

# FACULTAD DE INGENIERÍA

---

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

## Laboratorios 5 y 6

**Autor:**

Campoverde Pacora Marco

Cuicapusa Palomino Michael

Galvez Sedelmayer Andres

Veramendi Montedoro Gonzalo

**Curso:**

Robótica

**Docente:**

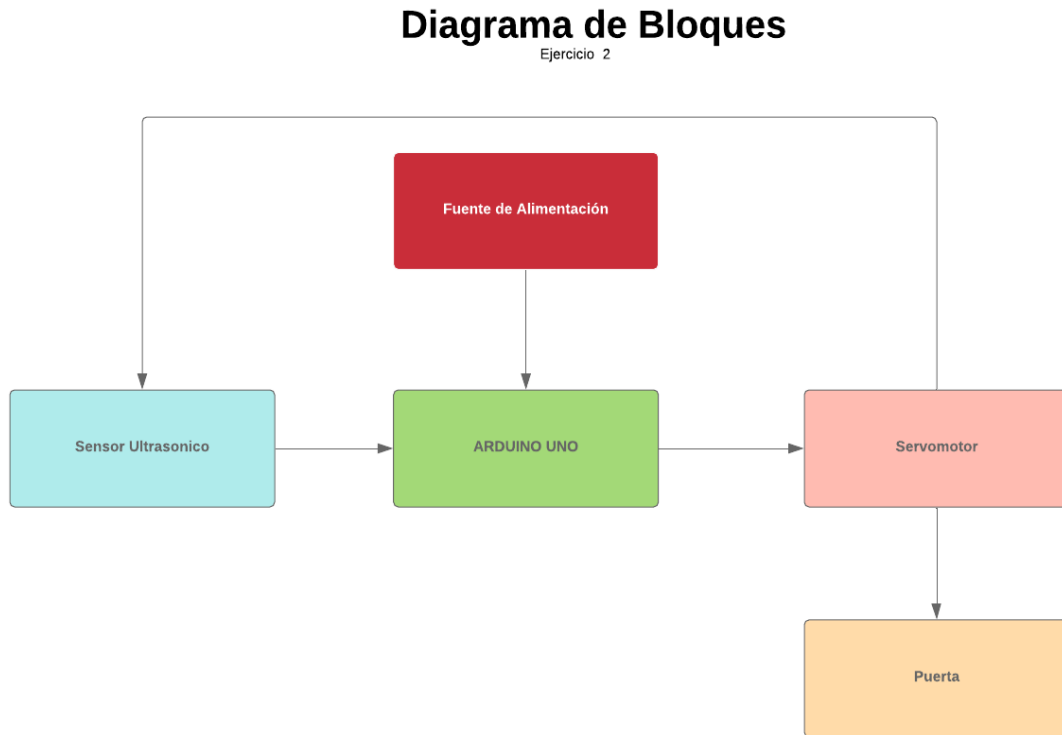
Luz Adanaqué Infante

LIMA – PERÚ

2019

## Proyecto N° 2: Implementar un sistema donde se abra una puerta cuando se detecta la cercanía de un objeto.

**1. Diseñar el diagrama de bloques de los proyectos asignados.**



El diagrama de bloques muestra los componentes que interactúan dentro del sistema, en este caso tenemos el sensor ultrasónico el cual enviara la señal a nuestro Arduino y este a su vez activara un servomotor al darse el escenario esperado.

