

Proyecto final Curso IoT

Control domótico: encendido-apagado de las luces y de la calefacción del hogar desde cualquier lugar

1. Objetivo:

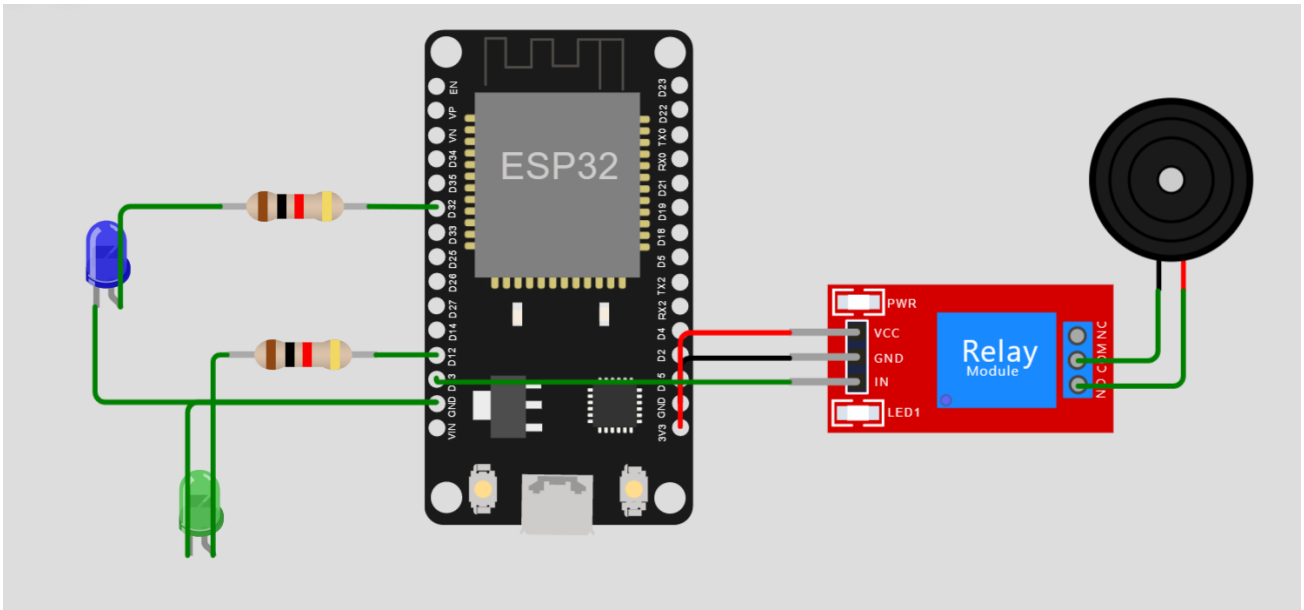
Desarrollar un proyecto básico de IoT con el equipamiento y aplicaciones vistas a lo largo del curso.

2. Introducción:

Aplicación/ejemplo desarrollado:

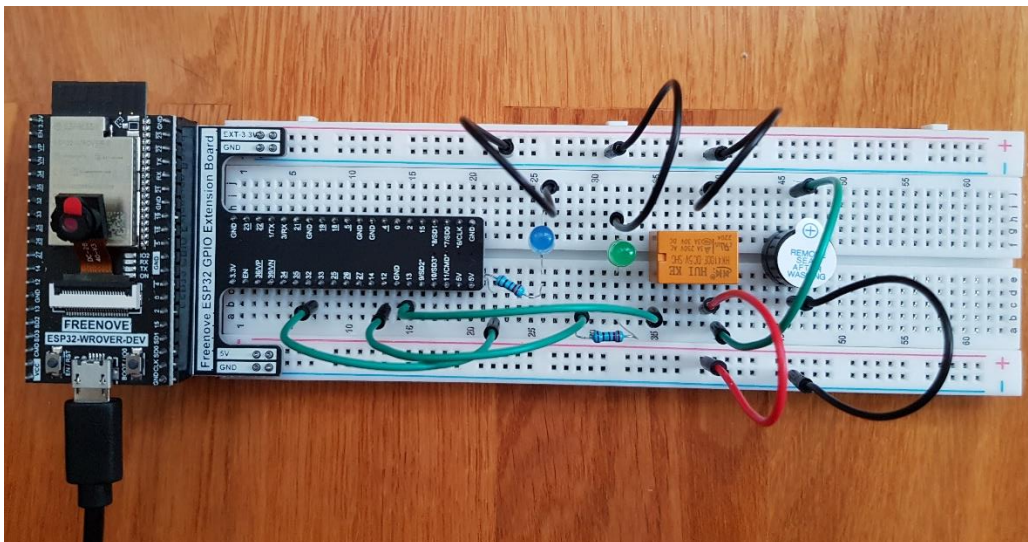
- Control domótico que podría interesar a cualquier persona.
- Control de luces: encendido y apagado en cualquier momento y desde cualquier lugar.
 - Por simplicidad las lámparas son dos leds.
- Control de la calefacción (también podrían ser las persianas...): encendido y apagado.
 - La calefacción es simulada a través de un zumbador
- El circuito se puede ampliar:
 - Incluyendo un sensor de temperatura para tener información sobre la necesidad o no de encender la calefacción
 - Con el encendido y apagado de algún tipo de motor, etc.
- Para la interconexión se utiliza el bróker Mosquitto
- El control se hace desde unos paneles de control:
 - Uno disponible en la App instalada en el móvil
 - Y el otro en el dashboard creado en Node-Red, para control desde el portátil
- Conexiones a través de WiFi (hogar) y a través del móvil o del WiFi del lugar de trabajo, lugares de ocio...

3. Esquema físico del circuito



En el montaje realizado:

- El microcontrolador es el ESP32 Wrover Dev
- El relé es el incluido en el kit, que se alimenta a 5V (salida disponible en nuestro microcontrolador). Conexiones algo distintas, ver Foto.
- Las resistencias utilizadas son de 220 Ω



4. Programa realizado en el Visual Studio Code

```
#include <Arduino.h>
#include <WiFi.h>
#include <PubSubClient.h>
```

```

#include <Wire.h>

// Red WiFi
const char* ssid      = "xxxx";
const char* password = "xxxx";

// Broker
const char* mqttServer = "test.mosquitto.org";
const int  mqttPort   = 1883;

// Se definen varios temas: para control calefaccion y control de cada lampara
const char* tema = "Calefaccion";
const char* tema_led_verde = "Led Verde";
const char* tema_led_azul = "Led Azul";

WiFiClient ClienteWiFi;
PubSubClient clienteMQTT(ClienteWiFi);

int pinRele = 13;
int pinLedVerde = 12;
int pinLedAzul = 32;

void setup() {
  pinMode(pinRele, OUTPUT);
  digitalWrite(pinRele, LOW);

  pinMode(pinLedVerde, OUTPUT);
  digitalWrite(pinLedVerde, LOW);
  pinMode(pinLedAzul, OUTPUT);
  digitalWrite(pinLedAzul, LOW);

  // Start serial
  Serial.begin(9600);
  delay(10);

  // Conectando con la red WiFi
  Serial.println();
  Serial.print("Conectando a " + String(ssid) + " ");

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
    Serial.println();
  }
  Serial.println("Conectado");
}

```

```

// Indicamos el Broker con el que nos comunicamos
clienteMQTT.setServer(mqttServer, mqttPort);
}

// Funcion que se ejecuta cuando se recibe un mensaje de alguno de los temas
void callback(char* tema_recibido, byte* contenido, unsigned int longitudContenido)
{
    String mensaje = "";
    Serial.print("Ha llegado un mensaje de: ");
    Serial.print(tema_recibido);
    Serial.print(" indicando: ");
    for (int i = 0; i < longitudContenido; i++) {
        mensaje = mensaje + (char)contenido[i];
        Serial.print((char)contenido[i]);
    }
    Serial.println();

    // Activa o desactiva el rele
    if(mensaje == "ON Calefaccion") {
        digitalWrite(pinRele, HIGH);
        clienteMQTT.publish(tema, "ON_CONFIRMADO");
    }
    else if(mensaje == "OFF Calefaccion") {
        digitalWrite(pinRele, LOW);
        clienteMQTT.publish(tema, "OFF_CONFIRMADO");
    }

    // Enciende o apaga las luces
    if(mensaje == "ON Led Verde") {
        digitalWrite(pinLedVerde, HIGH);
        clienteMQTT.publish(tema_led_verde, "Luz Verde ON");
    }
    else if(mensaje == "OFF Led Verde") {
        digitalWrite(pinLedVerde, LOW);
        clienteMQTT.publish(tema_led_verde, "Luz Verde OFF");
    }

    if(mensaje == "ON Led Azul") {
        digitalWrite(pinLedAzul, HIGH);
        clienteMQTT.publish(tema_led_azul, "Luz Azul ON");
    }
    else if(mensaje == "OFF Led Azul") {
        digitalWrite(pinLedAzul, LOW);
        clienteMQTT.publish(tema_led_azul, "Luz Azul OFF");
    }
}

void loop() {

