y personas en el video mediante técnicas de detección de objetos como YOLO.</p> <p>4. Si se detecta un objeto o persona sospechosa según los criterios establecidos, el sistema registra un evento.</p> <p>5. El sistema guarda el evento y los detalles de este, así como la clase detectada, la confianza y la marca de tiempo.</p>
Precondiciones:	El sistema de visión artificial está en funcionamiento y recibiendo la entrada de video de la cámara.
Escenarios:	<p>Escenario 1: El sistema detecta un evento sospechoso.</p> <ul style="list-style-type: none"> • El sistema registra el evento y guarda los detalles relevantes. • El sistema emite una alerta enviando un correo electrónico y

Caso de uso #: 012	
	SMS a los destinatarios especificados.

Tabla 12 - Detección de eventos

Caso de uso #: 013	
Nombre:	Registro de eventos
Actores Principales:	Sistema de visión artificial
Flujo de eventos:	<ol style="list-style-type: none"> 1. El sistema de visión artificial detecta un evento sospechoso. 2. El sistema registra los detalles del evento, como la clase detectada, la ubicación y la marca de tiempo. 3. Los detalles del evento se almacenan en la base de datos.
Precondiciones:	El sistema de visión artificial está en funcionamiento y configurado correctamente.
Escenarios:	Escenario 1: El sistema registra con éxito el evento y almacena los detalles en la base de datos.

Caso de uso #: 013	
	Escenario 2: El sistema no puede detectar un evento sospechoso y no se realiza el registro.

Tabla 13 - Registro de eventos

Caso de uso #: 014	
Nombre:	Generación de alertas por correo electrónico
Actores Principales:	Sistema de visión artificial, Servidor de correo electrónico
Flujo de eventos:	<ol style="list-style-type: none"> 1. El sistema de visión artificial registra un evento sospechoso. 2. El sistema evalúa la gravedad del evento según la puntuación de confianza y otros criterios establecidos. 3. Si el evento cumple con los criterios de alerta definidos, el sistema genera una alerta. 4. El sistema de visión artificial se comunica con el servidor de correo electrónico

Caso de uso #: 014	
	<p>para enviar la alerta por correo electrónico.</p> <p>5. El servidor de correo electrónico envía la alerta a los destinatarios especificados, como administradores de seguridad o personal encargado.</p>
Precondiciones:	El sistema de visión artificial ha registrado un evento sospechoso que cumple con los criterios de alerta.
Escenarios:	<p>Escenario 1: El sistema genera una alerta por correo electrónico.</p> <ul style="list-style-type: none"> • El sistema de visión artificial crea el contenido del correo electrónico, incluyendo la información relevante sobre el evento, como la ubicación, la clase detectada y la puntuación de confianza. • El sistema se autentica con el servidor de correo electrónico utilizando las credenciales de autenticación previamente configuradas.

Caso de uso #: 014	
	<ul style="list-style-type: none"> • El sistema envía el correo electrónico a los destinatarios especificados. • Los destinatarios reciben el correo electrónico de alerta y pueden tomar las medidas correspondientes, como verificar la situación, activar protocolos de seguridad o realizar un seguimiento del evento.

Tabla 14 - Generación de alertas por correo electrónico

Caso de uso #: 015	
Nombre:	Generación de alertas por SMS
Actores Principales:	Sistema de visión artificial, Servidor de SMS
Flujo de eventos:	<p>2. El sistema de visión artificial registra un evento sospechoso.</p> <p>3. El sistema evalúa la gravedad del evento según la puntuación de confianza y otros criterios establecidos.</p>

Caso de uso #: 015	
	<p>4. Si el evento cumple con los criterios de alerta definidos, el sistema genera una alerta.</p> <p>5. El sistema de visión artificial se comunica con el servidor de SMS para enviar la alerta por SMS.</p> <p>6. El servidor de SMS envía la alerta a los destinatarios especificados, como administradores de seguridad o personal encargado.</p>
Precondiciones:	El sistema de visión artificial ha registrado un evento sospechoso que cumple con los criterios de alerta.
Escenarios:	<ul style="list-style-type: none"> • Escenario 1: El sistema genera una alerta por SMS. El sistema de visión artificial crea el contenido del SMS, incluyendo la información relevante sobre el evento, como la ubicación, la clase detectada y la puntuación de confianza. • El sistema se autentica con el servidor de SMS utilizando las credenciales de autenticación previamente configuradas.

Caso de uso #: 015	
	<ul style="list-style-type: none"> • El sistema envía el SMS a los destinatarios especificados. • Los destinatarios reciben el SMS de alerta y pueden tomar las medidas correspondientes, como verificar la situación, activar protocolos de seguridad o realizar un seguimiento del evento.

Tabla 15 - Generación de alertas por SMS

Caso de uso #: 016	
Nombre:	Activar alarma
Actores Principales:	Sistema Visión Artificial
Flujo de eventos:	<ol style="list-style-type: none"> 1. El Sistema Visión Artificial envía señal de evento sospechoso. 2. El Sistema Visión Artificial activa alarma de seguridad. 3. La alarma se activa.
Precondiciones:	<ul style="list-style-type: none"> • Conexión con la cámara de vigilancia. • Conexión con la red neuronal. • Conexión con la base de datos. • Tener un evento potencialmente sospechoso. • Conexión con la alarma.

Caso de uso #:	016
Escenarios:	<ul style="list-style-type: none">• Alarma activada con éxito.

Tabla 16 - Activar alarma

Interfaz gráfica del prototipo.

A continuación, se detallan las pantallas que contiene el sistema de consulta denominado plataforma web watchdog.

Inicio de sesión

Permite a los usuarios administradores iniciar sesión utilizando nombre de usuario y contraseña.



El prototipo muestra una interfaz de usuario para el inicio de sesión de un sistema web. La pantalla está dividida en dos secciones principales. A la izquierda, hay un formulario de inicio de sesión con el título "Iniciar sesión" y un icono de un cuadro con una X. El formulario contiene dos campos de entrada: "Usuario:" con el texto "admin" y "Contraseña:" con caracteres ocultos por asteriscos. Debajo de los campos hay un botón azul con el texto "Iniciar sesión" y un enlace azul que dice "¿Olvido la contraseña?". A la derecha de la sección de formulario, hay un área grande con una X diagonal, lo que indica que el contenido de esa parte del prototipo no está definido o es un espacio reservado.

Ilustración 20 - Inicio de sesión sistema web

Gestión de usuarios

Permite administrar los usuarios del sistema, deshabilitar un usuario, registrar un usuario, modificar los datos del usuario.

Usuarios

https://www.videovigilancia.com/usuarios

Bienvenido, usuario

Gestión de usuarios

opción
opción
opción

Correo Contraseña

Usuario Rol

Nombre Apellidos

Teléfono Enviar correos

Registrar

Columna 1	Columna 2	Columna 3	Columna 4	Columna 5	Columna 6	Columna 7	Acciones

<< 1 2 3 >>

Ilustración 21 - Gestión de usuarios sistema web

Gestión de eventos

Permite al usuario poder gestionar los sucesos que hayan sido registrados automáticamente por el sistema de visión artificial, estos sucesos representan una posible alerta para evitar un robo o hurto en un estacionamiento.

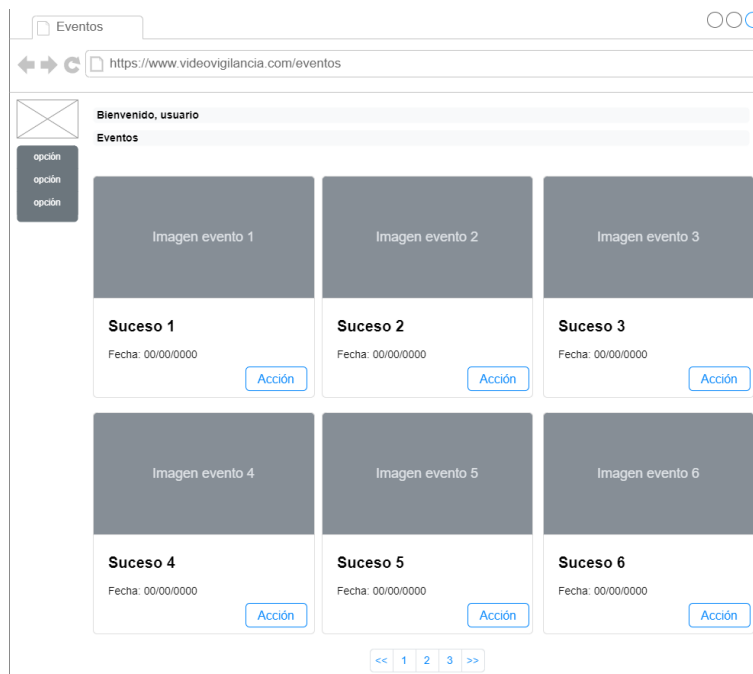


Ilustración 22 - Gestión de eventos sistema web

Configuración

Pantalla de configuración, contiene los valores de configuración para capturar eventos, con el número de imágenes por evento a ser capturadas y la puntuación de confianza en un rango entre 0 y 1 siendo el mínimo 0 y el máximo 1, el nivel de confianza por defecto es 0.85 puede ser superior para mayor confianza.

La configuración de tiempo nos permite dejar un tiempo entre un evento y otro, si todos los valores se encuentran en cero, el sistema registra los eventos que detecte sin dejar un tiempo de por medio.

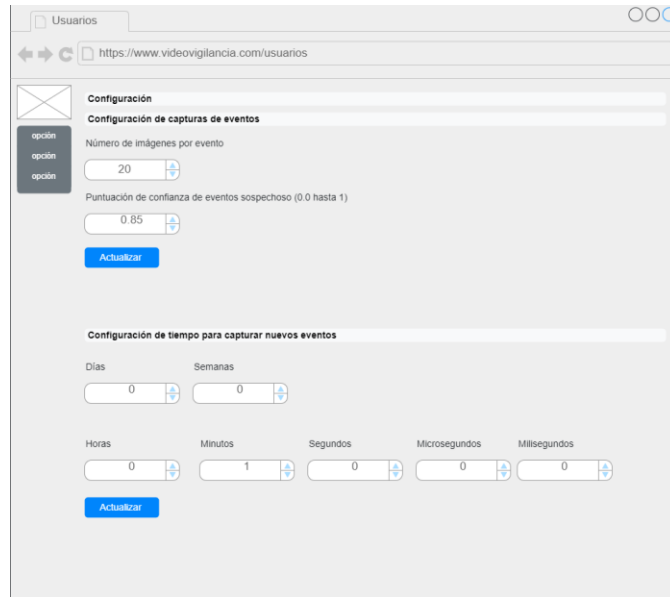


Ilustración 23 - Configuración de para captura de imágenes y tiempo para registro de nuevos eventos

A continuación, se detallan las diferentes pantallas que posee el sistema de visión artificial.

Iniciar sesión en el sistema

El usuario puede iniciar sesión previamente registrado por el administrador en la base de datos para poder acceder al sistema.



Ilustración 24 - Inicio de sesión sistema de visión artificial

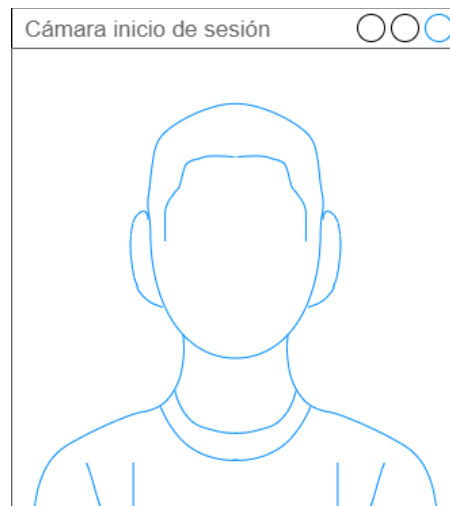


Ilustración 25 - Captura de fotografía inicio de sesión sistema de visión artificial

Registrar un nuevo usuario

El usuario si posee los permisos para poder registrar un usuario se habilita donde despliego la pantalla para registrarlo con los datos usuario, correo y la fotografía para poder identificarlo en futuros inicios de sesión.



Ilustración 26 - Registro en sistema de visión artificial

Captura de eventos

Para capturar eventos se debe de iniciar el sistema para empezar a monitorizar, el sistema utiliza un modelo entrenado con inteligencia artificial para la captura de eventos, de igual forma este registra en base de datos y envía una alerta al correo configurado, además se emitirá una alerta temprana

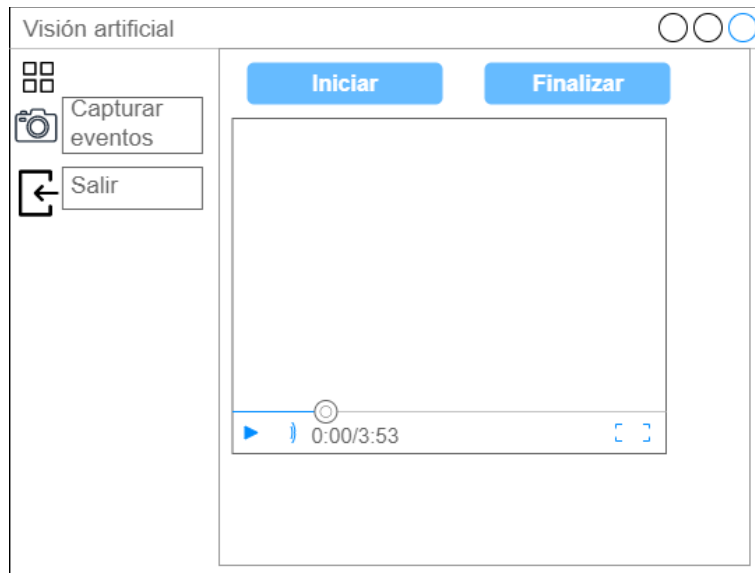


Ilustración 27 - Iniciar monitoreo

Organización, monitoreo y evaluación para la innovación

El desarrollo del proyecto de innovación se ha llevado a cabo siguiendo una cuidadosa organización. Se estableció un cronograma detallado que definió las diferentes etapas del proyecto, desde la investigación inicial hasta la implementación final. Este cronograma sirvió como una guía clara para el equipo de trabajo, permitiendo una gestión efectiva del tiempo y los recursos disponibles.

Se realizaron reuniones periódicas para evaluar el avance del proyecto y tomar decisiones oportunas haciendo uso de las bondades de SCRUM a pesar de basar nuestra metodología en KANBAN. Esto permitió identificar posibles desviaciones y tomar acciones correctivas de manera oportuna.

Se implementaron herramientas de seguimiento y gestión de proyectos para poder cumplir con el plazo establecido para el desarrollo del prototipo, esto mejora la gestión de los recursos y del tiempo.

Se encontraron problemas técnicos para la implementación de algoritmos de detección específicos en el sistema de videovigilancia, estos problemas se resolvieron investigando más sobre la visión artificial y realizando pruebas exhaustivas, ajustes iterativos y exploración de alternativas, se superaron estos obstáculos y se encontraron soluciones efectivas.

La evaluación de los resultados fue un aspecto clave en cada etapa del proyecto. Se llevaron a cabo análisis y pruebas rigurosas para medir la eficacia y eficiencia del sistema de videovigilancia implementado. Se recopilaron datos relevantes y

se compararon los resultados obtenidos con los objetivos establecidos. Esta evaluación continua permitió identificar áreas de mejora y realizar los ajustes necesarios para optimizar el rendimiento del sistema.

El desarrollo de proyectos innovadores se realiza según métodos Agile, especialmente el método Kanban. Cree un tablero Kanban para visualizar y administrar de manera efectiva los flujos de trabajo y las tareas del proyecto. Este tablero se dividió en columnas que representaban las distintas etapas del proceso, como "Por hacer", "En progreso" y "Completado".

El cronograma se manejó con Kanban, priorizando las tareas según su importancia. Cada tarea se reflejaba en una tarjeta Kanban que detallaba los requisitos, las responsabilidades y los plazos de esta forma se facilitaba la comunicación permitiendo un seguimiento preciso del progreso de cada tarea.

En cuanto al proyecto iba avanzando las tarjetas en estado "completado" eran más y se encontraron con cuellos de botella, pero la metodología facilitaba identificar esos cuellos de botella y así poder tomar acciones con antelación, se hizo especial énfasis en la mejora continua. De esta manera, el enfoque Kanban permitió una gestión ágil y adaptativa, fomentando la eficiencia y la calidad en el desarrollo del proyecto de innovación.

Cronograma y presupuesto

A continuación, se detalla la viabilidad económica del proyecto. Se consideran los costos asociados, como la adquisición de hardware y software, recursos humanos, infraestructura, conectividad, licencias y posibles gastos de mantenimiento. Se debe realizar un análisis detallado de los costos y compararlos con los beneficios esperados, como la mejora en la seguridad del estacionamiento y la optimización de recursos. Además, se puede evaluar la posibilidad de obtener financiamiento o buscar fuentes de financiación alternativas.

Costo del personal

El proyecto se trabajó en una modalidad ad-honorem para la realización del trabajo de tesis, sin embargo, es importante contemplar el costo teórico que como equipo aportaremos para el proyecto para poder determinar el costo real del proyecto. Hemos asignado un salario por hora de \$5.00 que es un valor con el que estamos de acuerdo para prestar nuestros servicios en el caso de que para la realización del proyecto se nos contratara con un salario. Teniendo esto en cuenta, para el equipo de 4 personas que realizaremos el proyecto, se tendría el siguiente costo de personal:

Personal	Salario por hora	Horas	Salario diario	Días	Salario	Meses	costo
1	\$5.00	8	\$40.00	26	\$ 1,040.00	3	\$3,120.00
2	\$5.00	8	\$40.00	26	\$ 1,040.00	3	\$3,120.00
3	\$5.00	8	\$40.00	26	\$ 1,040.00	3	\$3,120.00
4	\$5.00	8	\$40.00	26	\$ 1,040.00	3	\$3,120.00
						Tota l	\$ 12,480.00

Tabla 17 - Costo de personal

Costo hardware y software

Para el desarrollo del proyecto, se necesitaron los siguientes insumos:

- Hardware El equipo necesario para realizar el proyecto una cámara de videovigilancia Software.
- Los costos del software son asociados al costo de la infraestructura necesaria para desplegar el proyecto, siendo requeridos un dominio para acceder al sitio, así como un servidor basado en un sistema operativo Linux para poder alojar el software desarrollado.

En la presente tabla se definen los costos incurridos en la adquisición de componentes para el desarrollo del prototipo, se han definido el servidor y dominio el cual no aplicará ya que se trabajará de forma local por lo cual el costo aparece en cero dólares.

Concepto	Cantidad	Costo unitario	Costo total
Servidor para implementación de solución AWS (1 año)	1	\$0	\$0
DNS para el sitio videovigilancia.com.sv (1 año)	1	\$0	\$0
Cámara de videovigilancia	1	\$125.28	\$125.28
Kit Arduino	1	\$75	\$75
Router inalámbrico pruebas	1	\$20.00	\$20.00
Total			\$220.28

Tabla 18 - Costo de hardware y software

Para el desarrollo del proyecto se usó un presupuesto de \$12,480.00 en un escenario donde se deban de pagar honorarios al equipo desarrollador, en este caso que se proveerá el servicio ad-honorem solo se incurrirá en los gastos asociados al hardware y software requeridos para el proyecto, cuyo costo total es de \$220.28.

El costo de \$220.28 es un costo que se encuentra en las posibilidades del equipo, por lo cual la realización del prototipo es factible. Esto puede incluir mejoras en la seguridad del estacionamiento, optimización de recursos, reducción de pérdidas o robos, y mejora en la eficiencia operativa. Estos beneficios pueden traducirse en ahorros financieros o en la generación de valor para ITCA-FEPADE.

Tomando en cuenta que para el prototipo se utilizara una única cámara se realizar el análisis operativo.

Sabiendo que el sistema tiene como objetivo principal el solventar y mejorar los procesos de videovigilancia, y a la vez tener un mejor control. Muchas de las empresas hoy en día pueden obtener estos beneficios a nivel operacional, ya que con esta herramienta se busca optimizar todos estos procesos del día a día para el personal de seguridad. Y que a diferencia de los sistemas habituales que necesitan de intervención humana, este facilita el hecho de la interacción con la misma. Y a la vez tomar decisiones en base a la facilidad que proporciona la visión artificial.

A continuación, se presenta una propuesta de venta de producto y servicio que podemos ofrecer, ya sea para público general o empresas que estén interesadas en la adquisición de nuestro servicio de videovigilancia, hemos tomado en consideración los siguientes aspectos de venta:

- **Cámaras**

Se toma en cuenta la cobertura de las cámaras necesarias, basadas en el área de instalación, estas tienen una cobertura de 10 metros cuadrados a la redonda.

- **Mano de obra**

Instalación de cámaras, sistema de detección de eventos, Sistema Watchdog. Basado en los costos de implementación de $\$220.28 * 1.95$ (95% de ganancia) = $\$429.54$

- **Servidor**

Configuración de base de datos y sistemas ligados al funcionamiento general.

- **Licencia de software**
Derecho de pertenencia y uso del software, el cual es un pago único de adquisición
- **Soporte técnico**
Si se desea recibir mejoras o nuevas implementaciones dentro del sistema, se puede optar por esta opción; la cual no es obligatoria hacia el público de venta.

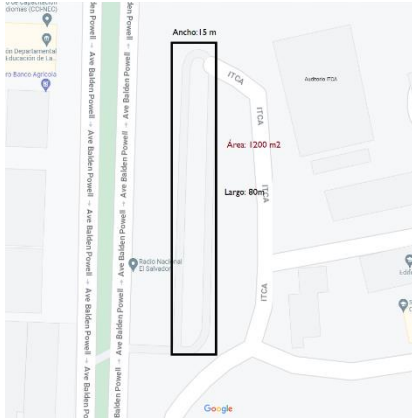
Cabe mencionar que la implementación puede variar dependiendo el área, lugar de trabajo o institución en la cual se realice, tomando en cuenta los costos de equipo y mano de obra, así como el soporte que también se requiera.

Para datos de ejemplo se tomó en cuenta las instalaciones de ITCA-Fepade, específicamente el parqueo principal de la institución, sirviendo este como modelo de implementación del proyecto, se tiene en cuenta lo siguiente:

Área de despliegue

Se estima establecer las cámaras en puntos estratégicos, con la finalidad de que cubran la mayor parte del área, teniendo en cuenta, esquinas, puntos céntricos. Las cámaras, al cubrir un área de 10 metros a la redonda, tendrían una cobertura de 4 cámaras en las esquinas, y en el área central 8 por cada lado. Siendo un total de 14 cámaras distribuidas en el área.

Tomar en cuenta que, entre más área a cubrir, aumentará la obtención de más eventos, por ende, más cámaras serán necesarias para su instalación.



Lugar: Parqueo principal de ITCA-FEPADE

Dimensiones aprox: 15m * 80m

Área aprox: 1200 m²

Equipo necesario:

- 14 cámaras Hikvision
- 1 dispositivo Arduino

Teniendo los datos analizados, a continuación, se presenta un ejemplo detalle de los servicios que se pueden ofrecer al mercado:

Servicio

Implementación de sistema de videovigilancia mediante Visión Artificial

Detalle de servicio

Concepto	Cantidad	Precio unitario	Subtotal
Cámaras	14	\$125.28	\$1,753.92
KIT Arduino	1	\$75.00	\$75.00
Mano de obra	1	\$429.54	\$429.54
Soporte técnico	1	\$100.00	\$100.00
Licencia de software	1	\$350.00	\$350.00
Total			\$2,708.46

Tabla 19 - Detalle de servicio

El servicio al público por lo general debería incluir 12 meses de soporte técnico y de licencia renovable de software; teniendo como pago la cantidad de **\$2,708.46**, y un pago anual recurrente por uso, mantenimiento y soporte de **\$450.00**.

Siendo retribuable: la mano de obra, soporte técnico y licencia de software, por cada producto: **\$879.54**

Proyección de ventas

Si trazamos como objetivo de venta y distribución del producto en la región, por la cantidad de 150 unidades anuales, y realizando una inversión en marketing a nivel de redes sociales mensual de \$60, con el fin de potenciar la adquisición de clientes, tenemos que:

Concepto	Cantidad	Precio	Subtotal
Facebook Ads	12 meses	\$60.00	\$720.00
Producto	150	\$879.54	\$131,931.00
Total			\$131,211.00

Tabla 20 - Proyección

Proyección anual = \$131,211.00

Proyección de ganancia mensual = \$10,934.25

Con esta proyección, se puede esperar tener una ganancia aproximada de \$10,934.25 por mes en promedio.

Requisitos operativos

Requisitos operativos	
Cámara de videovigilancia	Valor
Cantidad	1
Resolución mínima	5 mpx
Conectividad	WiFi o Ethernet
Personal	Cantidad
Supervisión de eventos.	1
Revisión de grabaciones	1
Gestión de alertas.	1

Tabla 21 - requisitos operativos

Evaluación de recursos		
Equipo	Especificación	Capacidad
Servidor	Procesador	Intel® Core™i7-13700
	RAM	64.0 GB
	Tipo de sistema	Sistema operativo Windows 10 de 64 bits, Procesador x64
	Tarjeta grafica	NVIDIA GeForce RTX 4070
	Memoria de video dedicada	12GB GDDR6X
	Unidad de almacenamiento (Sistema)	Disco de Estado Sólido SSD 128 GB
	Unidad de almacenamiento (Grabaciones)	NAS (4TB iniciales)

Tabla 22 - evaluación de recursos

Evaluación y selección de equipos:	
Cámara	Hikvision H.265 Exir Mini Bullet Network Camera 5MP
Servidor	Servidor adquirido (vea evaluación de recursos)
Software	AEFP (sistema desarrollado)

Tabla 23 - selección de equipos

Instalación y configuración de equipos:	
Cámara	Se instalará en un lugar estratégico, con una visión optima a los vehículos y además se debe conectar a una intranet para acceder a ella mediante una ip.
Servidor	Se instalará en un ambiente fresco y seco. Además, se debe conectar a internet y a una intranet para acceder a la cámara.
Software	Se instalará en el servidor.

Tabla 24 - instalación y configuración de equipos

Además, capacitar al personal encargado de operar el sistema de videovigilancia. Esto incluye instrucciones sobre cómo utilizar el software, monitorear las cámaras, revisar las grabaciones, generar informes y manejar eventos o alertas

Y por último realizar pruebas exhaustivas del sistema para verificar su rendimiento y funcionalidad. Ajusta la configuración según sea necesario para garantizar la detección precisa, la calidad del video y la estabilidad del sistema.

Fase de operación

Fase de operación	
Monitoreo continuo	El personal designado debe estar atento a eventos o actividades sospechosas y responder de acuerdo con los procedimientos establecidos.
Revisión de grabaciones	Realiza revisiones periódicas de las grabaciones almacenadas para buscar eventos o actividades pasadas relevantes.
Gestión de eventos y generación de informes	Establecer un protocolo para la revisión eficiente de las grabaciones y la documentación de los eventos importantes. Crea informes detallados que documenten los eventos relevantes, las acciones tomadas y los resultados obtenidos.

Tabla 25 - fase de operación

Capacitación y formación

Se preparó materiales de capacitación claros y concisos que respalden los temas a cubrir. Esto incluye manuales de usuario y ejercicios prácticos. Asegurando que los materiales son comprensibles y están disponibles en formatos accesibles. Se utilizó una combinación de sesiones de capacitación presenciales, capacitación en línea, tutoriales en video y materiales de referencia escritos.

La factibilidad operativa de este proyecto es alta, lo que significa que es viable y práctico implementar y operar el sistema de videovigilancia en el estacionamiento. Se muestra un resumen de los aspectos clave que respaldan esta factibilidad:

- Equipos y tecnología: Existe disponibilidad de equipos y tecnología necesarios, como cámaras de videovigilancia, software de videovigilancia, servidores de almacenamiento y conectividad de red, que son adecuados para el proyecto.
- Implementación: Se cuenta con un plan de implementación detallado que incluye la selección y configuración adecuada de los equipos, la instalación física de las cámaras y la integración del software necesario.
- Capacitación del personal: Se prevé un programa de capacitación para el personal encargado de operar el sistema. Esto garantizará que estén familiarizados con el software, los procedimientos de monitoreo, la revisión de grabaciones y la gestión de eventos.
- Monitoreo y gestión de eventos: Se establece un programa de monitoreo continuo para supervisar el

estacionamiento en tiempo real y detectar eventos sospechosos.

El presenta cronograma refleja las actividades a realizar durante el periodo y fechas estimadas.

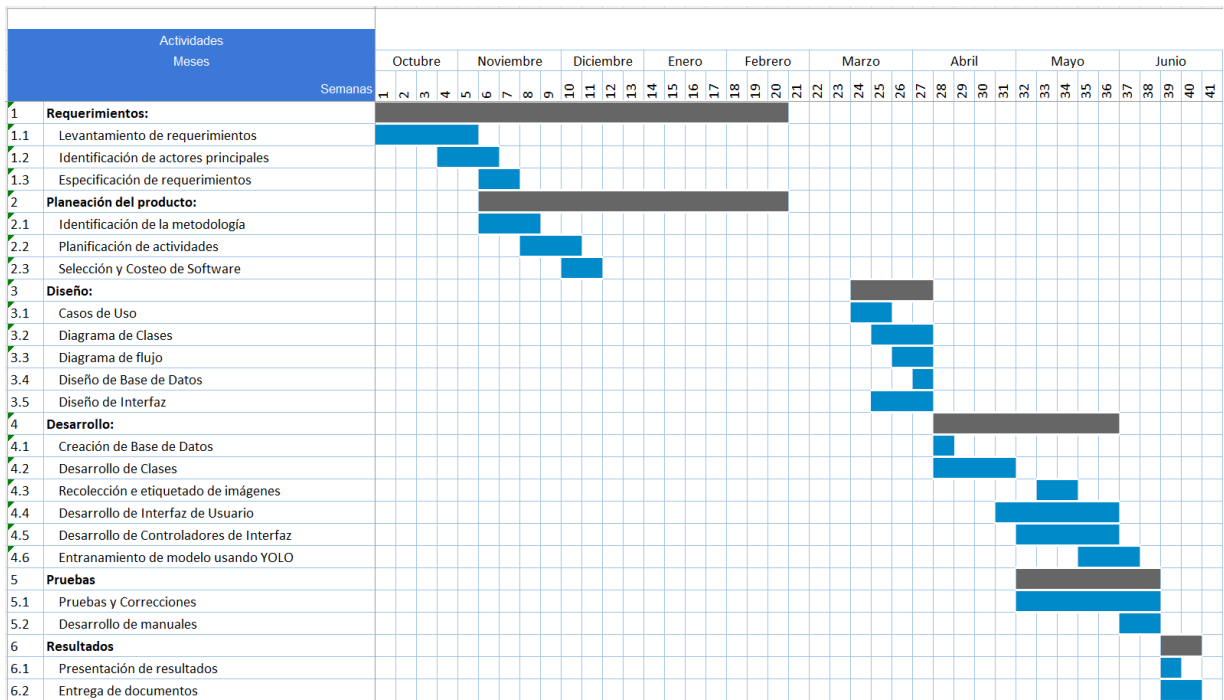


Ilustración 28 - Cronograma de actividades

CAPÍTULO III

RESULTADOS OBTENIDOS DE LA INNOVACIÓN.

Cambios en necesidades abordados

Se presentan los cambios significativos que se han observado en las necesidades que se identificaron en la descripción inicial del problema y cómo el proyecto de innovación ha contribuido a abordar estas necesidades de manera efectiva.

Es importante destacar que se trata de un prototipo el cual ayudará en gran manera si se lanza a producción ya que ahorrará tiempo a los vigilantes y volverá el estacionamiento potencialmente más seguro para los usuarios y será de gran ayuda para los vigilantes ya de esta manera pueden recibir notificaciones en tiempo real de los actos vandálicos o sucesos sospechosos.

El prototipo es capaz de detectar eventos sospechosos, pero con un número limitado de eventos así mismo a este punto del desarrollo únicamente es capaz de trabajar con una sola cámara.

De esta manera se mejoraría la capacidad de anticiparse a posibles incidentes y tomar medidas preventivas de manera oportuna, el personal podría recibir alertas de seguridad inmediatas cuando se detecta alguna actividad sospechosa, lo que les permitiría intervenir rápidamente y minimizar cualquier riesgo potencial.

El desarrollo del sistema es ideal para el estacionamiento ya que volverá más efectiva la seguridad en el parqueo ya que el prototipo de momento tiene efecto detectando los eventos sospechosos, demostrando ser eficiente en la labor que se le solicita.

El proyecto de innovación ha generado un impacto significativo en las necesidades identificadas en la descripción del problema, mejorando en la seguridad y la detección temprana

de eventos sospechosos, demostrando ser una solución sólida y efectiva que cumple con los requisitos de los estacionamientos en términos de protección y prevención de incidentes. Los resultados muestran que puede reducir el riesgo, mejorar la respuesta ante situaciones sospechosas y brindar mayor tranquilidad a los usuarios y al personal de seguridad.

Pruebas y demostraciones de la eficacia, eficiencia y efectividad

Los resultados del entrenamiento se presentan a continuación en la carpeta weights encontramos los pesos equivalentes al modelo que usaremos dentro del prototipo.

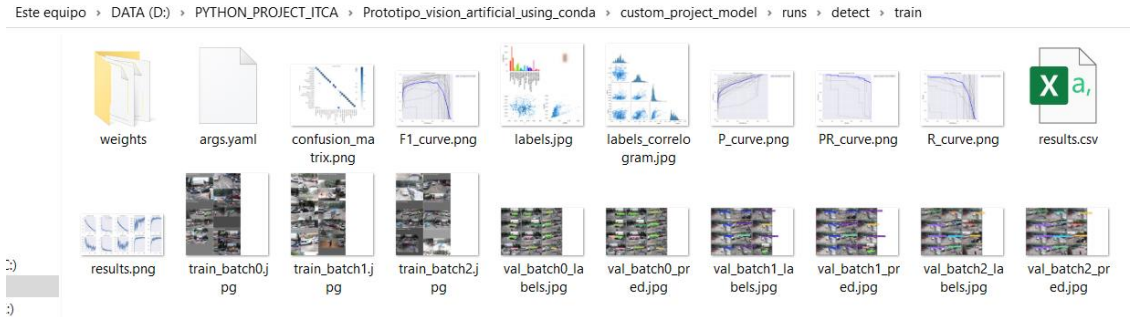


Ilustración 29 - resultados

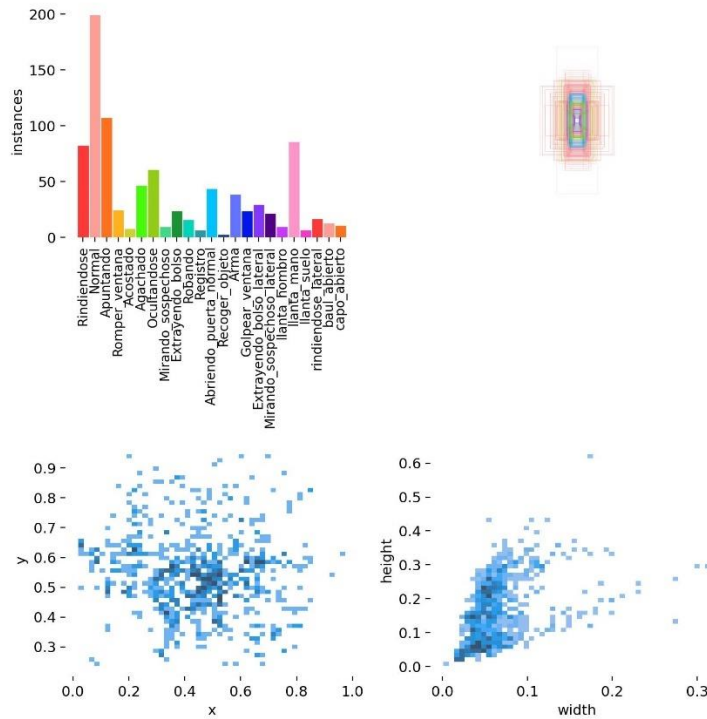


Ilustración 30 - clases e instancias

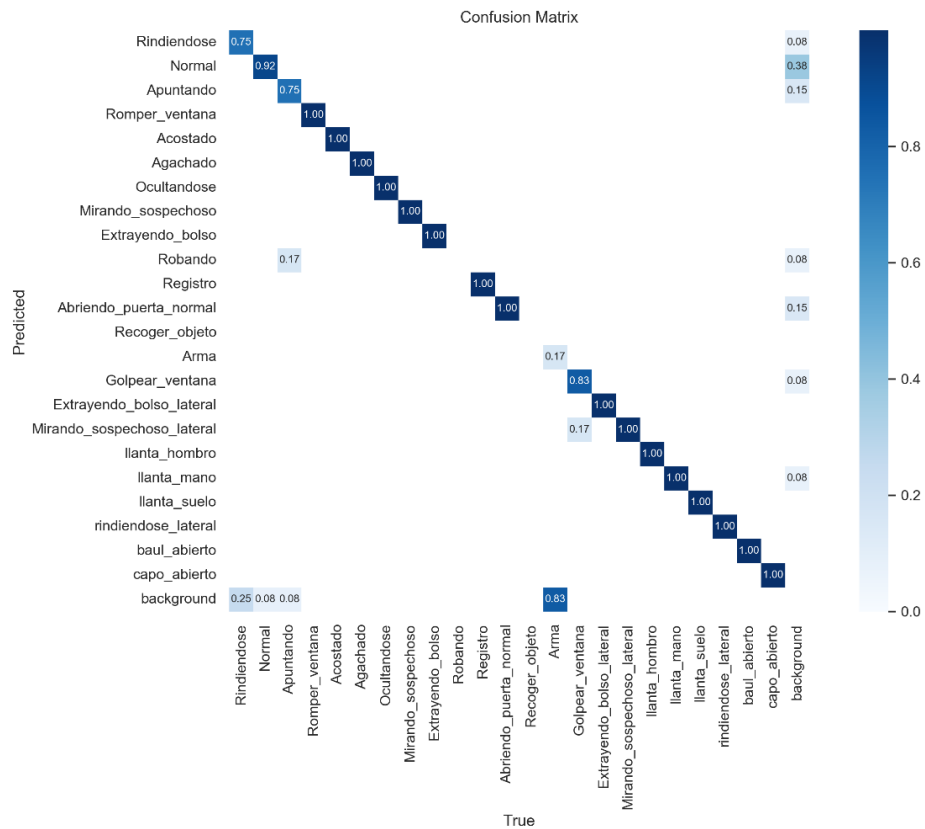


Ilustración 31 - Matriz de confusión

La matriz de confusión muestra las clases y el puntaje de confusión por cada una de ellas, dentro de ellas el valor más alto de 0.90 es más eficiente entre más bajo sea puede afectar al predecir valores con este modelo³¹.

³¹ COCO - common objects in context. (s/f). Cocodataset.org. Recuperado el 12 de junio de 2023, de <https://cocodataset.org/>

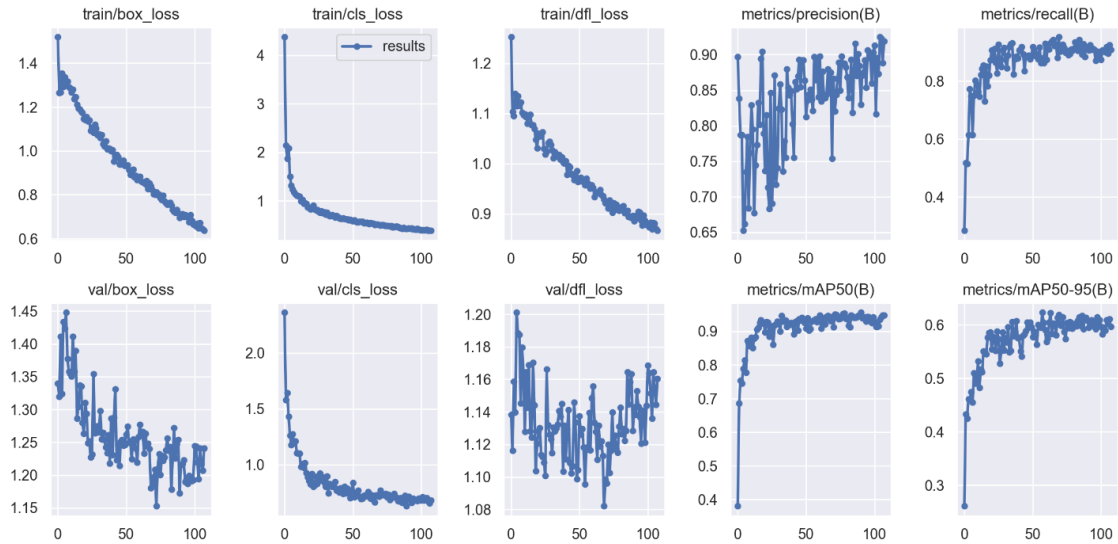


Ilustración 32 - Resultados entrenamiento modelo yolov8 personalizado

Dentro del entrenamiento encontramos:

1. **box_loss**: pérdida de regresión del cuadro delimitador (error cuadrático medio).

Dentro de los resultados la pérdida disminuye al ir aumentando las épocas de entrenamiento.

2. **cls_loss**: la pérdida de clasificación (Cross Entropy).

El modelo pierde valor cada vez que las épocas aumentan, esto indica que entre más cercano a 0 nos encontramos en cada época el modelo deja de aprender de los datos que está procesando.

3. **dfl_loss**: es el desequilibrio de clases.

En este punto se observa un desequilibrio con respecto a las imágenes que está procesando, si recibió 200 imágenes para reconocer una persona normal y 20 imágenes para reconocer una persona acostada, cada vez que quiera conocer una persona acostada aumenta la pérdida porque a los datos menos frecuente intentara darle más importancia en el aprendizaje esto se puede

observar más en la validación pues los datos son menores respecto al entrenamiento.

4. Precisión: mide cuanto de las predicciones del cuadro delimitador (bbox) son correctas.

5. Recall: mide cuanto de los verdaderos del cuadro delimitador (bbox) se predijeron correctamente.

La precisión tiende a tener valores entre 0.75 a 0.95 indica que no está muy baja sin embargo es posible tener falsos positivos para algunas clases al momento de realizar predicciones las cuales puede indicar solo algunas muestras positivas, esto se debe a la alta recuperación que se muestra en la gráfica recall.

6. mAP (mean Average Precision) es la precisión media en el umbral IoU (intersección sobre unión) de 0.5 (mAP 50:95) es el promedio de mAP sobre diferentes umbrales de IoU, que van de 50 a 95

Se utiliza como umbral para determinar si un resultado previsto es un verdadero positivo o un falso positivo. La relación IoU es la cantidad de superposición entre el cuadro de delimitación alrededor de un objeto previsto y el cuadro de delimitación alrededor de los datos de referencia del terreno³².

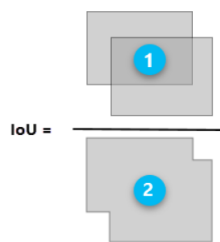


Ilustración 33 - Intersección sobre unión

³² Cómo funciona Calcular precisión para la detección de objetos. (s/f). Arcgis.com. Recuperado el 12 de junio de 2023, de <https://pro.arcgis.com/es/pro-app/latest/tool-reference/image-analyst/how-compute-accuracy-for-object-detection-works.htm>

Resultados de las pruebas.

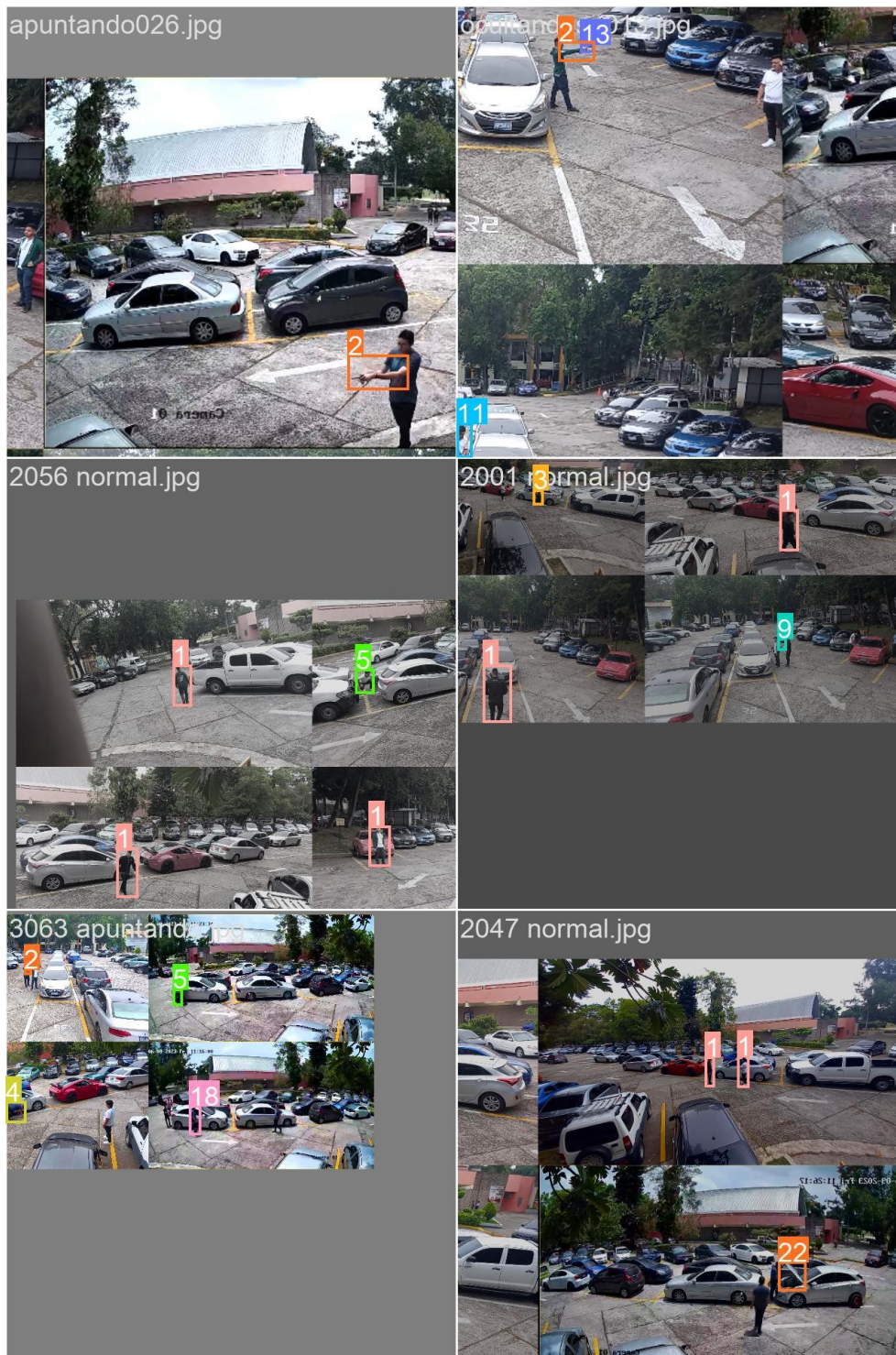


Ilustración 34 - entrenamiento 1



Ilustración 35 - entrenamiento 2



Ilustración 36 validación 1



Ilustración 37 - validación 2

Dentro de las pruebas con este modelo se obtuvieron los siguientes resultados en un prototipo funcional.