## 2. Elegir el tipo de sensor y de actuador acorde a la aplicación.

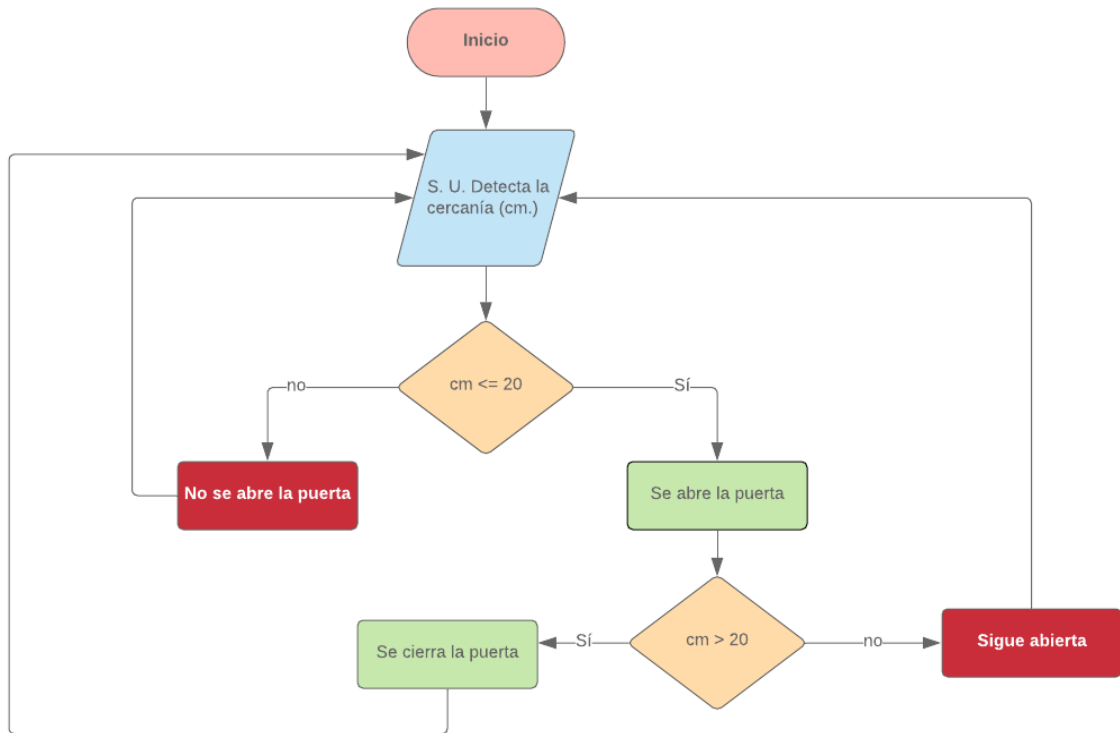


Para el servomotor se active y por consiguiente realice la acción de abrir la puerta, se escogió el sensor HC-SR04, el cual es un sensor de ultrasónico capaz de poder detectar la proximidad de objeto.

