



C5.1 Interface Node-RED y NodeMCU ESP32

Arquitectura Cliente-Servidor, utilizando interface Node-red, ESP32 y un sensor



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 **C5.1_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
| | | C5.1_TituloActividad.md
| | | C5.2_TituloActividad.md
| | img
| | docs
| | | A5.1_TituloActividad.md
| | | A5.2_TituloActividad.md
```



Desarrollo

1. Basado en las actividades referentes a sensores, y protocolos de comunicación realice un Dashboard utilizando la interface Node-red y el NodeMCU ESP32; el cual permitirá a un cliente a través de su

dispositivo móvil acceder a una dirección de un servidor Web local, y al ingresar al sitio deberá observar un dashboard que visualice el comportamiento de un sensor en tiempo real.

2. Para poder observar que la interface muestra el comportamiento de un sensor deberá a este someterse a condiciones cambiantes a fin de observar esas variaciones en el dashboard.

Video

Video Demostrativo

3. Agregue el programa creado para lograr la condición solicitada.

```
#include <DHTesp.h>
#include <WiFi.h>
#include <WebServer.h>
const char* ssid = "INFINITUM8AA4_2.4";
const char* password = "R47Za4aA3b";
WebServer server(80);
DHTesp dht;
const int SENSOR_PIN = 4;
TempAndHumidity reading;

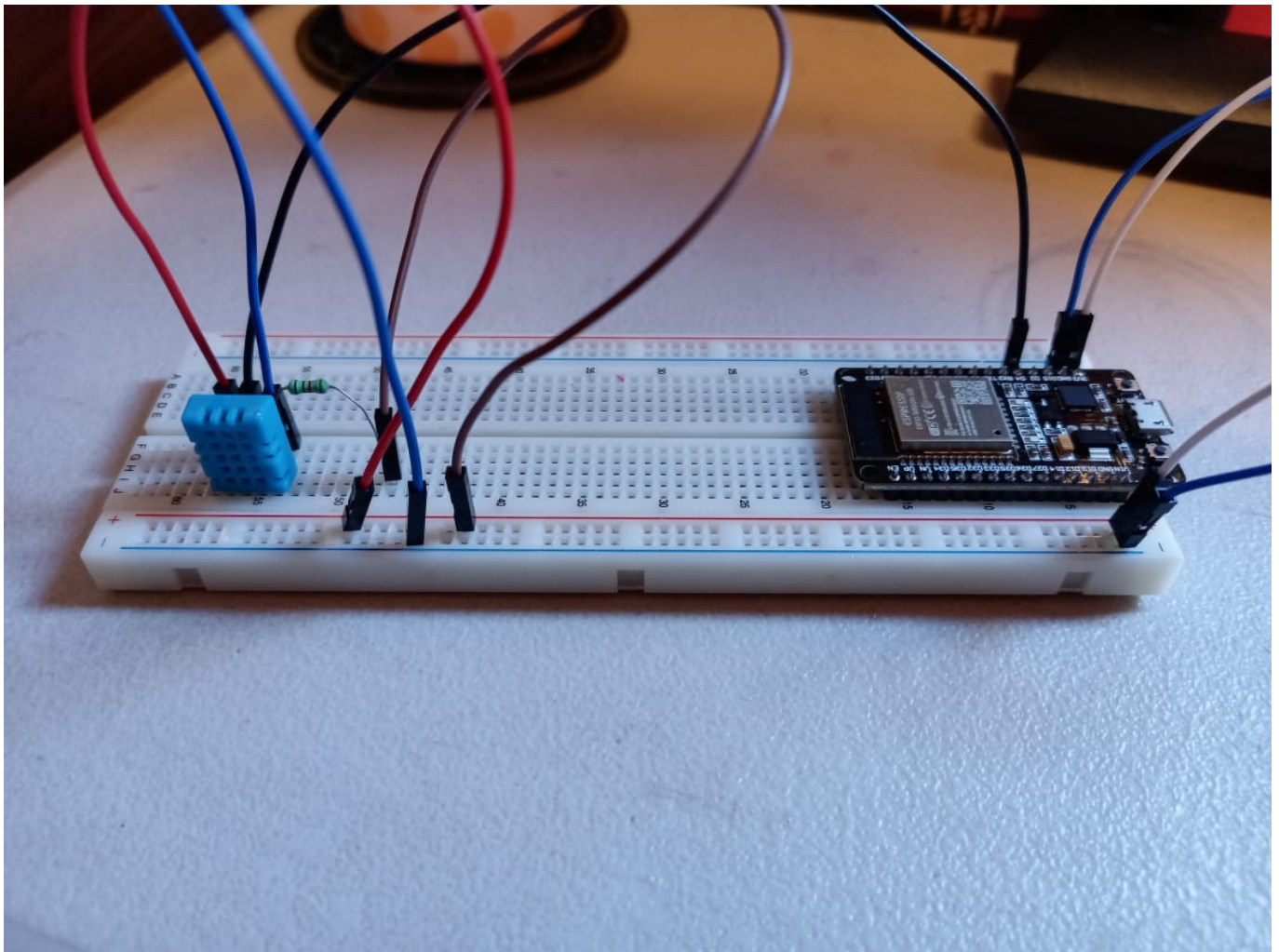
void setup() {
  // Consola serial
  Serial.begin(115200);
  // establecer el tipo del Sensor que se utilizara
  dht.setup(SENSOR_PIN, DHTesp::DHT11);
  // establecer la conexion wifi
  WiFi.begin(ssid, password);
  Serial.print("intenteando conectar a ");
  Serial.print(ssid);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("se ha establecido una coneccion, dirijete a ");
  Serial.println(WiFi.localIP());
  // HTTP server
  server.on("/", handle_root);
  server.begin();
  delay(100);
  Serial.println("Servidor Listo!");
}

