UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE INGENIERÍA ELECTRÓNICA

“DISEÑO E IMPLEMENTACION DE UN MODULO ELECTRONICO PARA SISTEMA DE SEGURIDAD CON PYTHON”

Trabajo de titulación previo a la obtención del

Título de **Ingeniero Electrónico**

AUTORES: **DIAZ MONTAÑO JORGE JOHAN**

FUENTES ROBELLY EDWIN FERNANDO

TUTOR: **MSC. VICENTE PEÑARANDA**

GUAYAQUIL – ECUADOR

2021 - 2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, **Diaz Montaño Jorge Johan** con documento de identificación N° **0930618079** y **Fuentes Robelly Edwin Fernando** con documento de identificación N° **0930646799** manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación

Guayaquil, 29 de Julio del año 2022

Atentamente,

(f) Diaz Montaño Jorge Johan

C.I: 0930618079

(f) Fuentes Robelly Edwin Fernando

C.I: 0930646799

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Nosotros, **Diaz Montaño Jorge Johan** con documento de identificación No. **0930618079** y **Fuentes Robelly Edwin Fernando** con documento de identificación No. **0930646799**, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación: "**DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO ELECTRÓNICO PARA SISTEMA DE SEGURIDAD CON PYTHON**", el cual ha sido desarrollado para optar por el título de: **Ingeniero Electrónico**, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, 29 de Julio del año 2022

Atentamente,

(f) Diaz Montaño Jorge Johan
C.I: 0930618079

(f) Fuentes Robelly Edwin Fernando
C.I: 0930646799

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo Msc. Vicente Peñaranda con documento de identificación N° **0916113426**, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: "**DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO ELECTRÓNICO PARA SISTEMA DE SEGURIDAD CON PYTHON**", realizado por: **Diaz Montaño Jorge Johan** con documento de identificación N° **0930618079** y por **Fuentes Robelly Edwin Fernando** con documento de identificación N° **0930646799**, obteniendo como resultado final el trabajo de titulación bajo la opción proyecto técnico, que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 29 de Julio del año 2022

Atentamente,

_____
Msc. Vicente Avelino Peñaranda Idrovo
C.I: 0916113426

DEDICATORIA

El presente trabajo de titulación lo dedico principalmente a mis padres porque con su ejemplo han forjado en mi un ser humano con valores y principios.

En segundo lugar este trabajo va dedicado a mis profesores de la Universidad Politécnica Salesiana, en especial a mi tutor de Tesis el Ing. Vicente Peñaranda por su dedicación y comprensión.

Diaz Jorge

DEDICATORIA

Dedico este proyecto técnico a mis padres por su apoyo incondicional en cada paso que doy.

Fuentes Fernando

AGRADECIMIENTO

Mi principal agradecimiento es para Dios nuestro creador por haberme regalado la vida y los padres maravillosos que tengo Marilú Montaño y Jorge Diaz; porque gracias a ellos y a su amor, he podido alcanzar un logro importante en mi vida.

En segundo lugar agradezco a mis profesores que me han impartido sus conocimientos; también agradezco a mi compañero de Tesis que juntos logramos hacer un gran equipo.

Diaz Jorge

AGRADECIMIENTO

Agradezco a todos los docentes de la Universidad Politécnica Salesiana que supieron guiarme en los años de estudios de alguna forma se interesaron que culmine mi tesis satisfactoriamente.

Fuentes Edwin

RESUMEN

AÑO	ALUMNOS	DIRECTOR DE PROYECTO	TEMA DE PROYECTO DE TITULACIÓN
2022	DIAZ MONTAÑO JORGE JOHAN EDWIN FERNANDO FUENTES ROBELLY	MSC. VICENTE PEÑARANDA	"DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO ELECTRÓNICO PARA SISTEMA DE SEGURIDAD CON PYTHON "

Debido a que en la actualidad el índice de la delincuencia ha aumentado, las instituciones y hogares ya no son seguros porque son un blanco fácil para los delincuentes, por ende con la tecnología se busca mejora las técnicas de seguridad.

El presente proyecto trata del diseño e implementación de un módulo electrónico para sistema de seguridad con Python por medio del reconocimiento facial que permite simular el acceso a una puerta de una casa u oficina, solo permite ingresar a las personas autorizadas, el módulo consta con una Webcam (cámara web), que apunta al rostro de la persona, toda la información de inscripción de los usuarios se almacena en la base del sistema.

Palabras Clave: Raspberry Pi 4, OpenCV, Raspbian, Webcam

ABSTRACT

YEAR	STUDENTS	PRJ. DIRECTOR	SUBJECT
2022	DIAZ MONTAÑO JORGE JOHAN EDWIN FERNANDO FUENTES ROBELLY	MSC. VICENTE PEÑARANDA	"DESIGN AND IMPLEMENTATION OF AN ELECTRONIC MODULE FOR A SYSTEM SECURITY WITH PYTHON"

Due to the fact that currently the crime rate has increased, institutions and homes are no longer safe because they are an easy target for criminals, therefore technological advances seek to modernize security techniques.

This project deals with the design and implementation of an electronic module for a security system with Python through facial recognition that allows simulating access to a door of a house or office, it only allows authorized persons to enter, the module consists of a Webcam (web camera), pointing to the person's face, all user registration information is stored in the system base.

Keywords: Raspberry Pi 4, OpenCV, Raspbian, Webcam

ABREVIATURAS

VA: Visión Artificial

FGE: Fiscalía General del Estado

OpenCV: Open Source Computer Vision library

PCA; Principal Component Analysis

LDA: Linear Discriminat Analysis

FLD: Discriminant Lineal Fisher - Discriminante Lineal de Fisher

LBPH: Local Binary Pattern Histogram

EFP: Error Falso Positivo

EFN: Error Falso Negativo

GPU: Unidad de Procesamiento Gráfico

RAM: Random Access Memory

USB: Universal Serial Bus

HDMI: High-Definition Multimedia Interface

GPIO: General Purpose Input/Output, Entrada/Salida de Propósito General

CSI: Camera Serial Interface

IP: Internet Protocol

WLAN: Wireless LAN

N.C: Normalmente Cerrado.

N.O: Normalmente Abierto

UPS: Uninterruptable Power Supply

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ANEXOS

INTRODUCCIÓN

Los sistemas de visión computarizada son herramientas relativas a los más recientes avances de la validación de usuarios que hay en todo el mundo, siendo el más confiable.

Con el avance de la tecnología y el desarrollo de ordenadores portátiles como lo es la Raspberry Pi, se logra que los sistemas de seguridad por visión computarizada ayuden a solucionar inconvenientes de mejor modo como es el caso de la seguridad en el hogar o en trabajo.

Para explicar el desarrollo del proyecto de titulación se ha dividido el documento como se especifica a continuación:

La primera parte contiene el problema donde se puntúa los antecedentes, la justificación del trabajo de titulación, importancia y alcance, las delimitación del problema y define los objetivos; la segunda parte presenta el marco teórico donde se habla de los principales conceptos y fundamentos teóricos de los sistemas de visión artificial ligados al reconocimiento facial; la tercera parte se detalla el diseño del proyecto y su funcionamiento; la cuarta parte muestra la implementación del módulo; la quinta parte proyecta el análisis y resultados del módulo; la sexta parte detalla las conclusiones al momento de finalizar la realización del módulo de seguridad; la séptima parte muestra las recomendaciones; la octava parte contiene los proyectos vinculados a la investigación; la novena parte están las referencias bibliográficas y por último la décima parte contiene los anexos.

1. EL PROBLEMA

1.1. Antecedentes

Un gran problema de la sociedad es el estado de vulnerabilidad con respecto a la seguridad, como son en los hogares, trabajos, centros de educación, entre otros; ya que el índice de delincuencia ha aumentado de acuerdo con las estadísticas proporcionadas por la Fiscalía General del Estado del Ecuador que es el órgano autónomo de la Función Judicial; por ejemplo informa que durante el año 2020 al 2021, la delincuencia registra una variación 12,1%, que refleja un aumente en el robo a domicilios en el período enero – noviembre del 2021 (Fiscalía General del Estado del Ecuador, 2021). Lastimosamente los delincuentes siempre se las ingenian para vulnerar la seguridad, llevando con esto a los dueños de los hogares, trabajos, centros de educación, entre otros; a utilizar sistemas de seguridad contra quienes lo burlan, por lo cual se presenta esta opción que propone poner una gran dificultad a los delincuentes al momento de llevar a cabo su delito.

1.2. Justificación del Trabajo de Titulación

Teniendo en cuenta el índice de inseguridad que hay actualmente en el país, cualquier persona puede ser víctima de la delincuencia tanto en el hogar, instituciones o empresas; por eso surge la necesidad de invertir en un buen sistema de seguridad que proteja a las personas ya sean propietarios, empleados o seres queridos; teniendo en cuenta los métodos de seguridad más habituales que son alarmas o los candados; sin embargo como la delincuencia ha avanzado en sus modus operandi, se ha pensado con la presente propuesta el desarrollo de un sistema por medio de la visión artificial, que mejore la seguridad.

1.3. Importancia y alcances

El presente proyecto tiene una gran importancia debido a que, con el desarrollo del módulo práctico para sistema de seguridad mediante el reconocimiento facial, incrementa el nivel de seguridad de los hogares, instituciones o empresas, la ventaja del sistema electrónico es que las personas que ingresan al lugar son las que constan en el registro del sistema, para mayor seguridad este sistema hace un registro de los usuarios que entran o intentaron ingresar, se toma captura del rostro del individuo y se envía por correo al programador y al supervisor.

Tomando en cuenta el problema que tiene actualmente el país en cuanto a la delincuencia, fue necesario la realización de este módulo de seguridad para proteger y minimizar las denuncias de robo a domicilios, instituciones u oficinas y crear un ambiente de seguridad para las personas.

1.4. Delimitación del problema

1.4.1 Temporal

El trabajo de titulación fue realizado en los años 2021 al 2022.

1.4.2 Espacial

El trabajo de titulación está destinado para el uso práctico de los estudiantes de la carrera Ingeniería Electrónica de la Universidad Politécnica Salesiana de Guayaquil (Campus Centenario), con la creación de este módulo didáctico los estudiantes Salesianos tendrán una idea más clara de cómo funcionan los sistemas de seguridad con visión artificial y como el nivel de seguridad aumenta al ser notificado a cualquier hora del día de las personas que entran o intentaron ingresar al establecimiento o a una área no autorizada.

1.4.3 Académica

Por el gran aporte de la tecnología en el mundo actual, hoy en día se puede hacer diversos cambios en el campo de seguridad. El objetivo de la Universidad Politécnica Salesiana es formar ingenieros con un perfil técnico altamente capacitados para desarrollarse en el campo laboral, por ese motivo la universidad crea carreras técnicas para resolver esta problemática.

Una problemática de nuestro entorno son los robos a domicilios a nivel nacional, el robo a domicilios marca un aumento considerable, de 6.643 en el año 2020 paso a 7.449 al año 2021 (Fiscalía General del Estado del Ecuador, 2021).

A través de la realización del proyecto de titulación del Módulo Seguridad Programado en Python mediante el reconocimiento facial, se da una opción confiable para resolver los problemas de robos que se da día a día; para la realización de este proyecto se aplicaron conocimientos técnicos adquiridos durante los cursos regulares y seminarios profesionales dictados a lo largo de la Carrera de Ingeniería Electrónica, en las materias tales como: Circuitos Eléctricos, Teoría de Control, Teoría Electromagnética, Programación y Redes de Comunicación.

1.5. Objetivos

1.5.1 Objetivo General

Desarrollo e implementación de un sistema de seguridad por reconocimiento facial programado en Python.

1.5.2 Objetivos Específicos

- Programación en Python para un módulo didáctico para identificación facial.
- Diseño e implantación de un módulo didáctico para reconocimiento facial.
- Desarrollo del software sobre la plataforma Raspberry Pi.
- Desarrollo de 5 prácticas para uso del módulo didáctico.

2. MARCO TEÓRICO REFERENCIAL

2.1. Seguridad en el Ecuador.

Según estudios en FGE, entre enero y noviembre del 2021 con respecto al mismo periodo del año 2020, hay un incremento de robo a domicilios y va en aumento a medida que pasa el tiempo.

En el 2020 las denuncias de robo a domicilios fueron de 6.643 y en el periodo del 2021 las denuncias subieron a 7.449, teniendo un incremento con una variación del 12,1% (Fiscalía General del Estado del Ecuador, 2021). Lastimosamente los delincuentes siempre se las ingenian para vulnerar la seguridad, llevando con esto a los dueños de los hogares, trabajos, centros de educación, entre otros; a utilizar sistemas de seguridad contra quienes lo burlan, por lo cual se presenta esta opción que propone poner una gran dificultad a los delincuentes al momento de llevar a cabo su delito. No obstante, existen también excelentes dispositivos de seguridad; aquí se presenta una lista de tres sistemas antirrobo como se observa en la tabla 1.

2.1.1. Cerraduras antibumping

La cerradura antibumping es un sistema de seguridad que refuerza la capacidad del cilindro de la cerradura que lo hace resistente a los golpes, con los que es casi imposible que la puerta se abra. (*MAPFERE*, n.d.).

2.1.2. Ring video doorbell

Ring Video Doorbell, es un sistema de seguridad que cuenta con una cámara integrada y con comunicación bidireccional para que puedas hablar con la persona que esté en la puerta con la app de Ring (*RING*, n.d.).

2.1.3. Temporizadores

Temporizadores, este sistema de seguridad controla el acceso de las puertas de los hogares, instituciones o empresas, la persona que abra la puerta es la que tenga acceso a la aplicación instalada en el celular (*MAPFRE*, n.d.).

A continuación en la tabla 1, se muestra el cuadro comparativo acerca de los costos de instalación de varios sistemas antirrobo.

Sistema antirrobo	Costo de instalación (USD $)	Imagen del sistema
Cerraduras antibumping	560	
Ring video doorbell	570	
Temporizadores	600	

Tabla 1. Sistema antirrobo con el costo de instalación

2.2. Estadísticas de robo a domicilios.

A nivel nacional, el robo a domicilios en el año 2020 las denuncias fueron 6.643 y paso a 7.449 en el año 2021 (Fiscalía General del Estado del Ecuador, 2021). Como se observa en la figura 1, los robos a domicilios son más común en la noche.

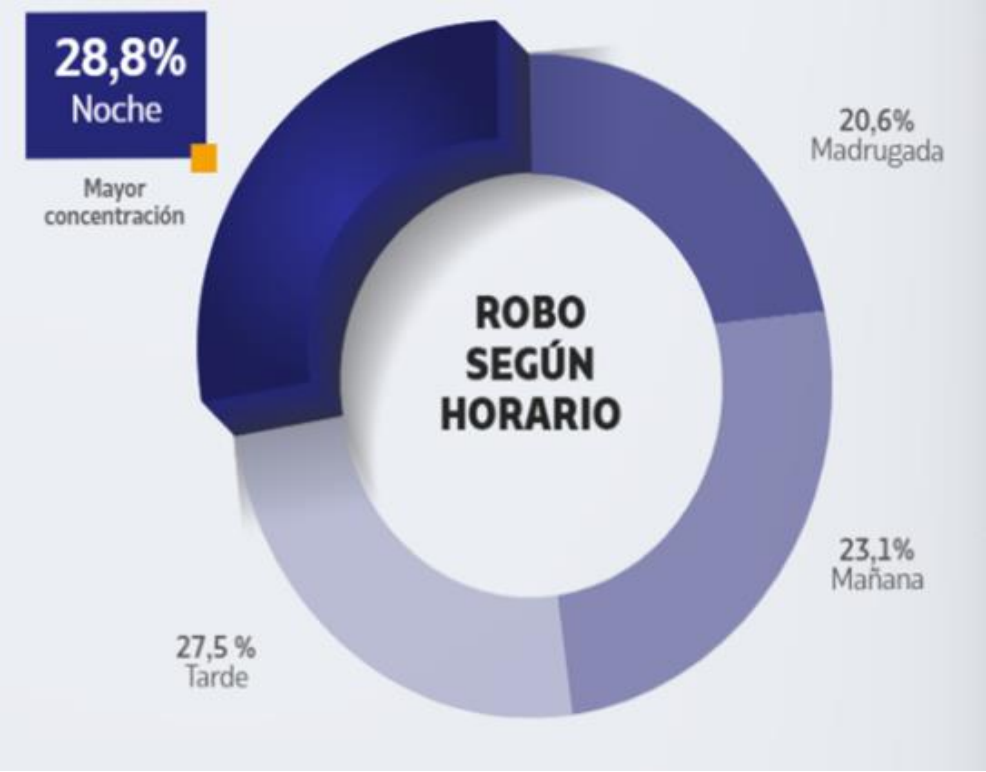


Figura 1. Robo de hogares en el 2020 y 2021 (Fiscalía General del Estado del Ecuador, 2021).

En la tabla 2 se observa el total de denuncias por mes de robo a domicilios durante el año 2021:

Cantidad de denuncias de robo a domicilios 2021		
Delito	**Mes**	**Total de denuncias**
Robo a domicilio	Enero	681
	Febrero	678
	Marzo	657
	Abril	591
	Mayo	596
	Junio	608
	Julio	712
	Agosto	691
	Septiembre	737
	Octubre	771
	Noviembre	721

Tabla 2. Cantidad de denuncias de robo a domicilios en el año 2021 (Fiscalía General del Estado del Ecuador, 2021).

Como se observa en la tabla 2, el robo a domicilio aumento en el mes de Octubre.

2.3. Visión Artificial como método a la seguridad.

A lo largo del tiempo la investigación de la visión artificial a menudo ha seguido caminos similares a la del ojo humano, utilizando una cámara para desempeñar el papel de ojos; y una computadora para analizar las imágenes digitales.

Las primeras investigaciones de la visión artificial se realizaron en la década de los 60, donde se realizaron sistemas de visión para la comprensión del mundo; la mayoría de las soluciones ofrecidas no eran transferibles a la visión en el mundo natural. A pesar de esto; esa línea de investigación abrió el camino para muchas direcciones de investigación interesantes; todas moviéndose lentamente hacia el objetivo final de una visión artificial similar a la humana sin restricciones (Bharath & Petrou, 2008).

2.4. Tipos de métodos de reconocimiento de rostro para sistemas de seguridad por medio de la visión artificial

Uno de los mayores problemas al momento de realizar un sistema de seguridad por visión artificial es la detención y la identificación del rostro; antes que comenzar la programación del módulo hay que tener claro el funcionamiento que va a realizar y donde se va a instalar el sistema de seguridad para así tener una detección precisa de la imagen captada para el reconocimiento del rostro del usuario. Hay que tener en cuenta que el sistema no solo debe capturar y detectar el rostro para el reconocimiento de las personas; sino que también toma otros aspecto como son:

- Posición, tamaño y expresiones del rostro.

- Objectos colocados en el rostro como son los gafas, gorra, entre otros.

- Expresiones faciales al momento de la captura del rostro como son los estados de ánimo: feliz, triste, enojado, etc.

- Las condiciones del ambiente como es la iluminación.

Para detectar estos aspectos y no tener inconvenientes al momento de la detección del rostro se desarrollaron las técnicas de reconocimiento facial (Gibrán García C., 2019), las cuales son:

- Métodos que utilizan imágenes de intensidad las cuales se dividen en 2 tipos: las que están basadas en características y las que están dirigidas a los enfoques holísticos.

- Métodos por medio de las secuencias de video.

2.4.1. Métodos que utilizan imágenes de intensidad

Los métodos que utilizan imágenes de intensidad se dividen en dos tipos las cuales son: basadas en características y las que están dirigidas a los enfoques holísticos.

Las que están basada en características; su función principal es procesar la imagen para después identificarla, luego extraer y medir los rasgos faciales del rostro

como son ojos, nariz y boca para calcular las relaciones geométricas entre esos puntos faciales para así obtener el vector de características geométricas.

Y los métodos que están dirigidos a los enfoques holísticos; es lo contrario al método de categoría de características es decir que utiliza toda la imagen del rostro para hacer una comparación de rostros; este método se divide en dos grupos:

Enfoques estadísticos donde la imagen se representa en una matriz con valores de intensidad donde se realiza comparaciones de los rostros guardados en la base de datos con la imagen del rostro de entrada.

Y la Inteligencia Artificial (IA), es la que utiliza las redes neuronales para el reconocimiento facial (Gibrán García C., 2019).

2.4.2. Método por medio de las secuencias de video.

Este método consta de tres módulos:

Un primer módulo que es para detección del rostro; un segundo módulo que ayuda a rastrear la cara que se muestra en el video; y el tercer módulo que se encarga de reconocer el rostro (Gibrán García C., 2019).

2.5. Modelos utilizados para el reconocimiento Facial

Teniendo en cuenta los problemas y los métodos que se utilizan para mejorar los sistemas de seguridad por medio del reconocimiento facial; la parte más compleja es comprender como funciona la comparación de los rostros para reconocer y diferenciar los usuarios registrados y los no registrados; para poder comprender mejor esa comparación se realizó los cálculos matemáticos; el proceso de reconocimiento consiste en tomar una captura del rostro (Cajas Idrovo & Viri Ávila, 2017); esta imagen contiene filas y columnas por lo cual se transforma en un vector unitario que contiene n-dimensional (n = a×b) donde n es el número de dimensiones.

Después de transformar la imagen principal en un vector unitario se entiende que ese vector proyecta el resultado en el subespacio y el resultante es un vector de menor dimensión; para que se cumpla este proceso se usa el método de reducción de dimensiones más conocido como el método de extracción de características, luego la proyección de la imagen principal se compara con la proyección de varias imágenes obtenidas de la base de datos (Cajas Idrovo & Viri Ávila, 2017); la proyección más similar de las imágenes de la base de datos con la imagen principal; será el resultado del proceso de comparación de rostros (Ottado, 2010).

Para la realización del proceso de comparación de rostros se utilizaron los modelos: Eigenfaces, Fisherfaces y LBPH; después se compararon los resultados obtenidos y se utilizó en el módulo de seguridad el modelo que mejor resultado dio.

2.5.1. Modelo Eigenfaces

El modelo Eigenfaces es uno de los métodos de reconocimiento de captura de rostro se realiza mediante la proyección lineal del espacio de las imágenes en un subespacio de menor dimensiones (Cajas Idrovo & Viri Ávila, 2017).

La reducción de las dimensiones se realiza usando la Técnica Principal Component Analysis (PCA), la cual utiliza la proyección lineal para separar las imágenes que son proyectadas (Ottado, 2010).

Para sacar la ecuación 1, hay que tener en cuenta que este es un conjunto de n-imagen; donde (n) es el número del valor de esas imágenes que están en el espacio con n-dimensiones:

$$\{a_i\} \ \ i = (1,2, \dots, n) \qquad \text{(Ecuación 1)}$$

Entonces se entiende, que cada una de las imágenes pertenece a una clase $\{A_1, A_2, \dots, A_C\}$. Donde c = clase

Se realizó también la transformación lineal del espacio de separación de la imagen con n-dimensiones al espacio de características de m-dimensiones; donde (m) es menor que (n), como resultado tenemos nuevos vectores que son de características: $b_k \in L^m$, las cuales se definen en la ecuación 2:

$$bk = w^T a_k \quad k = (1,2, \dots, N) \qquad \text{(Ecuación 2)}$$

W= Matriz con columnas ortonormales.

En la ecuación 3 se tiene la matriz de distribución S_T:

$$S_T = \sum_{k=1}^{N} (a_k - \mu)\ (a_k - \mu)^T \qquad \text{(Ecuación 3)}$$

$\mu =$ Medida total de las imágenes de la ecuación 1

A W^T se le aplica la trasformada lineal, como resultado da $W^T S_T W$ que es la distribución de los vectores $\{y_1, y_2, \dots, y_N\}$.

Luego se toma la proyección de la matriz de las columnas ortonormales W_{opt} que aumenta el determinante de la distribución de toda la matriz de las imágenes; entonces se tiene la ecuación 4:

$$W_{opt} = arg_w^{max} |W^T S_T W| \qquad \text{(Ecuación 4)}$$

$$= [\mathrm{w1}, \mathrm{w2}, \dots, W_m]$$

2.5.2. Modelo Fisherfaces

El método Fisherfaces es una técnica de reconocimiento facial que ayuda a clasificar y reducir la dimensión de la captura de rostro utilizando el método FLD o también conocido como el método Discriminant Lineal Fisher (Ottado, 2010).

Este método selecciona la matriz con columnas ortonormales; de manera que la distribución de las clases y de la intra-clases el cociente de estas dos sea mayor (Carlos H. Esparza Franco et al., 2017); en la ecuación 5 se ve la matriz de dispersión S_B :

$$S_B = \sum_{i=1}^{c} (\mu_i - \mu)\ (\mu_i - \mu)^T \quad \text{(Ecuación 5)}$$

En la ecuación 6 se define la matriz de distribución intra-clases S_W:

$$S_W = \sum_{i=1}^{c}\ \sum_{ak \in A1} \mathrm{N}_i (\mu_i - \mu)\ (\mu_i - \mu)^T \quad \text{(Ecuación 6)}$$

μ_i = Imagen media perteneciente a la clase Ai

N_i= Número de imágenes de la clase Ai.

El algoritmo FisherFace busca que en la proyección de la matriz con columnas ortonormales la separación de las clases sea mayor, entonces se tiene que la ecuación 7 es:

$$W_{opt} = arg^{max}_{w} \left| \frac{W^T S_B W}{W^T S_W W} \right| \quad \text{(Ecuación 7)}$$

$= [\mathrm{w1}, \mathrm{w2}, \ldots, W_m]$

Lugo se tiene que resolver las ecuaciones dadas con PCA (Principal Component Analysis); donde S_B y S_W son dadas a partir de los datos de las imágenes que son proyectadas sobre el subespacio (Carlos H. Esparza Franco et al., 2017); donde el resultado final se aplica Fisherfaces; todo esto se aprecia en la ecuación 8:

$$S_B w_i = \lambda_i S_W w_i \qquad i = 1,2, \ldots, m \qquad \text{(Ecuación 8)}$$

Unos de los principales problemas del reconocimiento facial es el patrón que se utiliza para la comparación de rostros; ya que los N-números de muestra siempre es menor a las dimensión de entrada; por lo que esto provoca que la matriz de distribución intra-clases (S_{W}) se transforme en singular; para resolver este problema se utiliza PCA para poder realizar la reducción de las dimensiones en el espacio de características ($N\mathrm{umero} - c\mathrm{lase}$); y al finalizar se aplica un FLD (Discriminant Lineal Fisher) definido en la ecuación 7 y así poder reducir la dimensión a ($c\mathrm{lase} - 1$) donde c=clase. De esta manera S_{W} ya no es singular.

En la ecuación 9 se observa que la matriz W_{opt} dado por:

$$W^T{}_{\mathrm{pca}} = W^T{}_{\mathrm{fld}} W^T{}_{\mathrm{pca}} \qquad \text{(Ecuación 9)}$$

La optimización del problema dada en la ecuación 9 por tanto se reescribe, donde:

$$W_{pca} = arg^{max}_{w} |W^T S_T W| \qquad \text{(Ecuación 10)}$$

$$W_{fdl} = arg^{max}_{w} \frac{|W^T W_{pca}^T S_B W_{pca} W|}{|W^T W_{pca}^T S_W W_{pca} W|}$$ (Ecuación 11)

Entonces como resumen se tiene que:

El reconocimiento facial se realiza mediante la proyección lineal en el espacio de la captura del rostro en el subespacio formado por el modelo Eigenfaces y esa imagen es comparada con las capturas guardadas en la base de datos (Carlos H. Esparza Franco et al., 2017).

Figura 2. Modelo EigenFace (Carlos H. Esparza Franco et al., 2017)

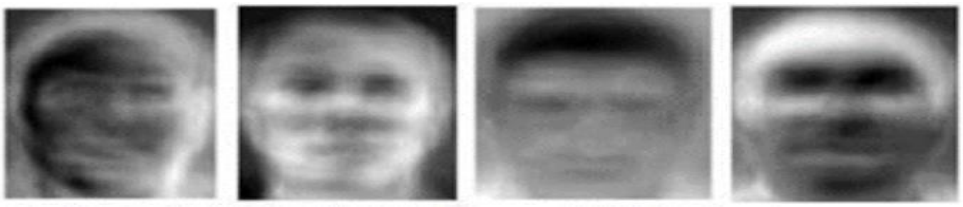


Figura 3. Modelo Fisherfaces (Carlos H. Esparza Franco et al., 2017)

Como resumen comparativo se tiene que los dos métodos son eficientes pero la cantidad de cálculo y la memoria que dispone utilizar estos métodos son muy elevados y esto nos generara problema al momento de utilizar el módulo de seguridad

como es: poner lento el programa y así provocar un reinicio al sistema haciendo que se apague el módulo de seguridad. En la figura 2 y la figura 3 se muestra los modelos EigenFace y FisherFace siendo el método FisherFace el más optimo ya que ocupa menos memoria de almacenamiento; pero unos de sus principales problemas es su sensibilidad a la luz eso hace poner lento al momento de capturar y guardar el rostro en la etapa de registro pero existe otro método que sirve para captura y reconocimiento de rostros que es incluso más rápido y ocupa menos memoria la cual la hace la adecuada para el desarrollo del módulo de seguridad; a continuación se habla más del modelo LBPH.

2.5.3. Modelo LBPH

El modelo LBPH (Histograma de patrón binario local) o también llamado el modelo de patrones binarios locales; es una técnica de reconocimiento facial donde su principal objetivo es no tomar en cuenta toda la imagen como un vector de gran dimensión (Carlos H. Esparza Franco et al., 2017); sino describir en la imagen las características de uno o varios objetos como es el uso de gorra, gafas, etc; donde estas características que se obtienen tendrán una baja dimensión gracias al uso de descripciones locales en las regiones del rostro de la imagen a comparar esto aporta información que ayuda a detectar el rostro al momento de reconocimiento del usuario (Sierra, 2015). Como resumen se tiene que el modelo LBPH su función principal al momento del reconocimiento de rostro es resumir la estructura de una imagen mediante la comparación de cada píxel con los pixeles de otra imagen almacenada en la base de datos; toma un píxel de la imagen a comparar y este se toma como el pixel principal es decir se le asigna una etiqueta y se limita el valor de los pixeles de

las imágenes almacenadas y el pixel principal se compara con los pixeles de las imágenes guardadas en la base de datos (Carlos H. Esparza Franco et al., 2017); en esta comparación se presentan dos casos:

El primer caso es si el pixel principal es mayor a los pixeles de las imágenes almacenadas se le asigna el número 0

El segundo caso es si los pixeles vecino de las imágenes guardas en la base de datos es mayor o igual que el pixel principal, entonces se le asigna el número 1, como se observa en la Figura 4.

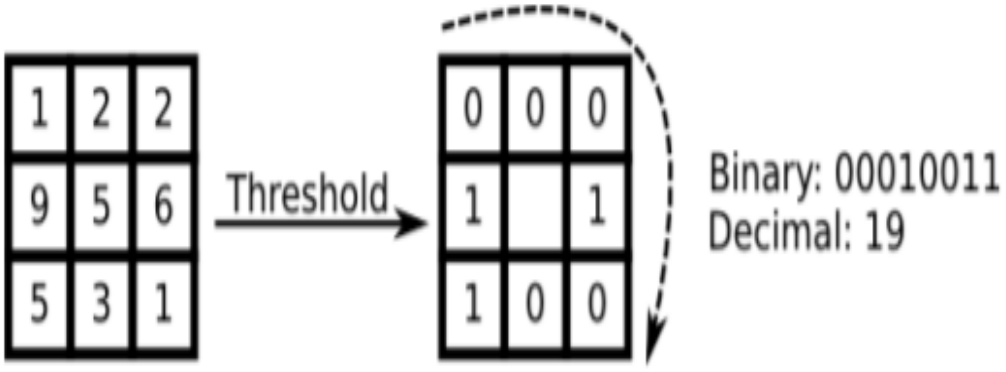


Figura 4. Funcionamiento del modelo LBPH (Sierra, 2015).

Descripción del algoritmo: Tomando en cuenta $LBPH(x_c + y_c)$ como el operador principal donde $(x_c + y_c)$ es nuestro pixel principal y i_c e i_n como intensidad de los pixeles de las imágenes almacenadas entonces se tiene como ecuación 12:

$$LBPH(x_c + y_c) = \sum_{p=0}^{p-1} 2^p \, s(i_p - i_c) \qquad \text{(Ecuación 12)}$$

$$\boldsymbol{s}\,(x) = \begin{cases} \mathbf{1}\ \boldsymbol{si}\ x \geq\ 0 \\ \mathbf{0}\ \boldsymbol{e.o.c} \end{cases} \qquad \text{(Ecuación 13)}$$

Tomando en cuenta la ecuación 12, esta ecuación nos permite capturar los puntos principales de las imágenes como son objetos y posiciones de los ojos, boca y nariz del rostro, el punto del pixel principal es $(x_c\ +\ y_c\,)$ y los pixeles de las imágenes guardadas es $\left(x_p\ +\ y_p\right)$; donde los pixeles de las imágenes guardadas se calculan como se observa en las ecuaciones 14 y 15:

$$x_p\ =\ x_c\ +\ R\ \cos(\frac{2\pi p}{p}) \qquad \text{(Ecuación 14)}$$

$$y_p\ =\ y_c\ +\ R\ \cos(\frac{2\pi p}{p}) \qquad \text{(Ecuación 15)}$$

R Radio del círculo que forman los pixeles de las imágenes almacenadas.

P= Número de puntos que tendrán las imágenes guardadas.

