ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

“Diseño y simulación de sistema de videovigilancia para el monitoreo de cultivos y control de acceso del personal”

PROYECTO INTEGRADOR

Previo a la obtención del Título de:

Ingeniero en Telecomunicaciones

Presentado por:

Jorge André Cabrera Savinovich
Dennys Saúl Pérez Sandoval

GUAYAQUIL – ECUADOR
Año: 2022

DEDICATORIA

El presente proyecto se lo dedico a Dios, que ha estado conmigo omnipresente en todo momento, en cada caída y en cada alegría. Y a mis padres Eliana Savinovich y Jorge Cabrera que con su apoyo incondicional siempre me alientan a nunca rendirme y ser humilde para llegar al éxito.

Jorge André Cabrera Savinovich

Este trabajo titulación va dedicado a mi madre Janeth Sandoval quien ha puesto confianza y determinación en mi toda la vida, por la motivación y apoyo durante toda mi carrera universitaria. A mi tío Miguel Arguello por haberme inspirado con ejemplos de perseverancia y fortaleza quien ahora me cuida desde el cielo.

Dennys Saúl Pérez Sandoval

AGRADECIMIENTOS

Agradezco a Dios por brindarme siempre el camino de la solución a cada obstáculo durante el proceso, a mis padres por sus aportes emocional y económico, a mi novia por su amor y tiempo brindado de forma incondicional, al Master Eduardo Chancay por sus consejos y aportes académicos, y a mi compañero de tesis por su dedicación y responsabilidad durante el proceso.

Jorge André Cabrera Savinovich

Quiero dar las gracias a Dios por darme salud y permitirme culminar esta etapa académica. A mi madre por todo el apoyo durante mi vida. A mi familia Sandoval por la constante motivación. Agradezco a mi compañero de tesis por su intachable responsabilidad y dedicación en la elaboración de este trabajo, y a mis mejores amigos Daniel Gusnay, Byron Alvarado, Joel calle y Gabriel Velásquez con quienes he vivido y compartido buenos momentos de estudios y diversión en estos años de carrera universitaria.

Dennys Saúl Pérez Sandoval

DECLARACIÓN EXPRESIVA

“Los derechos de titularidad y explotación, nos corresponde conforme al reglamento de propiedad intelectual de la institución; Jorge André Cabrera Savinovich Y Dennys Saúl Pérez Sandoval damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual”

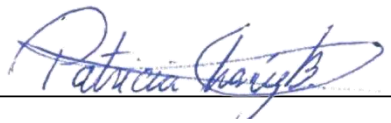
Jorge Cabrera Savinovich

Dennys Pérez Sandoval

EVALUADORES

PhD. Washington Media Moreira

PROFESOR DE LA MATERIA



PhD. Patricia Chávez Burbano

PROFESOR TUTOR

RESUMEN

El presente proyecto de titulación consiste en el diseño y simulación de un sistema de videovigilancia para el monitoreo de cultivos y control de acceso del personal de la industria PortiArroz S.A, el mismo resulta ser un sistema eficiente que brinda la seguridad necesaria para evitar el hurto de cultivos y restringir el acceso de personas no autorizadas a la industria. Previo al diseño, se receptó la información referente a los inconvenientes presentados y los requerimientos a cumplir solicitados por la gerencia de la industria. Para esto, se simuló un sistema general de seguridad que abarcó dos etapas: reconocimiento facial y monitoreo de cultivos con notificación de alerta.

La primera etapa es capaz de realizar un reconocimiento de características faciales a personas incorporando varios procesos: registro, verificación y marcación de usuario permitiendo el ingreso a la industria al personal autorizado; el componente electrónico principal es una Raspberry Pi en donde se instaló el sistema operativo necesario denominado Raspbian, empleando el desarrollo y ejecución del algoritmo de reconocimiento facial “Eigen Faces”.

La segunda etapa se enfoca en el sistema de videovigilancia a través de cámaras y envío de mensajes de alerta de forma remota por medio de la aplicación WhatsApp. Esta etapa fue simulada con dos cámaras USB con características de alto rendimiento tanto en calidad de imagen como en alcance, conectadas a un microprocesador Raspberry Pi en donde se instaló el sistema operativo de monitoreo MotionEyes configurando los parámetros necesarios para la grabación de video. Se añadió un envío automático de alertas al activar los sensores conectados al módulo ESP32 estableciendo una conectividad remota con un aplicativo creado para la activación de alarma. En esta parte del sistema se podrá monitorear todos los sucesos que ocurren en las áreas específicas.

El sistema de seguridad es eficiente y de bajo costo, el cual podrá ser empleado como una herramienta de gran importancia permitiendo salvaguardar la integridad de los trabajadores dentro de la industria, así como de evitar el hurto de cultivos ya que se avala una supervisión y asistencia de personal en su totalidad.

Palabras Claves: Simulación, Raspberry, ESP32, Servidor.

ABSTRACT

The following undergrad capstone project consists of the design and simulation of a video surveillance system for monitoring crops and controlling access of the personnel in the industry PortiArroz SA, it turns out to be an efficient system that provides the necessary security to avoid the theft of crops and restrict the access of unauthorized persons to the industry.Prior to the design, the information regarding the problems presented and the requirements to be fulfilled requested by the industry management was received. For this, a general security system was simulated that encompassed two stages: facial recognition, monitoring and alert notification.

The first stage can recognize people's facial characteristics, incorporating various processes: user registration, verification and marking, allowing authorized personnel to enter the industry; The main electronic component is a Raspberry Pi where the necessary operating system called Raspbian was installed, the development and execution of the "Eigen Faces" facial recognition algorithm.

The second stage focuses on the video surveillance system through cameras and the sending of alert messages remotely through the WhatsApp application. This stage was simulated with two USB cameras with high-performance characteristics in both image quality and range, connected to a Raspberry Pi microprocessor where the MotionEyes monitoring operating system was installed, configuring the necessary parameters for video recording. Automatic sending of alerts was added when activating the sensors connected to the ESP32 module, establishing remote connectivity with an application created for alarm activation. In this part of the system, it will be possible to monitor all the events that occur in the specific areas.

The security system is efficient and inexpensive, which can be used as a very important tool allowing safeguarding the integrity of workers within the industry, as well as avoiding the theft of crops as it is endorses a supervision and assistance of personnel in its entirety.

Keywords: *Simulation, Raspberry, ESP32, Server.*

ÍNDICE

ABREVIATURAS

IP	Protocolo de internet.
IOT	Internet de la Cosas
MQTT	Brokers de Transporte de Telemetría
HD	Alta Definición
DSI	Sistemas de Datos Internacionales
SNMP	Protocolo Simple de Manejo de Red
MIB	Base de Gestión de Información
GPRS	Paquetes Generales de Radio Servicio
HMI	interfaz Humano-Maquina
ICA	Análisis de Componentes Independientes
BSC	Ordenador de Placa Simple
PCA	Análisis de Componentes Principales
LAN	Red de Área Local.
GPIO	Pines de entrada / salida de uso general.
CCTV	Circuitos cerrados de televisión.
PoE	Alimentación por medio de cable Ethernet.
LDA	Análisis Discriminante Lineal
PTZ	Pan Tilt Zoom
DVR	Grabadora de Video Digital
NVR	Grabador de Video en la Red

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

CAPÍTULO 1

1. INTRODUCCIÓN

El avance tecnológico debido a la inseguridad en el país ha ido creciendo a un paso elevado permitiendo que gran parte de empresas, industrias, hogares y sectores financieros requieran de estos servicios para salvaguardar sus entidades o establecimientos utilizando equipos que permitan brindar la seguridad requerida. Existen empresas externas que brindan protección utilizando personal profesional enfocados en el control y seguridad de los establecimientos, pero no es lo suficientemente eficiente y el costo de contratación es elevado.

En la actualidad muchas empresas e industrias de varios sectores utilizan diferentes medios de seguridad para la protección de sus bienes y colaboradores, por ejemplo: cercado eléctrico, cámaras IP conectadas a una central para monitorear, sensores de movimiento o infrarrojo, etc. Otras empresas contratan personal de seguridad para el mismo fin ya que cada año se va incrementando la inseguridad en el país [1]. La Industria “Portiarroz S.A.” la cual se dedica al cultivo, venta de arroz y cacao ha venido presentando inconvenientes de seguridad en la actualidad ya que han notado faltantes en los cultivos y problemas en el control de acceso a los departamentos. Actualmente cuentan con un cercado para tratar de impedir que ingresen y roben, pero no es lo suficientemente seguro. por ende, necesitan una solución eficiente e innovadora para aminorar en su totalidad estos inconvenientes.

Teniendo claro el problema y planteando la solución para mejorar la seguridad en la industria, se propone diseñar un sistema de seguridad para prevenir el robo en las parcelas de cultivos y evitar pérdidas económicas. además, un sistema de reconocimiento facial para el ingreso del personal evitando el acceso de personas desconocidas y posibles robos, en el cual se utilizará dispositivos principales como el Raspberry Pi siendo un microrordenador de tarjeta reducida [2], y un módulo ESP32 integrado con Wifi y Bluetooth.

Además, el uso de base de datos y dispositivos finales para la detección de movimiento [3].

1.1 Descripción del problema

En Ecuador la delincuencia se ha vuelto un problema diario que afecta gravemente la seguridad de la población. El índice de delincuencia se ha visto incrementado notoriamente en los últimos meses, cada día se registran atracos, robos, amenazas, etc., tanto en negocios como a transeúntes. La recesión económica, la falta de trabajo y la inmoralidad son las principales causas que generan un entorno ideal para el florecimiento de estas actividades vandálicas [4]. Según datos estadísticos de la fiscalía general del Estado, las cifras registradas de robos en Ecuador entre Enero y Agosto del 2020 en comparación con el mismo período del 2021 ha incrementado aproximadamente 30% [5].

Las pequeñas y medianas empresas se están viendo especialmente vulnerables a este tipo de incidentes. De acuerdo con datos estadísticos, los responsables en entornos urbanos son bandas que operan en las ciudades de Quito, Guayaquil, Quevedo y Machala,utilizando armas de fuego y cuchillo [6] . De igual manera, empresas que se dedican al sector agrícola como la ganadería y cultivo, no se han vistos alejados de esta realidad [7].

Actualmente, la industria "Portiarroz S.A." ubicada en milagro, ciudad de Ecuador, se dedica a la comercialización de cultivos de arroz y cacao para venta nacional y exportación. Esta presenta inconvenientes de seguridad dado que frecuentemente personas externas cruzan las parcelas, ingresan a las áreas de cultivo y hurtan el cacao o arroz. Al ser una industria que abarca varias hectáreas de cultivo se dificulta la vigilancia en su totalidad por parte de los trabajadores. Esta hacienda al no contar con un sistema de seguridad es propensa a robos con frecuencia [8].

Hasta la fecha no se ha podido aminorar estos delitos ya que la cantidad de trabajadores no es suficiente para solventar una adecuada vigilancia en todas

las hectáreas. Por ello, es de gran importancia el desarrollo de un sistema integral de seguridad que permita salvaguardar la integridad de sus trabajadores, así como de evitar los robos de cultivos.

1.2 Justificación

La inseguridad que enfrenta Ecuador actualmente es relevante ante los numerables casos de robos a transeúntes, locales comerciales, instituciones y empresas. Esto plantea la cuestión de cuál sería la mejor forma de solventar y evitar estos inconvenientes. El acceso a personas no autorizadas puede ser evitado en cuanto los establecimientos determinen e implementen algunas opciones de seguridad como los sistemas de videovigilancia.

Actualmente hay un alto porcentaje en que estas organizaciones no cuentan con recursos respecto al tema de seguridad, por ello están expuestas permanentemente tanto a casos inoportunos dentro del establecimiento por parte de los trabajadores como las amenazas que se presentan por personas externas [9].

Debido a la falta de seguridad y dado el requerimiento de incorporar nuevas tecnologías que garanticen la seguridad en la industria “Portiarroz”, se propone implementar un proyecto, enfocado en el diseño y simulación del sistema de seguridad, el cual busca cubrir un extenso control de vigilancia en áreas de cultivo y tener un control de acceso por medio de reconocimiento facial a trabajadores en la industria, de tal manera que se cumpla con la necesidad de tener una supervisión eficiente.

En base a los requerimientos del cliente, mediante el sistema se podrá observar los sucesos a través de las cámaras y de forma remota mediante un teléfono móvil. Se realizará una tarjeta principal la cual tendrá un microcontrolador que será el encargado de recibir las señales de los sensores y como resultado mediante señales de salida activará los equipos encargados de alertar al usuario los cuales son: sirena y mensaje de texto.

Además, para el control de ingreso del personal se implementará un sistema de reconocimiento facial, mediante una tarjeta microcontroladora, desarrollo de algoritmos y programación en Python enlazada a una base de datos web, el cual permitirá tener un control de acceso y asistencia del personal de trabajo en la industria.

Con este sistema se espera avalar una supervisión en su totalidad que permita el monitoreo de vigilancia constantemente durante todo el día y así evitar el robo de equipos y de cultivos por atracadores.

1.3 Objetivos

1.3.1 Objetivo general

- Diseñar y simular un sistema integral de seguridad y reconocimiento facial controlado de forma inalámbrica para el monitoreo del cultivos y control de acceso de los trabajadores en la industria.

1.3.2 Objetivos específicos

- Diseñar un sistema de acceso que facilite el registro e ingreso del personal reduciendo el tiempo de verificación y validación volviéndolo eficiente y escalable.
- Elaborar un sistema de detección y comunicación mediante Wifi entre dos áreas específicas para mantener un monitoreo visual en caso de robo y activar una sirena como mecanismo de seguridad.
- Determinar estratégicamente los lugares donde se colocarán las cámaras, sensores, alarmas, computador principal y dispositivos adicionales a utilizarse para un rendimiento eficiente.
- Monitorear las áreas de cultivo desde un dispositivo e identificar de forma eficiente el lugar donde se lleva a cabo el hurto para posteriormente poder activar una alerta sonora y evitar pérdidas de cultivos.

- Desarrollar plan de pruebas para evaluar el tiempo de retardo en la comunicación entre dispositivos y a su vez, analizar la continuidad del sistema utilizando redes inalámbricas Wifi.

1.4 Estado del arte

El progreso de la tecnología referente a sistemas de seguridad ha llevado a que hogares, diferentes tipos de negocios y empresas adquieran los equipos necesarios para salvaguardar la vida de las personas y bienes materiales. Hay distintas formas en que se puede desarrollar e implementar un sistema de seguridad mediante cámaras de videovigilancia.

La gran variedad y demanda de seguridad que se necesita en la actualidad ha proporcionado la implementación de modelos electrónicos para seguridad. En la cuidad de Ambato se implementó un sistema de videovigilancia mediante protocolo de internet (IP) con la finalidad de ofrecer mayor seguridad en un barrio “las delicias”. Este sistema estaba integrado de switches, alarmas y una central de monitoreo de las cámaras. Esto representó una solución viable dado que el monitoreo de las cámaras fue de forma remota, los costos no son elevados, representando flexibilidad y escalabilidad [10].

Considerando diferentes puntos de vista, como los sistemas inteligentes en viviendas, se puede implementar otros diseños integrados, como por ejemplo sistemas embebidos con conexiones de internet de la cosas (IOT) [11], con la finalidad de convertir una casa o empresa en un sistema domótico para reducir los pagos excesivos de luz, mediante mecanismos de electrónica utilizando diferentes microprocesadores, una cámara de video, sensores de movimiento, relays, diodos y servo motor permitiendo el control del consumo eléctrico. Siendo una solución muy viable ya que se basan en brokers de transporte de telemetría (MQTT), aplicaciones de monitoreo remoto, y dispositivos electrónicos de bajos costos.

Por otra parte, [12] se basa en un sistema de seguridad Wireless con un microprocesador Arduino Yun, empleándolo para la intercomunicación con los demás equipos, utilizando un sensor de movimiento HC-SR501 conectado al Arduino, y una cámara Logitech C270 de alta resolución para el monitoreo. Se pudo analizar que algunos de los equipos utilizados como el Arduino Yun brinda una gran eficiencia respecto a la comunicación de equipos, pero no cuenta con capacidad de memoria suficiente para desarrollar las funciones que se prevé como solución ante la problemática descrita.

En España donde la electrónica está avanzada y la seguridad frágil. Se ha llegado a implementar el desarrollo de un sistema electrónico de monitoreo en Android [13]. El cual trata de una App que permite registrar los datos del cliente y almacenarla en una base de datos que se encontrará en el servidor. Una vez ingresado al App con el respectivo usuario y contraseña se puede solicitar la detección de movimiento por medio de los sensores, además de observar en tiempo real lo que la cámara filma. Se hace uso un Raspberry Pi, computadores, móbiles, Software de desarrollo Android Studio, MySQL, VLC (VideoLAN Client).

Permitiendo darle un excelente sistema de seguridad al cliente con equipos y softwares de alta gama, siendo una de las propuestas con mejores ideas para implementar debido al uso de VLC media player que es un reproductor y framework multimedia, libre y de código abierto, microcomputador Raspberry Pi de bajo costo y fácil codificación, y el uso de Base de Datos MySQL de código abierto y respaldado por la empresa Oracle [13].

La tecnología está creciendo de manera exponencial y los sistemas de seguridad siguen el paso [14]. En Madrid la delincuencia, el robo y el asesinato ha incrementado de a poco [15]. Por ende, se ha desarrollado un proyecto de la Universidad Politécnica de Valencia sobre un mecanismo de seguridad con bajo consumo de energía utilizando Raspberry Pi [16] permitiendo la detección de intrusos, de manera que el propietario puede estar al tanto en todo momento de la situación en la zona controlada. Este sistema es capaz de detectar intrusos, encender alarma, hacer fotos y enviarlas por mensaje a un dispositivo, y grabar video durante tiempo determinado.

Una de las características que cumple un factor fundamental en hogares y empresas es la seguridad mediante videovigilancia. La Universidad de Medellín ubicada en Colombia implementó el diseño de un sistema de seguridad con visión artificial utilizando microcontroladores como Raspberry Pi, software Open Source Linux, Python, lenguaje de programación OpenCV para la manipulación de imágenes, cámara full de alta definición (HD) y una pantalla táctil de 3,5 pulgadas [17]. Este proyecto permitió fortalecer la seguridad de las personas integrando un mecanismo de ingreso a personas mediante reconocimiento facial. Siendo de utilidad para implementarlo en diferentes ambientes gracias al uso de librerías de código abierto que utiliza el microcomputador Raspberry Pi.

El uso de inteligencia artificial y la nube en internet son medios que se utilizan por empresas para optimizar recursos, por lo cual, en la Escuela Superior Politécnica del Litoral (ESPOL) se analizó el diseño de un prototipo de sistema de comunicación interactivo para innovar la educación [18]. Este constaba de un microcontrolador Raspberry Pi para la ejecución de comandos y dispositivos externos como cámara, micrófonos y pantalla, permitiendo el registro de asistencia utilizando reconocimiento facial y envío de material de estudio digital por medio de la nube.

Resultando ideal para la implementación ya que entre sus características posee: Bluetooth +Wifi, salida estéreo y video compuesto, Power over Ethernet (PoE), conectores de interfaces multimedia de alta definición (HDMI) y puerto de sistemas de Datos Internacionales (DSI) para pantalla. Además, utiliza software OpenCv cuyo sistema operativo es Raspbian [18].

Existen otros tipos de microcontroladores como por ejemplo el ESP32 [19]. El cual permite la comunicación a través de Wifi y Bluetooth y mediante un servidor web se le puede crea una AppWeb que se comunique con el microcontrolador y realice las funciones que el usuario desee. Este proyecto se basa en envió de notificaciones mediante telegram cuando detecte algún movimiento utilizando un sensor de movimiento HC-SR501 y como resultado mediante un relay se encenderá un foco y enviará el ESP32 la notificación a usuario registrado mediante telegram.

La visualización por medio de equipos computarizados en los últimos años se ha elevado de manera significativa debido a la manipulación de herramientas electrónicas las cuales han dado la facilidad al uso de mecanismos para reconocer rostros y algoritmos de aprendizaje automatizado para el respectivo control de detección. [20].

En el 2018 en la Universidad de las Fuerzas Armadas implementó un prototipo de equipo inteligente para poder identificar personas mediante un video en tiempo real [21], permitiendo alertar e informar al operador o dueño mediante una alarma, la presencia de un individuo en el video tomado. El sistema estaba compuesto de un dispositivo microcomputador Raspberry Pi configurado con librerías de machine learning. Utilizaron tres tipos de algoritmos detectores de rostro y La captura del video se procesó mediante una cámara inteligente.

En los últimos años, en todo el mundo se ha presentado un avance tecnológico importante que implica la implementación de nuevos dispositivos para el desarrollo de proyectos de alta gama, esto involucra nuevas tecnologías que son utilizadas en sistemas de seguridad [22].

En diversos trabajos electrónicos se utilizan sistemas de comunicación digital, en el cual involucran diversos microcontroladores como el módulo 18F450 que incorpora [23] como procesador principal de la central de alarma en un sistema de seguridad; esta recepta las señales de los equipos para la respectiva comunicación.

Por lo general en este tipo de sistemas implementan contactos magnéticos que funcionan como switch, de tal manera que al abrir alguna puerta el circuito eléctrico se cierre y se pueda detectar la apertura no autorizada. Además [23] integra en su sistema sensores de proximidad para detección de movimiento y otros módulos como el modem GSM el cual es un sistema global para móviles permitiendo recibir un aviso por mensaje a un teléfono móvil, monitoreando todos estos sucesos a través de cámaras IP que estén conectadas a una switch conectadas a la computadora principal

En variados trabajos electrónicos utilizan protocolos de administración de red, en Chimborazo se implementó un método de vigilancia que involucra normas

técnicas similar a [23] pero utilizando cámaras IP mediante un protocolo simple de manejo de red (SNMP); haciendo uso del navegador SNMP que utiliza una Base de Gestión de Información (MIB).

Estos sistemas utilizaron una alarma con paquetes generales de radio servicio (GPRS) mediante un módulo GSM para avisar a un móvil cualquier suceso detectado por medio de cámaras IP. Todos los dispositivos trabajaron mediante IP con estándares abiertos. La utilización de estos protocolos permitió tener un sistema eficiente ya que facilitó el monitoreo y comunicación entre los dispositivos conectados en la red de área local (LAN) con una mayor estabilidad y convergencia de datos [24].

Existe variabilidad de sistemas que implican el uso de estándares específicos [25]. En la cuidad de Quito, ESPE implementó el estándar LONWorks en sistemas de seguridad como plataforma tecnología [26], permitiendo integrar varios sistemas como: control de acceso, detección de incendios y sistema intrusión, con el fin de salvaguardar el bienestar de las personas, así como de los materiales que hay en el lugar.

Por lo que respecta al sistema de detección de incendios se utilizó un sensor de humo ubicado en la planta baja de los edificios. Para el sistema electrónico de entrada al edificio, se colocó un lector de proximidad, pulsador de salida en cada puerta con una cerradura electromagnética. Y por último para el sistema anti-intrusión se utilizó contactos magnéticos y sensores de movimientos en cada puerta y/o ventana de interés activando las alarmas en caso de presentarse algún suceso; el estado de cada uno de los sensores se puede visualizar mediante la interfaz Human-Machine (HMI) [26].

La utilización de este estándar conllevó un alto costo ya que incorpora el manejo de varios sistemas en una sola plataforma, lo cual para el sistema implementado en [12] no resulta ser competente.

Los sistemas informáticos que se implementan actualmente siguen en tendencia a la miniaturización, es decir, tecnologías de compresión de los equipos que se utilizan en diversos procesos. Parte de ello se atribuye a los sistemas de control de acceso que van en aumento a su implementación en empresas para un

monitoreo eficiente del ingreso del personal de trabajo, proporcionando un mejor control, gestión y asistencia en actividades [27].

Por otra parte, en el proyecto [28] se implementó un sistema de control de acceso a una empresa mediante reconocimiento facial, el cual se basa en un sistema biométrico y el procesamiento digital de imágenes, haciendo uso de una cámara digital que toma fotos analizando los rasgos físicos de la imagen y los compara con imágenes almacenadas en una base de datos, permitiendo reconocer los patrones en el sistema estableciendo la identificación y caracterización única de cada persona.

El sistema biométrico consta del análisis de componentes independientes (ICA) que calcula los vectores de la base de datos de acuerdo con las características faciales de las personas, adecuando las dimensiones de la imagen por medio del Análisis Lineal de discriminante (LDA). Esto permitió tener un control de acceso eficiente y preciso de trabajadores, evitando el ingreso de personas no autorizadas [28] .

A diferencia de lo antes mencionado, en este proyecto se presenta la simulación de un sistema integral de seguridad que permitirá realizar un reconocimiento facial para el ingreso a la industria, así como la vigilancia en tiempo real a través de cámaras con un sistema de alerta mediante notificación móvil. Este sistema se desarrolla a base de algoritmos de programación en un entorno integral (IDE) en equipos electrónicos.

1.5 Alcance

Este proyecto tiene como propósito elaborar un prototipo de sistema de seguridad en base a diseños electrónicos y simulaciones. El mismo se desarrollará para ser implementado en las hectáreas de cultivos en la industria Portiarroz S.A ubicada en el cantón Milagro. Actualmente esta cuenta con departamentos donde hay servicio de internet: el departamento de jefatura, recursos humanos y calidad, en donde trabajan 8 colaboradores. Para el área de cultivo de cacao hay 9 trabajadores que se encargan de la supervisión y cultivo en las hectáreas.

Para el desarrollo de este prototipo se utilizará una Raspberry Pi como microordenador y otro microcontrolador denominado módulo EPS32. Adicionalmente, dispositivos electrónicos como cámaras, sensores de movimiento y una alarma; estos conectados y configurados con los algoritmos correspondientes, lo cual permitirán un correcto control de acceso a la industria por medio de reconocimiento facial y un eficiente sistema de vigilancia en los cultivos, permitiendo la asistencia inmediata de trabajadores al área de cultivo donde se presente inconvenientes.

Utilizando el tipo de entrenamiento Eigenfaces como reconocedor facial para identificar al personal, se permitirá tener un control de acceso de trabajadores y personas externas como medida de seguridad, el cual constará en un informe detallado dentro una base de datos MySQL actualizado en tiempo real para ser revisado por los departamentos encargados. Y mediante el sistema de vigilancia se obtendrían mensajes de alerta enviados a los usuarios principales que permitan la asistencia del personal y monitoreo mediante cámaras en el área de cultivo afectada; se permitirá el acceso a la central para el monitoreo con dispositivos móviles mediante TeamViewer.

1.6 Metodología

Como primer punto se inspeccionan las áreas de cultivo para hacer un análisis del terreno y determinar los puntos claves en donde se ubicarán los componentes electrónicos a utilizarse. En base a esto, se realizarán los diagramas electrónicos considerando las dimensiones de cada área de cultivo del cual se pueda precisar la cantidad de material adicional como cables a ocupar.

Luego de haber realizado la inspección de los cultivos se monitoreará las entradas y salidas en la industria para precisar el punto exacto donde se colocará el equipo junto a la cámara para el respectivo control de acceso. Una vez concluido ambos procesos de inspección, se armarán los circuitos correspondientes tanto del sistema de vigilancia como del control de acceso. Se empieza con la configuración de los algoritmos respectivos en la raspberry PI,

así como la programación del módulo EPS32 el cual permitirá la comunicación de forma inalámbrica entre los componentes. Luego de haber comprobado su funcionalidad, se procede a realizar las tarjetas impresas de los circuitos.

Se realiza la ubicación de la central de monitoreo del sistema en el departamento gerencial de la industria. Esta estará compuesta por la raspberry PI conectada a una computadora principal de donde se llevará a cabo el monitoreo mediante las cámaras de seguridad.

Para el sistema de control de acceso primero se crea una base de datos que contengan los rasgos faciales de cada trabajador de la empresa. Después, se configura el algoritmo del tipo de entrenamiento en la raspberry para comparar los rasgos faciales con la base de datos y permitir el acceso. Para efectuar una interacción entre el sistema de reconocimiento facial y trabajador, se desarrolla una interfaz interactiva en un software de programación. Se configura una pantalla touch que será el medio por el cual se realizará la interacción.

CAPÍTULO 2

2. Marco teórico

En este capítulo se detallarán equipos y elementos preseleccionados para el diseño y simulación del proyecto, acotando las características y funcionalidad de ellos.

2.1 Microcontroladores

Un microcontrolador se lo conoce como un circuito integrado programable, tiene la capacidad de ejecutar las órdenes grabadas en su procesador. Posee en su interior las tres unidades funcionales que son las más importantes de una computadora: unidad central de procesamiento, memoria y periféricos de acceso. Se detallará a continuación las familias de microcontroladores importantes [29].

2.2 Raspberry pi modelo 1

Este modelo de Raspberry fue creado en el 2012, fue el primer modelo en ser vendido. Como se puede observar en la Figura 2.1, viene integrado con 26 pines GPIO los cuales permiten comunicación con diferentes equipos y sensores electrónicos, posee procesador Broadcom de tipo BCM2835 con un único núcleo de 700MHz, viene adaptado con salida de video HDMI, RCA (Radio Corporation of America) y LCD para salida gráfica en Display, cuenta con 256 MB de SDRAM, 1 puerto ethernet, porta tarjeta SD para el sistema operativo, salida de audio mediante conector Jack de 3,5 mm y se requiere de una fuente de poder de 5 volteos y 2 amperios.

Figura 2.1 Raspberry Pi Modelo 1 [18] [30]

2.3 Raspberry pi Modelo 2

Este modelo fue estrenado y lanzado en el 2014, como se visualiza en la Figura 2.2, posee cuatro puertos USB, un puerto ethernet, salida de video HDMI, tiene 40 pines GPIO entre los cuales permiten las conexiones SPI, UART e I2C.

Figura 2.2: Raspberry Pi Modelo 2 [18] [30]

Modelo de SOC BCM2836 diferente al modelo anterior ya que cuenta con un procesador ARM Cortex A7 el cual tiene cuatro núcleos de 900MHz, viene adherido un porta tarjeta microSD de memoria, se amplió la memoria SDRAM a 1 GB, cuenta con gráfica Broadcom de videoCore IV y se suprime la conexión RCA.

2.4 Raspberry pi modelo 3

Este modelo como los anteriores tienen algunas extensiones, en este caso hablaremos del modelo 3 B+. Como se puede visualizar en la Figura 2.3, posee cuatro puertos USB, se expande la potencia SOC Broadcom BCM287 con un procesador ARMv8 de cuatro núcleos a 1.4 GHz, este modelo se le incorporó Bluetooth y WiFi 802.11n para mejorar la conectividad y trabaja en doble banda

a 2,4 GHz y 5GHz. Se mantienen los puertos HDMI de salida de video, puerto ethernet con una mejora en su velocidad de transmisión de 100 Mbits/s a 300Mbits/s y se mantienen los pines GPIO con sus respectivas funciones.

Figura 2.3 Raspberry Pi Modelo 3 B+ [18] [30]

2.5 Raspberry pi modelo 4

Este modelo es el último que está en el mercado en la actualidad, cuenta con algunas mejorar y actualizaciones. Visualizando la Figura 2.4, los puertos HDMI ahora son microHDMI, la velocidad de transmisión del puerto ethernet ya no está limitado a 300Mbits/s, posee un SOC Broadcom BCM2711 con cuatro núcleos Cortex-A72 y procesador ARMv8 de 64 bits, mantiene los medios de conectividad Bluetooth y WiFi 802.11n utilizando banda de doble transmisión a 2.4 GHz y 5 GHz. Mantiene los cuatro puertos USB y se expande la memora RAM hasta 8 GB.

