



## C5.2 Interface Node-RED y NodeMCU ESP32

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



### 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 cuenta 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 **C5.2\_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 actuadores, 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 poder interactuar y visualizar el comportamiento de un actuador en tiempo real.

2. Para poder observar que la interface muestra el comportamiento del actuador deberá a este someterse a condiciones cambiantes a fin de observar esas variaciones en la interface.
3. Agregue el programa creado para lograr la condición solicitada.

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
#include <Servo.h>
Servo myservo;

// Replace the next variables with your SSID/Password combination
const char* ssid = "REPLACE_WITH_YOUR_SSID";
const char* password = "REPLACE_WITH_YOUR_PASSWORD";

// Add your MQTT Broker IP address, example:
//const char* mqtt_server = "192.168.1.144";
const char* mqtt_server = "YOUR_MQTT_BROKER_IP_ADDRESS";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

void setup() {
  Serial.begin(115200);
  myservo.attach(13);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}

void setup_wifi() {
  delay(10);
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

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

  Serial.println("");
  Serial.println("Connected");
}
```

```
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* message, unsigned int length) {
    String movimiento;
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");

    for (int i = 0; i < length; i++) {
        Serial.print((char)message[i]);
        movimiento += (char)message[i];
    }

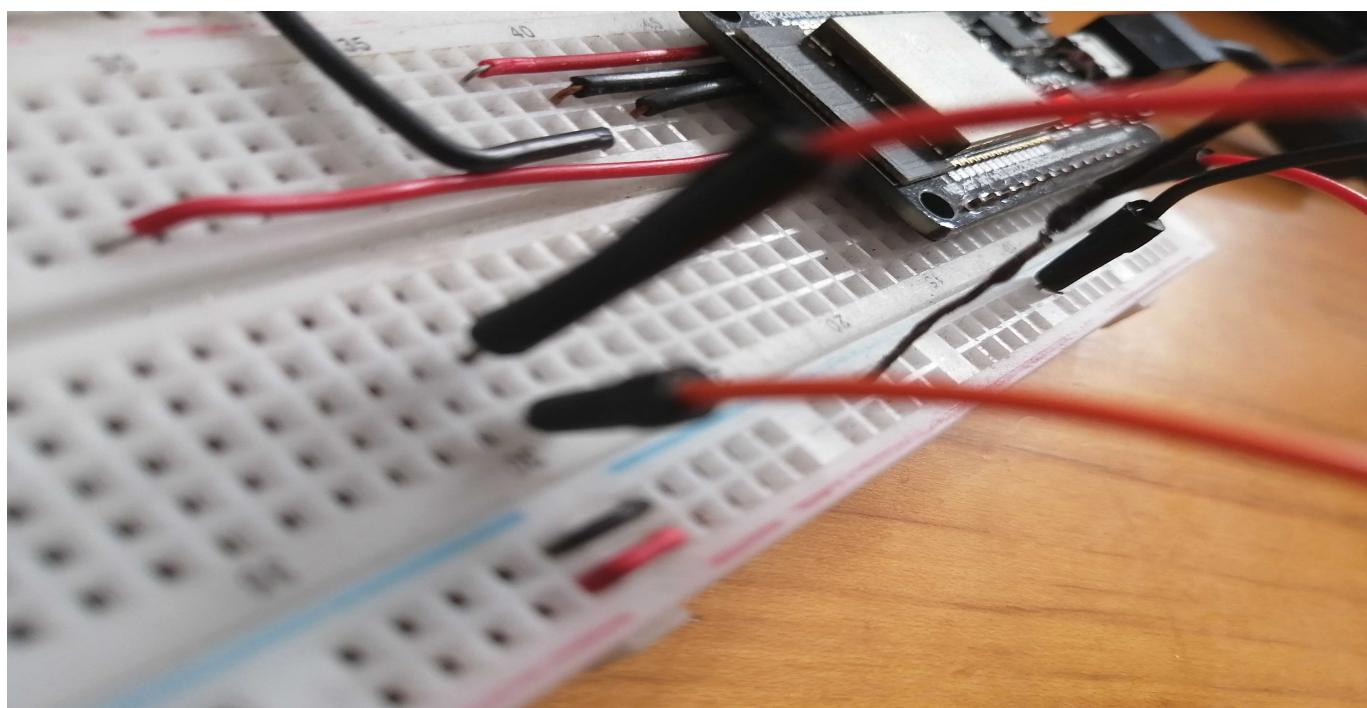
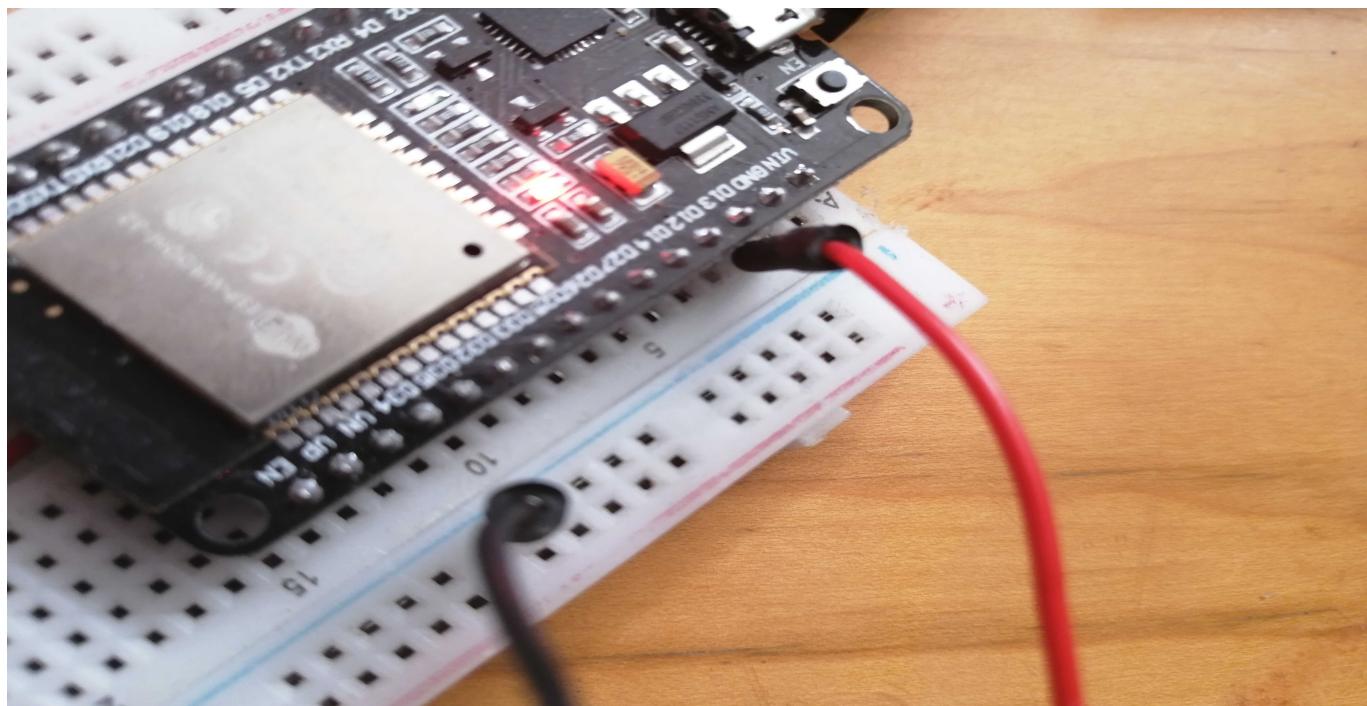
    if (String(topic) == "servo"){
        Serial.print("Changing output to ");
        int resultado = movimiento.toInt();
        int pos = map(resultado, 1, 100, 0, 180);
        Serial.println(pos);
        myservo.write(pos);
        delay(15);
    }
}

