🏆 A4.2 Learning activity

Circuito sensor de iluminación y actuación de luz NodeMCU ESP32 por medio de Wi-Fi

Instructions

- Realize an ambient light measurement and actuation system, using the **Wi-FI** communication protocol, as well as a NodeMCU **ESP32**, a **LDR** sensor, and an LED diode.
- Any activity or challenge must be carried out using the **MarkDown style with .md** extension and the VSCode development environment, and must be prepared as a **single page** document, that is, if the document has images, links or Any external document must be accessed from tags and links, and must be named with the nomenclature **A4.1_TituloAactivity_NombreAlumno.pdf.**
- It is a requirement that the .md contains a tag of the link to the repository of your document in GITHUB, for example **Link to my GitHub** and at the end of the challenge it should be uploaded to github.
- From the **.md** file, export a **.pdf** file that must be uploaded to classroom within its corresponding section, serving as evidence of its delivery, since being the **official** platform, the rating of your activity.
- Considering that the .PDF file, which was obtained from the .MD file, both must be identical.
- Your repository in addition to having a **readme**.md file in your root directory, with information such as student data, work team, subject, career, advisor data, and even logo or images, it must have a section of contents or index, which really are links or **links to your .md documents** , _ avoid using text_ to indicate internal or external links.
- A structure is proposed as indicated below, however any other that supports you can be used to organize your repository.

```
- readme.md
  - blog
    - C4.1_TituloActividad.md
    - C4.2_TituloActividad.md
    - C4.3_TituloActividad.md
    - C4.4_TituloActividad.md
  - img
  - docs
    - A4.1_TituloActividad.md
    - A4.2_TituloActividad.md
    - A4.3_TituloActividad.md
```

