



**UNIVERSIDAD ESTATAL
PENÍNSULA DE SANTA ELENA**

**FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES**

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

TRABAJO DE TITULACIÓN

Propuesta Tecnológica, previo a la obtención del Título de:

INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

“Sistema de detección de personas y cálculo de distancias euclidianas mediante el uso de Redes Neuronales Convolucionales integrando OpenCV y CUDA para medir el índice de cumplimiento del distanciamiento social de los estudiantes de la EEB Lcda. Angélica Villón Lindao”

AUTOR:

ALEJANDRO ROCA KATHERINE VIVIANA

PROFESOR TUTOR:

ING. SHENDRY ROSERO VASQUEZ

LA LIBERTAD - ECUADOR

2021

AGRADECIMIENTO

A Dios, por la vida y los talentos que me regala.

A mis padres, Amalia y Teófilo, por forjar en mí, cimientos sólidos que me llevaron a conseguir este objetivo.

A mis hermanos, Karen, Keyla, Kevin y Kayra, por los momentos felices y tristes que nos tocó vivir en familia y que me llevaron a valorar aún más este logro.

A mi familia política, Tomalá Guaranda, por la ayuda desinteresada, sin ellos no hubiera sido fácil continuar con mis estudios.

A mi tutor, Ing. Shendry Rosero Vásquez, por su guía y tiempo, que sirvieron para culminar este trabajo de manera satisfactoria.

A mis amigos de IRIS, los hermanos que la vida me regaló, por permitirme aprender de ellos y disfrutar juntos este logro que será el primero de muchos.

DEDICATORIA

A mis hijas, Kenia y Karol, por y para ustedes todos mis logros.

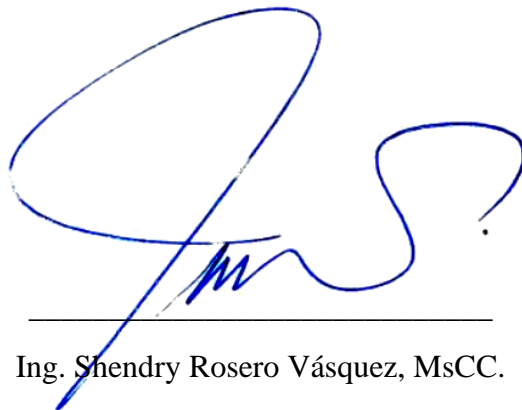
A mi esposo, amigo, compañero de vida y mi colega, Andrés, por decir si y atreverse a soñar conmigo, por mostrarme que soy capaz de cosas grandes y que juntos todo es posible.

A mi padre, Teófilo Inocente Alejandro Reyes por enseñarme a ser fuerte ante el dolor. Este título es tuyo padre mío, mi guerrero, mi eterno ángel. Gracias por tu vida, porque a pesar del cansancio de los años siempre buscaste lo mejor para tu familia Alejandro Roca. Hasta vernos otra vez, te amo.

APROBACIÓN DEL TUTOR

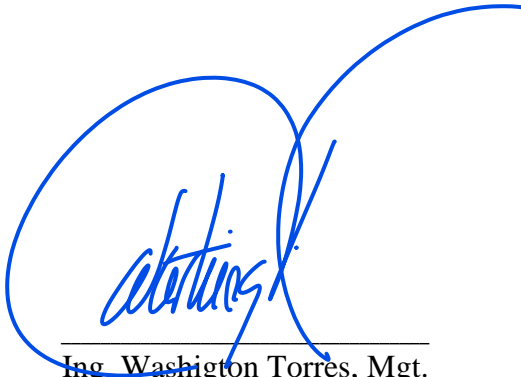
En mi calidad de Tutor del trabajo de titulación denominado: **“Sistema de detección de personas y cálculo de distancias euclidianas mediante el uso de Redes Neuronales Convolucionales integrando OpenCV y CUDA para medir el índice de cumplimiento del distanciamiento social de los estudiantes de la EEB Lcda. Angélica Villón Lindao”**, elaborado por el estudiante **Alejandro Roca Katherine Viviana**, de la carrera de **Tecnologías de la información** de la Universidad Estatal Península de Santa Elena, me permito declarar que luego de haber orientado, estudiado y revisado, lo apruebo en todas sus partes y autorizo al estudiante para que inicie los trámites legales correspondientes.

La Libertad, agosto del 2021

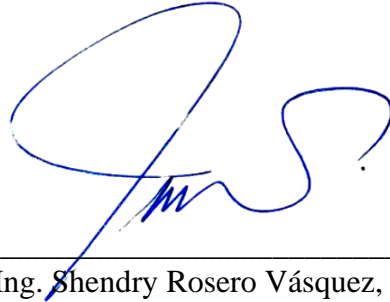


Ing. Shendry Rosero Vásquez, MsCC.

TRIBUNAL DE GRADO



Ing. Washigton Torres, Mgt.
DIRECTOR DE CARRERA



Ing. Shendry Rosero Vásquez, MsCC.
PROFESOR TUTOR



Ing. Marjorie Coronel, MGTL.
PROFESOR DE ÁREA



Ing. Iván Sánchez, MSc.
PROFESOR DE ÁREA

RESUMEN

La Escuela de Educación Básica “Lcda. Angélica Villón Lindao” se enfoca en brindar una educación acorde a las exigencias que la sociedad actual demanda. Con la aparición del virus COVID-19, las actividades educativas se vieron afectadas provocando el cierre de los centros de estudio a nivel mundial. En Ecuador, el Comité de Operaciones de emergencia (COE) Nacional mediante resolución resolvió declarar el distanciamiento social como el camino a la nueva normalidad, medida adoptada para evitar el contagio masivo en espacios de uso compartido. Reducir el aforo permitido en lugares de concurrencia es una manera de controlar el distanciamiento entre personas, sin embargo, depende de la actitud de cada persona cumplir o no esta medida. Los niños, niñas y adolescentes (NNA) en etapa de escolaridad son propensos a evadir los controles dispuestos por las autoridades del plantel. La solución planteada en este trabajo de titulación es desarrollar un sistema de Inteligencia Artificial (IA) basado en técnicas de visión por computadora capaz de detectar personas en un fotograma mediante la red convolucional YOLO (You Only Look Once) y calcular la distancia euclidiana entre puntos usando las coordenadas de las detecciones que devuelve la red neuronal con el método “*dist*” del módulo *Scipy*. PyQt5 es una librería escrita en lenguaje Python para el diseño y desarrollo de interfaces gráficas para entornos de escritorio, que integra el análisis del video en tiempo real y el cálculo de distancia. El resultado de este proyecto es una aplicación para Windows que renderiza el video con las detecciones enmarcadas en rojo si son infractores y en verde si no. Adicionalmente, se genera un reporte estadístico a partir de los registros guardados en una base de datos SQLite con el número de detecciones, el número de infractores, la distancia promedio y el porcentaje de cumplimiento agrupados por horas según la fecha elegida.

ABSTRACT

Lcda. Angélica Villón Lindao school focuses on providing an education according to the demands that today's society demands. With the appearance of the COVID-19 virus, society limited its daily activities such as education, causing the closure of study centers worldwide. In Ecuador, the National COE by resolution resolved to declare social distancing as the way to the new normal, a measure adopted to avoid massive contagion in shared use spaces. Reducing the capacity allowed in crowded places is a way to control the distance between people, however, it depends on the attitude of each person to comply with this measure or not. The NNA in the schooling stage are prone to evade the controls established by the school authorities. The solution proposed in this titling work is to develop an AI system based on computer vision techniques capable of detecting people in a frame using the YOLOv3 convolutional network and calculating the Euclidean distance between points using the coordinates of the detections returned by the network. neuronal with the *dist* method of the *Scipy* module. PyQt5 is the library used for the development of the desktop application that integrates the analysis of the video in real time and the calculation of distance. The result of this project is a desktop application for Windows environments that renders the video with the detections framed in red if they are infringing and in green if not. Additionally, a statistical report is generated from the records stored in a SQLite database with the number of detections, the number of offenders, the average distance and the percentage of compliance grouped by hours according to the chosen date.

DECLARACIÓN

El contenido del presente Trabajo de Graduación es de mi responsabilidad; el patrimonio intelectual del mismo pertenece a la Universidad Estatal Península de Santa Elena.



KATHERINE ALEJANDRO R.

Alejandro Roca Katherine Viviana

TABLA DE CONTENIDOS

ITEM	PÁGINA
AGRADECIMIENTO	I
DEDICATORIA	I
TRIBUNAL DE GRADO	III
RESUMEN	IV
ABSTRACT	V
DECLARACIÓN	VI
TABLA DE CONTENIDOS	VII
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABLAS	XIII
LISTA DE ANEXOS	XIV
INTRODUCCIÓN	1
CAPÍTULO I	3
1. FUNDAMENTACIÓN	3
1.1. Antecedentes	3
1.2. Descripción del proyecto	5
1.3. Objetivos del proyecto	9
1.3.1. Objetivo general	9
1.3.2. Objetivos Específicos	9
1.4. Justificación	10
1.5. Metodología del proyecto	12
1.5.1. Metodología de la investigación	12
1.5.2. Análisis de la entrevista	12
1.5.3. Variable	13
1.5.4. Beneficiarios del proyecto	13

1.5.5.	Metodología experimental	14
1.5.6.	Metodología de desarrollo	15
CAPITULO II		17
2.	La propuesta	17
2.1.	Marco Contextual	17
2.1.1.	Escuela de Educación Básica “Lcda. Angélica Villón Lindao”	17
2.1.2.	Lineamientos para la aplicación del plan de continuidad educativa, permanencia escolar y uso progresivo de las instalaciones educativas.	18
2.1.3.	Protocolo para el autocuidado e higiene de la población educativa durante el uso progresivo de las instalaciones educativas COVID-19	19
2.2.	Marco Conceptual	20
2.2.1.	Python 3.8	20
2.2.2.	PyQt5	20
2.2.3.	OpenCV 4.2	20
2.2.4.	Sqlite 3	20
2.2.5.	Scipy	20
2.2.6.	Entornos de desarrollo Integrado (IDE)	20
2.2.7.	Visual Studio Community	21
2.2.8.	CUDA	21
2.2.9.	Elementos a considerar para proyectos de visión por computadora	21
2.2.10.	Detección de objetos con OpenCV	22
2.2.11.	Redes Neuronales Convolucionales (CNN)	23
2.2.12.	Redes neuronales convolucionales basadas en regiones (R-CNN)	23
2.2.13.	YOLOv3 y su arquitectura neuronal	24
2.2.14.	Detección y clasificación de objetos con YOLOv3	25
2.3.	Marco Teórico	26

2.3.1.	La importancia de la Inteligencia Artificial (IA)	26
2.3.2.	Inteligencia Artificial frente al Covid-19	27
2.4.	Componentes de la propuesta	27
2.4.1.	Módulos del Sistema	27
2.4.2.	Requerimientos Funcionales	28
2.4.3.	Requerimientos no Funcionales	29
2.5.	Componentes de la propuesta.	30
2.5.1.	Arquitectura del sistema	30
2.5.2.	Diagrama de casos de uso	31
2.5.3.	Diagrama de clases	32
2.5.4.	Diagrama de flujo de procesos	33
2.5.5.	Diseño de Interfaces gráficas	35
2.6.	Estudios de factibilidad	38
2.6.1.	Factibilidad Técnica	38
2.6.2.	Factibilidad Económica	38
2.6.3.	Factibilidad Operativa	39
2.7.	Desarrollo del sistema	40
2.7.1.	Desarrollo del entorno de detecciones	40
2.7.2.	Desarrollo del código de distanciamiento	44
2.7.3.	Desarrollo del módulo de aplicación	46
2.7.4.	Desarrollo del módulo de reportes	49
2.8.	Implementación del sistema	51
2.9.	Pruebas	54
2.9.1.	Prueba #1: Detección de personas variando nivel de confianza	54
2.9.2.	Prueba #2: Calibrar la distancia de pixeles a metros	54
2.9.3.	Prueba #3: Infractores detectados variando distancia en pixeles.	55

2.9.4.	Prueba #4: Acceso a la aplicación	56
2.9.5.	Prueba #5: Generación de reporte	57
2.10.	Resultados	58
2.10.1.	Resultados Prueba #1	58
2.10.2.	Resultados Prueba #2	58
2.10.3.	Resultados Prueba #3	59
	Conclusiones	64
	Recomendaciones	65
	Bibliografía	66
	ANEXOS	72

ÍNDICE DE FIGURAS

ITEM	PÁGINA
Fig. 1. Fórmula distancia euclidiana.	7
Fig. 2. Métricas de la matriz de confusión.	15
Fig. 3. Modelo Incremental del software.	16
Fig. 4. Ubicación Geográfica de la escuela.	17
Fig. 5. Demostración del modelo edge detection.	22
Fig. 6. Capas red neuronal convolucional.	23
Fig. 7. Arquitectura R-CNN.	24
Fig. 8. Ejemplo salida SVM para una región candidata.	24
Fig. 9. Arquitectura de Darknet-53.	25
Fig. 10. Detección de objetos con Yolo.	26
Fig. 11. Modelo Vista Controlador.	30
Fig. 12. Caso de uso general del sistema.	31
Fig. 13. Caso de uso Ver reporte.	32
Fig. 14. Diagrama de clases.	33
Fig. 15. Diagrama de flujo de proceso del sistema.	34
Fig. 16. Proceso de detección de personas y cálculo de distancias euclidianas.	35
Fig. 17. Diseño preliminar de la vista del módulo de aplicación.	36
Fig. 18. Interfaz módulo de aplicación.	36
Fig. 19. Diseño preliminar de la vista del módulo de reportes.	37
Fig. 20. Interfaz módulo de reportes.	37
Fig. 21. Configuración para dnn-CUDA.	41
Fig. 22. Configuración CUDA para aceleración gráfica y análisis matemático.	41
Fig. 23. Instalación global de OpenCV mediante VS-Community con las nuevas características.	42
Fig. 24. Verificación del uso de GPU.	42
Fig. 25. Características tarjeta gráfica instalada.	42
Fig. 26. Comprobación de GPU.	43
Fig. 27. Detección de objetos con YOLOv3.	43
Fig. 28. Condición para detectar personas.	44
Fig. 29. Detección solo de personas.	44

Fig. 30. Bounding-boxes y punto a utilizar para medir la distancia entre personas.	45
Fig. 31. Iteración del método <code>scipy.spatial.distance</code> con la métrica 'euclidean'.	46
Fig. 32. Ejemplo de salida de matriz simétrica.	46
Fig. 33. Directorio raíz del proyecto.	47
Fig. 34. Clase <code>MainWindow</code> de la ventana principal.	47
Fig. 35. Envío de señal mediante función <code>emit()</code> .	48
Fig. 36. Class <code>TimerMessageBox</code> .	49
Fig. 37. Función <code>datosdia()</code> .	50
Fig. 38. Superposición de gráficas mediante el método <code>twinx()</code> .	51
Fig. 39. Repositorio de la aplicación en github.	52
Fig. 40. Directorio raíz de la aplicación.	52
Fig. 41. Instalación de dependencias.	53
Fig. 42. Copia del acceso directo en el escritorio.	53
Fig. 43. Ejecución del sistema.	53
Fig. 44. Distancia en pixeles de personas no infractora.	54
Fig. 45. Distancia en pixeles de personas infractoras.	55
Fig. 46. Escena con falso positivo.	55
Fig. 47. Personas cumpliendo la distancia.	56
Fig. 48. Personas infractoras.	56
Fig. 49. Prueba 3 parte 1.	60
Fig. 50. Prueba 3 parte 2.	63

ÍNDICE DE TABLAS

ITEM	PÁGINA
Tabla 1. Beneficiarios del proyecto.	14
Tabla 2. Matriz de confusión.	14
Tabla 3 Requerimientos Funcionales Módulo de Aplicación.	28
Tabla 4 Requerimientos Funcionales Módulo de reportes.	29
Tabla 5 Requerimientos no funcionales del sistema.	29
Tabla 6. Caso de uso general.	31
Tabla 7. Caso de uso Ver reportes.	32
Tabla 8. Recursos de hardware.	38
Tabla 9. Recursos de software.	38
Tabla 10. Costo del Sistema de control de distanciamiento social.	39
Tabla 11. Eje x y eje y de las gráficas.	51
Tabla 12. Prueba de acceso a la aplicación.	57
Tabla 13. Prueba generación de reporte.	57
Tabla 14. Prueba variando nivel de confianza.	58
Tabla 15. Prueba de conversión de pixeles a centímetros.	59
Tabla 16. Información de la prueba 3 parte 1.	59
Tabla 17. Matriz de confusión prueba 3 parte 1.	61
Tabla 18. Información de la prueba 3 parte 2.	61
Tabla 19. Matriz de confusión prueba 3 parte 2.	62

LISTA DE ANEXOS

Nº	DESCRIPCIÓN	
	Anexo 1. Formato de entrevista.	72
	Anexo 2. Misión y Visión de la Escuela de Educación Básica “Lcda. Angélica Villón Lindao”	73
	Anexo 3. Reunión con directora de la escuela.	73
	Anexo 4. Presentación del sistema a la directora del plantel.	74
	Anexo 5. Patio de la escuela.	74
	Anexo 6. Muestra de fotogramas con flexómetro en varias posiciones.	75
	Anexo 7. Solicitud para el ingreso a la escuela.	76

INTRODUCCIÓN

El COVID-19 es una enfermedad infecciosa que presenta cuadros respiratorios leves, moderados y hasta graves, que afectan a personas mayores y a aquellas que posean enfermedades pre existentes. El virus se propaga principalmente a través de las gotículas de saliva o las secreciones nasales que se generan cuando una persona infectada tose o estornuda, por lo que es importante tomar precauciones al toser y estornudar, por ejemplo: tapándose la boca con el codo flexionado, mantener distancia social, usar mascarilla [1].

La sociedad está regresando gradualmente a algunas actividades que se habían interrumpido cuando empezó esta pandemia, sin embargo, es importante que se continúe con el distanciamiento social, protocolo que solo funcionaría si todos participamos [2]. En Ecuador, el Ministerio de Educación ejecutó el Plan Educativo COVID-19, que contempla 2 fases: Aprendemos juntos en casa y Juntos aprendemos y nos cuidamos cuya finalidad es el retorno progresivo de los estudiantes a las instituciones educativas [3].

En la actualidad, la IA con el uso de técnicas de visión por computadora y redes neuronales ha permitido el desarrollo de sistemas capaces de clasificar objetos dentro de una imagen o video. Distancia2 usa algoritmos de IA para analizar videos y verificar si la distancia es menor a 2 metros [4].

El presente trabajo propone el desarrollo de una aplicación de escritorio para entornos Windows capaz de analizar fotogramas y detectar personas por medio de la red convolucional YOLOv3, y, calcular la distancia que existe entre personas mediante la aceleración matemática de CUDA.

En el primer capítulo se detallan los antecedentes, la descripción del problema que tiene su origen tras la aparición del COVID-19, la metodología, las herramientas y técnicas utilizadas para el desarrollo de sistemas de IA definiendo, además, los objetivos a alcanzar y la justificación para el uso de aplicaciones que miden el índice de cumplimiento de distanciamiento social.

En el segundo capítulo se describe el marco contextual, conceptual y teórico, los requerimientos funcionales y no funcionales, para finalmente detallar los

componentes de la solución. Según la metodología de desarrollo incremental este proyecto está dividido en 4 etapas. La primera comprende la implementación del entorno de detecciones integrando CUDA y OpenCV. La detección se realiza capturando un fotograma que es pre procesado y enviado a la red neuronal para su clasificación.

La segunda etapa describe el cálculo de la distancia entre las personas detectadas para posteriormente determinar los infractores presentes en una imagen. El módulo de aplicación es la tercera etapa, en esta se incorpora el script Python de las detecciones con una aplicación de escritorio desarrollada en PyQt5. La última etapa corresponde a la visualización de reportes estadísticos a partir de los registros almacenados en la base de datos.

CAPÍTULO I

1. FUNDAMENTACIÓN

1.1. Antecedentes

Con la aparición del COVID-19 a finales del 2019 y al conocer la repercusión que este virus tiene dentro de la salud de las personas, los gobiernos a nivel mundial han tomado medidas para evitar su propagación. La Organización Mundial de la Salud (OMS) indica que el virus se transmite principalmente de persona a persona a través de las gotículas respiratorias y por contacto. Por esta razón los cuidados que se deben tener para evitar el contagio del virus son: uso de mascarilla, evitar el contacto de la cara con las manos, desinfectar superficies de uso frecuente, lavado de manos, distanciamiento social de 2 metros y quedarse en casa si presenta síntomas [5].

Desde el inicio de la pandemia, los niños, niñas y adolescentes (NNA) han sido afectados por el cierre de sus escuelas y por el desarrollo de actividades académicas virtuales, sincrónicas y asincrónicas. El 8 de junio, El Fondo de las Naciones Unidas para la infancia (UNICEF) informó que más de 70 países planeaban la apertura de escuelas y, a finales de junio, millones de estudiantes habían regresado a las aulas[6]. Al principio se manejaba la hipótesis de que los NNA eran asintomáticos o presentaban síntomas leves, sin embargo, en los últimos meses se ha identificado que un número reducido a desarrollado una respuesta inflamatoria sistémica que comparte características comunes con otras afecciones respiratorias pediátricas que necesitan hospitalización [7].

