



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.

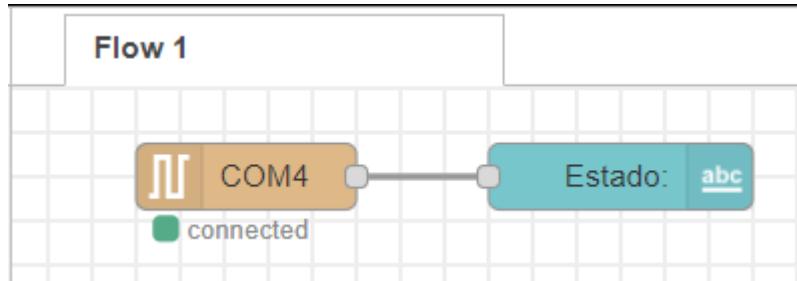
```
| readme.md  
| | blog  
| | | C5.1_TituloActividad.md  
| | | C5.2_TituloActividad.md  
| | IMG  
| | docs  
| | | A5.1_TituloActividad.md  
| | | A5.2_TituloActividad.md
```



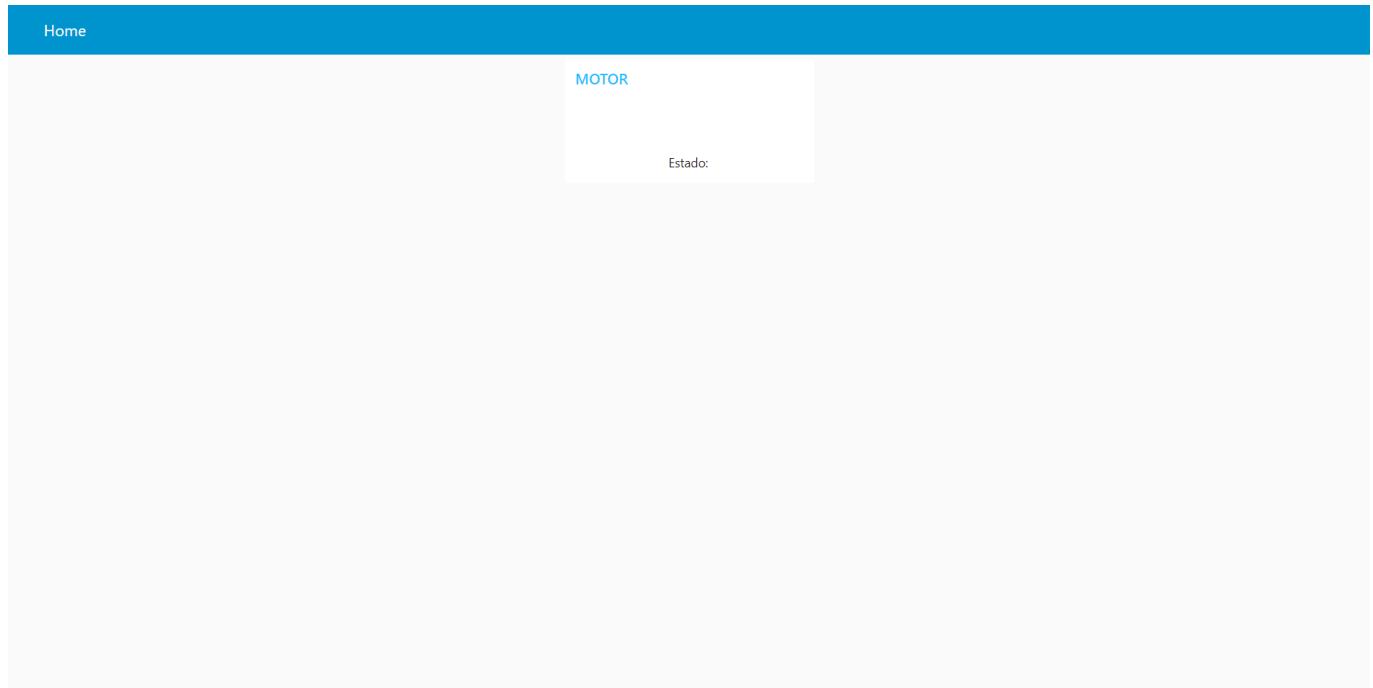
Desarrollo

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.

```
int motor1Pin1 = 27;
int motor1Pin2 = 26;
int enable1Pin = 14;

// Propiedades PWM
const int freq = 30000;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 200;