Si los puntos encontrados en el radio del círculo no se asemejan a las coordenadas de la imagen a comparar, el punto encontrado entonces se lo interpola como se observa en la ecuación 16 (OpenCV implementa una interpolación bilineal):

$$f(x,y) \approx [1\ -\ x\ x] \begin{bmatrix} f(0,0) & f(0,1) \\ f(1,0) & f(1,1) \end{bmatrix} \begin{bmatrix} 1\ -\ y \\ y \end{bmatrix} \qquad \text{(Ecuación 16)}$$

En la figura 5 se muestra, la insensibilidad a la luz :

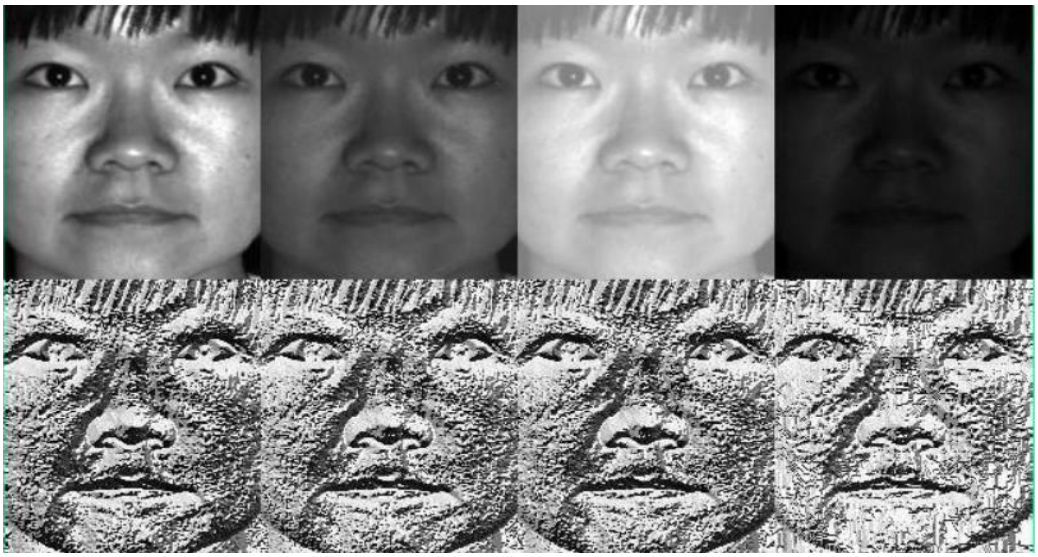


Figura 5. LBPH (Programador Clic, s.f.)

2.6. Algoritmo de la Visión Artificial

2.6.1. Cascada de clasificadores Haar

En el año 2001 Paul Viola y Michael Jones dos investigadores desarrollaron el método de cascada de clasificación Haar; este método de clasificación es una forma de detectar objetos en una o varias imágenes (Cajas Idrovo & Viri Ávila, 2017).

Para poder entender el funcionamiento de la cascada de clasificadores Haar; Viola y Jones presentaron un esquema detallando el proceso que realiza el algoritmo para clasificar las imágenes:

Como primer paso está el gesto de entrada que es la imagen capturada o también llamada la imagen principal.

El segundo paso a la clasificación esta la integración de la imagen que es la transformación de la imagen principal de la cual como resultado se tiene una nueva imagen; y los pixeles de la imagen nueva son sumada a los pixeles de la imagen principal.

En tercer paso esta la extracción de las características aquí se compara cada pixel de las imágenes almacenadas con el pixel primario de la captura de la imagen principal.

En el cuarto paso esta la clasificación, aquí se toma a las imágenes similares en la comparación con la imagen principal y se deja a un lado las imágenes que no coinciden.

Y en el quinto y último paso se tiene a la Imagen con el gesto detectado.

Todos estos pasos se pueden observar en la figura 6.

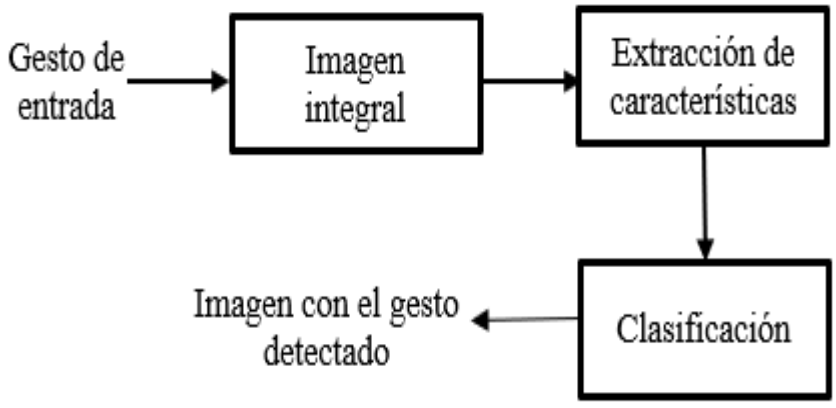


Figura 6. Etapas metodología Haar

2.7. Tipo de errores

En el proceso del reconocimiento del rostro, el sistema puede generar dos tipos de errores al momento de la comparación de las imágenes (Miguel Ángel Vázquez López, 2014):

Los Errores falsos positivos (EFP): Es cuando el usuario estando no registrado en el sistema intenta ingresar al módulo de seguridad; la imagen que es capturada por el sistema se detecta como una imagen conocida y es declara erróneamente como imagen encontrada o usuario registrado.

Los Errores falsos negativos (EFN): Es cuando el usuario estando registrado en el sistema intenta ingresar al módulo de seguridad; la imagen que es capturada por el sistema se detecta como una imagen no conocida y es declara erróneamente como imagen no encontrada o usuario no registrado.

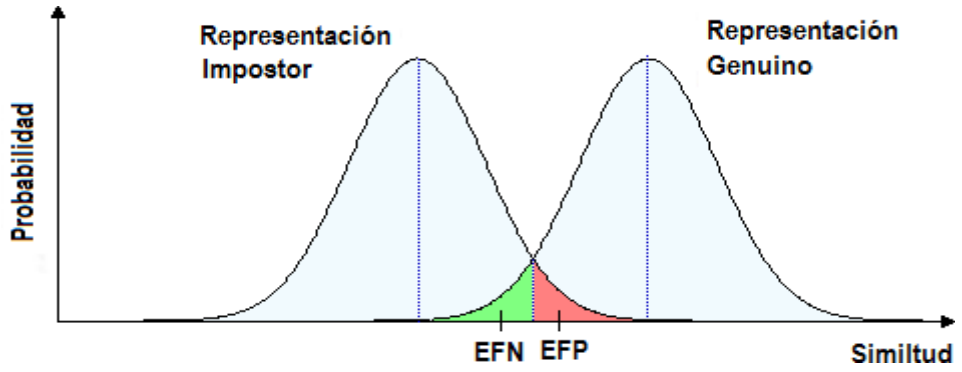


Figura 7. Probabilidades de un usuario no registrado y un usuario registrado sea detectado por el sistema. (Miguel Ángel Vázquez López, 2014).

En la figura 7, se representa lo que es la distribución de probabilidad de un usuario no registrado y uno que está registrado en el sistema; con la gráfica mostrada en la

Figura 7 se da a conocer que dependiendo el modelo que se utilice para el almacenamiento y comparación de las imágenes hay una gran probabilidad de tener este tipo de errores al momento de reconocer al usuario (Miguel Ángel Vázquez López, 2014).

2.8. Etapa del Proceso de la Visión computarizada

Una vez conocido como funciona el reconocimiento facial por medio de la visión computarizada o mejor conocida como visión artificial y los tipos de errores que se presentan al momento de la comparación; es de suma importancia definir los pasos a realizar el módulo de seguridad para que así tener una programación ordenada y si se quiere cambiar o mejorar algo se pueda buscar fácilmente en las líneas de código del programa.

Como primer paso está en obtener la captura del rostro (Ordieres et al., 2006).

El segundo paso es el preprocesar dicha captura; el objetivo de este segundo paso es mejorar la imagen de forma que cuando se ingrese al sistema el usuario registrado tenga mayores posibilidades de éxito en el inicio de sección.

El tercer paso se tiene la segmentación; en este paso se elige el modelo de almacenamiento para previo reconocimiento de la imagen (Ordieres et al., 2006).

En el cuarto paso están los cálculos característicos que realiza el sistema al momento para el previo reconocimiento de la captura; en si es la comparación de la imagen principal con las imágenes almacenadas en el sistema.

Como último paso está el reconocimiento de la imagen, en este paso se da a conocer si el usuario fue encontrado en el sistema o no (Ordieres et al., 2006).

Todos estos pasos se observan en el diagrama de bloques de las etapas de un sistema de visión artificial que corresponde a la figura 8.

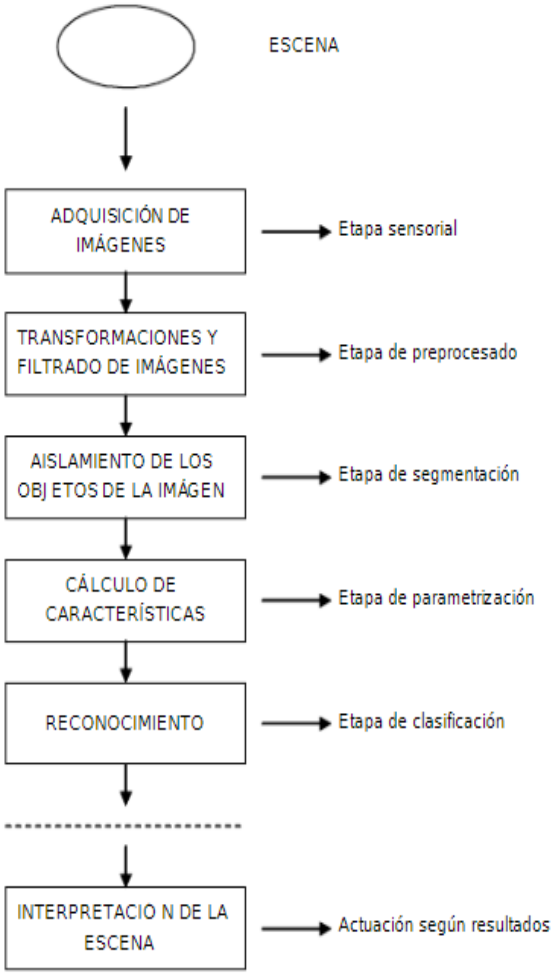


Figura 8. Diagrama de bloques de las etapas para el reconocimiento y almacenamiento de la captura del rostro del usuario. (Ordieres et al., 2006).

2.8.1. Patrones

El patrón es el punto o característica que puede ser reconocible de acuerdo a sus singularidades o cualidades (Miguel Ángel Vázquez López, 2014).

2.8.2. Reconocimientos de patrones

Reconocimientos de patrones en el campo de seguridad por medio de visión computarizada es la actividad de reconocer los objetos o expresiones que se muestran en la imagen de manera que se pueda clasificar la imagen como usuario registrado o no registrado (CEUPE, s.f.).

El funcionamiento del reconocimiento de patrones tiene cuatro etapas (CEUPE, s.f.), que son:

- **Adquisición de datos:** El responsable de este proceso es un sensor, este transforma las magnitudes físicas o químicas en magnitudes eléctricas.
  Las variables captadas son por Ej.: color, temperatura, intensidad lumínica, inclinación, etc.

- **Formulación de característica:** Esta formulación de características se usa posteriormente para la clasificación de objetos.

- **Selección de atributos:** Esta selección describe a los objetos.

- **Clasificación de objetos:** En esta parte se clasifican los objetos de acuerdo a los atributos que presentan, asignándolas en un grupo u otro, para esta parte se utilizan tecnologías de machine learning.

- **Clasificación supervisada:** Esta utiliza modelos ya prestablecidos para la clasificación objetos o comportamientos.

- **Clasificación no supervisada:** Este se encarga de identificar similitudes entre objetos.

- **Clasificación parcialmente supervisada:** Es un panorama donde existen modelos en algunas clases pero no en todas, entonces se entiende que la clasificación parcialmente supervisada es la combinación de la clasificación supervisada y no supervisada.

2.8.3. Similitud

La similitud que se presenta en la etapa de reconocimiento de la imagen.

Al hablar de similitud se refiere a los puntos parecidos de uno o más objetos que aparecen en la captura del rostro en el proceso de reconocimiento del usuario; es decir se evalúa la similitud que hay en la imagen principal con las imágenes almacenadas en el sistema tomando en cuenta principalmente las expresiones y objetos que se presentan en el rostro del usuario (Miguel Ángel Vázquez López, 2014).

2.8.4. Etapas requeridas para el reconocimiento de los patrones

En la visión artificial la técnica para el reconocimiento de patrones, está compuesta por distintos bloques que se ejecutan de manera constante sobre los patrones. En la figura 9 se observa la estructura de un sistema para el debido reconocimiento de los patrones (Miguel Ángel Vázquez López, 2014).

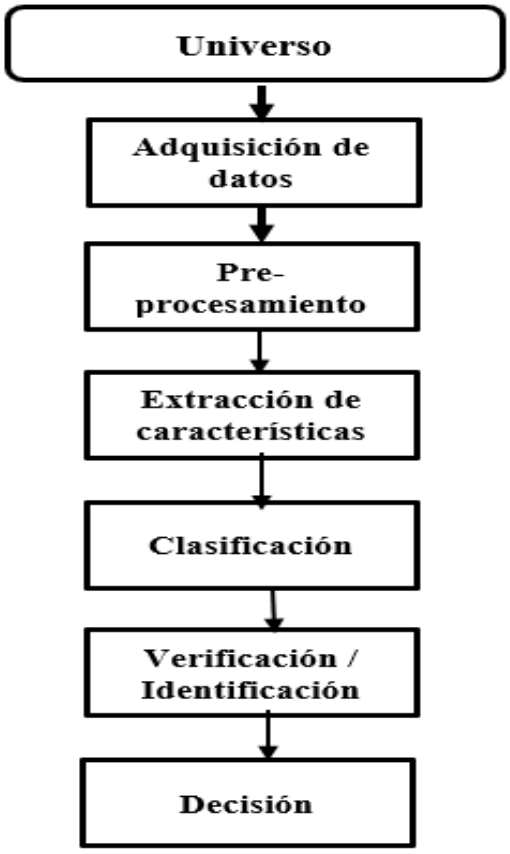


Figura 9. Etapas requeridas para el reconocimiento de los patrones

Etapa de adquisición de datos: En esta primera parte se registran las variables físicas de la imagen (Miguel Ángel Vázquez López, 2014).

Etapa de pre-procesamiento: En esta segunda parte se seleccionan los datos relevantes de la imagen (Cajas Idrovo & Viri Ávila, 2017).

Etapa de extracción de características: Una vez seleccionado los datos relevantes de la imagen se procede a guardar estos datos.

Etapa de clasificación: En esta parte se compara la imagen principal con las imágenes almacenadas del sistema para así clasificarlas.

Etapa de verificación e identificación: En esta parte se evalúa el resultado de la etapa de la clasificación de los datos de la imagen principal con los datos de las imágenes guardadas para así analizar si la imagen fue asignada a la categoría correcta.

Etapa de decisión: En esta parte se muestra el resultado de la comparación de la imagen.

2.9. Raspberry Pi

La Raspberry Pi es un mini ordenador que fue necesaria para la realización del proyecto de titulación; en este ordenador se realizó la programación del módulo de seguridad por medio del programa Python.

2.9.1. Partes de una Raspberry Pi 4

Para este proyecto de titulación se compró la Raspberry Pi 4 ya que es la que tiene mayor capacidad actualmente y por su gran cantidad de puertos que nos ayuda para la instalación de los materiales electrónicos que lleva el módulo de seguridad; a continuación se muestra en la figura 10 las partes que contiene una Raspberry Pi 4:

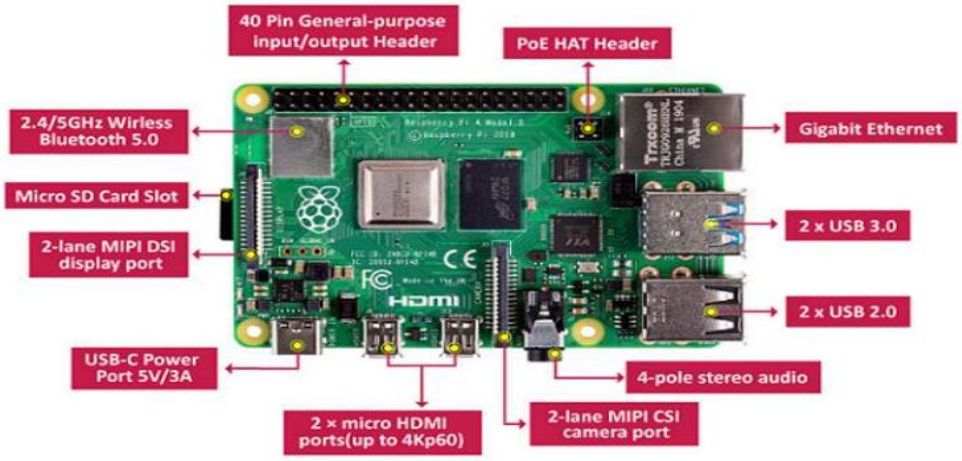


Figura 10. Partes de una Raspberry Pi 4. **(1+D Electrónica, s.f.)**

2.10. Python

Toda la programación del módulo de seguridad se realizó en el programa de alto nivel Python, se utilizó la versión 3.10. Python está disponible para Windows y Linux, una de las diferencia entre las dos es que en Windows se tiene que instalar Python y la mayor parte de las librerías se instalan automáticamente en cambio en Linux el programa Python ya está instalado pero se tiene que descargar e instalar todas las librerías necesarias para poder realizar la programación como es el caso de la librería OpenCV (Challenger Pérez et al., 2014).

2.11. OpenCV

OpenCV también llamado Open Source Computer Vision Library es una librería de código abierto contiene más de 2500 algoritmos; en proyectos se la utiliza para el

reconocimientos de objetos por medio de la visión artificial, las ventajas principales de esta librería es que funciona en muchas plataformas como es Windows, Linux y MacOs (PEÑA MERINO & JUIMY MILTON YEFF, 2011).

2.11.1. Manipulación de imágenes

Ya conociendo para que se utiliza principalmente la Liberia OpenCV en Python, se procedió hacer algunas pruebas de manipulación y edición de imagen para así garantizar la correcta instalación de la librería (CHRISTIAN FERNANDO SALAZAR ESPINOZA, 2016); En las figuras 11, 12 y 13 se muestran algunos resultados obtenidos en la programación en Python utilizando la librería OpenCV:

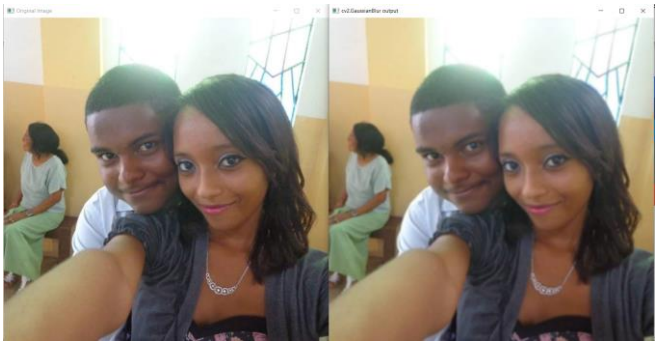


Figura 11. Imagen editada Liberia OpenCV con la función GasussianBlur.

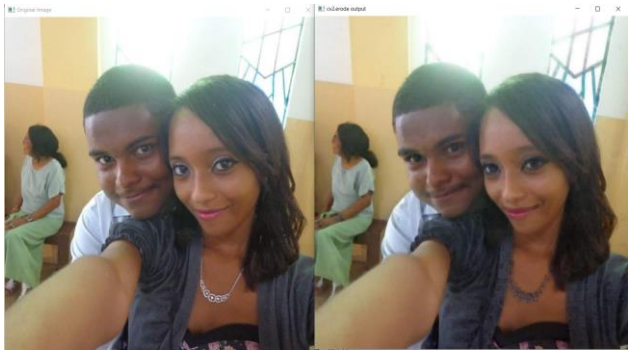


Figura 12. Imagen editada Liberia OpenCV con la función Erode.

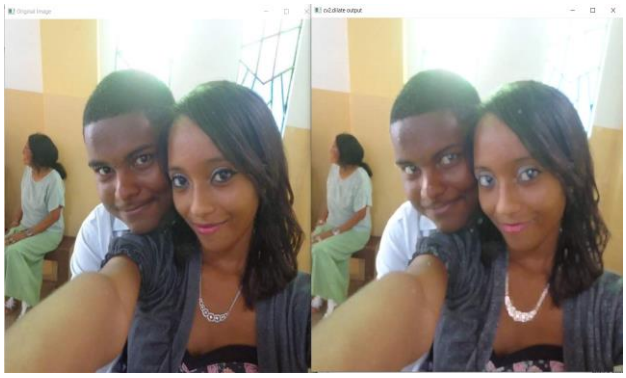


Figura 13. Imagen editada Liberia OpenCV con la función Dilate.

2.11.2. Tipos de cámaras para la obtención de las imágenes

Para la realización del módulo de seguridad se necesitó instalar una cámara para realizar la captura de rostro; en la Raspberry PI 4 hay tres tipos de cámaras que se pueden instalar y que garantiza el funcionamiento ideal (CHRISTIAN FERNANDO SALAZAR ESPINOZA, 2016):

Cámaras Web USB: La cámara Web USB es un tipo de cámara digital que se conecta al ordenador a través del puerto USB (CHRISTIAN FERNANDO SALAZAR ESPINOZA, 2016).

Cámaras IP: La cámara IP es un tipo de cámara que se utiliza principalmente para él envió de señal de video y audio a través del Internet utilizando un router Asymmetric Digital Subscriber Line o también conocido como ADSL.

Cámaras CSI: Este tipo de cámara se observa en los celulares móvil.

2.12. APP

La app o mejor conocida como aplicación de software; son diseñadas para utilizarse en los celulares; su principal objetivo es facilitar al usuario la realización de una o varias tareas en el día a día (Cobos, 2012).

2.12.1. IP Webcam

Como una función adicional que se implementó en el proyecto de titulación es utilizar la cámara del celular para hacer el registro y el reconocimiento del usuario, para hacer eso posible se instaló la aplicación IP Webcam que nos permite convertir la cámara del dispositivo móvil en una cámara IP con múltiples opciones (Khlebovich, s.f.).

3. DISEÑO

3.1. FUNCIONALIDAD

El presente proyecto se desarrolló por secciones las cuales se presentan a continuación:

La figura 14 muestra la pantalla principal del módulo:

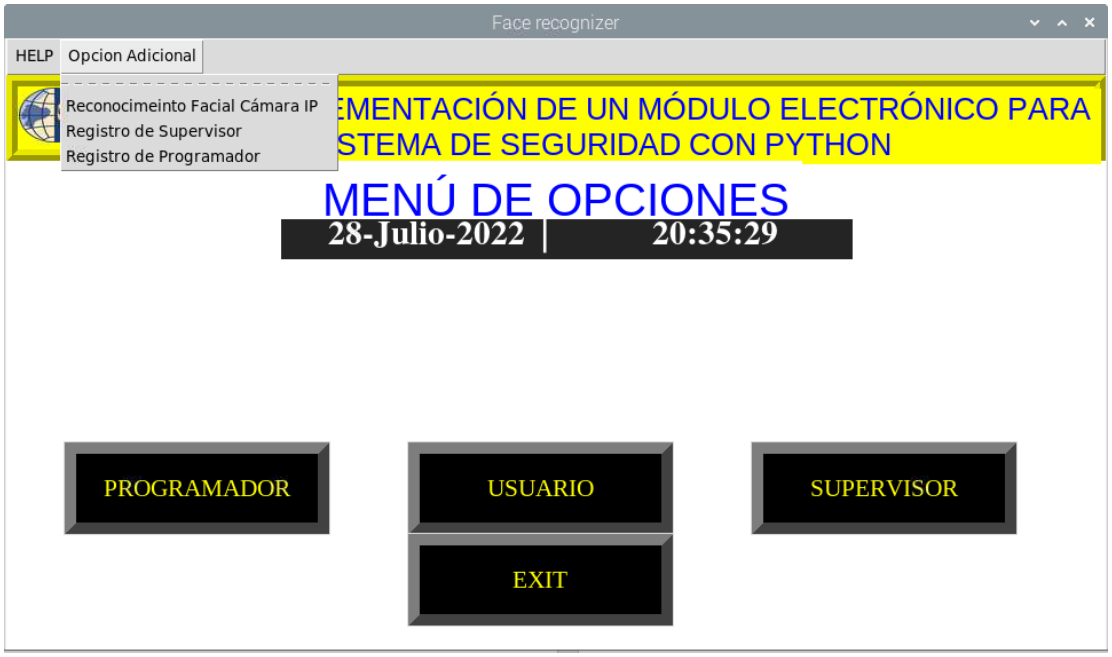


Figura 14. Menú principal

En la figura 15 se observa el caso si el usuario elije la opción Programador, el sistema le pide que ingrese la contraseña, si la contraseña es incorrecta lo devolverá a la pantalla principal, si la contraseña es correcta el sistema procederá abrir la cámara para verificar si el usuario que está iniciando sección está registrado en el sistema, si este es el caso se enciende el led rojo, se abre la ventana del modelo de entrenamiento LBPH y se envía una captura del usuario que ingresó a la opción programador y se envía un correo al supervisor (si el usuario no está registrado se enciende la alarma, el sistema captura el rostro de la persona y envía un correo al supervisor como el caso anterior, pero esta vez con el título “Usuario Sospechoso”).

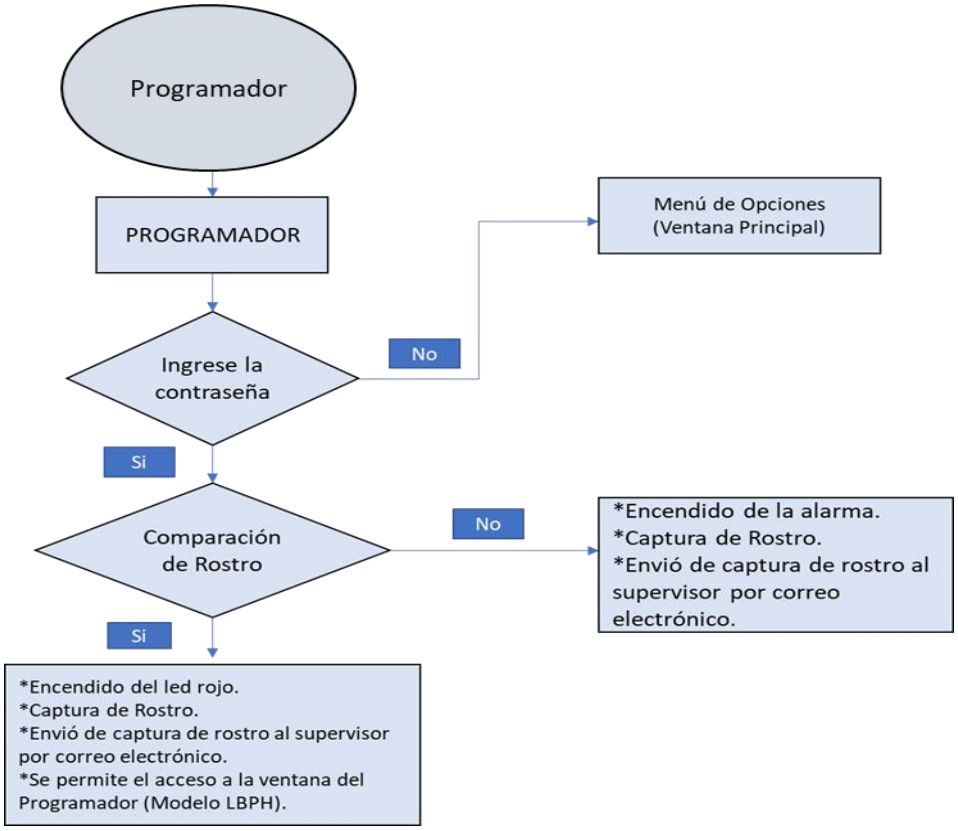


Figura 15. Diagrama de flujo de la opción Programador.

Si el usuario (programador), elije la opción Programador, se abre una nueva ventana de opciones con el modelo de entrenamiento LBPH y se cierra la ventana principal, esta nueva ventana se presenta las siguientes opciones a elegir, como se ve en la figura 16:

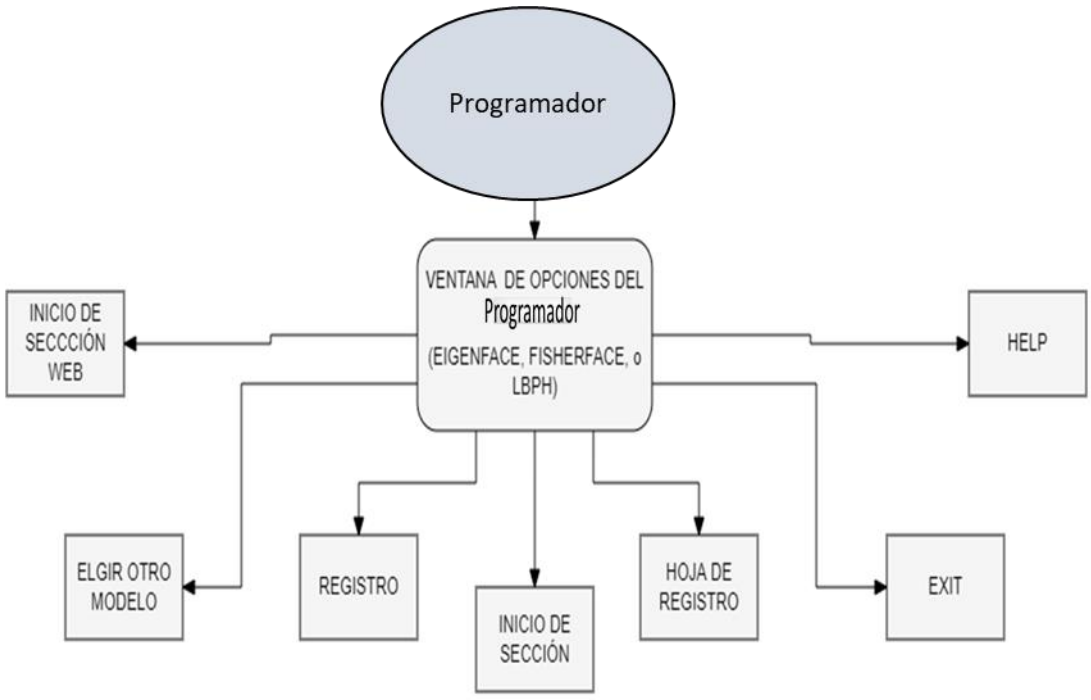


Figura 16. Diagrama de flujo de la ventana del Programador

En la figura 17 se muestra la opción Registro de Usuario, para que el programador o supervisor registre a las personas, se presiona el botón registro, donde el sistema le pide al Programador o Supervisor que ingrese la contraseña si la contraseña es correcta se abre la ventana de registro de usuario (donde el usuario tendrá que llenar los datos que pide el sistema para poder hacer el registro), una vez llenado eso, se presiona el botón registro (si falta un parámetro de llenar el sistema avisa al individuo que llene todo los parámetros, caso contrario el registro es exitoso), después se debe

de presionar el botón almacenamiento (este botón contiene el código de entrenamiento donde se almacena los datos del usuario).

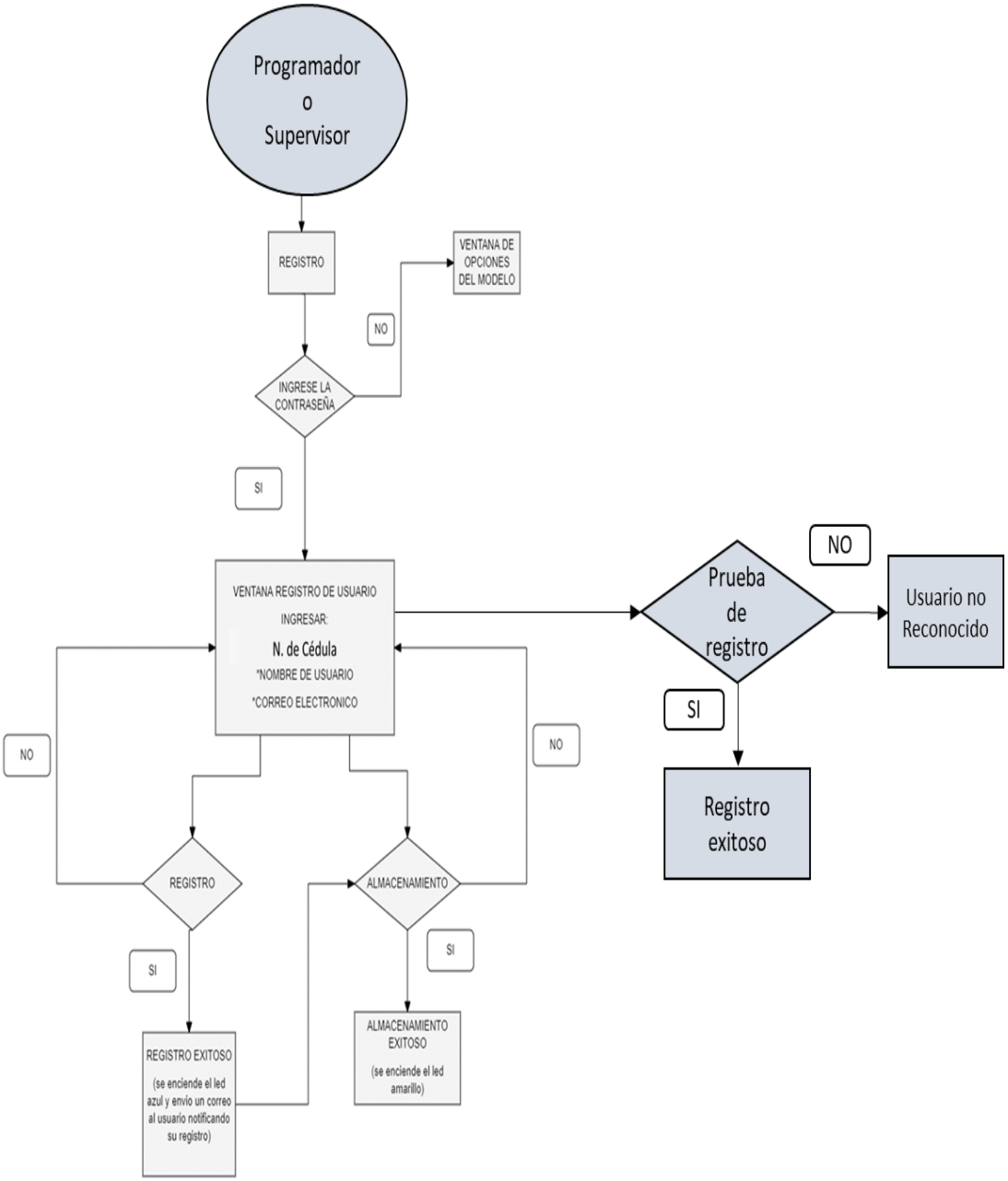


Figura 17. Diagrama de flujo del registro y almacenamiento del usuario

Si el usuario presiona el botón de inicio de sección se abre la ventana para llenar la asistencia (ventana de inicio de sección) donde se debe de ingresar el asunto (entrada o salida) y se presiona el botón asistencia manual, seguido de esto la cámara se activa y comienza a comparar el rostro de la persona con las que tiene en el registro, si el asunto o el rostro del usuario no consta en el sistema se toma captura del rostro y se envía un correo al administrador con el asunto “Usuario sospechoso” donde se adjunta la captura de rostro del individuo y se activa la alarma; caso contrario se toma captura del rostro del usuario y se envía un correo con el asunto “Ingreso+Asunto” donde se adjunta la captura de rostro, se activa el relé 1 (cerradura eléctrica) y el relé 2 (enciende el led rojo), como se observa en la figura 18.