### 3. Elaborar la lógica de control correspondiente a la aplicación.

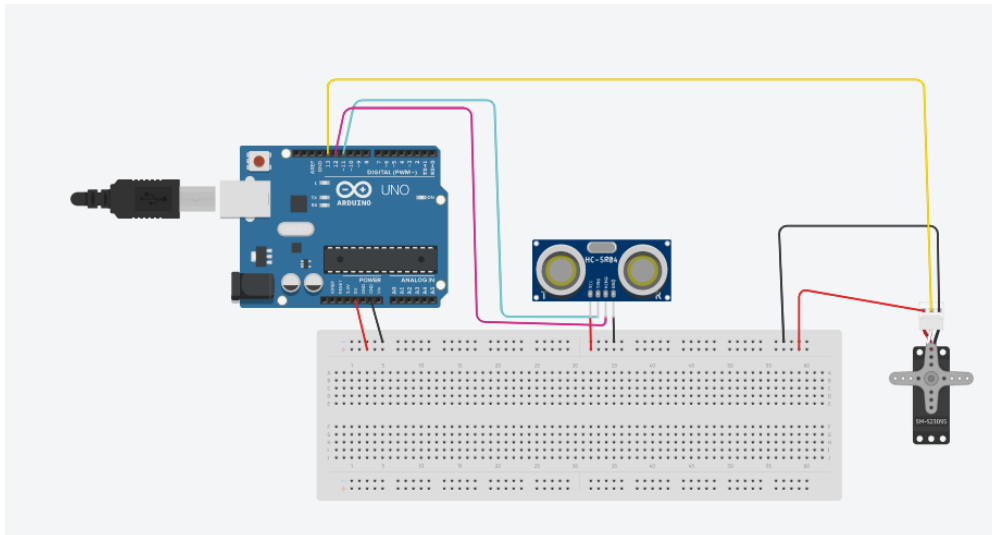
## Lógica de Control

Ejercicio 2



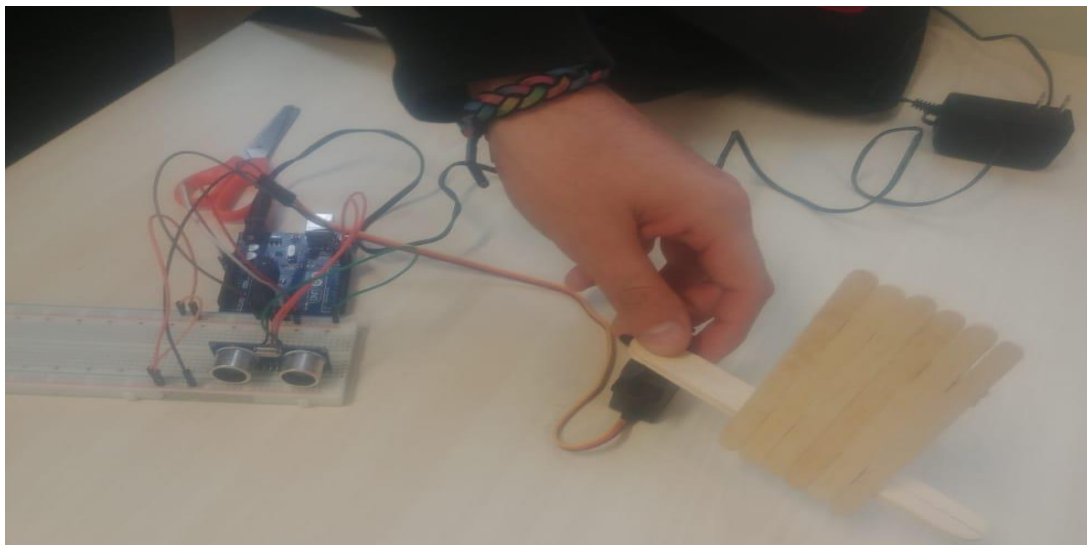
La lógica de control para el ensamblado inicia siempre detectando la cercanía de un objeto en centímetros, si la cercanía es menor a 20cm, se procede a activar el servomotor y por consiguiente abrir una puerta en caso contrario se sigue con la medición de la cercanía.

#### 4. Simular el circuito correspondiente.



Fue necesario el simular nuestro circuito en la herramienta Tinkercad, ya que se necesita definir como serán las conexiones de nuestro sistema y observar su comportamiento.

#### 5. Probar el funcionamiento del circuito



```

✓ → 🏠 ⬆ ⬇
sketch_oct05c $
#include <Servo.h>
const int Trigger = 11;
const int Echo = 12;
const int ledPIN = 4;
int angulo;
Servo myservo;
void setup() {
    Serial.begin(9600);
    pinMode(Trigger, OUTPUT);
    pinMode(Echo, INPUT);
    digitalWrite(Trigger, LOW);
    myservo.attach(13);
}

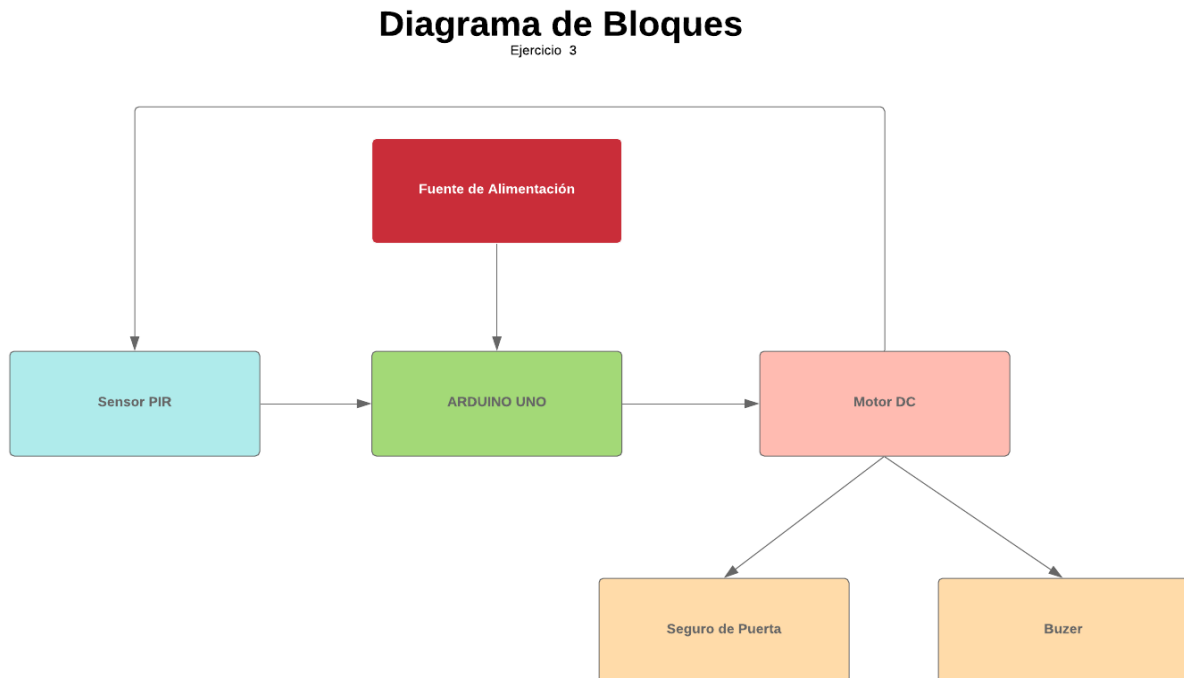
void loop()
{
    long t;
    long d;
    digitalWrite(Trigger, HIGH);
    delayMicroseconds(10);
    digitalWrite(Trigger, LOW);
    pinMode(ledPIN, OUTPUT);
    t = pulseIn(Echo, HIGH);
    d = t/59;
    if(d<=20){
        angulo= 90;
        digitalWrite(ledPIN, LOW);
        myservo.write(angulo);
    }else{
        angulo= 0;
        digitalWrite(ledPIN, HIGH);
        myservo.write(angulo);
    }
    Serial.print("Angulo: ");
    Serial.println(angulo);
    Serial.print("Distancia: ");
    Serial.print(d);
    Serial.print("cm");
    Serial.println();
    delay(500);
}