En Ecuador, 126 niños y adolescentes menores de 19 años fueron reportados como casos sospechosos de Síndrome Inflamatorio Multisistémico (SIM) hasta noviembre del 2020 [8], lo que preocupa a la comunidad educativa por ser un posible foco de contagio. Por tal motivo, el retorno a clases presenciales se ha aplazado por varias ocasiones mientras se ejecuta un plan de inoculación efectivo que, involucre no sólo a docentes sino a toda la comunidad educativa del país.

La Escuela de Educación Básica Lcda. Angélica Villón Lindao, se encuentra ubicada en la calle 18 de agosto entre Paquisha y Fausto Fajardo del cantón Santa Elena, provincia de Santa Elena. Fue fundada el 12 de septiembre del 2000 por la profesora a la que se atribuye su nombre. Cuenta con una oferta académica que va desde educación inicial hasta educación general básica, teniendo inglés como segunda lengua en todos sus niveles.

En la actualidad tiene como directora a la Lcda. Ibelice Tomalá V. y Coordinador Administrativo al Arq. Mauricio Yunda. En 20 años sigue ofreciendo una educación de calidad basada en el derecho de todos y todas a recibir una enseñanza que promueva el aprendizaje durante toda la vida, que a pesar de los cambios que trajo la pandemia se empeña en brindar atención a los grupos marginados y vulnerables, procurando el desarrollo académico también de manera virtual.

La Lcda. Ibelice Tomalá en la entrevista (Anexo 1), manifestó que la administración del establecimiento educativo cambia a razón de 2 años, además mencionó que con la aparición del COVID-19 han tenido que afrontar nuevos retos a nivel institucional. Por tal motivo, los directivos actuales y personal docente están desarrollando un plan para el regreso a clases presenciales alineados a los protocolos del Ministerio de Educación [9].

Las Unidades Educativas privadas y fiscales del Ecuador no cuentan con un sistema que permita el control del distanciamiento social, medida de bioseguridad dispuesta por las autoridades de Gobierno y que se contempla en, “Protocolos para el autocuidado e higiene de la población educativa durante el uso progresivo de las instalaciones educativas COVID-19” [9]. Por lo tanto, el cumplimiento de esta medida es fundamental para el cuidado interpersonal, evitando el contacto con personas infectadas.

Por lo antes mencionado, alrededor del mundo se están implementando sistemas que ayuden al control del distanciamiento físico. Soccial Distance es un sistema de localización de alta precisión para entornos cerrados que permite controlar el aforo

en tiempo real además de notificar a la persona si está a menos de 2 metros distancia. Este sistema es parte del proyecto piloto Aulas COVID Free que se está implementando en el colegio privado Liceo Hispano del municipio de Valencia, España[10]. No obstante, este proyecto necesita el uso de tags en forma de tarjetas o pequeños wearables y localizadores, tecnología que difícilmente podría ser adquirida por escuelas locales.

Distancia2, es una plataforma de monitoreo que contiene un módulo que utiliza inteligencia artificial para medir la distancia entre personas utilizando cámaras de vigilancia. Los datos generados son enviados a una aplicación web, donde mediante algoritmos de inteligencia artificial analizan los videos y verifican si la distancia es menor a 2 metros, enviando una señal de alerta a la plataforma que actualiza los mapas de calor en el panel de control en tiempo real [4]. Sin embargo, este proyecto necesita de conexión a internet mientras la solución planteada en esta tesis trabaja localmente.

En el siguiente trabajo [11], el Laboratorio de Investigación en Inteligencia Artificial Industrial de la Espol y TIA S.A. desarrollaron el sistema de estimación del cumplimiento del distanciamiento físico que utiliza visión por computadora e inteligencia artificial para detectar a las personas presentes en un área definida a partir de señales de videos. Esta información luego es reportada en tiempo real y puede ser utilizada para analizar videos de lugares públicos, centros comerciales, fábricas y otro tipo de establecimientos.

Finalmente, en base a los trabajos antes expuestos, se ve la importancia de contar con un sistema a nivel local dentro de los establecimientos educativos que mide el índice de cumplimiento del distanciamiento social y salvaguardar la salud de las personas evitando así la propagación del virus que desde su aparición a afectado al mundo entero.

1.2. Descripción del proyecto

Debido a la necesidad de un sistema que mida el índice de cumplimiento de distanciamiento social dentro de la Escuela de Educación Básica Lcda. Angélica

Villón Lindao (Anexo 1), se propone la implementación de un sistema de detección autónomo para la visualización de video en tiempo real de los alumnos y la interacción de ellos en el patio del plantel. El sistema tuvo un módulo de aplicación y uno de reporte que sirvió de apoyo para medir el cumplimiento de este protocolo.

Los recursos físicos mínimos para el desarrollo e implementación de este proyecto fueron: procesador de sexta generación o superior, 8GB de memoria RAM o superior, webcam de 720p con autoenfoco, una tarjeta gráfica Nvidia GT710 o superior que ayudo en la aceleración del renderizado, procesamiento de los datos y cálculos matemáticos.

Este proyecto estuvo dividido en 4 etapas, en cada una de ellas se detalló las herramientas necesarias y su función dentro de la solución del proyecto.

Etapa 1: Implementación del entorno de detecciones

Se creó un entorno global de desarrollo para las herramientas de aceleración gráfica, análisis matemáticos y renderizado de video. La herramienta principal para este propósito fué CUDA que es un entorno de desarrollo creado por NVidia para realizar proyectos basados en IA y visión por computadora. Este requiere de un componente de compatibilidad para tarjetas gráficas NVidia llamado cuDNN. Todos estos componentes se los descargó de los sitios web oficiales de la marca, teniendo en cuenta que para este proyecto se usó una unidad de procesamiento gráfico o GPU NVidia GeForce GT710 con compilación 3.5.

En la creación del entorno global se utilizó Visual Studio Community, CMake y Opencv-contrib, este último en su versión para GPU. Para instalar OpenCV, se abrió su carpeta contendora en CMake, allí se configuró las características para GPU, aceleración gráfica y de video. Finalmente, al abrir Visual Studio se depuró y compiló la carpeta CMakeTargets que contiene las nuevas configuraciones de OpenCV. Posteriormente, estas se instalaron automáticamente dentro del SO.

El siguiente paso dentro de esta etapa y uno de los más relevantes en este proyecto es la detección de objetos usando análisis con GPU, donde se comprobó la vinculación entre OpenCV y CUDA mediante un pequeño script en el CMD de Windows. Dentro del intérprete de Python se importó el módulo cv2 y usando las funciones *getDevice()* y *printCudaDeviceInfo()* de la clase cuda se imprimió en consola toda la información de la tarjeta gráfica instalada.

Para la primera detección de objetos se descargó los pesos de YOLOv3, las configuraciones y etiquetas de COCO. Se realizó un script en Python para la captura de video con visión por computadora, luego se separaron los fotogramas y estos se analizaron con la red convolucional de YOLO, que dentro de un bucle detectó y devolvió coordenadas. Las coordenadas sirvieron para ubicar y enmarcar cada objeto dentro de los fotogramas del video. YOLO cuenta con 80 clases y para este proyecto se configuró el uso de la clase persona.

Etapa 2: Implementación del código de distanciamiento

Los datos que devolvió Yolov3 fueron el centro en x, el centro en y, el alto y el ancho, estos valores sirvieron para el cálculo de las coordenadas de los pies de cada persona y posterior cálculo de la distancia entre personas. Para medir la distancia de separación es necesario al menos 2 puntos que serán el inicio y fin del vector trazado, donde la longitud de ese vector se conoce como distancia euclidiana [12]. Se tiene 2 puntos bidimensionales, $A(x_1, y_1)$ y $B(x_2, y_2)$, la distancia entre A y B está definida por (Fig. 1):

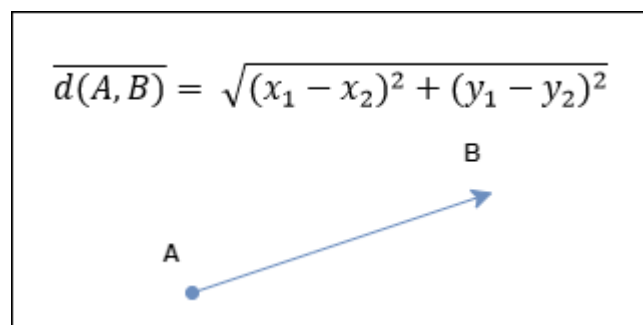

$$\overline{d(A, B)} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Fig. 1. Fórmula distancia euclidiana [12].

La fórmula que se muestra en la Fig. 1 es el principio que utiliza el módulo *scipy.spatial.distance* [13] para el cálculo de distancia entre pares de puntos. Estos valores de distancia se compararon con una distancia referencial en píxeles para saber si la persona está o no infringiendo la distancia permitida. Para la calibración de la distancia referencial se hizo pruebas de medición de longitud entre objetos de forma manual con ayuda de un flexómetro que permitió la aproximación de metros a píxeles.

Si la distancia es menor a 1.5 metros, el infractor es enmarcado de color rojo caso contrario será enmarcado de color verde. El día y la hora, el número de infractores son datos que se guardaron en una base de datos local SQLite.

Etapa 3: Implementación del módulo de aplicación

El módulo de aplicación se construyó en PyQt5 que es entorno de trabajo que permite la creación de interfaces gráficas para entornos Windows. La ventana principal contiene 2 pestañas:

- Pestaña de video que muestra la renderización del video en tiempo real por medio de señales que actualizan los fotogramas dentro de una etiqueta.
- Pestaña de reportes que se detalla en la Etapa 4 de este proyecto.

La función de detección y cálculo de distancias euclidianas se encuentra embebida dentro del código principal, además existe otra señal que permite guardar los registros de las detecciones en una base SQLite. Adicionalmente, en la pestaña de video se muestra el número de personas detectadas, el número de personas incumplidas y las que cumplen el distanciamiento.

Etapa 4: Implementación del módulo de reportes

Este módulo permite la visualización de los datos en forma de diagramas estadísticos mediante Matplotlib que hace consultas a la base de datos y genera reportes estadísticos de histograma de barras y gráfico de líneas dependiendo de una fecha seleccionada, además se muestra el número de detecciones guardadas y el total de infractores hasta la hora de la última detección guardada.

Debido a que este sistema usa técnicas de visión por computadora, se realizaron validaciones de conexión de cámara que permiten inicio del sistema si y solo si existe un cámara disponible.

Para el desarrollo de este proyecto se usaron las siguientes tecnologías:

- **Python 3.8:** Lenguaje de programación [14].
- **PyQT 5:** Para la creación de la interfaz del módulo de aplicación y reportes [15].
- **OpenCV 4.2:** Biblioteca para el análisis y procesamiento de imágenes y videos [16].
- **SQLite 3:** Motor de base de datos [17].
- **YOLOv3:** Método de detección de objetos basado en *Darknet-53* [18].
- **Scipy:** Librería de Python para análisis matemáticos, cálculo de distancia eucladiana [19].

Este proyecto recae en las siguientes líneas de investigación: Desarrollo de software y, Tecnología y Gestión de la Información [20]. Por lo tanto, la implementación del sistema de distanciamiento social, me permitió monitorear mediante video en tiempo real la distancia entre personas dentro del patio de la Escuela de Educación Básica Lcda. Angélica Villón Lindao del cantón Santa Elena. Finalmente, estos datos permitirán a los directivos del plantel educativo tomar medidas correctivas ante el desacato al protocolo de distanciamiento social.

1.3. Objetivos del proyecto

1.3.1. Objetivo general

Implementar un sistema de monitoreo mediante técnicas de inteligencia artificial y visión por computadora para el análisis del índice de cumplimiento del distanciamiento social de los estudiantes en el patio de EEB Lcda. Angélica Villón Lindao.

1.3.2. Objetivos Específicos

- Implementar el script en lenguaje python para detección de personas con YOLOv3.

- Determinar el valor de conversión de píxeles a centímetros por medio de pruebas de medición con distancia referencial de 150 centímetros.
- Generar reporte del número de infractores detectados en un día de clases.

1.4. Justificación

El distanciamiento social es un factor recurrente entre las medidas de bioseguridad, por el elevado riesgo de que la velocidad de transmisión se incremente de manera descontrolada y que la capacidad de los servicios de salud no es suficiente para atender los casos que requieran hospitalización, especialmente en la Unidad de Cuidados Intensivos (UCI), con el consecuente incremento de la letalidad. Intensificar la comunicación del riesgo e implementar comunicación de crisis permite ampliar el alcance de las medidas de promoción de la salud [21].

El Ministerio de Educación inició la verificación de condiciones para el retorno opcional a clases en aproximadamente 2.071 instituciones educativas unidocentes y bidocentes que tienen hasta 35 estudiantes, de régimen Costa-Galápagos. Para lograr este proceso, esta cartera de Estado cuenta con los protocolos de retorno seguro a las instituciones educativas, con base a la identificación de la realidad de cada territorio y trabajando de la mano con el Comité de Operaciones de Emergencia(COE) Nacional y los COEs Cantonales [22] .

Bajo este contexto se encuentra el Plan de “continuidad educativa, permanencia escolar y uso progresivo de las instalaciones educativas” - Juntos aprendemos y nos cuidamos, cuyo objetivo primordial es sostener el proceso educativo y donde todos los ciudadanos ecuatorianos debemos apoyar las medidas ante la emergencia sanitaria que atraviesa el mundo. En esta fase encontramos los “Protocolo para el autocuidado e higiene de la población educativa durante el uso progresivo de las instalaciones educativas COVID-19” [9].

La finalidad de la visión por computador es la de modelar y automatizar procesos de reconocimiento visual, es decir, distinguir entre objetos [23].

La escuela de Educación Básica Lcda. Angélica Villón Lindao, siguiendo las directrices dispuestas por las autoridades de gobierno está implementando medidas de bioseguridad siendo el distanciamiento social una de las más importantes. El protocolo de bioseguridad de distanciamiento social para los alumnos, personal docente y administrativo establece una distancia mínima de 1,5 metros entre personas. Ante la necesidad de un control más eficiente en el patio del plantel, este proyecto implementa un sistema de monitorización de video que ayuda a medir el índice de cumplimiento de la distancia entre personas en tiempo real.

Esta aplicación se desplegó en entorno de escritorio y permite ver el número de personas en el interior del patio, el número de infractores detectados y el total de infractores en un día. Además, se puede visualizar en un histograma el número de infractores en relación al tiempo transcurrido. El video enmarca en color verde a las personas que cumplan la normativa y en rojo a las personas que infrinjan la medida.

El sistema es autónomo, es decir, realiza las detecciones y guarda los datos en una base de datos local sin necesidad de un operador. Los datos guardados generan un reporte donde se detalla el número de infractores totales dependiendo del día seleccionado. Toda esta información permitió a las autoridades del plantel tomar decisiones con respecto al comportamiento de los alumnos, personal docente y administrativo en el patio del plantel.

Este proyecto está alineado al Plan Nacional de Desarrollo “Toda una vida”, especificado en los siguientes ejes:

Eje 1.- Derechos para Todos Durante Toda la Vida

Objetivo 1.- Garantizar una vida digna con iguales oportunidades para todas las personas [16].

Política 1.15.- Promover el uso y el disfrute de un hábitat seguro, que permita el acceso equitativo a los espacios públicos con enfoque inclusivo [24].

Eje 2.- Economía al servicio de la sociedad.

Objetivo 5.- Impulsar la productividad y competitividad para el crecimiento económico sostenible de manera redistributiva y solidaria [24].

Política 5.6.- Promover la investigación, la formación, la capacitación, el desarrollo y la transferencia tecnológica, la innovación y el emprendimiento, la protección de la propiedad intelectual, para impulsar el cambio de la matriz productiva mediante la vinculación entre el sector público, productivo y las universidades [24].

1.5. Metodología del proyecto

1.5.1. Metodología de la investigación

Para el desarrollo de este proyecto se realizó una investigación de tipo exploratoria [25] mediante la revisión de trabajos, estudios y bibliografía que mostró evidencia sobre el uso de la Inteligencia Artificial para el cumplimiento de protocolos de bioseguridad en la actualidad. En [26] se enfatiza el uso de modelos de Inteligencia Artificial entrenados para la detección de objetos en el desarrollo de múltiples tareas tecnológicas que requieren seguimiento constante. Los sistemas inteligentes facilitan la obtención de datos relevantes para la toma de decisiones sanitarias [27]. Estos trabajos contribuyeron al análisis, desarrollo y despliegue de este proyecto.

La Metodología de investigación diagnóstica [25] se utilizó mediante una entrevista a la directora del centro de estudios donde se detalló cómo se realizará el control del distanciamiento social en los estudiantes (Anexo 1). La entrevista, de tipo directa recolectó información sobre temas como: el número de estudiantes, el plan de control de medidas de bioseguridad, eventos dentro de la escuela y soluciones tecnológicas.

1.5.2. Análisis de la entrevista

En relación con la entrevista (Anexo 1), el instrumento fue aplicado a la directora Lcda. Ibelice Tomalá con el propósito de detallar actividades, funcionamiento del plantel, su estructura organizacional y las medidas que tomarán para el regreso progresivo a las clases presenciales. Por tal motivo, los directivos, personal docente y administrativos de la escuela, han empezado a trabajar en el Plan Institucional de

Continuidad Educativa, Permanencia Escolar y uso progresivo de las Instalaciones Educativas (PICE)[28] para que la comunidad educativa entre en la Fase 2 de este proceso y tengan la autorización de alternar la educación virtual con la presencial.

Durante la entrevista se conoció que:

- La administración cambia a razón de 2 años.
- La escuela posee 2 jornadas de clases.
- Debido a la pandemia tuvieron que reducir su planta de docentes por la disminución de alumnos.

También se determinó que entre las disposiciones que tomarán ante el regreso a clases presenciales están:

- Distancia permitida entre estudiantes y docentes 1.5m.
- Los estudiantes deben portar mascarilla durante su permanencia en el plantel.
- Al ingreso del plantel y de los salones de clases se desinfectará las manos de los estudiantes con alcohol antibacterial.
- La jornada de clases estará dividida en matutina y vespertina.
- El retorno de los estudiantes a clases presenciales es voluntario.
- Queda prohibida cualquier actividad colectiva dentro de la institución.

1.5.3. Variable

La variable del proyecto corresponde a el número de infractores detectados durante un día de clases. Esta variable estará sujeta a factores como: la luminosidad del entorno, el ángulo de cobertura de la cámara, el enfoque de la cámara y su perspectiva respecto a la superficie que se va a medir.

1.5.4. Beneficiarios del proyecto

La solución tecnológica de este proyecto beneficiará directa e indirecta a la población de la Escuela de Educación Básica Lcda. Angélica Villón Lindao detallada a continuación:

BENEFICIARIOS DIRECTOS	
Directivos	5
BENEFICIARIOS INDIRECTOS	
Estudiantes	160
Docentes	12
Total	177

Tabla 1. Beneficiarios del proyecto [29].

1.5.5. Metodología experimental

Con la finalidad de analizar la objetividad de los resultados en los procesos de reconocimiento de imágenes, detección de personas y distancia entre ellas se efectuó una comparación entre los datos obtenidos y los observados mediante la verdad fundamental [30]. Esos valores fueron utilizados para evaluar la exactitud del sistema en clasificar a una “persona infractora” y a una “persona no infractora” por medio de la creación de la matriz de confusión [31].

		PREDICCIÓN	
		POSITIVO	NEGATIVO
REALIDAD	POSITIVO	VERDADERO POSITIVO(VP)	FALSO NEGATIVO(FN)
	NEGATIVO	FALSO POSITIVO(FP)	VERDADERO NEGATIVO(VN)

Tabla 2. Matriz de confusión.

En la matriz de confusión (Tabla 2Tabla 2. Matriz de confusión.) se considera como:

- VP: la cantidad de “infractor” que el sistema clasificó como “infractor”.
- FN: la cantidad de “infractor” que el sistema clasificó como “no infractor”.
- FP: la cantidad de “no infractor” que el sistema clasificó como “infractor”.
- VN: la cantidad de “no infractor” que el sistema clasificó como “no infractor”.

Finalmente, esta tabla sirvió para el cálculo de la:

- Exactitud(accuracy): porcentaje de predicciones que fueron correctas.

- Precisión: porcentaje de exactitud en los resultados de la predicción.
- Sensibilidad(recall): efectividad para identificar etiquetas positivas.
- Puntaje de referencia(F-Score): combinación entre precisión y sensibilidad.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

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

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

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Fig. 2. Métricas de la matriz de confusión [30].

1.5.6. Metodología de desarrollo

Uno de los valores fundamentales para el éxito de los proyectos es la información la cual corresponde a: objetivos, estructura, organización, planificación, presupuesto, comunicación, y que toda esta información esté actualizada en todo instante. Por tal razón, es necesario tener en cuenta los tres grandes componentes de un proyecto: Procesos, Personas, y Productos (genéricamente denominadas las 3Ps) [32].

Para ese propósito, se utilizó una adaptación de la metodología de desarrollo incremental debido a que esta combina elementos del modelo en cascada, que es aplicada para trabajos de Redes Neuronales Artificiales [33], esto permite que el software vaya evolucionando en funcionalidad desde su primera versión hasta que cumpla con los requerimientos al inicio planteados.

