



Instituto Tecnológico José Mario Molina Pasquel y Henríquez
Campus Chapala

Programación para Sistemas de Embebidos

Unidad 4. Comunicaciones

Control de alarma

8°M

González Osuna Jennifer Cassandra

Junio de 2022.

Contenido

Objetivo..... 3

Justificación..... 3

Marco teórico..... 4

 Componentes..... 4

Desarrollo..... 10

 Materiales..... 10

 Proceso..... 10

Resultados..... 20

 Diseños..... 20

 Circuitos..... 21

 Códigos..... 22

 Prototipo..... 44

 Análisis de costos..... 44

Conclusiones..... 45

Bibliografía..... 45

Objetivo.

- a. Objetivo del proyecto.
Desarrollar un sistema de alarma que permita la seguridad de un inmueble mediante la información de sensores magnéticos, infrarrojos y variables recibidas de la interfaz web.
- b. Objetivo académico.
Comprende el funcionamiento de los periféricos de comunicaciones de un microcontrolador.

Justificación.

En la actualidad, nuestro país presenta un índice impresionante de delincuencia donde los policías no tienen el poder de proteger a todos los domicilios de cada lugar, donde el obtener un sistema de alarma se vuelve cada vez algo más necesario para estar protegidos de la delincuencia que se pueda presentar en nuestros hogares.

Aquellos que están al tanto de todas las capacidades de un sistema de alarma, el costo del mismo puede ser un problema a la hora de decidir si instalar el equipamiento extra o no. Si bien el costo es una preocupación válida, deberá tomar en consideración los riesgos que puede llegar a correr.

Sin embargo, proteger a su familia y a sus posesiones es importante para la mayoría de las personas. Un sistema de seguridad casera tal vez disminuya el riesgo de tener a un ladrón entrando en su hogar. Los estudios muestran que los criminales evitarán las aventuras riesgosas y generalmente se alejarán de las casas donde haya algún tipo de mecanismo de seguridad. La mayoría de los ladrones viajan por el barrio buscando las casas que sean más fáciles de asaltar, y las eligen antes que aquellas que tengan sistemas de seguridad que puedan poner su libertad en riesgo. Hasta si llegan a ingresar en su hogar seguramente se irán rápidamente (tomando menos cosas y minimizando el daño) cuando suene una alarma.

Utilizar un sistema de seguridad casero podrá proveerle comodidad para la familia. Saber que usted, su familia y sus posesiones están a salvo de los robos es invaluable. Los sistemas de seguridad son baratos y proveen un valioso sentido de la seguridad, razón por la cual se justifica este proyecto.

Marco teórico.

Componentes.

ESP32

El ESP32 cuenta con un total de **34 pines digitales**. Estos pines, al igual que en cualquier **placa Arduino**, permiten agregar **LEDs, botones, zumbadores**, y un largo etc. a nuestros proyectos.

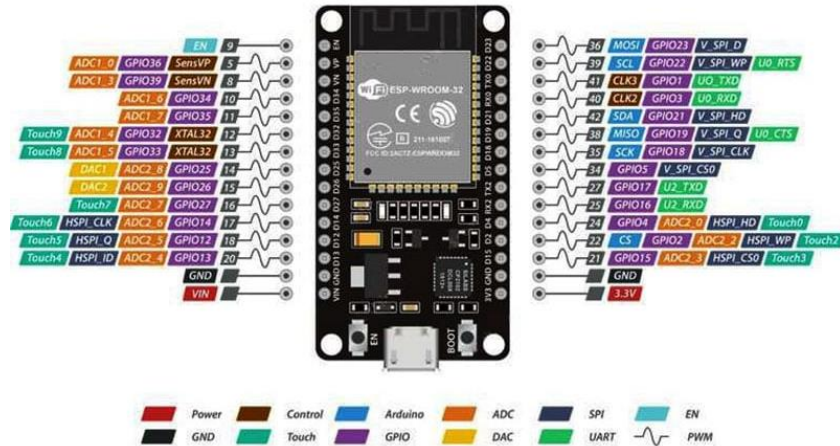


Figura 1.0 ESP32 Pinout (DescubreArduino, 2018).

La mayoría de estos pines admiten el uso de pull-up, pull-down internos y también el estado de alta impedancia. Esto los hace ideales para conectar botones y teclados matriciales, así como para aplicar técnicas de control de LEDs como la conocida Charlieplexin.

Algunos de los pines también pueden ser utilizados para interactuar con sensores analógicos, es decir, como si fueran los pines analógicos de una placa Arduino. Para esto el ESP32 cuenta con un conversor analógico digital de 12-bits y 18 canales, es decir, que puedes tomar lecturas de hasta 18 sensores analógicos.

Esto te permite desarrollar aplicaciones conectadas muy compactas, incluso cuando se empleen varios sensores analógicos. Para los ESP32 que no poseen memoria empotrada o simplemente cuando la memoria es insuficiente para tu aplicación, es posible adicionar más memoria de forma externa:

- Se pueden agregar hasta 16 MiB de memoria flash externa. De esta forma puedes desarrollar aplicaciones más complejas.

- También admite, hasta 8 MiB de memoria SRAM externa. Por lo tanto, es difícil que te encuentres limitado en memoria al implementar una aplicación utilizando esta plataforma.

Sensor infrarrojo

Está conformado por tres componentes básicos que mencionaremos a continuación

- Sensores pasivos: Tienen la finalidad de medir las radiaciones de los cuerpos u objetos y están integrado por un fototransistor.
- Sensores activos: Por lo general conforman un mismo circuito y están constituido por un emisor y un receptor.



Figura 1.1. Sensor Infrarrojo (prometec, 2018).

Características generales del sensor infrarrojo

- Su composición es aplicable para la medición climática
- En algunos casos colabora con el ahorro de energía
- Los sensores infrarrojos emiten luz que no es captada por el ojo humano
- Su estructura de base es utilizada en la fabricación de robots
- Comprenden diversas configuraciones, lo podemos encontrar de una, dos y cuatro píxeles de material piroeléctrica.

Sensor Magnético

Sensor magnético para puertas y ventanas. El mecanismo del sensor MC-38 es normalmente cerrado (NC), por lo cual, manda un 1 lógico cuando ambas partes del sensor están en contacto y 0 cuando están separadas

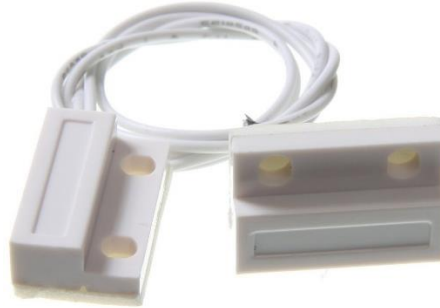


Figura 1.2. Sensor magnético

Keypad



Figura 1.3 Keypad (Luis Llamas, 2017)

Un teclado matricial agrupa los pulsadores en filas y columnas formando una matriz, lo que permite emplear **un número menor de conductores** para determinar la pulsación de las teclas.

La siguiente imagen muestra, a modo de ejemplo, una disposición rectangular de 4x4, aunque el funcionamiento es análogo en otras disposiciones. Al detectar la pulsación en la columna X y la fila Y, sabremos que se ha pulsado la tecla (X,Y).

Internamente la disposición de los pulsadores es la siguiente, que es muy similar a la disposición que vimos al tratar sobre display LED matriciales.

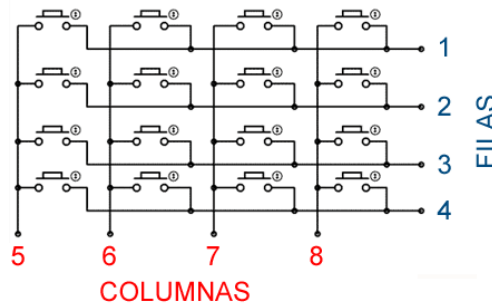


Figura 1.4 Columna y filas de un keypad (Luis llamas, 2017).

LED

Un LED (acrónimo del concepto inglés light-emitting diode) es un diodo emisor de luz. En su interior hay un semiconductor que, al ser atravesado por una tensión continua, emite luz, lo que se conoce como electroluminiscencia. Existen distintos tipos de led en función de las tecnologías usadas para su fabricación y montaje sobre circuitos electrónicos.



