

UNIVERSIDAD DE CIENCIAS Y ADMINISTRACIÓN



Licenciatura en Ingeniería en Sistemas Computacionales

PROYECTO CASA DOMÓTICA

TELEMÁTICA

Grupo: SC08BA

Clave de la materia: ISC848

Profesor: INGENIERO SERGIO MANUEL MARTÍNEZ CHÁVEZ

Integrantes:

OREA PÉREZ MIGUEL ENRIQUE

GUILLERMO PEREA CESAR

ANAYA SOTO ALBERTO

LOPEZ DIAZ IVAN

Fecha de entrega: 05/MAYO/2021

ÍNDICE

| | |
|--|----|
| 1. INTRODUCCIÓN | 4 |
| 2. OBJETIVOS DEL PROYECTO | 4 |
| 3. QUÉ ES LA DOMOTICA | 5 |
| 3.1. OBJETIVOS DE LOS SISTEMAS DOMÓTICOS | 5 |
| 3.2. DISPOSITIVOS DEL SISTEMA | 5 |
| 4. DESCRIPCIÓN DE COMPONENTES | 6 |
| 4.1. PROTOBOARD | 6 |
| 4.2. LED | 6 |
| 4.3. RESISTENCIA | 7 |
| 4.4. MÓDULO WIFI | 7 |
| 4.5. SERVOMOTOR | 7 |
| 4.6. MÓDULO BLUETOOTH HC-05 | 8 |
| 4.7. PLACA ARDUINO “UNO” | 9 |
| 4.8. SENSOR DE HUELLA DACTILAR | 9 |
| 5. DISEÑO | 10 |
| 5.1. DESCRIPCIÓN DE IMPLEMENTACIÓN DE CADA ETAPA | 10 |
| 5.2. DIAGRAMA A BLOQUES POR ETAPAS | 10 |
| 5.3. DISEÑO DEL PROTOTIPO | 11 |
| 5.4. VIDEO DESCRIPTIVO DEL DISEÑO E IMPLEMENTACIÓN | 12 |
| 6. PROGRAMACIÓN | 12 |
| 6.1. MÓDULO BLUETOOTH | 12 |
| 6.2. IMPLEMENTACIÓN DE LA APLICACIÓN MÓVIL | 15 |
| 6.3. MÓDULO WIFI | 16 |
| 6.4. MÓDULO DE HUELLA DACTILLAR | 19 |
| 7. PRESUPUESTO | 25 |
| 8. RESULTADOS | 26 |
| 10. CONCLUSIONES | 26 |
| 11. REFERENCIAS | 26 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1: Smart Home | 5 |
| Figura 2: Protoboard | 6 |
| Figura 3: Led | 7 |
| Figura 4: Resistencia..... | 7 |
| Figura 5: Modulo WIFI..... | 7 |
| Figura 6: ServoMotor | 8 |
| Figura 7: MÓDULO BLUETOOTH HC-05 | 9 |
| Figura 8: PLACA ARDUINO “UNO” | 9 |
| Figura 9: SENSOR DE HUELLA DACTILAR | 10 |
| Figura 10: Diagrama de puerta(blueooth) | 10 |
| Figura 11: Diagrama de ventana(WIFI) | 10 |
| Figura 12: Diagrama de leds (Huella Digital) | 11 |
| Figura 13: Diseño de la casa part.1 | 11 |
| Figura 14: Diseño de la casa part.2 | 11 |
| Figura 15: Diseño de la casa part.3 | 11 |
| Figura 16: Diseño de la casa part.4 | 11 |
| Figura 17: Diseño de la casa part.5 | 12 |
| Figura 18: Diseño de la casa part.6 | 12 |
| Figura 19: Diseño de la casa part.7 | 12 |
| Figura 20: Conexión de placa Arduino con módulo bluethooth | 12 |
| Figura 21: Conexión de placa Arduino con módulo bluethooth con “EN” | 13 |
| Figura 22: Conexión de placa Arduino con módulo bluethooth sin “EN” | 13 |
| Figura 23: Aplicación móvil | 15 |
| Figura 24: Módulos disponibles | 16 |
| Figura 25: Programación de botones | 16 |
| Figura 26: Programación de listas | 16 |
| Figura 27: Conexión de protoboard, módulo wifi y servomotor..... | 17 |
| Figura 28: Interfaz gráfica de sitio web | 19 |
| Figura 29: Pantalla de acceso | 19 |
| Figura 30: Conexión de placa Arduino con sensor de huella dactilar | 19 |
| Figura 31: Conexión de placa Arduino con protoboard y led..... | 23 |
| Figura 32: Modelo de conexión para varios leds..... | 23 |
| Figura 33: Pantalla de funcionamiento del sensor de huellas..... | 24 |
| Figura 34: Presupuesto 1..... | 25 |
| Figura 35: Presupuesto 2..... | 25 |
| Figura 36: Presupuesto 3..... | 25 |
| Figura 37: Presupuesto 4..... | 25 |
| Figura 38: Presupuesto 5..... | 25 |

Proyecto Casa Inteligente

Orea Pérez Miguel Enrique

Guillermo Perea Cesar

Anaya Soto Alberto

López Diaz Iván

Universidad de Ciencias y Administración

UCAD

1. INTRODUCCIÓN

La infraestructura del hogar debe adaptarse a las nuevas tecnologías, y esto hay que tenerlo en cuenta a la hora de diseñar y construir una vivienda, igual que ocurre con la distribución de electricidad o de agua. Llevamos muchos años ya con la instalación eléctrica convencional y empezamos a observar necesidades enfocadas a la simplificación de las tareas domésticas que hasta ahora no habían sido relevantes. La respuesta será probablemente la "Domótica". La vivienda inteligente es el resultado de la integración de sistemas y equipos que permiten cumplir las necesidades de sus habitantes referentes a la seguridad, confort, gestión y control, telecomunicaciones y ahorro de energía. Para que esta infraestructura funcione se necesitan unos conocimientos teóricos y prácticos para su correcta instalación. El sistema domótico de la vivienda debe permitir tener conectadas todas las estancias de la vivienda con un control total para el usuario de la iluminación, de las persianas, toldos, de la climatización de cada una de las habitaciones, etc.

2. OBJETIVOS DEL PROYECTO

El principal objetivo de este proyecto es dar a conocer el tipo de aplicaciones y mejoras que se pueden introducir en una vivienda para dotarla de "inteligencia" y los beneficios que los usuarios de la misma pueden obtener gracias a esta automatización.

Se busca diseñar e implementar una solución domótica para el control de una vivienda para hacerla moderna, ecológica y rentabilizarle.

Los principales objetivos son:

- Dar a conocer los tipos de tecnologías que se pueden utilizar para automatizar la vivienda.
- Explicar en profundidad el estándar a utilizar que mejor se adapten a nuestras necesidades.
- Ver el modo de funcionamiento y de comunicación entre elementos del sistema. Implementar los elementos necesarios para dotar a la vivienda de seguridad, confort y ahorro de energía.

Para ello se realizará un estudio global sobre la domótica y las tecnologías existentes en el mercado. Se reducirán los distintos componentes que son necesarios para automatizar una vivienda, teniendo en cuenta la "Guía técnica de aplicación de instalaciones de sistemas de automatización gestión técnica de energía y seguridad para viviendas y edificios" del Ministerio de Industria, Turismo y Comercio.

3. QUÉ ES LA DOMOTICA

El término domótica proviene de la unión de las palabras domus (que significa casa en latín) y tica (de automática, palabra en griego, 'que funciona por sí sola'). Se entiende por domótica al conjunto de sistemas capaces de automatizar una vivienda, aportando servicios de gestión energética, seguridad, bienestar y comunicación, y que pueden estar integrados por medio de redes de comunicación pudiendo ser controlados desde dentro y fuera del hogar. La tendencia de futuro en la demanda se centra básicamente en conseguir un hogar totalmente conectado, capaz de integrar las nuevas tecnologías que van apareciendo, con sistemas sencillos y totalmente gestionables, pero garantizando la seguridad y privacidad, todo esto de manera transparente en el hogar. En definitiva, la domótica es el uso simultáneo de electricidad, electrónica, informática y comunicaciones aplicadas a la gestión de las viviendas. Los sistemas de automatización, gestión técnica de energía y seguridad para viviendas y edificios, se conocen internacionalmente como HBES (Home and Buildings electronic Systems).