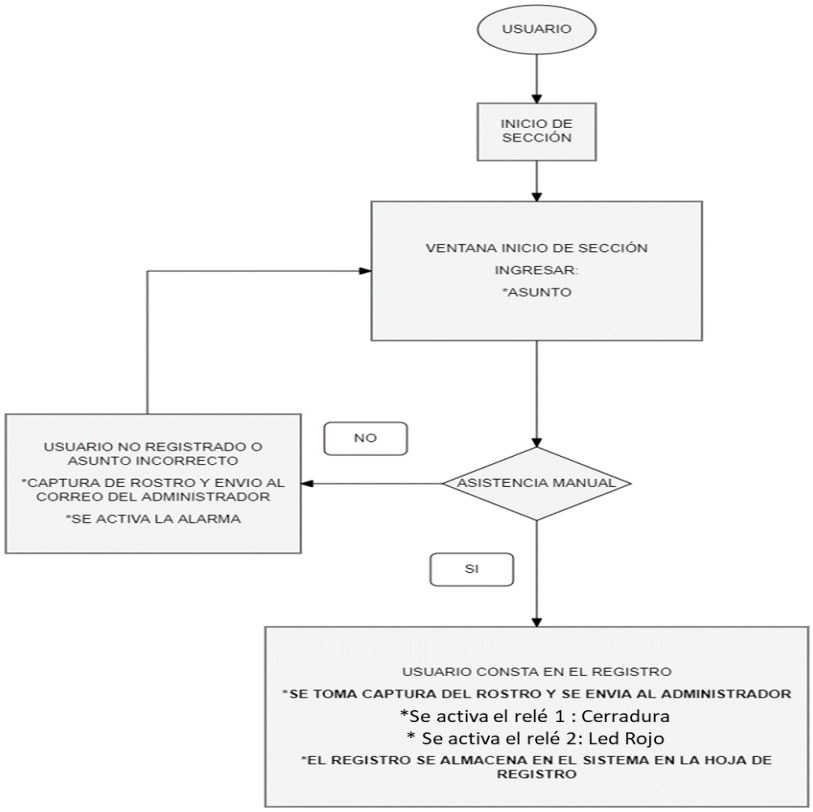


Figura 18. Diagrama de flujo de Inicio de sección

En la opción de hoja de registro, en esta parte se observa el registro de usuarios que inician sección con el módulo, como se observa en la figura 19, el programador o supervisor colocan el nombre del asunto (entrada o salida), luego procede hacer clip en el botón registro, si el nombre del asunto es incorrecto el sistema pide que ingrese nuevamente el nombre del asunto, caso contrario se muestra la hoja de registro de los usuarios que iniciaron sección con el módulo de seguridad.

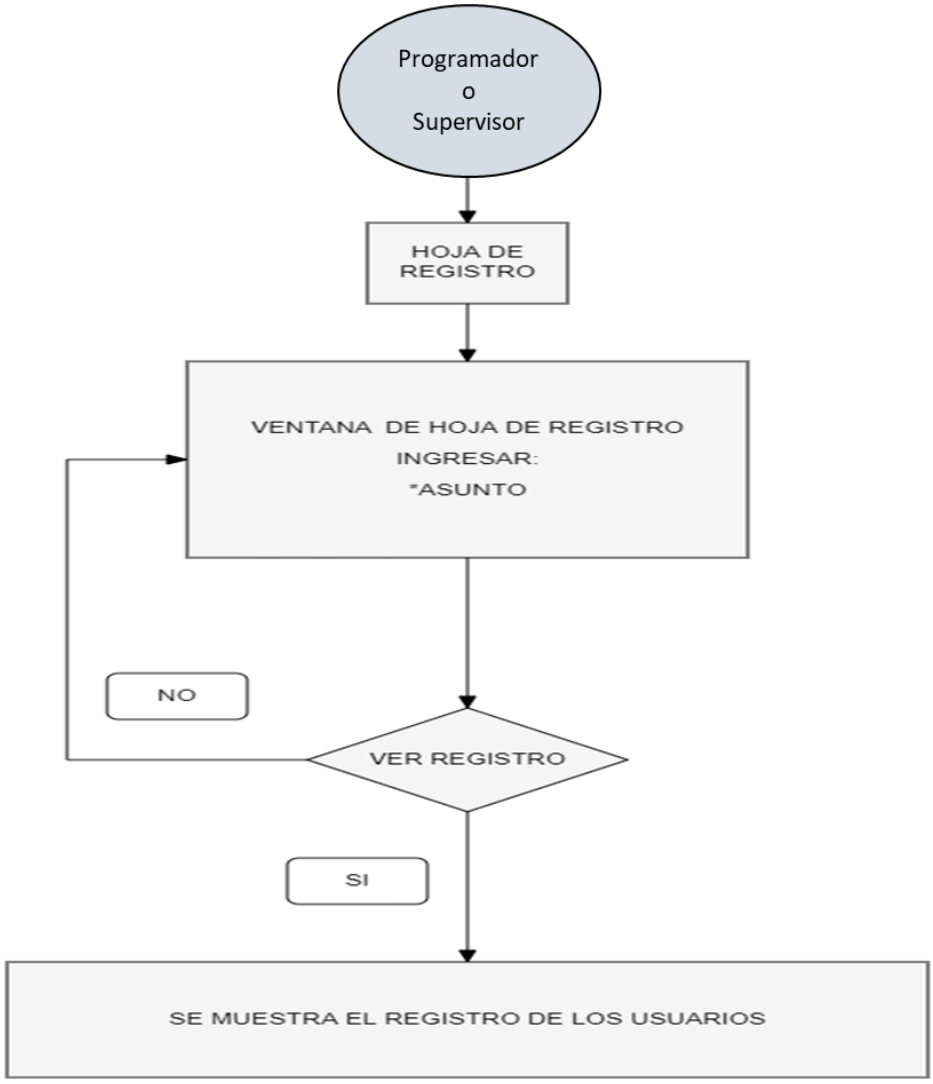


Figura 19. Diagrama de flujo de la opción hoja de registro.

Para el inicio de Cámara IP es lo mismo que en los puntos anteriores ya mencionado solo que en este punto se necesita de una app externa llamada IP WEBCAM que ayuda activar la cámara IP del celular, en las opciones para el ingreso, se pide al usuario ingresar la porción de HOST de su dirección IP e ingresar su nombre de usuario; según el botón que se elija, se compila el proceso que se observa en la figura 20.

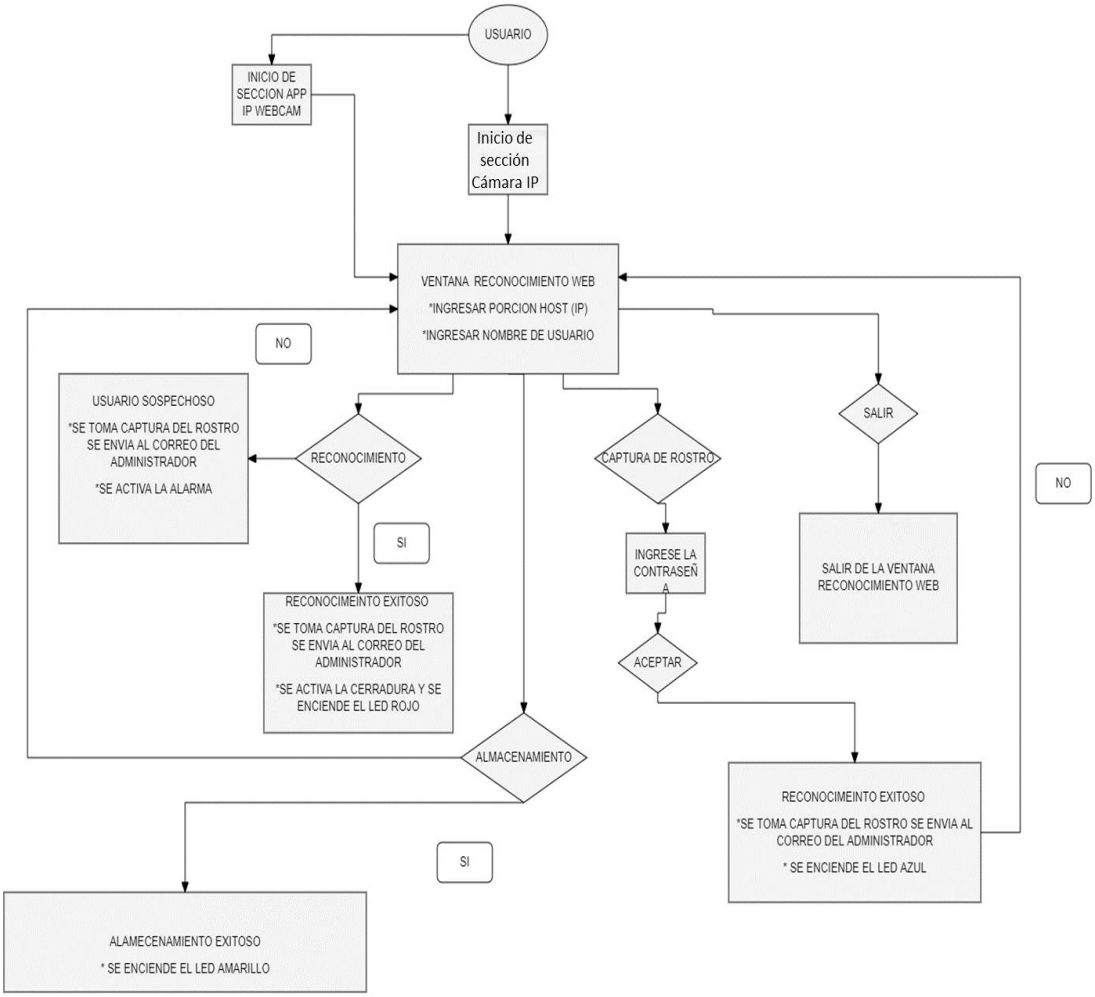


Figura 20. Diagrama de flujo del inicio de sección Cámara IP.

Como se observa en la figura 21 se tiene la opción Help que ayuda a revisar el nombre de los autores, contactos, también se logra apreciar la opción de cambiar la contraseña y por último, se tiene la opción que cierra todas las ventanas.

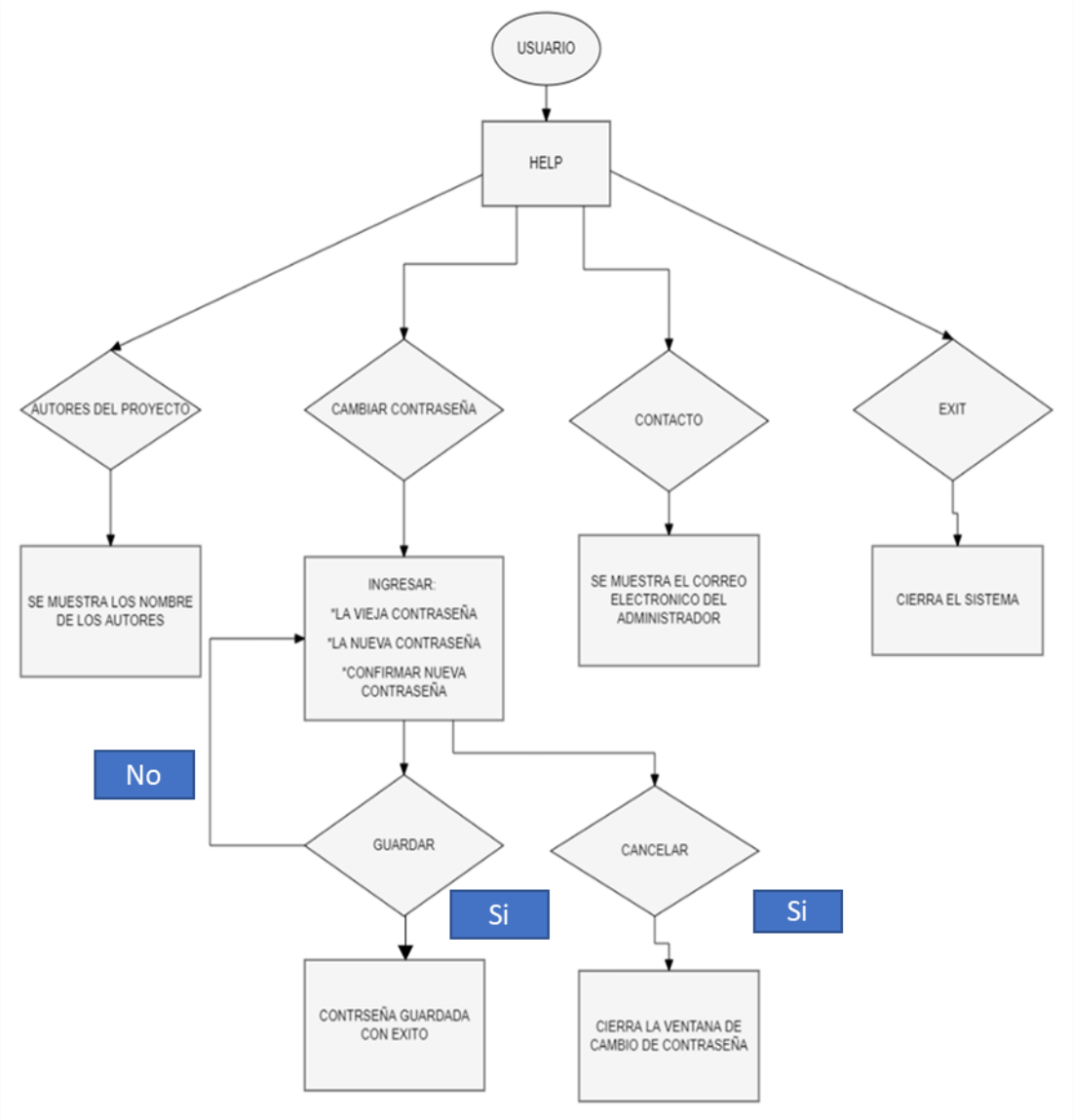


Figura 21. Diagrama de flujo de la opción HELP

En la figura 14, se tiene la pantalla principal, si la persona elije la opción USUARIO se abre la ventana de “Menú de Inicio de Sesión”, el cual dentro de la ventana contiene el botón para iniciar sección este proceso se muestra en la figura 18.

Tomando en cuenta la figura 14, si se elige la opción supervisor el sistema procede a realizar el siguiente proceso de la figura 22:

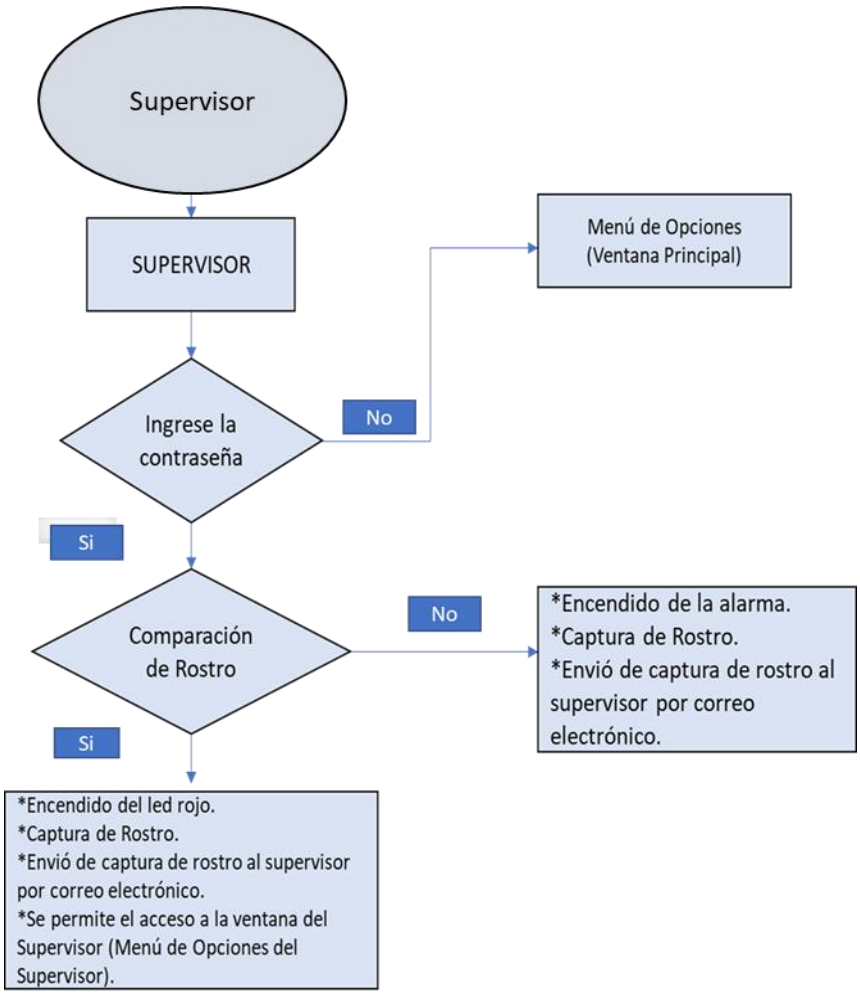


Figura 22. Diagrama de flujo botón Supervisor

El menú de opciones del supervisor contiene lo mismo que la ventana del programador, solo que en la ventana del supervisor se quita las botoneras de los modelos FisherFaces y EigenFaces.

En este último diagrama de la figura 23, se observa las diferentes maneras de cerrar el sistema:

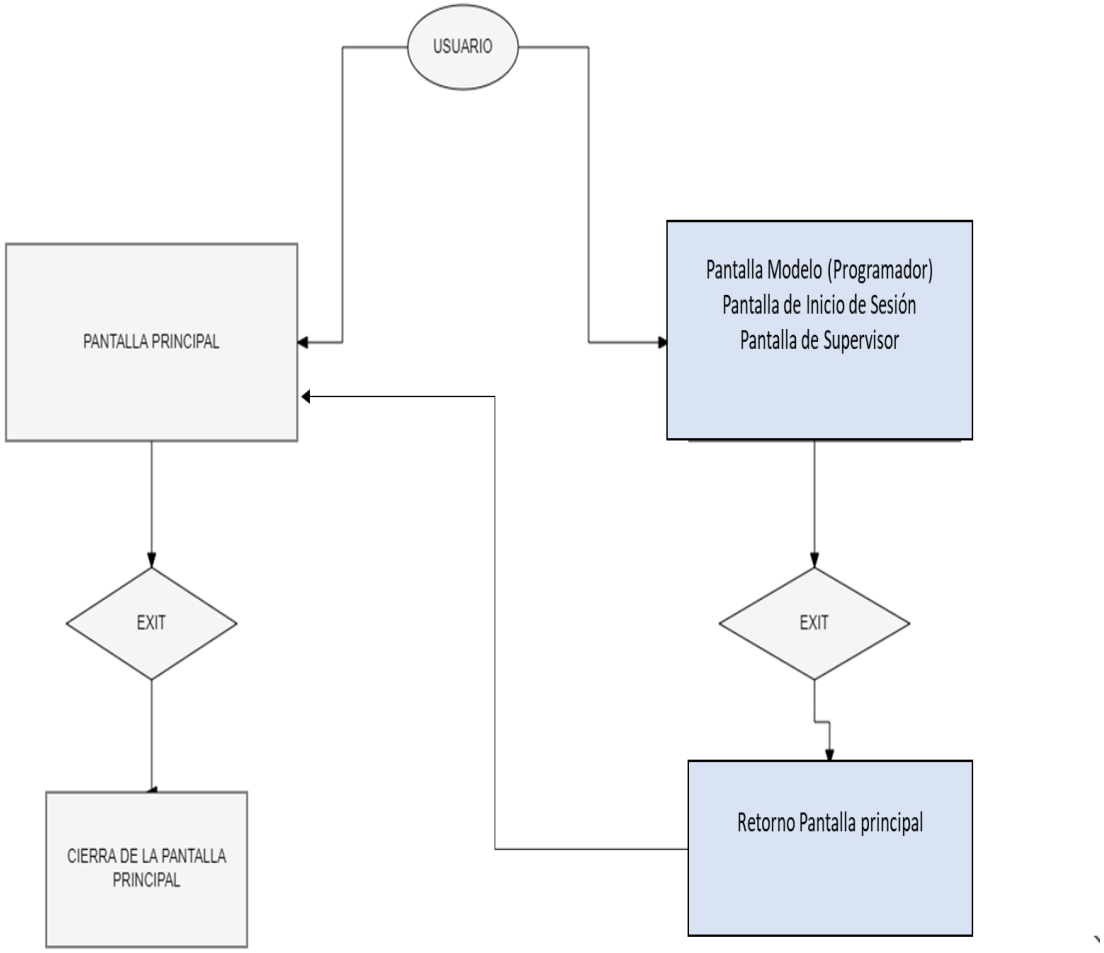


Figura 23. Diagrama de flujo de la salida del sistema

3.2. Diseño del módulo

Para hacer el armazón o armadura del módulo de seguridad, se verifica los elementos eléctricos y electrónicos que se van a montar al módulo. Con la información recolectada se realizó el diseño del módulo de seguridad, como se observa en el anexo C.

3.3. Diseño de planos eléctricos y diseño de la interfaz gráfica

Se realizó el diseño de los planos eléctricos mediante el software Fritzing, en el cual se encuentra el circuito eléctrico y el diseño de la interfaz, esta última se realizó en PowerPoint, como se observa en el anexo D.

3.4. Diseño del diagrama de bloque y eléctrico del módulo previo a la programación.

El diagrama de bloque del esquemático de la figura 24 y 25, describe los elementos más importantes para realizar la programación que son: Laptop, celular, Raspberry Pi 4, fuentes de alimentación, módulo Relé, pantalla táctil, Chapa Solenoide, ventilador, cable ethernet, zumbador, parlante y diodos led y en la figura 26 se observa el diagrama eléctrico del módulo de seguridad .

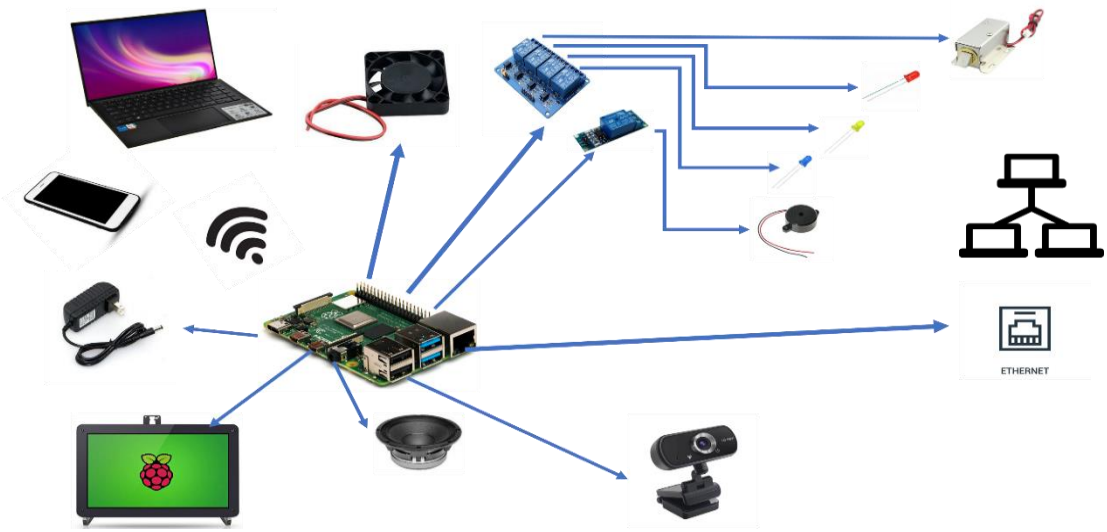


Figura 24. Esquema del módulo.

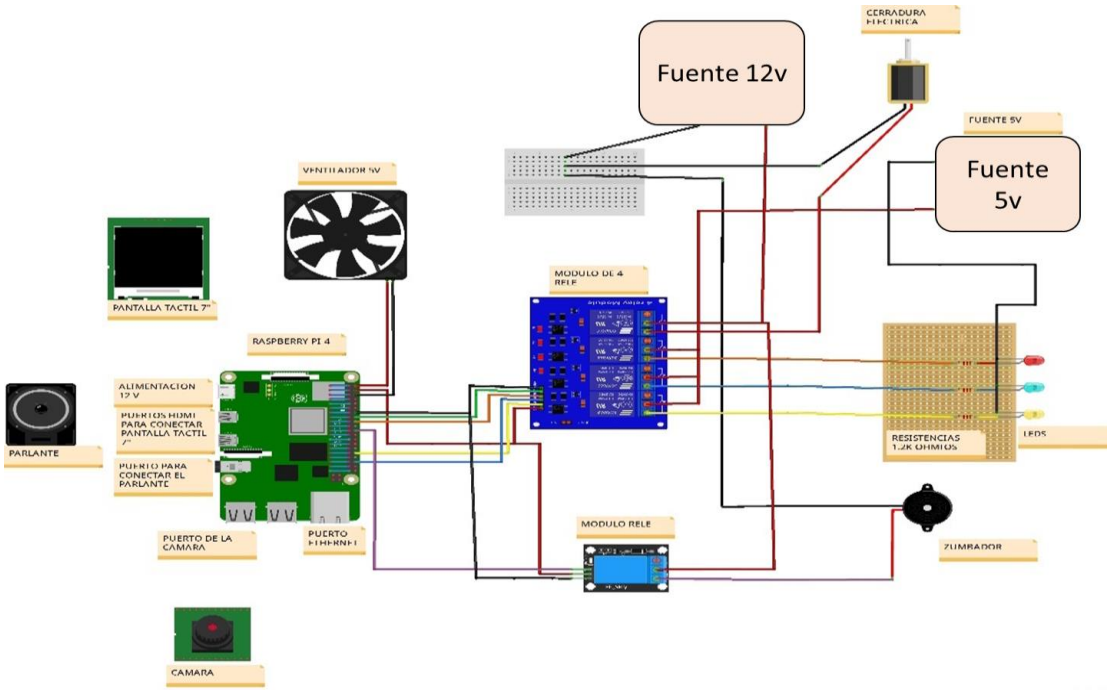


Figura 25. Esquema Eléctrico

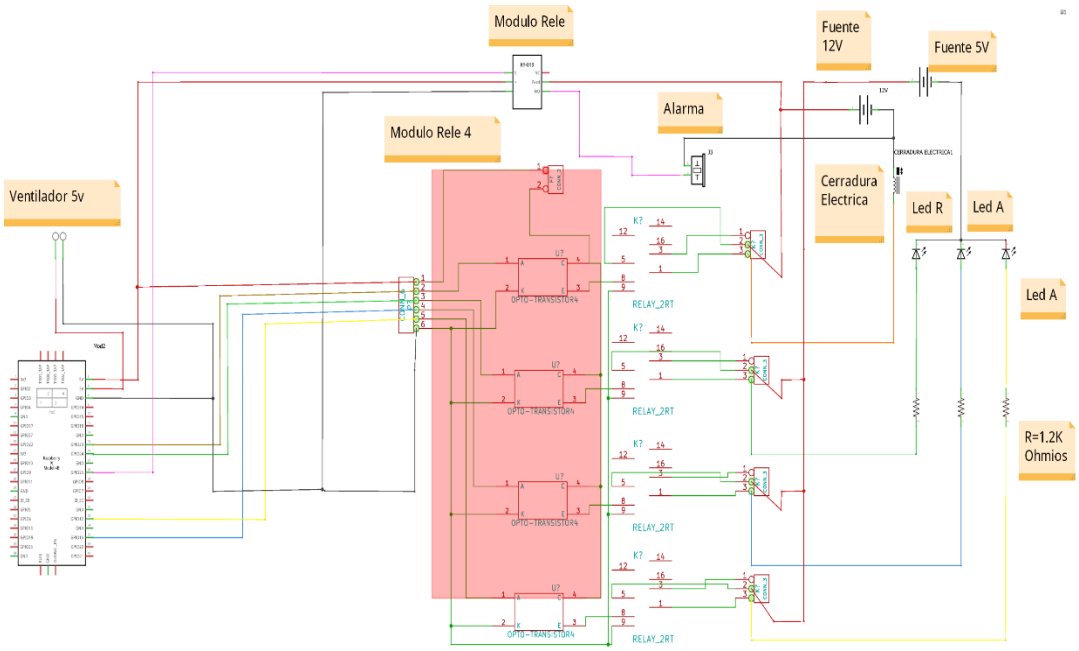


Figura 26. Diagrama electrónico

3.5. Configuración de la comunicación Router- Raspberry Pi

La programación fue realizada en Python para la Raspberry Pi, a continuación se detalla lo más importante de la programación. (Ver Anexo E).

4. IMPLEMENTACIÓN

4.1 Diseño del armazón

Se procedió a realizar el listado de componentes a colocar en el módulo para crear el respectivo diseño del armazón con la ayuda de la herramienta AutoCAD (el listado de componentes se observa en el anexo B), luego se verificó el diseño para evitar errores o falta de espacio al realizar el montaje de los componentes en el módulo como se ve en el anexo C.

4.2 Montaje de componentes en el armazón

Previamente determinadas las medidas que tiene el armazón y los componentes a colocar, se procede con la compra de los materiales, se tomó en cuenta la dimensión de la pantalla táctil que tiene las siguientes medidas: 16cm ancho x 10cm de alto, en la figura 27 se observa módulo de seguridad con los parlantes ya colocados.

Figura 27. Previo al montaje de la pantalla táctil en el armazón

En la figura 28 se observa los módulos relé ya instalados dentro del módulo de seguridad.

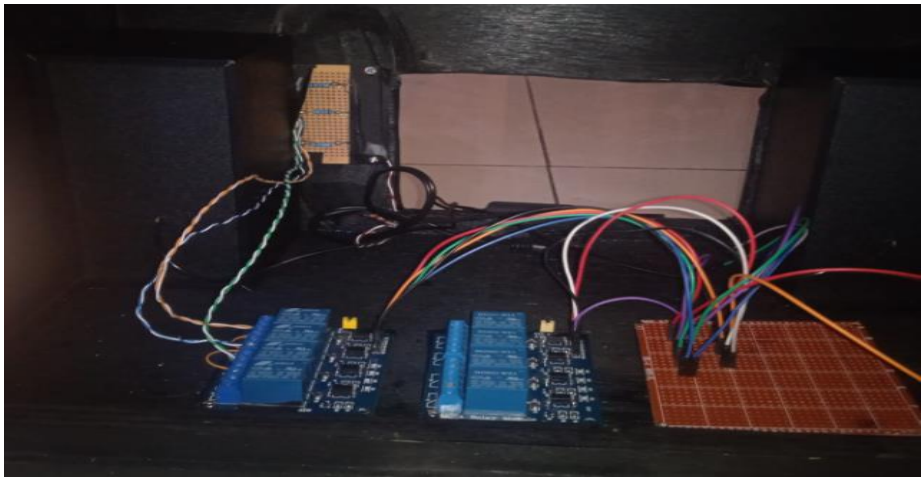


Figura 28. Módulos relé ya instalados

Como se observa en la figura 29, el armazón cuenta con la cerradura eléctrica que sirve para simular el acceso a un domicilio, empresa o lugar de estudio .

Figura 29. Armazón con la cerradura

Se colocó la pantalla táctil en el armazón, la cual se puede visualizar en la figura 30.