void setup() {
    // Pines de salida
    pinMode(motor1Pin1, OUTPUT);
    pinMode(motor1Pin2, OUTPUT);
    pinMode(enable1Pin, OUTPUT);

    // Configuración LED PWM
```

```
ledcSetup(pwmChannel, freq, resolution);
ledcAttachPin(enable1Pin, pwmChannel);

Serial.begin(115200);
Serial.print("Inicio de Motor");
}

void loop() {
    ledcWrite(pwmChannel, dutyCycle);
    dutyCycle = 255;

    // Forward
    Serial.println("Motor en Movimiento");
    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, HIGH);
    delay(5000);

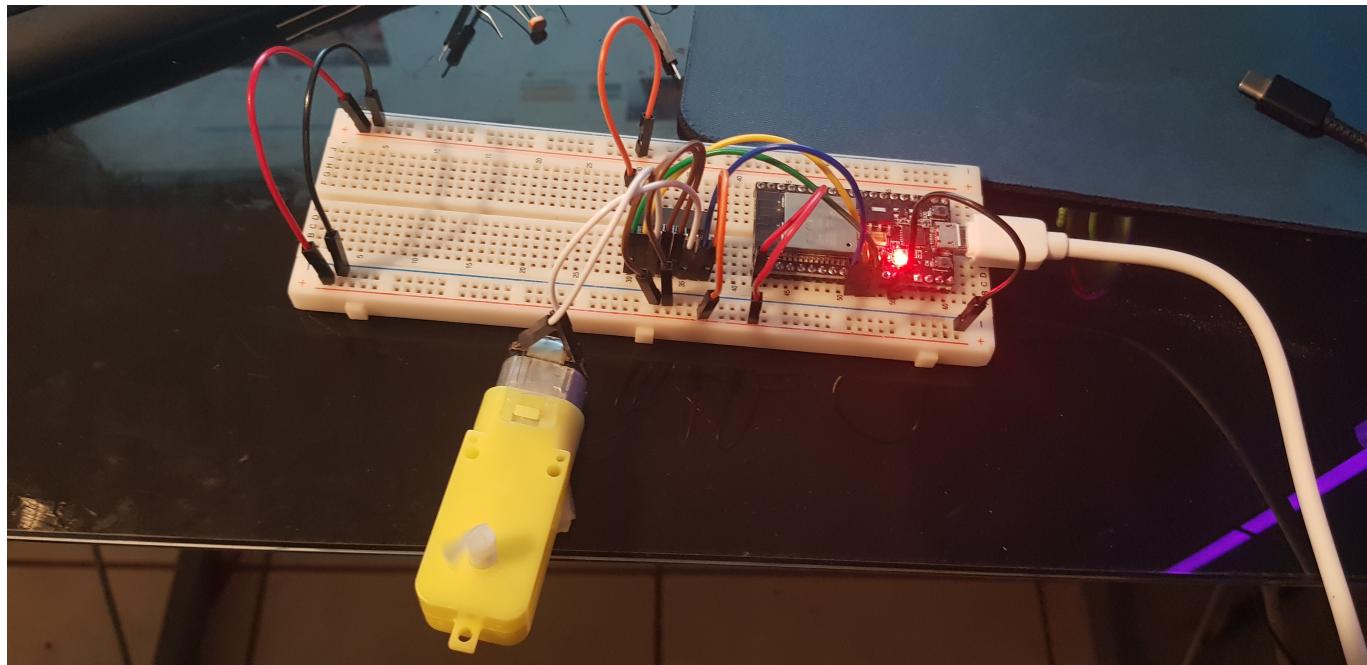
    // Se detiene el motor
    Serial.println("Motor Detenido");
    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, LOW);
    delay(1000);

    // Reversa
    Serial.println("Motor en Reversa");
    digitalWrite(motor1Pin1, HIGH);
    digitalWrite(motor1Pin2, LOW);
    delay(5000);

    // Se detiene el motor
    Serial.println("Motor Detenido");
    digitalWrite(motor1Pin1, LOW);
    digitalWrite(motor1Pin2, LOW);
    delay(1000);

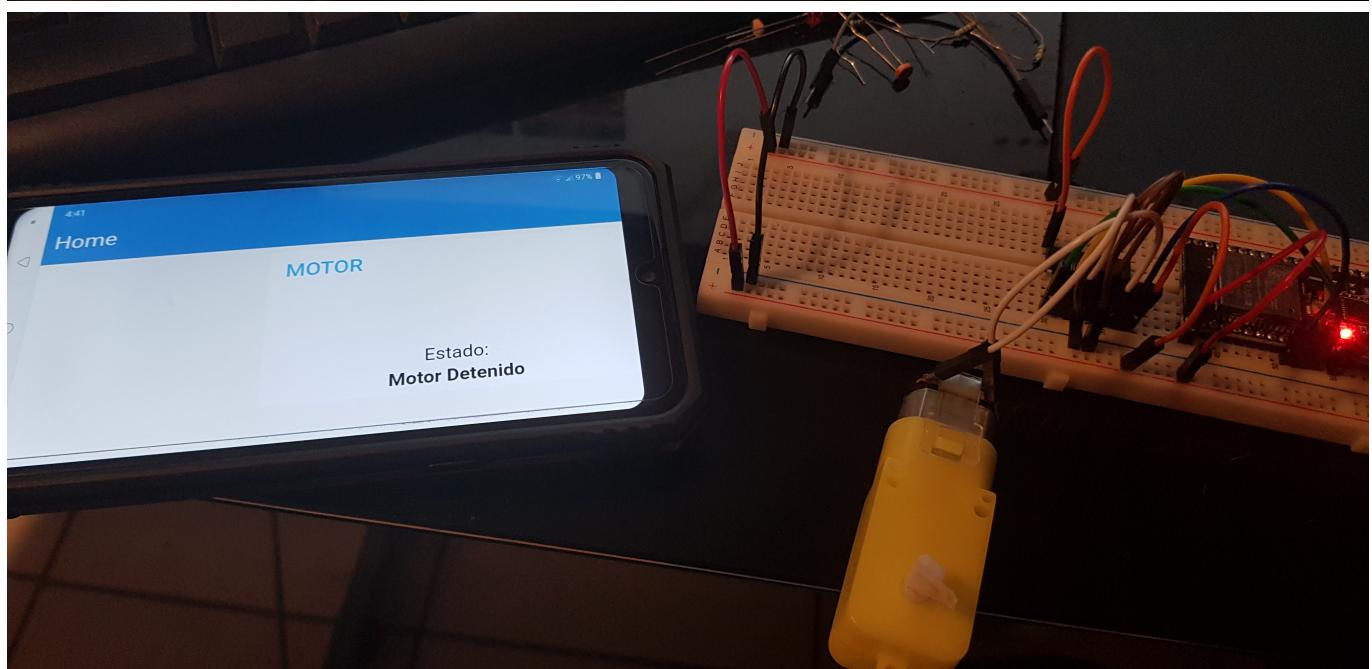
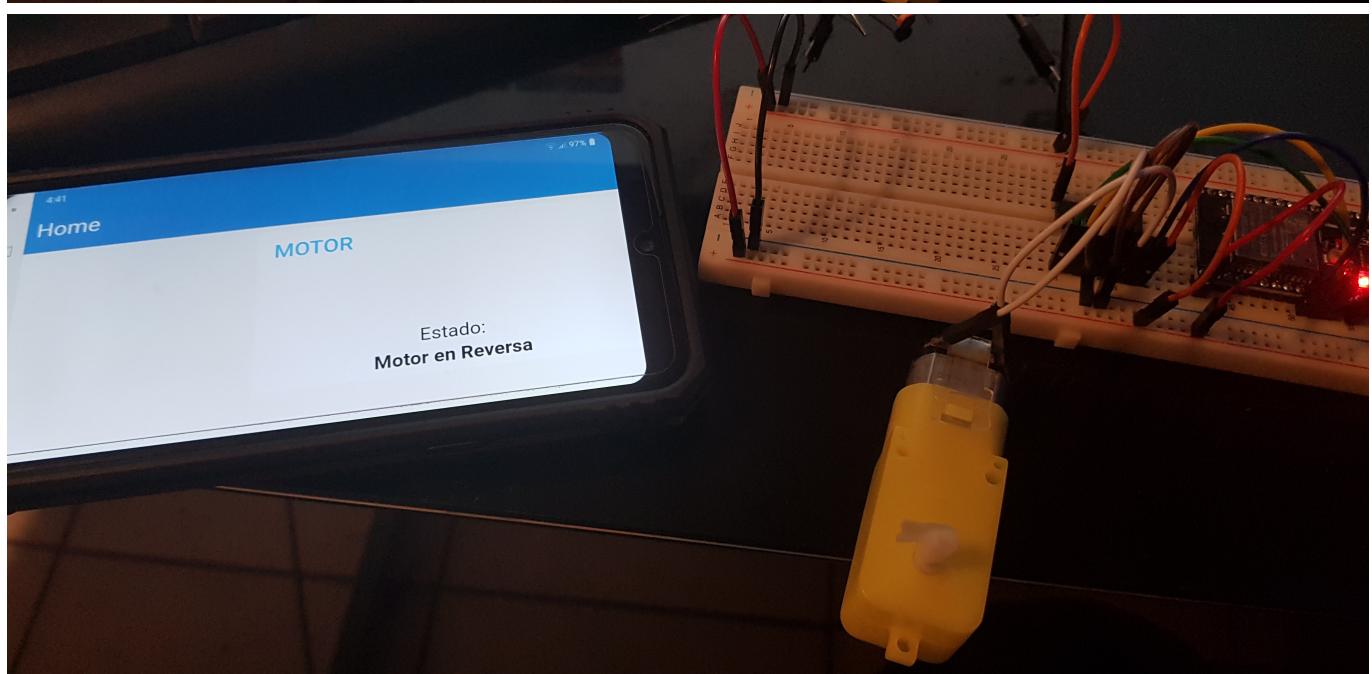
}
```