Una vez que se definen los incrementos del modelo (Fig. 3), se debe organizar cada incremento para que las funciones y requerimientos no afecte la funcionalidad del incremento anterior [34].

Los incrementos para este proyecto son:

Incremento 1

En esta parte se crea el entorno y se integran las librerías necesarias para la detección de personas.

Incremento 2

En esta etapa se desarrolla el procedimiento para calcular la distancia entre personas en tiempo real. Este procedimiento se integrará al módulo de detección de personas.

Incremento 3

En esta etapa se desarrolla el módulo de aplicación y a su vez se integra el módulo de detección de personas.

Incremento 4

En esta última etapa se desarrolla el módulo de reportes y se integra al de aplicación, también, se realizarán pruebas a fin de corregir errores.

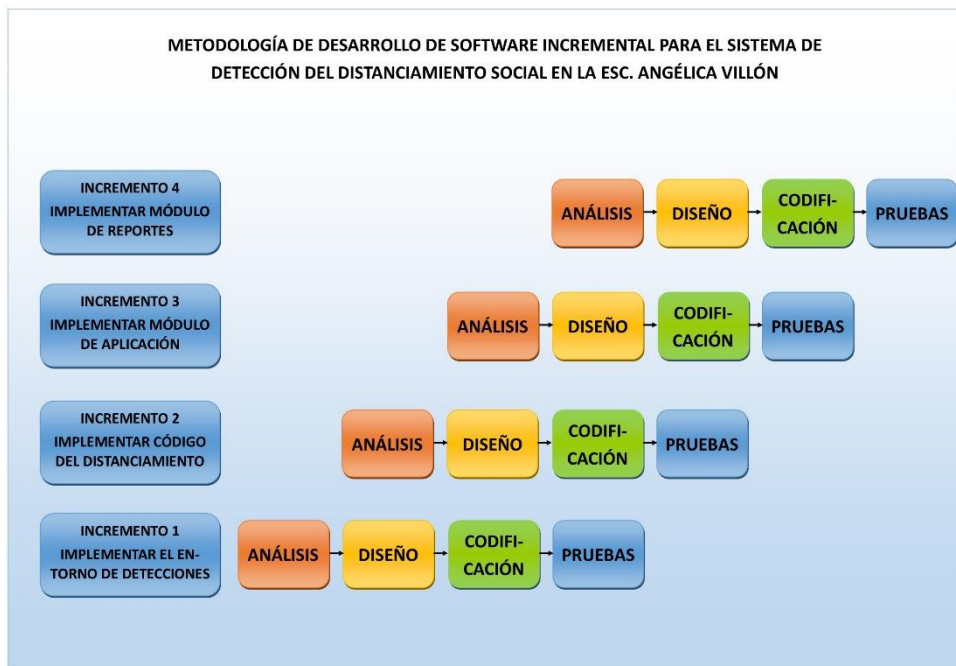


Fig. 3. Modelo Incremental del software

CAPITULO II

2. La propuesta

2.1. Marco Contextual

2.1.1. Escuela de Educación Básica “Lcda. Angélica Villón Lindao”

La escuela de Educación Básica “Lcda. Angélica Villón Lindao” fue creada el 12 de septiembre del 2000 por la profesora a quien se le atribuye su nombre. Luego de su fallecimiento el 11 de noviembre del 2008, la escuela quedó siendo administrada por sus hijos y familiares más cercanos. La escuela está ubicada en Santa Elena, calle 18 de agosto entre Paquisha y Fausto Fajardo (Anexo 1).

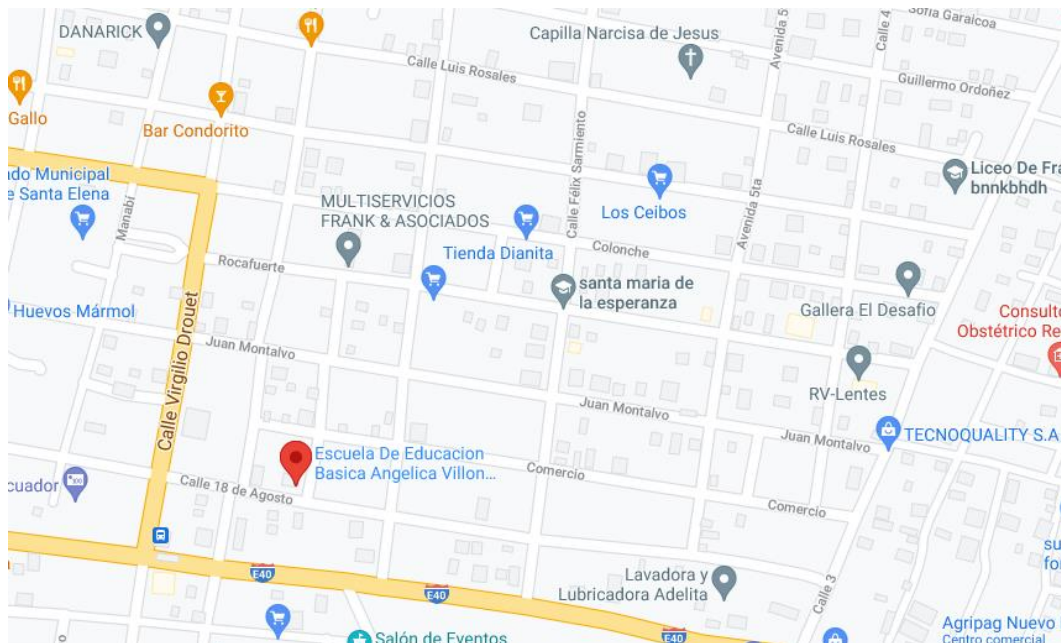


Fig. 4. Ubicación Geográfica de la escuela [35]

Misión

La Escuela de Educación Básica “Lcda. Angélica Villón Lindao” ofrece un conjunto de talentos humanos técnicos y pedagógicos al servicio de la educación, desarrollando procesos educativos para los estudiantes con o sin necesidades educativas especiales atendiendo con absoluta responsabilidad y equidad, sin

discriminación alguna, sirviendo a la comunidad con un enfoque en derechos (Anexo 2).

Visión

La Escuela de Educación Básica “Lcda. Angélica Villón Lindao” anhela brindar una educación actualizada acorde a las exigencias que la sociedad actual demanda, con la atención y diversidad, con docentes capacitados comprometidos al cambio, para enrumbar a la formación de talentos humanos que se inserten con facilidad de estudios y a la sociedad, que adquieran un pensamiento crítico reflexivo a través de la investigación científica con una práctica permanente de valores éticos y morales por el medio de la inclusión permitiendo a las personas con necesidades educativas especiales o sin discapacidad ser parte de la comunidad educativa (Anexo 2).

2.1.2. Lineamientos para la aplicación del plan de continuidad educativa, permanencia escolar y uso progresivo de las instalaciones educativas.

En [36] se indica que:

Capítulo 2: Fases del proceso educativo en el contexto de la emergencia sanitaria

Artículo 6.- Fases del proceso general. - El proceso educativo que se aplicará en el sistema educativo ecuatoriano, durante e inmediatamente después de que se declare como superada la emergencia sanitaria en el país producida por COVID-19, está compuesto por tres fases:

- a. Fase 1: Aprendamos juntos en casa.
- b. Fase 2: Juntos aprendemos y nos cuidamos.
- c. Fase 3: Todos de regreso a la escuela.

Artículo 9.- Fase 2: Juntos aprendemos y nos cuidamos. – Es la etapa en la que se garantiza la continuidad educativa y la permanencia escolar de todos los niños, niñas, adolescentes, jóvenes y adultos que forman parte del sistema educativo ecuatoriano, y que considera como recurso el uso progresivo y alternado del estudio desde casa y el uso de las instalaciones educativas con los mecanismos de educación no presencial.

Artículo 12.- Aprobación del uso progresivo de las instalaciones educativas. – Cualquier institución educativa podrá voluntariamente solicitar, bajo la figura de proyecto piloto, la autorización del uso progresivo de las instalaciones educativas a la autoridad territorial o COE competente. Para esto deberá cumplir las condiciones detalladas en el presente instrumento.

Artículo 13.- Condiciones generales para la aprobación del uso progresivo de las instalaciones educativas

- c. Todas las instituciones educativas, independientemente de su sostenimiento, oferta o modalidad deberán aplicar el Protocolo de Autocuidado e Higiene realizado por el órgano competente.

2.1.3. Protocolo para el autocuidado e higiene de la población educativa durante el uso progresivo de las instalaciones educativas COVID-19

Este instrumento emite los lineamientos y directrices para el estricto cumplimiento en las instituciones educativas unidocentes, bidocentes y multigrado del sistema educativo básica y media a nivel nacional de todos los sostenimientos, modalidades y ofertas educativas, que hagan uso progresivo de las instalaciones educativas durante la emergencia sanitaria en Ecuador por la pandemia COVID-19; según las disposiciones emitidas por el COE-N hasta la fecha de emisión del presente documento [37].

Cuidados que debemos tener [37]:

- Usar mascarillas es obligatorio.
- Lavar las manos con frecuencia.
- Evitar tocar el rostro con tus manos.
- Mantener 2 metros de distancia entre personas.
- Desinfectar áreas de uso frecuente.
- Quédate en casa si presentas síntomas.

2.2. Marco Conceptual

2.2.1. Python 3.8

Es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos. La elegante sintaxis de Python y su tipado dinámico, junto a su naturaleza interpretada lo convierten en un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en muchas áreas, para la mayoría de plataformas [9].

2.2.2. PyQt5

Es un conjunto completo de enlaces de Python para Qt v5. Se implementa con más de 35 módulos de extensión y permite que Python se utilice como un lenguaje de desarrollo de aplicaciones alternativo a C++ en todas las plataformas compatibles, incluidos iOS y Android [10].

2.2.3. OpenCV 4.2

Open Source Computer Vision Library, es una biblioteca de software de aprendizaje automático y visión artificial de código abierto. OpenCV fue construido para proporcionar una infraestructura común para aplicaciones de visión computarizada y para acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto con licencia BSD, OpenCV facilita a las empresas el aprovechamiento y la modificación del código [11].

2.2.4. Sqlite 3

Es una biblioteca de lenguaje C que implementa un motor de base de datos SQL pequeño, rápido, autónomo, de alta fiabilidad, con todas las funciones [12].

2.2.5. Scipy

Es un ecosistema de software de código abierto para matemáticas, ciencias e ingeniería y computación científica en Python que incluye herramientas generales y especializadas para la gestión de datos y la computación de alto rendimiento [14].

2.2.6. Entornos de desarrollo Integrado (IDE)

Los IDE son herramientas que permiten a los desarrolladores programar aplicaciones nuevas con rapidez, ya que no necesitan establecer ni integrar manualmente varias herramientas como parte del proceso de configuración [38]. A

pesar de que las aplicaciones pueden programarse con un editor y un compilador (a veces, con un depurador), pero, en entornos profesionales, casi siempre se utiliza un IDE. Entre las herramientas propias de un IDE se encuentra: editor de texto, compilador e intérprete, depurador y constructor de interfaces gráficas [39].

2.2.7. Visual Studio Community

Es un completo IDE extensible y gratuito que permite crear aplicaciones modernas para cualquier plataforma sea esta Windows, Android e iOS, además de aplicaciones web y servicios en la nube [40].

2.2.8. CUDA

Es una plataforma de computación paralela y modelo de programación desarrollado por NVIDIA para procesamiento gráfico (GPU), que ayuda a los desarrolladores a acelerar drásticamente las aplicaciones informáticas aprovechando la potencia de las GPU, donde la parte secuencial de la carga de trabajo se ejecuta en la CPU, que está optimizada para el rendimiento de un solo subproceso, mientras que la parte de proceso intensivo de la aplicación se ejecuta en miles de núcleos de GPU en paralelo. El kit de herramientas de CUDA incluye bibliotecas aceleradas por GPU, un compilador, herramientas de desarrollo y el tiempo de ejecución de CUDA [41].

2.2.9. Elementos a considerar para proyectos de visión por computadora

Para el desarrollo de aplicaciones que requieran el análisis de imágenes por computador se tienen disponibles un amplio conjunto de elementos que van desde las fuentes de luz necesarias para iluminar los objetos hasta los algoritmos encargados de clasificarlos [42].

- La *iluminación* es un factor relevante pues permite resaltar los aspectos del objeto de interés y garantiza que la cámara capte los objetos con la misma intensidad sin importar su ubicación dentro de la escena [42]. Una buena iluminación simplifica de manera considerable el análisis y posterior interpretación de la escena captada, mayor exactitud en las medidas para evitar resultados no deseados [43].
- La *cámara* es un dispositivo que permite captar, registrar y reproducir imágenes mediante la transformación de la luz en señales eléctricas [42]. Conocer características intrínsecas (ubicación de la cámara, altura, ángulo

de visión y sus especificaciones ópticas), y las características extrínsecas (rotación y traslación, que relacionan las coordenadas del mundo real con el sistema de coordenadas de la cámara) ayuda a su calibración[31].

- La imagen obtenida es enviada al *módulo de proceso y software* para su pre procesamiento y transformación [42].

2.2.10. Detección de objetos con OpenCV

Es una librería de software libre utilizada comúnmente para el reconocimiento y detección de objetos en la cual ofrece más de 2500 optimizados, entre los que se incluyen algoritmos de visión por computador y aprendizaje automático tanto clásicos como de última generación [44]. El medio de detección es por edge detection algorithm, que consiste en tomar una imagen por segmentos formando líneas, las cuales, son mucho más fáciles de analizar al momento del reconocimiento. Estas pueden ser procesadas de tal manera que, bajo un análisis matricial, se pueden detectar de manera sencilla los patrones que por los cuales la red neuronal debidamente entrenada intentará reconocer [26].



Fig. 5. Demostración del modelo edge detection [26].

Para este fin OpenCV cuenta con módulos que permiten la aplicación de técnicas de Deep learning de clasificación al momento de trabajar con imágenes. El módulo DNN (*Deep Neural Network module*) en la versión 3.3 de OpenCV añadió cambios que facilitan el uso de redes preentrenadas en tres sencillos pasos [44]:

1. Cargar el modelo de la red.
2. Preprocesar la imagen para adecuarla a la entrada de la red.
3. Enviar la imagen a la red y obtener las clasificaciones de salida.

2.2.11. Redes Neuronales Convolucionales (CNN)

Son redes neuronales muy efectivas para tareas de detección de objetos y clasificación y segmentación de imágenes. Tienen una arquitectura similar a las redes neuronales ordinarias, siendo la principal diferencia la aparición de capas especializadas en la extracción de características de manera automática [44]. Estas utilizan la operación de convolución, que es una operación matemática que transforma dos funciones en una tercera, la cual representa la magnitud de cuánto se superpone una función sobre la otra [45].

Estas redes están compuestas principalmente por capas convolucionales, capas de reducción (en inglés, pooling layer), y capas totalmente-conectadas (en inglés, fully-connected) [45]. Donde la primera aplica filtros de convolución a la imagen para reconocer determinados patrones, la segunda se encarga de reducir el tamaño de los datos y resumir las características resumidas y la última proporciona el resultado final de clasificación de imagen [44].

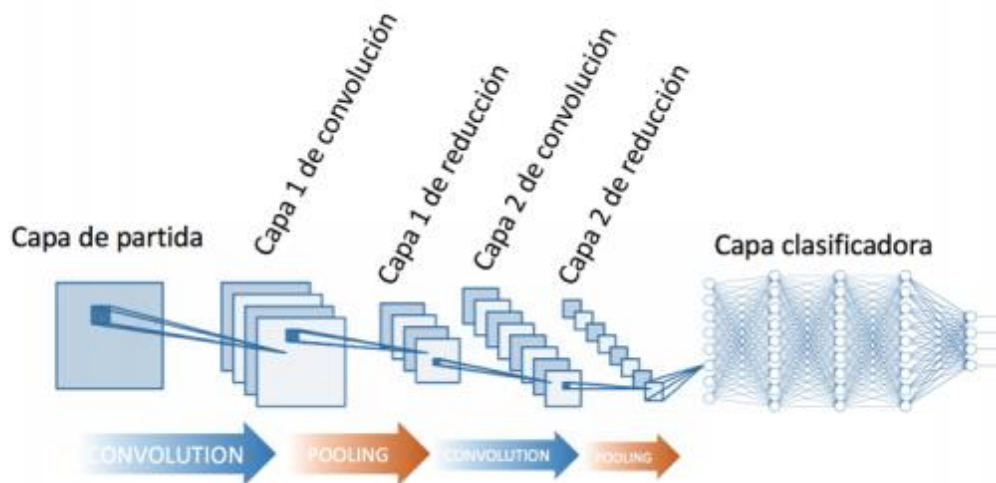


Fig. 6. Capas red neuronal convolucional [44].

2.2.12. Redes neuronales convolucionales basadas en regiones (R-CNN)

Las R-CNN se basan en buscar regiones de interés dentro de la imagen, para esto se buscan alrededor de 2000 regiones candidatas que puedan contener un objeto que serán redimensionadas y enviada posteriormente a la CNN para la extracción de las

características. Los datos extraídos son enviadas en forma de vector a las máquinas de vectores de soporte (SVM) que genera una salida que puntúa la probabilidad de pertenencia a una clase.

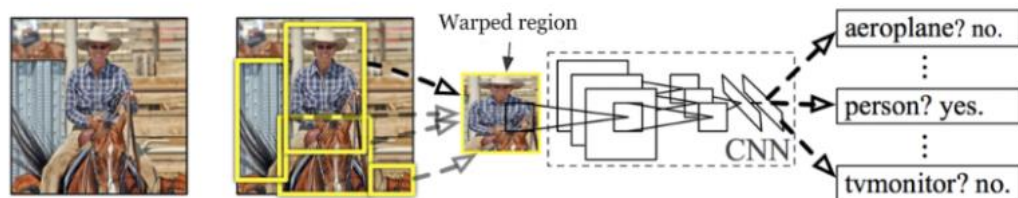


Fig. 7. Arquitectura R-CNN [44]

Finalmente, se escoge entre la mayor clase puntuada para etiquetar a la región candidata y proporcionar los *bounding box* de los objetos detectados en cada región.

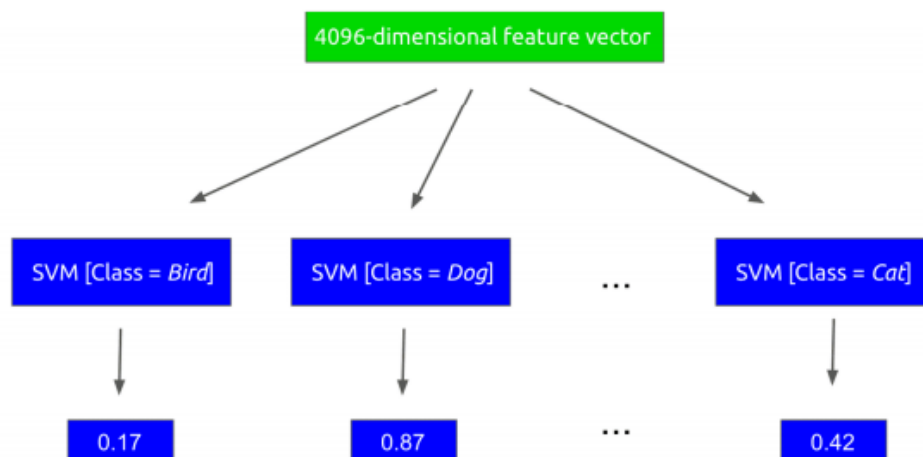


Fig. 8. Ejemplo salida SVM para una región candidata [44]

2.2.13. YOLOv3 y su arquitectura neuronal

You Only Look Once, es un sistema de detección de objetos en tiempo real de última generación [13] que utiliza una única red neuronal convolucional [45] de tipo R-CNN para “ver” a la imagen una sola vez permitiendo así ser rápido en comparación con otras redes [44]. YoloV3 es capaz de detectar y localizar múltiples objetos dentro de la imagen, logrando tener un elevado porcentaje de acierto sin restarle velocidad de detección, además de la integración fácilmente con OpenCV.

La arquitectura de YOLOv3 está compuesta por 53 capas convolucionales, razón por la cual recibe el nombre de Darknet-53. Cada capa convolucional es seguida de

una normalización de lote (en inglés, batch normalization) y la función de activación Leaky ReLU. No se utiliza ninguna capa de reducción, en su lugar, se utilizan capas convolucionales con paso=2. Con esto, se reduce la dimensionalidad del mapa de características, evitando perder las características de bajo nivel que se le atribuyen a las capas de reducción-máxima [45].

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig. 9. Arquitectura de Darknet-53 [45].

2.2.14. Detección y clasificación de objetos con YOLOv3

YOLO es un sistema que utiliza una red neuronal convolucional para la detección de objetos en tiempo real, este sistema aplica una única red neuronal a la imagen completa por esta razón es muy rápida. Para esto la red divide la imagen en regiones y predice múltiples cajas delimitadoras (en inglés, bounding box), y la probabilidad de detección de las clases de entrenamiento para cada caja delimitadora. Por último, utiliza un método de supresión de no-máximos para eliminar múltiples detecciones del mismo objeto. Como resultado, el sistema imprime los objetos que detectó, su confianza y el tiempo de ejecución [45].