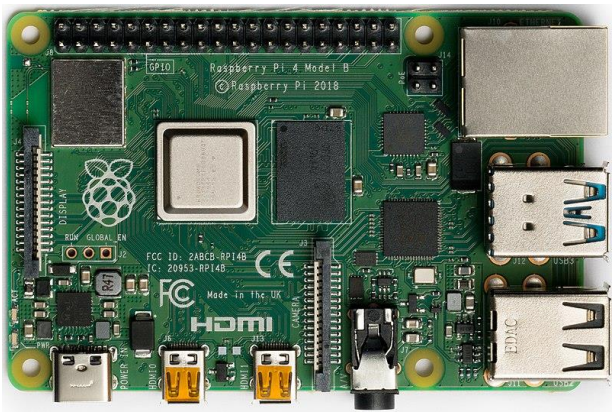


Figura 2.4 Raspberry Pi Modelo 4 B [18] [30]

2.6 Microcontrolador ESP82

Este chip electrónico fue desarrollado por la empresa Espressif, generalmente es utilizado en ambientes industriales ya que efectúa trabajos en lugares de altas temperaturas. En la Figura 2.5 se muestra el microprocesador ESP82, internamente este microprocesador tiene incorporada una antena WiFi, filtros y amplificadores de potencia que permiten la tolerancia al ruido. Por lo general, esta placa es ideal para ser utilizada en proyecto Iot, por ejemplo, en los relojes inteligentes (Wearables), ya que conlleva un consumó mínimo de energía y una prolongada duración de batería.

Figura 2.5 Microprocesador ESP82 [31]

2.7 Microcontrolador ESP8266

La placa ESP8266 [32], fue desarrollada por la empresa Nodemcu, actualmente es uno de los microprocesadores más populares, utilizado principalmente en proyectos que se requiera conexión a distancia de dispositivos por medio de WiFi, por ejemplo, controlar una alarma o puerta de garaje de forma remota; esto es posible gracias a su módulo WiFi incorporado y al alto rendimiento del procesador.

Entre sus principales características destacan el amplificador WiFi incluido en el chip permitiendo tener una mayor cobertura de gestión, el procesador de 32 bits que puede escalar de 80MHz hasta 160 MHz, 80K y 1 Mb de memoria DRAM y flash respectivamente [31]. Además, como se visualiza en la Figura 2.6, tiene 15 pines de entrada y salida generales (GPIO) programables, cuenta con 2

botones conectados al pin GPIO 0 y Reset para activar y/o resetear el firware del chip.

Figura 2.6 Microprocesador ESP8266 [33]

2.8 Microcontrolador ESP32

Esta placa pertenece a la familia de la empresa Espressif Systems. Se diferencia respecto a las demás ya que incorpora tecnología Bluettooth, WiFi como comunicación dual y otras características que la vuelven superior al microprocesador ESP8266. Su procesador de 32 bits puede tener una escalabilidad hasta de 400KHz, lo que permite un mayor rendimiento y respuesta rápida frente a procesos.

En la Figura 2.7 se observa el microprocesador ESP32 con sus pines, a comparación de los demás microprocesadores en mención, este incluye dos núcleos lo cual permite que se administre datos por medio de sensores intercambiando información de forma simultánea a la nube, lo que precisa en ser una placa con un buen desempeño necesaria a utilizar en este proyecto [34]. Adicionalmente, tiene incorporado amplificadores de potencia y de recepción; su compatibilidad de tecnología Bluetooth es de baja energía.

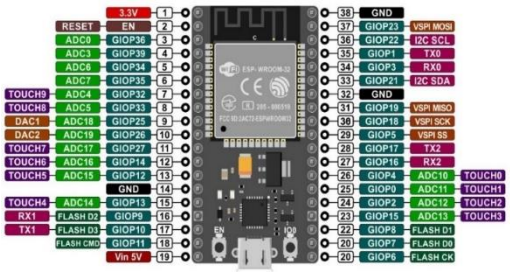


Figura 2.7 Microprocesador ESP32 [35]

2.9 Cámaras analógicas

Es un tipo de cámara empleada en sistemas de circuitos cerrados de televisión (CCTV). Para este tipo de cámaras, se maneja una conexión punto a punto ya que utilizan el cable coaxial como medio de conectividad; esto resulta en un manejo no tan eficiente. Al utilizar estas cámaras en algunos sistemas, el tráfico de video analógico no puede conllevar a riesgos asociados a la red, esto es debido a que tienen una conexión pasiva, por lo que problemas externos a la red no afecta el sistema en que se emplean [36] .

En la Figura 2.8, se muestra el funcionamiento de las cámaras analógicas, en el cual, el lente capta la imagen/video, esta es procesada por amplificadores y chips de procesamiento de imagen, aplica un conversor para visualización de video. Por lo general, para estos dispositivos se debe tener una infraestructura elaborada de cableado de energía y video. Las cámaras analógicas deben ser escogidas teniendo en cuenta estas particularidades:

- **Sensibilidad**

La sensibilidad presentada en estas cámaras se representa en lux; a mayor sensibilidad, implica en que estas trabajen a mayor lux.

- **Resolución**

La cantidad de detalles que se puede observar en la imagen de video. La resolución se mide en pixeles.

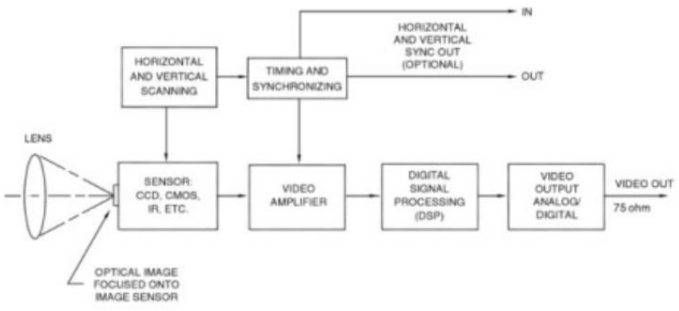


Figura 2.8 Diagrama de bloques de funcionamiento CCTV [37]

2.10 Cámara Web MTQ HD

La cámara Web MTQ HD, es tipo de cámara HD que posee resolución de 1080 pixeles. Se conecta al computador/microprocesador a través del puerto USB. Mediante esta se puede transmitir imágenes en alta definición y visualizar en el monitor las áreas en donde estén instaladas. Son cámaras que no requieren de alimentación directa ya que a través del cable USB obtiene la alimentación necesaria para funcionar como se observa en la Figura 2.9.

Figura 2.9 Cámara Web MTQ HD [38]

Características:

- Alta resolución.
- Lente giratorio de 360°; mayor cobertura.
- Grabación de 1080 pixeles.
- Incorpora un estabilizador óptico.

2.11 Cámaras Bullet

Estas cámaras se caracterizan por tener un diseño de forma rectilíneo hacia fuera, por lo que cubren una zona fija de vigilancia ya que tienen un rango de visión establecido. En la Figura 2.10 se muestra la cámara bullet modelo SCO-6085R, esta cámara está equipada con un sensor de imagen de alto rendimiento que permite tener una eficiente calidad de imagen. Además, en su lente esta incorporado la iluminación de luz baja con la cual se pueden obtener imágenes y videos de buena resolución en lugares donde hay poca luz [39]. En adición, tiene

varios leds infrarrojos que se activan para dar mayor luminosidad en un área de hasta 25 metros.

Características:

- Alta resolución Full HD 1920 x 1080 pixeles.
- Ángulo de visión vertical de 60° y horizontal de 120°.
- Lente de longitud focal (permite acercamiento)
- Distancia de alcance de hasta 100 metros.

Figura 2.10 Cámara bullet SCO-6085R luz [39]

2.12 Cámaras Domo

Son cámaras que están disponibles comúnmente en el mercado de la electrónica y se emplean mayoritariamente en ambientes de seguridad en áreas internas, ya que tiene un diseño en forma de cúpula/esférica de vidrio con su lente en el interior. En la Figura 2.11 se muestra una cámara domo de marca Hilook Thc-t120, esta se caracteriza por tener un ángulo de vigilancia de 130°, por lo que puede tener una cobertura de visión general dentro de un área extensa [40]. Esta cámara cuenta con la tecnología de alta definición analógica (AHD) lo que garantiza una resolución de alta definición (HD) de 1080 pixeles en captación de imágenes y transmisión de video. Además, incorpora el IP67 que es un nivel de protección en su estructura contra el polvo y lluvia.

Características:

- Resolución de 1920 x 1080 pixeles.
- Ángulo de visión vertical de 80° y horizontal de 100°.

- Ángulo de inclinación de 0 a 70 °.
- Distancia de alcance de hasta 100 metros.
- Alcance de 70 m (visión diurna) y de 25 m (visión nocturna).

Figura 2.11 Cámara Domo Hilook Thc-t120 [40]

2.13 Cámaras Digitales

Son conocidas como cámaras de red o cámaras IP ya que su transmisión de audio y video se realiza mediante cable directamente a la red. Generalmente poseen un puerto de entrada Ethernet para conexión vía internet. Son cámaras especializadas para enviar señales que atribuyen a imágenes, audios y videos por medio de internet utilizando variados protocolos de red. Estás cámaras representan una alta eficiencia en algunos sistemas ya que estas pueden integrar aplicativos de detección de movimiento; ante esto ejecuta grabación de imágenes del suceso [41]. Además, gran parte de las cámaras IP tienen puertos de entrada y salida como se puede observar en la Figura 2.12 en donde se puede conectar dispositivos electrónicos como sensores, indicadores, cable para alimentación mediante Ethernet (PoE), entre otros.

Figura 2.12 Componentes de una cámara IP [42]

Internamente en estas cámaras vienen integrado un sensor de imágenes, lente, microprocesador de imágenes, placa de video, módulo Ethernet para la conectividad inalámbrica y comunicación de datos. Con la variabilidad de las redes IP, se puede lograr la visualización, control y múltiples gestiones de las cámaras IP en tiempo real desde cualquier lugar en donde haya conexión de red [43]. En la Figura 2.13 se puede observar detalladamente el diagrama que contiene bloques del funcionamiento de cámaras digitales; el lente capta la imagen mediante un sensor de imagen, esto es guardado en el almacenamiento y posteriormente visualizado en un monitor. Por tal razón, la utilización de estas cámaras en algunos sistemas resulta ser más competente respecto a las cámaras analógicas. Entre las principales ventajas se tienen las siguientes:

- Cámaras con alta resolución.
- Excelente calidad de captación de imágenes.
- Control y manejo de forma inalámbrica.
- Escalabilidad de funciones.

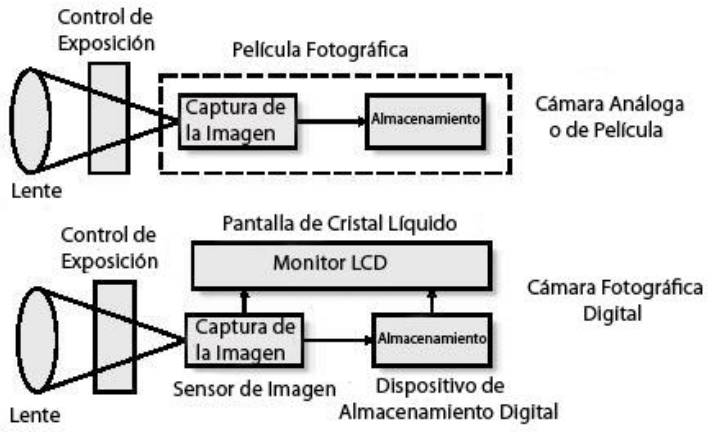


Figura 2.13 Diagrama de bloques de funcionamiento de cámaras digitales [37]

2.14 Cámara IP Pan Tilt Zoom (PTZ)

Estas cámaras tienen dos funcionalidades específicas, la primera hace referencia al “Pan Tilt Zoom” el cual es un enfoque de acercamiento o alejamiento en una rotación vertical u horizontal, y la segunda modalidad trata

del seguimiento automático ya que incorpora sensores de movimiento, temperatura y humedad [44]. En la Figura 2.14 se muestra la cámara IP PTZ de marca Hikvision, esta cámara puede ser controlada de forma inalámbrica sin necesidad que el usuario esté en el lugar en donde se haya instalado, además cuenta con un alto enfoque de acercamiento que permite el monitoreo hasta una distancia de 1540 pies e incorpora tecnología de alto rendimiento que entrega una alta calidad de imagen y video en cualquier clima que se presente.

Características:

- Alta resolución Full HD 1920 x 1080 pixeles.
- Posee zoom óptico de 60 x
- Expansión de microSD hasta 128GB
- Alcance de 150 m (visión diurna) y de 40 m (visión nocturna).
- No requiere de mucho mantenimiento ni atención.

Figura 2.14 Cámara IP Hikvision PTZ [44]

2.14.1 Cámara IP TV- IP440PI Trendnet

Estas cámaras pueden llegar a cubrir un área extensa de vigilancia ya que tiene una mayor disponibilidad de funciones respecto a la rotación horizontal, vertical y de acercamiento. En la Figura 2.15 se observa la cámara IP TV-IP44PI de la marca Trendnet; su operación es automática y continua ya que contiene la funcionalidad de posición predefinida, por lo que la cámara puede moverse a los puntos de vigilancia fijados [45].

Características:

- Alta resolución HD 2M y 1080 pixeles.

- Ángulo de visión vertical de 90° y horizontal de 360°.
- Alcance de 200 m (visión diurna) y de 100 m (visión nocturna).
- Tiene tecnología PoE.

Figura 2.15 Cámara IP TV-IP440PI Trendnet [45]

2.14.2 Cámara Raspberry Pi V1.3

La cámara Pi como se observa en la Figura 2.16, es un módulo que se introduce directamente en la interfaz CSI de la Raspberry. Fabricada por la fundación Raspberry Pi en Reino Unido. Es una cámara de alta definición que ofrece nitidez en las imágenes y claridad en los videos. Son cámaras de circuito cerrado con funcionalidades específicas.

Características:

- Compatible con modelo A y B de Raspberry Pi.
- Módulo de cámara Univisión 5647 de 5 MP.
- Imágenes de calidad 2592x1994.
- Admite grabaciones de 1080p, 720p y 480p.
- Interfaz CSI de 15 pines.
- Tiene lente de enfoque fijo Onboard.

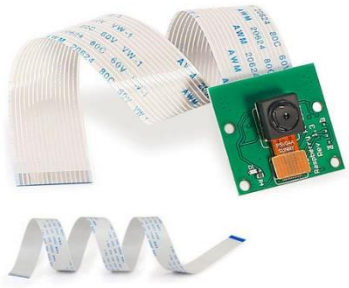


Figura 2.16 Cámara Pi Rev 1.3 [46]

2.15 Pantallas Táctiles y teclados inalámbricos

Las pantallas táctiles en algunos lugares son llamadas pantallas touch, es conocida por su importante funcionamiento que permite mediante un toque en la superficie admitir entrada de datos o funciones de movimientos, a continuación, se mencionarán algunos tipos de pantallas.

2.15.1 Pantalla Touch Screen TFT

Es un tipo de pantalla táctil de 3,5 pulgadas, es sumamente compatible con Raspberry modelos Pi 3B, B+ y 2B y su sistema operativo Raspbian. Posee una resolución amplia y de alta definición con 320 x 480 pixeles; en la Figura 2.17 se visualiza una pantalla touch screen de 3.5 pulgadas.

Características:

- Interfaz de la pantalla tipo SPI.
- Pantalla tipo TFT.
- Color ilustrativo 65536.
- Luz LED de fondo.Tamaño: 85,42 mm x 55,60 mm.

Figura 2.17 Pantalla Touch Screen 3,5 Pulgadas [47]

2.15.2 Pantalla TFT MegaSystem

Es una pantalla táctil compatible con los microcontroladores ESP y Arduino, no es tan amigable con Raspberry Pi pero se la puede configurar para que cumpla con configuraciones básicas realizadas en Raspberry. Como se muestra en la Figura 2.18 es de dimensiones reducidas y apta para acoplar en sistemas inteligentes

Características.

- Resolución: 320x240.
- Driver: ILI9341.
- BUS: SPI.
- Touch: Si.
- Tamaño: 2.8”.
- Dimensiones pcb: 77x43mm.

Figura 2.18 Pantalla Táctil Spi 2,8 pulgadas [48]

2.15.3 Teclado Mini Keyboard

Características

- Diseño ergonómico, fácil de transportar y de utilizar.
- Batería de iones de Litio recargable.
- Rango de operación: 15 metros (máximo), sin obstaculizar la señal.
- Se requiere un sistema operativo de Windows, Linux, Android/for Google/for Smart TV, Mac OS
- Conexión: 2.4G wireless: 10 m
- Conexión USB

Figura 2.19 Teclado táctil inalámbrico [49]

2.16 Sistema operativo

Un sistema Operativo contiene múltiples programas los cuales tienen como funcionalidad la distribución y manejo de memoria, medios de almacenamiento y los distintos recursos del equipo computacional, por ejemplo: mouse, teclado, parlantes, tarjeta de red, entre otros. En la actualidad, existen distintos sistemas

operativos en los cuales se puede programar y son: Windows, Linux, MAC, entro otros [50]. A continuación, se detallará alguno de ellos.

2.16.1 Linux Ubuntu

Es un sistema basado en Debian en las cuales sus principales características que lo hacen viable para la elaboración de programas son: Facilidad en el manejo de programas y características, Actualizaciones de frame y del sistema frecuentemente, Fácil de instalar para poder desarrollar y libertad de manejarlo de forma correcta y distribución continua [51].

Las características más destacadas de Ubuntu son:

- **Compatibilidad**. Es compatible con gran cantidad de ordenadores. Además, los servidores que requieran comunicación con Ubuntu Server tienen la facilidad de integrarse de manera confiable en la intranet central.
- **Seguridad.** Administra de forma confiable y segura los sistemas de seguridad solicitando clave de acceso cada que se quiera instalar algún programa. Es más resistente a virus que otros sistemas operativos.
- **Facilidad de gestión.** Ubuntu brinda la tecnología LTSP (Linux terminal server project), permite la conexión mediante programas especializados al entorno virtual remotamente, facilitando la comunicación entre programador y máquina.
- **Economía.** Es un programa de código abierto lo cual es beneficioso para poder instalar y manipular.
- **Soporte.** Al momento de generar algún error, existe un sin número de fuentes en el internet que facilitan el desarrollo del código y evitan la obtención de errores.
- **Educativo.** Posee la facilidad de comunicación aplicando múltiples idiomas y animaciones con métodos característicos para personas o niños con necesidades especiales.

- **Aplicaciones.** Tiene integrado una amplia variedad de programas los cuales son de gran beneficio para la utilizarlos a nivel educativo.

2.16.2 Sistema Operativo Windows

Windows es conocido como uno de los softwares más importantes de la industria, creado por la famosa empresa Microsoft. Windows fue creado a finales del año 1985 con la primera versión 1.0 ofreciendo baja funcionalidad, al paso de los años se fueron mejorando las versiones e innovando y adaptando funcionalidades al sistema de tal forma que en la actualidad se trabaja con Windows 10 y es una de las mejores versiones existentes [52]. Algunas de las características de este sistema es que facilita la administración de los recursos en un ordenador, controla el acceso a datos, organiza archivos, proporciona servicios de ejecución, entre otras funciones que se detallarán.

Características Generales:

- Interfaz de usuario gráfica.
- Multitarea (ejecuta al mismo tiempo varias tareas).
- Integración de recursos multimedia (textos, imagen y sonido).
- implementación de programas importantes para múltiples usos: Un Bloc de notas, un escritor de textos (WordPad), etc.

2.16.3 Sistema Operativo Mac

Fundado por la prestigiosa empresa Apple, es uno de los sistemas más conocido por el medio debido a su importante integración de interfaz gráfica. Su nombre oficial es Mac Os que significa “Macintosh Operating System” su principal ventaja es que procesa de forma eficiente y más rápido la información en comparación con otros sistemas, pero no es tan eficiente para desarrollo de software [53].

2.17 Lenguaje de programación

Es un lenguaje que se utiliza formalmente en la cual el programador puede tipear de un conjunto de órdenes, datos, acciones y lazos consecutivos y algoritmos que se realizan mediante una serie de instrucciones para poder construir programas que tengan como función el control físico y lógico de una maquina o equipo [54]. Es la forma usual en la que una persona llamada programador puede comunicarse con una maquina dándole la facilidad de precisar aspectos como:

- Datos que debe trabajar un software.
- Almacenamiento y transmisión de datos.
- Acciones dependientes que debe tomar el programa mediante las circunstancias que se presenten.

2.17.1 Python

Es un lenguaje de programación el cual se familiariza fácilmente con servidores, Aplicaciones iOS, Linux, Android, etc. Este suceso se debe a que posee una sintaxis de código fácil de aprender, es versátil multiparadigma y multiplataforma el cual sobresale por su sintaxis de cogido limpio y legible. Cuenta con licencia de código abierto y permite la automatización de procesos y elaboración de trabajos en entornos de clientes como así mismo de servidor.

Python fue inventado a inicio de los 90’s por un ingeniero holandés llamado Guido Van Rossum [55],Actualmente labora en CWI de Ámsterdam, el cual se conoce como el Centro de Investigaciones de Ciencias de la Computación. Actualmente es usado a nivel mundial, tiene un logotipo definido como se puede observar en la figura 7. cuenta con características fundamentales que se detallarán a continuación:

- **Simplificado y rápido**: Permite la simplificación en la programación, es uno de los lenguajes comúnmente utilizado para realizar scripting.

- **Distinguido y flexible**: Es práctico y brinda facilidades al programador para lograr programar de forma entendible e interpretable. Posee una sintaxis elegante y permite desarrollar de forma procesal u orientada a objetos.
- **Eficiencia programable**: Es un lenguaje fácil de aprender, por medio de métodos de aprendizajes moderados a partir de curvas gráficas. Cuenta con interpretes los cuales permiten el desarrollo del código de forma interactiva.
- **Ordenado y escalable**: Se mantiene estandarizado y organiza de forma eficaz sus módulos. Posee una sofisticada variedad de librerías que permiten el desarrollo de programas avanzados, procesamiento de imágenes, servidor web, etc.
- **Portable**: Es un lenguaje muy accesible. Se lo puede usar prácticamente en cualquier sistema, ya sea de seguridad, finanzas, inventarios, etc.

2.17.2 OpenCV – Python

OpenCV es una biblioteca especializada para utilizar en Python, tiene como objetivo la detección de rostro mediante algoritmos de visión por computadora. Su creación fue en la empresa Intel en 1999 por Gary Bradsky, científico estadounidense, ingeniero especializado en sistemas.

Es conocida por su admisión amplia de lenguajes de programación tales como C, Java, C++, Python, etc. Se lo puede utilizar en varias plataformas de sistemas operativos, por ejemplo: Windows, Linux, Android y iOS. También admite interfases que se basan en OpenCL y CUD, las cuales se desarrollan para mecanismos de GPU (Graphic Processing Unit) de velocidad alta [56].

Una de las ventajas que tiene OpenCV en Python es que la librería es compatible con otras librerías integradas en Python, por ejemplo: Numpy la cual realiza funciones numéricas y las convierte en arreglos. Esto quiere decir que los mecanismo y operaciones que realice OpenCv se convierten en arreglos a

través de la librería Numpy y representaran una mejor visualización al momento de correr el código. Otras librerías que son adaptables a OpenCV son Matplotlib y Scipy las cuales permiten graficar y mostrar resultados definidos.

OpenCv trabaja con códigos de colores BGR que significa Blue, Green and Red. Otras librerías como PIL y Matplotlib utilizan RGB, este código es muy utilizado en la actualidad para la transformación de imágenes [57].

2.17.3 Lenguaje de programación C++

Lenguaje conocido por su gran importancia en la potenciación de la programación orientada a objetos [58]. Creado por el danés Bjarne Stroustrup en el año de 1979 mediante la captación de distintos lenguajes de programación con sus diferentes características para mejorar la versión del lenguaje C. entre sus principales características se puede mencionar que cuenta con un estándar ISO, programación pura mediante tipeo de códigos, soporta sobrecargos y expresiones lamba conocidas como expresiones anónimas y eficiencia en trabajos de hardware. Permite la compilación mediante IDE de programación las cuales son Visual Studio y Code-Blocks.

2.18 Características de un sistema de seguridad

Dentro de un sistema de seguridad se deben precisar características que son muy importantes para el desarrollo y funcionamiento integral del servicio [59].

- **Disponibilidad:** hace referencia al tiempo en que el sistema está disponible para ser usado, por lo cual, se plantea el tiempo de funcionamiento del sistema de 24/7, es decir, 24 horas, los 7 días a la semana.

- **Confiabilidad:** se recalca la capacidad de que el sistema cumpla con el funcionamiento respectivo del cual fue planificado.

- **Control de acceso:** consiste en la verificación y comprobación de acceso a los servidores y sistemas de almacenamiento; teniendo una restricción para usuarios no autorizados.

2.18.1 Sistemas autónomos de control de acceso

Son sistemas de fácil manipulación, no requieren almacenamiento de información, funcionan como una llave electrónica la cual permitirá el ingreso o acceso al personal autorizado [17]. en la figura 2.20 se observa los ejemplos tales como: teclado numérico, botón de acceso, módulo magnético.

Figura 2.20 Control de acceso autónomo [17]

2.18.2 Sistemas de control de acceso de red

Este sistema desarrollado es inteligente y más eficiente, posee características que permiten obtener un registro y acceder al mismo de las personas que ingresan y salen. Se puede verificar y analizar todo tipo de dato como, por ejemplo: Nombre, Fecha, Hora, entro otros [17]. En la figura 2.21 se detalla los diferentes sistemas que abarcan el control como, por ejemplo: Tarjeta RFID, Biometría, Torniquete, reconocimiento facial, entro otros.

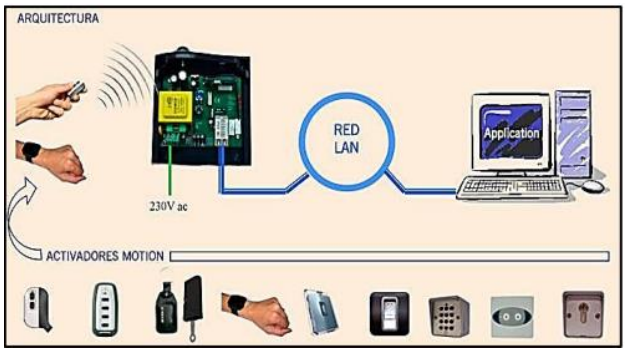


Figura 2.21 Control de acceso de red [17]

Uno de los sistemas más importantes y eficientes es el del reconocimiento facial, a continuación, se detalla su funcionabilidad.

2.19 Reconocimiento facial

El reconocimiento facial es conocido como una de las nuevas tecnologías la cual tiene como finalidad la detección y verificación de una persona utilizando como patrón un rostro, imagen o video. Además, utiliza seguridad biométrica. Se utiliza este tipo de sistemas para salvaguardar y proteger los sectores privados y restricción de acceso de personas no autorizadas [60].

En la actualidad, existe gran cantidad de personas que están al tanto de los procesos y mecanismos que abarca el reconocimiento facial, un claro ejemplo de en donde se lo utiliza es en el FaceID utilizado para desbloquear IPhone [61]. Todo sistema de reconocimiento facial sigue un modelo de funcionamiento, y es de la siguiente manera:

- La cámara logra la detección del rostro de una persona, logra marcar mediante un cuadro, donde se encuentra el rostro.
- Se analiza y estudia la captura del rostro que se realizó para de esa forma recopilar las características del rostro y almacenarlo en una base de datos.
- Se transforma la imagen analógica que vendría a ser información analógica del rostro en un conjunto replicado de datos digitales, Básicamente es una fórmula utilizada para los cálculos de reconocimiento.
- Se compara el rostro al momento de realizar la detección con imágenes almacenadas en un directorio que se tomaron para el respectivo proceso de detección y validación facial.

2.20 Machine Learning

Es conocido como un proceso autónomo que mediante algoritmos permite registrar e identificar patrones de datos y construir predicciones. Es una disciplina la cual se encuentra en el campo investigativo de la inteligencia artificial. Su principal función es permitir a los ordenadores o computadores realizar y proyectar tareas especializadas con un fin especifico de forma autónoma.

En el año 2004, Jeff Hawking Ingeniero Informático define en su libro a machine learning como: “La capacidad de predecir el futuro” [62]. A continuación, se detallará los algoritmos de Machine Learning:

- **Aprendizaje supervisado:** cuentan con un mecanismo previo de aprendizaje el cual se basa en un sistema de etiquetas que se encuentra asociado a diferentes datos que faciliten la toma de decisiones.
- **Aprendizaje no supervisarlo:** tiene como función la detección de patrones que permita organizarlos de alguna manera
- **Aprendizaje por Refuerzo:** se caracteriza por la toma decisiones en diferentes situaciones, permite el desarrollo en varios lenguajes de programación.

  Existen algoritmos de reconocimiento basados en esquemas de aprendizajes en métodos estadísticos en los cuales a continuación se detallan los tipos de modelamiento.

2.20.1 Algoritmo Eigen Faces

Se conoce a Eigen Faces como un algoritmo especializado en el enfoque a la detección y manipulación de un rostro en el cual busca reconocer mediante la captura de variación las principales características de la imagen facial para utilizar esta información en el proceso de codificación y comparar la similitud de las imágenes de rostros de forma holística [63]. Además, es una herramienta

matemática la cual se la conoce como PCA que significa Análisis de componentes principales.

2.20.2 Algoritmo FisherFace

Es un algoritmo basado en análisis discriminante lineal (LDA), el cual emplea procesos estadísticos para determinar un porcentaje de similitud al comparar características propias de objetos. El algoritmo encuentra una combinación lineal que puede utilizarse en el área de reconocimiento facial; en este campo el algoritmo ejecuta patrones binarios para la extracción de características físicas faciales, y mediante la combinación lineal, genera histogramas estadísticos vectoriales codificados, el cual establece un cálculo de similitud de imagen teniendo como base los histogramas.

2.21 Tkinter

Es una importante librería que tiene como función la creación de GUI (Interfaz Gráfica de Usuario) también conocido como Frame en la cual permite la visualización, el control de widgets y la manipulación mediante una serie de atributos. Tkinter forma parte de la biblioteca Tcl/Tk como unbinding, permite la vinculación entre ambas partes para poder establecer conexión con el lenguaje de programación Python [64].

2.22 Almacenamiento de datos

Se conoce a una Base de datos como un mecanismo de recolección de información de datos estadísticos o personales que se encuentran estructurados de tal forma que tengan accesibilidad, se puedan administrar y actualizar. Existen diferentes tipos de almacenamientos ya sean digitales o virtuales como por ejemplo Base de Datos MySQL, almacenamiento en DropBox de forma virtual, almacenamiento en GoogleDrive, entre otros [65].

2.23 Sensores electrónicos de movimiento

Los sensores son dispositivos electrónicos que detectan algún movimiento presente dentro del lugar o área en donde se encuentren instalados. Consumen poca electricidad lo que conlleva a optimizar el consumo de energía eléctrica [66]. Respecto en sistemas de seguridad implementados, hay dos tipos de sensores que son comúnmente utilizados debidos a sus amplias y eficientes características:

2.23.1 Sensor Ultrasónico HC-SR04

Este sensor es usado mayormente en detección de obstáculos y el más utilizado dentro de la familia de tipo ultrasónico. Se caracteriza por calcular la distancia entre el sensor y un objeto a través de ultrasonidos. En la Figura 2.22 se muestra el sensor ultrasónico HC-SR04, tiene incorporado 4 pines: el pin Vcc que es la alimentación de 5 DC (Corriente Directa), el pin Trigger, el pin Echo, y el pin GND que es tierra [59]. Su funcionamiento consiste en que este sensor emite pulsos ultrasónicos por medio del pin Trigger, las ondas viajan y rebotan contra el objeto, y se recepta dichas señales por medio del pin echo; determinando la distancia del objeto [67].