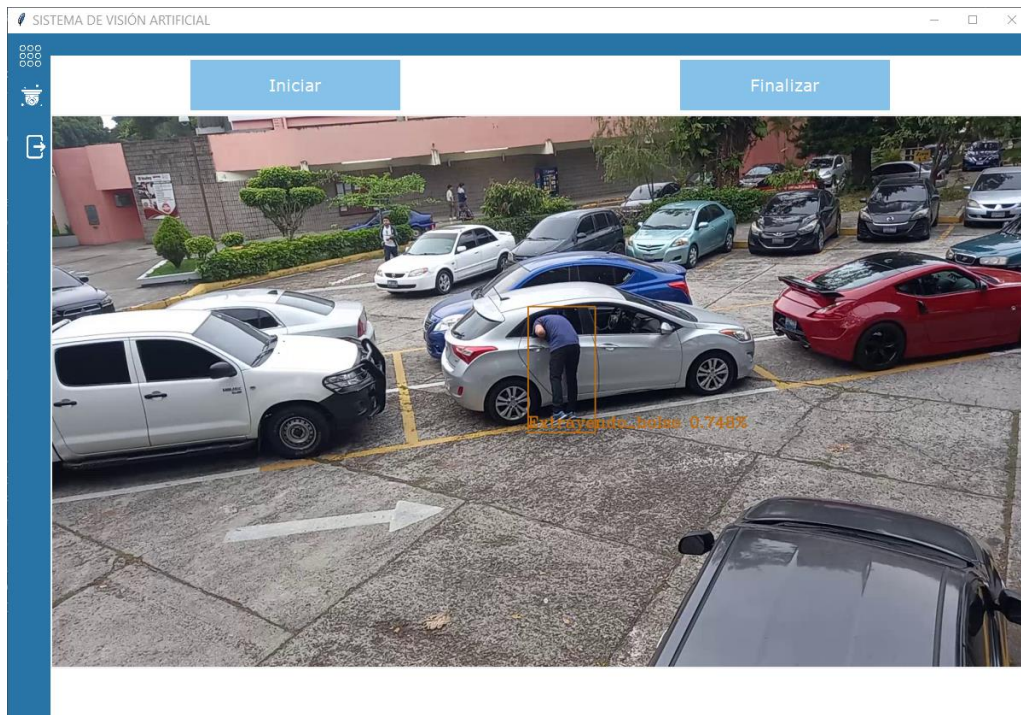


Ilustración 38 - prueba

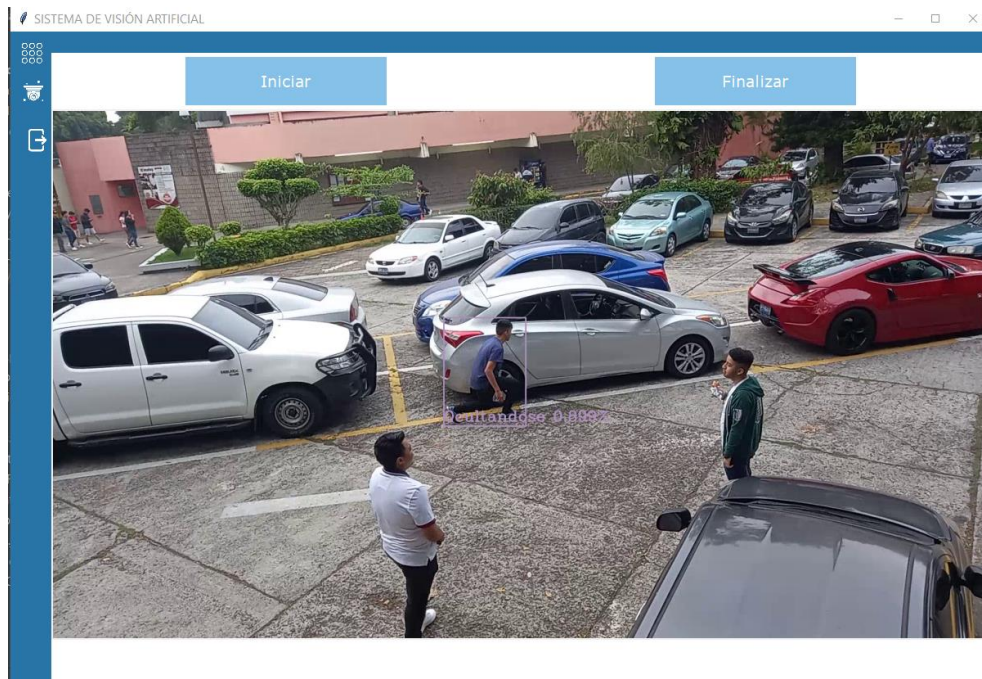


Ilustración 39 - prueba

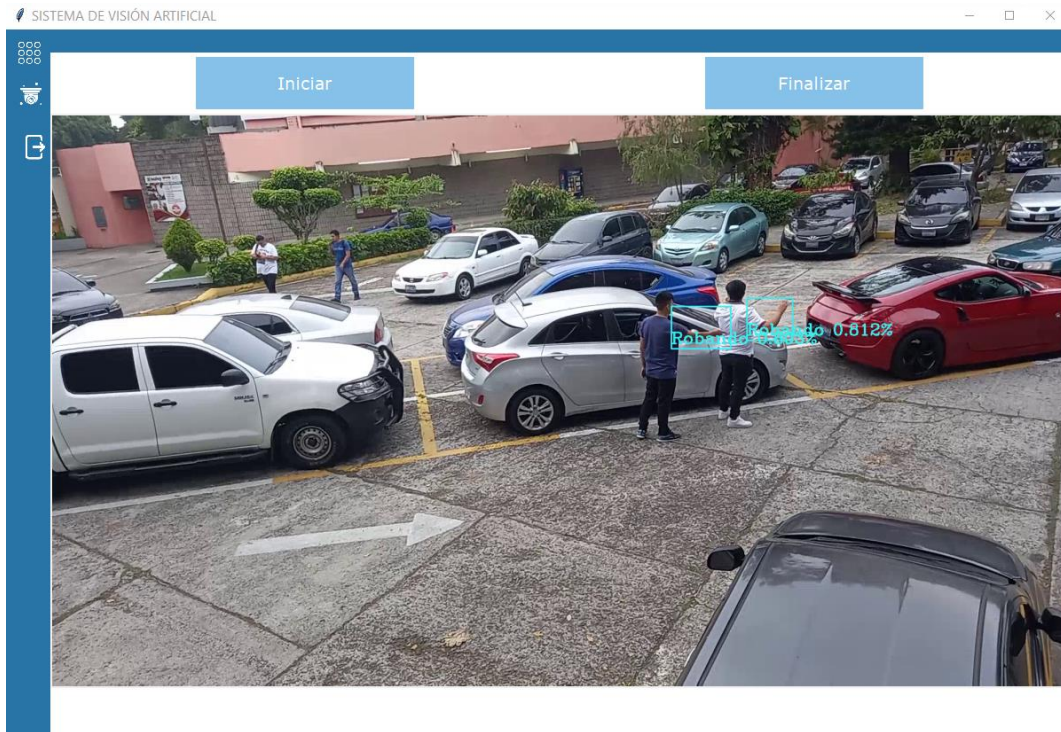


Ilustración 40 - prueba

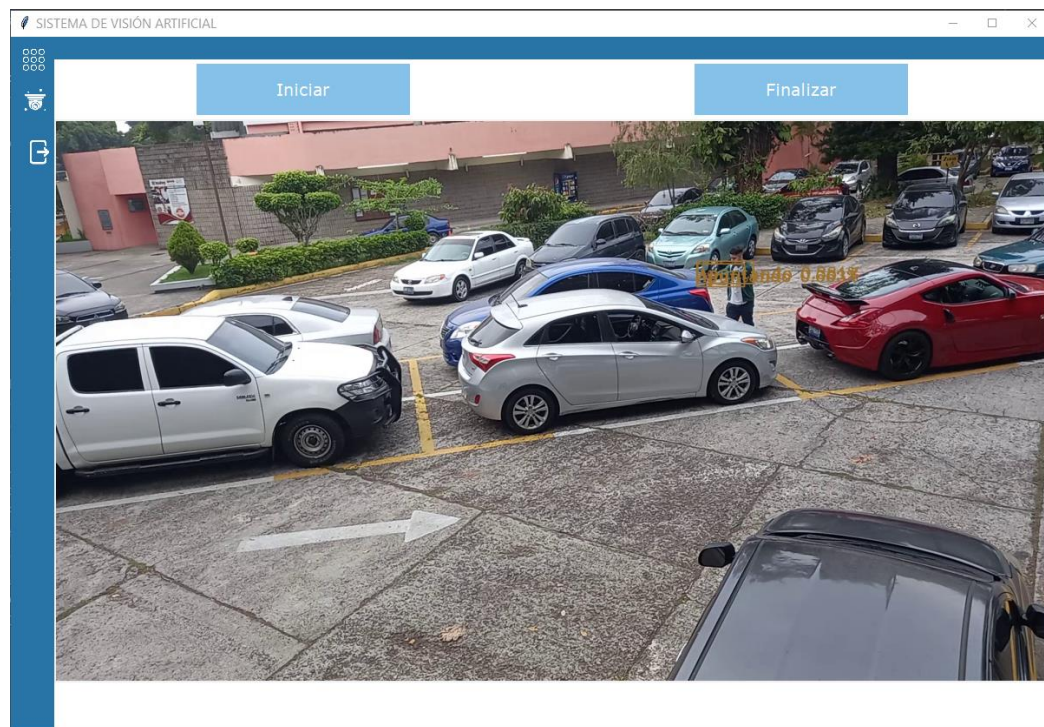


Ilustración 41 - prueba

Percepciones y evaluación del usuario

El viernes 09 de junio del 2023 se realizó una entrevista con el encargado de seguridad del estacionamiento de ITCA-FEPAD, se abordaron preguntas relacionadas con la seguridad en el estacionamiento y experiencias en otros estacionamientos que fueran de utilidad para integrar nuevos eventos al prototipo.

Dentro de las preguntas realizadas se consultó, ¿Considera que un sistema de alerta automatizada en tiempo real ayudaría a la seguridad en los estacionamientos? La percepción del encargado de seguridad fue que está muy bien un sistema de alerta pues ayudaría a previsualizar sucesos y el encargado de monitorizar las cámaras de seguridad podrá comunicarse por radio al personal de seguridad más cercano de esa forma tomar acciones rápidas, igual forma la activación de un sonido de alarma ayudaría a verificar en el momento si llegara a ocurrir algún suceso. Así mismo agrego que se tendrá un mayor control y ayudaría a los protocolos ya establecidos en ITCA como en otros estacionamientos donde se pueda usar un sistema con estas características.

Conclusiones

Con este proyecto innovador se obtuvo conocimiento sobre las nuevas tecnologías referente a la visión artificial, fue un reto enorme el desarrollo de este, llenó de satisfacción el resultado ya que mejora significativamente la detección temprana de eventos sospechosos y por lo tanto fortalece la seguridad en las áreas en las cuales se puede implementar para este proyecto será el estacionamiento de ITCA-FEPADE.

El prototipo de videovigilancia para ITCA-FEPADE fue un éxito después de diversos obstáculos, el prototipo analiza las imágenes de cámaras colocadas en lugares estratégicos para tener una visión buena del estacionamiento y mediante el uso de algoritmos de detección de movimiento y objetos mediante los cuales el prototipo es capaz de detectar situaciones potencialmente peligrosas y generar alertas para una respuesta rápida y eficiente.

Para el desarrollo de este cabe mencionar que algunas de las limitaciones más importante que tuvimos fue el tiempo de desarrollo del sistema de videovigilancia ya que este fue un poco limitado lo cual afecto en la cantidad de eventos que detecta el sistema y la cantidad de pruebas realizadas.

En el proceso de desarrollo del prototipo se obtuvo un aprendizaje significativo de todas estas nuevas tecnologías, así como de las capacidades como desarrolladores ya que impulsó a adquirir y estudiar conocimientos especializados para implementar y configurar el sistema de manera efectiva. Se Afronto con contingencias que requerían soluciones creativas y flexibilidad para superar los obstáculos y garantizar un rendimiento óptimo del sistema que fueron solventadas con éxito.

En resumen, el proyecto de innovación ha resultado exitoso en el desarrollo de un sistema de videovigilancia de aparcamientos basado en tecnologías de visión artificial. El prototipo promete mejorar la seguridad en el área de estacionamiento de ITCA-FEPADE. El proyecto impulsa a dar un paso más como desarrolladores y se concluye que gracias al conocimiento y la experiencia se sentaron unas bases sólidas para futuros desarrollos y mejoras en el campo de la videovigilancia y la visión artificial en general.

Recomendaciones y propuestas

Desarrollar una aplicación móvil

Sería ideal desarrollar una aplicación móvil que permita a los usuarios acceder y recibir notificaciones sobre las condiciones de estacionamiento en tiempo real aumentando así la flexibilidad para los usuarios y el personal de seguridad ya que en la actualidad siempre andamos un smartphone en nuestros bolsillos.

Integrar sistemas de detección de placas de vehículos

Integrar un sistema de control de acceso al estacionamiento, se puede lograr mediante la integración de sistemas de detección de placas de vehículos. De esta manera se podría tener un mayor control sobre los vehículos que circulan en el estacionamiento y a su vez puede ayudar en tareas más simples como indicar a donde dejo el cliente su automóvil.

Explorar técnicas de seguimiento de objetos

Ampliar la detección y clasificación de objetos puede traer una mejora muy grande al proyecto ya que actualmente es capaz de reconocer una cantidad muy reducida de estos, se podría investigar y desarrollar técnicas de seguimiento de objetos en tiempo real y también mejorar la calidad y la cantidad de cámaras del estacionamiento para llevar a cabo este propósito. Esto sería útil para detectar armas blancas o armas de fuego con antelación.

Mejorar la eficiencia del sistema

El proyecto tiene un amplio margen de mejora ya que no fue realizado por expertos en área, por lo cual se considera que hay mucho margen para mejorar la eficiencia de este y así consumir menos recursos del servidor, reduciendo así el tiempo de respuesta de detección de eventos, optimizando los algoritmos de procesamiento de imágenes.

Detección de colisiones o accidentes

Una función interesante podría ser la introducción de algoritmos y técnicas de procesamiento de imágenes para detectar colisiones o accidentes entre vehículos en un estacionamiento, ayudando así a responder rápidamente en situaciones de emergencia y así mejorar la seguridad de los usuarios en el estacionamiento.

Socialización de los resultados

Plan de Socialización

Identificación de la audiencia objetivo

- **Comunidad académica:** Investigadores, profesores y estudiantes relacionados con el desarrollo de software, campo de visión artificial, sistemas de videovigilancia y tecnologías afines.
- **Personal de la institución:** Directivos, profesores y personal administrativo involucrado en la implementación y desarrollo del proyecto.
- **Partes externas interesadas:** Empresas del sector, organizaciones relacionadas con la seguridad y la tecnología.

Selección de estrategias de socialización

- **Publicaciones científicas:** Elaboración de artículos técnicos y científicos para su publicación en revistas especializadas, brindando información técnica sobre el proyecto, los métodos utilizados y los resultados alcanzados.
- **Eventos internos:** Organizar reuniones y presentaciones internas para que ITCA-FEPADE comparta los resultados del proyecto, genere debates y recopile ideas y sugerencias.

- **Demostraciones prácticas:** La demostración de un prototipo desarrollado permite a las partes interesadas verlo en acción en tiempo real y comprender sus capacidades y beneficios.
- **Informes técnicos:** Crear un informe que refleje los aspectos técnicos del proyecto entre los cuales tenemos: descripción del sistema, la metodología implementada, los resultados obtenidos, conclusiones obtenidas y recomendaciones sugeridas.

Cronograma de socialización

- Definir un cronograma que incluya las fechas clave para la participación en conferencias y eventos, envío de artículos científicos, organización de demostraciones prácticas y presentaciones internas.

Evaluación y seguimiento

- A partir de las evaluaciones y sugerencias obtenidas realizar las mejoras y actualizaciones necesarias para el correcto funcionamiento y desarrollo del prototipo.
- Realizar ajustes en el plan de socialización según los resultados obtenidos, identificando áreas de mejora y oportunidades adicionales de difusión de los resultados del proyecto.

ARTÍCULO PARA LA PUBLICACIÓN

Prototipo de sistema de videovigilancia automatizado por medio de visión artificial para ITCA-FEPADE.

Resumen

El presente artículo presenta un estudio detallado sobre el desarrollo de un sistema de visión artificial para la videovigilancia automatizada en un entorno de estacionamiento, el objetivo principal del prototipo es detectar y prevenir eventos sospechosos o peligrosos en tiempo real, brindando una mayor seguridad y eficiencia en la gestión de los espacios de estacionamiento.

Introducción

Se requieren más soluciones avanzadas para prevenir cualquier situación riesgosa. Ahí es donde entra la visión artificial, una tecnología increíble que puede analizar y procesar automáticamente imágenes y videos para identificar cosas y eventos importantes. El enfoque de este innovador proyecto es el desarrollo de un prototipo de sistema de visión artificial basado en redes neuronales para la videovigilancia de los estacionamientos del ITCA-FEPADE.

Como lenguaje de programación se ha utilizado Python en conjunto con la biblioteca OpenCV y las redes neuronales convolucionales (CNN) se puede desarrollar sistemas de visión artificial. Se opto por trabajar con Python y OpenCV porque pueden procesar imágenes y videos de manera eficiente además por la amplia documentación de estos y las redes neuronales convolucionales se han utilizado para entrenar un modelo que ayudara a detectar y reconocer objetos en las imágenes capturadas

por la cámara de videovigilancia. Se desarrolla un sistema de alerta que será capaz de enviar notificaciones a través de correo electrónico, SMS y activar alertas físicas mediante una alarma programada con Arduino.

Cabe destacar importancia de seleccionar cuidadosamente el equipo a utilizar. Para asegurar la selección anterior, se ha apegado al estándar DORI (Distancia, Observación, Reconocimiento e Identificación). Este estándar ayuda a definir los criterios y parámetros que permiten evaluar la calidad y capacidad de los sistemas de videovigilancia.

Marco teórico

Se utiliza OpenCV³³, librería de visión artificial y procesamiento de imágenes, que proporciona una amplia gama de funciones y algoritmos para la detección de objetos, reconocimiento facial y manipulación de imágenes y videos, se emplean redes neuronales convolucionales (CNN)³⁴ para el reconocimiento de objetos en tiempo real, aprovechando la capacidad de estas redes para aprender y extraer características relevantes de las imágenes. Se utilizó Django³⁵, un framework de desarrollo web en Python³⁶ que facilita la creación de interfaces

³³ OpenCV team. (2023). ABOUT. Open CV. <https://opencv.org/about/>

³⁴ ¿Qué son las redes neuronales convolucionales? (s/f). Ibm.com. Recuperado el 12 de junio de 2023, de <https://www.ibm.com/es-es/topics/convolutional-neural-networks>

³⁵ Django Software Foundation and individual contributors. (2023). Django. Django. <https://www.djangoproject.com/>

³⁶ Documentation. (s/f). Python.org. Recuperado el 12 de junio de 2023, de <https://docs.python.org/3/>

interactivas, el manejo de formularios y la interacción con la base de datos.

¿Qué es la visión artificial?

La visión artificial es un campo de la inteligencia artificial (IA) que permite a los ordenadores y sistemas extraer información significativa a partir de imágenes digitales, videos y otras entradas visuales, y tomar medidas o realizar recomendaciones en función de esa información. Si la IA permite a los ordenadores pensar, la visión artificial les permite ver, observar y comprender. La visión artificial funciona de manera muy similar a la visión humana, excepto que los humanos tienen una ventaja inicial³⁷.

Redes neuronales convolucionales

Los ordenadores usan filtros o lentes para ver las diferentes características de las redes neuronales convolucionales: uno ve los bordes diagonales, otro los colores, etc. Las CNN son una serie de redes que fueron creadas pensando en cómo funciona el cerebro, capaces de aprender en los diferentes niveles de abstracción: en la primera capa se diferencian formas simples, colores o bordes; en la siguiente se pueden distinguir combinaciones de bordes y colores; mientras que la última capa se fija en la forma con el fin de conseguir

³⁷ Bernard Marr . (2019). 7 Amazing Examples Of Computer And Machine Vision In Practice. FORBES. <https://www.forbes.com/sites/bernardmarr/2019/04/08/7-amazing-examples-of-computer-and-machine-vision-in-practice/?sh=5915a2d81018>

averiguar qué es exactamente. Las CNN están diseñadas para procesar datos en forma de cuadrículas o matrices, como imágenes.

Las principales características de las redes neuronales convolucionales son las siguientes:

- **Capas convolucionales:** Utilizan operaciones de convolución para aplicar filtros o núcleos a la imagen de entrada, extrayendo características como bordes, texturas o patrones relevantes.
- **Capas de pooling:** Reducen la dimensionalidad de las características obtenidas, disminuyendo la cantidad de parámetros y manteniendo la información más relevante.
- **Capas totalmente conectadas:** En la etapa final de la red, se utilizan capas completamente conectadas para realizar la clasificación o la predicción final basada en las características extraídas.

La arquitectura de una CNN generalmente consta de varias capas convolucionales y de pooling, seguidas de una o varias capas completamente conectadas. A medida que los datos fluyen a través de la red, las capas convolucionales aprenden a reconocer características cada vez más complejas, lo que permite una representación jerárquica y progresiva de la imagen. Una de las principales ventajas de las CNN es su capacidad para aprender de forma automática características relevantes de las imágenes, sin necesidad de una extracción manual de características³⁸.

³⁸ NAB. (2018). ¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador. aprendemachinlearning. <https://www.aprendemachinlearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>

YOLO ("Solo miras una vez")

A diferencia de los métodos tradicionales que requieren secuencias separadas de pasos para reconocer y clasificar objetos, YOLO aborda este problema de manera integral, realizando ambos procesos en un solo paso. YOLO es un algoritmo de detección de objetos de video e imágenes en tiempo real de código abierto. Utiliza una red neuronal convolucional que divide la imagen en una cuadrícula y asigna cajas delimitadoras (bounding boxes) a las regiones donde se encuentra un objeto. YOLO utiliza una red neuronal convolucional para predecir las coordenadas del cuadro delimitador, la clase del objeto y la probabilidad de detección de cada objeto en la imagen³⁹.

Ultralytics es una librería de Python que permite realizar la implementación de YOLO a través del modelo YOLOv8, el cual admite varios modos que se pueden usar para realizar diferentes tareas. Estos modos son:

Entrenar: para entrenar un modelo YOLOv8 en un conjunto de datos personalizado.

- Val: para validar un modelo YOLOv8 después de haberlo entrenado.
- Predecir: Para hacer predicciones usando un modelo YOLOv8 entrenado en nuevas imágenes o videos.
- Exportar: para exportar un modelo YOLOv8 a un formato que se puede usar para la implementación.

³⁹ Yolo v8. (s/f). Ultralytics. de <https://ultralytics.com/yolov8> (S/f-a). Uab.cat. Recuperado el 6 de junio de 2023, de https://ddd.uab.cat/pub/tfg/2017/tfg_71066/paper.pdf

- **Track:** Para rastrear objetos en tiempo real usando un modelo YOLOv8.
- **Benchmark:** para comparar la velocidad y la precisión de las exportaciones de YOLOv8 (ONNX, TensorRT, etc.).

OpenCV (Biblioteca de visión artificial de código abierto)

Es una biblioteca de procesamiento de imágenes y visión por computadora de código abierto que proporciona una amplia gama de funciones y algoritmos para manipular imágenes y videos. Esta es una biblioteca de código abierto que proporciona herramientas y algoritmos para la visión artificial y el procesamiento de imágenes. Un producto con licencia de Apache 2, OpenCV facilita a las empresas el uso y la modificación de su código. Obtenga imágenes de escenas completas, busque imágenes similares en bibliotecas de imágenes, use el flash para eliminar los ojos rojos en las imágenes, rastree el movimiento de los ojos, reconozca paisajes, establezca marcadores para superponer con realidad aumentada y mucho más.

Algunas características y funcionalidades clave de OpenCV incluyen⁴⁰:

- Procesamiento de imágenes
- Detección de objetos
- Seguimiento de objetos
- Reconocimiento de patrones
- Calibración de cámaras

⁴⁰ Rubén Crespo Cano. (2017). CURSO DE INTRODUCCIÓN A OPENCV Y PYTHON. [Tesis doctoral o de maestría, UNIVERSIDAD DE ALICANTE]. <https://github.com/rcrespocano/opencv-python/blob/master/opencv-and-python.ipynb>.

- Visión estéreo
- Interoperabilidad

MTCNN

Las redes neuronales convolucionales en cascada multitarea, o MTCNN, han demostrado su eficacia en la detección de rostros en una variedad de puntos de referencia. La salida de esta red nos da todos los posibles candidatos de los rostros. En la segunda etapa de la red, se utiliza la salida de la primera etapa como entrada de una red CNN y a esta se le llama "The Refine Network (R-Net)", esta reduce el número de posibles candidatos, calibra las bounding box y se hace una supresión no máxima (NMS) para unir nuestros posibles candidatos⁴¹.

Evento sospechoso

Un evento sospechoso en un sistema de videovigilancia es cualquier actividad o comportamiento que pueda indicar un accidente, una brecha de seguridad o una actividad delictiva. Algunos ejemplos de eventos sospechosos pueden incluir:

Movimiento inusual: Detección de movimientos que no siguen los patrones normales, como personas merodeando en áreas restringidas o comportamientos erráticos.

Reconocer que un sujeto que se aproxime a las puertas o ventanas de un vehículo lleve en sus manos alguna herramienta

⁴¹ Paz Centeno, I. (s/f). mtcnn: MTCNN face detection implementation for TensorFlow, as a PIP package.

como algún destornillador, cuchillo, varilla, arma de fuego o incluso una roca.

Entre otros.

TWILIO

- **Twilio:** Twilio es una plataforma de comunicaciones en la nube que proporciona servicios para enviar y recibir mensajes de texto (SMS), así como realizar y recibir llamadas telefónicas. Permite la integración de la funcionalidad de mensajería y llamadas en aplicaciones y **sistemas.**
- **API de Twilio:** Twilio ofrece una API (Application Programming Interface) que permite a los desarrolladores interactuar con sus servicios y enviar mensajes de texto desde aplicaciones mediante programación. Esta API proporciona métodos y parámetros específicos para enviar mensajes de texto a través de la plataforma Twilio.
- **Autenticación y credenciales de Twilio:** Para utilizar la API de Twilio, es necesario contar con las credenciales de autenticación proporcionadas por la plataforma. Estas credenciales incluyen un identificador de cuenta (Account SID) y un token de autenticación (Auth Token), que se utilizan para autorizar las solicitudes de envío de mensajes de texto.
- **Integración de Twilio en el sistema:** Para integrar Twilio en el sistema de detección de eventos sospechosos, es necesario utilizar el SDK (Software Development Kit) de Twilio para el lenguaje de programación utilizado en el proyecto. A través del SDK, se pueden implementar los

métodos y funciones necesarios para enviar mensajes de texto a través de la API de Twilio.

Estándar DORI

El estándar DORI (Distancia, Observación, Reconocimiento, Identificación) es una métrica utilizada en videovigilancia para evaluar y establecer los requisitos de calidad y rendimiento de los sistemas de cámaras de seguridad. Proporciona una guía para determinar la idoneidad de una cámara o sistema de videovigilancia en función de la distancia a la que se encuentra el objeto de interés y los detalles que se pueden capturar en diferentes escenarios. El estándar DORI ayuda a los profesionales de la seguridad y a los instaladores de sistemas de videovigilancia a evaluar y seleccionar las cámaras adecuadas para diferentes escenarios y necesidades⁴².

Metodología de la investigación

Se presenta la metodología y estrategia utilizada para el desarrollo del proyecto de videovigilancia en estacionamientos. Se describen las técnicas y métodos empleados, así como las tecnologías y herramientas seleccionadas. Se explica cómo estos enfoques fueron apropiados para alcanzar los objetivos establecidos, garantizando un sistema eficiente y efectivo de videovigilancia.

⁴² Argos.red. DORI en camaras CCTV. <https://argos.red/dori-en-camaras-cctv/>

- **Definición de requerimientos**

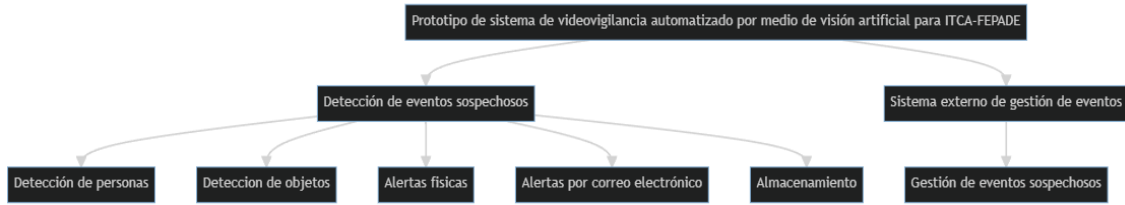
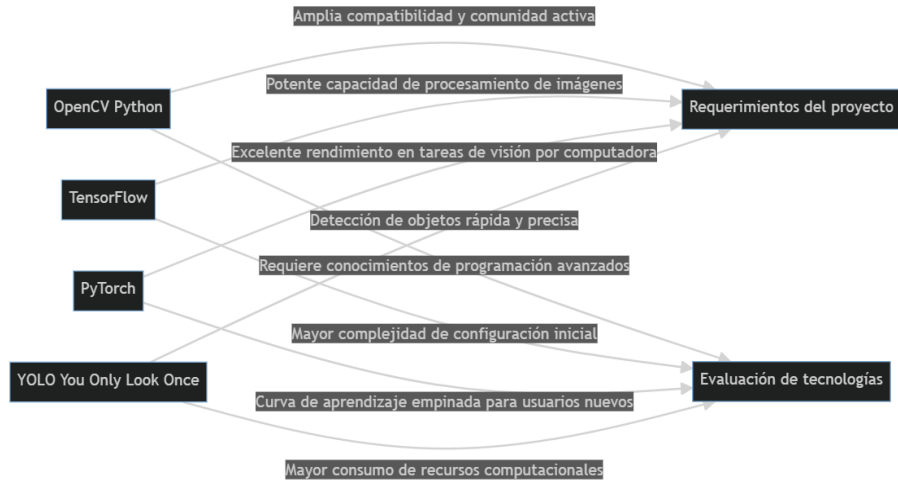


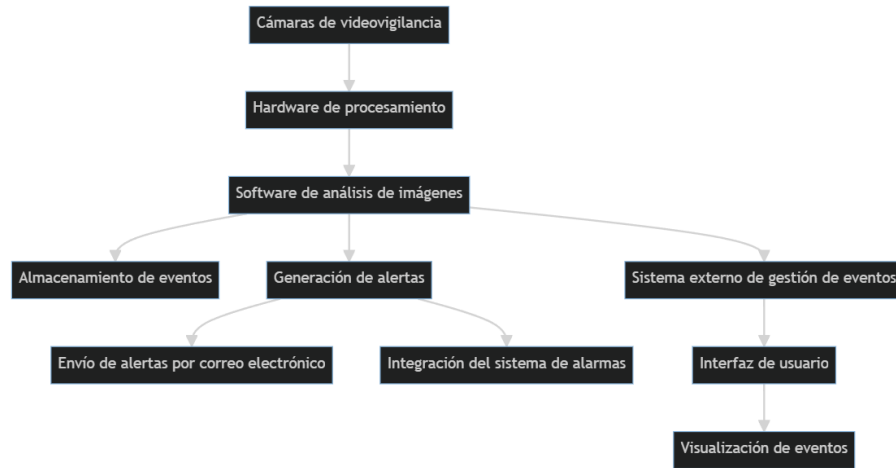
Diagrama de requerimientos

- **Investigación y selección de tecnologías**



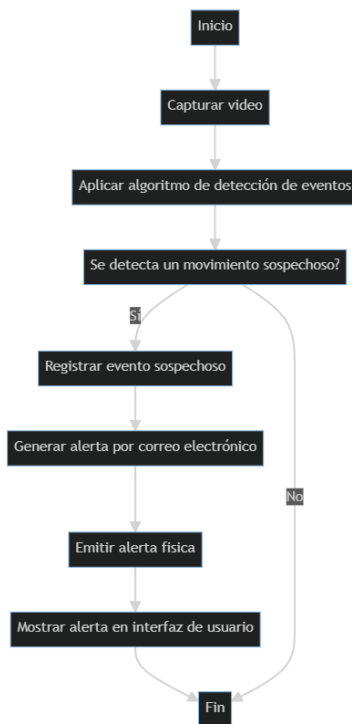
Selección de tecnologías

- **Diseño del sistema**



Arquitectura del sistema

- **Desarrollo del prototipo**



Proceso de detección de eventos

- **Evaluación y validación del prototipo**



Comparación de algoritmos de detección

Resultados y análisis.

Rendimiento del sistema: Se realizó pruebas para evaluar la precisión y eficacia del sistema en la detección de eventos sospechosos obteniendo resultados satisfactorios.

Sistema de alertas: Probar los procesos establecidos, incluidas las notificaciones electrónicas y físicas, arrojó buenos resultados en términos de precisión y eficiencia.

Tiempos de respuesta: El tiempo de respuesta se midió desde el momento en que el sospechoso planteó el incidente hasta que se emitió la alerta, y estamos muy satisfechos con los resultados, ya que se respondió al incidente de inmediato.

Limitaciones y mejoras potenciales: Se identificaron algunas limitaciones del sistema, como la dependencia de condiciones óptimas de iluminación y la necesidad de hardware especial para su implementación.

Conclusiones

Se desarrollo un prototipo de sistema de visión artificial mediante el uso de redes neuronales convoluciones con el cual podemos detectar eventos sospechosos en tiempo real.

Los sistemas de visión artificial resultan muy útiles para los estacionamientos en general ya que ayudan en gran manera al personal de seguridad y ayudan a tener un mejor control sobre los eventos sospechosos que sucedan en el área dando como resultado una respuesta mucho más rápida a disco eventos.

La integración de herramientas como Python, OpenCV y Django facilito el desarrollo y mejoro la compatibilidad entre los aplicativos desarrollados proporcionando una plataforma flexible y extensible para una mayor mejora y personalización.

El uso de algoritmos avanzados para el análisis de movimientos nos ha ayudado a mejorar la eficiencia y precisión del análisis de imágenes.

La viabilidad y aplicabilidad de los sistemas de visión artificial en un estacionamiento sugiere que puede resultar altamente efectiva la implementación en otras áreas.

Referencias

¿Qué son las redes neuronales convolucionales? (s/f). Ibm.com. Recuperado el 12 de junio de 2023, de <https://www.ibm.com/es-es/topics/convolutional-neural-networks>

About. (2019, marzo 6). OpenCV. <https://opencv.org/about/>

Argos.red. DORI en camaras CCTV. <https://argos.red/dori-en-camaras-cctv/>

Bernard Marr . (2019). 7 Amazing Examples Of Computer And Machine Vision In Practice. FORBES. <https://www.forbes.com/sites/bernardmarr/2019/04/08/7-amazing-examples-of-computer-and-machine-vision-in-practice/?sh=5915a2d81018>

Django Software Foundation and individual contributors. (2023). Django. Django. <https://www.djangoproject.com/>

Documentation. (s/f). Python.org. Recuperado el 12 de junio de 2023, de <https://docs.python.org/3/>

NA8. (2018). ¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador. aprendemachinelearning. <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>

OpenCV team. (2023). ABOUT. Open CV. <https://opencv.org/about/>

Paz Centeno, I. (s/f). mtcnn: MTCNN face detection implementation for TensorFlow, as a PIP package.

Rubén Crespo Cano. (2017). CURSO DE INTRODUCCIÓN A OPENCV Y PYTHON. [Tesis doctoral o de maestría, UNIVERSIDAD DE ALICANTE]. <https://github.com/rcrespocano/opencv-python/blob/master/opencv-and-python.ipynb>.

Yolo v8. (s/f). Ultralytics. de <https://ultralytics.com/yolov8> (S/f-a). Uab.cat. Recuperado el 6 de junio de 2023, de https://ddd.uab.cat/pub/tfg/2017/tfg_71066/paper.pdf

MANUAL DE PROGRAMADOR

APLICATIVO SERVIDOR

ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA-FEPADE

ESCUELA DE COMPUTACIÓN



MANUAL DE PROGRAMADOR

Prototipo de sistema de videovigilancia automatizado por medio
de visión artificial para ITCA-FEPADE

PRESENTADO POR

Edgardo Antonio Argueta Bautista

Jefferson Francisco Pineda Quintanilla

Kevin Rodrigo Flores Acosta

Nicolás Gabriel Escobar Pérez

Versión 1.0

Santa Tecla, La Libertad, El Salvador.

ÍNDICE

OBJETIVO	142
ALCANCE DEL DOCUMENTO	143
DEFINICIONES	144
REQUERIMIENTOS TÉCNICOS DEL SISTEMA.....	145
REQUISITOS DE HARDWARE	145
REQUISITOS DE SOFTWARE	146
PROGRAMACIÓN.....	147
DIAGRAMA DE BASE DE DATOS	147
PROCESO ENTRENAMIENTO DE MODELO USANDO YOLO V8	147
ENTRENAMIENTO YOLOV8 COMANDOS	151
PROGRAMACIÓN DEL PROTOTIPO DE VISIÓN ARTIFICIAL	155
PAQUETES UTILIZADOS	157
CONFIGURACIÓN EN PYCHARM	159
DESCRIPCIÓN DE FUNCIONES POR ARCHIVO.....	165
MAIN.PY	165
CONNECTION.PY.....	168
FUNCTION_DB.PY.....	168
LOGIN_FUNCTIONS.PY.....	170
PARAMETROS.PY.....	170
SEND_MAIL.PY	171
CONFIGURACIONES	172
INSTALACIÓN DE VISUAL STUDIO	172
INSTALACIÓN DE CUDA Y CUDNN	174
INSTALACIÓN Y CONFIGURACIÓN DE ANACONDA3	185

INSTALACIÓN DE PYTHON	190
INSTALACIÓN DE PYCHARM	193
INSTALACIÓN DE MYSQL	197
INSTALACIONES OPCIONALES	205
CREACIÓN DE ARCHIVO EJECUTABLE	211
CONCLUSIÓN.....	215

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 42	DIAGRAMA DE RELACIONAL DE BASE DE DATOS	147
ILUSTRACIÓN 43	ESTRUCTURA DE PROYECTO DE ENTRENAMIENTO	148
ILUSTRACIÓN 44	MODELOS ULTRALYTICS	148
ILUSTRACIÓN 45	LISTADO DE CLASES	150
ILUSTRACIÓN 46	ARCHIVO DATA_CUSTOM.YAML	150
ILUSTRACIÓN 47	ARCHIVO TRAIN.PY	150
ILUSTRACIÓN 48	ARCHIVO TEST.PY	151
ILUSTRACIÓN 49	CONFIGURACIÓN DESCARGA PYTORCH	152
ILUSTRACIÓN 50	RESULTADO DE ENTRENAMIENTO	154
ILUSTRACIÓN 51	ESTRUCTURA DEL SISTEMA PROTOTIPO	155
ILUSTRACIÓN 52	CONFIGURACIÓN ABRIR PROYECTO	159
ILUSTRACIÓN 53	SELECCIONAR PROYECTO	159
ILUSTRACIÓN 54	CONFIAR EN LOS ARCHIVOS	160
ILUSTRACIÓN 55	ABRIR PROYECTO NUEVA VENTANA	160
ILUSTRACIÓN 56	CONFIGURACIÓN COMPILADOR BASE	161
ILUSTRACIÓN 57	ALERTA DE ERROR, CORRECCIÓN.	161
ILUSTRACIÓN 58	INSTALACIÓN PYTORCH EN EL PROYECTO	162
ILUSTRACIÓN 59	INSTALACIÓN DE PYTORCH EXITOSA EN EL PROYECTO	162
ILUSTRACIÓN 60	INSTALACIÓN DE REQUERIMIENTOS	163
ILUSTRACIÓN 61	SELECCIONAR LOS PAQUETES A INSTALAR	163
ILUSTRACIÓN 62	INSTALANDO PAQUETES	164
ILUSTRACIÓN 63	PROYECTO EJECUTÁNDOSE	164
ILUSTRACIÓN 64	DESCARGA VISUAL STUDIO	172
ILUSTRACIÓN 65	ARCHIVOS DESCARGADOS	173
ILUSTRACIÓN 66	SELECCIONAR DESARROLLO CON C++	173
ILUSTRACIÓN 67	INSTALANDO COMPONENTES DE C++	174
ILUSTRACIÓN 68	DESCARGAR CUDA CONFIGURACIÓN	175
ILUSTRACIÓN 69	DESCARGA DE CUDA	175
ILUSTRACIÓN 70	EJECUTAR EL EJECUTABLE DE CUDA COMO ADMINISTRADOR	175
ILUSTRACIÓN 71	DIRECTORIO TEMPORAL DEL INSTALADOR DE CUDA	175

ILUSTRACIÓN 72	INSTALANDO CUDA	176
ILUSTRACIÓN 73	VERIFICANDO COMPATIBILIDAD	176
ILUSTRACIÓN 74	ACEPTAR TÉRMINOS Y CONDICIONES	177
ILUSTRACIÓN 75	INSTALACIÓN EXPRESS	177
ILUSTRACIÓN 76	SOLICITARA INSTALACIÓN DE VISUAL STUDIO	178
ILUSTRACIÓN 77	PREPARANDO PARA INSTALACIÓN	178
ILUSTRACIÓN 78	INSTALACIÓN DE CUDA	179
ILUSTRACIÓN 79	INSTALACIÓN FINALIZADA DE CUDA	179
ILUSTRACIÓN 80	DESCARGA DE CUDNN	180
ILUSTRACIÓN 81	EXTRACCIÓN DE ARCHIVOS DESCARGADOS DE CUDNN	180
ILUSTRACIÓN 82	DIRECTORIO DE INSTALACIÓN DE CUDA	181
ILUSTRACIÓN 83	ADICIÓN DE ARCHIVOS CUDNN A CUDA CARPETA BIN	181
ILUSTRACIÓN 84	ADICIÓN DE ARCHIVOS CUDNN A CUDA CARPETA INCLUDE	182
ILUSTRACIÓN 85	ADICIÓN DE ARCHIVOS CUDNN A CUDA CARPETA LIB/X64	182
ILUSTRACIÓN 86	CONFIGURACIONES VARIABLES DE ENTORNO CUDA	183
ILUSTRACIÓN 87	CONFIGURACIÓN VARIABLES DE ENTORNO CUDA	183
ILUSTRACIÓN 88	PANEL DE CONTROL NVIDIA CON LA VERSIÓN INSTALADA	184
ILUSTRACIÓN 89	DESCARGA DE ANACONDA3	185
ILUSTRACIÓN 90	EJECUTABLE DE ANACONDA3	185
ILUSTRACIÓN 91	EJECUCIÓN DEL INSTALADOR DE ANACONDA3	186
ILUSTRACIÓN 92	ESTAR DE ACUERDO CON LOS TÉRMINOS	186
ILUSTRACIÓN 93	INSTALAR SOLO PARA USO DEL EQUIPO	187
ILUSTRACIÓN 94	CAMBIO DE DIRECTORIO DE INSTALACIÓN	187
ILUSTRACIÓN 95	ADICIÓN DE VARIABLES DE ENTORNO CON EL INSTALADOR	188
ILUSTRACIÓN 96	INSTALANDO ANACONDA3	188
ILUSTRACIÓN 97	ANACONDA INSTALADO	189
ILUSTRACIÓN 98	COMPROBACIÓN DE INSTALACIÓN	189
ILUSTRACIÓN 99	DESCARGA DE PYTHON 3.9.6	190
ILUSTRACIÓN 100	ADICIÓN DE VARIABLES DE ENTORNO DESDE EL INSTALADOR DE PYTHON	191
ILUSTRACIÓN 101	INSTALANDO PYTHON	191

ILUSTRACIÓN 102	PYTHON INSTALADO CORRECTAMENTE	192
ILUSTRACIÓN 103	COMPROBAR INSTALACIÓN	192
ILUSTRACIÓN 104	DESCARGA DE PYCHARM	193
ILUSTRACIÓN 105	EJECUTAR EL INSTALADOR DE PYCHARM	194
ILUSTRACIÓN 106	DIRECTORIO DE INSTALACIÓN	194
ILUSTRACIÓN 107	OPCIONES SELECCIONADAS PARA INSTALAR PYCHARM	195
ILUSTRACIÓN 108	FOLDER DE INICIO PYCHARM	195
ILUSTRACIÓN 109	INSTALACIÓN DE PYCHARM	196
ILUSTRACIÓN 110	REINICIAR EQUIPO	196
ILUSTRACIÓN 111	DESCARGA DE MYSQL	197
ILUSTRACIÓN 112	INSTALACIÓN PERSONALIZADA MYSQL	198
ILUSTRACIÓN 113	OPCIONES A INSTALAR	198
ILUSTRACIÓN 114	INSTALACIÓN DE OPCIONES MYSQL	199
ILUSTRACIÓN 115	ELEGIR TIPO DE CONFIGURACIÓN DE DESARROLLO	199
ILUSTRACIÓN 116	MÉTODO DE AUTENTICACIÓN POR DEFECTO	200
ILUSTRACIÓN 117	CONFIGURACIÓN DE CREDENCIALES DE ACCESO	200
ILUSTRACIÓN 118	SERVICIO DE WINDOW MYSQL	201
ILUSTRACIÓN 119	PERMISOS DE ARCHIVOS DEL SERVIDOR	201
ILUSTRACIÓN 120	APLICANDO CONFIGURACIÓN MYSQL	202
ILUSTRACIÓN 121	CONFIGURACIÓN DE MYSQL	202
ILUSTRACIÓN 122	FINALIZAR INSTALACIÓN	203
ILUSTRACIÓN 123	INSTALACIÓN COMPLETADA	203
ILUSTRACIÓN 124	CONFIGURACIÓN EN MYSQL WORKBENCH	204
ILUSTRACIÓN 125	EJECUTANDO BASE DE DATOS	204
ILUSTRACIÓN 126	DESCARGA MAKESENSE	205
ILUSTRACIÓN 127	DESCARGA DE NODE JS v16.16	205
ILUSTRACIÓN 128	ACEPTAR TÉRMINOS DE NODE JS	206
ILUSTRACIÓN 129	INSTALACIÓN NODE JS PERSONALIZADA	206
ILUSTRACIÓN 130	INSTALAR HERRAMIENTA PARA MÓDULOS NATIVOS	207
ILUSTRACIÓN 131	CONFIRMAR INSTALACIÓN	207
ILUSTRACIÓN 132	CERRAR SIN CONTINUAR	208

ILUSTRACIÓN 133	MAKESENSE EJECUTÁNDOSE	209
ILUSTRACIÓN 134	UTILIZANDO MAKESENSE	209
ILUSTRACIÓN 135	EJECUTANDO LABELIMG DESDE CONSOLA CONDA	210
ILUSTRACIÓN 136	LABELIMG EJECUTÁNDOSE	211
ILUSTRACIÓN 137	CARPETA ESTRUCTURA PROYECTO	213
ILUSTRACIÓN 138	UBICACIÓN ARCHIVO EJECUTABLE	213
ILUSTRACIÓN 139	UBICACIÓN ARCHIVO CONFIGURACIÓN PROYECTO EJECUTABLE	214

ÍNDICE DE TABLAS

TABLA 1 REQUISITOS DE HARDWARE Y SOFTWARE

145

Introducción

El presente documento contempla los pasos a seguir para la instalación y configuración del entorno de desarrollo o mantenimiento del prototipo de sistema de videovigilancia automatizado por medio de visión artificial para ITCA-FEPADE.

