



C3.8 Programación Microcontrolador NodeMCU ESP32

Arduino y sensor de tacto integrado al NodeMCU ESP32



Instrucciones

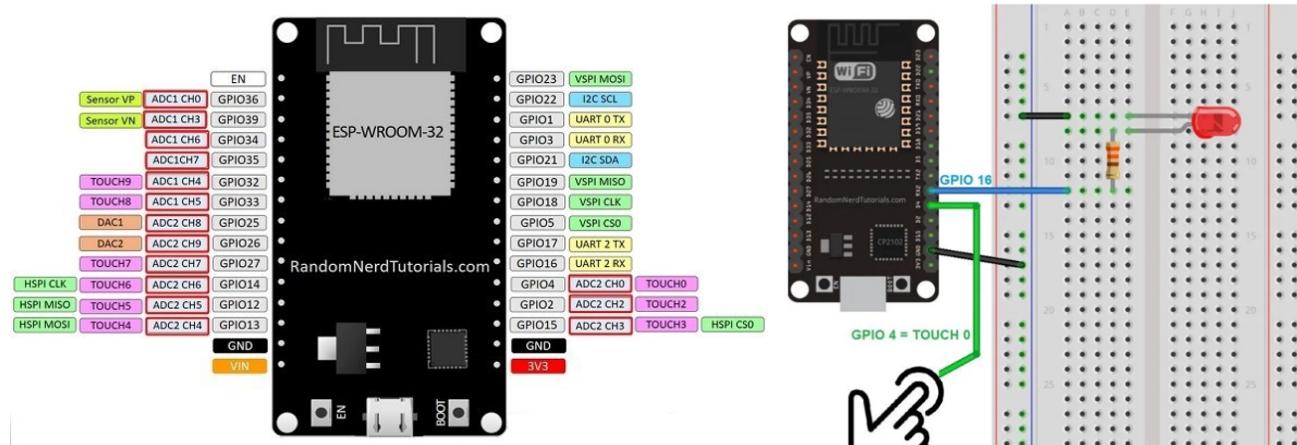
- De acuerdo con la información presentada por el asesor referente al tema, desarrollar lo que se indica dentro del apartado siguiente.
- Toda actividad o reto se deberá realizar utilizando el estilo **MarkDown con extension .md** y el entorno de desarrollo VSCode, debiendo ser elaborado como un documento **single page**, es decir si el documento cuanta con imágenes, enlaces o cualquier documento externo debe ser accedido desde etiquetas y enlaces.
- Es requisito que el archivo .md contenga una etiqueta del enlace al repositorio de su documento en Github, por ejemplo **Enlace a mi GitHub**
- Al concluir el reto el reto se deberá subir a github el archivo .md creado.
- Desde el archivo **.md** se debe exportar un archivo **.pdf** con la nomenclatura **C3.8_NombreAlumno_Equipo.pdf**, el cual deberá subirse a classroom dentro de su apartado correspondiente, para que sirva como evidencia de su entrega; siendo esta plataforma **oficial** aquí se recibirá la calificación de su actividad por individual.
- Considerando que el archivo .pdf, fue obtenido desde archivo .md, ambos deben ser idénticos y mostrar el mismo contenido.
- Su repositorio ademas de que debe contar con un archivo **readme.md** dentro de su directorio raíz, con la información como datos del estudiante, equipo de trabajo, materia, carrera, datos del asesor, e incluso logotipo o imágenes, debe tener un apartado de contenidos o indice, los cuales realmente son ligas o **enlaces a sus documentos .md**, *evite utilizar texto* para indicar enlaces internos o externo.
- Se propone una estructura tal como esta indicada abajo, sin embargo puede utilizarse cualquier otra que le apoye para organizar su repositorio.

```
| readme.md  
| | blog  
| | | C3.1_TituloActividad.md  
| | | C3.2_TituloActividad.md  
| | | C3.3_TituloActividad.md  
| | | C3.4_TituloActividad.md  
| | | C3.5_TituloActividad.md  
| | | C3.6_TituloActividad.md  
| | | C3.7_TituloActividad.md  
| | | C3.8_TituloActividad.md  
| | img  
| | docs  
| | | A3.1_TituloActividad.md  
| | | A3.2_TituloActividad.md
```

Desarrollo

1. Basado en el siguiente circuito y ensamblarlo, utilizando alguno de los simulados propuesto, explicando el resultado que se desea obtener del mismo.

