

Universidad de Lima
Facultad de Ingeniería
Fundamentos de Electricidad y Electrónica



SISTEMA DE SEGURIDAD CON SENSOR DE MOVIMIENTO

Caramantin Cruz, Leandro Aaron

Código 20222909

Chaparro Rodriguez, Sergio Alonso

Código 20220591

Jara Espinoza, Rodrigo

Código 20224280

Mendoza Parra, Santiago Enrique

Código 20221584

Sección: 428

Lima – Perú
Diciembre de 2023

ÍNDICE GENERAL

ÍNDICE DE FIGURAS.....	3
CAPÍTULO I: INTRODUCCIÓN.....	4
1.1. Descripción del Problema.....	4
1.2. Propuesta de Solución.....	5
1.3. Objetivos.....	6
CAPÍTULO II: MARCO TEÓRICO.....	7
CAPÍTULO III: DISEÑO DE LA SOLUCIÓN.....	10
3.1. Descripción de bloques.....	10
3.2. Diagrama circuital y funcionamiento.....	10
3.3. Costo de implementación.....	12
CAPÍTULO IV: RESULTADOS.....	13
CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES.....	23
BIBLIOGRAFÍA.....	24
ANEXO: CÓDIGO FUENTE.....	25

ÍNDICE DE FIGURAS

Figura 1.1 <i>Diagrama de bloques</i>	10
Figura 1.2 <i>Diagrama circuital</i>	10
Figura 1.3 <i>Circuito implementado en Tinkercad</i>	11
Figura 1.4 <i>Costo de implementación física</i>	12
Figura 2.1 <i>Sistema activado en implementación física</i>	13
Figura 2.2 <i>Sistema activado en Tinkercad</i>	13
Figura 2.3 <i>“Nada sospechoso” en implementación física</i>	14
Figura 2.4 <i>“Nada sospechoso” en Tinkercad</i>	14
Figura 2.5 <i>Detección de intruso en implementación física</i>	15
Figura 2.6 <i>Detección de intruso en Tinkercad</i>	15
Figura 2.7 <i>Prendiendo luces en implementación física</i>	16
Figura 2.8 <i>Prendiendo luces en Tinkercad</i>	16
Figura 2.9 <i>Activando alarma en implementación física</i>	17
Figura 2.10 <i>Activando alarma en Tinkercad</i>	17
Figura 2.11 <i>Cerrando puertas en implementación física</i>	18
Figura 2.12 <i>Cerrando puertas en Tinkercad</i>	18
Figura 2.13 <i>Puertas cerradas en implementación física</i>	19
Figura 2.14 <i>Puertas cerradas en Tinkercad</i>	19
Figura 2.15 <i>Llamando a las autoridades en implementación física</i>	20
Figura 2.16 <i>Llamando a las autoridades en Tinkercad</i>	20
Figura 2.17 <i>Desactivación del sistema en implementación física</i>	21
Figura 2.18 <i>Desactivación del sistema en Tinkercad</i>	21

CAPÍTULO I: INTRODUCCIÓN

1.1. Descripción del problema:

La seguridad en las comunidades y hogares es una preocupación fundamental en la actualidad. Según los datos recopilados por Prosegur (2023), se puede apreciar que el número de robos en la modalidad robos a vivienda han aumentado significativamente durante los últimos años, teniendo 163 casos en solo el mes de Enero en Lima Metropolitana. Además, en gran parte de estos casos, hubo uso de la fuerza física, ya sea amenaza o violencia en contra de las víctimas.

Este problema no solo tiene un impacto en la seguridad personal de las víctimas o víctimas potenciales, sino que también tiene consecuencias más amplias en la sostenibilidad de las comunidades. La inseguridad puede llevar a la pérdida de empleos y el desplazamiento de familias a áreas más seguras. Esto afecta la cohesión social y el desarrollo sostenible en varias dimensiones, incluyendo la económica, la social y la ambiental.

La poca eficacia policial en este tipo de casos, sumada a la falta de sistemas de seguridad efectivos y accesibles en muchas zonas de Lima contribuyen a este problema. La mayoría de personas no tienen el poder económico para adquirir sistemas de seguridad costosos y eso los deja vulnerables a este tipo de robos a domicilio.

Este problema está fuertemente relacionado con el Objetivo de Desarrollo Sostenible número 11: "Ciudades y comunidades sostenibles". Este objetivo busca garantizar que las ciudades y los asentamientos humanos sean sostenibles, lo que incluye abordar cuestiones de seguridad, además de cuestiones medioambientales. Aquí es donde pueden entrar en juego los sistemas de seguridad efectivos y ahorradores. La falta de seguridad en las comunidades y el alto costo de los sistemas de alarmas, con un consumo elevado de energía, puede obstaculizar el logro de un entorno urbano sostenible, inclusivo y resiliente, lo que hace que este problema sea relevante para el ODS 11.

1.2. Propuesta de solución:

Para abordar la problemática de la seguridad en las viviendas en el Perú, proponemos la implementación de un sistema de seguridad inteligente que utiliza un sensor de movimiento que activa una serie de mecanismos de prevención. El sistema se diseñaría para funcionar de la siguiente manera.

El sistema estaría desactivado y tendría que ser activado cuando se quiera usar, esto con un botón de encendido o apagado. Luego de encender el sistema, el sensor de movimiento pasará a estado operativo. El sensor de movimiento detectará cualquier actividad no autorizada en el área monitoreada, como la entrada de un intruso en una vivienda. Cuando se detecte una intrusión, el sistema activará una serie de medidas para confirmar la intrusión. Esto podría incluir el encendido de luces exteriores para iluminar la zona oscura, la activación de una alarma sonora para alertar a los residentes o la notificación a través de una aplicación móvil.

En caso se confirme la intrusión, el sistema podría cerrar automáticamente las puertas o portones de acceso, bloqueando el paso al intruso para garantizar la seguridad de los ocupantes. Asimismo, el sistema estaría conectado a una central de monitoreo o a una aplicación móvil que permitiría emitir una notificación de asalto a las autoridades.

1.3. Objetivos:

El objetivo principal es disminuir la inseguridad ciudadana a través de la implementación de un sistema de seguridad basado en un sensor de movimiento configurado en Arduino. Este estará diseñado para prevenir y operar en caso haya una intrusión.

Objetivos secundarios:

- Configurar el sensor de movimiento en Arduino y programar el funcionamiento del sistema.
- Realizar pruebas de funcionamiento para comprobar la eficacia del sistema en la detección de intrusos y la activación de las medidas de seguridad.

- Programar la frecuencia de la alarma (Buzzer) para que el sonido sea el adecuado.

Este proyecto se relaciona principalmente con el Objetivo de Desarrollo Sostenible número 11: Ciudades y comunidades sostenibles. La implementación de un sistema de seguridad contribuye directamente a la creación de comunidades más seguras y resilientes, promoviendo la sostenibilidad y la calidad de vida de los residentes.

CAPÍTULO II: MARCO TEÓRICO

Descripción de cómo funciona un sensor de movimiento y su relevancia en la automatización de la iluminación.

Descripción de los componentes utilizados, incluyendo el Arduino, el sensor de movimiento, los actuadores (luces), y otros componentes.

Para utilizar arduino de forma efectiva, es necesario comprender sus bases teóricas. En primer lugar, comprender los conceptos básicos de electricidad, como voltaje, corriente y resistencia es necesario para la implementación física del proyecto. En segundo lugar, es de vital importancia entender el funcionamiento del Arduino, siguiendo este la arquitectura básica de un microcontrolador (entrada/salida, memoria, CPU, etc.). En tercer lugar, saber de lenguaje C ya que Arduino se basa en este. Y por último, saber cómo leer y entender hojas de datos (datasheets) de componentes.