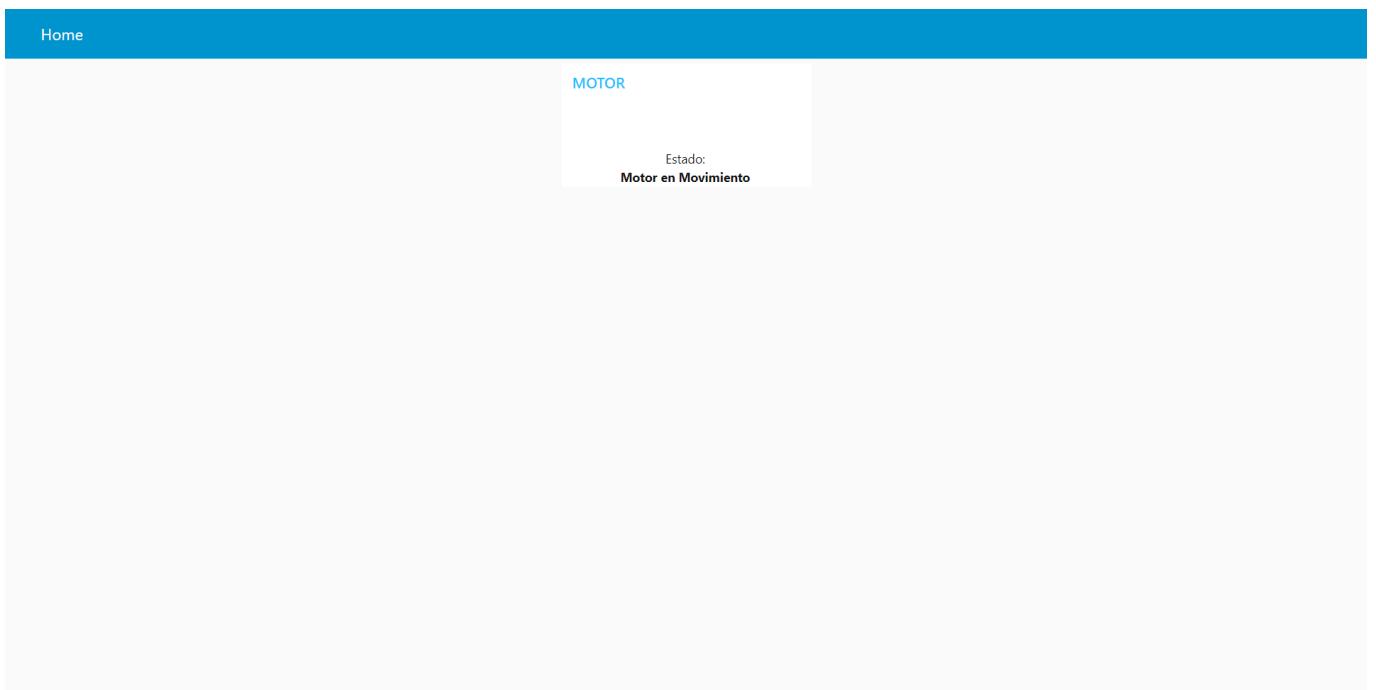
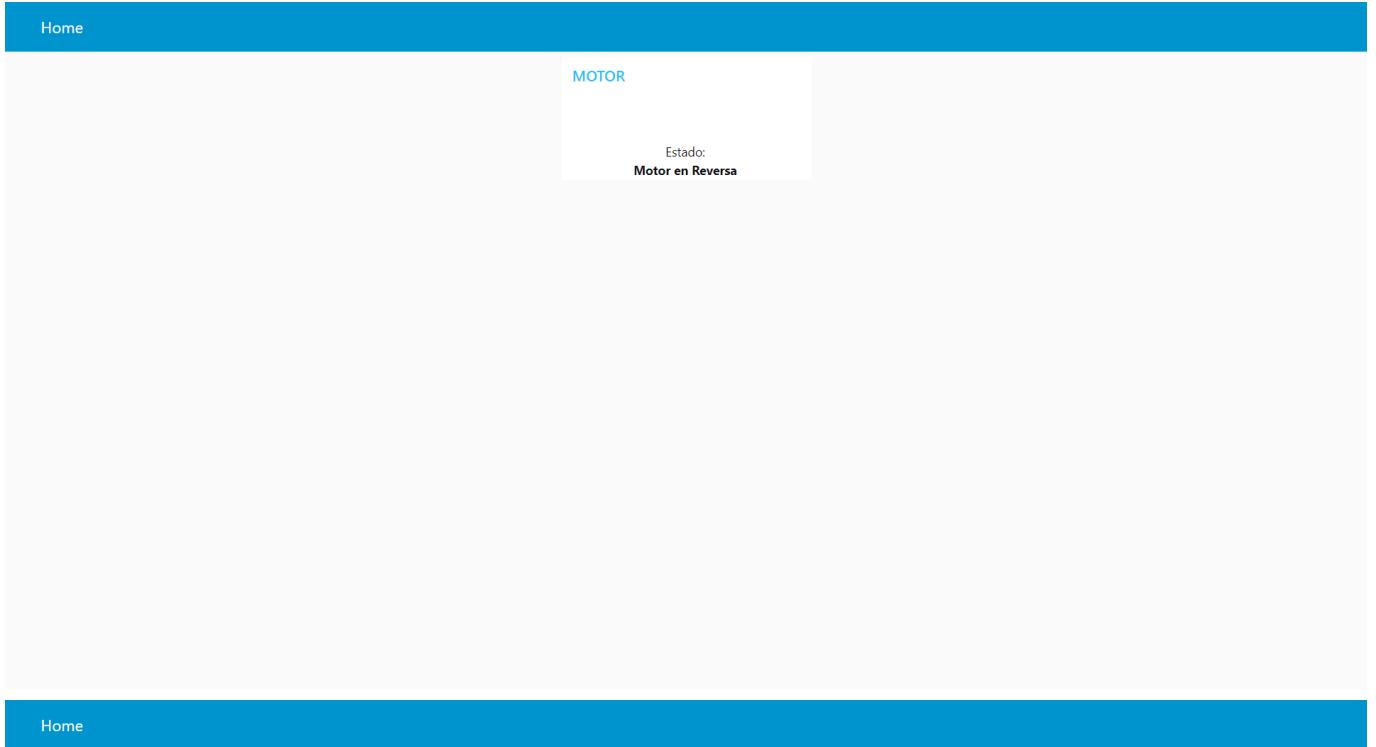
4. Agregue imágenes del circuito electrónico ensamblado.