Sources of support to develop the activity

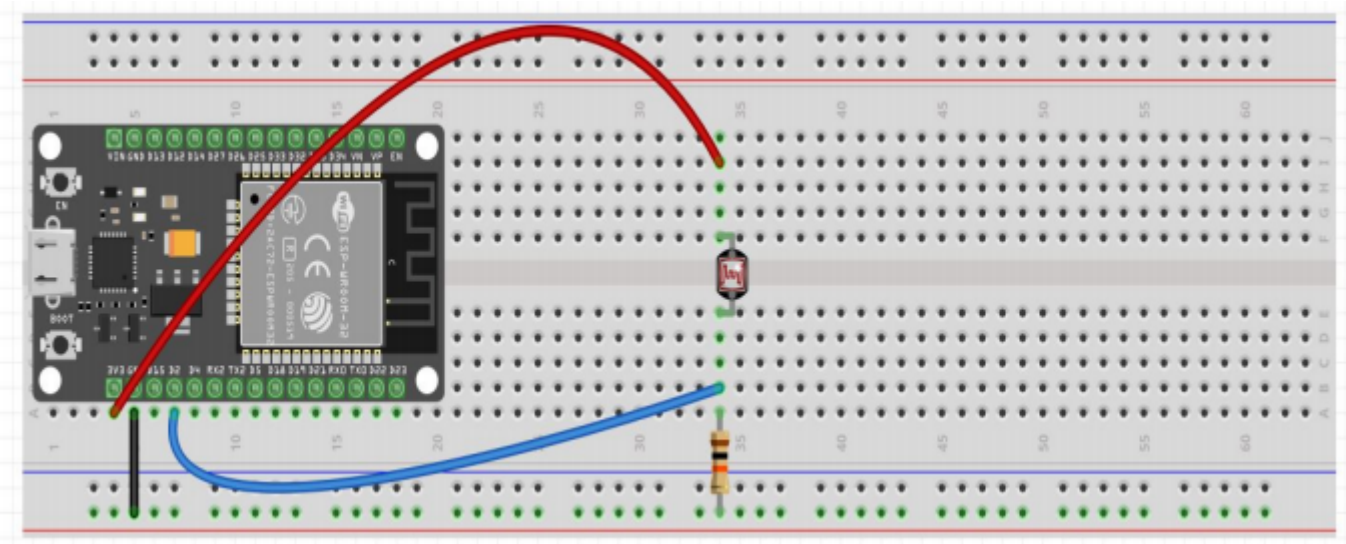
- ☑ ESP32 Web Server
- ☑ LDR with ESP32

Development
1.Use the following list of materials to prepare the activity

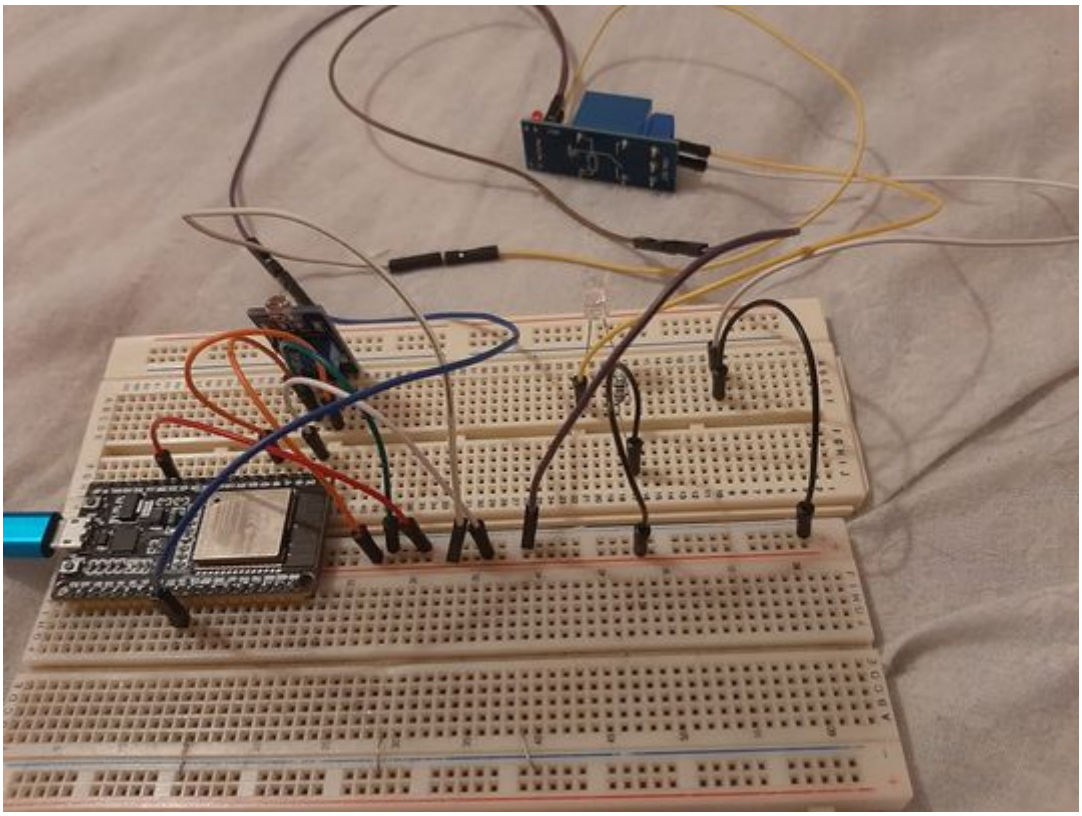
Quantity	Description
1	LDR Sensor module
1	5V voltage source
1	NodeMCU ESP32
1	BreadBoard
1	Jumpers M/M
1	3.3v Relay
1	Light Bulb

2. Based on the image shown in Figure 1, assemble an electronic circuit and add a Led to one of the device's terminals, in such a way that a system capable of comply with the following instructions:
    - The NodeMCU should be used as a standalone Web server, which should provide a visual interface, which will show an image representing an "ON and OFF" behavior depending on the ambient lighting condition.
