INSTITUTO TECNOLÓGICO DE SONORA
Educar para Trascender

Portafolio de evidencias

Integrantes:

Fiol Becheret Jassiel Uldarico

Manríquez Valenzuela Maribel

Navarro Villa Rey David

Rodríguez Nápoles Saul Antonio

Optativa I
Carlo Soto

Guaymas, Sonora

ISW
Ingeniero en Software

Índice:

I. Encender un led……1

II. Semáforo……....4

III. Semáforo con Botón……..7

IV. Led Secuencial……9

V. Sensor PIR……...12

VI. Sensor Sónico…….....13

VII. Motor DC……17

VIII. Domótica……..18

I. Encender un LED

Descripción:

En esta práctica debíamos encender únicamente un led.

Componentes:

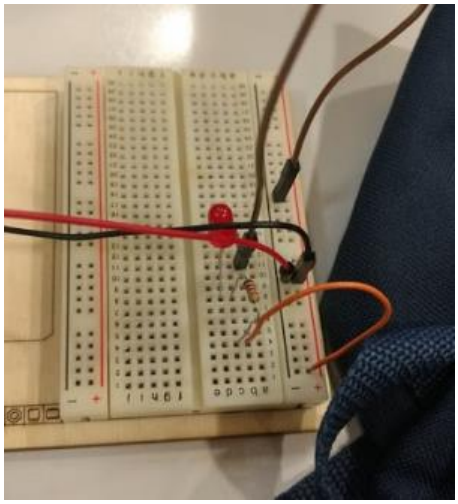
- Arduino uno
- Protoboard
- 1 LED
- Cables

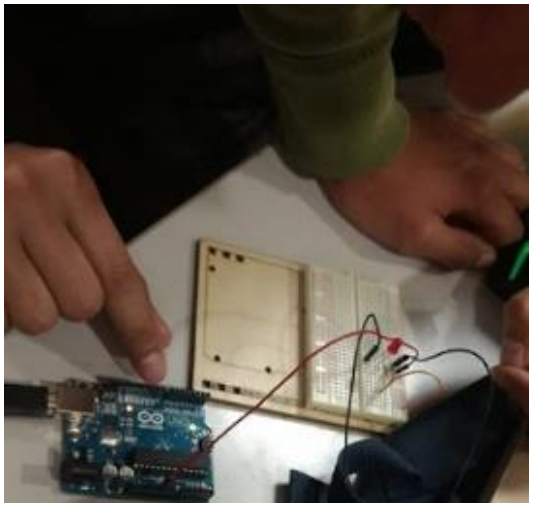
Principios:

En esta práctica definimos un pin y utilizamos un HIGH y LOW para poder encender y apagar el LED.

Procedimiento:

Paso 1: Unimos el arduino de la siguiente manera, para que coincida con la programación hecha.





Paso 2: Programar el código (Adjunto en el siguiente punto)

Paso 3: Compilar el código.

Código:

```
void setup() {
  // put your setup code here, to run once:
  pinMode(13, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13, HIGH);
  delay(1000);
  digitalWrite(13, LOW);
  delay(1000);
}
```

Conclusión:

Como vemos en esta práctica comenzamos haciendo lo más básico, semejante a un “Hola mundo”, pudimos comenzar a entender cómo funcionaba un arduino, así como conocer el lenguaje y el IDE con el que trabajaríamos.

II. Semáforo

Descripción:

En esta práctica nosotros utilizamos el arduino para poder crear un semáforo convencional, es decir; debíamos simular la secuencia del encendido típicamente utilizada, haciendo uso de 3 leds de color: verde, amarillo y rojo.