Características Electrónicas/Técnicas

- Voltaje de alimentación de 5V DC.
- Intensidad de corriente menor a 2mA.
- Angulo de detección hasta 18°.
- Detección de objetos de 2cm hasta 500 cm.

Figura 2.22 Sensor ultrasónico HC-SR04 [67]

2.23.2 Sensor PIR HC-SR501

Es un sensor infrarrojo de bajo costo, que actualmente contiene la tecnología más reciente como detector de movimiento [60]. En la Figura 2.23 se muestra el sensor PIR HC.SR501; como se aprecia, este incorpora un par de potenciómetros los cuales se pueden ajustar para variar la sensibilidad y respuesta de detecciones [68].

Características Electrónicas/Técnicas

- Voltaje de operación de 4.5V a 12V DC.
- Angulo de detección hasta de 110°.
- Rango ajustable de rastreo de 3 a 8 metros.
- Tiempo de respuesta de 5s a 20 s.

Ventajas

- Fácil instalación y uso.
- Consumo mínimo de potencia y energía eléctrica.
- Tolerante a interferencias y ruidos.
- Admite alto voltaje de alimentación.
- Mayor ángulo de detección.

Figura 2.23 Sensor Infrarrojo PIR [68]

2.23.3 Sirena de alarma HC-606-1200S

Esta potente sirena alarma de dos tonos cuenta con 112 db y una potencia de 15 watts para ser el mejor complemento de alarma, eficiente para espacios residenciales y comerciales [69].

Características

- Cubierta de ABS
- Color: Negro
- Voltaje: 12V DC
- Rango Voltaje: 6 ~ 15V DC
- Corriente: 1200 mA
- Sonido (dB/1m): 112

Figura 2.24 Sirena de alarma para evitar robos. [69]

2.24 Plataformas de sistemas de seguridad

Estas plataformas se basan en un sistema operativo de programación, las cuales permiten gestionar la videovigilancia mediante interfaces de control. Seguidamente se detallan las principales plataformas.

2.24.1 MotionEyeOS

Es una plataforma que consta de una interfaz web de libre uso [70] en la cual se puede realizar el monitoreo de cámaras de seguridad en tiempo real. Su programación debe ser desarrollada en una Raspberry Pi conectada a las cámaras que conforman el sistema. En la Figura 2.25 se muestra la interfaz de monitoreo MotionEye: esta posibilita el acceso al sistema de videovigilancia mediante una dirección IP generada al terminar la programación pertinente en la tarjeta microcontroladora.

En este aplicativo se permite la visualización y grabación simultánea de video, además de incluir modos de grabación, ya sea programada o continua.

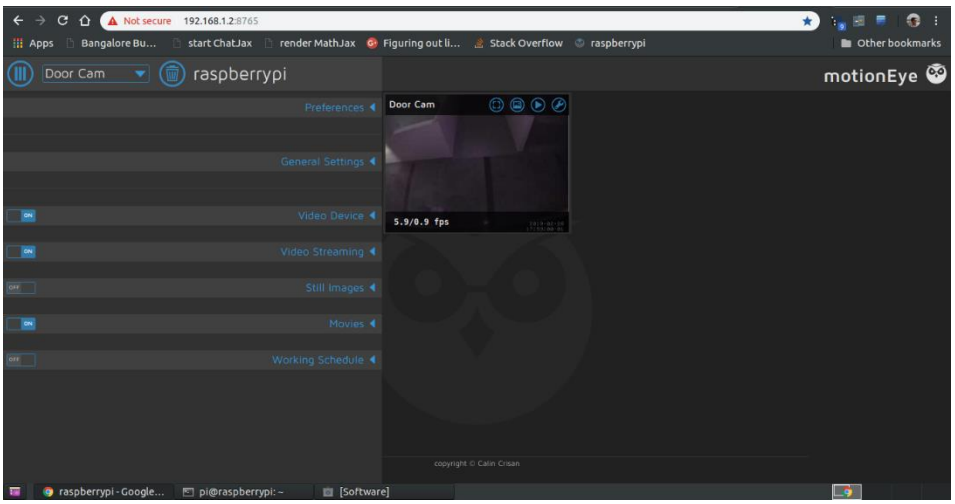


Figura 2.25 Interfaz de MotionEyeOS [70]

2.24.2 Sistema Digital Video Recorder (DVR)

Esta plataforma permite visualizar mediante pantalla las cámaras de seguridad. Además de poder configurar diversos parámetros como la programación de grabación, frames por segundo de cada cámara, mover cámaras, entre otros. No es un sistema tan utilizado ya que para la conexión de las cámaras se utiliza entradas analógicas conectadas con cable coaxial [71].

En la Figura 2.26 se visualiza el equipo DVR; este cumple con la función de digitalizar tanto las imágenes, grabaciones y audios que se reciben por medio de las cámaras. Estos dispositivos se pueden conectar a la red, de tal manera que

permite acceder remotamente a la plataforma del sistema desde cualquier dispositivo.

Características del equipo DVR

- Resolución de hasta 1080 pixeles.
- No requiere mantenimiento.
- Grabación continua.
- Utiliza el formato de compresión de video H.264.

Características de la plataforma DVR

- Plataforma de fácil entendimiento y uso.
- Permite programación o cronograma de grabación por eventos.

Figura 2.26 Dispositivo DVR Dahua [72]

2.24.3 Sistema Networking Video Recorder (NVR)

Es una plataforma que administra las imágenes y videos de cámaras de vigilancia a través de una red IP, es decir, utiliza tecnología Ethernet en un cableado de red. Esta posee un alto rendimiento de gestión dentro de un sistema pequeño de seguridad en donde hay un límite escalable de cámaras [73]. En la Figura 2.27 se observa el equipo video grabador NVR; este dispositivo tiene un costo elevado a comparación del equipo 2.26; este recibe las imágenes procesadas de forma digital proporcionando una mayor calidad de imagen con mayor resolución ya que en este se presenta menos ruido [74].

Se puede acceder al sistema, programar cámaras y diversas configuraciones a través de una conexión remota Telnet, SSH o explorador de internet (browser)**.**

Características del equipo NVR

- Resolución de 5M (2560 x 1920) pixeles.
- Grabación automática ante fallas eléctricas.
- Asistencia de formato de imágenes en movimiento (MPEG) 4.
- Monitoreo y grabación de hasta 16 cámaras IP.

Características de la plataforma NVR

- Grabación de cámaras 24/7.
- Fácil configuración y manejo.
- Soporte de asistencia Android y Iphone.

Figura 2.27 Dispositivo NVR 4116 [74]

CAPÍTULO 3

3. METODOLOGÍA

En este capítulo se detallan aspectos importantes sobre el sistema en general, ya que en capítulos anteriores se ha mencionado el problema del cual se requiere una solución, objetivos a cumplir y estudios realizados de enfoques similares al sistema y sus diferentes funcionalidades. El sistema cuenta con dos etapas: sistemas de acceso del personal y sistema de videovigilancia, de las cuales, se detallará algoritmos de detección facial, almacenamiento de imágenes y videos, notificaciones de alerta mediante protocolo de comunicación, proceso de visualización y grabación en tiempo real.

3.1 Diagrama general de bloques

Ese puede observar en la Figura 3.1 el diagrama de bloques general del presente proyecto, el cual corresponde al diseño y simulación de sistema de videovigilancia para el monitoreo de cultivos y control de acceso del personal. El desarrollo inicia con el diseño del sistema - bloque 1, en donde se realiza investigación de los parámetros y equipos electrónicos a utilizar considerando el análisis del entorno y requerimientos del cliente.

Figura 3.1 Solución general del sistema en diagrama de bloques

Posterior a la identificación de los equipos necesarios, se procede con el desarrollo de las estructuras de algoritmos - bloque 2, para programar los equipos con funciones específicas que cumplan con funcionalidad del sistema.

Luego se realiza las pruebas de operatividad - bloque 3, en donde se lleva a cabo la comprobación del funcionamiento de los componentes y evaluación de adquisición de datos en las tarjetas microcontroladoras. En el caso de ser necesario corregir los posibles errores que se puedan presentar.

Por último, se realiza la implementación del sistema en un ambiente simulado - bloque 4, de tal forma que permita verificar si cumple con la operatividad de seguridad dentro de la industria. Se pretende que con este sistema se brinde la seguridad pertinente mediante el uso de nuevas tecnologías.

3.2 Diagrama de flujo detallado de la solución general

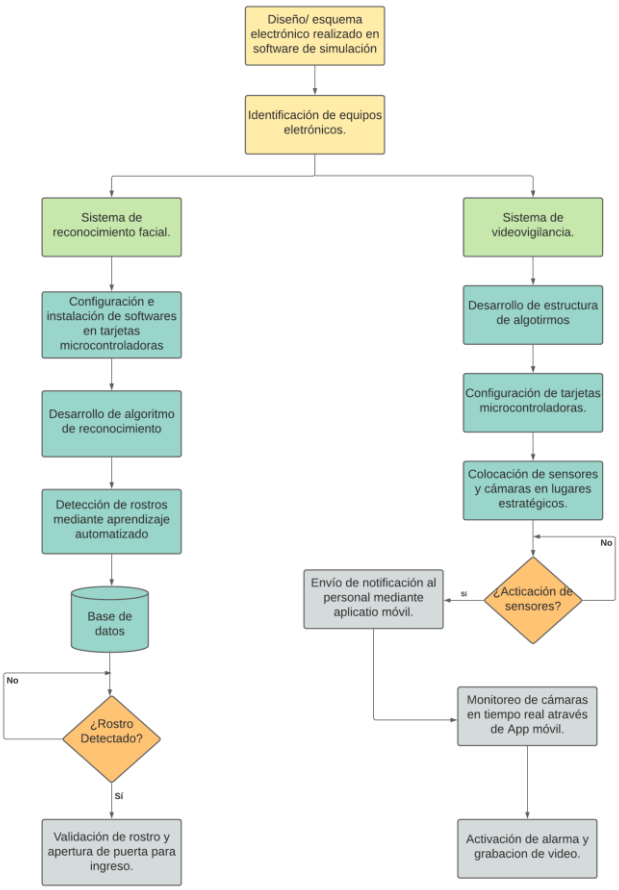


Figura 3.2 Diagrama de flujo detallado de la solución general

El proceso detallado para el desarrollo del sistema se lo explica por medio del diagrama de flujo mostrado en la Figura 3.2. Primero se realiza un diseño que corresponde a un esquema electrónico elaborado en un software de simulación para precisar el bosquejo del sistema general. Se identifican los equipos necesarios para cumplir con las funciones requeridas en el sistema. Como propuesta de solución general, se ha dividido en 2 etapas:

La primera fase hace referencia a un sistema de reconocimiento facial, el cual se implementa para tener un control de acceso del personal a la industria. En esta se realizan las instalaciones de los softwares requeridos en la tarjeta microcontroladora a ser utilizada, y se procede con el desarrollo de algoritmo que trata sobre el tipo de entrenamiento que permitirá la detección de rostro a través de un aprendizaje automatizado. Se crea un directorio donde se almacenan los rasgos faciales de las personas registradas, y posteriormente mediante el mismo algoritmo, se realiza el reconocimiento facial tomando características físicas del rostro de la persona comparando con los datos registrados para permitir su acceso.

La segunda fase trata del sistema de videovigilancia, el cual se encargará del monitoreo visual de los cultivos de cacao para evitar el robo del fruto y que se permita una asistencia inmediata del personal encargado. En este sistema se desarrollan los algoritmos para cumplir con funciones específicas, estos son configurados en las tarjetas microcontroladoras a utilizarse. Se procede con la instalación y configuración de los dispositivos como sensores y cámaras en el microcontrolador.

El sistema se enlaza a un almacenamiento web mediante una plataforma de monitoreo para permitir guardar las grabaciones de las cámaras y se pueda revisar los videos en cualquier momento. Cuando los sensores de activen, se enviará notificaciones de alerta al personal encargado mediante comunicación inalámbrica WiFi utilizando una aplicación de mensajería, permitiendo mediante una App móvil el monitoreo de cámaras en tiempo real y el control de activación de alarma para evitar posibles robos.

3.3 Diagramas de bloques específicos de la solución por etapas

Se detalla en diagrama de bloques en la Figura 3.3 del sistema de reconocimiento facial, el cual contiene un microcontrolador denominado Raspberry Pi alimentado por una fuente de poder DC, una Cámara USB y una pantalla LCD tipo SPI donde se observará la programación de la interfaz visual mostrando diferentes opciones seleccionables para realizar el proceso de registro de usuario almacenando en una base de datos la información de cada uno y para el proceso de ingreso mediante algoritmo de reconocimiento facial para validar y comprobar la similitud y dar acceso final a dicho usuario.

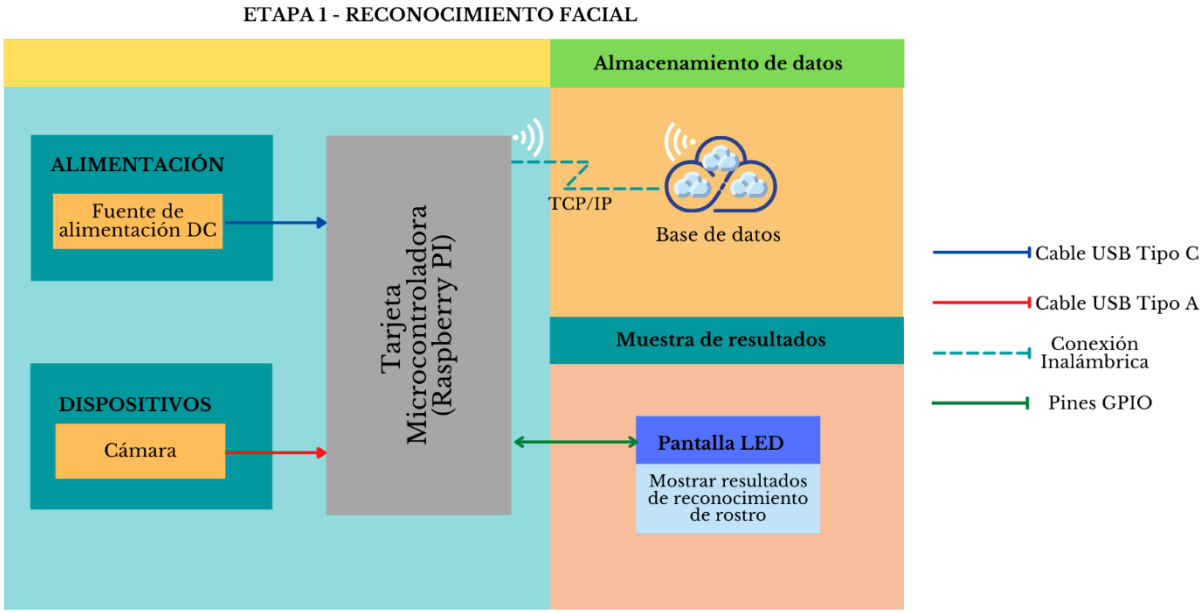


Figura 3.3 Diagrama de bloque, Reconocimiento Facial

Se observa el diagrama de bloques en la Figura 3.4 del sistema respecto a videovigilancia, el cual contiene un microcontrolador ESP32 adaptado con dos sensores de movimientos que al detectar algún suceso no permitido enviarán dicha información al microcontrolador para posteriormente notificar al usuario mediante comunicación TCP/IP a través de mensajería. Además, el usuario podrá comprobar el suceso utilizando un programa de monitoreo instalado en

otro microcontrolador denominado Raspberry Pi que tendrá conectado dos cámaras de videovigilancia para al momento de visualizar el suceso notificado se pueda encender la alarma inalámbricamente que estará conectada a un Relay que funcionará como activador por medio del ESP32.

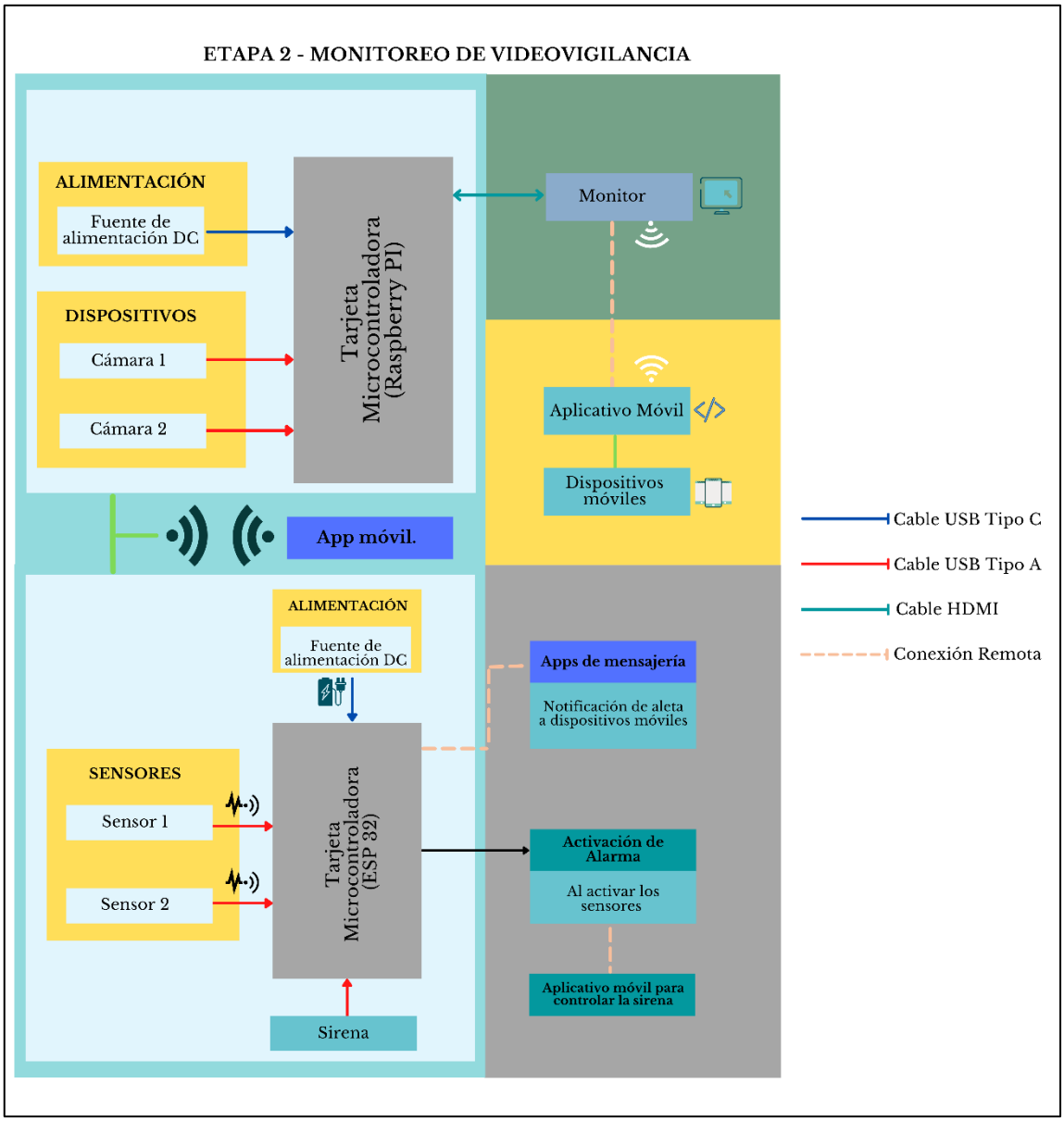


Figura 3.4 Diagrama de bloque Monitoreo del sistema de Videovigilancia

3.4 Diseño físico de la solución propuestas

Se puede observar los equipos a utilizar en la Figura 3.5 para el diseño físico de la solución propuesta de la etapa 1:

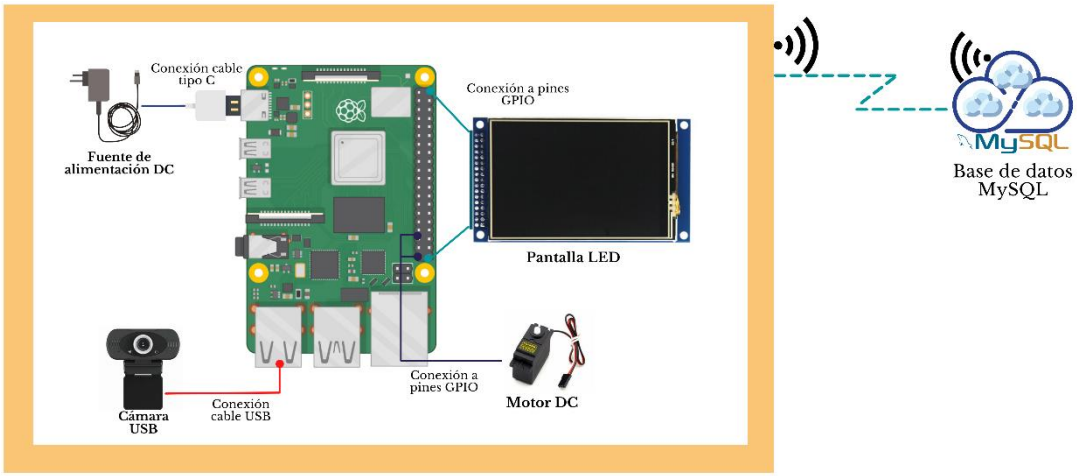


Figura 3.5: Diagrama de conexiones físicas de la etapa 1

Se utilizará un microcontrolador Raspberry Pi, una cámara USB adaptada en el microcontrolador, una pantalla LCD tipo SPI para el visualizar el proceso de detección y un servomotor para la apertura de la puerta.

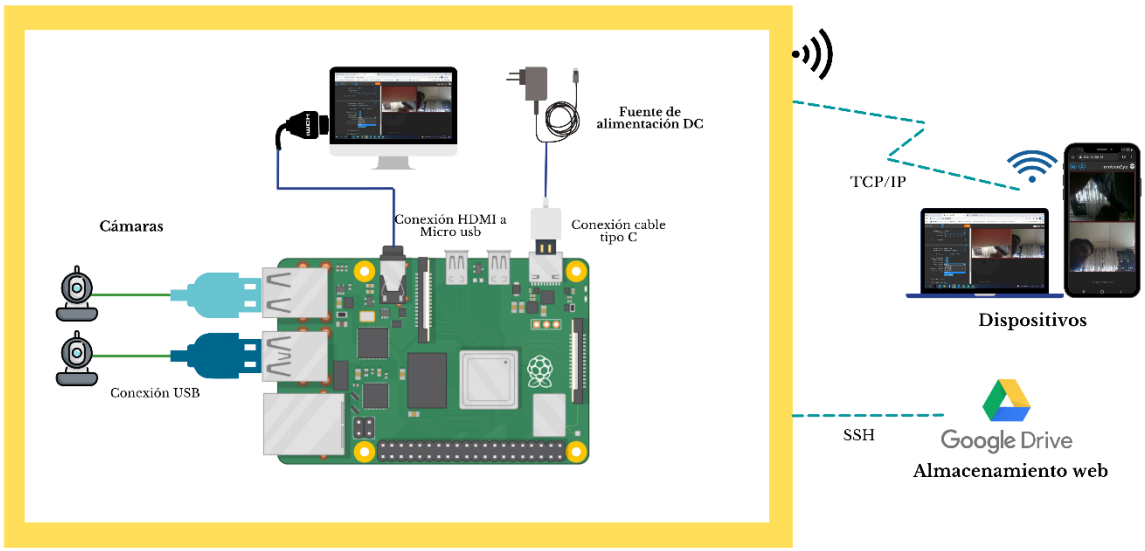


Figura 3.6 Diagrama de conexiones físicas de la etapa 2 – Monitoreo de cámaras

En la Figura 3.6 se evidencia el diagrama de conexiones físicas de la etapa 2 del proyecto respecto al monitoreo de cámaras. Este se segmenta con el uso de los siguientes dispositivos:

- Raspberry Pi 3 B+
- 2 cámaras web MTQ HD.
- 1 monitor con entrada HDMI.
- Fuente de alimentación.

Las cámaras se conectan al raspberry pi por medio de conectores USB, el monitor y fuente de alimentación se conectan con cables HDMI a micro USB y cable tipo V8 respectivamente. Dentro de este entorno, se emplean el uso de dispositivos electrónicos personales como celulares, tablets o laptops; para la interacción con el sistema de forma inalámbrica. Y el almacenamiento web mediante el protocolo de administración remota SSH.

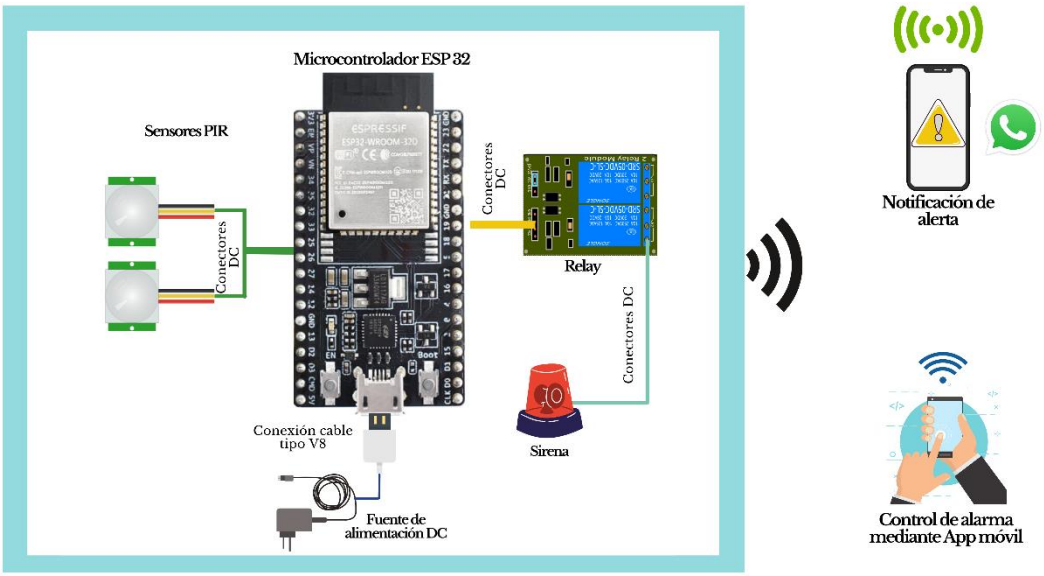


Figura 3.7 Diagrama de conexiones físicas de la etapa 2 – Lectura de sensores y notificación de alerta

Se puede evidenciar en la Figura 3.7 el diagrama de conexiones físicas de la etapa 2 correspondiente a la lectura de sensores y notificación de alerta. Para la implementación de esta sección, se hace uso de los siguientes dispositivos:

- Microcontrolador ESP 32.
- 2 sensores PIR.
- Fuente de alimentación DC.
- Sirena 12V/110V para alarma comunitaria.

Los sensores y la sirena se conectan a la placa ESP32 por medio de conectores DC, además la fuente de alimentación, la cual proporciona la energía para el funcionamiento del equipo, se conecta por medio de un cable USB tipo v8 a la tarjeta microcontroladora. La notificación de alerta es enviada a los dispositivos móviles a través de las aplicaciones de mensajería WhatsApp o Telegram. La comunicación entre la Raspberry y microcontrolador ESP32 se realiza mediante el desarrollo de una App móvil y la plataforma digital Firebase.

3.5 Etapa de reconocimiento facial

Se realiza el proceso de detección de rostro y de reconocimiento en esta etapa aplicando librerías especializadas en reconocimiento facial como OpenCV la cual trabaja con el lenguaje de programación Python, Es una librería compleja y completa la cual utiliza Machine Learning para realizar el proceso de detección y comparación de rostros.

3.5.1 Diagrama de flujo detallado del archivo reconocimiento facial.

En la Figura 3.8 se puede observar el diagrama de flujo de la etapa de reconocimiento facial, esta consta de tres procesos, si el usuario es nuevo debe registrarse ingresando su nombre y cédula correctamente para luego colocar su rostro frente a la cámara, se almacenarán las imágenes del rostro en un

directorio específico para luego proceder con el proceso de entrenamiento de cada imagen.

El usuario podrá verificar si consta registrado en la pestaña de verificar registro, este segundo proceso permite evitar pasar por el proceso de registro en caso de que el usuario no recuerde si está o no registrado.

Finalmente, si el usuario está registrado, al momento de colocar su cara para la detección saldrá un mensaje de “Acceso correcto más el nombre del usuario” y se abrirá la puerta principal, en caso de que no esté registrado se observará por pantalla un mensaje de Acceso incorrecto y no se podrá abrir la puerta de ingreso.

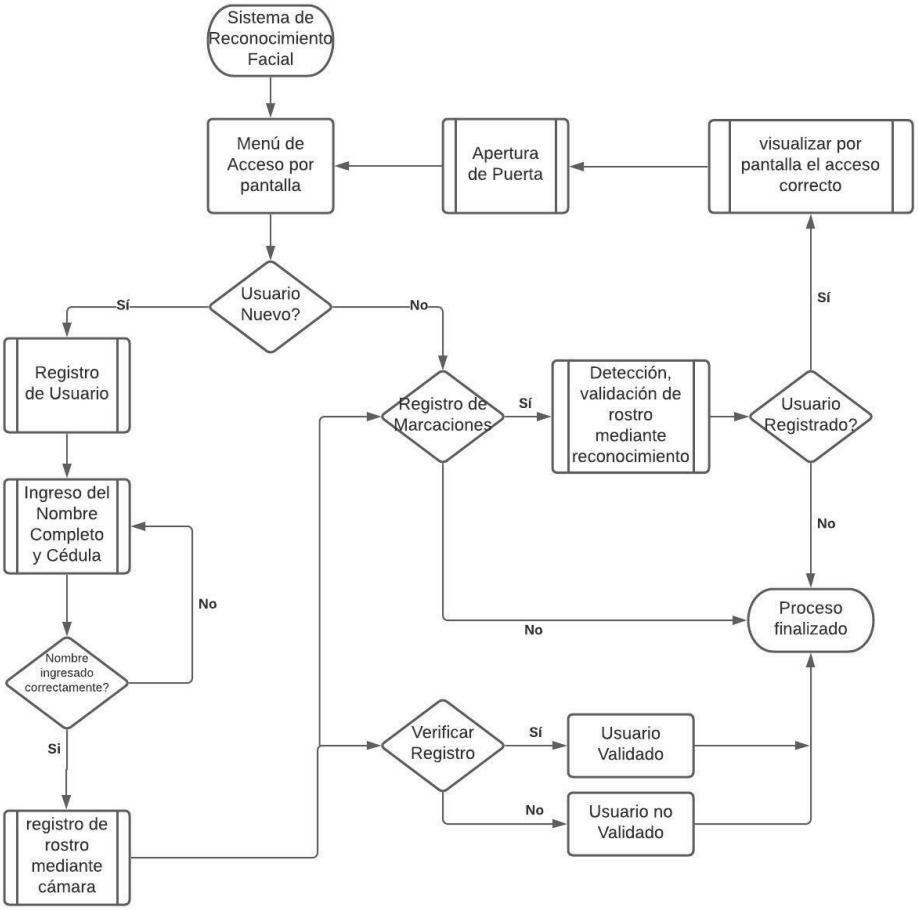


Figura 3.8 Diagrama de flujo del reconocimiento facial

3.5.2 Descripción de la función crear_registro()

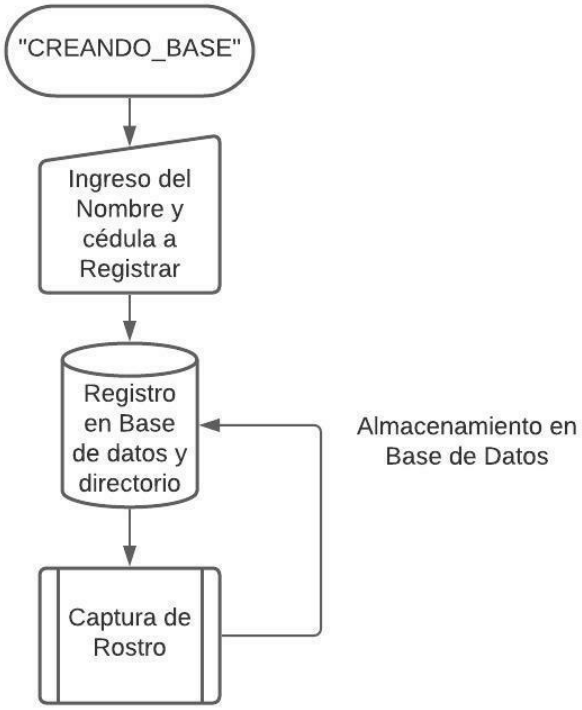


Figura 3.9 Diagrama de flujo de la función creando_base

Se puede observar en la Figura 3.9 el diagrama de flujo del proceso creando base el cual es la creación del directorio donde se almacenarán las imágenes como tal con el respectivo nombre de cada usuario y, además, se registrará la información en una base de datos. Se importan librerías que sirven para el debido proceso del algoritmo. La librería cv2 es de OpenCV permite dibujar un rectángulo alrededor del rostro para tomar 100 fotos y detectar sus características y poder almacenar las imágenes en blanco y negro para el respectivo proceso de reconocimiento.