Figura 1.5. LED (Tecnología, 2018).

2N2222A

Es un transistor uno de los dispositivos electrónicos más utilizados. Es un transistor BJT del tipo NPN. Un transistor tiene principalmente dos propósitos, el actuar como un amplificador o como un switch digital. El 2N2222A está construido con un material semiconductor dopado de 3 capas. Estos dos materiales se les conoce como semiconductor tipo N y tipo P. Dependiendo del semiconductor utilizado este puede ser silicio o germanio.

2N2222

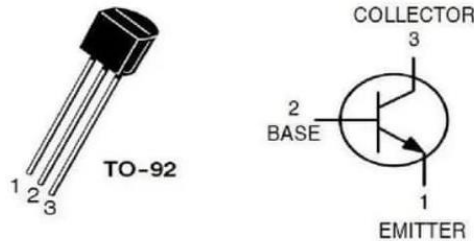


Figura 1.6. Transistor 2N2222A (HetPRO, 2017).

LM7805

Es un regulador de voltaje lineal positivo con voltaje de salida de 5V DC y corriente de salida máxima 1.5A. Tiene un encapsulado de TO-220 de tres pines.

Sirve para suministrar voltaje constante de 5V a partir de un voltaje de entrada mayor que puede ser mínimo 7V y máximo 35V, recomendable ingresar voltaje de entrada $\leq 18V$ DC.

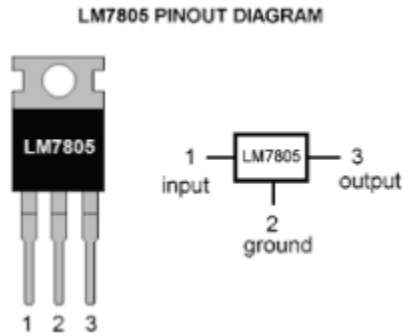


Figura 1.3 Tip 41C (Electro Store, 2015).

1. Input: Voltaje de entrada **INPUT**
2. Ground: Tierra **GND**
3. Output: Voltaje de salida **OUTPUT**

Relevador

Es un interruptor que puede ser controlador eléctricamente. Este dispositivo también puede entenderse como un controlador electro-mecánico. Fue inventado por el científico estadounidense Joseph Henry quien descubrió el fenómeno electromagnético de auto-inductancia e inductancia mutua. Este principio le permitió crear un tipo de **electroimán** que al activarse puede controlar a un interruptor, este es el principio del relevador.

Los elementos principales de un relevador son: (1) bobina de cobre, (2) **núcleo de hierro**, (3,4) balancín, (5,6 y 7) contactos. Cuando una corriente eléctrica se hace pasar por la bobina esta induce un campo magnético que permite que el núcleo de hierro actúe como un electro-imán lo que hace que el elemento enumerado como (3) y (2) se conecten haciendo subir o bajar el balancín (4). Este efecto permite abrir o cerrar el paso de corriente por los contactos 5, 6 y 7. Siendo el contacto (5) llamado común, el (6) el contacto normalmente cerrado y el (7) el contacto normalmente abierto.

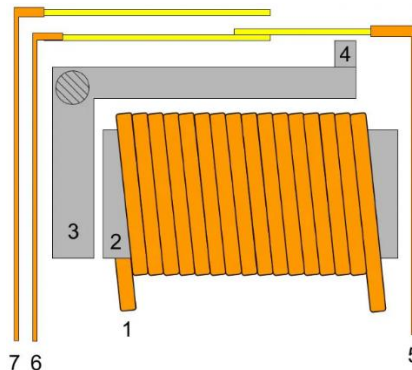


Figura 1. Elementos físicos de un relevador también llamado relé. (hetPRO, 2016).

Resistencias

Resistencia Una resistencia es un elemento pasivo de un circuito eléctrico. Generalmente, una resistencia cualquiera provoca una restricción al paso de la corriente, limitándola y, específicamente, regulándola. Claro está, una resistencia combinada con otros elementos como condensadores (capacitores), bobinas, diodos... forman complejos circuitos con funciones concretas. Una resistencia se puede definir como cualquier medio material que limita el paso de la corriente eléctrica.

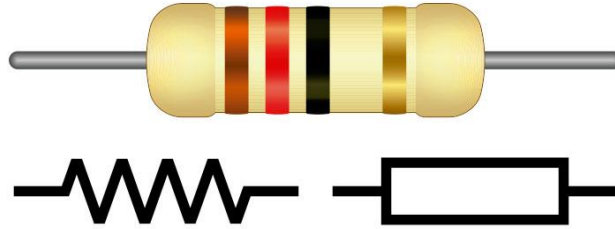


Figura 1.4. Tipos de resistencia según su potencia (Lugo, 2017)

Alarma / Sirena



Figura 1.7 Sirena

Sirena pieza cerámico (sin bobina ni imán), cuadrada, de un tono constante a 105 dB y alimentación de 12 Vcc. Mide 4 x 4,4 x 5,8 cm y está fabricada en plástico ABS color negro. Por su alto volumen y agudo sonido, es muy útil en sistemas de alarmas, ya que atrae la atención inmediatamente, ahuyentando a cualquier intruso.

Desarrollo.

Materiales.

Proceso.

Primeramente, se comenzaron a elegir los sensores que se iban a elegir, una vez obtenidos se comenzó a hacer pruebas del circuito, activando o leyendo los sensores magnéticos con un pin digital y elegir pines adc para los sensores infrarrojos. Y para el keypad se comprobó que las teclas presionadas entraran en el código. Ya activando todo junto con la lcd que se comprobó que ya imprimía datos, y se comenzó con el código. En la figura 2.0 se muestra las pruebas realizadas en protoboard.



Figura 2.0. Pruebas en protoboard.

Se comprobo tambien que se mandara el pulso alto en el pin digital asignado para activar la alarma mediante un relevador.

Al verificar que las conexiones estaban correctas pues la prueba base tanto del circuito como del código funcionaba, se comenzó a diseñar la pcb que se utilizaría en el prototipo, se muestra terminado en la figura 2.1 y en la figura 2.2 se muestra ya con los componentes colocados y ensamblados.

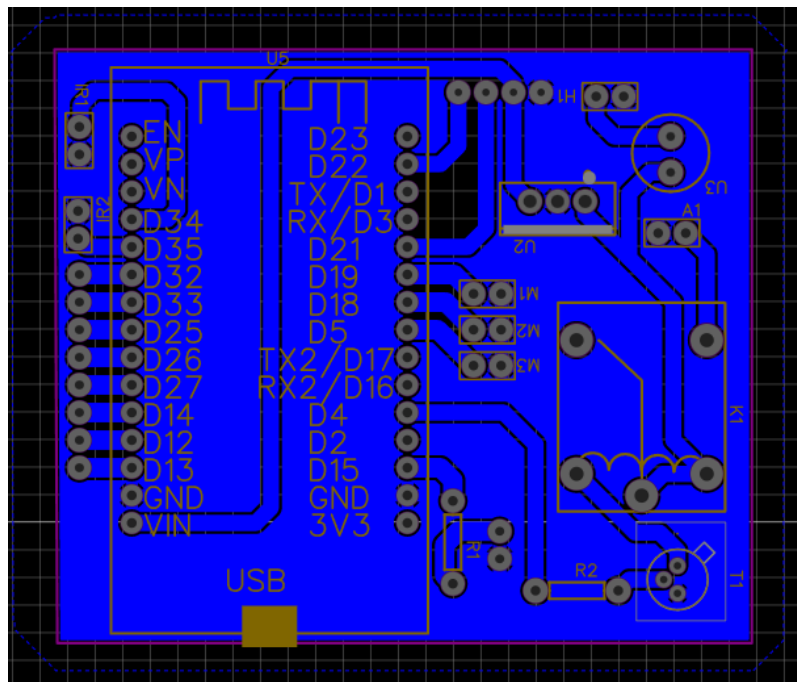


Figura 21. PCB del sistema de alarma.

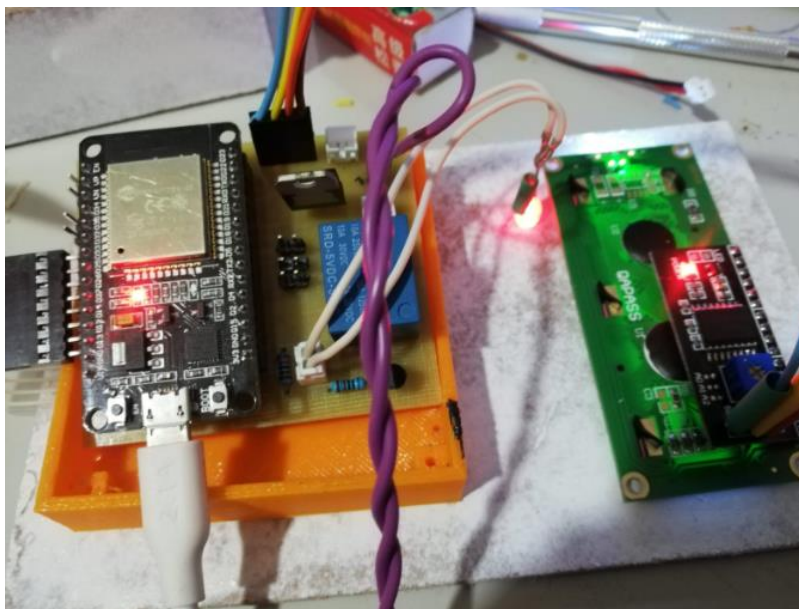


Figura 22 PCB ensamblada.

Se continua con el diseño de la tapa que tendrá la cajita de la alarma, se hace el diseño en Inventor Autodesk, se toman las dimensiones realizadas en el modelo para hacer el corte de las piezas en MDF. Este diseño se muestra en la figura 2.3.

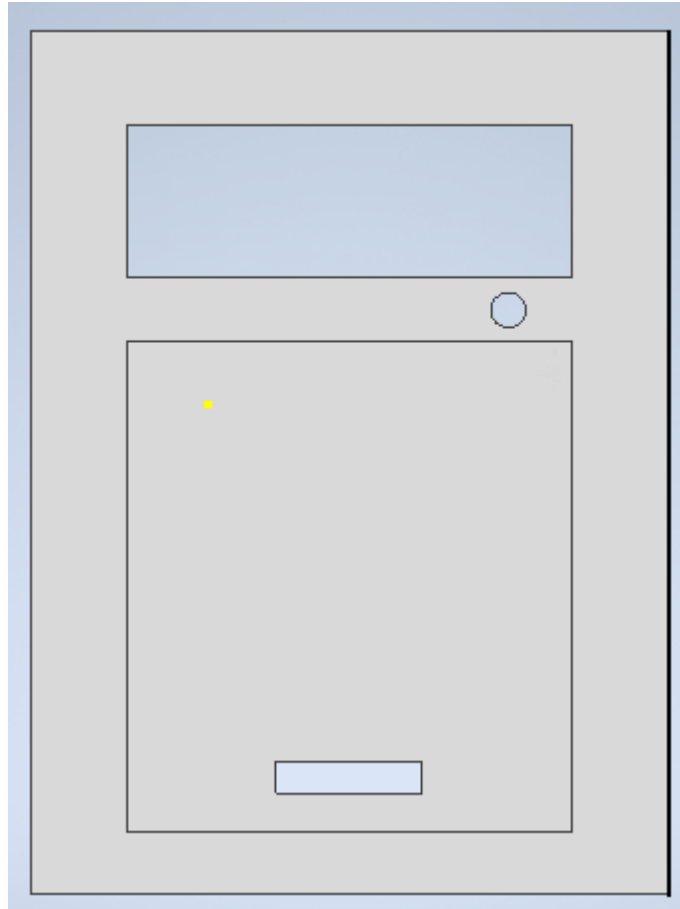


Figura 2.3 Modelo de la tapa del sistema de alarma (Autodesk Inventor, 2022).

Uno de los pasos más importantes fue instalar los sensores en la maqueta asignada para hacer todos los sistemas de domótica, en mi caso se me asigno la parte de hacer las ventanas. Por lo que tome medidas de los orificios hechos en la maqueta para hacer mis ventanas y crear el sistema que haría que las ventanas se recorrieran para poder abrir y cerrar. Estos diseños se muestran en la figura 2.4 a la figura 2.9.

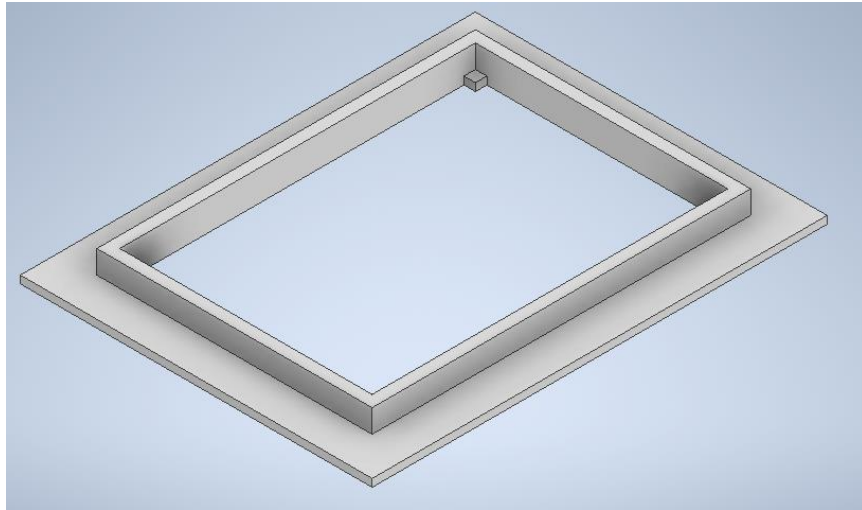


Figura 2.4 Modelo de marco de ventana 1 (Autodesk Inventor, 2022).

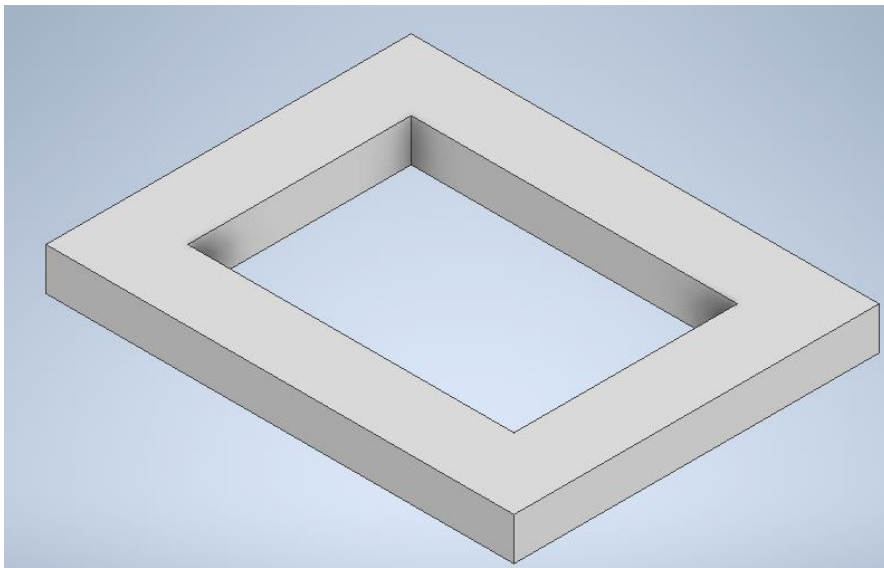


Figura 2.5 Modelo de ventana 1 parte derecha (Autodesk Inventor, 2022).