Figura 1: Smart Home

3.1. OBJETIVOS DE LOS SISTEMAS DOMÓTICOS

La instalación domótica tiene como objetivo encargarse de gestionar aspectos fundamentales del sistema como se define en la guía técnica de aplicación de instalaciones de sistemas de automatización como seguridad, confort, energía, comunicaciones, entre otras.

Todos estos objetivos se pueden alcanzar mediante tipos de formas de control remoto. Se puede aplicar el control remoto desde dentro de la vivienda (mando a distancia), desde fuera de la vivienda (telefónicamente o a través de internet) o programando funciones según se cumplan condiciones horarias o climatológicas.

3.2. DISPOSITIVOS DEL SISTEMA

Controlador: es el dispositivo que gestiona el sistema según la programación y la información que recibe. Puede haber un controlador solo, o varios distribuidos por el sistema.

Actuador: es un dispositivo capaz de ejecutar y/o recibir una orden del controlador y realizar una acción sobre un aparato o sistema (encendido/apagado, subida/bajada, apertura/cierre, etc.).

Sensor: es el dispositivo que monitoriza el entorno, tanto interior como exterior, captando información que transmite al sistema (sensores de agua, gas, humo, temperatura, viento, humedad, lluvia, iluminación, etc.).

Bus: es el medio de transmisión que transporta la información entre los distintos dispositivos por un cableado propio, por las redes de otros sistemas (red eléctrica, red telefónica, red de datos) o de forma inalámbrica.

Interface: nos referimos a los dispositivos (pantallas, móvil, Internet, conectores) y los formatos (binario, audio) en que se muestra la información del sistema para los usuarios (u otros sistemas) y donde los mismos pueden interactuar con el sistema.

4. DESCRIPCIÓN DE COMPONENTES

Bluetooth es una especificación industrial para

Redes Inalámbricas de Área Personal (WPAN) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2.4 GHz. Los principales objetivos que se pretenden conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles.
- Eliminar los cables y conectores entre estos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

4.1. PROTOBOARD

Es una placa de pruebas en los que se pueden insertar elementos electrónicos y cables con los que se arman circuitos sin la necesidad de soldar ninguno de los componentes. Las Protoboard tienen orificios conectados entre sí por medio de pequeñas láminas metálicas. Usualmente, estas placas siguen un arreglo en el que los orificios de una misma fila están

conectados entre sí y los orificios en filas diferentes no. Los orificios de las placas normalmente tienen una separación de 2.54 milímetros (0.1 pulgadas).

Una Protoboard es un instrumento que permite probar el diseño de un circuito sin la necesidad de soldar o desoldar componentes. Las conexiones en una Protoboard se hacen con solo insertar los componentes lo que permite armar y modificar circuitos con mayor velocidad.

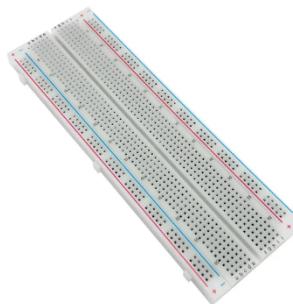


Figura 2: Protoboard

4.2. LED

Los diodos son componentes electrónicos que permiten el paso de la corriente en un solo sentido, en sentido contrario no deja pasar la corriente (como si fuera un interruptor abierto). Un diodo Led es un diodo que además de permitir el paso de la corriente solo en un sentido, en el sentido en el que la corriente pasa por el diodo, este emite luz. Cuando se conecta un diodo en el sentido que permite el paso de la corriente se dice que está polarizado directamente.



Figura 3: Led

4.3. RESISTENCIA

La resistencia es un dispositivo eléctrico que tiene la particularidad de oponerse al flujo de la corriente. Para medir el valor de las resistencias se usa un instrumento llamado ohmetro y las unidades en el S.I es el Ohm.

En general todo material presenta una resistencia natural, la cual depende de su estructura interna, las impurezas y composición atómica.

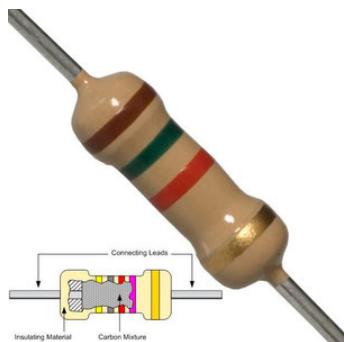


Figura 4: Resistencia

4.4. MÓDULO WIFI

El ESP8266 es un chip altamente integrado diseñado para las necesidades de un nuevo

mundo conectado. Ofrece una solución completa y autónoma de redes Wi-Fi, lo que le permite alojar la aplicación o servir como puente entre Internet y un microcontrolador.

El ESP8266 tiene potentes capacidades a bordo de procesamiento y almacenamiento que le permiten integrarse con sensores y dispositivos específicos de aplicación a través de sus GPIOs con un desarrollo mínimo y carga mínima durante el tiempo de ejecución. Su alto grado de integración en el chip permite una circuitería externa mínima, y la totalidad de la solución, incluyendo el módulo está diseñado para ocupar el área mínima en un PCB.



Figura 5: Modulo WIFI

4.5. SERVOMOTOR

También llamado servo, son dispositivos de accionamiento para el control de precisión de velocidad, par motor y posición. Constituyen un mejor desempeño y precisión frente a accionamientos mediante convertidores de frecuencia, ya que éstos no nos proporcionan control de posición y resultan poco efectivos en bajas velocidades.

Es un servomotor, aquel que contiene en su interior un encoder, conocido como decodificador, que convierte el movimiento

mecánico (giros del eje) en pulsos digitales interpretados por un controlador de movimiento. También utilizan un driver, que en conjunto forman un circuito para comandar posición, torque y velocidad.



Figura 6: ServoMotor

4.6. MÓDULO BLUETOOTH HC-05

El módulo Bluetooth HC-05 nos permite conectar nuestros proyectos con Arduino a un smartphone, celular o PC de forma inalámbrica (Bluetooth), con la facilidad de operación de un puerto serial. La transmisión se realiza totalmente en forma transparente al programador, por lo que se conecta en forma directa a los pines seriales de nuestro microcontrolador preferido (respetando los niveles de voltaje, ya que el módulo se alimenta con 3.3V). Todos los parámetros del módulo se pueden configurar mediante comandos AT.

La comunicación Bluetooth se da entre dos tipos de dispositivos: un maestro y un esclavo. Si nuestro objetivo es conectar nuestro proyecto a un smartphone android podemos utilizar tanto un módulo HC-06 o un HC-05 configurado como esclavo. El módulo Bluetooth HC-05 viene configurado de fábrica para trabajar como esclavo, es decir, preparado para escuchar peticiones de conexión, pero

podemos configurarlo para trabajar con Maestro utilizando comandos AT. Por otra parte, si nuestro objetivo es conectar dos proyectos, necesitaremos utilizar un módulo HC-05 configurado como maestro y un HC-06 (esclavo) o un HC-05 configurado como esclavo.

Este módulo cumple con las especificaciones del estándar Bluetooth 2.0 que es perfectamente compatible con celulares o smartphones Android, más no con los Iphone. Para trabajar con Iphone recomendamos utilizar el Módulo Bluetooth 4.0 BLE HM-10, que también es compatible con los celulares Android modernos.

Algunas de sus características son las siguientes;

- Voltaje de operación: 3.6V - 6V DC
- Consumo corriente: 50mA
- Bluetooth: V2.0+EDR
- Frecuencia: Banda ISM 2.4GHz
- Modulación: GFSK(Gaussian Frequency Shift Keying)
- Potencia de transmisión: 4dBm, Class 2
- Sensibilidad: -84dBm a 0.1% BER
- Alcance 10 metros
- Interface comunicación: Serial TTL
- Velocidad de transmisión: 1200bps hasta 1.3Mbps
- Baudrate por defecto: 38400,8,1,n.
- Seguridad: Autenticación y encriptación
- Temperatura de trabajo: -20C a +75C
- Compatible con Android

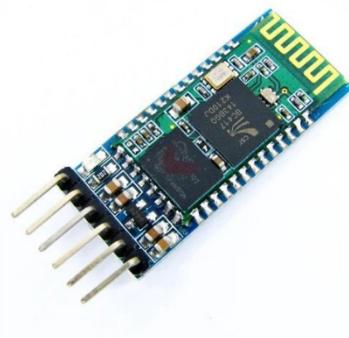


Figura 7: MÓDULO BLUETOOTH HC-05

4.7. PLACA ARDUINO “UNO”

Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteo. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador.