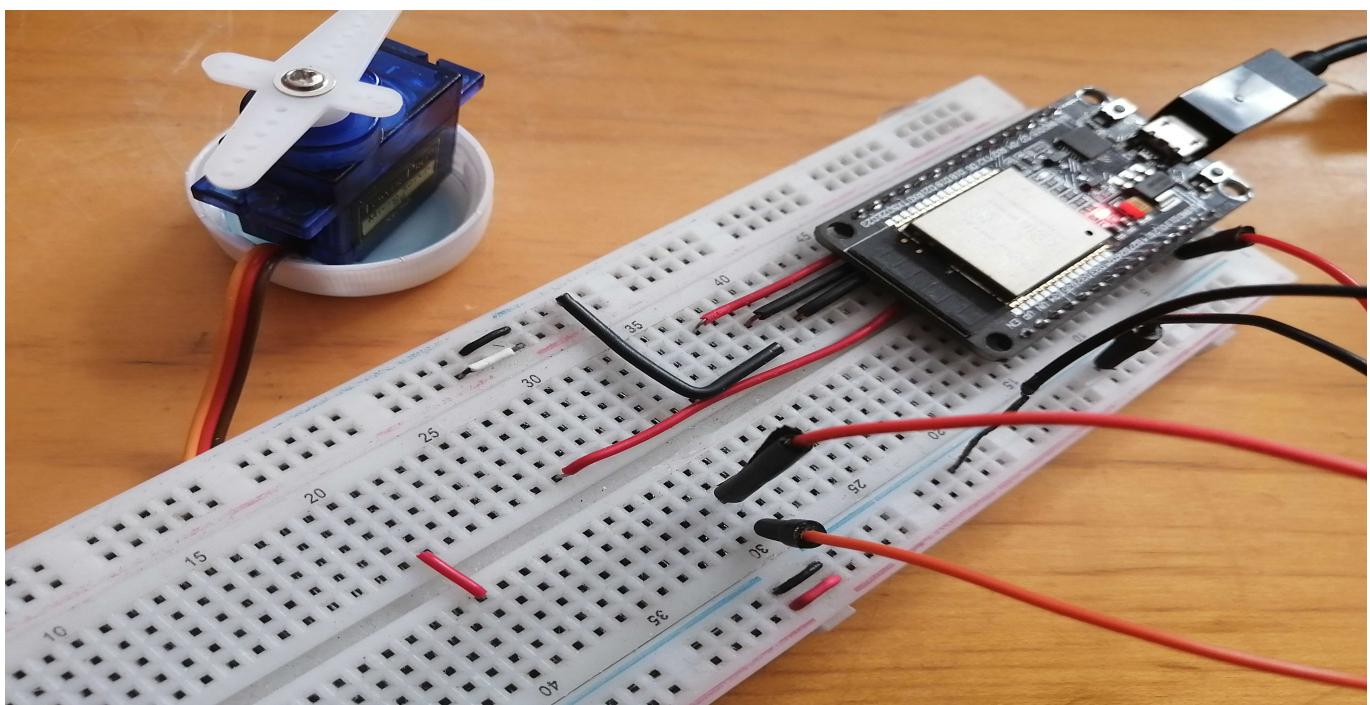
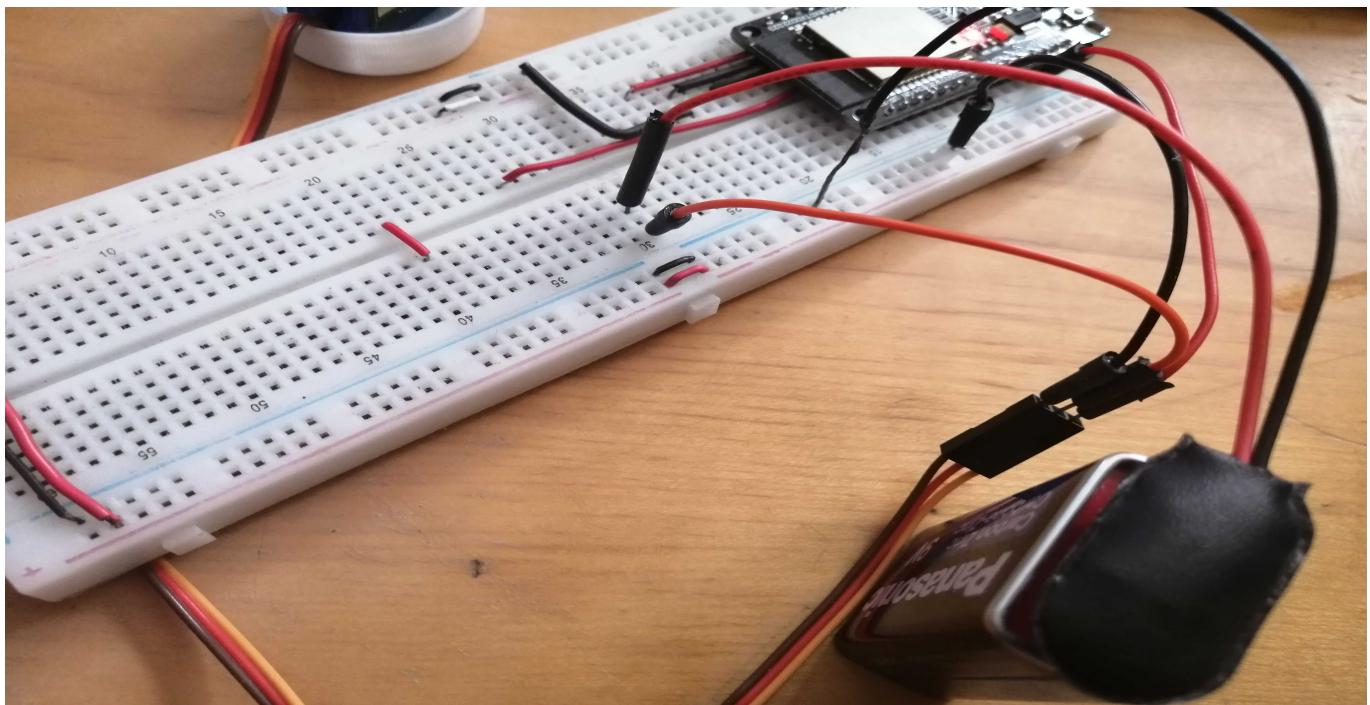
void reconnect() {
    // Loop until we're reconnected
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        // Attempt to connect
        if (client.connect("ESP8266Client")) {
            Serial.println("connected");
            // Subscribe
            client.subscribe("servo");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            // Wait 5 seconds before retrying
            delay(5000);
        }
    }
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