Componentes:

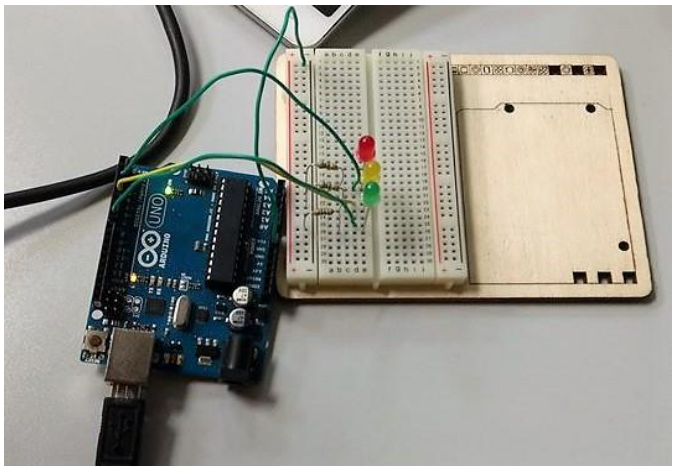
- Arduino uno
- Protoboard
- 3 diodos LED
- Cables

Principios:

Para poder hacer una simulación creíble, se deberá jugar con la longitud del tiempo que tarda. Se deben definir 2 ciclos (Uno largo y otro corto) ya que el tiempo que se invierte para mostrar los colores rojo y verde será más largo que al momento de mostrar el color amarillo.

Procedimiento:

Paso 1: Construir el circuito como se muestra en la siguiente imagen



Paso 2: Programar el código (Adjunto en el siguiente punto)

Paso 3: Compilar el código

Código:

```
int rojo = 7;
int amarillo = 4;
int verde = 2;

void setup() {
  // put your setup code here, to run once:
  pinMode(verde, OUTPUT);
  pinMode(amarillo, OUTPUT);
  pinMode(rojo, OUTPUT);

}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(verde, HIGH);
  delay(2000);
  digitalWrite(verde, LOW);
  delay(500);

  digitalWrite(amarillo, HIGH);
  delay(2000);
  digitalWrite(amarillo, LOW);
  delay(500);

  digitalWrite(rojo, HIGH);
  delay(2000);
  digitalWrite(rojo, LOW);
  delay(500);
}
```

Conclusión:

La realización de esta práctica fue bastante sencilla, ya que anteriormente realizamos una en donde debíamos prender un solo led, sólo debíamos hacer lo mismo, pero incluyendo más accesorios (Resistencias, leds y cables) y en cuanto al código incluir los dos ciclos para definir cuándo deben encender los leds.

III. Semáforo con Botón

Descripción:

En esta práctica se realizó un semáforo que se activaba mediante un botón, siguiendo la función de un semáforo normal.

Componentes:

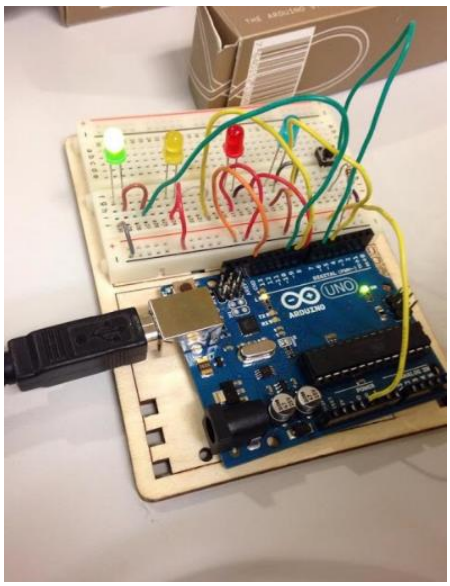
- Arduino uno
- Protoboard
- Botón
- 4 leds
- Cables

Principios:

En esta práctica se jugó con el delay de los leds, además, al tener declarado 3 leds, se usaron dos ciclos, uno alto y otro bajo, para el delay.

Procedimiento:

Paso 1: Unimos el arduino de la siguiente manera, para que coincida con la programación hecha.



Paso 2: Se hace la programación (adjunta en el siguiente paso).

