



Universidad de San Carlos de Guatemala  
Facultad de Ingeniería  
Escuela de Mecánica Eléctrica

**DISEÑO DE UN SISTEMA BIOMÉTRICO PARA EL CONTROL DE ASISTENCIA DEL  
PERSONAL ADMINISTRATIVO DE LA ESCUELA SUPERIOR DE ARTE DE LA  
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

**Gabriel Esaú Can López**

Asesorado por el Ing. Kenneth Issur Estrada Ruiz

Guatemala, octubre de 2023



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**DISEÑO DE UN SISTEMA BIOMÉTRICO PARA EL CONTROL DE ASISTENCIA DEL  
PERSONAL ADMINISTRATIVO DE LA ESCUELA SUPERIOR DE ARTE DE LA  
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA  
FACULTAD DE INGENIERÍA  
POR

**GABRIEL ESAÚ CAN LÓPEZ**

ASESORADO POR EL ING. KENNETH ISSUR ESTRADA RUÍZ

AL CONFERÍRSELE EL TÍTULO DE

**INGENIERO ELECTRICISTA**

GUATEMALA, OCTUBRE DE 2023



UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERÍA



**NÓMINA DE JUNTA DIRECTIVA**

DECANO	Ing. José Francisco Gómez Rivera (a.i.)
VOCAL II	Ing. Mario Renato Escobedo Martinez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Ing. Kevin Vladimir Armando Cruz Lorente
VOCAL V	Br. Fernando José Paz Gonzáles
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

**TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO**

DECANO	Ing. José Francisco Gómez Rivera (a.i.)
EXAMINADOR	Ing. Kenneth Issur Estrada Ruiz
EXAMINADOR	Ing. Armando Alonso Rivera Carrillo
EXAMINADOR	Ing. Natanael Jonathan Requena Gómez
SECRETARIO	Ing. Hugo Humberto Rivera Pérez



## **HONORABLE TRIBUNAL EXAMINADOR**

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

**DISEÑO DE UN SISTEMA BIOMETRICO PARA EL CONTROL DE ASISTENCIA DEL  
PERSONAL ADMINISTRATIVO DE LA ESCUELA SUPERIOR DE ARTE DE LA  
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Tema que me fuera asignado por la Dirección de Escuela de Mecánica Eléctrica de ingeniería con fecha 9 de marzo de 2022.



**Gabriel Esau Can López**



Universidad de San Carlos de  
Guatemala



Facultad de Ingeniería  
Unidad de EPS

Guatemala, 17 de agosto de 2023.  
REF.EPS.DOC.352.08.2023.

Ing. Oscar Argueta Hernández  
Director Unidad de EPS  
Facultad de Ingeniería  
Presente

Estimado Ingeniero Argueta Hernández.

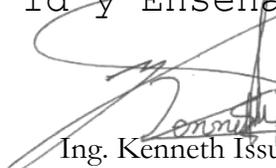
Por este medio atentamente le informo que como Asesor-Supervisor de la Práctica del Ejercicio Profesional Supervisado (E.P.S.), del estudiante universitario **Gabriel Esaú Can López** de la Carrera de Ingeniería Eléctrica, Registro Académico No. **201807395** y **CUI 3055 64439 0301**, procedí a revisar el informe final, cuyo título es **“DISEÑO DE UN SISTEMA BIOMETRICO PARA EL CONTROL DE ASISTENCIA DEL PERSONAL ADMINISTRATIVO DE LA ESCUELA SUPERIOR DE ARTE DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA”**.

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

“Id y Enseñad a Todos”

  
  
Ing. Kenneth Issur Estrada Ruiz  
Asesor-Supervisor de EPS  
Área de Ingeniería Eléctrica

c.c. Archivo  
KIER/ra



Universidad de San Carlos de  
Guatemala



Facultad de Ingeniería  
Unidad de EPS

Guatemala 28 de agosto de 2023.  
REF.EPS.D.254.08.2023.

Ing. Armando Alonso Rivera Carrillo  
Director Escuela de Ingeniería Mecánica Eléctrica  
Facultad de Ingeniería  
Presente

Estimado Ingeniero Rivera Carrillo.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **"DISEÑO DE UN SISTEMA BIOMETRICO PARA EL CONTROL DE ASISTENCIA DEL PERSONAL ADMINISTRATIVO DE LA ESCUELA SUPERIOR DE ARTE DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"** que fue desarrollado por el estudiante universitario, **Gabriel Esaú Can López**, quien fue debidamente asesorado y supervisado por el Ing. Kenneth Issur Estrada Ruiz.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor y Supervisor de EPS, en mi calidad de Director apruebo su contenido solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,  
"Id y Enseñad a Todos"

Ing. Oscar Argueta Hernández  
Director Unidad de EPS

/ra



REF. EIME 48.2023.

El director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del director de EPS, del asesor, con el Visto Bueno del coordinador de área, al Informe final de EPS del estudiante Gabriel Esaú Can López: **"DISEÑO DE UN SISTEMA BIOMETRICO PARA EL CONTROL DE ASISTENCIA DEL PERSONAL ADMINISTRATIVO DE LA ESCUELA SUPERIOR DE ARTE DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**, procede a la autorización Correspondiente.



Ing. Armando Alonso Rivera Carrillo

Guatemala, 15 de septiembre de 2023.



LNG.DECANATO.OI.717.2023

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **DISEÑO DE UN SISTEMA BIOMÉTRICO PARA EL CONTROL DE ASISTENCIA DEL PERSONAL ADMINISTRATIVO DE LA ESCUELA SUPERIOR DE ARTE DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por: **Gabriel Esaú Can López**, después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

  
Ing. José Francisco Gómez Rivera  
Decano a.i.



Guatemala, octubre de 2023

JFGR/gaoc



## **ACTO QUE DEDICO A:**

- Dios** Por ser quien me ha dado toda la sabiduría y paciencia para llegar hasta este punto de mi vida profesional.
- Mis padres** Gabriel Can y Barbara López, por apoyarme en el transcurso de mi carrera de forma incondicional por aconsejarme y guiarme por el camino correcto.
- Mis hermanos** Esteban Can y Joel Can por estar presentes y ayudarme en los retos que he afrontado a lo largo de mi vida estudiantil.
- Mis tíos** David López, Alberto López e Isabel López por ser un apoyo moral y motivarme en todo momento a culminar mis estudios.
- Mis amigos** Carlos Fuentes y Pablo Bravo, por ser una fuente de inspiración y apoyo en el transcurso de mis estudios siendo de las principales personas que me motivaron a culminar este objetivo.



## **AGRADECIMIENTOS A:**

**Universidad de San  
Carlos de  
Guatemala**

Por ser mi casa de estudios y brindarme la oportunidad de formarme como profesional.

**Mi madre**

Bárbara Elizabeth por todo el amor y cariño incondicional que me proporciono en este tiempo, por todo el apoyo y motivación que me dio para cumplir con esta meta.

**Mi padre**

Gabriel Can por motivarme a ser un profesional estando presente en toda mi formación, por el apoyo y consejos que me dio durante todos estos años de carrera universitaria.

**Mis tíos**

David López, Alberto López e Isabel López por ser un apoyo moral y sentirse orgullosos de los logros que he conseguido a lo largo de mi vida.

**Mis amigos**

Carlos Fuentes, Pablo Bravo, Manuel López, por haber sido un gran apoyo emocional en el transcurso de mi carrera universitaria.



# ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	V
LISTA DE SÍMBOLOS .....	VII
GLOSARIO .....	IX
RESUMEN.....	XI
OBJETIVOS.....	XIII
INTRODUCCION.....	XV
1. ANTECEDENTES GENERALES DE LA ESCUELA SUPERIOR DE ARTES DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA.....	1
1.1. Historia de la institución.....	1
1.2. Misión .....	2
1.3. Visión.....	2
1.4. Organigrama.....	3
1.5. Ubicación geográfica .....	3
2. MARCO TEÓRICO.....	5
2.1. Arduino .....	5
2.1.1. Componentes de Arduino .....	6
2.1.1.1. Entradas y salidas .....	6
2.1.1.2. Salidas de alimentación .....	7
2.1.1.3. Salidas de comunicación serial.....	7
2.1.1.4. Control de velocidad de ejecución .....	8
2.1.1.5. Microcontrolador .....	9

2.1.2.	Tipos de Arduino .....		9
	2.1.2.1.	Arduino Uno .....	10
	2.1.2.2.	Arduino Leonardo.....	10
	2.1.2.3.	Arduino 101 .....	10
	2.1.2.4.	Arduino Zero.....	11
	2.1.2.5.	Arduino Mega 2560 .....	11
2.1.3.	Arduino IDE .....		11
2.2.	Sensores .....		13
	2.2.1.	Tipos de sensores .....	13
	2.2.2.	Sensores por magnetismo.....	14
	2.2.3.	Sensor por efecto Hall.....	14
	2.2.4.	Sensor por conductividad eléctrica.....	14
	2.2.5.	Sensores termoelectricos .....	15
	2.2.6.	Sensores fotoelectricos .....	15
	2.2.7.	Sensores piezoelectricos.....	15
	2.2.8.	Sensores por ultrasonido.....	15
	2.2.9.	Sensores por radiofrecuencia.....	16
2.3.	Actuadores .....		16
	2.3.1.	Electromagneticos.....	16
	2.3.2.	Electromotores .....	16
2.4.	Sistema biométrico.....		17
	2.4.1.	Captura de información .....	18
	2.4.2.	Extracción de las características .....	19
	2.4.3.	Comparación .....	19
	2.4.4.	Selección o filtrado .....	19
	2.4.5.	Almacenamiento de los datos .....	19
2.5.	Tipos de detectores biométricos .....		20
	2.5.1.	Sensor de huella digital .....	20
		2.5.1.1. Escáner óptico.....	21

	2.5.1.2.	Escáner capacitivo.....	21
	2.5.1.3.	Escáner por ultrasonido.....	21
	2.5.1.4.	Escáner térmico.....	21
	2.5.1.5.	Lenguajes de programación .....	21
2.6.		Python .....	22
3.		MARCO METODOLÓGICO .....	23
3.1.		Diagnóstico de la situación actual .....	23
3.2.		Diseño teórico del sistema biométrico para el control de asistencia.....	23
3.3.		Diagrama de Bloques .....	25
3.4.		Dispositivos utilizados.....	26
3.5.		Diseño teórico de conexión de equipo.....	28
4.		DISEÑO PROPUESTO .....	29
4.1.		Código de interfaz gráfica.....	29
4.2.		Código de Arduino .....	35
4.3.		Código para la base datos.....	37
4.4.		Diagrama de conexión de Arduino .....	41
4.5.		Simulación de circuito electrónico .....	42
5.		MANUAL DE USUARIO Y MANTENIMIENTO .....	45
5.1.		Ventana principal.....	45
5.2.		Venta de registro .....	51
6.		RESULTADOS .....	55
		CONCLUSIONES .....	57
		RECOMENDACIONES .....	59

REFERENCIAS .....61  
ANEXOS.....63

# ÍNDICE DE ILUSTRACIONES

## FIGURAS

<b>Figura 1.</b>	Organigrama ESA.....	3
<b>Figura 2.</b>	Foto aérea.....	4
<b>Figura 3.</b>	Placa Arduino UNO R3.....	5
<b>Figura 4.</b>	Pines de entrada y salida.....	6
<b>Figura 5.</b>	Salidas de alimentación para Arduino.....	7
<b>Figura 6.</b>	Puertos de comunicación.....	8
<b>Figura 7.</b>	Oscilador de frecuencia.....	8
<b>Figura 8.</b>	Microcontrolador.....	9
<b>Figura 9.</b>	<i>Software</i> Arduino.....	12
<b>Figura 10.</b>	Sistema biométrico.....	18
<b>Figura 11.</b>	Diagrama de bloques.....	26
<b>Figura 12.</b>	Conexión de sensor óptico con Arduino.....	28
<b>Figura 13.</b>	Conexión de sensor óptico con Arduino.....	41
<b>Figura 14.</b>	Huellas dactilares.....	42
<b>Figura 15.</b>	Simulación.....	43
<b>Figura 16.</b>	Ventana principal.....	46
<b>Figura 17.</b>	Mensaje en monitor serial.....	47
<b>Figura 18.</b>	Ingreso de una persona a las instalaciones.....	48
<b>Figura 19.</b>	Ventana de control de ingreso y salida de personal.....	49
<b>Figura 21.</b>	Ventana de emergencia.....	50
<b>Figura 22.</b>	Control de ingreso y salida de personal en Excel.....	50
<b>Figura 23.</b>	Ventana de ingreso de personal.....	51
<b>Figura 24.</b>	Ventana con datos registrados.....	52



## LISTA DE SÍMBOLOS

<b>Símbolo</b>	<b>Significado</b>
<b>A</b>	Amperio
<b>AC</b>	Corriente alterna
<b>DC</b>	Corriente directa
<b>Hz</b>	Hertz
<b>Kb</b>	Kilo bit
<b>KB</b>	Kilo byte
<b><math>\Omega</math></b>	Ohm
<b>V</b>	Voltios
<b>W</b>	Watts



## GLOSARIO

<b>Bootloader</b>	Es un <i>software</i> alojado en la memoria flash que nos permite programar Arduino a través del puerto serie sin necesidad de usar un programador externo.
<b>COM</b>	Es un puerto de comunicación que permite a un ordenador comunicarse con un dispositivo serie.
<b>ESA</b>	Escuela Superior de Artes
<b>ICSP</b>	Es el acrónimo en inglés que significa programación serial en circuito, esta es una tecnología que permite la reprogramación de un microchip sin la necesidad de removerla del PBC donde este instalado.
<b>IDE</b>	Es una aplicación de <i>software</i> que ayuda a los programadores a desarrollar código de <i>software</i> de manera más efectiva.
<b>JACK</b>	Es un conector para comparar señales analógicas.
<b>PCB</b>	Por sus siglas en inglés significa <i>printed circuit board</i> y es una placa de circuito impreso utilizada para los circuitos electrónicos.