ESP32 DEVKIT V1 - DOIT



2. Analice y escriba el programa que se muestra a continuación.

```
// set pin numbers
const int touchPin = 4;
const int ledPin = 16;

// change with your threshold value
const int threshold = 20;
// variable for storing the touch pin value
int touchValue;

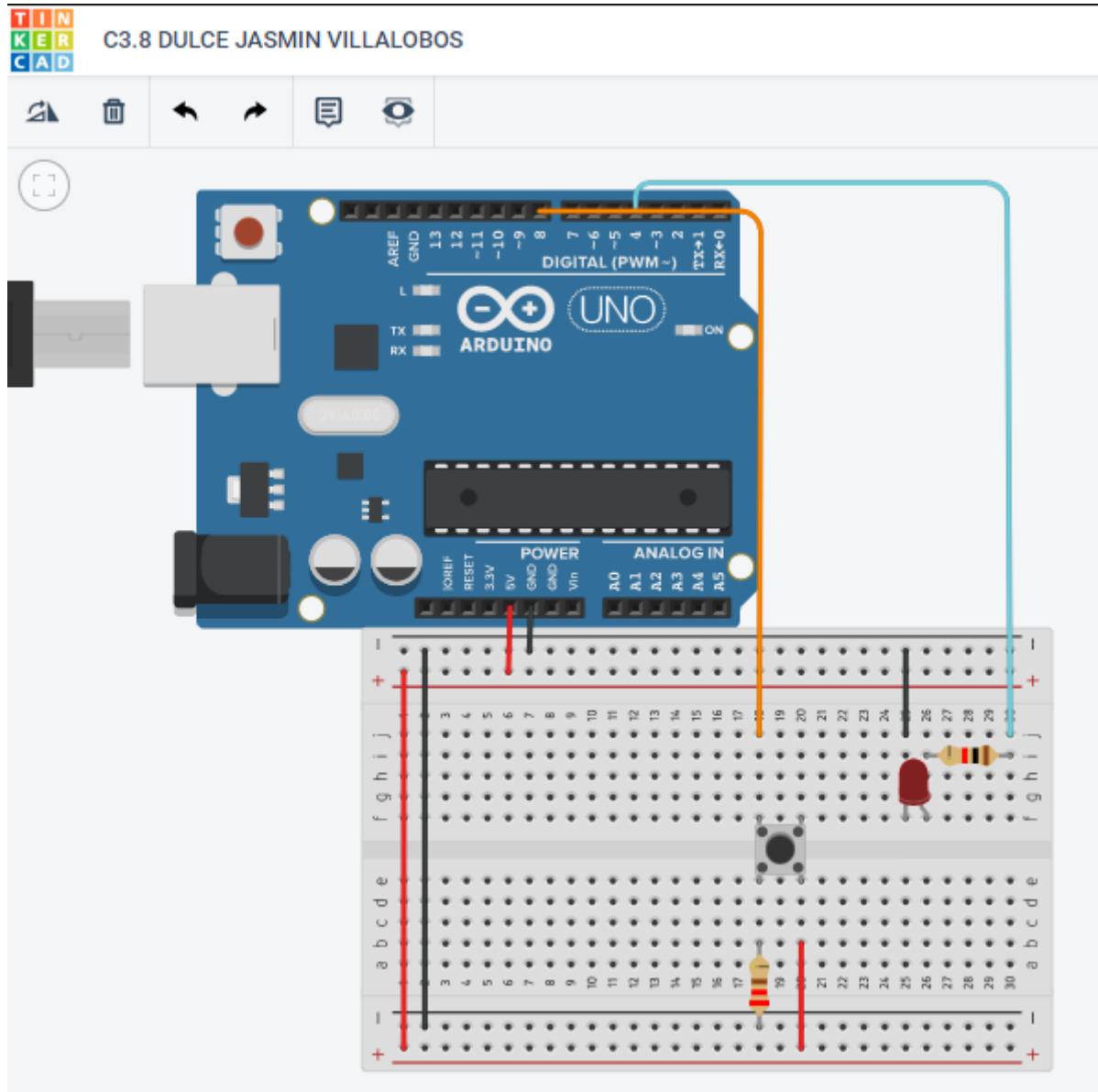
void setup(){
    Serial.begin(115200);
    delay(1000); // give me time to bring up serial monitor
    // initialize the LED pin as an output:
    pinMode (ledPin, OUTPUT);
}

void loop(){
    // read the state of the pushbutton value:
    touchValue = touchRead(touchPin);
    Serial.print(touchValue);
    // check if the touchValue is below the threshold
    // if it is, set ledPin to HIGH
    if(touchValue < threshold){
        // turn LED on
        digitalWrite(ledPin, HIGH);
        Serial.println(" - LED on");
    }
    else{
        // turn LED off
        digitalWrite(ledPin, LOW);
        Serial.println(" - LED off");
    }
    delay(500);
}
```