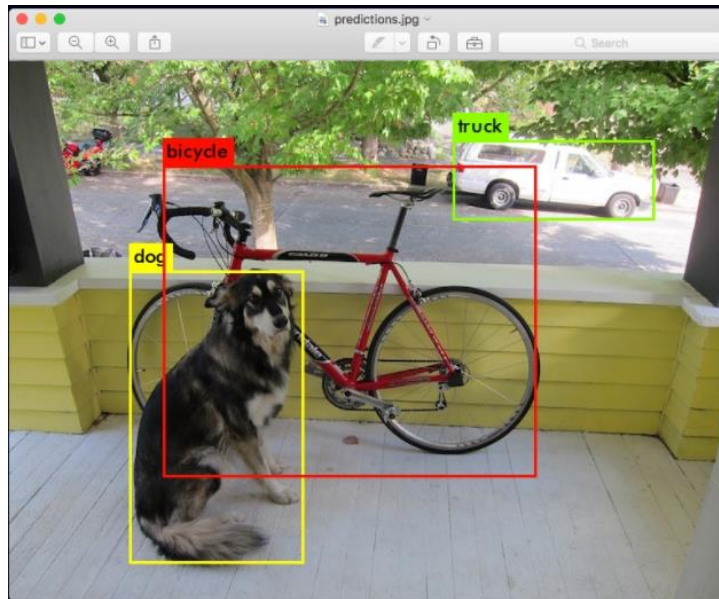


Fig. 10. Detección de objetos con Yolo [18].

2.3. Marco Teórico

2.3.1. La importancia de la Inteligencia Artificial (IA)

La inteligencia artificial plantea modelos computacionales de aprendizaje basado en redes neuronales biológicas humanas que permite el desarrollo de sistemas inteligentes con la capacidad para sentir, razonar, participar y aprender, procesando mayor cantidad de datos en menor tiempo. La IA abarca campos como el reconocimiento de voz, procesamiento del lenguaje natural, visión por computadora, robótica avanzada, captura de conocimiento, planificación y optimización [27].

La eficacia de la Inteligencia artificial se ve reflejada en el campo de la medicina, mejorando imágenes de partes médicas, detectando enfermedades, manejando imágenes microscópicas; en la industria orientado al control de calidad; robótica generando robots capaces de localizar objetos, reconociendo formas, evitando obstáculos y la interacción con las personas; vigilancia y seguridad para la detección de personas, identificando patrones de comportamiento y monitoreando espacios públicos [26].

En el artículo “Viabilidad de las redes neuronales en la detección de objetos” de la Revista Ibérica de Sistemas y Tecnologías de la Información, nos dice: “La inteligencia artificial correctamente entrenada es un medio óptimo de posibilidades

para la implementación en el seguimiento constante de procesos en la supervisión de seguridad en cámaras de vigilancia” [26].

2.3.2. Inteligencia Artificial frente al Covid-19

Los problemas sanitarios que surgen a partir de la aparición del Covid-19 y su repercusión sobre los sistemas de salud se vieron colapsados ante la creciente ola de contagios. Frente a estos problemas, la Inteligencia Artificial está ayudado a minimizar estos efectos a través del uso de técnicas de aprendizaje profundo cuya arquitectura se basa en redes neuronales artificiales, mediante el reconocimiento de imágenes [27].

El aprendizaje profundo se caracteriza porque emplea información gráfica para efectuar su análisis, es así que, los sistemas de reconocimiento de rostro empleados comúnmente en seguridad física, han sido modificados para detectar si la comunidad está cumpliendo la distancia social reglamentaria. Este software emplea cámaras de video estándar o aquellas dispuestas en una ciudad para video vigilancia, permitiendo monitorear el flujo peatonal en zonas críticas, realizando un reconocimiento sobre la distancia mínima e indicando una alerta a las autoridades si alguien no cumple con la norma [27].

2.4. Componentes de la propuesta

2.4.1. Módulos del Sistema

Módulo de Aplicación

En este módulo se podrá visualizar:

- Video en tiempo real.
- Clasificación y detección de personas.
- Número de personas detectadas.
- Número de personas infractoras.
- Número de personas que cumplen el distanciamiento.
- Persona que no cumple el distanciamiento enmarcada en rojo.
- Persona que cumple el distanciamiento enmarcada en verde.

Además, se lleva a cabo el cálculo interno de la distancia entre personas detectadas y se guardará en la base de datos local información como:

- Cantidad de personas.
- Cantidad de infractores.
- Distancia promedio de los infractores.

Módulo de reportes

En este módulo se lleva a cabo la consulta a la base de datos local que dependiendo de la fecha escogida permite la generación de los siguientes reportes:

- Gráfico de líneas (Distancia promedio vs tiempo) combinado con un histograma de barras (Número de Infractores vs tiempo).
- Gráfico de líneas (Aproximado de personas detectadas vs tiempo) combinado con un histograma de barras (Porcentaje de cumplimiento vs tiempo).

Además, en la ventana se muestra la distancia promedio de los infractores, el porcentaje de cumplimiento, la fecha y hora de la última detección guardada, el número de detecciones guardadas y el número de infractores totales.

2.4.2. Requerimientos Funcionales

Módulo de Aplicación

CODIGO	REQUERIMIENTO
RF-001	El sistema podrá mostrar el video en tiempo real de las detecciones de personas.
RF-002	El sistema mostrará el número de personas detectadas.
RF-003	El sistema mostrará el número de personas infractoras.
RF-004	El sistema mostrará el número de personas que cumplen la distancia.
RF-005	El sistema mostrará el porcentaje de predicción de la clase persona en cada detección.
RF-006	El sistema mostrará la persona detectada enmarcada en verde.
RF-007	El sistema mostrará la persona infractora enmarcada en rojo.
RF-008	El sistema mostrará un mensaje alertando cuando una detección es guardada.

Tabla 3 Requerimientos Funcionales Módulo de Aplicación

Módulo de reportes

CODIGO	REQUERIMIENTO
RF-007	El sistema permite seleccionar la fecha del reporte a mostrar.
RF-008	El sistema mostrará dos gráficos estadísticos combinando histograma de barras y diagrama de líneas.

RF-009	El sistema mostrará la distancia promedio de los infractores
RF-010	El sistema mostrará el porcentaje de cumplimiento del distanciamiento social.
RF-011	El sistema mostrará el número de infracciones por horas en forma de barras de color azul.
RF-012	El sistema mostrará el porcentaje de cumplimiento por horas en forma de barras de color verde.
RF-013	El sistema podrá ver la fecha y hora de la última detección guardada.
RF-014	El sistema mostrará el número total de detecciones.
RF-015	El sistema mostrará el número total de infractores.

Tabla 4 Requerimientos Funcionales Módulo de reportes

2.4.3. Requerimientos no Funcionales

CÓDIGO	ESPECIFICACIÓN
RNF-001	El sistema podrá detectar a una persona hasta 15m desde la posición de la cámara.
RFN-002	Los reportes se generarán a partir de la consulta a la base de datos.
RFN-003	El sistema hará captura de video por medio de una sola cámara.
RFN-004	El sistema fue desarrollado para plataformas Windows.
RFN-005	La captura de video se la realiza mediante una webcam de 720p con un rango de visión de 120° y autoenfoque.
RFN-006	El sistema solo detecta personas, discriminando otros objetos.

Tabla 5 Requerimientos no funcionales del sistema

EFICIENCIA

- El sistema podrá procesar como mínimo 5 fotogramas por segundo.
- El sistema debe ser capaz de operar adecuadamente sin necesidad de un operador.
- La información enviada a la base de datos podrá ser visualizada después de 5 segundos.
- La generación de reportes debe realizarse en menos de 5 segundos después de escoger la fecha.

SEGURIDAD

- La aplicación no cuenta con un sistema de autenticación.
- El sistema no tiene roles, debido que el sistema es autónomo y será supervisado por el director o inspector de la escuela.
- La aplicación guardará la información dentro de una base de datos local.

USABILIDAD

- El director o inspector de la escuela le tomará menos de 1 hora entender el funcionamiento de la aplicación.
- El sistema mostrará mensajes de error informativos y que el usuario final pueda entender.
- El sistema debe tener interfaces gráficas de fácil manipulación para el usuario final.

2.5. Componentes de la propuesta.

2.5.1. Arquitectura del sistema

Este sistema esta implementado con la arquitectura modelo vista controlador (MVC) que separa la parte gráfica de la aplicación, de los procesos lógicos y de los datos de la misma [46].

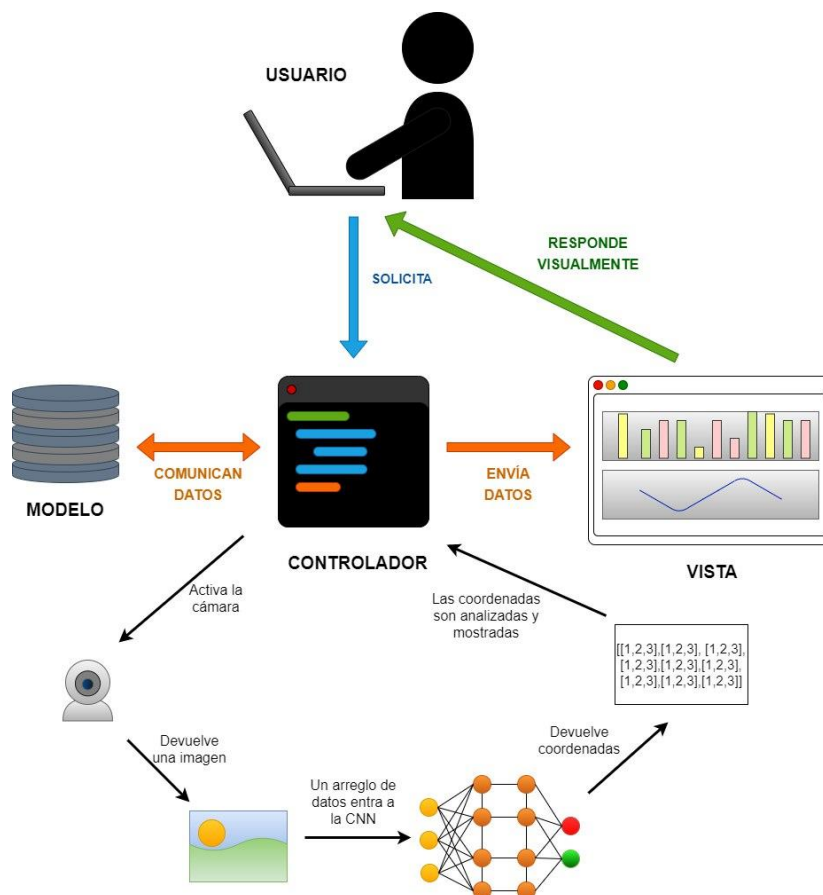


Fig. 11. Modelo Vista Controlador.

En Fig. 11 el MVC está compuesto por las siguientes elementos:

- Vista: El usuario solicita al sistema información al controlador.
- Controlador: Recibe las peticiones de la vista y se encarga de activar todas las funciones del sistema, entrega y recibe datos desde el modelo.
- Modelo: Manipula los datos debido a su conexión directa con la base de datos.

2.5.2. Diagrama de casos de uso

El diagrama de casos de uso captura la funcionalidad de un sistema, de un subsistema, o de una clase dividiéndola en transacciones significativas para el usuario final. Los usuarios del sistema se denominan actores y las particiones funcionales se conocen con el nombre de casos de uso [47].

Diagrama de caso de uso general

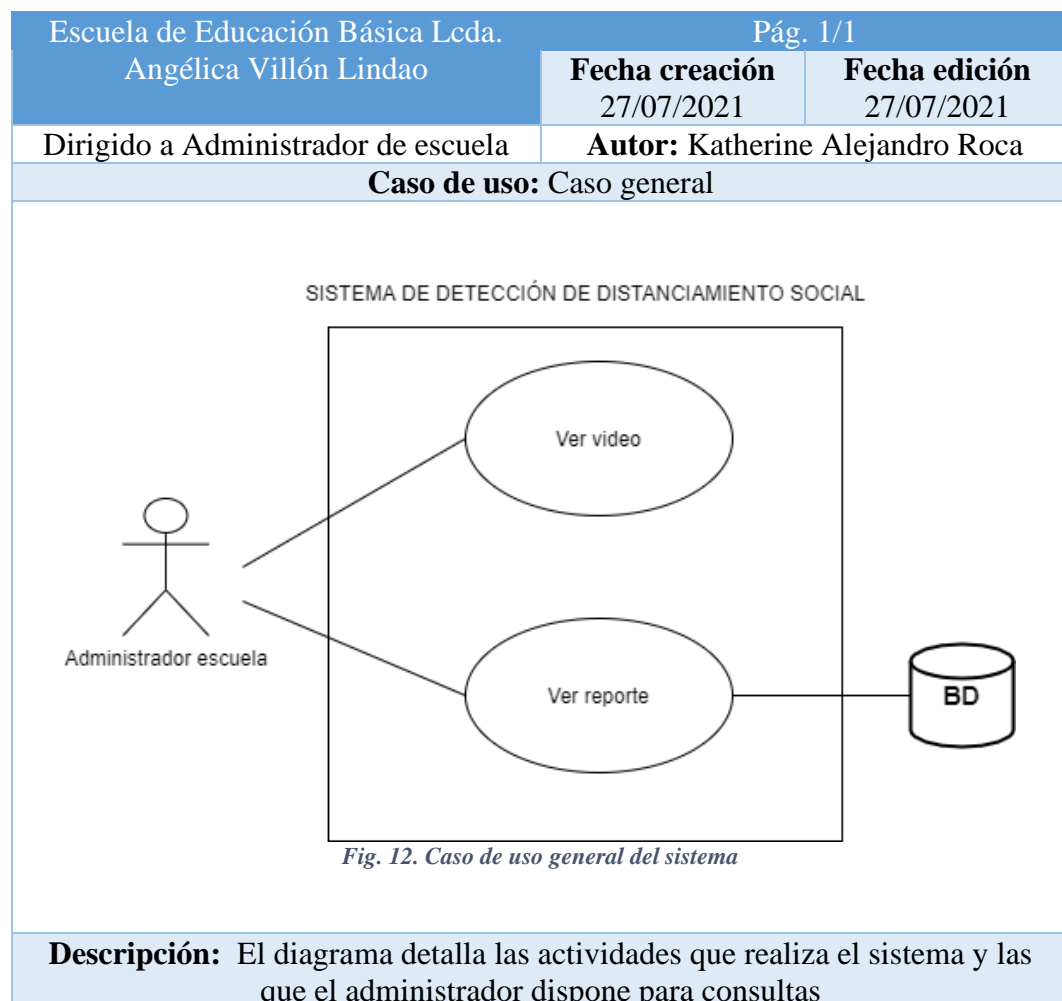


Tabla 6. Caso de uso general

Diagrama de caso de uso Ver reporte

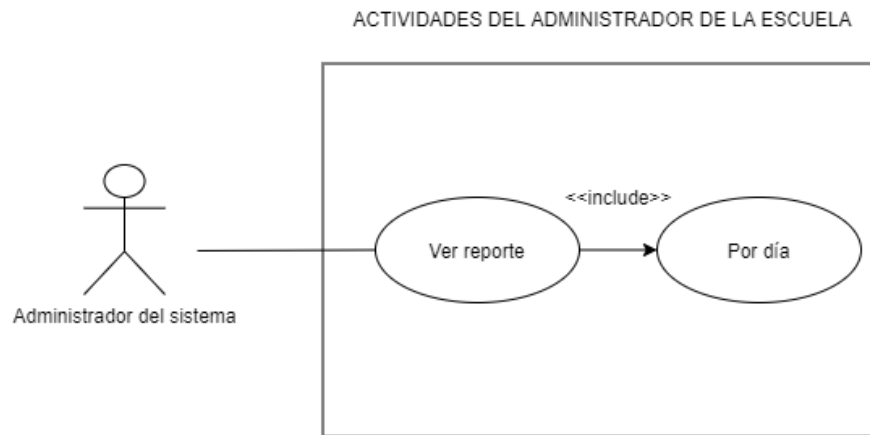


Fig. 13. Caso de uso Ver reporte

Caso de uso	Ver reportes
Actor	Administrador de la escuela
Descripción	Permite al administrador mostrar reportes que detallan el número de detecciones, porcentaje de cumplimiento, distancia promedio de los infractores.
Evento desencadenante	El administrador da clic en la pestaña Reportes
Pasos realizados	
1. El administrador inicia el programa. 2. Da clic en la pestaña Reportes. 3. Selecciona la fecha. 4. El administrador visualiza los reportes en pantalla.	
Pre-condiciones	Tener registros en la base de datos de detecciones guardadas.
Post-condiciones	Ninguna
Requerimientos cumplidos	Visualizar reportes

Tabla 7. Caso de uso Ver reportes

2.5.3. Diagrama de clases

Los diagramas de clase describen los tipos de objetos de un sistema, así como los distintos tipos de relaciones que pueden existir entre ellos. Los diagramas de clase

se convierten así en la técnica más potente para el modelado conceptual de un sistema software [48].

registro	
1	id: interger
2	incumplidos: interger
3	total: interger
4	dist_promedio: real
5	fecha: text

Fig. 14. Diagrama de clases

Los datos almacenados dentro de la base de datos SQLite, están detallados en la Fig. 14, donde:

- *id* es el identificador de la detección guardada siendo un valor entero único que no puede ser nulo y es auto incrementable.
- *incumplidos* es un valor entero que determina el número de personas infractoras detectadas en un registro.
- *total* es un valor entero que determina el número de personas detectadas en un registro.
- *dist_promedio* es un valor que puede ser entero o decimal que determina la distancia promedio de los infractores en un registro.
- *fecha* es un valor de tipo texto que determina la fecha y hora de un registro.

2.5.4. Diagrama de flujo de procesos

Diagrama de flujo del sistema

En Fig. 15 se detalla el flujo de procesos del sistema, que empieza validando la conexión de la cámara para iniciar el renderizado del video y el proceso de detección de personas infractoras para su posterior registro en la base de datos, el cual, servirá para la generación de reportes. Estos procesos son asíncronos y no necesitan que uno de ellos se detenga para continuar con el otro.

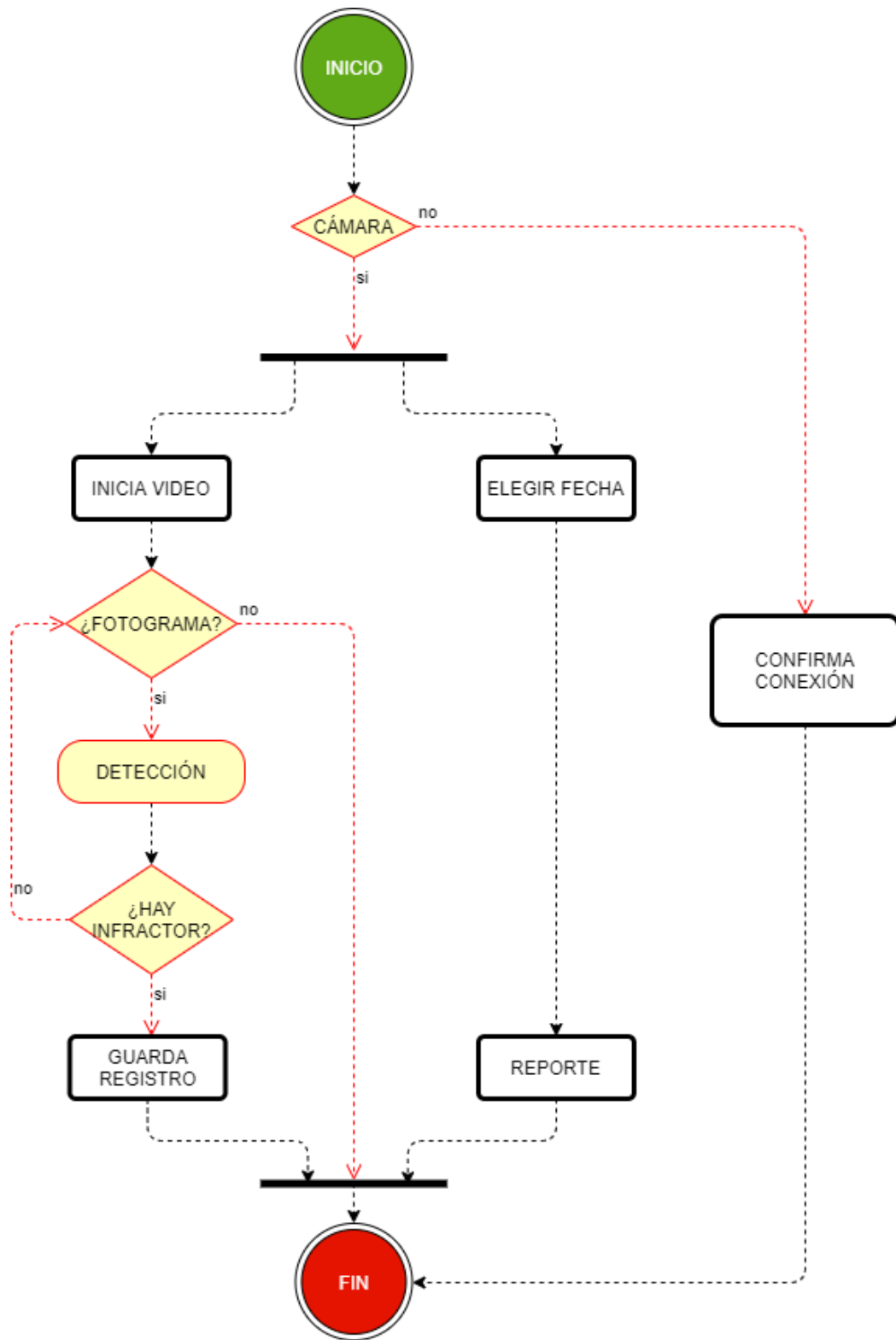


Fig. 15. Diagrama de flujo de proceso del sistema

Proceso de detección de personas y cálculo de distancias euclidianas.

