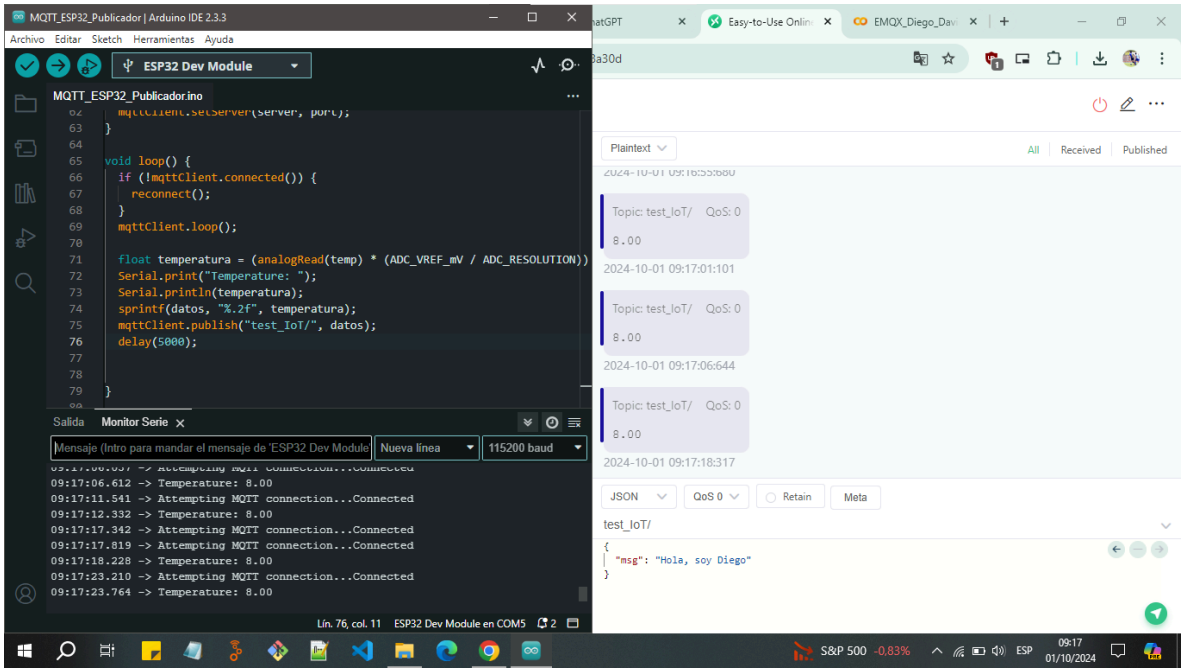
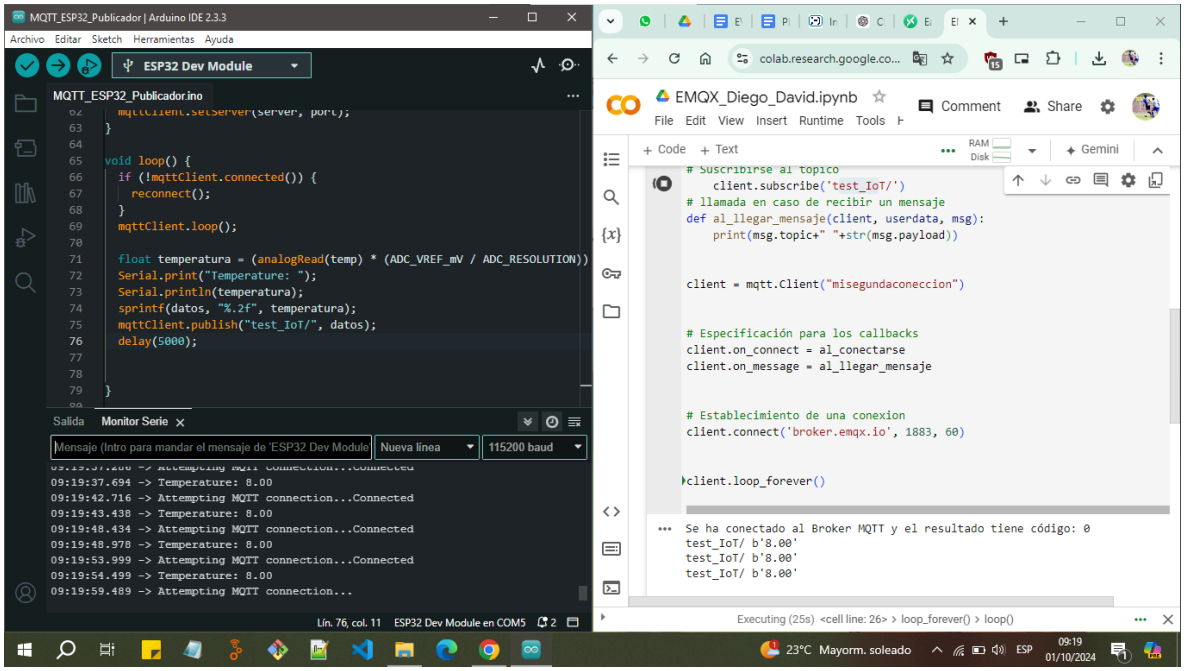
EVIDENCIAS PRACTICA 1
PRIMER MÓDULO MQTT
(PARTE 2)

Grupo:
Integrantes: David Flores y Diego García.
Fecha: 01/10/2024

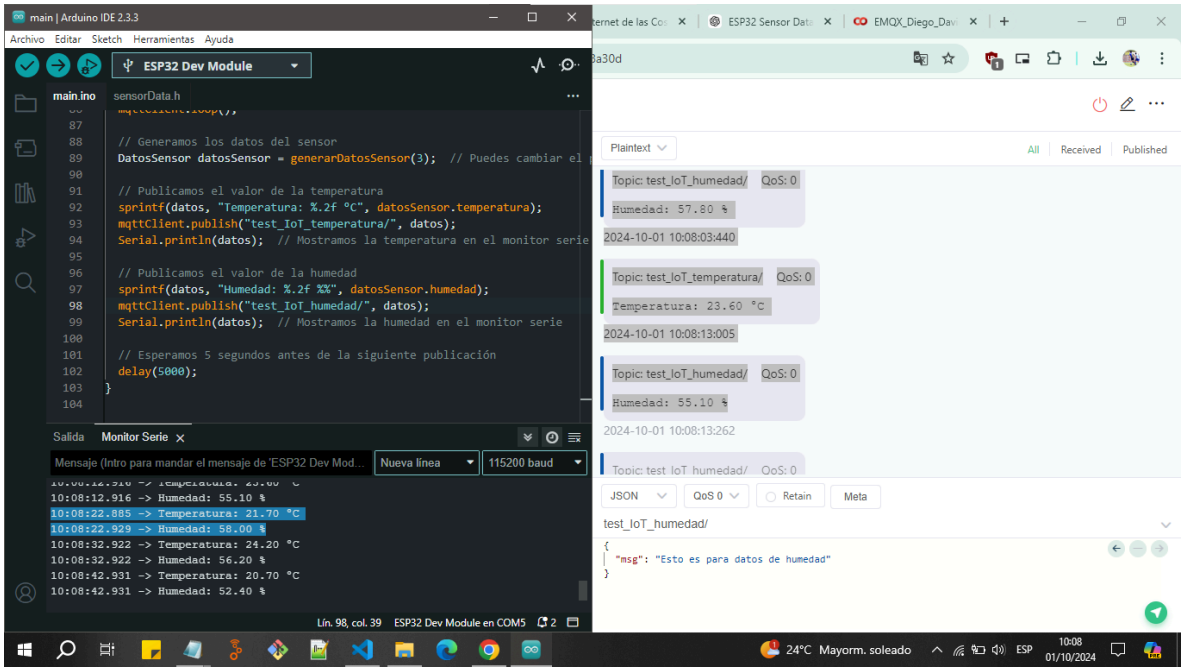
Actividad 4: ESP32 como publicador



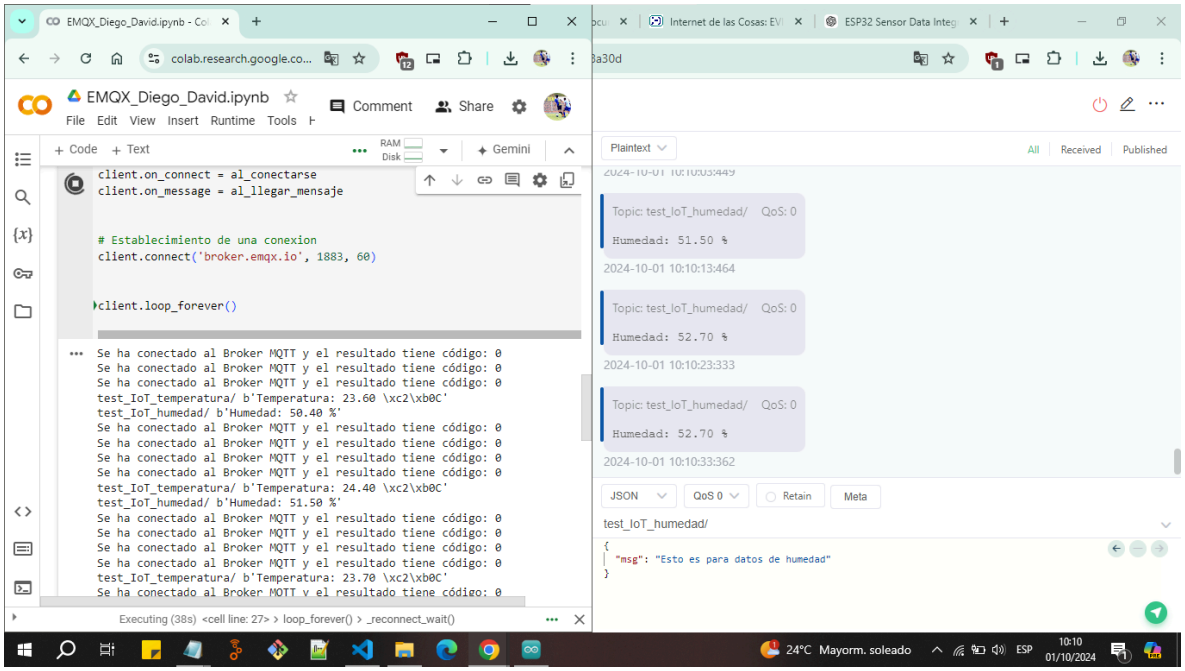
Actividad 4: ESP32 y Python



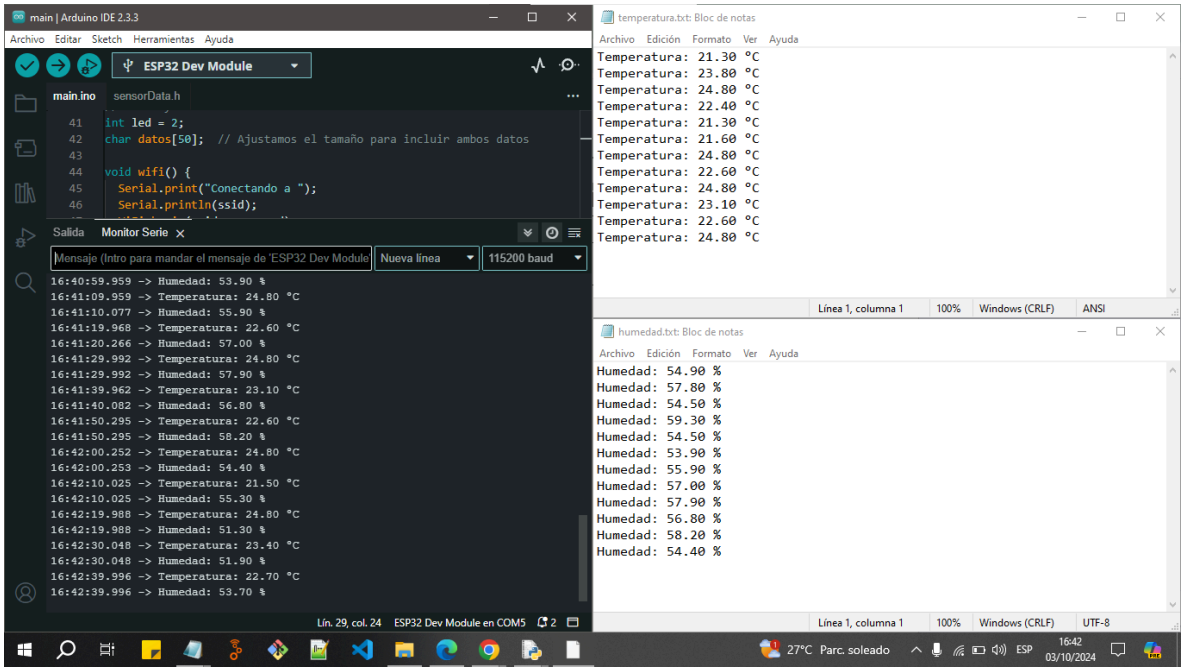
Actividad de cierre: Evidencias etapa 1

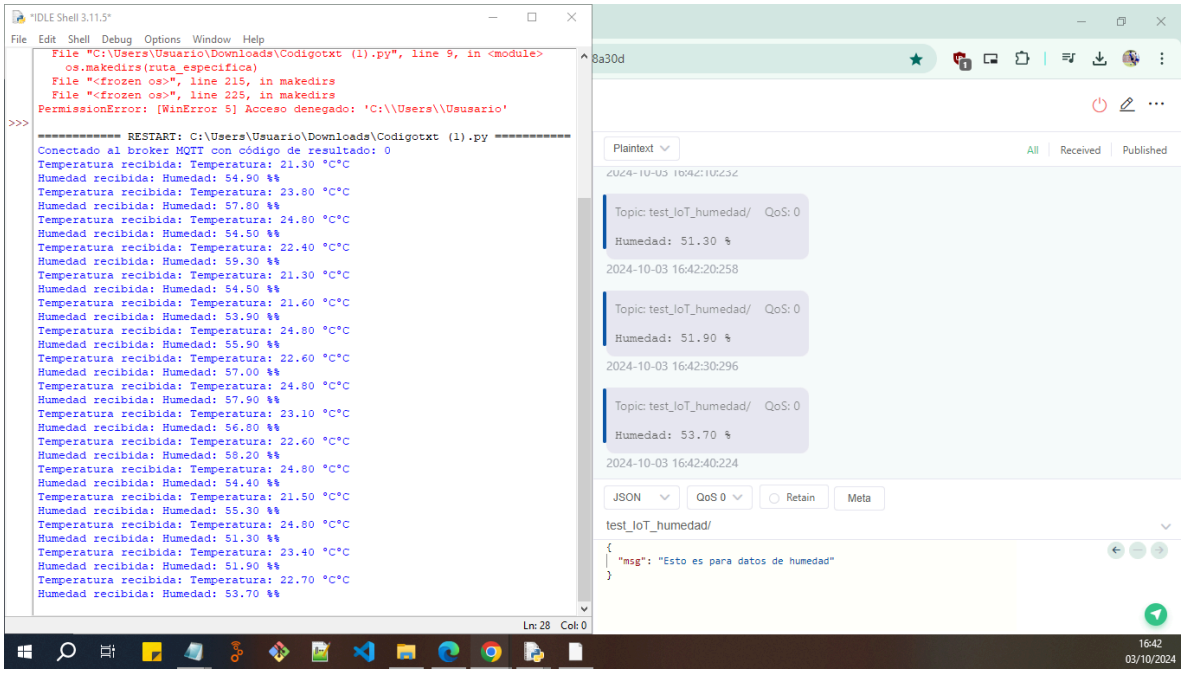


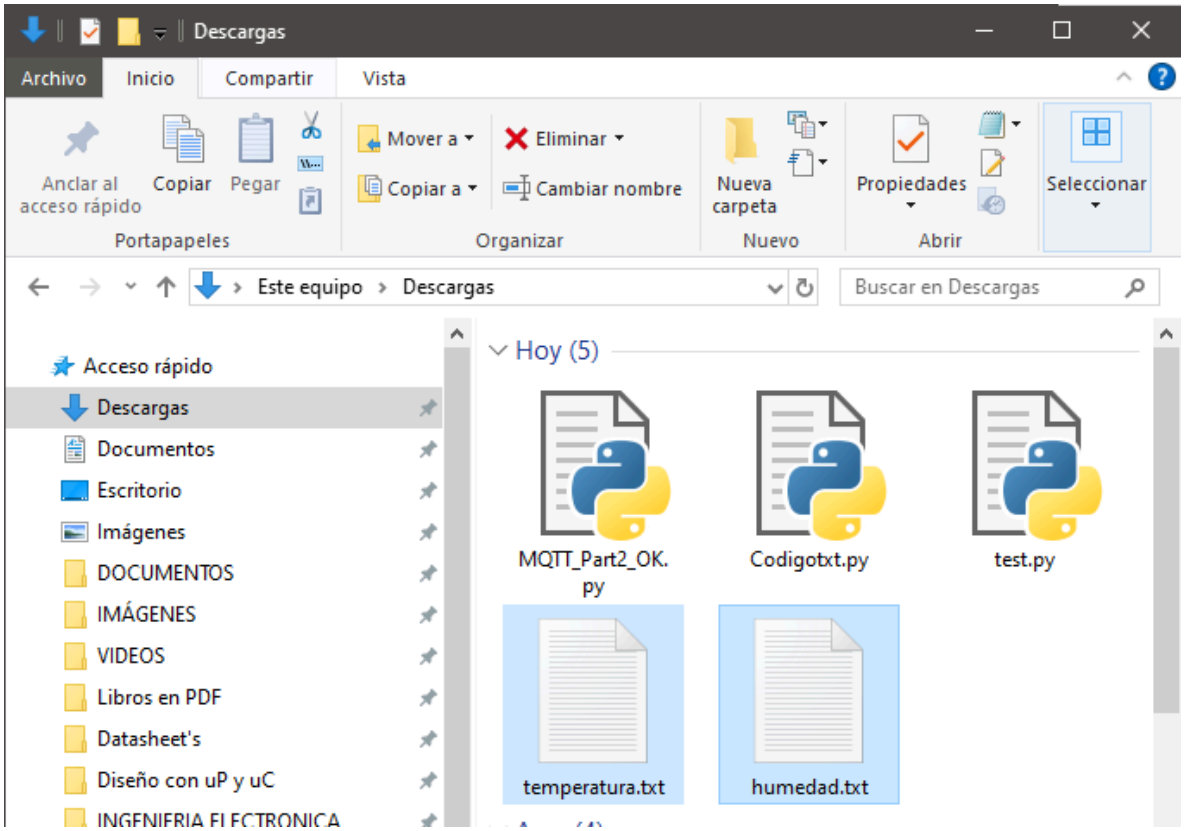
Actividad de cierre: Evidencias etapa 2



Actividad de cierre: Evidencias etapa 3







Finalmente el código que se usó fue el siguiente:

```
import paho.mqtt.client as mqtt
import os

# Especificar la ruta completa donde se guardarán los archivos
ruta_especifica = "C:\\Users\\Usuario\\Downloads"  # Cambia esta ruta
```

```
por la que necesites

# Verificar si la carpeta existe, si no, crearla
if not os.path.exists(ruta_especifica):
    os.makedirs(ruta_especifica)

# Variables para almacenar las rutas completas de los archivos
temperatura_file = os.path.join(ruta_especifica, "temperatura.txt")
humedad_file = os.path.join(ruta_especifica, "humedad.txt")

# Función que maneja los mensajes recibidos de los tópicos MQTT
def on_message(client, userdata, message):
    topic = message.topic
    payload = message.payload.decode("utf-8")

    # Verificar el tópico y escribir en el archivo correspondiente
    if topic == "test_IoT_temperatura/":
        with open(temperatura_file, "a") as f:
            f.write(f"{payload}\n")  # Guardar cada dato en una nueva
línea
        print(f"Temperatura recibida: {payload}°C")

    elif topic == "test_IoT_humedad/":
        with open(humedad_file, "a") as f:
            f.write(f"{payload}\n")  # Guardar cada dato en una nueva
línea
        print(f"Humedad recibida: {payload}%")

# Función que se ejecuta cuando la conexión al broker se realiza con
éxito
def on_connect(client, userdata, flags, rc):
    print(f"Conectado al broker MQTT con código de resultado:
"+str(rc))

    # Suscribirse a los tópicos de temperatura y humedad
    client.subscribe("test_IoT_temperatura/")
    client.subscribe("test_IoT_humedad/")

# Configuración del cliente MQTT
broker = "broker.emqx.io"
port = 1883

client = mqtt.Client("PythonSubscriber")  # Crear un nuevo cliente
MQTT
client.on_connect = on_connect  # Establecer la función de conexión
client.on_message = on_message  # Establecer la función para manejar
los mensajes

# Conectarse al broker MQTT
client.connect(broker, port)

# Mantener la conexión activa y esperar los mensajes
client.loop_forever()
```