    - The lighting sensor should be measuring the amount of light that exists in the environment.
    - When detecting the LDR sensor, the absence of light should show an "ON" status on the Web interface, and otherwise the "OFF" status should be represented.
    - A relay in series with a light bulb should be added to the circuit in figure 1, which will be activated in the opposite condition to the "ON" and "OFF" state, that is, once the sensor detects little or no lighting, it will activate the relay and therefore it will light the bulb, and otherwise it will turn it off.

Source of support Random Nerd Tutorial



3. Place the picture of the assembled circuit here



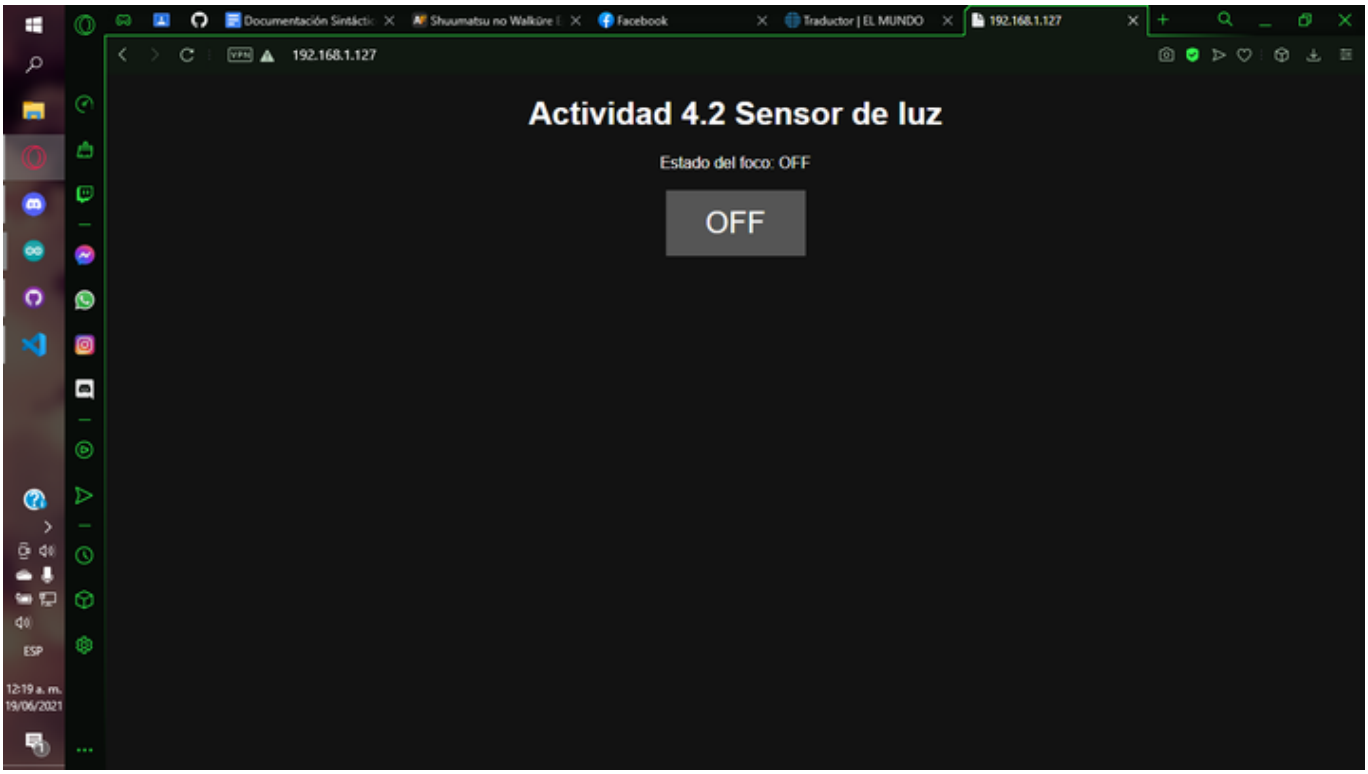
4. Place the program created within the Arduino environment in this place