Código:

```
int redLed1=12;
int yellowLed=11;
int greenLed1=10;
int redLed2=7;
int greenLed2=6;
int buttonPin=2;
int state=0;
void setup() {
  // put your setup code here, to run once:
  pinMode(redLed1, OUTPUT);
  pinMode(yellowLed, OUTPUT);
  pinMode(greenLed1, OUTPUT);
  pinMode(redLed2, OUTPUT);
  pinMode(greenLed2, OUTPUT);
   pinMode(buttonPin, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
 digitalWrite(greenLed1, HIGH);
 digitalWrite(redLed2, HIGH);

 state=digitalRead(buttonPin);

 if(state==HIGH){
 digitalWrite(greenLed1, LOW);
 digitalWrite(yellowLed, HIGH);

 delay(1000);

 digitalWrite(yellowLed, LOW);
 digitalWrite(redLed1, HIGH);
 digitalWrite(redLed2, LOW);
 digitalWrite(greenLed2, HIGH);
 delay(200);

 digitalWrite(redLed1, LOW);
 digitalWrite(greenLed2, LOW);
 }

}
```

Conclusión:

Viendo la práctica del semáforo sin botón, y realizar esta es sencilla, ya que, a la práctica anterior únicamente se le tenía que implementar el botón a lo anteriormente hecho.

IV. LED secuencial (PWM)

Descripción:

En la siguiente práctica controlaremos la intensidad con la que se encienden y apagan los leds para que se vea de forma consecutiva, haciendo parecer que la luz viaja de un led a otro.

Componentes:

- Arduino uno
- Protoboard
- 4 diodos LED
- Cables

Principios:

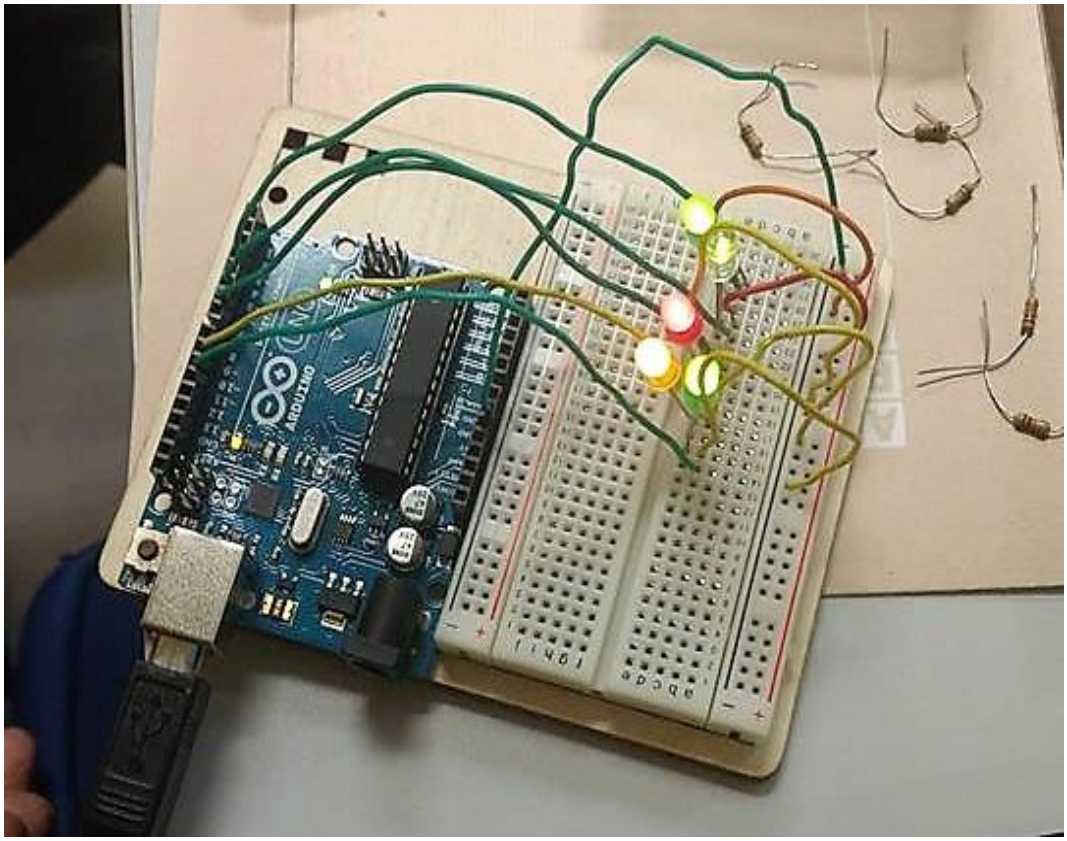
Ya que no se cuenta con un convertidor digital análogo utilizaremos la modulación por ancho o de pulso (PWM) la cual nos servirá para poder modificar la cantidad de energía que se envía.