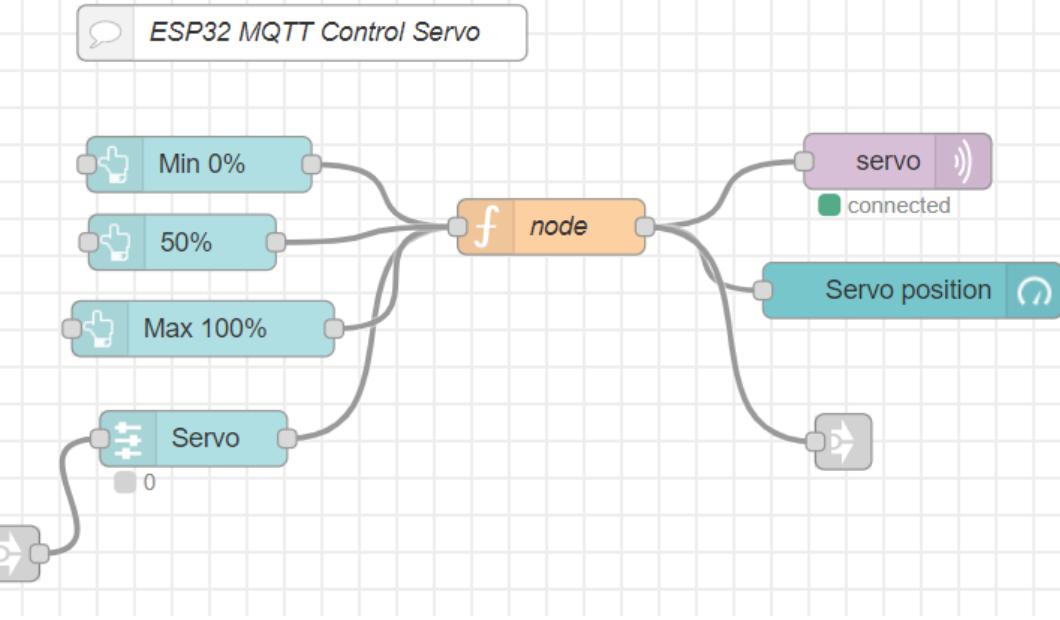
    delay(100);
}
```

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.



CONNECTING TO ARRIS-510Z

....

WiFi connected

IP address:

192.168.0.11

Attempting MQTT connection...connected

Message arrived [servo] 5050

Changing output to 89

Message arrived [servo] 5050

Changing output to 89

Message arrived [servo] 00

Changing output to -1

Message arrived [servo] 00

Changing output to -1

Autoscroll  Mostrar marca temporal