Se detalla cada uno de los softwares utilizados para el desarrollo del sistema de visión utilizando Python en la versión 3.9.6 y sus respectivas librerías, se detalla también los softwares utilizados para el entrenamiento de modelos de predicción utilizando YOLO v8 ejecutado con la GPU para un mejor rendimiento.

Describe cada una de las librerías y sus diferentes funciones en el funcionamiento del proyecto, se detalla la estructura del proyecto, de igual forma los archivos y funciones de cada uno para una mejor comprensión al momento de revisar y comprende el código.

Objetivo

Describir cada uno de los procesos a seguir para la configuración y desarrollo del prototipo de sistema de videovigilancia automatizado por medio de visión artificial para ITCA-FEPADE.

Alcance del documento

Este documento está dirigido a desarrolladores del prototipo de sistema de videovigilancia automatizado por medio de visión artificial para ITCA-FEPADE con el fin de ofrecer una guía y consultas sobre la configuración y desarrollo del sistema.

Definiciones

- **Conda:** Conda es un sistema de gestión de entornos y paquetes de código abierto que se ejecuta en Windows, macOS y Linux. Conda instala, ejecuta y actualiza rápidamente los paquetes y sus dependencias.
- **CUDA:** son las siglas de Compute Unified Device Architecture (Arquitectura Unificada de Dispositivos de Cómputo) que hace referencia a una plataforma de computación en paralelo que incluye un compilador y un conjunto de herramientas de desarrollo creadas por Nvidia que permiten a los programadores usar una variación del lenguaje de programación C (CUDA C) para codificar algoritmos en GPU de Nvidia.
- **CPU:** Unidad central de procesamiento, es el procesador encargado de interpretar las instrucciones de un programa informático.
- **Directorio:** sinónimo de archivos, folder, carpeta y su ubicación dentro de nuestra computadora.
- **GPU:** Unidad de procesamiento gráfico funciona como un coprocesador ayudando a acelerar algunas funciones de procesamiento gráfico liberando de trabajo la CPU.
- **NVIDIA® CUDA® Toolkit:** proporciona un entorno de desarrollo para crear aplicaciones aceleradas por GPU de alto rendimiento. Con CUDA Toolkit, puede desarrollar, optimizar e implementar sus aplicaciones en sistemas integrados acelerados por GPU.
- **PyTorch:** es una biblioteca de tensores optimizada para el aprendizaje profundo mediante GPU y CPU, de código abierto basada en la biblioteca de Torch, utilizado para

aplicaciones como visión artificial y procesamiento de lenguajes naturales.

Requerimientos técnicos del sistema

Requisitos de hardware

Para el desarrollo del prototipo se trabajó con las siguientes características de hardware y software o superior.

Equipo	Especificación	Capacidad
Laptop	Procesador	Intel® Core™i7-6700 HQ CPU @ 2.60 GHz 2.59 GHz o superior
	RAM	16.0 GB o superior
	Tipo de sistema	Sistema operativo Windows 10 de 64 bits, Procesador x64 o superior
	Tarjeta grafica	NVIDIA GeForce GTX 965M o superior
	Memoria de video dedicada	4096 MB GDDR 5 o superior
	Unidad de almacenamiento	Disco de Estado Sólido SSD 128 GB o superior

Tabla 26 Requisitos de hardware y software

Requisitos de software

Para el desarrollo del prototipo se han utilizado las siguientes herramientas y lenguaje de programación.

Lenguaje de programación.

Python en la versión 3.6.9

Herramienta de desarrollo.

Editor de código utilizado PyCharm Community Edition 2022.3.2

Base de datos.

Base de datos utilizada MySQL Ver 15.1 o superior, para Win64 (AMD64).

Otras Tecnologías.

CUDA 11.8 se ha utilizado para la ejecución de procesos mediante la GPU de la computadora.

Pytorch se ha utilizado para ejecutar procesos de entrenamiento utilizando la GPU la computadora.

Anaconda3 se ha utilizado para ejecutar comando y entrenar modelos en un entorno virtual.

Programación

Este apartado detalla el desarrollo del proyecto, diagrama de base de datos, estructura del proyecto Python, entrenamiento del modelo.

Diagrama de base de datos

Diagrama relacional de las tablas que contiene el sistema.

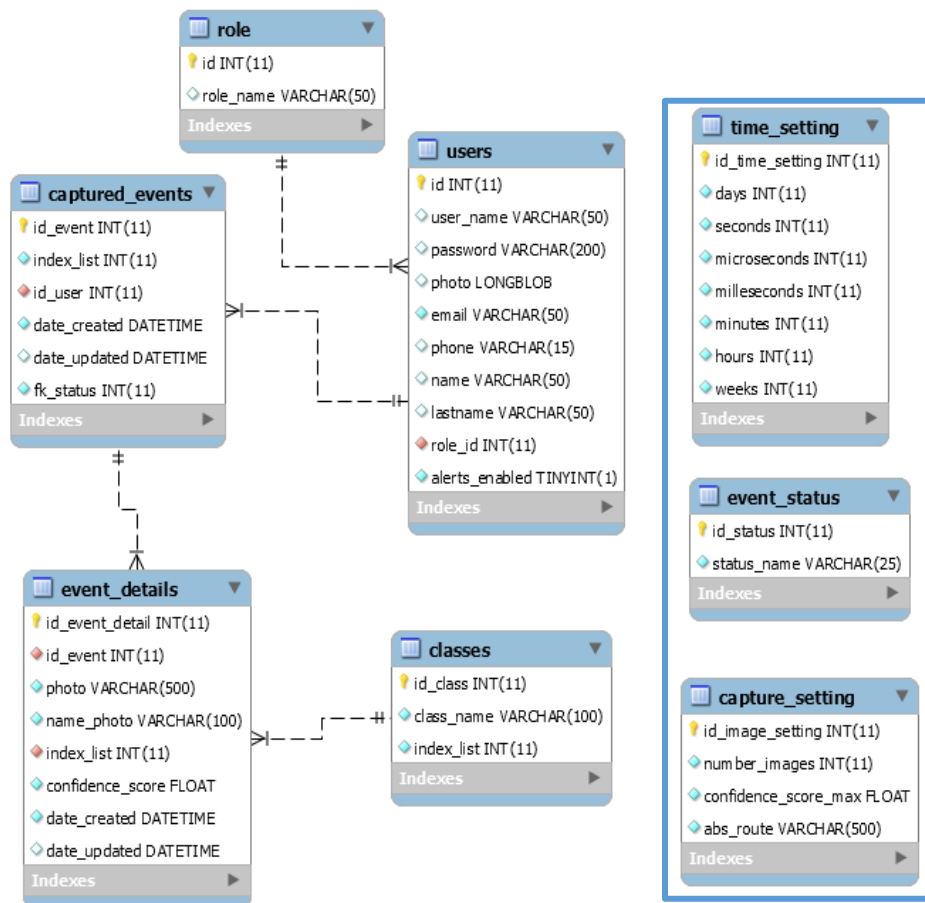


Ilustración 42 Diagrama de relacional de base de datos

Proceso entrenamiento de modelo usando YOLO v8

Para el entrenamiento se seguirá la siguiente estructura de carpetas y archivos.

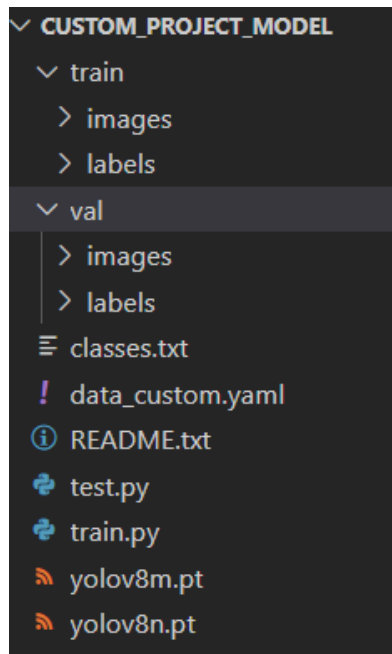


Ilustración 43 Estructura de proyecto de entrenamiento

Para la descarga de modelos base consultar la página oficial de ultralytics

<https://docs.ultralytics.com/tasks/detect/#models>

Model	size (pixels)	mAP ^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Ilustración 44 modelos ultralytics

Tomado de fuente

<https://docs.ultralytics.com/tasks/detect/#models>

Para el entrenamiento se ha usado YOLOv8m que es la versión media con los parámetros listados en la tabla, donde

- **Size:** Tamaño de las imágenes
- **mAPval:** Precisión promedio media 50-95
- **Speed CPU ONNX (ms):** Velocidad de la cpu con ONNX(ms), ajuste del rendimiento del tiempo de ejecución de ONNX
- **Speed A100 TensorRT (ms):** Velocidad de la inferencia con tensores utilizando CUDA
- **Etiquetado de imágenes:** Para el etiquetado de imágenes se ha utilizado uno de los softwares sugeridos, makesense o LabelImg, una vez realizado el etiquetado se exportan y se colocan en los directorios correspondiente.
- **Train:** Es el directorio que contiene las imágenes y las etiquetas para usarlas en el entrenamiento del modelo
- **Val:** Es el directorio que contiene las imágenes y las etiquetas para usarlas en la validación del modelo, es una muestra de imágenes que ayudara a validar el entrenamiento.
- **Classes:** Es un archivo de texto que contiene un listado de las clases las cuales se usan para el etiquetado.

```

Rindiendose
Normal
Apuntando
Romper_ventana
Acostado
Agachado
Ocultandose
Mirando_sospechoso
Extrayendo_bolso
Robando
Registro
Abriendo_puerta_normal
Recoger_objeto
Arma
Golpear_ventana
Extrayendo_bolso_lateral
Mirando_sospechoso_lateral

```

Ilustración 45 Listado de clases

- **Data_custom.yaml:** Contiene la configuración para poder entrenar un modelo, donde se indican la ubicación de las imágenes de entrenamiento y validación, número de clases `nc` y las clases en el orden del archivo `classes`

```

1 train: D:\PYTHON_PROJECT_ITCA\Prototipo_vision_artificial_using_conda\custom_project_model\train
2 val: D:\PYTHON_PROJECT_ITCA\Prototipo_vision_artificial_using_conda\custom_project_model\val
3
4 nc: 23
5
6 names: ["Rindiendose", "Normal", "Apuntando", "Romper_ventana", "Acostado", "Agachado", "Ocultandose",
7         "Mirando_sospechoso", "Extrayendo_bolso", "Robando", "Registro", "Abriendo_puerta_normal", "Recoger_objeto",
8         "Arma", "Golpear_ventana", "Extrayendo_bolso_lateral", "Mirando_sospechoso_lateral",
9         "llanta_hombro", "llanta_manos", "llanta_suelo", "rindiendose_lateral", "baul_abierto", "capo_abierto"]

```

Ilustración 46 Archivo `data_custom.yaml`

- **Train.py:** Es un archivo con configuraciones para entrenar un nuevo modelo con base en uno ya existente.

```

from ultralytics import YOLO

model = YOLO("yolov8m.pt")

model.train(data = "data_custom.yaml", batch=8, imgsz=640, epochs=100, workers=1)

```

Ilustración 47 Archivo `train.py`

Se está usando el modelo `yolov8m.pt` para entrenar un nuevo modelo, dentro de los parámetros se han configurado: (Estos valores pueden variar dependiendo de los resultados mostrados en

el primer entrenamiento y obtener nuevos modelos con nuevos valores)

- **Data:** indica el archivo de configuración con la información necesario para entrenar el modelo.
- **batch:** Indica el número de imágenes por lote.
- **imgsz:** Indica el tamaño de las imágenes de entrada como entero o (w,h) ancho, alto.
- **Epochs:** Número de épocas para entrenar.
- **Workers:** Número de subprocesos de trabajo para la carga de datos (por rango si DDP).
- **Test.py:** Es un archivo con configuraciones para realizar pruebas utilizando una cámara o video depende de la configuración mediante el uso de un modelo pre-entrenado para poder predecir las cuales son guardados en el proyecto.

```
from ultralytics import YOLO

model = YOLO("yolov8m_theft_detection.pt")

model.predict(source="robos.mp4", show=True, save=True, conf=0.5)
```

Ilustración 48 Archivo test.py

Entrenamiento yolov8 comandos

Se listan cada uno de los pasos para crear un entorno virtual y realizar entrenamiento de modelos.

1. Creación del entorno virtual.

```
conda create -n yolov8_custom_project_example_name python=3.9
```

2. Activar el entorno virtual para poder trabajar.

```
conda activate yolov8_custom_project_example_name
```

3. Instalar el paquete de ultralytics

```
pip install ultralytics
```

4. Instalar pytorch, PyTorch es una biblioteca de tensores optimizada para el aprendizaje profundo mediante GPU y CPU.

Es necesario realizar la instalación de acuerdo con los recursos y versiones instaladas.

url <https://pytorch.org/get-started/locally/#windows-pip>

PyTorch Build	Stable (2.0.1)		Preview (Nightly)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python		C++ / Java	
Compute Platform	CUDA 11.7	CUDA 11.8	ROCm 5.4.2	CPU
Run this Command:	<pre>pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118</pre>			

Ilustración 49 Configuración descarga Pytorch

Tomado Fuente: <https://pytorch.org/get-started/locally/#windows-pip>

Ejecutar la siguiente línea de comando para realizar la instalación.

```
pip install torch==2.0.0+cu118 torchvision==0.15.1+cu118 torchaudio --force-reinstall --extra-index-url https://download.pytorch.org/whl/cu118
```

5. Comprobar la instalación de pytorch, deberá aparecer como el siguiente:

La versión instalada y si CUDA está disponible en verdadero.

```

(yolov8_custom)
D:\PYTHON_PROJECT_ITCA\Prototipo_vision_artificial_using_conda\
custom_project_model>python

Python 3.9.16 (main, Mar  8 2023, 10:39:24) [MSC v.1916 64 bit
(AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more
information.

>>> import torch

>>> torch.__version__

'2.0.0+cu118'

>>> torch.cuda.is_available()

True

>>>exit()

```

6. Entrenamiento del modelo

En este paso podremos ejecutar el archivo de Python llamado `train.py` con las configuraciones establecidas o realizarlo desde consola con la siguiente línea, iniciará el entrenamiento del nuevo modelo el cual será usado para predicciones dentro del prototipo.

```

yolo task=detect mode=train epochs=100 data=data_custom.yaml
model=yolov8m.pt imgsz=640 batch=8

```

Los resultados se guardan en el directorio donde se está trabajando, en la carpeta llamada `weights` se aloja el nuevo modelo llamado `pesos` para poder hacer predicciones.

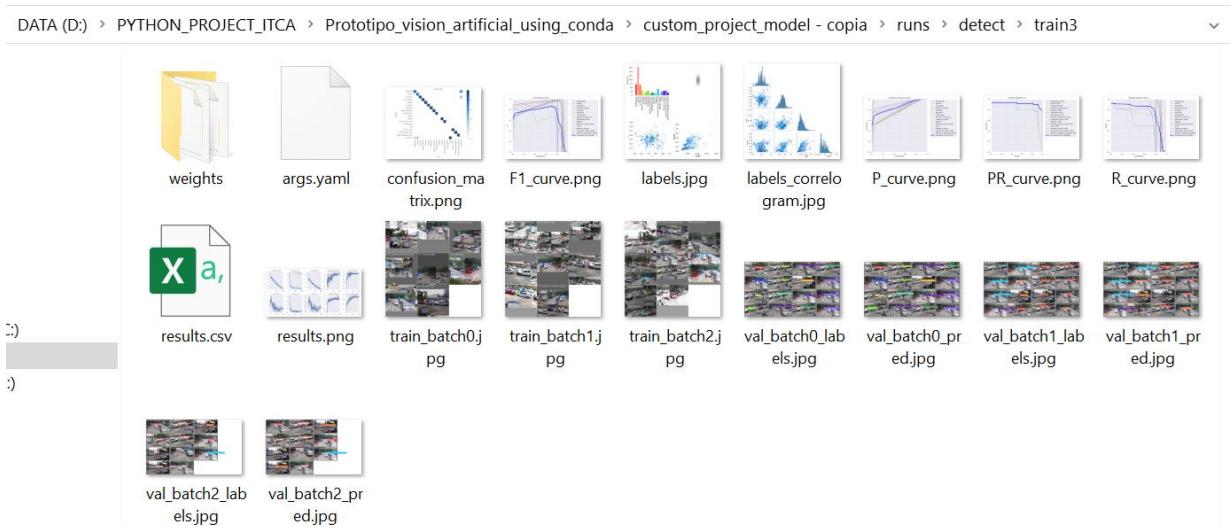


Ilustración 50 Resultado de entrenamiento

7. Para probar el modelo generado se debe ejecutar el archivo `test.py` o ejecutarla desde la línea de comando siguiente.

```
yolo task=detect mode=predict model=yolov8m_custom.pt show=True
conf=0.5 source=1.mp4/ip de camara/url/0 para web cam
```

Programación del prototipo de visión artificial

Esta sección describe la estructura del proyecto de captura y procesamiento de video, detalle de funciones.

La estructura del proyecto es la siguiente:

Nombre	Fecha de modifica...	Tipo	Tamaño
base_de_datos	06/06/2023 09:06 ...	Carpeta de arc...	
event_images	07/06/2023 12:18 ...	Carpeta de arc...	
images	06/06/2023 09:06 ...	Carpeta de arc...	
images_db	06/06/2023 09:06 ...	Carpeta de arc...	
login_images	07/06/2023 12:17 ...	Carpeta de arc...	
pesos_red_yolo	06/06/2023 09:06 ...	Carpeta de arc...	
classes.txt	06/06/2023 09:06 ...	Documento de...	1 KB
connection.py	06/06/2023 09:06 ...	Python File	1 KB
function_db.py	06/06/2023 09:06 ...	Python File	11 KB
login_functions.py	06/06/2023 09:06 ...	Python File	2 KB
main.py	06/06/2023 09:06 ...	Python File	32 KB
parametros.py	06/06/2023 09:06 ...	Python File	2 KB
README.txt	06/06/2023 09:06 ...	Documento de...	6 KB
README_PROJECT.txt	06/06/2023 09:06 ...	Documento de...	1 KB
requirements.txt	06/06/2023 09:06 ...	Documento de...	1 KB
send_mail.py	06/06/2023 09:06 ...	Python File	4 KB
settings_db.json	06/06/2023 09:06 ...	JSON File	1 KB

Ilustración 51 Estructura del sistema prototipo

- **Base de datos:** Este directorio contiene el script de base de datos.
- **Event_images:** Contiene las imágenes capturadas en tiempo real ordenadas por fecha, este directorio se debe registrar en base de datos.
- **Images:** Este directorio contiene imágenes, iconos del proyecto.
- **Images_db:** Directorio para obtener todas las imágenes en base de datos (opcional).

- **Login_images:** Este directorio es utilizado para escribir imágenes temporales al iniciar el sistema con detección de rostros.
- **Pesos_red_yolo:** Este directorio contiene el modelo llamado pesos utilizado para predecir eventos en tiempo real.
- **Classes:** Este archivo contiene el listado de clases utilizado para el entrenamiento.
- **Connection:** Contiene la conexión a base de datos.
- **Function_db:** Contiene funciones que permiten registrar, leer datos de base de datos.
- **Logins_functions:** Contiene las funciones para logueo
- **Main:** Contiene la funcionalidad del sistema
- **Parámetros:** Se definen los parámetros de estilo e inicialización de parámetros.
- **Send_mail:** Contiene la funcionalidad para el envío de correos.
- **Send_sms:** contiene las funcionalidades para el envío de mensajes.
- **Settings_db:** Contiene las configuraciones de conexión, servicios, y rutas de archivos locales.
- **Requirements:** Contiene las librerías a instalar en el proyecto.

Paquetes utilizados

- **Base64:** Se ha utilizado para codificar imágenes en formato base64
- **Random:** Se ha utilizado para generar valores numéricos entre 0 y 255
- **Traceback:** Se ha utilizado para imprimir mensajes de seguimiento
- **Tkinter:** Se ha Utilizado para el desarrollo de la interfaz gráfica del prototipo.
- **Os:** Se ha Utilizado para gestionar archivos del sistema escritura, lectura.
- **PIL:** Sirve para la utilización de imágenes con tkinter
- **Opencv-python cv2 as cv:** Se ha utilizado para el procesamiento de video e imágenes
- **Matplotlib:** Se ha utilizado para la leer los pixeles de una imagen
- **Mtcnn:** Se ha utilizado para detectar rostros con la función `detect_faces(pixels)` en una imagen.
- **Json:** Se ha utilizado para capturar la información de archivo de configuración json
- **Ultralytics:** Se ha utilizado para procesar el modelo preentrenado y obtener predicciones.
- **Datetime:** Se ha Utilizado para configuración del tiempo
- **Imutils:** Se ha utilizado para redimensionar imágenes o llamadas "frame".
- **Numpy:** Se ha utilizado para convertir valores a arreglos de numpy del resultado del modelo.

- **Re:** Se ha utilizado para verificar existencia de patrones específicos en una cadena de texto o valor.
- **APScheduler:** Se ha utilizado para programar código que puede ejecutarse en diferentes intervalos de tiempo.

Configuración en PyCharm

Desde el directorio donde se ubicaron los archivos del proyecto se debe iniciar en PyCharm con las siguientes acciones.

Pasos

1. Ir a archivos y presionar abrir

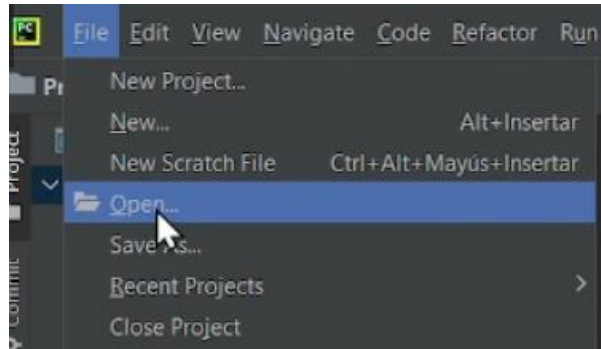


Ilustración 52 Configuración abrir proyecto

2. Seleccionar el directorio donde se encuentra los archivos del proyecto y presionar ok.

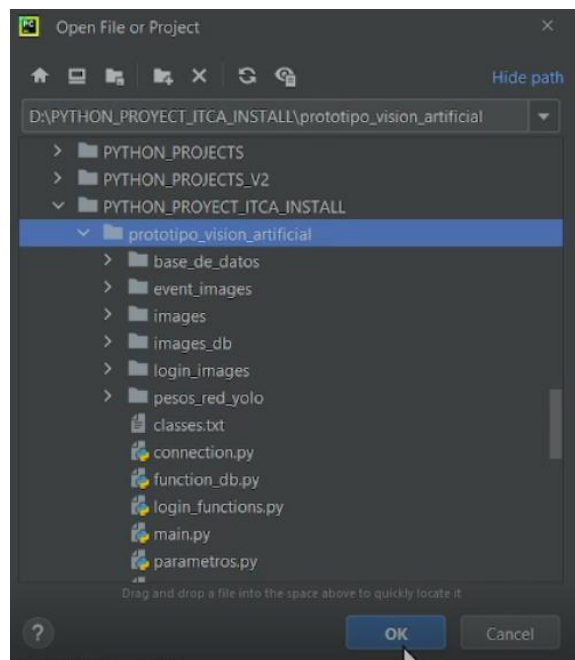


Ilustración 53 Seleccionar proyecto

3. Aparecerá la siguiente ventana presionar confiar en el proyecto (Trust Project)

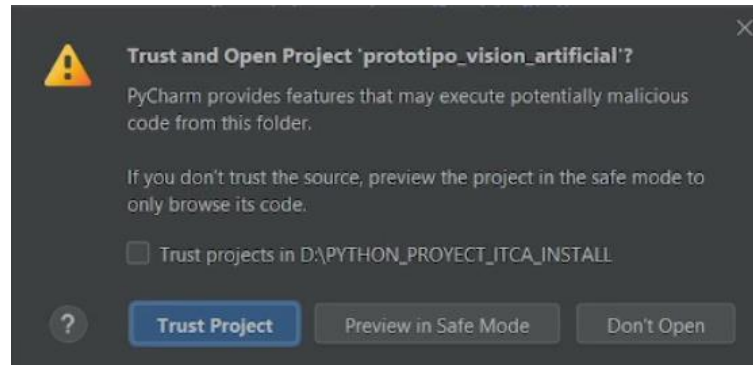


Ilustración 54 Confiar en los archivos

4. Solicitará seleccionar donde se abrirá el proyecto si en la ventana actual o en una nueva, presionar en la ventana actual (this Window).

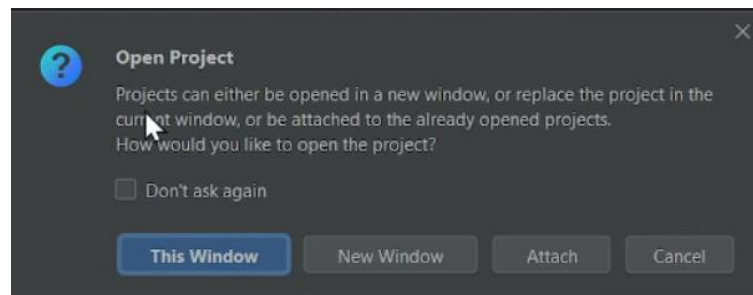


Ilustración 55 Abrir proyecto nueva ventana

5. Se mostrará un modal para crear un entorno virtual, acá lo importante es seleccionar el intérprete base, para el cual se usara la versión de Python 3.9.6 y se buscara donde se instaló la versión, seleccionar el ejecutable Python.exe, quedando como esta en la siguiente captura, la ruta seguramente cambiara.

La ubicación del entorno virtual se dejará en el directorio donde se está trabajando, y las dependencias se usarán los

requerimientos que se incluyen en el archivo requirements.txt.

Presionaremos "ok" y el editor de código empezara a configurar el entorno.

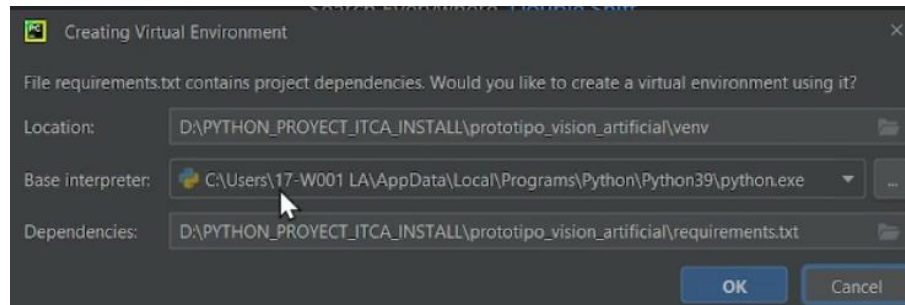


Ilustración 56 Configuración compilador base

6. En este paso se mostrara un error el cual se muestra en la siguiente pantalla, este mensaje ocurre porque intento instalar las dependencias y no encontró algunas dependencias de pytorch.

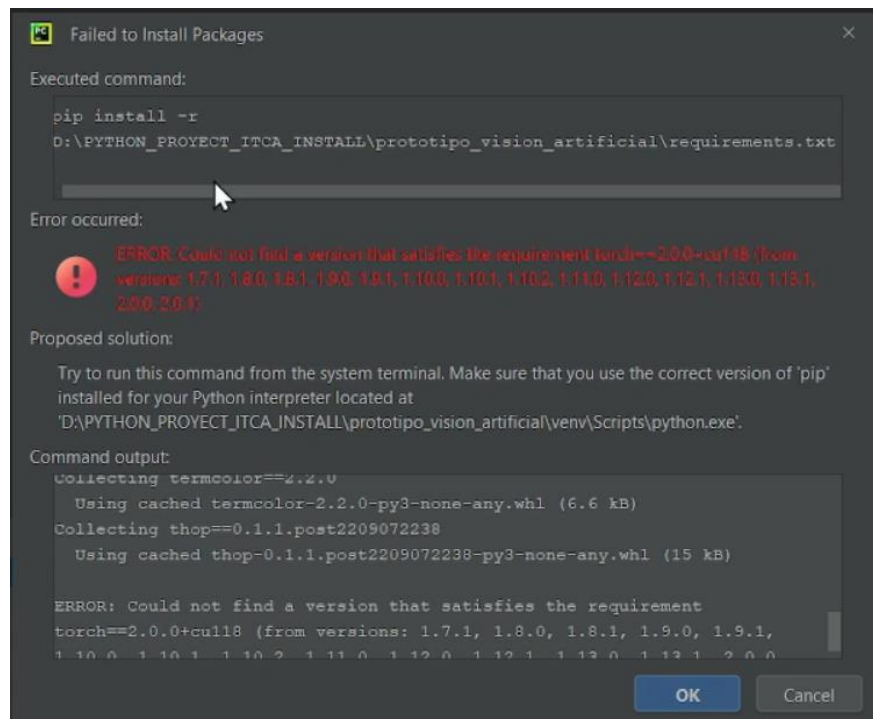


Ilustración 57 Alerta de error, corrección.

Para corregir esto se instalará la siguiente librería desde la terminal de PyCharm

```
pip install torch==2.0.0+cu118 torchvision==0.15.1+cu118 torchaudio --force-reinstall --extra-index-url https://download.pytorch.org/whl/cu118
```



```
Terminal: Local + v
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

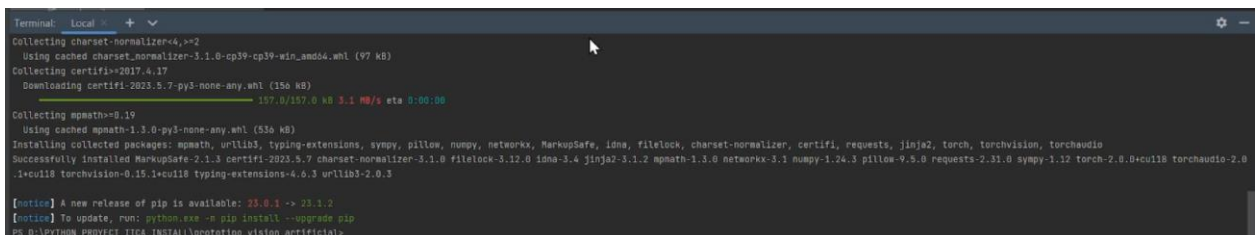
Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS D:\PYTHON_PROYECT_ITCA_INSTALL\prototipo_vision_artificial> pip install torch==2.0.0+cu118 torchvision==0.15.1+cu118 torchaudio --force-reinstall --extra-index-url https://download.pytorch.org/whl/cu118
Looking in indexes: https://pypi.org/simple, https://download.pytorch.org/whl/cu118
Collecting torch==2.0.0+cu118
  Using cached https://download.pytorch.org/whl/cu118/torch-2.0.0%2Bcu118-cp39-cp39-win_amd64.whl (2611.4 MB)
Collecting torchvision==0.15.1+cu118
  Using cached https://download.pytorch.org/whl/cu118/torchvision-0.15.1%2Bcu118-cp39-cp39-win_amd64.whl (4.9 MB)
Collecting torchaudio
```

Ilustración 58 instalación Pytorch en el proyecto

Esta instalación utilizará los paquetes que estén en cache correspondiente a la instalación realizada para el entrenamiento del modelo.

Cuando se complete la instalación aparecerá la confirmación en la terminal que fue exitosa.

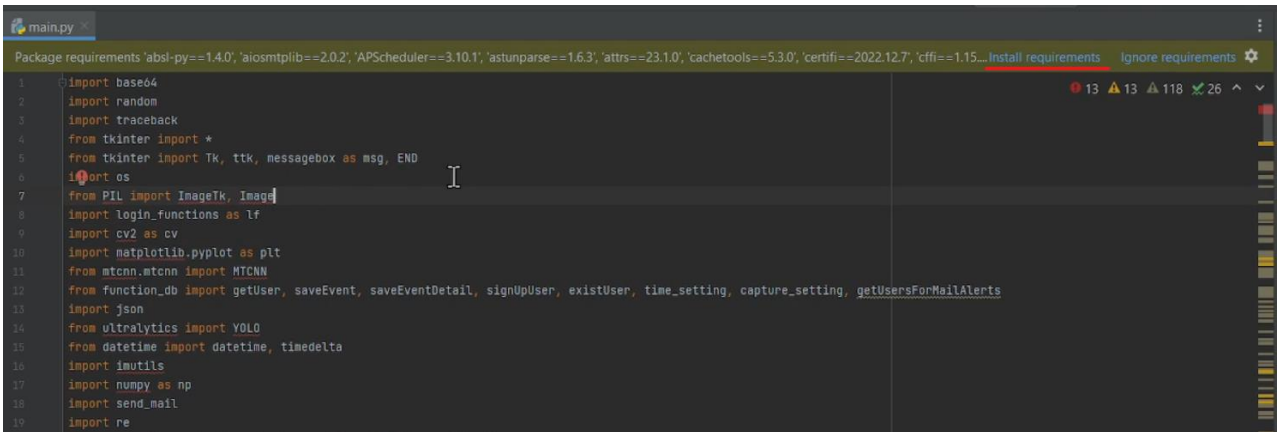


```
Terminal: Local + v
Collecting charset-normalizer<4, >=2
  Using cached charset_normalizer-3.1.0-cp39-cp39-win_amd64.whl (97 kB)
Collecting certifi==2017.4.17
  Downloading certifi-2023.5.7-py3-none-any.whl (156 kB)
  197.0/197.0 KB 0.1 MB/s eta 0:00:00
Collecting mpmath==0.19
  Using cached mpmath-1.3.0-py3-none-any.whl (536 kB)
Installing collected packages: mpmath, urllib3, typing-extensions, sympy, pillow, numpy, networkx, MarkupSafe, idna, filelock, charset-normalizer, certifi, requests, Jinja2, torch, torchvision, torchaudio
Successfully installed MarkupSafe-2.1.3 certifi-2023.5.7 charset-normalizer-3.1.0 filelock-3.12.0 idna-3.4 Jinja2-3.1.2 mpmath-1.3.0 networkx-3.1 mumpy-1.24.3 pillow-9.5.0 requests-2.31.0 sympy-1.12 torch-2.0.0+cu118 torchvision-0.15.1+cu118 typing-extensions-4.5.3 urllib3-2.0.3

[notice] A new release of pip is available: 23.0.1 -> 23.1.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS D:\PYTHON_PROYECT_ITCA_INSTALL\prototipo_vision_artificial>
```

Ilustración 59 Instalación de Pytorch exitosa en el proyecto

7. Instalar los requerimientos para que funcione el proyecto, abriremos el archivo main.py el cual aparecerá como en la siguiente imagen, acá presionar instalar requerimientos (Install requirements)



```
1 import base64
2 import random
3 import traceback
4 from tkinter import *
5 from tkinter import TK, ttk, messagebox as msg, END
6 import os
7 from PIL import ImageTk, Image
8 import login_functions as lf
9 import cv2 as cv
10 import matplotlib.pyplot as plt
11 from mtcnn.mtcnn import MTCNN
12 from function_db import getUser, saveEvent, saveEventDetail, signUpUser, existUser, time_setting, capture_setting, getUsersForMailAlerts
13 import json
14 from ultralytics import YOLO
15 from datetime import datetime, timedelta
16 import inutils
17 import numpy as np
18 import send_mail
19 import re
```

Ilustración 60 Instalación de requerimientos

Aparecerán los paquetes a instalar, en este paso presionaremos install

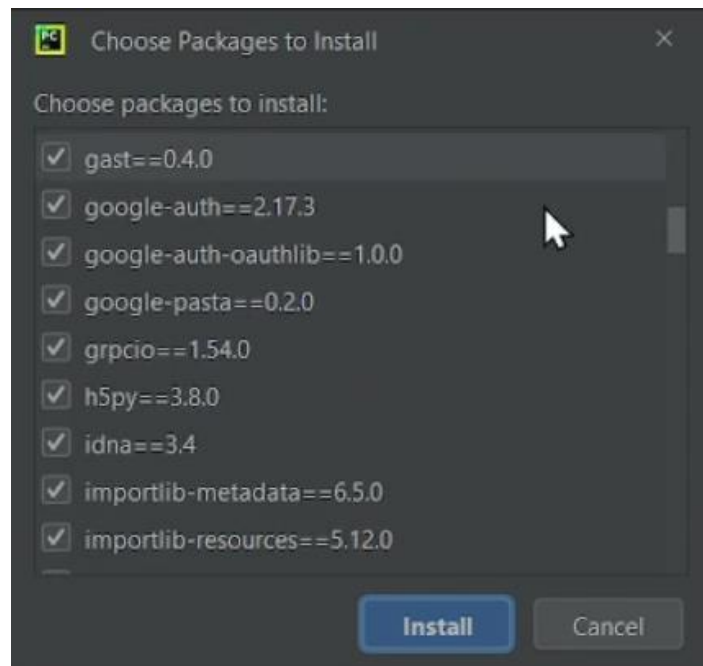


Ilustración 61 Seleccionar los paquetes a instalar

8. En la parte inferior del editor aparecerá una barra donde se verán las instalaciones que está realizando, presionamos ver todo y aparecerá el progreso del paquete que está instalando.

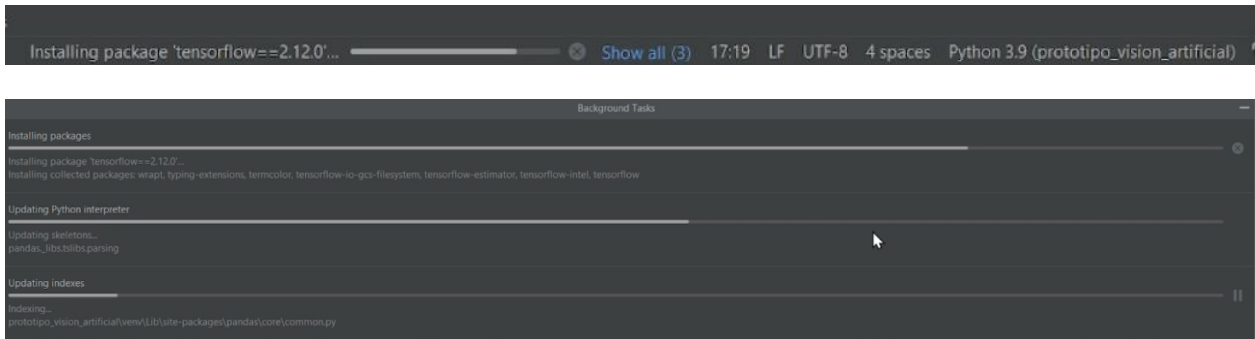


Ilustración 62 Instalando paquetes

9. Completada la instalación se procede a ejecutar el archivo main.py el cual muestra la siguiente ventana del proyecto.

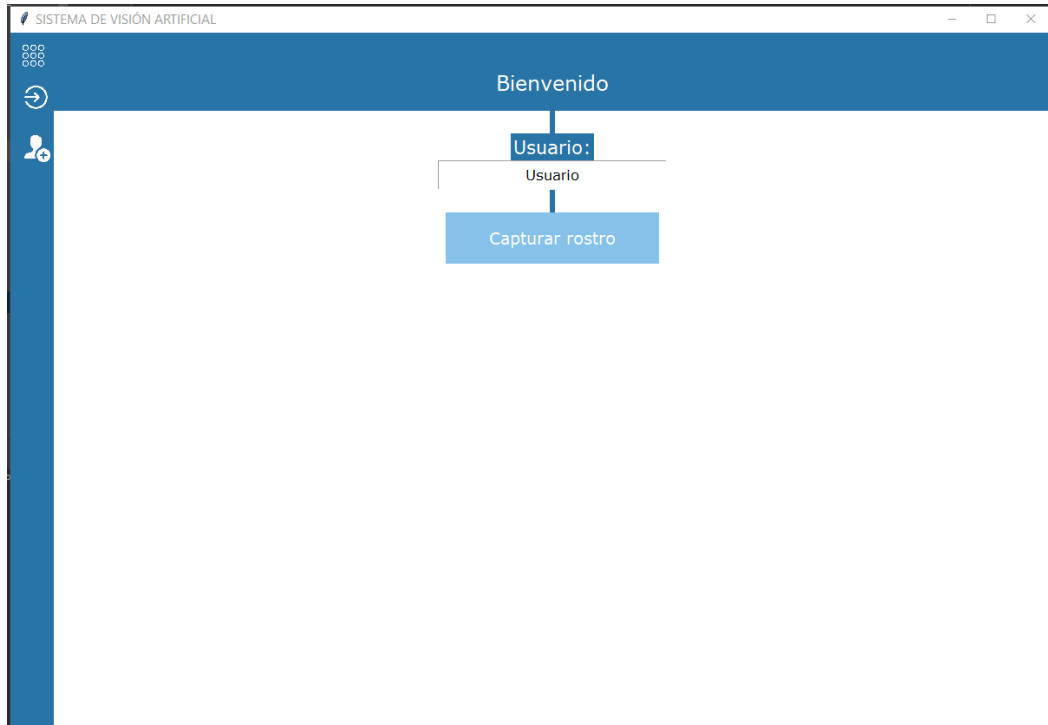


Ilustración 63 Proyecto ejecutándose

Descripción de funciones por archivo

main.py

Este archivo contiene las importaciones de paquetes instalados, contiene la inicialización de parámetros, frame de utilizado para mostrar los componentes gráficos.

Funciones

- **main_dashboard()**

Esta función se encarga de mostrar los diferentes paginas usando TNotebook de tkinter, en esta función se define cual página mostrar dependiendo si está registrado o no.

- **screen_home()**

Función encargada de seleccionar la página a mostrar en el notebook de tkinter, de igual forma se selecciona el frame de inicio configurado.

- **screen_login()**

Función encargada de seleccionar la página a mostrar en el notebook de tkinter, de igual forma se selecciona el frame de login configurado.

- **screen_signup()**

Función encargada de seleccionar la página a mostrar en el notebook de tkinter, de igual forma se selecciona el frame de registro configurado.

- **screen_events()**

Función encargada de seleccionar la página a mostrar en el notebook de tkinter, de igual forma se selecciona el frame de eventos configurado.