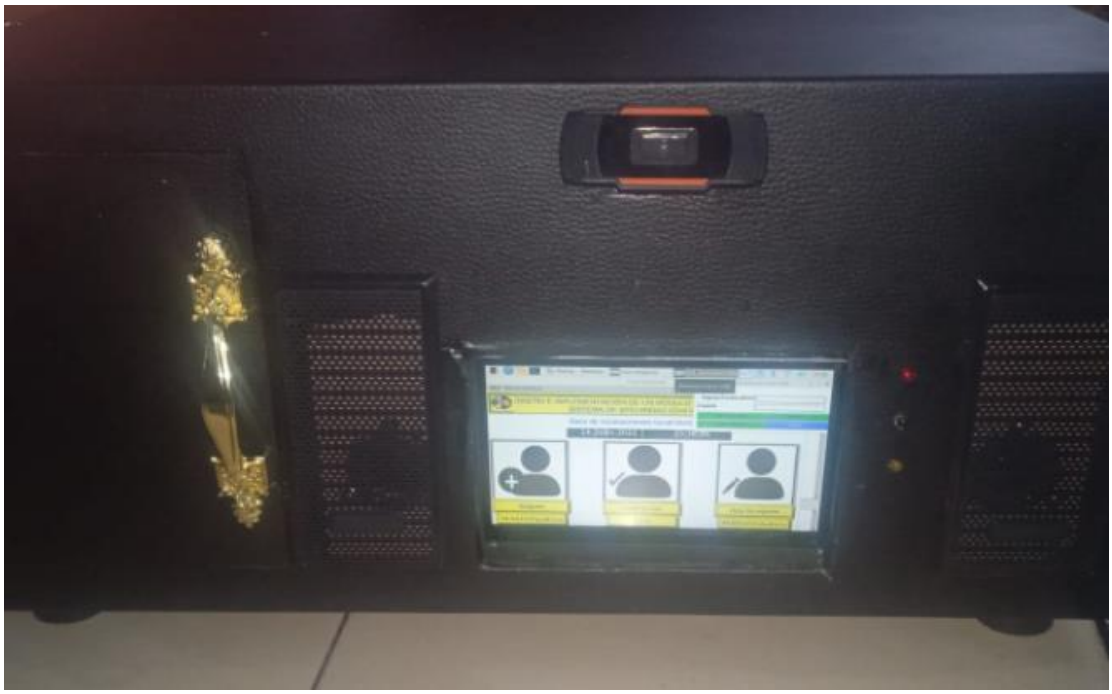


Figura 30. Pantalla táctil ya instalada en el armazón

5. ANÁLISIS DE RESULTADOS

5.1 Análisis del proyecto

Se verificó que el proyecto realice los procesos adecuadamente como lo son: registro, almacenamiento e inicio de sesión; además de comprobar el correcto funcionamiento de las botoneras que se encuentran en la interfaz; que se observa en la pantalla táctil en la Figura 30.

5.1.1 Comunicación a la Raspberry Pi

Para la primera comunicación a la Raspberry Pi fue necesario comunicar por vía conexión ethernet hacia el router para asignar una IP a la Raspberry Pi y así mediante el programa VNC Viewer poder visualizar la pantalla de la Raspberry y proceder a programar, en la figura 31 se observa la comunicación entre la módulo de seguridad y la laptop .

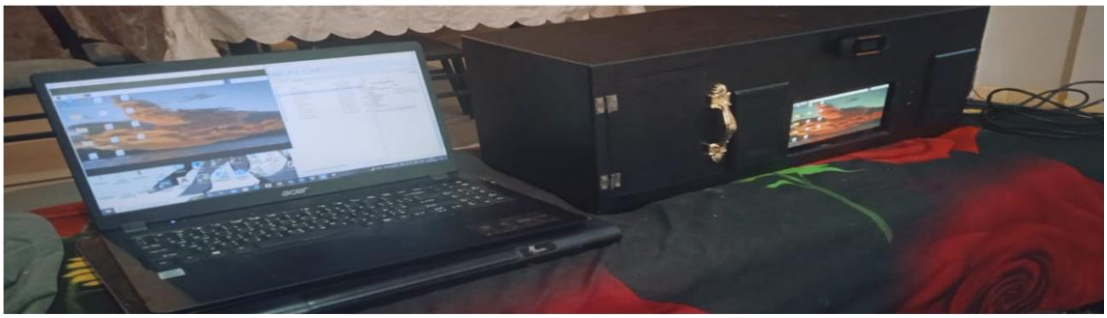


Figura 31. Comunicación entre dispositivos

5.1.2 Pruebas realizadas

En la figura 32 se realizó verificación del correcto funcionamiento del módulo para eso se confirma el funcionamiento adecuado del programa realizado en la lenguaje de alto nivel de programación Python.

Figura 32. Verificación del correcto funcionamiento del módulo

Cuando se procede hacer el registro; si es correcto se activa el relé 3 (enciende el led azul y después se apaga) y se visualiza en la pantalla un mensaje satisfactorio donde se muestra el número de la inscripción y el nombre del usuario en la interfaz gráfica de la pantalla táctil y se le envía un correo electrónico al usuario registrado confirmándole su registro exitoso, como se observa en la figura 33, 34 y 35.

Figura 33. Mensaje del registro en el módulo.

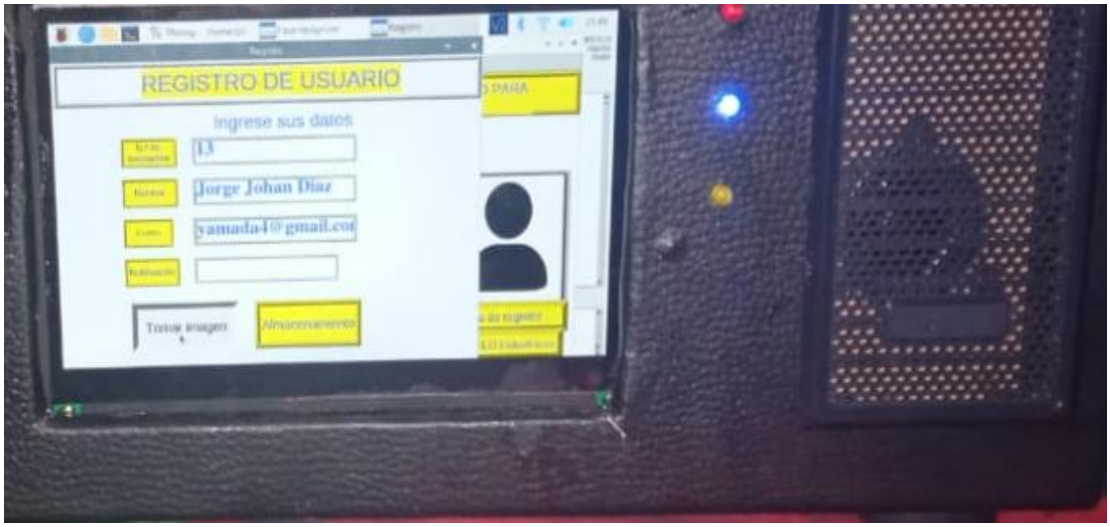


Figura 34. Encendido del led azul.

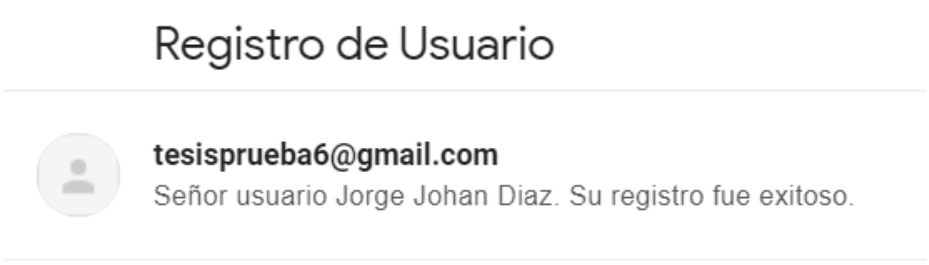


Figura 35. Mensaje del registro en el correo electrónico.

En la figura 36; la interfaz gráfica de la pantalla se observa un botón que dice almacenamiento al presionar este botón almacena todos los datos del usuario en un archivo .ylm; en la pantalla también se visualiza un mensaje de satisfacción que dice “modelo almacenado” y el relé 4 se activa (enciende el led color amarillo), como se observa en la figura 37.

Figura 36. Mensaje de almacenamiento exitoso

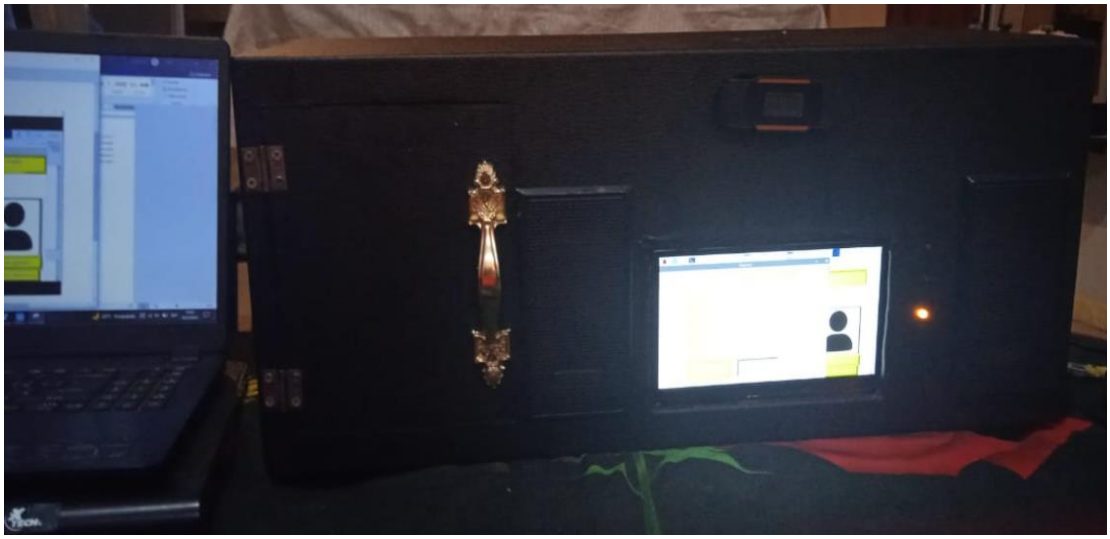


Figura 37. Encendido del led amarillo.

En la parte de inicio de sección, se pide al usuario ingresar el asunto; si el asunto es correcto, se toma captura del rostro de la persona que inicio sección y esta captura se envía por correo electrónico al administrador como asunto “Ingreso de Usuario” se activa el relé 1 (activa la cerradura eléctrica) y también se activa el relé 2 (enciende el led rojo) y se muestra el registro de inicio de sección para ese usuario; este proceso se observa en las figuras: 38, 39 y 40.

Figura 38. Correo electrónico de Inicio de sección

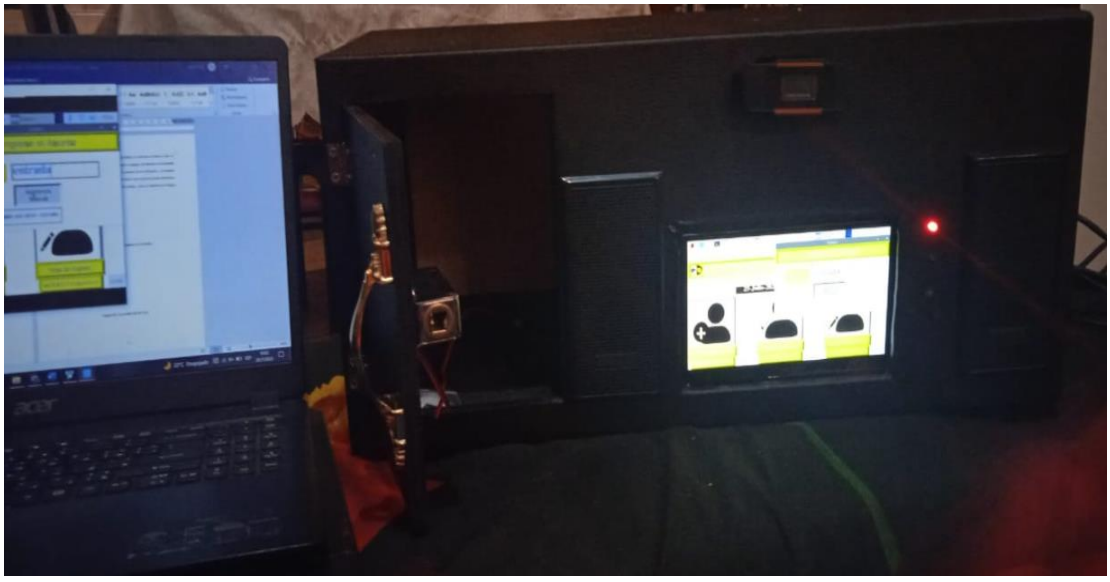


Figura 39. Activación de la cerradura eléctrica y encendió del led rojo

Figura 40. Mensaje de inicio de sección del usuario.

Al contrario si el registro es incorrecto, se toma captura del rostro de la persona que inicio sección y esta captura se envía por correo electrónico al administrador como asunto “Usuario Sospechoso”, como se observa en la figura 41; y se activa el relé 5 (enciende la alarma).

Figura 41. Correo electrónico de Usuario Sospechoso

6. CONCLUSIONES

- Para la implementación del módulo de seguridad se usó los respectivos materiales como son: leds, cerradura eléctrica, pantalla táctil, zumbador y el módulo relé; pero es importante conocer el funcionamiento del módulo relé, que su estado es normalmente abierto o normalmente cerrado y según como actúe el módulo relé se procede a programar para su correcto funcionamiento en el módulo de seguridad.

- Al momento de programar en Python los proceso a realizar el módulo de seguridad, se los dividió por secciones, teniendo así los procesos: captura, almacenamiento y reconocimiento del rostro de la persona; todo esto se realizó de esa manera, para llamar a las variables con facilidad al momento de la programación de las respectivas ventanas: del programador, supervisor y la del usuario.

- Se cumplió con el objetivo general y los objetivos específicos. Por lo cual, el Módulo Electrónico para Sistema de Seguridad con Python, queda funcionando correctamente para el uso practico de los estuantes de la Universidad Politécnica Salesiana (Sede Guayaquil) de la carrera de Ingeniería Electrónica.

7. RECOMENDACIONES

- Como el módulo no cuenta con una batería se recomienda conectarlo a un UPS (Uninterruptable Power Supply).

- El proceso de registro tarda 10 minutos, en el transcurso de ese tiempo no presionar otros botones ya que el sistema se congela.

- El registro de usuario por cámara de celular pone lento al sistema, por eso es recomendable hacerlo por la cámara del módulo de seguridad, pero si desea registrar por el método cámara IP del celular, tendrá que esperar máximo 15 minutos.

- Se recomienda al usuario si hace algún proceso con la cámara IP con el módulo, estar atento en colocar correctamente la porción de HOST de la IP del celular .

- En caso de tener algún inconveniente con la red WIFI, como otra opción de tener conectividad a Internet, está la conexión Ethernet.

8. PROYECTOS DE INVESTIGACIÓN VINCULADOS

Diseño e implementación de un sistema de seguridad para un automóvil con autenticación por reconocimiento facial utilizando técnicas de visión artificial.

Autor: Christian Fernando Salazar Espinoza

Universidad: Escuela Superior Politécnica de Chimborazo.

Sistema de Reconocimiento Facial Mediante Técnicas de Visión Tridimensional.

Autor: Ing. Miguel Ángel Vázquez López

Universidad: Centro de Investigaciones en Óptica, A.C.

Sistema de detección y conteo de vehículos utilizando visión artificial.

Autores: Peña Merino, Juimy Milton Yeff

Universidad: Universidad Nacional De Piura.

Diseño de un sistema de autenticación biométrica basado en reconocimiento facial.

Autor: Sandra Elizabeth Garrochamba Sánchez

Universidad: Universidad Nacional De Loja.

Diseño e implementación de un sistema de seguridad vehicular mediante reconocimiento facial a través de visión artificial.

Autor: Marco Vinicio Caja Idrovo, Pablo Andrés Viri Ávila

Universidad: Universidad Politécnica Salesiana.

Desarrollo de un control de acceso a través del reconocimiento Facial utilizando Raspberry Pi y una aplicación Android.

Autor: Catherine Rossana Rivas Ortiz

Universidad: Universidad Politecnica Salesiana.

9. REFERENCIAS BIBLIOGRÁFICAS

1+D Electrónica. (s.f.). *1+D Electrónica*. (T. R. B-4GB, Productor) Recuperado el 18 de Junio de 2022, de 1+D Electrónica: https://www.didacticaselectronicas.com/index.php/sistemas-de-desarrollo/raspberry/tarjetas-raspberry/tarjeta-raspberry-pi-4-b-4gb-pi4-tarjetas-de-desarrollo-sistemas-de-desarrollo-minipc-mini-computadores-raspberry-pi-4-modelo-b-de-4gb-detail

ACERVO LIMA. (s.f.). *ACERVO LIMA*. (P. |. OBJETOS, Productor) Recuperado el 18 de Junio de 2022, de ACERVO LIMA: https://es.acervolima.com/python-cascadas-de-haar-para-la-deteccion-de-objetos/

CEUPE. (s.f.). *CEUPE Magtazine*. Recuperado el 18 de Junio de 2022, de https://www.ceupe.com/blog/reconocimiento-de-patrones.html?dt=1655535559535

Cobos, T. L. (2012). *virtualis*. Recuperado el 9 de Julio de 2022, de virtualis: https://www.revistavirtualis.mx/index.php/virtualis/article/view/64

CONTAVAL. (18 de Febrero de 2016). *CONTAVAL*. Recuperado el 18 de Junio de 2022, de CONTAVAL: https://www.contaval.es/que-es-la-vision-artificial-y-para-que-sirve/

Khlebovich, P. (s.f.). *Pavel Khlebovich*. Recuperado el 9 de Julio de 2022, de Googleplay: https://play.google.com/store/apps/details?id=com.pas.webcam&hl=es_EC&gl=US

Maginvent. (s.f.). *Maginvent*. Recuperado el 18 de Junio de 2022, de Maginvent: https://www.maginvent.org/articles/pidht/pidtoot/Reconocimiento_Patrones.html#:~:text=Reconocimiento%20de%20Patrones%20es%20una,o%20mas%20clases%20de%20categor%C3%ADas.

Pardo, I. C. (s.f.). *Visión por computadora*. Recuperado el 18 de Junio de 2022, de Visión por computadora: https://carlosjuliopardoblog.wordpress.com/2017/05/12/filtros-haar-deteccion-de-rostros/

Programador Clic. (s.f.). *programador clic*. Recuperado el 18 de Junio de 2022, de https://programmerclick.com/article/95011695978/

¿Qué es una cerradura antibumping? ¿Existen otras? -canalHOGAR. (n.d.). Retrieved June 18, 2022, from https://www.hogar.mapfre.es/hogar/seguridad-en-casa/antibumping-y-otras-cerraduras-de-seguridad/

Bharath, A. A. (Anil A., & Petrou, M. (2008). *Next generation artificial vision systems : reverse engineering the human visual system*. 438. https://books.google.com/books/about/Next_Generation_Artificial_Vision_System.html?hl=es&id=TfAeAQAAIAAJ

Cajas Idrovo, M. V., & Viri Ávila, P. A. (2017). Diseño e implementacion de un istema de seguridad vehicular mediante reconocimiento facial a traves de visión artificial. In *Universidad Politécnica Salesiana Sede Cuenca*. https://dspace.ups.edu.ec/bitstream/123456789/13566/1/UPS-CT006920.pdf

Cardona, E., Steeven, D., Ospina, V., & Mateo, D. (2019). Raspberry pi : la tecnología reducida en placa. *Tecnología En Sistemas de Información*, 1–13. https://repository.usc.edu.co/bitstream/handle/20.500.12421/4250/RASPBERRY PI.pdf?sequence=3&isAllowed=y

Carlos H. Esparza Franco, Christian Tarazona Ospina, Esdras E. Sanabria Cuevas, & Daniel A. Velazco Capacho. (2017). RECONOCIMIENTO FACIAL BASADO EN EIGENFACES, LBHP Y FISHERFACES EN LA BEAGLEBOARD-xM. *Revista Colombiana De Tecnologias De Avanzada (Rcta)*, *2*(26). https://doi.org/10.24054/16927257.v26.n26.2015.2387

Challenger Pérez, I., Díaz Ricardo, Y., & Becerra García, R. (2014). El lenguaje de programación Python/The programming language Python. *Revista Ciencias Holguín*, *20*, 1–13. http://www.linuxjournal.com/article/2959

CHRISTIAN FERNANDO SALAZAR ESPINOZA. (2016). *Diseño e Implementación de un Sistema de Seguridad para un Automóvil con Autenticación por Reconocimiento Facial Utilizando Técnicas de Visión Artificia*. 122. http://www.dspace.espol.edu.ec/xmlui/handle/123456789/3550?locale-attribute=en

Fontecha Zabaleta, A. J. (2017). PYTHON EN LA SEGURIDAD INFORMÁTICA. *Universidad Piloto de Colombia*, 1–10. https://bit.ly/2CQx5vZ

Gibrán García C. (n.d.). *Técnicas de reconocimiento facial: vista general - Visión artificial %*. Retrieved June 18, 2022, from https://naps.com.mx/blog/tecnicas-reconocimiento-facial/

Fiscalía General del Estado del Ecuador. (2021). *Fiscalía General del Estado | Cifras de robos*. 8-09-2021. https://www.fiscalia.gob.ec/estadisticas-de-robos/

Juan Vicente Martínez Pérez, & Jordi Linares Pellicer. (2012). Sistema de reconocimiento Facial y Realidad Aumentada Para Dispositivos Móviles. *3Ciencias*, 1–10. https://riunet.upv.es/bitstream/handle/10251/34375/Mart%EDnez P%E9rez, J. V. - Sistema de reconocimiento facial.pdf?sequence=1

MARISOL QUISPE ADUVIRI. (2013). RECONOCIMIENTO DE GESTOS PARA LA INTERACCIÓN POR COMPUTADOR, CON REALIDAD AUMENTADA. *Pontificia Universidad Catolica Del Peru*, *8*(33), 44. https://repositorio.umsa.bo/bitstream/handle/123456789/7814/T.2768.pdf?sequence=1&isAllowed=y

Miguel Ángel Vázquez López. (2014). *Sistema de Reconocimiento Facial Mediante Técnicas de Visión Tridimensional*. https://cio.repositorioinstitucional.mx/jspui/bitstream/1002/436/1/15950.pdf

Ordieres, J., Limas, M., Ascacibar, F. J., Alba-Elías, F., González-Marcos, A., Pernía-Espinoza, A., & Vergara, E. (2006). *Técnicas y algoritmos básicos de visión artificial Recurso electrónico - En línea* (Issue December 2016). https://www.researchgate.net/publication/231521316_Tecnicas_y_algoritmos_basicos_de_vision_artificial_Recurso_electronico_-_En_linea

Ottado, G. (2010). Reconocimiento de caras : Eigenfaces y Fisherfaces. *Universidad de La República Uruguay*, 1–15. https://eva.fing.edu.uy/file.php/514/ARCHIVO/2010/TrabajosFinales2010/informe_final_ottado.pdf

PEÑA MERINO, & JUIMY MILTON YEFF. (2011). *SISTEMA DE DETECCIÓN Y CONTEO DE VEHÍCULOS UTILIZANDO VISIÓN ARTIFICIAL*. 1–110. https://repositorio.unp.edu.pe/bitstream/handle/UNP/1311/CIE-PEÑ-MER-17.pdf?sequence=1&isAllowed=y

Preguntas frecuentes sobre Ring Video Doorbell y las cámaras de seguridad – Ring Help. (n.d.). Retrieved June 18, 2022, from https://support.ring.com/hc/es/articles/115004666066-Preguntas-frecuentes-sobre-Ring-Video-Doorbell-y-las-cámaras-de-seguridad

Romo Marín, D. F. (2020). *Software de Reconocimiento Óptico de caracteres*. 1–73. https://rinacional.tecnm.mx/bitstream/TecNM/2620/1/SOFTWARE DE RECONOCIMIENTO ÓPTICO DE CARACTERES.pdf

SANDRA ELIZABETH GARROCHAMBA SÁNCHEZ. (2015). DISEÑO DE UN SISTEMA DE AUTENTICACIÓN BIOMÉTRICA BASADO EN RECONOCIMIENTO FACIAL. *Universidad Nacional De Loja*, *0*(0), 151. http://dspace.unl.edu.ec/jspui/bitstream/123456789/17025/1/TESIS WILSON FERNANDO.pdf

Sierra, M. (2015). *Estudio comparativo de modelos de identificación facial basados en correlación*. https://idus.us.es/bitstream/handle/11441/28426/tfgMariaSierraZapata.pdf?sequence=1&isAllowed=y

ANEXOS

ANEXO A. CRONOGRAMA DE DURACIÓN DEL PROYECTO.

CRONOGRAMA DEL PROYECTO														
DURACIÓN DE EJECUCIÓN DEL PROYECTO		14 MESES												
No.	Descripción de la actividad	MESES												
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Cotización y compra de materiales para realizar el proyecto.														
2	Programación del sistema de seguridad electrónica.														
3	Realización de la estructura externa e interna del proyecto.														
3	Armado del sistema de seguridad electrónica.														
4	Comprobar el funcionamiento adecuado del sistema de seguridad.														

Tabla 3. Cronograma de la duración del proyecto

ANEXO B. LISTADO DE MATERIALES UTILIZADOS.

LISTADO DE MATERIALES DEL MÓDULO
GENERALES
1 RASPBERRY PI 4
1 PANTALLA TÁCTIL DE 7 PULGADAS
1 FUENTE DE 5V – 1.5A
2 FUENTE DE 12V – 1.5A
3 RESISTENCIA DE 1.2K OHMIOS
1 LED ROJO
1 LED AMARILLO
1 LED AZUL
1 CÁMARA WEB HD 720P (USB)
2 ALTA VOZ (5V)
1 REGLETA (6 TOMAS 110V)
2 MÓDULO DE 4 RELÉ (5V)
1 ZUMBADOR (12V)
1 CHAPA ZELENOIDE (12V)
1 VENTILADOR (5V)

Tabla 4. Listado de materiales

ANEXO C. DISEÑO DEL ARMAZÓN.