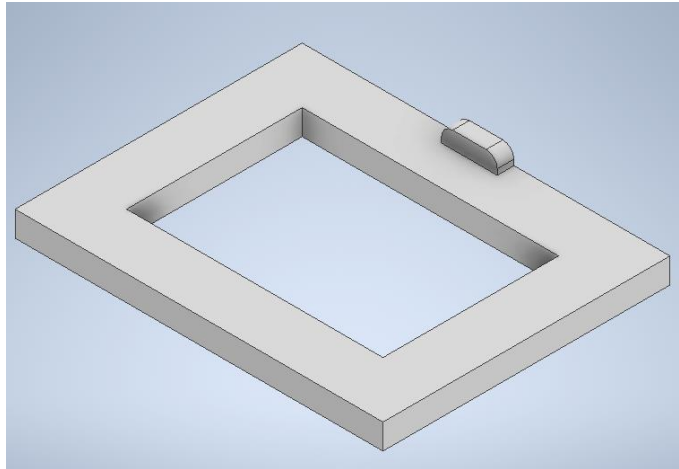


Figura 2.6 Modelo de ventana 1 parte izquierda (Autodesk Inventor, 2022).

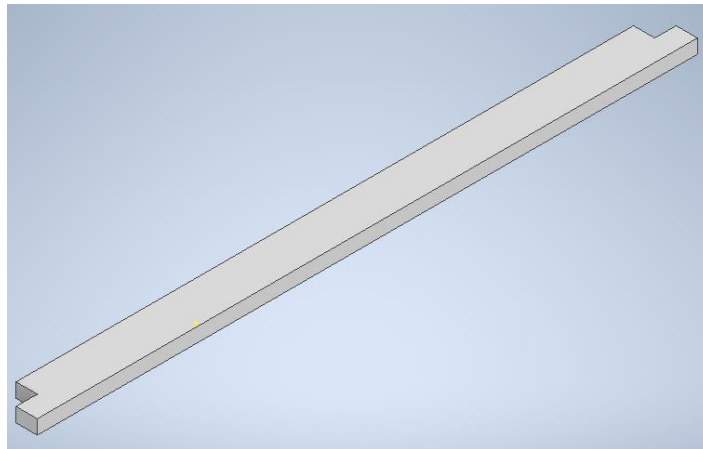


Figura 2.6 Modelo del sujetador de ventana 1 (Autodesk Inventor, 2022).

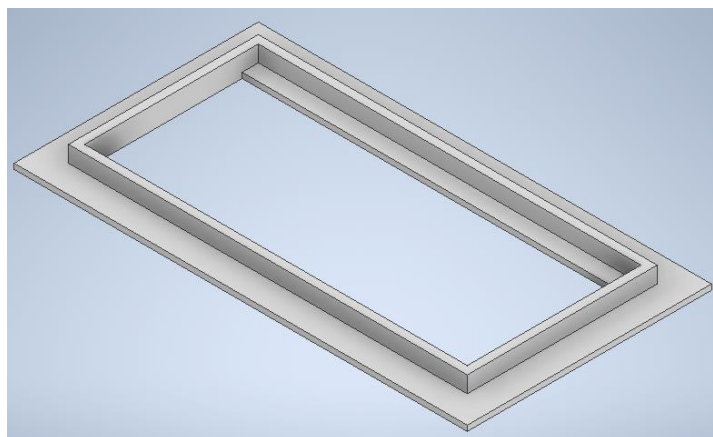


Figura 2.7 Modelo de marco de ventana 2 (Autodesk Inventor, 2022).

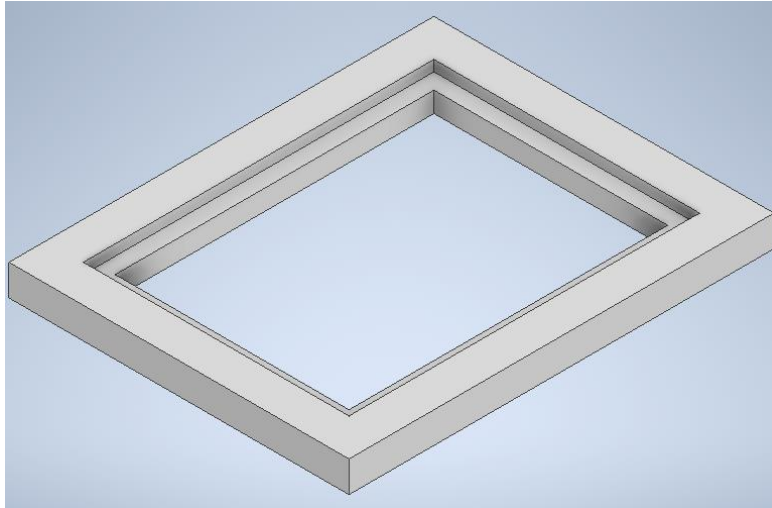


Figura 2.8 Modelo de ventana 2 parte izquierda (Autodesk Inventor, 2022).

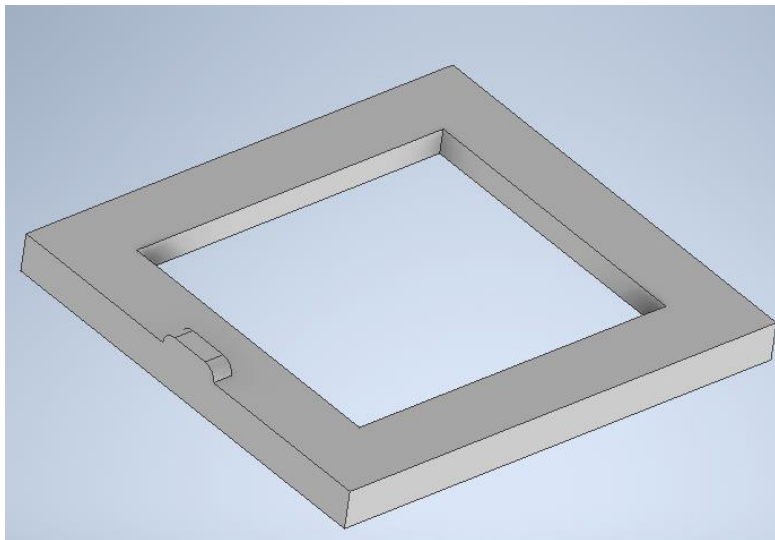


Figura 2.9 Modelo de ventana 2 parte izquierda (Autodesk Inventor, 2022).

Se imprimieron en 3D con filamento PLA de color amarillo de 1.75mm, una vez impresas estas piezas se instalaron en la maqueta en los lugares asignados y posteriormente se hicieron las conexiones de los sensores al sistema de alarma, pero esta vez ya con los sensores en sus lugares asignados (puertas y ventanas). El proceso de instalación se muestra en la figura 3.0 a la figura



Figura 3.0 Instalación del sensor magnético en la puerta principal (Autodesk Inventor, 2022).



Figura 3.1 Instalación del sensor magnético e infrarrojo en ventana 1 (Autodesk Inventor, 2022).



Figura 3.2. Conexión de sensores magnéticos e infrarrojos al sistema de alarma (Autodesk Inventor, 2022).



Figura 3.3 Instalación del sensor magnético e infrarrojo en ventana 1 (Autodesk Inventor, 2022).

Por último, como proceso, se comprobó el funcionamiento del sistema de alarma. A continuación, se muestra el orden de su funcionamiento.

- Figura 3.4 muestra la pantalla principal o MENÚ, donde el usuario puede activar la alarma presionando el botón 'A', donde activará la alarma, lista para mandar una alerta a la interfaz de usuario en caso de que una ventana o puerta sea abierta mientras la alarma está activada. O presionar el botón 'B' para que el usuario pueda cambiar la contraseña asignada para que se desactive la alarma.
- Figura 3.5, se muestra un pequeño contador de 10 segundos para activar la alarma
- Figura 3.6. Una vez activada la alarma si se detecta alguna puerta o ventana que ha sido abierta o un vidrio roto (para esto se utilizan los sensores infrarrojos) salta la sirena, empieza a sonar y no se desactivará hasta que el usuario la desactive ingresando la contraseña correcta, de no ser así mostrará error y seguirá sonando.



Figura 3.4 Pantalla principal del sistema de alarma.



Figura 3.5 Contador del sistema de alarma al iniciar la alarma.



Figura 3.6 Sirena encendida y contraseña requerida para desactivar alarma.

Resultados.

Diseños.