Necesitamos conectar el led a un pin que permite salida PWM.

Tuvimos que jugar con los valores de intensidad de los leds para que la secuencia fuese visible para el ojo humano.

Procedimiento:

Paso 1: Construir el circuito como se muestra en la siguiente imagen



Paso 2: Programar el código (Adjunto en el siguiente punto)

Paso 3: Compilar el código

Código:

```
int led[5] = {3,5,6,9,10};
int i, j;
int repeat = 10;
void setup() {
  // put your setup code here, to run once:
  for(i=0; i<5; i++)
  pinMode(led[i], OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  for(j=0; j<repeat; j++){
    for(i=0; i<256; i++){
      analogWrite(led[0],i);
    }
    for(i=50; i>-1; i--){
      analogWrite(led[0],i);
      analogWrite(led[1],255-i);
    }
    for(i=0; i<52; i++){
      analogWrite(led[1],255-i);
      analogWrite(led[2],i);
    }
    for(i=50; i>-1; i--){
      analogWrite(led[2],i);
    }


   for(i=0; i<52; i++){
     analogWrite(led[2],i);
   }
   for(i=50; i>-1; i--){
     analogWrite(led[2],i);
     analogWrite(led[1],255-i);
   }
   for(i=0; i<52; i++){
     analogWrite(led[1],255-i);
     analogWrite(led[0],i);
   }
   for(i=50; i>-1; i--){
     analogWrite(led[0],i);
   }

 }

 for(i=0; i<repeat; i++){
   analogWrite(led[0],random(0,256));
   delay(random(0,501));
   analogWrite(led[1],random(0,256));
   delay(random(0,501));
   analogWrite(led[2],random(0,256));
   delay(random(0,501));
   analogWrite(led[3],random(0,256));
   delay(random(0,501));
   analogWrite(led[4],random(0,256));


     delay(random(0,501));

   }
 }
```

Conclusión:

La práctica fue sencilla, sólo debíamos hacer lo mismo, pero incluyendo más accesorios incluir los dos ciclos para definir cuándo deben encender los leds.

V. Sensor PIR

Descripción:

Para esta práctica utilizamos un sensor que conectado de cierta manera a una placa de Arduino y mediante algunas líneas de código es capaz de detectar cualquier movimiento.

Componentes:

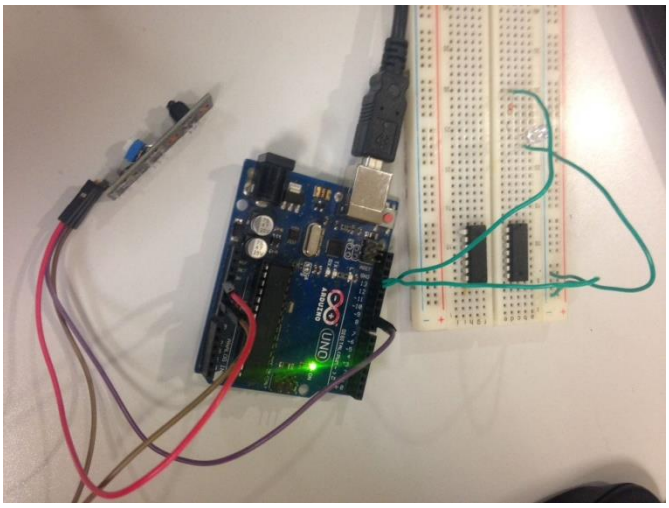
- Arduino UNO
- Sensor PIR
- 1 Led
- Cables

Principios:

Se programó nuestra placa de Arduino para que al momento de que el sensor detectara algún movimiento, encendiera un led conectado a la protoboard y nos imprimiera en consola “Detectando algo”.