```

Como último paso, se procede a ensamblar nuestro circuito, observando que realiza con éxito lo esperado que en este caso es el abrir una puerta cuando detecte que un objeto se encuentra a una proximidad menor a 20 cm.

## Proyecto N° 3 Implementar un sistema donde se encienda una alarma y se active el bloqueo de una puerta ante un impacto

### 1. Diseñar el diagrama de bloques de los proyectos asignados.



El diagrama de bloques muestra los componentes que interactúan dentro del sistema, en este caso tenemos el sensor PIR el cual enviara la señal a nuestro Arduino y este a su vez activara un servomotor al darse el escenario esperado el cual es accionar un seguro a la puerta y a la vez activar una alarma.

### 2. Elegir el tipo de sensor y de actuador acorde a la aplicación.

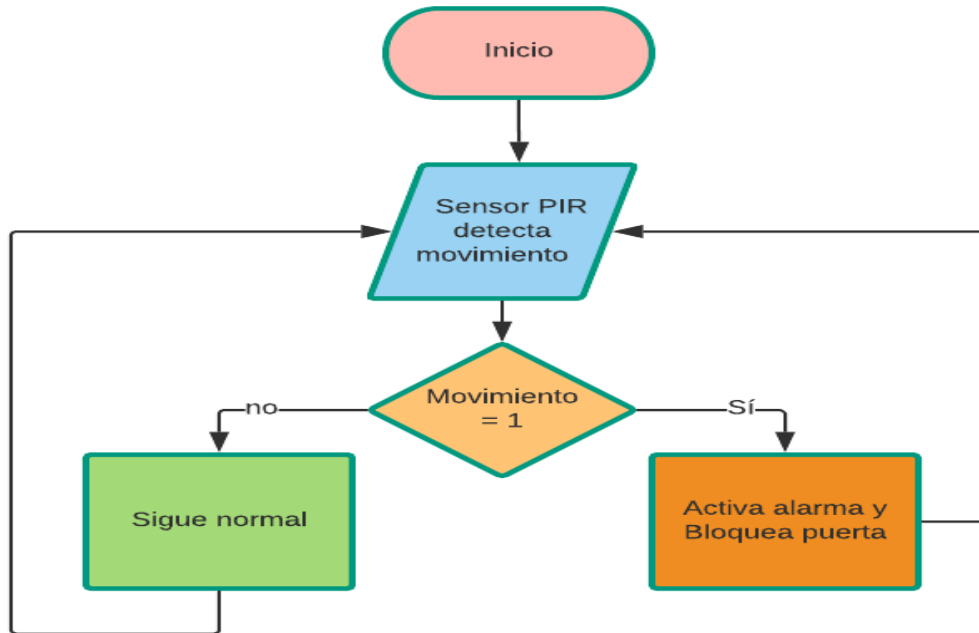


Para el servomotor y la alarma se active y por consecuente realice la acción de bloquear la puerta y emitir la alarma respectivamente, se escogió el Pir HC-SR501, el cual es un sensor de presencia o movimiento capaz de poder detectar la el movimiento.