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

- Evidencia física realizada por: **Jose Alfredo Venegas Medina**
- [Presentacion del circuito \(VIDEO\)](#)





Facebook FINAL - Goog Avance Final PROYECTO FIN PROYECTO FIN (Scala) | Online SWISH -- SWI-

meet.google.com/dzw-vqtu-obz

Jose Venegas está presentando

18:32 Tú

```

CS2 Arduino 1.0.0
Archivo, Editar, Programa, Herramientas, Ayuda
G87
int motor1Pin1 = 27;
int motor1Pin2 = 26;
int enable1Pin = 14;

// Propiedades PWM
const int freq = 30000;
const int pwmChannel = 0;
const int resolution = 8;
int dutyCycle = 200;

void setup() {
    // Pines de salida
    pinMode(motor1Pin1, OUTPUT);
    pinMode(motor1Pin2, OUTPUT);
    pinMode(enable1Pin, OUTPUT);

    // Configuración LED PWM
    ledcSetup(pwmChannel, freq, resolution);
    ledcAttachPin(enable1Pin, pwmChannel);

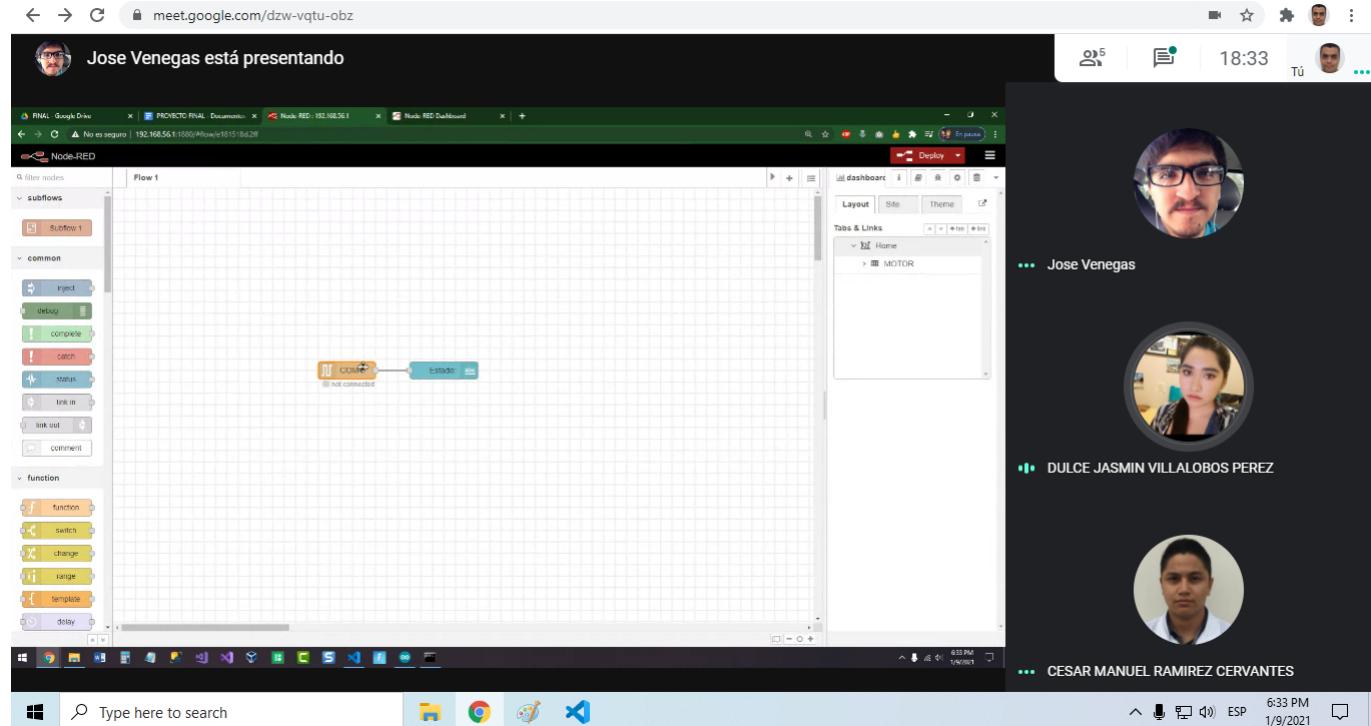
    Serial.begin(115200);
    Serial.print("Inicio de Motor");
}

```

Windows taskbar: Facebook, FINAL - Goog, Avance Final, PROYECTO FIN, PROYECTO FIN, (Scala) | Online, SWISH -- SWI-

Type here to search

6:32 PM 1/9/2021



A screenshot of a Google Meet video call interface. The main video frame shows a breadboard with various components and wires, connected to a smartphone. The phone's screen displays a mobile application with the title "MOTOR" and the status "Estado: Motor en Reversa". The video call interface includes a participant list on the right side with five users: CESAR MANUEL RAMIREZ CERVANTES, DULCE JASMIN VILLALOBOS PEREZ, CESAR MANUEL RAMIREZ CERVANTES, and Jose Venegas. The browser tab bar at the top shows multiple open tabs related to projects and meetings.

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 - MI GITHUB](#)