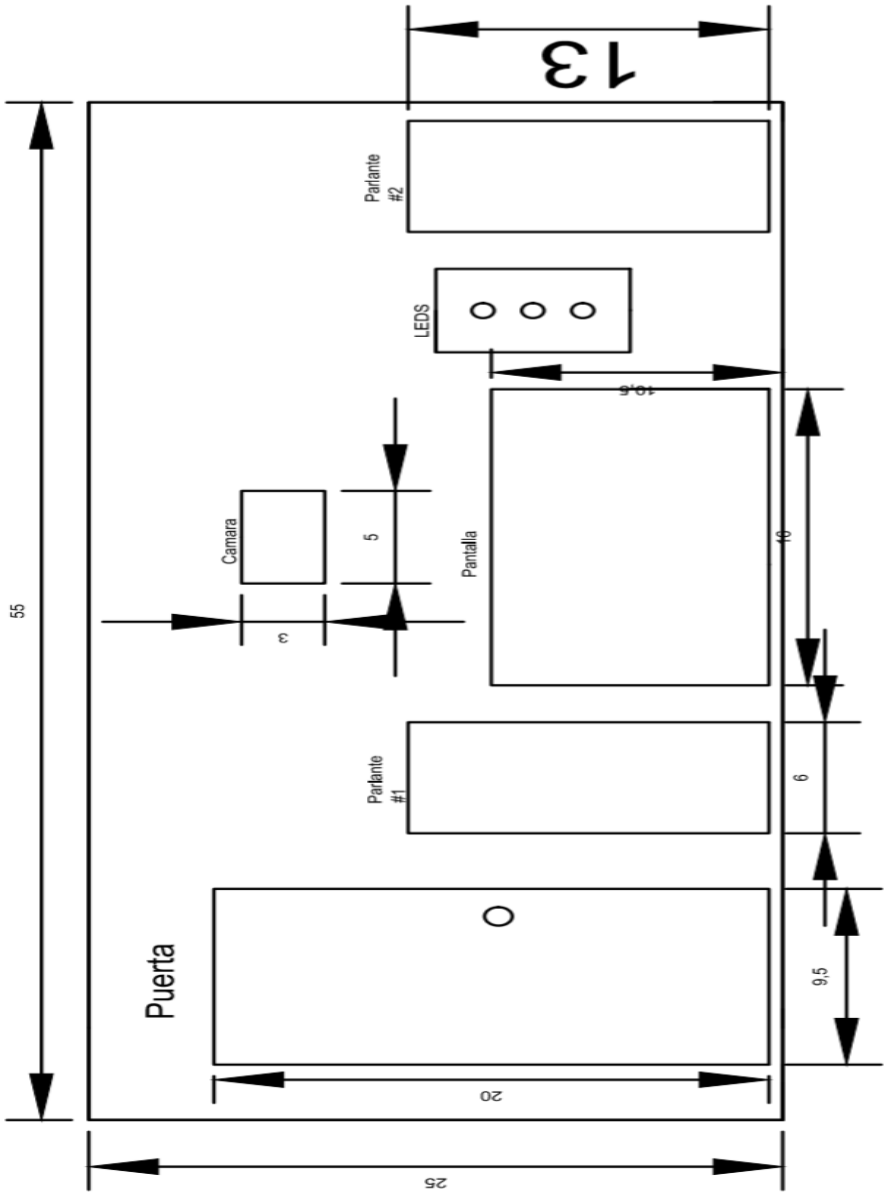


Figura 42. Dimensiones del armazón parte frontal

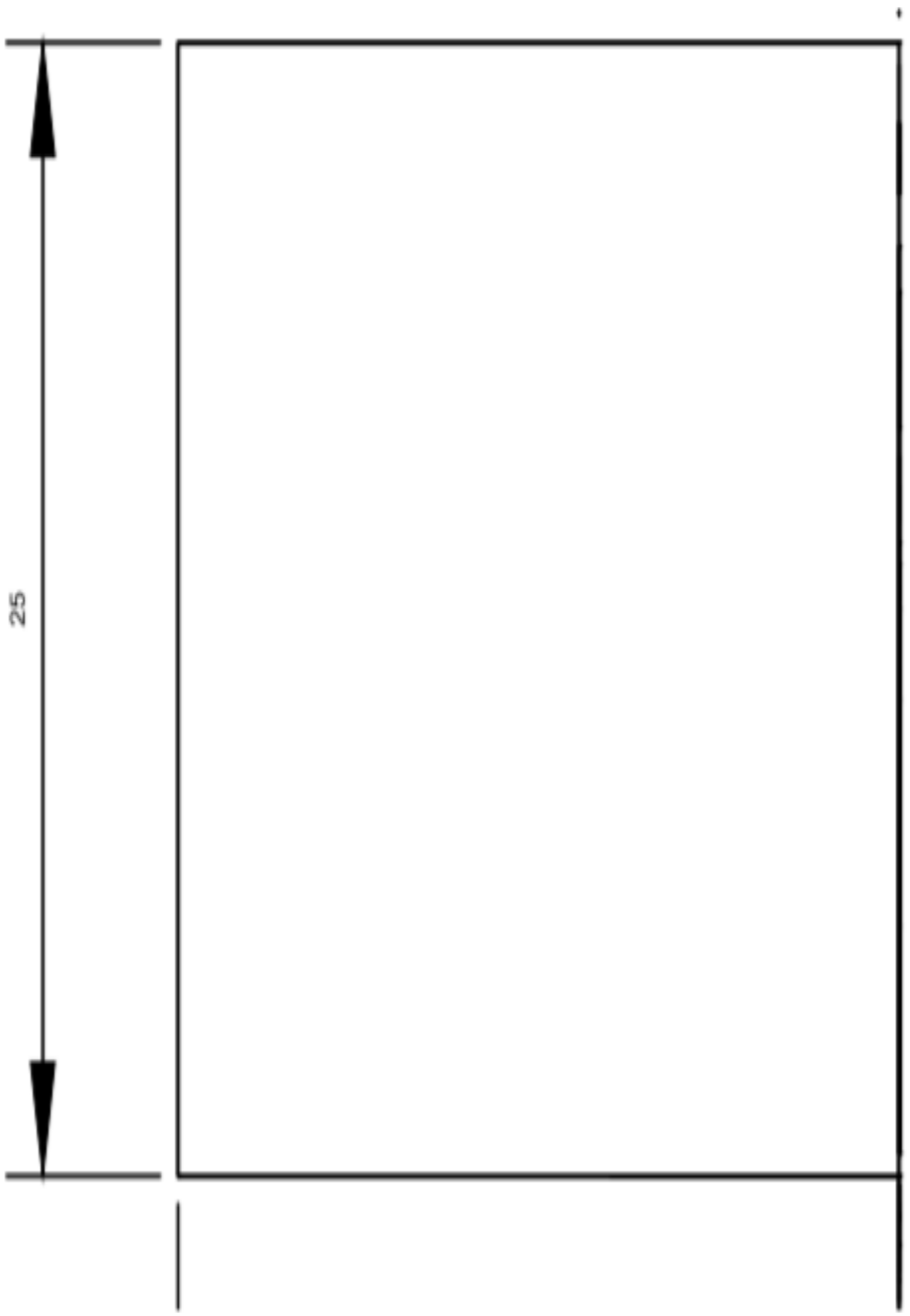


Figura 43. Dimensiones del armazón lado izquierdo y derecho

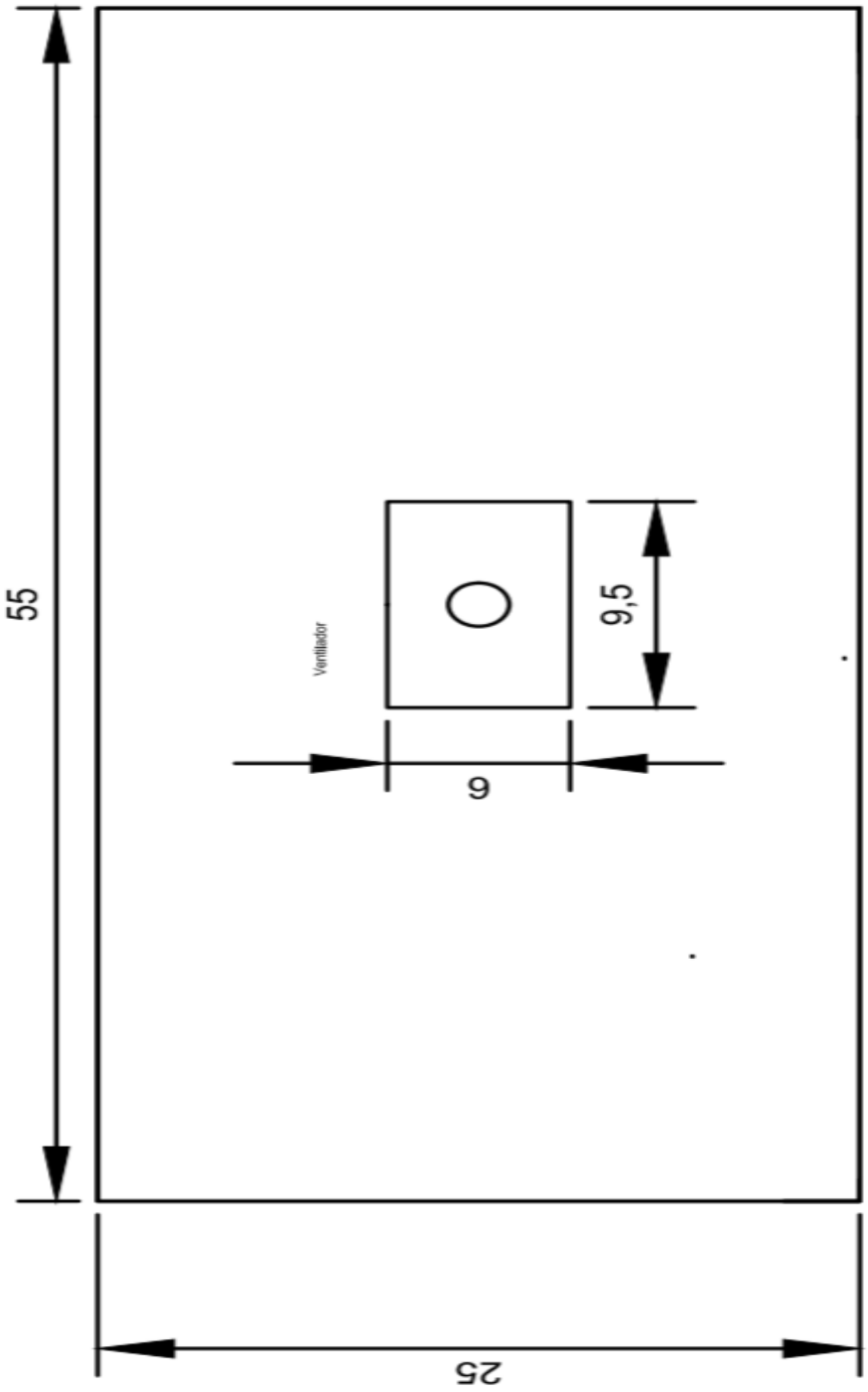
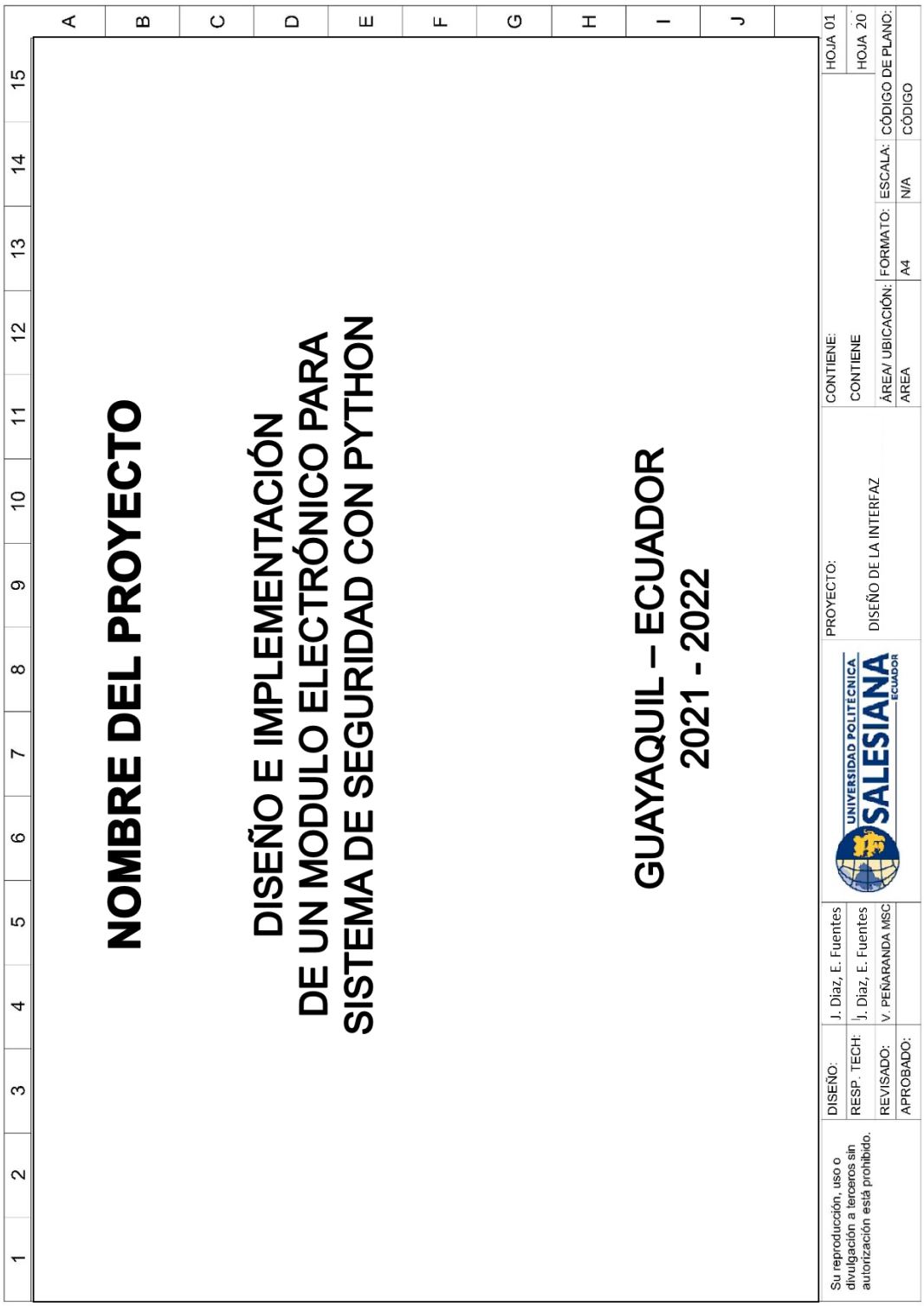


Figura 44. Dimensiones del armazón parte posterior

ANEXO D. DISEÑO DE PLANOS ELÉCTRICOS E INTERFAZ GRÁFICA



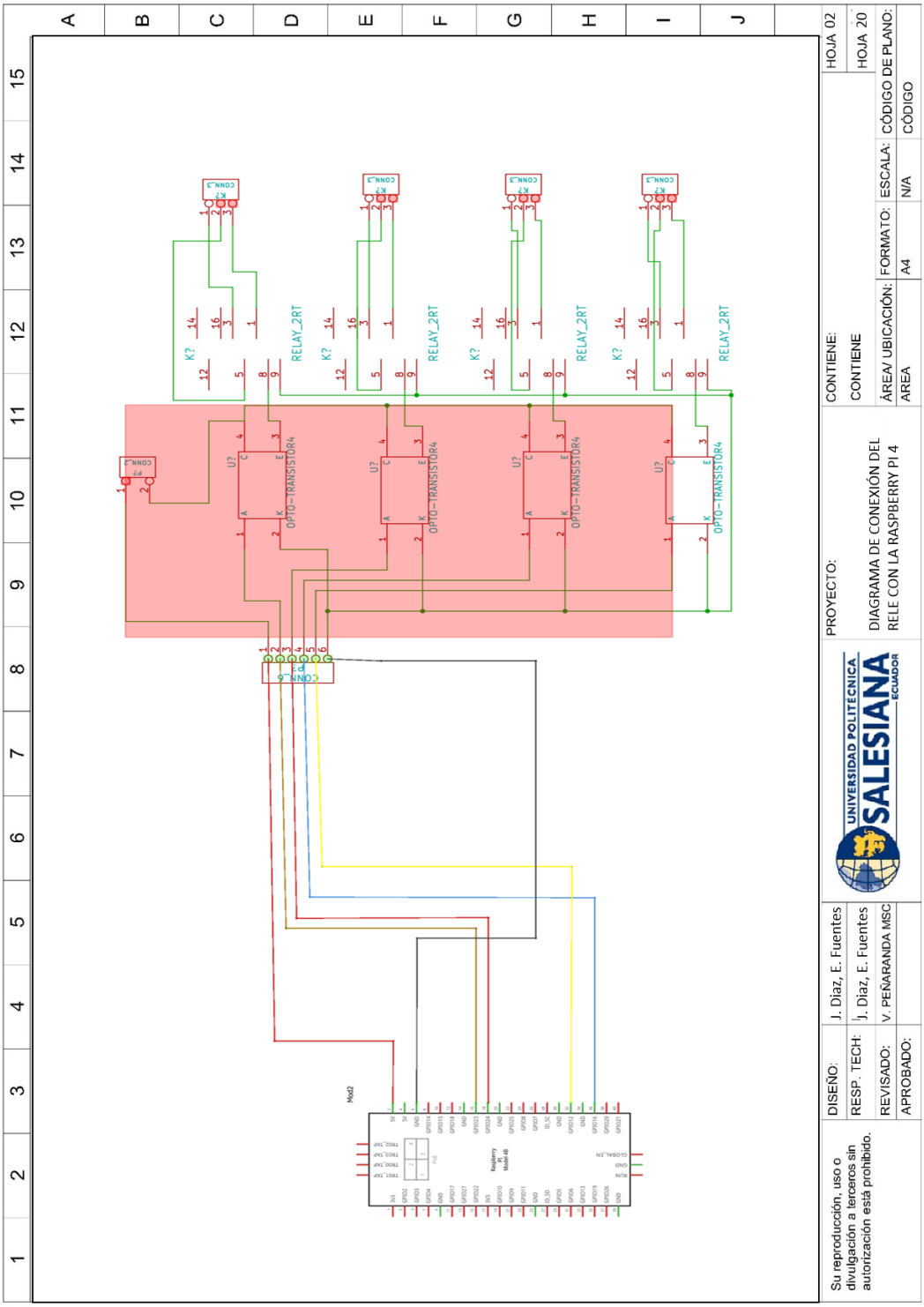


Figura 45. Diagrama de conexión del relé con la Raspberry PI 4 – Fritzing 1

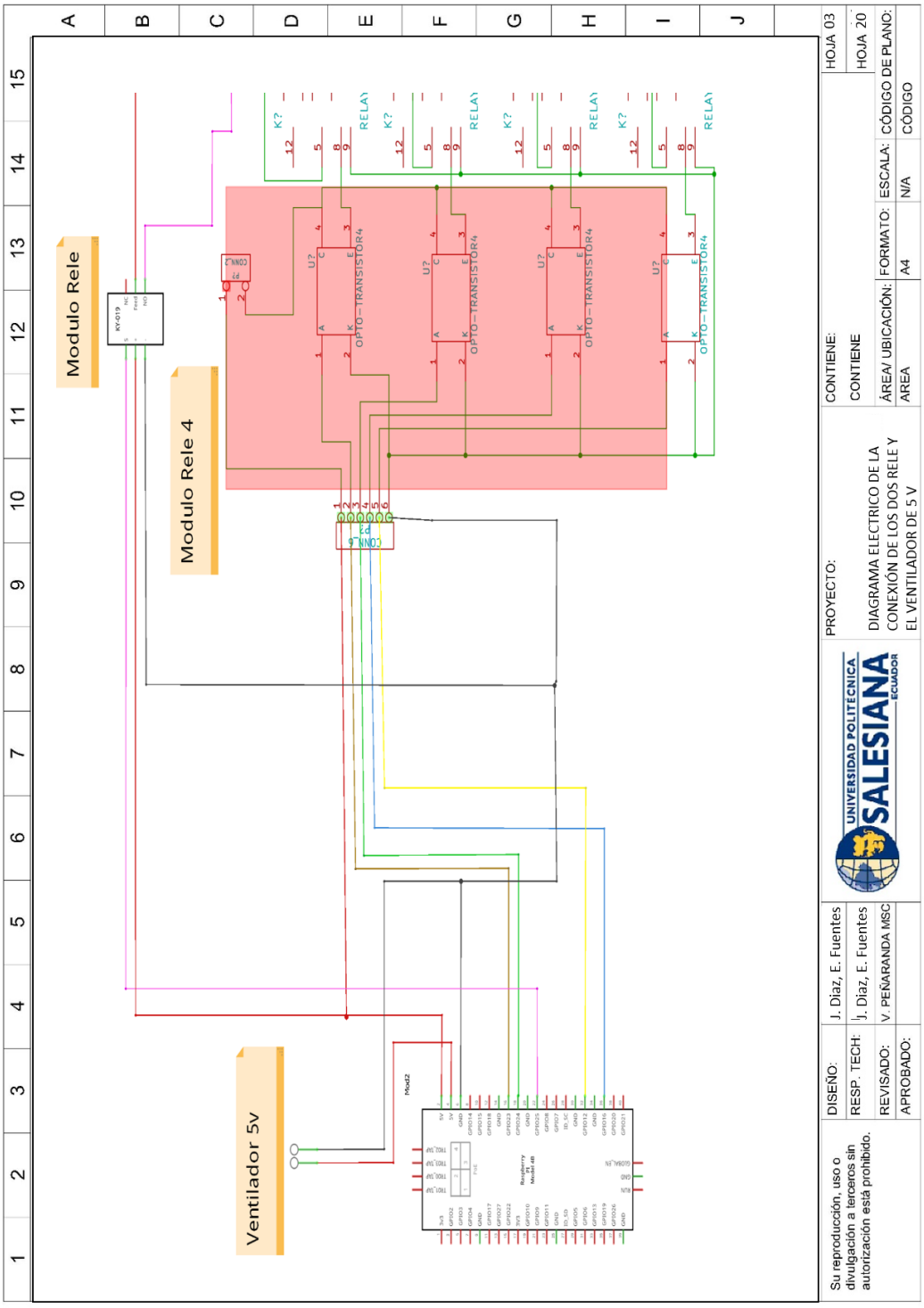


Figura 46. Diagrama eléctrico de la conexión de los relés y el ventilador de 5v – Fritzing 2

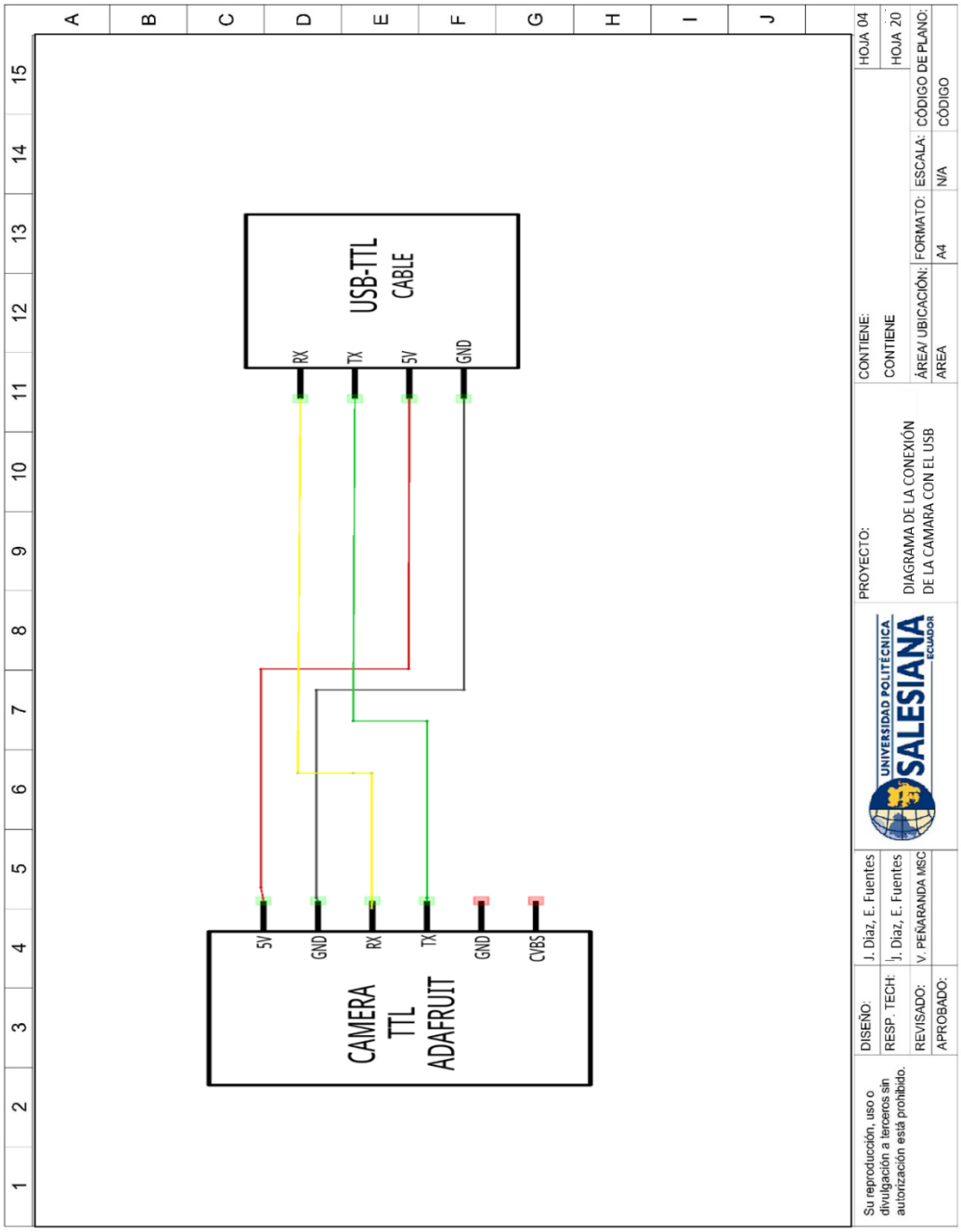


Figura 47. Diagrama de la conexión de la cámara con el USB – Fritzing 3

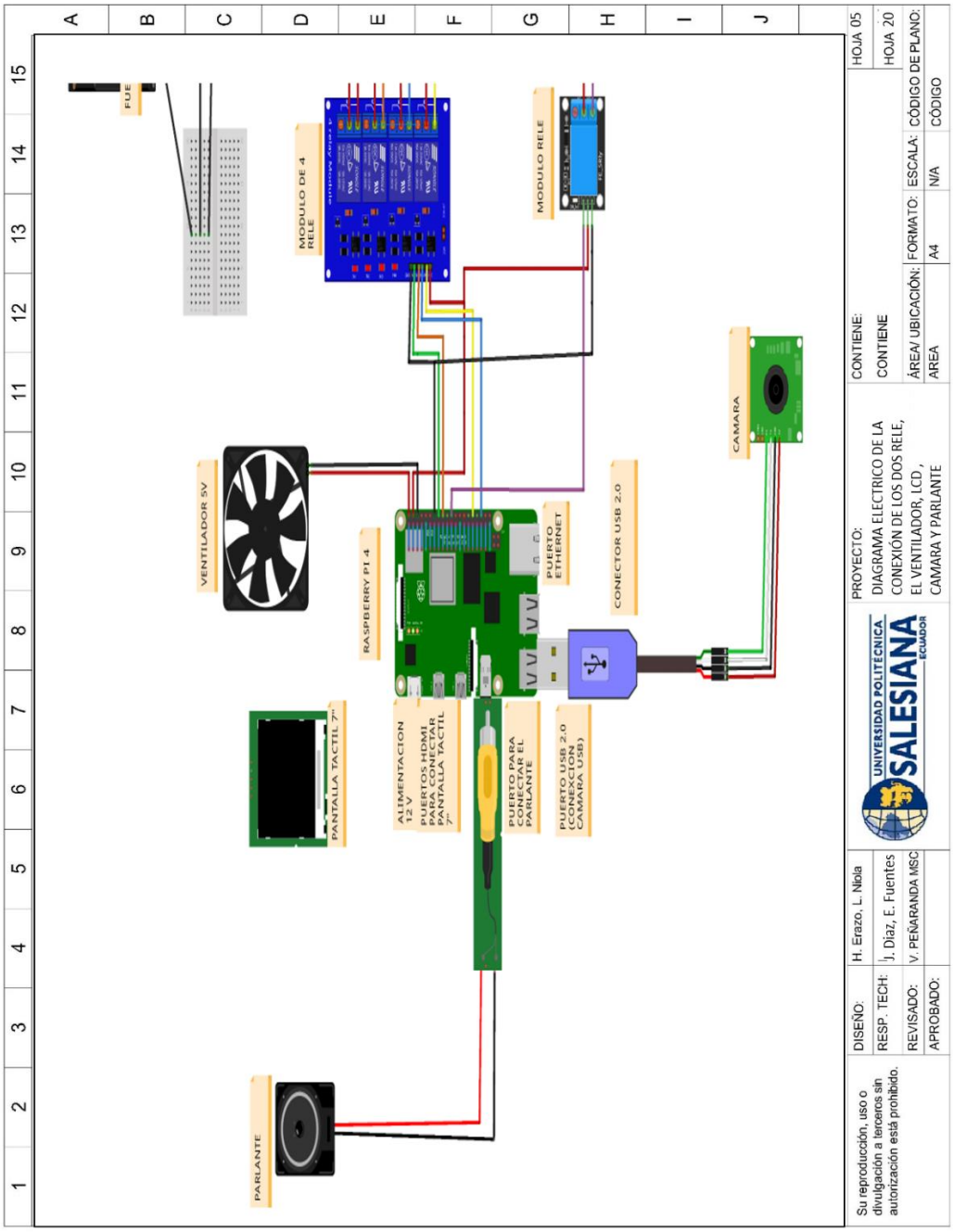


Figura 48. Diagrama eléctrico de la conexión de los materiales con la Raspberry Pi 4 – Fritzing 4

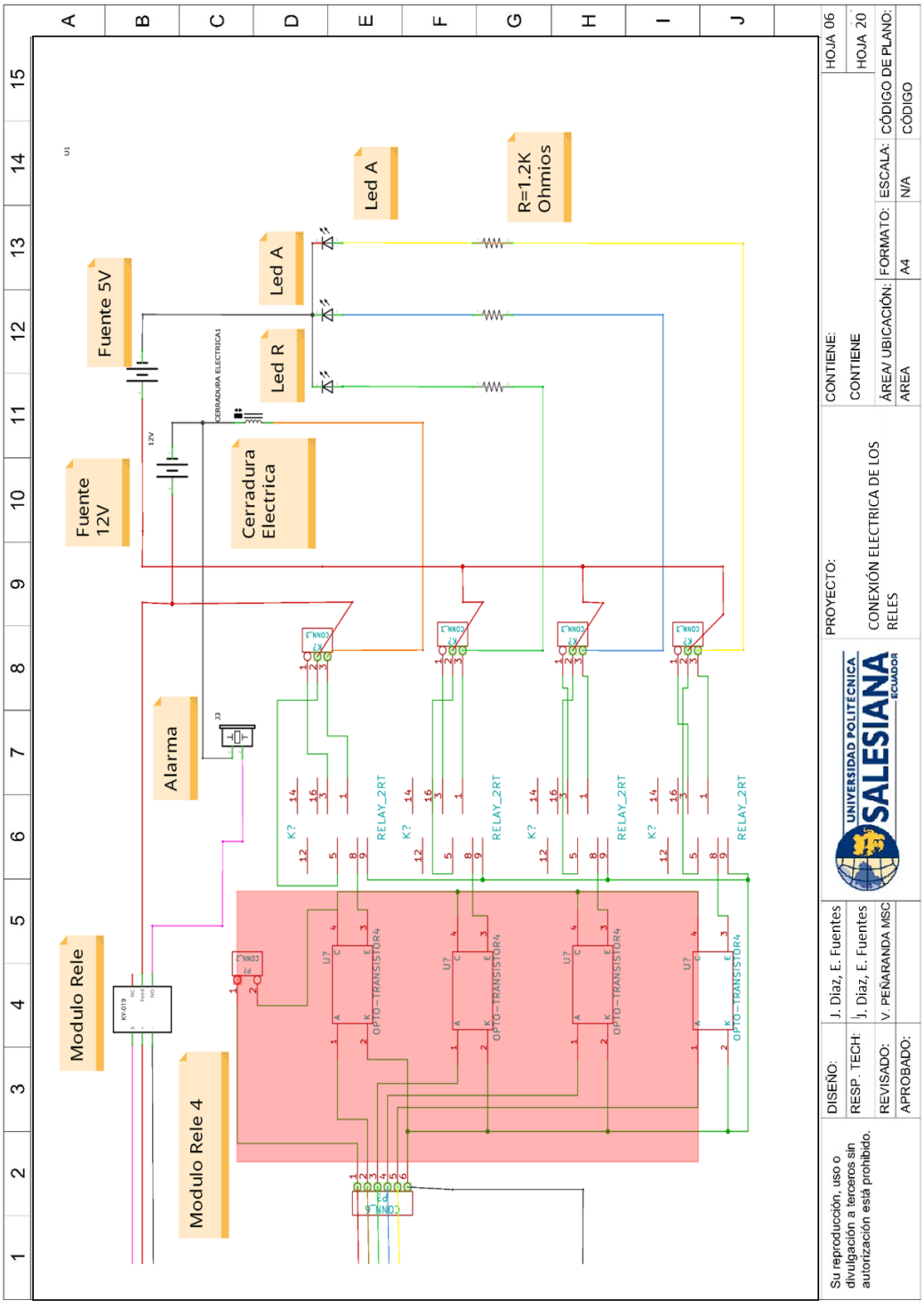


Figura 49. Conexión eléctrica de los relés con: la alarma, cerradura electrica, resistensias y leds – Fritzing 5

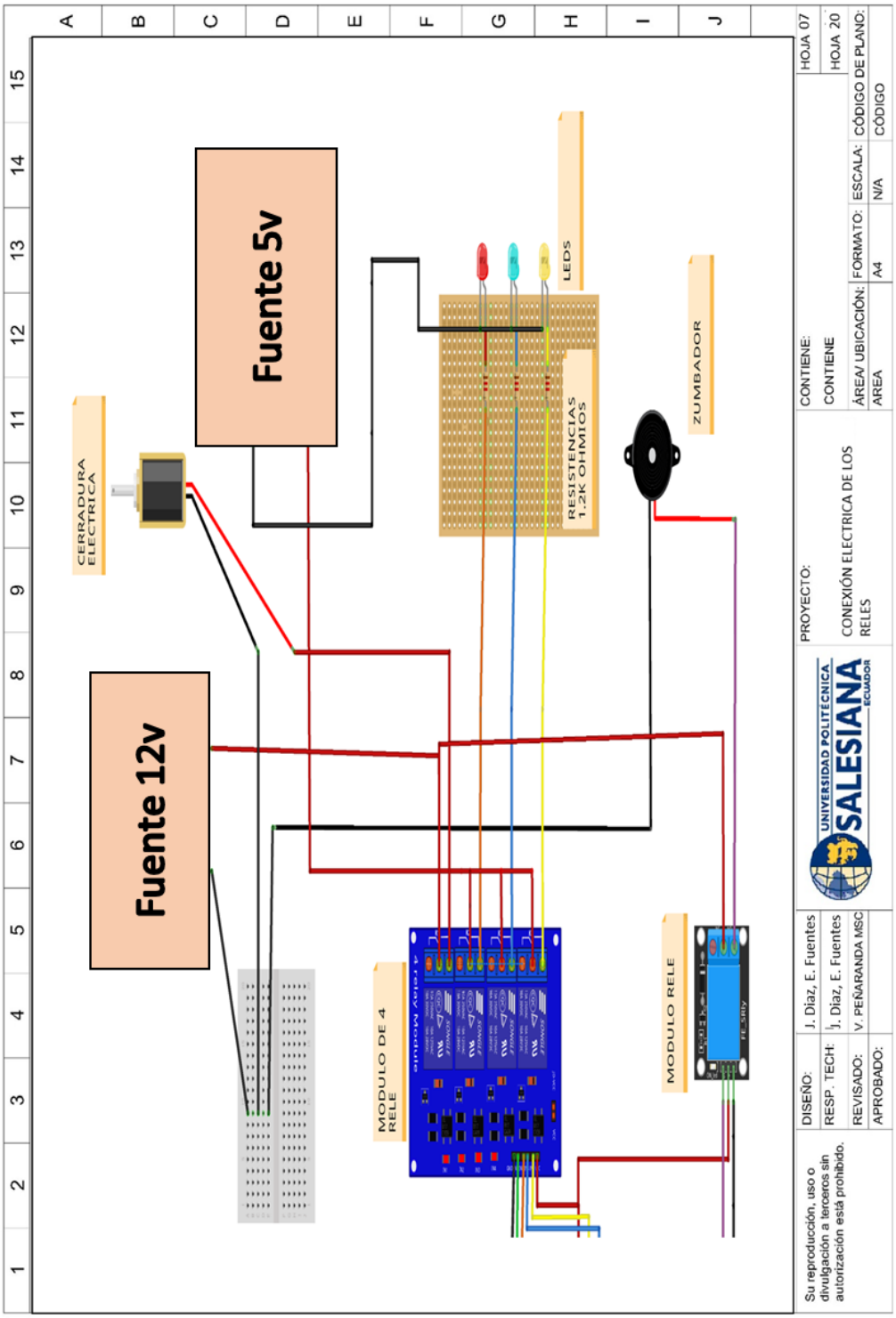


Figura 50. Conexión de los relés con: la alarma, cerradura eléctrica, resistencias y leds – Fritzing 6