void loop() {
  reading = dht.getTempAndHumidity();
  server.handleClient();
}

void handle_root() {
  server.send(200, "application/json", "{\"temperature\": " +
String(reading.temperature) + ", \"humidity\": " + String(reading.humidity) +
```

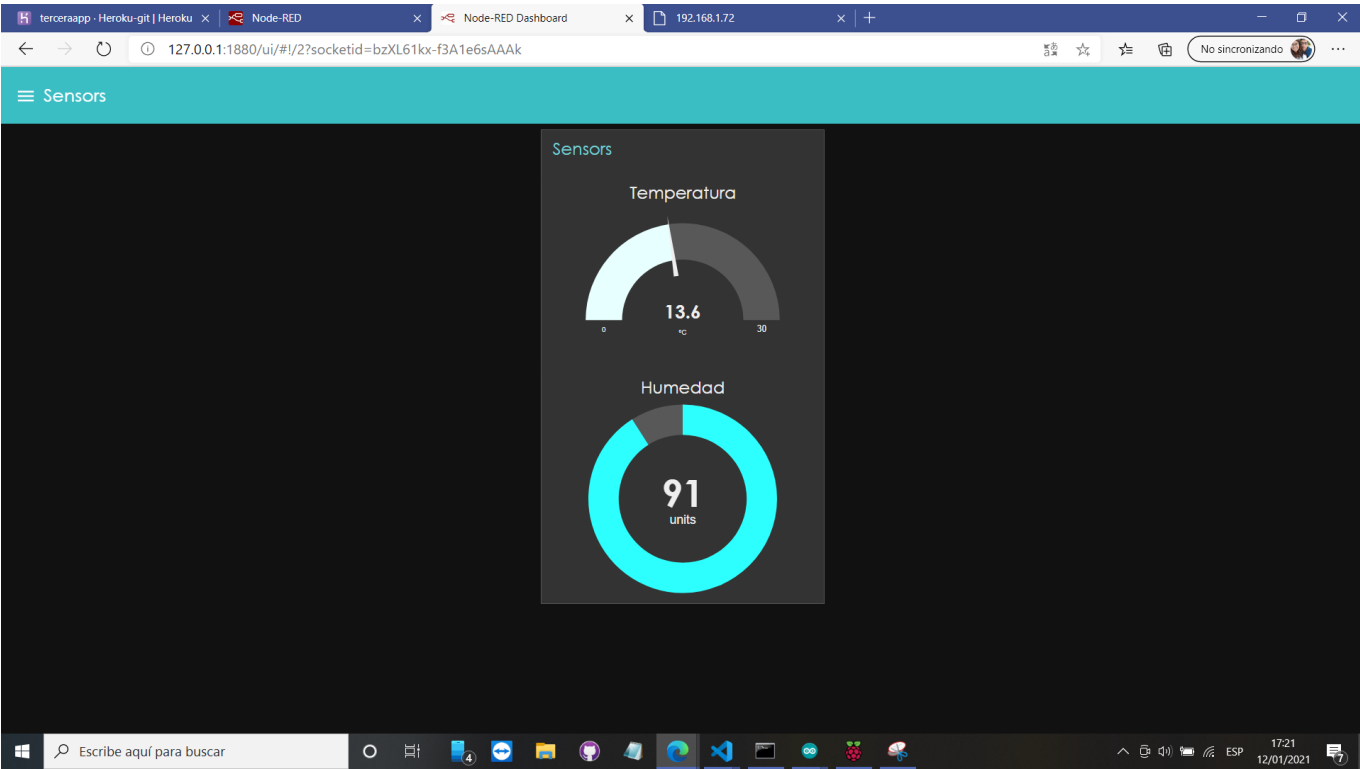
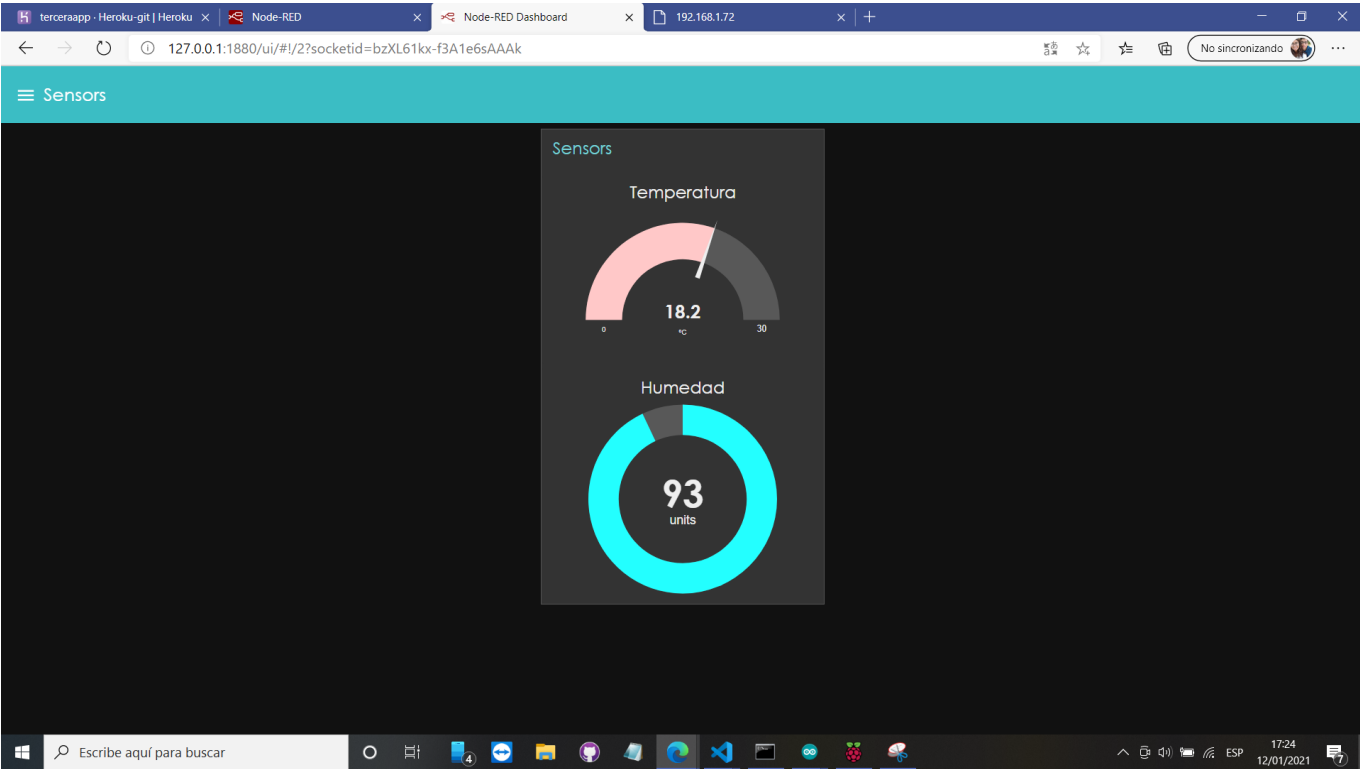
```
"}");  
}
```

4. Agregue imágenes del circuito electrónico ensamblado.



5. Inserte aquí las imágenes que considere como evidencias para demostrar el resultado obtenido y las reuniones del equipo de trabajo.