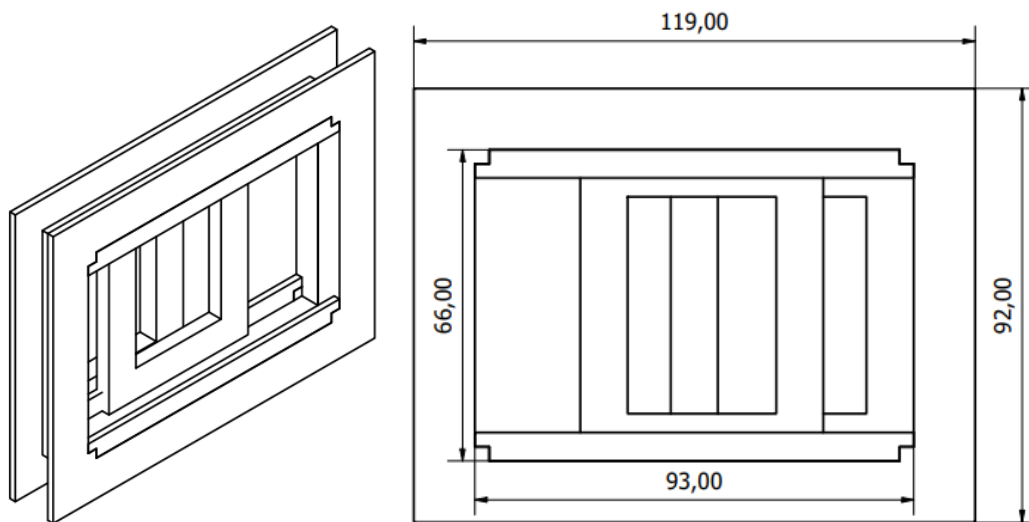


Figura 3.7 Plano y dimensiones de la ventana 1(Inventor, 2022).

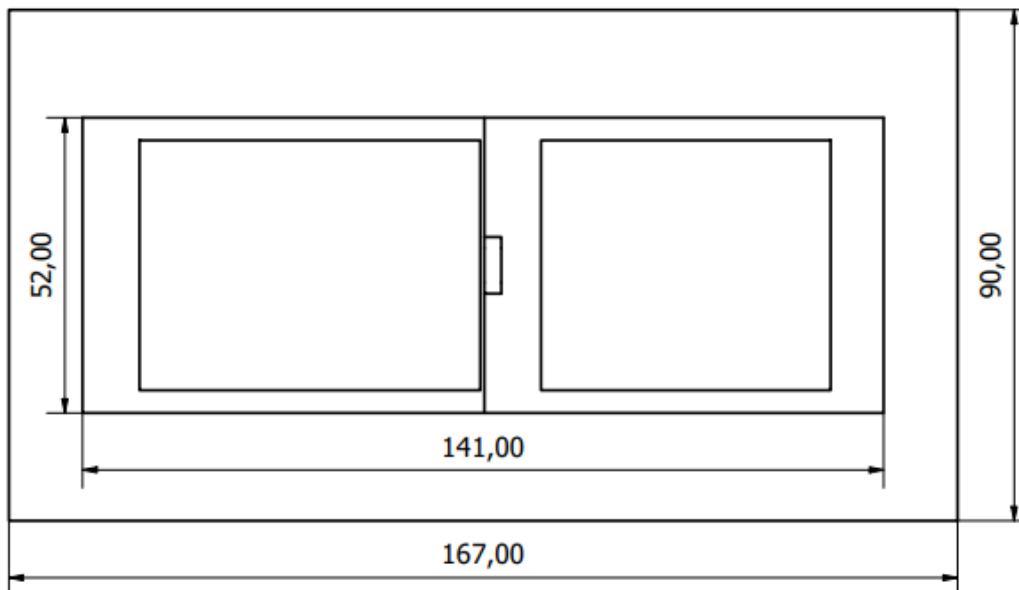


Figura 3.8 Dimensiones de la ventana 2 (Inventor, 2022).

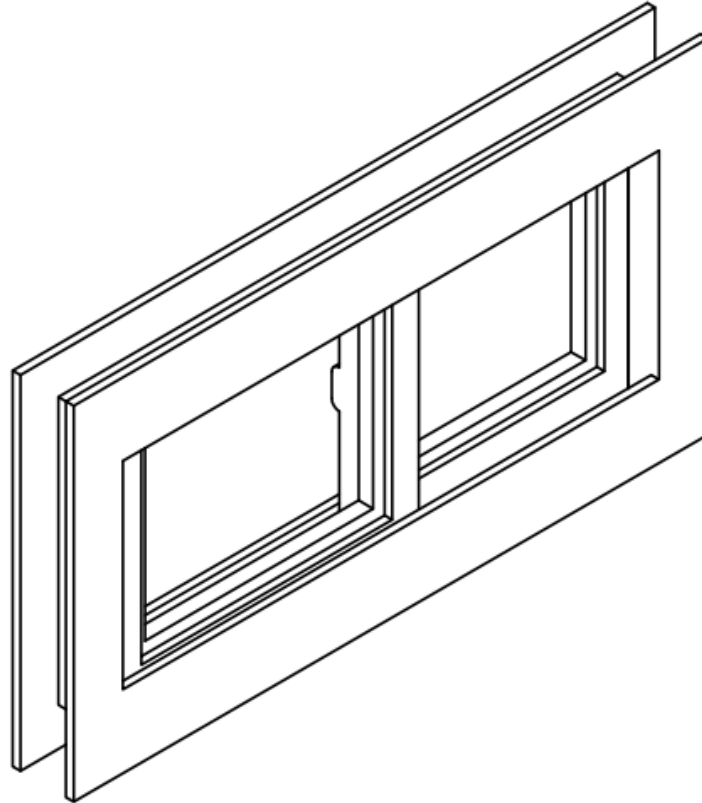


Figura 3.9 Plano de la ventana 2 (Inventor, 2022).

Circuitos.

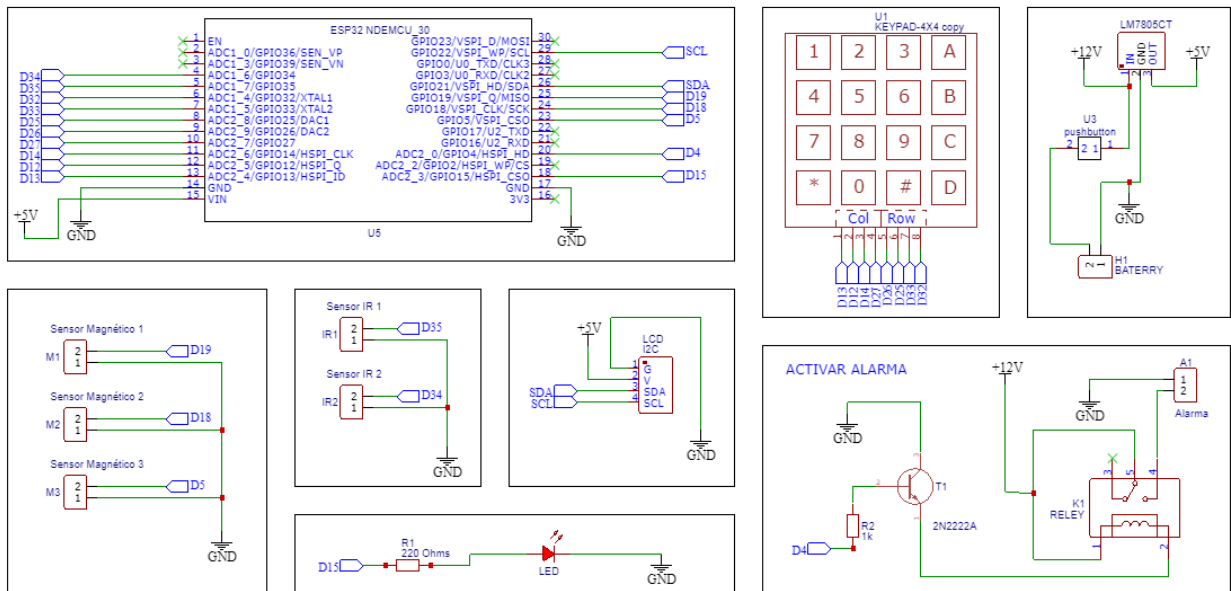


Figura 4.0. Circuito de conexión del sistema de alarma (EasyEDA, 2022).

