

EVIDENCIAS EVALUACIÓN FINAL Node-RED

NOMBRE: DIEGO ANDRES GARCIA DIAZ. **CÓDIGO:** 2195533

FECHA: 14/11/2024

ACTIVIDAD PRÁCTICA: Configura un flujo en Node-RED que reciba datos de un sensor a través de MQTT, muestre en un dashboard la lectura de dos variables provenientes de esos datos. Describe los pasos necesarios y suba un documento con las evidencias del mismo, use un tópico diferente a los usados anteriormente.

Inicialmente se respondieron las preguntas de respuesta múltiple, posteriormente se procedió a realizar la parte práctica de dicha evaluación final, donde se realizó un flujograma en Node-RED, con el fin de tener una interfaz gráfica para observar datos de sensores (variables), ya sean simulados o con los sensores en físico, usando la ESP32 o directamente desde un código en Python por ejemplo.

- 1) Se realizó un programa para cargar en la ESP32 con el fin de hacer lectura de un sensor DHT11 para leer temperatura y humedad, también un sensor HC-SR04 para leer distancia:

The screenshot displays the Node-RED web interface. On the left, the Arduino IDE editor shows a C++ sketch for an ESP32 module. The sketch includes code for serial printing and MQTT publishing. The main workspace shows a flow with several nodes: a 'MQTT In' node, a 'JSON Parse' node, and a 'Dashboard' node. The 'MQTT In' node is configured with the topic 'evaluacionFinal'. The 'JSON Parse' node is configured with the payload type 'object'. The 'Dashboard' node is configured with the topic 'evaluacionFinal'. The right sidebar shows the 'debug' console with logs of the received MQTT messages, including the topic, payload, and qos.

```
practica_5.ino
138 } else {
139 // Mostrar en el monitor serial
140 Serial.print("Humedad: ");
141 Serial.print(h);
142 Serial.print("% Temperatura: ");
143 Serial.print(t);
144 Serial.print("°C Distancia:");
145 Serial.print(distancia);
146 Serial.println("cm");
147 }
148 // Publicar la distancia, temperatura y humedad en el broker MQTT
149 //sprintf(datos, "Distancia: %d, Temp: %.f, Humedad: %.f", distancia, t, h);
150 Serial.println(datos);
151 mqttClient.publish("evaluacionFinal", datos.c_str());
152 delay(3000); // Espera antes de la siguiente lectura
153 }
154 }
```

Salida Monitor Serie x

Mensaje (Intro para mandar el mensaje de "ESP32 Dev Mod... Nueva línea 115200 baud

09:47:09.998 -> {"Dispositivos":"ESP32","Temperatura":22.29999924,"Humedad":57.00000076,"Distancia":12cm}

09:47:13.041 -> Humedad: 57.00% Temperatura: 22.30°C Distancia:12cm

09:47:13.041 -> {"Dispositivos":"ESP32","Temperatura":22.29999924,"Humedad":57.00000076,"Distancia":12cm}

09:47:16.050 -> Humedad: 57.00% Temperatura: 22.20°C Distancia:12cm

09:47:16.050 -> {"Dispositivos":"ESP32","Temperatura":22.20000076,"Humedad":57.00000076,"Distancia":12cm}