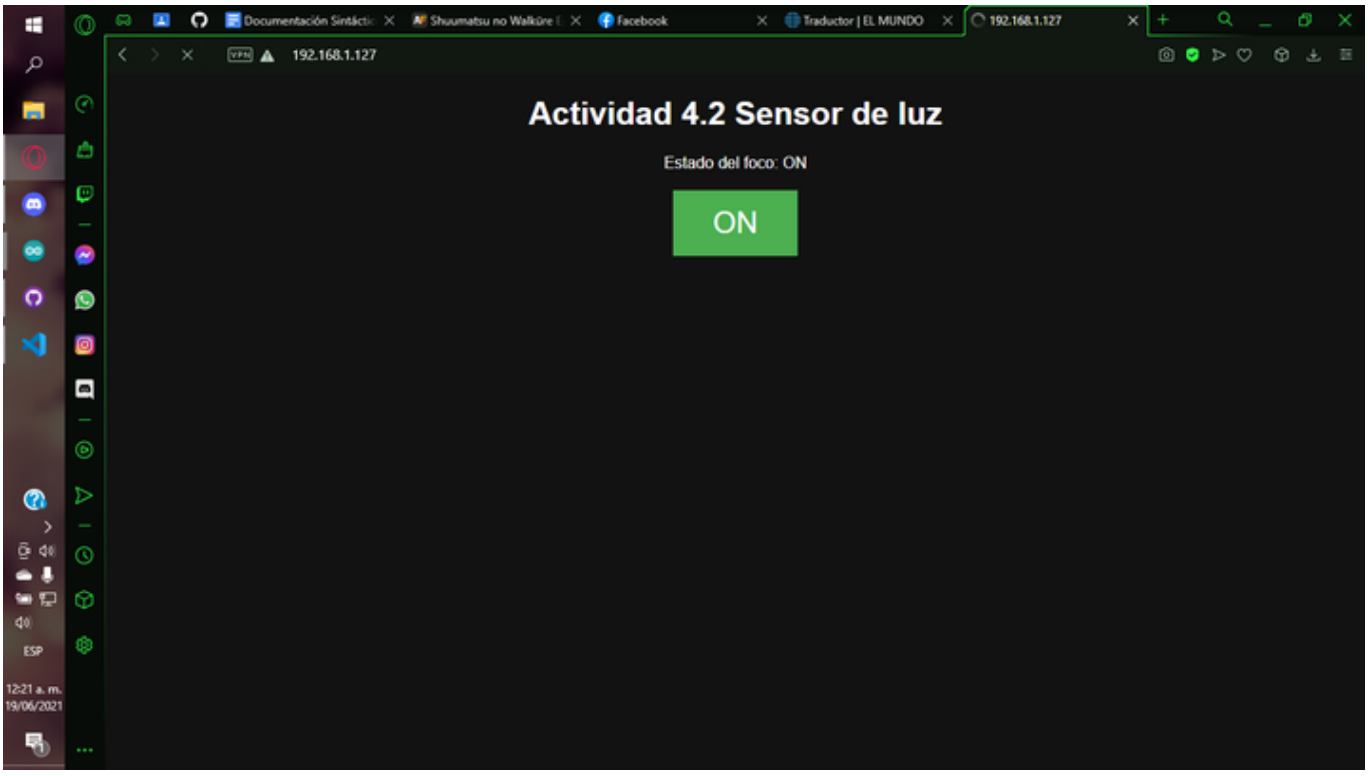
```
#include <WiFi.h>
//Servidor web en puerto 80
WiFiServer server(80);

//Datos de la red wifi
const char* ssid = "INFINITUM537D";
const char* password = "1263743512";

//Variables globales
int contConexion = 0;
String header; //Variable para guardar el HTTP request
const int Ent = 34; //Pin de la entrada del sensor
const int Sal = 5; //Pin salida al relay

//Codigo HTML
String PAG = "<!DOCTYPE html><html>"
"<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">"
"<META HTTP-EQUIV='Refresh' CONTENT='1'>"
"<link rel=\"icon\" href=\"data:,\">"
"<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto;
text-align: center;}"
".button { background-color: #4CAF50; border: none; color: white; padding: 16px
40px;"
"text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}"
".button2 {background-color: #555555;}</style></head>"
"<body><h1>Actividad 4.2 Sensor de luz </h1>";

String ON = "<p> Estado del foco: ON </p>"
```

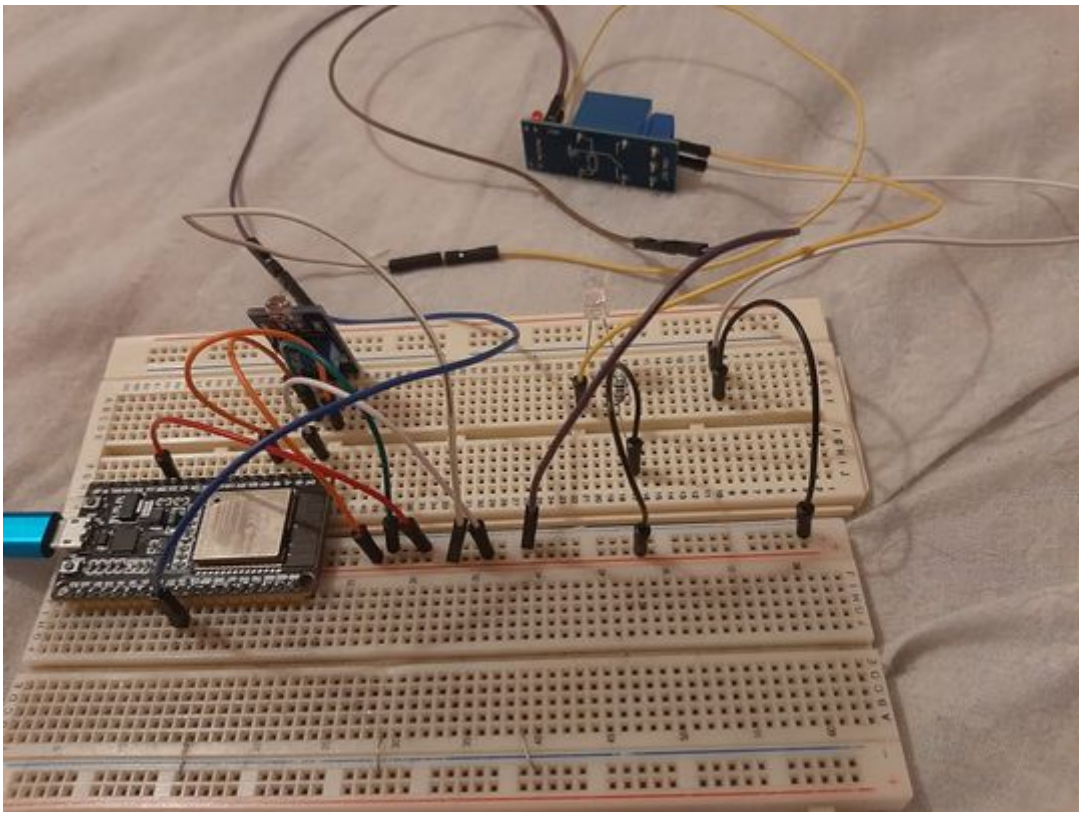
```
"<p><a href=\"/26/on\"><button class=\"button\">ON</button></a></p>";
String END = "</body></html>";

String OFF = "<p> Estado del foco: OFF </p>"
"<p><a href=\"/26/off\"><button class=\"button button2\">OFF</button></a></p>";


void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  Serial.println("");

  pinMode(Ent,INPUT);
  pinMode(Sal,OUTPUT);
  digitalWrite(Sal,LOW);
  //Conexion wifi
  WiFi.begin(ssid,password);
  //Cuenta hasta 50 si no se puede conectar lo cancela
  while(WiFi.status() != WL_CONNECTED and contConexion < 50)
  {
    ++contConexion;
    delay(500);
    Serial.print(".");
  }
  if(contConexion<50)
  {
      Serial.println("");
      Serial.println("WiFi conectado");
      Serial.println(WiFi.localIP());
      server.begin(); // Iniciamos el servidor
  }
  else
  {
     Serial.println("");
     Serial.println("Error de conexion");
  }
}

void loop() {
  // put your main code here, to run repeatedly:
  WEB(analogRead(Ent));
}


void WEB(int LDR){
  WiFiClient client = server.available();   // Escucha a los clientes entrantes

  if (client) {                             // Si se conecta un nuevo cliente
    Serial.println("New Client.");          //
    String currentLine = "";                //
    while (client.connected()) {            // loop mientras el cliente está
conectado
      if (client.available()) {             // si hay bytes para leer desde el
cliente
```