Fuente de consulta: [Random Nerd Tutorials](#)

3. Inserte aquí las imágenes que considere como evidencias para demostrar el resultado obtenido.

Evidencia en el simulador, para hacer la parte de touch como no era posible, utilice un boton.
Asi quedo el circuito.

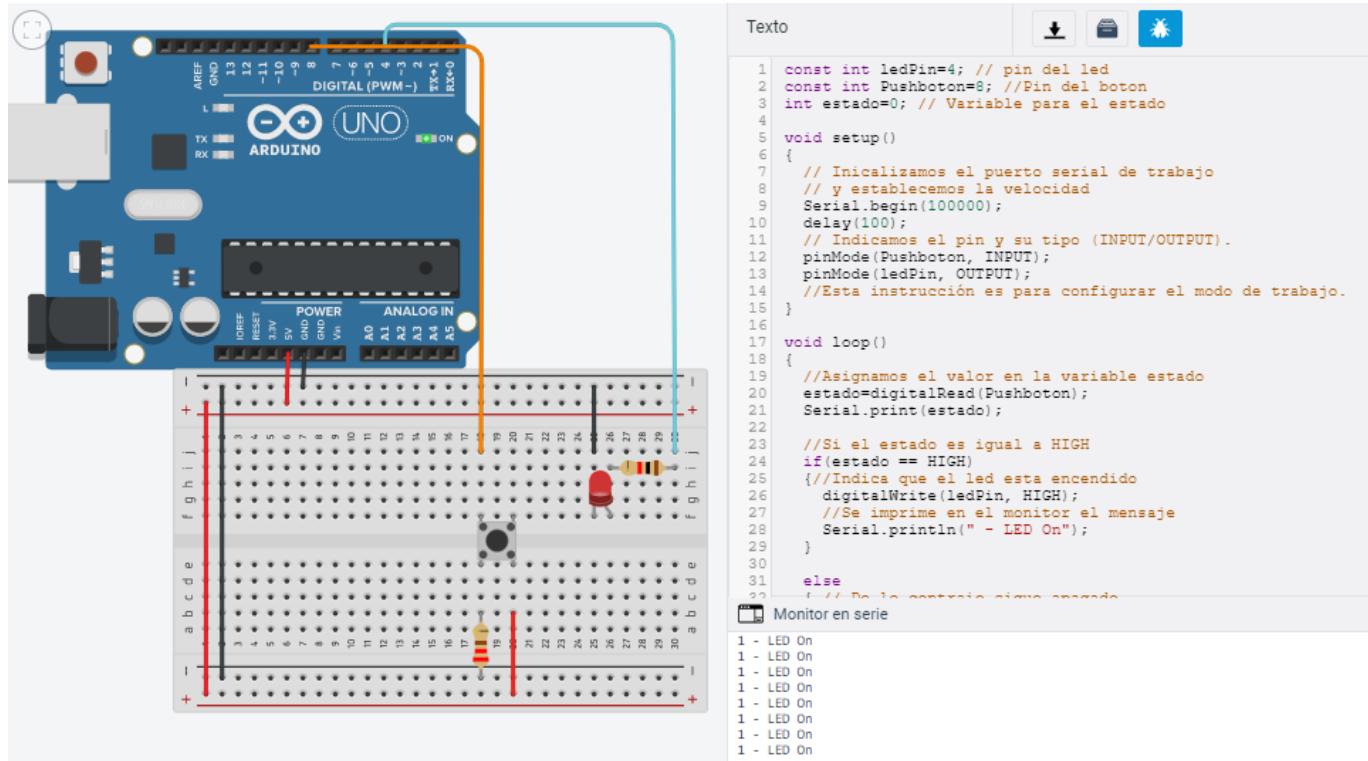


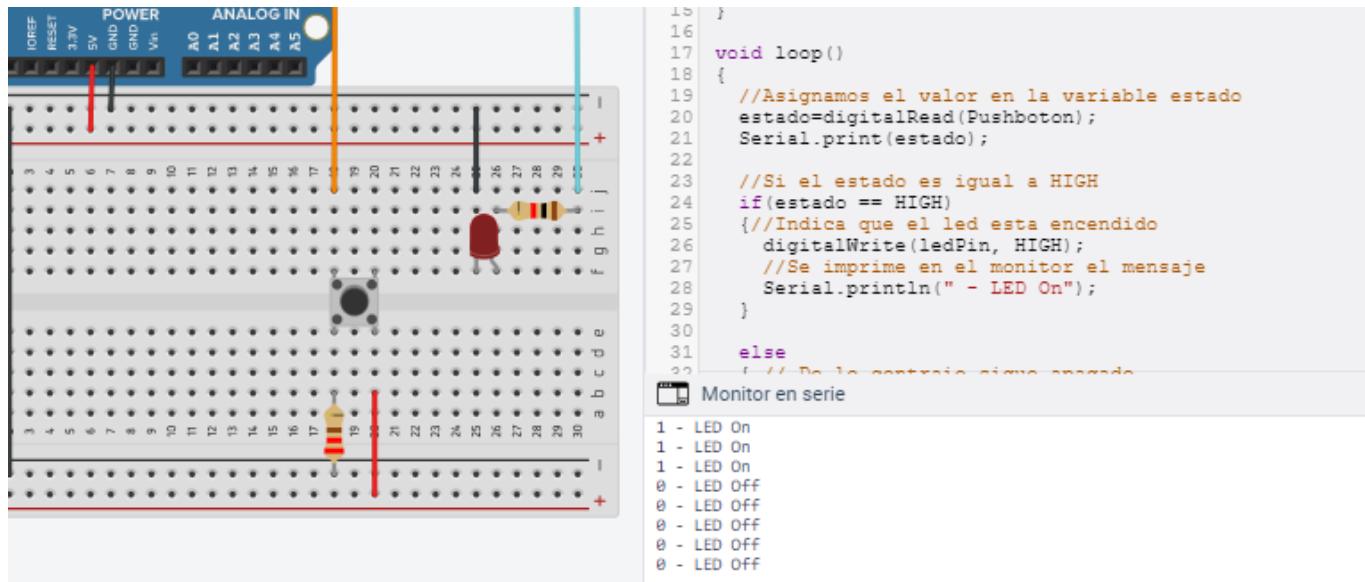
```

1 const int ledPin=4; // pin del led
2 const int Pushboton=8; //Pin del boton
3 int estado=0; // Variable para el estado
4
5 void setup()
{
7 // Inicializamos el puerto serial de trabajo
8 // y establecemos la velocidad
9 Serial.begin(100000);
10 delay(100);
11 // Indicamos el pin y su tipo (INPUT/OUTPUT).
12 pinMode(Pushboton, INPUT);
13 pinMode(ledPin, OUTPUT);
14 //Esta instrucción es para configurar el modo de trabajo.
15 }
16
17 void loop()
{
19 //Asignamos el valor en la variable estado
20 estado=digitalRead(Pushboton);
21 Serial.print(estado);
22
23 //Si el estado es igual a HIGH
24 if(estado == HIGH)
25 //Indica que el led esta encendido
26 digitalWrite(ledPin, HIGH);
27 //Se imprime en el monitor el mensaje
28 Serial.println(" - LED On");
29 }
30
31 else
32 { // De lo contrario sigue apagado
33 digitalWrite(ledPin, LOW);
34 // Por lo que al comenzar solo imprime que esta apagado
35 Serial.println(" - LED Off");
36 }
37 delay(500);
38 }