**POO**

Significa programación orientada a objetos y es un paradigma de programación

**PWM**

Por sus siglas en ingles significa modulación de ancho de pulso y se utiliza para modular la cantidad de energía que se transmite.

## RESUMEN

En este informe final sobre el proyecto titulado: diseño de un sistema biométrico para el control de asistencia del personal administrativo de la escuela superior de arte de la universidad de San Carlos de Guatemala, detallamos los distintos tipos de dispositivos utilizados para crear el sistema y las distintas opciones que pudimos utilizar, a la vez explicamos por qué escogimos cada dispositivo, se plantea el modelo inicial en el cual se basó el diseño final.

A partir de la elección del sensor biométrico se decidieron los dispositivos a utilizar para transmitir los datos y conectarse al ordenador, de esta forma también escogimos el lenguaje de programación que tuviera la mayor accesibilidad a conectarse con el transmisor de la información.

En el diseño final se presenta las conexiones necesarias para poder implementarse de una forma física, también se comparte el código de programación desarrollado para el correcto funcionamiento y se da un manual de funcionamiento donde se detalla cuáles son las opciones que presenta el programa y la forma correcta en que debe interactuar el usuario con el programa, se comparte tanto el código principal de la aplicación así como los códigos creados para funciones específicas, como los son la transmisión de datos desde el microcontrolador como el código de la base de datos.



## **OBJETIVOS**

### **General**

Diseñar un sistema biométrico para el control de asistencia del personal administrativo de la Escuela Superior de Arte de la universidad de San Carlos de Guatemala.

### **Específicos**

1. Realizar una base de datos para la administración de las huellas dactilares del personal administrativo de la Escuela Superior de Arte.
2. Realizar una interfaz gráfica para el control y registro de las huellas dactilares del personal administrativo de la Escuela Superior de Arte.
3. Realizar un algoritmo para la recopilación y comparación de huellas dactilares del personal administrativo de la Escuela Superior de Arte.



## INTRODUCCION

El proyecto realizado dentro de la Escuela Superior de Artes de la Universidad de San Carlos de Guatemala fue realizar un diseño para la implementación de un sistema biométrico para tomar la asistencia del personal administrativo con la finalidad de poder reducir costos y optimizar la recopilación de esta información al evitar la digitación de los listados.

La realización de este diseño abarcará desde un diseño electrónico en el cual se tendrá que disponer de que materiales serán los necesarios para recopilar y almacenar las huellas digitales y el diseño de una aplicación grafica para la interacción y control del registro de huellas, así mismo se tendrá que realizar una base de datos en la cual se tendrá almacenados los datos de los empleados y se registrará la hora de salida y entrada de cada empleado.

El sistema biométrico a diseñar será diseñado mediante la utilización de un *software* de simulación y utilizando como microcontrolador un Arduino y un sensor de huellas dactilares para el registro y verificación del personal administrativo, se utilizaran estos dispositivos dado que el sistema no requiere mucha potencia de procesamiento y se requiere un bajo costo para ser rentable, para el diseño de la interfaz gráfica se utilizara el lenguaje de programación Python debido a las accesibilidad para crear un enlace con la plataforma de Arduino.



# **1. ANTECEDENTES GENERALES DE LA ESCUELA SUPERIOR DE ARTES DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

## **1.1. Historia de la institución**

La creación de la Escuela Superior de Arte (ESA) fue el producto del trabajo y múltiples gestiones administrativas que duraron aproximadamente 22 años, Se crea la primera comisión para el desarrollo de este proyecto en el acuerdo de Rectoría No. 461-85 en el cual se nombran a los maestros Marco Augusto Quiroa, Antonio Crespo Morales, Víctor Hugo Cruz y Jorge Álvaro Sarmientos, junto a representantes de la USAC tales como el Licenciado Luis Vallejo y al ingeniero Jorge Cáceres para la creación del anteproyecto global, donde se establecen las fases de trabajo y procedimientos que seguirá la comisión para la creación del proyecto final (Morales,2010).

En el año 2002 se emite el acuerdo de Rectoría No. 443-2002 donde se amplía el acuerdo de creación de la comisión y se incluye formalmente a los profesionales antes mencionados los cuales estuvieron trabajando en este proyecto de forma totalmente desinteresado y Ad-honorem (Morales,2010).

La comisión creada hace entrega del proyecto final en junio de 2002, siendo este proyecto aprobado en la sesión del 19 de abril de 2006 en el acuerdo de rectoría No. 602-2006, donde se amplía y modifica el Acuerdo de Rectoría No. 443-2002 para nombrar la Comisión de Acompañamiento para la Ejecución del Proyecto de la Escuela superior de Arte (Morales,2010).

## **1.2. Misión**

según la página oficial de la Escuela Superior de Artes (2023) la misión de esta escuela es “Formar artistas con capacidad de caracterizar lo guatemalteco y universalizarlo, fomentando el desarrollo social a través de los valores éticos y estéticos contenidos en la creación artística”.

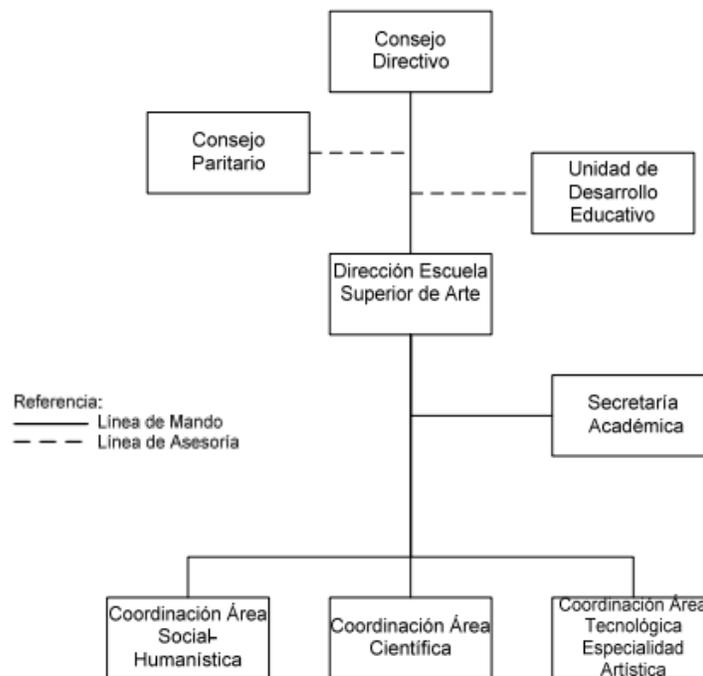
## **1.3. Visión**

según la página oficial de la Escuela Superior de Artes (2023) la misión de esta escuela es "Elevar el nivel cultural del guatemalteco, a través de la creación artística, enriqueciendo la cultura nacional por medio del trabajo profesional de los artistas”.

## 1.4. Organigrama

**Figura 1.**

*Organigrama ESA*



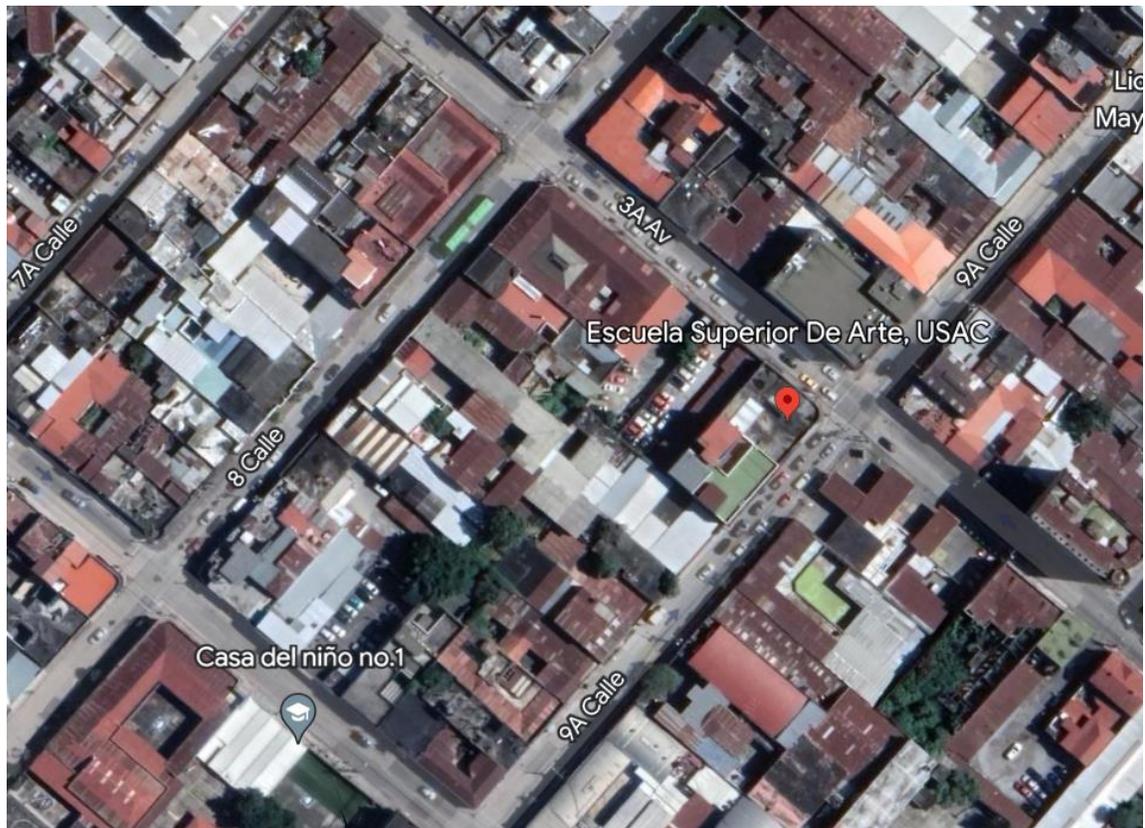
*Nota.* Organigrama general de la Escuela Superior de Arte. Obtenido de S. Morales (2010). *Manual de organización de la escuela de arte.* (<https://ddo.usac.edu.gt/wp-content/uploads/2015/01/MANUAL-DE-ORGANIZACION-DE-LA-ESCUELA-DE-ARTE.pdf>), consultado el 16 de abril de 2023. De dominio público.

## 1.5. Ubicación geográfica

La Escuela Superior de Arte comparte el Edificio del Paraninfo Universitario (2. ° avenida 12-40 zona 1) con el Centro Cultural Universitario.

## Figura 2.

Foto aérea



*Nota.* Foto aérea de las instalaciones de la Escuela Superior de Arte. Obtenido de Google Earth. (2023). *Mapa de la ubicación geográfica de la escuela superior de arte* [Dirección]. (<https://earth.google.com/>), consultado el 16 de abril de 2023. De dominio público.

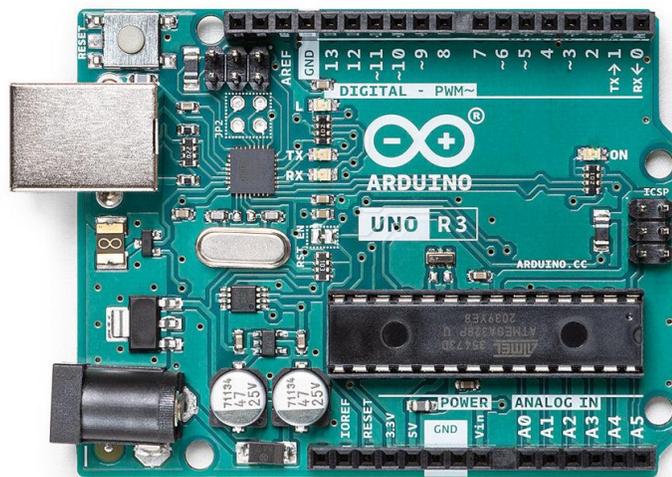
## 2. MARCO TEÓRICO

### 2.1. Arduino

Arduino es una plataforma de desarrollo basada en una placa electrónica de *hardware* libre que incorpora un microcontrolador reprogramable y una serie de pines hembra. Estos permiten establecer conexiones entre el microcontrolador y los diferentes sensores y actuadores, las PCBs superficies planas fabricadas en un material no conductor, la cual consta de distintas capas de material conductor una PCB es la forma más compacta y estable de construir un circuito electrónico, por lo tanto, la placa Arduino no es más que una PCB que implementa un determinado diseño de circuitería interna (*What is Arduino?, s.f.*).