Componentes usados:

- Resistencias de 300 ohmios.
- 6 LEDs de diferentes colores.
- Un Micro Servo 9g.
- Un botón.
- Un sensor PIR.
- Un buzzer.
- Un Arduino Mega 2560
- Una pantalla LCD I2C.

Descripción:

- Resistencias de 300 ohmios:

En la electrónica, podemos utilizar un dispositivo simple llamado resistencia, que mide la cantidad de luz que incide en él y transmitirla como una señal.

- LED:

Pequeño componente electrónico que se parece un poco a una bombilla, pero es más eficiente y requiere un voltaje más bajo para funcionar

- Micro Servo 9g:

Componente que se asemeja a un pequeño motor que puede ser programado para que gire o se detenga en una posición específica.

- Dimensiones (largo x ancho x alto) = 0,86 x 0,45 x 1,0 pulgadas (22,0 x 11,5 x 27 mm)
- Peso = 0,32 onzas (9 gramos)
- Peso con cable y conector = 0,37 onzas (10,6 gramos)
- Par de calado a 4,8 voltios = 16,7 oz/pulg. (1,2 kg/cm)
- Voltaje de funcionamiento = 4,0 a 7,2 voltios
- Velocidad de funcionamiento a 4,8 voltios (sin carga) = 0,12 segundos/60 grados
- Longitud del cable del conector = 9,75 pulgadas (248 mm)
- El conector universal tipo "S" se adapta a la mayoría de los receptores.

- Botón:

La funcionalidad del botón se basa en activar el LCD cuando se desea verificar su operatividad.

- Sensor PIR:

Componentes electrónicos que permiten que un elemento electrónico interactúe con el mundo real, posee 3 pines (una a tierra, una a energía y la señal que es conectada al arduino).

- Voltaje: 5V ~ 20V
- Consumo de energía: 65 m
- Salida TTL: 3,3 V, 0V
- Tiempo de bloqueo: 0,2 segundos
- Rango de detección: menos de 120 grados, aproximadamente 7 metros

- Buzzer:

El buzzer o llamado zumbadores, es un altavoz que permite emitir sonidos en diversas frecuencias dependiendo a la programación aplicada en el arduino

- Material del zumbador: PPO.
- Color: Negro.
- Forma del zumbador: redonda.
- Número de pines: 2.
- Frecuencia de oscilación: 2048 Hz.
- Voltaje: 2 - 4 V.
- Nivel de sonido: 85 dB

- Arduino Mega 2560:

La principal función del Arduino Mega es la recepción y transmisión de datos de los sensores previamente establecidos, para así poder manejar el flujo de datos.

- Rendimiento de hasta 16 MIPS a 16MHz, 256k bytes (de los cuales 8k se usan para el gestor de arranque)
- EEPROM de 4k bytes
- SRAM interna de 8kbytes
- Registros de trabajo de uso general 32×8
- Contador en tiempo real con oscilador separado
- Cuatro canales PWM de 8 bits
- Cuatro USART seriales programables
- Controlador/interfaz serie SPI periférica

- Pantalla LCD I2C:

Los datos capturados por este sensor, serán capturados por el Arduino para luego ser mostrados en la pantalla LCD

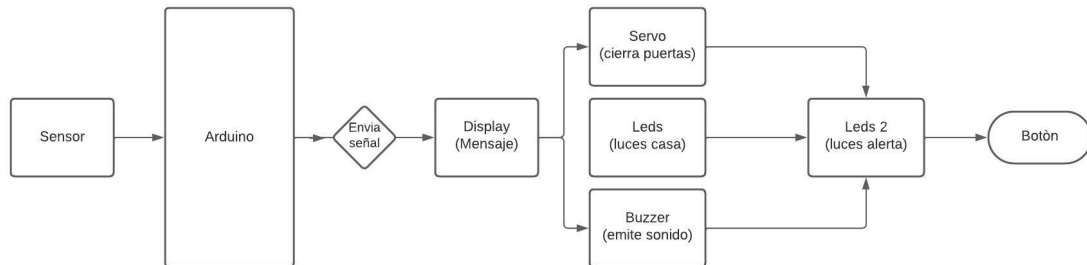
Lugar de compra de los componentes: <https://hifisac.com/en>

CAPÍTULO III: DISEÑO DE LA SOLUCIÓN

3.1. Diagrama de bloques del proyecto

Figura 1.1

Diagrama de bloques



3.2. Diagrama circuital y funcionamiento

Figura 1.2

Diagrama circuital

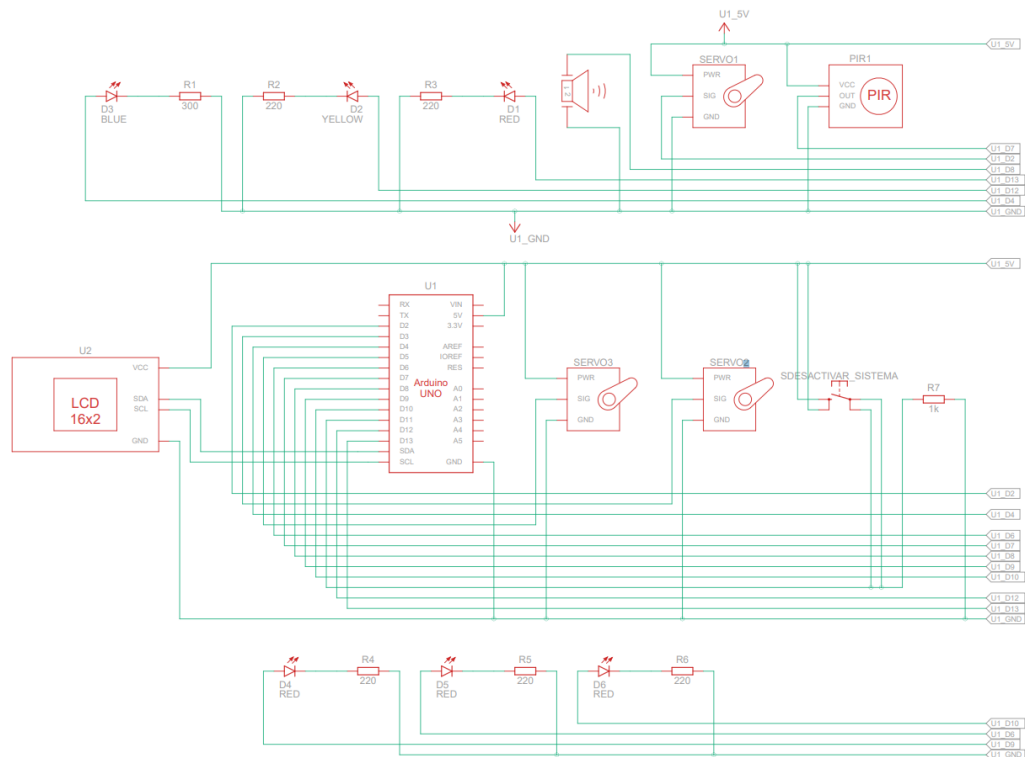
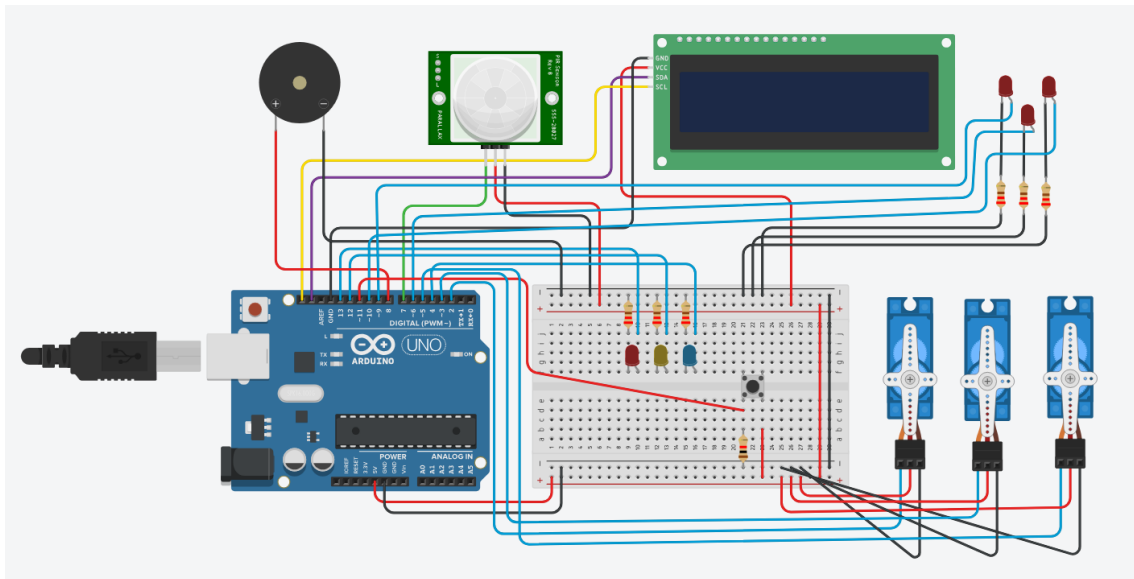