09:47:19.079 -> Humedad: 57.00% Temperatura: 22.20°C Distancia:12cm

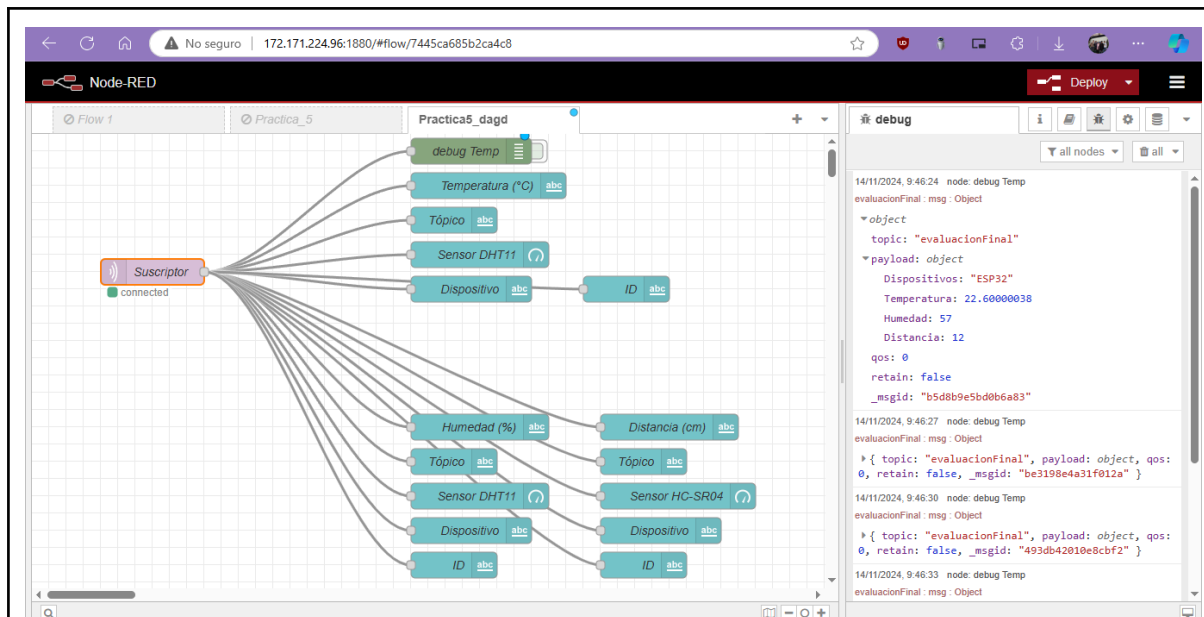
09:47:19.079 -> {"Dispositivos":"ESP32","Temperatura":22.20000076,"Humedad":57.00000076,"Distancia":12cm}

09:47:22.148 -> Humedad: 57.00% Temperatura: 22.20°C Distancia:12cm

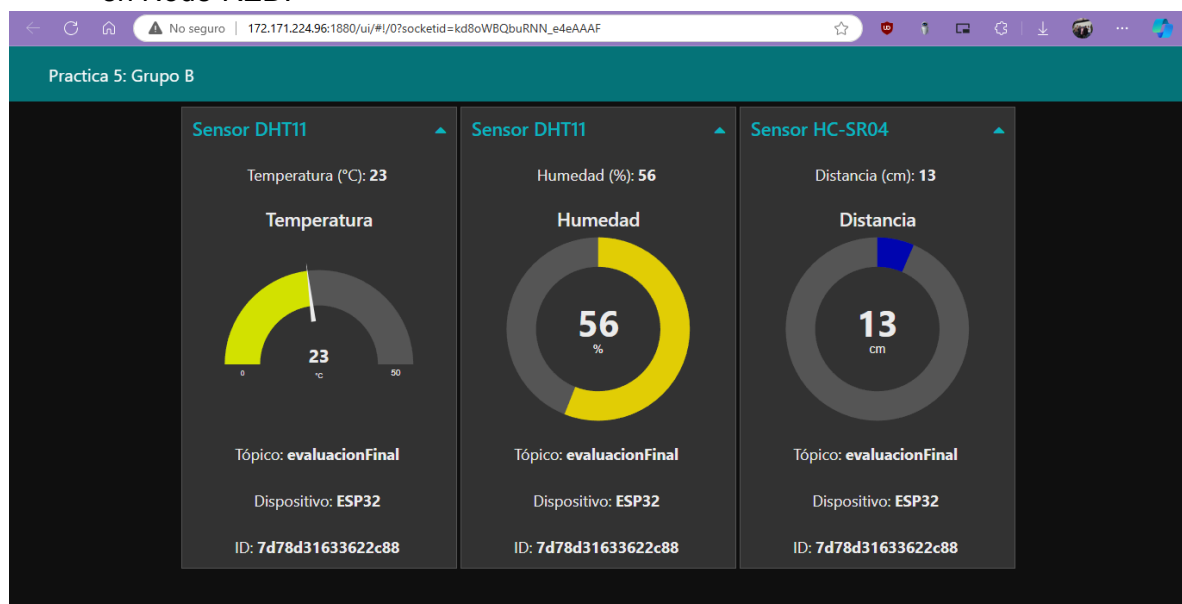
09:47:22.148 -> {"Dispositivos":"ESP32","Temperatura":22.20000076,"Humedad":57.00000076,"Distancia":12cm}

Lin. 151, col. 23 ESP32 Dev Module en COM5 2

- 2) Se realiza el flujograma según las necesidades y según como se quieran organizar los diferentes elementos en el dashboard:



3) Finalmente se observan los datos de los sensores en la interfaz gráfica realizada en Node-RED:



4) Evidencia uso de ESP32 con los sensores mencionados anteriormente:

