ENTREGA #2 DISEÑO ELECTRÓNIO

CODIGO FUENTE

```
#include <UbiConstants.h>
#include <UbiTypes.h>
#include <UbidotsEsp32Mqtt.h>
#include "DHT.h"
#include <TFT_eSPI.h>
#include <SPI.h>

/****************************************
 * Define Constants
 ****************************************/
#define DHTPIN 2          // Pin conectado al sensor DHT11
#define DHTTYPE DHT22     // Tipo de sensor DHT11
const char *UBIDOTS_TOKEN = "BBUS-UmLWaIxEkCI6nYJel8pIL3zd96IPJa";  // Token de Ubidots
const char *WIFI_SSID = "Lasso";      // Nombre de tu red Wi-Fi
const char *WIFI_PASS = "24825207";   // Contraseña de tu red Wi-Fi
const char *DEVICE_LABEL = "esp32";   // Nombre del dispositivo en Ubidots
const char *TEMP_LABEL = "temp";      // Etiqueta para la temperatura
const char *HUMIDITY_LABEL = "humidity"; // Etiqueta para la humedad
const char *SWITCH_LED_RED = "sw1";   // Etiqueta para el switch del LED rojo
const char *SWITCH_LED_GREEN = "sw2"; // Etiqueta para el switch del LED verde
const int PUBLISH_FREQUENCY = 5000;   // Frecuencia de envío en milisegundos

const uint8_t RED_LED = 12;          // Pin del LED rojo
const uint8_t GREEN_LED = 13;        // Pin del LED verde
unsigned long timer;

// Inicializa el sensor DHT
DHT dht(DHTPIN, DHTTYPE);

// Inicializa el cliente Ubidots
Ubidots ubidots(UBIDOTS_TOKEN);

// Variables para rastrear el estado de los LEDs
int RED_led = 0; // 0 = apagado, 1 = encendido
int GREEN_led = 0; // 0 = apagado, 1 = encendido

// Inicializa la pantalla OLED
TFT_eSPI tft = TFT_eSPI(135, 240); // Pantalla de la TTGO OLED

/****************************************
 * Funciones Auxiliares
```

```
 ***************************************/
void callback(char *topic, byte *payload, unsigned int length)
{
  Serial.print("Mensaje recibido [");
  Serial.print(topic);
  Serial.print("]: ");

  // Verifica si el mensaje es para el switch del LED rojo (sw1)
  if (strstr(topic, SWITCH_LED_RED) != NULL)
  {
    if ((char)payload[0] == '1')
    {
      digitalWrite(RED_LED, HIGH); // Encender el LED rojo
      Serial.println("LED ROJO ON");
      tft.fillCircle(50, 110, 20, TFT_RED);
      RED_led = 1; // Actualizar el estado del LED rojo
    }
    else
    {
      digitalWrite(RED_LED, LOW); // Apagar el LED rojo
      Serial.println("LED ROJO OFF");
      tft.fillCircle(50, 110, 20, TFT_DARKGREY);
      RED_led = 0; // Actualizar el estado del LED rojo
    }
  }

  // Verifica si el mensaje es para el switch del LED verde (sw2)
  if (strstr(topic, SWITCH_LED_GREEN) != NULL)
  {
    if ((char)payload[0] == '1')
    {
      digitalWrite(GREEN_LED, HIGH); // Encender el LED verde
      Serial.println("LED VERDE ON");
      tft.fillCircle(150, 110, 20, TFT_GREEN);
      GREEN_led = 1; // Actualizar el estado del LED verde
    }
    else
    {
      digitalWrite(GREEN_LED, LOW); // Apagar el LED verde
      Serial.println("LED VERDE OFF");
      tft.fillCircle(150, 110, 20, TFT_DARKGREY);
      GREEN_led = 0; // Actualizar el estado del LED verde
    }
  }
```

```
  Serial.println();
}

void initDisplay()
{
  tft.init();
  tft.setRotation(1);
  tft.fillScreen(TFT_BLACK);
  tft.setTextSize(2);
  tft.setTextColor(TFT_GREEN, TFT_BLACK);
  tft.setTextDatum(MC_DATUM);
  tft.drawString("Inicializando...", tft.width() / 2, tft.height() / 2);
  delay(2000);
  tft.fillScreen(TFT_BLACK);
}

/****************************************
 * Funciones principales
 ****************************************/

void setup()
{
  // Configuración inicial
  Serial.begin(115200);
  dht.begin(); // Inicia el sensor DHT

  // Inicializa la pantalla OLED
  initDisplay();

  // Configura los pines de los LEDs como salida
  pinMode(RED_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);

  // Conexión Wi-Fi y Ubidots
  ubidots.connectToWifi(WIFI_SSID, WIFI_PASS);
  ubidots.setCallback(callback);
  ubidots.setup();
  ubidots.reconnect();

  // Suscribirse a los temas de los switches
  ubidots.subscribeLastValue(DEVICE_LABEL, SWITCH_LED_RED);
  ubidots.subscribeLastValue(DEVICE_LABEL, SWITCH_LED_GREEN);
```

```
  timer = millis();
}

void displayData(float temperature, float humidity)
{
  tft.fillScreen(TFT_BLACK);
  tft.setTextSize(3);
  tft.setCursor(0, 20);
  tft.setTextColor(TFT_GREEN, TFT_BLACK);

  // Mostrar temperatura y humedad
  tft.printf("Temp: %.2f C\n", temperature);
  tft.printf("Hum: %.2f %%\n", humidity);

  // Dibujar el círculo rojo según el estado del LED rojo
  if (RED_led == 1)
  {
    tft.fillCircle(50, 110, 20, TFT_RED); // Círculo rojo encendido
  }
  else
  {
    tft.fillCircle(50, 110, 20, TFT_DARKGREY); // Círculo rojo apagado
  }

  // Dibujar el círculo verde según el estado del LED verde
  if (GREEN_led == 1)
  {
    tft.fillCircle(150, 110, 20, TFT_GREEN); // Círculo verde encendido
  }
  else
  {
    tft.fillCircle(150, 110, 20, TFT_DARKGREY); // Círculo verde apagado
  }

  Serial.printf("Temp: %.2f C, Hum: %.2f %%\n", temperature, humidity);
}

void loop()
{
  // Reconexión a Ubidots si es necesario
  if (!ubidots.connected())
  {
    ubidots.reconnect();
    ubidots.subscribeLastValue(DEVICE_LABEL, SWITCH_LED_RED);
```

```
  ubidots.subscribeLastValue(DEVICE_LABEL, SWITCH_LED_GREEN);
 }

 // Enviar datos periódicamente
 if (abs((long)(millis() - timer)) > PUBLISH_FREQUENCY)
 {
  float temperature = dht.readTemperature(); // Lectura de temperatura
  float humidity = dht.readHumidity();      // Lectura de humedad

  // Verifica si las lecturas son válidas
  if (isnan(temperature) || isnan(humidity))
  {
   Serial.println("Error al leer el sensor DHT11");
   tft.fillScreen(TFT_RED);
   tft.setTextDatum(MC_DATUM);
   tft.drawString("Error sensor!", tft.width() / 2, tft.height() / 2);
  }
  else
  {
   // Muestra los datos en la pantalla OLED
   displayData(temperature, humidity);

   // Enviar datos a Ubidots
   ubidots.add(TEMP_LABEL, temperature);   // Añadir temperatura
   ubidots.add(HUMIDITY_LABEL, humidity); // Añadir humedad
   ubidots.publish(DEVICE_LABEL);        // Publicar al dispositivo
  }

  timer = millis(); // Reinicia el temporizador
 ubidots.subscribeLastValue(DEVICE_LABEL, SWITCH_LED_RED);
 ubidots.subscribeLastValue(DEVICE_LABEL, SWITCH_LED_GREEN);
 }

 ubidots.loop();
}
```