```

```

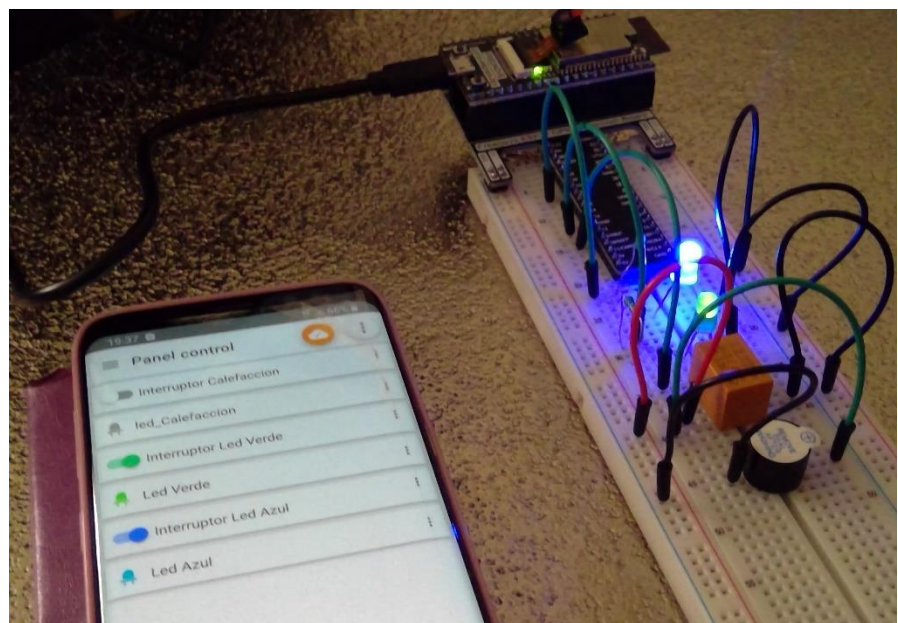
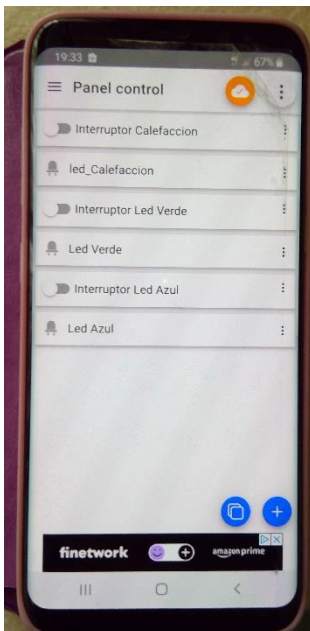
// Establece la conexión con el broker
while (!clienteMQTT.connected())
{
  Serial.print("Conectando al broker ...");
  Serial.println();
  if (clienteMQTT.connect("casa")) {
    Serial.println("Conectado");
    clienteMQTT.subscribe(tema);
    clienteMQTT.subscribe(tema_led_verde);
    clienteMQTT.subscribe(tema_led_azul);
  }
  else delay(5000);
}

// Función a la que se llama cuando llega un mensaje del broker
clienteMQTT.setCallback(callback);

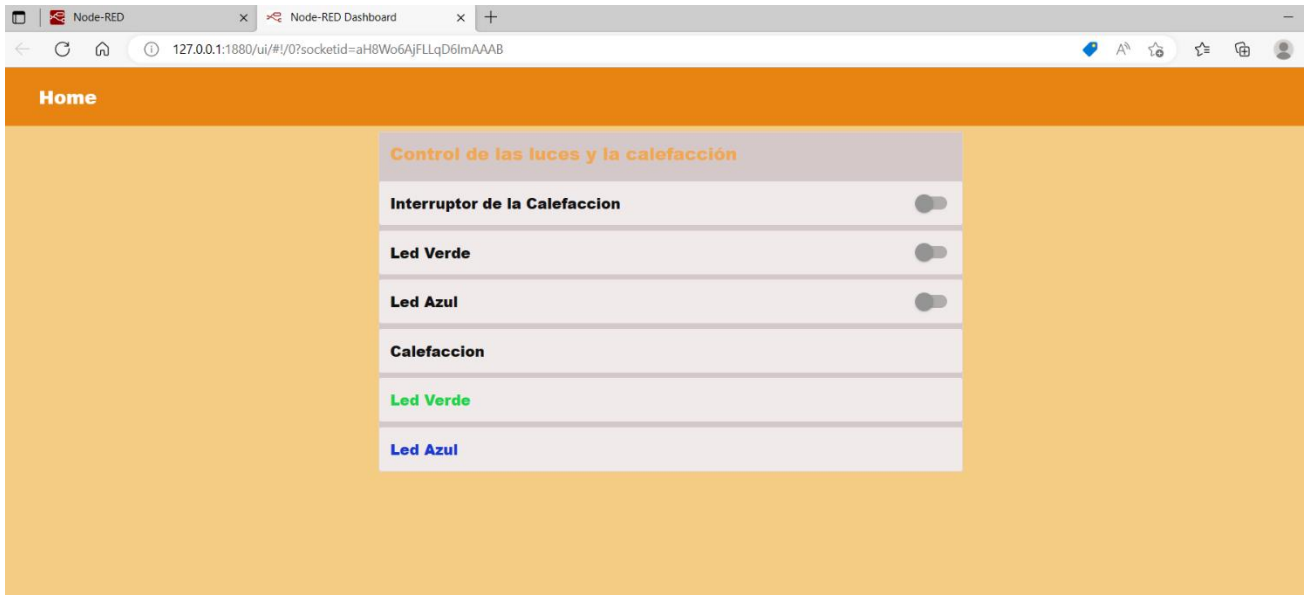
// Cliente escuchando
clienteMQTT.loop();
}