3.5.3 Descripción de la función train()

En la Figura 3.10 se puede observar el diagrama de flujos del proceso Entrenando, el cual permitirá entrenar las imágenes.

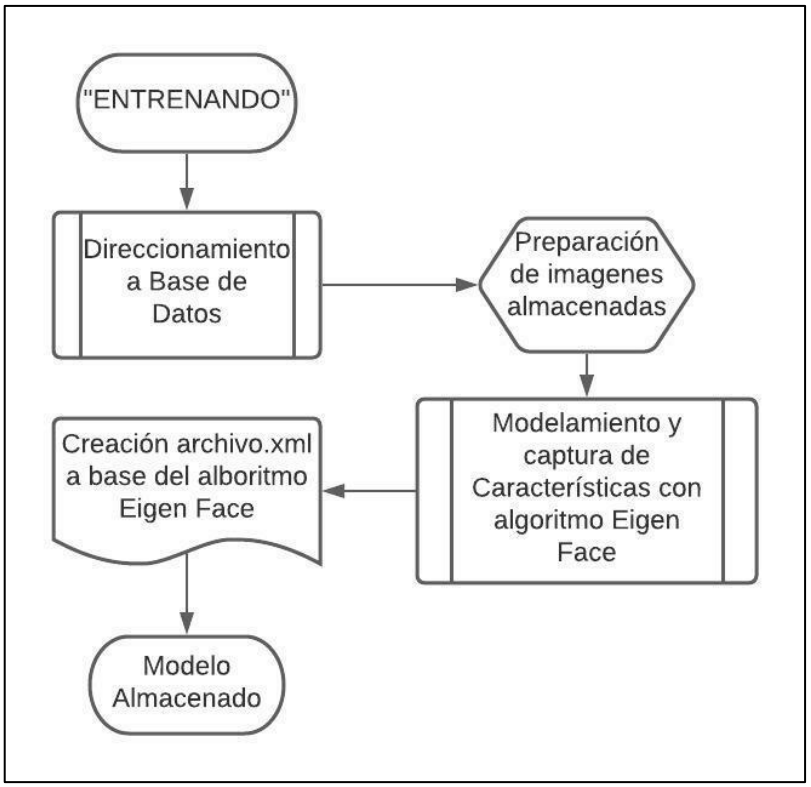


Figura 3.10 Diagrama de flujo de la función entrenando.py

Se ingresa al directorio llamado “Data” para entrenar cada archivo donde se encuentran las imágenes capturadas de cada usuario y crear una lista con los nombres de los mismos usuarios. Luego, automáticamente se crea un archivo.xml con las características principales de cada usuario utilizando el algoritmo de reconocimiento Eigen Face y finalmente mostrará una vez concluido todos los procesos, un mensaje de Modelo almacenado.

3.5.4 Descripción de la función Detect()

Se detalla en la Figura 3.11 la solución mediante diagrama de flujos la cual, representa la parte final del sistema, reconocimiento facial y es donde se realiza la comparación entre los rostros almacenados en la base de datos utilizando el archivo.xml el cual guarda las características de cada roto, y el video en tiempo real para la verificación, comprobación y Acceso del sistema. El algoritmo Eigen Face permite tomar las características en tiempo real y

compararlas con las imágenes almacenadas haciendo posible el Acceso del usuario o la denegación de este mismo.

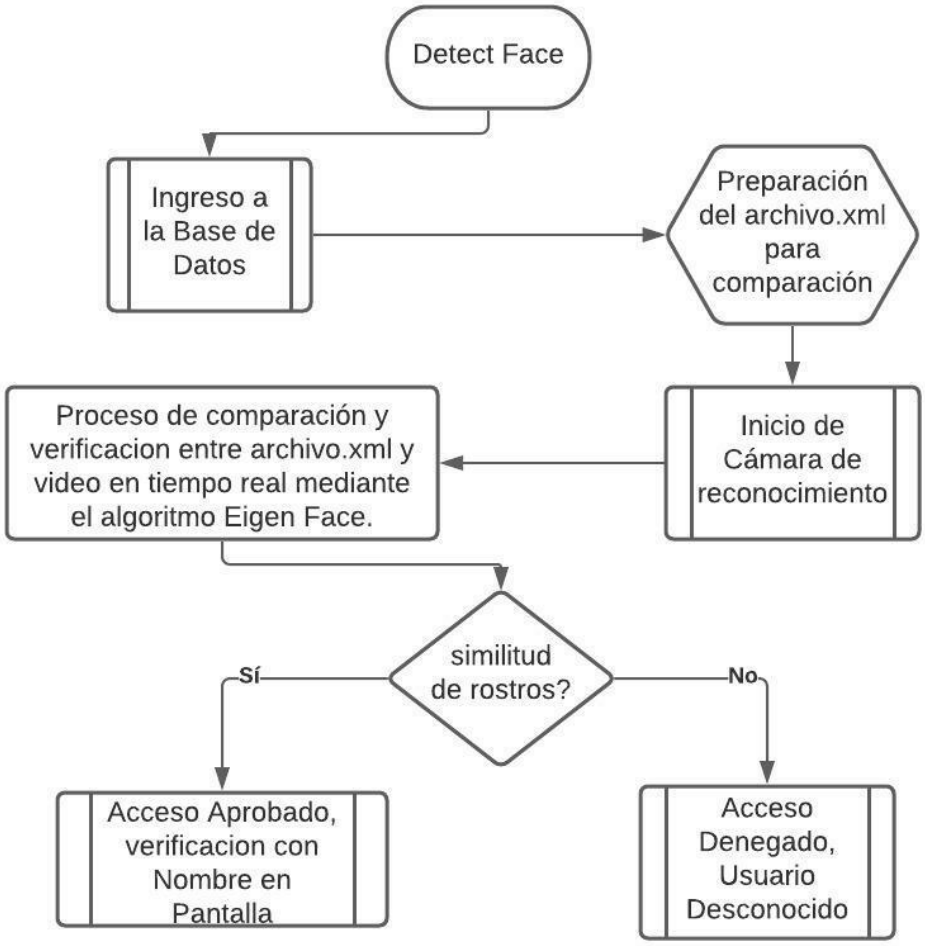


Figura 3.11 Diagrama de flujo del archivo reconocimiento facial.py

3.5.5 Activación de interfaz para trabajar remotamente

Para poder trabajar remotamente mediante Wifi se deberá activar la opción de SSH la cual mediante el programa de Putty el cual permite conexión remota, se logrará establecer los protocolos de conexión y con el comando sudo raspi-config se podrá entrar a la Configuration Tool y en Interface Options activar el modo SSH como se puede observar en la Figura 3.12.

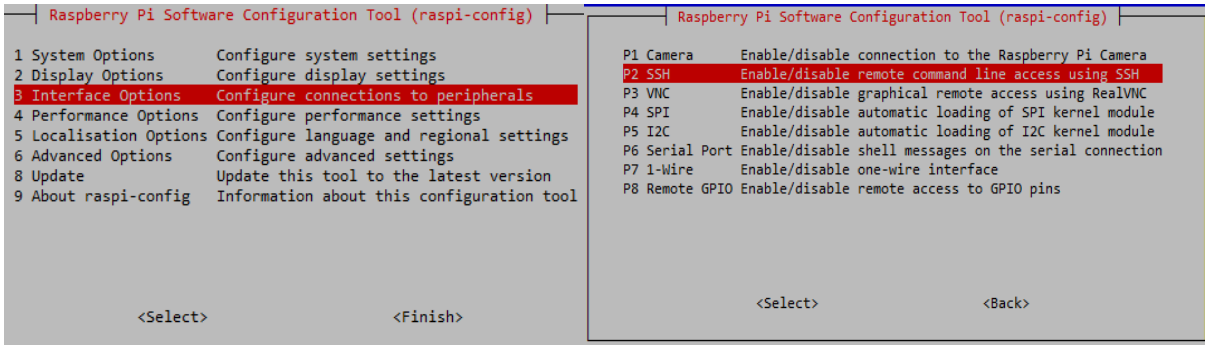


Figura 3.12 Activación de Interfaz SSH

3.5.6 Instalación del lenguaje de programación Python

Una vez instalado Raspbian en el Rasperry Pi con sus respectivas configuraciones para poder operar de forma remota, se procede a la instalación de python3 como se muestra en la Figura 3.13 mediante una línea de código. Este será el lenguaje que permitirá realizar toda la programación con un fin específico el cual viene con librerías por defecto incluidas como por ejemplo Numpy, Datetime, entre otras.

```
Sudo nano apt-get install python3
```

Figura 3.13 Comando para descargar Python 3.7

3.5.7 Instalación de OpenCV

Para la instalación correcta de OpenCv se necesita algunas líneas de códigos, como vemos en la Figura 3.14 la línea 1 permite tener actualizadas todas las listas de paquetes y tener versiones actualizadas de todos los paquetes requeridos en cada librería instalada. La línea 2 permite actualizar los paquetes que estén con versiones anteriores respetando las debidas configuraciones del software. En la línea 3 se realiza la instalación de algunos paquetes que serán eficientes y útiles al momento de trabajar con OpenCv. En la línea 4 se procede a la respectiva instalación de la librería OpenCv y la versión requerida actualmente.

```
1. Sudo apt-get update
2. Sudo apt-get upgrade
3. Sudo apt-get install libhdf5-dev libhdf5-serial-dev libatlas-base-dev libjasper-dev
   libqtgui4 libqt4-test
4. Sudo pip3 install opencv-contrib-python==4.1.0.25
```

Figura 3.14 Comandos para instalar OpenCV

3.5.8 Instalación de librerías tkinter e imutils.

```
sudo apt-get install Python-tk
sudo pip3 install imutils
```

Figura 3.15 Instalación de tkinter e imutils en consola

En la Figura 3.15 se observan dos líneas de comandos, la primera es la instalación de la librería tkinter la cual permite realizar la interfaz gráfica para mostrar un menú interactivo de forma visual. La segunda línea es la instalación de imutils la cual permite procesar imágenes de forma rápida y correcta.

```
sudo apt install mariadb-server
sudo mysql_secure_installation
systemctl status mariadb.service
sudo ufw allow mysql
CREATE USER 'root'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON * . * TO 'root'@'localhost';
FLUSH PRIVILEGES;
mysql -u root -p
CREATE DATABASE “name”;
CREATE TABLE “name”;
```

Figura 3.16 creación de base de datos y tabla respectivamente

En la Figura 3.16 se detalla las líneas de comando para poder instalar la Base de Datos mariadb, la cual permitirá registrar la información de cada

usuario. La primera línea instala la base respectivamente en raspbian. La segunda línea hace que sea segura la instalación de la base con los privilegios de instalación. La tercera y cuarta línea verifica que la instalación se hizo correcta mente y comprueba que esté operativa la base.

La quinta, sexta y séptima línea crea un usuario administrador con su respectiva clave para otorgarle los accesos y permisos privilegiados y pueda realizar cualquier cambio. La octava línea, una vez realizado los pasos anteriores se reinicia el sistema y se entra con el respectivo usuario y clave creado. Las dos últimas líneas crean una base de datos con el nombre a decisión propia y una tabla para registrar la información de los usuarios respectivamente.

3.5.9 Simulación del sistema

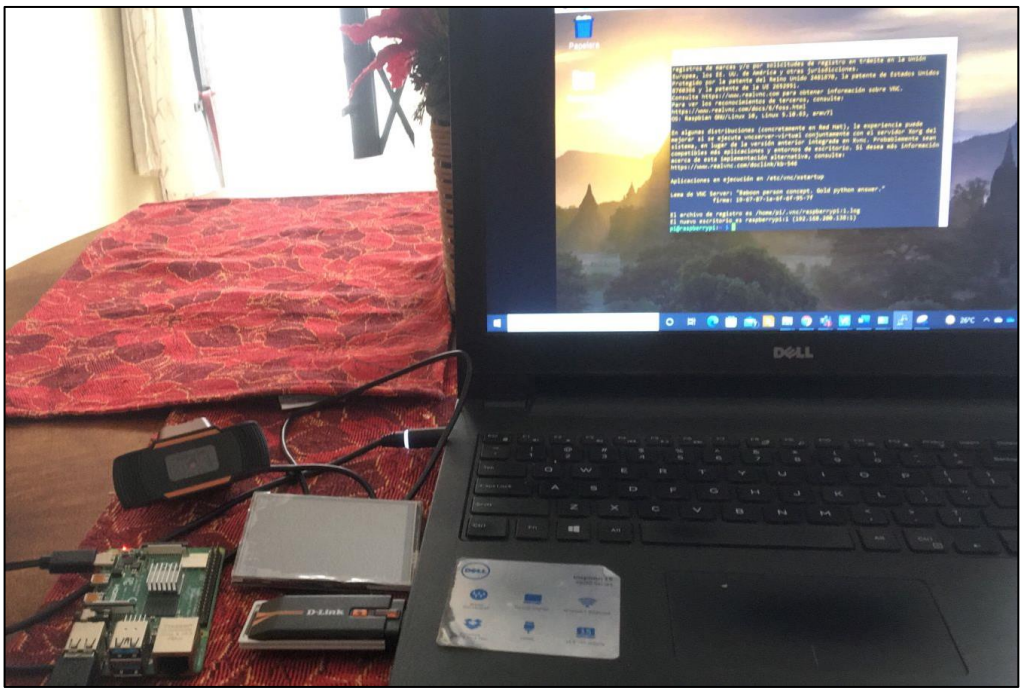


Figura 3.17 implementación general física del prototipo de RF

El sistema cuenta con varios equipos a utilizar, entre ellos se tiene una cámara USB, una laptop para las respectivas pruebas y configuraciones, teclado USB para el registro y una display TFT de 3,5inch RPi para poder visualizar la

interfaz gráfica del sistema e ingresar los datos para registrarlos como se muestra en la Figura 3.17.

En la Figura 3.18 se detalla las líneas de comando que servirán para tener operativa la pantalla y poder visualizar el sistema como tal.

```
Sudo rm -rf LCD-show
Git clone https://github.com/goodtft/LCD-show.git
Sudo apt install git
Sudo Chmod -R 755 LCD-show
Cd LCD-show/
Sudo ./LCD-show
```

Figura 3.18 comandos para activar y utilizar la pantalla TFT de 3,5 pulgadas

Una vez que se hayan ejecutado los comandos de la Figura 3.18, se observará en la pantalla TFT el entorno gráfico de Raspbian como se visualiza en la Figura 3.19 y se podrá navegar de manera libre y ejecutar el programa principal para la respectiva simulación.

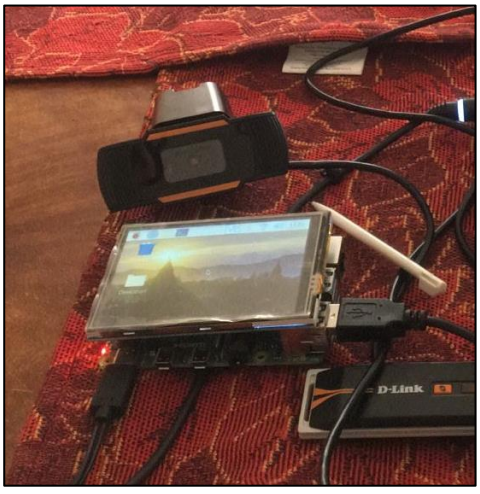


Figura 3.19 Entorno gráfico visible en raspberry pi utilizando una pantalla TFT

Cuando se ejecuta el archivo programa_principal.py mostrará un menú interactivo en el cual se puede seleccionar la pestaña que se desea visualizar,

en este caso como se muestra en la Figura 3.20 se comenzará con el proceso de registro y creación del usuario en la base de datos.

Figura 3.20 Ejecución del programa principal, pestaña de registro en pantalla LCD TFT

Una vez inicializado el proceso de registro aparecerá una pantalla donde se visualizará en tiempo real el rostro del usuario a registrar enmarcado con un cuadro de color verde el cual tiene como función capturar esa fracción de rostro como se muestra en la Figura 3.21. Luego, almacenarla en el directorio llamado Data en una carpeta con el respectivo nombre del usuario los rostros capturados como se muestra en la Figura 3.22. Finalmente se realizará el proceso de entrenamiento y se registrará el nombre y cédula en la base de datos creada llamada Tesis en la tabla IndustriaA para llevar el respectivo control de registros tal cual se observa en la Figura 3.23.

Figura 3.21 Respectiva captura para el registro del rostro

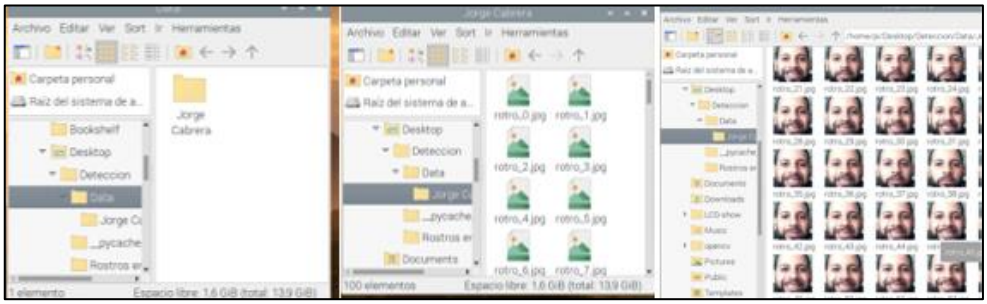


Figura 3.22 Proceso de registro de usuario en directorio y base de datos

```
MariaDB [Tesis]> SELECT * FROM IndustriaA;
+--------+------------+---------------+
| id_usu | cta_usu    | nombre        |
+--------+------------+---------------+
|     62 | 0928315316 | Jorge Cabrera |
+--------+------------+---------------+
1 row in set (0.001 sec)
```

Figura 3.23 Registro de información en base de datos

Una vez realizado el registro del usuario se procede a la siguiente pestaña verificar registro del menú interactivo para validar o verificar que el usuario fue creado como se muestra en la Figura 3.24. Se ingresa el nombre del usuario y se oprime el botón verificar constancia. Si el usuario está registrado saldrá un mensaje de usuario validado como se muestra en la Figura 3.25 y si el usuario no está registrado saldrá un mensaje de usuario no validado mostrando el resultado en la Figura 3.26.

Figura 3.24 Visualización gráfica de verificación y comprobación

Figura 3.25 mensaje de comprobación de usuario validado

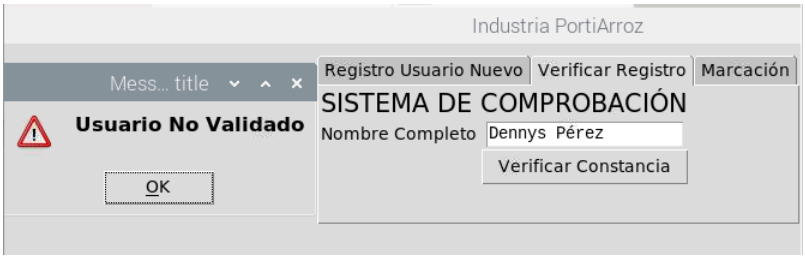


Figura 3.26 mensaje de comprobación de usuario no validado

Finalmente, la última pestaña del menú interactivo permite realizar el Acceso a la industria, como se visualiza en la Figura 3.27 se solicita ingresar el nombre para la respectiva comprobación de registro en la base de datos. Luego, se abrirá una ventana la cual comparará las fotos almacenadas en la carpeta de cada usuario y verificará si el rostro comparado es igual al almacenado como se muestra en la Figura 3.28 para final mente mostrar un

mensaje de Acceso correcto o Acceso incorrecto como se muestra en la Figura 3.29 y se abrirá la puerta dependiendo del mensaje mostrado oprimiendo el botón de Abrir puerta.

Figura 3.27 Marcación para Acceder mediante el menú interactivo

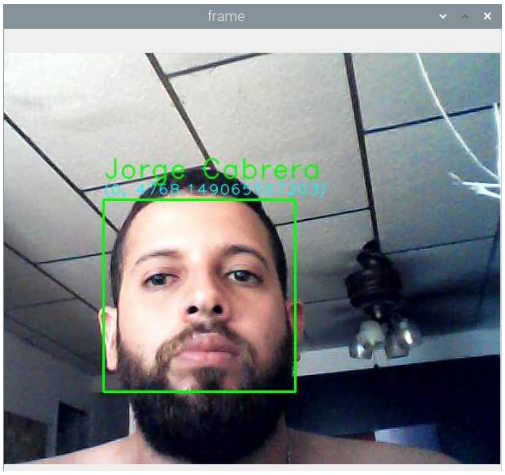


Figura 3.28 comparación de rostro con imagen almacenada en el registro

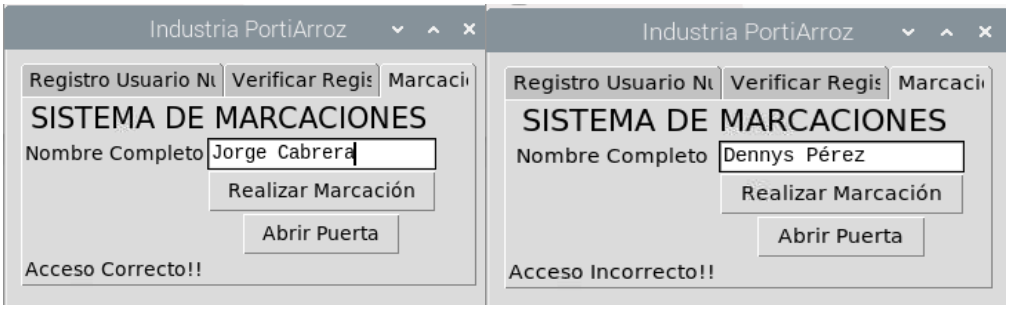


Figura 3.29 mensaje de Acceso correcto o Acceso incorrecto al usuario

3.6 Etapa de monitoreo de vigilancia y notificación de alerta

Para esta segunda etapa, se diseña y simula el monitoreo de vigilancia y el circuito que envía la notificación de alerta; se detallan los procesos correspondientes a continuación.

3.6.1 Monitoreo y grabación de cámaras.

Se detalla en la Figura 3.30 el diagrama de flujo del proceso de monitoreo y grabación mediante cámaras en la plataforma de videovigilancia. Primero se inicia con la carga del sistema operativo de MotionEye en la microSD conectada a la Raspberry Pi. Luego que termine el proceso se asigna una dirección IP, que mediante esta se accede al sistema de cámaras por medio de un navegador web; se configura un user y password para ingresar al sistema. Dentro de las

opciones de configuración se enlaza la grabación de video con un almacenamiento de datos web (Google Drive). En caso de que se desee que las cámaras graben, se habilita la opción de grabar; la grabación se subirá automáticamente a la carpeta de video en Google Drive una vez que se desactive la opción.

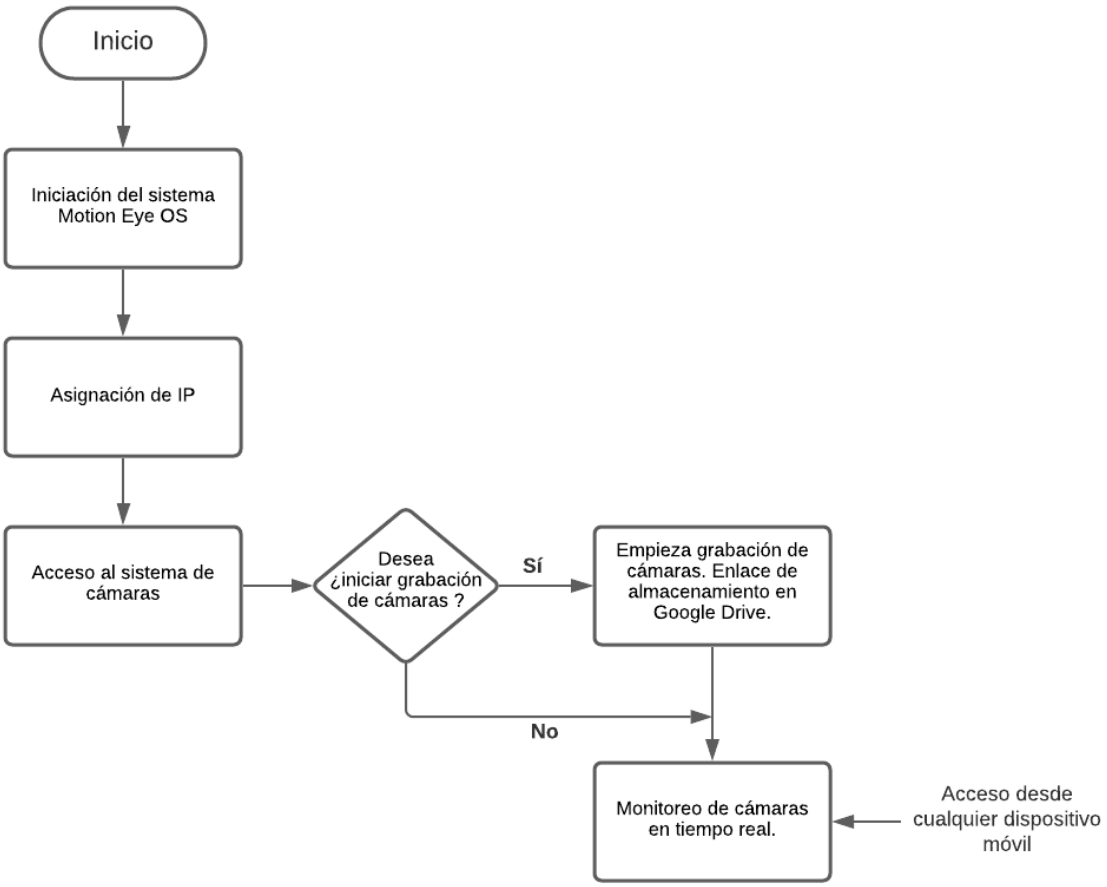


Figura 3.30 Proceso de monitoreo y grabación de cámaras ilustrado en un diagrama de flujo

3.6.2 Instalación de Putty

Una vez instalado el sistema operativo Raspbian en la tarjeta microcontroladora Raspberry Pi, se instala el programa Putty para la configuración de la tarjeta, este programa es un emulador gratuito que permite la conexión al servidor de la Raspberry Pi por medio del protocolo Secure Shell (SSH) para

acceder remotamente a la interfaz de inicio. Como se observa en la Figura 3.31 su instalación se la realiza desde la página oficial mediante el link mostrado.

https://www.putty.org/download

Figura 3.31 Enlace para descargar programa Putty

Mediante Putty se ingresa a la interfaz de configuración de inicio a través de la IP de la Raspberry Pi como se muestra en la Figura 3.21. En el mismo se procede a configurar parámetros importantes como el user y password de acceso en el apartado de System Options, activar el Wifi para conectividad inalámbrica, de igual forma la expansión de memoria para tomar todo el espacio de la tarjeta microSD y la computadora virtual de red (VNC) en la sección de Advanced Options, habilitar los periféricos disponibles para las cámaras en el apartado de Performance Options y finalizar.

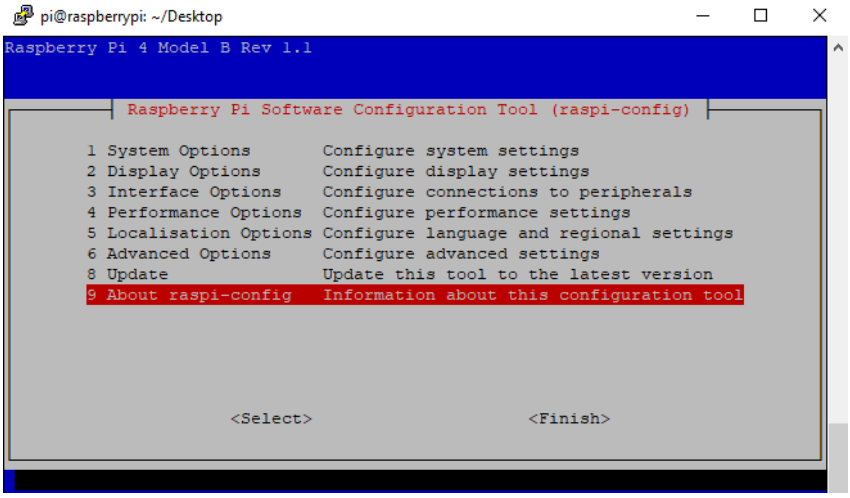


Figura 3.32 Interfaz de Configuración de raspberry PI a través de Putty

3.6.3 Instalación del sistema operativo MotionEye

Motion Eyes permite el control y monitoreo de cámaras mediante una interfaz web. Se accede a la Raspberry pi mediante el programa Putty, con el user y password configurados en el punto anterior. Primero se debe asegurar que el sistema operativo Raspbian esté actualizado, por lo que se debe ejecutar el siguiente comando mostrado en la Figura 3.33.

```
Sudo apt – get update
```

Figura 3.33 Comando para verificación de SO actualizado

En caso de que no esté actualizado el sistema operativo, se procede con la actualización mediante el comando mostrado en la Figura 3.34.

```
Sudo apt- get upgrade
```

Figura 3.34 Comando para actualizar SO

Para el proceso de instalación de MotionEye, los paquetes y librerías, se debe ejecutar los siguientes indicados en la Figura 3.35.

```
Sudo nano /etc/default/motion
Sudo nano /etc/motion/motion.conf
```

Figura 3.35 Comandos de instalación de MotionEye, librerías y paquetes

Con la correcta instalación del sistema operativo, se ejecuta los comandos de la Figura 3.36. para cambiar las configuraciones del MotionEye, habilitar las cámaras y modificar la altura y ancho de las mismas (pixeles). Por último, se inicia el sistema.

```
Sudo nano /etc/motion/motion.conf
Web cam_localhost on
Width 1280
Height 720
Sudo Service motion start
```

Figura 3.36 Comandos configuración de MotionEye, habilitar cámaras y modificar pixeles

3.6.4 Instalación del programa Network Mapper (nmap)

Se instala el aplicativo nmap que permite explorar la red y visualizar los hosts activos en la misma, e incluso mediante este programa se puede observar la IP de cada host asignado en la red y el sistema operativo que utiliza. Se observa en la Figura 3.37 el link de descarga que se realiza desde la página oficial.