En la Fig. 16 se detalla el flujo de actividades en el proceso de detecciones, que inicia activando el uso de la GPU y permite la ejecución de varias tareas al mismo tiempo [49] para luego capturar el fotograma que será analizado por YOLO, detentando así, personas. Posteriormente mide la distancia entre las personas detectadas, determinando los datos de los infractores, los cuales, se guardarán en la base de datos.

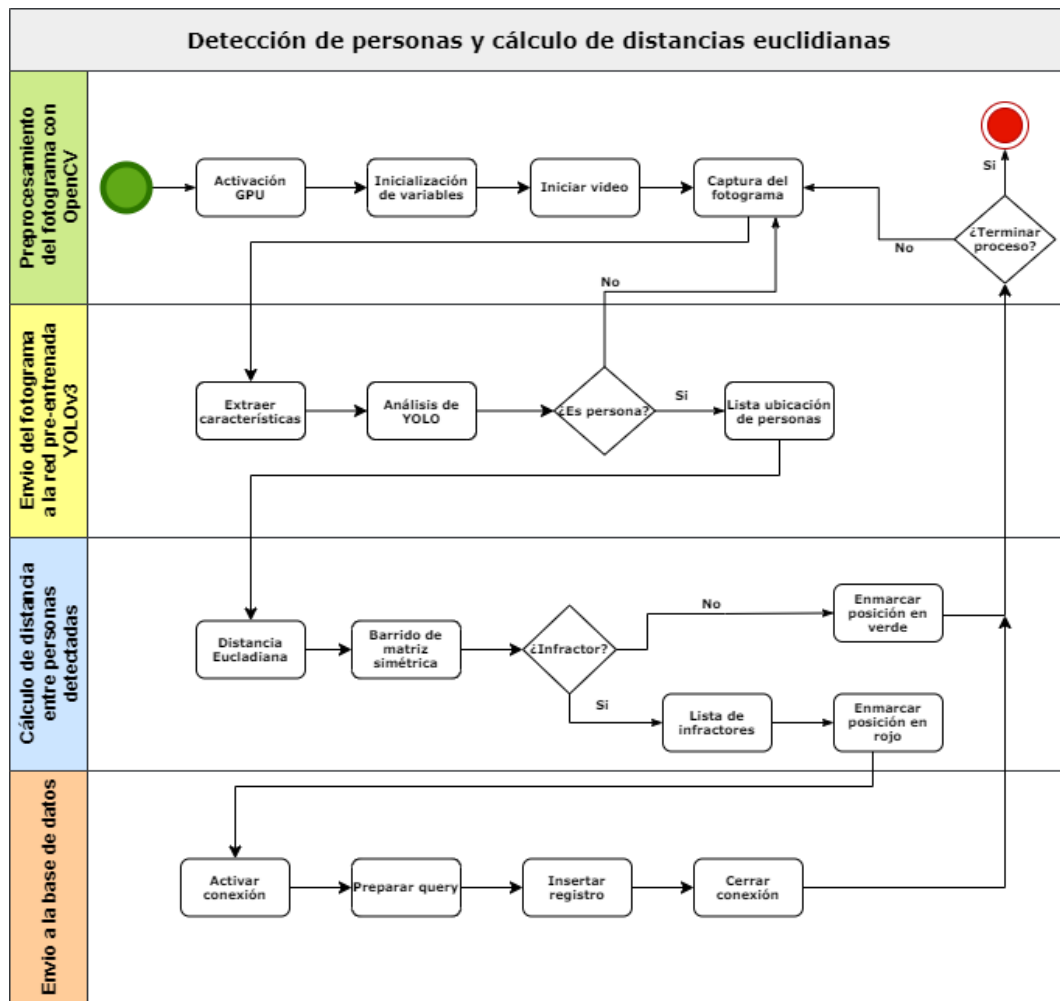


Fig. 16. Proceso de detección de personas y cálculo de distancias euclidianas

2.5.5. Diseño de Interfaces gráficas

Los diseños de las interfaces se basaron en los requerimientos presentados por el cliente, buscando siempre que el sistema sea amigable e intuitivo.

Módulo de aplicación

En la ventana principal se visualiza el video renderizado en tiempo real, mostrando las detecciones, infractores; además, se muestra un menú superior con las opciones de video y reportes.

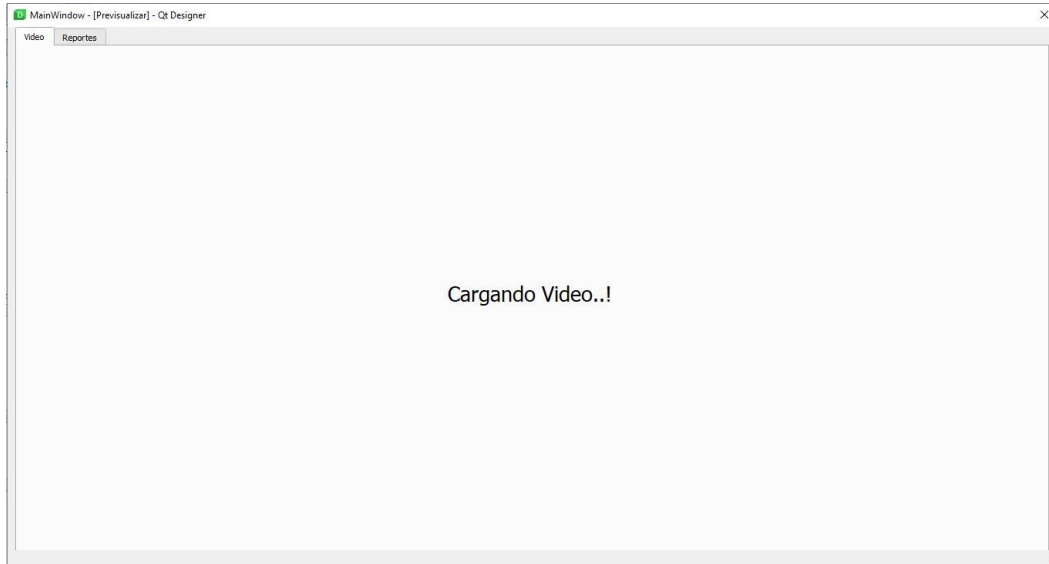


Fig. 17. Diseño preliminar de la vista del módulo de aplicación

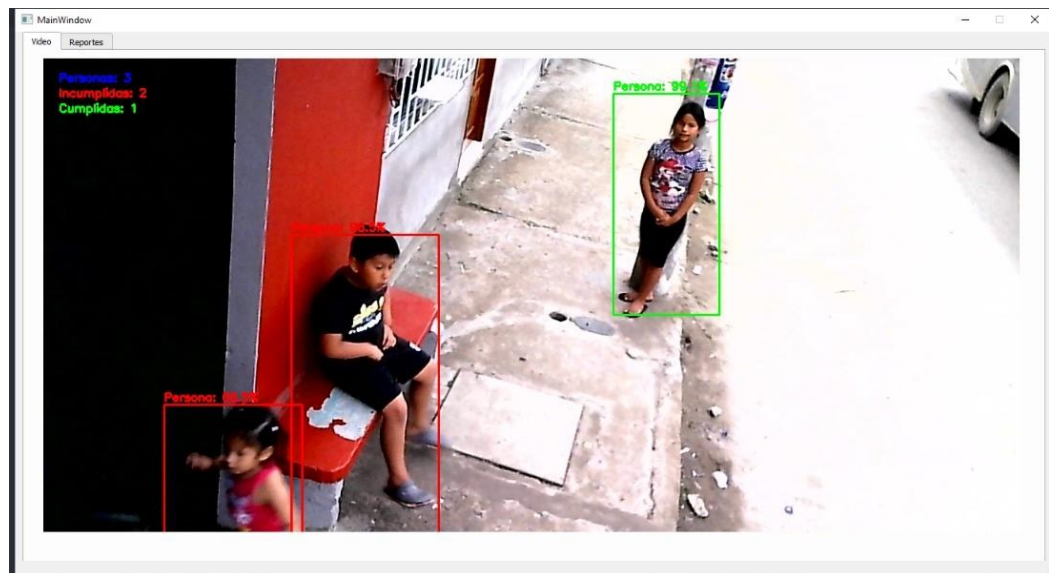


Fig. 18. Interfaz módulo de aplicación

Módulo de reportes

La pestaña reportes muestra la disposición de los elementos dentro la ventana, donde se escoge una fecha que es el filtro para los gráficos de histogramas de barras e histogramas de líneas que detallan información de número de infractores con sus

distancias promedios y de la misma forma el porcentaje de cumplimiento con un aproximado de personas detectadas, también muestra el total de detecciones y el número total de personas infractoras.

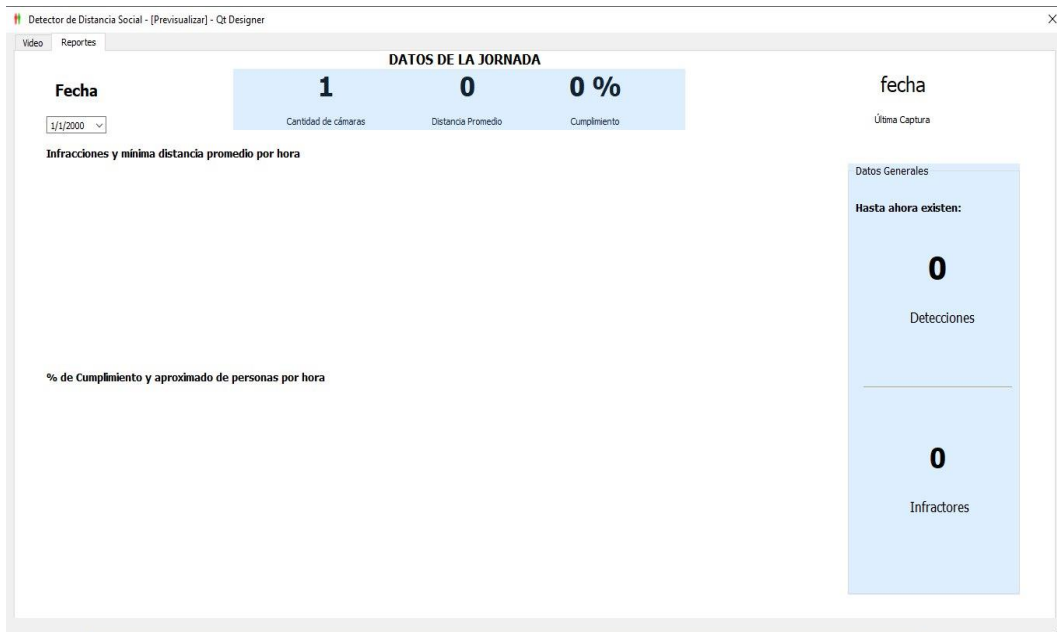


Fig. 19. Diseño preliminar de la vista del módulo de reportes.

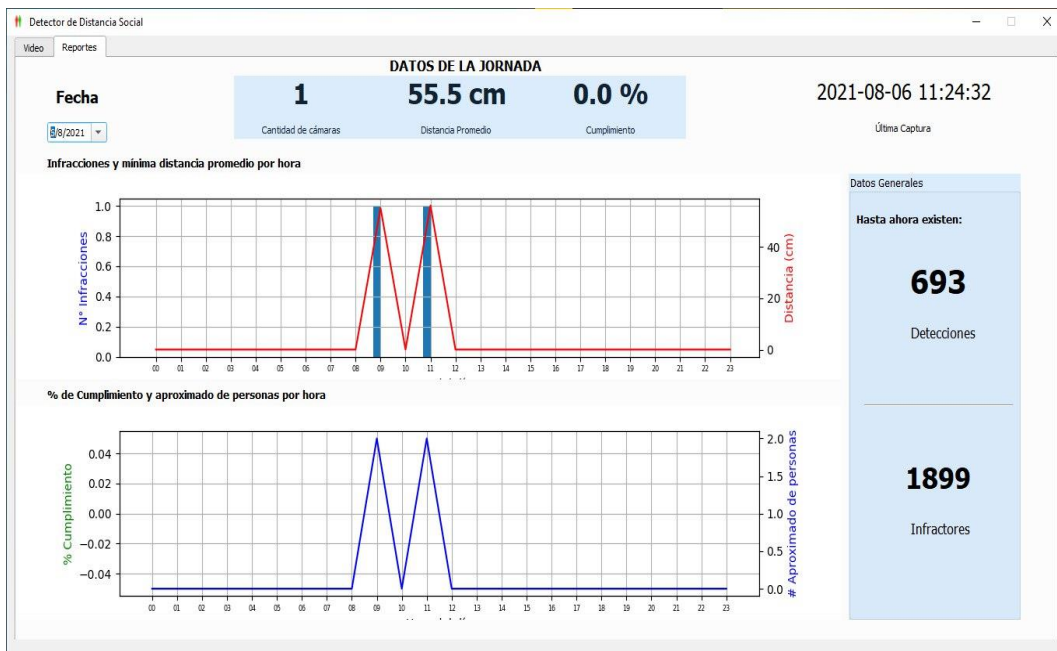


Fig. 20. Interfaz módulo de reportes.

2.6. Estudios de factibilidad

2.6.1. Factibilidad Técnica

Para el desarrollo del proyecto se detallan los recursos mínimos necesarios.

Cantidad	Descripción
1	PC Core i3 6ta Gen. 8RAM Nvidia GT710
1	Webcam 720p

Tabla 8. Recursos de hardware.

Cantidad	Descripción
1	Python 3.7.9
1	Open CV 4.2
1	YOLOv3
1	PyQt5
1	SQLite3
1	CUDA
1	cuDNN
1	Visual Studio Code

Tabla 9. Recursos de software.

2.6.2. Factibilidad Económica

Se detalla a continuación el valor de los recursos necesarios para el desarrollo del proyecto.

Presupuesto del sistema de control del distanciamiento social					
El tiempo de elaboración de la aplicación es de 4 meses					
Componente	Costo \$	Cantidad	Tiempo	Subtotal	Total
Personal					\$ 4100
Analista	600	1	2 meses	1200	
Programador	500	1	4 meses	2000	
Documentador	300	1	3 meses	900	
Hardware					\$ 760
PC Core i3 6ta Gen. 8RAM Nvidia GT710	0	1	4 meses	700	

Webcam 720p	0	1	4 meses	60	
Software					\$ 0
Python 3.7.9	0	1	4 meses	0	
Open CV 4.2	0	1	4 meses	0	
YOLOv3	0	1	4 meses	0	
PyQt5	0	1	4 meses	0	
SQLite3	0	1	4 meses	0	
CUDA	0	1	4 meses	0	
cuDNN	0	1	4 meses	0	
Visual Studio Code	0	1	4 meses	0	
Servicios básicos					\$ 180
Energía eléctrica	10	1	4 meses	40	
Internet	35	1	4 meses	140	
Otros					\$ 80
Transporte	20	1	4 meses	80	
TOTAL					\$ 5120

Tabla 10. Costo del Sistema de control de distanciamiento social.

Los valores detallados de personal, hardware, software, servicios básicos y otros serán asumidas por el autor del presente proyecto de Tesis. Por lo tanto, el desarrollo del Sistema para controlar el distanciamiento social de los estudiantes de la Escuela de Educación Básica Lcda. Angélica Villón tendrá un costo de \$0 debido a que es un trabajo de titulación, no obstante, si se desea desarrollar el proyecto, su costo es de \$5120.

Costos de Implementación

Para la implementación del sistema se deberá contar con equipos que tengan las mismas características descritas en la Tabla 8, los valores de adquisición de estos equipos serán cubiertos por los administradores de la escuela.

2.6.3. Factibilidad Operativa

El sistema es 80% autónomo, esto significa que no necesita de un operador para su correcto funcionamiento. Además, cumple el objetivo de reconocer personas y determinar si infringen o no el protocolo de distanciamiento social para el que fue

creado. La generación de los reportes es intuitiva y solo basta elegir la fecha para obtener los gráficos estadísticos donde se detalla la información requerida. Por lo tanto, no es necesario realizar una capacitación para el uso del sistema, pero sí una inducción para dar a conocer su funcionamiento.

2.7. Desarrollo del sistema

2.7.1. Desarrollo del entorno de detecciones

Creación del entorno

La creación de un entorno de desarrollo que vincula OpenCV y CUDA permite la aceleración por GPU de aplicaciones en una amplia gama de dominios, desde el procesamiento de imágenes hasta el aprendizaje profundo, el análisis numérico y la ciencia computacional [41] ayudando al rendimiento de los sistemas basados en Inteligencia Artificial y Visión por Computadora.

Procedimiento:

1. Descargar CUDA y cuDNN desde la página oficial: <https://developer.nvidia.com/cuda-zone> [41].
2. Descargar Cmake desde la página oficial: <https://cmake.org/install/> [50].
3. Descargar e instalar Visual Studio Community desde la página oficial: <https://visualstudio.microsoft.com/es/vs/community/> [40].
4. Descargar opencv-contrib del repositorio de github: https://github.com/opencv/opencv_contrib [51].
5. Abrir el proyecto desde CMake y elegir un directorio de salida.

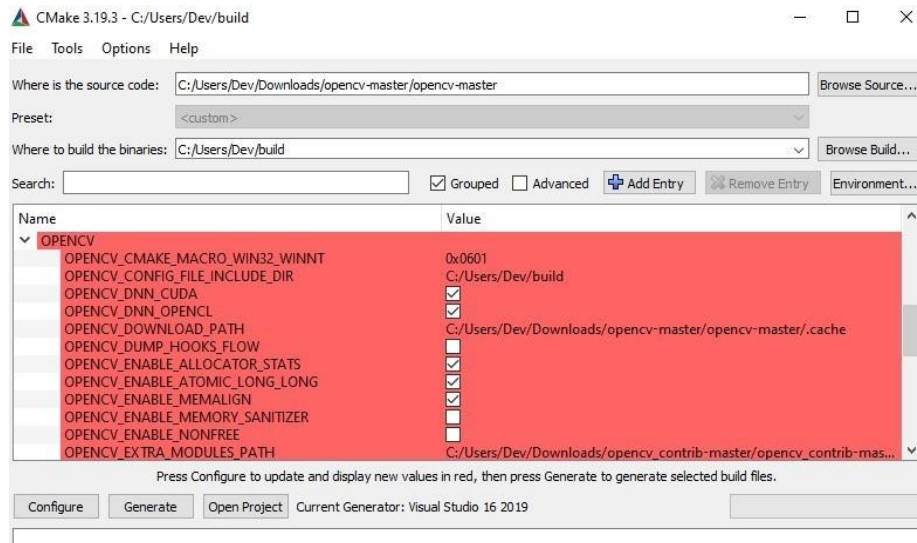


Fig. 21. Configuración para dnn-CUDA.

6. Habilitar las opciones de CUDA para aceleración de gráficos y cálculos matemáticos.



Fig. 22. Configuración CUDA para aceleración gráfica y análisis matemático

7. Abrir el nuevo directorio con las configuraciones realizadas en Visual Studio para la depuración del proyecto.
8. Elegir el archivo INSTALL para la depuración y la instalación de las nuevas características.

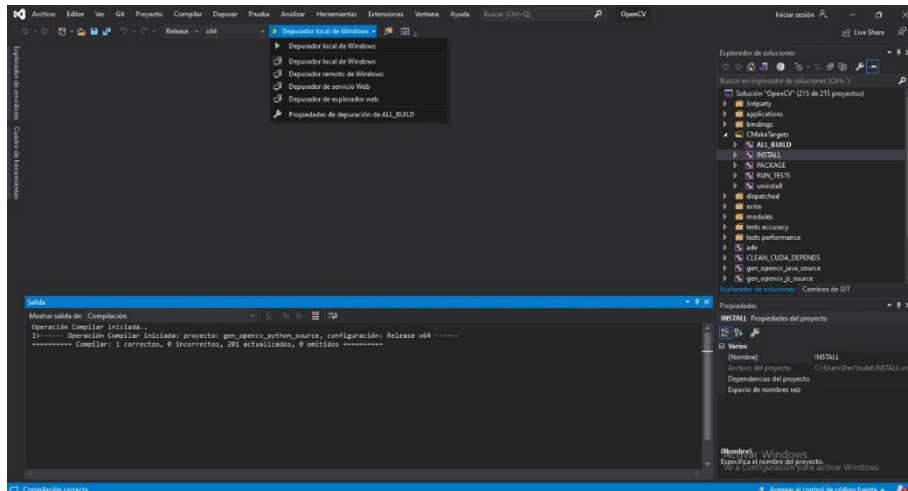


Fig. 23. Instalación global de OpenCV mediante VS-Community con las nuevas características.