```

= 0;
;
{
n(115200);
ach(13);
};
  
```

node-red

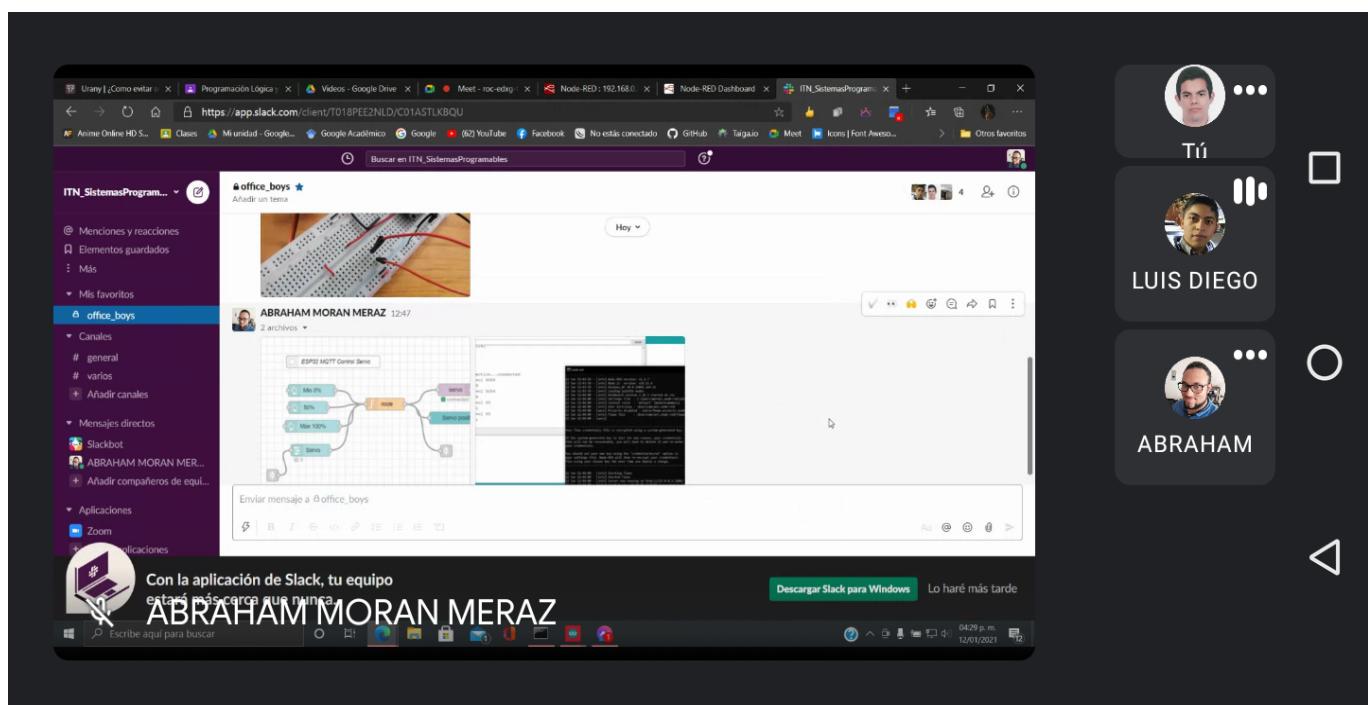
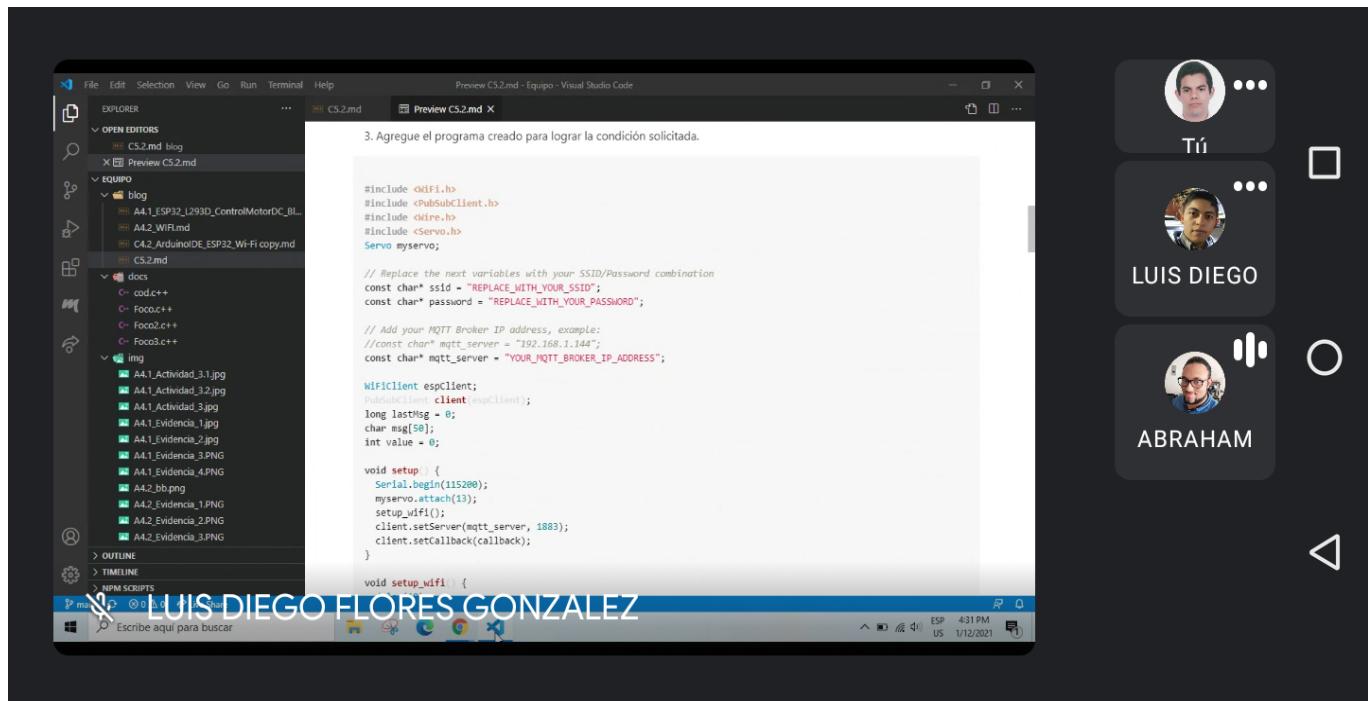
12 Jan 12:43:58 - [info] Node-RED version: v1.2.7  
12 Jan 12:43:58 - [info] Node.js version: v14.15.4  
12 Jan 12:43:58 - [info] Windows\_NT 10.0.19042 x64 LE  
12 Jan 12:43:59 - [info] Loading palette nodes  
12 Jan 12:44:00 - [info] Dashboard version 2.26.2 started at /ui  
12 Jan 12:44:00 - [info] Settings file : C:\Users\moran\.node-red\settings.js  