Figura 3.37 Enlace para descargar programa nmap

Luego de instalar el programa se procede hacer el escaneo de la red, configurado en un perfil de Ping Scan para determinar la IP que está asociada al raspberry PI. En la Figura 3.38 se muestra el escaneo completo de la red y se evidencia la dirección IP y MAC con nombre Raspberry Pi Foundation.

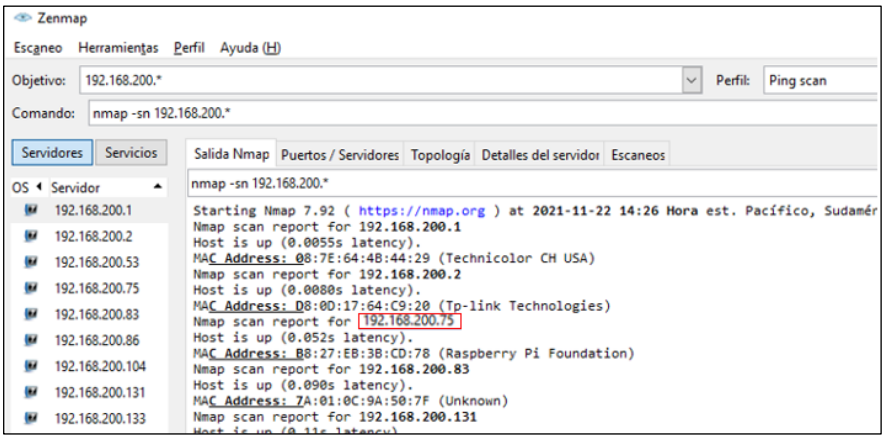


Figura 3.38 Dirección IP para acceso a MotionEye

3.6.5 Inicio de sesión y configuración de MotionEye

Tras obtener la dirección IP con el proceso detallado en el punto 3.6.2, se inicia sesión de MotionEye colocando la dirección IP en un navegador Web. Como se muestra en la Figura 3.39 aparece una interfaz de Login en el cual se coloca el username y password configurados en el punto 3.6.1.

Luego que se acceder a las configuraciones de la plataforma se procede añadir las cámaras que conforman el sistema de videovigilancia, esto se realiza ingresando a la opción de add cámara mostrado en la Figura 3.40 y seleccionando las cámaras que se encuentren activas; en el menú de Video Device se configura el nombre de cada cámara y la resolución.

Por último, se logra visualizar en la Figura 3.41 las cámaras activas accediendo de forma inalámbrica a la plataforma de monitoreo.

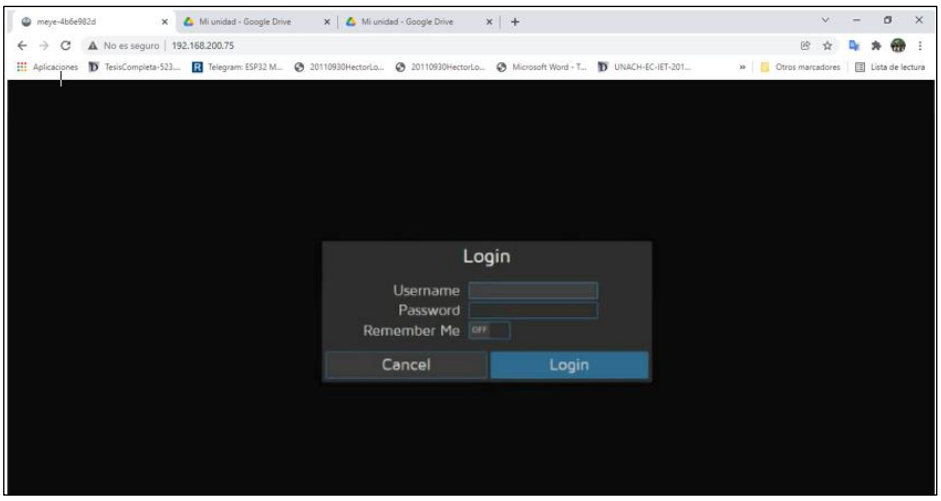


Figura 3.39 Interfaz de Login de MotionEye

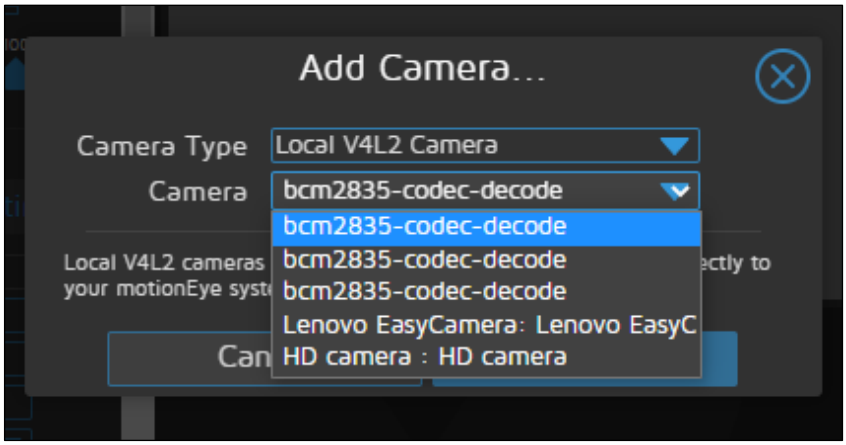


Figura 3.40 Agregación de cámaras en la plataforma MotionEye

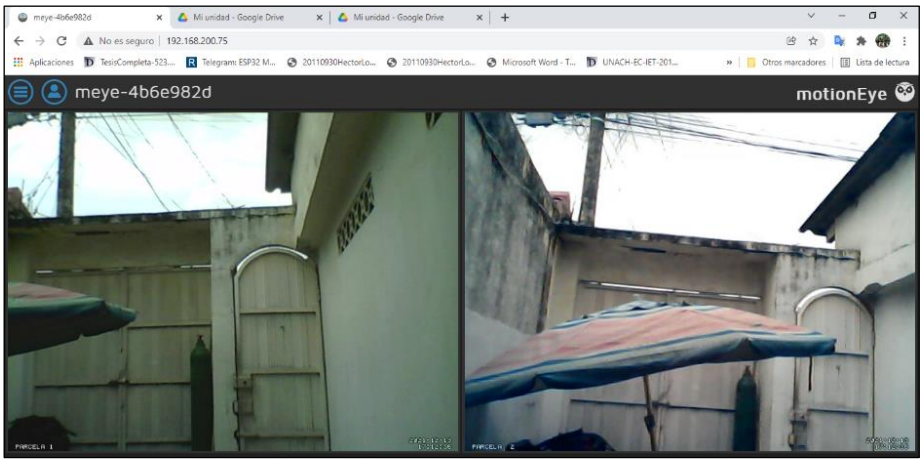


Figura 3.41 Monitoreo de cámaras en la plataforma MotionEye

3.6.6 Configuración de almacenamiento de video en Google drive.

Dentro del menú de opciones se procede a enlazar el almacenamiento de la nube Google Drive con la plataforma MotionEye, de tal forma que las grabaciones de video sean subidas, archivadas y puedan ser revisadas posteriormente en este servidor.

En la Figura 3.42 se muestra la configuración realizada para la cámara 1 que tiene como nombre “PARCELA 1”. En el apartado de File Storage se habilita la opción de “Upload Media Files” y “Upload Movies” seleccionando como servicio de carga Google Drive. En la sección de Location se escribe el nombre de la carpeta en donde se alojarán las grabaciones de la cámara respectiva; en este caso cámara 1.

Se habilita las opciones de “Include subfolders” y “Clean Cloud” para que se permita crear subcarpetas referentes a cada día en que se activen las grabaciones.

Cabe aclarar que este proceso se debe realizar para cada cámara instalada ya que las grabaciones de las cámaras se guardan en carpetas distintas.

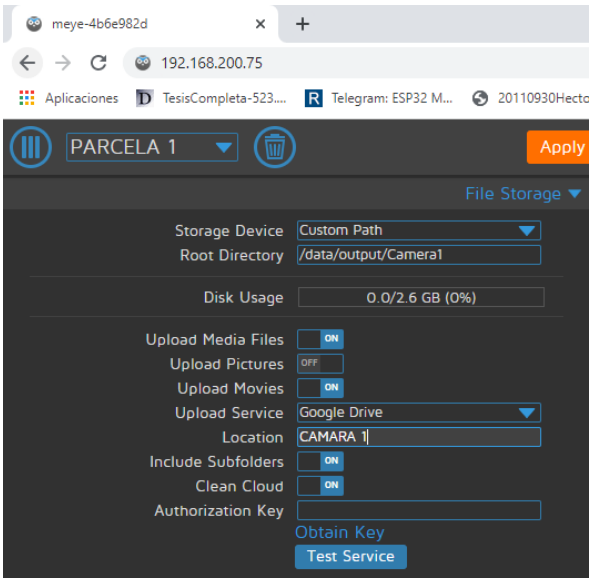


Figura 3.42 Configuración de enlace de almacenamiento de video con Google Drive

Para obtener la clave de autorización del enlace, se procede a dar clic en la opción obtener clave; automáticamente se redirige a elegir una cuenta de Google como se muestra en la Figura 3.43, la cual estará asociada al almacenamiento de grabaciones. En la figura 3.44 se visualiza la clave que proporciona el servidor para el enlace.

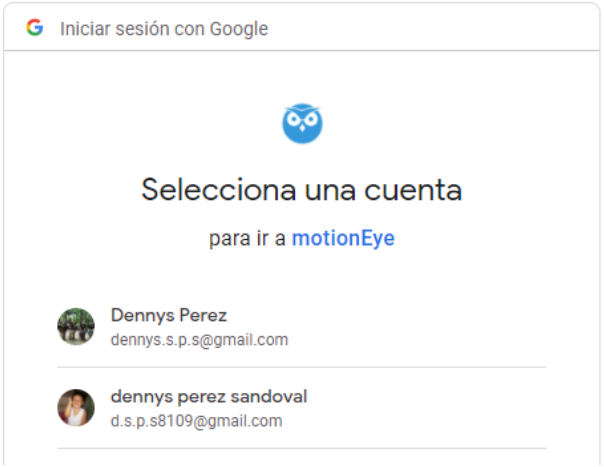


Figura 3.43 Selección de cuenta de Google para enlace de almacenamiento

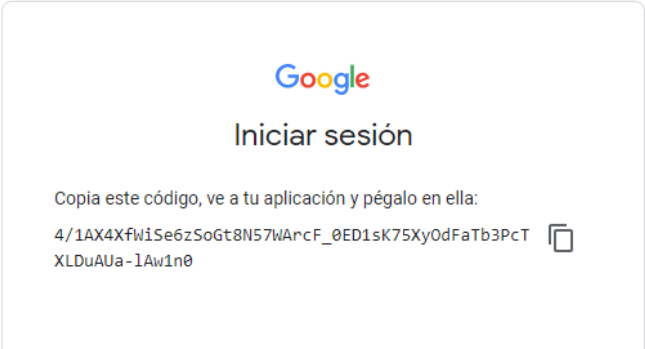


Figura 3.44 Clave de autorización para enlace de almacenamiento

La clave de autorización se copia y pega en la sección de “Autorization Key” mostrado en la Figura 3.42. Luego se aplican los cambios realizados. Para activar las grabaciones de cada cámara se selecciona la opción de grabación mostrado en el recuadro rojo en la Figura 3.45; de la misma forma para detener las grabaciones se selecciona la opción de parar grabación. Se puede observar en la Figura 3.46 se guardan los videos en las carpetas respectivas con los nombres de fechas en la sección de Location.

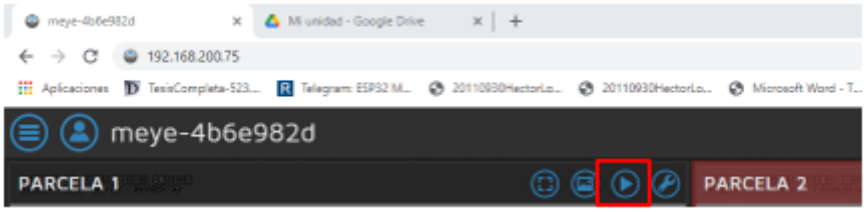


Figura 3.45 Habilitar el proceso de grabación mediante cámaras

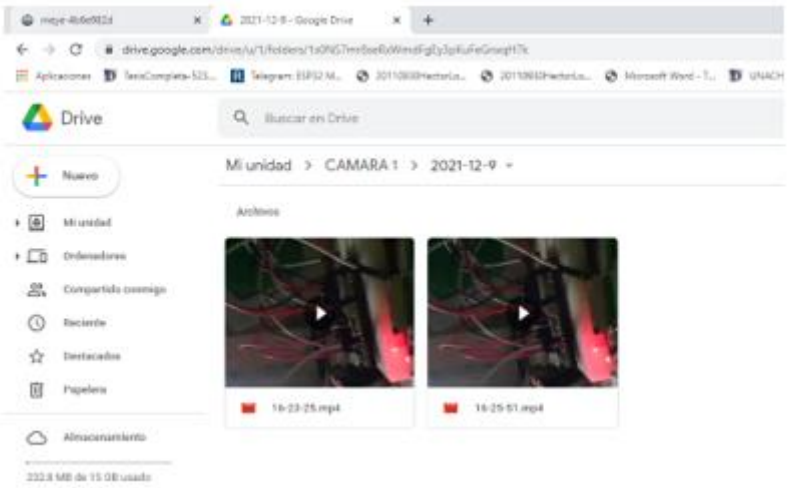


Figura 3.46 Almacenamiento de grabaciones en Google Drive

3.6.7 Proceso de notificación y activación de alarma

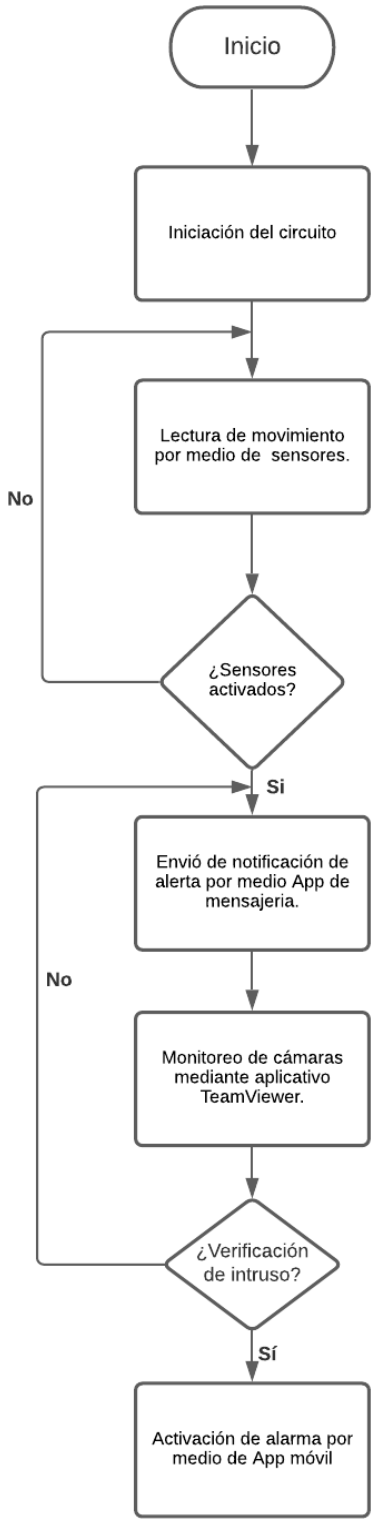


Figura 3.47 Proceso de notificación y activación de alarma mediante diagrama de flujo

Se muestra el proceso de notificación y activación de alarma en la Figura 3.47. Primero comienza por la iniciación del circuito, el cual está conformado por sensores PIR, módulo ESP32, led, relay y alarma. Se ejecuta el algoritmo respectivo y empieza la lectura de movimiento de los sensores. Al momento de activar algún sensor, se envía una notificación por medio de la mensajería de WhatsApp indicando el área en donde se detectó una actividad al cruzar las parcelas; el relay se activa. Al recibir la notificación, el personal encargado puede acceder al sistema de cámaras mediante la aplicación Team Viewer para verificar el ingreso de personas no autorizadas al área de cultivos.

Posteriormente, en caso de ser necesario se activa la alarma mediante el aplicativo desarrollado en App Inventor, el cual incorpora un control de encendido y apagado de alarma enlazado a la plataforma Firebase que permite la comunicación entre el módulo ESP 32 y la App creada.

3.6.8 Diagrama del circuito para la notificación de alerta

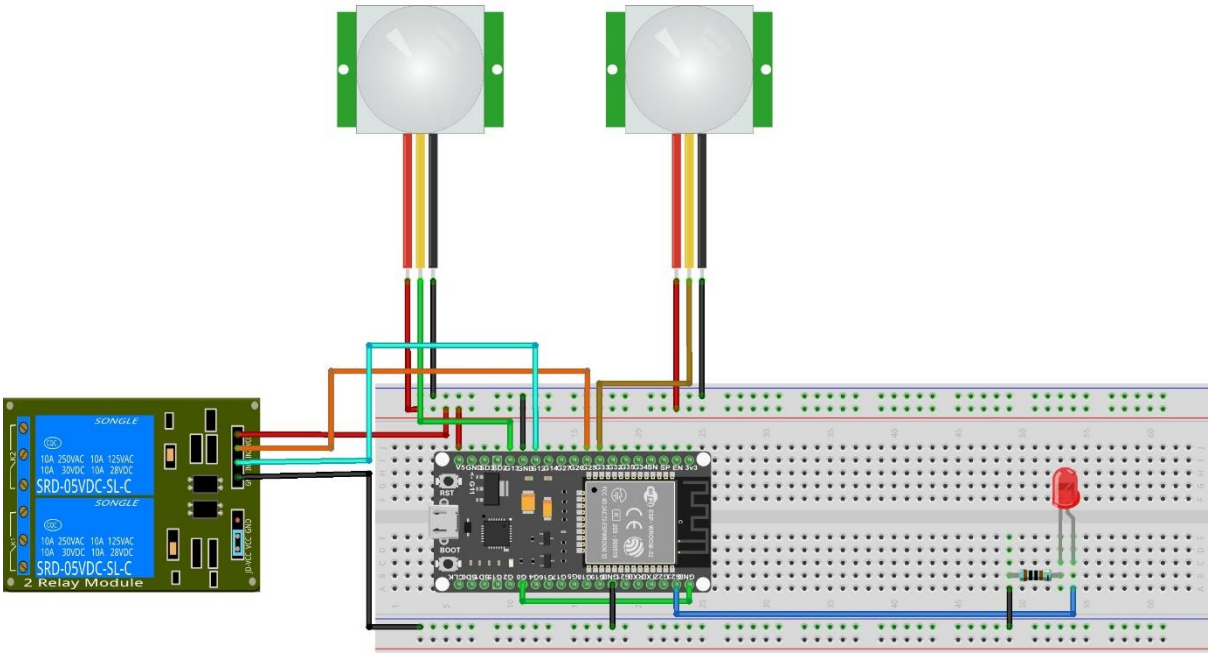


Figura 3.48 Diagrama esquemático de conexiones del circuito

Se visualiza por medio de la Figura 3.48 el respectivo diagrama esquemático donde se encuentran las conexiones del circuito, en el que se evidencia el empalme de los componentes descritos en el punto 3.4 .

3.6.9 Instalación de entorno de programación Arduino

Para la implementación del algoritmo respectivo, se utiliza el entorno de programación de Arduino, el cual se lo descarga desde la página oficial de Arduino. Este permite compilar el código que se estructuró para el cumplimiento de las funciones del sistema. Como se aprecia en la Figura 3.49, este programa contiene cinco secciones principales: menú, botones de acceso rápido, editor de texto, área de mensajes y consola.

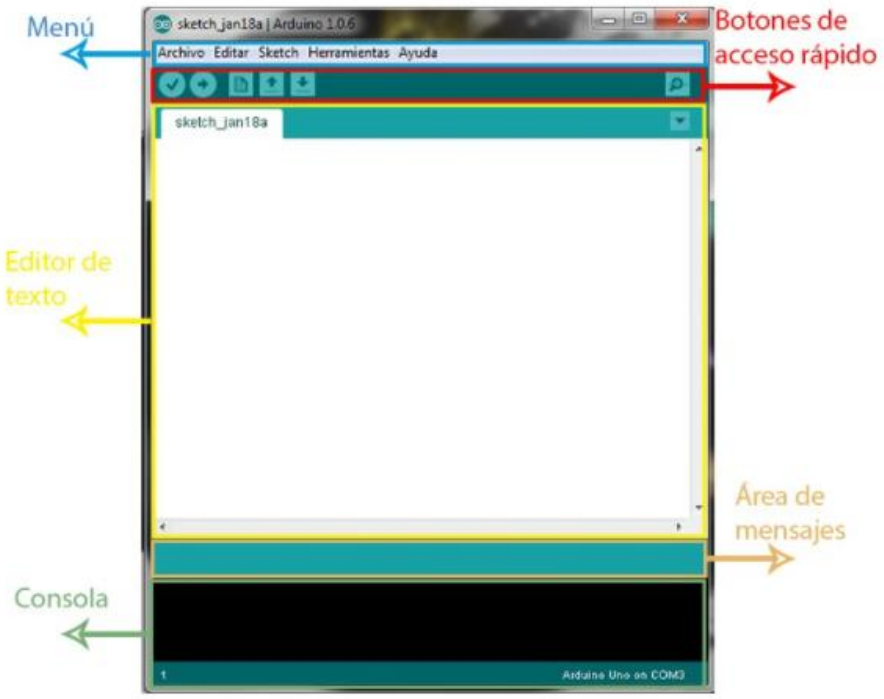


Figura 3.49 Entorno IDE Arduino y sus partes

En este proyecto se enfoca en la sección de texto, ya que en esta se escribe el código fuente del algoritmo con las funciones respectivas para la funcionalidad del circuito. Además del manejo del área de mansajes que permite

mostrar la carga de programas y errores, junto a la sección de consola que muestra el texto de salida de la compilación del código.

3.6.10 Configuración del Bot de alerta de WhatsApp

Se debe establecer un chat de comunicación en donde se reciben los mensajes que corresponden a la notificación de alerta. Para esto se hace uso de un Bot público con dominio de España; este se maneja con el número +34 698 28 89 73, para lo cual, se agrega dicho número en la agenda de contactos y como se muestra en la Figura 3.50 se envía un mensaje por WhatsApp autorizando permitir recibir mensajes. Posteriormente, el Bot proporciona el número celular establecido que recibirá las notificaciones y el apikey que corresponde a un código de autenticación de usuario; se precisa su utilidad en la descripción del algoritmo en el punto 3.6.

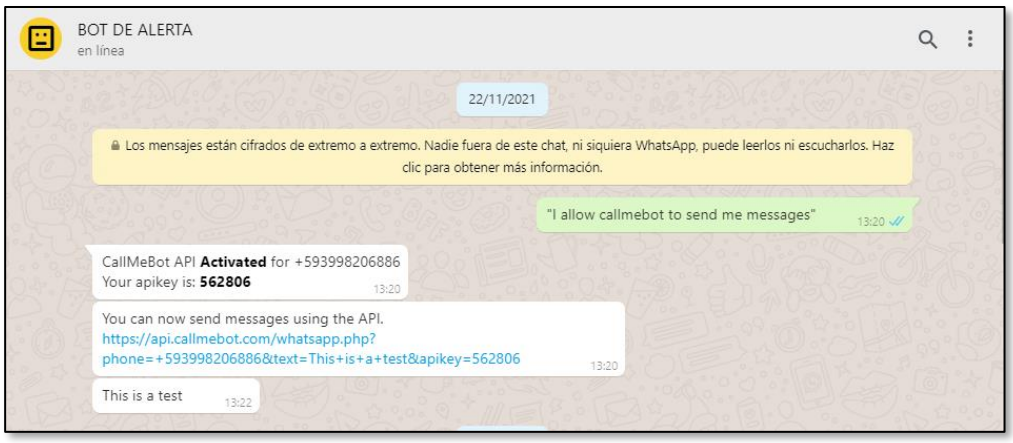


Figura 3.50 Algoritmo de detección de intrusos y envió de notificación– Parte 1

3.6.11 Creación y configuración de base de datos en Firebase

Para establecer la comunicación entre el módulo ESP32 y el aplicativo en App inventor se debe crear una base de datos para el control remoto de la alarma. Para esto se debe crear una cuenta en la página de Firebase, a esta se puede acceder directamente con una cuenta de Google. Luego se crea un nuevo

proyecto al que definimos como LED CONTROL como se evidencia en la Figura 3.51.

Figura 3.51 Creación de proyecto de Firebase

Se crea una base de datos en tiempo real seleccionando la opción referente como se muestra en la Figura 3.52. Se debe obtener la databaseURL (Host) y contraseña (auth) de la base de datos para establecer conectividad con el ESP32. Para esto se debe ingresar al apartado de cuentas de servicio, en la sección de SDK de Firebase se obtiene el host y en la sección de secretos de base de datos el auth, como se muestra en la Figura 3.53 y 3.54 respectivamente; estos códigos se los ingresa en el algoritmo desarrollado en la plataforma de Arduino.

Por consiguiente, en la sección de base de datos en tiempo real, se crea la base de datos de control, con alarma “0” en estado apagado y “1” en encendido, como se evidencia en la Figura 3.55.

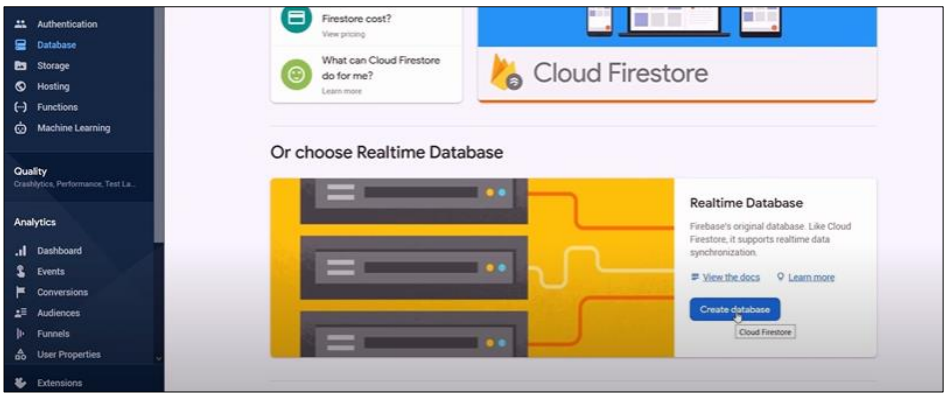


Figura 3.52 Creación de base de datos en tiempo real

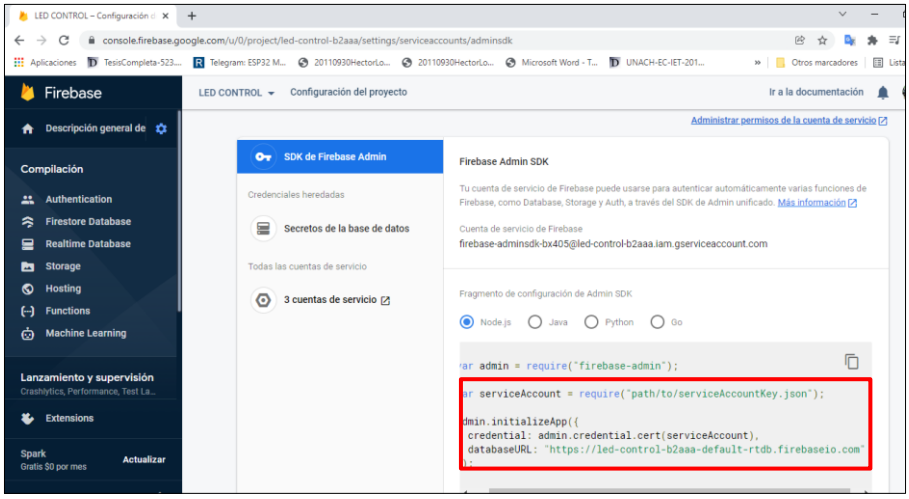


Figura 3.53 Enlace de DatabaseURL

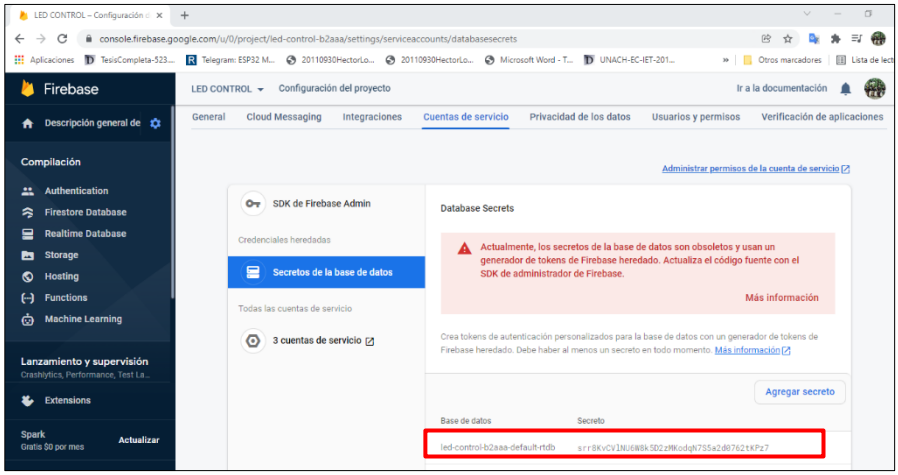


Figura 3.54 Contraseña de base de datos de Firebase

Figura 3.55 Base de datos para el control de alarma

3.6.12 Descripción del archivo etapa2.ino

El archivo en mención detalla el algoritmo ejecutado en el microcontrolador ESP32 respecto a la lectura de sensores, notificación y activación de alarma.