### 3. Elaborar la lógica de control correspondiente a la aplicación.

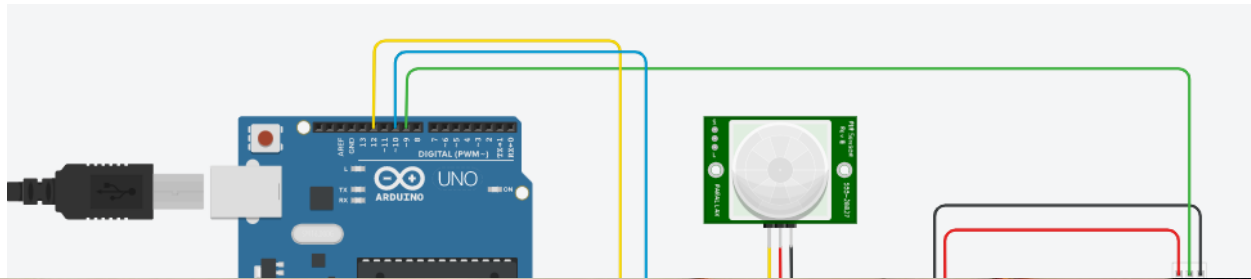
#### Lógica de Control

##### Ejercicio 3

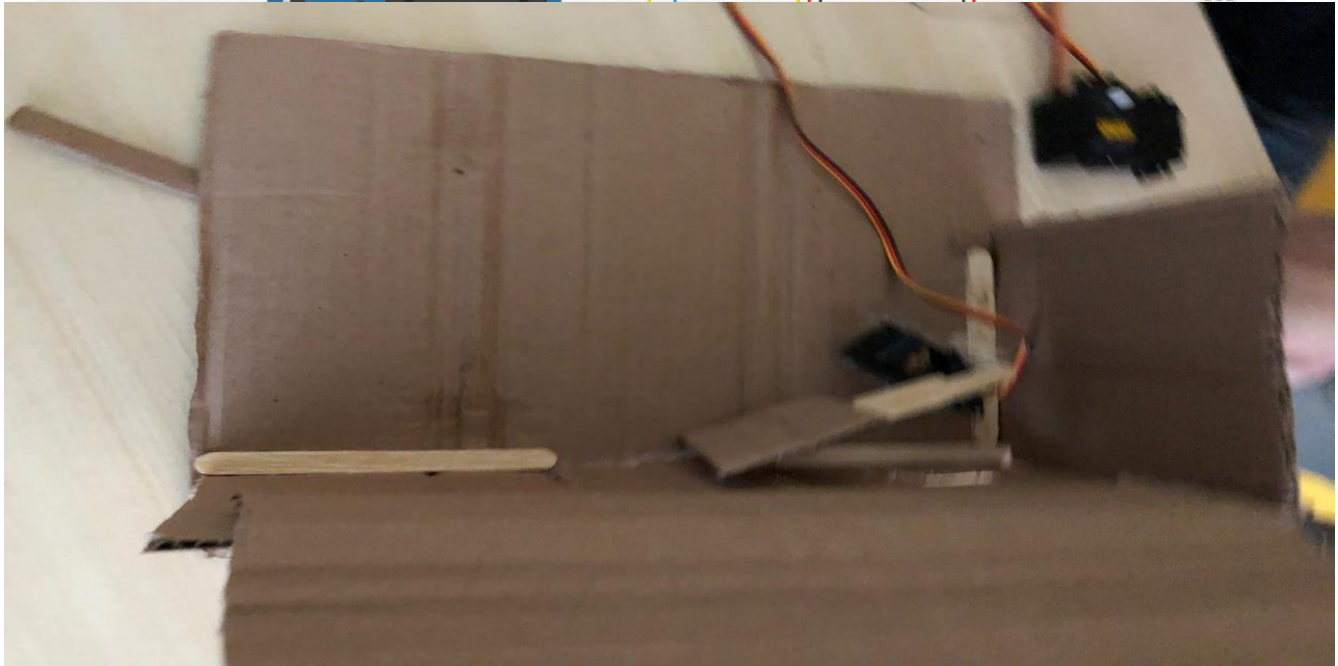


La lógica de control para el ensamblado inicia siempre detectando el movimiento, si el movimiento es igual a uno es decir si existe, se procede a activar el servomotor y por consiguiente bloquear una puerta y activar su alarma en caso contrario se sigue con la medición del PIR.