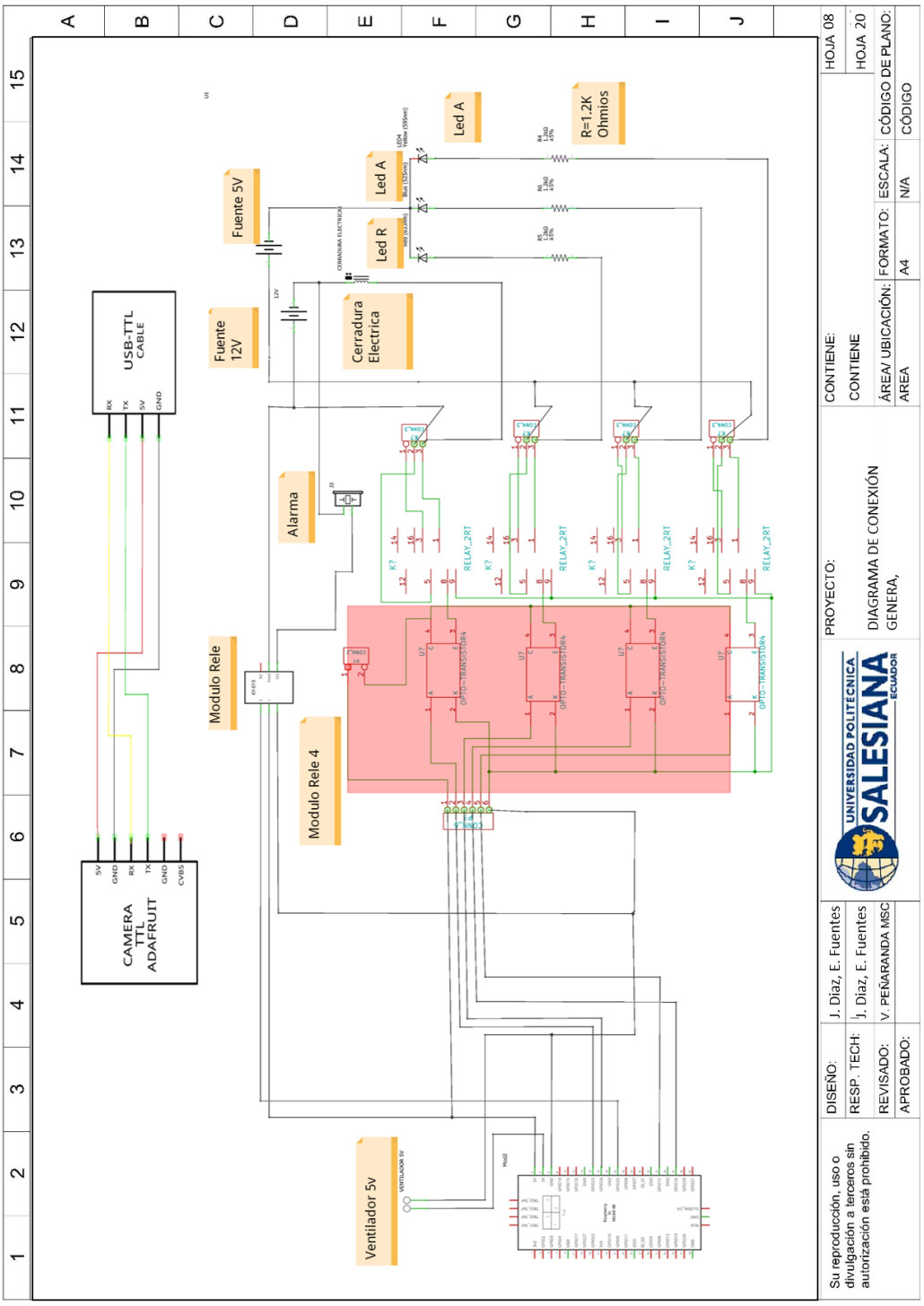


Figura 51. Diagrama de conexión General – Fritzing 7

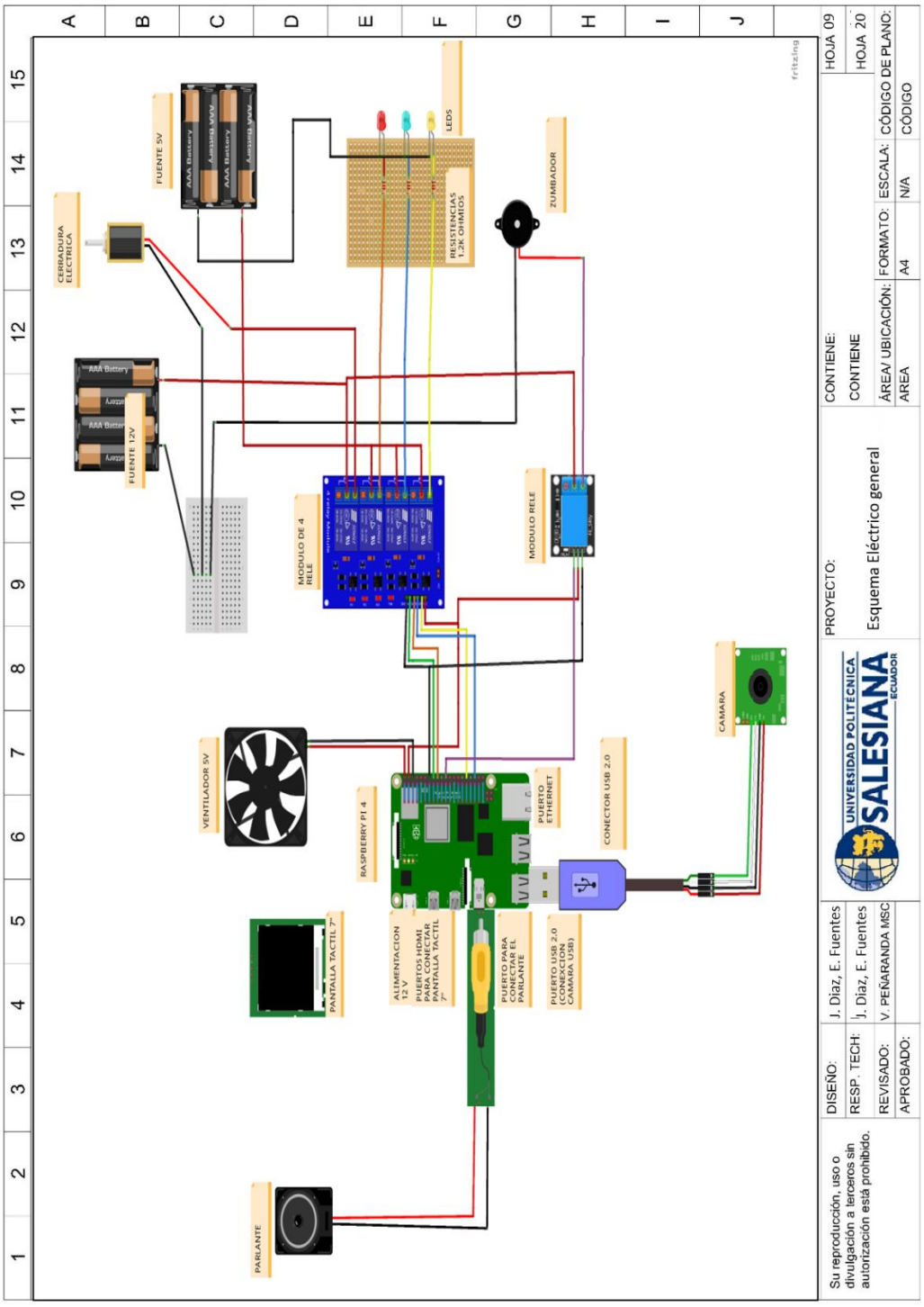


Figura 52. Esquema eléctrico General – Fritzing 8

Figura 53. Diseño de la interfaz de la pantalla principal - PowerPoint 9

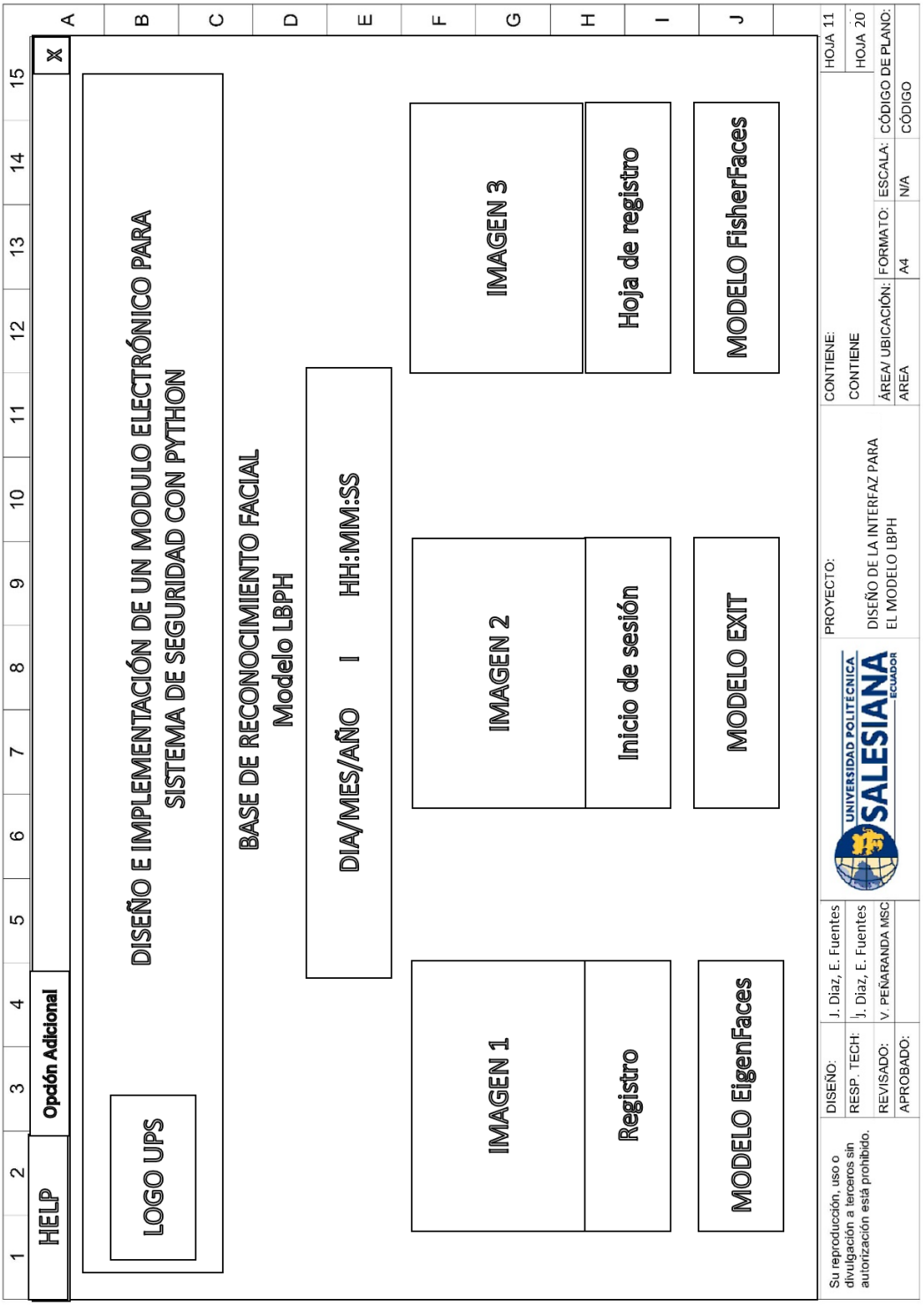


Figura 54. Diseño de la interfaz para el modelo LBPH (Programador) – PowerPoint 10

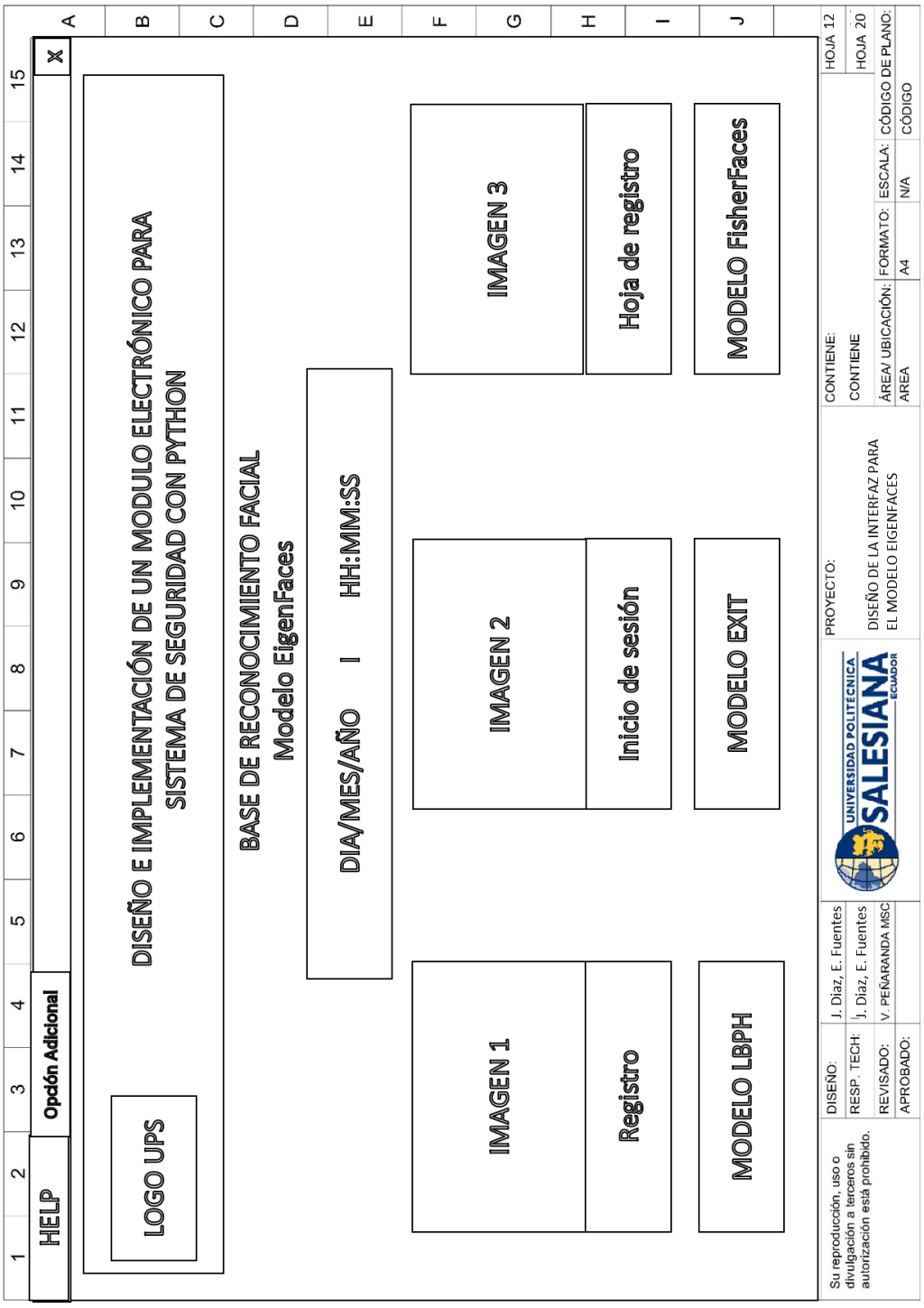


Figura 55. Diseño de la interfaz para el modelo EigenFaces (Programador) – PowerPoint 11

Figura 56. Diseño de la interfaz para el modelo FisherFaces (Programador) – PowerPoint 12

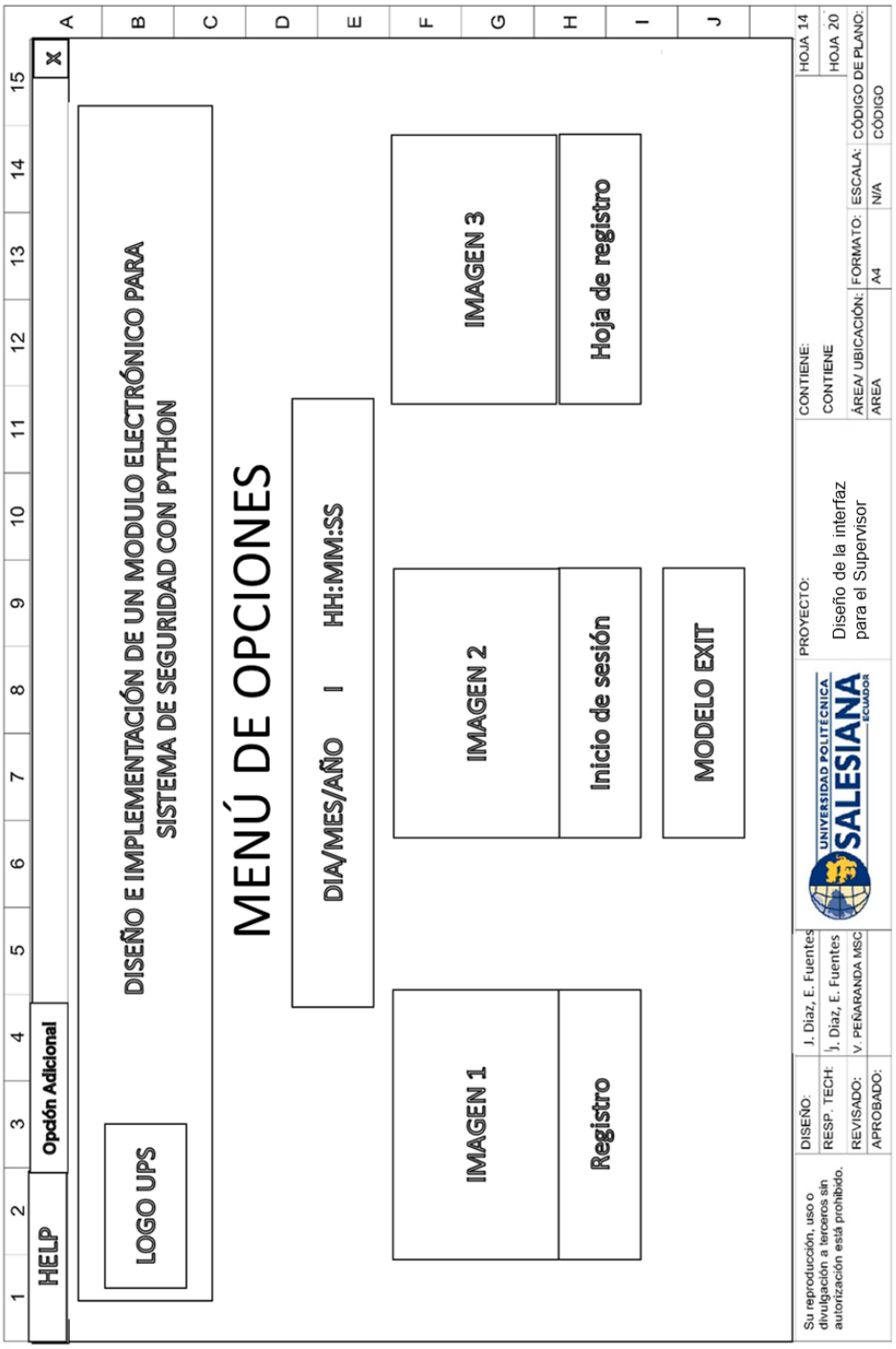


Figura 57. Diseño de la interfaz para el Supervisor– PowerPoint 13

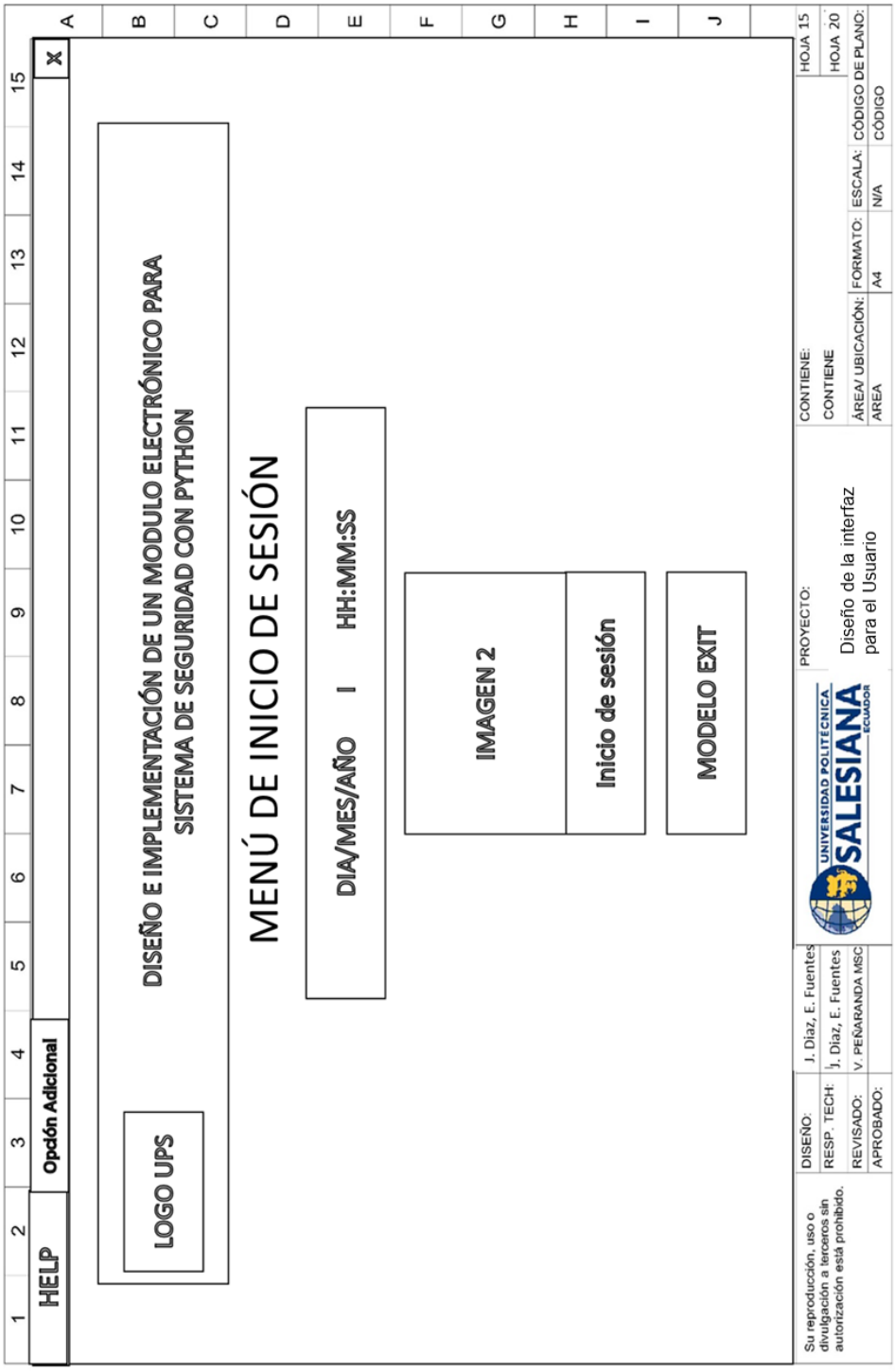


Figura 58. Diseño de la interfaz para el Usuario – PowerPoint 14

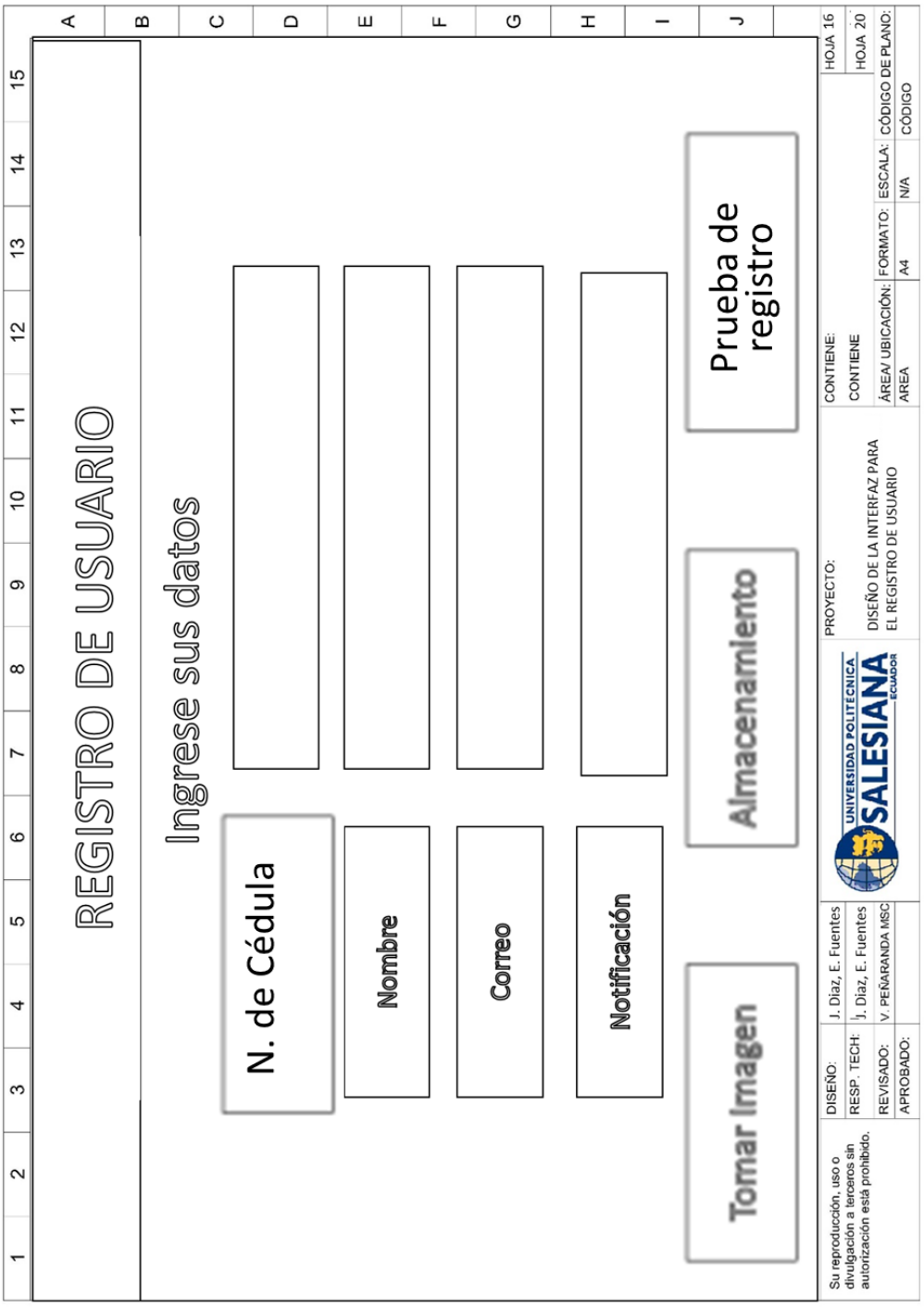


Figura 59. Diseño de la interfaz para el registro Usuario – PowerPoint 15

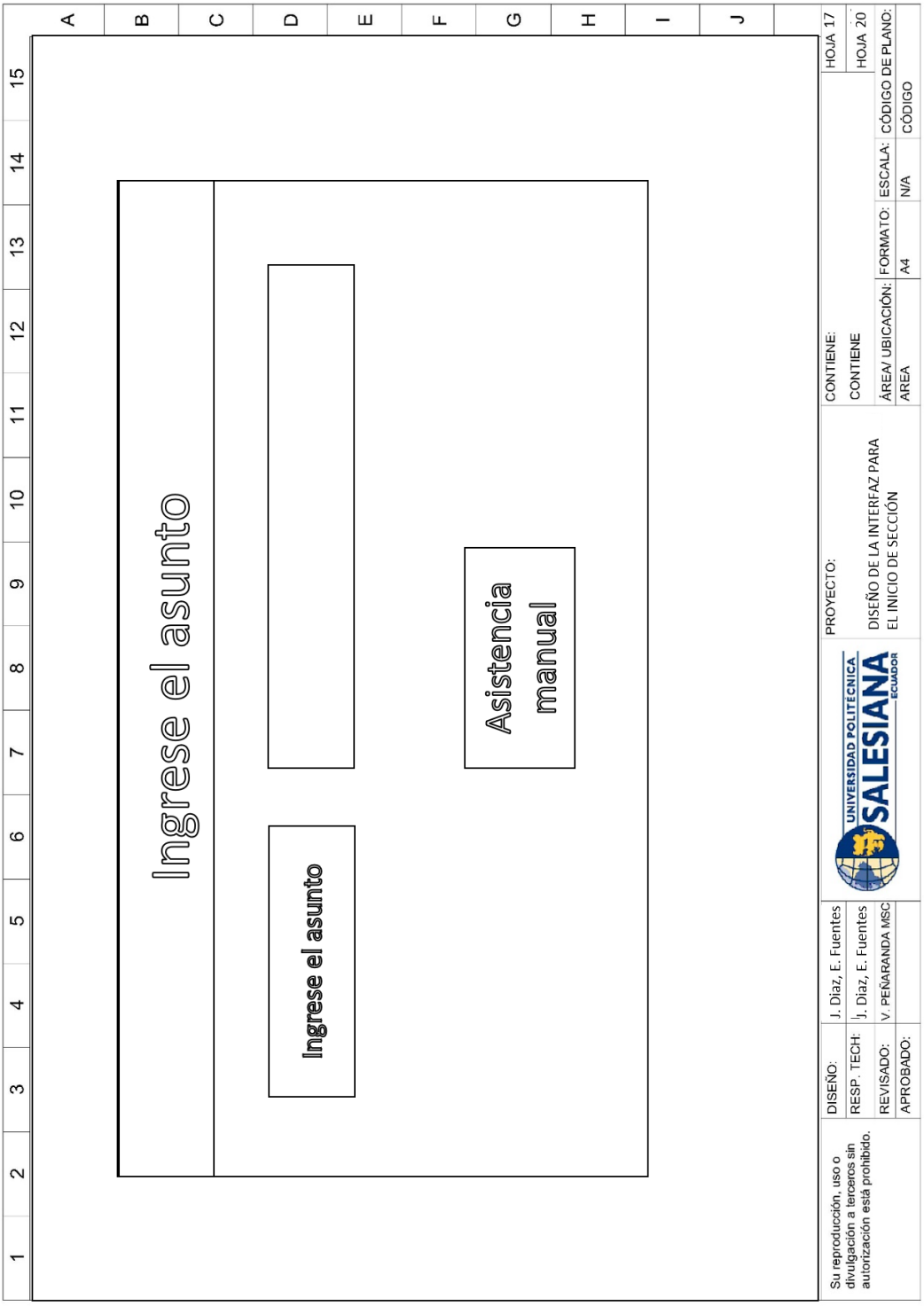


Figura 60. Diseño de la interfaz para el inicio de sección **–** PowerPoint 16

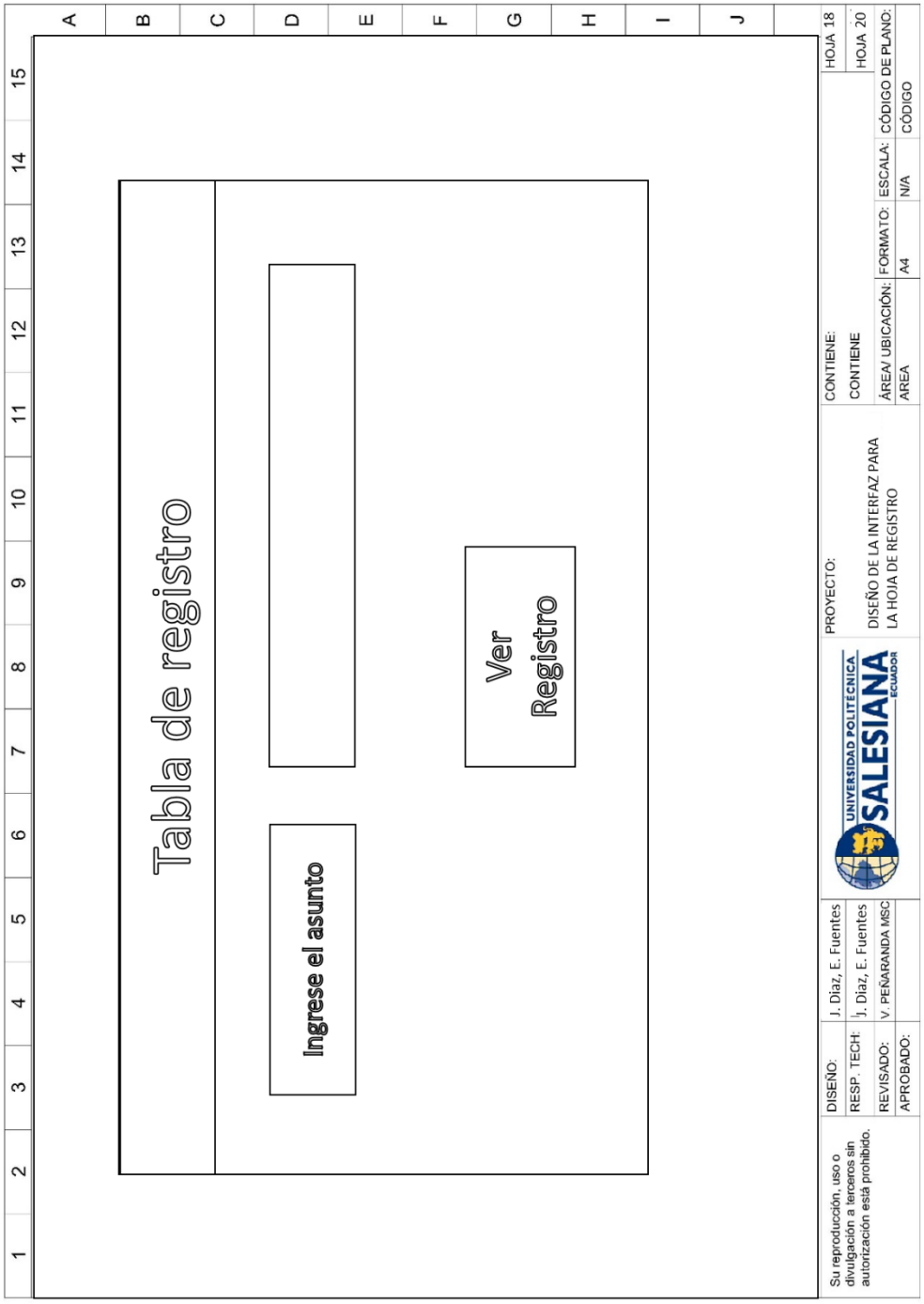


Figura 61. Diseño de la interfaz para Hoja de registro– PowerPoint 17

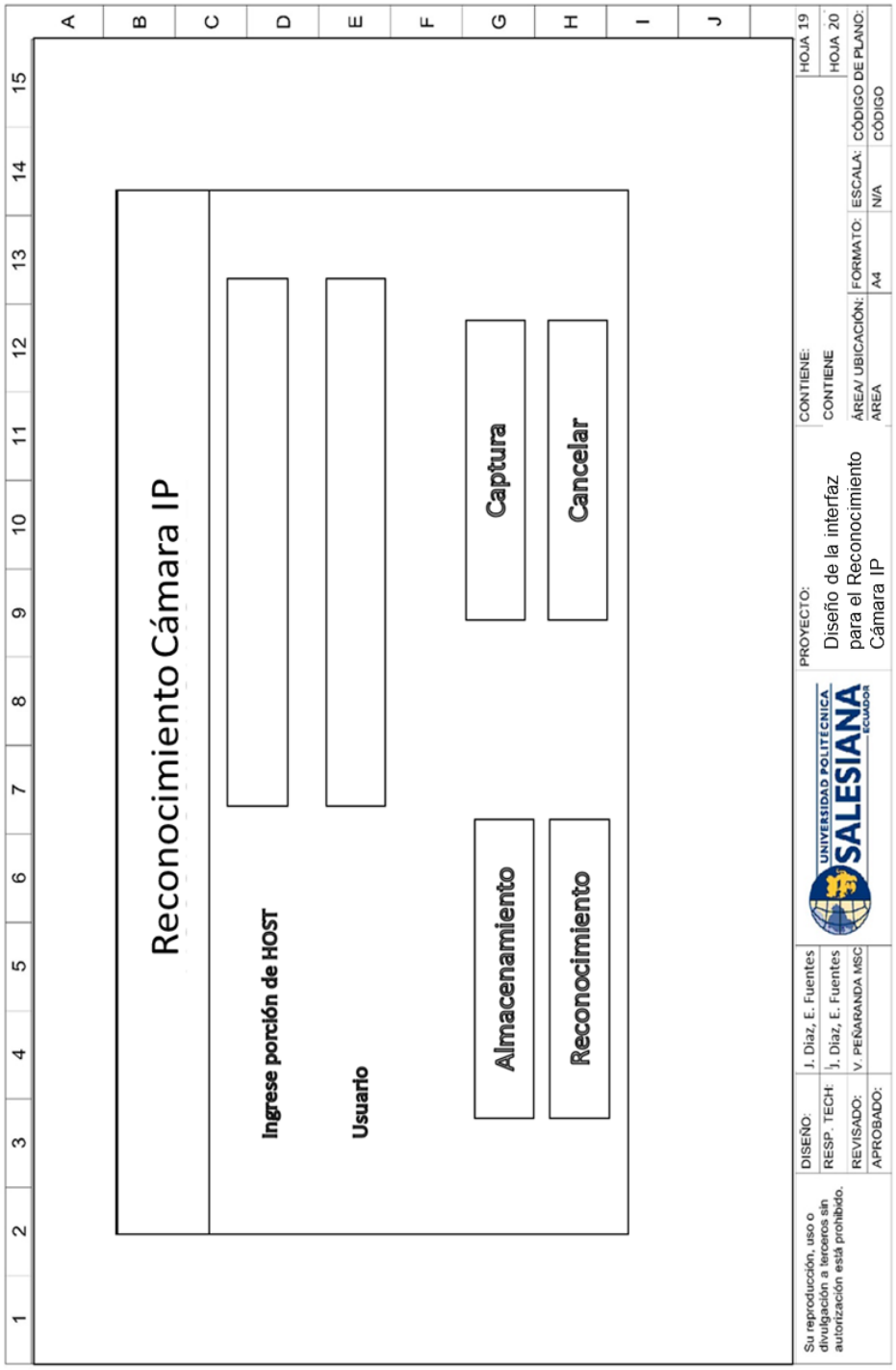


Figura 62. Diseño de la interfaz para el Reconocimiento Cámara IP – PowerPoint 18