9. Verificar el uso de la GPU mediante script en lenguaje Python.

```

C:\Windows\system32\cmd.exe - python
Microsoft Windows [Versión 10.0.19042.746]
(c) 2020 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Dev>python
Python 3.8.5 (default, Sep 3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> i = cv2.cuda.getCudaDevice()
>>> cv2.cuda.printCudaDeviceInfo(i)

```

Fig. 24. Verificación del uso de GPU.

```

C:\Windows\system32\cmd.exe - python
>>> cv2.cuda.printCudaDeviceInfo(i)
*** CUDA Device Query (Runtime API) version (CUDA static linking) ***
Device count: 1
Device 0: "GeForce GT 710"
  CUDA Driver Version / Runtime Version      11.20 / 10.10
  CUDA Capability Major/Minor version number: 3.5
  Total amount of global memory:             2048 Mbytes (2147483648 bytes)
  ( 1) Multiprocessors x (192) CUDA Cores/MP: 192 CUDA Cores
  GPU Clock Speed:                           0.95 GHz
  Max Texture Dimension Size (x,y,z)         1D=(65536), 2D=(65536,65536), 3D=(4096,4096,4096)
  Max Layered Texture Size (dim) x layers    1D=(16384) x 2048, 2D=(16384,16384) x 2048
  Total amount of constant memory:           65536 bytes
  Total amount of shared memory per block:   49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                  32
  Maximum number of threads per block:       1024
  Maximum sizes of each dimension of a block: 1024 x 1024 x 64
  Maximum sizes of each dimension of a grid: 2147483647 x 65535 x 65535
  Maximum memory pitch:                     2147483647 bytes
  Texture alignment:                         512 bytes
  Concurrent copy and execution:             Yes with 1 copy engine(s)
  Run time limit on kernels:                 Yes
  Integrated GPU sharing Host Memory:        No
  Support host page-locked memory mapping:   Yes
  Concurrent kernel execution:               Yes
  Alignment requirement for Surfaces:        Yes
  Device has ECC support enabled:            No
  Device is using ICC driver mode:           No
  Device supports Unified Addressing (UVA):   Yes
  Device PCI Bus ID / PCI location ID:       1 / 0
  Compute Mode:
     Default (multiple host threads can use ::cudaSetDevice() with device simultaneously)
deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 11.20, CUDA Runtime Version = 10.10, NumDevs = 1
>>>

```

Fig. 25. Características tarjeta gráfica instalada.

Descarga de la red neuronal YOLOv3 y configuración para detección de personas.

Procedimiento:

1. Se descargó yolov3.cfg, yolov3.weights y coco.names desde la página oficial: <https://pjreddie.com/darknet/yolo/> [18].
2. Se modificó el archivo yolo.py desde: <https://github.com/nandinib1999/object-detection-yolo-opencv/blob/master/yolo.py> [52] para que reconozca y trabaje con la GPU.

```
1 #comprobar GPU
2 indice_gpu = cv2.cuda.getDevice()
3 if indice_gpu >= 0:
4     red.setPreferableBackend(cv2.dnn.DNN_BACKEND_CUDA)
5     red.setPreferableTarget(cv2.dnn.DNN_TARGET_CUDA)
6     #imprimir información de GPU
7     cv2.cuda.printCudaDeviceInfo(indice_gpu)
```

Fig. 26. Comprobación de GPU

3. Detección de objetos con YOLOv3.

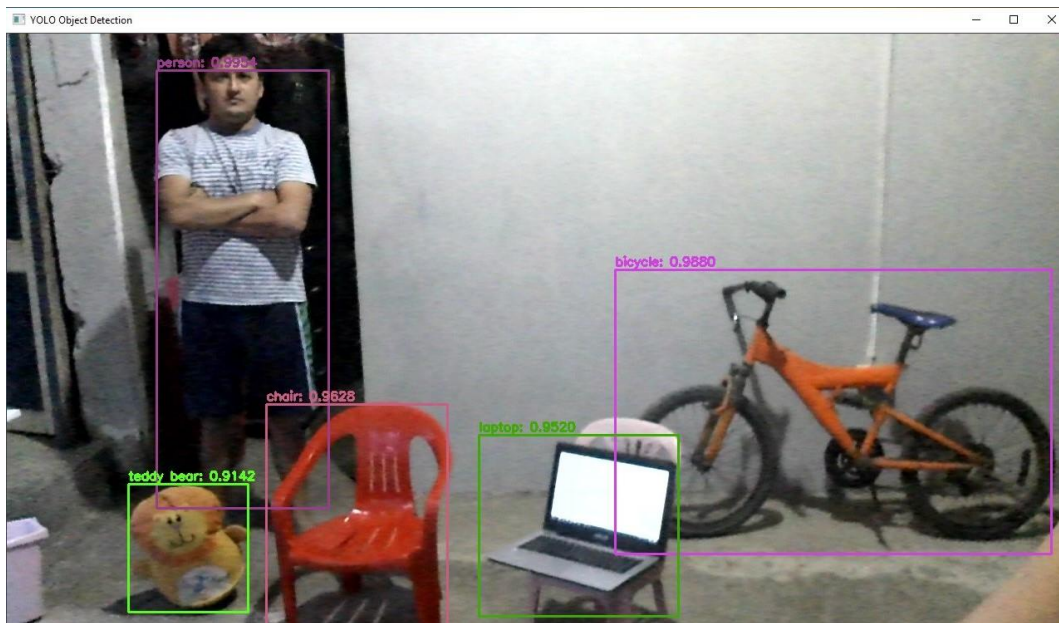


Fig. 27. Detección de objetos con YOLOv3

4. Modificación para detectar solo personas.

```

1 def dibujar_cajas(imagen, cajas, lista_confianza, ids_clases, idxs, color):
2     if len(idxs) > 0:
3         for i in idxs.flatten():
4             # extraer las coordenadas de las cajas
5             x, y = cajas[i][0], cajas[i][1]
6             ancho, alto = cajas[i][2], cajas[i][3]
7
8             if etiquetas[ids_clases[i]] == 'person':
9                 # dibujar la caja que enmarca a cada persona con su respectivo nivel de confianza
10                #color = [int(c) for c in colors[classIDs[i]]]
11                cv2.rectangle(imagen, (x, y), (x + ancho, y + alto), color, 2)
12                texto = "{}: {:.1f}%".format('Persona', lista_confianza[i]*100)
13                cv2.putText(imagen, texto, (x, y - 5), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
14
15    return imagen

```

Fig. 28. Condición para detectar personas

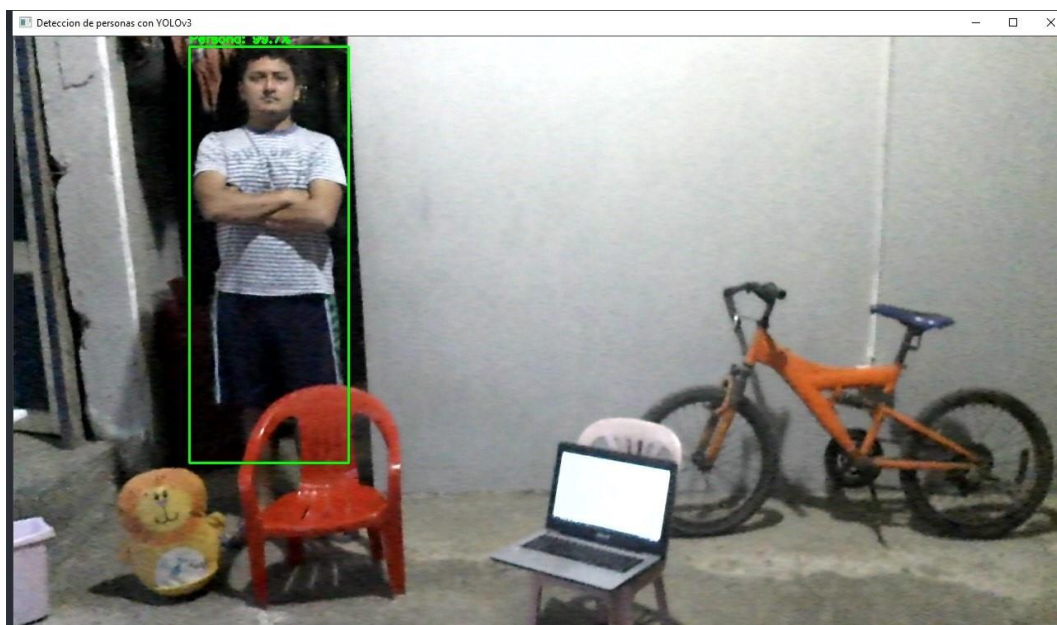


Fig. 29. Detección solo de personas

2.7.2. Desarrollo del código de distanciamiento

Una vez establecidas las *bounding-boxes* de cada persona detectada se procede a obtener el punto (x_1, y_1) que será utilizado para calcular la distancia entre puntos. Cada *bounding-boxes* contiene 6 valores: p_c que indica el valor de la clase a la que pertenece, las coordenadas (b_x, b_y) que señala el centro del cuadro, b_h y b_w es el ancho y alto de la caja y el valor c es la confianza de que la imagen detectada pertenece a una clase [53].

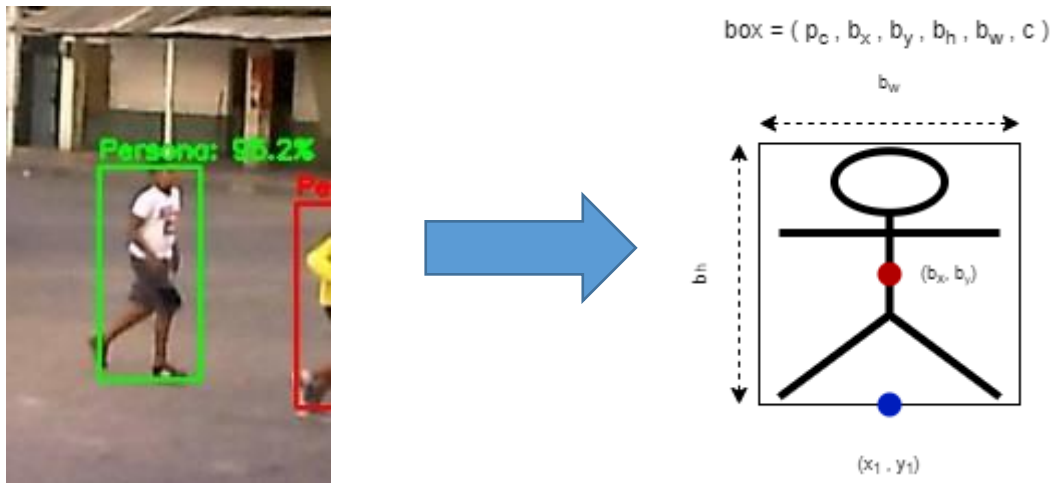


Fig. 30. Bounding-boxes y punto a utilizar para medir la distancia entre personas

El cálculo de las coordenadas (x_1, y_1) está determinado por los pies de cada persona detectada (Fig. 30), mediante una operación matemática de suma y división de coordenadas correspondientes. Ese valor es enviado a `resultados = []` que es una lista de n elementos que dependerá del número de personas detectadas en el fotograma analizado en ese instante.

El módulo `scipy.spatial.distance` permite el cálculo de la distancia entre cada par de 2 colecciones de entrada y una métrica de distancia [43] como se observa en la Fig. 31. Para nuestro trabajo son 2 matrices iguales que permitirán obtener la distancia de todas contra todas las personas detectadas y `euclidean` es la métrica a utilizar para medir distancia. Este método recorre cada posición i, j de la primera matriz con cada uno de los elementos de la segunda matriz (Fig. 31) dando como resultado una matriz simétrica (Fig. 32) que además tiene su diagonal principal llena de ceros.

Finalmente, se itera la matriz resultante para verificar si la distancia en píxeles es menor al valor de referencia, 400 píxeles es aproximadamente 150 cm, si esto es así, se añade los índices a un diccionario de infractores para posteriormente compararlo con la lista de resultados que contiene las detecciones encontradas, y así enmarcarlos en rojo si son infractores o mantenerlos en verde si no lo son.

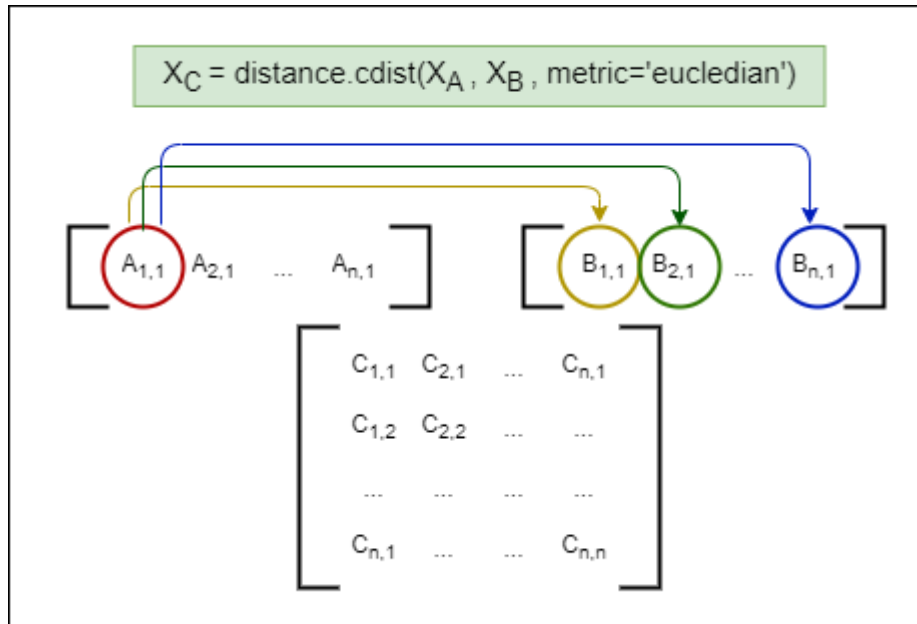


Fig. 31. Iteración del método `scipy.spatial.distance` con la métrica 'euclidean'

```

X_c = distance.cdists([(2,3) (5,2) (4,3)], [(2,3) (5,2) (4,3)], metric = ' euclidean')

X_c = [ 0  3,16  2 ]
      [ 3,16  0  1,41 ]
      [ 2  1,41  0 ]

donde X_c es una matriz simétrica con diagonal principal llena de ceros

```

Fig. 32. Ejemplo de salida de matriz simétrica

2.7.3. Desarrollo del módulo de aplicación

Qt Designer es una aplicación a partir de Qt5 que posee las herramientas para la creación de interfaces gráficas en entornos de escritorios. El sistema tendrá dos pestañas video y reportes. La primera contendrá una etiqueta *label* donde se renderiza el video mediante el método `setPixmap()`, que irá iterando una imagen tras otra. Cada elemento que conforma la vista tiene diferentes métodos que ayudan al funcionamiento de la aplicación. El archivo que contiene las vistas de estas dos pestañas tiene el nombre de *interface.ui* ubicada dentro de la carpeta *view* del directorio raíz (Fig. 33).

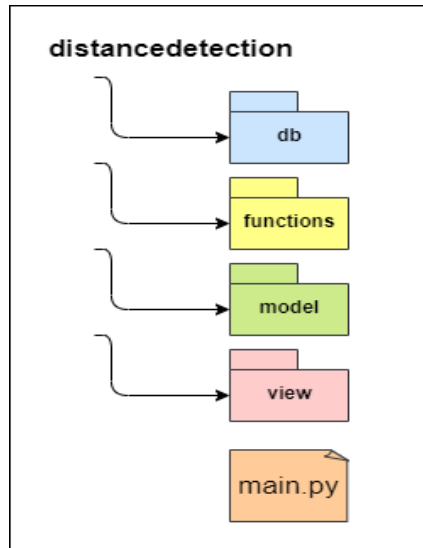


Fig. 33. Directorio raíz del proyecto

Para cargar la aplicación y sus diferentes elementos se utiliza la función *loadUi()* del método *uic* de PyQt5 que necesita la ruta del archivo *.ui* que permite el renderizado de la vista desde su controlador (Fig. 34).

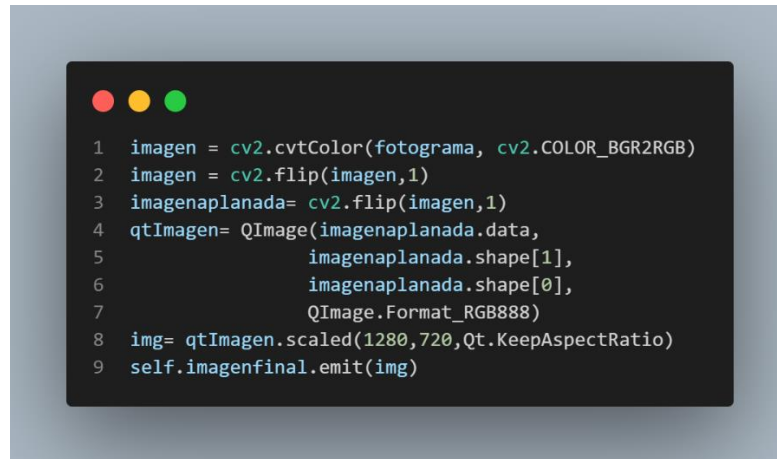
```
class MainWindow (QMainWindow):
    def __init__(self):
        QMainWindow.__init__(self)
        self.path = os.getcwd()
        path_ui = os.path.join(self.path,r"view\interface.ui")
        uic.loadUi(path_ui,self)
```

Fig. 34. Clase *MainWindow* de la ventana principal

El controlador *interface.py* importa la clase *videoStreaming(QThread)* del archivo *distance_detection_class.py* que posee las funciones de detecciones de personas y cálculo de distancia. Esta clase posee el método *run()*, que es un método por defecto de *QThread* y se encarga de iniciar las detecciones. *QThread* es una clase de Qt5 que permite generar hilos de procesamiento paralelos al hilo principal que se ejecuta a partir de *QApplication*. El análisis de los fotogramas es independiente de los procesos ejecutados por la aplicación, esto se realiza de forma asíncrona enviando información por medio de señales.

Dentro de la clase se declaran tres señales, la primera para enviar una imagen como argumento al controlador de la interfaz, la segunda para guardar información a la

base de datos y la última que permitirá la validación de la integración de la cámara al sistema. La función *emit()* es la encargada de enviar información de un fichero a otro sin detener la ejecución del código.



```
1 imagen = cv2.cvtColor(fotograma, cv2.COLOR_BGR2RGB)
2 imagen = cv2.flip(imagen,1)
3 imagenaplanada= cv2.flip(imagen,1)
4 qtImagen= QImage(imagenaplanada.data,
5                 imagenaplanada.shape[1],
6                 imagenaplanada.shape[0],
7                 QImage.Format_RGB888)
8 img= qtImagen.scaled(1280,720,Qt.KeepAspectRatio)
9 self.imagenfinal.emit(img)
```

Fig. 35. Envío de señal mediante función *emit()*

Normalmente la ejecución de un código se realiza línea por línea y cuando se llama a un segundo fichero el proceso del primero se detiene hasta que termine la ejecución del segundo, luego se reanuda el proceso del primer fichero hasta terminar. Las señales mediante la función *emit()* permiten la ejecución de dos procesos simultáneamente evitando conflictos de dependencia entre controladores.

El registro de los infractores es acumulado durante cierto tiempo mientras existan personas enfocadas por la cámara y, además, el número de infractores sea mayor en la siguiente iteración. Si no existen cambios durante 15 fotogramas se emite la señal de guardar el registro. La clase *TimerMessageBox(QMessageBox)* permite ver en pantalla una notificación con el mensaje “Detección guardada” (Fig. 36).


```
1 class TimerMessageBox(QMessageBox):
2     def __init__(self, timeout=3, parent=None):
3         super(TimerMessageBox, self).__init__(parent)
4         self.setWindowTitle("Atención..!!")
5         self.setIcon(QMessageBox.Warning)
6         self.time_to_wait = timeout
7         self.setText("Detección Guardada".format(timeout))
8         self.setStandardButtons(QMessageBox.NoButton)
9         self.timer = QTimer(self)
10        self.timer.setInterval(1000)
11        self.timer.timeout.connect(self.changeContent)
12        self.timer.start()
```

Fig. 36. Class *TimerMessageBox*.

2.7.4. Desarrollo del módulo de reportes

La pestaña reportes tiene etiquetas que cambian de valor según las consultas a la base de datos mediante el método *setText()*, además, los gráficos se muestran dentro de un *gridLayout* que tiene un *widgets* secundario donde se almacena la imagen de *matplotlib*. El archivo *db.py* permite la creación, conexión de la base de datos *SQLite*, la creación de la tabla que guardará el registro, la obtención de los datos según el día escogido, la hora de la última detección de un día determinado y el número total de infractores.