12 Jan 12:44:00 - [info] Context store : 'default' [module=memory]  
12 Jan 12:44:00 - [info] User directory : \Users\moran\.node-red  
12 Jan 12:44:00 - [warn] Projects disabled : editorTheme.projects.enabled=false  
12 Jan 12:44:00 - [info] Flows file : \Users\moran\.node-red\flows\_LAPTOP-3DGUI0FA.json  
12 Jan 12:44:00 - [warn]

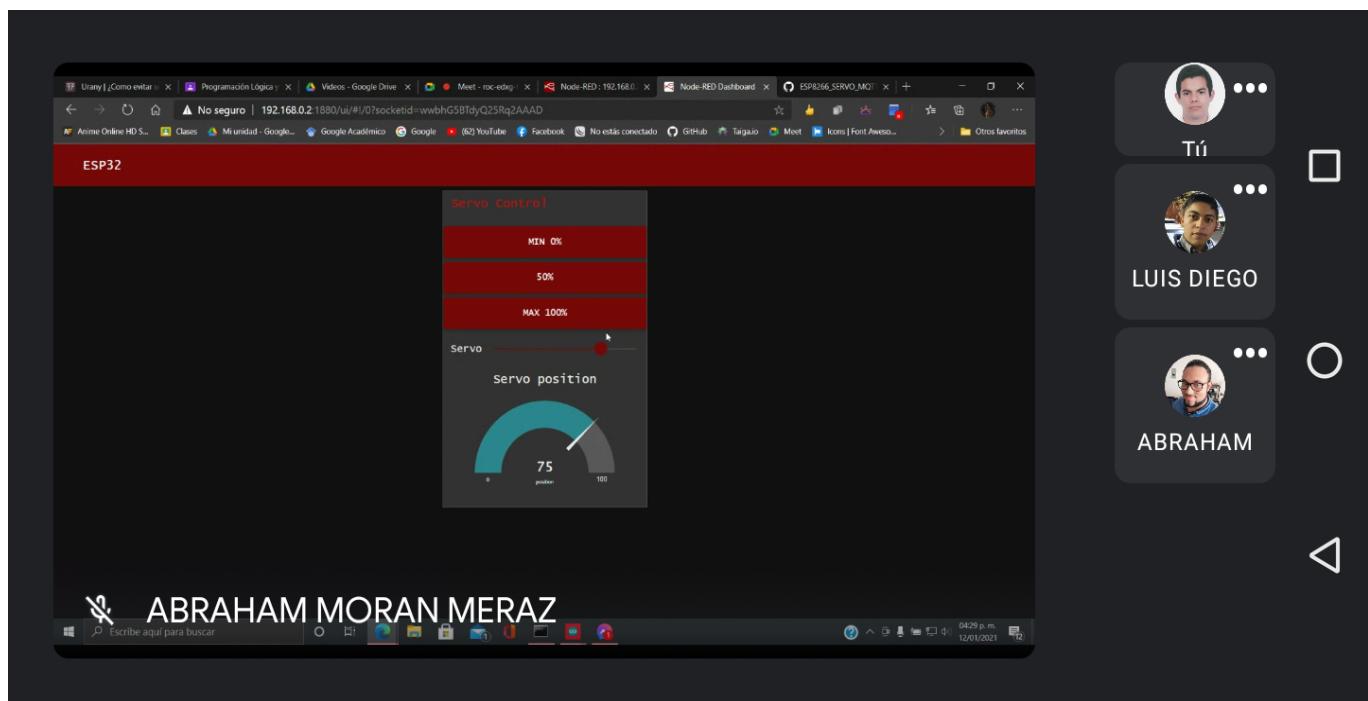
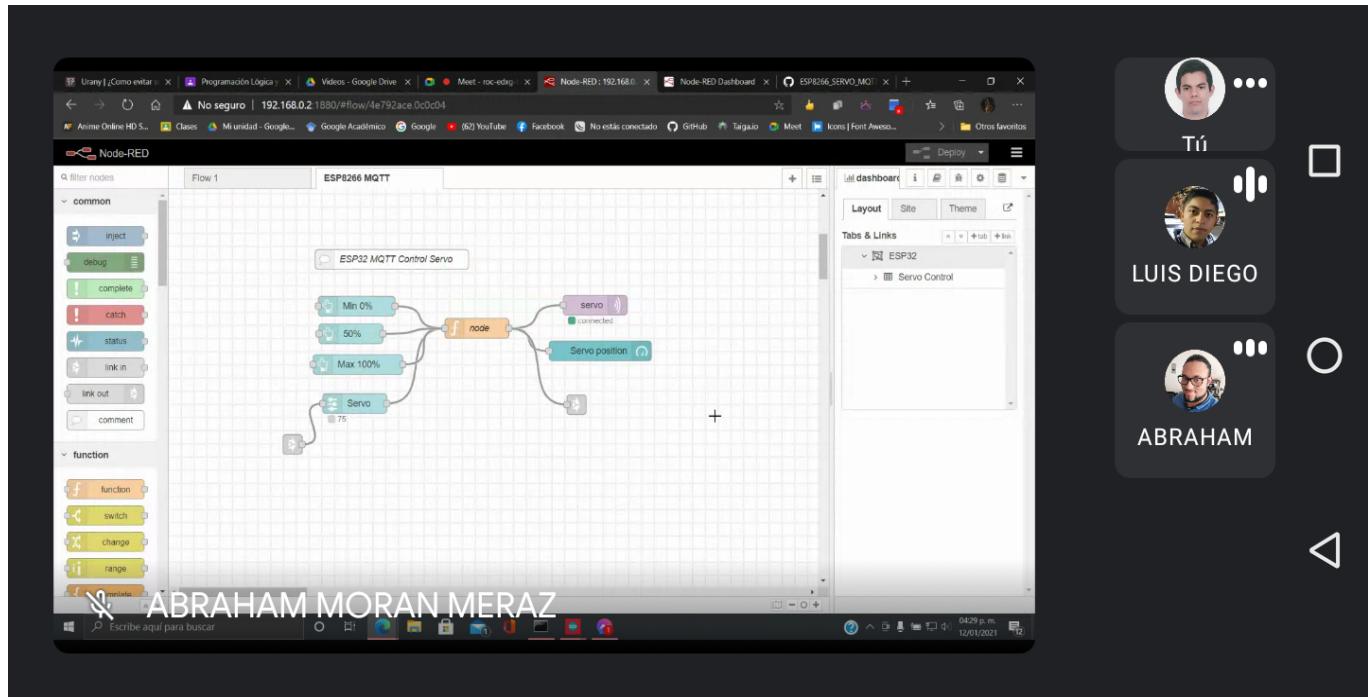