#### Figura 3.

Placa Arduino UNO R3



*Nota.* muestra frontal de una placa Arduino uno. Obtenido de Arduino.cc (2020). *Arduino uno rev3.* (<https://store.arduino.cc/products/arduino-uno-rev3>), consultado el 16 de abril de 2023. De dominio público.

## 2.1.1. Componentes de Arduino

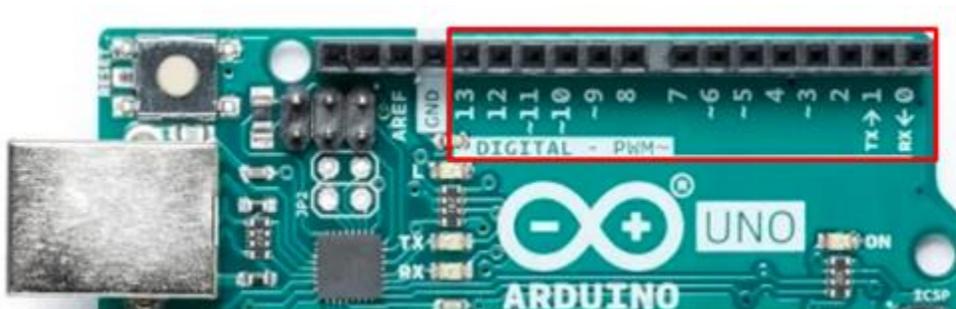
Los componentes de Arduino son mayormente entradas y salidas, existen diversos tipos de salidas y entradas que conforman la placa principal y comunican información al microcontrolador.

### 2.1.1.1. Entradas y salidas

Las entradas pueden ser de tipo digital o analógica, en cuanto a las salidas tendremos salidas únicamente digitales, no existen salidas analógicas, pero tendremos opción a imitar una salida analógica mediante salidas PWM, siendo estas salidas digitales las cuales pueden variar los tiempos de salida de estado alto a bajo (Lukianova, 2021).

#### Figura 4.

*Pines de entrada y salida*



*Nota.* Fotografía de pines de entrada y salidas. Obtenido de Arduino.cc (2020). *Partes de un Arduino.* (<https://arduino.cl/principales-partes-de-un-arduino/>), consultado el 15 de abril de 2023. De dominio público.

### 2.1.1.2. Salidas de alimentación

Posee salidas de alimentación de voltaje las cuales alimentan a la tarjeta y los sensores que se deseen conectar los voltajes de salida son de 5v y 3.3v, para la alimentación posee una entrada USB y un Jack de alimentación de 9v (Lukianova, 2021).

#### Figura 5.

*Salidas de alimentación para Arduino*



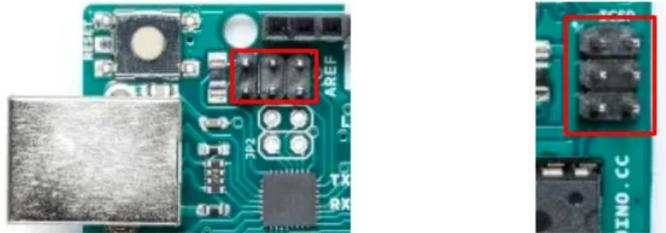
*Nota.* Fotografía de salidas de alimentación. Obtenido de Arduino.cc (2020). *Partes de un Arduino.* (<https://arduino.cl/principales-partes-de-un-arduino/>), consultado el 15 de abril de 2023. De dominio público.

### 2.1.1.3. Salidas de comunicación serial

Cuenta además con 2 conexiones para comunicación ICSP, la cual nos permite programar el *Bootloader* del microcontrolador que utiliza Arduino (Lukianova, 2021).

## Figura 6.

### *Puertos de comunicación*



*Nota.* Fotografía de puertos de comunicación de una placa Arduino uno. Obtenido de Arduino.cc (2020). *Partes de un Arduino.* (<https://arduino.cl/principales-partes-de-un-arduino/>), consultado el 15 de abril de 2023. De dominio público.

#### 2.1.1.4. Control de velocidad de ejecución

Para el control de ejecución de los programas se tiene un dispositivo específico en este caso contamos con un oscilador de 16Mhz el cual tiene la cualidad de modificar la frecuencia de comunicación (Lukianova, 2021).

## Figura 7.

### *Oscilador de frecuencia*



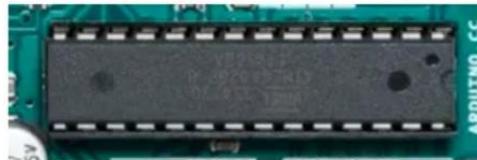
*Nota.* Fotografía de pines de Oscilador de frecuencia. Obtenido de Arduino.cc (2020). *Partes de un Arduino.* (<https://arduino.cl/principales-partes-de-un-arduino/>), consultado el 15 de abril de 2023. De dominio público.

### 2.1.1.5. Microcontrolador

Este es el dispositivo principal que ejecuta nuestro código y nos permite la interacción con las salidas y entradas, el tipo de microcontrolador será diferente según la placa que se esté utilizando, para el caso del Arduino uno R3 se utiliza el microcontrolador ATmega 328.

#### Figura 8.

*Microcontrolador*



*Nota.* Fotografía de chip microcontrolador ATmega 328. Obtenido de Arduino.cc (2020). *Partes de un Arduino.* (<https://arduino.cl/principales-partes-de-un-arduino/>), consultado el 15 de abril de 2023. De dominio público.

### 2.1.2. Tipos de Arduino

Arduino cuenta con una alta diversidad de placas oficiales teniendo un aproximado de 24 placas distintas que se pueden adquirir de forma oficial al ser un proyecto *open source* y *open hardware*, cualquier persona cuenta con acceso a la tecnología para replicarla y modificarla teniendo como resultado placas no oficiales con usos específicos (De Luis, 2021).

Para elegir una placa de Arduino debemos de buscar una que se adapte a nuestras necesidades ya que cada una tendrá disto propósito y precio según aumenten sus prestaciones, siendo los modelos utilizados por principiantes: Arduino UNO, Arduino Leonardo, Arduino 101 y Arduino Explora (De Luis, 2021).

Las diferencias entre un modelo y otro radican en el tamaño, la cantidad de entradas y salidas, la conectividad y tipo de procesador.

#### **2.1.2.1. Arduino Uno**

Esta es la tarjeta más popular y se considera la base de partida para la creación de otras placas, dispone de 14 pines digitales y 5 analógicos y requiere 5 Voltios para funcionar, se puede alimentar con un cable USB o con adaptador de corriente monta el procesador ATMEGA328P con 32Kb de memoria, este modelo lo podemos observar en la figura 3 (De Luis, 2021).

#### **2.1.2.2. Arduino Leonardo**

Es una placa similar al Arduino Uno en su forma física, Integra el microcontrolador ATmega32u4 y cuenta con 20 pines de entrada y salida digitales, un cristal oscilador de 16MHz, una conexión micro USB, jack de alimentación opcional y un botón de reinicio, su característica principal es que puede emular un teclado o ratón al conectarse a un ordenador (De Luis, 2021).

#### **2.1.2.3. Arduino 101**

Este modelo cuenta con una placa con base en el Arduino uno, con la diferencia de contar con el procesador Intel Curie, diseñado para proporcionar alto rendimiento con un bajo consumo de energía, además incorpora el estándar Bluetooth de baja energía (BLE), acelerómetro de seis ejes y un giroscopio (De Luis, 2021).

#### **2.1.2.4. Arduino Zero**

Este modelo se considera la SAMD21 de Atmel, con núcleo ARM Cortex, lo que permite mayor procesamiento. También permite programar otros dispositivos al admitir un puerto COM. del Arduino uno, posee un mayor rendimiento gracias al microprocesador (De Luis, 2021).

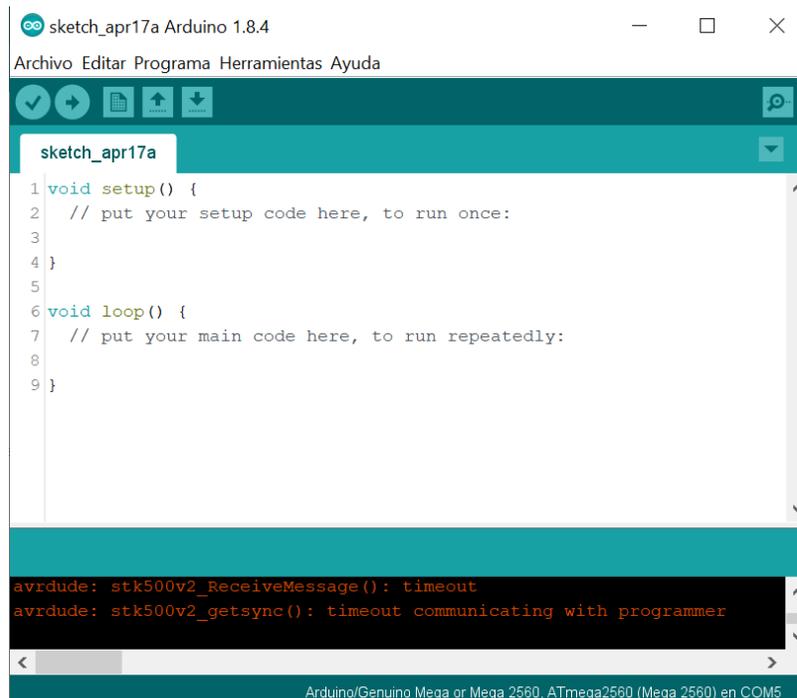
#### **2.1.2.5. Arduino Mega 2560**

Este modelo se caracteriza por su aumento de pines de entrada y salida integra el microprocesador ATmega2560, una memoria flash de 256 KB, 8 KB de RAM y 70 pines, entre analógicos y digitales, cuenta con 4 puertos UART y su velocidad de procesamiento es de 16MHz. A nivel de espacio, es capaz de almacenar programas cuatro veces más grandes que el Arduino uno (De Luis, 2021).

#### **2.1.3. Arduino IDE**

Un IDE es un conjunto de herramientas de *software* necesarias para desarrollar y grabar el código necesario para que nuestro Arduino funcione, El IDE de Arduino nos permite escribir, depurar, editar y grabar nuestro programa (Aguayo, 2022).

**Figura 9.**  
*Software Arduino*



*Nota.* Captura de pantalla del *software* Arduino IDE. Elaboración propia, realizado con Arduino IDE 1.8.4.

El IDE de Arduino cuenta con un menú de 5 opciones, la primera opción es archivo, en este menú podremos acceder a las opciones para guardar, abrir y editar los programas hechos en esta IDE. El menú de editar nos permite modificar el código que estemos escribiendo en el IDE, tiene opciones como rehacer, deshacer, copiar, cortar, etc. El menú de Programa tiene las opciones de verificar y compilar nuestro archivo e importa librerías externar. En el menú Herramientas encontramos opciones para auto-formatear el código, también la opción de abrir el monitor serial entre otras. El menú Ayuda accede a artículos e información de la página web oficial de Arduino para mostrar tutoriales y ejemplos de ayuda. El

menú de Monitor Serial nos permite abrir una ventana del IDE para enviar y recibir datos textuales de la placa Arduino usando el cable USB.

## **2.2. Sensores**

Los sensores son elementos que se encargan de interactuar con una condición física eléctrica para obtener información y son capaces de convertirla en una magnitud eléctrica, la cual es procesada por una unidad de control para dar una orden de salida la cual es convertida en una señal eléctrica la cual se envía a un acondicionador o actuador este último dispositivo se encarga de transformar la energía eléctrica en otro tipo de energía, los sensores son conocidos también como transductores (Guarella, Heredia, Rodríguez & Bagatto, 2011).

### **2.2.1. Tipos de sensores**

Los sensores tienen una respuesta en dependencia de la magnitud física detectada y el principio físico en el que se basan por esta razón Guarella, Heredia, Rodriguez y Bagatto (2011) clasifican los sensores según el principio de funcionamiento teniendo como resultado la siguiente clasificación:

- Magnético
- Efecto Hall
- Conductividad eléctrica
- Termoeléctricos
- Fotoeléctricos
- Piezoeléctricos
- Por ultrasonido

### **2.2.2. Sensores por magnetismo**

Según Guarella et. al (2011) este tipo de sensores basan su funcionamiento en el fenómeno electromagnético, esto sucede al tener una variación del flujo magnético obtendremos una corriente por inducción magnética. Este tipo de sensores usa una bobina para detectar el cambio del flujo magnético.

### **2.2.3. Sensor por efecto Hall**

Estos sensores se basan en el efecto hall el cual se produce en semiconductor al ser recorrido por una corriente y estar sometido por un campo magnético, generando en sus extremos una diferencia de potencial, la tensión generada en los extremos del sensor es proporcional a la intensidad del flujo magnético (Guarella et. al, 2011).