```

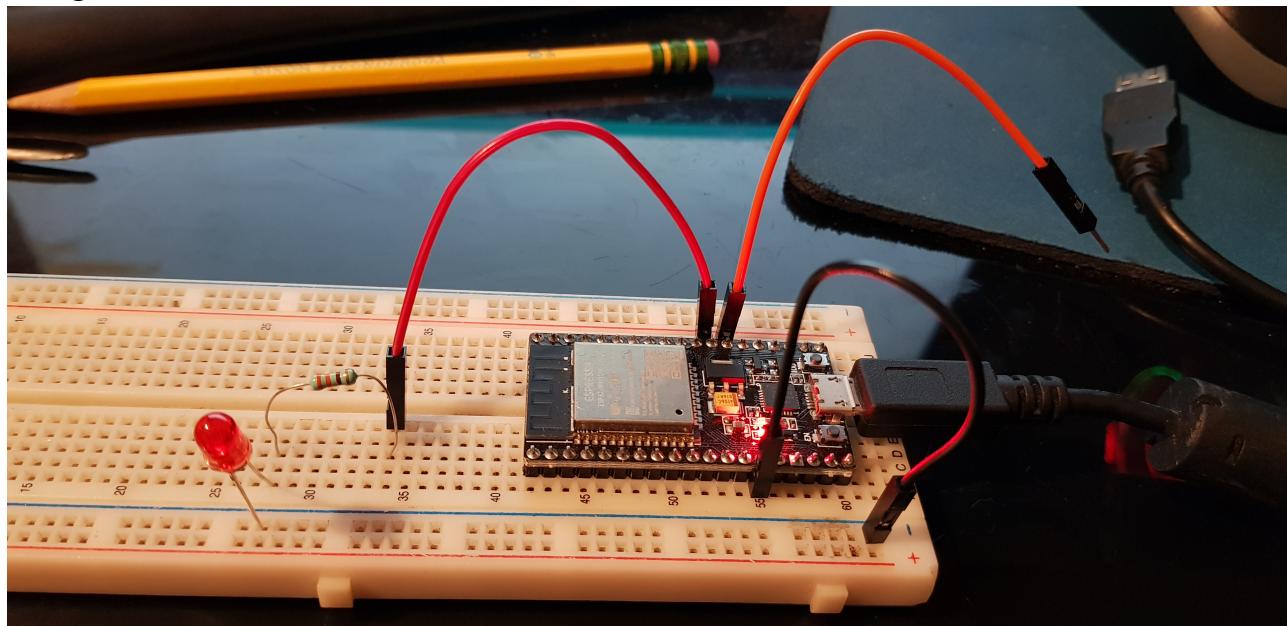
En la primera vemos que el led esta encendido y en la pantalla se imprime 1 LED ON, en la segunda imagen esta apagado, por lo que el mensaje que imprime es 0 LED OFF.