You flow credentials file is encrypted using a system-generated key.

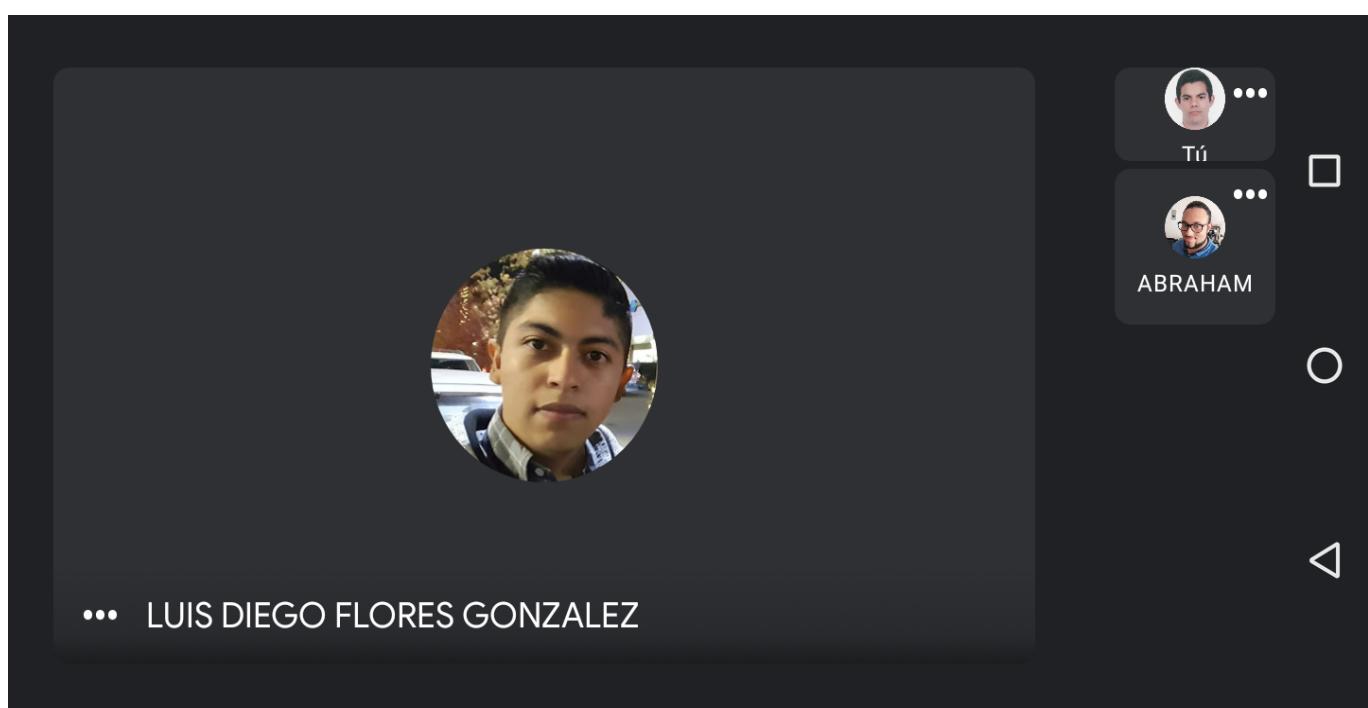
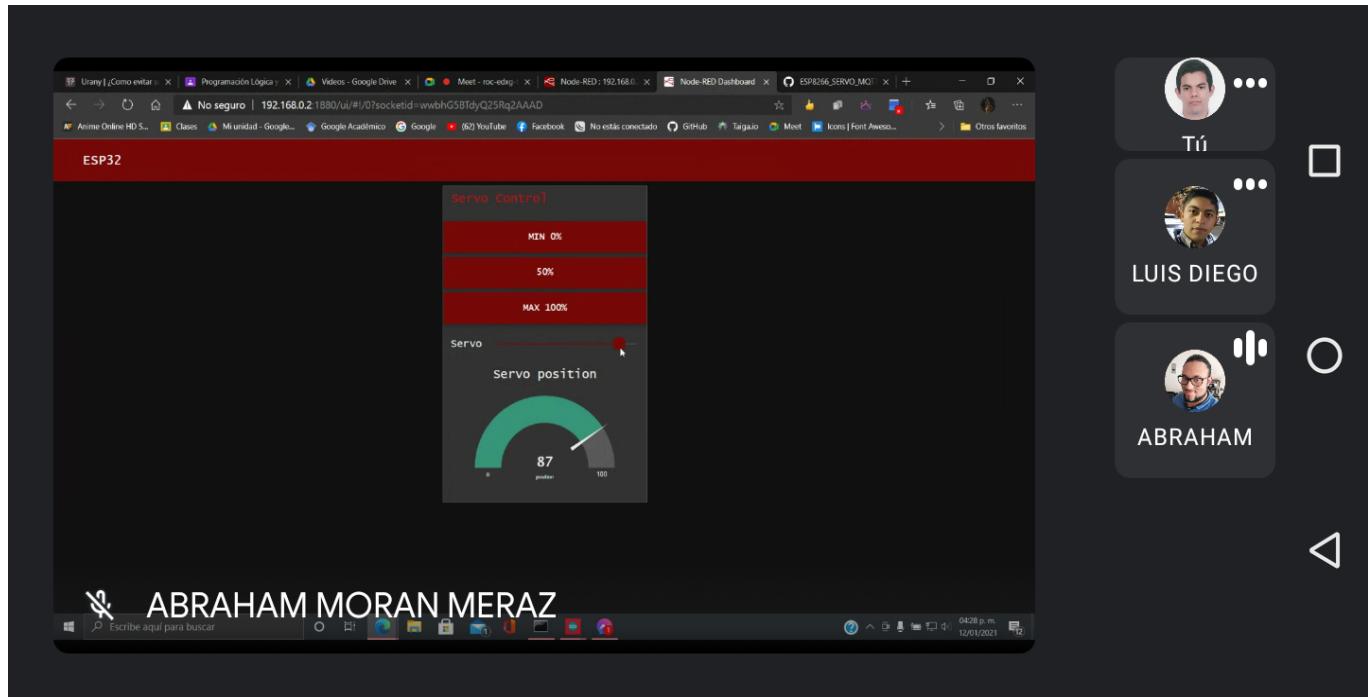
If the system-generated key is lost for any reason, your credentials file will not be recoverable, you will have to delete it and re-enter your credentials.

You should set your own key using the 'credentialSecret' option in your settings file. Node-RED will then re-encrypt your credentials file using your chosen key the next time you deploy a change.

12 Jan 12:44:00 - [info] Starting flows  
12 Jan 12:44:00 - [info] Started flows  
12 Jan 12:44:00 - [info] Server now running at http://127.0.0.1:1880/  
12 Jan 12:44:00 - [info] [mqtt-broker:10e78a89.5b4fd5] Connected to broker: mqtt://localhost:1883







## 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



>>>>Enlace a mi GitHub: Diego Flores<<<<



>>>>Enlace a mi GitHub: Abraham  
Moran<<<<



>>>>Enlace a mi GitHub: Eduardo Ruiz<<<<