Figura 1.3

Circuito implementado en Tinkercad



- **Arduino MEGA 2560:** El arduino tiene dos cables en la parte inferior por donde se transmite la energía y la tierra hacia el protoboard. También, es donde están conectados los componentes del sistema.
- **Sensor:** En la vida real sería el sensor de proximidad a mayor escala, el cual pueda detectar movimiento en el área determinada por el usuario. En el proyecto, este está conectado en el pin 7 del Arduino MEGA 2560 y el código comienza a correr cuando el sensor detecta un movimiento.
- **Display LCD I2C:** Esto sería una representación tanto de las acciones que estará ejecutando el circuito y los mensajes que se le enviarán a la persona y la llamada a las autoridades después de detectar algún movimiento. El display LCD tiene 4 conexiones, las cuales son GND (tierra) y VCC (energía), las cuales van conectadas al Arduino y protoboard respectivamente, y SDA y SCL, las cuales están conectadas al Arduino también. A través del código es donde se programan los mensajes que muestran el proceso actual del sistema.
- **Buzzer:** Representa una alarma en casa, en caso de que el sensor detecte alguna presencia no deseada cuando el usuario haya dejado el sensor activo. El buzzer tiene dos cables, el primero está conectado al pin 8 del Arduino y el segundo va a tierra. Una vez el sensor detecta movimiento, el buzzer está programado para sonar hasta que se desactiva el sistema a través de un botón.

- Servo Micro: Sería la representación de una puerta de casa, la cual se cerraría y echaría llave, para asegurar las áreas determinadas y evitar que alguien acceda a la casa. El Micro Servo consta de 3 salidas, dos de ellas son GND (tierra) y VCC (energía). La otra es la señal, la cual va conectada al pin 2, en este caso el Micro Servo va a adoptar una posición inicial de 0° y al momento de activarse la alarma va a girar 180°.
- LEDs 5mm: Los LEDs representan las luces de la casa y que se está llamando a la policía. Los primeros 3 LEDs se prenden progresivamente, representando que se están prendiendo las luces de la vivienda, estos están conectados a los pines 13, 12 y 2. Los últimos 3 LEDs representan la “llamada a las autoridades”, estos están conectados a los pines 10, 9 y 6 y van a parpadear indefinidamente, hasta que el sistema sea desactivado manualmente.
- Botón: El botón lo tendrá el usuario en su app de celular para que solo él pueda desactivar la alarma. Este botón está conectado al pin 11 del Arduino.
- Cable jumper: Estos representarán al cableado y conexiones de electricidad que se establecerán a la hora de la instalación del sistema de alarmas.

3.3. Costo de la implementación

Figura 1.4

Costo de implementación física

Descripción	Cantidad	Precio Total
[SB-156] Protoboard Breadboard Standard de color Blanco de 830 Puntos en Blister	1	S/ 8.00
[YHE12-05-ST] Buzzer o Zumbador Activo de 5Voltios DC Frecuencia: +/- 2300±300Hz Pitch 7.6mm Azul	1	S/ 1.20
[SG90A] Servo Micro 9gr SG90 1.6Kg 5V con Accesorios	1	S/ 9.30
[LR-5-RRPRF] LED Rojo Superbrillante 5mm 12000-14000MCD	1	S/ 0.40
[LV-5-VVTPRF] LED VERDE SEMITRANSARENTE DE 5MM PROFESIONAL	1	S/ 0.40
[LB-5-WC] Led Blanco Transparente 5mm 3-3.2VDC 20mA	1	S/ 0.20
[LY-5-YYTPRF] Led amarillo semitransparente de 5mm profesional 10000mcd	1	S/ 0.40
[MEGA-2560R3] Arduino Mega 2560 R3 Comp. con Arduino + Cable USB	1	S/ 98.60
[DYP-ME003] Sensor de Movimiento PIR en Módulo HC-SR501. Con Lente Fresnel.	1	S/ 7.60
[LCD-216A] Display Alfanumerico LCD de 2x16 Caracteres 1602A Driver HD44780	1	S/ 14.60
[SCH4015PPMC] Cable Jumper PIN-PIN Set de 40 pines x 15cm de Largo de Cobre/Aluminio Cables Dupont	1	S/ 4.50
[300R-025W] Resistencia 1/4W de Carbon CFR25 5 %, Dimensiones: 6.8 x 2.5 mm Pack x 100pzs (300 Ohm)	1	S/ 1.50
[LY-5-YYTPRF] Led amarillo semitransparente de 5mm profesional 10000mcd	3	S/ 1.20
[LR-5-RRPRF] LED Rojo Superbrillante 5mm 12000-14000MCD	1	S/ 0.40
[SCH4015PHMC] Cable jumper PIN-HEM Set 40 pines x 15cm Cobre/Aluminio Dupont	1	S/ 4.50
TOTAL		S/ 152.80

CAPÍTULO IV: RESULTADOS

Activación del sistema: Se prende el sistema para que esté alerta.