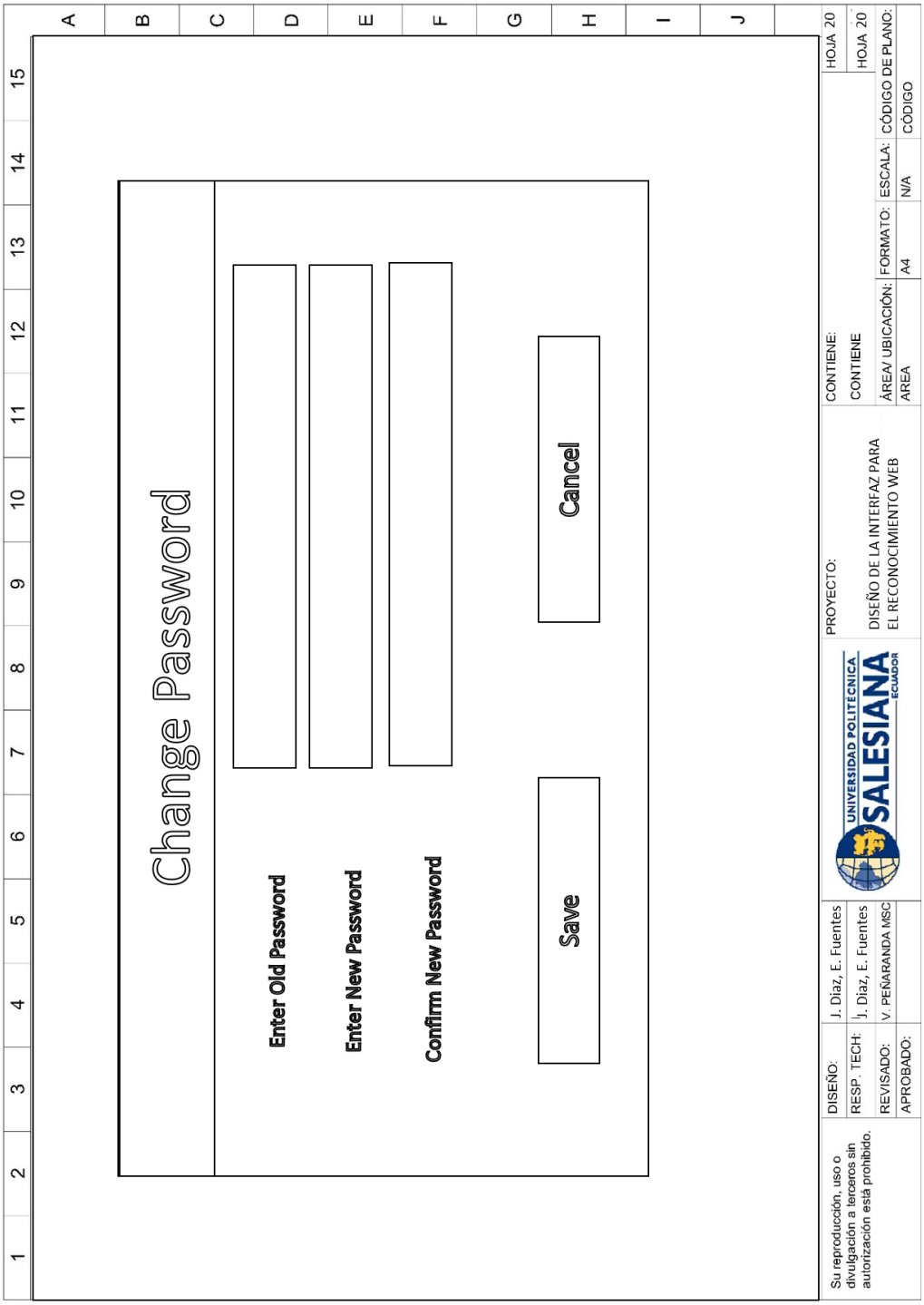


Figura 63. Diseño de la interfaz para el cambio de contraseña **–** PowerPoint 19

ANEXO E. PROGRAMACIÓN

El primer paso para programar en Python es establecer una comunicación vía ethernet con el router a la Raspberry Pi 4, para revisar la IP que está utilizando la Raspberry Pi, en una laptop se descargó la aplicación “advance IP SCANNER”, que ayuda reconocer la IP que está utilizando la Raspberry Pi, como se observa en la figura 64.

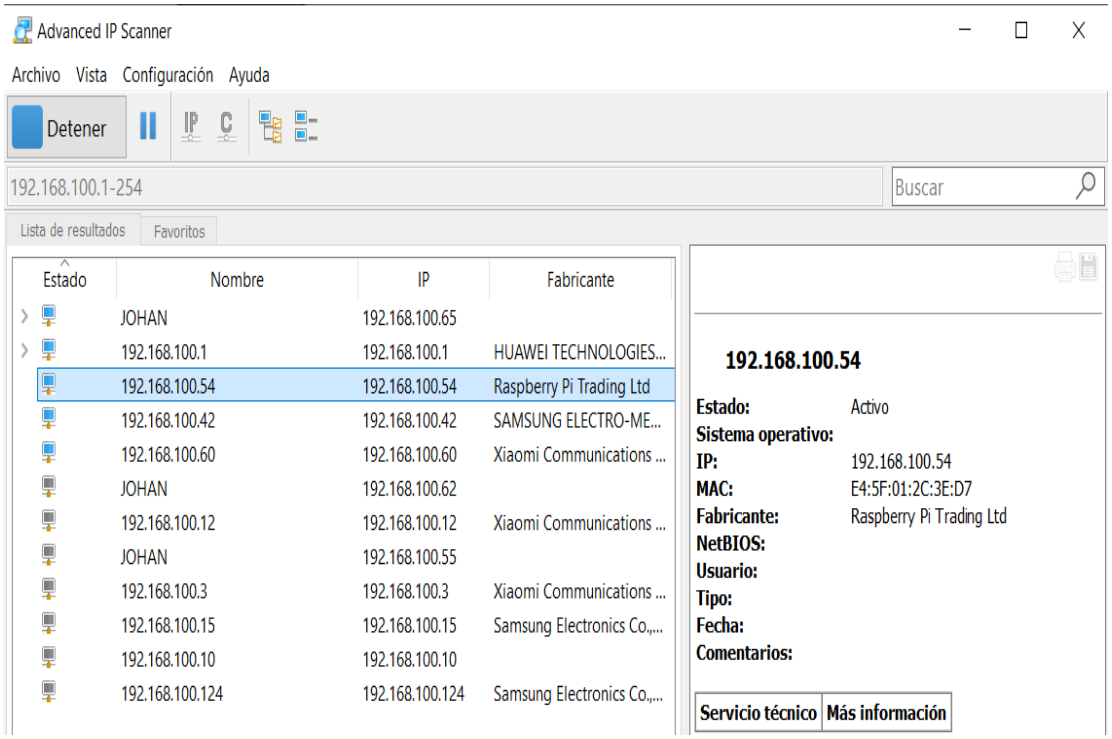


Figura 64. Se asignó una dirección IP para la Raspberry PI 192.168.100.54.

Como se observa en la figura 65, para poder entrar a la Raspberry se instaló el programa VNC Viewer, el cual permite conectar remotamente a nuestra Raspberry Pi 4, para poder conectarse se agrega un nuevo dispositivo en el programa y se ingresó la IP de la Raspberry PI.

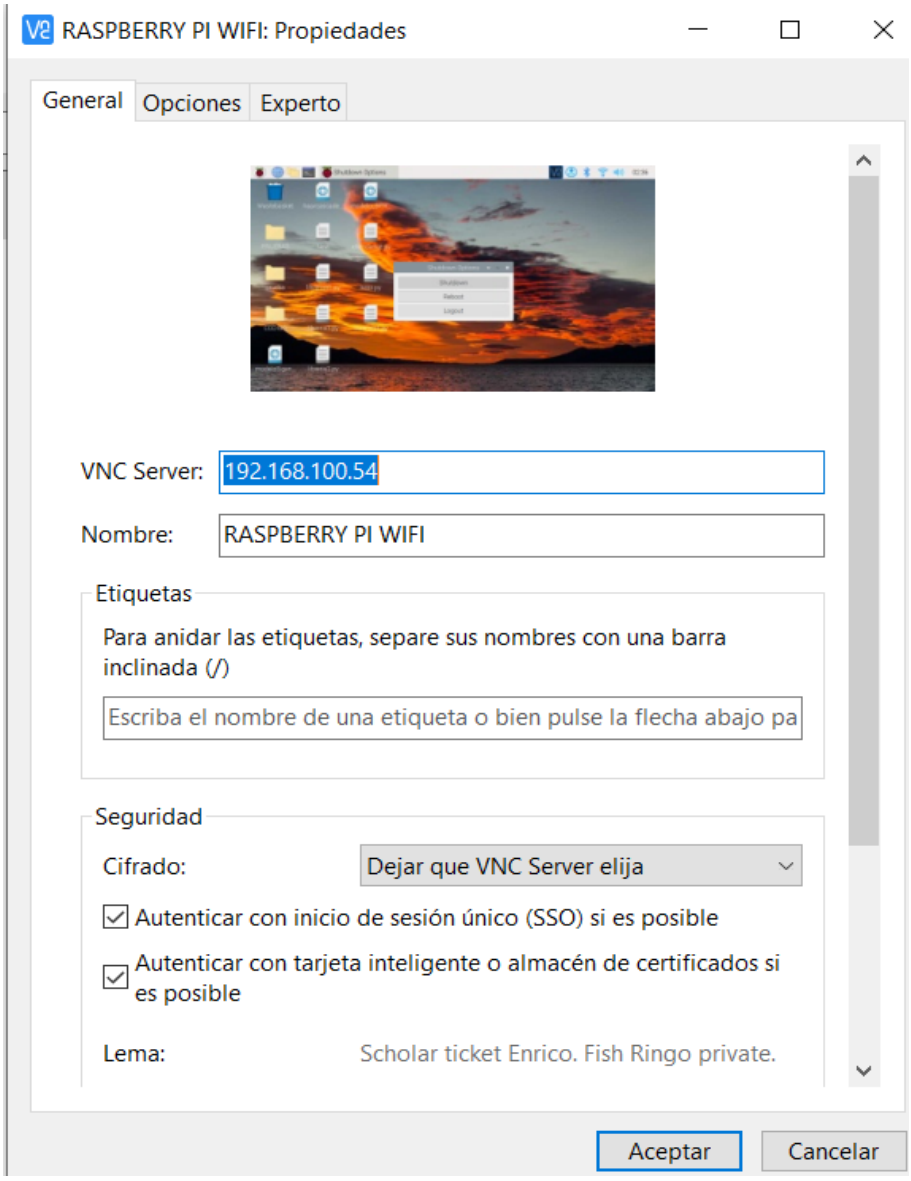


Figura 65. Agregado de la IP del servidor para la conexión de la Raspberry Pi 4**.**

Como se observa en la figura 66, una vez dentro de la Raspberry se cambió el método de conexión de la red, paso de utilizar Ethernet a Wlan.

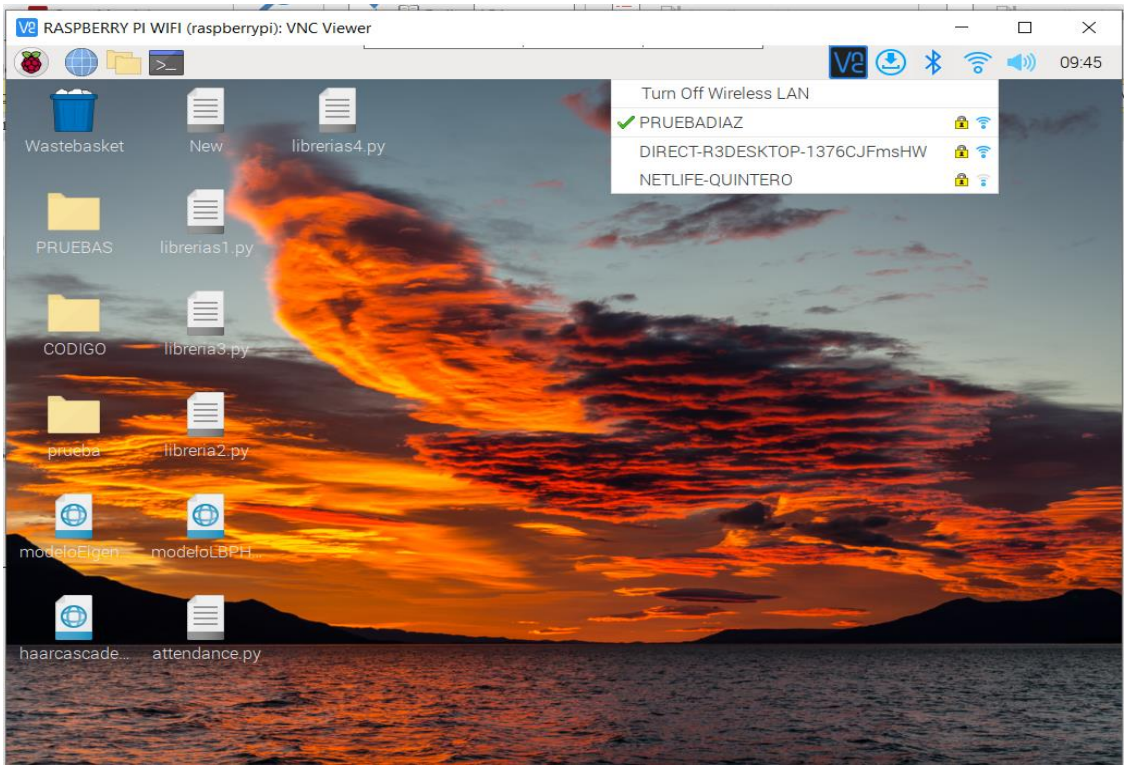


Figura 66. Cambio de conexión de Ethernet a Wlan

Para la programación en Python dentro de la Raspberry se utilizó el programa que ya viene instalado en la Raspberry Pi 4, llamado Thonny, como se observa en la figura 67.

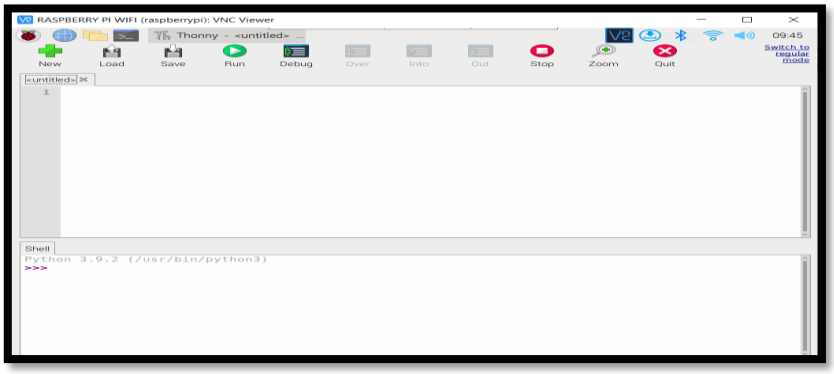


Figura 67. Programa Thonny

Declaración de librerías a utilizar.

Para empezar a desarrollar el programa se instalaron las librerías y se las nombra, como se ve en las siguientes figuras 68, 69, 70, 71, 72 y 73:

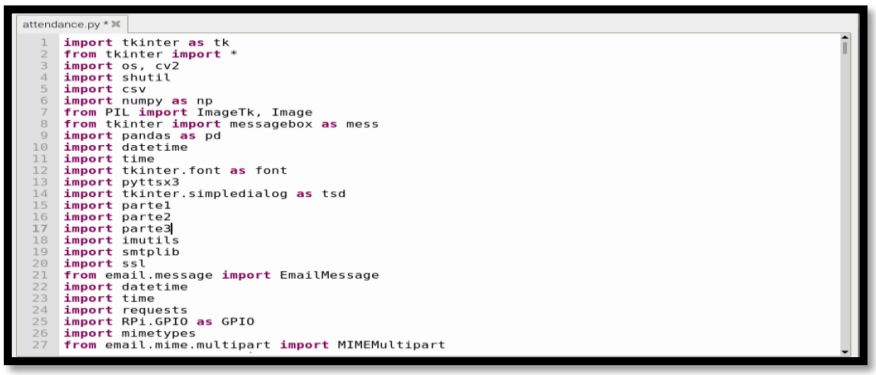


Figura 68. Declaración de librerías menú principal.

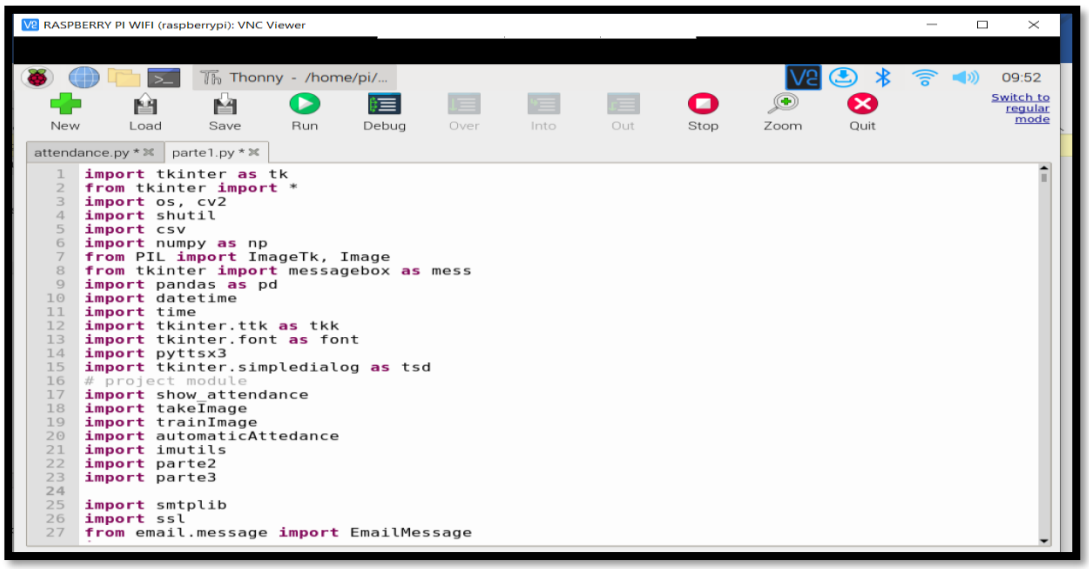


Figura 69. Declaración de librerías para la ventana modelo (programador) , Supervisor y Usuario

Figura 70. Declaración de librerías para la captura.

attendance.py * | parte1.py * | takeImage.py | trainImage.py | automaticAttedance.py | show_attendance.py | takemanually.py

```
1  import csv
2  import os, cv2
3  import numpy as np
4  import pandas as pd
5  import datetime
6  import time
7  from PIL import ImageTk, Image
8  import RPi.GPIO as GPIO
9
```

Figura 71. Declaración de librerías para el almacenamiento.

attendance.py * | parte1.py * | takeImage.py | trainImage.py | automaticAttedance.py | show_attendance.py | takemanually.py

```
1  import tkinter as tk
2  from tkinter import *
3  import os, cv2
4  import shutil
5  import csv
6  import numpy as np
7  from PIL import ImageTk, Image
8  import pandas as pd
9  import datetime
10 import time
11 import tkinter.ttk as tkk
12 import tkinter.font as font
13 import datetime
14 import ssl
15 import smtplib
16 import time
17 import imutils
18 import smtplib
19 import mimetypes
20 # Importamos los módulos necesarios
21 from email.mime.multipart import MIMEMultipart
22 from email.mime.image import MIMEImage
23 import RPi.GPIO as GPIO
24 import mimetypes
25 from email.mime.multipart import MIMEMultipart
26 from email.mime.image import MIMEImage
```

Figura 72. Declaración de librerías para el reconocimiento.

Figura 73. Declaración de librerías para ver la hoja de registro

Programación de la pantalla principal.

El nombre del archivo para la pantalla principal es “attendace.py”.

En la Figura 74, se observa que se llama a los puertos 13, 23 y 25; estos puertos se van a utilizar para activar el led rojo (puerto 13), cerradura (puerto 23) y alarma (puerto 25).

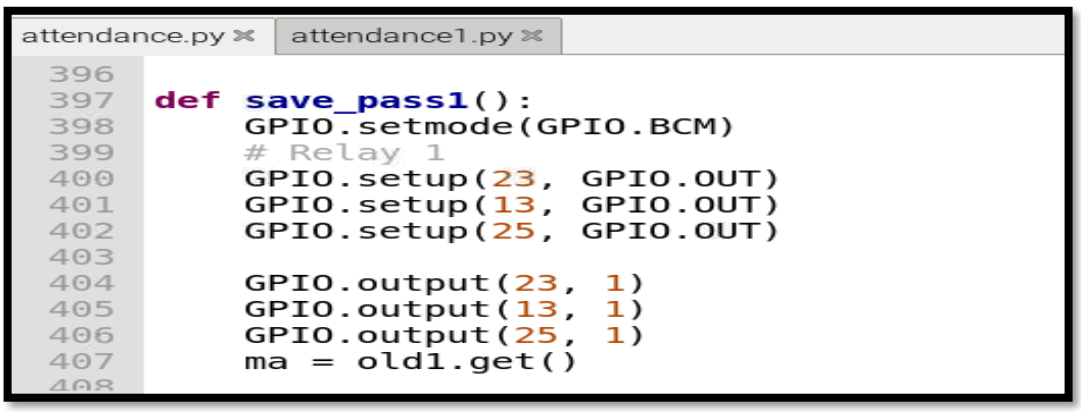
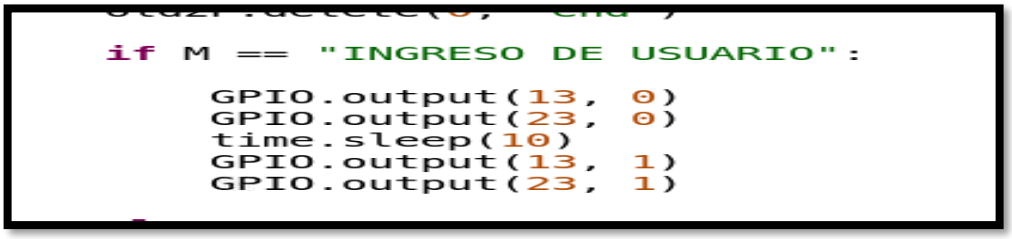


Figura 74. Programación de la pantalla Principal Parte 1

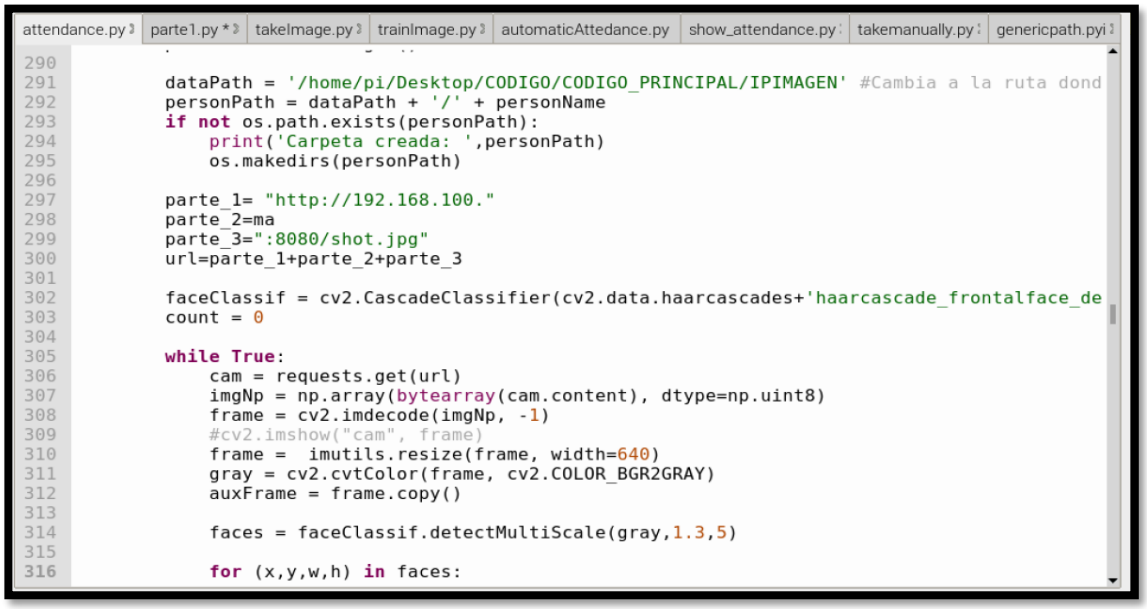
En la figura 75 se muestra el código de activación de los puertos GPIO, si el reconocimiento del Usuario es exitoso, se activa los pines 13 y 23 y después de 10 segundos desactiva los puertos GPIO.



```
if M == "INGRESO DE USUARIO":
        GPIO.output(13, 0)
        GPIO.output(23, 0)
        time.sleep(10)
        GPIO.output(13, 1)
        GPIO.output(23, 1)
```

Figura 75. Programación de la pantalla Principal Parte 2

En la Figura 76, se captura el rostro de la persona y se almacena en una carpeta con el nombre del usuario.



attendance.py | parte1.py * | takeImage.py | trainImage.py | automaticAttedance.py | show_attendance.py | takemanually.py | genericpath.pyi

```
290
291         dataPath = '/home/pi/Desktop/CODIGO/CODIGO_PRINCIPAL/IPIMAGEN' #Cambia a la ruta dond
292         personPath = dataPath + '/' + personName
293         if not os.path.exists(personPath):
294             print('Carpeta creada: ',personPath)
295             os.makedirs(personPath)
296
297         parte_1= "http://192.168.100."
298         parte_2=ma
299         parte_3=":8080/shot.jpg"
300         url=parte_1+parte_2+parte_3
301
302         faceClassif = cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_de
303         count = 0
304
305         while True:
306             cam = requests.get(url)
307             imgNp = np.array(bytearray(cam.content), dtype=np.uint8)
308             frame = cv2.imdecode(imgNp, -1)
309             #cv2.imshow("cam", frame)
310             frame =  imutils.resize(frame, width=640)
311             gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
312             auxFrame = frame.copy()
313
314             faces = faceClassif.detectMultiScale(gray,1.3,5)
315
316             for (x,y,w,h) in faces:
```

Figura 76. Programación de la Captura de rostro

En la figura 77 se procede al almacenamiento del rostro, en esta parte una vez el usuario se registró en el sistema, se comienza almacenar el rostro en un archivo XML y este a su vez se utiliza en el reconocimiento del usuario para comparar los rostros del registro con la persona que inicia sección en el módulo de seguridad.

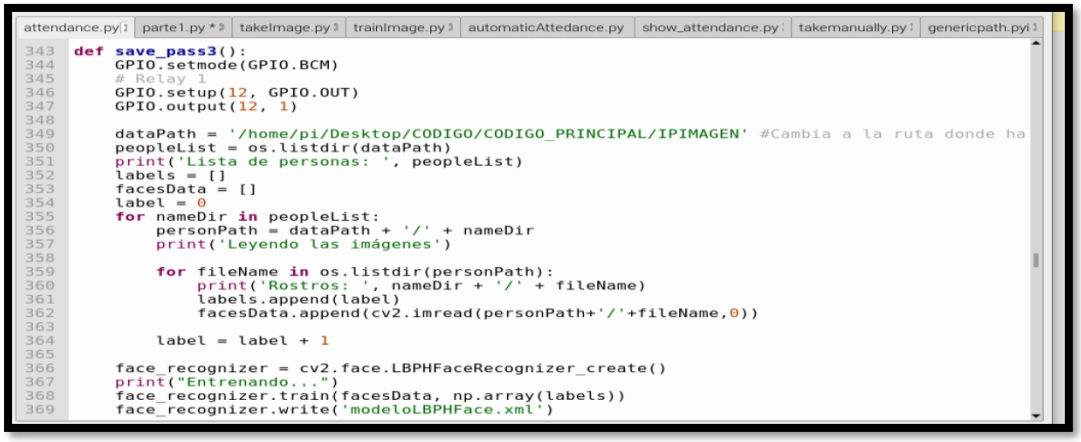


Figura 77. Programación del almacenamiento o entrenamiento

En la figura 78 se observa la pantalla principal del módulo de seguridad.

Figura 78. Pantalla principal

Programación de la pantalla modelo de opciones del programador.

En esta parte se detalla la programación de la pantalla de modelo de opciones (pantalla del Programador).

En la figura 79 se ven codigos de comandos de la programacion de la ventana del registro de usurios y en la figura 80 se programa el ingreso de datos del registro para la captura, almacenamiento y registro del usuario .

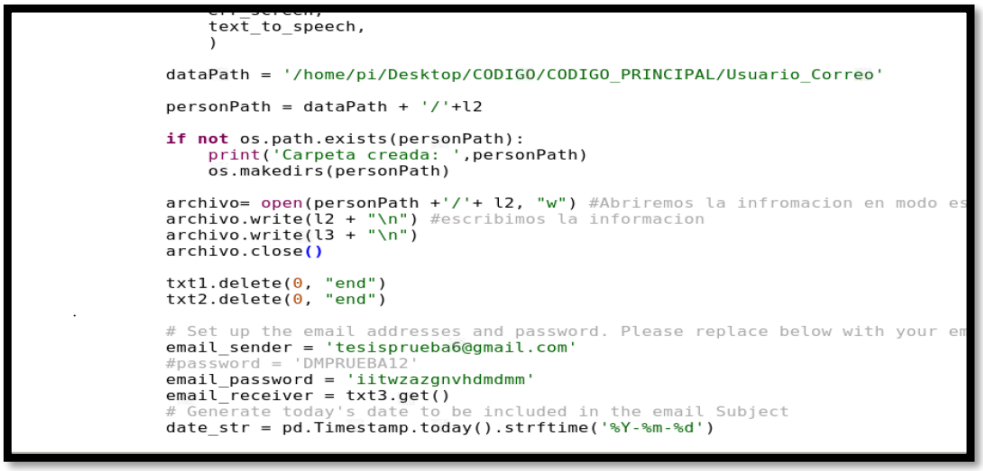
```
ImageUI = Tk()
ImageUI.title("Registro")
ImageUI.geometry("780x550")
ImageUI.configure(background="white")
ImageUI.resizable(0, 0)
titl = tk.Label(ImageUI, bg="white", relief=RIDGE, bd=10, font=("arial", 35))
titl.pack(fill=X)

# image and title
titl = tk.Label(
    ImageUI, text="REGISTRO DE USUARIO", bg="yellow", fg="blue", font=("arial",
    )
dialog_text = "Registro"

text_to_speech(dialog_text)
titl.place(x=160, y=12)
```

Figura 79. Programación de la ventana registro de Usuario

```
def take_image():

    l1 = txt1.get()
    l2 = txt2.get()
    l3 = txt3.get()

    takeImage.TakeImage(
        l1,
        l2,
        haarcasecade_path,
        trainimage_path,
        message,
        err_screen,
        text_to_speech,
        )

    dataPath = '/home/pi/Desktop/CODIGO/CODIGO_PRINCIPAL/Usuario_Correo'

    personPath = dataPath + '/'+l2

    if not os.path.exists(personPath):
        print('Carpeta creada: ',personPath)
```

Figura 80. Llamado de las variables de ingreso de datos del registro para la captura, almacenamiento y registro del usuario

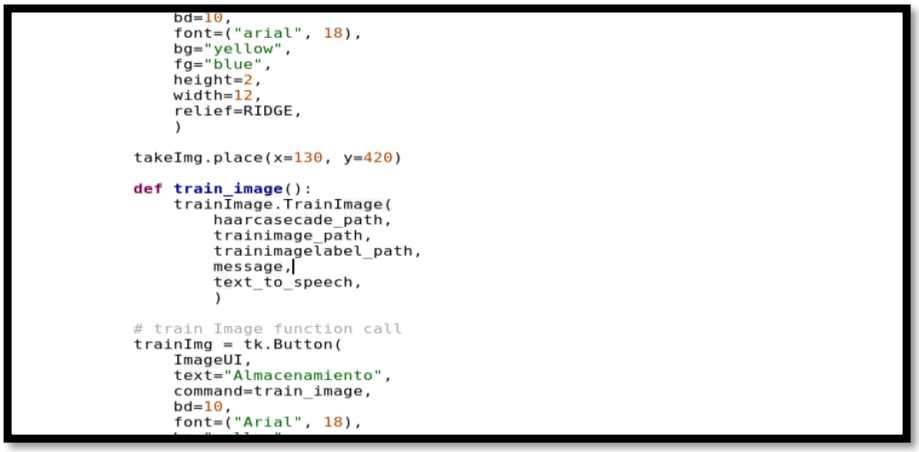
En la figura 81 se programa el envío de correo del usuario registrado.



```
        text_to_speech,
        )

dataPath = '/home/pi/Desktop/CODIGO/CODIGO_PRINCIPAL/Usuario_Correo'

personPath = dataPath + '/'+l2

if not os.path.exists(personPath):
    print('Carpeta creada: ',personPath)
    os.makedirs(personPath)

archivo= open(personPath +'/'+ l2, "w") #Abriremos la infromacion en modo es
archivo.write(l2 + "\n") #escribimos la informacion
archivo.write(l3 + "\n")
archivo.close()

txt1.delete(0, "end")
txt2.delete(0, "end")

# Set up the email addresses and password. Please replace below with your em
email_sender = 'tesisprueba6@gmail.com'
#password = 'DMPRUEBA12'
email_password = 'iitwzazgnvhdmdmm'
email_receiver = txt3.get()
# Generate today's date to be included in the email Subject
date_str = pd.Timestamp.today().strftime('%Y-%m-%d')
```

Figura 81. Programación para el envío de correo del registro del usuario.