Códigos.

A continuación, se muestra el código finalizado, ya con los datos que se envían al servidor y los que se reciben de este.

```
#include <EEPROM.h>

#include <LiquidCrystal_I2C.h>

#include <Keypad.h>

#include <WiFi.h>

#include <HTTPClient.h> // nos permite hacer peticiones http


// static const char *ssid = "RDC-SOON";
// static const char *passwordIoT = "2030SAXL25";


const char *ssid = "RED JALISCO PRIVADA";
const char *passwordIoT = "RedJalisco";


// Url's para hacer las peticiones
String example = "http://10.149.73.213/";
// const char* example2 = "http://example.com/sensor1";

String recibir = "http://10.149.73.213/datos.php?action=5&ncontrol=ch18031192&label=red"; // Dato que voy a recibir
String sendData1 = "http://10.149.73.213/datos.php?action=3&ncontrol=ch18031192&label="; // Enviar datos
String sendData2 = "http://10.149.73.213/datos.php?action=3&ncontrol=ch18031192&label="; // Enviar datos
String sendData3 = "http://10.149.73.213/datos.php?action=3&ncontrol=ch18031192&label="; // Enviar datos
String sendData4 = "http://10.149.73.213/datos.php?action=3&ncontrol=ch18031192&label="; // Enviar datos
String sendData5 = "http://10.149.73.213/datos.php?action=3&ncontrol=ch18031192&label="; // Enviar datos
String sendData6 = "http://10.149.73.213/datos.php?action=3&ncontrol=ch18031192&label="; // Enviar datos
String sendData7 = "http://10.149.73.213/datos.php?action=3&ncontrol=ch18031192&label="; // Enviar datos
String sendData8 = "http://10.149.73.213/datos.php?action=3&ncontrol=ch18031192&label=";

// const char* usingIP = "http://192.168.4.1/";


String answer; // estado de alarma
String answer2; // estado del sensor M2
```

```
String answer3; // Estado de sensor IR1
String answer4; // Estado de sensor M3
String answer5; // Estado de sensor IR2
String answer6; // Estado de sensor M1
String answer7; // Estado de sensor M4
String answer8; // Estado de sensor M4

const int requestInterval = 7000; // 7s

LiquidCrystal_I2C lcd(0x27, 16, 2);

// Definir pines
#define buzzer 4

#define sensorMagnetico1 19
#define sensorMagnetico2 18
#define sensorMagnetico3 5
#define sensorMagnetico4 36

#define sensorInfrarrojo1 34
#define sensorInfrarrojo2 35
#define led 15

// Definir valores del contador cuando la puerta se abra
int a = 15; // 30 900
volatile unsigned long tiempo2 = 0;
volatile unsigned long tiempo1 = 0;
volatile unsigned long tiempoMillis = 0;
unsigned long s = 0;
unsigned long se;
int contador = 0;
int flag = 0;
volatile unsigned long timer = 0;
```

```
uint8_t recibido = 0, obtenido = 0;
```

```
bool recibi = false;
```

```
//-----> variables puestas por sato <-----
```

```
bool posSendData = true;
```

```
//-----
```

```
// Definir simbolos y botones del keypad
```

```
const byte ROWS = 4; // 4 filas
```

```
const byte COLS = 4; // 4 columnas
```

```
char keypressed;
```

```
char keyMap[ROWS][COLS] = {
```

```
    {'1', '2', '3', 'A'},
```

```
    {'4', '5', '6', 'B'},
```

```
    {'7', '8', '9', 'C'},
```

```
    {'*', '0', '#', 'D'}
```

```
};
```

```
byte rowPins[ROWS] = {13, 12, 14, 27};
```

```
byte colPins[COLS] = {26, 25, 33, 32};
```

```
Keypad myKeypad = Keypad(makeKeymap(keyMap), rowPins, colPins, ROWS, COLS);
```

```
// lectura de los sensores magneticos
```

```
uint8_t sensorM1 = 0; // Puerta principal
```

```
uint8_t sensorM2 = 0; // ventana 1 (abierto/cerrada)
```

```
uint8_t sensorM3 = 0; // ventana 2 (abierto/cerrada)
```

```
uint8_t sensorM4 = 0; // Puerta cochera
```

```
// Lectura de los sensores infrarrojos
```

```
float sensorIR1 = 0; // Ventana 1 Rota
```

```
float sensorIR2 = 0; // Ventana 2 Rota
```

```
// Alarma - Sirena
```



```
int Alarm = 0; // Estado dela sirena
```

```
// definir variables para alarma
```

```
long duration;
```

```
int distance, initialDistance, currentDistance, i;
```

```
int screenOffMsg = 0;
```

```
String password = "2996";
```

```
String tempPassword;
```

```
boolean activated = 0; // True / False; // estado de la alarma
```

```
boolean isActivated;
```

```
boolean alarmActivated = false;
```

```
boolean enteredPassword; // estado de la contraseña ingresada para detener la ala
```

```
boolean passChangeMode = false;
```

```
boolean passChanged = false;
```

```
// ESTADO DE LA PUERTA Y ALARMA
```

```
bool est = false;
```

```
bool puerta = false;
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600); // Se inicia el serial
```

```
  lcd.init();
```

```
  lcd.backlight();
```

```
  // nos conectamos a la red
```

```
  WiFi.begin(ssid, passwordIoT);
```

```
  Serial.println("Connecting");
```

```
  while (WiFi.status() != WL_CONNECTED)
```

```
  {
```

```
    delay(500);
```

```
    Serial.print(".");
```

```
  }
```

```
Serial.println("");

Serial.print("Conectado a la red con la IP: ");

Serial.println(WiFi.localIP());

Serial.println();

tiempo1 = millis();

/* //Se inicia correctamente la pantalla lcd

  lcd.setCursor(4, 0);

  lcd.print("LCD Test");

*/

pinMode(sensorMagnetico1, INPUT_PULLUP);
pinMode(sensorMagnetico2, INPUT_PULLUP);
pinMode(sensorMagnetico3, INPUT_PULLUP);
// pinMode(sensorMagnetico4, INPUT_PULLUP);
pinMode(sensorInfrarrojo1, INPUT_PULLUP);
pinMode(sensorInfrarrojo2, INPUT_PULLUP);
pinMode(led, OUTPUT);
pinMode(buzzer, OUTPUT);
}

void loop()
{
  if (WiFi.status() == WL_CONNECTED)
  {
    sendData = sendData + "activated&valor=" + (String)activated;
    answer = getRequest(sendData);
    tiempoMillis = millis();
    if ((tiempoMillis - timered) >= 1000)
    {
      answer = getRequest(recibir); // recibir los datos de emmanuel
      Serial.print("\n\nRespuesta de duckduckgo.com: ");
      Serial.print(answer); // puro numero para manipularlo
```

```
Serial.print(" ");  
recibido = answer.toInt();  
Serial.println(recibido);  
timered = tiempoMillis;  
}  
  
if (recibido == 1 && recibi == true)  
{  
    obtenido = recibido;  
    recibi = false;  
}  
  
if (recibido == 0 && recibi == false)  
{  
    obtenido = recibido;  
    recibi = true;  
}  
  
Serial.print("Alarma = ");  
Serial.print(activated);  
  
// PUERTA PRINCIPAL  
sensorM1 = digitalRead(sensorMagnetico1);  
Serial.print(" Puerta =");  
Serial.print(sensorM1);  
  
// VENTANA 1 ---- ABIERTA / CERRADA  
sensorM2 = digitalRead(sensorMagnetico2);  
Serial.print(" Ventana 1 =");  
Serial.print(sensorM2);  
  
// VENTANA 1 ROTA ---- INTRUSO  
sensorIR1 = analogRead(sensorInfrarrojo1);  
Serial.print(" Ventana1 Rota =");
```

```
Serial.print(sensorIR1);

// VENTANA 2 ---- ABIERTA / CERRADA
sensorM3 = digitalRead(sensorMagnetico3);
Serial.print(" Ventana 2 =");
Serial.print(sensorM3);
delay(500);

// VENTANA 2 ROTA ---- INTRUSO
sensorIR2 = analogRead(sensorInfrarrojo2);
Serial.print(" Ventana2 Rota =");
Serial.print(sensorIR2);

// PUERTA COCHERA
sensorM4 = digitalRead(sensorMagnetico4);
Serial.print(" Puerta de la cochera ");
Serial.println(sensorM4);
sendDataWiFi();

//----- SI ESTA ACTIVA LA ALARMA
if (alarmActivated == true)
{
    digitalWrite(led, HIGH);
    digitalWrite(buzzer, LOW);
    Alarm = 0;
    activated = 1;
    Serial.println(Alarm);

    //-----SI ALGUNA VENTANA SE ABRE-----//
    if (sensorM2 == 1 || sensorM3 == 1 || sensorIR2 > 0 || sensorIR1 > 0)
    { // Si algun sensor magnetico da valor d
        Alarm = 1;
        Serial.println(Alarm);
        lcd.clear();
        enterPassword();
    }
}
```

```
}  
  
//-----SI LA PUERTA SE ABRE-----//  
  
if (sensorM1 == 1 && flag == 0 || sensorM4 == 1 && flag == 0) // Si la puerta principal o puerta de cochera se abre  
{  
  
    activated = 1; // activated = true;  
  
    digitalWrite(buzzer, LOW);  
  
    Alarm = 0;  
  
    Serial.print("Alarma=");  
  
    Serial.print(Alarm);  
  
    tiempo2 = millis();  
  
    s = millis();  
  
    Serial.print(" Segundo");  
  
    Serial.print(" ");  
  
    se = s / 1000;  
  
    Serial.print(se);  
  
    Serial.println(" contador:");  
  
    Serial.println(" ");  
  
    writePass(); // INGRESAR CONTRASEÑA ANTES DE QUE SE ACTIVE LA SIRENA  
  
}  
  
  
if (sensorM1 == 0)  
{  
  
    // activated = 0; //activated = false;  
  
}  
}  
  
  
//-----MENÚ DEL SISTEMA DE ALARMA-----//  
  
if (!alarmActivated)  
{  
  
    digitalWrite(led, LOW);  
  
    digitalWrite(buzzer, LOW);  
  
    Alarm = 0;
```

```
if (screenOffMsg == 0)
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("A Activar");
    lcd.setCursor(0, 1);
    lcd.print("B Cambiar clave");
    digitalWrite(buzzer, LOW);
    Alarm = 0;
    activated = 0;
    screenOffMsg = 1;
}

//-----SI SE PRESIONA A ALARMA ACTIVADA-----

keypressed = myKeypad.getKey();
if (keypressed == 'A' || obtenido == 1)
{
    digitalWrite(buzzer, LOW);
    Alarm = 0;
    // activated = 1;
    activateAlarm();
}

//-----CAMBIAR LA CONTRASEÑA-----

//---GUARDAR LA NUEVA CONTRASEÑA EN LA EEPROM PARA QUE SE QUEDE COMO PREDETER

else if (keypressed == 'B')
{
    lcd.clear();
    int i = 1;
    digitalWrite(buzzer, LOW);
    Alarm = 0;
    tempPassword = "";
    lcd.setCursor(0, 0);
    lcd.print("Clave actual");
    lcd.setCursor(0, 1);
```

```
lcd.print(">");

passChangeMode = true;

passChanged = true;

while (passChanged)

{

    keypressed = myKeypad.getKey();

    if (keypressed != NO_KEY)

    {

        if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '4' || keypressed == '5' || keypressed == '6' ||

keypressed == '8' || keypressed == '9')

        {

            tempPassword += keypressed;

            lcd.setCursor(i, 1);

            lcd.print("*");

            i++;

        }

    }

    if (i > 5 || keypressed == '#')

    {

        tempPassword = "";

        i = 1;

        lcd.clear();

        lcd.setCursor(0, 0);

        lcd.print("Clave actual");

        lcd.setCursor(0, 1);

        lcd.print(">");

    }

    if (keypressed == 'D')

    {

        i = 1;

        digitalWrite(buzzer, LOW);

        Alarm = 1;

        if (password == tempPassword)
```

```
{
    tempPassword = "";
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Cambiar clave");
    lcd.setCursor(0, 1);
    lcd.print(">");
    while (passChangeMode)
    {
        keypressed = myKeypad.getKey();
        if (keypressed != NO_KEY)
        {
            if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '4' || keypressed == '5' || keypressed == '6' |
keypressed == '8' || keypressed == '9')
            {
                tempPassword += keypressed;
                lcd.setCursor(i, 1);
                lcd.print(tempPassword);
                i++;
                digitalWrite(buzzer, LOW);
                Alarm = 1;
            }
        }
        if (i > 5 || keypressed == '#')
        {
            tempPassword = "";
            i = 1;
            digitalWrite(buzzer, LOW);
            Alarm = 1;
            lcd.clear();
            lcd.setCursor(0, 0);
            lcd.print("Cambiar clave");
            lcd.setCursor(0, 1);
```



```
        lcd.print(">");
    }
    if (keypressed == 'D')
    {
        i = 1;
        digitalWrite(buzzer, LOW);
        Alarm = 1;
        password = tempPassword;
        EEPROM.write(1, 100);
        passChangeMode = false;
        passChanged = false;
        screenOffMsg = 0;
    }
}
}
}
}
}
}
}
}
}

//-----CONTADOR AL ACTIVAR LA ALARMA-----/
void activateAlarm()
{
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Alarma activa");
    lcd.setCursor(0, 1);
    lcd.print(" en...");

    int countdown = 9; // 9 segundos para que se active la alarma
    while (countdown != 0)
```

```
{  
    lcd.setCursor(13, 1);  
    lcd.print(countdown);  
    countdown--;  
    digitalWrite(buzzer, LOW);  
    Alarm = 0;  
    activated = 1;  
    delay(500);  
    digitalWrite(led, HIGH);  
    delay(500);  
    digitalWrite(led, LOW);  
}  
lcd.clear();  
lcd.setCursor(0, 0);  
lcd.print("Alarma activada");  
alarmActivated = true;  
digitalWrite(buzzer, LOW);  
Alarm = 0;  
}  
  
//-----INGRESAR LA CONTRASEÑA PARA DESACTIVAR LA ALARMA Y LA SIRENA-  
void enterPassword()  
{  
    int k = 5;  
    Alarm = 1;  
    tempPassword = "";  
    activated = true;  
    lcd.clear();  
    lcd.setCursor(0, 0);  
    lcd.print(" * ALARMA * ");  
    lcd.setCursor(0, 1);  
    lcd.print("Clave>");  
    digitalWrite(buzzer, HIGH);
```

```
Alarm = 1;

while (activated)

{
    // Serial.println("SIRENA ACTIVADA");
    digitalWrite(buzzer, HIGH);
    Alarm = 1;

    keypressed = myKeypad.getKey();
    if (keypressed != NO_KEY)
    {
        if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '4' || keypressed == '5' || keypressed == '6' ||
            keypressed == '8' || keypressed == '9')
        {
            tempPassword += keypressed;
            lcd.setCursor(k, 1);
            lcd.print(tempPassword);
            k++;
        }
    }
    if (k > 9 || keypressed == '#')
    {
        tempPassword = "";
        k = 5;
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print(" * ALARMA * ");
        lcd.setCursor(0, 1);
        lcd.print("Clave>");
        digitalWrite(buzzer, HIGH);
        Alarm = 1;
    }
    if (keypressed == 'D')
    {