- **design_sign_up()**
Función encargada del diseño para el registro de usuarios.
- **design_video_capture()**
Función encargada del diseño para capturar y mostrar la cámara.
- **design_login()**
Función encargada del diseño del login, carga de componentes.
- **Menú_lateral(menu_val)**
Función encargada de ampliar los ítems en la pantalla principal, recibe el estado actual verdadero o falso, depende si esta desplegado o no.
- **on_enter()**
Función que elimina el texto de un componente entry del usuario cuando está enfocado o el cursor esta sobre este.
- **on_leave(e)**
Función que ingresa un texto en un componente entry de usuario si este está vacío y está fuera de foco.
- **on_leave_email(e)**
Función utilizada para validar el correo cada vez que está fuera de foco o el cursor se mueve a otro componente.
- **on_leave_name(e)**
Función utilizada para validar el nombre cada vez que está fuera de foco o el cursor se mueve a otro componente.
- **on_leave_lastname(e)**
Función utilizada para validar el apellido cada vez que está fuera de foco o el cursor se mueve a otro componente.
- **getEnter(screen)**
Función utilizada para agregar un componente label en la frame indicado llamado screen.

- **app()**
Función que inicia la pantalla principal llamando a la función `main_dashboard()`.
- **sign_up_capture(user1, emailst, namest, lastnamest)**
Función encargada de registrar un usuario, recibe como parámetros el usuario (fotografía), el correo, el nombre y el apellido, retorna el mensaje confirmando éxito o error.
- **logout()**
Cambia los valores y nos lleva a la pantalla principal cuando se encuentre un usuario dentro del sistema.
- **login_validate(user, entry)**
Función que permite a un usuario acceder al sistema, recibe como parámetro la fotografía del usuario y el nombre de usuario en `entry`.
- **get_phones_for_alerts()**
Función que carga los números de teléfono de los usuarios configurados para él envío de alertas.
- **load_db_settings()**
Función que se encarga de cargar las configuraciones de la base de datos en las tablas `capture_setting` y `time_setting`.
- **start_video_capture()**
Función que se encarga de iniciar la captura de video, utiliza el modelo para predecir se auxilia de funciones para almacenar los registro en la base de datos, emitir alertas vía SMS y activar el dispositivo Arduino.
- **end_video()**
Función que se encarga de finalizar la captura de video y destruir las ventanas correspondientes.

- **start_video()**

Función que carga las configuraciones e inicia la captura de video pulsando desde el botón iniciar en pantalla.

connection.py

Este archivo contiene la conexión a base de datos

- **get_connection()**

Función encargada de retornar la conexión a la base de datos

function_db.py

Este archivo contiene las funciones para gestión de datos con la base de datos.

- **capture_setting()**

Función que retorna las configuraciones para captura de video desde la base de datos en la tabla capture_setting.

- **time_setting()**

Función que se encarga de retornar las configuraciones de la tabla time_setting para agregar un intervalo de tiempo para no almacenar registros en base de datos los valores por defecto se dejaron en 0 de esta forma se mantendrá todo el tiempo activo.

- **saveEvent(clase, user_login_id)**

Almacena un registro en base de datos, recibe como parámetros la clase, el id del usuario activo en ese momento, retorna el id del evento para poder usarlo en el detalle del evento.

- **saveEventDetail(id_event, frame, path_image, relative_path , image_quality, name_photo, cls, cs)**
 Registra el detalle del evento en base de datos, recibe como parámetros, el id del evento, la imagen capturada como frame, la ruta para guardarla como path_image, la ruta relativa para ser almacenada en base de datos, la calidad de la imagen, el nombre de la foto, la clase y la puntuación de confianza como cs.
- **convertToBinaryData(photo)**
 Permite convertir una imagen a formato binario para ser almacenado como blob en la base de datos tabla users.
- **signUpUser(user_name, image, email, name, lastname)**
 Permite el registro de un nuevo usuario en la base de datos, recibe como parámetro, nombre de usuario, fotografía, correo, nombre y apellido, para las credenciales se usa por defecto "@passw0rd" retorna el id y el número de filas afectadas.
- **existUser(name)**
 Verifica si existe el usuario en la tabla users con el nombre del usuario, si este ya existe retorna 0 caso contrario retorna el id y el número de filas afectadas.
- **getUser(name, path)**
 Función que se encarga de obtener el usuario y su fotografía utiliza como parámetros el nombre y la url donde guardara temporalmente la fotografía para ser comparada e iniciar sesión.
- **getUsersForPhoneAlerts ()**
 Función que se encarga de obtener los números de teléfono de la tabla usuario a los cuales les enviara la notificación

de alerta, retorna id con valor cero y el un arreglo de números telefónicos.

- **getUsersForMailAlerts()**

Función que se encarga de obtener los correos de la tabla usuario a los cuales les enviara el correo de alerta, retorna id con valor cero y el un arreglo de correos.

login_functions.py

Este archivo contiene las funciones para comparar imagen y verificar la compatibilidad para iniciar en el sistema.

- **fase(img, faces)**

Función que recibe como parámetro una fotografía capturada y rostros detectados por el modelo fase_detection de mtcnn y guarda únicamente si encuentra rostros en el directorio temporal de login_imagenes.

- **compatibility(img1, img2)**

Función que se encarga de ver la compatibilidad entre dos imágenes y determinar el parecido a través de fuerza bruta una función de BFMatcher el cual extrae las características de una imagen y los compara con la segunda. Retorna la similitud entre ellas, un valor entre 0 y 1

parametros.py

Archivo que contiene la inicialización de los parámetros

send_sms.py

- **send(body, to)**

Función que se encarga de enviar las alertas a través del proveedor de SMS Twilio, recibe como parámetros, cuerpo del mensaje y el número de teléfono de a quien se enviara.

send_mail.py

- **send_mail_async(body, subject, to)**

Función que se encarga de enviar las alertas, verifica la conexión establecida con el servidor smtp, recibe como parámetros, cuerpo del mensaje, el asunto y a quien se enviara.

- **send(body, subject, to)**

Función que se encarga de preparar de forma asíncrona el envío de correos, recibe como parámetros el cuerpo del mensaje, el asunto y a quien se enviara.

Configuraciones

Las siguientes configuraciones e instalaciones del software son requeridas para el funcionamiento optimo del prototipo.

Instalación de visual studio

Visual estudio y las librerías de C o C++ son utilizadas por CUDA para su funcionamiento correcto.

Pasos

1. Descargar de la página oficial el instalador de visual studio

<https://learn.microsoft.com/en-us/visualstudio/releases/2019/release-notes>

2. Descargar el instalador de la versión community 2019 o profesional ambas son funcionales.

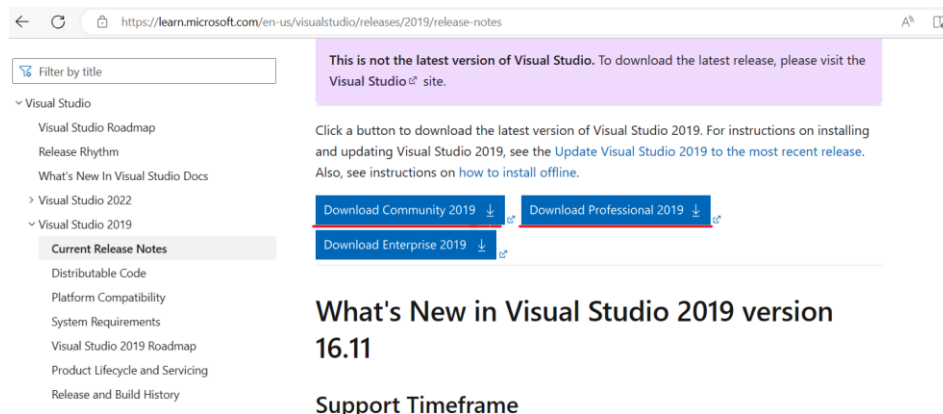


Ilustración 64 Descarga visual studio

3. Ejecutar el instalador descargado, los primeros pasos dar en continuar.

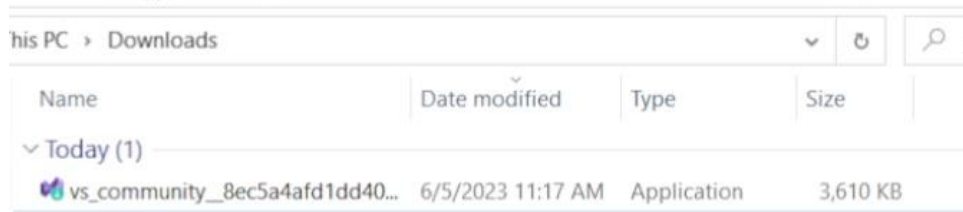


Ilustración 65 Archivos descargados

Marcar la parte de desarrollo con C++ las siguientes acciones y presionar en instalar.

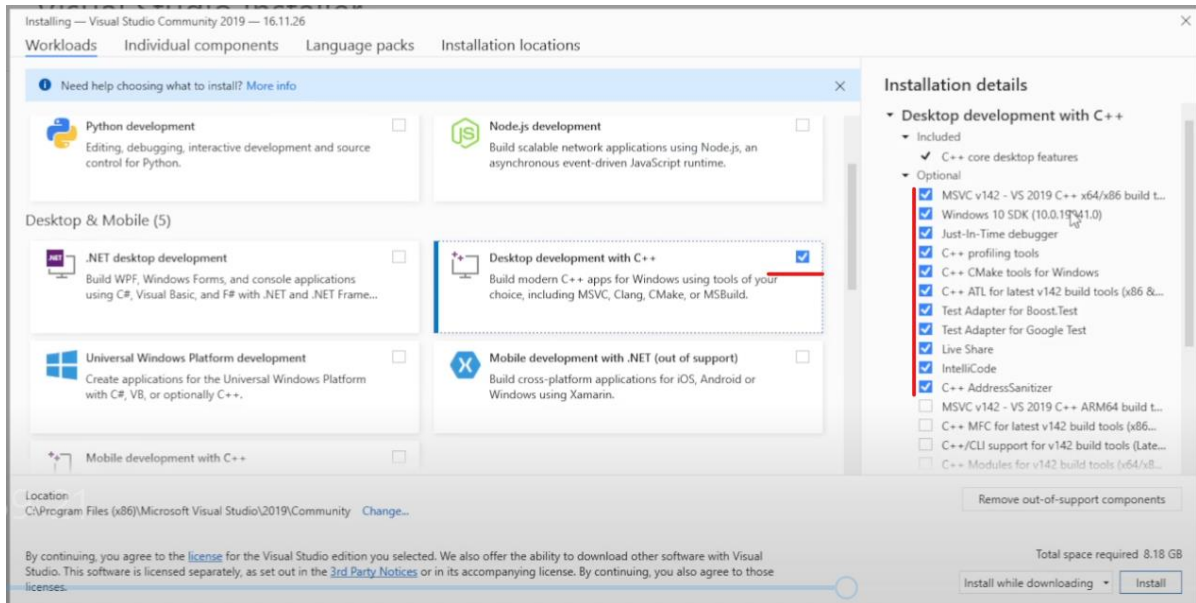


Ilustración 66 Seleccionar desarrollo con C++

4. Iniciaré la descarga en la computadora, una vez finalizada cerrar el instalador.

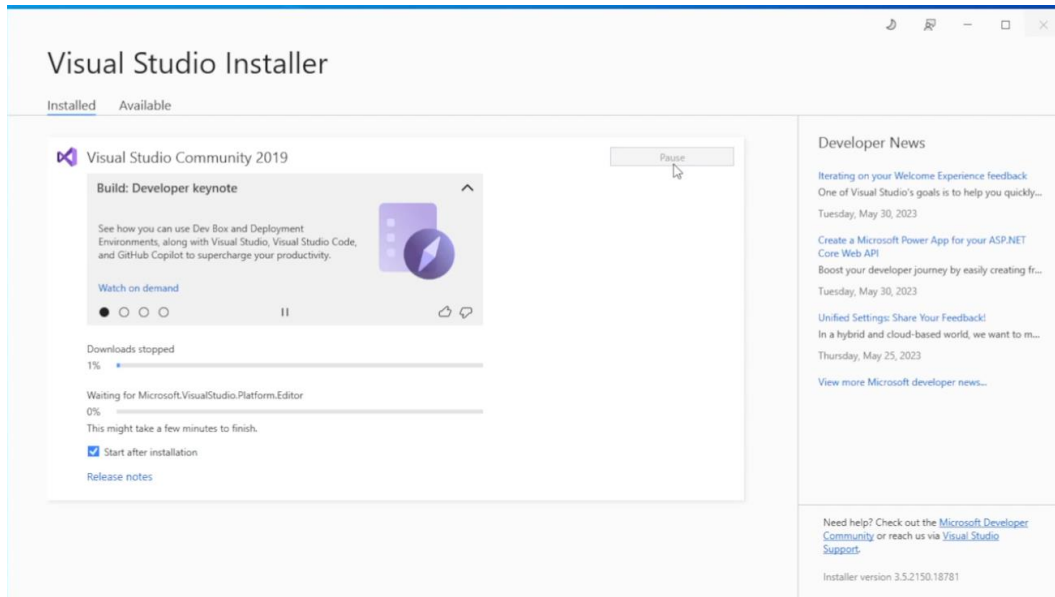


Ilustración 67 Instalando componentes de C++

Instalación de CUDA y CUDNN

CUDA permite ejecutar procesos de inteligencia artificial usando la GPU conocido como Unidad de Procesamiento Grafico, juega un papel fundamental para la captura de video y predicción usando YOLO v8, de igual forma se complementa con Pytorch para el entrenamiento de modelos con datos personalizados.

Pasos

1. Descargar el instalador de CUDA

https://developer.nvidia.com/cuda-11-8-0-download-archive?target_os=Windows&target_arch=x86_64&target_version=10&target_type=exe_local

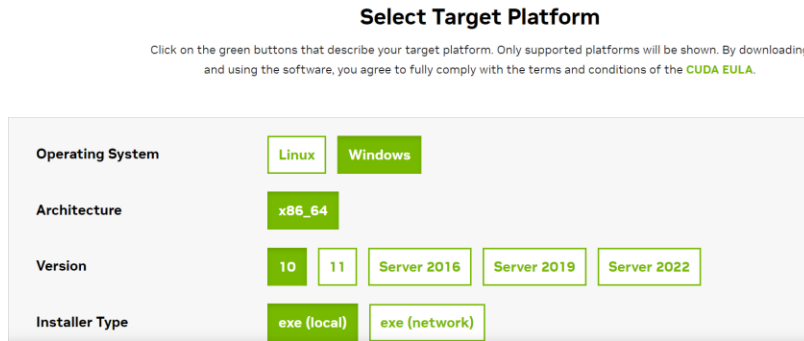


Ilustración 68 Descargar CUDA configuración

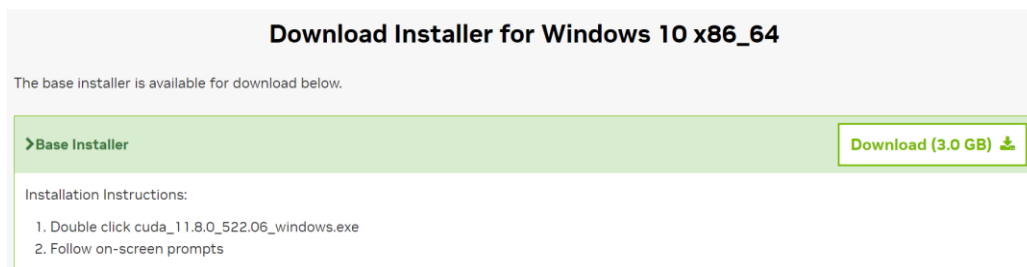


Ilustración 69 Descarga de CUDA

2. Ejecutar el instalador descargado.



Ilustración 70 Ejecutar el ejecutable de CUDA como administrador

3. Digitar para extraer los archivos de instalación, por defecto se instala en el directorio

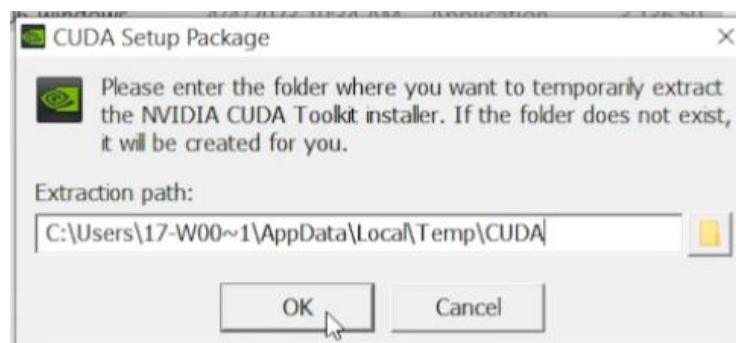


Ilustración 71 Directorio temporal del instalador de CUDA

4. Extraer archivos

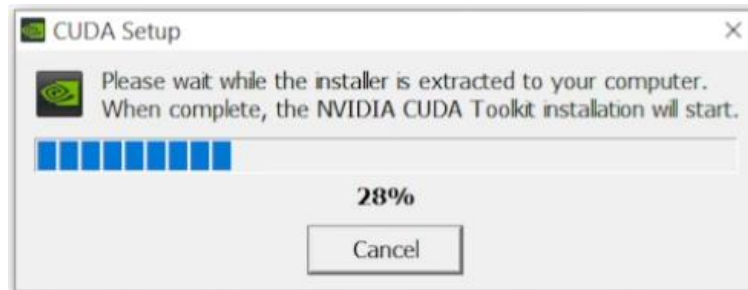


Ilustración 72 Instalando CUDA

5. Esperar que la verificación de compatibilidad de NVIDIA CUDA con la computadora.



Ilustración 73 Verificando compatibilidad

6. Aceptar los términos y condiciones y continuar.



Ilustración 74 Aceptar términos y condiciones

7. Utilizar la configuración recomendada (Express) y presionamos siguiente.



Ilustración 75 Instalación express

8. Instalar visual studio, si no está instalado aparecerá la siguiente ventana. En este apartado instalar visual studio para ello verificar el paso instalación de visual studio. También es posible que aparezca cuando no es compatible con el visual studio instalado, si fuera el caso tocara realizar la actualización a la versión requerida.

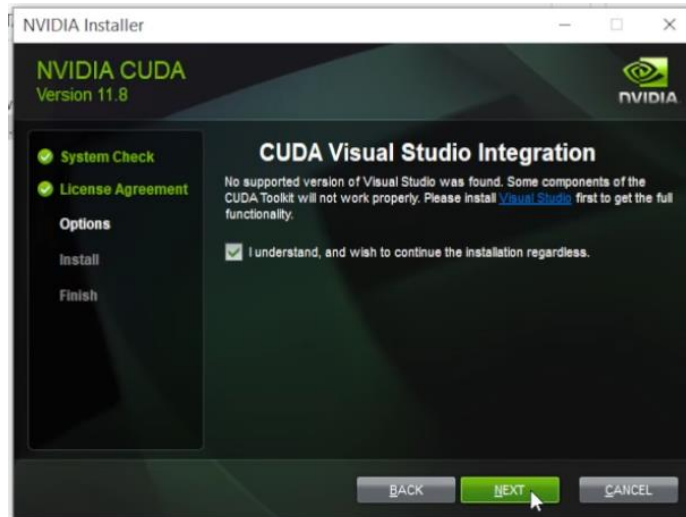


Ilustración 76 Solicitara instalación de visual studio

9. Iniciar la instalación

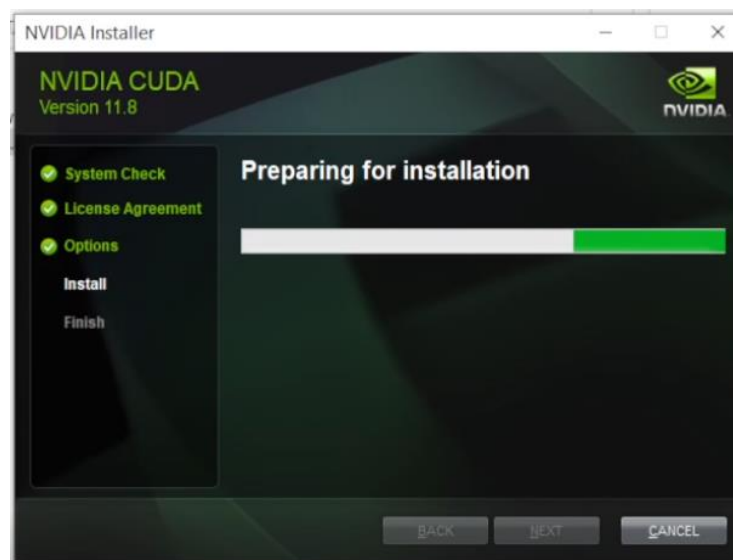


Ilustración 77 Preparando para instalación

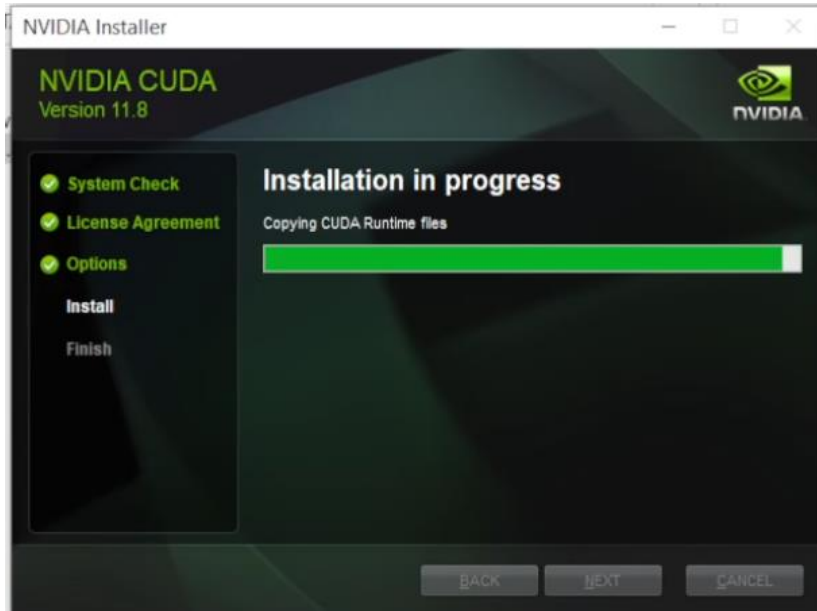


Ilustración 78 Instalación de CUDA

10. Finalizara la instalación y se mostrara todos los componentes instalados en el equipo.



Ilustración 79 Instalación finalizada de CUDA

11. Cerrar el instalador y proceder a configurar CUDNN, descargar CUDNN desde la siguiente ruta.

NOTA: para descargar CUDNN deberá estar registrado

<https://developer.nvidia.com/cudnn>.

<https://developer.nvidia.com/rdp/cudnn-archive>

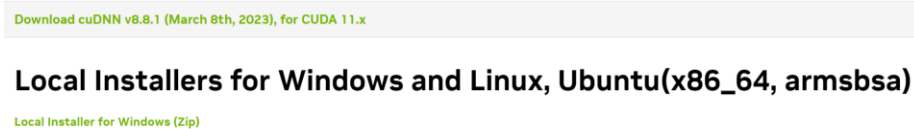


Ilustración 80 Descarga de CUDNN

12. Una vez descargado se procede a configurar los archivos, extraer los archivos del paquete zip descargado, aparecerán los siguientes.

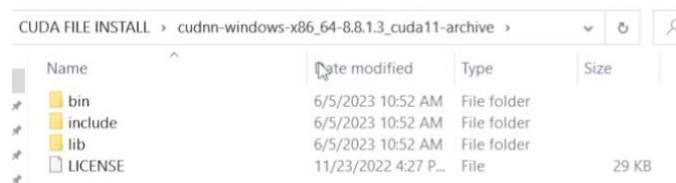


Ilustración 81 Extracción de archivos descargados de CUDNN

13. Ubicarse en la carpeta donde se instaló CUDA y copiar los archivos descargados de CUDNN en la carpeta correspondiente en CUDA, los archivos a copiar corresponden con el nombre en ambas carpetas.

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8

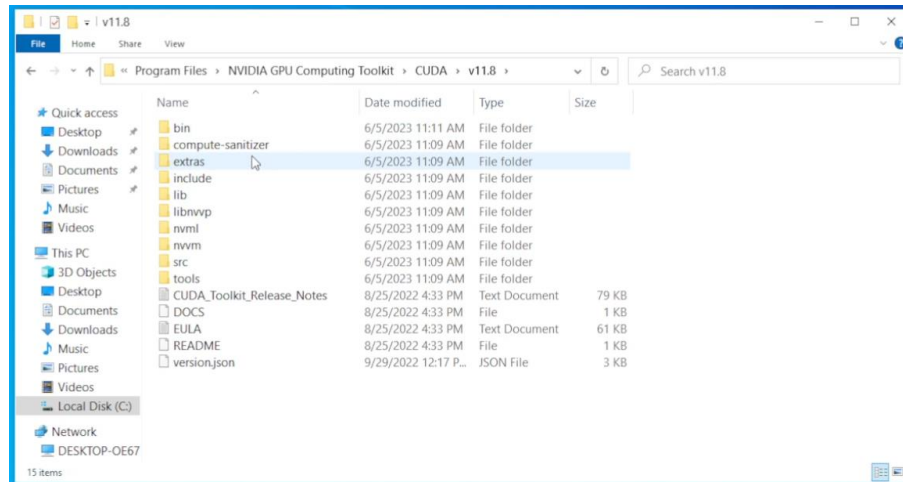


Ilustración 82 Directorio de instalación de CUDA

14. Las carpetas quedaran de la siguiente forma.

Carpeta bin

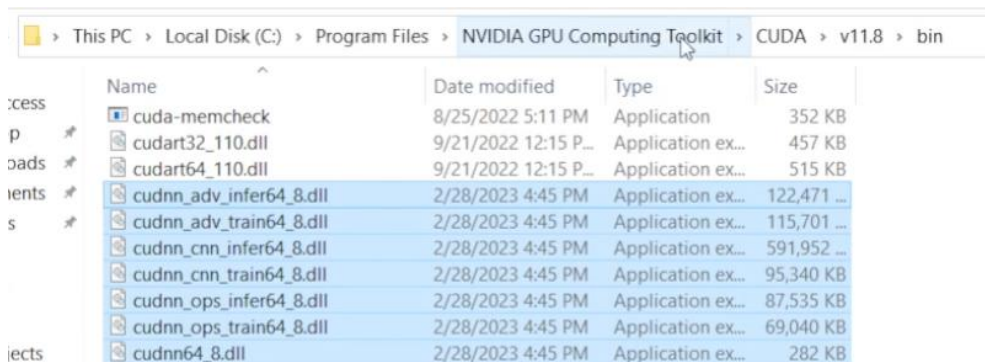
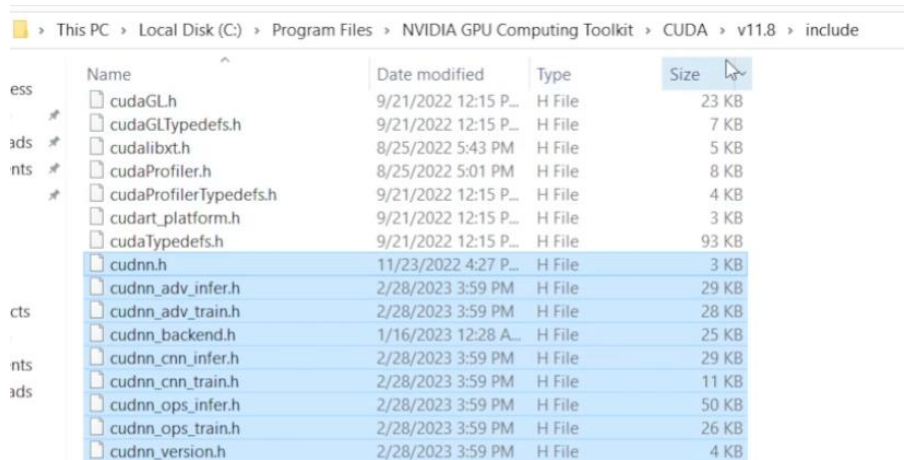


Ilustración 83 Adición de archivos CUDNN a CUDA carpeta bin

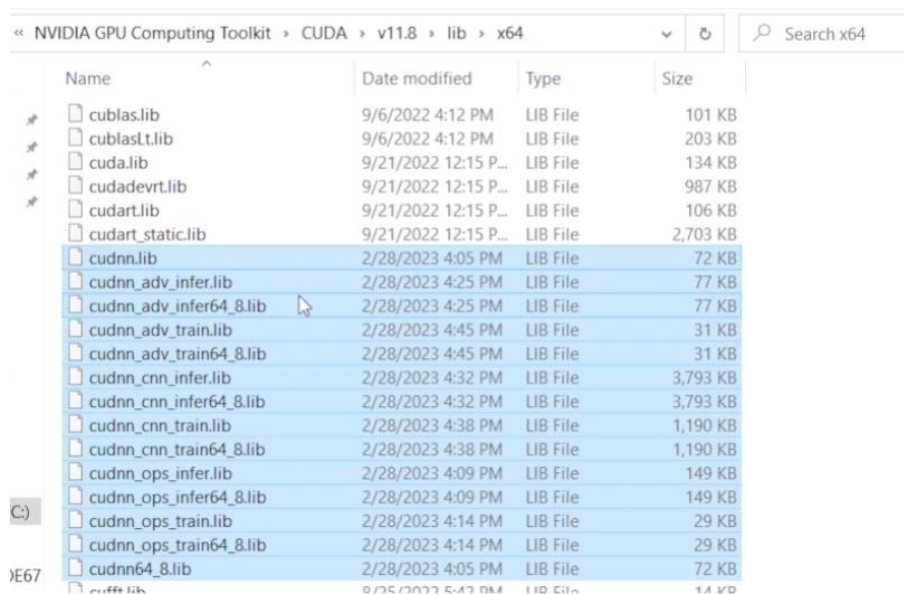
Carpeta include



Name	Date modified	Type	Size
cudaGL.h	9/21/2022 12:15 P...	H File	23 KB
cudaGLTypedefs.h	9/21/2022 12:15 P...	H File	7 KB
cudaLibx.h	8/25/2022 5:43 PM	H File	5 KB
cudaProfiler.h	8/25/2022 5:01 PM	H File	8 KB
cudaProfilerTypedefs.h	9/21/2022 12:15 P...	H File	4 KB
cudaPlatform.h	9/21/2022 12:15 P...	H File	3 KB
cudaTypedefs.h	9/21/2022 12:15 P...	H File	93 KB
cuda.h	11/23/2022 4:27 P...	H File	3 KB
cudaAdvInfer.h	2/28/2023 3:59 PM	H File	29 KB
cudaAdvTrain.h	2/28/2023 3:59 PM	H File	28 KB
cudaBackend.h	1/16/2023 12:28 A...	H File	25 KB
cudaCnnInfer.h	2/28/2023 3:59 PM	H File	29 KB
cudaCnnTrain.h	2/28/2023 3:59 PM	H File	11 KB
cudaOpsInfer.h	2/28/2023 3:59 PM	H File	50 KB
cudaOpsTrain.h	2/28/2023 3:59 PM	H File	26 KB
cudaVersion.h	2/28/2023 3:59 PM	H File	4 KB

Ilustración 84 Adición de archivos CUDNN a CUDA carpeta include

Carpeta lib/x64



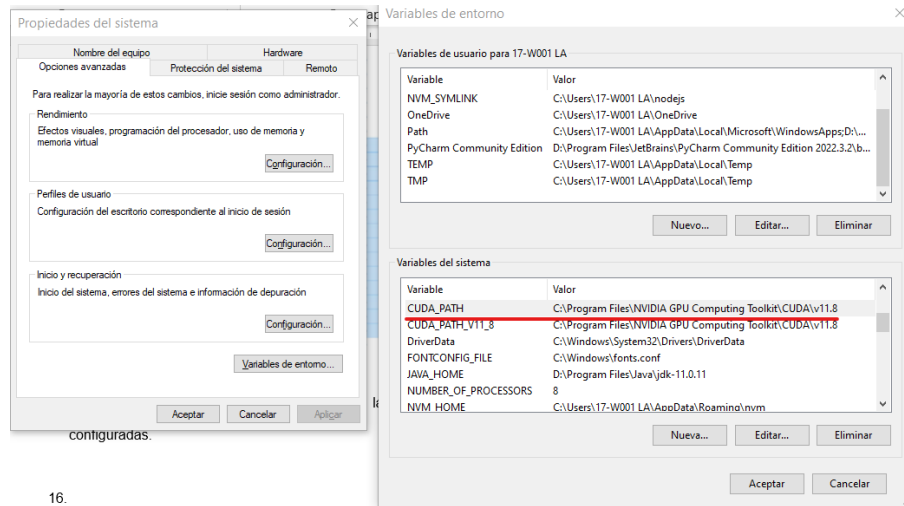
Name	Date modified	Type	Size
cublas.lib	9/6/2022 4:12 PM	LIB File	101 KB
cublasLt.lib	9/6/2022 4:12 PM	LIB File	203 KB
cuda.lib	9/21/2022 12:15 P...	LIB File	134 KB
cudaDevrt.lib	9/21/2022 12:15 P...	LIB File	987 KB
cudaStatically.lib	9/21/2022 12:15 P...	LIB File	106 KB
cudaStatically.lib	9/21/2022 12:15 P...	LIB File	2,703 KB
cuda.lib	2/28/2023 4:05 PM	LIB File	72 KB
cudaAdvInfer.lib	2/28/2023 4:25 PM	LIB File	77 KB
cudaAdvInfer64_8.lib	2/28/2023 4:25 PM	LIB File	77 KB
cudaAdvTrain.lib	2/28/2023 4:45 PM	LIB File	31 KB
cudaAdvTrain64_8.lib	2/28/2023 4:45 PM	LIB File	31 KB
cudaCnnInfer.lib	2/28/2023 4:32 PM	LIB File	3,793 KB
cudaCnnInfer64_8.lib	2/28/2023 4:32 PM	LIB File	3,793 KB
cudaCnnTrain.lib	2/28/2023 4:38 PM	LIB File	1,190 KB
cudaCnnTrain64_8.lib	2/28/2023 4:38 PM	LIB File	1,190 KB
cudaOpsInfer.lib	2/28/2023 4:09 PM	LIB File	149 KB
cudaOpsInfer64_8.lib	2/28/2023 4:09 PM	LIB File	149 KB
cudaOpsTrain.lib	2/28/2023 4:14 PM	LIB File	29 KB
cudaOpsTrain64_8.lib	2/28/2023 4:14 PM	LIB File	29 KB
cuda64_8.lib	2/28/2023 4:05 PM	LIB File	72 KB

Ilustración 85 Adición de archivos CUDNN a CUDA carpeta lib/x64

15. Una vez configurado el sistema, verificar que las variables de entorno estén configuradas.

- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\bin

- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\libnvvp
- C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\extras\CUPTI\lib64



16.

Ilustración 86 configuraciones variables de entorno CUDA

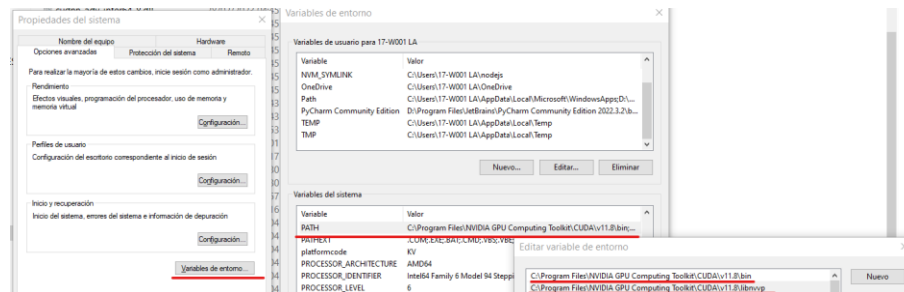


Ilustración 87 Configuración variables de entorno CUDA

16. Reiniciar el sistema, una vez configurado CUDA en el panel de control de NVIDIA se podrá observar que el sistema reconoce las versiones instaladas.

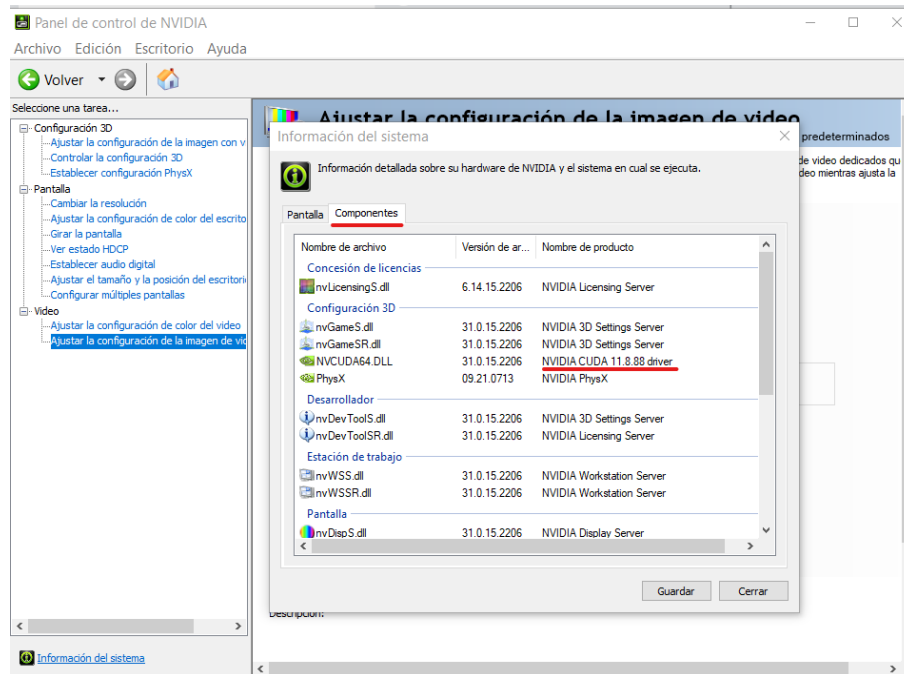


Ilustración 88 Panel de control NVIDIA con la versión instalada

Instalación y configuración de anaconda3

Para el desarrollo del proyecto se ha utilizado anaconda para ejecutar comandos y crear entornos virtuales para el entrenamiento del modelo.

Pasos

1. Descarga e instalación acceder al enlace de descarga y buscar, se está trabajando con la última versión a la fecha 06/06/2023

<https://www.anaconda.com/>

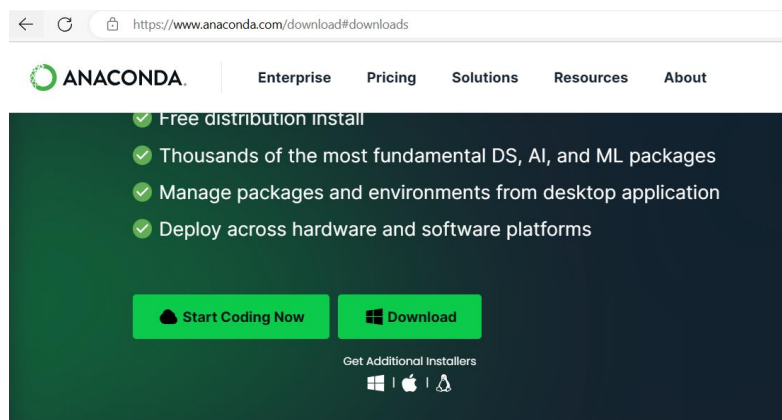


Ilustración 89 Descarga de Anaconda3


 Anaconda3-2023.03-Windows-x86_64.exe

Ilustración 90 Ejecutable de Anaconda3

2. Ejecutar el instalador, aceptar los términos y condiciones,



Ilustración 91 Ejecución del instalador de Anaconda3

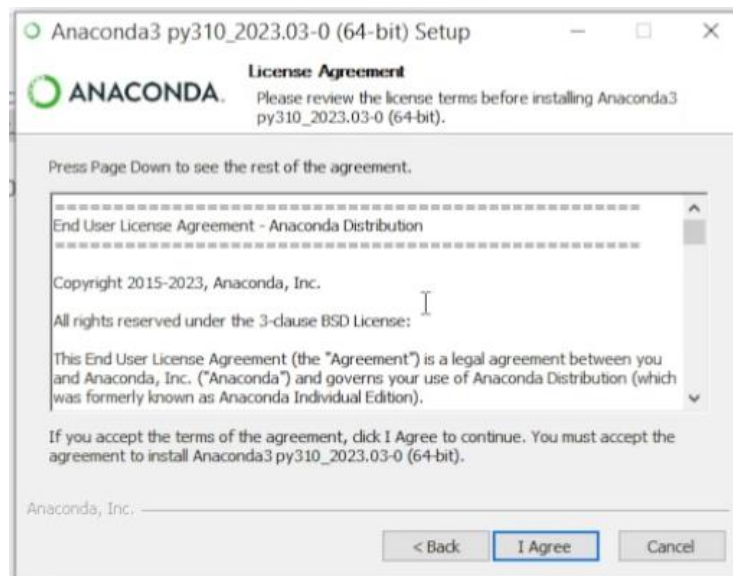


Ilustración 92 Estar de acuerdo con los términos

3. Seleccionar usuarios a usar el software y presionar siguiente.

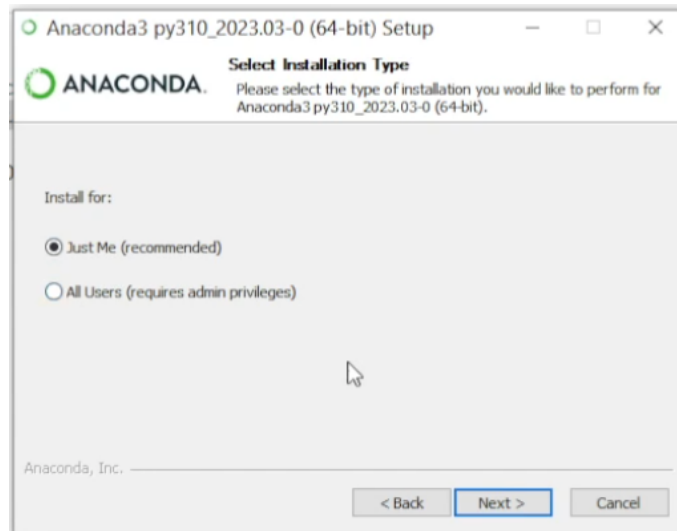


Ilustración 93 Instalar solo para uso del equipo

4. Aparecerá la ruta de instalación, si el nombre de una carpeta que se encuentre en la ruta de instalación contiene espacios nos pedirá cambiarla, es recomendable hacer el cambio.

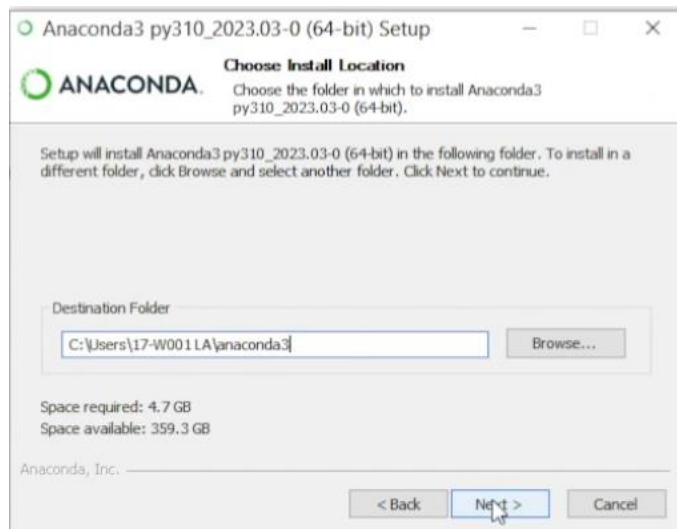


Ilustración 94 Cambio de directorio de instalación

5. Configurar el modal para instalación, las opciones para las variables de entorno se pueden agregar manualmente o dar click en Add Anaconda3 to my PATH environment variable para que sean agregadas por el instalador y procedemos a instalar.

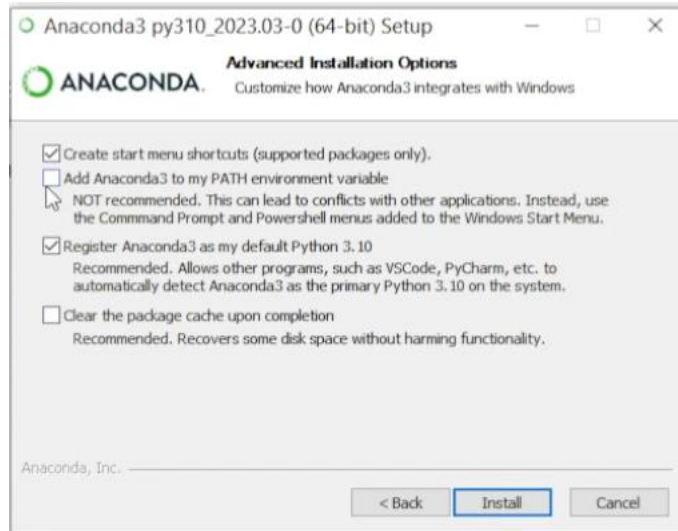


Ilustración 95 Adición de variables de entorno con el instalador

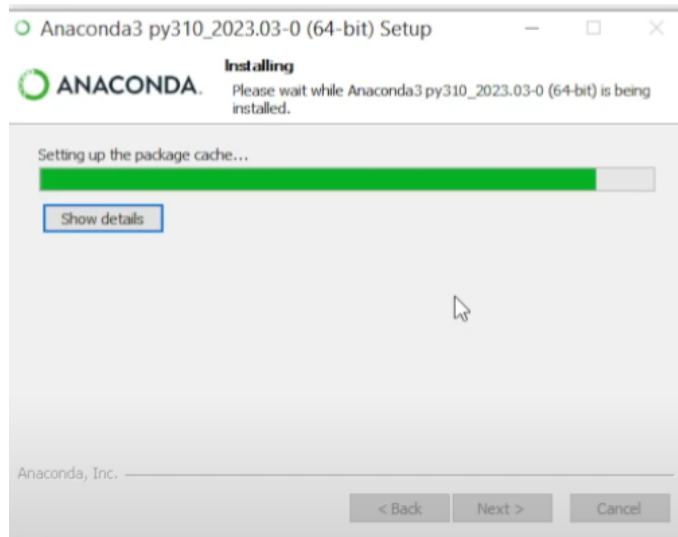


Ilustración 96 Instalando Anaconda3

6. Una vez instalado, presionar siguiente y estará instalado anaconda3, presionamos finalizar.



Ilustración 97 Anaconda instalado

7. En variables de entorno verificar que se encuentre en PATH la siguiente variable correspondiente al equipo donde se instaló.