En la figura 82 se crean las botoneras del módulo de seguridad.



```
        bd=10,
        font=("arial", 18),
        bg="yellow",
        fg="blue",
        height=2,
        width=12,
        relief=RIDGE,
        )

takeImg.place(x=130, y=420)

def train_image():
    trainImage.TrainImage(
        haarcasecade_path,
        trainimage_path,
        trainimagelabel_path,
        message,
        text_to_speech,
        )

# train Image function call
trainImg = tk.Button(
    ImageUI,
    text="Almacenamiento",
    command=train_image,
    bd=10,
    font=("Arial", 18),
```

Figura 82. Creación de botoneras.

En las figuras 83, 84, 85, 86 , 87 se obsevan las diferentes pantallas realizadas para el módulo de seguridad para los diferentes tipos de personas que son: el programador, supervisor y el usuario.

Figura 83. Pantalla del Menú del Programador (modelo LBPH)

Figura 84. Pantalla del modelo EigenFaces

Figura 85. Pantalla del modelo FisherFaces

Figura 86. Pantalla del Menú Supervisor

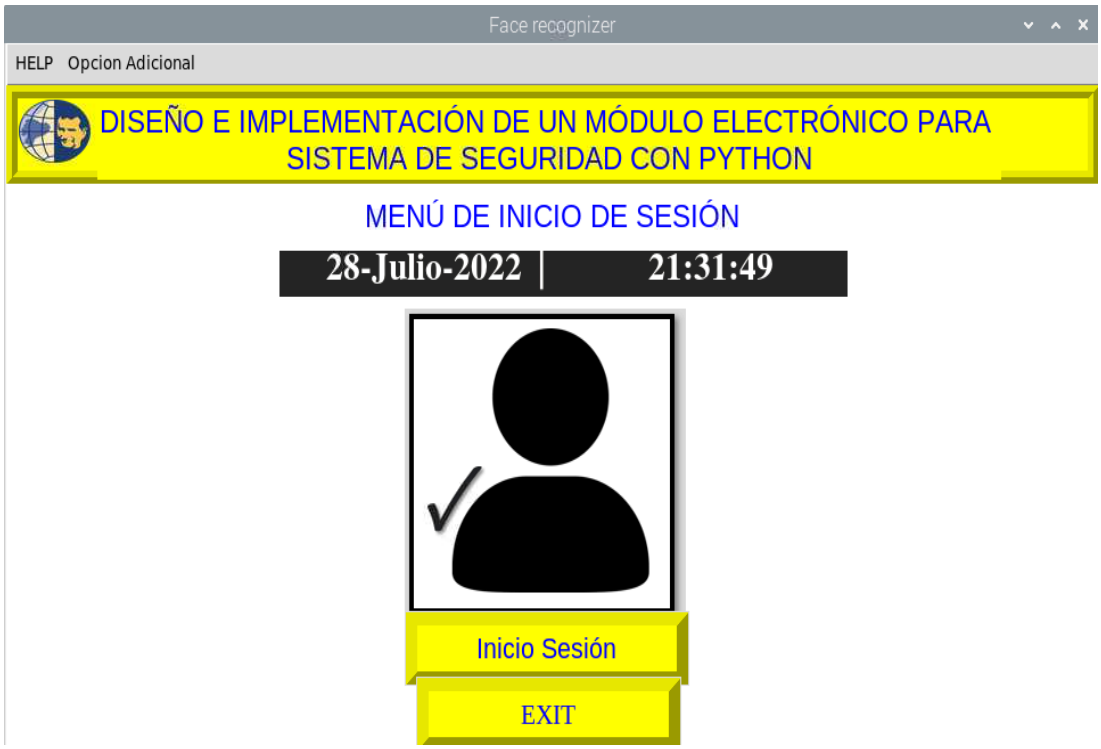


Figura 87. Pantalla de Inicio de Sesión para el Usuario

Programación de la captura de rostro

En la figura 88 se muestra los comandos utilizados para la programación de registro del usuario.

```
try:
    cam = cv2.VideoCapture(0)
    detector = cv2.CascadeClassifier(haarcasecade_path)

    Enrollment = l1 #variables de entrada
    Name = l2 #variables de entrada



    #carperta donde voy a guardar todo la imagen
    directory = Enrollment + "_" + Name
    path = os.path.join(trainimage_path, directory)
    os.mkdir(path)


    sampleNum = 0 #contador

    while True:

        ret, img = cam.read()
        if ret == False: break
        img =  imutils.resize(img, width=640)
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        auxFrame = img.copy()
        faces = detector.detectMultiScale(gray, 1.3, 5)
```

Figura 88. Programación para la captura de rostro

Programación de almacenamiento

En esta parte se muestra los comandos utilizados para el almacenamiento del rostro.

Se observa en la Figura 89, que se llama al pin 22, se crea el modelo de entrenamiento o almacenamiento que es LBPH y este a su vez detecta todas las imágenes almacenadas en el sistema y los lee en forma de cascada y los guarda en un archivo “.yml”.

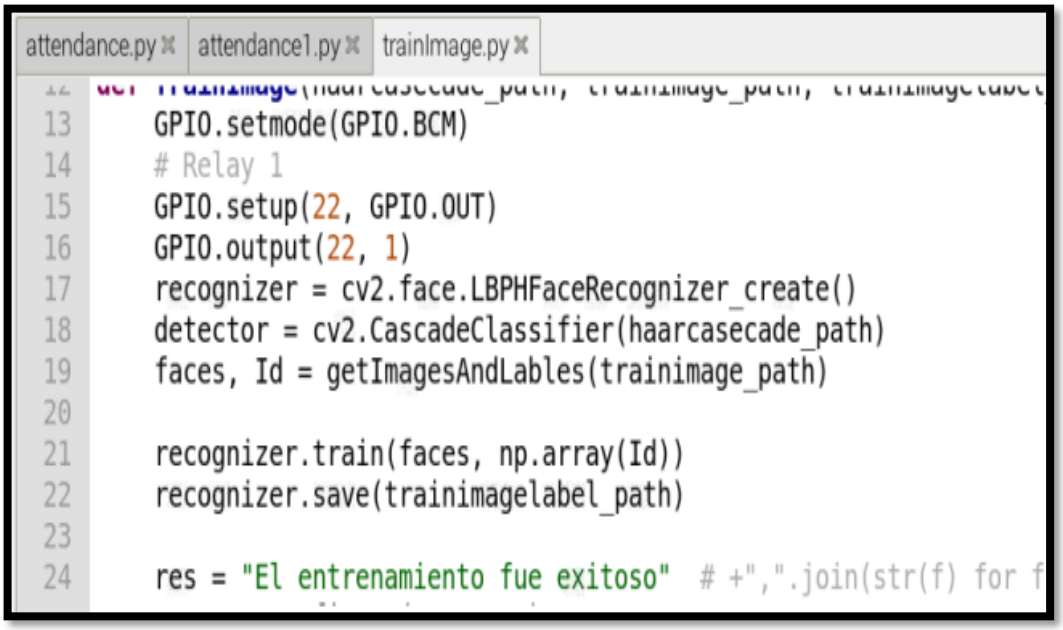


Figura 89. Programación del almacenamiento de rostro para los usuarios

Programación de reconocimiento facial (Inicio de Sesión)

En esta parte se muestra los comandos utilizados para el reconocimiento facial.

En la Figura 90 se ingresa el nombre del asunto, una vez hecho eso, se procede hacer clip al botón asistencia manual el cual dentro de este botón contienen una variable que se encarga de hacer todo el proceso de reconocimiento fácil mediante la comparativa de datos.

```
def subjectChoose(text_to_speech):
    GPIO.setmode(GPIO.BCM)
    # Relay 1
    GPIO.setup(23, GPIO.OUT)
    GPIO.setup(24, GPIO.OUT)
    GPIO.output(23, 1)
    GPIO.output(24, 1)
    dataPath2= "/home/pi/Desktop/CODIGO/CODIGO_PRINCIPAL/IPIMAGENRESULTADOS"


    dialog_text = "Inicio de sesion"
    text_to_speech(dialog_text)
    def FillAttendance():
        sub =  tx.get()
        now = time.time()
        future = now + 20
        print(now)
        print(future)
        if sub == "":
            t = "Por favor ingrese el nombre del sujeto!!!"
            text_to_speech(t)
        else:
            try:
                recognizer = cv2.face.LBPHFaceRecognizer_create()
```

Figura 90. Programación del reconocimiento facial

Programación para muestra de la hoja del registro

En las figuras 91 y 92 se programa la hoja de registro que muestras los usuarios que ingresan al sistema con el módulo de seguridad.

```
def subjectchoose(text_to_speech):
    dialog_text = "Registro de usuarios"
    text_to_speech(dialog_text)

    def calculate_attendance():

        Subject = tx.get()
        if Subject=="":
            t='Por favor ingrese el Asunto.'
            text_to_speech(t)
        os.chdir(f"/home/pi/Desktop/CODIGO/CODIGO_PRINCIPAL/Attendance/{Subject}")
        filenames = glob(f"/home/pi/Desktop/CODIGO/CODIGO_PRINCIPAL/Attendance/{Subject}/{Su
        df = [pd.read_csv(f) for f in filenames]
        newdf = df[0]
        for i in range(1, len(df)):
            newdf = newdf.merge(df[i], how="outer")
        newdf.fillna(0, inplace=True)
        newdf["Attendance"] = 0
        for i in range(len(newdf)):
            newdf["Attendance"].iloc[i] = str(int(round(newdf.iloc[i, 2:-1].mean() * 100)))+
            #newdf.sort_values(by=['Enrollment'],inplace=True)
        newdf.to_csv("attendance.csv", index=False)

        root = tkinter.Tk()
        root.title("Registro de "+Subject)
```

Figura 91. Programación de la hoja del registro parte 1

```
subject = Tk()
subject.title("Subject...")
subject.geometry("580x320")
subject.resizable(0, 0)
subject.configure(background="white")
# subject_logo = Image.open("UI_Image/0004.png")
# subject_logo = subject_logo.resize((50, 47), Image.ANTIALIAS)
# subject_logo1 = ImageTk.PhotoImage(subject_logo)
titl = tk.Label(subject, bg="yellow", relief=RIDGE, bd=10, font=("arial", 30))
titl.pack(fill=X)
# l1 = tk.Label(subject, image=subject_logo1, bg="black",)
# l1.place(x=100, y=10)
titl = tk.Label(
    subject,
    text="Tabla del registro",
    bg="yellow",
    fg="blue",
    font=("arial", 25),
)
titl.place(x=170, y=12)

def Attf():
```

Figura 92. Programación de la hoja del registro parte 2

ANEXO F. EJERCICIOS HECHO CON EL MÓDULO

OBJETIVO

- Desarrollo de 5 prácticas para uso del módulo didáctico.

DESARROLLO DE LAS 5 PRACTICAS PARA USO DEL MÓDULO DIDÁCTICA

1. Registrar a 3 personas e iniciar sección con el modelo LBPH con la cámara del módulo (Detalle los resultados obtenidos)

2. Iniciar sección con el modelo EigenFaces con la cámara del módulo (Detalle los resultados obtenidos)

3. Iniciar sección con el modelo FisherFaces con la cámara del módulo (Detalle los resultados obtenidos)

4. Iniciar sección con los tres modelos de entrenamiento con la cámara del celular (Detalle los resultados obtenidos)

5. Con DOS personas no registradas Inicie sección con los modelos de entrenamiento tanto cámara del módulo y la cámara del celular (Detalle los resultados obtenidos)

RESULTADOS OBTENIDOS

USUARIOS REGISTRADOS PARA LOS MODELOS DE ENTRENAMIENTO

Se registraron a 3 usuarios, en la figura 93 se observa las carpetas de los usuarios registrados. En figura 94, 95 y 96 se ve la captura del rostro de los usuarios registrados en el sistema y se asignó a 2 usuarios no registrados para probar los distintos modelos de entrenamiento programados en el sistema para así determinar el mejor modelo a utilizar para el registro y ingreso de los usuarios.

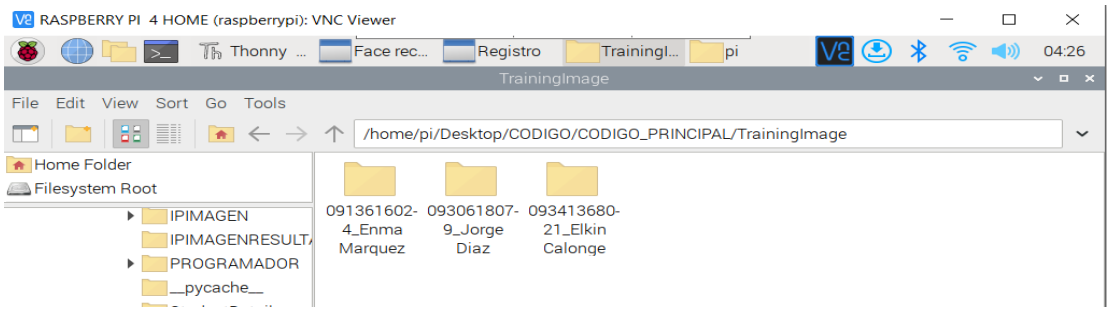


Figura 93. Carpetas de Usuarios registrados

Figura 94. Usuario 1 Jorge Diaz Registrado

Figura 95. Usuario 3 Elkin Calonge Registrado

Figura 96. Usuario 4 Enma Marquez Registrado

TABLA DE RESULTADOS DE INGRESO DE USUARIOS

USUARIOS	MODELO LBPH	MODELO LBPH CAMARA CELL.	MODELO FISHERFACE	MODELO FISHERFACE CAMARA CELL.	MODELO EIGENFACES	MODELO EIGENFACES CAMARA CELL.
USUARIO 1	2	2	1	2	1	2
USUARIO 3	2	2	1	2	2	2
USUARIO 4	2	2	1	1	1	1
USUARIO NO REGISTRADO 1	1	1	1	1	2	2
USUARIO NO REGISTRADO 2	1	1	1	2	2	1
SI INGRESO= 2						
NO INGRESO= 1						

Tabla 5. Tabla de resultados de ingreso de usuarios

GRAFICA DE ANÁLISIS DE INGRESO DE USUARIOS

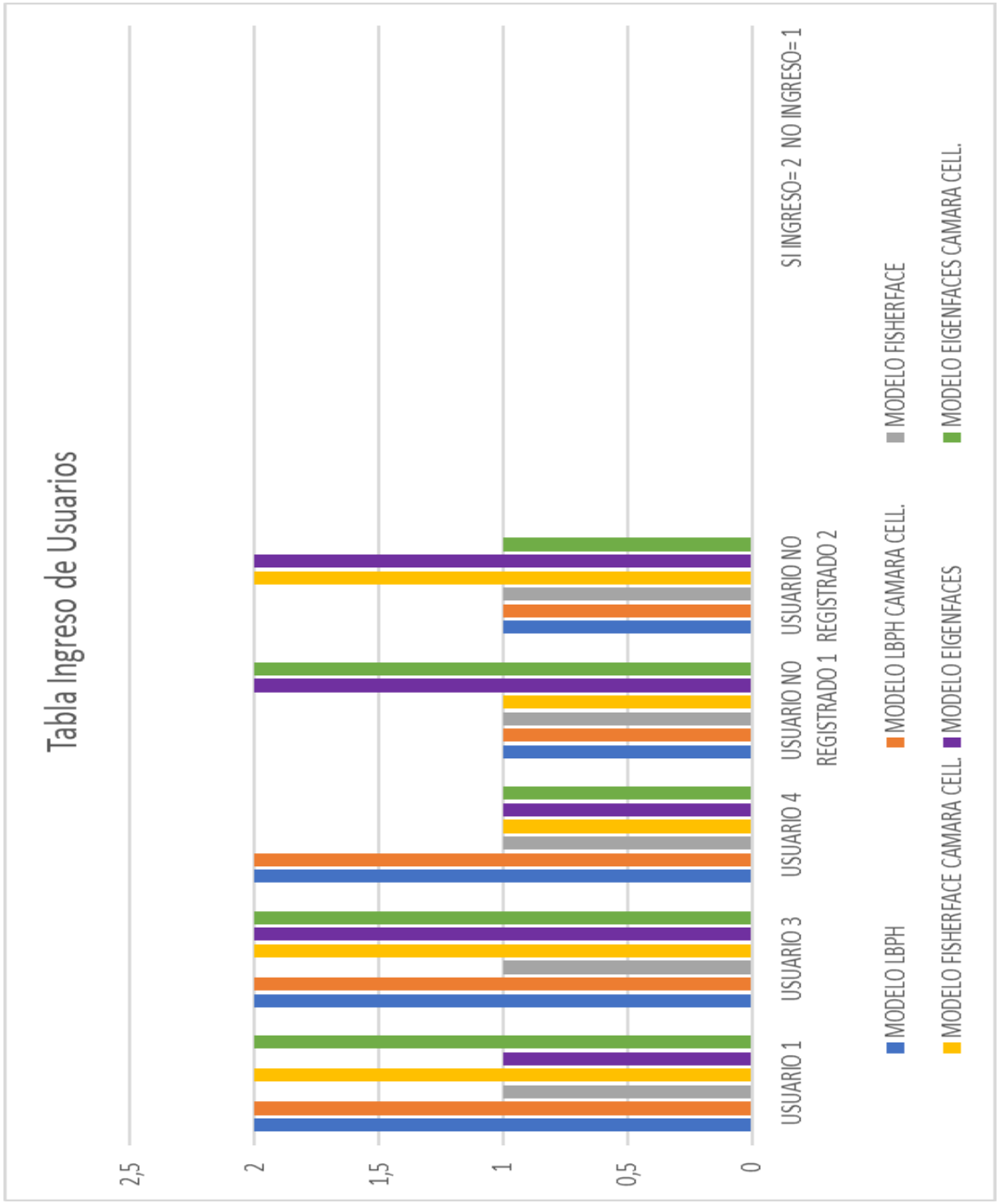


Figura 97. Grafica Ingreso de Usuarios

En la figura 97 se observa la gráfica Ingreso de usuarios con respecto a los resultados obtenidos en la tabla 5 de los resultados de ingreso de usuarios se obtiene como análisis:

USUARIO 1

USUARIO NÚMERO 1, JORGE JOHAN DIAZ:

MODELO DE ENTRENAMIENTO LBPH: Ingreso exitoso.

MODELO DE ENTRENAMIENTO LBPH CÁMARA CELULAR: Ingreso exitoso.

MODELO DE ENTRENAMIENTO FISHERFACE: Detectado como usuario no registrado.

MODELO DE ENTRENAMIENTO FISHERFACE CÁMARA CELULAR: Ingreso exitoso.

MODELO DE ENTRENAMIENTO EIGENFACE: No fue exitoso, detectado como usuario número 3.

MODELO DE ENTRENAMIENTO EIGENFACE CÁMARA CELULAR: Ingreso exitoso.

USUARIO 3

USUARIO NÚMERO 3, ELKIN CALONGE:

MODELO DE ENTRENAMIENTO LBPH: Ingreso exitoso.

MODELO DE ENTRENAMIENTO LBPH CÁMARA CELULAR: Ingreso exitoso.

MODELO DE ENTRENAMIENTO FISHERFACE: Detectado como usuario no registrado.

MODELO DE ENTRENAMIENTO FISHERFACE CÁMARA CELULAR: Ingreso exitoso.

MODELO DE ENTRENAMIENTO EIGENFACE: Ingreso exitoso.

MODELO DE ENTRENAMIENTO EIGENFACE CÁMARA CELULAR: Ingreso exitoso.

USUARIO 4

USUARIO NÚMERO 4, ENMA MARQUEZ:

MODELO DE ENTRENAMIENTO LBPH: Ingreso exitoso.

MODELO DE ENTRENAMIENTO LBPH CÁMARA CELULAR: Ingreso exitoso.

MODELO DE ENTRENAMIENTO FISHERFACE: No fue exitoso, detectado como usuario número 1.

MODELO DE ENTRENAMIENTO FISHERFACE CÁMARA CELULAR: No fue exitoso, detectado como usuario número 1.

MODELO DE ENTRENAMIENTO EIGENFACE: Detectado como usuario no registrado.

MODELO DE ENTRENAMIENTO EIGENFACE CÁMARA CELULAR: Detectado como usuario no registrado.

USUARIO NO REGISTRADO 1

MODELO DE ENTRENAMIENTO LBPH: Usuario no registrado.

MODELO DE ENTRENAMIENTO LBPH CÁMARA CELULAR: Usuario no registrado.

MODELO DE ENTRENAMIENTO FISHERFACE: Usuario no registrado.

MODELO DE ENTRENAMIENTO FISHERFACE CÁMARA CELULAR: Usuario no registrado.

MODELO DE ENTRENAMIENTO EIGENFACE: No fue exitoso, detectado como usuario 3.

MODELO DE ENTRENAMIENTO EIGENFACE CÁMARA CELULAR: No fue exitoso, detectado como usuario 4.

USUARIO NO REGISTRADO 2

MODELO DE ENTRENAMIENTO LBPH: Usuario no registrado.

MODELO DE ENTRENAMIENTO LBPH CÁMARA CELULAR: Usuario no registrado.

MODELO DE ENTRENAMIENTO FISHERFACE: Usuario no registrado.

MODELO DE ENTRENAMIENTO FISHERFACE CÁMARA CELULAR: No fue exitoso, detectado como usuario 4.

MODELO DE ENTRENAMIENTO EIGENFACE: No fue exitoso, detectado como usuario 3.

MODELO DE ENTRENAMIENTO EIGENFACE CÁMARA CELULAR: Usuario no registrado.

CAPTURAS DE LOS RESULTADOS DE LAS PRÁCTICAS

Figura 98. Mensaje de registro Usuario 1

Figura 99. Mensaje de Ingreso con la cámara del módulo del Usuario 1

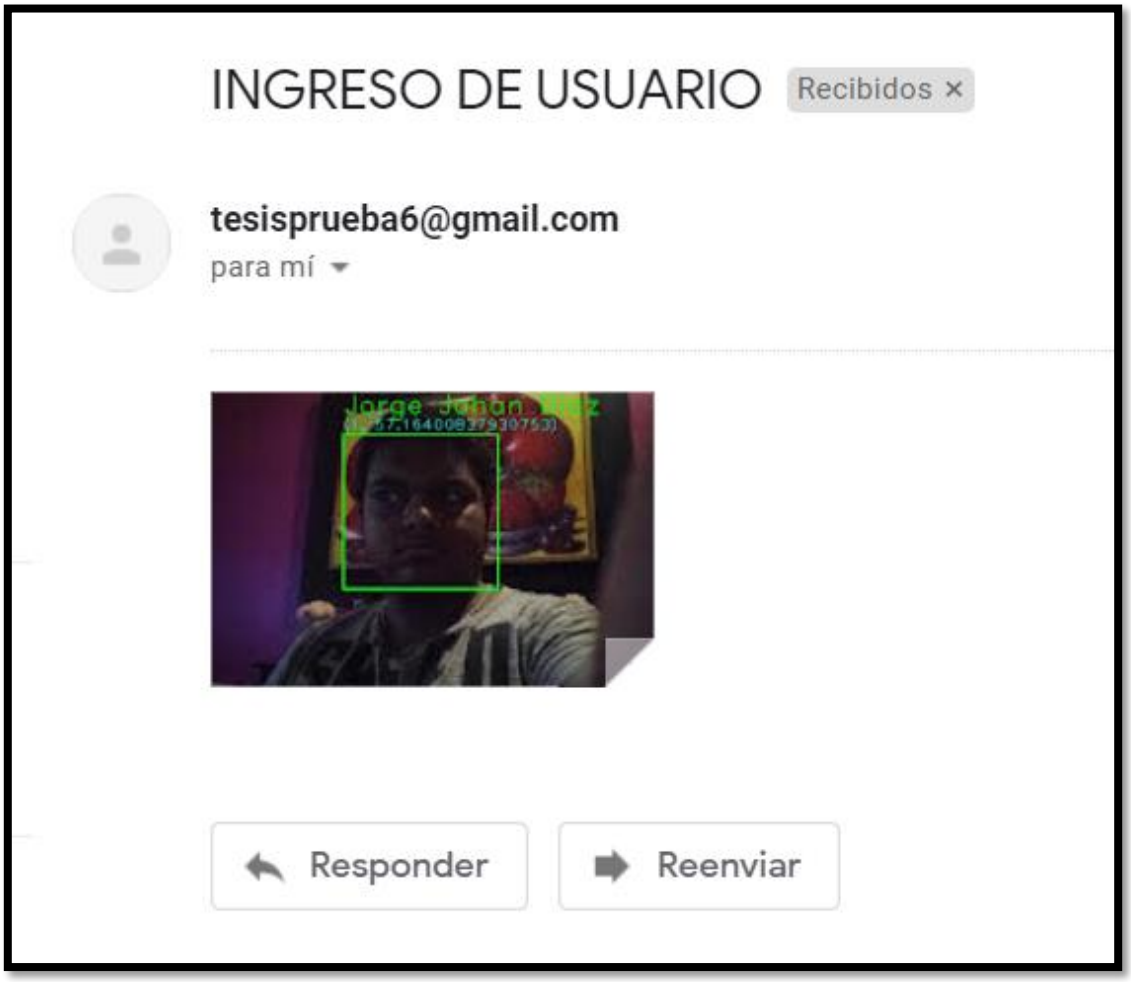


Figura 100. Mensaje de Ingreso con la cámara del celular del Usuario 1

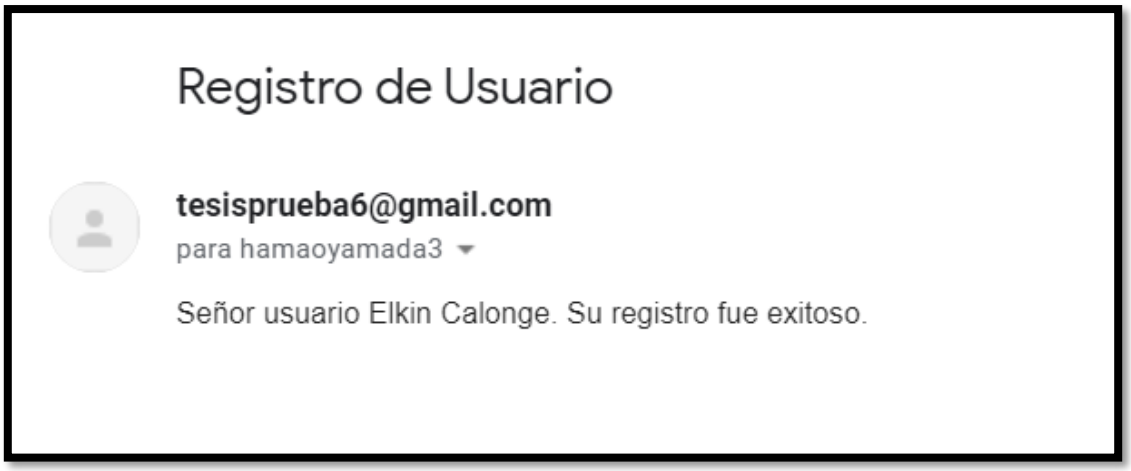


Figura 101. Mensaje de registro Usuario 3

Figura 102. Mensaje de Ingreso con la cámara del módulo del Usuario 3

Figura 103. Mensaje de Ingreso con la cámara del celular del Usuario 3

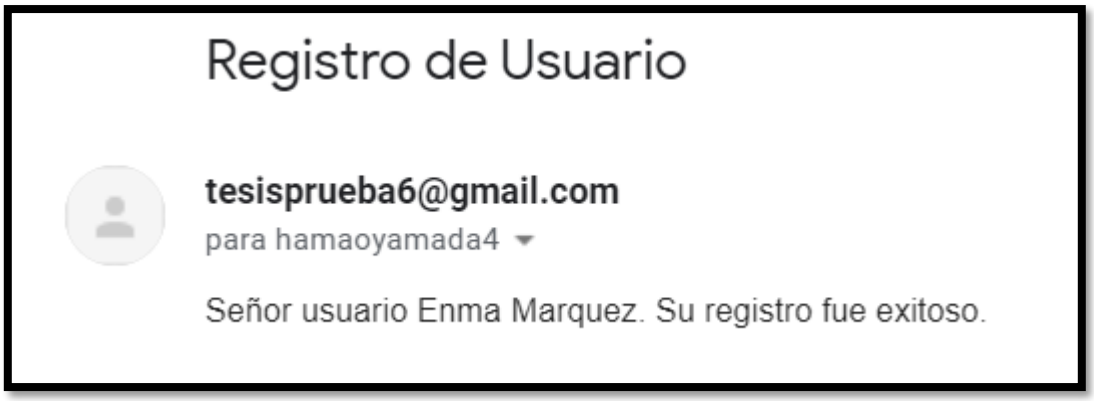


Figura 104. Mensaje de registro Usuario 4

Figura 105. Mensaje de Ingreso con la cámara del módulo del Usuario 4

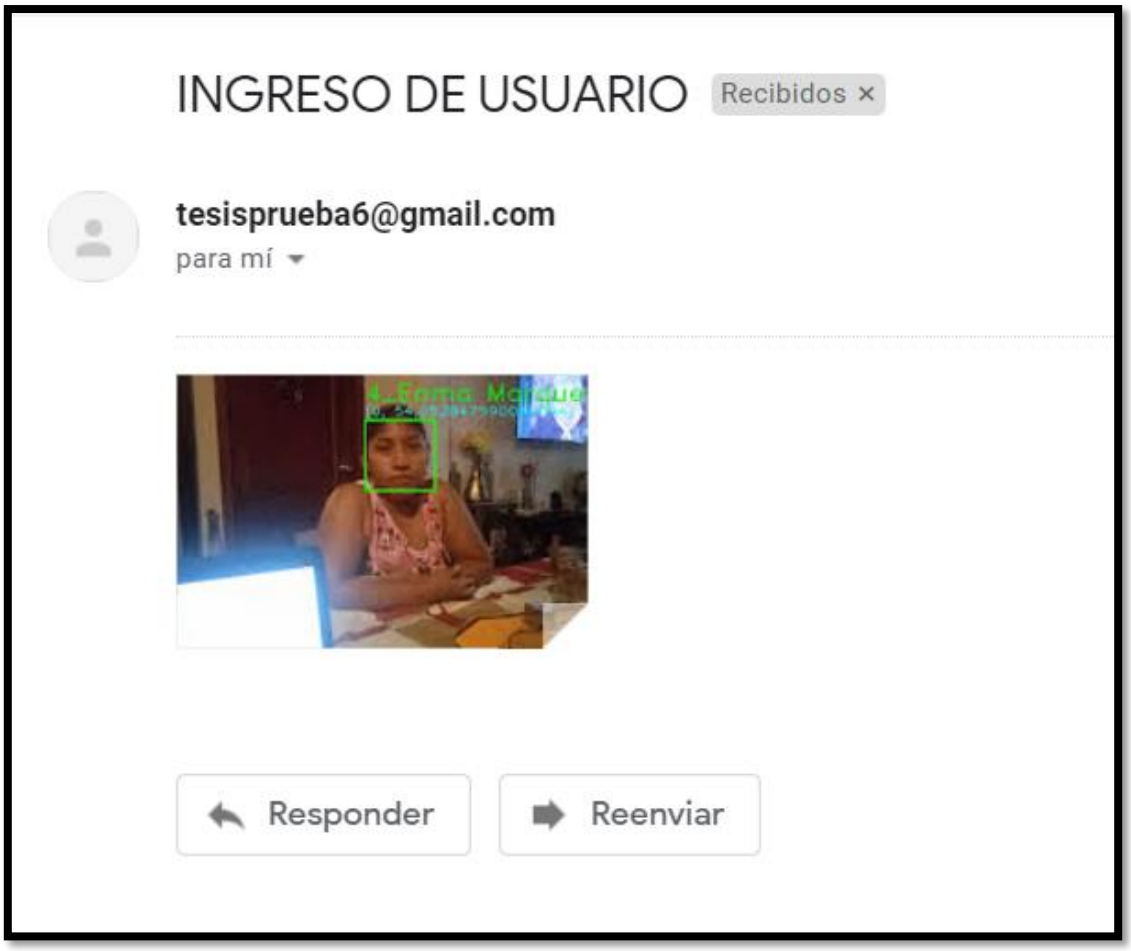


Figura 106. Mensaje de Ingreso con la cámara del celular del Usuario 4

Figura 107. Mensaje de usuario sospechoso 1

Figura 108. Mensaje de usuario sospechoso 2

ANÁLISIS GENERAL AL REALIZAR LAS 5 PRACTICAS

Como análisis general al terminar estas 5 practicas, el resultado obtenido es que como mejor modelo de inicio de sección es el modelo LBPH; los modelo FisherFaces y EigenFaces no son confiables ya que ocupan mayor espacio de almacenamiento, por ende ralentizan al sistema y son sensibles a la luz dando resultados erróneos al momento de iniciar sección.