La consulta a la base de datos se hace mediante la creación de un objeto de la clase *QSqlQuery* que proporciona el medio para la ejecución y manipulación de sentencias *SQL*, los métodos *prepare()* y *exec()* devuelven un booleano que indica si la sentencia es correcta y se ejecuta sin errores, respectivamente (Fig. 37). La función devuelve un arreglo procedente de la consulta *SQL* y es agregada a una lista para ser utilizada en las gráficas que se muestran en la pestaña reportes.

```
1 def datosdia(dia):
2     datos = []
3     incum = 0
4     total = 0
5     dist_prom = 0
6     n = 0
7     hora = ''
8     q = QSqlQuery()
9     if q.prepare("""SELECT sum(incumplidos),sum(total), sum(dist_promedio),
10                    count(fecha),strftime('%H',fecha)
11                    FROM registro WHERE date(fecha)=" + dia + "
12                    GROUP BY strftime('%H',fecha);"""):
13         if q.exec():
14             while q.next():
15                 incum = q.value(0)
16                 total = q.value(1)
17                 dist_prom = q.value(2)
18                 n = q.value(3)
19                 hora = q.value(4)
20                 datos.append([incum,total,dist_prom,n,hora])
21     return datos
22
```

Fig. 37. Función *datosdia()*.

La información que devuelve las funciones *datosdia(dia)*, *ultimadetdia(dia)* y *totaldetinf()* permite por medio de Matplotlib la visualización de los datos en forma de gráficos estadísticos, para esto se vincula Matplotlib con PyQt5 mediante el archivo *canvas.py* que se encuentra dentro de la carpeta *functions* (Fig. 33) donde se importa *FigureCanvasQTAgg* que contiene los módulos necesarios para la renderización de las gráficas, posteriormente se crea una clase *MplCanvas* que se importa en *interface.py*.

Las gráficas son objetos heredados de *MplCanvas* que mediante el método *twinx()* permite la superposición del gráfico de histograma de barras y el histograma de líneas, dando como resultado la representación de 4 figuras estadísticas en la pestaña reportes (Fig. 38). La función *cargarreporte()* se encarga de dibujar los gráficos que se actualizan cada vez que hay un cambio en la fecha y esto ocurre mediante el evento *dateChanged* que pertenece al elemento *dateEdit*.

```

1 self.grafica_uno = MplCanvas()
2 self.ax = self.grafica_uno.axes.twinx()
3 self.grafica_dos = MplCanvas()
4 self.ax1 = self.grafica_dos.axes.twinx()

```

Fig. 38. Superposición de gráficas mediante el método *twinx()*

A continuación, se define las etiquetas de los *ejes x* y *eje y* que conforma las gráficas (Tabla 11), siendo en el caso de las *y* valores que devuelve la función *diacompleto(dia)*. Si existe registros en ese día, lo que se hace es iterar en los índices el arreglo de datos agregando valores a las listas de tamaño 24 inicializadas en ceros en los índices que corresponde. Finalmente, se etiquetan los ejes con *set_ylabel()* y *set_xlabel()* para que, llamando a *draw()*, se dibujen las gráficas.

Nº	Gráfico	eje x	eje y
Gráfico 1	Histograma de barras	Horas del día	Número de infracciones
	Histograma de líneas		Distancia promedio
Gráfico 2	Histograma de barras		Porcentaje de cumplimiento
	Histograma de líneas		Aproximado de personas

Tabla 11. Eje *x* y *eje y* de las gráficas

Finalmente, se crea un archivo *main.py* (Fig. 33) que tiene el método *QApplication* que permite la creación de la aplicación QT y que contiene la función *main()* que da inicio a la aplicación por medio de la clase *MainWindow()*. Cabe indicar que, para que inicie la aplicación debe existir una cámara detectada y capaz de renderizar un video caso contrario se muestra un mensaje y se cierra el programa.

2.8. Implementación del sistema

Se debe tener en cuenta que para la implementación del sistema se requiere equipos con características recomendables como las descritas en la Tabla 8.

Para la ejecución de este proyecto es necesario la vinculación de OpenCV y CUDA (40), posteriormente realice los siguientes pasos:

1. Descargar el repositorio desde:

<https://github.com/andatogua/DistanceDetection> y pegar en el disco C.

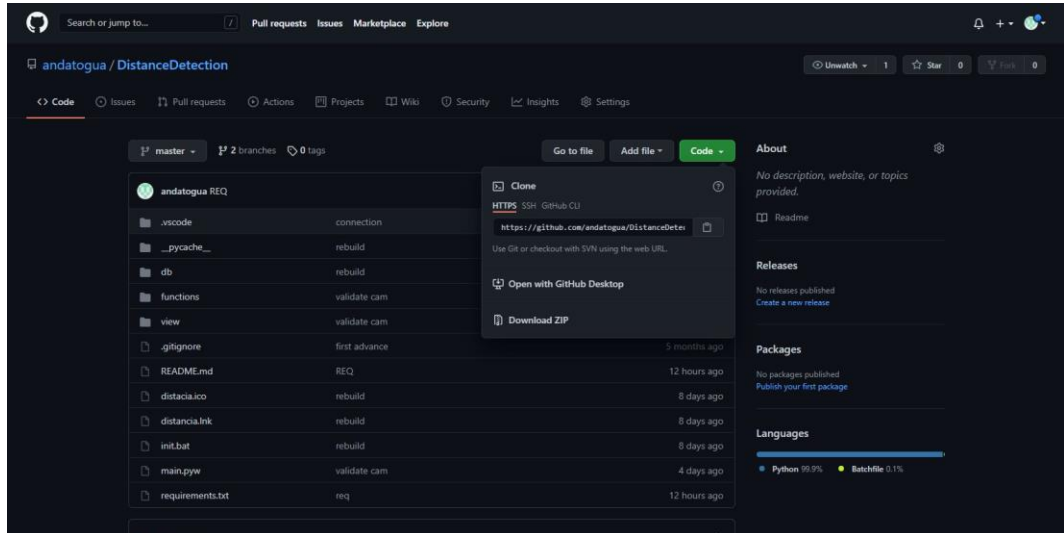


Fig. 39. Repositorio de la aplicación en github.

2. Abra una consola y desde el directorio raíz ejecute el comando
`pip install -r requirements.txt`

```
C:\Windows\system32\cmd.exe
C:\>cd DistanceDetection

C:\DistanceDetection>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 0A94-28B0

Directorio de C:\DistanceDetection

11/08/2021 09:38 <DIR>      .
11/08/2021 09:38 <DIR>      ..
11/03/2021 02:01                22 .gitignore
11/03/2021 10:26 <DIR>      .vscode
11/03/2021 04:47 <DIR>      db
31/07/2021 00:54           27.484 distancia.ico
31/07/2021 00:55           1.656 distancia.lnk
06/08/2021 10:01 <DIR>      functions
31/07/2021 00:43                27 init.bat
04/08/2021 21:45           745 main.pyw
24/01/2021 20:49 <DIR>      model
11/08/2021 09:39                522 README.md
06/08/2021 11:24           40.960 registro.sqlite
11/08/2021 09:40                73 requirements.txt
04/08/2021 21:55 <DIR>      view
20/07/2021 21:51 <DIR>      __pycache__
                8 archivos          71.489 bytes
                8 dirs    6.011.961.344 bytes libres

C:\DistanceDetection>
```

Fig. 40. Directorio raíz de la aplicación

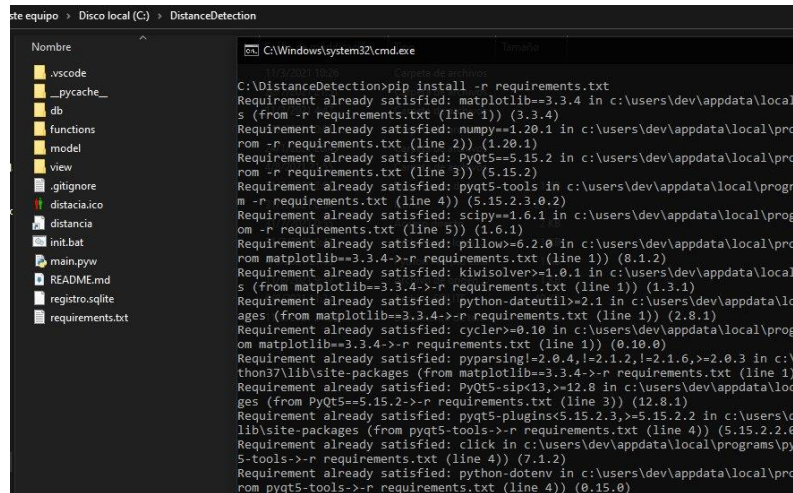


Fig. 41. Instalación de dependencias.

- Al finalizar la instalación de dependencias, se copia y pega “distancia” en el escritorio.

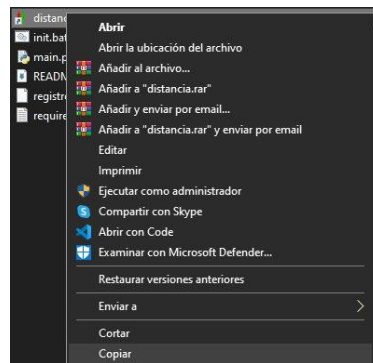


Fig. 42. Copia del acceso directo en el escritorio

- Se ejecuta el programa desde el acceso directo.

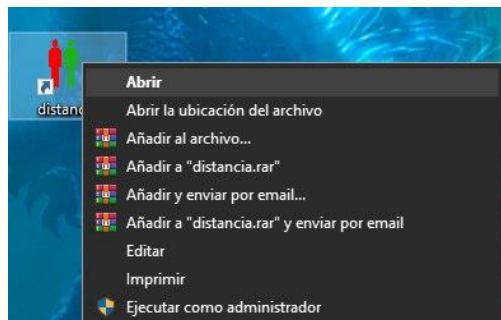


Fig. 43. Ejecución del sistema.

2.9. Pruebas

2.9.1. Prueba #1: Detección de personas variando nivel de confianza

En este apartado se toma una imagen de prueba donde se observan grupos de personas en filas y en distintas perspectivas con respecto al observador (cámara) y se determinó como afecta el cambio el nivel de confianza de 0.5, 0.4, 0.3 y 0.2 a los resultados finales, así se encontró el valor óptimo de confianza que genere la mayor cantidad de detección de personas.

2.9.2. Prueba #2: Calibrar la distancia de pixeles a metros

Para realizar esta prueba se tomó las siguientes consideraciones:

- Altura de la cámara con respecto al suelo, 3 metros.
- Se utilizó un flexómetro para medir 1,50 metros.
- Nivel de confianza de 0.3
- La posición del flexómetro cambia para cada muestra tomada (Anexo 6).
- El número de muestras es de 10 fotogramas.
- Se tomaron escenas donde hay 2 personas que cumplen la distancia permitida.

En la Fig. 46 se observó que las 2 personas están cumpliendo el distanciamiento social, sin embargo son enmarcadas en rojo siendo un problema de perspectiva que se debe considerar y que resulta por la superposición de las detecciones.

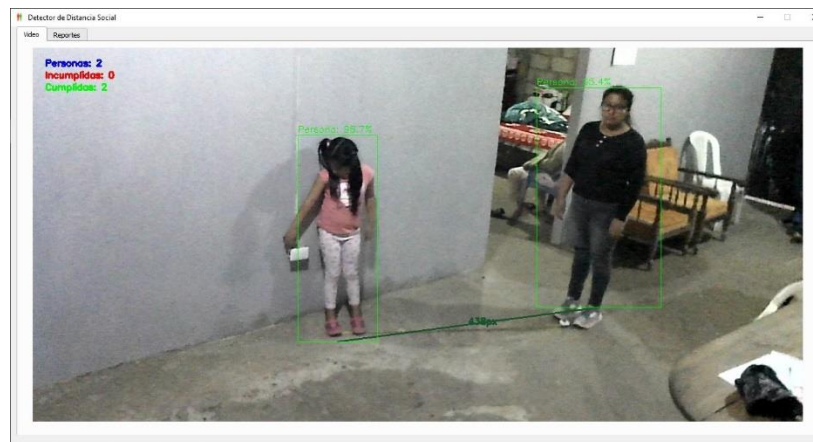


Fig. 44. Distancia en pixeles de personas no infractora.



Fig. 45. Distancia en pixeles de personas infractoras.

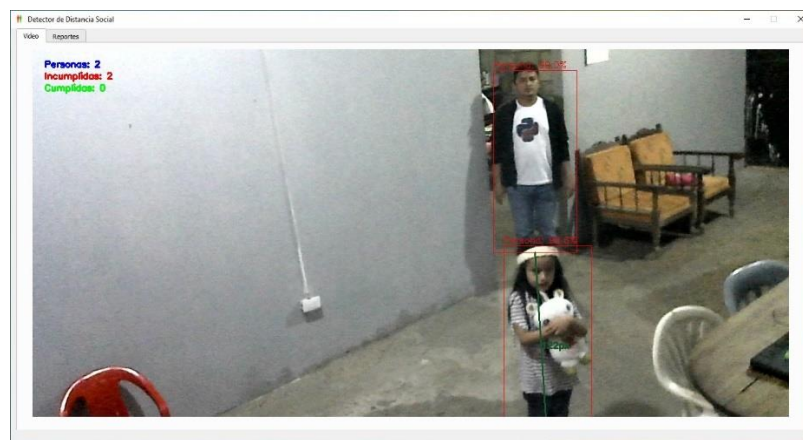


Fig. 46. Escena con falso positivo

2.9.3. Prueba #3: Infractores detectados variando distancia en pixeles.

Para esta prueba se toman 2 imágenes de grupo de personas en fila y ubicadas desde diferentes perspectivas que cumplen y no cumplen el distanciamiento social con un nivel de confianza de 0.3, variando el valor de la distancia. El sistema enmarcó de rojo a los infractores y de verde a los no infractores.



Fig. 47. Personas cumpliendo la distancia



Fig. 48. Personas infractoras

2.9.4. Prueba #4: Acceso a la aplicación

ACCESO A LA APLICACIÓN	
PRUEBA #:	4
Objetivo:	Comprobar el correcto acceso al sistema.
Usuario:	Administrador del sistema
Condiciones:	El sistema debe tener conectada una cámara.
Pasos	

1. Ingreso al sistema.		
2. Verifica si existe una cámara conectada y capaz de renderizar un video.		
Resultados de la prueba		
Resultados esperados	Estado	
Una vez iniciado el sistema, valida la integración de la cámara sino la encuentra envía un mensaje de error y cierra la aplicación ().	Exitoso	Fallido
	X	

Tabla 12. Prueba de acceso a la aplicación

2.9.5. Prueba #5: Generación de reporte

GENERACIÓN DE REPORTES		
PRUEBA #:	5	
Objetivo:	Comprobar la generación de reporte en forma de histograma de barras y líneas.	
Usuario:	Administrador del sistema	
Condiciones:	El administrador del sistema debe seleccionar una fecha.	
Pasos		
1. Ingreso al sistema. 2. Dirigirse a la pestaña reportes. 3. Escoger la fecha del reporte. 4. El administrador del sistema observa los reportes.		
Resultados de la prueba		
Resultados esperados	Estado	
Al seleccionar la fecha se visualizará los reportes que cambiaran a medida que escoja otra fecha.	Exitoso	Fallido
	X	

Tabla 13. Prueba generación de reporte

2.10. Resultados

2.10.1. Resultados Prueba #1

INFORMACIÓN DE ENTRADA		
Entrada	Imagen	
Confianza	50%	
Detecciones	N° Personas reales	N° Personas detectadas
Personas	14	9
Porcentaje de detecciones	64%	

a.

INFORMACIÓN DE ENTRADA		
Entrada	Imagen	
Confianza	40%	
Detecciones	N° Personas reales	N° Personas detectadas
Personas	14	11
Porcentaje de detecciones	79%	

b.

INFORMACIÓN DE ENTRADA		
Entrada	Imagen	
Confianza	30%	
Detecciones	N° Personas reales	N° Personas detectadas
Personas	14	13
Porcentaje de detecciones	93%	

c.

INFORMACIÓN DE ENTRADA		
Entrada	Imagen	
Confianza	20%	
Detecciones	N° Personas reales	N° Personas detectadas
Personas	14	16
Porcentaje de detecciones	>100% (Detecciones erróneas)	

d.

Tabla 14. Prueba variando nivel de confianza.

En base a los resultados de la Tabla 14 se determinó que el nivel de confianza óptimo es de 0.3 que equivale a un 93% de personas detectadas en una imagen.

2.10.2. Resultados Prueba #2

FOTOGRAMA N#1	DISTANCIA EN PÍXELES	DISTANCIA REAL EN CENTÍMETROS
1	415	150

2	409	
3	412	
4	398	
5	406	
6	417	
7	410	
8	399	
9	405	
10	408	
Promedio:	407,9	

Tabla 15. Prueba de conversión de pixeles a centímetros

El resultado mostrado en la Tabla 15 determina que el valor equivalente a 150 centímetros es aproximadamente igual a 408 pixeles, esto indica que un centímetro es aproximadamente igual a 3 pixeles.

2.10.3. Resultados Prueba #3

Parte 1

INFORMACIÓN DE ENTRADA		
Entrada	Imagen (Fig. 49 - a)	
Confianza	0.3	
Distancia mínima(pixeles)	35	
Detecciones	N° Infractores	N° no Infractores
Detecciones del sistema	0	24
Detecciones visuales	2	22
Total de personas detectadas	24	
Total de personas	24	

(a)

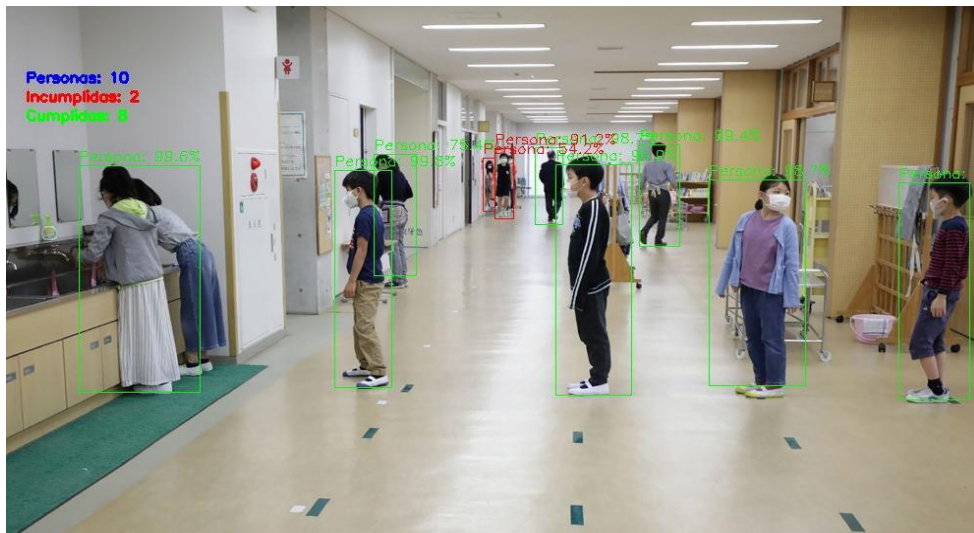
INFORMACIÓN DE ENTRADA		
Entrada	Imagen (Fig. 49 - b)	
Confianza	0.3	
Distancia mínima(pixeles)	35	
Detecciones	N° Infractores	N° no Infractores
Detecciones del sistema	2	8
Detecciones visuales	5	7
Total de personas detectadas	10	
Total de personas	12	

(b)

Tabla 16. Información de la prueba 3 parte 1



(a)



(b)

Fig. 49. Prueba 3 parte 1

Los resultados de la (a) mostraron un porcentaje de detección de persona del 100%, logrando detectar a las 24 personas presentes en la escena, mientras que en la (b) solo se detectó a 10 de las 12 personas, esto se debió a la superposición de una persona con otra. Por otro lado, al definir la distancia mínima de 35 píxeles en la (a) se presenta un error al no detectar a las dos profesoras infractoras, mientras que en la (b) por el solapamiento antes mencionado no se detectó a las 5 personas infractoras.

Entrada: Fig. 49 (b)		Predicción	
		Infractor	No Infractor
Situación real	Infractor	2	0
	No Infractor	0	8
RESULTADOS			
Precisión		1	
Sensibilidad		1	
Valor de referencia		1	
Exactitud		1	

Tabla 17. Matriz de confusión prueba 3 parte 1.

Los resultados de la prueba 3 indica que el modelo es 100% preciso y es capaz de detectar la mayor cantidad de infractores presentes en una imagen. Sin embargo, es importante aclarar que el sistema puede omitir detecciones por el solapamiento de personas.

Parte 2

INFORMACIÓN DE ENTRADA		
Entrada	Imagen1	
Confianza	0.3	
Distancia mínima(pixeles)	70	
Detecciones	N° Infractores	N° no Infractores
Detecciones del sistema	4	20
Detecciones visuales	2	22
Total de personas detectadas	24	
Total de personas	24	

(a)

INFORMACIÓN DE ENTRADA		
Entrada	Imagen2	
Confianza	0.3	
Distancia mínima(pixeles)	70	
Detecciones	N° Infractores	N° no Infractores
Detecciones del sistema	3	7
Detecciones visuales	5	7
Total de personas detectadas	10	
Total de personas	12	

(b)

Tabla 18. Información de la prueba 3 parte 2

Los resultados de la Fig. 50 (a) mostraron un porcentaje de detección de persona del 100%, logrando detectar a las 24 personas presentes en la escena, mientras que en la Fig. 50 (b) solo se detectó a 10 de las 12 personas, esto se debió a la superposición de una persona con otra. Por otro lado, al definir la distancia mínima de 70 píxeles en la Fig. 50 (a) se presenta a las dos profesoras infractoras pero un error al detectar como infractores a 2 estudiantes que están cumpliendo la distancia, siendo esto un error por solapamiento de personas, mientras que en la Fig. 50 (b) por el solapamiento antes mencionado no se detectó a las 5 personas infractoras pero mostró como infractor a un que visualmente si se encuentra cumpliendo la medida.