Algunas de sus características son;

- Características técnicas de Arduino Uno r3
- Microcontrolador: ATmega328
- Voltage: 5V
- Voltaje entrada (recomendado): 7-12V
- Voltaje entrada (límites): 6-20V
- Digital I/O Pins: 14 (de los cuales 6 son salida PWM)

- Entradas Analógicas: 6
- DC Current per I/O Pin: 40 mA
- DC Current para 3.3V Pin: 50 mA
- Flash Memory: 32 KB (ATmega328) de los cuales 0.5 KB son utilizados para el arranque
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz



Figura 8: PLACA ARDUINO “UNO”

4.8. SENSOR DE HUELLA DACTILAR

El lector huella digital funciona colocando el dedo sobre un sensor que reconoce la huella dactilar del trabajador, el sensor digitaliza el dedo del usuario, captura la imagen tridimensional de la huella dactilar, la cual se compara con una imagen de la huella dactilar del usuario previamente capturada para comprobar su correspondencia. Si se encuentra la correspondencia el empleado es identificado como ya registrado

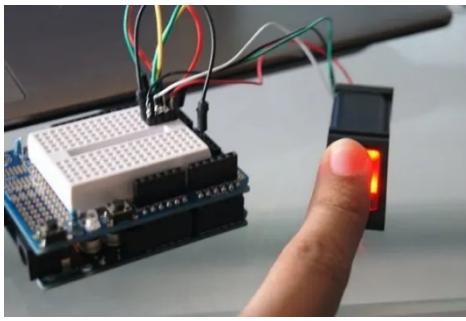


Figura 9: SENSOR DE HUELLA DACTILAR

5. DISEÑO

5.1. DESCRIPCIÓN DE IMPLEMENTACIÓN DE CADA ETAPA

Implementación bluetooth. A través de la aplicación, valida si el módulo bluetooth tiene alguna acción predeterminada, si es así, cumple la función establecida de cambio de estados, en dado caso que no sea así, termina el ciclo de funcionamiento.

Implementación wifi. Desde la interfaz web, verifica si hay algún evento establecido en el módulo wifi, en dado caso que sea cierto activa el evento de cambio de estados establecido en el módulo, si no es así, termina el ciclo de funcionamiento.

Implementación dactilar. El sensor valida la huella ingresada para dicho funcionamiento establecido, activando dicho evento si la huella digital es reconocida por el mismo sensor, al no ser así, no se activará ningún evento o acción dentro de la implementación.

5.2. DIAGRAMA A BLOQUES POR ETAPAS

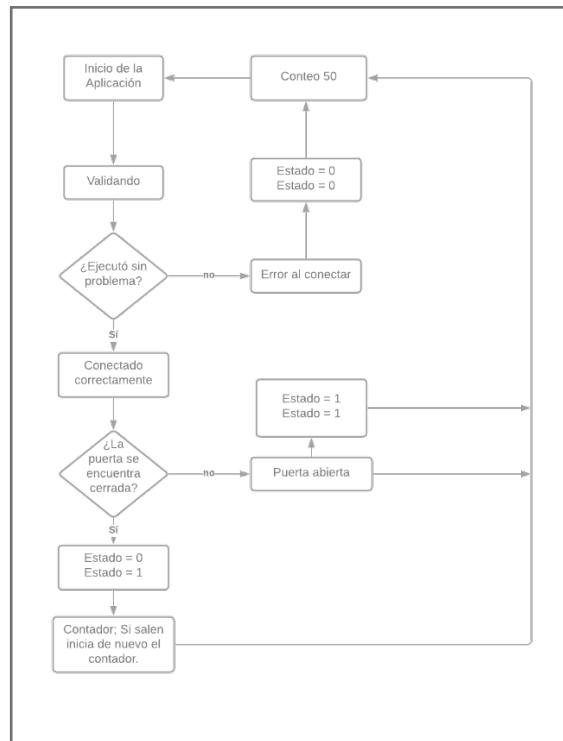


Figura 10: Diagrama de puerta(blueooth)

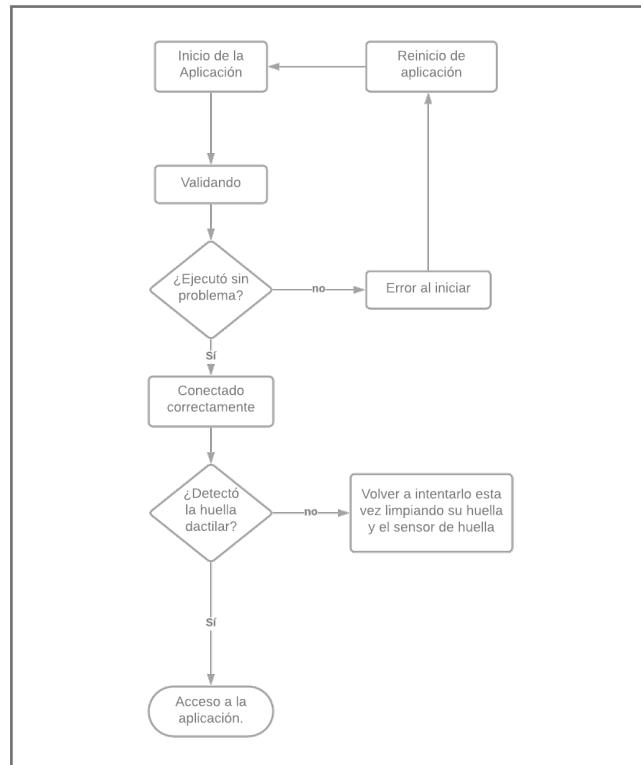


Figura 11: Diagrama de ventana(WIFI)

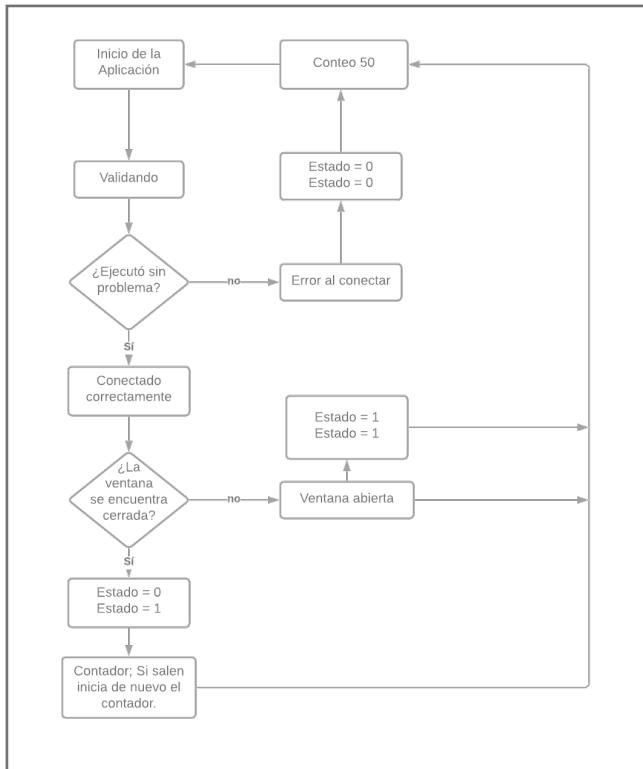


Figura 12: Diagrama de leds (Huella Digital)