#### 4. Simular el circuito correspondiente.

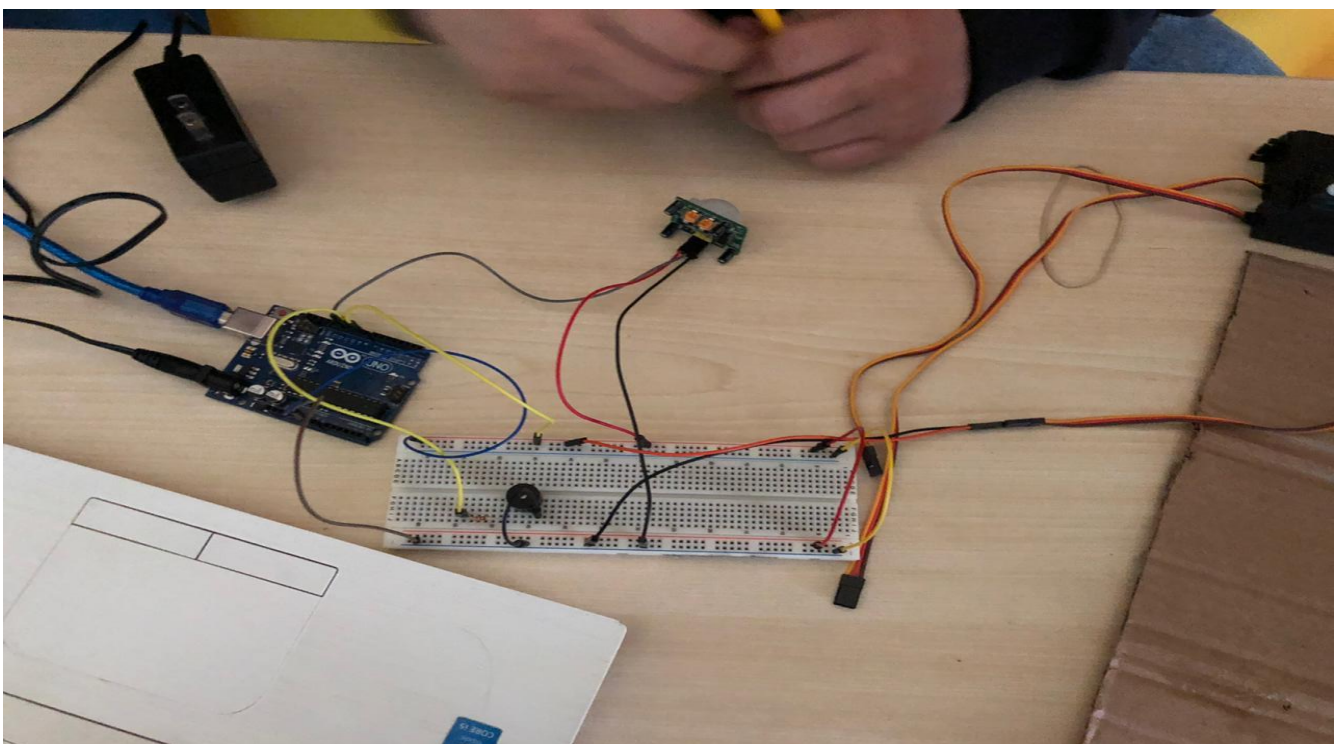


Fue



necesario el simular nuestro circuito en la herramienta TinkerCad, ya que se necesita definir como serán las conexiones de nuestro sistema y observar su comportamiento.

#### 5. Probar el funcionamiento del circuito





Archivo Editar Programa Herramientas Ayuda



```
✓ ↻ 📄 ⬆️ ⬇️
sketch_oct12a$
#include <Servo.h>
Servo myservo; //creamos un objeto servo
int PIRval = 0;

void setup()
{
  myservo.attach(9); // asignamos el pin 9 al servo.
  Serial.begin(9600);
  pinMode(12, INPUT);
  pinMode(10, OUTPUT);
}

void loop()
{
  PIRval = digitalRead(12);
  myservo.write(0);
  if (PIRval == HIGH) {
    digitalWrite(10, HIGH);
    myservo.write(180);
    delay(5000);

  }else{
    digitalWrite(10, LOW);
    myservo.write(0);
    delay(5000);
  }
}
```

Como último paso, se procede a ensamblar nuestro circuito, observando que realiza con éxito lo esperado que en este caso es el bloquear una puerta que al simultaneo active una alarma cuando detecte que el movimiento de un objeto.