### **2.2.4. Sensor por conductividad eléctrica**

Este sensor funciona con mediante el cambio de conductividad en un material por donde circula corriente eléctrica, la conductividad dependerá del número de electrones libres en el material y la temperatura a la cual se encuentre, como ejemplo de este tipo de sensores tenemos a la sonda Lambda, siendo este un sensor de oxígeno para gases de escape en los sistemas de inyección de combustible (Guarella et. al, 2011).

### **2.2.5. Sensores termoeléctricos**

Este sensor funciona mediante el cambio de resistencia que sufren los metales al entrar en contacto con el calor, debido a la dilatación que sufren estos objetos se pueden crear configuraciones que aumenten o disminuyan su resistencia al entrar en contacto con una alta temperatura, de esta manera es que se da origen a los termistores PTC o NTC (Guarella et. al, 2011).

### **2.2.6. Sensores fotoeléctricos**

Este tipo de sensores pueden generar energía eléctrica a partir de la radiación luminosa, tanto la visible como la infrarroja y ultravioleta, también existen los sensores que varían su resistencia eléctrica, cuando se produce un aumento de radiación luminosa como lo son las fotorresistencias (Guarella et. al, 2011).

### **2.2.7. Sensores piezoeléctricos**

Este sensor se basa en el efecto piezoeléctrico el cual consiste en la aparición de una diferencia de potencial en un material cuando se deforma por una fuerza, regularmente estos sensores están hechos por cristales naturales como el cuarzo (Guarella et. al, 2011).

### **2.2.8. Sensores por ultrasonido**

Estos sensores funcionan detectando sonidos con frecuencias mayores superiores al límite audible por el oído humano, estas ondas se propagan por aire y cambian su frecuencia al rebotar en un objeto (Guarella et. al, 2011).

### **2.2.9. Sensores por radiofrecuencia**

Este sensor detecta ondas radioeléctricas emitidas en el espacio y recibidas por un receptor. Las ondas de radiofrecuencia son generadas por corriente alterna a alta frecuencia (Guarella et. al, 2011).

## **2.3. Actuadores**

Son los dispositivos que se encargan de transformar las señales recogidas por el sensor y transformarlas en un nuevo tipo de energía, puede ser en energía eléctrica, mecánica, hidráulica o térmica, este dispositivo pasa de una señal eléctrica de baja potencia a una señal más potente (Guarella et. al, 2011).

### **2.3.1. Electromagnéticos**

Este tipo de actuador funciona mediante el magnetismo, pudiendo ser de origen natural o artificial, creado por la electricidad, en esta clasificación también entran otros fenómenos provocados por el electromagnetismo como lo son la inducción electromagnética (Guarella et. al, 2011).

### **2.3.2. Electromotores**

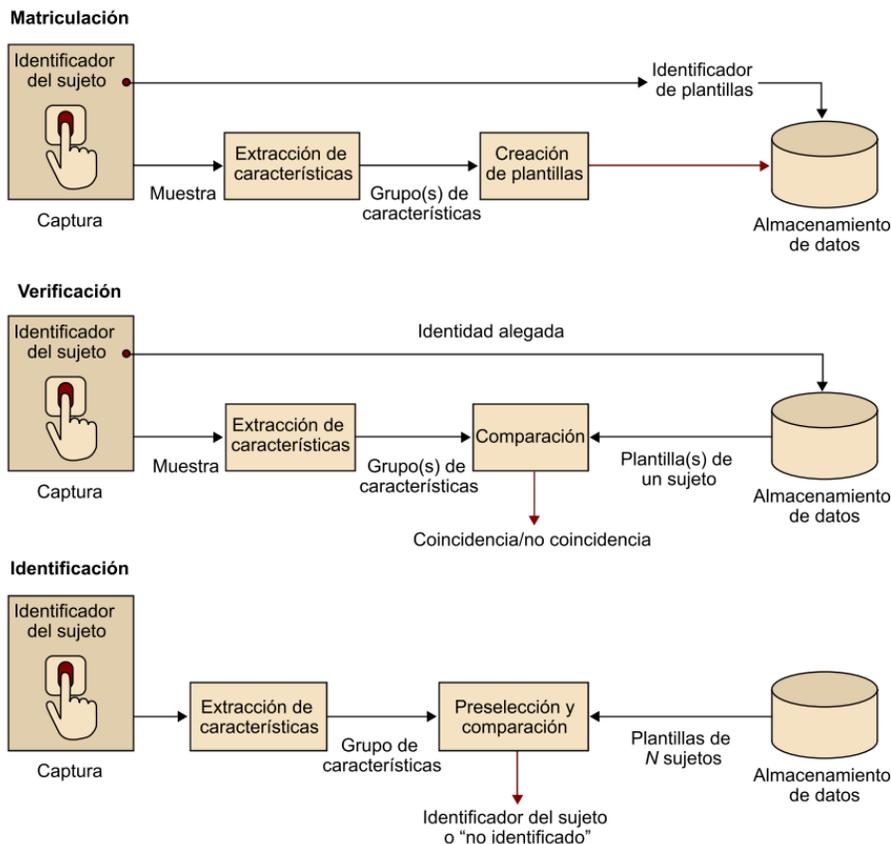
Este tipo de actuador basa su funcionamiento en la transformación de energía eléctrica en energía mecánica rotacional de manera que se genere movimiento por la interacción de dos campos magnéticos, siendo uno generado por la circulación de corriente a través de un conductor y el segundo campo magnético debe ser un campo magnético fijo (Guarella et. al, 2011).

## **2.4. Sistema biométrico**

La biometría, según Guarella et. al (2011) la define como una ciencia que se dedica al analizar las distancias y posiciones entre las partes del cuerpo con la finalidad de identificar a una persona del resto de la población, existen varios rasgos biométricos como lo son las huellas dactilares, el iris, la retina, etc. Estos rasgos ya han sido estudiados desde el siglo XIX en aplicaciones forenses cuando se analizaban las huellas dactilares para identificar a los individuos.

Los rasgos biométricos son una combinación de anatomía y de comportamiento humano, teniendo como resultado un parecido de esta característica entre parientes próximos estos rasgos son esenciales para la identificación de las personas al ser características que no se pueden compartir o extraviar dando como resultado sistemas de identificación más cómodos, seguros. Los sistemas biométricos constan de los siguientes procesos (Guarella et. al, 2011).

**Figura 10.**  
*Sistema biométrico*



Nota. Diagrama de procesos de un sistema biométrico. Adaptado de F. Serratos (2008). *La biometría para la identificación de las personas*. ([https://sistemamid.com.ar/panel/uploads/biblioteca/2015-03-22\\_12-05-01117594.pdf](https://sistemamid.com.ar/panel/uploads/biblioteca/2015-03-22_12-05-01117594.pdf)), consultado el 17 de abril de 2023.

### 2.4.1. Captura de información

Este sistema captura la información con una representación digital del rasgo biométrico, la información capturada se suele denominar muestra y también puede incorporar otros periféricos para introducir información no biométrica (Serratos, 2008).

#### **2.4.2. Extracción de las características**

Este paso se realiza mediante un dispositivo que extraiga las características a comparar, genera una representación más compacta con el objetivo de facilitar la comparación de información (Serratosa, 2008).

#### **2.4.3. Comparación**

En este proceso se recibe como entrada un registro donde se compara con una plantilla y se calcula la distancia entre los dos, en el proceso de verificación existe un umbral interno que solo el administrado puede configurar, el sistema considera si la distancia supera el umbral y determina si los datos provienen de la misma persona (Serratosa, 2008).

#### **2.4.4. Selección o filtrado**

Es un método para aumentar la eficiencia de la comparación reduciendo el tiempo de respuesta del sistema ya que se tendrán registros de cientos o miles de personas es necesario tener la respuesta más rápida posible (Serratosa, 2008).

#### **2.4.5. Almacenamiento de los datos**

En este proceso se almacenan los datos de identificación del usuario como el DPI, nombre, edad, etc. Y la plantilla biométrica, esta información se almacena en sistemas centralizados o en tarjetas inteligentes (Serratosa, 2008).

## **2.5. Tipos de detectores biométricos**

Según Sensores biométricos (s.f.) un detector biométrico es un dispositivo capaz de transformar una característica biológica humana en una señal eléctrica, estos detectores suelen ser sensores hechos por materiales semiconductores gobernados por microprocesadores o microcontroladores que en conjunto tienen la capacidad de procesar las características físicas.

Siendo los rasgos biométricos muy variados existe una gran diversidad de sensores biométricos entre estos podemos encontrar los sensores de huellas dactilares, reconocimiento facial, reconocimiento de iris, reconocimiento de voz, entre otros, tomando en cuenta el desarrollo de la tecnología no se descarta el hecho de tener en futuro detectores de ADN, el lóbulo de la oreja, entre otros, (Sensores biométricos, s.f.).

### **2.5.1. Sensor de huella digital**

Este dispositivo electrónico nos permite identificar las huellas digitales siguiendo un proceso de lectura, guardado e identificación de la huella dactilar utilizando un escáner el cual se encarga de analizar las características fisiológicas de la huella, el sensor guarda las formas y distancias existentes entre las crestas y valles que conforman la huella digital (Lector de huellas digital, 2021).

Los lectores de huella digital se pueden clasificar en ópticos, escáner de capacitancia, escáner ultrasónico, escáner térmico, cada uno funciona con distinto principio físico y cada uno tiene sus ventajas y desventajas (Lector de huellas digital, 2021).

#### **2.5.1.1. Escáner óptico**

Este tipo de escáner toma una imagen visual de la huella digital utilizando mediante una cámara digital y son los pixeles de esta imagen la que se registra y compara (Lector de huellas digital, 2021).

#### **2.5.1.2. Escáner capacitivo**

Estos sensores utilizan capacitores por lo que generan una imagen de la huella mediante la corriente eléctrica que atraviesa las crestas y valles, suelen ser más compactos que los dispositivos ópticos y tienen gran precisión (Lector de huellas digital, 2021).

#### **2.5.1.3. Escáner por ultrasonido**

Utiliza ondas sonoras de alta frecuencia que penetran la piel del dedo y son reflejadas hacia el sensor nuevamente y de esta manera se forma una imagen digital de la huella digital (Lector de huellas digital, 2021).

#### **2.5.1.4. Escáner térmico**

Este sensor mide la diferencia de temperatura entre las crestas y valles de las huellas dactilares dando como resultado un mapa térmico que dibuja de forma precisa la forma de la huella dactilar (Lector de huellas digital, 2021).

#### **2.5.1.5. Lenguajes de programación**

Un lenguaje de programación es formal y nos permite escribir una serie de ordenes ya acciones consecutivas de datos, para crear programas que le den

instrucciones físicas y lógicas a una máquina, este lenguaje está formado por un conjunto de símbolos, palabras claves, reglas semánticas y sintácticas que nos permiten la comunicación entre el programador y una máquina (Pulido, 2015).

## 2.6. Python

Python es un lenguaje de programación que inicio a principios de los 90s, siendo sus bases la programación de ABC, el cual era un programa que se desarrolló en el CWI, de este lenguaje de programación surge Python en su versión 0.9.0 en 1991, teniendo características como las clases con herencia, manejo excepción, funciones y los tipos modulares (Rojas, s.f.).

En este mismo año se crea la versión 1.0 en la cual se incluyen herramientas de programación funcionales como *lambda*, *reduce*, *filter* y *map*. A partir de esta primera versión se desarrolla este programa a partir de su gran popularidad uniéndose a equipos como *BeOpen Python Labs*, siendo un *software* de licencia libre al ser apoyado por la *Free Software Foundation*, la Última versión de este *Software* es la 3.0 en la cual rompe la compatibilidad del lenguaje con las versiones anteriores con el objetivo de tener un solo modo obvio de hacer las cosas, evitando las formas repetidas de programar una misma tarea (Rojas, s.f.).

### **3. MARCO METODOLÓGICO**

#### **3.1. Diagnóstico de la situación actual**

El área de administración del para el personal de la escuela superior de artes de la universidad de San Carlos de Guatemala actualmente no cuenta con un sistema para el registro de asistencia, por lo que recurren al uso de planillas impresas donde cada persona registra su nombre y hora de entrada y salida, teniendo problemas para confirmar la veracidad de esta información. A esto agregamos el problema de tener que digitar todos los datos de entrada y salida del personal el cual se puede acumular durante una semana o una quincena, esto agrega tiempo de trabajo y se vuelve ineficiente con el tiempo, otro inconveniente encontrado es que se debe contar con un responsable que atienda la entrada, al tener una puerta con chapa eléctrica para el ingreso.

#### **3.2. Diseño teórico del sistema biométrico para el control de asistencia**

Para el funcionamiento de un sistema de control de asistencia gobernado por un sensor biométrico, tenemos que crear una serie de etapas que garanticen el funcionamiento y la correcta interacción con las personas que utilicen este sistema, por este motivo se dividirá en 4 etapas su funcionamiento, la primera constara de la recopilación de la información biométrica de los usuarios, la segunda etapa será la de almacenamiento de la información en una base de datos ya que se requiere poder volver a utilizar los datos registrados, como tercer etapa tendremos la interfaz gráfica la cual servirá para que un administrador del sistema pueda registrar a los usuarios y obtener los resultados del control de asistencia, como cuarta y última etapa tendremos el control de ingreso del

personal etapa donde se registrara la hora de entrada de las personas y se permitiría el ingreso a las instalaciones.