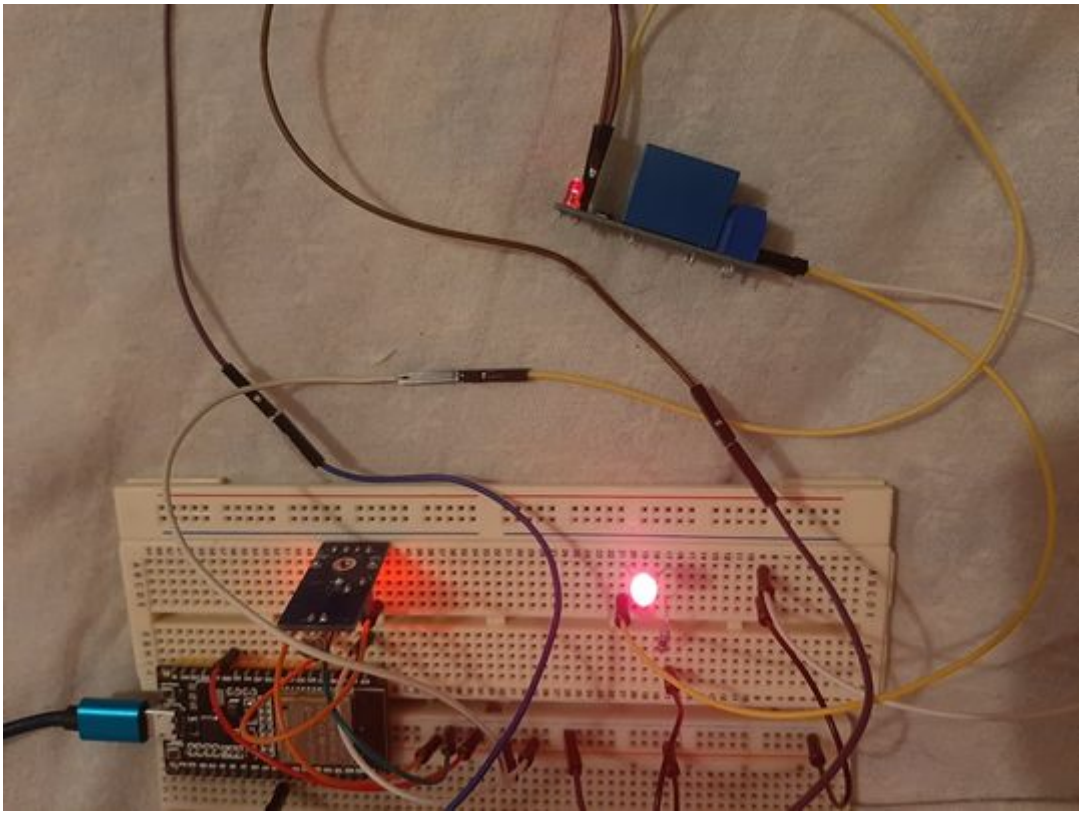
```
        char c = client.read();             // lee un byte
        Serial.write(c);                    // imprime ese byte en el monitor
serial
        header += c;
        if (c == '\n') {                    // si el byte es un caracter de salto
de linea
          // si la nueva linea está en blanco significa que es el fin del
          // HTTP request del cliente, entonces respondemos:
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println("Connection: close");
            client.println();

            client.println(PAG);

            if(LDR>2000)
            {
              Serial.println("ON");
              digitalWrite(Sal,LOW);
              client.println(ON);
            }
            else
            {
              Serial.println("OFF");
              digitalWrite(Sal,HIGH);
              client.println(OFF);
            }

             client.println(END);
            // la respuesta HTTP temina con una linea en blanco
              client.println();
              break;
          } else { // si tenemos una nueva linea limpiamos currentLine
            currentLine = "";
          }
        } else if (c != '\r') {  // si C es distinto al caracter de retorno de
carro
          currentLine += c;      // lo agrega al final de currentLine
        }
      }
    }
    // Limpiamos la variable header
    header = "";
    // Cerramos la conexión
    client.stop();
    Serial.println("Client disconnected.");
    Serial.println("");
  }
}
```

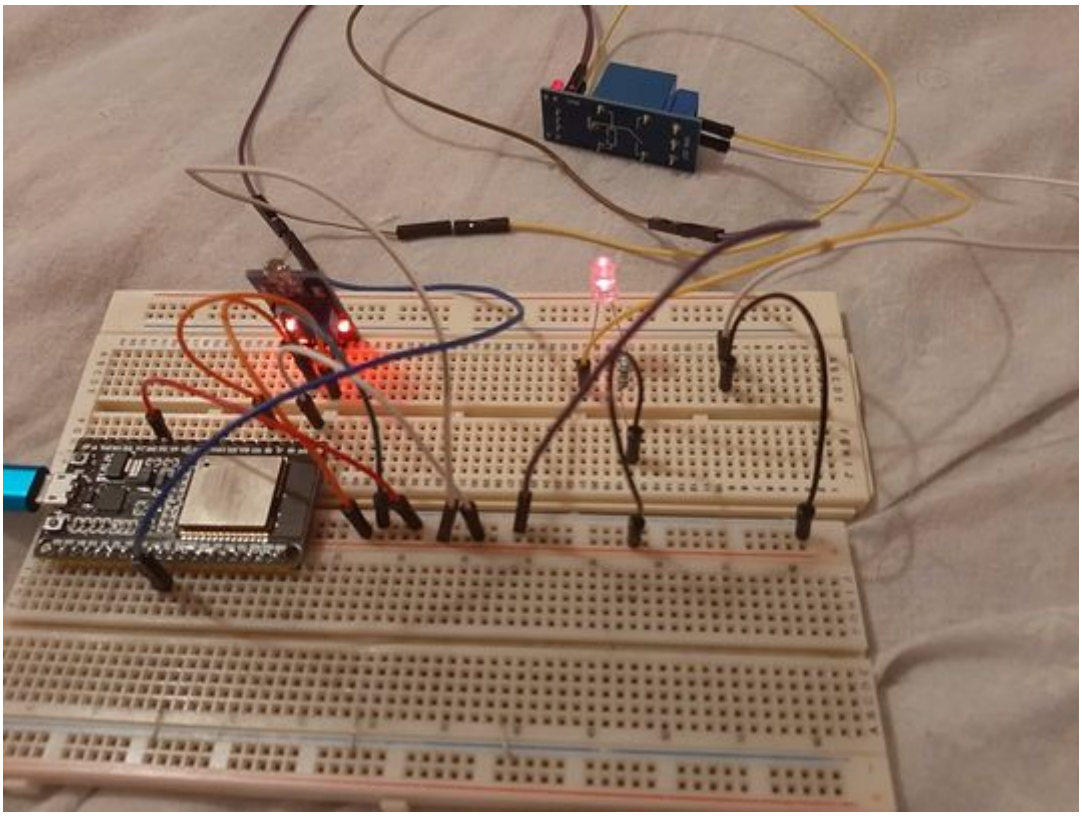
5. Place here evidence that you consider important during the development of the activity.