Figura 2.1

Sistema activado en implementación física

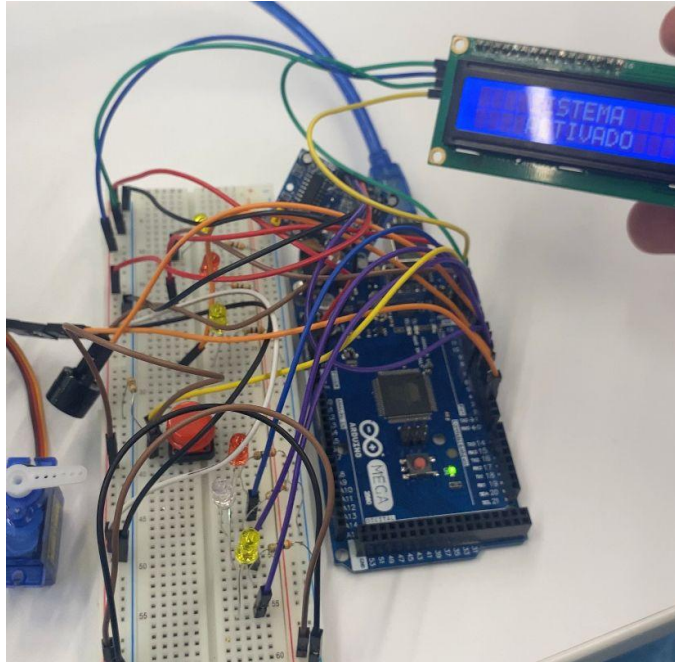
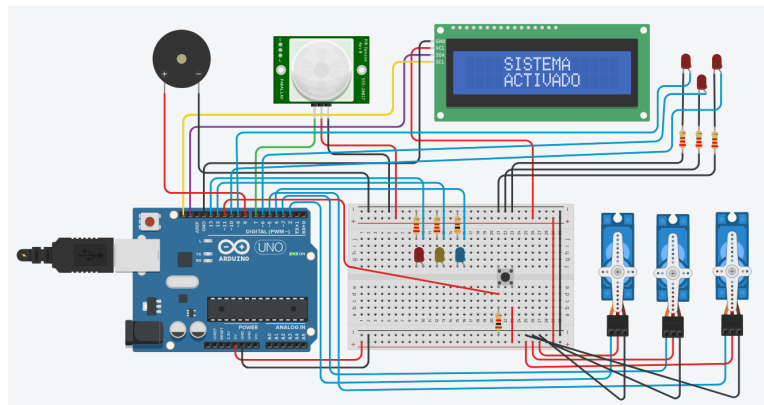


Figura 2.2

Sistema activado en Tinkercad



Cuando el programa es cargado en el arduino, la pantalla LCD muestra el mensaje “SISTEMA ACTIVADO” para avisarnos que el sistema acaba de ser encendido.

El sistema no detecta movimiento: se muestra el mensaje “NADA SOSPECHOSO”

Figura 2.3

“Nada sospechoso” en implementación física

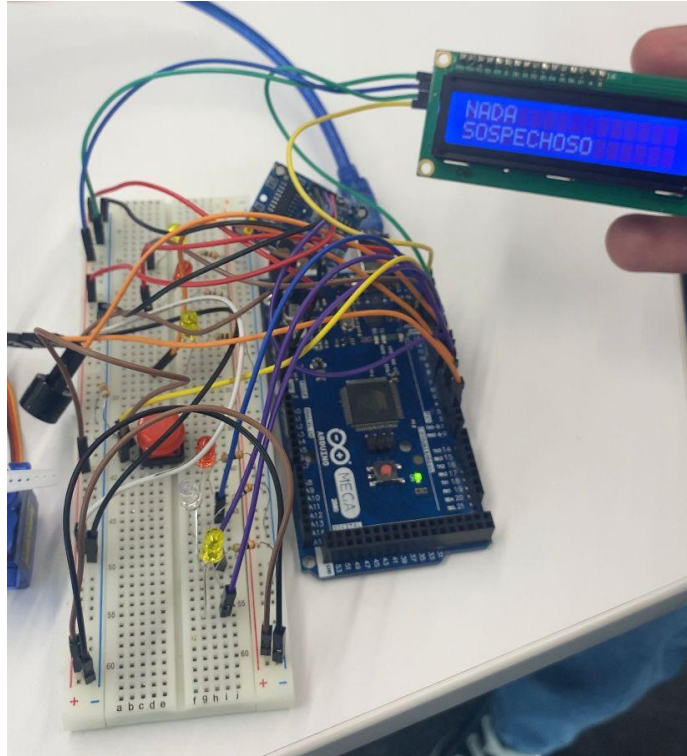
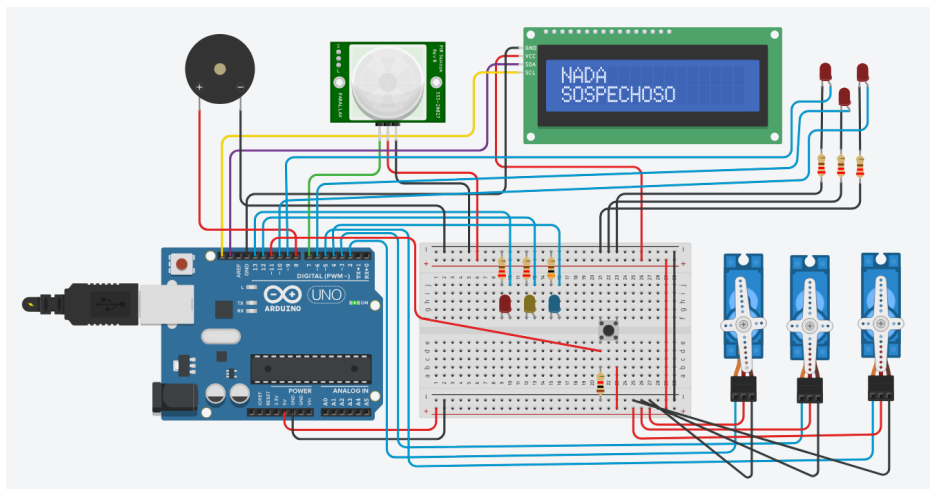


Figura 2.4

“Nada sospechoso” en Tinkercad



Luego de ser activado, el sistema de seguridad se encuentra a la espera de un movimiento para que se active el mecanismo de seguridad. Mientras tanto, se muestra el mensaje “NADA SOSPECHOSO”.

El sensor detecta movimiento: se muestra en el LCD “INTRUSO DETECTADO”

Figura 2.5

Detección de intruso en implementación física

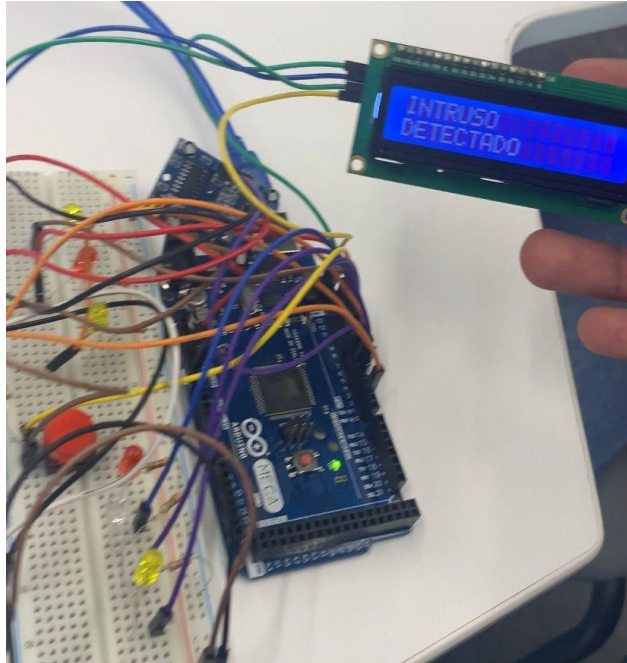
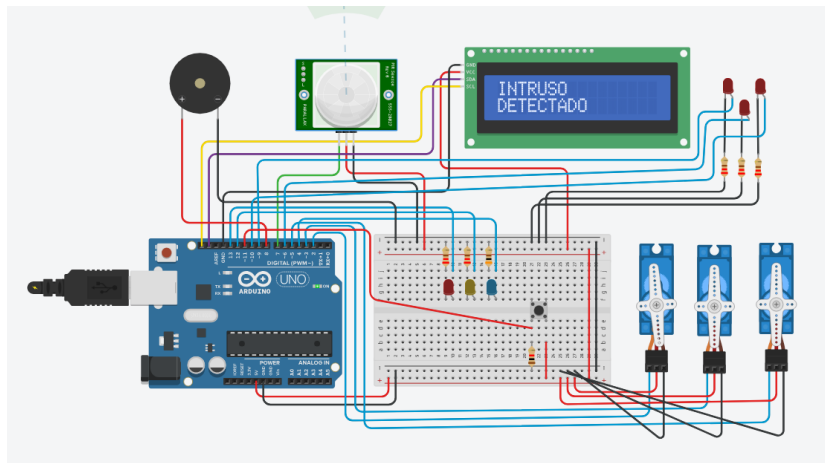


Figura 2.6

Detección de intruso en Tinkercad



Una vez el sensor de movimiento detecte movimiento, el sistema de seguridad empezará a operar. Lo primero que hace es indicar que se detectó un movimiento a través del mensaje “INTRUSO DETECTADO”.