Para la primer etapa se tienen una gran variedad de opciones de sensores para recopilar la información biométrica de una persona, cada sensor tendrá sus ventajas y desventajas dependiendo de aplicación que se requiera, En este caso optamos por un sensor de huellas dactilares específicamente el sensor óptico “*Adafruit Optical Fingerprint Sensor*”, este sensor se escogió debido a la compatibilidad con Arduino y su bajo costo, el sensor debe tener compatibilidad con Arduino dado que buscamos aprovechar los puertos seriales que posee este microcontrolador para la comunicación entre el sensor óptico y la aplicación que contenga la interfaz gráfica, por esta razón se decide utilizar el Arduino Mega 2560 ya que requerimos utilizar 2 puertos seriales.

La segunda etapa requiere una base de datos que se comunique con nuestra interfaz gráfica y que al mismo tiempo pueda comunicarse con el puerto serial de Arduino, tomando en cuenta estos requisitos optamos primero por utilizar Python como lenguaje principal dado que tiene la versatilidad y herramientas suficientes para desarrollar la interfaz gráfica y la lógica de conexión entre los datos recibidos por Arduino y la base de datos. Como base de datos utilizaremos SQLite dato su facilidad y capacidad suficiente para almacenar la información del personal administrativo de la Escuela Superior de Arte de la Universidad de San Carlos de Guatemala.

La etapa de interfaz gráfica contara con las opciones necesarias para registrar los datos del personal administrativo, así como visualizar los datos de las personas previamente registradas, también contara con una ventana que muestre que persona ha ingresado y el registro de personas que ingresaron en el día.

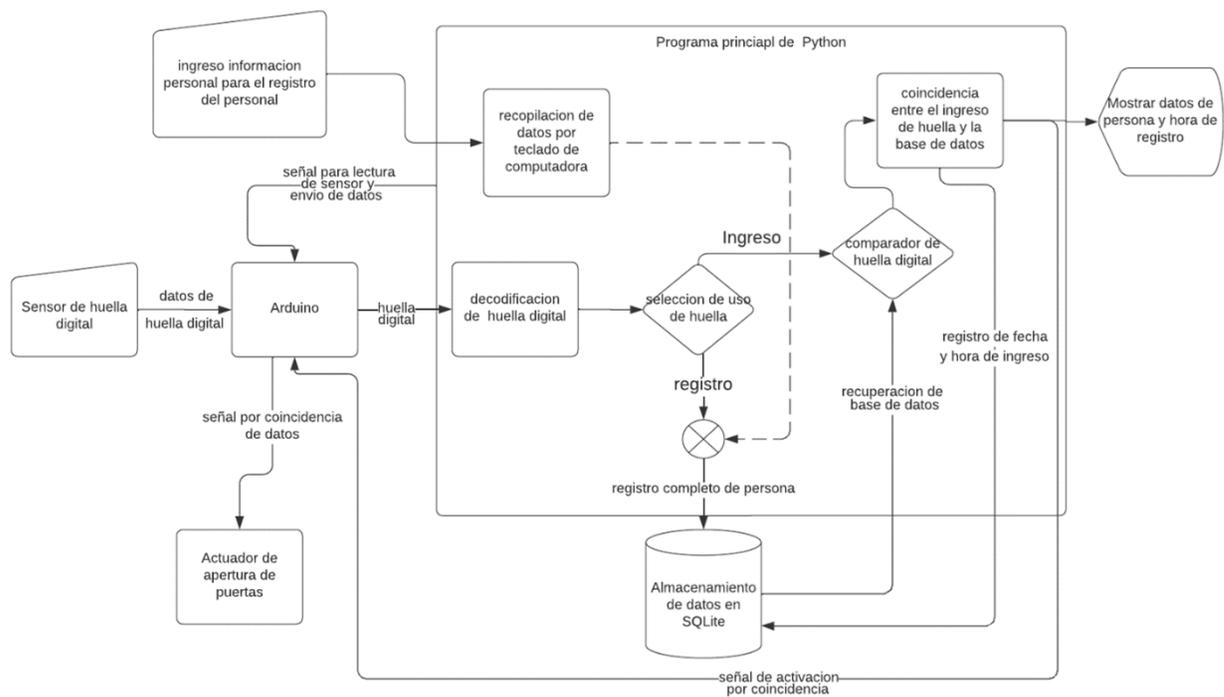
En la última etapa tendremos para el control de ingreso se contará con una parte electrónica la cual activará una cerradura eléctrica yale para permitir el ingreso de las personas a las instalaciones, para activar esta cerradura se tendrá una comparación entre los registros biométricos guardados en la base de datos y la información obtenida por el sensor al intentar ingresar la persona, de esta manera el programa determinar qué persona está ingresando y guarda la hora de ingreso.

### **3.3. Diagrama de Bloques**

En la figura 11. Se muestra cual es proceso lógico que sigue nuestra aplicación y como interactúa con Arduino para la obtención de datos de la huella digital y como acciona la puerta automática al verificar que el usuario existe en los registros de la base de datos.

**Figura 11.**

*Diagrama de bloques*



*Nota.* Diagrama de bloques para la obtención y comparación de huella digital. Elaboración propia, realizado con Lucidchart.

### 3.4. Dispositivos utilizados

El diseño propuesto consideró los siguientes dispositivos electrónicos para su funcionamiento:

- Arduino Mega 2560.
- Cable USB tipo A macho a tipo B macho.
- Fuente de alimentación de 12v.
- Resistencia de 1kΩ.
- Transistor 2n2222

- Diodo 1N4001
- Relé 2 polos 1 tiro de 9v.
- Sensor óptico “Adafruit Optical Fingerprint Sensor”.
- Cerradura eléctrica Yale 849396.

El diseño de este proyecto utilizo los siguientes programas para simular el funcionamiento completo al implementar el diseño.

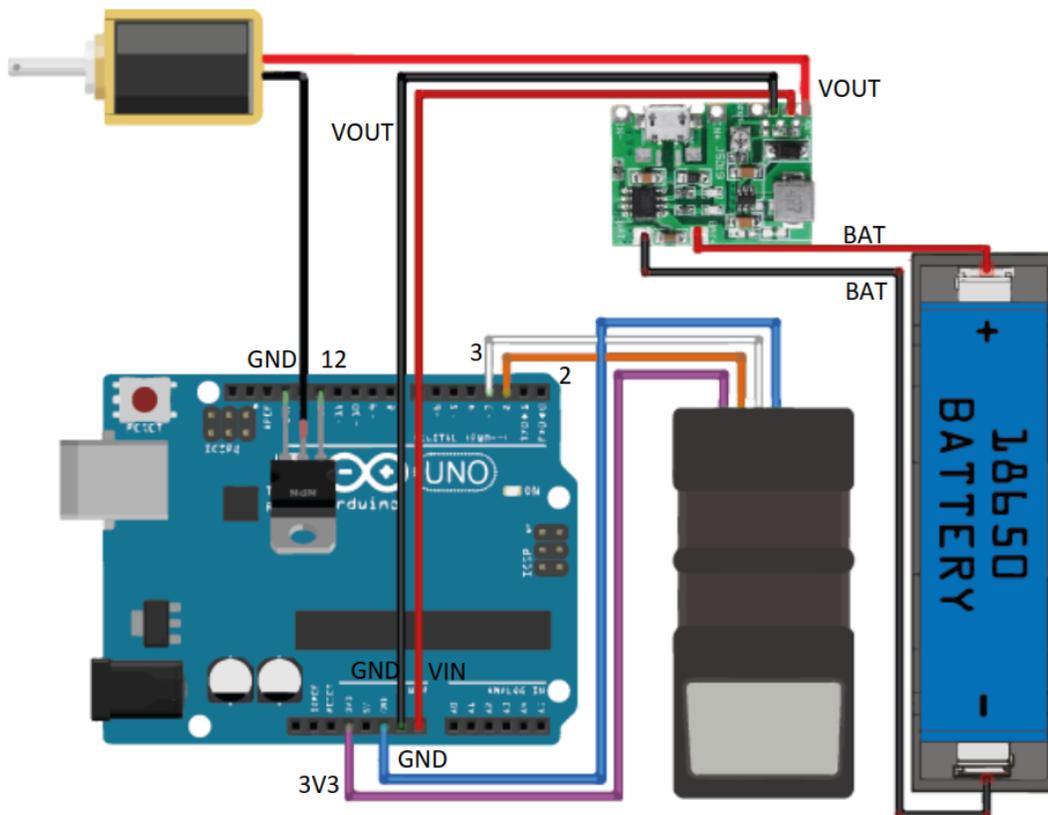
- PROTEUS 8 PROFESSIONAL
- DB Browser for SQLite
- Visual Studio Code
- Virtual Serial Port Driver 6.9
- Arduino IDE

### 3.5. Diseño teórico de conexión de equipo

Para el diseño teórico de la conexión del sensor de huellas digitales nos basamos en la información proporcionada por la página AUTOBOTIC en la cual muestra la conexión del sensor hacia una placa Arduino uno.

**Figura 12.**

*Conexión de sensor óptico con Arduino*



*Nota.* Esquema de conexión para sensor óptico con Arduino . Adaptado de Ada, L. (2022). *Optical Fingerprint Module Sensor (AS608) For Arduino* [Sensor óptico del módulo de huellas dactilares (AS608) para Arduino] <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf> consultado el 22 de agosto de 2022.

## 4. DISEÑO PROPUESTO

Para el diseño propuesto tomamos en cuenta 2 factores fundamentales los costos de implementación y la versatilidad de uso tanto para los usuarios como para el administrador del sistema siendo esta la persona con más contacto con el sistema.

De esta manera planteamos un sistema sencillo de utilizar por el personal que debe tomar su asistencia y amigable para las personas que deben utilizar y administrar los datos de asistencia. Por esta razón dividimos el diseño en 4 etapas donde nos enfocamos en un tema específico del sistema biométrico que analiza desde la obtención de datos hasta el almacenamiento y la interacción maquina usuario que se tendrá en su uso diario.

### 4.1. Código de interfaz gráfica

La interfaz gráfica se desarrolló en el lenguaje de programación Python dada la versatilidad que tiene este lenguaje para interactuar con bases de datos y comunicarse con microcontroladores externos mediante una comunicación con el puerto serial. Para la interfaz gráfica utilizamos la librería Tkinter la cual nos permite crear ventanas para interactuar con la información obtenida por nuestro sensor biométrico. El código se desarrolló mediante POO.

El código principal de la interfaz gráfica es el siguiente:

```
if __name__ == "__main__":  
    global is_secondary_open
```

```

is_secondary_open = False
ventana = Tk()
ventana.title('ingreso de personal')
ventana.minsize(height = 400, width=600)
ventana.geometry('800x500')
ventana.call('wm', 'iconphoto', ventana._w,
PhotoImage(file='Usac_logo.png'))
bg = PhotoImage(file = 'logo-usac-2.png')
fondo = Label(ventana, image = bg)
fondo.place(x = 0, y = 0, relheight=1, relwidth=1)
#ventana.call('wm', 'iconphoto', ventana._w,
PhotoImage(file='descarga.png'))
app = Ventana(ventana)
#se crea el menu
menubar = Menu(ventana)
#se crean las opciones dentro de menu
file_menu = Menu(menubar, tearoff=0)
file_menu.add_command(label="registro", command=
lambda:nueva_ventana(app))
file_menu.add_separator()
file_menu.add_command(label="Salir", command=ventana.quit)
#se crea la pestaña menu
menubar.add_cascade(label="Archivo", menu=file_menu)
ventana.config(menu=menubar)
app.mainloop()

##### PRINCIPAL VENTANA
#####

class Ventana(Frame):
    def __init__(self, master):
        super().__init__(master)
        self.nombre = StringVar()
        self.apellido = StringVar()
        self.dependencia = StringVar()
        self.fecha = StringVar()

```

```

self.hora = StringVar()
self.fondo = StringVar()
self.deteccion = StringVar(value='presione ingresar la
huella')

self.master.columnconfigure(0, weight = 1)
self.master.rowconfigure(0, weight = 1)
self.master.rowconfigure(1, weight = 5)
self.stop_hilo = False
self.base_datos = comunicacion()
self.fondo = '#95A5A6'
self.bg = PhotoImage(file = 'logo-usac-2.png')
self.widget()

def fin(self):
    global is_secondary_open
    is_secondary_open = False
    self.stop_hilo = True
    self.destroy()

def inicia_hilo(self):
    self.deteccion.set("ingrese su huella")
#Crea un nuevo hilo y asigna el método "leer_puerto_serial"
como función target
    hilo = threading.Thread(target=self.leer_puerto_serial)
    # Inicia el hilo
    hilo.start()

def leer_puerto_serial(self):
    # Abre el puerto serial
    arduino = serial.Serial('COM1', 9600)
    time.sleep(2)
    arduino.write(b'2')
    stop = 1
    texto = ""
    union = ""