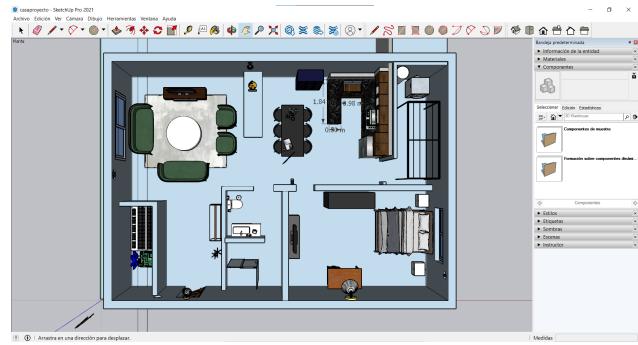


Figura 14: Diseño de la casa part.2

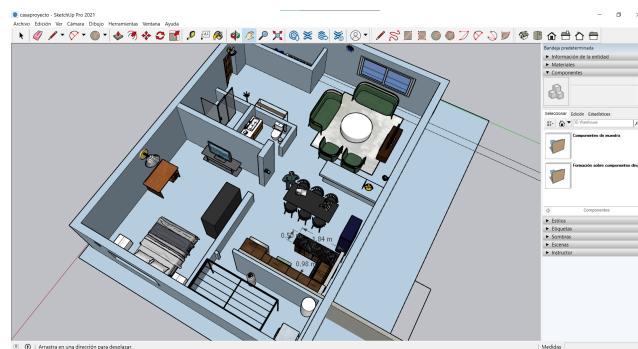


Figura 15: Diseño de la casa part.3

5.3. DISEÑO DEL PROTOTIPO

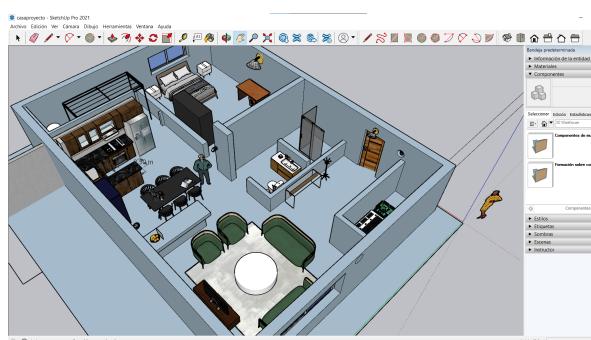


Figura 13: Diseño de la casa part.1

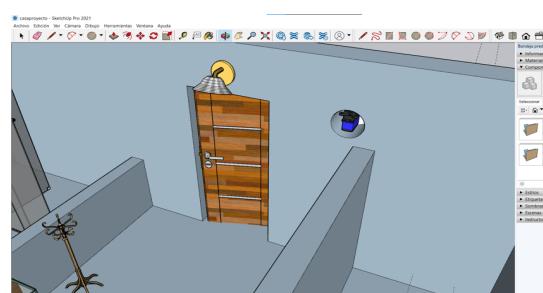


Figura 16: Diseño de la casa part.4

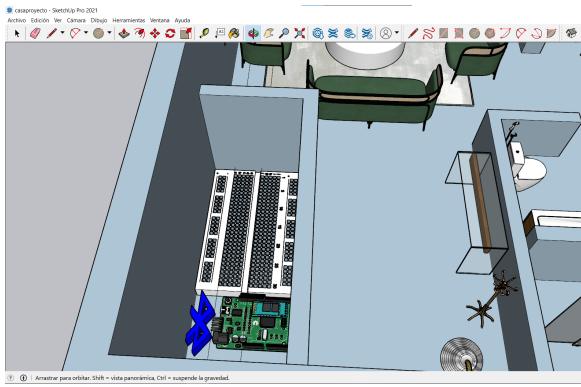


Figura 17: Diseño de la casa part.5

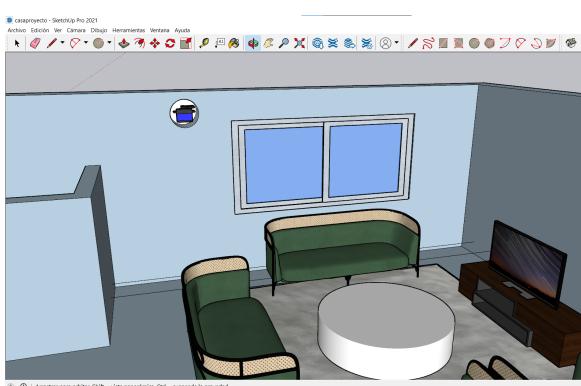


Figura 18: Diseño de la casa part.6

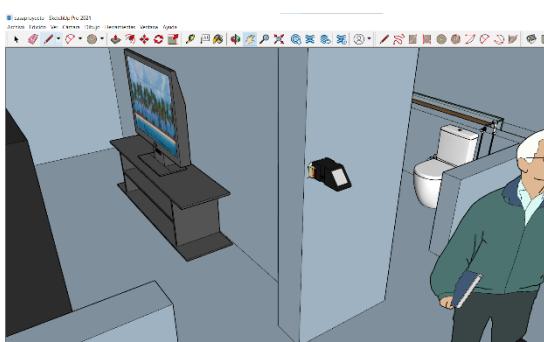


Figura 19: Diseño de la casa part.7

5.4. VIDEO DESCRIPTIVO DEL DISEÑO E IMPLEMENTACIÓN

<https://drive.google.com/file/d/1Q5zK3XhxIoZHEbXOUK7HUWrkVtPRF25R/view?usp=sharing>

6. PROGRAMACIÓN

6.1. MÓDULO BLUETOOTH

Este módulo se utilizará para abrir y cerrar la puerta de esta casa domótica. Para llevar a cabo este proceso se debe obtener los siguientes componentes:

- Modulo bluetooth HC-05
- Placa Arduino
- Cables
- Servomotor

Pero antes de empezar la construcción de los componentes se necesitará la programación para el módulo bluetooth para que funcione correctamente. Para comenzar a programar el módulo primeramente se conectarán el módulo bluetooth con la placa Arduino, como se muestra en la siguiente figura:

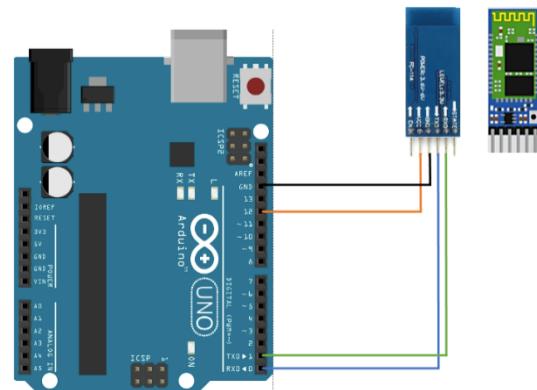


Figura 20: Conexión de placa Arduino con módulo bluetooth

Donde:

VCC: Es el pin de conexión para alimentar el módulo bluetooth.

GND: Es el pin de conexión a tierra del módulo bluetooth.

TXD: Es el pin que establece la comunicación

entre el módulo bluetooth. En este caso establece la transmisión del módulo bluetooth al Arduino.

RXD: Es el pin que establece la comunicación entre el módulo bluetooth. En este caso establece la recepción del módulo bluetooth al Arduino.

EN: Es el pin que permite al módulo ser configurable o programable.

La razón de porque los pines del módulo se conectarán en los puertos seriales digitales es que es más fácil de programar o configurar el módulo ya que no se necesitarán librerías y funciones especiales para la simulación de este.

La conexión del pin VCC del módulo al pin digital 12 del Arduino tendrá como objetivo alimentar el módulo una vez que se haya cargado el programa al Arduino, evitando que se tengan problemas en los puertos series.

Se conectará el pin GND del módulo al pin digital GND de Arduino.

Se conectará el pin TXD del módulo al pin 0 del Arduino, este pin 0 es la recepción del Arduino. Esta conexión consiste en que el módulo bluetooth transmitirá datos o señales y el Arduino será el receptor de dichos datos o señales.

Se conectará el pin RXD del módulo al pin 1 del Arduino, este pin 1 es la transmisión del Arduino. Esta conexión consiste en que el módulo bluetooth recibirá datos o señales que el Arduino proporcione.

Algo muy importante del módulo bluetooth es que si este trae un botón en el pin EN no se conectará directamente al Arduino, ya que este

pin se activará más adelante en la configuración o programación del módulo.