Prender luces: Pantalla LCD “PRENDIENDO LUCES” y se prenden las luces

Figura 2.7

Prendiendo luces en implementación física

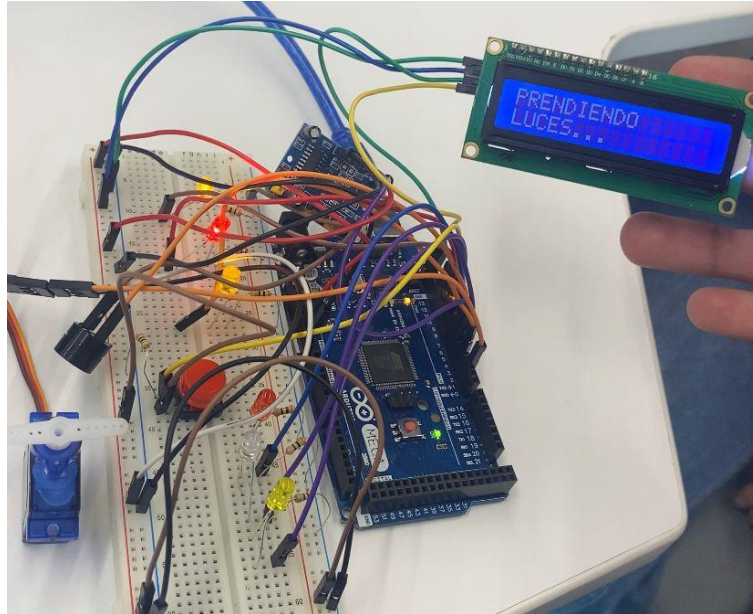
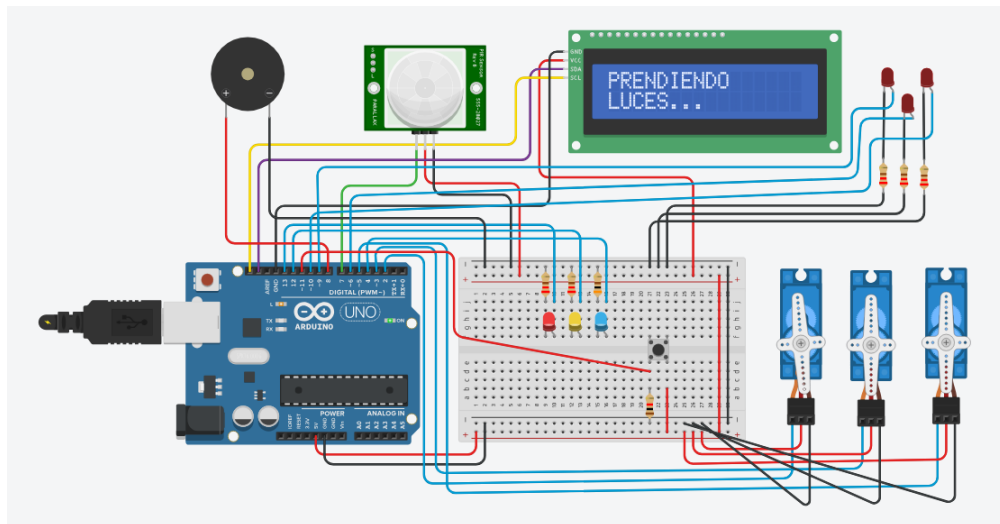


Figura 2.8

Prendiendo luces en Tinkercad



Luego de 2 segundos, el sistema prende las luces de la vivienda. Este proceso se puede ver evidenciado a través del mensaje en la pantalla LCD “PRENDIENDO LUCES..”.

Activar alarma: Pantalla LCD: “Activando alarma” y la alarma empieza a sonar

Figura 2.9

Activando alarma en implementación física

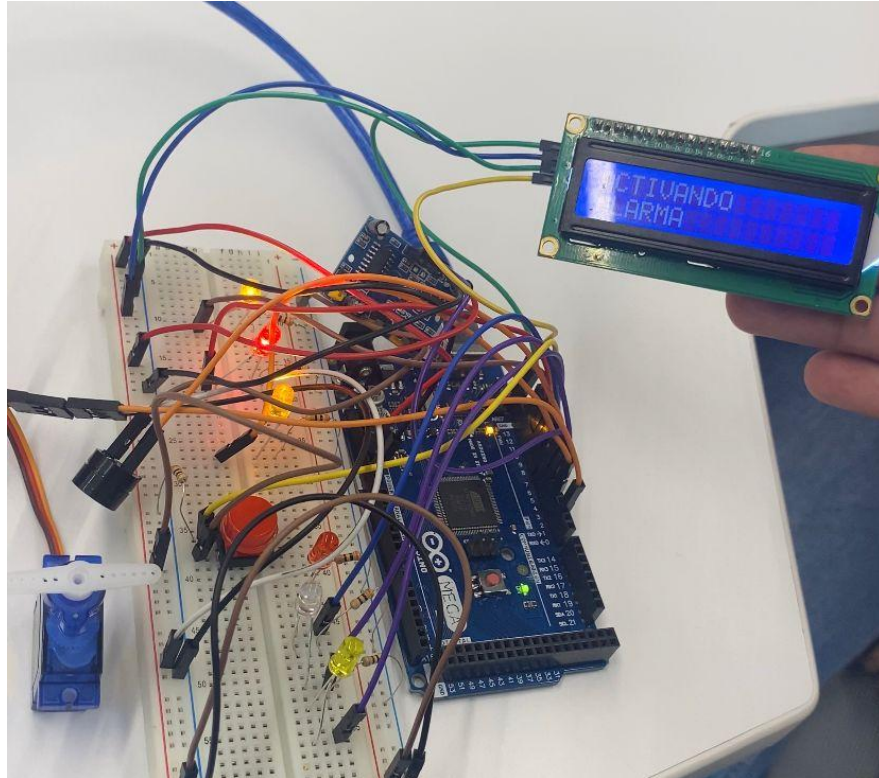
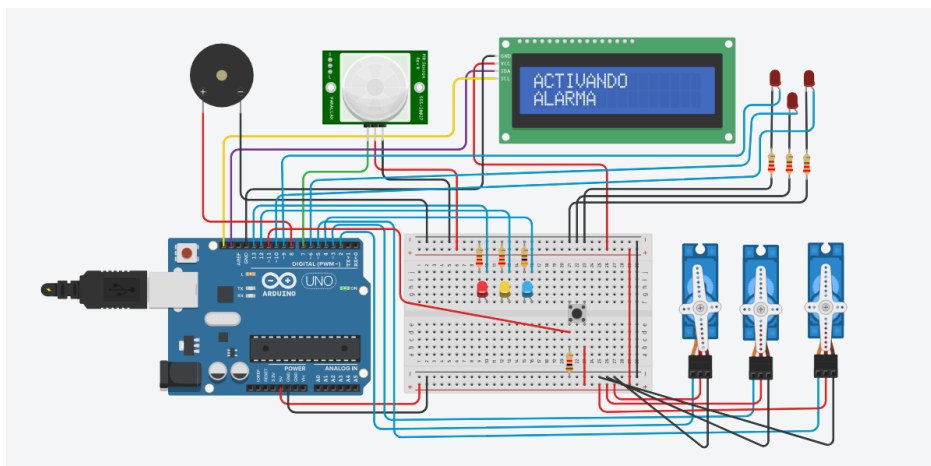


Figura 2.10

Activando alarma en Tinkercad



Luego de 2 segundos, el sistema activa la alarma, la cual sonará hasta que el sistema sea desactivado manualmente. Esto se evidenciará a través de los mensajes “ACTIVANDO ALARMA” y “ALARMA ACTIVADA”.