```

```

boton = ""
comparador = ""
while not self.stop_hilo and not boton == "no":
    # Lee los datos del puerto serial y los muestra en la
etiqueta

    datos = arduino.readline()
    d_comp = datos.decode().rstrip()
    if d_comp == "\x16":
        pass
    elif d_comp == "0":
        boton = "no"
    elif d_comp == "guardar":
        boton = "guardar"
    else:
        # Añade la línea al texto multilínea
        texto = re.sub(r"\s+", "", datos.decode())
        if boton == "comparar":
            comparador += texto
        elif boton == "guardar":
            union += texto

    print(union)
    self.deteccion.set("Huella Detectada")
    self.comparar_huella(union, arduino)
    arduino.close()

def widget(self):
#frame 1
    # self.tiempo_ultima_accion = time.time()
    self.frame_uno = Frame(self.master, bg = self.fondo,
height = 200, width = 800)
    self.frame_uno.grid(column=0, row=0, sticky='nsew')
#frame 2 muestra si no se reconoce el usuario
    self.frame_dos = Frame(self.master, bg = self.fondo,
height = 300, width = 800,)
    self.frame_dos.grid(column = 0, row = 1, sticky = 'nsew')
#configuracion de los frame creacion de columnas internas

```

```

self.frame_uno.columnconfigure([0,1,2], weight = 1)
self.frame_uno.rowconfigure([0,1,2,3,4,5], weight = 1)
self.frame_dos.columnconfigure(0, weight = 1)
self.frame_dos.rowconfigure(0, weight = 1)
    #etiquetas en primera columna indicadores de los
datos
    Label(self.frame_uno, text = 'Asistencia',fg='black', bg=
self.fondo,
        font=('Kaufmann BT', 13, 'bold')).grid(columnspan=2,
column=0, row=0, pady=5)
    Label(self.frame_uno, text = 'Nombre', fg='black', bg =
self.fondo,
        font=('Rockwell',13, 'bold')).grid(column=0, row=1,
pady=5)
    Label(self.frame_uno, text = 'apellido', fg='black', bg
= self.fondo,
        font=('Rockwell', 13, 'bold')).grid(column=0, row=2,
pady=5)
    Label(self.frame_uno, text = 'dependencia', fg='black',bg
= self.fondo,
        font=('Rockwell', 13, 'bold')).grid(column=0, row=3,
pady=5)
    Label(self.frame_uno, text = 'fecha', fg='black', bg
=self.fondo,
        font=('Rockwell', 13, 'bold')).grid(column=0,row=4,
pady=5)
    Label(self.frame_uno, text = 'hora', fg='black', bg
=self.fondo,
        font=('Rockwell', 13, 'bold')).grid(column=0,row=5,
pady=5)
    #label variable cambia segun datos
    Label(self.frame_uno, textvariable =self.nombre ,
fg='black', bg ='white',
        font=('Rockwell',13,
'bold'),width=20,height=1).grid(column=1, row=1, pady=5)

```

```

        Label(self.frame_uno, textvariable =self.apellido ,
fg='black', bg ='white',
        font=('Rockwell', 13,
'bold'),width=20,height=1).grid(column=1, row=2, pady=5)
        Label(self.frame_uno, textvariable =self.dependencia,
fg='black',bg ='white',
        font=('Rockwell', 13,
'bold'),width=20,height=1).grid(column=1, row=3, pady=5)
        Label(self.frame_uno, textvariable =self.fecha,
fg='black', bg ='white',
        font=('Rockwell', 13,
'bold'),width=20,height=1).grid(column=1,row=4, pady=5)
        Label(self.frame_uno, textvariable =self.hora,
fg='black', bg ='white',
        font=('Rockwell', 13,
'bold'),width=20,height=1).grid(column=1,row=5, pady=5)
        #crear botones
        #Label(self.frame_uno, image = self.bg).grid(column=2)
        Button(self.frame_uno, text= 'exportar',font = ('Arial',
9, 'bold'), bg= 'deep sky blue',
        width=20, bd=3, command =
self.guardar_datos).grid(column=2, row=5, pady=5, padx=5)
        Button(self.frame_uno, text= 'registro',font = ('Arial',
9, 'bold'), bg= 'deep sky blue',
        width=20, bd=3, command =
self.ventana_tabla).grid(column=2, row=2, pady=5, padx=5)
        Button(self.frame_uno, text= 'Ingresar huella',font =
('Arial', 9, 'bold'), bg= 'deep sky blue',
        width=20, bd=3, command =
self.inica_hilo).grid(column=2, row=3, pady=5, padx=5)
        #label del frame2 indicador de correcta deteccion de la huella
        Label(self.frame_dos, textvariable=self.deteccion,
fg='black', bg = self.fondo,
        font=('Rockwell', 20, 'bold')).grid(column=0, row=0)

```

## 4.2. Código de Arduino

El Arduino programamos la conexión del sensor óptico hacia Arduino y la conexión hacia la aplicación hecha en Python para esto utilizamos 2 puertos seriales dado que por el tipo de sensor utilizado se simulo como una entrada en el puerto serial 1 y la conexión hacia Python se realiza en el segundo puerto seria además de configurarse las salidas digitales destinadas para el control de la puerta que se debe activar al momento de ingresar el personal registrado en la base de datos.

A continuación, se presenta el código creado para el funcionamiento de la placa Arduino

```
String huella;
String comparador;
char llave;
const int registro = 4;
const int activar = 5;
const int cerrar = 6;
bool ingreso = false;
bool ms1 = true;
bool abrir = false;
int contador = 10;
void setup() {
    Serial.begin(9600);
    Serial1.begin(9600);
    //Serial.setTimeout(tiempo en ms); cambia el tiempo de espera para
    leer la cadena
    pinMode(registro, INPUT);
    pinMode(activar, OUTPUT);
    pinMode(cerrar, INPUT);
```

```

}

void loop() {
  delay(10);
  contador++;
  if (contador >1000) {
    contador = 0;
  }

  if(abrir == true && contador >250){
    abrir = false;
    digitalWrite(activar, LOW);
  }

  if (Serial1.available()>0) {
    llave = Serial1.read();
  }

  if (llave == '1' ){
    abrir = true;
    contador =0 ;
    llave = '0';
    digitalWrite(activar,HIGH);
  }
  //ingreso indica si ya se detecto un caracter en el monitor
  while (llave == '2' || ingreso == true) {

    llave = '0';
    //se envia el mensaje una sola vez
    if(ms1 == true ){
      Serial.println("ingrese la huella");
      Serial1.println("guardar");
      ms1 = false;
      ingreso = true;
    }
  }
}

```

```

//se hace el registro de la huella
    if (Serial.available()) { // pregunta si hay un caracter
disponible para almacenar, esto sucede inmediatamente se ingresa un
valor al monitor
        huella = Serial.readString();
        ingreso = false;
        ms1 = true;
        Serial.println("huella registrada :)");
        Serial1.println(huella);
        Serial1.println("0");
    }
}
}

```

### 4.3. Código para la base datos

Para el código de la base de datos utilizamos SQLite siendo una base de datos que nos permite almacenar los datos de los empleados de forma estructurada y eficiente, lo que facilita su acceso y gestión. Además, SQLite es una base de datos muy fácil de integrar en una aplicación, lo que la hace una excelente opción para cualquier programa que requiera almacenar y manipular grandes cantidades de información.

La creación de una base de datos en SQLite puede ser un proceso técnico, que requiere conocimientos especializados en programación y bases de datos. Sin embargo, una vez creada la base de datos, su uso es muy sencillo e intuitivo. Además, el uso de una base de datos en un programa de registro de personal puede ofrecer numerosos beneficios, como la posibilidad de hacer búsquedas rápidas, la generación de informes detallados y la automatización de tareas repetitivas.

Para la creación de la base de datos utilizamos el programa DB Browser for SQLite donde definimos y creamos las tablas y registros a utilizar por nuestro programa principal. Se crearon las siguientes tablas, la tabla “Registro” donde se almacenará la información de los empleados, este registro será el principal ya que aquí se almacena la información codificada de la huella digital. La siguiente tabla será “horario” en esta tabla se almacena la información de la hora en la al cual se ingresa o egresa los empleados del departamento.

Para la interacción con la base de datos se crea un código especial donde se tienen las funciones necesarias para la interactuar con el programa principal, el código es el siguiente creado en Python.

```
import sqlite3

class comunicacion():
    def __init__(self):
        self.conexion = sqlite3.connect('base1.db')
        #conexion a la base de datos

        #####INGRESA LOS DATOS DE HORA Y PERSONA INGRESADA A LA BASE DE DATOS
        def registra_hora(self, nombre, apellido, dependencia, fecha,
hora):
            cursor = self.conexion.cursor()
            # nombre_tabla
            bd = '''INSERT INTO horario
(NOMBRE, APELLIDO, DEPENDENCIA, FECHA, HORA)
VALUES ('{}', '{}', '{}', '{}', '{}')'''.format(nombre, apellido, dependencia, fe
cha, hora)
            cursor.execute (bd)
            self.conexion.commit()
            cursor.close()
```

```

def mostrar_registro(self):
    cursor = self.conexion.cursor()
    bd = "SELECT * FROM horario"
    cursor.execute(bd)
    Registro = cursor.fetchall()
    cursor.close()    ###se agrego no se porque
    return Registro

def eliminar_horario(self):
    curosor = self.conexion.cursor()
    #                nomb tabla y nombre columnas
    bd = ''' DELETE FROM horario '''
    curosor.execute(bd)
    self.conexion.commit()
    curosor.close()

def inserta_datos(self, nombre, apellido, dependencia, huella):
    cursor = self.conexion.cursor()
    #                nombre_tabla
    bd =          =          '''INSERT          INTO          Registro
(NOMBRE, APELLIDO, DEPENDENCIA, HUELLA)
VALUES ('{}', '{}', '{}', '{})'''.format(nombre, apellido, dependencia, huella)
    cursor.execute(bd)
    self.conexion.commit()
    cursor.close()

#mostrar los datos, retorna los datos en una variable llamda datos
def mostrar_datos(self):
    cursor = self.conexion.cursor()
    bd = "SELECT * FROM Registro"
    cursor.execute(bd)
    Registro = cursor.fetchall()

```

```

        cursor.close()      ###se agrego no se porque
        return Registro

    def eliminar_datos(self, nombre):
        curosor = self.conexion.cursor()
        # nomb tabla y nombre columnas
        bd = ''' DELETE FROM Registro WHERE NOMBRE = '{}'
''' .format(nombre)
        curosor.execute(bd)
        self.conexion.commit()
        curosor.close()

#no se requiere mostrar la huella al actualizar los datos
    def actualiza_datos(self, Id, nombre, apellido, dependencia):
        cursor = self.conexion.cursor()
        bd = '''UPDATE Registro SET NOMBRE = '{}', APELLIDO = '{}',
DEPENDENCIA = '{}' WHERE Id = '{}' ''' .format(nombre, apellido, dependencia,
Id)

        cursor.execute(bd)
        Registro = cursor.rowcount
        self.conexion.commit()
        cursor.close()
        return Registro

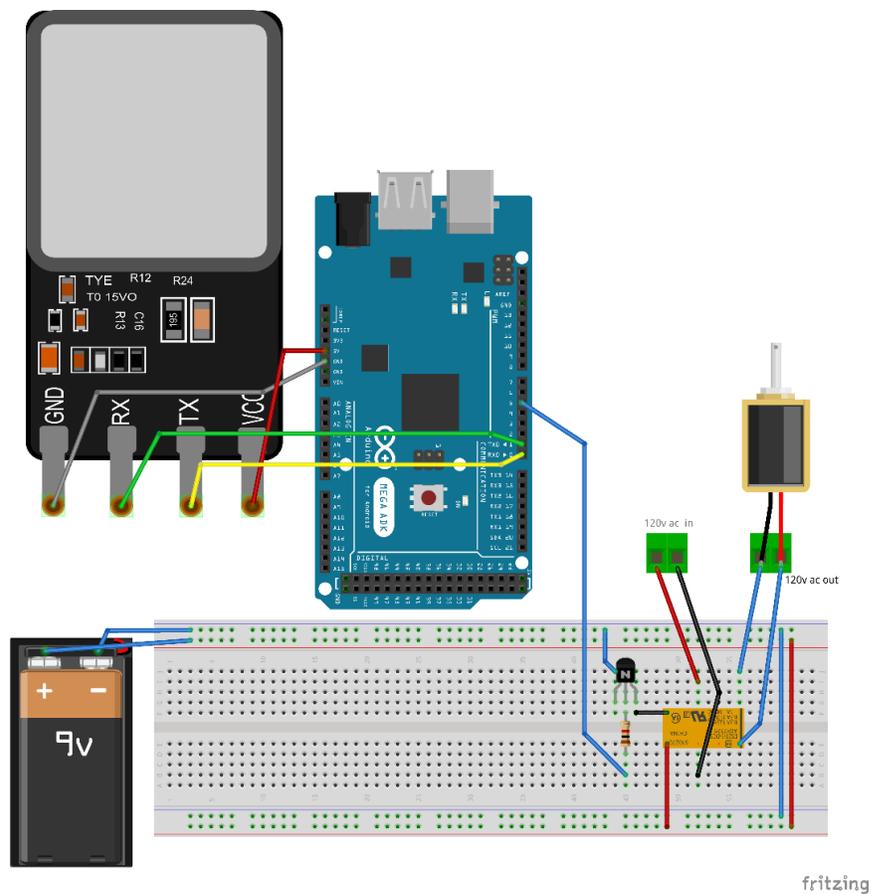
```

#### 4.4. Diagrama de conexión de Arduino

El circuito de conexión de los dispositivos electrónicos hacia el Arduino cuenta con una conexión directa del sensor óptico “Adafruit Optical Fingerprint Sensor”, para la lectura de las huellas digitales. Para el funcionamiento de la chapa eléctrica se cuenta con un circuito electrónico el cual activa un relé para utilizar el voltaje 120 v ac.