Procedimiento:

Paso 1: Se conectó el Arduino y los distintos componentes de la siguiente manera:



Paso 2: Se escribió el código necesario para el correcto funcionamiento de los componentes

Paso 3: Se ejecutó el código

Código:

```
const int sensorPin = 9;
int ledPin = 13;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(sensorPin, INPUT);
  pinMode(ledPin, OUTPUT);

}

void loop() {
  // put your main code here, to run repeatedly:
int value = 0;
value = digitalRead(sensorPin);

if (value == LOW){
  Serial.println("Detectando algo");
  digitalWrite(ledPin,HIGH);

}else {
  digitalWrite(ledPin, LOW);
  }
delay(250);

}
```

Conclusión: La práctica estuvo en verdad muy sencilla, pero destaca por encima de las anteriores por ser nuestra primera experiencia con sensores de movimiento como lo es el PIR.

VI. Sensor Ultrasónico

Descripción:

Durante esta práctica se realizó mediante el uso de Arduino un sensor que detecta el movimiento de cualquier tipo de objeto a una distancia de aproximadamente 50 cm, al momento de detectarlo enciende un led y una vez pasados 400ms te muestra en consola a qué distancia está dicho objeto.

Componentes:

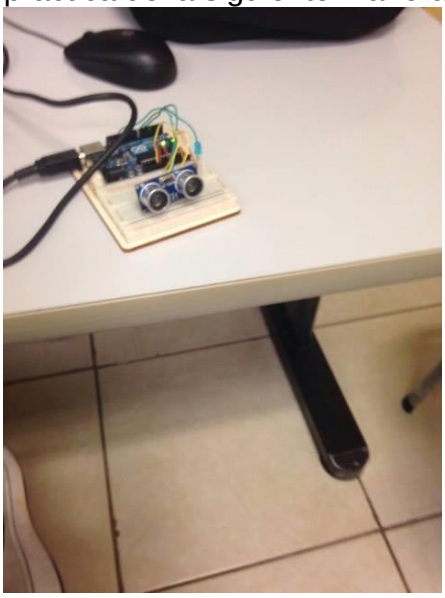
- Arduino uno
- Sensor sónico HC-SR04
- 1 leds
- Cables

Principios:

Se utilizó un sensor sónico HC-SR04 para dtectar movimiento en un rango de 50 cm.

Procedimiento:

Paso 1: Unimos el Arduino y los componentes necesarios para realizar la práctica de la siguiente manera:



Paso 2: Se escribe el código necesario(Mismo que se adjuntará en el siguiente punto) para hacer funcionar los componentes.

Paso 3: Subir a la placa y compilar el código

Código:

```
#define Pecho 6
#define Ptrig 7
long duracion, distancia;
long distanciaPermitida = 50; //centimetros

void setup() {
  Serial.begin (9600);
  pinMode(Pecho, INPUT);     // define el pin 6
  pinMode(Ptrig, OUTPUT);    // define el pin 7
  pinMode(13, 1);            // led
  }

void loop() {

  generarPulsaciones();

  calcularDistacia();
}

void generarPulsaciones(){
  digitalWrite(Ptrig, LOW);
  delayMicroseconds(2);
  digitalWrite(Ptrig, HIGH);   // genera el puls
  delayMicroseconds(10);
  digitalWrite(Ptrig, LOW);
}

void calcularDistacia(){
  duracion = pulseIn(Pecho, HIGH);  //recibe o lee las pul
  distancia = (duracion/2) / 29;           // calcula la

  if (distancia >= 500 || distancia <= 0){  // si la dista
    Serial.println("------fuera del rango-------");
  }
  else {
    Serial.print(distancia);          // envia el valor d
    Serial.println("cm");
    digitalWrite(13, 0);              // se apaga el led
  }

  encenderLed(distancia);
}

void encenderLed(long distancia){

  if (distancia <= distanciaPermitida && distancia >= 1){
    digitalWrite(13, 1);                    // se enciend
    Serial.print("se detecto algo a ");
    Serial.print(distancia);
    Serial.println(" cm de distancia ");        // envia
  }
  delay(400);  // espera 400ms para que se logre ver la di
}
```

Conclusión:
Realizar esta práctica fue una experiencia que consideramos nueva, puesto que después de venir de encender leds en las prácticas anteriores, en esta se interactúa con algo más a doc a lo que es el IoT.