D:\Program Files\anaconda3\Scripts

8. Para comprobar la versión instalada en el equipo ejecutamos el cmd de Windows, ejecutamos el comando "conda -version".

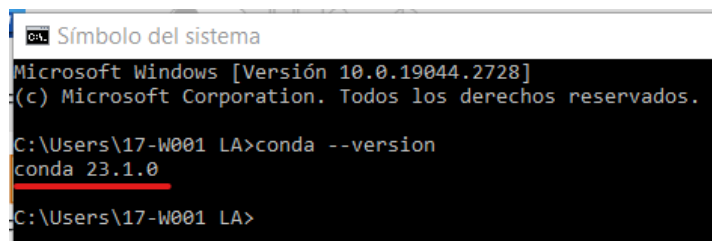


Ilustración 98 Comprobación de instalación

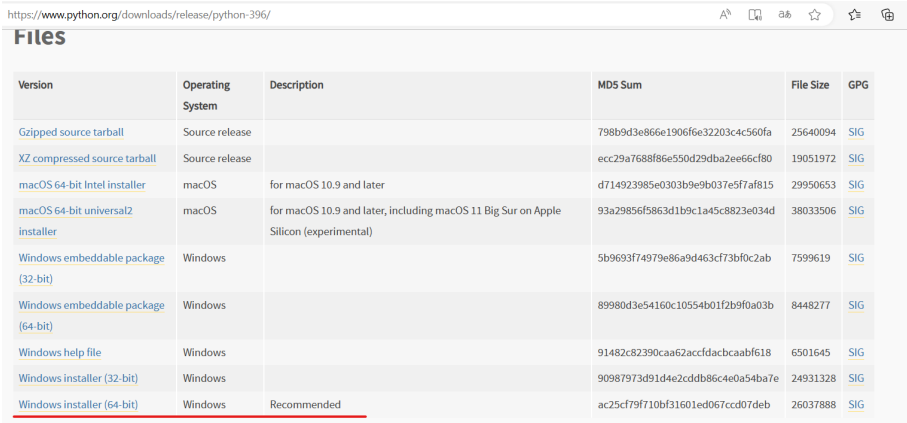
Instalación de Python

Anaconda instala la versión de Python 3.10, es posible trabajar con esa versión, para el proyecto se instaló la versión 3.9.6 como una alternativa para trabajar.

Pasos

1. Descarga e instalación, descargar desde el enlace.

<https://www.python.org/downloads/release/python-396/>



The screenshot shows the Python 3.9.6 download page with a table of files. The table has columns for Version, Operating System, Description, MD5 Sum, File Size, and GPG. The 'Windows installer (64-bit)' row is highlighted with a red underline.

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		798b9d3e866e1906f6e32203c4c560fa	25640094	SIG
XZ compressed source tarball	Source release		ecc29a7688f86e550d29dba2ee66cf80	19051972	SIG
macOS 64-bit Intel installer	macOS	for macOS 10.9 and later	d714923985e0303b9e9b037e5f7af815	29950653	SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later, including macOS 11 Big Sur on Apple Silicon (experimental)	93a29856f5863d1b9c1a45c8823e034d	38033506	SIG
Windows embeddable package (32-bit)	Windows		5b9693f74979e86a9d463c773bf0c2ab	7599619	SIG
Windows embeddable package (64-bit)	Windows		89980d3e54160c10554b01f2b9f0a03b	8448277	SIG
Windows help file	Windows		91482c82390caa62accfdacbcabf618	6501645	SIG
Windows installer (32-bit)	Windows		90987973d91d4e2cddb86c4e0a54ba7e	24931328	SIG
Windows installer (64-bit)	Windows	Recommended	ac25cf797f10bf31601ed067ccd07deb	26037888	SIG

Ilustración 99 Descarga de Python 3.9.6

2. Una vez descargado proceder a su instalación, marcar las opciones de Add Python 3.9 to PATH para que agregue las variables de entorno por defecto y damos click en Instalar ahora.



Ilustración 100 Adición de variables de entorno desde el instalador de Python

3. Iniciar la instalación

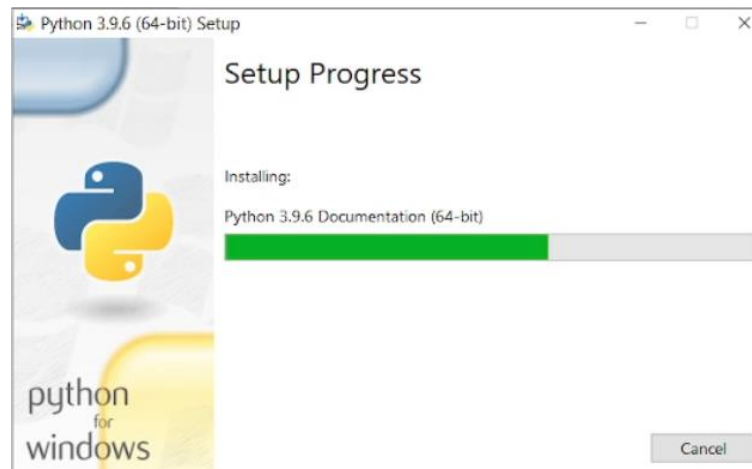


Ilustración 101 Instalando Python

4. Aparecerá en el modal que fue configurado exitosamente procedemos a cerrar.

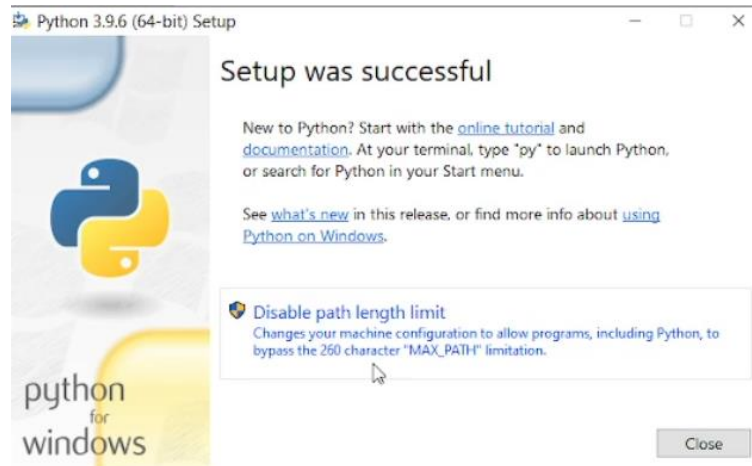


Ilustración 102 Python instalado correctamente

5. Verificar las variables de entorno creadas, dentro de PATH se debe de verificar que estén las siguientes, si no se encuentran debe agregarlas, recuerda que la ruta depende de donde se realizó la instalación.

```
C:\Users\PC01\AppData\Local\Programs\Python\Python39
```

```
C:\Users\PC01\AppData\Local\Programs\Python\Python39\Scripts
```

6. Verificar la instalación desde el CMD de Windows, al crear las variables de entorno desde el instalador no debería generar conflictos y dar la versión 3.9.6

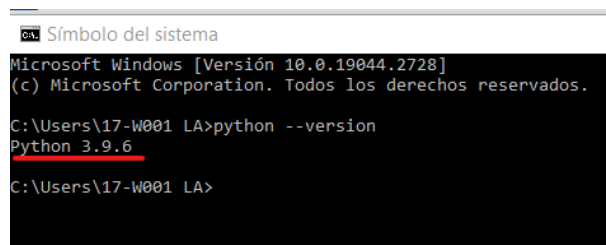


Ilustración 103 Comprobar instalación

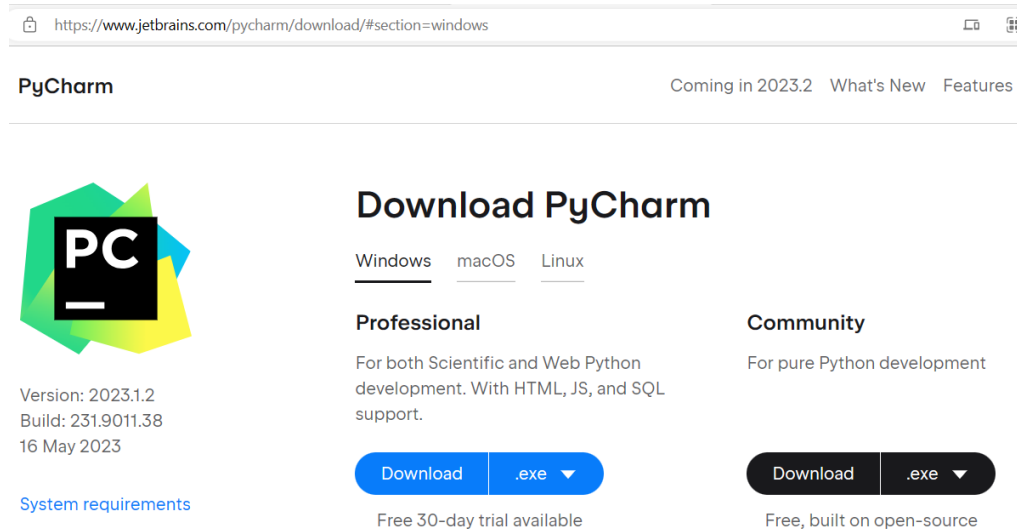
Instalación de PyCharm

PyCharm se ha utilizado como editor de código, nos proporciona un entorno amigable y fácil de utilizar, la versión Community es gratis y de código abierto, puede utilizar otro editor de código si ve conveniente.

Pasos

1. Descarga e instalación, acceder al enlace de descarga y descargar la versión community

[Download PyCharm: Python IDE for Professional Developers by JetBrains](https://www.jetbrains.com/pycharm/download/#section=windows)



The screenshot shows the JetBrains website page for downloading PyCharm on Windows. The browser address bar displays the URL: <https://www.jetbrains.com/pycharm/download/#section=windows>. The page header includes the PyCharm logo and navigation links for "Coming in 2023.2", "What's New", and "Features". The main content area features the PyCharm logo (a stylized 'PC' in a black square on a colorful background) and version information: "Version: 2023.1.2", "Build: 231.9011.38", and "16 May 2023". Below this is a link for "System requirements". The "Download PyCharm" section has tabs for "Windows", "macOS", and "Linux", with "Windows" selected. Under "Professional", it states "For both Scientific and Web Python development. With HTML, JS, and SQL support." and provides a "Download" button with a ".exe" dropdown menu, along with the note "Free 30-day trial available". Under "Community", it states "For pure Python development" and provides a "Download" button with a ".exe" dropdown menu, along with the note "Free, built on open-source".

Ilustración 104 Descarga de PyCharm

2. Ejecutar el instalador, aparecerá el siguiente modal presionar en siguiente.



Ilustración 105 Ejecutar el instalador de PyCharm

3. Indicar la ruta de instalación y presionar siguiente.

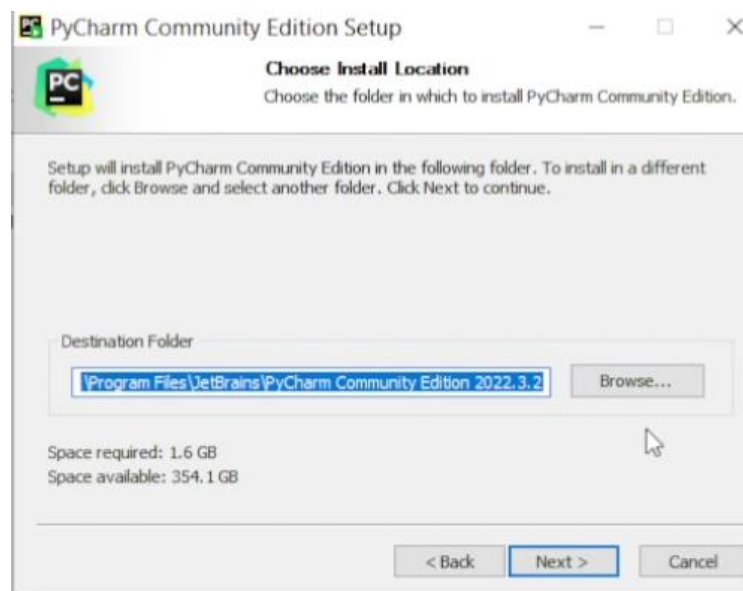


Ilustración 106 Directorio de instalación

- Indicar las opciones de instalación, se recomienda marcar las que aparecen en el recuadro y presionar siguiente.

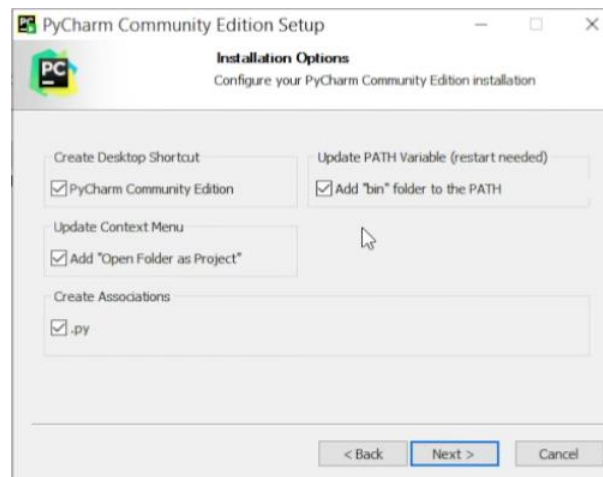


Ilustración 107 Opciones seleccionadas para instalar PyCharm

- Elegimos una carpeta para inicio de PyCharm y presionamos instalar

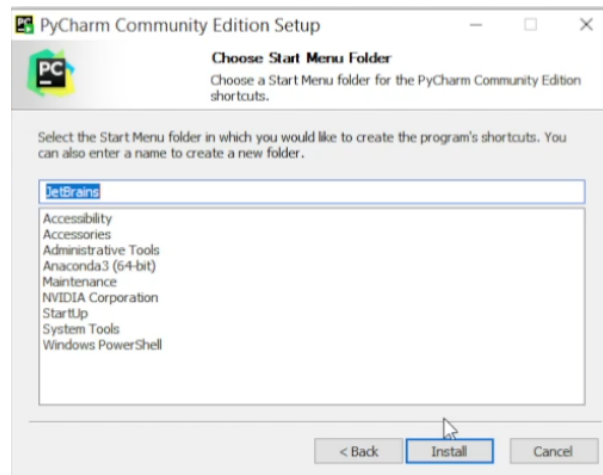


Ilustración 108 Folder de inicio PyCharm

6. Se realizará la instalación

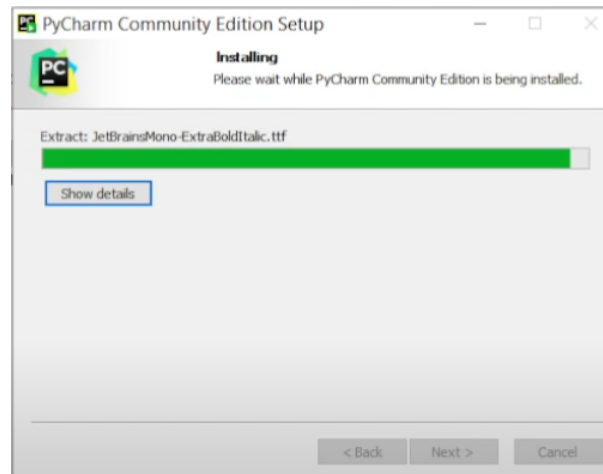


Ilustración 109 Instalación de PyCharm

7. Una vez finalizado pedirá reiniciar la computadora y presionar finalizar.

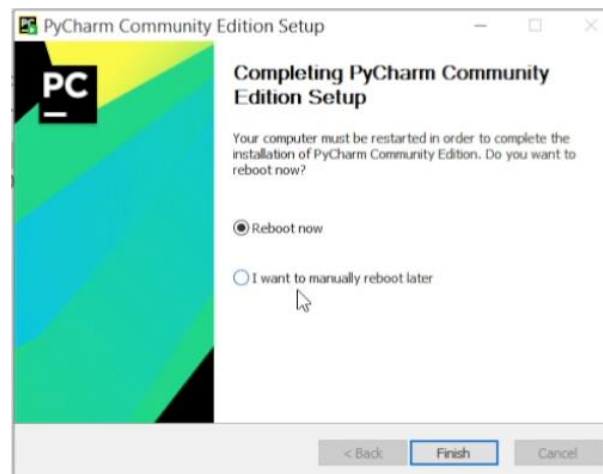


Ilustración 110 Reiniciar equipo

El equipo se reiniciará, a partir de acá ya podremos utilizar el editor de código.

Instalación de MySQL

MySQL es el gestor utilizado para base de datos, nos servirá para gestionar información útil sobre eventos detectados y configuraciones del sistema.

Pasos

1. Descargar desde el siguiente enlace

<https://downloads.mysql.com/archives/installer/>

MySQL Product Archives

MySQL Installer (Archived Versions)

⚠ Please note that these are old versions. New releases will have recent bug fixes and features!
To download the latest release of MySQL Installer, please visit [MySQL Downloads](#).

Product Version:
Operating System:

Windows (x86, 32-bit), MSI Installer <small>(mysql-installer-web-community-8.0.32.0.msi)</small>	Jan 3, 2023	2.4M	Download
Windows (x86, 32-bit), MSI Installer <small>(mysql-installer-community-8.0.32.0.msi)</small>	Jan 3, 2023	437.3M	Download

Ilustración 111 Descarga de MySQL

2. Una vez descargado ejecutar el instalador, aparecerá el siguiente modal, marcar opción personalizada (Custom) y presionar siguiente.

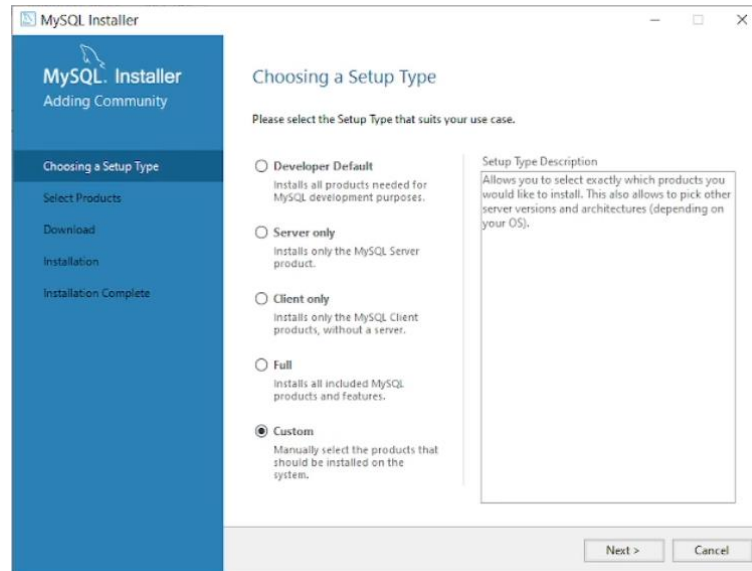


Ilustración 112 Instalación personalizada MySQL

3. Seleccionar los productos a instalar, en el recuadro mover los productos que serán instalados presionar siguiente.

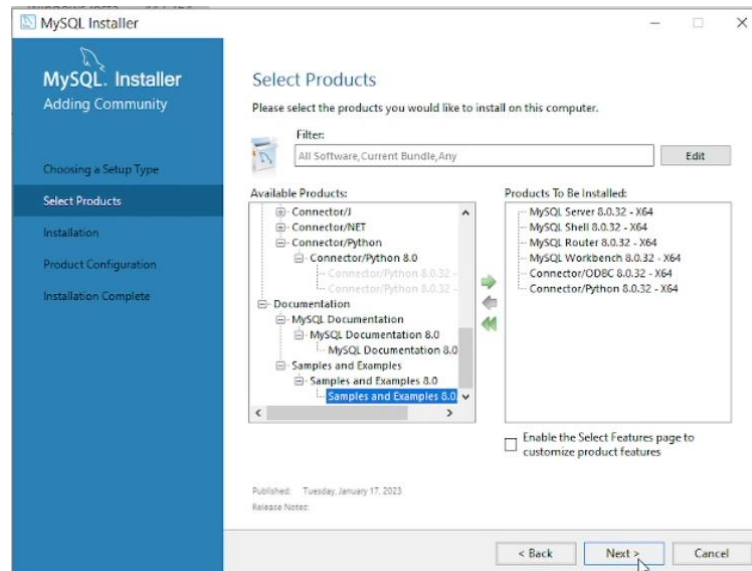


Ilustración 113 Opciones a instalar

4. Se instalarán los productos seleccionados al presionar ejecutar, luego presionamos siguiente.

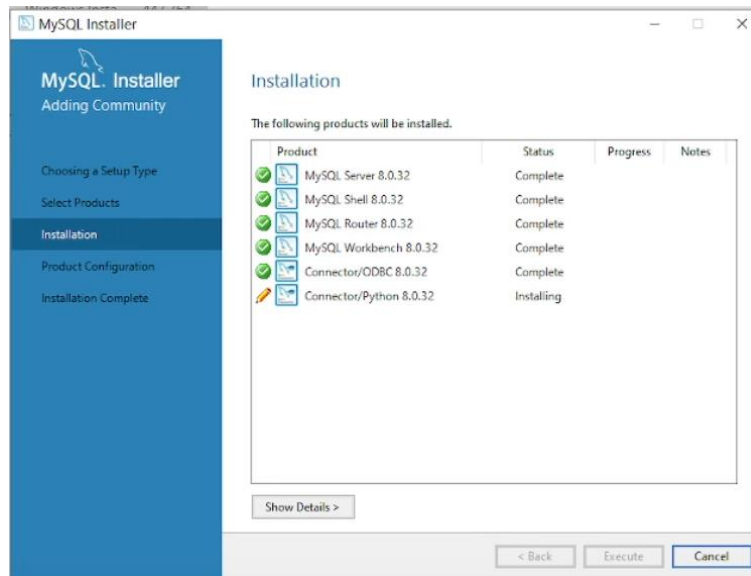


Ilustración 114 Instalación de opciones MySQL

5. Aparecerá el siguiente modal el cual no realizaremos cambio y presionamos siguiente

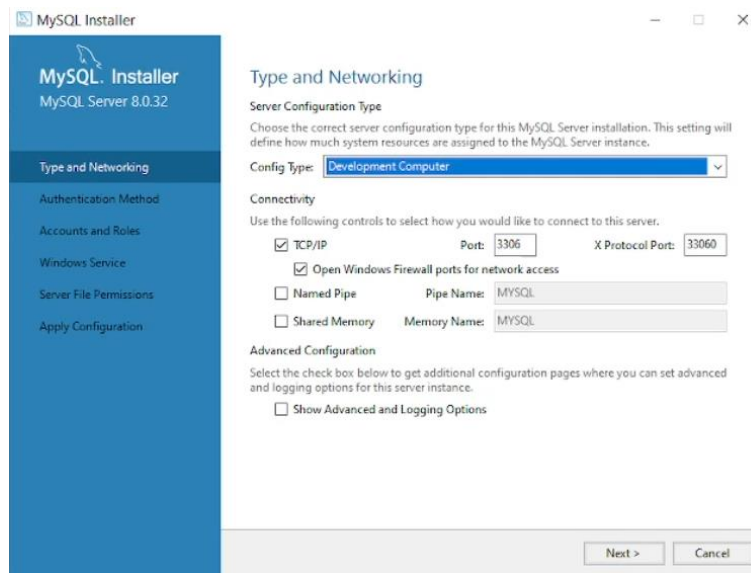


Ilustración 115 Elegir tipo de configuración de desarrollo

6. Aparecerá el modal siguiente y presionamos siguiente

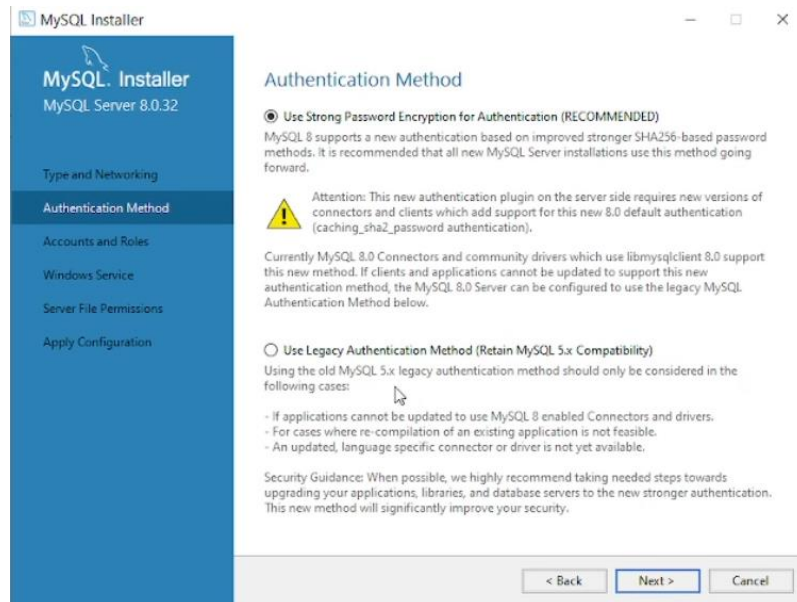


Ilustración 116 Método de autenticación por defecto

7. Digitamos una contraseña segura para inicio de sesión del usuario "root" y presionar siguiente.

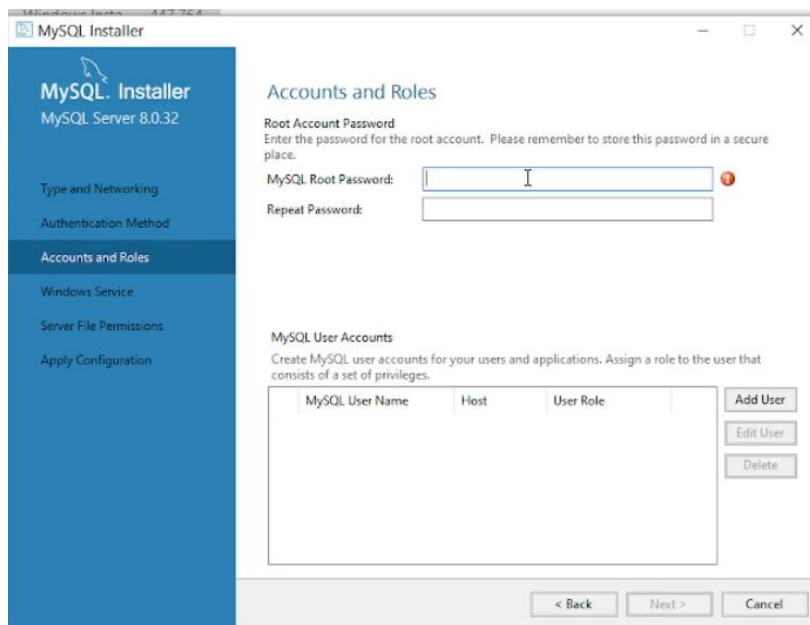


Ilustración 117 Configuración de credenciales de acceso

8. Para el siguiente modal presionar siguiente sin hacer modificaciones.

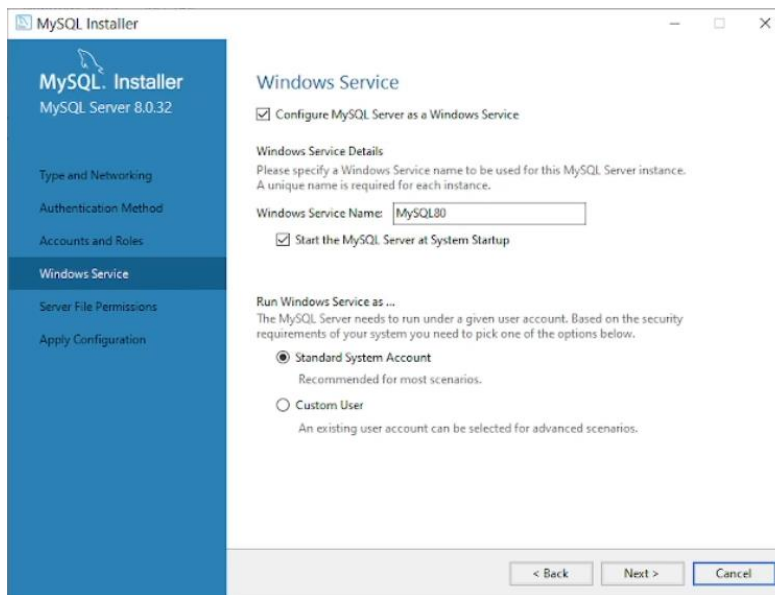


Ilustración 118 Servicio de Window MySQL

9. Configuramos los permisos de archivos del servidor, en este caso dejaremos la primera opción.

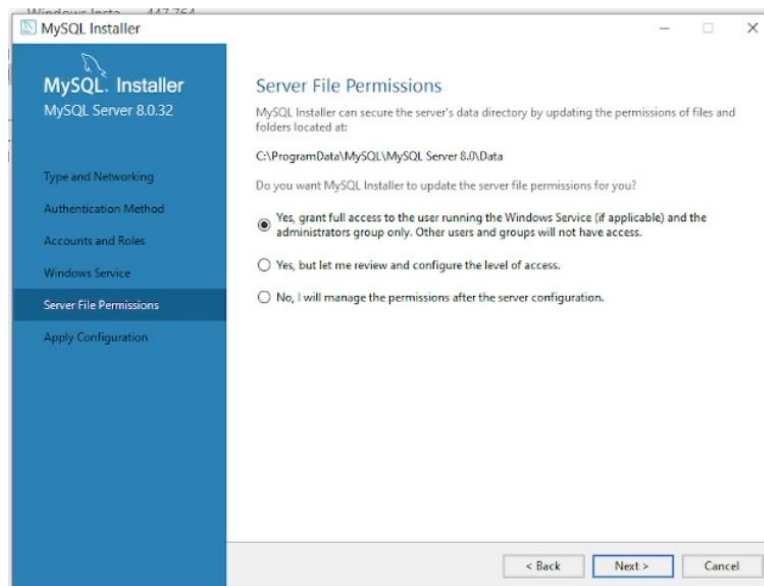


Ilustración 119 Permisos de archivos del servidor

10. En el siguiente modal presionar en ejecutar

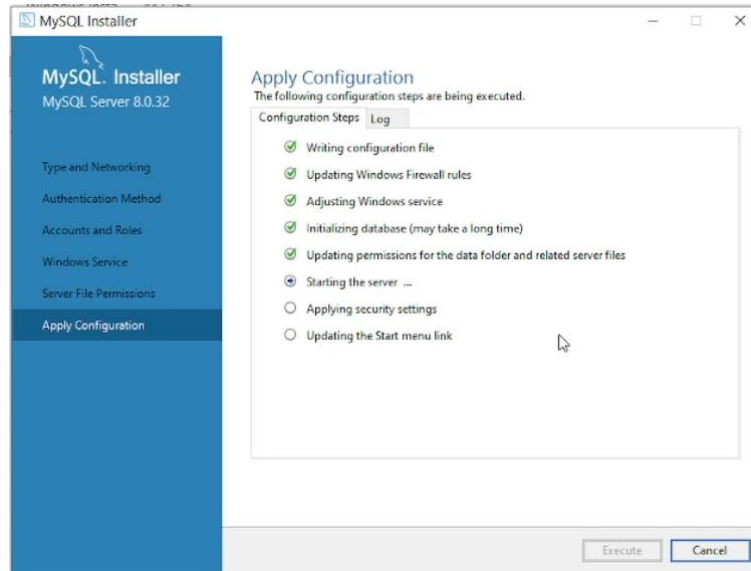


Ilustración 120 Aplicando configuración MySQL

11. Aparecerá la configuración del producto, presionar siguiente.

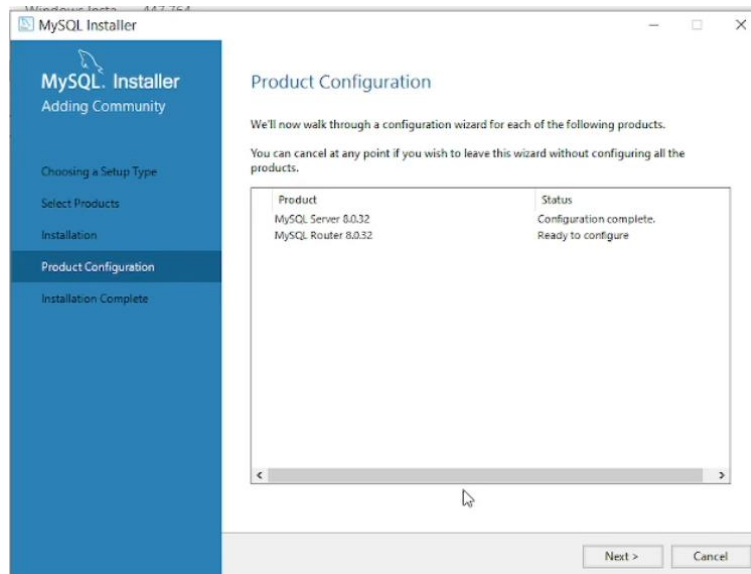


Ilustración 121 Configuración de MySQL

12. En el siguiente modal presionamos finalizar

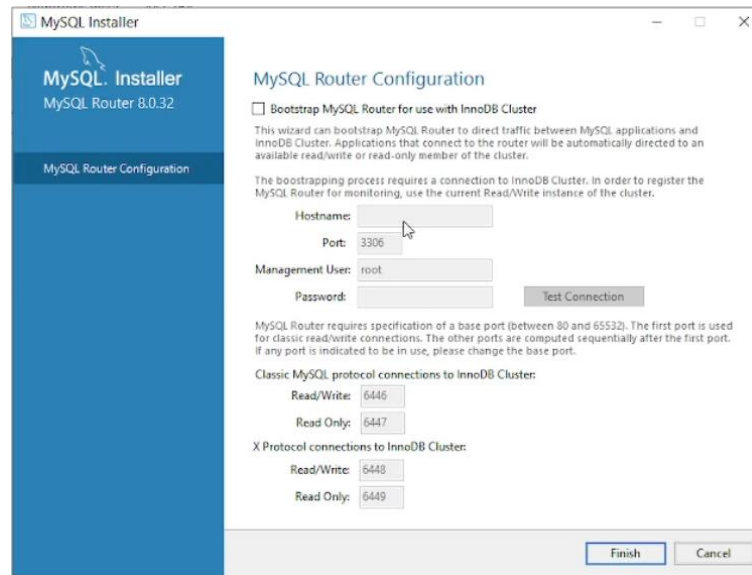


Ilustración 122 Finalizar instalación

13. Nos aparecerá la pantalla final con la instalación completada, presionar finalizar.

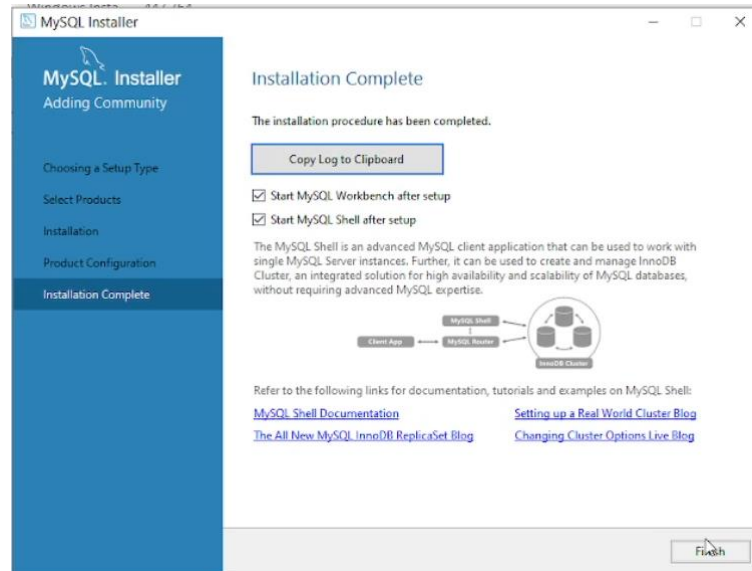


Ilustración 123 Instalación completada

14. Iniciar Wowkbench y configurar una conexión.

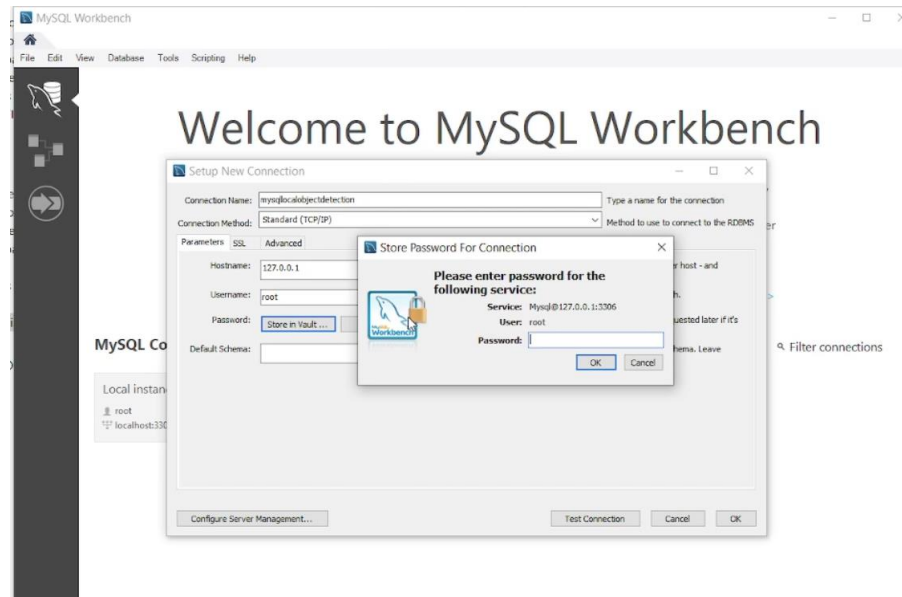


Ilustración 124 Configuración en MySQL Workbench

15. Este todo listo para trabajar en base de datos

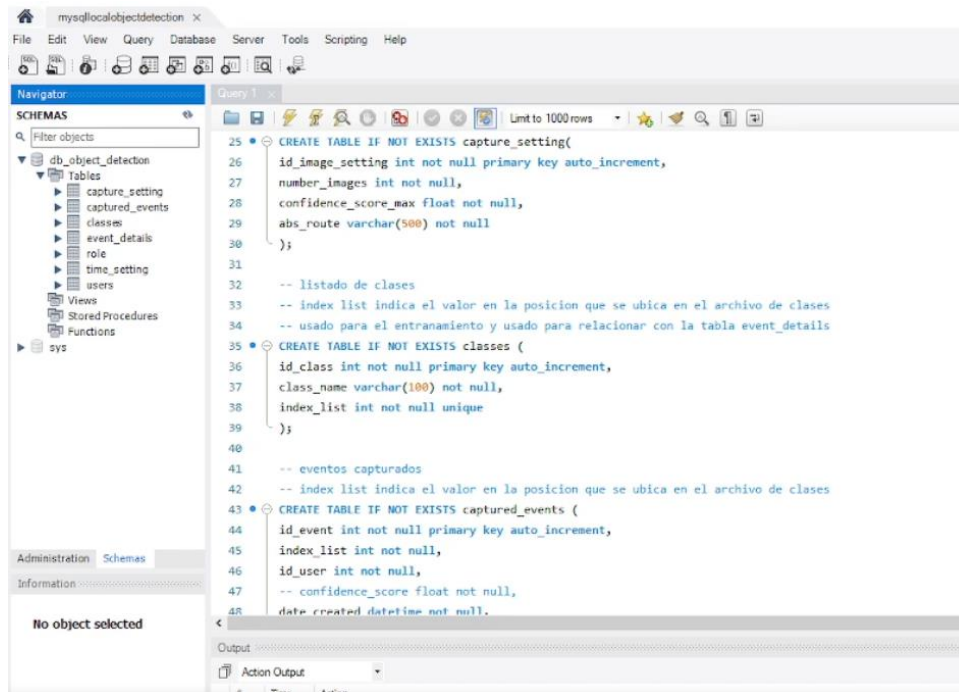


Ilustración 125 Ejecutando base de datos

Instalaciones opcionales

Esta instalación es opcional para etiquetar imágenes podremos usar diferentes herramientas entre estas `make sense` es una herramienta que permite etiquetar para detección de objetos y `labelImg` también usada para este fin.

Make sense

Pasos

1. Descarga e instalación

Para poder utilizar `make sense` de forma local se debe usar una máquina virtual para que no nos genere conflictos con las instalaciones ya creadas u otra computadora, de igual forma existe la alternativa de usarla en línea desde el enlace oficial.

<https://www.makesense.ai/>

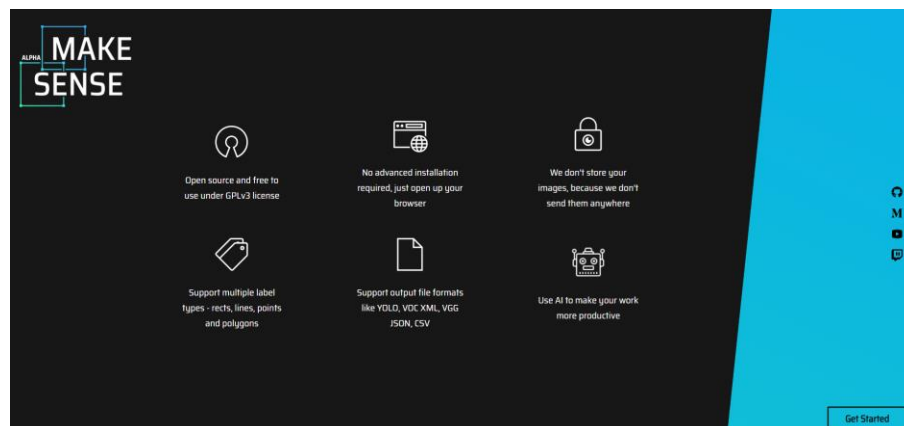


Ilustración 126 Descarga makesense


2. Si se usa de forma local instalar node js en la versión 16.16 para ello ir al siguiente enlace y descargarlo

<https://nodejs.org/en/blog/release/v16.16.0>

Windows 32-bit Installer: <https://nodejs.org/dist/v16.16.0/node-v16.16.0-x86.msi>

Windows 64-bit Installer: <https://nodejs.org/dist/v16.16.0/node-v16.16.0-x64.msi>

Ilustración 127 Descarga de node Js v16.16

 node-v16.16.0-x64.msi

3. Ejecutar el instalador, aceptar los términos y presionar siguiente.



Ilustración 128 Aceptar términos de node js

4. En el siguiente paso presionar siguiente.

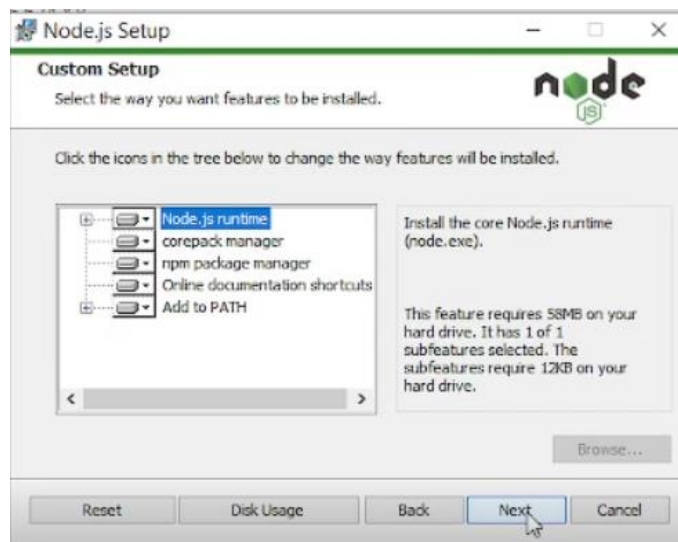


Ilustración 129 instalación node js personalizada

5. En este paso marcamos la caja de chequeo y presionar siguiente

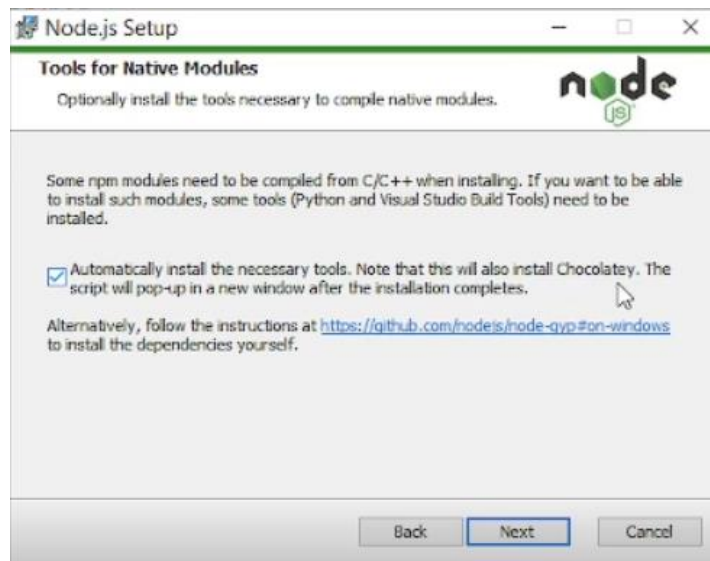


Ilustración 130 Instalar herramienta para módulos nativos

6. Procederá a instalar y esperar para finalizar.

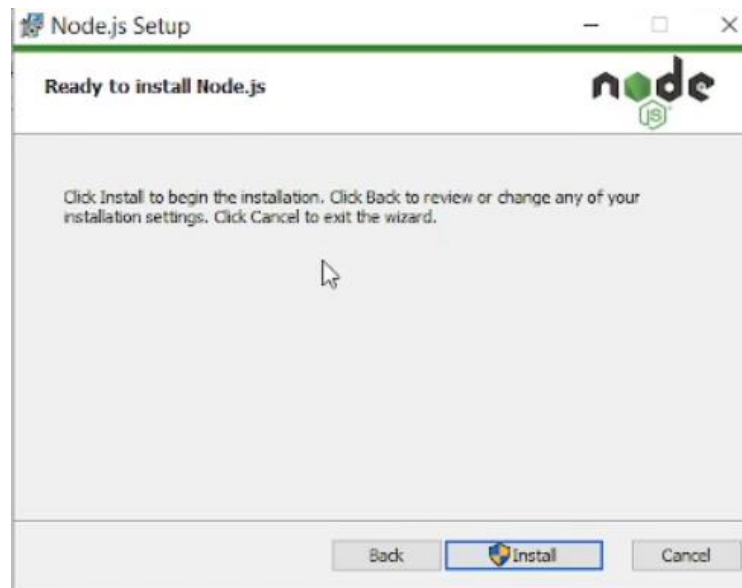


Ilustración 131 Confirmar instalación

7. Se desplegará una consola y se debe cerrar desde la opción cerrar sin presionar ninguna tecla.

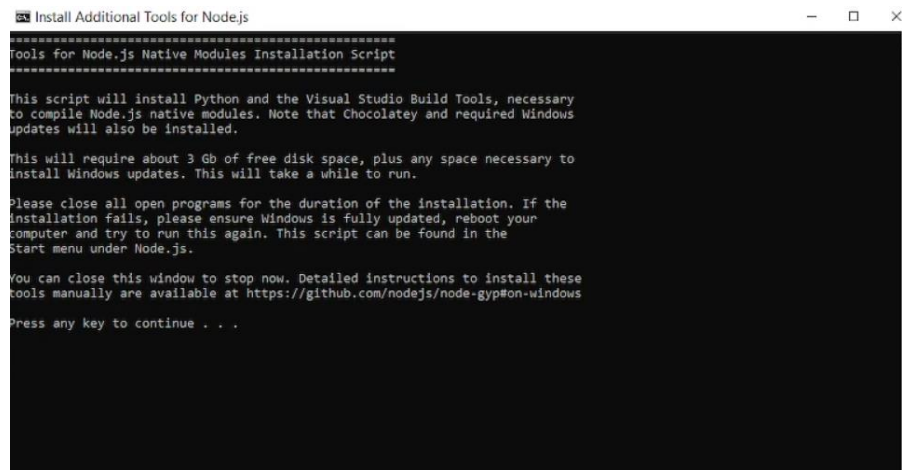


Ilustración 132 Cerrar sin continuar

8. A partir de este punto seguir los pasos que nos indica la página oficial

Clonar los archivos desde GitHub

<https://github.com/SkalskiP/make-sense>

```
# clone repository
```

```
git clone https://github.com/SkalskiP/make-sense.git
```

```
# navigate to main dir
```

```
cd make-sense
```

```
# install dependencies
```

```
npm install
```

```
# serve with hot reload at localhost:3000
```

```
npm start
```

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. All rights reserved.