**Figura 13.**

*Conexión de sensor óptico con Arduino*



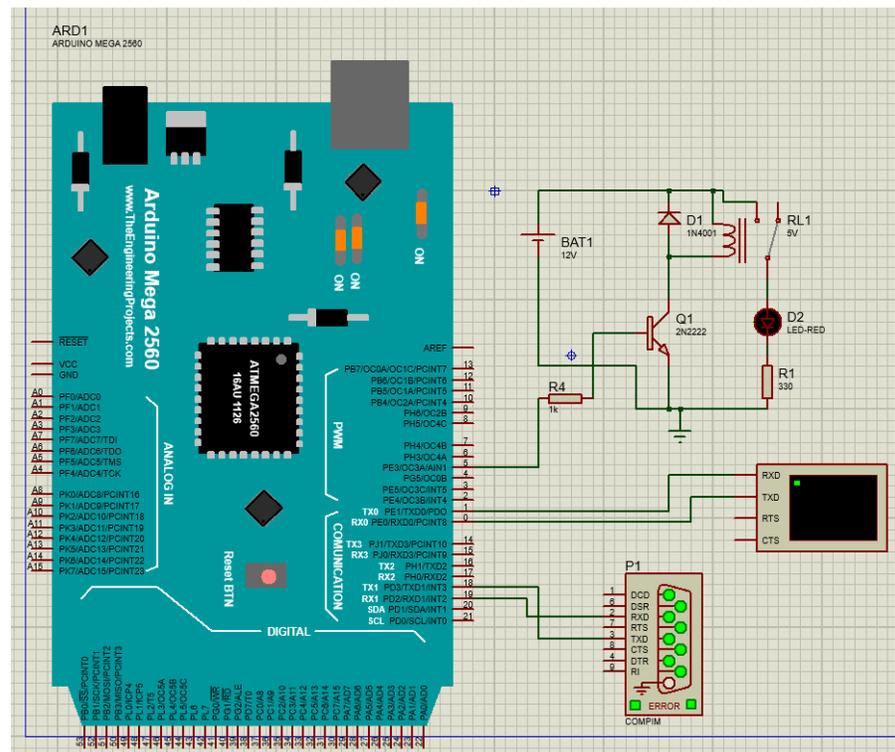
*Nota.* Diagrama de conexión de sensor óptico con Arduino. Elaboración propia, realizado con fritzing.



De esta manera podemos asegurarnos de tener una entrada que sea única y no repetible como lo son las huellas dactilares.

En la figura 15. Se muestra el diseño implementado en Proteus 8, como se puede observar contamos con el monitor serie sustituyendo al sensor Adafruit Optical Fingerprint Sensor y dado que no podemos simular directamente la chapa eléctrica colocamos un diodo led y por último colocamos un puerto COM para realizar la comunicación con la interfaz gráfica realizada en Python.

**Figura 15.**  
*Simulación*



*Nota.* Simulación del sistema biométrico en Proteus 8. Elaboración propia, realizado con Proteus 8 professional.



## **5. MANUAL DE USUARIO Y MANTENIMIENTO**

En el presente manual se presentan las ventanas, mensajes de emergencia y función de los botones que se presentan en la aplicación realizada para el diseño del sistema biométrico, así como las recomendación y uso del sensor biométrico.

### **5.1. Ventana principal**

Esta venta es la primera en mostrarse al iniciar la aplicación, en la figura 16. Se puede observar cómo se diseñó esta ventana, esta ventana tiene la función comunicarse con el sensor biométrico y mostrar la información de la persona que ingresa a la vez que registra la hora de ingreso o salida del usuario.

**Figura 16.**

*Ventana principal*



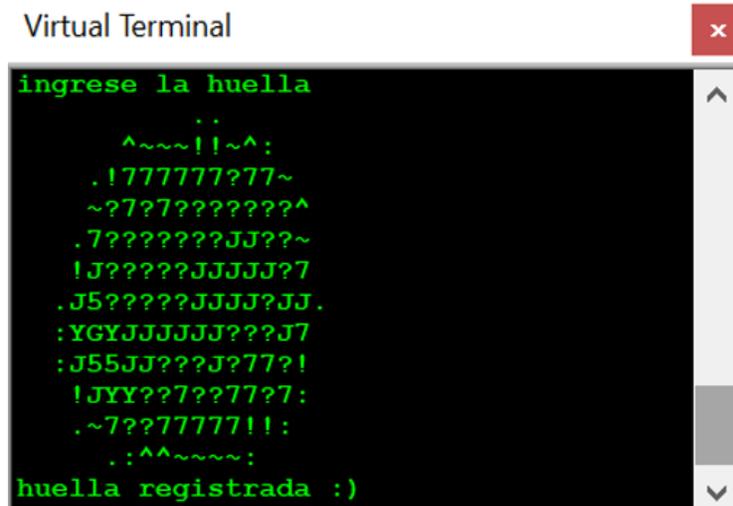
*Nota.* Captura de pantalla de la ventana principal nombrada como ventana de ingreso de personal. Elaboración propia, realizado con Python.

Para su funcionamiento tenemos un texto que da instrucciones en la imparte inferior de la ventana, tal y como indica el primer paso es presionar el botón ingresar huella el cual nos activa el reconocimiento de huellas dactilares en el simulador Proteus 8.

Como resultado tendremos un mensaje que nos indica que podemos ingresar la huella digital el monitor serie, el procedimiento se muestra en la siguiente imagen.

**Figura 17.**

*Mensaje en monitor serial*



*Nota.* Captura de pantalla de la ventana del monitor serie después de haber ingresado una huella dactilar. Elaboración propia, realizado con Proteus 8 Profesional.

esta información será almacenada y comparada por programa principal si la persona existe en el registro o si no se reconoce la huella digital, en la figura 18. Se muestra cómo cambia la pantalla principal al momento de ingresar una persona.

**Figura 18.**

*Ingreso de una persona a las instalaciones*

The screenshot shows a web application window titled "Ingreso de personal" with a sub-header "Asistencia". The form contains the following data:

Nombre	Gabriel Esaú
Apellido	Can Lopez
Dependencia	presupuestos
Fecha	23/03/2023
Hora	00:34:58

Buttons: Registro, Ingresar huella, Exportar

**Bienvenido**

*Nota.* Captura de pantalla de la ventana principal después de haber ingresado una huella dactilar. Elaboración propia, realizado con Python.

Para llevar un control de las personas que van ingresando en el día tenemos dos botones "Registro" y "Exportar", el botón Registro es el que nos servirá si únicamente queremos chequear que personas han ingresado en el día, La figura 19. Muestra la ventana que se genera al presionar el botón.

**Figura 19.**

*Ventana de control de ingreso y salida de personal*



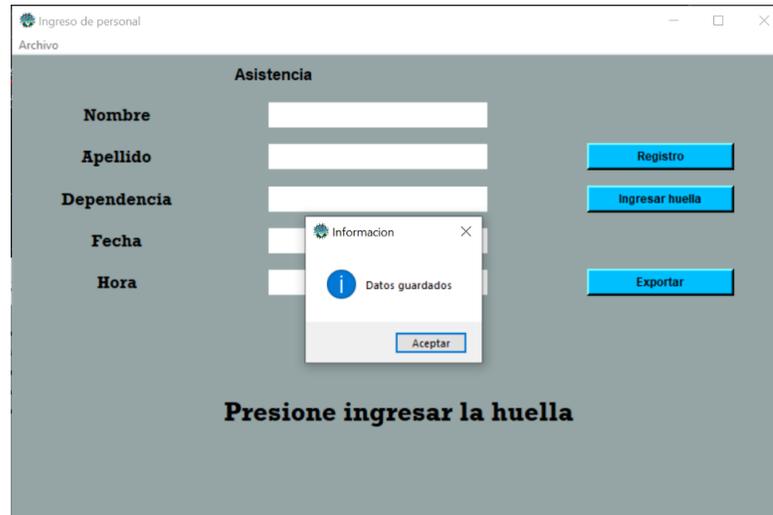
Nombre	Apellido	Dependencia	Fecha	Hora
Manuel	López	Eficiencia	23/03/2023	00:32:31
María	Lucía	Administracion	23/03/2023	00:33:48
Gabriel Esau	Can Lopez	presupuestos	23/03/2023	00:34:58

*Nota.* Captura de pantalla de la tabla de control de ingreso y salida de personal. Elaboración propia, realizado con Python.

El botón Exportar tiene la función de limpiar exportar el registro de la base de datos hacia un archivo de Excel esto con la finalidad de no saturar la base de datos, además de facilitar el uso de los datos para la persona que este a cargo del control de asistencia, el archivo de Excel tendrá como nombre la palabra “Datos” más la fecha y hora en la que fue exportada la información. La figura 21. Muestra el mensaje de emergencia que surge al presionar el botón y la figura 22. Muestra cómo se guarda el archivo de Excel.

**Figura 21.**

*Ventana de emergencia*



*Nota.* Captura de pantalla de aviso por ventana de emergencia. Elaboración propia, realizado con Python

**Figura 22.**

*Control de ingreso y salida de personal en Excel*

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		Nombres	Apellido	Dependencia	Fecha	Hora													
2	0	Manuel	López	Eficiencia	23-03-23_	00:32:31													
3	1	María	Lucía	Administr:	23-03-23_	00:33:48													
4	2	Gabriel Es Can Lopez	presupue:	23-03-23_	00:34:58														
5																			
6																			
7																			
8																			
9																			
10																			
11																			
12																			
13																			
14																			
15																			
16																			
17																			
18																			
19																			
20																			

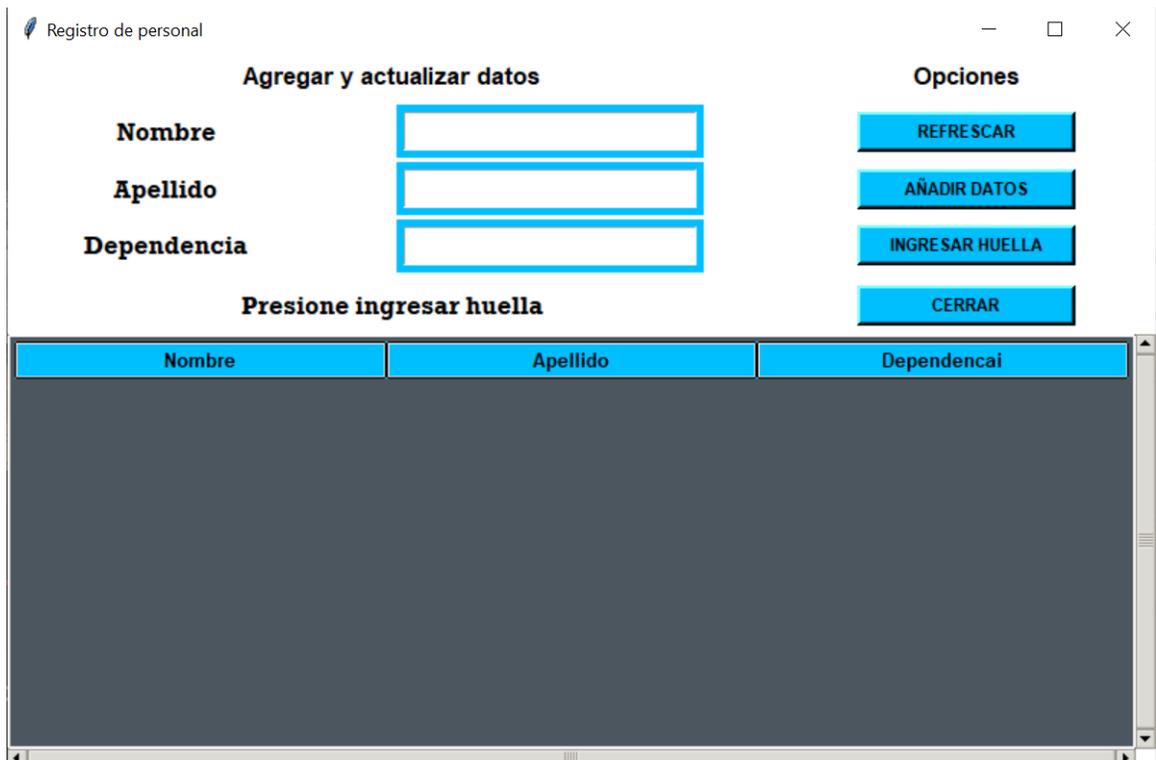
*Nota.* Captura de pantalla tabla generada en Excel al exportar los registros de la base de datos. Elaboración propia, realizado con Excel.