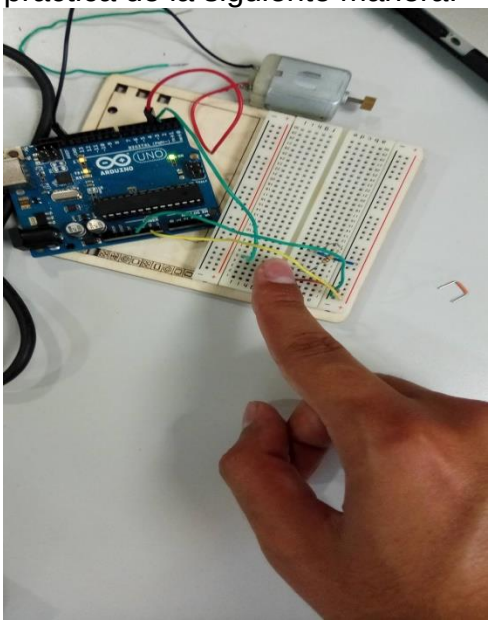
VII. Motor DC

Descripción:
En este proyecto se utilizó Arduino para hacer que un motor se moviera presionando un botón.

Componentes:
- Arduino uno
- Botón
- Motor DC

Principios:
Conectamos y programamos la placa de Arduino de manera que al presionar un botón el Motor DC se moviera y en consola nos imprimiera la palabra “Prende”.

Paso 1: Conectamos el Arduino y los componentes necesarios para realizar la práctica de la siguiente manera:



Paso 2: Se escribe el código necesario para hacer funcionar los componentes.

Paso 3: Subir a la placa y compilar el código

Código:

```
const int motorPin = 3;
const int botonPin = 2;
int estadoBtn = 0;
void setup() {
  Serial.begin(9600);
  pinMode(motorPin,  OUTPUT) ;
  pinMode(botonPin, INPUT);


}

void loop() {
  estadoBtn = digitalRead(botonPin);

  if( estadoBtn == HIGH){
  analogWrite(motorPin,255);
  Serial.println("Prende");
  }
  else{
    analogWrite(motorPin, 0);
    }

}
```

Conclusión: Al principio tuvimos dificultades por el “comportamiento errático” que presentaba el botón al momento de hacerlo funcionar, pero pudimos solventar el problema guiándonos con algunos tutoriales en Internet, excelente práctica.

VIII. Domótica

Descripción:

En este proyecto se hicieron varias prácticas, utilizando los distintos sensores los cuales interactúan entre sí.

Componentes:

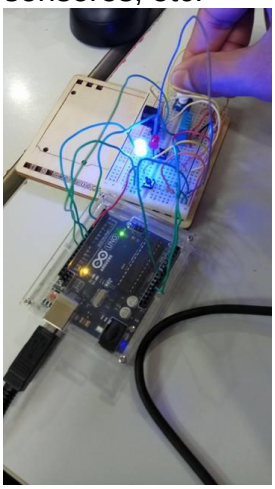
- Arduino uno
- Protoboard
- 2 diodos LED
- Cables
- Potenciómetro
- Sensor de temperatura
- Botón
- Buzzer

Principios:

En este proyecto, todo interactúa entre sí, en donde dependiendo de las condiciones, se activa la alarma o no, a cierta hora del día se activa junto con el buzzer para hacer sonido, etc.

Procedimiento:

Paso 1: Unir el arduino de la siguiente manera, conectando todos los cables, sensores, etc.



Paso 2: Realizar la programación correspondiente de cada sensor.

Paso 3: Compilar el código escrito anteriormente.