Cerrando puertas: Pantalla LCD: “CERRANDO PUERTAS” y el Servo da un giro de 180°

Figura 2.11

Cerrando puertas en implementación física

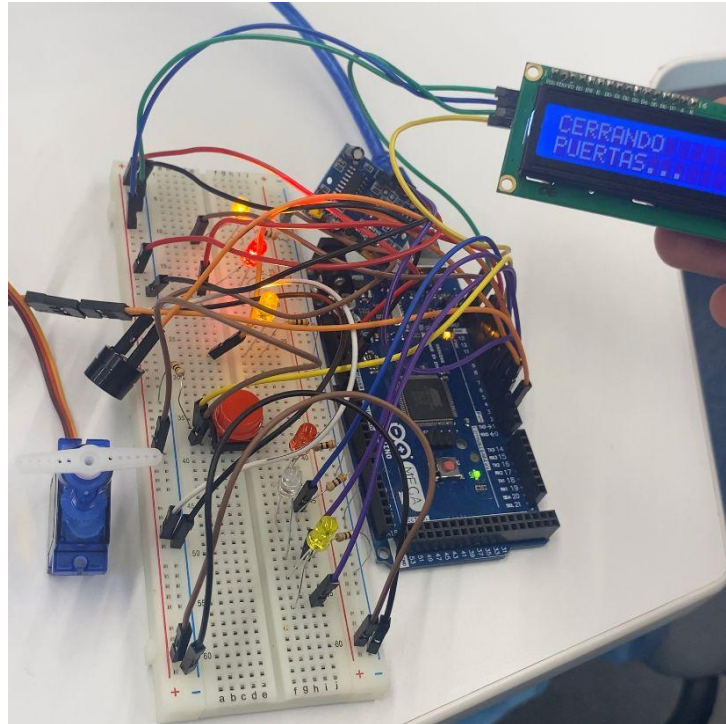
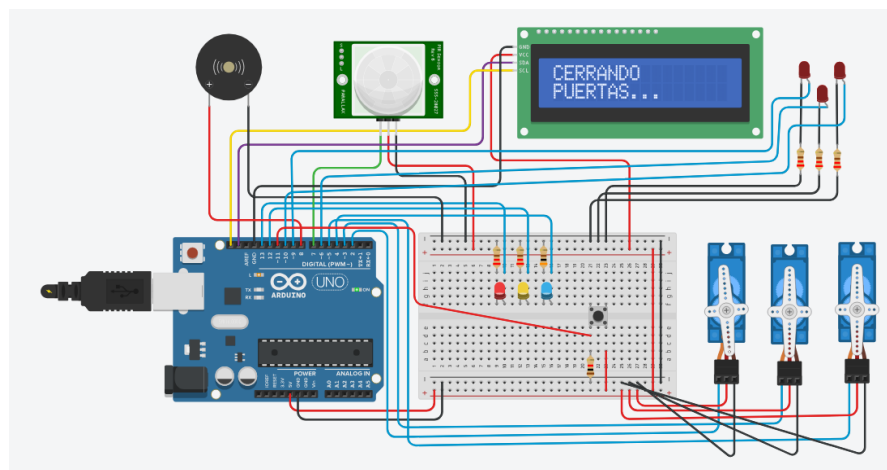


Figura 2.12

Cerrando puertas en Tinkercad



Como tercera medida, el sistema de seguridad empezará a cerrar las puertas de la casa, mientras se muestra el mensaje “CERRANDO PUERTAS...”.

Mensaje en pantalla LCD: “Puertas Cerradas”

Figura 2.13

Puertas cerradas en implementación física

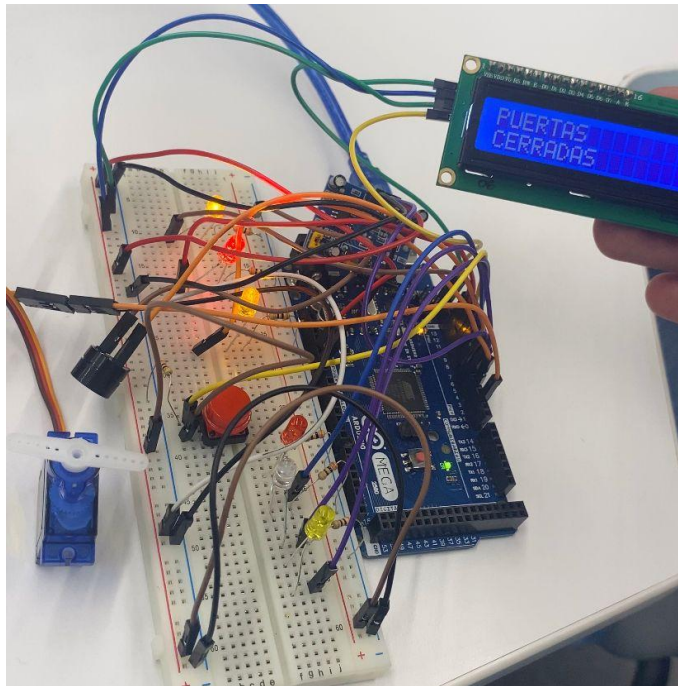
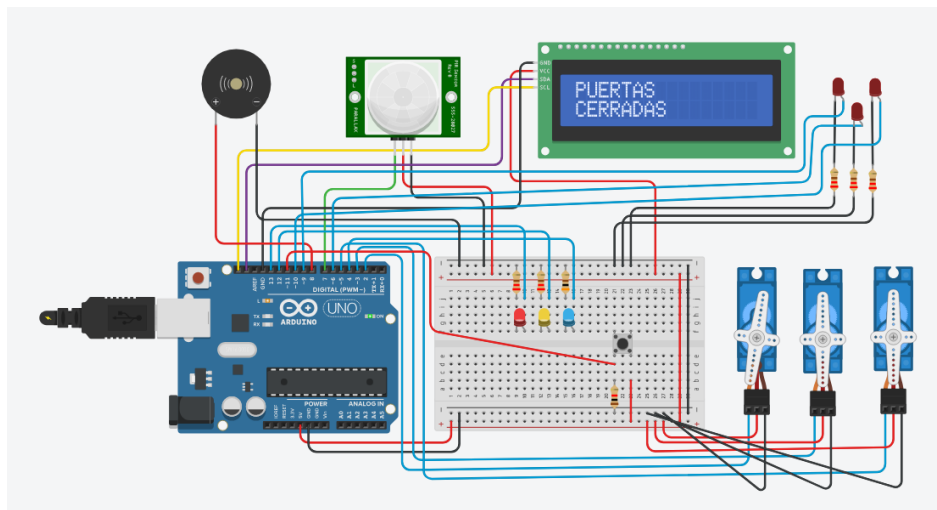


Figura 2.14

Puertas cerradas en Tinkercad



Una vez cerradas las puertas, la pantalla LCD dará la confirmación a través del mensaje “PUERTAS CERRADAS”.