```

```
if (tempPassword == password)
{
    activated = false;
    activated = 0;
    alarmActivated = false;
    digitalWrite(buzzer, LOW);
    Alarm = 0;
    screenOffMsg = 0;
    // obtenido = 0;
    // recibi = false;
    // regresar la bandera a 0
    flag = 0;
}
else if (tempPassword != password)
{
    lcd.setCursor(0, 1);
    lcd.print("Error!");
    delay(2000);
    lcd.clear();
    digitalWrite(buzzer, HIGH);
    Alarm = 1;
    lcd.setCursor(0, 0);
    lcd.print(" * ALARMA * ");
    lcd.setCursor(0, 1);
    lcd.print("Clave> ");
}
}
}
}

//-----DESACTIVAS ALARMA ANTES 15 SEG O ACTIVA SIRENA-----

void writePass()
{
```

```
int k = 5;

tempPassword = "";

// activated = true;

// lcd.clear();

lcd.setCursor(0, 0);

lcd.print(" PUERTA ABIERTA ");

lcd.setCursor(0, 1);

lcd.print("Clave>");

while (activated)
{
    keypressed = myKeypad.getKey();

    Serial.print(contador);

    // Serial.println(" Alarma");

    if (keypressed != NO_KEY)
    {
        if (keypressed == '0' || keypressed == '1' || keypressed == '2' || keypressed == '4' || keypressed == '5' || keypressed == '6' ||
            keypressed == '8' || keypressed == '9')
        {
            tempPassword += keypressed;

            Serial.println(tempPassword);

            lcd.setCursor(k, 1);

            lcd.print("*****");

            k++;
        }
    }

    tiempo2 = millis();

    if (((tiempo2 - tiempo1) >= 1000) && (contador < a))
    {
        contador++;

        Serial.println(contador);

        tiempo1 = tiempo2;

        if (contador >= a)
        {
```

```
Alarm = 0;

// Serial.println("FUERA.");

// delay(5000);

contador = 0;

flag = 1;

activated = false;

enterPassword();

}

}

if (k > 9 || keypressed == '#')
{
    tempPassword = "";
    k = 5;
    lcd.clear();
    digitalWrite(buzzer, LOW);
    Alarm = 0;
    lcd.setCursor(0, 0);
    lcd.print(" PUERTA ABIERTA ");
    lcd.setCursor(0, 1);
    lcd.print("Clave>");
}

if (keypressed == 'D')
{
    if (tempPassword == password)
    {
        activated = false;
        activated = 0;
        alarmActivated = false;
        digitalWrite(buzzer, LOW);
        Alarm = 0;
        // recibi = false;
        // obtenido = 0;
```

```
    screenOffMsg = 0;
}
else if (tempPassword != password)
{
    lcd.setCursor(0, 1);
    lcd.print("Error!");
    digitalWrite(buzzer, HIGH);
    Alarm = 1;
    delay(2000);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(" * ALARMA * ");
    lcd.setCursor(0, 1);
    lcd.print("Clave> ");
}
}
}
```

```
String getRequest(String serverName)
{
    HTTPClient http;
    http.begin(serverName);

    // Enviamos petición HTTP
    int httpResponseCode = http.GET();

    String payload = "...";

    if (httpResponseCode > 0)
    {
        Serial.print("HTTP Response code: ");
        Serial.println(httpResponseCode);
    }
}
```

```
    payload = http.getString();
}
else
{
    Serial.print("Error code: ");
    Serial.println(httpResponseCode);
}

// liberamos
http.end();

return payload;
}

void sendDataWIFI()
{

    if (activated == 0 && posSendData == true)
    { // si la alarma está desactivada
        // Enviar estado de la alarma
        sendData1 = sendData1 + "activated&valor=" + (String)activated;
        answer = getRequest(sendData1);
        if (sensorM2 == 0)
        {
            sendData2 = sendData2 + "sensorM2&valor=" + (String)sensorM2;
            answer2 = getRequest(sendData2);
            delay(100);
            posSendData = false;
        }
        if (sensorIR1 == 0)
        {
            sendData3 = sendData3 + "sensorIR1&valor=" + (String)sensorIR1;
            answer3 = getRequest(sendData3);
            delay(100);
        }
    }
}
```



```
    posSendData = false;
}
if (sensorM3 == 0)
{
    sendData4 = sendData4 + "sensorM3&valor=" + (String)sensorM3;
    answer4 = getRequest(sendData4);
    delay(100);
    posSendData = false;
}
if (sensorIR2 == 0)
{
    sendData5 = sendData5 + "sensorIR2&valor=" + (String)sensorIR2;
    answer5 = getRequest(sendData5);
    delay(100);
    posSendData = false;
}
if (sensorM1 == 0)
{
    sendData6 = sendData6 + "sensorM1&valor=" + (String)sensorM1;
    answer6 = getRequest(sendData6);
    delay(100);
    posSendData = false;
}
if (sensorM4 == 0)
{
    sendData7 = sendData7 + "sensorM4&valor=" + (String)sensorM4;
    answer7 = getRequest(sendData7);
    delay(100);
    posSendData = false;
}
// posSendData = false;
delay(100);
}
```

```
else if (activated == 1 && posSendData == false)
{ // Si alarma está activa enviar datos

    sendData1 = sendData1 + "activated&valor=" + (String)activated;

    answer = getRequest(sendData1);

    delay(100);

    obtenido = 0;

    sendData8 = sendData8 + "red&valor=" + (String)obtenido;

    answer = getRequest(sendData8);

    delay(100);

    if (sensorM2 == 1)
    {

        sendData2 = sendData2 + "sensorM2&valor=" + (String)sensorM2;

        answer2 = getRequest(sendData2);

        delay(100);

        posSendData = true;

    }

    if (sensorIR1 == 1)
    {

        sendData3 = sendData3 + "sensorIR1&valor=" + (String)sensorIR1;

        answer3 = getRequest(sendData3);

        delay(100);

        posSendData = true;

    }

    if (sensorM3 == 1)
    {

        sendData4 = sendData4 + "sensorM3&valor=" + (String)sensorM3;

        answer4 = getRequest(sendData4);

        delay(100);

        posSendData = true;

    }

    if (sensorIR2 == 1)
    {

        sendData5 = sendData5 + "sensorIR2&valor=" + (String)sensorIR2;
```

```
    answer5 = getRequest(sendData5);  
    delay(100);  
    posSendData = true;  
}  
if (sensorM1 == 1 && activated == true)  
{  
    sendData6 = sendData6 + "sensorM1&valor=" + (String)sensorM1;  
    answer6 = getRequest(sendData6);  
    delay(100);  
    posSendData = true;  
}  
if (sensorM4 == 1)  
{  
    sendData7 = sendData7 + "sensorM4&valor=" + (String)sensorM4;  
    answer7 = getRequest(sendData7);  
    delay(100);  
    posSendData = true;  
}  
Serial.println("WIFI");  
// posSendData = true;  
}  
}
```