En la Figura 3.56 se muestra la parte 1 del código para la notificación de alerta. Se inicia llamando a las librerías principales, las cuales serán necesarias para desempeñar funciones respectivas como: WiFi.h para la conectividad inalámbrica, IOXhop_FirebaseESP32.h para la comunicación entre el módulo ESP32 con la App de Firebase, ArduinoJson.h para el almacenamiento de la estructura de datos, entre otros. Luego se ingresa el host y auth de Firebase; estos datos son proporcionados por la base de datos el cual muestra su proceso en el punto anterior, para establecer conectividad entre la App creada en App inventor y el módulo ESP32. Se indica el nombre de red y contraseña para conectarse por medio de Wifi.

```
#include <FirebaseESP32.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <IOXhop_FirebaseESP32.h>
#include <HTTPClient.h>
#include "time.h"
#include <ESP32Time.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>

#define FIREBASE_HOST "led-control-b2aaa-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "srr8KvCVlNU6W8k5D2zMKodqN7S5a2d0762tKPz7"

const char* ssid = "NETLIFE_JAVIER";
const char* password = "Javier8109_**";
```

Figura 3.56 Algoritmo de detección de intrusos y envió de notificación– Parte 1

La siguiente parte del código consiste en la parte 2 mostrado en la Figura 3.57, esta hace referencia a la asociacion del número celular y apikey del móvil al cual se enviará la notificacion por Whatsapp; estos fueron generados por medio del Bot en el punto 3.6.5 . Se definen los puertos de entrada del led, relay, sensor 1 y 2, estado, valor y conteo de activaciones respectivamente.

```
String apiKey = "562806";
String phone_number = "+593998206886";
String url;

String fireStatus = "";
int led = 23;
const int relay = 26;
int inputPin = 13;
int inputPin2 = 14;
int pirState = LOW;
int pirState2 = LOW;
int val = 0;
int val2 = 0;
int conteo = 0;
```

Figura 33.57 Algoritmo de detección de intrusos y envió de notificación– Parte 2

En la parte 3 del código mostrado en la Figura 3.58, se declara los pines de salida del relay y sensores conectados al microcontrolador; pin 23. Se genera una IP para establecer la conexión WiFi con el comando WiFi.localIP. Además, se conecta el módulo ESP32 a Firebase enviando un mensaje de estado: Sistema preparado.

```
void setup() {
  Serial.begin(115200);
  pinMode(relay, OUTPUT);
  pinMode(inputPin, INPUT);
  pinMode(inputPin2, INPUT);
  pinMode(23, OUTPUT);
  Serial.print("Connecting to WiFi to");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {

  }
  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP());
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.setString("ALARMA", "0");
  bot.sendMessage(chat_id, "Sistema Preparado!!", "");
```

Figura 3.58 Algoritmo de detección de intrusos y envió de notificación– Parte 3

En la parte 4 del código, mostrado en la Figura 3.59, se establece el inicio de la lectura de los sensores. Si la lectura de entrada del sensor es alta (high), es decir, detecta movimiento, se cambia el estado del pin de entrada pirstate procesando en el entorno de Arduino el mensaje de “movimiento detectado”. Y por medio de la función message_to_whatsapp se envía un mensaje a WhatsApp “movimiento detectado en la parcela”, el numero de la parcela varía de acuerdo con el sensor activado.

Se ejecuta la conectividad con el Firebase para poder ejecutar la acción de encendido o apagado de la alarma, si el estado es igual a 1, se activa la alarma, muestra el mensaje de alarma encendida y se activa el led, caso contrario el estado igual a 0, la alarma esta apagada y no se activa el led. El estado de alarma es controlado por medio de la App desarrollada en App Inventor.

```
void loop() {
  val = digitalRead(inputPin);  // read input value
  if (val == HIGH) {            // check if the input is HIGH
    if (pirState == LOW) {
      Serial.println("Motion detected parcela 1!");
      mensaje = "Movimiento detectado" + conteoString;
      bot.sendMessage(chat_id, mensaje, "hola0");
      message_to_whatsapp("Movimineto Detectado parcela 1.");
      pirState1 = HIGH;
    }
  }

  val2 = digitalRead(inputPin2);
  if (val2 == HIGH) {
    if (pirState2 == LOW) {
      Serial.println("Motion detected parcela 2!");
      mensaje = "Movimiento detectado parcela 2" + conteoString;
      message_to_whatsapp("Movimineto Detectado parcela 2.");
      pirState2 = HIGH;
    }
  }
```

Figura 3.59 Algoritmo de detección de intrusos y envió de notificación– Parte 4

3.6.12 Desarrollo de Aplicativo

Como se menciona en puntos anteriores, para entablar una conectividad con el monitoreo de vigilancia y control de alarma, se desarrolla un aplicativo haciendo uso del entorno de desarrollo web Mit App Inventor, el cual permite crear aplicativos móviles para Android mediante bloques de programación.

En la Figura 3.60 se muestra el diseño de la interfaz principal, la cual se la ha compuesto de un recuadro de estado de alarma para encender y apagar la misma, y otro recuadro de visor web para enlazar la plataforma de MotionEye por medio de la dirección IP otorgada en el punto 3.6.2 .

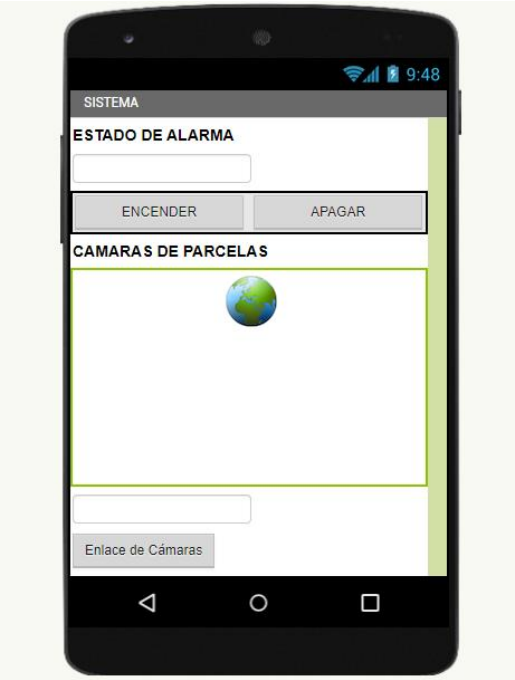


Figura 3.60 Interfaz principal del aplicativo

Se realiza la programación respectiva mediante bloques. En la Figura 3.61 se evidencia el conjunto de bloques de programación que hace referencia al control de la alarma. Se definen dos bloques de control que cumplen su función mediante un clic, el primero para encender la alarma y el segundo para apagar. Dentro de ambos bloques se integra otro bloque de control de ejecutar el cual

incorpora bloques de procedimientos (color morado); en estos se llama a la base de datos de Firebase con la etiqueta de alarma y un valor a guardar de “1” en estado encendido y “0” en estado apagado. Por último, para la visualización de los estados en mención, se agrega un bloque de texto (color verde).

La programación por bloques para enlace de cámaras con visor web se muestra en la Figura 3.62, de igual forma, se utiliza un bloque de control que es el botón de enlace de cámaras. Seguidamente se agrega el bloque de ejecutar, dentro del mismo se añade dos bloques de texto que corresponden al ingreso del localizador uniforme de recueros (URL) y el bloque de procedimiento que llama al visor web para enlazar las cámaras.

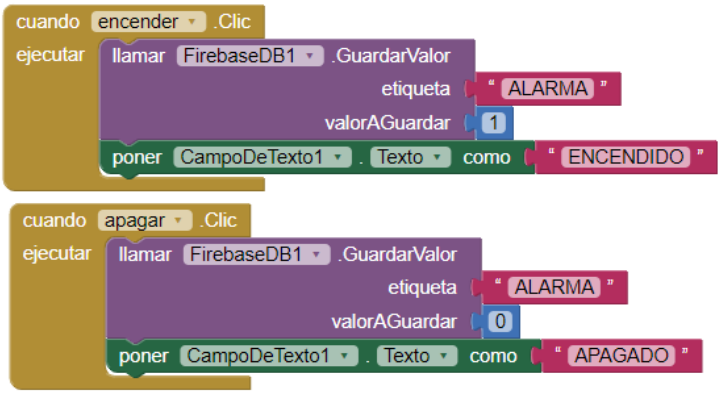


Figura 3.61 Programación por bloques de encendido y apagado de alarma

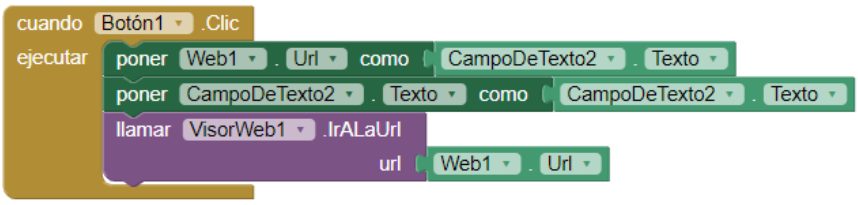


Figura 3.62 Programación por bloques para enlace de MotionEye con visor web

3.6.13 Simulación del sistema de monitoreo de vigilancia y notificación de alerta.

En la Figura 3.63 se muestra la implementación física de la etapa de monitoreo de videovigilancia y circuito de notificación de alerta. Como se indica en el punto 3.6.2 se obtiene la dirección IP de la plataforma de software MotionEye, mediante esta se puede acceder al monitoreo de cámaras de forma

inalámbrica dentro de la red desde cualquier dispositivo laptop, tablet, celulares, entre otros. Se ingresa al monitoreo de cámaras mediante un teléfono móvil como se evidencia en la Figura 3.64.

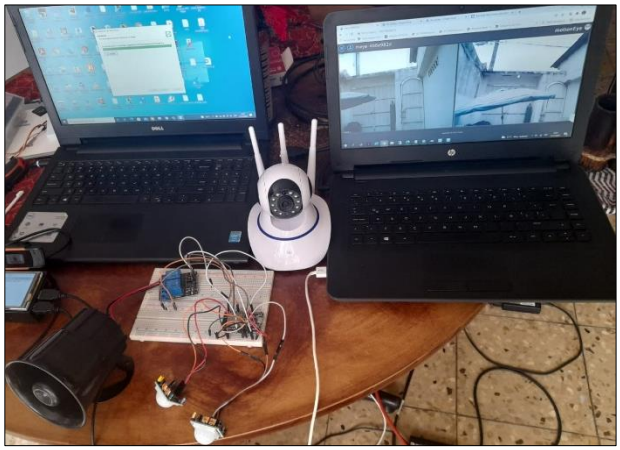


Figura 3.63 Implementación física de la etapa de monitoreo y notificación de alerta

En el caso que se desee acceder a la plataforma de videovigilancia y no se encuentre conectado dentro de la red, se realiza el acceso remoto mediante el aplicativo TeamViewer. Se descarga la aplicación en los dispositivos de acceso y mediante el ID y contraseña que proporciona TeamViewer se permite el acceso a la plataforma de monitoreo como se muestra en la Figura 3.65. El monitoreo de las cámaras mediante la aplicación se evidencia en la Figura 3.66.

Figura 3.64 Monitoreo de cámaras mediante un teléfono móvil

Figura 3.65 Conexión con ID y contraseña a laptop con plataforma ME

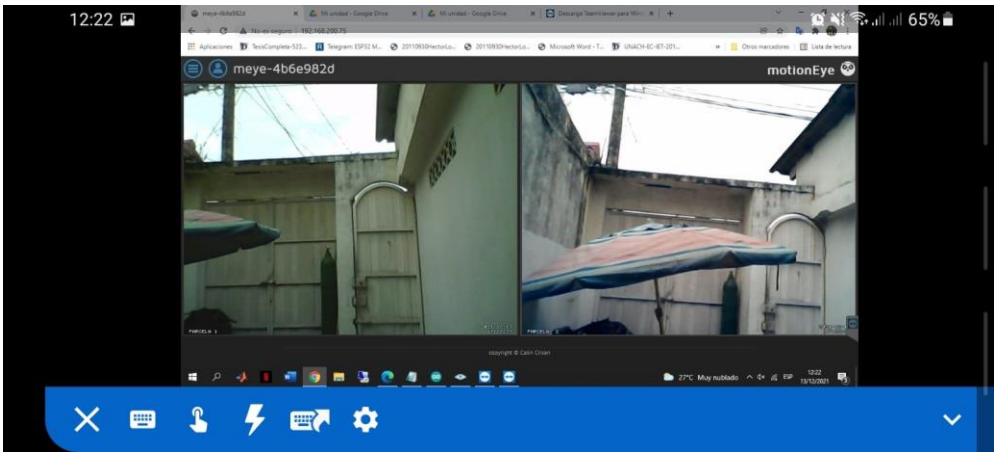


Figura 3.66 Monitoreo de cámaras de videovigilancia a través de TeamViewer

Se ejecuta el algoritmo de lectura de sensores y envío de notificaciones de alerta en el entorno de programación de Arduino. En la cual como se muestra en la Figura 3.67, al detectar movimiento se activa la alarma, la misma se ve reflejada con valor de “1” en la base de datos en tiempo real de Firebase, de igual forma en caso de que no se detecte movimiento, el estado de alarma se mantendrá en “0” como se evidencia en la Figura 3.68.

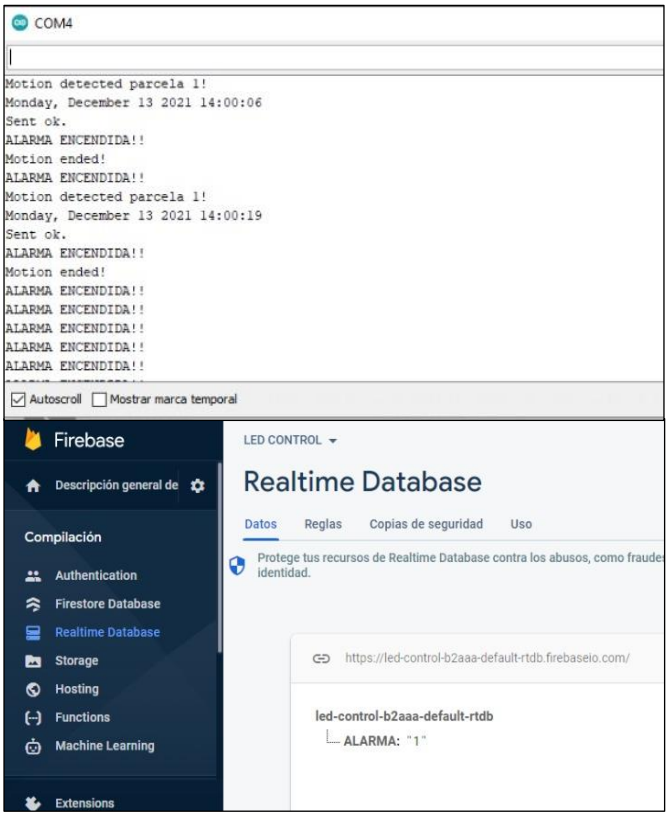


Figura 3.67 Sensor activado – Estado de alarma encendido

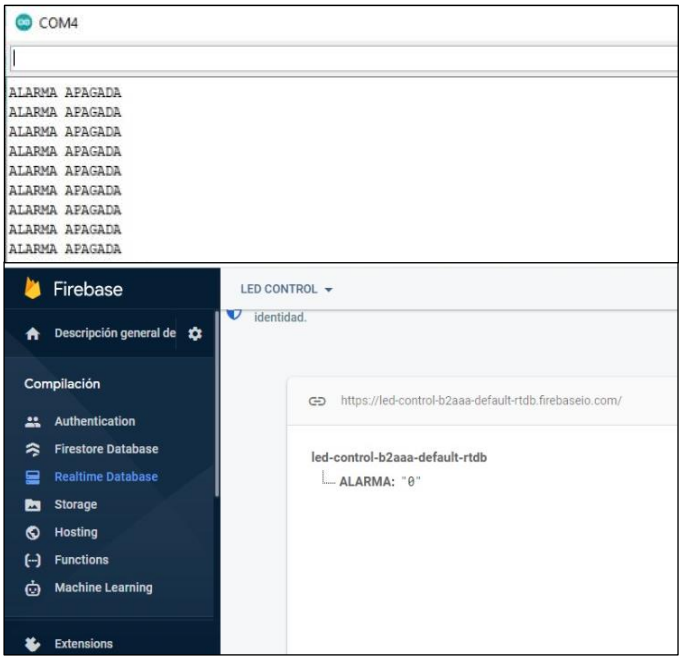


Figura 3.68 Sensor inactivo – Estado de alarma apagado

Una vez activado algún sensor se envía la notificación a WhatsApp como se muestra en la Figura 3.69. El usuario puede acceder a la aplicación

desarrollada en App inventor tal cual se muestra en la Figura 3.70, para validar el ingreso de persona no autorizada y proceder activar la alarma de forma remota.

Figura 3.69 Alerta mediante notificación por WhatsApp

Figura 3.70 App de monitoreo de cámaras y control de alarma

CAPÍTULO 4

4. RESULTADOS Y ANÁLISIS

En este capítulo se detallarán los resultados obtenidos del presente proyecto previo a la obtención del título. En el cual se mostrarán los procedimientos de manera ilustrativa referente a la etapa de reconocimiento facial y videovigilancia, tablas de resultados, gráficos estadísticos y resultados finales de la simulación.

4.1 Etapa Reconocimiento Facial

4.1.1 Proceso de registro

Para poder tener acceso en la industria se debe empezar con el proceso de registro de usuario nuevo, para esto es importante colocar el nombre correctamente y la cédula con sus 10 dígitos ya que en caso de que no sea de esa forma, se mostrará un mensaje por pantalla de advertencia como se muestra en la Figura 4.1.

Figura 4.1 Ingreso incompleto de registro

Una vez que el usuario haya ingresado correctamente su nombre y cédula como se puede observar en la Figura 4.2. se abrirá una ventana la cual permitirá captar los rasgos faciales del usuario como se visualiza en la Figura 4.3 para posteriormente realizar el entrenamiento respectivo y almacenar el modelo en rostro en un archivo y finalmente una vez concluido el proceso como se puede

visualizar en la Figura 4.4, mostrará un mensaje de !!Usuario registrado correctamente!!

Figura 4.2 Registro de Usuario

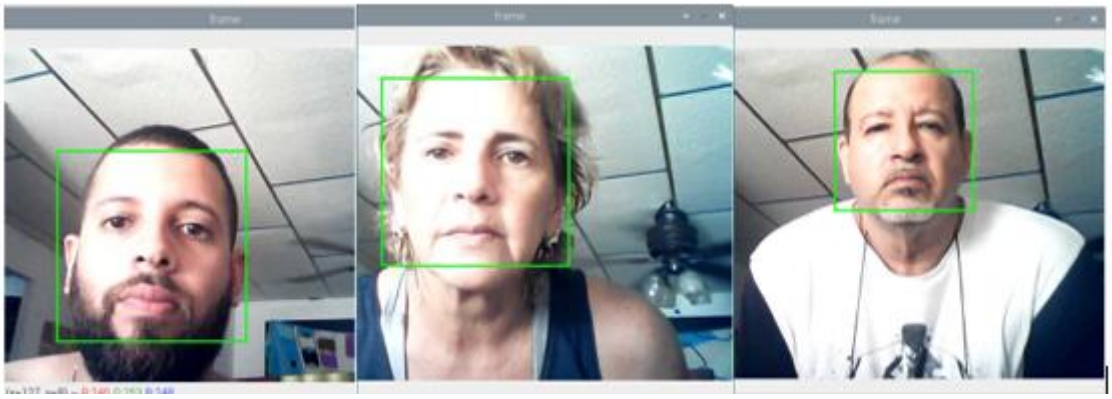


Figura 4.3 Captura de Rostro

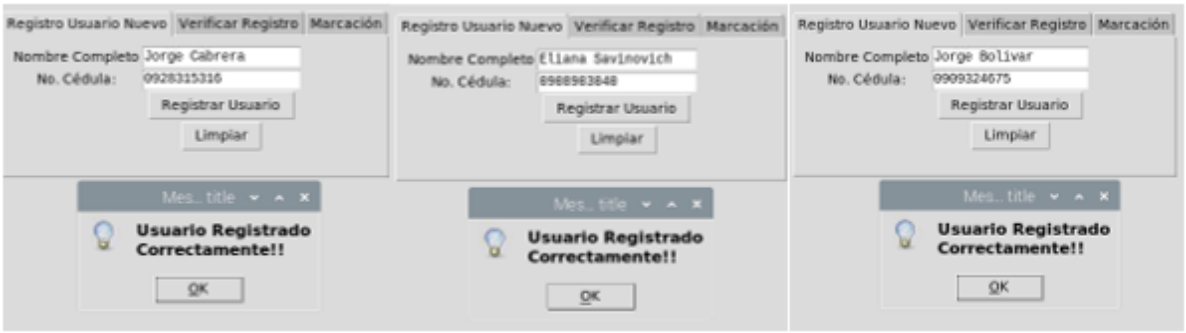
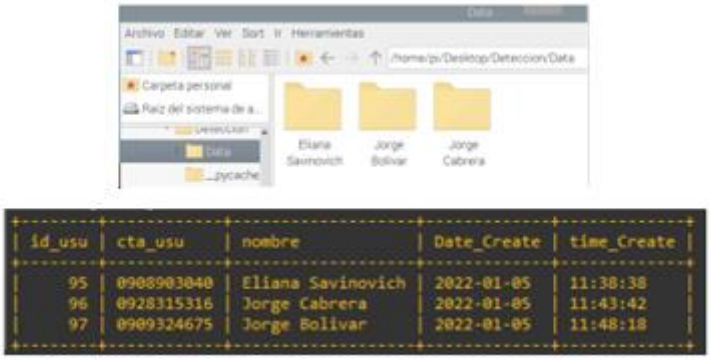


Figura 4.4 Registro exitoso del usuario

Una vez hecho el proceso de registro los datos del usuario de almacenarán en una base de Datos con la fecha y hora actual en el instante que realizo en proceso respectivo de registro. Además, Se puede observar en la Figura 4.5 que las imágenes del rostro se almacenan en un directorio para el respectivo proceso de reconocimiento.



id_usu	cta_usu	nombre	Date_Create	time_Create
95	0908903040	Eliana Savinovich	2022-01-05	11:38:38
96	0928315316	Jorge Cabrera	2022-01-05	11:43:42
97	0909324675	Jorge Bolivar	2022-01-05	11:48:18

Figura 4.5 Almacenamiento en directorio y base de datos

El sistema está diseñado para que al momento de usar algún tapabocas o material que impida visualizar el rostro en su totalidad, no permita capturar las características del mismo usuario y deniegue el proceso de registro y entrenamiento como se puede visualizar en la Figura 4.6, no se muestra el marco de color verde debido a que el usuario está portando un tapabocas.

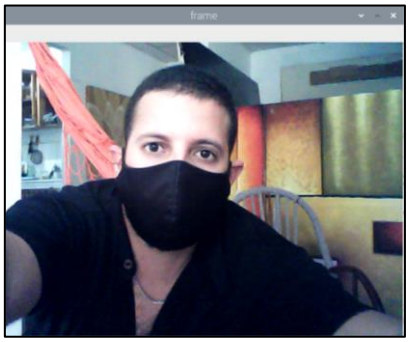


Figura 4.6 Verificación de captura de rostro usando mascarilla

La nitidez de la imagen de captura dependerá exclusivamente de la distancia a la que se encuentre el usuario al momento de realizar el proceso de registro.

Figura 4.7 Calidad de imagen vs distancia.

Como se observa en la Figura 4.7 la primera imagen esta capturada a una distancia de 5 metros, la segunda imagen esta capturada a una distancia de 3

metros, la tercera a una distancia de 2 metros y la cuarta imagen a una distancia de 0.5 metros. A mayor distancia menor calidad y a menor distancia de aproximación al a cámara mejor calidad de imagen.

4.1.2 Proceso de Verificación

Esta pestaña permite comprobar si el usuario esta registrado en el sistema en caso de que no recuerde si realizo el proceso de registro, así evita pasar por la pestaña uno y con tan solo escribir su nombre el sistema le mostrará un mensaje informando que ya se encuentra registrado como se observa en la Figura 4.8, o no está registrado como se visualiza en la Figura 4.9.

Figura 4.8 Comprobación de registro en pestaña de verificación

Figura 4.9 Comprobación de usuario no registrado

4.1.3 Proceso de Marcación y Apertura de puerta

Para el proceso final de Acceso, el sistema reconocere el rostro mostrado frente a la cámara y se podrá visualizar el respectivo nombre de cada usuario, en este caso se puede observar a Jorge Cabrera, Eliana Savinovich y Jorge Bolívar que están registrados correctamente como se muestra en las Figuras

4.10, 4.11y 4.12 y podrán acceder a la industria posteriormente aperturando la puerta tal cual la Figura 4.13 para el respectivo ingreso.

Figura 4.10 Acceso correcto de Usuario 1

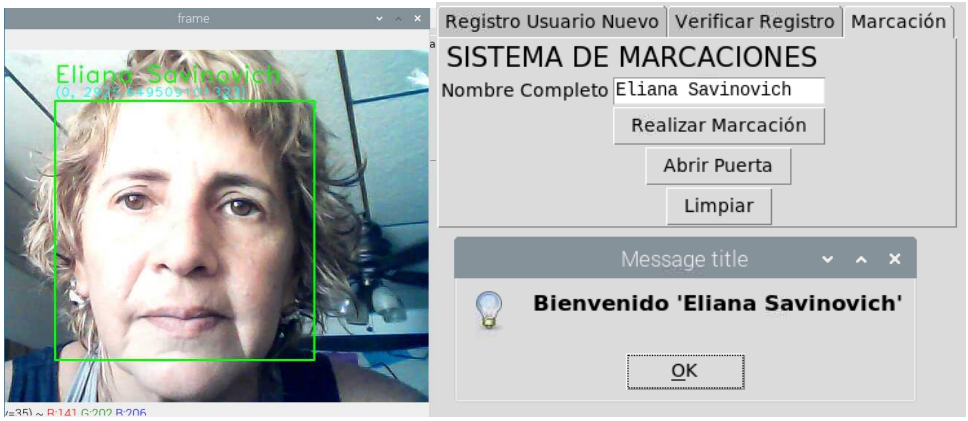


Figura 4.11 Acceso correcto de Usuario 2

Figura 4.12 Acceso correcto de Usuario 3

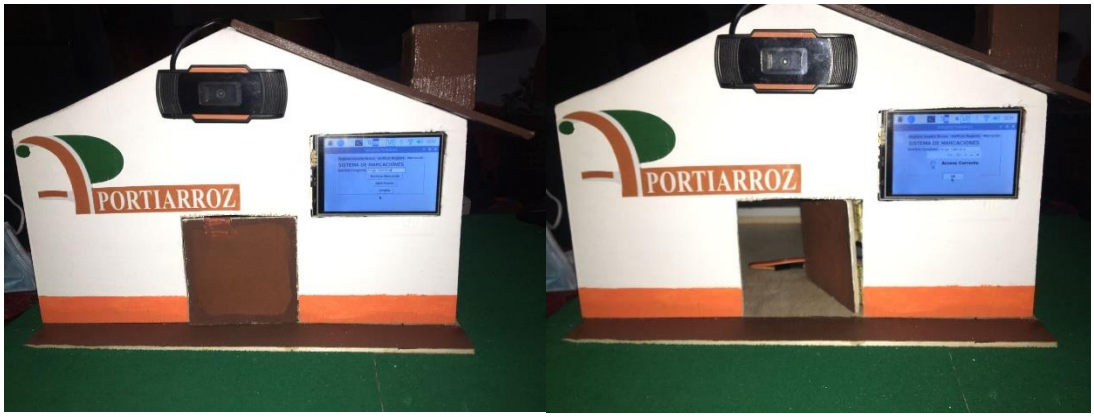


Figura 4.13 Apertura de puerta para el Ingreso

4.2 Etapa Monitoreo de Videovigilancia y Notificación de Alerta

En esta etapa se presentan las pruebas y resultados respecto al monitoreo de videovigilancia y notificación de alerta de seguridad.

4.2.1 Pruebas de Monitoreo y Grabación del sistema

Para esta etapa se cumple con el objetivo de la simulación de videovigilancia a través de cámaras con el propósito de monitorear el ingreso de personas externas al área de cultivo en la industria.

Figura 4.14 Diseño del sistema de monitoreo de Videovigilancia

En la Figura 4.14 se puede evidenciar el diseño del sistema en la industria con la distribución de las cámaras externas ubicadas de forma simulada en las parcelas en donde se realizará la inspección de los cultivos. De igual forma, se muestra en la Figura 4.15 las conexiones internas del componente principal Raspberry Pi con las cámaras. Se logra el monitoreo de las cámaras como se muestra en la Figura 4.16 previamente realizado el proceso de configuración descrito de forma detallada en el punto 4.6.

Figura 4.15 Conexiones internas del sistema de Videovigilancia

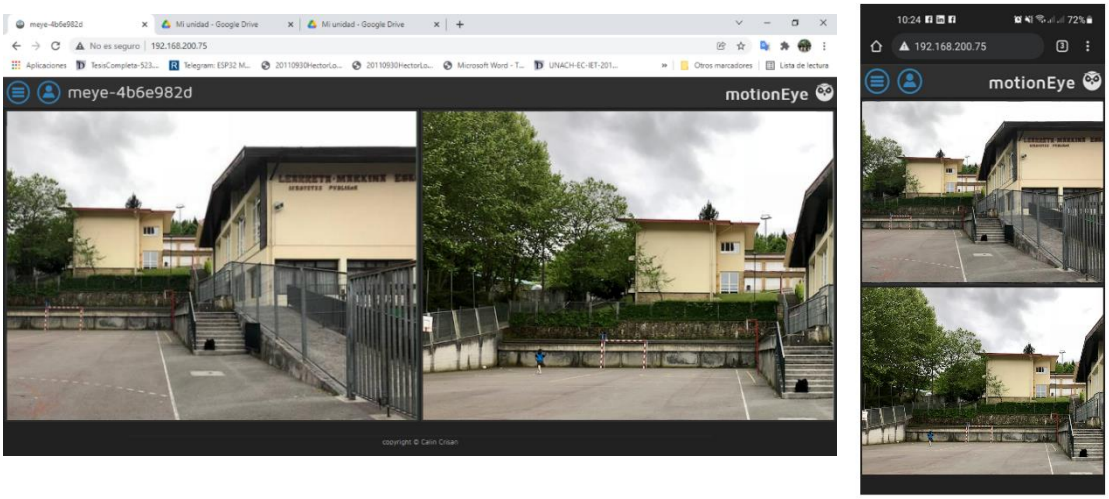


Figura 4.16 Simulación de monitoreo de cámaras

Por otro lado, el sistema garantizó la grabación y almacenamiento de videos en el servidor de datos Google Drive; estos corresponden a cada cámara instalada como se muestra en la Figura 4.17. La grabación puede ser configurada automáticamente ya sea mantener la grabación de forma constante días u horarios específicos mediante la configuración realizada en el punto 4.6.6. Los videos puedes ser vistos desde cualquier dispositivo en el cual esté enlazado el correo de Google principal.

Cabe mencionar que el servidor Google Drive facilita 15 Gigabits (GB) de almacenamiento gratuito, por lo cual es posible adquirir más Gigas de forma monetaria.