192.168.1.127
Actividad 4.2 Sensor de luz
Estado del foco: OFF
OFF

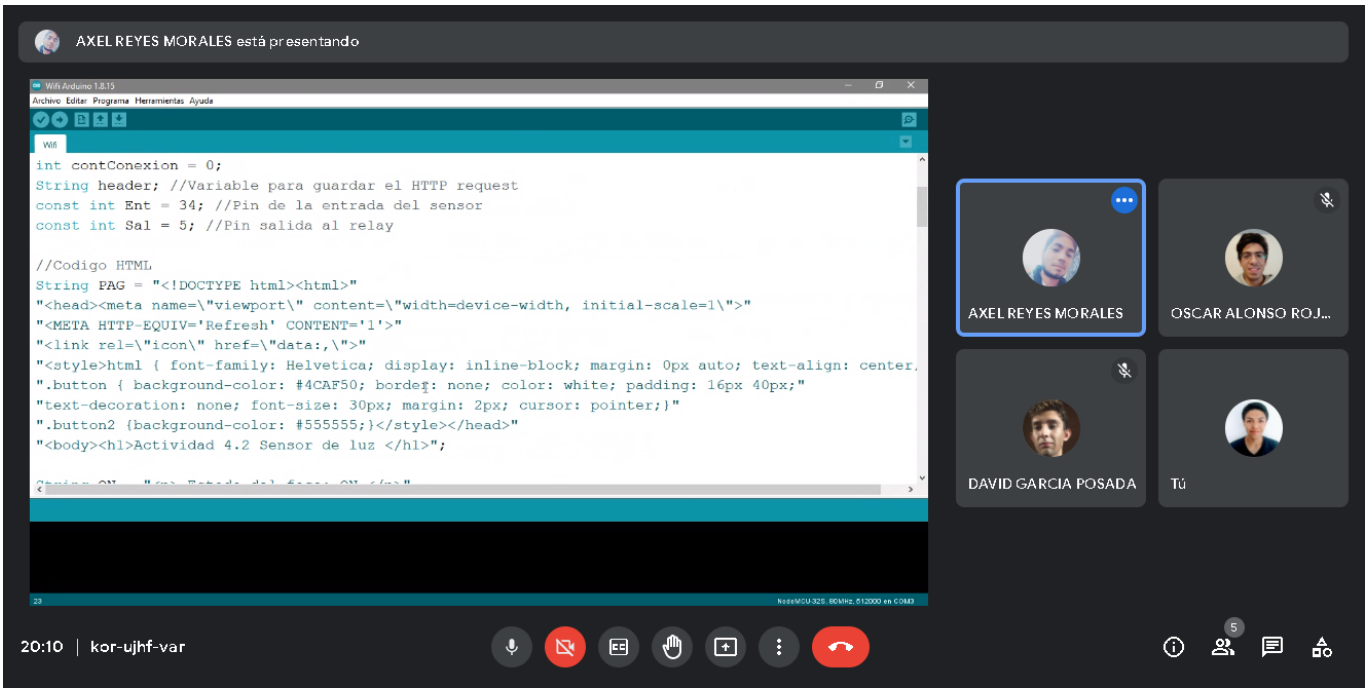

192.168.1.127
Actividad 4.2 Sensor de luz
Estado del foco: ON
ON

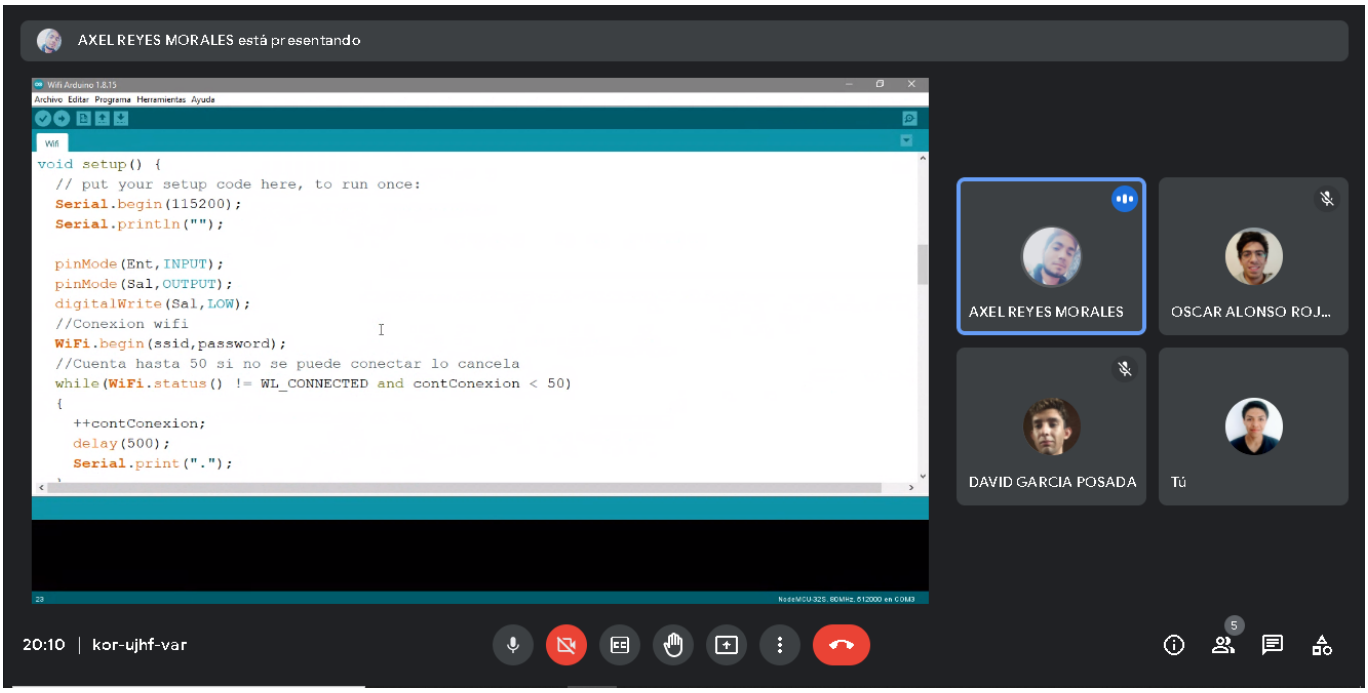


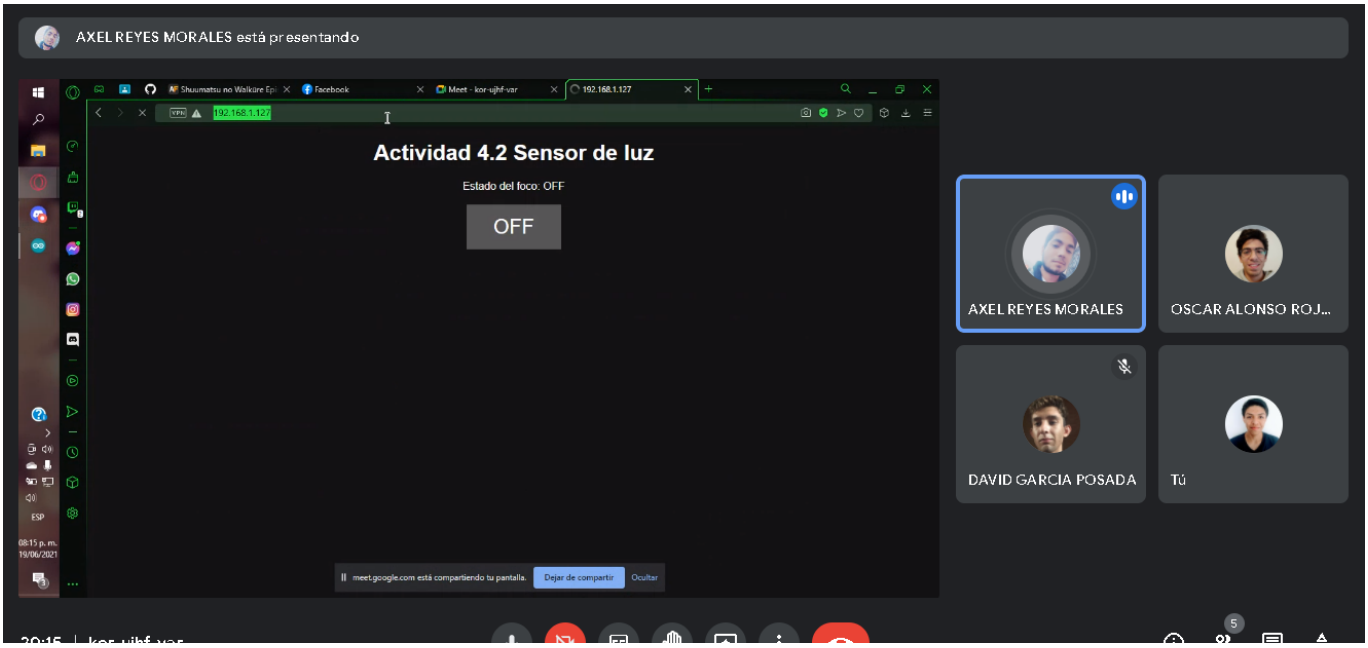


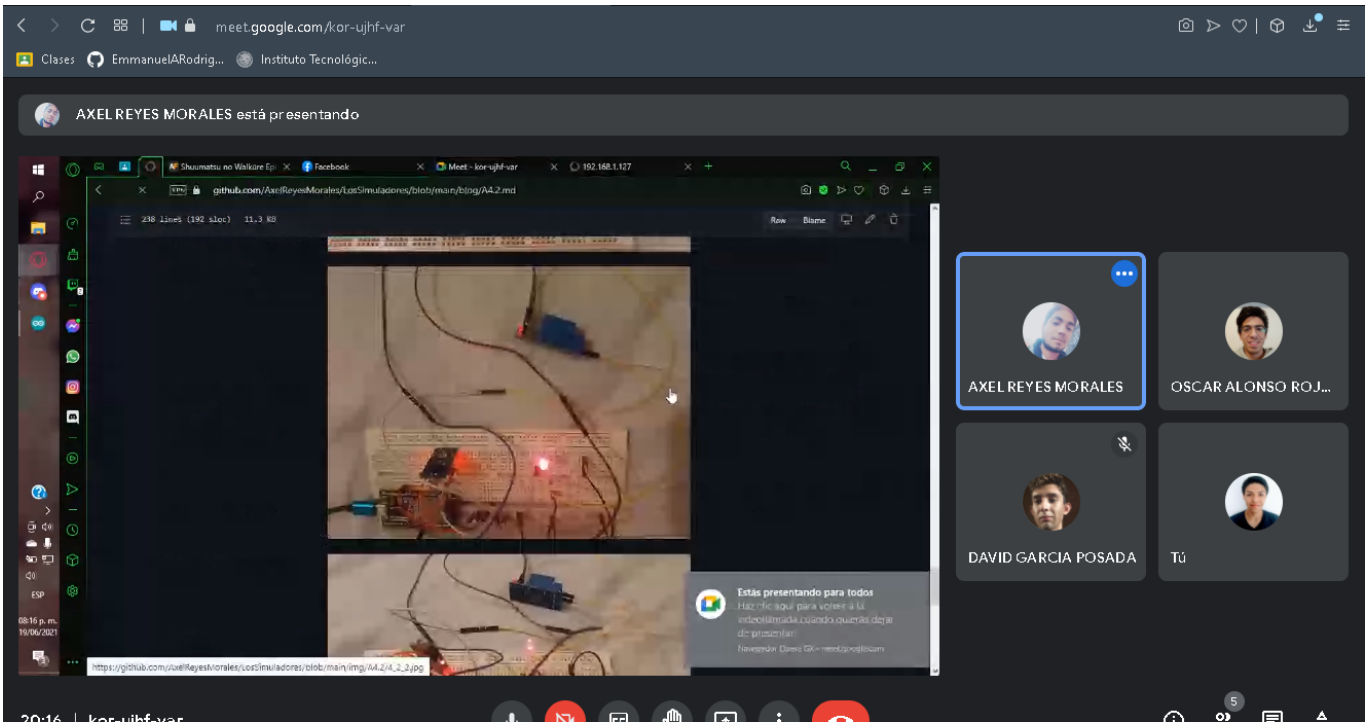


6. Insert images of **evidence** such as meetings of the team members held for the development of the activity.




AXEL REYES MORALES está presentando
Wifi Arduino 1.8.15
Archivo Editar Programa Herramientas Ayuda
Wifi
void setup() {
// put your setup code here, to run once:
Serial.begin(115200);
Serial.println("");
pinMode(Ent,INPUT);
pinMode(Sal,OUTPUT);
digitalWrite(Sal,LOW);
//Conexion wifi
WiFi.begin(ssid,password);
//Cuenta hasta 50 si no se puede conectar lo cancela
while(WiFi.status() != WL_CONNECTED and contConexion < 50)
{
++contConexion;
delay(500);
Serial.print(".");
AXEL REYES MORALES
OSCAR ALONSO ROJ...
DAVID GARCIA POSADA
Tú
20:10 | kor-ujhf-var


AXEL REYES MORALES está presentando
192.168.1.127
Actividad 4.2 Sensor de luz
Estado del foco: OFF
OFF
meet.google.com está compartiendo tu pantalla.
Dejar de compartir
Ocultar
AXEL REYES MORALES
OSCAR ALONSO ROJ...
DAVID GARCIA POSADA
Tú



Conclusions

Axel: In conclusion I can say that in this practice the implementation of a web service was approved, which was created from the ESP32, where we would be observing the state in which the focus was to be turned on by means of the LDR light sensor, because if the light sensor marked the absence of the light the order should be sent that the focus was turned on and on the page would be turned on, otherwise the focus would stay off and on the page would be turned off in the same way.

David: during the development of this activity we were able to connect a LDR via wifi with the esp32 and with this we could observe the way it works the connection and how to pass information to a web page, during this activity we found some difficulties for connecting the esp32 with the wifi and pass information but via some reserch we were able to resolve the problem

Emmanuel: In conclusion, a circuit was made where a lighting sensor and an ESP32 were used where the Wi-fi function was implemented, where together with a web server the lighting sensor would detect if there was darkness or absence of light and a bulb would light and it would send a signal to the web service so that it would mark that it would turn on. If there was light, the bulb would not turn on and the web service would mark that the bulb is off.

Oscar Rojas: In this activity we make use of the ESP32 to start a web server, wich can be accessed by the devices in the same network as the ESP32, in this case we connected the ESP32 to the network with the built-in Wi-Fi module, the web server hosted a web page with one button, that show if the led connected to the relay, wich is controlled by the ESP32, is on or off, in the circuit we implemented a LDR sensor to measure the amount of light and change the behaviour of the led, when the LDR cant measure light then the led turn on and on the opposite side when it detects light the led turns off, the web page refresh this behavior every time the led its turned on or off.

Repositories

Axel Go to my repository

David Go to my repository

Emmanuel Go to my repository

Oscar Rojas Got to my repository