Llamar a las autoridades: Se muestra el mensaje “LLAMANDO A LAS AUTORIDADES” y los leds empiezan a parpadear .

Figura 2.15

Llamando a las autoridades en implementación física

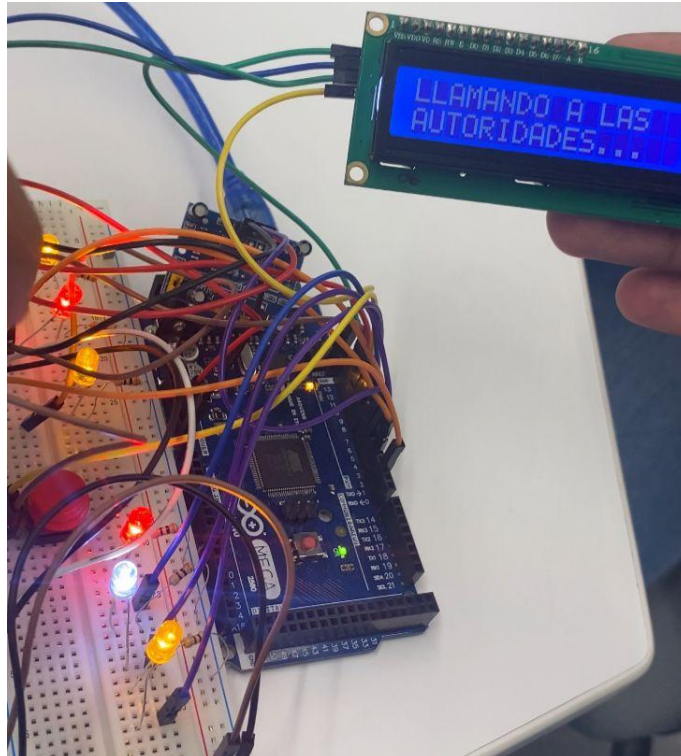
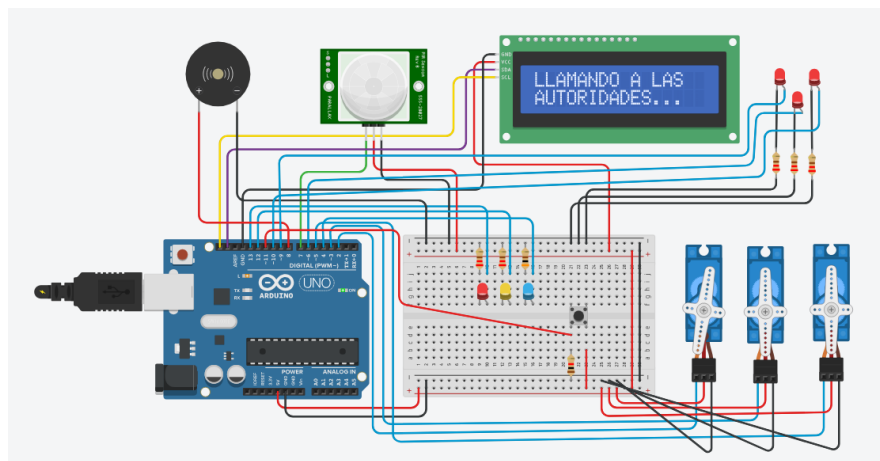


Figura 2.16

Llamando a las autoridades en Tinkercad



Como cuarta y última medida de seguridad, el sistema emitirá un mensaje a las autoridades correspondientes, notificando que hay una intrusión en la vivienda. Los LEDs empezarán a parpadear, indicando que la notificación está siendo enviada.

Desactivar sistema: Se mantiene presionado el botón y el sistema de seguridad se desactiva y vuelve a su estado inicial, esperando un movimiento.

Figura 2.17

Desactivación del sistema en implementación física

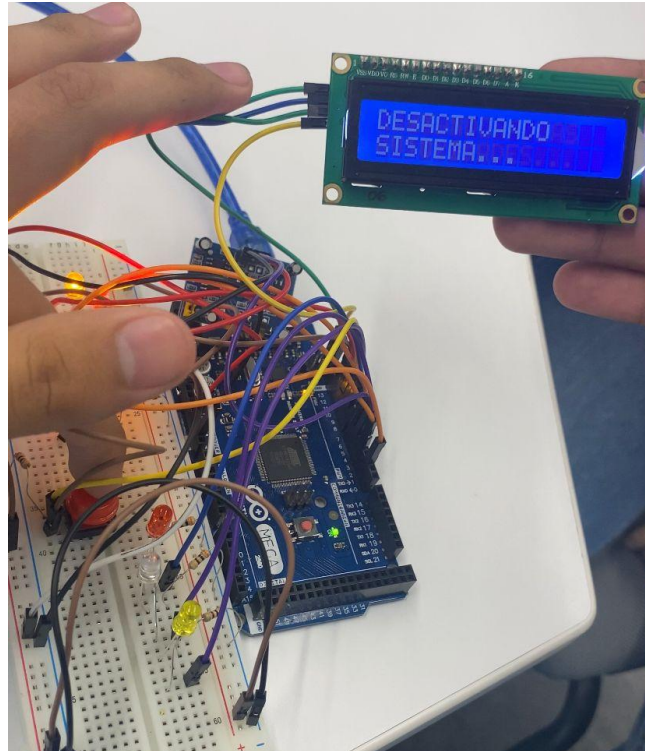
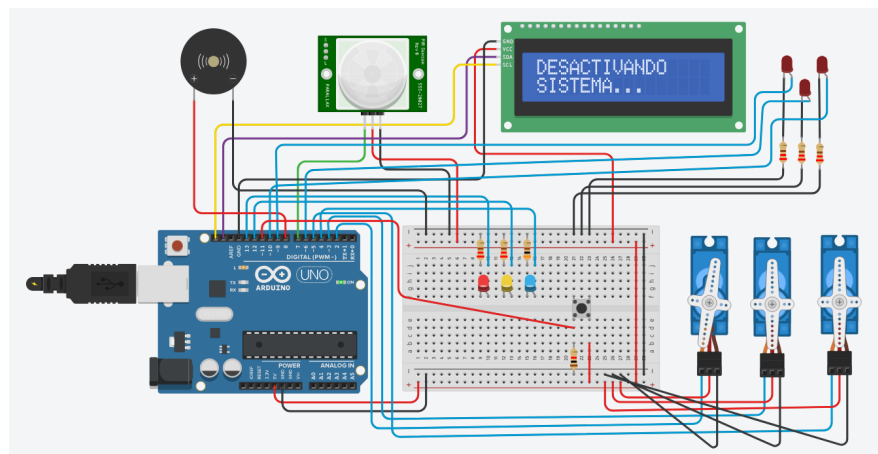


Figura 2.18

Desactivación del sistema en Tinkercad



Luego de que el sistema esté operativo, la forma de apagarlo es dejando presionado durante un breve momento el botón. Este botón tiene la función de desactivar manualmente cada una de las acciones anteriores y reiniciar el sistema al estado “NADA SOSPECHOSO”.

Link del circuito implementado en Tinkercad:

<https://www.tinkercad.com/things/6ear2TZCD26-respaldo-proyecto-integrador-funda>

CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

- La capacidad de encenderse cuando te acercas sugiere que has tenido éxito en la implementación de un sistema de detección de proximidad utilizando Arduino y sensores adecuados.
- Este proyecto puede servir como una base sólida para proyectos futuros. Puedes expandir y mejorar esta idea, integrando más sensores, mejorando la eficiencia del código, o agregando funcionalidades adicionales.
- En este proyecto, la programación es parte fundamental del funcionamiento de manera operativa y adecuada; resaltando la eficacia de estos algoritmos para la interpretación de señales del sensor.
- Además, este proyecto sirve para medir la efectividad de los sensores de detección, examinar la capacidad de procesamiento y almacenamiento del Arduino Mega para manejar tareas relacionadas con estos sensores.