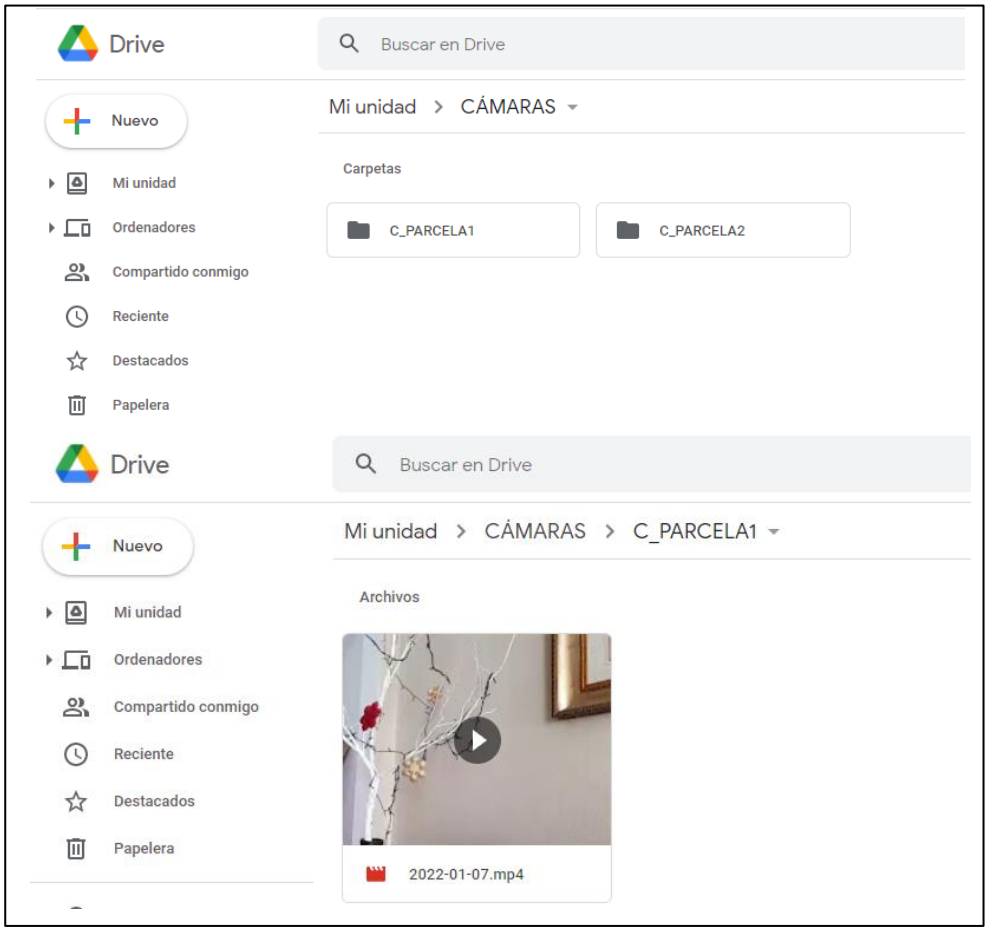


Figura 4.17 Grabación de Cámaras en Google Drive

4.2.2 Alcances y Restricciones

Alcances

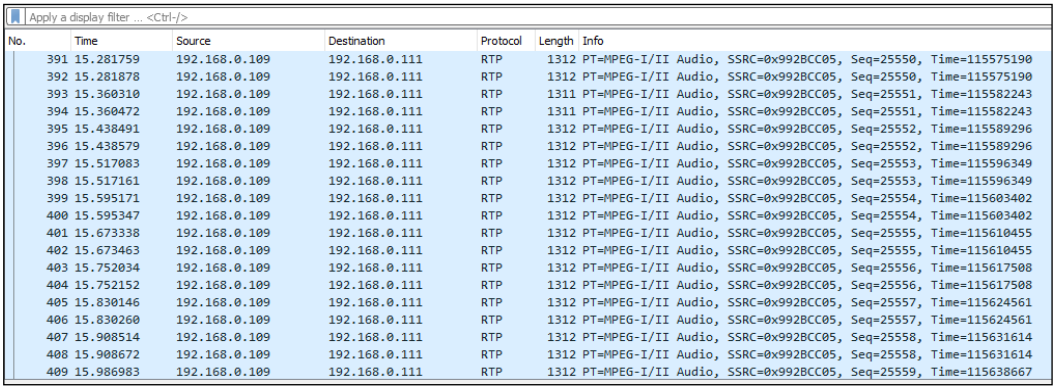
- El sistema puede implementarse en cualquier lugar en donde se requiera videovigilancia.
- La plataforma de monitoreo MotionEye es de fácil uso y permite el monitoreo constante y rápido de cámaras desde cualquier dispositivo digital electrónico.
- Utiliza un almacenamiento en la nube lo que implica menos costo en su implementación ya que no se requiere la adquisición de dispositivos videograbadores.

Restricciones

- Dado que el sistema tiene como módulo principal una Raspberry Pi, solo puede ser instalado un total de 16 cámaras para que no haya una sobrecarga en el microprocesador.

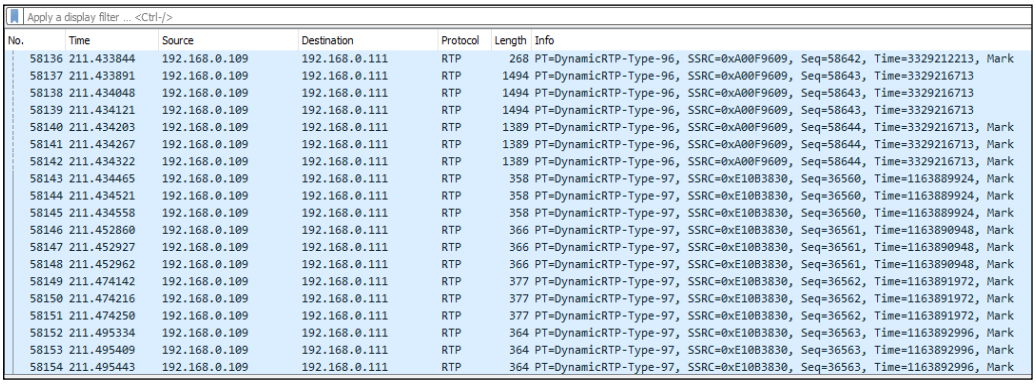
4.2.3 Transmisión de paquetes

Mediante el programa Wireshark se puede evidenciar que cada cámara configurada en el Raspberry Pi que conforman el sistema de videovigilancia transmite los paquetes de protocolo de transporte en tiempo real (RTP) del envío de audio y video como se muestra en las Figuras 4.18 y 4.19 respectivamente, lo cual se logra una transmisión constante.


Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
391	15.281759	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25550, Time=115575190
392	15.281878	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25550, Time=115575190
393	15.360310	192.168.0.109	192.168.0.111	RTP	1311	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25551, Time=115582243
394	15.360472	192.168.0.109	192.168.0.111	RTP	1311	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25551, Time=115582243
395	15.438491	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25552, Time=115589296
396	15.438579	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25552, Time=115589296
397	15.517083	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25553, Time=115596349
398	15.517161	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25553, Time=115596349
399	15.595171	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25554, Time=115603402
400	15.595347	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25554, Time=115603402
401	15.673338	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25555, Time=115610455
402	15.673463	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25555, Time=115610455
403	15.752034	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25556, Time=115617508
404	15.752152	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25556, Time=115617508
405	15.830146	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25557, Time=115624561
406	15.830260	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25557, Time=115624561
407	15.908514	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25558, Time=115631614
408	15.908672	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25558, Time=115631614
409	15.986983	192.168.0.109	192.168.0.111	RTP	1312	PT=MPEG-I/II Audio, SSRC=0x992BCC05, Seq=25559, Time=115638667

Figura 4.18 Transmisión de paquetes RTP de audio de cámaras



Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
58136	211.433844	192.168.0.109	192.168.0.111	RTP	268	PT=DynamicRTP-Type-96, SSRC=0xA00F9609, Seq=58642, Time=3329212213, Mark
58137	211.433891	192.168.0.109	192.168.0.111	RTP	1494	PT=DynamicRTP-Type-96, SSRC=0xA00F9609, Seq=58643, Time=3329216713
58138	211.434048	192.168.0.109	192.168.0.111	RTP	1494	PT=DynamicRTP-Type-96, SSRC=0xA00F9609, Seq=58643, Time=3329216713
58139	211.434121	192.168.0.109	192.168.0.111	RTP	1494	PT=DynamicRTP-Type-96, SSRC=0xA00F9609, Seq=58643, Time=3329216713
58140	211.434203	192.168.0.109	192.168.0.111	RTP	1389	PT=DynamicRTP-Type-96, SSRC=0xA00F9609, Seq=58644, Time=3329216713, Mark
58141	211.434267	192.168.0.109	192.168.0.111	RTP	1389	PT=DynamicRTP-Type-96, SSRC=0xA00F9609, Seq=58644, Time=3329216713, Mark
58142	211.434322	192.168.0.109	192.168.0.111	RTP	1389	PT=DynamicRTP-Type-96, SSRC=0xA00F9609, Seq=58644, Time=3329216713, Mark
58143	211.434465	192.168.0.109	192.168.0.111	RTP	358	PT=DynamicRTP-Type-97, SSRC=0xE10B3830, Seq=36560, Time=1163889924, Mark
58144	211.434521	192.168.0.109	192.168.0.111	RTP	358	PT=DynamicRTP-Type-97, SSRC=0xE10B3830, Seq=36560, Time=1163889924, Mark
58145	211.434558	192.168.0.109	192.168.0.111	RTP	358	PT=DynamicRTP-Type-97, SSRC=0xE10B3830, Seq=36560, Time=1163889924, Mark
58146	211.452860	192.168.0.109	192.168.0.111	RTP	366	PT=DynamicRTP-Type-97, SSRC=0xE10B3830, Seq=36561, Time=1163890948, Mark
58147	211.452927	192.168.0.109	192.168.0.111	RTP	366	PT=DynamicRTP-Type-97, SSRC=0xE10B3830, Seq=36561, Time=1163890948, Mark
58148	211.452962	192.168.0.109	192.168.0.111	RTP	366	PT=DynamicRTP-Type-97, SSRC=0xE10B3830, Seq=36561, Time=1163890948, Mark
58149	211.474142	192.168.0.109	192.168.0.111	RTP	377	PT=DynamicRTP-Type-97, SSRC=0xE10B3830, Seq=36562, Time=1163891972, Mark
58150	211.474216	192.168.0.109	192.168.0.111	RTP	377	PT=DynamicRTP-Type-97, SSRC=0xE10B3830, Seq=36562, Time=1163891972, Mark
58151	211.474250	192.168.0.109	192.168.0.111	RTP	377	PT=DynamicRTP-Type-97, SSRC=0xE10B3830, Seq=36562, Time=1163891972, Mark
58152	211.495334	192.168.0.109	192.168.0.111	RTP	364	PT=DynamicRTP-Type-97, SSRC=0xE10B3830, Seq=36563, Time=1163892996, Mark
58153	211.495409	192.168.0.109	192.168.0.111	RTP	364	PT=DynamicRTP-Type-97, SSRC=0xE10B3830, Seq=36563, Time=1163892996, Mark
58154	211.495443	192.168.0.109	192.168.0.111	RTP	364	PT=DynamicRTP-Type-97, SSRC=0xE10B3830, Seq=36563, Time=1163892996, Mark

Figura 4.19 Transmisión de paquetes RTP de videos de cámaras

Se obtiene de forma gráfica la cantidad de paquetes RTP (audio y video) versus tiempo que se transmiten desde las cámaras hasta el dispositivo de monitoreo mediante la IP correspondiente. En la Figura 4.20 se muestra que la cantidad de paquetes de audio resulta menor a la cantidad de paquetes que se transmiten en video evidenciado en la Figura 4.21, por lo cual se precisa que el ancho de banda que ocupa la retrasmisión de video es mayor; es decir, se requiere de un aumento de ancho de banda al conectar más cámaras en el sistema.

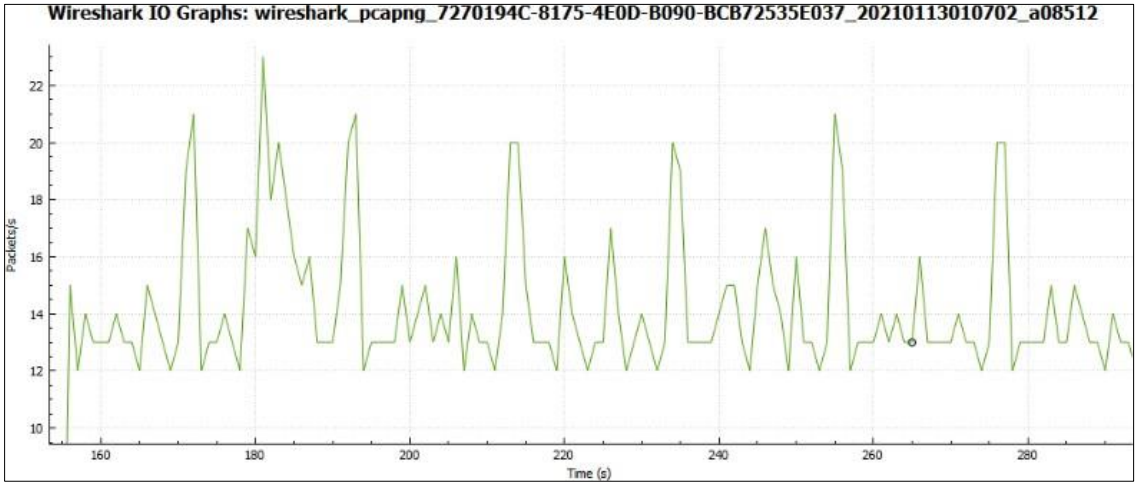


Figura 4.20 Gráfica de paquetes RTP de audio vs tiempo

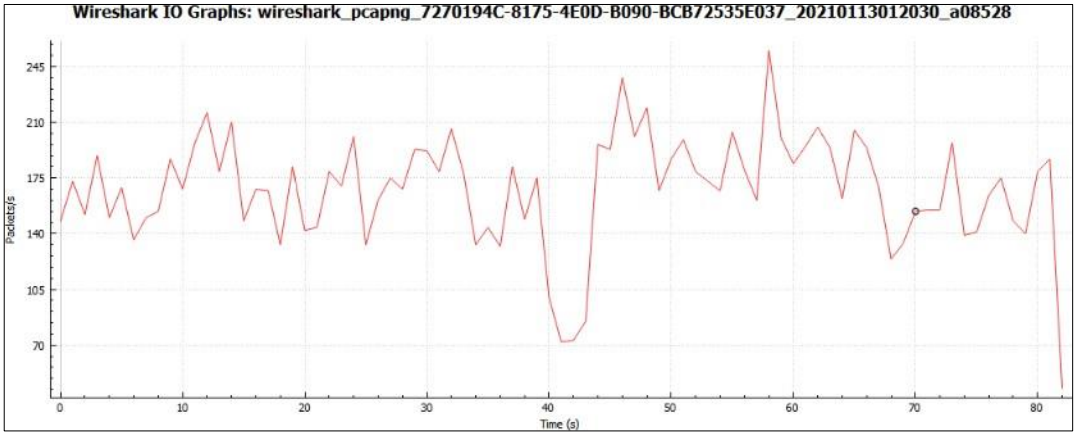


Figura 4.21 Gráfica de paquetes RTP de video vs tiempo

4.2.4 Velocidad de internet

Dentro de las configuraciones realizadas en el punto 4.6 se debe tomar en cuenta la velocidad de internet para la transmisión de audio y video de cámaras. Cada cámara ocupa una velocidad aproximada de 3 Megabits por segundo (Mbps); esta capacidad máxima de transmisión de datos resulta adecuada y eficiente considerando varios factores como la resolución y velocidad de imagen, así como las horas de funcionamiento de las cámaras. Por lo tanto, se precisa en la tabla 4.1 la cantidad de velocidad de internet que se debe tener en la red, adecuada a la cantidad de cámaras a utilizar.

# Cámaras a utilizar en el sistema	Velocidad asignada (Mbps)	Velocidad requerida (Mbps)
2	6	10
4	12	20
6	18	20
8	24	30
10	30	35
12	36	40
14	42	45
16	48	55

Tabla 4.1 Resultado del Ancho de banda requerido de acuerdo con la cantidad de cámaras

.

4.2.5 Pruebas de Notificación de alerta

El sistema general de seguridad cuenta con un proceso de notificación de alerta, el cual se implementa en la simulación usando el módulo ESP32 y sensores infrarrojos PIR. La conexión de los componentes se encuentra realizada en el diagrama esquemático de la Figura 4.48.

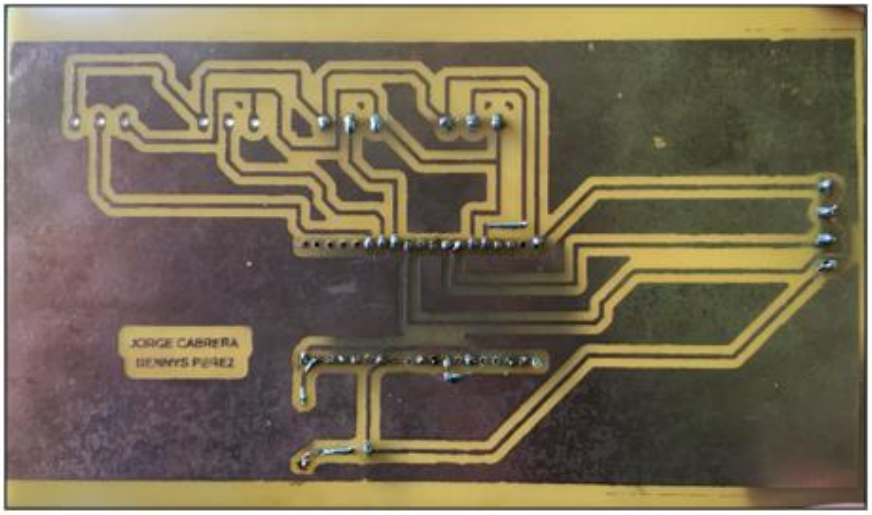


Figura 4.22 Placa del circuito de notificación de alerta impreso

Dentro del implementación del circuito, se realizó la placa electrónica de circuito impreso mostrado en la Figura 4.22, permitiendo el soporte y conectividad de los componentes soldados en la PCB de la Figura 4.23, los cuales son: ESP32, sensores PIR, resistencia, led y una alarma

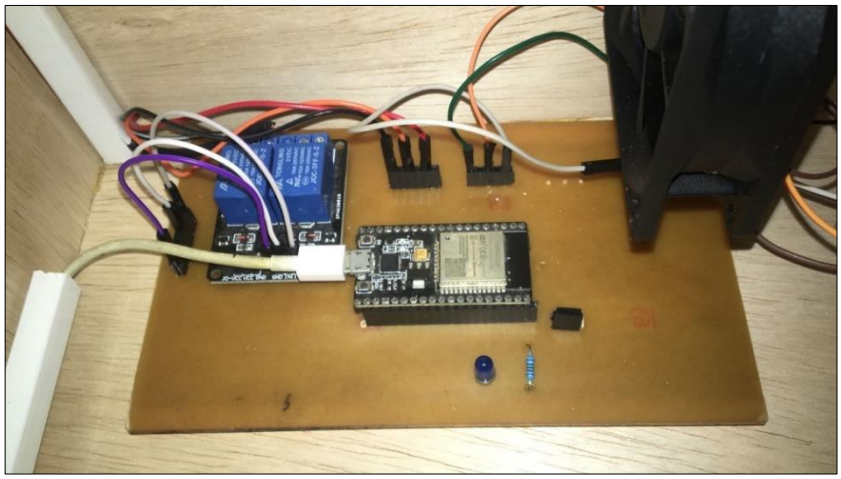


Figura 4.23 Conexiones internas del sistema de Notificación de Alerta

El proceso de notificación empieza ejecutando el algoritmo desarrollado que se detalla en el punto 4.6.10. Al activar los sensores el microprocesador ESP32 recibe el aviso; este mediante el algoritmo procede a enviar la notificación de alerta a los teléfonos celulares registrados con el apikey a través de la mensajería de WhatsApp. Dentro de la programación se hizo referencia en la simulación la ubicación de los sensores en las parcelas, por lo cual al activarse la notificación de alerta refleja el aviso de “Movimiento Detectado en la Parcela 1”; el mensaje varía dependiendo del sensor activado como se muestra en la Figura 4.24.

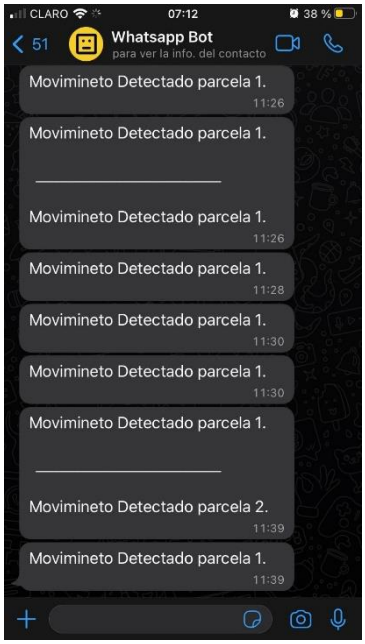


Figura 4.24 Envío de notificación de alerta a WhatsApp

4.2.6 Proceso de activación de alarma

Dentro del sistema se encuentra incorporada una alarma, la cual da aviso de alerta a los trabajadores para que asistan al área de cultivo en donde se presentan sucesos de robo. Esto se desarrolló dentro del algoritmo ejecutado para esta estaba del sistema, en la cual, al activar los sensores, el módulo ESP32 mantiene una conectividad con una base de datos de red Firebase en donde el encendido y apagado de la alarma se enlaza a un aplicativo móvil creado en MIT App Invertor. El resultado de estado de alarma se muestra en la

Figura 4.25 y 4.26 la cual toma un valor de 0 en estado apagado y un valor de 1 cuando esté encendida.

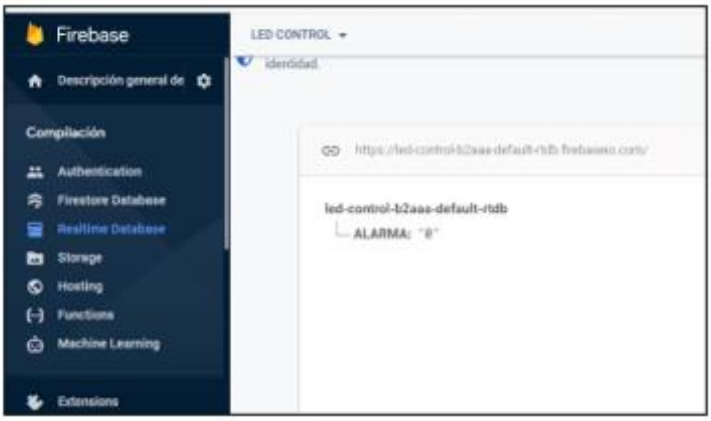


Figura 4.25 Estado apagado de alarma en Firebase

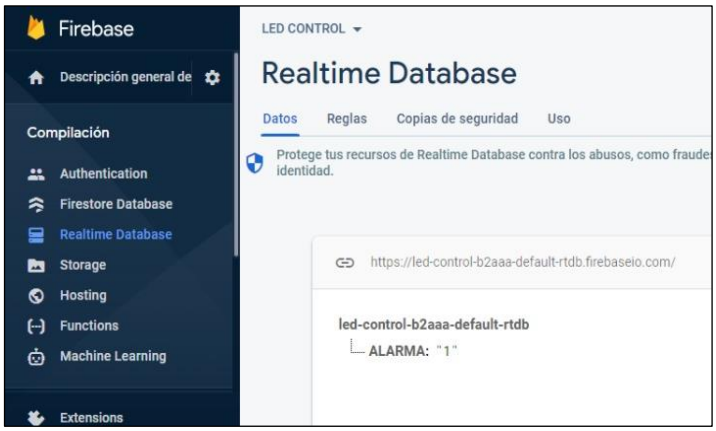


Figura 4.26 Estado Encendido de alarma en Firebase

4.2.7 Aplicativo móvil de monitoreo y control

Para mantener una conectividad entre el monitoreo de cámaras y activación de alarma, se implementó un aplicativo móvil creado en MIT App Inventor, el cual está desarrollado a través de programación por bloques. Mediante esta App, es posible enlazar las cámaras de videovigilancia usando la dirección IP y controlar el estado de alarma en caso de haber confirmado que una persona no autorizada sustrae algún cultivo. Este aplicativo está diseñado para ser instalado en los dispositivos Android, a través de un archivo apk. En la Figura 4.27 se muestra el aplicativo en funcionamiento y los resultados de activación y desactivación de alarma.

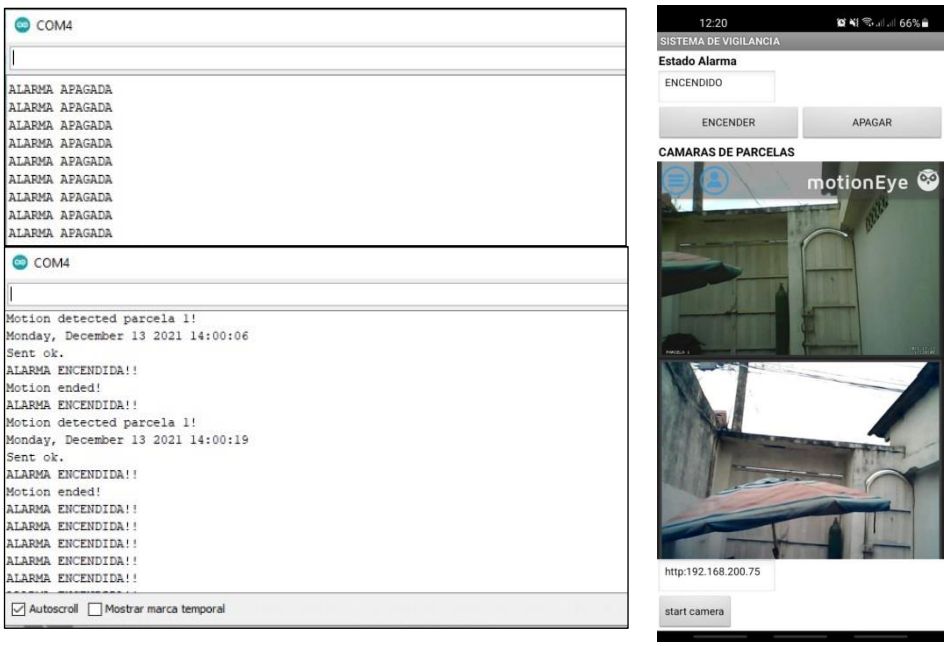


Figura 4.27 Aplicativo móvil de monitoreo de cámaras y control de alarma

4.2.8 Tiempo de respuesta de notificación

Luego de la simulación de esta parte de sistema se realizaron varias pruebas que se enfocan en el tiempo de respuesta que se obtiene al activarse los sensores y del envió de notificaciones a WhatsApp por el microprocesador ESP32 variando la cantidad de celulares registrados en el algoritmo.

Se realizaron variaciones en el algoritmo en donde se logró registrar 16 dispositivos para la recepción de notificación de alerta. Los resultados de la simulación se muestran en la Tabla 4.2 en donde se precisa que al aumentar la cantidad de dispositivos aumenta el tiempo de respuesta. Sin embargo, en base a los requerimientos del cliente, se precisa la utilización de 5 dispositivos registrados, por lo que se mantiene un tiempo de respuesta optimo menor a 15 segundos. En la Figura 4.28 se muestran de manera gráfica los resultados obtenidos.

Cantidad de dispositivos.	Tiempo de respuesta (Segundos)	Cantidad de dispositivos.	Tiempo de respuesta (Segundos)
1	6.20	9	9.70
2	6.93	10	10.57
3	7.53	11	10.81
4	7.88	12	11.00
5	8.24	13	11.95
6	8.73	14	12.56
7	8.92	15	13.66
8	9.17	16	14.40

Tabla 4.2 Resultados de tiempo de respuesta vs cantidad de dispositivos

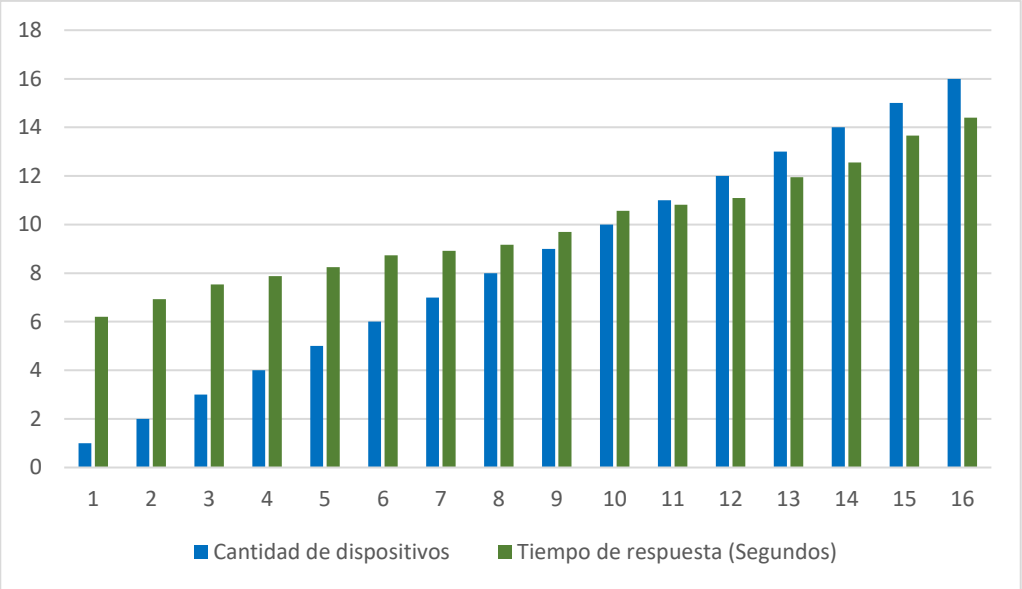


Figura 4.28 Resultados gráficos de tiempo de respuesta vs cantidad de dispositivos

CONCLUSIONES

Posterior a las pruebas realizadas del sistema en general, tanto para la etapa de reconocimiento facial como de monitoreo de videovigilancia, se puede concluir que:

- Se desarrolló un sistema de acceso que permite el registro e ingreso del usuario mediante reconocimiento facial mejorando el tiempo de verificación y validación tradicional en un 50% permitiendo la apertura sistemática de la puerta de ingreso en caso de que el sistema lo apruebe sin necesidad de que el usuario realice la acción de empujar la puerta. El sistema valida si el usuario se encuentra registrado en la base de datos, Luego aprueba el ingreso accionando un motor para su correcta activación lo cual resulta eficiente y rentable.

- El sistema elaborado para la detección y monitoreo de cultivos mostró resultados favorables en su comunicación mediante Wifi permitiendo un envió de notificación en un tiempo mínimo de 6.20 segundos, volviéndolo un sistema seguro y confiable ya que al no poseer un método de seguridad están propensos a robos. Además, el sistema presentó adaptabilidad al conectar más cámaras y sensores a conveniencia, lo que proporciona una mayor escalabilidad en su funcionamiento.

- Se determinó las zonas y lugares estratégicos donde se colocaron los dispositivos, sensores y cámaras permitiendo abarcar las áreas en su totalidad evitando tener zonas sin ser detectadas para que el encargado de supervisar y monitorear tenga facilidad de manipular y tomar acciones rápidas en caso de observar algún problema o suceso indebido.

- Se logró diseñar una aplicación móvil en donde se pudo visualizar las cámaras en tiempo real para el monitoreo de cultivos e ingreso de personas no autorizadas dándole al personal encargado la factibilidad de monitorear en cualquier lugar, dentro o fuera de la red los sucesos de robo para posteriormente mediante un botón activar la alarma y evitar pérdidas.

- El uso del sistema en general permite darle facilidad de monitoreo y cuidado a los cultivos y empresa como tal con bajos porcentajes de consumo eléctrico y a través de comunicación inalámbrica vía Wifi se obtiene de forma rápida la alerta esperada.

RECOMENDACIONES

En base a las pruebas realizadas y los resultados obtenidos, se argumentan indicaciones para el buen funcionamiento del sistema:

- El sistema de videovigilancia simulado al mantener una transmisión de datos constantes entre sus componentes y dispositivos requiere ser energizado eléctricamente todo el tiempo, por ende, es recomendable ubicar la central en un lugar en donde no se presente altas temperaturas y humedad.

- El internet es un medio de comunicación fundamental, por ende, es recomendable y de suma importancia utilizar una conexión rápida, eficiente y segura de unos 20 Mbps mínimo para evitar errores en el sistema o retardo en la comunicación entre dispositivos.

- Mantener actualizada la plataforma de monitoreo de cámaras MotionEye en caso de que se presenten actualizaciones, de tal manera que opere de manera eficiente y no presente algún problema en el rendimiento.

- Se sugiere activar la configuración de limpieza de grabaciones en la plataforma para dos días, de forma que automáticamente se elimine las grabaciones de video innecesarias que no requieren revisión y así ahorrar almacenamiento en la nube.