C:\make_sense\make-sense-develop\make-sense-develop>npm start
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

> make-sense@1.11.0-alpha start
> vite --port 3000

VITE v3.0.2 ready in 2650 ms

  Local: http://127.0.0.1:3000/
  Network: use --host to expose
Browserslist: caniuse-lite is outdated. Please run:
  npx update-browserslist-db@latest
  Why you should do it regularly: https://github.com/browserslist/update-db#readme
```

Ilustración 133 Makesense ejecutándose

9. Ejecutándose en entorno local

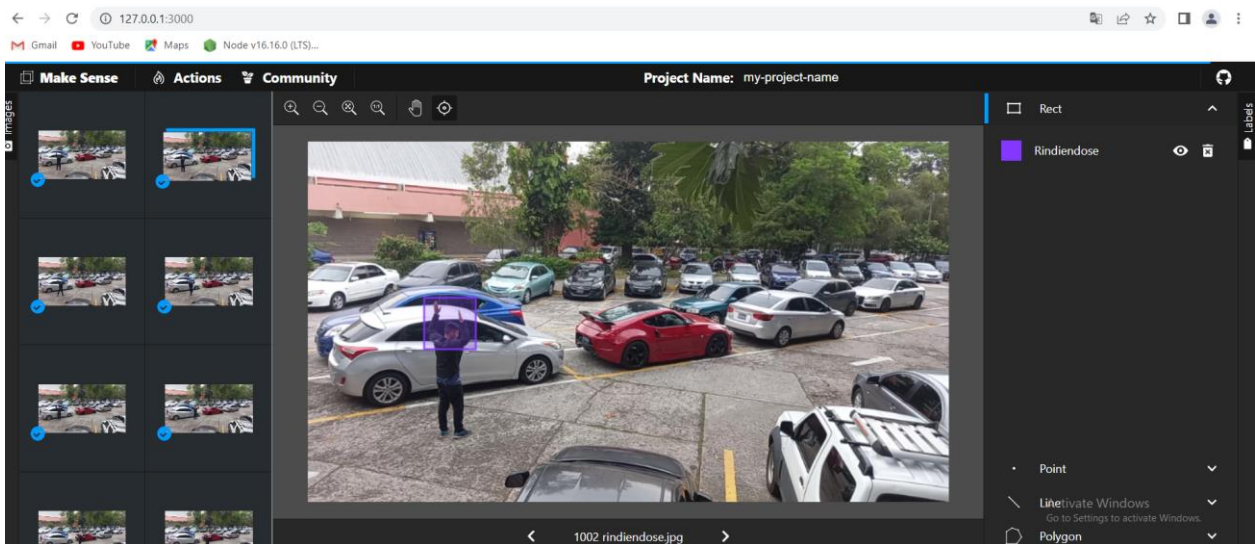


Ilustración 134 Utilizando makesense

LabelImg

Label img es otra alternativa para etiquetado de imágenes para ello se usa la consola de anaconda.



Pasos

1. Creación del entorno virtual, los datos pueden variar.

```
conda create -n yolov8_custom_project_example_name python=3.9
```

1. Activar el entorno virtual para poder trabajar.

```
conda activate yolov8_custom_project_example_name
```

2. Instalar LabelImg

```
pip install labelImg
```

3. Ejecutar labelImg, el cual deberá abrir el siguiente programa para etiquetado de imágenes.

```
labelImg
```

```
Anaconda Prompt - labelImg
(base) D:\PYTHON_PROJECT_ITCA\Prototipo_vision_artificial_using_conda\custom_project_model>conda activate yolov8_custom
(yolov8_custom) D:\PYTHON_PROJECT_ITCA\Prototipo_vision_artificial_using_conda\custom_project_model>labelImg
```

Ilustración 135 Ejecutando LabelImg desde consola conda

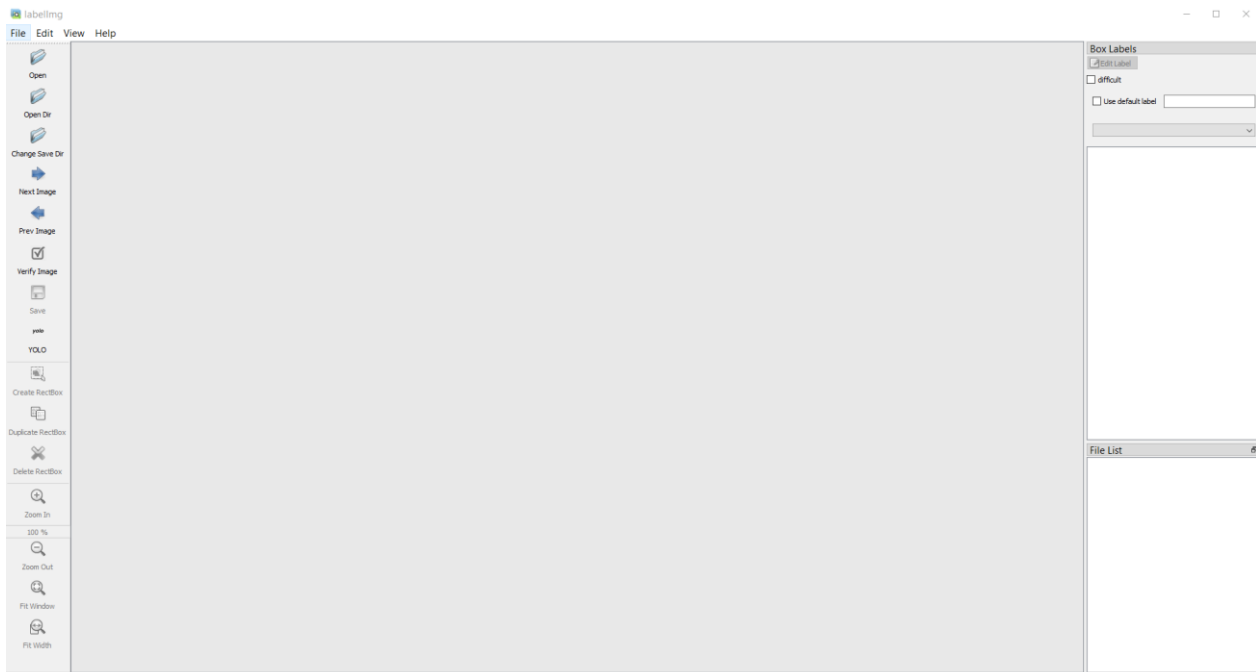


Ilustración 136 LabelImg ejecutándose

Creación de archivo ejecutable

Para compilar y crear un archivo ejecutable del proyecto, se utiliza la librería PyInstaller, la cual permite empaquetar la solución en un formato ejecutable para el sistema operativo Windows.

Se debe instalar la herramienta apoyándose del gestor de dependencias pip con el siguiente comando:

```
pip install pyinstaller
```

Posterior a la instalación, navegar a la carpeta del proyecto, en la cual se debe ejecutar el comando:

```
pyinstaller main.py --collect-all ultralytics --collect-all  
mtcnn --add-data settings_db.json;. --add-data
```

```
base_de_datos;base_de_datos --add-data classes.txt;. --add-data
images;images --add-data pesos_red_yolo;pesos_red_yolo
```

Para el comando anterior, el uso básico es `pyinstaller.py`, sin embargo, se necesita utilizar las siguientes opciones adicionales:

```
--collect-all ultralytics
```

Para agregar las librerías y dependencias de Ultralytics (YOLOv8) al empaquetado del proyecto.

```
--collect-all mtcnn
```

Para agregar las librerías y dependencias de mtcnn (detección de rostros con TensorFlow) al empaquetado del proyecto.

```
--add-data settings_db.json;.
```

Para incluir el archivo de configuración de credenciales al empaquetado del proyecto.

```
--add-data base_de_datos;base_de_datos
```

Para incluir la carpeta con el archivo con el script de base de datos al empaquetado del proyecto.

```
--add-data classes.txt;.
```

Para incluir el archivo con el listado de movimientos sospechosos soportados al empaquetado del proyecto.

```
--add-data images;images
```

Para incluir la carpeta con las imagenes del software al empaquetado del proyecto.

```
--add-data pesos_red_yolo;pesos_red_yolo
```

Para incluir la carpeta que contiene el modelo entrenado con visión artificial para la detección de movimientos sospechosos al empaquetado del proyecto.

Una vez concluye el proceso, se genera el empaquetado de la solución en el directorio "dist" del proyecto:

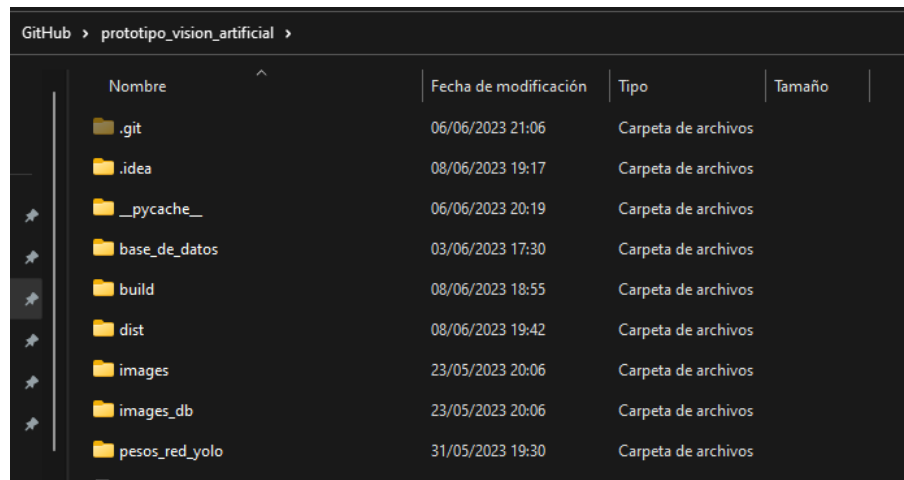


Ilustración 137 Carpeta estructura proyecto

Dentro de esta carpeta se encuentra la subcarpeta main y dentro de esta se encuentra el ejecutable en el archivo main.exe:

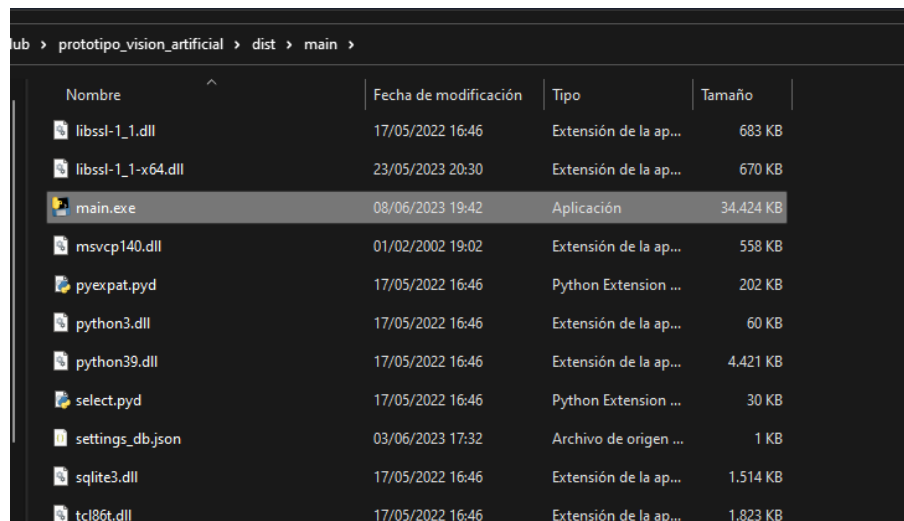
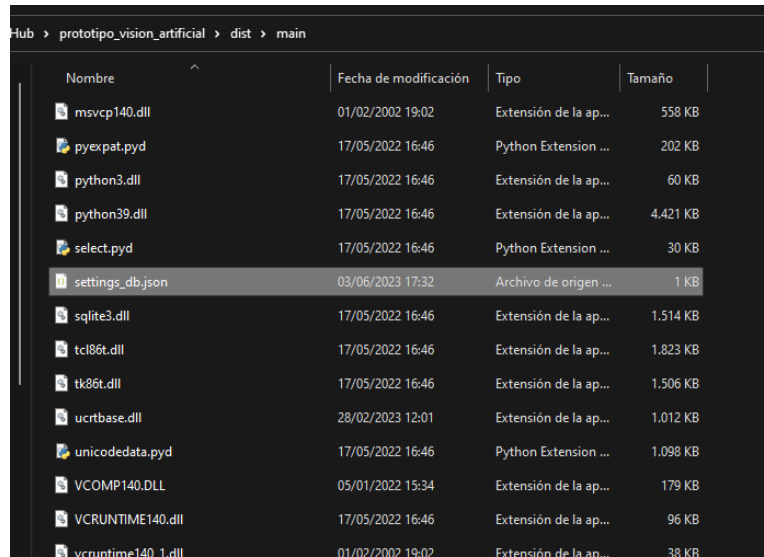


Ilustración 138 Ubicación archivo ejecutable

Adicionalmente vale destacar, que en la misma ubicación se encuentra el archivo `settings_db.json` en el cual se establecen las credenciales de base de datos, servicios y rutas del proyecto:



The screenshot shows a file explorer window with the following table of files:

Nombre	Fecha de modificación	Tipo	Tamaño
msvcp140.dll	01/02/2002 19:02	Extensión de la ap...	558 KB
pyexpat.pyd	17/05/2022 16:46	Python Extension ...	202 KB
python3.dll	17/05/2022 16:46	Extensión de la ap...	60 KB
python39.dll	17/05/2022 16:46	Extensión de la ap...	4.421 KB
select.pyd	17/05/2022 16:46	Python Extension ...	30 KB
settings_db.json	03/06/2023 17:32	Archivo de origen ...	1 KB
sqlite3.dll	17/05/2022 16:46	Extensión de la ap...	1.514 KB
tcl86t.dll	17/05/2022 16:46	Extensión de la ap...	1.823 KB
tk86t.dll	17/05/2022 16:46	Extensión de la ap...	1.506 KB
ucrtbase.dll	28/02/2023 12:01	Extensión de la ap...	1.012 KB
unicodedata.pyd	17/05/2022 16:46	Python Extension ...	1.098 KB
VCOMP140.DLL	05/01/2022 15:34	Extensión de la ap...	179 KB
VCRUNTIME140.dll	17/05/2022 16:46	Extensión de la ap...	96 KB
vcruntime140_1.dll	01/02/2002 19:02	Extensión de la ap...	38 KB

Ilustración 139 Ubicación archivo configuración proyecto ejecutable

Conclusión

Este manual de programador representa una guía para otro desarrollador al momento de instalar y configurar el entorno de trabajo, se ha registrado todo el proceso de forma que sea de utilidad y pueda simular todo el proceso seguido.

Cada uno de los softwares instalados contiene los pasos para un aprendizaje rápido la configuración puede variar según el equipo y rutas de instalación más sin embargo se ha intentado plasmar la idea central de su instalación y configuración.

En el desarrollo del proyecto se describieron los archivos y funciones internas que pueden orientar al lector al revisar el código su funcionamiento y respuesta a integrar nuevas funciones y mejoras.

Como todo proceso de software siempre hay puntos de mejoras, se espera le sea de utilidad y fácil de comprender.

MANUAL DE PROGRAMADOR

APLICATIVO WEB

ESCUELA ESPECIALIZADA EN INGENIERÍA ITCA-FEPADE

ESCUELA DE COMPUTACIÓN



MANUAL DE PROGRAMADOR

Prototipo de sistema de videovigilancia automatizado por medio
de visión artificial para ITCA-FEPADE

PRESENTADO POR

Edgardo Antonio Argueta Bautista

Jefferson Francisco Pineda Quintanilla

Kevin Rodrigo Flores Acosta

Nicolás Gabriel Escobar Pérez

Versión 1.0

Santa Tecla, La Libertad, El Salvador.

INDICE

INTRODUCCIÓN	219
ARQUITECTURA DEL SISTEMA	220
REQUISITOS TÉCNICOS	221
REQUISITOS GENERALES	221
INSTALACIÓN Y CONFIGURACIÓN	222
ESTRUCTURA DEL PROYECTO	224
API Y SERVICIOS	227
DIAGRAMAS	228
FUNCIONALIDADES Y CARACTERÍSTICAS	228
GUÍA DE DESARROLLO	230
CONSIDERACIONES DE SEGURIDAD	234
FLUJO DE TRABAJO Y CONTROL DE VERSIONES	234
NOTAS DE LANZAMIENTO	235
NOTA LANZAMIENTO - VERSIÓN 1.0.0	235
REFERENCIAS Y RECURSOS ADICIONALES	236

Introducción

Hoy el buen uso de la tecnología es muy importante a nivel educativo, como profesional, muchas de ellas nos dan facilidad a desarrollar productos digitales, que mejoran procesos, pero aplicarlos requiere de un conocimiento amplio.

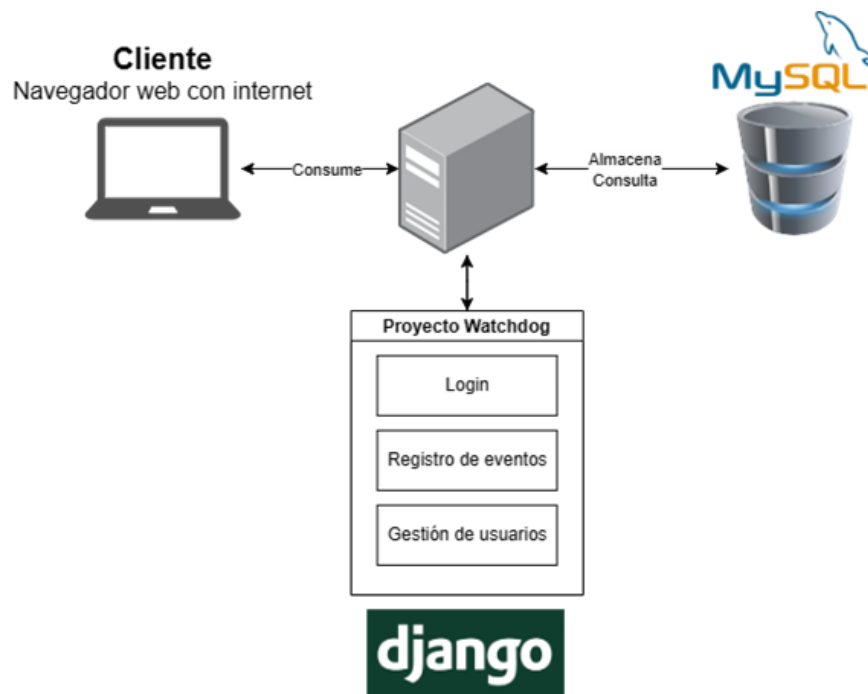
El siguiente documento, tiene como finalidad el proveer una guía para los desarrolladores/programadores que se involucren posteriormente a la hora de manipular el prototipo de visión artificial, específicamente al sistema administrativo llamado como: **"Watchdog"**. Se explicará a nivel técnico los diferentes apartados que este posee, así como sus funcionalidades, usos de tecnologías e implementación de estos.

El sistema tiene como objetivo el poder recolectar los diferentes "eventos sospechosos" y mostrarlos con mejor detalle, además de la gestión de usuarios de la plataforma.

Arquitectura del sistema

A continuación, se muestra un diagrama el cual detalla la arquitectura general del sistema de "Watchdog", en donde hay 3 apartados importantes y que intervienen en su uso.

- 1. Base de datos:** Almacena la información capturada de los eventos sospechosos, así como la información de los usuarios registrados de ambos sistemas.
- 2. Aplicativo Django:** Creado con Python junto de la mano del Framework de Django que permite realizar aplicaciones webs. A la vez es alojada en un servidor, donde se puede acceder mediante un dominio.
- 3. Cliente/Laptop:** Los usuarios desde sus dispositivos como laptops o máquinas de escritorio pueden acceder al sistema web mediante la url del servidor donde se aloje el aplicativo web.



Requisitos técnicos

Requisitos generales

- Python: una versión estable; para el desarrollo se utilizó la versión 3.11.3. Nota: Agregar Python al PATH de nuestro sistema operativo para el buen funcionamiento.

```
C:\Users\krfac>python --version
Python 3.11.3
```

- Django: Se instala posterior a la instalación de Python. Ejecutando el siguiente comando:

```
C:\Users\krfac>pip install django
```

- Gestor de base de datos MySQL, instalado ya sea en nuestra máquina (XAMPP), o un servidor de base de datos dedicado para el almacenamiento de los datos, con sus credenciales de accesos.

Entorno

En este entorno podemos correr el aplicativo en la misma computadora, los requerimientos mínimos para levantar el proyecto en el entorno son:

- Sistema operativo: Windows o Linux (Según preferencia)
- RAM: 4 GB mínimo
- Procesador: Intel(R) Core i5 de quinta generación
- Acceso a internet

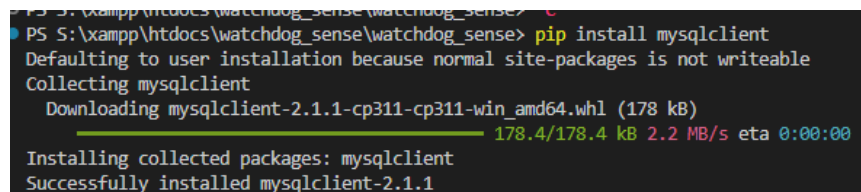
Instalación y configuración

Para la instalación y configuración del proyecto es necesario conocer que el proyecto inicialmente se aloja dentro de [Github.com](https://github.com), lo cual es necesario los permisos para acceder.

Una vez poseemos los accesos a GitHub, podemos hacer uso de [Git](https://git-scm.com/), o de cualquier herramienta que posea una interfaz gráfica. Para obtener el proyecto dentro de la computadora y poder iniciar la configuración, preceder a abrir el editor de código o IDE. En este caso, para el desarrollo se utilizó [Visual Studio Code](https://code.visualstudio.com/), desde este momento conocido como VSC.

Una vez obtenido el proyecto, se debe abrir en el IDE, y se procede a abrir la terminal de VSC. Es necesario ejecutar ciertos comandos para el uso de librerías dentro del proyecto, los cuales se detallan a continuación:

1. Instalación de driver para MySQL



```
PS 5: \xampp\htdocs\watchdog_sense\watchdog_sense> pip install mysqlclient
Defaulting to user installation because normal site-packages is not writeable
Collecting mysqlclient
  Downloading mysqlclient-2.1.1-cp311-cp311-win_amd64.whl (178 kB)
    178.4/178.4 kB 2.2 MB/s eta 0:00:00
Installing collected packages: mysqlclient
Successfully installed mysqlclient-2.1.1
```

2. Verificación de credenciales en settings.py (Se encuentra dentro del proyecto, en su raíz. En este apartado se debe ingresar las credenciales de acceso, que se nos provean o

que hayamos montado en XAMPP o servidor destinado.

```
3 DATABASES = {
4     'default': {
5         'ENGINE': 'django.db.backends.sqlite3',
6         'NAME': BASE_DIR / 'db.sqlite3'
7         # "USER":
8         # "PASSWORD"
9     },
10    'db_object_detection': {
11        'ENGINE': 'django.db.backends.mysql',
12        'NAME': 'db_object_detection',
13        'USER': 'root',
14        'PASSWORD': '',
15        'HOST': 'localhost',
16        'PORT': '3306',
17    }
18 }
```

3. Luego proceder a ejecutar el siguiente comando, el cual recolectará todos los recursos que necesita el proyecto. Estos recursos contienen los archivos CSS, JS, Imágenes, vídeos, etc.

```
PS C:\xampp\htdocs\watchdog_sense\watchdog_sense> python manage.py collectstatic
You have requested to collect static files at the destination
location as specified in your settings:

C:\xampp\htdocs\watchdog_sense\watchdog_sense\staticfiles

This will overwrite existing files!
Are you sure you want to do this?

Type 'yes' to continue, or 'no' to cancel: yes
```

4. Ahora ya tenemos todo listo, proceder a levantar el proyecto, ejecutando el siguiente comando:

```
PS C:\xampp\htdocs\watchdog_sense\watchdog_sense> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 11, 2023 - 11:49:38
Django version 4.2.1, using settings 'watchdog_sense.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Si todo salió en orden, podremos abrir el navegador con la url que la misma consola provee (Ejemplo: `http:127.0.0.1:8000`), y podrá ya empezar a utilizarlo.



WATCHDOG

Inicio de sesión

Nombre de usuario

krfaaf

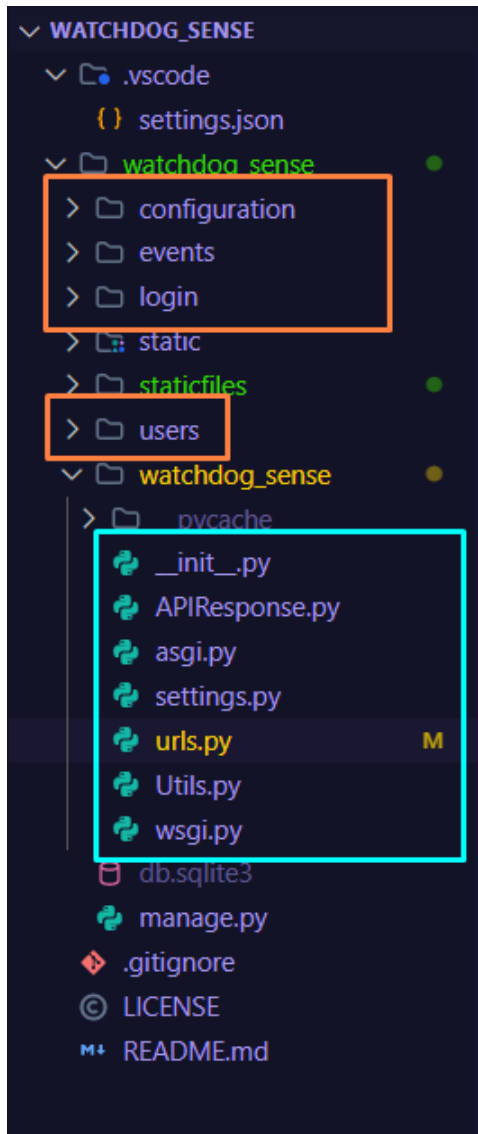
Contraseña

.....

Ingresar

Estructura del proyecto

La estructura de todo de el proyecto se basa en el Framework Django, este nos permite tener un proyecto, con muchas "apps", cada una de ellas funcionan como un módulo dentro del aplicativo y con características específicas.



Estructura general del proyecto

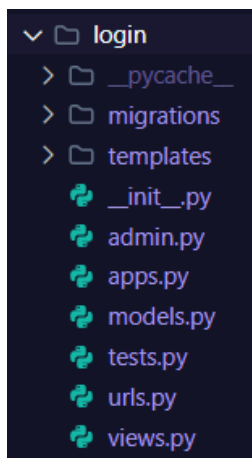
1. Naranja: Apps que contienen las diferentes vistas del aplicativo.
2. Celeste: Archivos de configuración general del aplicativo

Archivos de configuración:

1. APIResponse.py: Clase que gestiona las diferentes respuestas HTTP de nuestro aplicativo.
2. *Archivo general de configuración*
3. settings.py: Permite controlar muchos aspectos generales de configuración, como: Credenciales, rutas estáticas, Librerías, Lenguaje, entre otros.
4. urls.py: Permite controlar las rutas de acceso al proyecto.
5. Utils: Clase que permite gestionar funciones generales de uso dentro del aplicativo.
6. *Archivo general de configuración.*

Estructura de Apps

Dentro de cada App, se encuentran diferentes archivos también, que hace a cada app distinta, la estructura de cada una es la siguiente:



Tomaremos la app de "Login" para mostrar ejemplo:

1. `__pycache__`: Se crea automáticamente al crear una app.
2. `migrations`: Contiene las diferentes cláusulas que se ejecutan a nivel de base de datos para la gestión de estas.
3. `templates`: Contiene las diferentes vistas en formato Django HTML.
4. `admin.py`: Se crea automáticamente al crear una app.
5. `apps.py`: Se crea automáticamente al crear una app.
6. `models.py`: Permite poder definir clases basadas en tablas de la base de datos.
7. `tests.py`: Permite realizar test del código antes de implementarlo en el.
8. `urls`: Define las diferentes rutas de acceso a la información de la App, por ejemplo: crear, modificar, eliminar. Estos pueden contener vistas o solamente retornar información para su uso.
9. `views`: Contiene las diferentes funciones que se definen previamente en el "urls.py", donde ya va la lógica de aplicativo y acceso de datos.

API y servicios

Como se mencionó anteriormente, se utiliza una clase la cual nos permite el hecho de poder retornar información mediante peticiones HTTP; esta clase, permite controlar en un solo sitio las diferentes respuestas, a continuación, se muestran la estructura:

Clase APIResponse

Contiene una colección de funciones que permiten llamar desde los archivos "views", con la finalidad de que estos los utilicen y se envíe la información a las diferentes vistas, y así sean interpretados o incluso validados en diferentes ámbitos de la interfaz gráfica del sistema.

```
from django.http import JsonResponse

# Clase creada para la respuesta dinámica de nuestras peticiones

class APIResponse:
    @staticmethod
    def success(message='', data=None, records=None, status=200):
        response_data = {
            'status': "SUCCESS",
            'message': message,
            'data': data,
            'records': records
        }
        return JsonResponse(response_data, status=status)

    @staticmethod
    def error(message='', data=None, status=400):
        response_data = {
            'status': "ERROR",
            'message': message,
            'data': data
        }
        return JsonResponse(response_data, status=status)

    @staticmethod
    def nodata(data=None):
        response_data = {
            'status': "NO_DATA",
            'data': data
        }
        return JsonResponse(response_data)

    @staticmethod
    def exist(message='', data=None, records=None, status=200):
        response_data = {
            'status': "EXIST",
            'message': message,
```

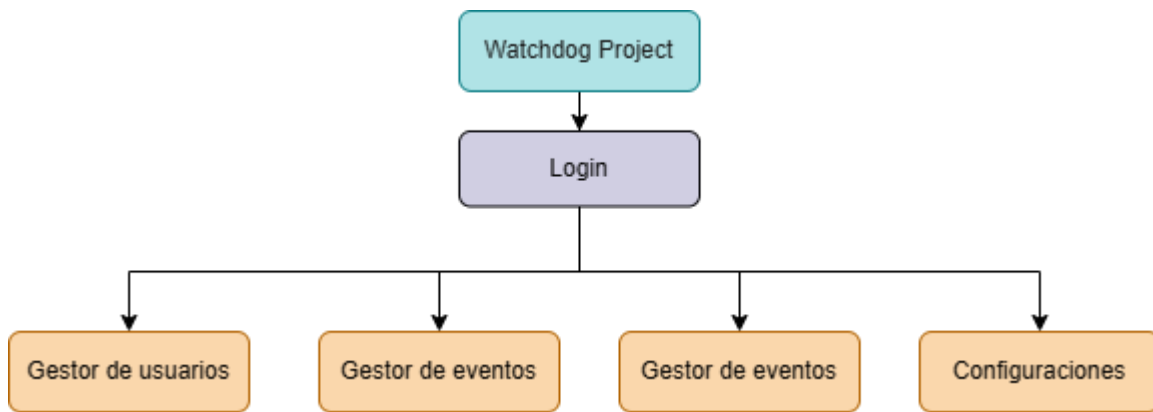
El uso depende de la acción que se este realizando y la respuesta que se requiere gestionar ya en la vista:

```
if enc_pass == Pass:
    return APIResponse.success(data=userData, message='Logged', records=len(userData))
else:
    return APIResponse.nodata()
```

Diagramas

Flujos de pantallas

A continuación, se muestra el flujo de pantallas que contiene el proyecto.



Funcionalidades y características

Las funcionalidades del sistema son las siguientes:

1. Login: permite acceder a los usuarios dentro del sistema, es necesario que posean credenciales válidas para poder acceder.
2. Gestor de eventos: permite ver los eventos que se han detectado mediante el prototipo del sistema de videovigilancia

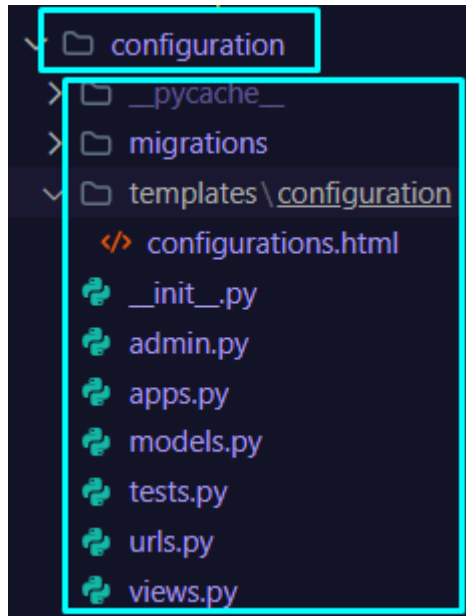
3. Gestor de usuarios: Permite gestionar los diferentes usuarios que utilizarán en el sistema, los cuales son:
 - a. Administrado: Gestiona eventos, configuraciones generales y usuarios
 - b. Operador: Gestiona eventos sucedidos
4. Configuraciones generales: Permite controlar variables o configuraciones que utiliza tanto el sistema de visión artificial, como el administrativo.

Guía de desarrollo

Si es necesaria la implementación de una App, es necesario conocer estos pasos, ya que son vitales para la buena configuración de este.

1. Ejecutar comando para la creación de la nueva App, esto permite crear la carpeta de la nueva app, con todas las configuraciones bases para que sea reconocido por el proyecto Django.

```
PS C:\xampp\htdocs\watchdog_sense\watchdog_sense> py manage.py startapp configuration
```

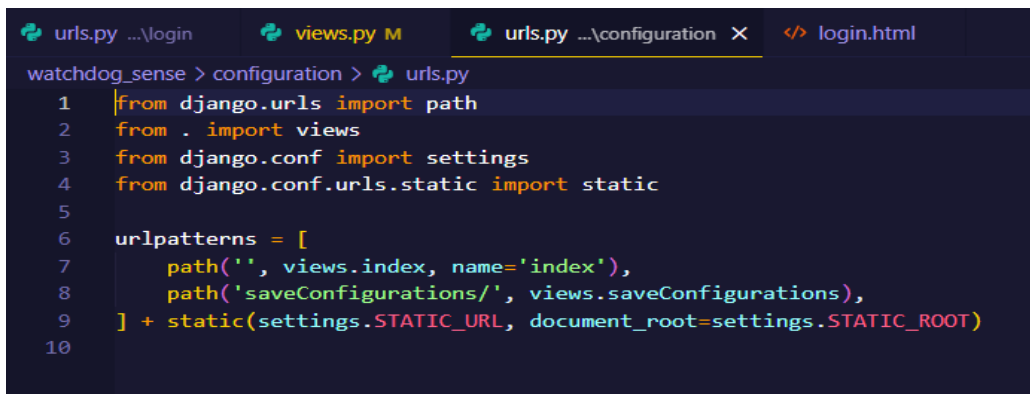


2. Crear template del proyecto Django HTML: Acceder a la carpeta del App, y crear el directorio "templates". Luego dentro de "templates", se crea otro directorio con el mismo nombre de la app, llamado "configuration". Esto puede ser un poco confuso, pero es la manera en la que Django interpreta y sabe dónde se alojan los templates. Luego

dentro de "configuration", creamos el HTML, con el código general que se usa en HTML y su estructura.

3. Crear archivo urls.py: En este archivo es necesario ya que, por medio de este archivo, se podrá acceder a vistas dentro de la App, como, por ejemplo, el archivo HTML, y mostrarlo, así como el acceso a acciones específicas, como: guardar un registro en la base de datos; entonces no es únicamente para mostrar vistas.

La estructura es la siguiente:



```
watchdog_sense > configuration > urls.py
1 | from django.urls import path
2 | from . import views
3 | from django.conf import settings
4 | from django.conf.urls.static import static
5
6 | urlpatterns = [
7 |     path('', views.index, name='index'),
8 |     path('saveConfigurations/', views.saveConfigurations),
9 | ] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)
10
```

Se define el arreglo "urlpatterns" y se detalla mediante una función "path" la url a la cual se debe acceder, se nota que al inicio solamente se ve que la primera tiene comillas simples, esto quiere decir que se accede por defecto a esa pantalla al entrar al módulo, luego se accede al archivo views que será el "controlador", o contenedor de funciones dentro de la página.

Y al final se anexa el static con otras configuraciones para el acceso a recursos.

4. Configuración del archivo views.py: Siguiendo de guía el paso anterior, es necesario definir en este archivo las funciones que se utilizan. En este caso, se usa una función index, el cual se accede por la ruta "configuration/", por

defecto, y se tiene la ruta "configuration/saveConfigurations".

Entonces cada función que se implementa en el url.py, debe estar reflejada en la views.py, de la siguiente manera:

```
def index(request):
    try:
        with connections['db_object_detection'].cursor() as cursor:
            cursor.execute("select cs.id_image_setting, cs.number_images, cs
            configs = cursor.fetchall()

            if len(configs) > 0:
                # Se encontraron registros
                config = configs[0]
                #print(user)

                idConfig = config[0]
                numberImages = config[1]
                confidenceScore = config[2]
                absoluteRoute = config[3]

                configData = {
```

5. Luego como último paso, ir al archivo general del proyecto "settings.py", para poder realizar una configuración, ya que, sin esto, todo lo que se hizo anteriormente, no funcionará.