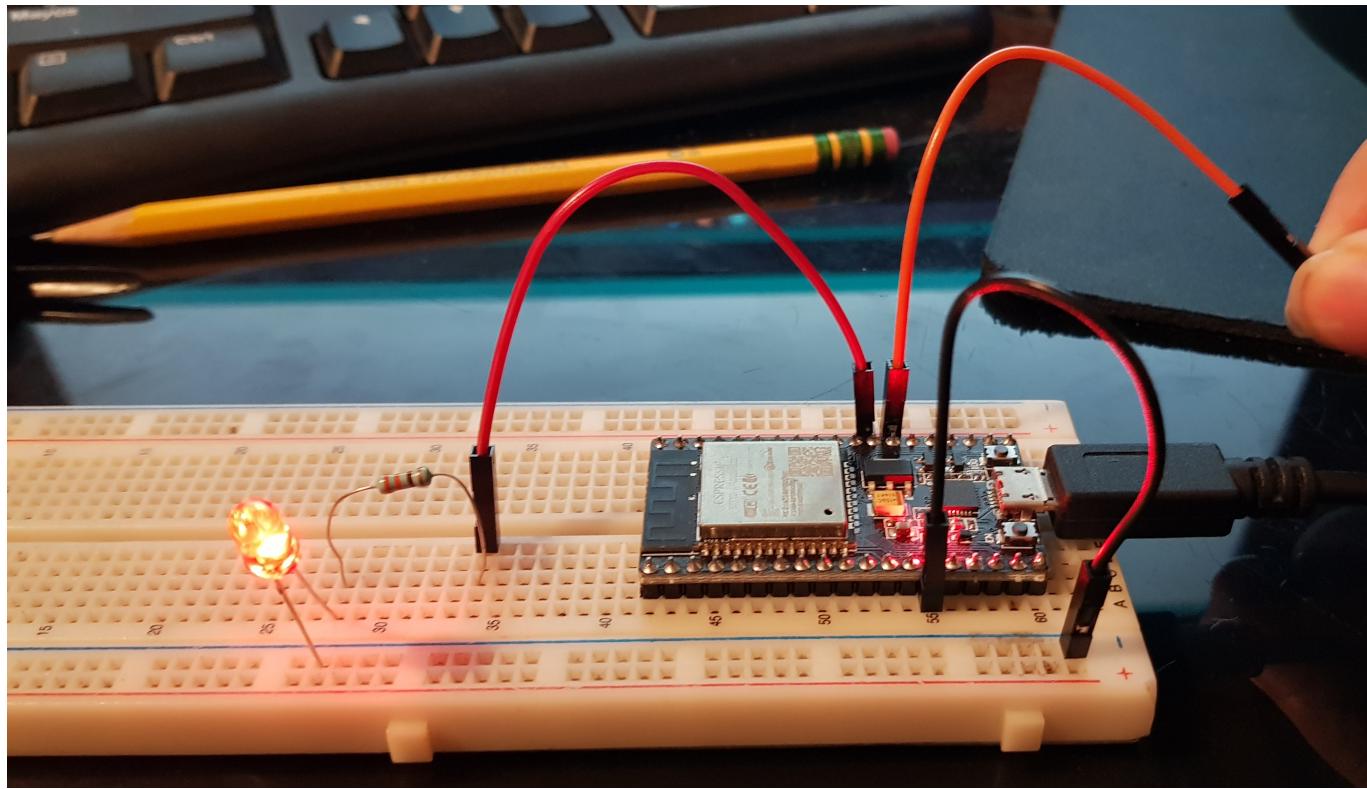




4. Evidencia proporcionada por el compañero con los componentes:

Venegas Medina Jose Alfredo:





C3.8 Arduino 1.8.12

Archivo Editar Programa Herramientas Ayuda



C3.8 §

```
//Se declaran los pines GPIO16 para el LED u GPIO4 para el touch
const int touchPin = 4;
const int ledPin = 16;

//Cambiar con el valor umbral
const int threshold = 20;
//Variable para almacenar el valor del pin touch
int touchValue;

void setup() {
  //Iniciar comunicacion con el monitor en serie
  Serial.begin(115200);
  delay(1000);
  //Asignamos el LED como salida
  pinMode(ledPin,OUTPUT);

}

void loop() {
  //Lee el estado del pin touch
  touchValue = touchRead(touchPin);
  //Se imprime el valor obtenido del pin touch
  Serial.print(touchValue);

  //Si el valor del pin touch es menor al umbral, se prende el led e imprime "LED encendido"
  if(touchValue < threshold){
    digitalWrite(ledPin, HIGH);
    Serial.println("- LED on");
  }

  //Si no, entonces el LED se apaga e imprime "LED apagado"
  else{
    digitalWrite(ledPin, LOW);
    Serial.println("- LED off");
  }

  delay(500);
}
```

COM4

```
19:18:29.583 -> 61- LED off
19:18:30.058 -> 59- LED off
19:18:30.566 -> 57- LED off
19:18:31.076 -> 56- LED off
19:18:31.554 -> 56- LED off
19:18:32.064 -> 35- LED off
19:18:32.574 -> 18 - LED on
19:18:33.053 -> 17 - LED on
19:18:33.563 -> 15 - LED on
19:18:34.074 -> 14 - LED on
19:18:34.551 -> 13 - LED on
19:18:35.063 -> 13 - LED on
19:18:35.573 -> 59- LED off
19:18:36.083 -> 60- LED off
19:18:36.557 -> 60- LED off
19:18:37.066 -> 9 - LED on
19:18:37.579 -> 60- LED off
19:18:38.055 -> 60- LED off
19:18:38.565 -> 60- LED off
19:18:39.076 -> 60- LED off
```

 Autoscroll Mostrar marca temporal

 Rubrica

Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	20
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	80

 ENLACE - MI GITHUB