BIBLIOGRAFÍA

[1] «Internacional,» 19 Octubre 2021. [En línea]. Available: https://elpais.com/internacional/2021-10-20/ecuador-el-pais-donde-las-balas-no-distinguen-barrios-ni-horarios.html. [Último acceso: 1 Noviembre 2021].

[2] R. Velasco, «Hardzone,» 24 Agosto 2021. [En línea]. Available: https://hardzone.es/reviews/perifericos/analisis-raspberry-pi-3-modelo-b/. [Último acceso: 1 Noviembre 2021].

[3] A. Electronic, «AV Electronic,» 2021. [En línea]. Available: https://avelectronics.cc/producto/tarjeta-de-desarrollo-esp32-wifi-bluetooth/. [Último acceso: 1 Noviembre 2021].

[4] P. Orta Jarrín, «Diseño e implementación de un sistema de seguridad de video vigilancia mediante camaras IP Bajo administración SNMP, utilizando una alarma GPRS,» 2013. [En línea]. Available: http://dspace.unach.edu.ec/bitstream/51000/716/1/UNACH-EC-IET-2013-0009.pdf. [Último acceso: 13 Octubre 2021].

[5] Fiscalía General Del Estado, «Estadísticas FGE,» 8 Septiembre 2021. [En línea]. Available: https://www.fiscalia.gob.ec/estadisticas-de-robos/.

[6] A. B. Rosero, «El comercio,» 30 Noviembre 2020. [En línea]. Available: https://www.elcomercio.com/actualidad/seguridad/robos-locales-comerciales-violencia-ecuador.html. . [Último acceso: 14 Octubre 2021].

[7] Oficina Internacional del trabajo, [En línea]. Available: https://www.ilo.org/wcmsp5/groups/public/---ed_protect/---protrav/---safework/documents/publication/wcms_117460.pdf. . [Último acceso: 10 Octubre 2021].

[8] Narvis, «Portiarroz,» 2015. [En línea]. Available: https://portiarroz.com.ec/. [Último acceso: 31 octubre 2021].

[9] P. Torres Osorio y L. Paternina Alvarez, «Diseño e implementación de un sistema de seguridad perimetral para la red de comercializadora internacional oceanos S.A,» 2011. [En línea]. Available: https://repositorio.utb.edu.co/bitstream/handle/20.500.12585/2124/0062307.pdf?sequence=1&isAllowed=y.%20[%C3%9Altimo%20acceso:%2015%20Octubre%202021].. [Último acceso: 17 Octubre 2021].

[10] D. L. C. Toro, «bibdgital,» 11 05 2015. [En línea]. Available: https://bibdigital.epn.edu.ec/bitstream/15000/10770/1/CD-6313.pdf. [Último

acceso: 11 10 2021].

[11] J. D. Fonseca, «Universidad Politécnica Salesiana,» 10 agosto 2020. [En línea]. Available: https://dspace.ups.edu.ec/bitstream/123456789/18986/1/UPS%20-%20TTS056.pdf. [Último acceso: 12 octubre 2021].

[12] M.-O. Schwartz, «Adafruit,» Anne Barela, 31 marzo 2014. [En línea]. Available: https://learn.adafruit.com/wireless-security-camera-arduino-yun?view=all. [Último acceso: 12 octubre 2021].

[13] V. M. R. Marcos, «Universidad Carlos III de Madrid,» Junio 2017. [En línea]. Available: https://core.ac.uk/download/pdf/288499822.pdf. [Último acceso: 12 Octubre 2021].

[14] Scielo, [En línea]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202019000500176. [Último acceso: 10 Octubre 2021].

[15] sociedad, «EL INDEPENDIENTE,» 28 Abril 2021. [En línea]. Available: https://www.elindependiente.com/sociedad/2021/04/28/madrid-la-tercera-gran-ciudad-europea-con-menos-homicidios/. [Último acceso: 20 Octubre 2021].

[16] A. T. Escrihuela, «etsinf,» 24 Septiembre 2015. [En línea]. Available: https://riunet.upv.es/bitstream/handle/10251/56039/Memoria.pdf?sequence=1. [Último acceso: 12 Octubre 2021].

[17] D. Castaño Saavedra y J. Alonso Sierra, «Universidad Católica de Colombia,» 2019. [En línea]. Available: https://repository.ucatolica.edu.co/bitstream/10983/24032/1/Final%20Trabajo%20de%20grado.pdf. [Último acceso: Octubre 2021].

[18] J. Muñoz, «espolec-sharepoint,» 2019. [En línea]. Available: https://espolec-my.sharepoint.com/personal/administracion-fiec_espol_edu_ec/_layouts/15/onedrive.aspx?id=%2Fpersonal%2Fadministracion%2Dfiec%5Fespol%5Fedu%5Fec%2FDocuments%2FRepositorioSubdecanato%2F2020%2FINTEGRADORA%20IIT%2D2019%2FTELECOMUNICACIONES%2FM. [Último acceso: 20 Octubre 2021].

[19] Pena, «Random Nerd,» 18 Diciembre 2019. [En línea]. Available: https://randomnerdtutorials.com/esp32-relay-module-ac-web-server/. [Último acceso: 20 Octubre 2021].

[20] G. Márquez, «Universidad Tecnica de Ambato,» 2016. [En línea]. Available: https://repositorio.uta.edu.ec/bitstream/123456789/23065/1/Tesis_t1117ec.pdf. [Último acceso: 2021].

[21] J. S. Espinoza, «repositorio ESPE,» 5 Marzo 2018. [En línea]. Available: http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/14065/T-ESPE-057625.pdf?sequence=1&isAllowed=y. [Último acceso: 12 Octubre 2021].

[22] Comisión económica para América Latina (CEPAL), 2015. [En línea]. Available: https://archivo.cepal.org/pdfs/GuiaProspectiva/Alvarez2015Implementacion.pdf. [Último acceso: 10 Octubre 2021].

[23] A. d. Aviles Salazar y K. L. Cobeña Mite, «Universidad Politécnica Salesiana,» Febrero 201. [En línea]. Available: https://dspace.ups.edu.ec/bitstream/123456789/10401/1/UPS-GT001444.pdf.

[24] P. Orta Jarrín y D. Santillan , «Universidad de Chimborazo,» 2013. [En línea]. Available: http://dspace.unach.edu.ec/bitstream/51000/716/1/UNACH-EC-IET-2013-0009.pdf.

[25] R. F. Ayala Sandoval, «Universidad de las Fuerzas Armadas,» [En línea]. Available: http://repositorio.espe.edu.ec/bitstream/21000/8077/1/AC-EAC-ESPE-047724.pdf. [Último acceso: 8 Octubre 2021].

[26] C. Novillo Montoya, «Universidad de Guayaquil,» 2014. [En línea]. Available: http://repositorio.ug.edu.ec/bitstream/redug/6529/1/TesisCompleta-523.pdf. [Último acceso: 9 Octubre 2021].

[27] M. Cajas Idrovo y P. Viri Ávila, «2017,» [En línea]. Available: https://dspace.ups.edu.ec/bitstream/123456789/13566/1/UPS-CT006920.pdf. [Último acceso: Octubre 2021].

[28] R. Castro Arias, «Universidd Técnica de Ambato,» Febrero 2016. [En línea]. Available: https://repositorio.uta.edu.ec/bitstream/123456789/20347/1/Tesis_t1107ec.pdf. [Último acceso: Octubre 2021].

[29] «HETPRO,» 11 Agosto 2021. [En línea]. Available: https://hetpro-store.com/TUTORIALES/microcontrolador/. [Último acceso: 2 Diciembre 2021].

[30] L. Llamas, «Modelos y Características de Raspberry Pi,» 17 Noviembre 2017. [En línea]. Available: https://www.luisllamas.es/modelos-de-raspberry-pi/. [Último acceso: 20 Noviembre 2021].

[31] Espressif Systems, «microprocessor guide,» 2020. [En línea]. Available: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. [Último acceso: 18 Noviembre 2021].

[32] HZ hard, «Microcontrolador ESP8266,» Julio 2019. [En línea]. Available:

https://hardzone.es/reportajes/tema/esp8266-2n2222-arduino/. [Último acceso: 18 Noviembre 2021].

[33] TecMikro, «Microprocesador ESP8266,» 2020. [En línea]. Available: https://tecmikro.com/modulos-shields/598-modulo-wifi-esp8266-nodemcu.html. [Último acceso: 18 Niviembre 2021].

[34] J. Guerra Carmenate, «Microprocesador ESP32,» 2018. [En línea]. Available: https://programarfacil.com/esp8266/esp32/. [Último acceso: 19 Noviembre 2021].

[35] BricoGeek, Septiembre 2020. [En línea]. Available: https://tienda.bricogeek.com/arduino-compatibles/1274-esp32-wroom-wifi-bluetooth.html. [Último acceso: 19 Noviembre 2021].

[36] C. Novillo Montaya, «Implementación de sistema de seguridad con videocámaras,» 2014. [En línea]. Available: http://repositorio.ug.edu.ec/bitstream/redug/6529/1/TesisCompleta-523.pdf. [Último acceso: 20 Noviembre 2021].

[37] H. Kruegle, «Block diagram of analog cameras in CCTV,» 2010. [En línea]. Available: https://www.sciencedirect.com/book/9780750690287/cctv-surveillance. [Último acceso: 20 Noviembre 2021].

[38] «Ficha técnica Cámara Web MTQ HD,» [En línea]. Available: https://cdn.cyberpuerta.mx/storage/articles/multimedia/651312_ft.pdf. [Último acceso: 20 Noviembre 2021].

[39] [En línea]. Available: Cámara bullet SCO-6085R. [Último acceso: 1 Diciembre 2021].

[40] HIKVISION, [En línea]. Available: https://www.hikvision.com/es-la/products/HiLook-Turbo-HD-Product/Turbo-HD-Camera/1080P/THC-T120-P/. [Último acceso: 1 Diciembre 2021].

[41] CISCO, [En línea]. Available: https://www.cisco.com/c/dam/global/es_mx/solutions/small-business/centro_recursos/pdf/ip_camera_systems_for_more_than_just_security_spa.pdf. [Último acceso: 19 Noviembre 2021].

[42] Wanscam, «Componentes de cámaras IP,» 2018. [En línea]. Available: https://www.wanscam.es/wanscam-hw0024-camara-ip-wifi-interior-color-negra-alta-resolucion-hd-con-ranura-memoria-grabacion.html. [Último acceso: 20 Noviembre 2021].

[43] S. Martí, «Diseño de un sistema de televigilancia sobre IP,» 2013. [En línea].

Available: https://riunet.upv.es/bitstream/handle/10251/34082/memoria.pdf. [Último acceso: 20 Noviembre 2021].

[44] HIKVISION, [En línea]. Available: https://www.hikvision.com/es-la/products/IP-Products/PTZ-Cameras/. [Último acceso: 30 Noviembre 2021].

[45] TRENDnet, [En línea]. Available: https://www.trendnet.com/store/langsp/products/surveillance-camera/indoor-outdoor-2mp-1080p-hd-poe-ir-ptz-speed-dome-network-camera-TV-IP440PI. [Último acceso: 30 Noviembre 2021].

[46] R. P. Fundation, «Rasperry Pi,» 2012. [En línea]. Available: https://www.raspberrypi.com/documentation/accessories/camera.html. [Último acceso: 20 Noviembre 2021].

[47] TostaTronic, «Pantalla TFT Touch,» Guadalajara, 2021.

[48] R. ECUADOR, «Mercado Libre,» 2019. [En línea]. Available: https://articulo.mercadolibre.com.ec/MEC-502954423-pantalla-tactil-touch-28-spi-240x320-arduino-tft-puntero-_JM. [Último acceso: 2 Diciembre 2021].

[49] m. Keyboard, «Mas Tecnologia PC,» 2020. [En línea]. Available: http://www.mastecnologiapc.com/shop/teclado-mini-keyboard/. [Último acceso: 18 Enero 2022].

[50] APPccesible, «CILSA,» 2017. [En línea]. Available: https://desarrollarinclusion.cilsa.org/tecnologia-inclusiva/que-es-un-sistema-operativo/. [Último acceso: 21 Noviembre 2021].

[51] D. G. IONOS, «Ubuntu Linux,» 11 Noviembre 2019. [En línea]. Available: https://www.ionos.es/digitalguide/servidores/know-how/ubuntu-un-sistema-para-todos-basado-en-linux/. [Último acceso: 21 Noviembre 2021].

[52] R. Adeva, «ADSL ZONE,» 11 Junio 2021. [En línea]. Available: https://www.adslzone.net/noticias/streaming-tv/condenado-crear-kodi-addon-supremacy/. [Último acceso: 2 Diciembre 2021].

[53] GCFGlobal, «macOs,» 1998. [En línea]. Available: https://edu.gcfglobal.org/es/curso-de-mac-os/que-es-macos/1/. [Último acceso: 03 Diciembre 2021].

[54] «RockContent,» 20 Abril 2019. [En línea]. Available: https://rockcontent.com/es/blog/que-es-un-lenguaje-de-programacion/. [Último acceso: 21 Noviembre 2021].

[55] A. Robledano, «OpenWebinars,» 23 Septiembre 2019. [En línea]. Available: https://openwebinars.net/blog/que-es-python/. [Último acceso: 21 Noviembre 2021].

[56] O. S. C. Vision, «OPENCV,» Google, 2013. [En línea]. Available: https://docs.opencv.org/4.x/d0/de3/tutorial_py_intro.html. [Último acceso: 21 Noviembre 2021].

[57] J. A. Rodrigo, «Reconocimiento Facial,» cienciadedatos.net, Mayo 2021. [En línea]. Available: https://www.cienciadedatos.net/documentos/py34-reconocimiento-facial-deeplearning-python.html. [Último acceso: 21 Noviembre 2021].

[58] AndroidWeb, «C++,» [En línea]. Available: https://lenguajesdeprogramacion.net/cpp/. [Último acceso: 04 Diciembre 2021].

[59] Universidad de Guayaquil, «Caracteristicas de un sistema de seguridad,» 2014. [En línea]. Available: http://repositorio.ug.edu.ec/bitstream/redug/6529/1/TesisCompleta-523.pdf. [Último acceso: 20 Noviembre 2021].

[60] «Electronic IDentification,» 15 Octubre 2021. [En línea]. Available: https://www.electronicid.eu/es/blog/post/como-funciona-reconocimiento-facial/es. [Último acceso: 21 Noviembre 2021].

[61] Kaspersky, «latam.Kaspersky,» 2021. [En línea]. Available: https://latam.kaspersky.com/resource-center/definitions/what-is-facial-recognition. [Último acceso: 22 Noviembre 2021].

[62] I. S.A, «IBERDROLA,» 2021. [En línea]. Available: https://www.iberdrola.com/innovacion/machine-learning-aprendizaje-automatico. [Último acceso: 2021].

[63] J. P. Labonia, «IDIS-Eigen Faces,» 14 Julio 1991. [En línea]. Available: https://proyectoidis.org/eigenface/. [Último acceso: 22 Noviembre 2021].

[64] Python, «tkinter-Python,» 2001. [En línea]. Available: https://docs.python.org/3/library/tkinter.html. [Último acceso: 22 Noviembre 2021].

[65] TechTarget, «ComputerWeekly,» Junio 2021. [En línea]. Available: https://www.computerweekly.com/es/definicion/Base-de-datos-o-DB. [Último acceso: 22 Noviembre 2021].

[66] M. Villavicencio Zambrano, «Sensores de movimiento para ahorro de energía eléctrica,» 2018. [En línea]. Available:

https://repositorio.uleam.edu.ec/bitstream/123456789/2108/1/ULEAM-IEL-0049.pdf. [Último acceso: 19 Noviembre 2021].

[67] Echatronics, «Sensor ultrasónico HC-SR04,» 2016. [En línea]. Available: https://naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html. [Último acceso: 21 Noviembre 2021].

[68] Punto Flotante S.A., «Sensor Infrarrojo de movimiento PIR,» 2017. [En línea]. Available: https://puntoflotante.net/MANUAL-DEL-USUARIO-SENSOR-DE-MOVIMIENTO-PIR-HC-SR501.pdf. [Último acceso: 19 Noviembre 2021].

[69] H. tecnologies, «Security Profile,» [En línea]. Available: https://hymtecnologies.com/producto/sirena-de-alarma-hc-606-1200s/. [Último acceso: 18 Enero 2022].

[70] «MotionEyeOS,» 2018. [En línea]. Available: https://raspberry-valley.azurewebsites.net/MotionEye-OS/. [Último acceso: 21 Noviembre 2021].

[71] «Manueal de Usuario DVR,» [En línea]. Available: http://seguridad100.com/Manuales/2404-2408-2416manual%20usuario.pdf. [Último acceso: 3 Diciembre 2021].

[72] C. J. Rojas, «DVR: qué son, tipos y cuáles son sus principales características,» [En línea]. Available: https://www.tecnoseguro.com/faqs/cctv/dvr-que-es-tipos-caracteristicas. [Último acceso: 3 Diciembre 2021].

[73] Netsuite, «Manual del usuario de Grabador de Video en Red,» [En línea]. Available: https://4820964.app.netsuite.com/core/media/media.nl?id=7467143&c=4820964&h=3ef749ed41a96bbaafe9&_xt=.pdf. [Último acceso: 3 Diciembre 2021].

ANEXOS

ETAPA CONTROL DE ACCESO

ANEXO 1

Función Crear_Registro

```
def crear_registro():
    personName = txt.get(0.1, "end-1c")
    Id_Number = txt1.get(0.1, "end-1c")
    if len(Id_Number) != 10 or len(personName) == 0:
        messagebox.showwarning("Message title", "Ingreso Incompleto!!")
    elif Id_Number in listar_Usuario()[1]:
        messagebox.showwarning("Message title", "Uuario ya Registrado!!")
    else:
        dataPath = '/home/pi/Desktop/Deteccion/Data'#Cambia a la ruta donde hayas almacenado Data
        personPath = dataPath + '/' + personName
        if not os.path.exists(personPath):
            print('Carpeta creada: ',personPath)
            os.makedirs(personPath)
        #cap = cv2.VideoCapture(0,cv2.CAP_DSHOW)
        cap = cv2.VideoCapture(0)
        #cap = cv2.VideoCapture('george.mp4')
        faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
        count = 0
        while True:

            ret, frame = cap.read()
            if ret == False: break
            frame =  imutils.resize(frame, width=640)
            gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
            auxFrame = frame.copy()
            faces = faceClassif.detectMultiScale(gray,1.3,5)
            for (x,y,w,h) in faces:
                cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
                rostro = auxFrame[y:y+h,x:x+w]
                rostro = cv2.resize(rostro,(150,150),interpolation=cv2.INTER_CUBIC)
                cv2.imwrite(personPath + '/rotro_{}.jpg'.format(count),rostro)
                count = count + 1
            cv2.imshow('frame',frame)
            k =  cv2.waitKey(1)
            if k == 27 or count >= 100:
                print('Captura finalizada')

                break
        cap.release()
        cv2.destroyAllWindows()
        train()
        registro_mySQL()
        messagebox.showinfo("Message title", "Usuario Registrado!!")
```

ANEXO 2

Funcion Train

```
def train():
    dataPath = '/home/pi/Desktop/Deteccion/Data'#Cambia a la ruta donde hayas almacenado Data
    peopleList = os.listdir(dataPath)
    print('Lista de personas: ', peopleList)
    labels = []
    facesData = []
    label = 0
    for nameDir in peopleList:
        personPath = dataPath + '/' + nameDir
        print('Leyendo las imágenes')
        for fileName in os.listdir(personPath):
            print('Rostros: ', nameDir + '/' + fileName)
            labels.append(label)
            facesData.append(cv2.imread(personPath+'/'+fileName,0))
            image = cv2.imread(personPath+'/'+fileName,0)
        label = label + 1
    face_recognizer = cv2.face.EigenFaceRecognizer_create()
    # Entrenando el reconocedor de rostros
    print("Entrenando...")
    face_recognizer.train(facesData, np.array(labels))
    # Almacenando el modelo obtenido
    face_recognizer.write('modeloEigenFace.xml')
    print("Modelo almacenado...")
```

ANEXO 3

Funcion Motor_On

```
def motor_On():
    dataPath = '/home/pi/Desktop/Deteccion/Data' #Cambia a la ruta donde hayas almacenado Data
    imagePaths = os.listdir(dataPath)
    verifyNa = txtA.get(0.1, "end-1c")
    if verifyNa in imagePaths:
        messagebox.showinfo("Message title", "Acceso Correcto")
        servoPIN = 17
        GPIO.setmode(GPIO.BCM)
        GPIO.setup(servoPIN, GPIO.OUT)
        p = GPIO.PWM(servoPIN, 50) # GPIO 17 for PWM with 50Hz
        p.start(2.5) # Initialization
        p.ChangeDutyCycle(12.5)
        time.sleep(2.0)
        p.ChangeDutyCycle(2.5)
        time.sleep(3.0)
        GPIO.setwarnings(False)
        p.stop()
        GPIO.cleanup()
    else:
        messagebox.showwarning("Message title", "Acceso Incorrecto")
```

ANEXO 4

Función Detect (Reconocimiento)

```
def detect():
    dataPath = '/home/pi/Desktop/Deteccion/Data' #Cambia a la ruta donde hayas almacenado Data
    imagePaths = os.listdir(dataPath)
    print('imagePaths=',imagePaths)
    face_recognizer = cv2.face.EigenFaceRecognizer_create()
    # Leyendo el modelo
    face_recognizer.read('modeloEigenFace.xml')
    cap = cv2.VideoCapture(-1)
    faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
    count = 0
    count2 = 0
    lista = []
    while True:
        ret,frame = cap.read()
        if ret == False: break
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        auxFrame = gray.copy()
        faces = faceClassif.detectMultiScale(gray,1.3,5)
        for (x,y,w,h) in faces:
            rostro = auxFrame[y:y+h,x:x+w]
            rostro = cv2.resize(rostro,(150,150),interpolation= cv2.INTER_CUBIC)
            result = face_recognizer.predict(rostro)
            cv2.putText(frame,'{}'.format(result),(x,y-5),1,1.3,(255,255,0),1,cv2.LINE_AA)
            # EigenFaces
            if result[1] < 5700:
                cv2.putText(frame,'{}'.format(imagePaths[result[0]]),(x,y-25),2,1.1,(0,255,0),1,cv2.LINE_AA)
                cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
                count = count + 1
                if imagePaths[result[0]] not in lista:
                    lista.append(imagePaths[result[0]])
            else:
                cv2.putText(frame,'Desconocido',(x,y-20),2,0.8,(0,0,255),1,cv2.LINE_AA)
                cv2.rectangle(frame, (x,y),(x+w,y+h),(0,0,255),2)
                count2 = count2 + 1
        cv2.imshow('frame',frame)
        k =  cv2.waitKey(1)

        if k == 27 or count >= 20:
            lbl = Label(tab3, text="Acceso Correcto!!")
            lbl.grid(column=0, row=7)
            print('Usuario Validado!!')
            break
        elif k == 27 or count2 >= 30:
            lbl = Label(tab3, text="Acceso Incorrecto!!")
            lbl.grid(column=0, row=7)
            print('Usuario No Validado!!')
            break
    #lbl.grid(column=0, row=7)
    cap.release()
    cv2.destroyAllWindows()
```

ANEXO 5

Librerías

```
from tkinter import *
from tkinter import filedialog
from PIL import Image
from PIL import ImageTk
import cv2
import os
import imutils
import numpy as np
import pymysql
from datetime import datetime
import RPi.GPIO as GPIO
import time
```

ANEXO 6

Programa Principal Tkinter

```
captura = None
root = Tk()
root.title("Industria PortiArroz")
root.geometry('350x200')


#lblb = Label(root, text="Usuario Nuevo?")
#lblb.grid(column=0, row=4)
tab_control = ttk.Notebook(root)
tab1 = ttk.Frame(tab_control)
tab2 = ttk.Frame(tab_control)
tab3 = ttk.Frame(tab_control)

tab_control.add(tab1, text="Registro Usuario Nuevo",padding=10)
lbl = Label(tab1, text="Nombre Completo")
lbl.grid(column=0, row=4)
txt = Text(tab1, width=20, height=1)
txt.grid(column=1, row=4)
lbl1 = Label(tab1, text="No. Cédula:")
lbl1.grid(column=0, row=5)
txt1 = Text(tab1, width=20, height=1)
txt1.grid(column=1, row=5)
btn = Button(tab1, text="Registrar Usuario",command=crear_registro)
btn.grid(column=1, row=6)
btn2 = Button(tab1, text="Limpiar",command=LimpiarRegistro)
btn2.grid(column=1, row=7)

tab_control.add(tab2, text="Verificar Registro")
mensaje_principal = Label(tab2, text="SISTEMA DE COMPROBACIÓN", font=("Arial Bold",15))
mensaje_principal.grid(column=0, row=0, columnspan=2)
lbl2 = Label(tab2, text="Nombre Completo")
lbl2.grid(column=0, row=2)
txt2 = Text(tab2, width=20, height=1)
txt2.grid(column=1, row=2)
btn3 = Button(tab2, text="Verificar Constancia",command=comprobarDatos)
btn3.grid(column=1, row=3)
btn4 = Button(tab2, text="Limpiar",command=LimpiarBuscador)
btn4.grid(column=1, row=4)
```

```
tab_control.add(tab3, text="Marcación")
mensaje_marcacion = Label(tab3, text="SISTEMA DE MARCACIONES", font=("Arial Bold",15))
mensaje_marcacion.grid(column=0, row=0, columnspan=2)
lblA = Label(tab3, text="Nombre Completo")
lblA.grid(column=0, row=1)
txtA = Text(tab3, width=20, height=1)
txtA.grid(column=1, row=1)
btn5 = Button(tab3, text="Realizar Marcación",command=detect)
btn5.grid(column=1, row=3)
btn6 = Button(tab3, text="Abrir Puerta",command=motor_On)
btn6.grid(column=1, row=4)
btn7 = Button(tab3, text="Limpiar",command=LimpiarReconocimiento)
btn7.grid(column=1, row=5)
tab_control.pack(padx=10, pady=10)
#tab_control.pack(expand=1, fill="both")
root.mainloop()
```

ETAPA VIDEOVIGILANCIA

ANEXO 7

```
//#include <FirebaseESP32.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <IOXhop_FirebaseESP32.h>
#include <HTTPClient.h>
#include "time.h"
#include <ESP32Time.h>
#include <UniversalTelegramBot.h>
#include <ArduinoJson.h>


#define FIREBASE_HOST "led-control-b2aaa-default-rtdb.firebaseio.com"
#define FIREBASE_AUTH "srr8KvCVlNU6W8k5D2zMKodqN7S5a2d0762tKPz7"

const char* ssid = "Claro_CABRERA0000083003";
const char* password = "8105060432023";
//const char* ssid = "iPhone de Jorge";
//const char* password = "12345678";

const char* ntpServer = "pool.ntp.org";
const long  gmtOffset_sec = -9000;
const int   daylightOffset_sec = -9000;

String token_Bot = "2134690674:AAH4yYCfipsHi1eM7U7HtRk7CiKl6kxQREQ";
String chat_id = "2010245364";

WiFiClientSecure client;
ESP32Time rtc;
UniversalTelegramBot bot(token_Bot, client);

String apiKey = "784428";
String phone_number = "+593980922607";
//String apiKey = "589828";
//String phone_number = "+593998206886";
String url;
```

```
String fireStatus = "";
int led = 23;
const int relay = 25;
int inputPin = 13;
int inputPin2 = 33;
int pirState = LOW;
int pirState2 = LOW;
int val = 0;
int val2 = 0;
int conteo = 0;
String tiempo="";
String mensaje = "";
String conteoString = "";

void printLocalTime()
{
  struct tm timeinfo;
  if(!getLocalTime(&timeinfo)){
    Serial.println("Failed to obtain time");
    return;
  }
  Serial.println(&timeinfo, "%A, %B %d %Y %H:%M:%S");
}

void setup() {
  Serial.begin(115200);
  pinMode(relay, OUTPUT);
  pinMode(inputPin, INPUT);
  pinMode(inputPin2, INPUT);
  pinMode(23, OUTPUT);
  Serial.print("Connecting to WiFi to");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
  Serial.print(".");
  delay(500);
  }
  Serial.println("");
  Serial.print("Connected to ");
  Serial.println(ssid);
  Serial.print("IP Address is : ");
  Serial.println(WiFi.localIP());
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.setString("ALARMA", "0");
  bot.sendMessage(chat_id, "Sistema Preparado!!", "");
  //init and get the time
  configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
  printLocalTime();
}

void loop() {
  val = digitalRead(inputPin);
  if (val == HIGH) {
    //digitalWrite(relay, LOW);
    if (pirState == LOW) {
```

```
    Serial.println("Motion detected parcela 1!");
    printLocalTime();
      conteo = conteo + 1;
    conteoString = String(conteo);
    mensaje = "Movimiento detectado Parcela 1" + conteoString;
    bot.sendMessage(chat_id, mensaje, "hola0");
    message_to_whatsapp("Movimineto Detectado parcela 1.");

    pirState = HIGH;
  }
}
else {
  //digitalWrite(relay, HIGH);
  if (pirState == HIGH){

    Serial.println("Motion ended!");

    pirState = LOW;
  }
}
val2 = digitalRead(inputPin2);
if (val2 == HIGH) {
  //digitalWrite(relay, LOW);
  if (pirState2 == LOW) {

    Serial.println("Motion detected parcela 2!");
    printLocalTime();
      conteo = conteo + 1;
    conteoString = String(conteo);
    mensaje = "Movimiento detectado parcela 2" + conteoString;
    bot.sendMessage(chat_id, mensaje, "hola0");
    message_to_whatsapp("Movimineto Detectado parcela 2.");

    pirState2 = HIGH;
  }
}
else {
  //digitalWrite(relay, HIGH);
  if (pirState2 == HIGH){

    Serial.println("Motion ended!");

    pirState2 = LOW;
  }
}

fireStatus = Firebase.getString("ALARMA");

if (fireStatus == "1") {
  Serial.println("ALARMA ENCENDIDA!!");
  digitalWrite(23, HIGH );
  digitalWrite(relay, LOW);
}
else if (fireStatus == "0") {
  Serial.println("ALARMA APAGADA");
  digitalWrite(23, LOW);
  digitalWrite(relay, HIGH);
```

```
  }
  else {
    Serial.println("Wrong Credential! Please send ON/OFF");
  }
}

void  message_to_whatsapp(String message)
{
  url = "https://api.callmebot.com/whatsapp.php?phone=" + phone_number + "&apikey=" + apiKey + "&text=" +
urlencode(message);

  postData();
}

void postData()
{
  int httpCode;
  HTTPClient http;
  http.begin(url);
  httpCode = http.POST(url);
  if (httpCode == 200)
  {
    Serial.println("Sent ok.");
  }
  else
  {
    Serial.println("Error.");

  http.end();
}

String urlencode(String str)
{
    String encodedString="";
    char c;
    char code0;
    char code1;
    char code2;
    for (int i =0; i < str.length(); i++){
      c=str.charAt(i);
      if (c == ' '){
        encodedString+= '+';
      } else if (isalnum(c)){
        encodedString+=c;
      } else{
        code1=(c & 0xf)+'0';
        if ((c & 0xf) >9){
            code1=(c & 0xf) - 10 + 'A';
        }
        c=(c>>4)&0xf;
        code0=c+'0';
        if (c > 9){
            code0=c - 10 + 'A';
        }
        code2='\0';
        encodedString+='%';
        encodedString+=code0;
```

```
        encodedString+=code1;
        //encodedString+=code2;
      }
      yield();
    }
    return encodedString;
}
```