## 5.2. Venta de registro

Esta ventana es un submenú de la pantalla principal, esta ventana tiene una de las funciones más importantes la cual es registrar los datos del personal y registrar de forma digital la huella dactilar que detecta el sensor biométrico, cuenta con 3 entradas para resaltadas por un marco celeste donde se debe ingresar los datos correspondientes de la persona que se desea registrar.

### Figura 23.

*Ventana de ingreso de personal*



The screenshot shows a window titled "Registro de personal" with standard window controls (minimize, maximize, close) in the top right corner. The main content area is divided into two sections. On the left, under the heading "Agregar y actualizar datos", there are three input fields with blue borders, labeled "Nombre", "Apellido", and "Dependencia". Below these fields is the instruction "Presione ingresar huella". On the right, under the heading "Opciones", there are four blue buttons: "REFRESCAR", "AÑADIR DATOS", "INGRESAR HUELLA", and "CERRAR". At the bottom of the window, there is a table with three columns: "Nombre", "Apellido", and "Dependencia". The table body is currently empty and has a dark grey background.

*Nota.* Captura de pantalla de ventana de ingreso de personal. Elaboración propia, realizado con Python.

El primero de los botones es “Refrescar” este botón tiene la función de mostrar una tabla con los datos de las personas que se han registrado previamente, en la figura 24 se muestra la tabla que devuelve el botón “Refrescar”.

**Figura 24.**

*Ventana con datos registrados.*

The screenshot shows a window titled "Registro de personal" with standard window controls. It features a form with three input fields: "Nombre" (Gabriel Esaú), "Apellido" (Can Lopez), and "Dependencia" (presupuestos). To the right, under "Opciones", are four buttons: "REFRESCAR", "AÑADIR DATOS", "INGRESAR HUELLA", and "CERRAR". Below the form, a table displays the following data:

Nombre	Apellido	Dependencia
Gabriel Esaú	Can Lopez	presupuestos
Manuel	López	Eficiencia
Edward	Palencia	Mantenimiento
María	Lucía	Administración

*Nota.* Captura de pantalla de ventana de registro de personal después de importar los registros de la base de datos. Elaboración propia, realizado con Python.

Para registrar a un nuevo usuario se deben escribir los datos del nuevo usuario en los campos correspondientes (Nombre, Apellido, Dependencia), posterior a tener los datos se debe presionar el botón “INGRESAR HUELLA” este botón permitirá que se ingrese la huella dactilar en el sensor óptico mostrando el

mismo mensaje que se observa en la figura 17. En el momento en que aparezca el mensaje de huella registrada se debe presionar el botón “AÑADIR DATOS” de esta manera se registrara el nuevo usuario en la base de datos.

Si se desea eliminar un usuario en caso se haya escrito mal algún dato o el usuario se de baja, se debe dar doble clic en los datos del usuario mostrado en la tabla esto mostrará un mensaje de emergencia preguntando si se desea eliminar el registro, se presionará aceptar para completar el proceso de eliminación del registro.



## 6. RESULTADOS

- La simulación desarrollada en Arduino tuvo dos complicaciones, la primera fue el hecho de las limitaciones del *software* Proteus 8, al no tener entre sus módulos un equivalente al sensor óptico seleccionado para el proyecto, a la vez no se tenía opción para poder ingresar una huella dactilar de forma directa, por esta razón se optó por utilizar el monitor serial ya que de esta manera podemos ingresar datos digitales al microcontrolador Arduino y realice una programación que simule el funcionamiento del sensor óptico “Adafruit Optical Fingerprint Sensor”, teniendo como resultado una interfaz que cumple con la función de recolectar, decodificar y enviar los datos ingresados en el monitor serial al programa principal desarrollado en Python.
- La aplicación cuenta con una interfaz amigable y ergonómica para el usuario de manera que será fácil adaptarse al nuevo sistema además de contar con la utilidad de poder exporta los datos de la tabla de control de ingreso y egreso del sistema a un archivo de Excel, teniendo de esta manera una forma más accesible de manipular la información y evitamos que nuestra base de datos se sature ya que esta información al ser exportada será eliminada de la base de datos principal.
- El sistema optimizara los tiempos de entrada y salida de los empleados al evitar los sistemas convencionales para registrar la hora de entrada, dado que no se necesita de un guardia o secretaria que ingrese los datos de cada usuario, dando un tiempo de entrada menor además de reducir

tiempos de trabajo ya que no se deben de volver a digitar los registros almacenados.

- El programa al desarrollarse en un programa de código abierto tiene posibilidades de mejoras, mantenimientos y optimización de código y dado que Python es un lenguaje de programación con una gran variedad de módulos y librerías tiene posibilidades de mejoras tanto en conectividad a redes de internet como a otros sensores o actuadores externos.

## CONCLUSIONES

1. El diseño propuesto tanto en la parte del código de Arduino como el código realizado para la interfaz gráfica y la base de datos se comunicaron e intercambiaron los datos de forma correcta manteniendo un funcionamiento continuo durante las simulaciones realizadas, lo cual nos asegura el funcionamiento del sistema al ejecutarse en las instalaciones de la Escuela Superior de Artes.
2. En el desarrollo de la base de datos posterior a la investigación teórica se determinó que es más factible realizarla en SQLite dado su versatilidad, fácil uso y el poco espacio en memoria que requiere.
3. Al desarrollar el programa en Python se decidió optar por la librería Tkinter para el desarrollo de la interfaz gráfica, esto permitió tener una mayor facilidad para establecer la comunicación entre la interfaz gráfica y la base de datos.
4. Se realizó el algoritmo de obtención de datos mediante la programación orientada a objetos, dado que necesitamos recurrir a una estructura que cree las plantillas para el intercambio de información entre el código que interpreta los datos y la base de datos, con la finalidad de reducir las líneas de código que serían necesarias para la creación de las ventanas y widgets que conforman la interfaz gráfica.



## RECOMENDACIONES

1. Se recomienda tomar en cuenta las correcciones correspondientes para poder implementar este diseño en un proyecto físico. Dado que al realizarse las pruebas en un simulador se hicieron cambios para que funcionase bajo las restricciones que impuso el *software*, el primer cambio es la programación que debe llevar el microcontrolador Arduino ya que el manual del sensor óptico da una librería que debe usarse para guardar las huellas dactilares.
2. Se recomienda realizar el cambio de la versión de Arduino Mega por un Arduino UNO o un Arduino NANO ya que estos tienen un menor costo y menores dimensiones, este cambio se puede realizar ya que el Arduino Mega fue necesario únicamente por las condiciones impuestas por el simulador.
3. Se recomienda leer el manual proporcionado en este informe de manera que se pueda tener un entendimiento completo del programa y se eviten problemas como la pérdida de información.
4. Se recomienda leer los comentarios escritos a lo largo del código de programación ya que darán una mejor perspectiva y entendimiento del código.

5. Se recomienda crear un repositorio de git o github en caso de querer realizar una modificación o un rediseño del código fuente de esta manera siempre se tendrá acceso a la versión original del programa y podemos tener una línea de tiempo que nos muestre los cambios o mejoras realizadas.
  
6. Se recomienda utilizar la aplicación DB browser para darle mantenimiento a la base de datos, dado que podemos interactuar de manera grafica con las tablas creadas y también podemos modificar la información almacenada.

## REFERENCIAS

Aguayo, P. (2022, 8 julio). *Software de Arduino*. *Arduino.cl - Compra tu Arduino en Línea*. <https://arduino.cl/programacion/>

De Luis, E. R. (2021, septiembre 22). *Empezar con Arduino: qué placa y kits de iniciación comprar*. *Xataka*. <https://www.xataka.com/makers/empezar-arduino-que-placa-kits-iniciacion-comprar>

Guarella, J. E., Heredia, J. P., Rodríguez, L., & Bagatto, I. (2011). *Sensores y actuadores en motores*. *Buenos Aires*.

*Lector de huella digital*. (2021, 17 febrero). INLOC Robotics. <https://inlocrobotics.com/es/lector-de-huella-digital/>

Lukianova, N. (2021, 8 junio). *Principales partes de un Arduino*. *Arduino.cl - Compra tu Arduino en Línea*. <https://arduino.cl/principales-partes-de-un-arduino/>

Morales, S. (2010). *MANUAL DE ORGANIZACION DE LA ESCUELA DE ARTE* [Tesis de pregrado, Universidad de San Carlos de Guatemala]. <https://ddo.usac.edu.gt/wp-content/uploads/2015/01/MANUAL-DE-ORGANIZACION-DE-LA-ESCUELA-DE-ARTE.pdf>

Pulido, M. (2015). ¿Qué lenguaje de programación aprender primero? *Platzi*, <https://platzi.com/blog/lenguajes-de-programacion/>

Rojas, J. (s.f.). *La historia de Python*. Platzi. <https://platzi.com/blog/historia-python/>

*Sensores Biométricos*. (s.f.). INLOC Robotics. <https://inlocrobotics.com/es/ingenieria/sensores-biometricos/>

Serratosa, F. (2008). *La biometría para la identificación de las personas*. [https://sistemamid.com.ar/panel/uploads/biblioteca/2015-03-22\\_12-05-01117594.pdf](https://sistemamid.com.ar/panel/uploads/biblioteca/2015-03-22_12-05-01117594.pdf)

*What is Arduino?* [¿Que es arduino?] (s.f.). Arduino. <https://www.arduino.cc/en/Guide/Introduction/>

## ANEXOS

### Anexo 1.

#### Ficha técnica del sensor óptico “Adafruit Optical Fingerprint Sensor”

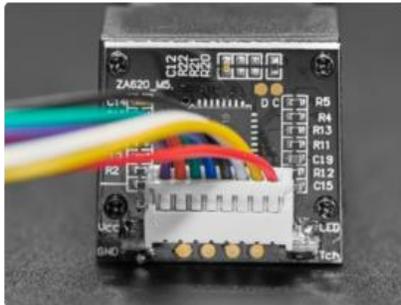
### Wiring for use with Arduino

Once you've tested the sensor, you can now use it within a sketch to verify a fingerprint. We'll need to rewire the sensor. Disconnect the green and white wires and plug the green wire into digital 2 and the white wire to digital 3. (For ESP8266 use 4 & 5, for devices with Hardware UART use 0 & 1)

It is normal for the sensor to blink the LED quickly once powered, after that the LED may stay off until you've started to request data from it



If your fingerprint sensor has individual socket wires (its this one) (<https://adafru.it/QfT>) then use the following wire setup:

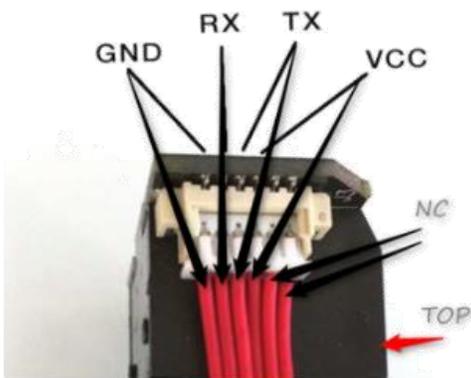
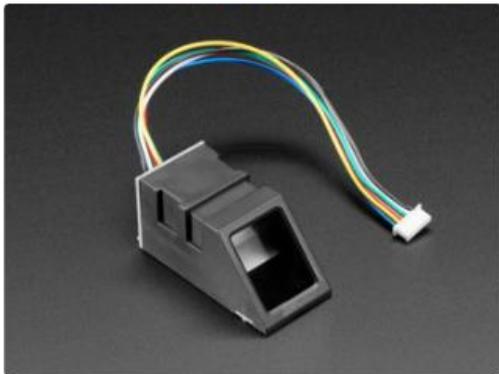


Red Wire to 3.3V  
Yellow wire is Serial TX  
White wire is Serial RX  
Black wire is ground

*Nota:* información sobre los pines de entrada del sensor óptico y método de conexión con Arduino. Adaptado de Ada, L. (2022). *Optical Fingerprint Module Sensor (AS608) For Arduino* [Sensor óptico del módulo de huellas dactilares (AS608) para Arduino] <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf> consultado el 25 de agosto de 2022.

## Anexo 2.

identificación de los pines de entrada



If your cable has a single slim connector on the end and has different color wires:

The first wire from the left should be the black wire ground

then the two data pins: Serial RX is the white wire

Serial TX is the green wire

Then the red power wire (3 or 5V)

You'll have to cut, strip and solder the wires.

If your sensor is an older one and has all the same-color wires, The first wire from the left is ground, then the two data pins, then power. You'll have to cut, strip and solder the wires.

RX is the same as the White wire  
TX is the same as the Green wire

*Nota:* información sobre como identificar los pines de entrada para distintas versiones del sensor. Adaptado de Ada, L. (2022). *Optical Fingerprint Module Sensor (AS608) For Arduino* [Sensor óptico del módulo de huellas dactilares (AS608) para Arduino] <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf> consultado el 25 de agosto de 2022.