En conclusión, si el módulo trae el botón en el pin EN no se conectará directamente al Arduino; en caso contrario tendrá que ir conectado directamente en los pines de poder del Arduino. A continuación, se mostrará la conexión de ambos casos en la siguiente figura.

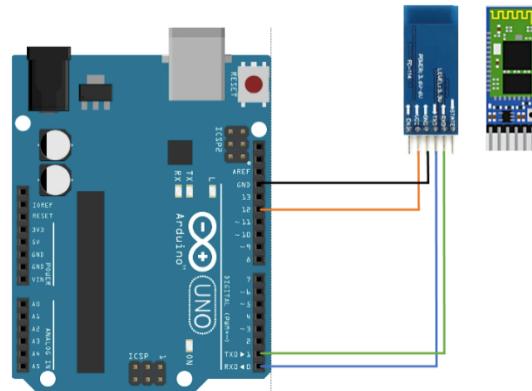


Figura 21: Conexión de placa Arduino con módulo bluetooth con "EN"

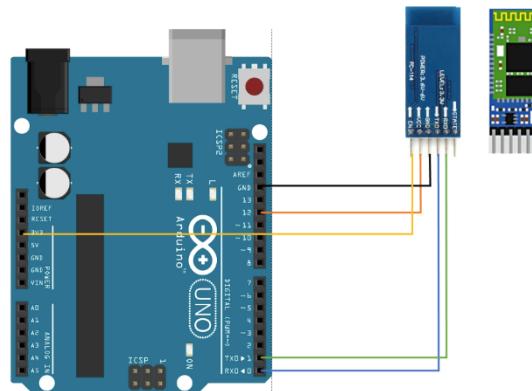


Figura 22: Conexión de placa Arduino con módulo bluetooth sin "EN"

Después de hacer las conexiones para el módulo bluetooth se procederá a hacer la conexión para el servomotor que se usará para abrir y cerrar la puerta.

El servomotor irá conectado en el pin digital 3 del Arduino, la tierra en GND y será conectado

al pin de 5v del Arduino. Como se muestra en la siguiente figura.

A continuación, se muestra la programación que se llevará a cabo para la configuración del módulo bluetooth y del servomotor.

```
include <Servo.h>
Servo motor;

//Declaramos las constantes y variables para los pines
const int LED = 13; // Led para notificar que la configuración del módulo a terminado
const int POWER = 12; //constante de alimentación para el pin 12
int estado=0; //Estado para los botones de la aplicación
char nombreB[8] = "Puerta" //nombre del módulo
char pin[4] = "1234"; //Clave de acceso
char velocidad = 4 // Velocidad del módulo 4 = 9600 baudios
char modo = '1' // 1 = maestro 0 = esclavo

// Configuración

void setup(){
motor.attach(3); // Motor conectado en el pin 3
pinMode(LED, OUTPUT); // pin 13 como salida
pinMode(POWER,OUTPUT); // pin 12 como salida
Serial.begin(38400); // Comunicarse con el módulo bluetooth a una velocidad de configuración de 38400 baudios
```

```
/* Solo si se tiene el botón en EN */
digitalWrite(LED, HIGH); // Modo de configuración (enciende led)
delay(4000); // 4 segundos para presionar botón
digitalWrite(LED, LOW); // Modo de configuración (apaga led)
/* En dado caso de no tener botón en el módulo omitir esta parte de código*/

digitalWrite(POWER, HIGH); // Alimentar módulo
delay(3000);
Serial.print("AT\r\n"); // Indica el inicio de la configuración del módulo

//Cambia el nombre del módulo
Serial.print("AT+NAME");
Serial.print(nombreB);
Serial.print("\r\n");

//Cambia el pin del módulo
Serial.print("AT+PIN");
Serial.print(pin);
Serial.print("\r\n");

//Cambia la velocidad del módulo
Serial.print("AT+BAUD");
Serial.print(velocidad);
Serial.print("\r\n");

//Cambia el modo del módulo
Serial.print("AT+MODE");
Serial.print(modo);
Serial.print("\r\n");

digitalWrite(LED, HIGH);
}

void loop(){
```

```

if(Serial.available()>0){ //Comprobar si
el puerto serial esta habilitado
estado = Serial.read(); //Se almacena
los valores que se le de desde la
aplicación
}

//Encendido
if (estado =='1'){
abrir();
}

//Apagado
if(estado=='2'){
cerrar
}
delay(5000);
}

// Función para abrir puerta
void abrir(){
motor.write(10); //angulo
delay(200);
motor.write(20); //angulo
delay(200);
}

//Función para cerrar puerta
void cerrar(){
motor.write(20); //Angulo
delay(200);
motor.write(10); //Angulo
delay(200);
}

```

6.2. IMPLEMENTACIÓN DE LA APLICACIÓN MÓVIL

El módulo bluetooth funcionará por medio de una aplicación móvil; esta aplicación se desarrollará en el sitio web “app inventor”,

donde este software ayudará a crear, diseñar programar e implementar los componentes necesarios para el funcionamiento correcto de la aplicación. Cabe resaltar que la programación que se llevará a cabo para la aplicación será programación por bloques que implica encastrar piezas prediseñadas de tal forma que genere una lista de acciones a seguir.

A continuación, se mostrará la programación y diseño de la aplicación

Primero se realizará el diseño de la aplicación donde se colocarán etiquetas y botones para la interfaz de la aplicación móvil, arrastrando solamente los componentes a la pantalla del simulador móvil que aparece. Como se muestra en la siguiente figura.



Figura 23: Aplicación móvil

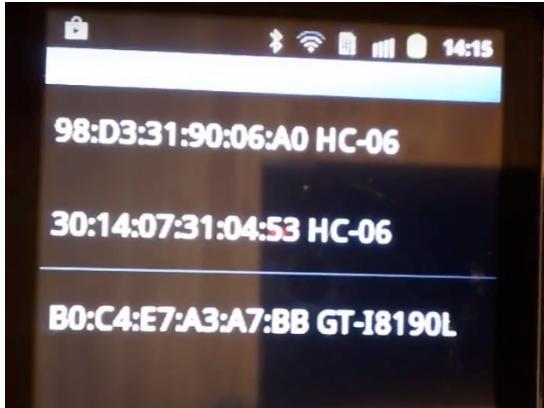


Figura 24: Módulos disponibles

Después de tener la interfaz terminada se procederá a la realización de la programación de cada uno de los componentes que se colocaron en la interfaz.

La programación para los botones es la siguiente:

- Se pondrá un bloque de botón que al presionar el botón llamará a otro bloque llamado bluetooth cliente que enviará un texto, en este caso “1”. Para esta aplicación el procedimiento será para ambos botones. Como se muestra en la siguiente figura.

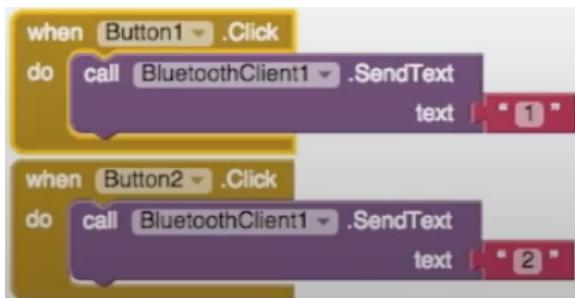


Figura 25: Programación de botones

- Para que los dos botones hagan su correcto funcionamiento se tiene que activar el módulo bluetooth programando el botón de búsqueda de bluetooth. Se harán dos programaciones para este botón, la

primera es antes de que sea oprimido y la segunda es después de que sea oprimido.

En la primera (antes) se colocará un bloque llamado selector de lista que al ejecutar pondrá otro bloque de selector de lista donde pondrá los nombres de módulos bluetooth sincronizados a nuestro teléfono.

En la segunda (después) se colocará un bloque llamado selector de lista que ejecutará un bloque llamado selector de lista donde recolectará la selección del módulo llamado.

Estos procedimientos se pueden ver en la siguiente figura.