REFERENCIAS

- Prosegur.** (2023). Reporte de robos a domicilios, vehículos y empresas enero. Blog de Prosegur.
<https://blog.prosegur.com.pe/reporte-de-robos-a-domicilios-vehiculos-y-empresas-enero/>
- Banzi, M., Cuartielles, D., Igoe, T., Martino, G., & Mellis, D.** (2014). Arduino. *The official Arduino web page at http://arduino.* cc.
https://anayamultimedia.es/primer_capitulo/introduccion-a-arduino-4a-edicion.pdf
- Romero, H. M., Huachaca, A. P., & Jara, H. R.** (2017). Monitoreo del consumo de energía eléctrica doméstica con arduino. In *Global Partnerships for Development and Engineering Education: Proceedings of the 15th LACCEI International Multi-Conference for Engineering, Education and Technology, July 19-21, 2017, Boca Raton, FL, United States* (p. 253). Latin American and Caribbean Consortium of Engineering Institutions. <https://dialnet.unirioja.es/servlet/articulo?codigo=7352987>
- SG90 datasheet.** (2023). Datasheetspdf.com.
<https://datasheetspdf.com/pdf-file/791970/TowerPro/SG90/1>
- DatasheetCafe** (2023, June 3). HC-SR501 PDF - PIR Sensor, Motion Detector (Manual). DatasheetCafe. <https://www.datasheetcafe.com/hc-sr501-pdf-22640/>
- Arduino® MEGA 2560 Rev3.** (2023).
<https://docs.arduino.cc/resources/datasheets/A000067-datasheet.pdf>

ANEXO: CÓDIGO FUENTE

```
// Proyecto Integrador

#include<LiquidCrystal_I2C.h> //import una libreria para el LCD
#include <Servo.h>

LiquidCrystal_I2C lcd(0x27,16,2);

Servo puerta1; //declaro el Servo con el nombre de puerta1
Servo puerta2;
Servo puerta3;

int led1 = 13; //asigno nombre a los pines
int led2 = 12;
int led3 = 4;

int pir = 7; //sensor
int pirdato; //recibe el dato que obtenga el sensor

int i;

int buzzer = 8; // El pin al que está conectado el zumbador

int llamada1 = 9;
int llamada2 = 6;
int llamada3 = 10;

int botonDesactivar = 11; // Modifica el número del pin según tu
conexión

void setup()
{
    pinMode(led1, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(led3, OUTPUT);

    pinMode(pir, INPUT);

    pinMode(buzzer, OUTPUT);

    //Inicio el sistema y muestro mensaje
```

```

    lcd.init();
    lcd.backlight();
    lcd.setCursor(4,0);
    lcd.print("SISTEMA");
    lcd.setCursor(4,1);
    lcd.print("ACTIVADO");
    delay(2000);

    puerta1.attach(2);
    puerta2.attach(3);
    puerta3.attach(5);

    pinMode(llamada1, OUTPUT);
    pinMode(llamada2, OUTPUT);
    pinMode(llamada3, OUTPUT);

    //Establezco los Servo a 0° como posición inicial
    puerta1.write(0); //de 0 a 180
    puerta2.write(0); //de 0 a 180
    puerta3.write(0); //de 0 a 180

    pinMode(botonDesactivar, INPUT);
}

void loop()
{
    //Mostrar "Nada sospechoso" en el LCD mientras no se detecta
    movimiento
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("NADA");
    lcd.setCursor(0,1);
    lcd.print("SOSPECHOSO");

    pirdato = digitalRead(pir); //recibe el dato del pin 7
    if(pirdato == HIGH) //Si el sensor detecta movimiento
    {
        lcd.clear(); //Emite mensaje "Intruso detectado"
        lcd.setCursor(0,0);
        lcd.print("INTRUSO");
        lcd.setCursor(0,1);
        lcd.print("DETECTADO");
        delay(2000);
    }
}

```

```

    lcd.clear(); //Luego de 2 segundos, muestra "Prendiendo luces"
    lcd.setCursor(0,0);
    lcd.print("PRENDIENDO");
    lcd.setCursor(0,1);
    lcd.print("LUCES...");
    for (i=0; i<255; i++) //Se prenden las luces progresivamente
    {
        analogWrite(led1,i);
        analogWrite(led2,i);
        analogWrite(led3,i);
        delay(10);
    }

    lcd.clear(); //Luego muestra "Activando alarma"
    lcd.setCursor(0,0);
    lcd.print("ACTIVANDO");
    lcd.setCursor(0,1);
    lcd.print("ALARMA");
    delay(2000); //Luego de 2 segundos, se activa la alarma
    activarAlarma();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("ALARMA"); //Se muestra "Alarma activada"
    lcd.setCursor(0,1);
    lcd.print("ACTIVADA");
    delay(2000);

    lcd.clear(); //Luego de 2 segundos, se muestra "Cerrando
puertas"
    lcd.setCursor(0,0);
    lcd.print("CERRANDO");
    lcd.setCursor(0,1);
    lcd.print("PUERTAS...");
    delay(2000);
    puerta1.write(180); //Se giran los Servo en 180°
    puerta2.write(180);
    puerta3.write(180);
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("PUERTAS"); //Se muestra "Puertas cerradas"
    lcd.setCursor(0,1);
    lcd.print("CERRADAS");

```

```

    delay(2000);

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("LLAMANDO A LAS"); //Luego de 2 segundos, se muestra
    "Llamando a las autoridades"
    lcd.setCursor(0,1);
    lcd.print("AUTORIDADES...");
    delay(2000);
    while(digitalRead(botonDesactivar) == LOW) //Y parpadean los
    LEDs hasta que se presiones el botón
    {
        parpadearLEDs();
    }
    //Si el botón está activado, entonces se desactiva el sistema
    if (digitalRead(botonDesactivar) == HIGH)
    {
        desactivarSistema();
    }
}
delay(1000);
}

void activarAlarma()
{
    tone(buzzer, 500); // Frecuencia de 500 Hz
}

void desactivarAlarma()
{
    noTone(buzzer);
}

void parpadearLEDs()
{
    //unsigned long startTime = millis();
    //millis() - startTime < 60000 - 1 minuto de parpadeo
    digitalWrite(llamada1, HIGH);
    digitalWrite(llamada2, HIGH);
    digitalWrite(llamada3, HIGH);
    delay(500);
    digitalWrite(llamada1, LOW);
    digitalWrite(llamada2, LOW);

```

```

        digitalWrite(llamada3, LOW);
        delay(500);
    }
    void desactivarLEDs()
    {
        digitalWrite(llamada1, LOW);
        digitalWrite(llamada2, LOW);
        digitalWrite(llamada3, LOW);
    }

    void desactivarSistema()
    {
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("DESACTIVANDO");
        lcd.setCursor(0, 1);
        lcd.print("SISTEMA...");
        delay(2000);

        // Desactivar todos los componentes
        desactivarAlarma();
        digitalWrite(led1, LOW);
        digitalWrite(led2, LOW);
        digitalWrite(led3, LOW);
        puerta1.write(0); // Abrir puertas
        puerta2.write(0);
        puerta3.write(0);
        desactivarLEDs();

        // Esperar un tiempo antes de volver al loop principal
        delay(2000);
    }

```