Funcionamiento



The image shows a Windows desktop with two windows open. The top window is a web browser displaying a Node-RED dashboard at the URL `127.0.0.1:1880/ui/#/1/2?socketid=bzXL61kx-f3A1e6sAAAk`. The dashboard has a teal header with a menu icon and the text "Sensors". The main content area is dark grey and contains two circular gauges. The first gauge, titled "Temperatura", has a red arc and shows a value of 27.9 °C on a scale from 0 to 30. The second gauge, titled "Humedad", has a cyan arc and shows a value of 95 units. The bottom window is a serial monitor titled "COM8". It has a text input field at the top with "Enviar" and a scrollable output area. The output text reads: `.intenteando conectar a INFINITUM8AA4_2.4....`, `se ha establecido una coneccion, dirigitse a 192.168.1.72`, and `Servidor Listo!`. At the bottom of the serial monitor, there are controls for "Autoscroll" (checked), "Nueva línea", "115200 baudio", and "Clear output".

Sensors

Temperatura

27.9 °C

Humedad

95 units

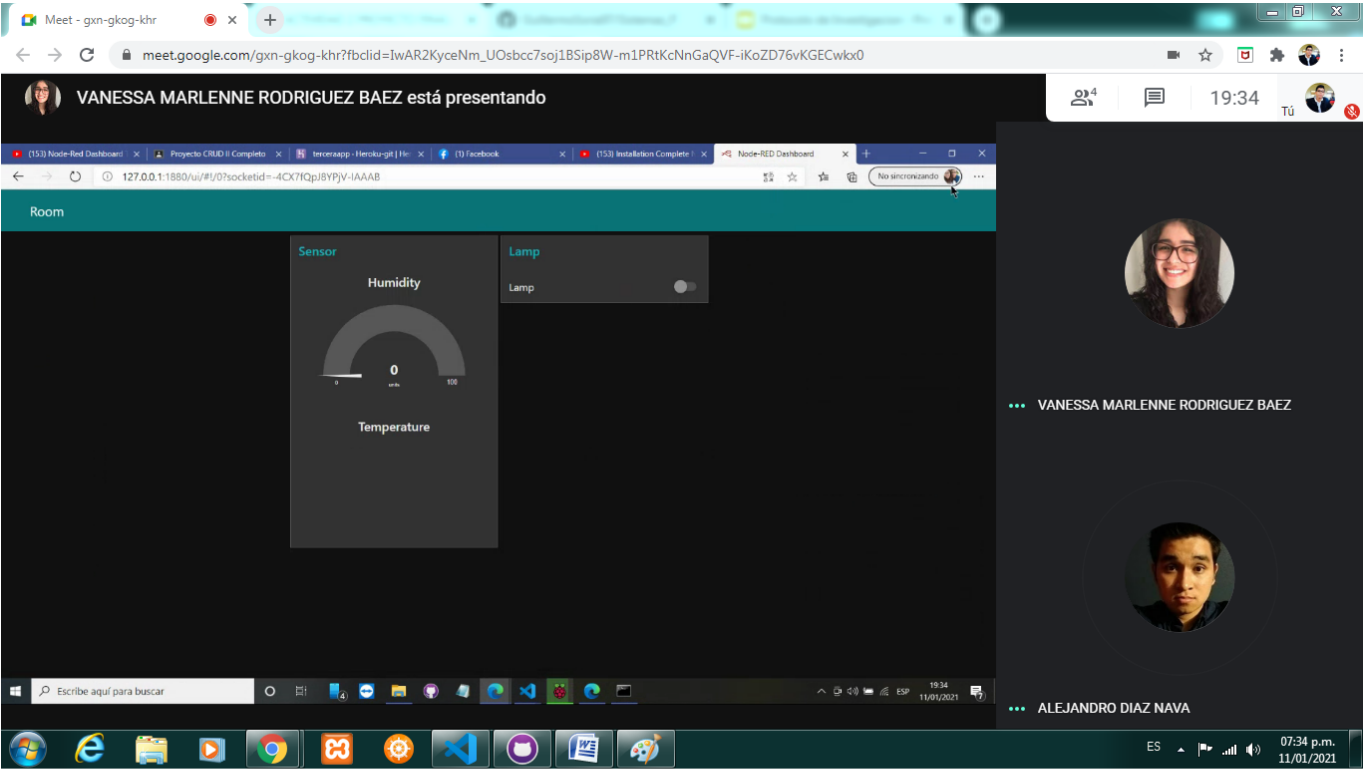
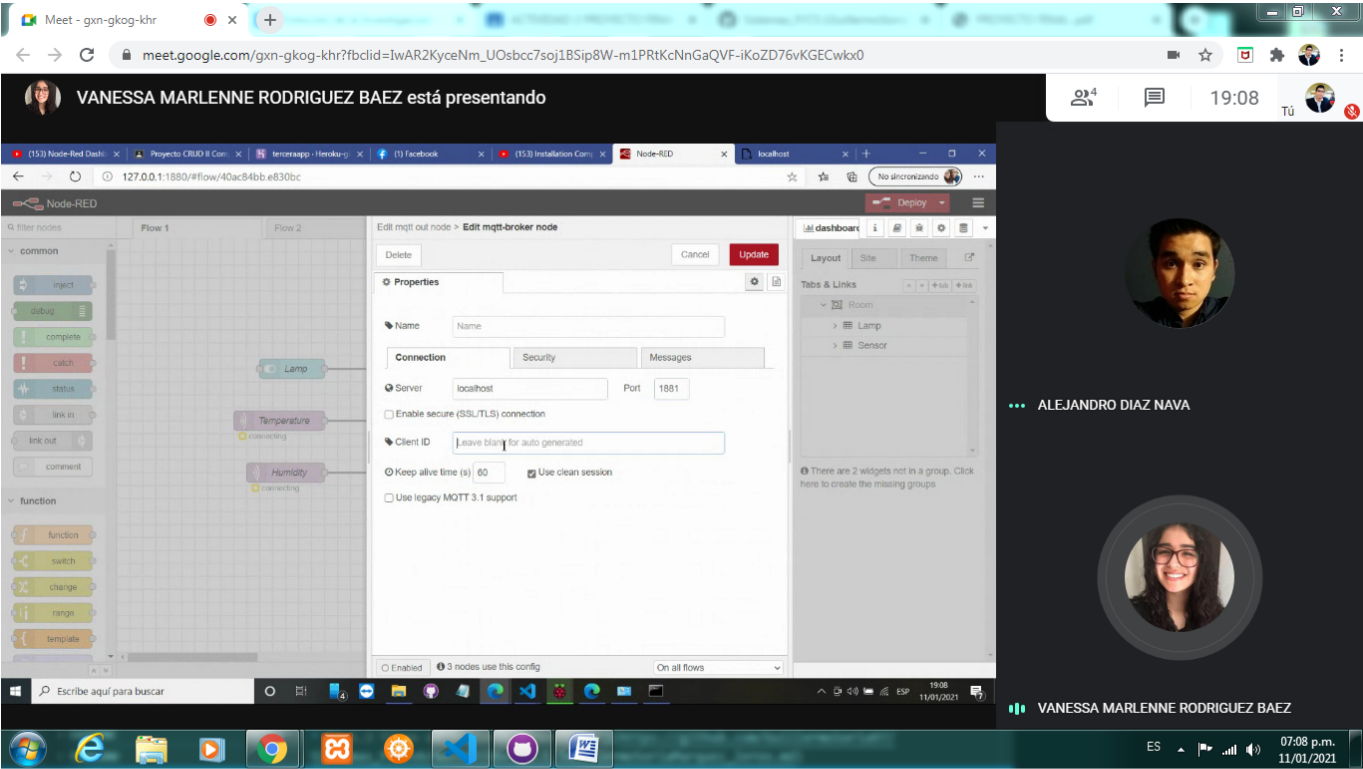
COM8

Enviar

.intenteando conectar a INFINITUM8AA4_2.4....
se ha establecido una coneccion, dirigitse a 192.168.1.72
Servidor Listo!

Autoscroll Nueva línea 115200 baudio Clear output

Reuniones



The screenshot shows a Google Meet window with the title "VANESSA MARLENNE RODRIGUEZ BAEZ está presentando". The main content area displays the Arduino IDE interface with the following code:

```
dtostrf(h, 6, 2, humidityTemp);

// Publishes Temperature and Humidity values
client.publish("room/temperature", temperatureTemp);
client.publish("room/humidity", humidityTemp);

Serial.print("Humidity: ");
Serial.print(h);
Serial.print(" %\t Temperature: ");
Serial.print(t);
Serial.print(" *C ");
Serial.print(f);
Serial.print(" *F\t Heat index: ");
Serial.print(hic);
Serial.println(" *C ");
// Serial.print(hif);
// Serial.println(" *F");
}
```

The right sidebar shows the participants: VANESSA MARLENNE RODRIGUEZ BAEZ and ALEJANDRO DIAZ NAVA. The bottom status bar indicates the time is 19:42 on 11/01/2021.

The screenshot shows a Google Meet window with the title "ALEJANDRO DIAZ NAVA está presentando". The main content area displays the Arduino IDE interface with the following code:

```
mqtt_esp8266

// Create a random client ID
String clientId = "ESP8266Client-";
clientId += String(random(0xffff), HEX);
// Attempt to connect
if (client.connect(clientId_c_str())) {
  Serial.println("connected");
  // Once connected, publish an announcement...
  client.publish("hola", "hello world");
  // ... and resubscribe
  client.subscribe("data");
} else {
  Serial.print("failed, rc=");
  Serial.print(client.state());
  Serial.println(" try again in 5 seconds");
  // Wait 5 seconds before retrying
  delay(5000);
}

void setup() {
  pinMode(BUILTIN_LED, OUTPUT); // Initialize the BUILTIN_LED pin as an output
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}

void loop() {
  Subido
  Leaving...
  Hard resetting via RTS pin...
}
```

The right sidebar shows the participants: ALEJANDRO DIAZ NAVA and VANESSA MARLENNE RODRIGUEZ BAEZ. The bottom status bar indicates the time is 07:44 p.m. on 11/01/2021.



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

[Link Díaz Navarro Alejandro](#)[Link Rodríguez Báez Vanessa Marlenne](#)[Link Soria Márquez Guillermo](#)