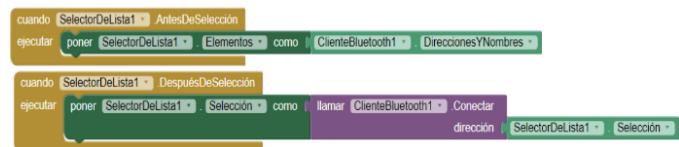


Figura 26: Programación de listas

6.3. MÓDULO WIFI

Este módulo se utilizará para abrir y cerrar la ventana de esta casa domótica. Para llevar a cabo este proceso se deben obtener los siguientes componentes:

- Módulo wifi ESP-M2.
- Protoboard
- Cables
- Servomotor

Las conexiones se llevarán a cabo en la protoboard colocando el módulo wifi encima de ella y posteriormente conectar el módulo con el servomotor. Las conexiones se pueden observar en la siguiente figura.

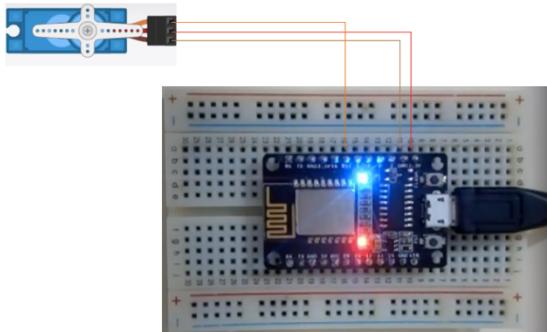


Figura 27: Conexión de protoboard, módulo wifi y servomotor

A continuación, se muestra la programación que se llevará a cabo para la configuración del módulo wifi y con su respectiva interfaz en la web para su funcionamiento correcto.

```
#include <ESP8266WiFi.h>
#include <servo.h>

const char* ssid = "WLAN_53CU";
//Nombre de la red

const char* password = "*****";
//Contraseña

WiFiServer server(80); //Iniciamos en el
//servidor en el puerto 80

servo.attach(16); //Pin donde trabajará el
//servomotor

int estado = LOW;

void setup(){
  serial.begin(9600); //Monitor serie para
  //acceder a la url
  pinMode(pinled,OUTPUT); //Pin 16 como
  //salida
```

```
digitalWrite(pinled,LOW); //Comience con
//un bajo promedio
```

```
WiFi.begin(ssid,password); //Utilizar los
//parámetros para entrar a internet
```

```
Serial.printf("\n\nConectando a la red:
%s\n", WiFi.ssid().c_str());
```

```
/* En caso de no conectarse mostrará
puntos */
```

```
while(WiFi.status() !=
WL_CONNECTED){
delay(500);
Serial.print(".");
}
```

```
Serial.println("");
Serial.println("wifi conectado");
```

```
server.begin(); //Inicializa servidor
Serial.println("Servidor inicializado");
```

```
Serial.printf("\n\nUtiliza esta URL para
conectar:
http://%s\n", WiFi.localIP().toString().");
//Pone la dirección URL
}
```

```
void loop(){
/* Si no está el servidor disponible no se
ejecutará nada*/
wificlient client = sever.available();
if(!client){
return;
}
```

```
Serial.println("nuevo cliente"); //Cuando
//se acceda a la URL
while(!client.available()){
delay(1);
}
```

```

//Guarda información enviada en la
variable peticion

String peticion =
client.readStringUntil('\r');
Serialprintln(peticion);
client.flush();

if(peticion.indexOf('/LED=ON') != -1)
{ estado = LOW;}
if(peticion.indexOf('/LED=OFF') != -1)
{estado = HIGH;}

digitalWrite(pinled, estado);
/* Configuración de la página*/
client.println("HTTP://1.1 200 OK");
client.println("");
client.println("");
client.println("");
client.println("");

client.print("<h1 align=center>Control de
ventana");

//Botones y funciones

if(estado == HIGH)
{client.print("<input type = 'image'
src='https://i.imgur.com/00i0BdB.jpg'
style='display:block; margin:auto'
width='30%'
onClick=location.href='/LED=ON'>");
abrir();
}
else
{client.print("<input type = 'image'
src='https://i.imgur.com/uQxhFE5.jpg'
style='display:block; margin:auto'
width='30%'
onClick=location.href='/LED=OFF'>");
cerrar();
}

client.println("</html>"); //Cierra código
html
delay(1);
Serial.println("Petición finalizada"); Se
finaliza la petición al cliente
Serial.println("");
}

// Función para abrir ventana
void abrir(){
servo.write(10); //angulo
delay(200);
servo.write(20); //angulo
delay(200);
}

//Función para cerrar ventana
void cerrar(){
servo.write(20); //Angulo
delay(200);
servo.write(10); //Angulo
delay(200);
}

```

Concluyendo con la programación para el módulo, se podrá abrir y cerrar la ventana por medio del sitio web creado. A continuación, se muestra la interfaz del sitio web para el control del módulo wifi.



Figura 28: Interfaz gráfica de sitio web

Cabe resaltar que una vez cargado el programa en el módulo, para ingresar al sitio web se debe de copiar la URL que se muestra en el monitor serie de Arduino al navegador. Como se muestra en la siguiente figura.

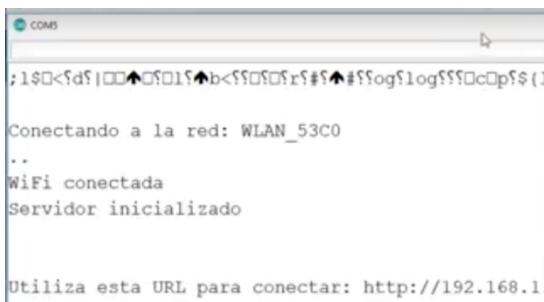


Figura 29: Pantalla de acceso

6.4. MÓDULO DE HUELLA DACTILLAR

Este módulo se utilizará para encender y apagar leds de esta casa domótica. Para llevar a cabo este proceso se deben obtener los siguientes componentes:

- Placa Arduino
- Sensor de huella dactilar
- Cables
- leds

Teniendo todos los componentes listos, primero se procederá a conectar la placa Arduino con el sensor de huellas de la siguiente manera:

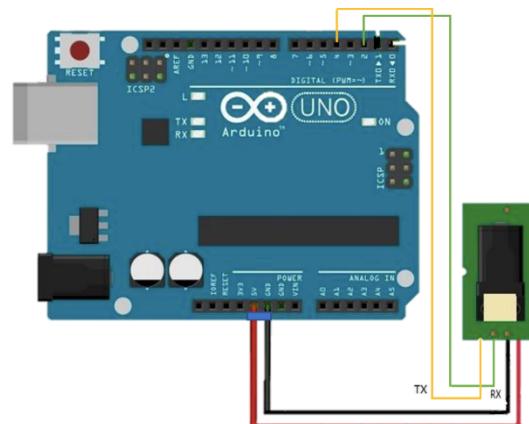


Figura 30: Conexión de placa Arduino con sensor de huella dactilar

Para el funcionamiento correcto de este módulo, se necesitarán dos programaciones; una para que se registren y guarden las huellas digitales en la base de datos y otra para que haga la función deseada de encender y apagar leds.

A continuación, se muestra la programación para la configuración y registro del sensor de huella dactilar.

```
#include <Adafruit_Fingerprint.h>
SoftwareSerial mySerial(2, 4); // Transmisión y recepción del modulo
Adafruit_Fingerprint finger =
Adafruit_Fingerprint(&mySerial);
uint8_t id;

void setup(){
Serial.begin(9600);
while (!Serial); // For Yun/Leo/Micro/Zero/...
}
```

```

delay(100);

Serial.println("\n\nRegistrando el sensor
de huellas dactilares");

// Establece la velocidad de datos para
el puerto serial del sensor
finger.begin(57600); //Por defecto

if (finger.verifyPassword()) {
    Serial.println("Sensor de huellas
detectado!");
} else {
    Serial.println("No se ha podido
detectar el sensor de huellas :(");
    while (1) {
        delay(1);
    }
}

//Proceso de registro
void loop(){
    Serial.println("Sensor listo para registrar
una huella!");

    Serial.println("Porfavor escribe el ID # (0
- 162) donde quieres guardar la
huella...");

    id = readnumber();
    if (id == 0) {// ID #0 no está permitido,
    intentar de nuevo
        return;
    }
    Serial.print("Registrando ID #");
    Serial.println(id);

    while (! getFingerprintEnroll() );
}

uint8_t readnumber(void) {
    uint8_t num = 0;

```

```

while (num == 0) {
    while (! Serial.available());
    num = Serial.parseInt();
}
return num;
}

uint8_t getFingerprintEnroll(void) {

int p = -1;
Serial.print("Esperando una huella
valida para registrar con el ID #");
Serial.println(id);

while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Imagen tomada");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.println(".");
            break;
        case FINGERPRINT_PACKETRECEIVEERR:
            Serial.println("Error al
comunicarse");
            break;
        case FINGERPRINT_IMAGEFAIL:
            Serial.println("Error de imagen");
            break;
        default:
            Serial.println("Error desconocido");
            break;
    }
}

// OK Aceptado!
p = finger.image2Tz(1);

```

```

switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Imagen convertida");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Imagen demasiada
desordenada");
        return p;
    case
FINGERPRINT_PACKETRECIEVEERR:
        Serial.println("Error al comunicarse");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("No se pudo encontrar
características de la huella dactilar");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("No se pudo encontrar
características de la huella dactilar");
        return p;
    default:
        Serial.println("Error desconocido");
        return p;
}

Serial.println("Retirar dedo");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER)
{
    p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Vuelve a introducir el
mismo dedo");
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Imagen tomada");
            break;
        case FINGERPRINT_NOFINGER:
            Serial.print(".");
            break;
        case
FINGERPRINT_PACKETRECIEVEERR:
            Serial.println("Error al
comunicarse");
            break;
        case FINGERPRINT_IMAGEFAIL:
            Serial.println("Error de imagen");
            break;
        default:
            Serial.println("Error desconocido");
            break;
    }
}

// OK aceptado!

p = finger.image2Tz(2);
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Imagen convertida");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Imagen demasiado
desordenada");
        return p;
    case
FINGERPRINT_PACKETRECIEVEERR:
        Serial.println("Error al comunicarse");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("");
        return p;
}

```

```

        case FINGERPRINT_INVALIDIMAGE:
            Serial.println("No se pudo encontrar
caracteristicas de la huella dactilar");
            return p;
        default:
            Serial.println("Error desconocido");
            return p;
    }

    // OK convertido!
    Serial.print("Creando modelo para
#"); Serial.println(id);

    p = finger.createModel();
    if (p == FINGERPRINT_OK) {
        Serial.println("Coincidencia de huella
dactilar!");
    } else if (p == FINGERPRINT_PACKETRECIEVEERR)
    {
        Serial.println("Error al comunicarse");
        return p;
    } else if (p ==
FINGERPRINT_ENROLLMISMATCH) {
        Serial.println("No hay coincidencia de
huella dactilar");
        return p;
    } else {
        Serial.println("Error desconocido");
        return p;
    }

    Serial.print("ID "); Serial.println(id);
    p = finger.storeModel(id);
    if (p == FINGERPRINT_OK) {
        Serial.println("Registrado!");
    } else if (p ==
FINGERPRINT_PACKETRECIEVEERR)
    {
        Serial.println("Error al comunicarse");
        return p;
    } else if (p ==
FINGERPRINT_BADLOCATION) {
        Serial.println("No se pudo almacenar
en esa ubicacion");
        return p;
    } else if (p ==
FINGERPRINT_FLASHERR) {
        Serial.println("Error al escribir en
flash");
        return p;
    } else {
        Serial.println("Error desconocido");
        return p;
    }
}

```

En algunos casos este código para registrar y almacenar las huellas digitales ya viene adjunto en algunos archivos que tengan que ver con funciones específicas con sensores de huellas dactilares. Este programa se cargará en la placa Arduino, siguiendo las siguientes instrucciones:

Paso 1. Se abre el Monitor serie de Arduino para empezar a grabar las huellas y siguen las instrucciones. La primera huella la guardamos en la posición 1 y luego se da entre y siguen las instrucciones.

Si la huella se registró correctamente mostrará el mensaje “Huella dactiles SI coinciden!”, seguido de la posición donde se guardó y el mensaje “Registrado!”

Paso 2. Para guardar más de una huella, el sensor permite hasta 162 huellas, se volverá a teclear ahora el número de la siguiente posición donde se quiere guardar, que en este ejemplo sería la posición 2, se teclea 2 y presione entre y siguen de nuevo las mismas instrucciones hasta grabar todas las huellas

necesarias, siempre indicando una posición diferente para que no se sobrescriba una ya guardada.

El objetivo de este módulo es encender y apagar leds, ya que se tiene la configuración del sensor de huellas dactilares se procederá primeramente a hacer las conexiones para este de la siguiente manera:

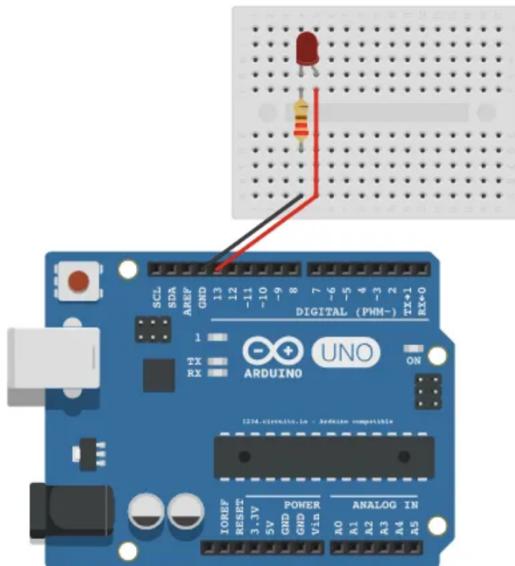


Figura 31: Conexión de placa Arduino con protoboard y led

Algo importante de mencionar es que esta figura es solo la conexión de un solo led y lo que se necesita es conectar mas leds para el objetivo, pero esto es solo la base para entender las conexiones de este módulo. En dado caso se conectará más leds a través de un solo pin y la unión de los leds deberá ser de la siguiente manera:

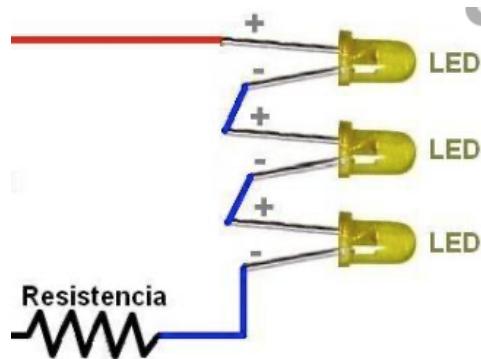


Figura 32: Modelo de conexión para varios leds

A continuación, se muestra la programación para encender y apagar los leds.

```
#include <Adafruit_Fingerprint.h> // Libreria para el Sensor de huella dactilar
SoftwareSerial mySerial(2, 4); // Crear Serial para Sensor Rx, TX del Arduino
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial); // Crea el objeto Finger comunicación pin 2 , 4
void setup()
{
pinMode(13, OUTPUT); //Salida led
Serial.begin(9600);
while
(!Serial); // Yun/Leo/Micro/Zero/...
delay(100);
Serial.println("Sistema de apertura con huella dactilar");
// Establece la velocidad de datos para el puerto serial del sensor
finger.begin(57600); // inicializa comunicación con sensor a 57600 Baudios
delay(5);
if (finger.verifyPassword()) {
Serial.println("Detectado un sensor de huella!");
} else {
Serial.println("No hay comunicación con el sensor de huella");
}
```

```

Serial.println("Revise las conexiones");
while (1) { delay(1); }

finger.getTemplateCount();
Serial.print("El sensor contiene ");
Serial.print(finger.templateCount);
Serial.println(" plantillas");
Serial.println("Esperando por una huella valida...");
}

void loop()
{
  getFingerprintIDez();
  delay(50);          //retardo de 50 milisegundos entre lecturas
}
//Encender led
void encender(){
digitalWrite(13,HIGH);
}

//Apagar led
void apagar(){
digitalWrite(13,LOW);
}

int getFingerprintIDez(){
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK) return -1;

  p = finger.image2Tz();
  if (p != FINGERPRINT_OK) return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK) {
    Serial.print("Mal registro");
    return -1;
  }
// Si hay coincidencias de huella
  Serial.print("ID #");
  Serial.print(finger.fingerID);
  Serial.print(" coincidencia del ");
  Serial.println(finger.confidence);
  if(finger.fingerID==1){
    Serial.print("BIENVENIDO PROFE ***");
  );
  encender();
}

```

```

if(finger.fingerID==4){
  Serial.print("Hasta luego *** ");
  apagar();
}
return finger.fingerID;
}

```

Este programa mostrará las huellas digitales almacenadas que anteriormente se configuraron en el sensor de huellas digitales, si ciertas huellas coinciden con las huellas autorizadas se prenderán los leds y mostrará un mensaje de autorización; en dado caso que no sean las autorizadas no mostrará nada ni hará nada.

En la siguiente figura muestra en monitor serie de las indicaciones y resultados del programa.

Figura 33: Pantalla de funcionamiento del sensor de huellas

7. PRESUPUESTO

ARDUINO
PROTOBOARD
CABLE JUMPERS
LED
RESISTENCIAS



Figura 34: Presupuesto 1

MODULO WIFI



Figura 35: Presupuesto 2

Nuevo | 4797 vendidos
Módulo Wifi Esp8266 Lua Nodemcu Node Mcu
★★★★★ 67 opiniones
MÁS VENDIDO 12º en Arduino
\$ 115
en 12x \$ 11¹
IVA incluido
Ver los medios de pago
Envío a todo el país FULL Conoce los tiempos y las formas de envío. Calcular cuándo llega
Devolución gratis Tiene 30 días desde que lo recibes. Conocer más
Vendido por TECNEU TECHNOLOGY MercadoLider | 27777 ventas
Stock disponible

MODULO BLUETOOTH (HC05)



Nuevo | 1072 vendidos
Modulo Bluetooth Hc05 Arduino
★★★★★ 6 opiniones
MÁS VENDIDO 16º en Arduino
\$ 92
en 12x \$ 9¹
IVA incluido
Ver los medios de pago
Envío a todo el país FULL Conoce los tiempos y las formas de envío. Calcular cuándo llega
Devolución gratis Tiene 30 días desde que lo recibes. Conocer más
Vendido por DIP MECATRONICA MercadoLider | 17068 ventas
Stock disponible

Figura 36: Presupuesto 3

SERVOMOTOR



Nuevo | 10905 vendidos
Servomotor Micro Sg90 Arduino Plc Raspberry
★★★★★ 82 opiniones
\$ 46
en 12x \$ 4¹
IVA incluido
Ver los medios de pago
Envío a todo el país FULL Conoce los tiempos y las formas de envío. Calcular cuándo llega
Devolución gratis Tiene 30 días desde que lo recibes. Conocer más
Vendido por TECNEU TECHNOLOGY MercadoLider | 27777 ventas
Stock disponible
Cantidad: 1 unidad (427 disponibles)

Figura 37: Presupuesto 4

SENSOR DE HUELLA



Nuevo | 64 vendidos
Lector Sensor Huella Dactilar Digital As608 Arduino Plc
★★★★★ 12 opiniones
\$ 366
en 12x \$ 36¹
IVA incluido
Ver los medios de pago
Envío gratis a todo el país FULL Conoce los tiempos y las formas de envío. Calcular cuándo llega
Devolución gratis Tiene 30 días desde que lo recibes. Conocer más
Vendido por TECNEU TECHNOLOGY MercadoLider | 27791 ventas
Stock disponible

Figura 38: Presupuesto 5

TOTAL, DEL PRESUPUESTO: \$880
Aproximadamente

8. RESULTADOS

El funcionamiento de cada uno de los módulos fue el correcto, cumpliendo con lo esperado. A pesar de todo, no significa que no hubo complicaciones durante el desarrollo del proyecto, los problemas más comunes que se presentaron se muestran en dos partes, la conexión de los componentes y la programación de los módulos donde:

En las conexiones de los componentes, los voltajes tienen que ser exactos ya que algunos componentes necesitan más ó menos voltaje para su correcto funcionamiento, también se recomienda usar los respectivos componentes mencionados para el desarrollo del proyecto y usar los mismos pines de las conexiones por que puede llegar a cambiar el funcionamiento del proyecto ya sea por ser distintos componentes o no hacer bien las conexiones.

En las programaciones de cada módulo puede variar el funcionamiento de cada uno, ya que en algunas versiones del IDE de trabajo y de los componentes no siempre es la misma sintaxis para programar.

10. CONCLUSIONES

Al hacer el diseño de la maqueta nos basamos en un pequeño departamento donde fuera útil al utilizar las puertas con un bluetooth, las ventanas con wifi y las lámparas con un lector de huellas haciendo más sencillo el funcionamiento de la casa y así aprendiendo el funcionamiento el tipo de conexión que se necesita para poder hacer estas funciones. Al observar el diseño de la aplicación que tendría el control sobre la casa y cada material que debimos utilizar para dicha conexión de los artículos ya que cada accesorio forma parte importante del funcionamiento si llegamos a implementar estas funciones en la vida junto con sus beneficios para las personas que

llegaran a tenerlo dependiendo de cada persona podrían variar como quiere que funcione cada accesorio.

11. REFERENCIAS

- [A. R. H. Balibrea, «“Tecnología domótica 1 para el control de una vivienda,”] Upct.es.,» [En línea]. Available: <https://repository.upct.es/bitstream/handle/10317/2793/pfc4381.pdf>. [Último acceso: 02 Mayo 2021].
- [Upv.es., [En línea]. Available: 2 <https://riunet.upv.es/bitstream/handle/10251/75797/SANCLEMENTE%20-%20DISE%C3%91O%20DE%20CASA%20DOM%C3%93TICA%20CONTROLADA%20POR%20ARDUINO.pdf>. [Último acceso: 02 Mayo 2021].
- [Unican.es., [En línea]. Available: 3 <https://repository.unican.es/xmlui/bitstream/handle/10902/3641/358731.pdf?sequence=1&isAllowed=y..> [Último acceso: 02 Mayo 201].
- [«“ESCUELA POLITÉCNICA DE EDIFICACIÓN 4 DE BARCELONA,” Upc.edu.,» [En línea].] Available: https://upcommons.upc.edu/bitstream/handle/2117/118851/Mem%C3%B2ria_VillaverdeMarta.pdf?sequence=1&isAllowed=y. [Último acceso: 02 Mayo 2021].
- [C. a. p. c. a. c. m. b. (. y. pic), «Youtube,» 5 02 Abril 2015. [En línea]. Available:] https://www.youtube.com/watch?v=sFb_TXKtJlo. [Último acceso: 02 Mayo 2021].

[M. d. u. c. D. (Automatizada), «Youtube,»
6 15 Agosto 2013. [En línea]. Available:
] <https://www.youtube.com/watch?v=V90xMG6Lixk>. [Último acceso: 02 Mayo 201].

[L. h. b. Arduino, « Makerelectronico.com,»
7 11 Abril 2020. [En línea]. Available:
] [https://www.makerelectronico.com/lecto-r-huella-biometrica-arduino/..](https://www.makerelectronico.com/lecto-r-huella-biometrica-arduino/) [Último
acceso: 02 Mayo 2021].

[Miguel, «“Práctica 1. Encender y apagar
8 un LED con Arduino,» Mecabot-ula.org, 14
] Enero 2017. [En línea]. Available:
<http://mecabot-ula.org/tutoriales/arduino/practica1-encender-y-apagar-con-arduino/>. [Último
acceso: 02 Mayo 2021].

[9] *NodeMCU - Control de led por internet.* 2018.

[10] *Servo motor control with Blynk over WIFI.* 2018.

[11] *Arduino: Cómo configurar un módulo bluetooth maestro (HC-05) | TechKrowd.* 2016.

[12] *Arduino: Cómo configurar un módulo bluetooth HC-06 (Slave) | TechKrowd.* 2016.

[13] *Arduino: Motor DC y transistor NPN (control de velocidad) | TechKrowd.* 2016.