Código:

```
// Example testing sketch for various DHT humidity/temperature sensors
// Written by ladyada, public domain

#include "DHT.h"

#define DHTPIN 3     // what digital pin we're connected to
const int ledAzul = 5; //Pin del led azul (movimiento)
const int ledRojo = 8; //Pin de led rojo (Alarma)
const int btnAzul = 7; //Pin del boton del led azul (movimiento);
const int btnRojo = 9; //Pin del boton del led Rojo (Alarma)
const int pinPot = 0;
int valorPot = 0;
int valorBtn; //Valor del boton que se utilizara en el if
int alarma = 10; //Pin de la alarma;
int salidaBoton = 0; //leer salida del boton
int activado = 0;

// Uncomment whatever type you're using!
//#define DHTTYPE DHT11   // DHT 11
#define DHTTYPE DHT22   // DHT 22  (AM2302), AM2321
//#define DHTTYPE DHT21   // DHT 21 (AM2301)

// Connect pin 1 (on the left) of the sensor to +5V
// NOTE: If using a board with 3.3V logic like an Arduino Due connect pin 1
// to 3.3V instead of 5V!
// Connect pin 2 of the sensor to whatever your DHTPIN is
// Connect pin 4 (on the right) of the sensor to GROUND
```

```
// Connect a 10K resistor from pin 2 (data) to pin 1 (power) of the sensor

// Initialize DHT sensor.
// Note that older versions of this library took an optional third parameter to
// tweak the timings for faster processors.  This parameter is no longer needed
// as the current DHT reading algorithm adjusts itself to work on faster procs.
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  pinMode(ledAzul, OUTPUT);          // Ponemos el pin del LED como salida
  pinMode(btnAzul, INPUT);           // Ponemos el pin del botón como entrada
  pinMode(pinPot, INPUT);
  Serial.println("DHTxx test!");
  dht.begin();
}

void loop() {

    valorPot = analogRead(pinPot);
    digitalWrite(ledRojo, LOW);
    digitalWrite(ledAzul, LOW);

    if(valorPot == 0) { //si el potenciometro llega a 0
    //activar sensor movimiento
    Serial.println("MOVIMIENTO (NOCHE)");
    activarMovimiento();
    }
```

```
// Reading temperature or humidity takes about 250 milliseconds!
// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
float h = dht.readHumidity();
// Read temperature as Celsius (the default)
float t = dht.readTemperature();
// Read temperature as Fahrenheit (isFahrenheit = true)
float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try again).
if (isnan(h) || isnan(t) || isnan(f)) {
  Serial.println("Failed to read from DHT sensor!");
  return;
}

// Compute heat index in Fahrenheit (the default)
float hif = dht.computeHeatIndex(f, h);
// Compute heat index in Celsius (isFahreheit = false)
float hic = dht.computeHeatIndex(t, h, false);

Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperature: ");
Serial.print(t);
Serial.print(" *C ");
Serial.print(f);
Serial.println(" *F\t");
```

```
    salidaBoton = digitalRead(btnAzul);
    if(salidaBoton == HIGH){
      activado = 1;
    }
    else {
      activado = 0;
    }

    if(activado == 1){ //si se activa el boton la alarma se desactiva
      Serial.println("ALARMA DESACTIVADA");
      digitalWrite(ledAzul, HIGH);
      noSonar();
      break;
    }
    else if (activado == 0 && i >= 50) { //si no se activa el boton la alarma empieza a sonar
      Serial.println("ALARMA ACTIVADA WIU WIU WIUW");
      sonar();
      digitalWrite(ledRojo, HIGH);
      delay(2000);
    }

    //Serial.println("HAY UN INTRUSO CHAVO");
    //delay(100);
  }
}



void sonar() {
  tone(alarma, 20000, 2000);
}

void noSonar(){
  noTone(alarma);
  delay(100);
}
```

Conclusión:

Esta práctica nos resultó un poco difícil, el hacer que todo trabaje en conjunto fue lo que más se nos dificulto, fuera de esto, la programación se pudo hacer realizando la investigación correspondiente.