```

5. Panel de control de la App: IoT MQTT Panel



6. Dashboard de Node-Red



Es un panel básico. Se puede cambiar apariencia, tipo de elementos...

7. Aplicación en el Aula

Que é para que	Como			Con que	Como e con que se valora	Duración
Actividade (Título e descrición)	Profesorado (en termo de Tarefas)	Alumnado (Tarefas)	Resultados ou Produtos	Recursos	Instrumentos e Procedementos de Avaliación	Sesións
Control domótico con IoT	Caracterización de IoT	Determina unha posible aplicación domótica de IoT	Circuito físico que represente a montaxe dos elementos a controlar	Portátiles con conexión a Internet mediante WiFi	Taboa de observación: montaxe e funcionamento	12
	Funcionalidade a través de aplicacións básicas. Ex.: control domótico	Identifica elementos físicos e software necesario para lograr unha comunicación dende fora ata o fogar	Programa no microcontrolador para conexión cara e dende o exterior	Programas instalados: Visual Studio Code/Arduino Ide, Node-Red (ou móbiles coa App: IoT MQTT Panel)	Rúbrica para avaliación da Memoria entregada	
	Elementos interconectados e sistemas necesarios	Deseña e monta o circuito electrónico necesario	Panel de control dende o móbil/PC	Compoñentes electrónicos: microcontrolador ESP32 Wrover Kit, protoboard, cables, resistencias, leds, relés, zumbadores/motores...		
	Sistemas gratuitos e protocolos de interconexión abertos	Realiza o programa que permitirá a interconexión a través de WiFi ou 5G, grazas ao Mosquitto	Documento/Memoria indicando elementos necesarios e pasos seguidos	Elementos de medida: polímetro		
	Exemplos de conexións a través do broker Mosquitto	Comproba o funcionamento e a interconexión				