Es por eso que es necesario:

- a. Acceder a al archivo apps.py en la App "configuration", copiar el nombre de la primera función; la cual se crea automáticamente al crear una App.

```
41 'django.contrib.contenttypes'
settings.py M views.py apps.py ...configuration X wsgi.py
watchdog_sense > configuration > apps.py
1 from django.apps import AppConfig
2
3
4 class ConfigurationConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'configuration'
7
```

b. Agregar la ruta de nuestra app, en el arreglo de aplicaciones que utiliza nuestro sistema, así como se ve en la imagen:

El nombre de la función que se copio es la que debe aparecer al final de la ruta.

```
# Application definition

INSTALLED_APPS = [
    'configuration.apps.ConfigurationConfig',
    'users.apps.UsersConfig',
    'events.apps.EventsConfig',
    'login.apps.LoginConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

6. Una vez agregada la url al arreglo, ya se podrá ver cómo funciona nuestra nueva página.

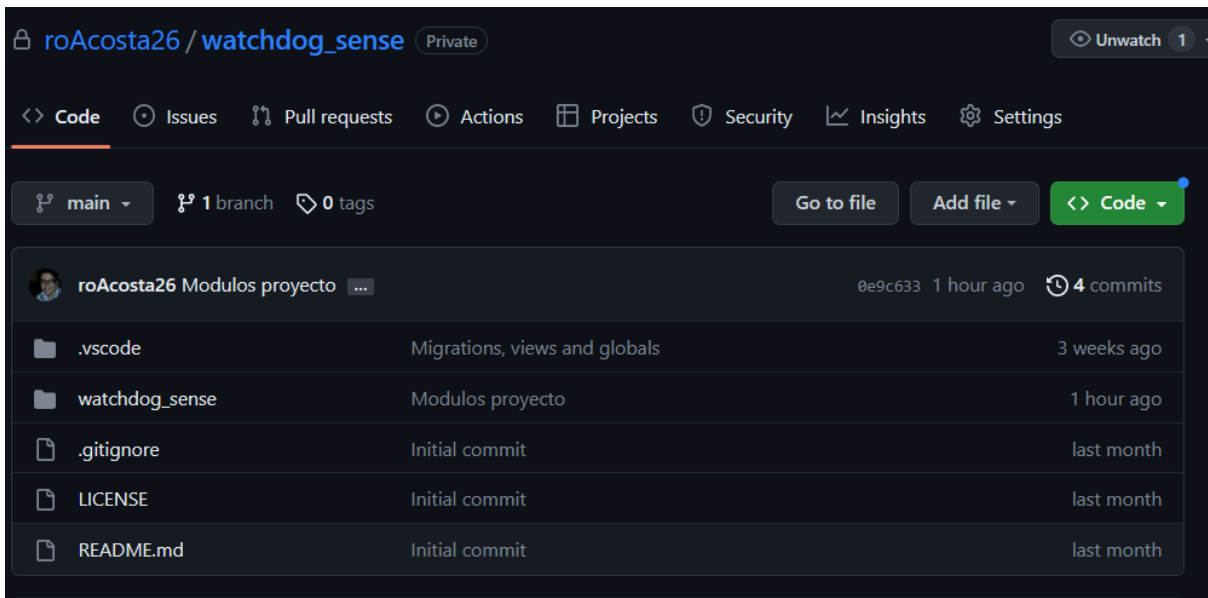
Consideraciones de seguridad

Actualmente se utilizan cookies, para el guardado de sesión, el cuál durará un día dentro del navegador; luego de pasado ese tiempo, se borrará automáticamente la sesión y se tendrá que iniciar sesión nuevamente.

Se recomienda usar metodologías como el guardado de sesiones en base de datos y tokens de validación para una mejor robustez dentro del aplicativo.

Flujo de trabajo y control de versiones

Como se mencionaba al inicio del documento, se utiliza Github como repositorio del proyecto, por lo cual es necesario un acceso.



Para el control de versionamiento, se utiliza Github Desktop, el cual permite realizar commits del mismo. Cada usuario o persona que necesite realizar un cambio es necesario que posea un usuario con credenciales, para poder interferir en el desarrollo de este.

Notas de lanzamiento

Nota lanzamiento - Versión 1.0.0

Fecha de lanzamiento: 6 de junio de 2023

Resumen de cambios:

- Creación de proyecto "Watchdog"
- Implementación de línea gráfica
- Funcionalidades
 - Log in
 - Gestor de usuarios
 - Registro de eventos
 - Configuraciones de sistema

Mejoras:

- Validaciones extras
- Mejoras en interfaz de componentes
- Guardado de configuraciones

Referencias y recursos adicionales

Django: [Download Django | Django \(djangoproject.com\)](https://www.djangoproject.com/)

GIT: [Git \(git-scm.com\)](https://git-scm.com/)

Python: [Download Python | Python.org](https://www.python.org/)

Repositorio: [roAcosta26/watchdog_sense \(github.com\)](https://github.com/roAcosta26/watchdog_sense)

MANUAL DE USUARIO

APLICATIVO SERVIDOR

ESCUELA ESPECIALIZADA EN INGENIERIA ITCA-FEPADE

ESCUELA DE COMPUTACIÓN



MANUAL DE USUARIO

Prototipo de sistema de videovigilancia automatizado por medio de visión artificial para ITCA-FEPADE.

PRESENTADO POR:

Edgardo Antonio Argueta Bautista

Jefferson Francisco Pineda Quintanilla

Kevin Rodrigo Flores Acosta

Nicolás Gabriel Escobar Pérez

Versión 1.0

Santa Tecla, La Libertad, El Salvador

ÍNDICE

OBJETIVO	244
ALCANCE DEL DOCUMENTO	245
REQUERIMIENTOS TÉCNICOS DEL SISTEMA	246
REQUISITOS DE HARDWARE	246
REQUISITOS DE SOFTWARE	247
OTROS REQUISITOS	247
INSTALACIÓN DE PYTHON	248
INSTALACIÓN DE MYSQL	251
INSTALACIÓN DEL SOFTWARE	259
USO DEL SOFTWARE	261
NAVEGACIÓN DEL PROTOTIPO	261
MENÚ DE NAVEGACIÓN	262
CONTENIDO DEL SOFTWARE	263
REGISTRO	263
INICIO DE SESIÓN	266
INICIAR/PARAR VIDEOVIGILANCIA	267
CIERRE DE SESIÓN	269
OTRAS FUNCIONALIDADES	269
ENVIÓ DE MENSAJES DE TEXTO (SMS)	271
ALERTA SONORA	271
GLOSARIO	272

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 140	DESCARGA DE PYTHON 3.9.6.....	248
ILUSTRACIÓN 141	ADICIÓN DE VARIABLES DE ENTORNO DESDE EL INSTALADOR DE PYTHON	249
ILUSTRACIÓN 142	INSTALANDO PYTHON.....	249
ILUSTRACIÓN 143	PYTHON INSTALADO CORRECTAMENTE.....	250
ILUSTRACIÓN 144	VERIFICANDO VERSIÓN DE PYTHON.....	251
ILUSTRACIÓN 145	DESCARGA DE MYSQL.....	251
ILUSTRACIÓN 146	INSTALACIÓN PERSONALIZADA MYSQL.....	252
ILUSTRACIÓN 147	OPCIONES A INSTALAR.....	252
ILUSTRACIÓN 148	INSTALACIÓN DE OPCIONES MYSQL.....	253
ILUSTRACIÓN 149	ELEGIR TIPO DE CONFIGURACIÓN DE DESARROLLO.....	253
ILUSTRACIÓN 150	MÉTODO DE AUTENTICACIÓN POR DEFECTO.....	254
ILUSTRACIÓN 151	CONFIGURACIÓN DE CREDENCIALES DE ACCESO.....	254
ILUSTRACIÓN 152	SERVICIO DE WINDOWS MYSQL.....	255
ILUSTRACIÓN 153	PERMISOS DE ARCHIVOS DEL SERVIDOR.....	255
ILUSTRACIÓN 154	APLICANDO CONFIGURACIÓN MYSQL.....	256
ILUSTRACIÓN 155	CONFIGURACIÓN DE MYSQL.....	256
ILUSTRACIÓN 156	FINALIZAR INSTALACIÓN.....	257
ILUSTRACIÓN 157	INSTALACIÓN COMPLETADA.....	257
ILUSTRACIÓN 158	CONFIGURACIÓN EN MYSQL WORKBENCH.....	258
ILUSTRACIÓN 159	EJECUTANDO BASE DE DATOS.....	258
ILUSTRACIÓN 160	ARCHIVO CONFIGURACIÓN SERVIDOR.....	259
ILUSTRACIÓN 161	MENÚ DE NAVEGACIÓN.....	262
ILUSTRACIÓN 162	MENÚ DE NAVEGACIÓN EXPANDIDO.....	262
ILUSTRACIÓN 163	CONTENIDO DEL SOFTWARE.....	263
ILUSTRACIÓN 164	BOTÓN REGISTRO.....	264
ILUSTRACIÓN 165	FORMULARIO REGISTRO.....	264
ILUSTRACIÓN 166	ALERTAS REGISTRO.....	265
ILUSTRACIÓN 167	CAPTURA DE ROSTRO REGISTRO.....	265
ILUSTRACIÓN 168	BOTÓN LOGIN.....	266

ILUSTRACIÓN 169	CAPTURA DE ROSTRO LOGIN.....	267
ILUSTRACIÓN 170	BOTÓN CAPTURA	268
ILUSTRACIÓN 171	INICIO CAPTURA EVENTOS	268
ILUSTRACIÓN 172	BOTÓN SALIR.....	269
ILUSTRACIÓN 173	CORREO DE ALERTA	270
ILUSTRACIÓN 174	MENSAJE DE ALERTA	271

ÍNDICE DE TABLAS

TABLA 1 REQUISITOS DE HARDWARE Y SOFTWARE

246

Introducción

El presente documento busca orientar a los usuarios que se encarguen de instalar y configurar el prototipo del sistema de visión artificial para su correcto funcionamiento en la máquina que se destine como servidor.

Se detallarán los requerimientos técnicos para poder ejecutar el software tanto a nivel de hardware como a nivel de paquetería necesaria.

Objetivo

Brindar a los usuarios una guía que les permita configurar el servidor del prototipo del sistema de visión artificial en cualquier máquina que cumpla con los requisitos y especificaciones necesarias.

Alcance del documento

Este documento está dirigido a los administradores del prototipo de sistema de videovigilancia automatizado por medio de visión artificial para ITCA-FEPADE con el fin de ofrecer una guía y consultas sobre la configuración y despliegue del sistema.

Requerimientos técnicos del sistema

Requisitos de hardware

Para el despliegue del prototipo se requiere un equipo con las siguientes características de hardware y software o superior.

Equipo	Especificación	Capacidad
Laptop	Procesador	Intel® Core™i7-6700 HQ CPU @ 2.60 GHz 2.59 GHz o superior
	RAM	16.0 GB o superior
	Tipo de sistema	Sistema operativo Windows 10 de 64 bits, Procesador x64 o superior
	Tarjeta grafica	NVIDIA GeForce GTX 965M o superior
	Memoria de video dedicada	4096 MB GDDR 5 o superior
	Unidad de almacenamiento	Disco de Estado Sólido SSD 128 GB o superior
	Cámara web	1mp o superior
Cámara IP	Lente	5mp o superior
	Resolución	1280px*720px o superior
	Protocolo de transmisión	RTSP

Tabla 27 Requisitos de hardware y software

Requisitos de software

Para poder ejecutar el prototipo se necesita poseer instalados en el equipo las siguientes herramientas.

Base de datos.

Base de datos utilizada MySQL Ver 15.1 o superior, para Win64 (AMD64).

Interprete.

Python 3.9.6 o superior

Otros requisitos

Para aprovechar al máximo las posibilidades del servidor se necesita una cuenta del proveedor Twilio, el cual permite realizar el envío de mensajes de texto, los cuales se utilizan para alertar a los usuarios de nuevos eventos.

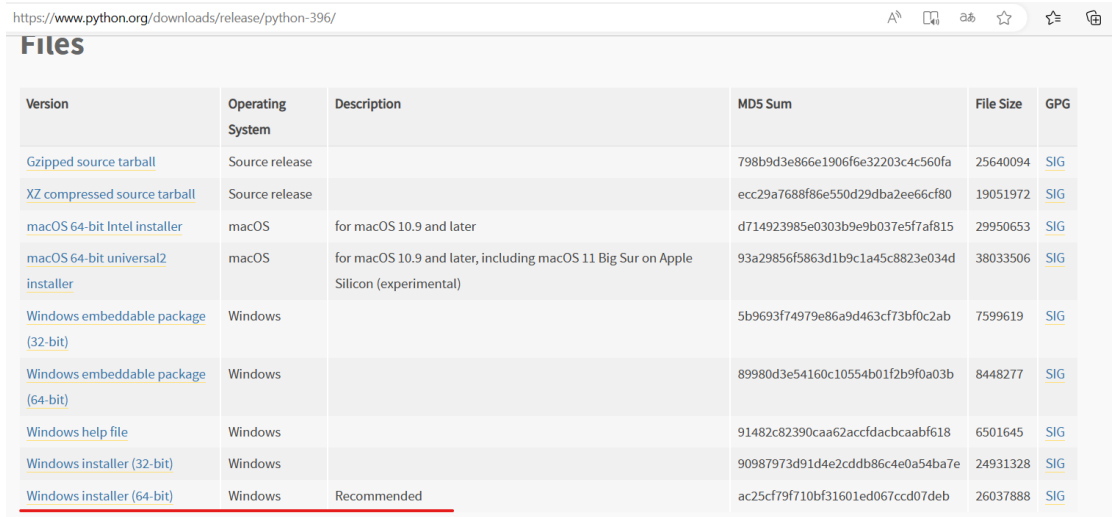
Instalación de Python

Para instalar Python en su versión 3.9.6, seguimos los siguientes pasos:

Pasos

7. Descarga e instalación, descargar desde el enlace.

<https://www.python.org/downloads/release/python-396/>



The screenshot shows a web browser window with the URL <https://www.python.org/downloads/release/python-396/>. The page title is "Files". Below the title is a table with the following columns: Version, Operating System, Description, MD5 Sum, File Size, and GPG. The table lists various download options for Python 3.9.6, including source releases, macOS installers, and Windows installers. The "Windows installer (64-bit)" row is highlighted with a red underline.

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		798b9d3e866e1906f6e32203c4c560fa	25640094	SIG
XZ compressed source tarball	Source release		ecc29a7688f86e550d29dba2ee66cf80	19051972	SIG
macOS 64-bit Intel installer	macOS	for macOS 10.9 and later	d714923985e0303b9e9b037e5f7af815	29950653	SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later, including macOS 11 Big Sur on Apple Silicon (experimental)	93a29856f5863d1b9c1a45c8823e034d	38033506	SIG
Windows embeddable package (32-bit)	Windows		5b9693f74979e86a9d463cf73bf0c2ab	7599619	SIG
Windows embeddable package (64-bit)	Windows		89980d3e54160c10554b01f2b9f0a03b	8448277	SIG
Windows help file	Windows		91482c82390caa62accfdacbaabf618	6501645	SIG
Windows installer (32-bit)	Windows		90987973d91d4e2cddb86c4e0a54ba7e	24931328	SIG
Windows installer (64-bit)	Windows	Recommended	ac25cf79f710bf31601ed067ccd07deb	26037888	SIG

Ilustración 140 Descarga de Python 3.9.6

8. Una vez descargado proceder a su instalación, marcar las opciones de Add Python 3.9 to PATH para que agregue las variables de entorno por defecto y damos click en Instalar ahora.

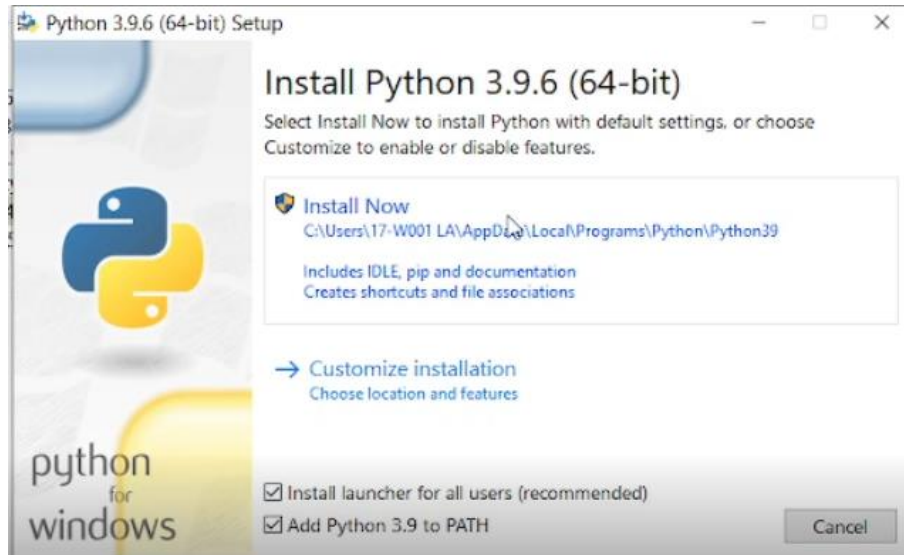


Ilustración 141 Adición de variables de entorno desde el instalador de Python

9. Iniciar la instalación

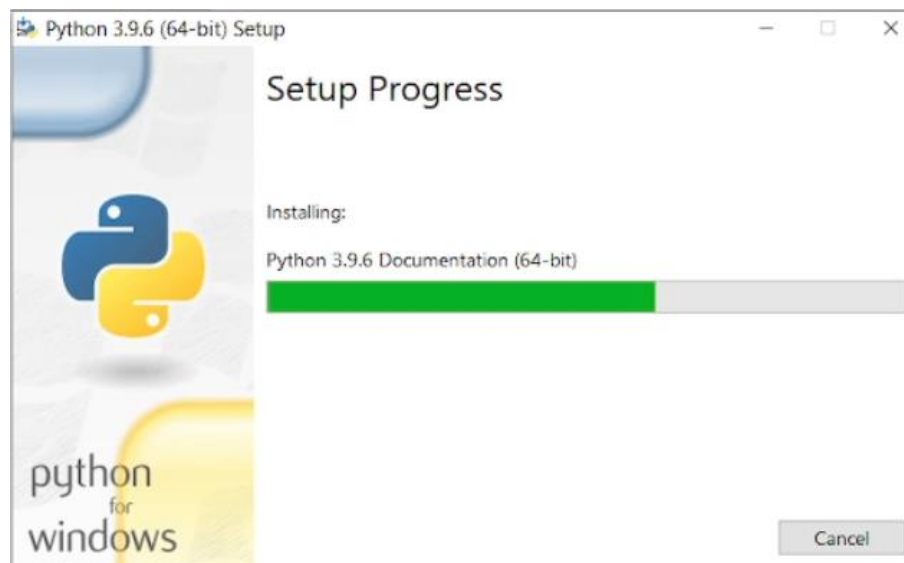


Ilustración 142 Instalando Python

10. Aparecerá en el modal que fue configurado exitosamente proceder a cerrar.

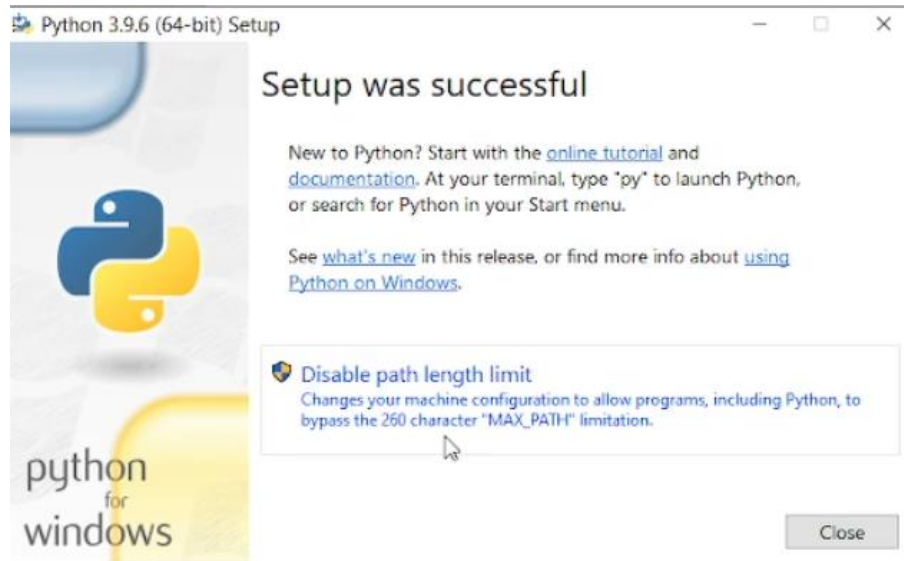


Ilustración 143 Python instalado correctamente

11. Verificar las variables de entorno creadas, dentro de PATH debemos de verificar que estén las siguientes, si no se encuentran se debera agregarlas, recuerda que la ruta depende de donde se realizó la instalación.

C:\Users\PC01\AppData\Local\Programs\Python\Python39

C:\Users\PC01\AppData\Local\Programs\Python\Python39\Scripts

12. Verificar la instalación desde el CMD de Windows, al crear las variables de entorno desde el instalador no debería generar conflictos y dar la versión 3.9.6

```
C:\> Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.2728]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\17-W001 LA>python --version
Python 3.9.6
C:\Users\17-W001 LA>
```

Ilustración 144 Verificando versión de Python

Instalación de MySQL

MySQL es el gestor utilizado para base de datos, nos servirá para gestionar información útil sobre eventos detectados y configuraciones del sistema.

Pasos

16. Descargar desde el siguiente enlace

<https://downloads.mysql.com/archives/installer/>

MySQL Product Archives

MySQL Installer (Archived Versions)

⚠ Please note that these are old versions. New releases will have recent bug fixes and features!
To download the latest release of MySQL Installer, please visit [MySQL Downloads](#).

Product Version:
Operating System:

Windows (x86, 32-bit), MSI Installer <small>(mysql-installer-web-community-8.0.32.0.msi)</small>	Jan 3, 2023	2.4M	Download
Windows (x86, 32-bit), MSI Installer <small>(mysql-installer-community-8.0.32.0.msi)</small>	Jan 3, 2023	437.3M	Download

Ilustración 145 Descarga de MySQL

17. Una vez descargado ejecutar el instalador, aparecerá el siguiente modal, marcar opción personalizada (Custom) y presionar siguiente.

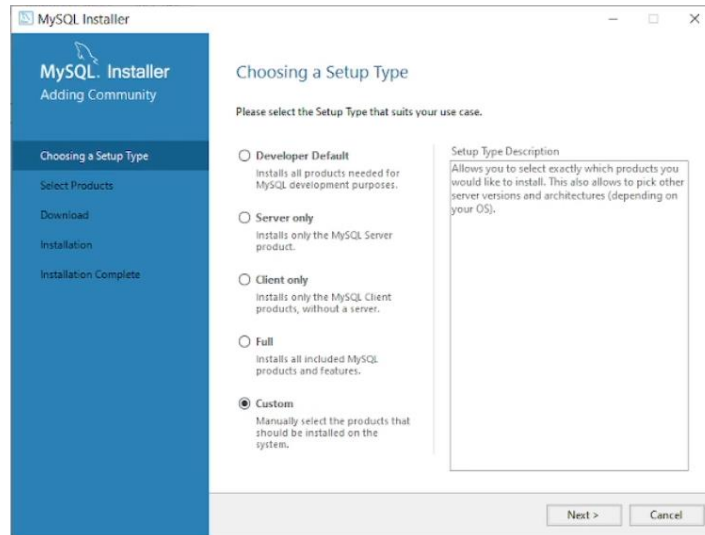


Ilustración 146 Instalación personalizada MySQL

18. Seleccionar los productos a instalar, en el recuadro mover los productos que serán instalados presionar siguiente.

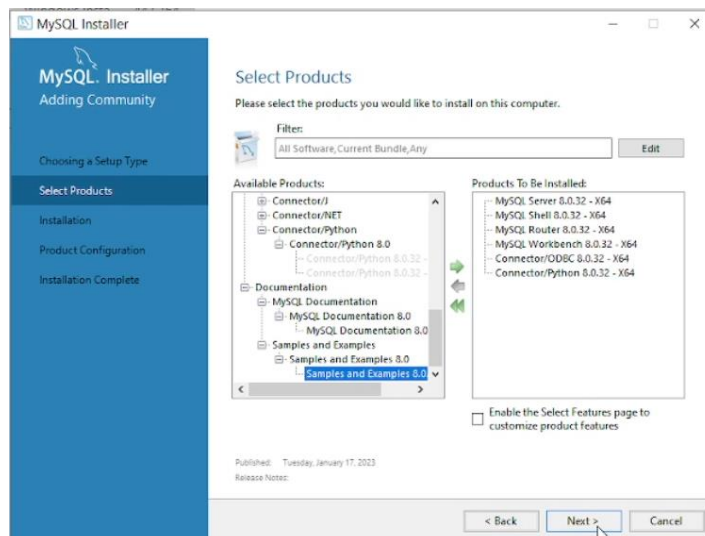


Ilustración 147 Opciones a instalar

19. Se instalarán los productos seleccionados al presionar ejecutar, luego presionar siguiente.

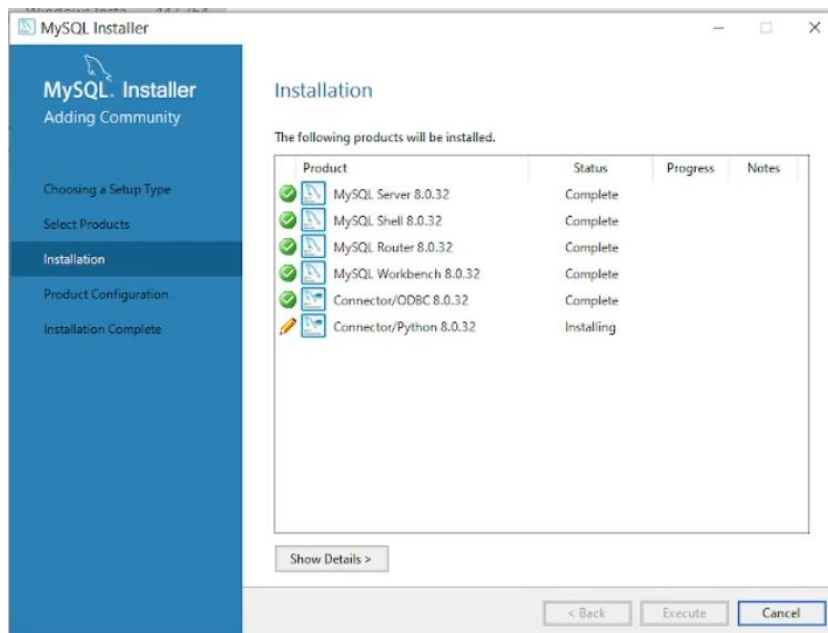


Ilustración 148 Instalación de opciones MySQL

20. Aparecerá el siguiente modal presionamos siguiente

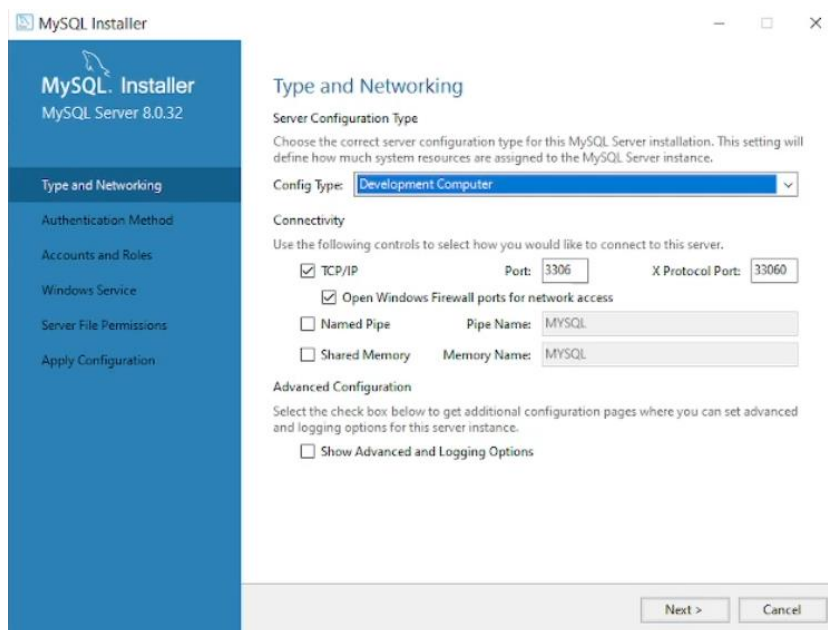


Ilustración 149 Elegir tipo de configuración de desarrollo

21. Aparecerá el modal siguiente presionar siguiente

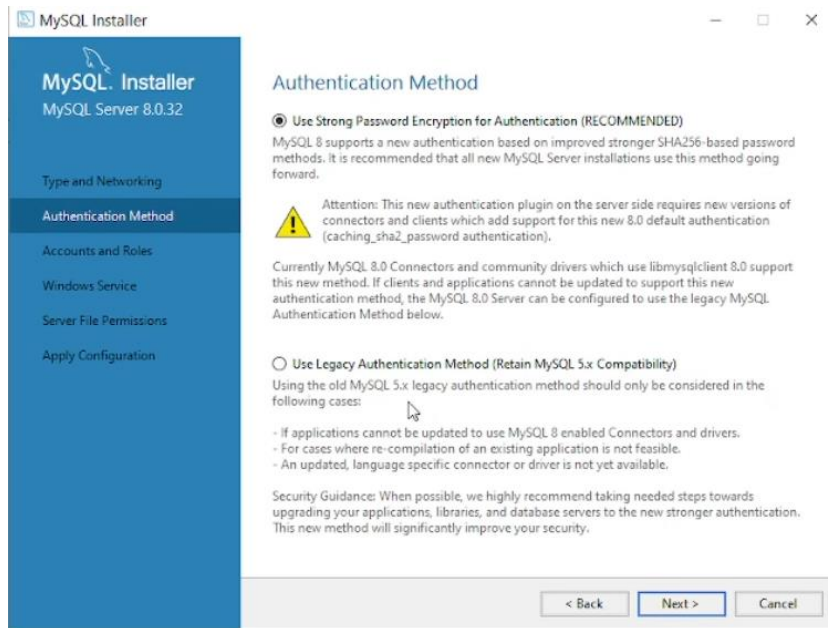


Ilustración 150 Método de autenticación por defecto

22. Digitar una contraseña segura para inicio de sesión del usuario "root" y presionamos siguiente.

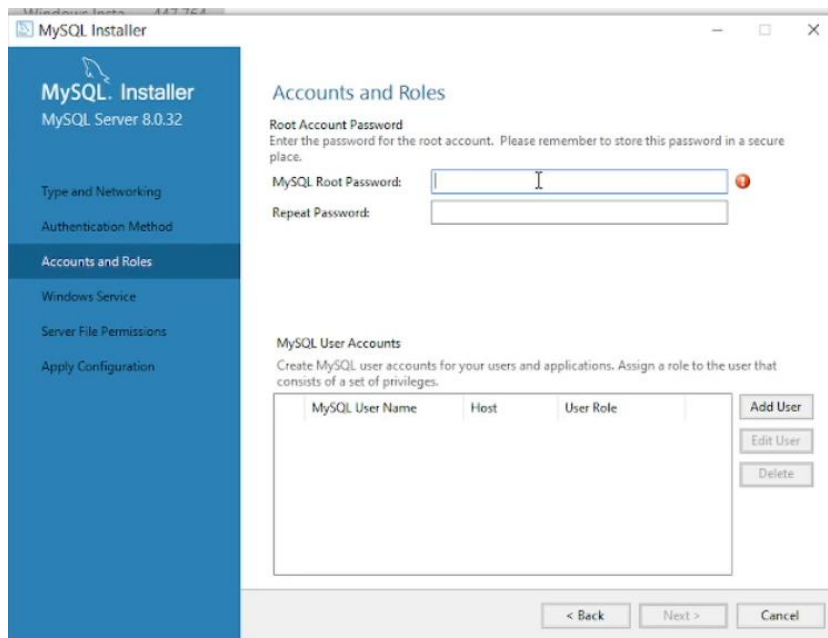


Ilustración 151 Configuración de credenciales de acceso

23. Para el siguiente modal presionar siguiente sin hacer modificaciones.

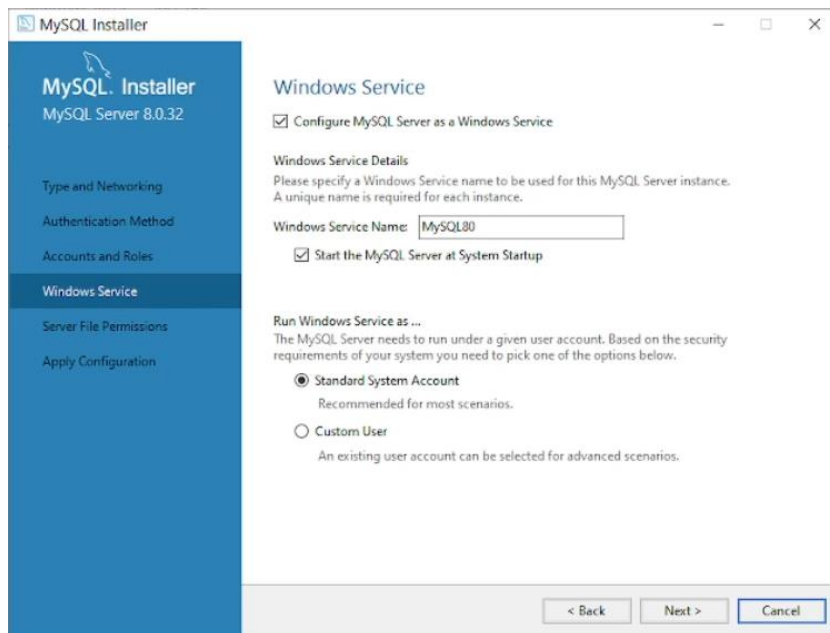


Ilustración 152 Servicio de Windows MySQL

24. Configurar los permisos de archivos del servidor, en este caso dejar la primera opción.

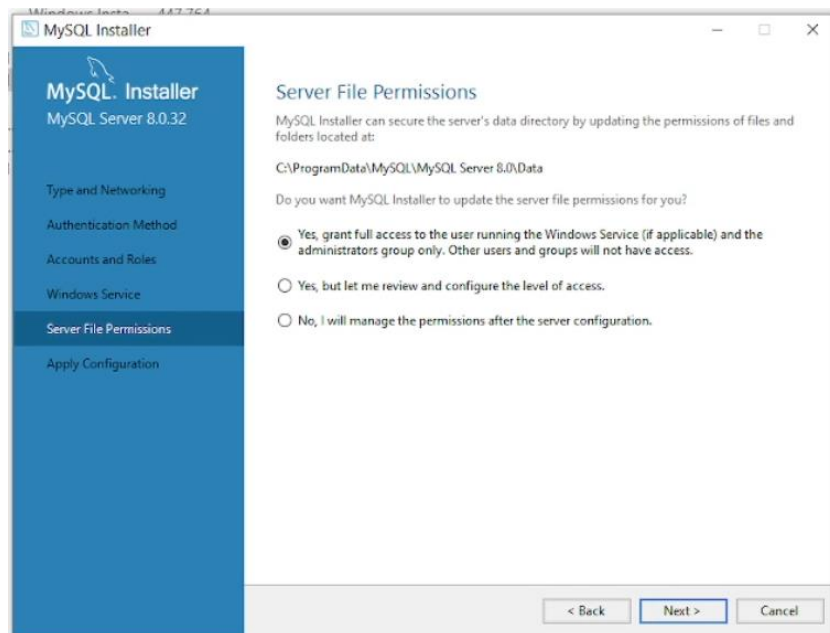


Ilustración 153 Permisos de archivos del servidor

25. En el siguiente modal presionar en ejecutar

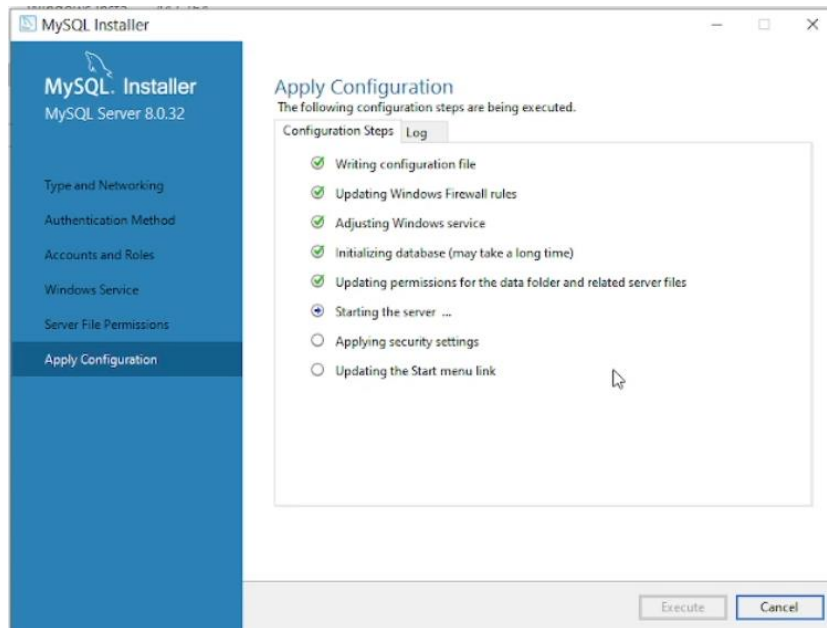


Ilustración 154 Aplicando configuración MySQL

26. Aparecerá la configuración del producto presionar siguiente.

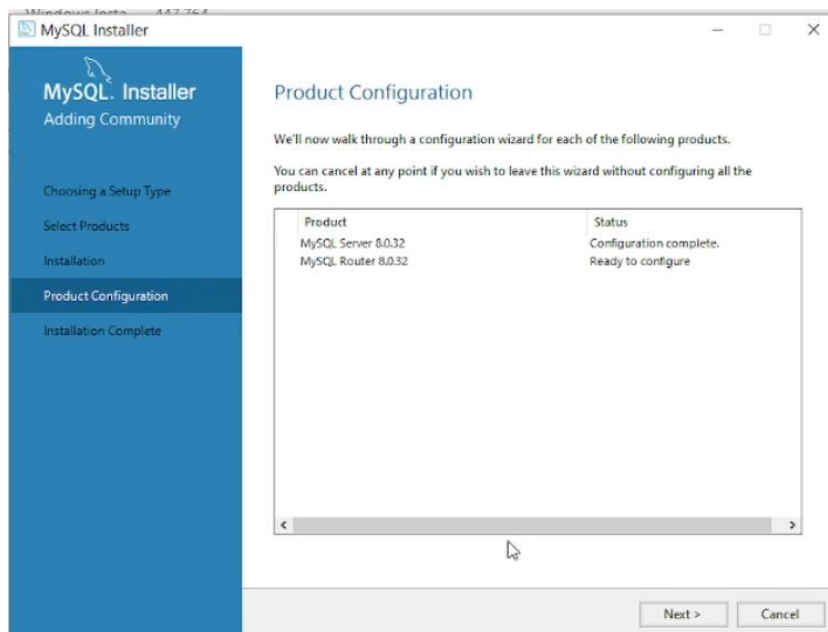


Ilustración 155 Configuración de MySQL

27. En el siguiente modal presionar finalizar

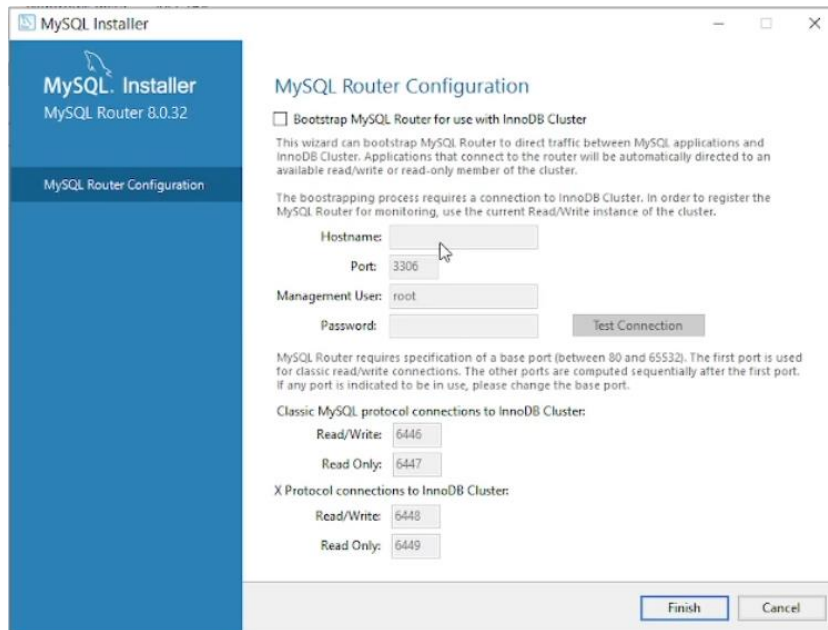


Ilustración 156 Finalizar instalación

28. Aparecerá la pantalla final con la instalación completada, presionar finalizar.

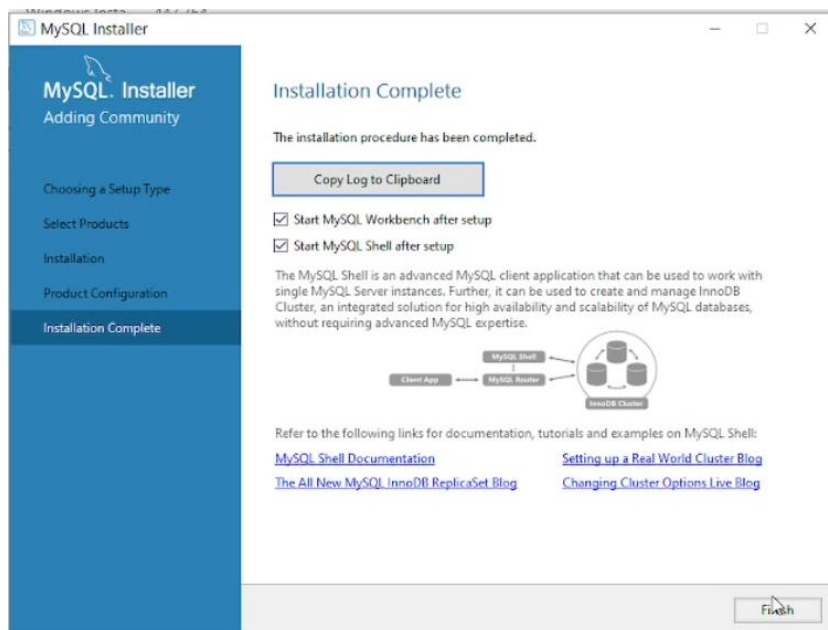


Ilustración 157 Instalación completada

29. Iniciar Wowkbench y configurar una conexión.

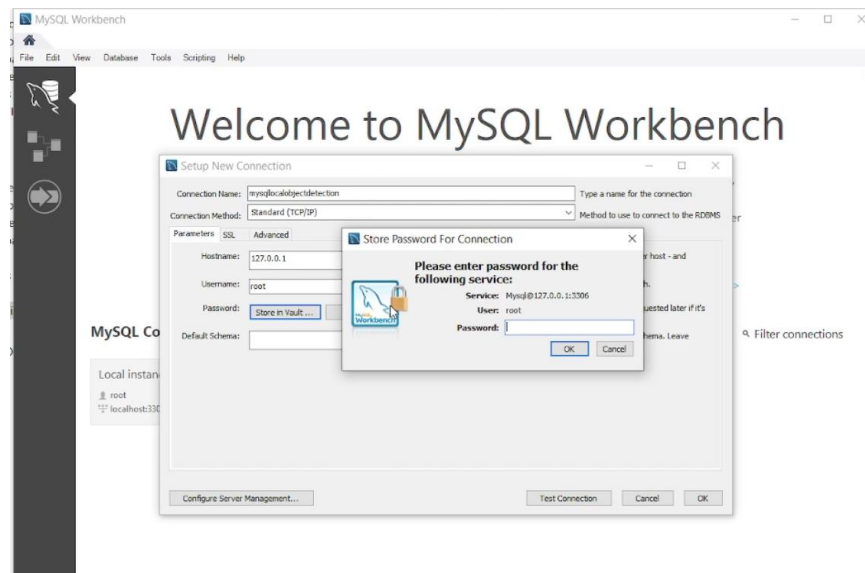


Ilustración 158 Configuración en MySQL Workbench

30. Esta todo listo para trabajar en base de datos

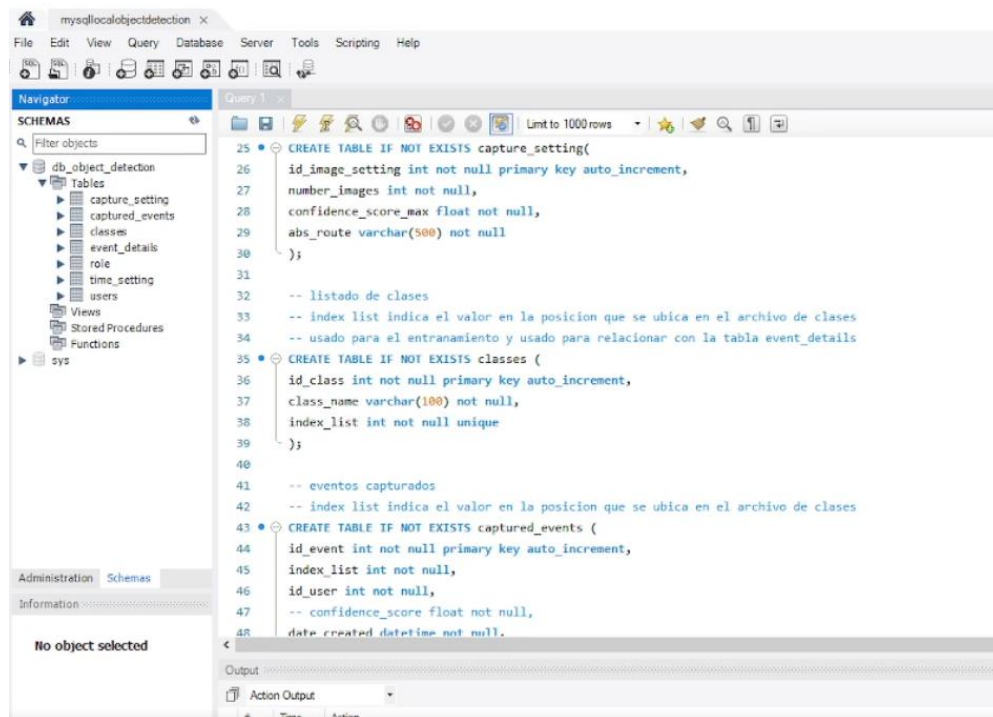


Ilustración 159 Ejecutando base de datos

Instalación del software

Una vez se ha obtenido el equipo necesario e instalado los requisitos previos, procedemos a instalar y configurar el programa, para esto seguir los siguientes pasos:

Paso 1

Mover el fichero zip, rar o carpeta que se haya entregado con el ejecutable a la ubicación de nuestra preferencia. En caso de ser un archivo comprimido, debemos de descomprimir el contenido con el gestor de archivos de nuestra preferencia.

Paso 2

Entrar a la carpeta del programa, y ubicar el archivo "settings_db.json", abrir para establecer las credenciales de cada servicio requerido:

```
"connection": [{
  "host": "127.0.0.1",
  "user": "root",
  "password": "",
  "database": "db_object_detection",
  "port": "3306"
}],
"paths": [{
  "images_temp_login": "C:\\Users\\Edgardo\\Documents\\GitHub\\watchdog_sense\\watchdog_sense\\static\\src\\login_images\\",
  "root": "C:\\Users\\Edgardo\\Documents\\GitHub\\watchdog_sense\\watchdog_sense\\static\\src\\",
  "event_images": "C:\\Users\\Edgardo\\Documents\\GitHub\\watchdog_sense\\watchdog_sense\\static\\src\\"
}],
"twilio": {
  "sid": "ACee86e58facb2eede85900f0bf914d6bf",
  "token": "e7b3aa940027da68b08a5077f6c0a158",
  "from": "+15416948574"
}
```

Ilustración 160 Archivo configuración servidor

Donde los parámetros requeridos son los siguientes:

Connection: Contenedor de la configuración de base de datos del proyecto.

- **Host:** dirección IP o dominio en el que se encuentra alojada la base de datos.
- **User:** usuario con acceso a la base de datos del proyecto.
- **Password:** contraseña del usuario de la base de datos.
- **Database:** nombre de la base de datos del proyecto.
- **Port:** Puerto en el que escucha el servidor de base de datos.

Paths: Contenedor de la configuración del almacenamiento de imágenes del proyecto.

- **Images_temp_login:** carpeta donde se guardan las imágenes utilizadas para el inicio de sesión.
- **Root:** carpeta principal donde se guardan las imágenes del proyecto.
- **Event_images:** carpeta donde se guardan las imágenes de los eventos detectados.

twilio: Contenedor de la configuración de la cuenta de Twilio del proyecto.

- **From:** número telefónico asignado por Twilio a la cuenta que envía los SMS.
- **Sid:** identificador de sesión provisto por Twilio en su portal.
- **Token:** clave de acceso única provista por Twilio en su portal.

Paso 3

Conectarse a la base de datos MySQL utilizando el gestor de bases de datos de preferencia, y crear la base de datos a utilizar por el proyecto (debe coincidir con lo definido en el archivo settings_db.json). Popular la base de datos con el archivo mysql_db_object_detection_v1.sql que se encuentra en la carpeta base_de_datos.

Paso 4

Ejecutar el archivo main.exe para iniciar el software, si el programa inicia correctamente hemos terminado de instalar y configurar el software. Si el programa se cierra se debe de validar que las configuraciones y base de datos se encuentren creadas correctamente.

Uso del software

El servidor del prototipo del sistema de visión artificial se compone por 3 funcionalidades clave, las cuales permiten gestionar los usuarios que utilizan el sistema y manejar los servicios de videovigilancia en tiempo real.

Navegación del prototipo

El servidor del prototipo se ejecuta como una aplicación de escritorio, la cual maneja una sección para usuarios sin autenticar, los cuales pueden realizar el proceso de registro e inicio de sesión, y otra sección para usuarios previamente autenticados quienes pueden iniciar y detener los servicios de videovigilancia en tiempo real.

Menú de navegación

Se muestran las opciones en la barra lateral izquierda, las cuales inicialmente aparecen en un estado retraído inicialmente, mostrando únicamente los iconos de navegación:

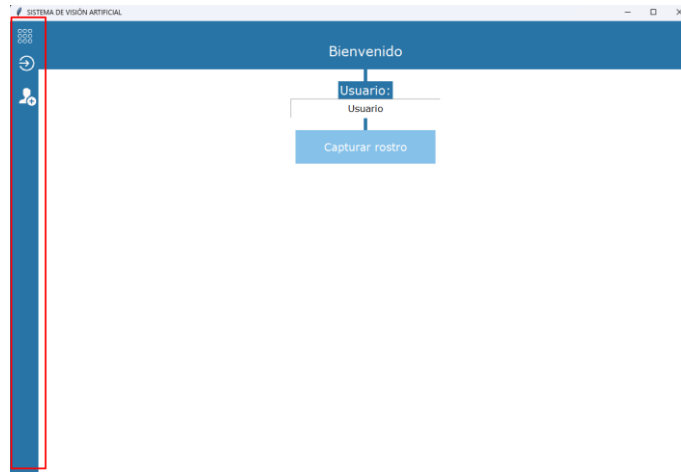


Ilustración 161 Menú de navegación

Si el usuario lo desea, puede expandir el menú y sus opciones mediante la primera opción representada por un icono de 9 puntos, el cual mostrara el menú con las opciones representadas tanto por el icono como por el nombre de la opción.

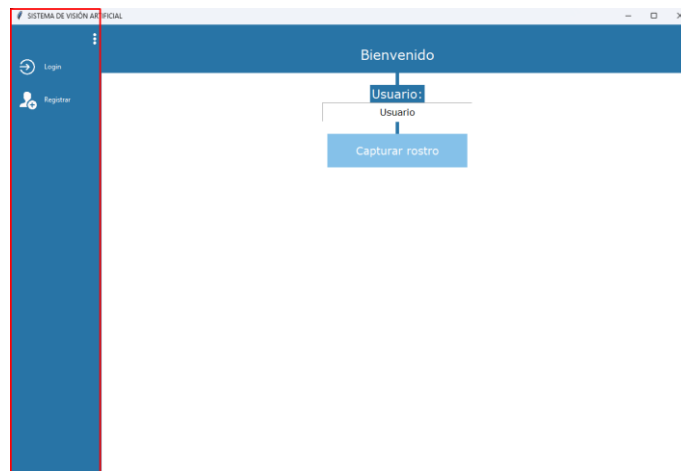


Ilustración 162 Menú de navegación expandido

Contenido del software

En el espacio restante de la aplicación (no utilizado por el menú), se mostrará el contenido de la opción que se haya seleccionado.

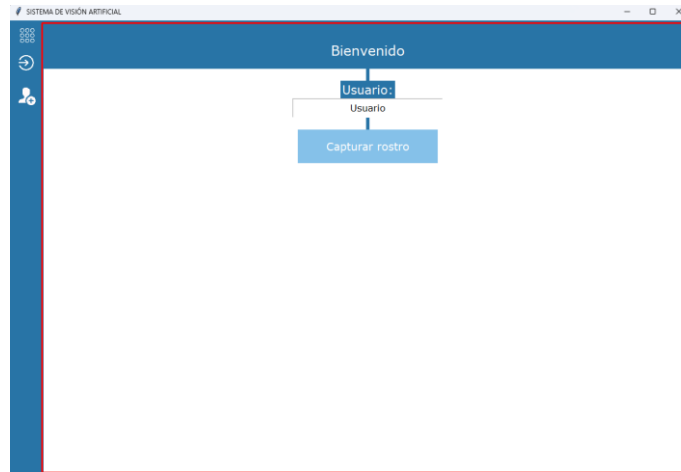


Ilustración 163 Contenido del software

Registro

Para el despliegue del servidor del prototipo se requiere tener una cuenta de usuario en el sistema, para lo cual se ha creado un registro de usuarios, el cual permite crear una cuenta a partir de los datos básicos del usuario, los cuales se asocian a los datos faciales del usuario los cuales son utilizados para autenticar en lugar de utilizar una contraseña.



Ilustración 164 Botón registro

Para iniciar, se debe escoger la opción "Registro" en el menú principal el cual es representado por el icono de una persona/usuario con un símbolo más por encima de este. Una vez se ha accedido a la pantalla, se mostrará un formulario donde se deberán de complementar los datos del usuario y posterior a esto realizar la captura del rostro mediante el botón "Capturar rostro" la cual servirá para autenticar al usuario en lugar de utilizar una contraseña.



Ilustración 165 Formulario registro

Si al momento de intentar capturar el rostro, se encuentra algún error en los datos del formulario o bien un usuario

duplicado, el sistema retroalimentara al usuario con una alerta para que pueda verificar los datos.

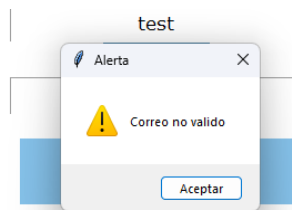


Ilustración 166 Alertas registro

Una vez los datos ingresados son correctos se mostrará una ventana emergente la cual cargará la cámara web de la computadora donde se realizará la captura del rostro. Una vez el usuario se haya posicionado para capturar el rostro, deberá de presionar la tecla "ESC" de su teclado para realizar la captura de datos.

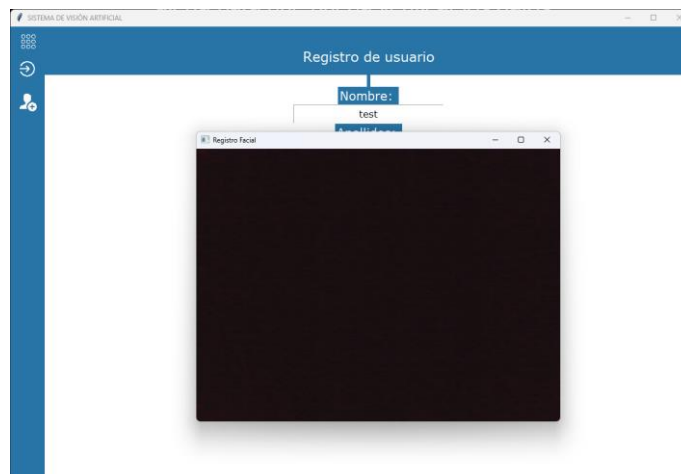


Ilustración 167 Captura de rostro registro

Si el proceso es exitoso, se dará paso al usuario hacia el sistema.

Inicio de sesión

Si el usuario que desea utilizar el prototipo ya posee un usuario, entonces deberá hacer uso de la opción "Login" del software, la cual es la opción que carga por defecto al iniciar el prototipo. Dicha opción está representada por una flecha con un círculo semicerrado alrededor.

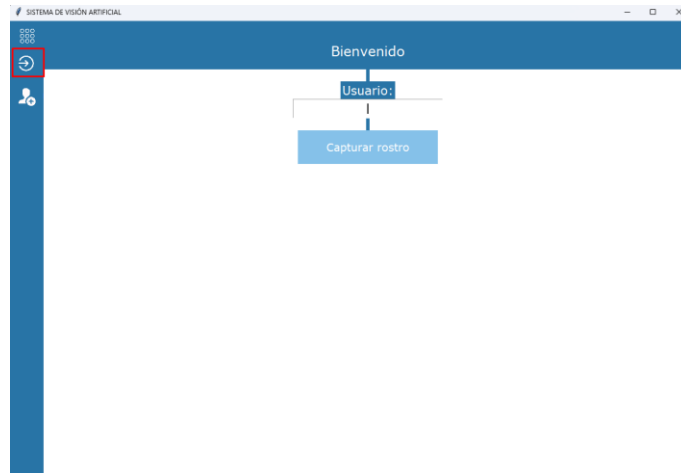


Ilustración 168 Botón login

En dicha pantalla, el usuario solo deberá de ingresar su nombre de usuario en el campo de texto "Usuario" y posteriormente utilizar el botón "Capturar rostro" con el cual se mostrará una ventana emergente la cual cargará la cámara web de la computadora donde se realizará la captura del rostro.

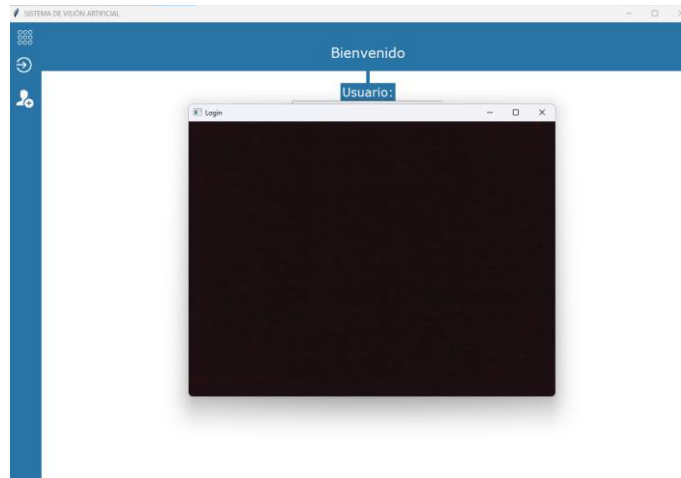


Ilustración 169 Captura de rostro login

Una vez el usuario se haya posicionado para capturar el rostro, deberá de presionar la tecla "ESC" de su teclado para realizar la captura de datos. Si el proceso es exitoso se brindará acceso hacia el sistema.

Iniciar/parar videovigilancia

Para gestionar los servicios de videovigilancia en tiempo real del prototipo, se debe de hacer uso de la opción "Capturar", la cual es representada con el icono de una cámara, que de igual modo es la opción que se carga por defecto posterior a haber iniciado sesión.

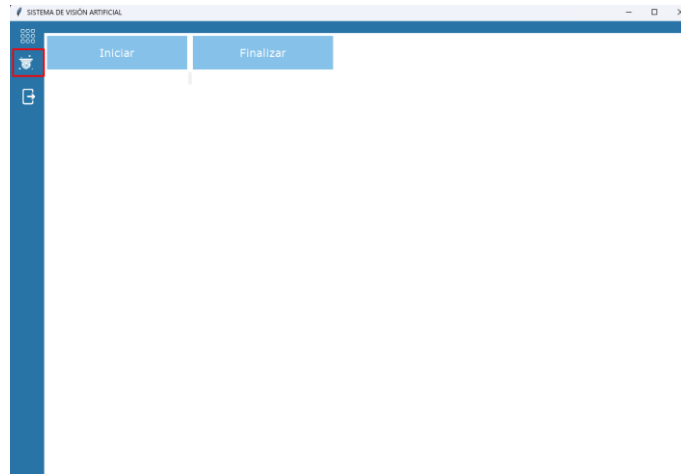


Ilustración 170 Botón captura

Dicha opción maneja 2 botones:

- Iniciar: botón que permite iniciar la detección de eventos mediante videovigilancia en tiempo real. Posterior a esto no se requiere la intervención del usuario.

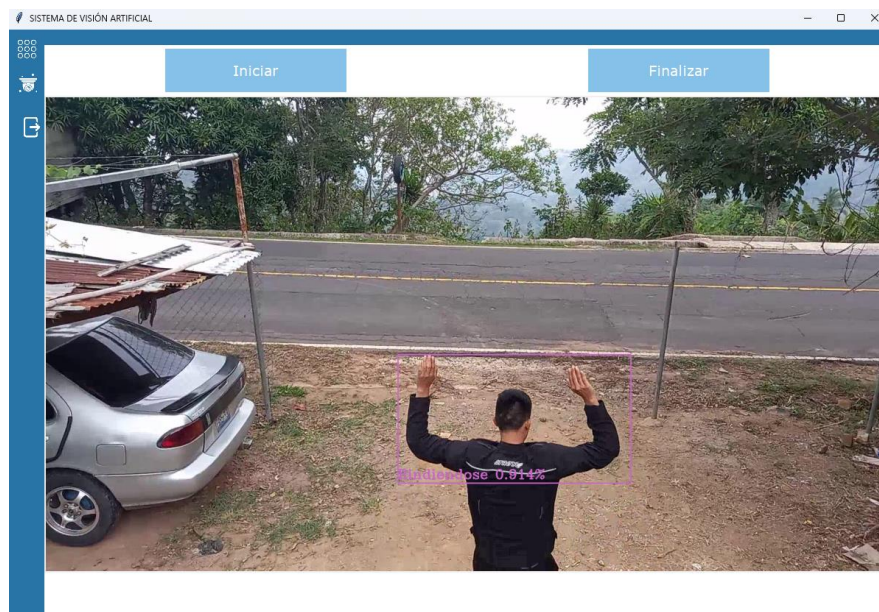


Ilustración 171 Inicio captura eventos

- Finalizar: botón que detiene la detección de eventos en tiempo real.

Cierre de sesión

Si el usuario requiere cerrar la sesión, puede realizarlo mediante la opción "Salir", la cual es representada como un rectángulo con una flecha que inicia desde la mitad de este y finaliza fuera del mismo.

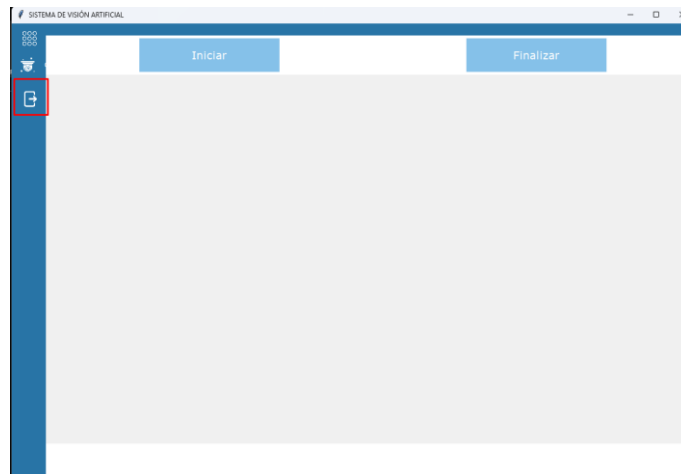


Ilustración 172 Botón salir

Otras funcionalidades

Mientras el software puede ser manipulado por un usuario para realizar diversas acciones, existen otras opciones las cuales están fuera del control del usuario, las cuales explicamos en este apartado.

Envió de correos

Al detectarse un evento el cual cumpla con las condiciones mínimas como para generar una alerta, entonces se procederá a enviar los correos de notificación a los correos de los usuarios


que apliquen para dicha opción haciendo uso del servidor SMTP de correos ingresado al momento de configurar las credenciales del proyecto.

Nuevo evento sospechoso 2023-06-06 20:59:52

PA **Prototipo** Vision Artificial <tesisvisionartificial2023@outlook.com>
Para: Usted

ATENCIÓN

El sistema de vision artificial ha detectado el siguiente evento sospechoso:



Fecha y hora: 2023-06-06 20:59:52

Tipo: Rindiendose

Indice de confiabilidad: 0.8539050817489624

Ver evento: [Link del evento](#)

Recomendamos la pronta atencion del evento y prudencia ante cualquier eventualidad.

[← Responder](#) [→ Reenviar](#)

Ilustración 173 Correo de alerta

Envió de mensajes de texto (SMS)

Al detectarse un evento el cual cumpla con las condiciones mínimas como para generar una alerta, entonces se procederá a enviar un mensaje de texto de notificación a los números de teléfono de los usuarios que apliquen para dicha opción haciendo uso del API de Twilio configurado a través de las credenciales al momento de configurar las credenciales del proyecto.

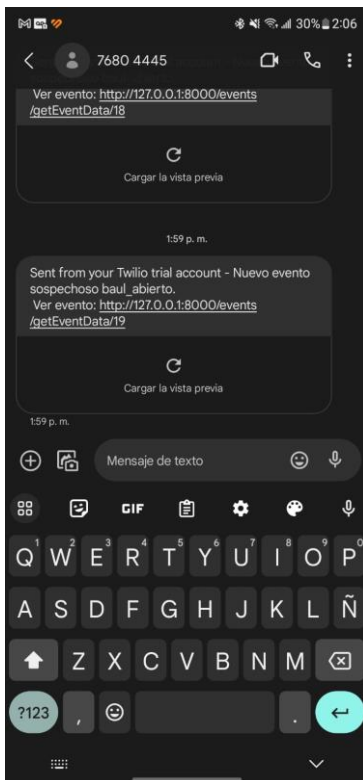


Ilustración 174 Mensaje de alerta

Alerta sonora

Al detectarse un evento el cual cumpla con las condiciones mínimas como para generar una alerta, entonces se procederá a emitir una alerta sonora a través de un dispositivo Arduino conectado a la máquina que ejecute el servidor del prototipo.

Glosario

Arduino: Es una plataforma de prototipado electrónico de código abierto que se utiliza para crear proyectos interactivos y dispositivos electrónicos. Incluye una placa con microcontrolador.

Base de datos: Es un conjunto organizado de datos estructurados y relacionados, almacenados y accesibles a través de un sistema de gestión de bases de datos (DBMS). Se utiliza para almacenar, administrar y recuperar información de manera eficiente.

Dirección IP: Es una serie de números que identifica de manera única un dispositivo en una red. Se utiliza para establecer la conexión y el intercambio de datos entre diferentes dispositivos en Internet.

Hardware: Se refiere a los componentes físicos y tangibles de una computadora o dispositivo electrónico, como la unidad central de procesamiento (CPU), la memoria, los discos duros, las tarjetas de video, etc.

Lente: Es una pieza óptica utilizada en cámaras y otros dispositivos para enfocar la luz y formar una imagen. Puede ser intercambiable y su calidad afecta la nitidez y calidad de las imágenes capturadas.

Login: Es el proceso de autenticación en un sistema informático o plataforma en línea. Consiste en proporcionar las credenciales de acceso, como un nombre de usuario y una contraseña, para verificar la identidad del usuario y permitir el acceso autorizado.

MySQL: Es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto ampliamente utilizado. Proporciona un entorno seguro y eficiente para almacenar, organizar y recuperar datos estructurados.

Procesador: Es el componente principal de una computadora que realiza las instrucciones y cálculos necesarios para ejecutar programas y realizar tareas.

Protocolo de transmisión: Es un conjunto de reglas y normas que definen cómo los dispositivos de una red se comunican y transmiten datos entre sí. Establece el formato de los mensajes, la secuencia de intercambio y la detección y corrección de errores.

Prototipo: Es un modelo inicial o muestra preliminar de un producto o sistema que se crea para probar y evaluar su funcionalidad antes de su producción o implementación completa.

Python: Es un lenguaje de programación de alto nivel y de propósito general, conocido por su simplicidad y legibilidad. Es ampliamente utilizado en el desarrollo de aplicaciones web, científicas y de automatización de tareas.

RAM (Random Access Memory): Es la memoria de acceso aleatorio, donde se almacenan temporalmente los datos y programas que están siendo utilizados activamente por la computadora. Es más rápida pero volátil, ya que pierde su contenido cuando se apaga la computadora.

Rar: Es otro formato de archivo comprimido similar al zip, pero con una mayor capacidad de compresión. Es utilizado para comprimir y empaquetar archivos en un solo archivo, facilitando su transporte y almacenamiento.

Resolución: Es la medida de la cantidad de detalle que se puede ver en una imagen o pantalla. Se expresa en píxeles y se representa mediante la cantidad de píxeles horizontales y verticales, como 1920x1080 (Full HD).

Sistema operativo: Es el software que controla y coordina el funcionamiento de una computadora. Proporciona una interfaz entre el usuario y el hardware, y administra recursos como archivos, memoria, procesos y dispositivos.

SMS (SMS): SMS es el acrónimo de "Short Message Service" (Servicio de Mensajes Cortos, en español). Se trata de un protocolo estándar utilizado para enviar mensajes de texto entre dispositivos móviles, como teléfonos móviles o smartphones.

Software: Es el conjunto de programas, instrucciones y datos que controlan las operaciones de una computadora o dispositivo electrónico. Incluye sistemas operativos, aplicaciones, controladores y otros programas.

Tarjeta gráfica: Es un componente de hardware especializado que se encarga de procesar y generar imágenes en una computadora. Se utiliza para mejorar el rendimiento gráfico, especialmente en aplicaciones de videojuegos y diseño.

Twilio: Es un proveedor de servicios que brinda APIs especializadas en el envío de mensajes de texto (SMS).

Visión artificial: Se refiere a la capacidad de una máquina o computadora para procesar y analizar imágenes o videos de manera similar a la visión humana, utilizando algoritmos y técnicas de inteligencia artificial.

Zip: Es un formato de archivo comprimido utilizado para reducir el tamaño de los archivos y facilitar su almacenamiento

y transferencia. Permite agrupar múltiples archivos en un único archivo zip y comprimirlos para ahorrar espacio.

MANUAL DE USUARIO

APLICATIVO WEB

ESCUELA ESPECIALIZADA EN INGENIERIA ITCA-FEPADE

ESCUELA DE COMPUTACIÓN



MANUAL DE USUARIO

Prototipo de sistema de videovigilancia automatizado por medio de visión artificial para ITCA-FEPADE.

PRESENTADO POR:

Edgardo Antonio Argueta Bautista

Jefferson Francisco Pineda Quintanilla

Kevin Rodrigo Flores Acosta

Nicolás Gabriel Escobar Pérez

Versión 1.0

Santa Tecla, La Libertad, El Salvador

ÍNDICE

Introducción	279
acceso	280
inicio de sesión	281
roles de sistema	283
estructura de menú	283
funcionalidades	284
REGISTRO DE EVENTOS	284
<i>cambiar estado de evento:</i>	284
GESTIÓN DE USUARIOS	285
<i>agregar nuevo usuario</i>	285
<i>modificar usuario:</i>	286
<i>eliminar usuario</i>	287
HERRAMIENTAS.....	288
<i>configuración general de parámetros</i>	288
<i>cierre de sesión</i>	288
glosario	290

Introducción

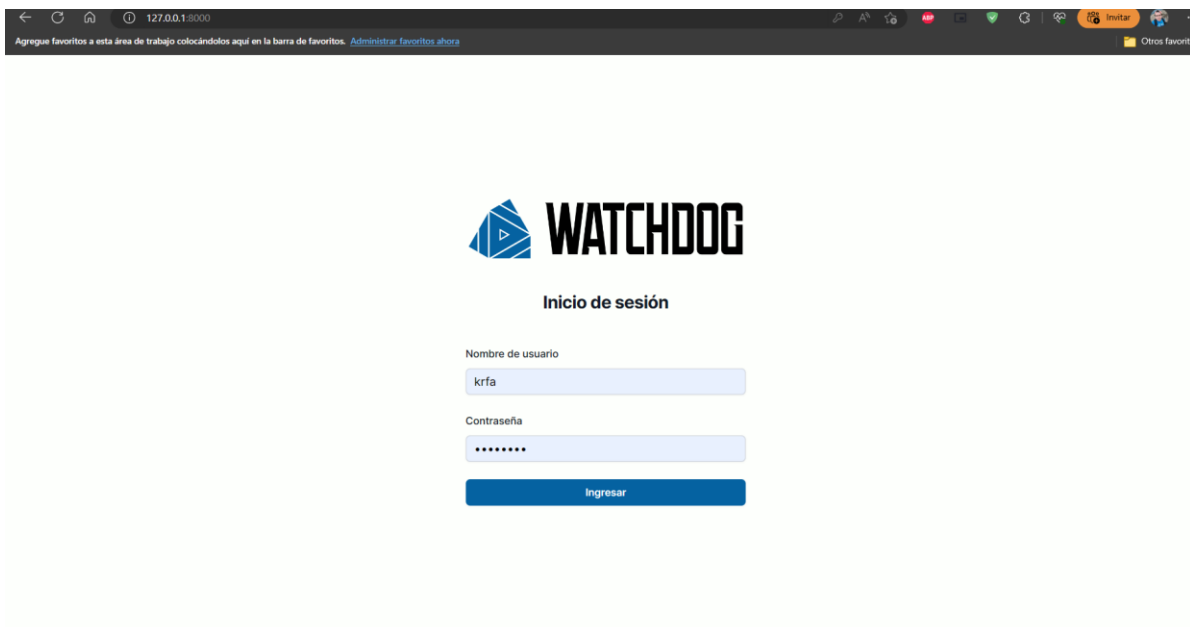
"Watchdog" es un proyecto que se desarrolla en conjunto al sistema de videovigilancia con visión artificial, el cual permite a los usuarios administradores y operadores, visualizar los diferentes eventos que son detectados, además de la gestión de los usuarios mismos, y configuraciones generales del mismo.

El siguiente documento, tiene como finalidad el de guiar a los diferentes usuarios que utilizarán en algún momento el sistema, y así proveerles una guía de usabilidad ideal.

Acceso

Para acceder al aplicativo, es necesario conocer el enlace de acceso, el cual puede ser proporcionado por la administración de informática en el despliegue. En este caso se ha tomado el ejemplo en un entorno de prueba, y el enlace de acceder será: "http://127.0.0.1:8000/"

Una vez con el enlace de acceso, abrimos cualquier navegador de preferencia, e ingresar el enlace en la barra de navegación. Se mostrará la pantalla inicial del aplicativo.



Inicio de sesión

Para poder acceder a las funcionalidades del sistema, es necesario poseer credenciales de acceso, las cuales serán proporcionadas por la administración del sistema.

En dado caso ya se tengan las credenciales, proceder a ingresar el usuario y luego la contraseña, en el apartado de inicio de sesión que se muestra a continuación



Inicio de sesión

Nombre de usuario

Contraseña

En dado caso las credenciales no sean correctas, no se concederá dicho acceso



Roles de sistema

El sistema posee varias funcionalidades las cuales deben ser utilizadas por diferentes tipos de roles dentro del sistema, los cuales son:

- **Administradores:** Tienen el control total del aplicativo, teniendo la posibilidad de:
 - Gestionar usuarios administradores y operadores
 - Gestionar configuraciones generales del aplicativo
 - Gestionar eventos registrados
- **Operadores:** Se centran en la monitorización de los eventos detectados y la validación de estos.

Estructura de menú

El menú general del sitio o "barra de navegación de usuario" se divide de la siguiente manera:

1. Logo del aplicativo
2. Opciones de menú principal
3. Opciones de menú secundaria

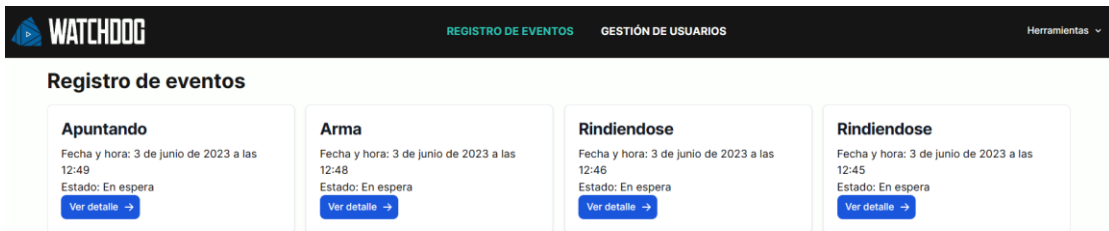


Funcionalidades

A continuación, se detallan los diferentes casos de usabilidad de las funciones que contiene el aplicativo:

Registro de eventos

Para acceder a este apartado, iniciar desde el menú de navegación "Registro de eventos", una vez dentro seleccionar la opción, se mostrará la pantalla con los eventos que se han registrado:



Cambiar estado de evento:

Validar el evento que se ha visualizado en el panel, proceder a seleccionar uno de la lista, y se mostrará un detalle del evento, junto con las diferentes imágenes que se detectaron desde el sistema de videovigilancia:



Cambiar el estado de este, seleccionar el estado y luego presionar "Actualizar estado", para guardarlo.



Gestión de usuarios

Para acceder a este apartado, seleccionar la opción "Gestión de usuarios" en el menú de navegación. Se mostrará un listado de usuarios previamente ingresados. Además, tendremos las funciones de agregar un nuevo, modificar y eliminar usuarios.

The screenshot shows the 'Gestión de usuarios' page in the WATCHDOG system. At the top, there's a navigation bar with 'WATCHDOG' logo, 'REGISTRO DE EVENTOS', 'GESTIÓN DE USUARIOS' (highlighted), and 'Herramientas'. Below the navigation bar, the title 'Gestión de usuarios' is displayed next to a '+ Nuevo usuario' button. A table lists three users with columns for 'ACCIONES', 'NOMBRE', 'APELLIDO', 'USUARIO', 'ROL', and 'CORREO'.

ACCIONES	NOMBRE	APELLIDO	USUARIO	ROL	CORREO
Modificar Eliminar	ROdrigo	Acosta	krfa	Administrador	krfacostak@hotmail.com
Modificar Eliminar	Kevin	Acosta	kero	Operador	krfacostak@hotmail.com
Modificar Eliminar	Kevin Rodrigo	Flores Acosta	roaco	Operador	krfacostak@hotmail.com

Agregar nuevo usuario

Para agregar un nuevo usuario presionar el botón "Nuevo usuario". Se mostrará una ventana emergente, en donde se ingresará la información general del nuevo usuario, así como sus credenciales de acceso.

The image shows a modal window titled "Información de usuario" with a close button (X) in the top right corner. The form contains the following fields:

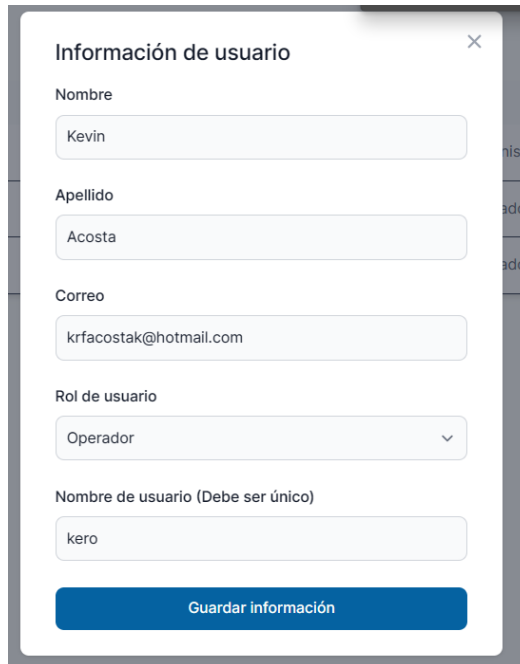
- Nombre:** Text input with the value "Prueba".
- Apellido:** Text input with the value "Usuario prueba".
- Correo:** Text input with the value "prueba@gmail.com".
- Rol de usuario:** Dropdown menu with "Operador" selected.
- Nombre de usuario (Debe ser único):** Text input with the value "prueba".
- Contraseña:** Password input with masked characters "....." and a visibility toggle icon.

At the bottom of the form is a blue button labeled "Guardar información".

Luego de ingresar la información en el formulario, presionar "Guardar información" para que quede registrado ya en el sistema.

Modificar usuario:

Para modificar la información de los usuarios, en el mismo listado que se muestra en pantalla, presionar el botón "Modificar". Se abrirá el panel nuevamente con la información del usuario que hayamos seleccionado:



Información de usuario

Nombre
Kevin

Apellido
Acosta

Correo
krfacostak@hotmail.com

Rol de usuario
Operador

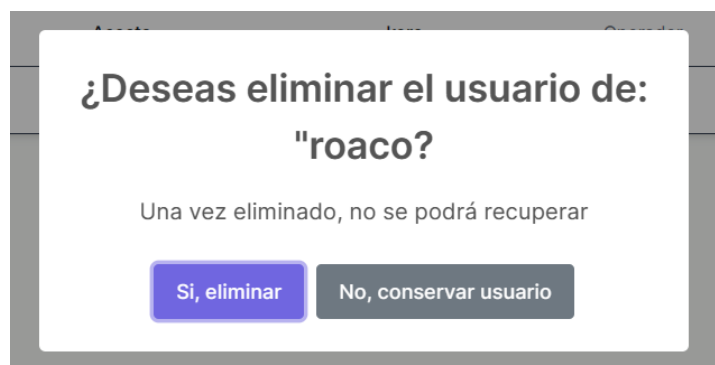
Nombre de usuario (Debe ser único)
kero

Guardar información

Luego de modificar la información pertinente, se procede a presionar el botón "Guardar información"

Eliminar usuario

Si se desea que un usuario ya no tenga acceso a nuestro aplicativo y queremos eliminarlo, bastará con presionar el botón "Eliminar" desde el listado de usuarios. Una vez presionados se nos mostrará un mensaje de confirmación para poder eliminarlo.



¿Deseas eliminar el usuario de:
"roaco?"

Una vez eliminado, no se podrá recuperar

Si, eliminar No, conservar usuario

Herramientas

Configuración general de parámetros

Si desea modificar parámetros generales del sistema, se puede acceder desde el menú de navegación, en el aparato de "Herramientas", y luego seleccionar "Configuraciones".



The screenshot shows the 'WATCHDOG' configuration interface. At the top, there is a navigation bar with 'WATCHDOG' on the left, 'REGISTRO DE EVENTOS' and 'GESTIÓN DE USUARIOS' in the center, and 'Herramientas' on the right. Below the navigation bar, there is a heading 'En este apartado se ingresa las configuraciones generales del sistema. Ingrese la información y proceda a guardarla'. The main content area contains three input fields: 'Número de imágenes (Cantidad de imágenes que se almacenarán por evento)' with the value '20', 'Nivel de confianza de eventos (#.##, Puntuación de confianza mínima permitido para registrar eventos)' with the value '0,45', and 'Ruta de recursos (En esta ruta se almacenarán las imágenes de los eventos detectados)' with the value 'D:/PYTHON_PROJECT/JTCA/Prototipo_vision_artificial/JTCA_FEPAD/event_images/'. At the bottom of the form is a blue button labeled 'Guardar configuraciones'.

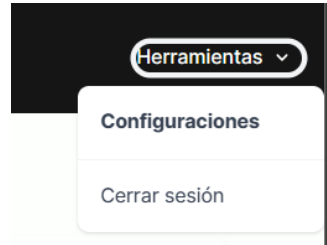
se mostrará unos apartados con un valor ya ingresado, los cuales se podrán modificar, y para guardar, presionar el botón "Guardar configuraciones"

NOTA: El uso de esta pantalla puede afectar el funcionamiento de ambas plataformas, tanto "Watchdog" como el de "Videovigilancia", por lo cual se recomienda utilizarlo únicamente si se sabe qué hacer.

Cierre de sesión

Para cerrar la sesión, podremos realizarlo desde el apartado de "Herramientas" en la barra de navegación.

Una vez presionado, se borrarán del navegador los datos ligados al usuario, mostrando la pantalla de inicio de sesión nuevamente.



Glosario

Credenciales: Conjunto de datos que están asociados a un usuario dentro del sistema, el cual le permite acceder al mismo de forma segura.

Login: Formulario de inicio de sesión principal del sistema.

Rol: Papel que desempeña el usuario dentro del aplicativo, cada rol tiene acciones que puede realizar.

Watchdog: Nombre clave del proyecto administrador de eventos sospechosos

Glosario

CCTV: Un sistema CCTV o Circuito Cerrado de Televisión funciona tras la instalación de un sistema de cámaras de seguridad que permiten comprobar desde otra ubicación (en forma remota), el funcionamiento o el estado de un lugar.

Conda: Conda es un sistema de gestión de entornos y paquetes de código abierto que se ejecuta en Windows, macOS y Linux. Conda instala, ejecuta y actualiza rápidamente los paquetes y sus dependencias.

CUDA: son las siglas de Compute Unified Device Architecture (Arquitectura Unificada de Dispositivos de Cómputo) que hace referencia a una plataforma de computación en paralelo que incluye un compilador y un conjunto de herramientas de desarrollo creadas por Nvidia que permiten a los programadores usar una variación del lenguaje de programación C (CUDA C) para codificar algoritmos en GPU de Nvidia.

Evento sospechoso: es una acción poco frecuente realizada por un individuo, como caminar a escondidas, esconder objetos, robar o hurtar pertenencias de otras personas.

Hurto: Consiste en tomar un bien inmueble ajeno sin que su dueño lo haya consentido, y con ánimo de lucro.

Inteligencia artificial: es el campo de la ciencia informática dedicado a la resolución de problemas cognitivos asociados comúnmente a la inteligencia humana, como el aprendizaje, la resolución de problemas y el reconocimiento de patrones.

NVIDIA® CUDA® Toolkit: proporciona un entorno de desarrollo para crear aplicaciones aceleradas por GPU de alto rendimiento. Con

CUDA Toolkit, puede desarrollar, optimizar e implementar sus aplicaciones en sistemas integrados acelerados por GPU.

PyTorch: es una biblioteca de tensores optimizada para el aprendizaje profundo mediante GPU y CPU, de código abierto basada en la biblioteca de Torch, utilizado para aplicaciones como visión artificial y procesamiento de lenguajes naturales.

Robo: es un delito contra el patrimonio, consistente en el apoderamiento de bienes ajenos de otras personas de manera fraudulenta, empleando para ello fuerza en las cosas o bien violencia o intimidación en las personas.

Software as a product: (Software como producto) es un producto, software, que está hecho para venderse a los usuarios, y los usuarios pagan por la licencia que les permite usarlo, donde los usuarios compran una suscripción.

Videovigilancia: es la grabación o visualización de imágenes grabadas por cámaras de seguridad en espacios públicos o privados, con la finalidad de garantizar la seguridad de las personas o instalaciones.

Visión artificial: La visión artificial es un campo de la IA que permite que las computadoras y los sistemas obtengan información significativa de imágenes digitales, videos y otras entradas visuales, y tomen acciones o hagan recomendaciones basadas en esa información. Si la IA permite que las computadoras piensen, la visión artificial les permite ver, observar y comprender.

Bibliografía

(S/f-a). Uab.cat. Recuperado el 6 de junio de 2023, de https://ddd.uab.cat/pub/tfg/2017/tfg_71066/paper.pdf.

(S/f-b). Github.com. de <https://github.com/rcrespocano/opencv-python/blob/master/opencv-and-python.ipynb>.

¿Qué es la visión artificial? (s/f). Ibm.com. de <https://www.ibm.com/mx-es/topics/computer-vision>

About. (2019, marzo 6). OpenCV. <https://opencv.org/about/>

Anaconda. (2023, April 26). Free Download | Anaconda. <https://www.anaconda.com/download>

Argos.red. DORI en camaras CCTV. <https://argos.red/dori-en-camaras-cctv/>

COCO - common objects in context. (s/f). Cocodataset.org. Recuperado el 12 de junio de 2023, de <https://cocodataset.org/>

Cómo funciona Calcular precisión para la detección de objetos. (s/f). Arcgis.com. Recuperado el 12 de junio de 2023, de <https://pro.arcgis.com/es/pro-app/latest/tool-reference/image-analyst/how-compute-accuracy-for-object-detection-works.htm>

Corbucci, S. (2011). Django. En *100 Cult Films* (pp. 65-66). British Film Institute.

datascientest. (2023). Convolutional Neural Network :
definición y funcionamiento. datascientest.
<https://datascientest.com/es/convolutional-neural-network-es>

Decreto Legislativo N°133.- Portal de Transparencia - El
Salvador. (n.d.). <https://www.transparencia.gob.sv/>

Documentos de Twilio: Referencia de la API, tutoriales e
integración. (s/f). Twilio.com. Recuperado el 28 de junio de
2023, de <https://www.twilio.com/es-mx/docs>

DORI en cámaras CCTV. (2022, agosto 1). *Sistemas de
videovigilancia Argos*. [https://argos.red/dori-en-camaras-
cctv/](https://argos.red/dori-en-camaras-cctv/)

Escobar, H. (2020, 25 marzo). Repositorio Institucional UCA:
Desarrollo de un sistema para el control de calidad de
cristales de azúcar utilizando visión artificial.
<http://repositorio.uca.edu.sv/jspui/handle/11674/2696>

Fuentes, C. (2021, 12 diciembre). Más del 60 % de los robos de
automóviles se registra en San Salvador y La Libertad.
Diario El Salvador. [https://diarioelsalvador.com/mas-del-60-
de-los-robos-de-automoviles-se-registra-en-san-salvador-y-
la-libertad/167778/](https://diarioelsalvador.com/mas-del-60-de-los-robos-de-automoviles-se-registra-en-san-salvador-y-la-libertad/167778/)

Fuentes, C. (2022, 22 agosto). Fiscalía ha procesado una
persona por día por robo de vehículos en últimos 4 años.
Diario El Salvador. [https://diarioelsalvador.com/fiscalia-
ha-procesado-una-persona-por-dia-por-robo-de-vehiculos-en-
ultimos-4-anos/262810/](https://diarioelsalvador.com/fiscalia-ha-procesado-una-persona-por-dia-por-robo-de-vehiculos-en-ultimos-4-anos/262810/)

Marr, B. (2019, 8 abril). 7 Amazing Examples Of Computer And
Machine Vision In Practice. Forbes.
[https://www.forbes.com/sites/bernardmarr/2019/04/08/7-
amazing-examples-of-computer-and-machine-vision-in-practice/](https://www.forbes.com/sites/bernardmarr/2019/04/08/7-amazing-examples-of-computer-and-machine-vision-in-practice/)

Marr, B. (2019, abril 8). *7 amazing examples of computer and
machine vision in practice*. Forbes.
<https://www.forbes.com/sites/bernardmarr/2019/04/08/7->

[amazing-examples-of-computer-and-machine-vision-in-practice/?sh=5915a2d81018](https://www.aprendemachinlearning.com/como-funcionan-las-convolucional-neural-networks-vision-por-ordenador/)

Na. (2018, noviembre 29). *¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador*. Aprende Machine Learning. <https://www.aprendemachinlearning.com/como-funcionan-las-convolucional-neural-networks-vision-por-ordenador/>

NVIDIA CUDA Toolkit - Free Tools and Training. (2023, April 20). NVIDIA Developer. <https://developer.nvidia.com/cuda-toolkit>

Object detection. YOLO. <https://viso.ai/deep-learning/object-detection/>

OpenCV. (2023, May 24). OpenCV - Open Computer Vision Library. <https://opencv.org/>

Paz Centeno, I. (s/f). mtcnn: MTCNN face detection implementation for TensorFlow, as a PIP package.

Prosegur. ¿Qué es un sistema CCTV y cómo funciona?. <https://www.prosegur.es/blog/seguridad/sistema-cctv-beneficios-seguridad-empresarial>

PyTorch documentation - PyTorch 2.0 documentation. (n.d.). <https://pytorch.org/docs/stable/index.html>

SMS with Twilio Programmable Messaging: Send and receive texts in your app. (s/f). Twilio.com. Recuperado el 28 de junio de 2023, de <https://www.twilio.com/es-mx/docs/sms>

Ultralytics Docs. (s. f.). <https://docs.ultralytics.com/>

Ultralytics. (s. f.). Brief summary of YOLOv8 model structure · Issue #189 · ultralytics/ultralytics. GitHub. <https://github.com/ultralytics/ultralytics/issues/189>

Wikipedia contributors. (2023b). Motor vehicle theft.
Wikipedia. https://en.wikipedia.org/wiki/Motor_vehicle_theft

Yolo v8. (s/f). Ultralytics. de <https://ultralytics.com/yolov8>

Anexos

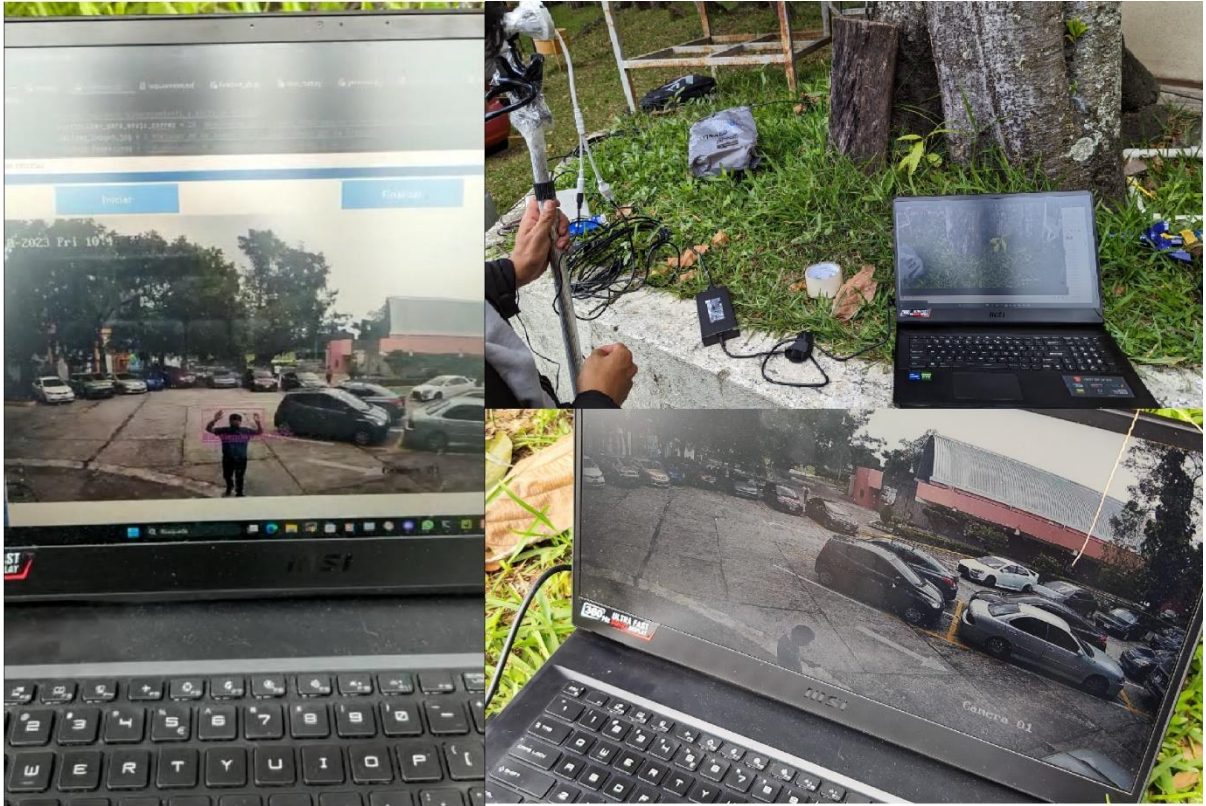


Ilustración 175 Pruebas y capturas de evento