Prototipo.



Figura 4.1. Prototipo instalado.

Análisis de costos.

Cantidad	Nombre del sensor o componente	Costo Total
2	Sensores Infrarrojos	\$18.00
1	Teclado Keypad	\$45.00
1	Pantalla LCD con módulo de i2C	\$140.00
3	Sensores magnéticos	\$75.00
1	ESP32 Wifi + Bluetooth 4.2 Ble Nodemcu	\$175.00
1	Led Ultrabrillante	\$2.00
1	Transistor 2N2222A	\$6.00

1	Regulador de voltaje LM7805	\$6.00
1	Relevador 5V	\$12.00
1	Switch ON/OFF	\$10.00
Varios	Pines, bases, cables, soldadura, etc.	\$64.00
1	Piezas de impresión en PLA	\$214.00

Total: \$767.00 pesos

Conclusiones

Con respecto al objetivo general se logró crear el sistema de alarma identificado los sensores utilizados para plantear como fue compuesta la caja de control, luego la interfaz de comunicación (el servidor creado especialmente para recibir datos de cada sistema y poder controlarlo tanto en general como individual) y finalmente la alarma tuvo una gran aplicación en la domótica gracias a su comunicación de WIFI, donde las características del funcionamiento eran arrojadas al servidor junto con el resto de los sistemas y así sabemos como crear una plataforma de comunicación para este tipo de sistemas.

Bibliografía.

News, E (2017). Resistencia. Fluke.

Newsletter, I. (2015). LM7805 Todo sobre el regulador de tensión. Hardware Libre.

Rojas, M. (2018). Historia del Led. Iliminet.

Tecnología, E. L. (2018). ¿Qué es un LED? LED

Tecnología. UAEH. (2021). Keypad modo de programar. UAEH.