Entrada: Fig. 50 (b)		Predicción	
		Infractor	No Infractor
Situación real	Infractor	2	0
	No Infractor	2	20
RESULTADOS			
Precisión		0.5	
Sensibilidad		1	
Valor de referencia		0.67	
Exactitud		0.92	

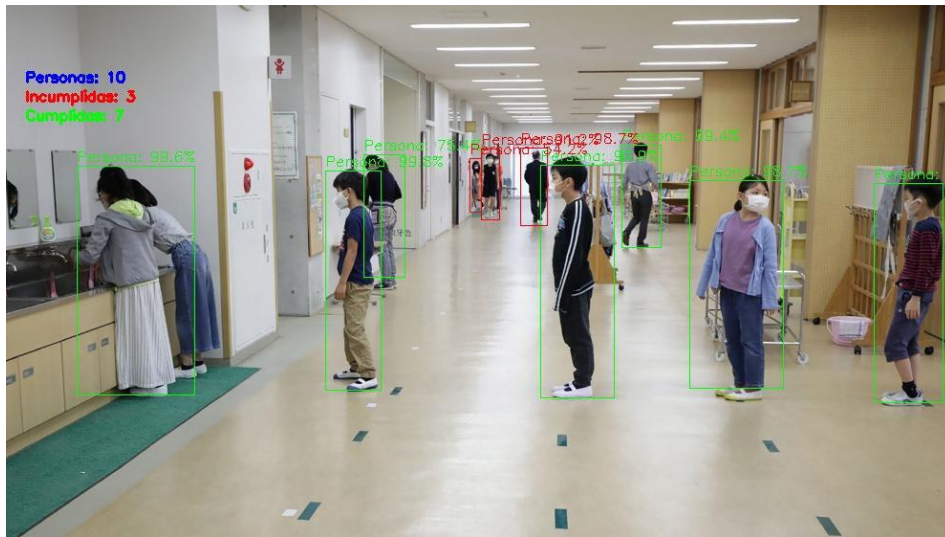
Tabla 19. Matriz de confusión prueba 3 parte 2.

En la

Tabla 19 se obtuvo una precisión del 50 % y una sensibilidad del 100% que indica que el algoritmo es capaz de detectar a los infractores en su mayoría, sin embargo se debe considerar que no se detectó a todas las personas presentes en la escena a analizar.



(a)



(b)

Fig. 50. Prueba 3 parte 2

Conclusiones

- Para el conjunto de pruebas realizadas referentes a la detección de personas, se utilizaron índices de confianza de 0.5, 0.4, 0.3, 0.2 de los cuales se determinó que el valor óptimo de confianza para clasificación de personas en una imagen es de 0.3, esto indica que el sistema acierta el 93% del total de personas.
- El valor equivalente de píxeles a centímetros que se obtuvo en las pruebas tiene un margen de error de aproximadamente más o menos dos centímetros, lo que representa un margen despreciable considerando la ubicación física de las personas. Esto significa que la distancia mínima permitida entre personas detectadas está en el rango de (148,152) centímetros.
- A partir de los reportes se pudo definir el rango de horas con mayor número de infracciones. Esta información gráfica ayuda al administrador del sistema a tomar decisiones frente al comportamiento de los infractores dentro de la escuela.
- Las pruebas realizadas a 2 imágenes diferentes para determinar el valor equivalente de píxeles a centímetros demostraron que, la tasa de conversión depende de la posición del observador con respecto a la escena, el solapamiento de objetos impide la detección y el cálculo de la distancia entre personas.
- Las métricas de la matriz de confusión permitieron medir la eficiencia, exactitud, precisión del modelo de clasificación de infractores y no infractores obteniendo un índice de sensibilidad mayor que el de precisión. Esto quiere decir que el sistema está orientado a reducir la detección de falsos positivos. No obstante, las métricas pueden interpretarse de manera distinta para cada caso en particular.

Recomendaciones

- Redimensionar el blob según la capacidad de pre procesamiento de la tarjeta gráfica. Si se desea tener mayor alcance en las detecciones se recomienda poseer una tarjeta con mayor número de núcleos CUDA.
- Realizar pruebas de ubicación de la cámara en varias posiciones de altura, grado de inclinación ayudará a encontrar el ángulo apropiado para obtener un mayor rango de visión de la escena, además la iluminación es un factor físico que debe corregirse para evitar la captura de resultados que no sean óptimos dentro de los sistemas que se basan en Visión Artificial.
- Reentrenar el modelo YOLOv3-tiny en la clase persona, para aumentar la confianza de la predicción, obtener coordenadas de ubicación más precisas y disminuir el margen de error en la conversión de píxeles a centímetros.
- Utilizar una cámara isométrica ayudaría a evitar errores de detección por solapamiento, además de conseguir una tasa de conversión con menor incertidumbre. Adicionalmente, los registros que se almacenen contendrá información precisa.
- Entrenar el modelo YOLOv3-tiny para las clases hombre y mujer ayudaría a identificar el índice de infractores dependiendo del sexo.

Bibliografía

- [1] «Coronavirus». <https://www.who.int/es/activities/preventing-noncommunicable-diseases/coronavirus> (accedido ago. 13, 2021).
- [2] «El distanciamiento social: continúe guardando distancia», *HealthyChildren.org*. <https://www.healthychildren.org/Spanish/health-issues/conditions/COVID-19/Paginas/Social-Distancing-Why-Keeping-Your-Distance-Helps-Keep-Others-Safe.aspx> (accedido ago. 13, 2021).
- [3] «Plan Educativo COVID 19 – Ministerio de Educación». <https://educacion.gob.ec/plan-educativo-covid-19/> (accedido ago. 13, 2021).
- [4] «Distancia2: inteligencia artificial para una movilidad más segura en época de COVID», *Moviliblog*, jun. 17, 2020. <https://blogs.iadb.org/transporte/es/distancia2-inteligencia-artificial-para-una-movilidad-mas-segura-en-epoca-de-covid/> (accedido jun. 09, 2021).
- [5] «WHO-2019-nCoV-HCW_Surveillance_Protocol-2020.1-spa.pdf». Accedido: jun. 09, 2021. [En línea]. Disponible en: https://apps.who.int/iris/bitstream/handle/10665/332994/WHO-2019-nCoV-HCW_Surveillance_Protocol-2020.1-spa.pdf
- [6] «Boletín Nota Política No2completa.pdf». Accedido: jun. 09, 2021. [En línea]. Disponible en: <https://repositorio.idep.edu.co/bitstream/handle/001/2384/Boletin%20Nota%20Politica%20No2completa.pdf?sequence=1&isAllowed=y>
- [7] «COVID-19-Paediatric-multisystem-inflammatory syndrome-20200501.pdf». Accedido: jun. 16, 2021. [En línea]. Disponible en: <https://www.rcpch.ac.uk/sites/default/files/2020-05/COVID-19-Paediatric-multisystem-%20inflammatory%20syndrome-20200501.pdf>
- [8] «BP-Síndrome-5-nov.pdf». Accedido: jun. 16, 2021. [En línea]. Disponible en: <https://www.salud.gob.ec/wp-content/uploads/2020/11/BP-Si%CC%81ndrome-5-nov.pdf>

- [9] «Plan de continuidad educativa – Ministerio de Educación». <https://educacion.gob.ec/plan-de-continuidad-educativa/> (accedido jun. 09, 2021).
- [10] E. P. S.L, «Nothingbutnet, la “startup” valenciana que permite controlar aforos y distancias de seguridad», *Valencia Plaza*. <https://valenciaplaza.com/nothingbutnet-startup-valenciana-aforos-distancias> (accedido jun. 09, 2021).
- [11] «ESPOL - Politécnicos crean sistema para monitorear el distanciamiento físico y el uso de mascarillas, a través de inteligencia artificial». <https://www.espol.edu.ec/es/noticias/polit%C3%A9cnicos-crean-sistema-para-monitorear-el-distanciamiento-f%C3%ADsico-y-el-uso-de> (accedido jun. 09, 2021).
- [12] «Curso-I.pdf». Accedido: ago. 09, 2021. [En línea]. Disponible en: <https://www.matem.unam.mx/~roli/librogeom/Libro/Curso-I.pdf>
- [13] «scipy.spatial.distance.cdist — SciPy v1.7.1 Manual». <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.cdist.html> (accedido ago. 02, 2021).
- [14] «Tutorial de Python — documentación de Python - 3.8.10». <https://docs.python.org/es/3.8/tutorial/index.html> (accedido jun. 09, 2021).
- [15] R. C. Limited, *PyQt5: Python bindings for the Qt cross platform application toolkit*. Accedido: jun. 09, 2021. [En línea]. Disponible en: <https://www.riverbankcomputing.com/software/pyqt/>
- [16] «About», *OpenCV*. <https://opencv.org/about/> (accedido jun. 09, 2021).
- [17] «SQLite Home Page». <https://www.sqlite.org/index.html> (accedido jun. 09, 2021).
- [18] «YOLO: Real-Time Object Detection». <https://pjreddie.com/darknet/yolo/> (accedido jun. 09, 2021).
- [19] «SciPy.org — SciPy.org». <https://www.scipy.org/> (accedido jun. 09, 2021).

- [20] I. Coronel, «Líneas de investigación». <http://facsisstel.upse.edu.ec/index.php> (accedido jun. 09, 2021).
- [21] «MSP y OPS/OMS fortalecen el análisis de la información de la pandemia de COVID-19 - OPS/OMS | Organización Panamericana de la Salud». <https://www.paho.org/es/noticias/9-6-2020-msp-opsoms-fortalecen-analisis-informacion-pandemia-covid-19> (accedido jun. 09, 2021).
- [22] «Ministerio toma las medidas necesarias para el regreso opcional y progresivo a las escuelas rurales que tienen régimen Costa – Ministerio de Educación». <https://educacion.gob.ec/ministerio-toma-las-medidas-necesarias-para-el-regreso-opcional-y-progresivo-a-las-escuelas-rurales-que-tienen-regimen-costa/> (accedido jun. 09, 2021).
- [23] C. Alonso, A. David, M. Jara, y E. Alcides, «Visión por computador: identificación, clasificación y seguimiento de objetos.», p. 17.
- [24] «Plan Nacional de Desarrollo 2017 – 2021 Toda una Vida – Secretaría Técnica Planifica Ecuador». <https://www.planificacion.gob.ec/plan-nacional-de-desarrollo-2017-2021-toda-una-vida/> (accedido ene. 14, 2021).
- [25] R. Hernández Sampieri, C. Fernández Collado, y P. Baptista Lucio, *Metodología de la investigación*. México: McGraw Hill Interamericana, 2014.
- [26] C. S. Rojas Moreno, C. A. Bravo, J. S. Rincon, C. E. Montenegro-Marín, y P. A. Gaona-Gracias, «Viabilidad de las redes neuronales en la detección de objetos», *RISTI*, n.º 28, p. 20, abr. 2020.
- [27] J. E. M. Díaz, «Inteligencia Artificial y Big Data como soluciones frente al COVID-19», *Rev. Bioét. Derecho*, vol. 0, n.º 50, Art. n.º 50, jul. 2020, doi: 10.1344/rbd2020.50.31643.
- [28] «Planes Institucionales de Continuidad Educativa». <https://www.unicef.org/ecuador/planes-institucionales-de-continuidad-educativa> (accedido jun. 16, 2021).
- [29] «Estadísticas AMIE – Ministerio de Educación». <https://educacion.gob.ec/estadisticas-amie/> (accedido ago. 13, 2021).

- [30] «paper.pdf». Accedido: ago. 16, 2021. [En línea]. Disponible en: https://ddd.uab.cat/pub/tfg/2017/tfg_71066/paper.pdf
- [31] «TTFIC-2020-IS-DE00016.pdf». Accedido: ago. 16, 2021. [En línea]. Disponible en: <http://repositorio.utmachala.edu.ec/bitstream/48000/15982/1/TTFIC-2020-IS-DE00016.pdf>
- [32] F. Ruiz-Bertol y J. Dolado, *PARMA: Un Modelo Incremental para la Representación y Gestión de Proyectos en Ingeniería del Software*. 2003, p. 368.
- [33] S. I. Mariño y C. R. Primorac, «Propuesta metodológica para desarrollo de modelos de redes neuronales artificiales supervisadas», *IJERI Int. J. Educ. Res. Innov.*, n.º 6, Art. n.º 6, may 2016.
- [34] J. D. S. Vázquez, «Aplicación del Modelo Incremental Para el Desarrollo del Sistema de Información Docente», p. 107.
- [35] «Escuela De Educacion Basica Angelica Villon Lindao. · 18 de agosto entre paquisha y, Fausto Fajardo, Santa Elena», *Escuela De Educacion Basica Angelica Villon Lindao. · 18 de agosto entre paquisha y, Fausto Fajardo, Santa Elena*. <https://www.google.com/maps/place/Escuela+De+Educacion+Basica+Angelica+Villon+Lindao./@-2.227096,-80.8543633,15z/data=!4m5!3m4!1s0x0:0x3c424bf57fce9012!8m2!3d-2.2271205!4d-80.8542583> (accedido jul. 19, 2021).
- [36] «MINEDUC-MINEDUC-2020-00044-A.pdf». Accedido: jul. 14, 2021. [En línea]. Disponible en: <https://educacion.gob.ec/wp-content/uploads/downloads/2020/09/MINEDUC-MINEDUC-2020-00044-A.pdf>
- [37] «Protocolos de autocuidado e higiene – Ministerio de Educación». <https://educacion.gob.ec/protocolos-de-autocuidado-e-higiene/> (accedido jul. 14, 2021).
- [38] «El concepto de IDE». <https://www.redhat.com/es/topics/middleware/what-is-ide> (accedido jul. 12, 2021).
- [39] J. C. Moreno Pérez, *Entornos de desarrollo*. Madrid: Síntesis, 2018.

- [40] «Visual Studio Community 2019 - Free IDE and Developer Tools». <https://visualstudio.microsoft.com/es/vs/community/> (accedido jul. 06, 2021).
- [41] «CUDA Zone», *NVIDIA Developer*, jul. 18, 2017. <https://developer.nvidia.com/cuda-zone> (accedido jul. 06, 2021).
- [42] «capitulo-2.pdf». Accedido: ago. 17, 2021. [En línea]. Disponible en: <http://ocw.uc3m.es/ingenieria-de-sistemas-y-automatica/sistemas-percepcion/material-de-clase-1/libro/capitulo-2.pdf>
- [43] «IME_218.pdf». Accedido: ago. 17, 2021. [En línea]. Disponible en: https://pirhua.udel.edu.pe/bitstream/handle/11042/2916/IME_218.pdf
- [44] Á. M. G. García, «Aplicación de percepción mejorada para personas con problemas visuales usando deep learning y dispositivos vestibles», p. 69.
- [45] «YOLO-Introducción-e-implementación-.pdf». Accedido: jul. 05, 2021. [En línea]. Disponible en: https://biorobotics.fi-p.unam.mx/wp-content/uploads/Courses/reconocimiento_de_patrones/tutoriales/YOLO-Introducci%C3%B3n-e-implementaci%C3%B3n-.pdf
- [46] C. S., «Cómo mantener el patrón modelo vista controlador en una aplicación orientada a la WEB», *Rev. Inven.*, vol. 4, p. 72, jul. 2009, doi: 10.26620/uniminuto.inventum.4.7.2009.72-78.
- [47] «UML - Casos de uso.pdf». Accedido: jul. 27, 2021. [En línea]. Disponible en: <https://repositorio.grial.eu/bitstream/grial/1155/1/UML%20-%20Casos%20de%20uso.pdf>
- [48] F. J. G. Peñalvo y C. P. Aguilar, «Diagramas de Clase en UML 1.», p. 8.
- [49] «¿GPU vs. CPU? ¿Qué es la computación por GPU? | NVIDIA». <https://www.nvidia.com/es-la/drivers/what-is-gpu-computing/> (accedido ago. 10, 2021).
- [50] «Installing | CMake». <https://cmake.org/install/> (accedido jul. 20, 2021).

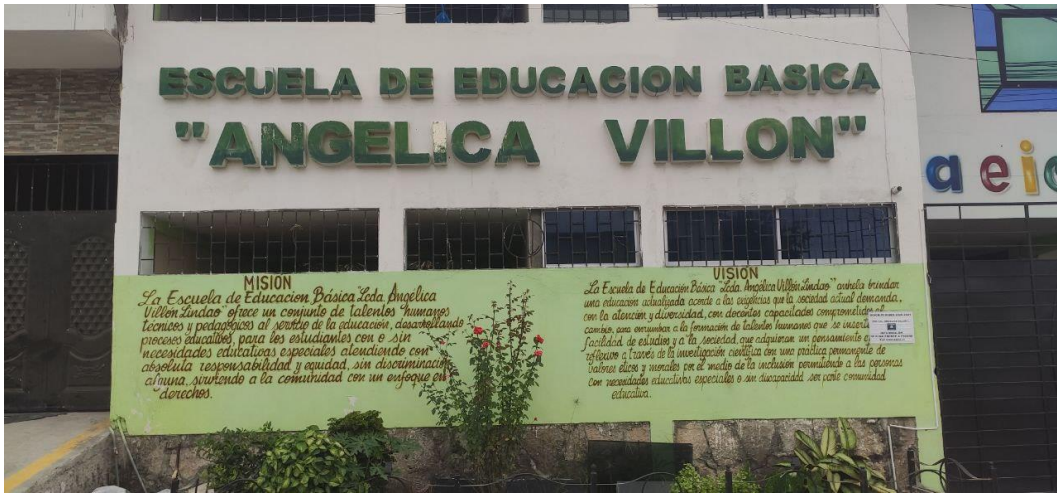
[51] «GitHub - opencv/opencv_contrib: Repository for OpenCV's extra modules», *GitHub*. https://github.com/opencv/opencv_contrib (accedido jul. 20, 2021).

[52] «object-detection-yolo-opencv/yolo.py at master · nandinib1999/object-detection-yolo-opencv», *GitHub*. <https://github.com/nandinib1999/object-detection-yolo-opencv> (accedido jul. 22, 2021).

[53] A. Dhiman, «YOLOv3 Object Detection in TensorFlow 2.x», *Analytics Vidhya*, nov. 13, 2020. <https://medium.com/analytics-vidhya/yolov3-object-detection-in-tensorflow-2-x-8a1a104c46a8> (accedido jul. 29, 2021).

ANEXOS

Entrevista dirigida al Arq. Mauricio Yunda Coordinador Administrativo de la Escuela de Educación Básica Angélica Villón	
Objetivo: Conocer cómo se realizaría el proceso de distanciamiento social en las instalaciones del plantel en el regreso a las clases presenciales.	
1	Describa brevemente la historia de la escuela
2	Describa el proceso de entrada, recreo y salida de los alumnos.
3	¿Cree que su proceso es el indicado en estos tiempos de pandemia?
4	¿Cuántos alumnos tiene matriculados en el presente periodo lectivo?
5	¿Cuántas jornadas de estudio dispone actualmente?
6	¿Cuántas personas conforman la plantilla docente y administrativa?
7	¿Hay alguna persona encargada de mantener el orden en el patio del plantel?
8	¿Con que frecuencia realiza actividades que concentre a alumnos, padres y profesores?
9	¿Qué medidas tomarán como directivos del plantel para acatar las medidas de bioseguridad decretados por las autoridades de gobierno?
10	¿Cree usted que contar con un sistema que automatice el control de distanciamiento social ayudaría a cumplir con esta medida?



Anexo 2. Misión y Visión de la Escuela de Educación Básica "Angélica Villón"



Anexo 3. Reunión con directora de la escuela



Anexo 4. Presentación del sistema a la directora del plantel.



Anexo 5. Patio de la escuela



Anexo 6. Muestra de fotogramas con flexómetro en varias posiciones.

Santa Elena, 2 de agosto de 2021

Lcda. Ibelice Tomalá Villón
Directora Académica
ESCUELA DE EDUCACIÓN BÁSICA "LCDA. ANGÉLICA VILLÓN LINDAO"
En su despacho.

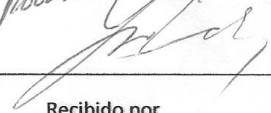
Yo, Katherine Alejandro Roca, portadora de la cédula de ciudadanía N° 0928412659, por medio de la presente me dirijo a usted, para solicitar me conceda el ingreso al establecimiento educativo para realizar la instalación y prueba del **SISTEMA DE DETECCIÓN DE DISTANCIAMIENTO SOCIAL**. La etapa de instalación y prueba es uno de los puntos clave dentro de mi tesis de titulación como Ing. en Tecnologías de la información que me permitirá tomar evidencia del trabajo realizado y corregir posibles errores de ejecución, además se hará la presentación formal del sistema y forma de funcionamiento.

Agradezco la atención que tenga el presente.

Atentamente,

KATHERINE ALEJANDRO R.

Katherine Alejandro Roca
katherine.alejandroroca@upse.edu.ec
Telf: 0964029670

RECIBIDA 03/08/21


Recibido por

