



## A4.2 Actividad de aprendizaje

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

### Instrucciones

- Realizar un sistema de medición y actuación de luz ambiental, utilizando el protocolo de comunicación **Wi-Fi**, así como un NodeMCU **ESP32**, un sensor **LDR**, un diodo LED.
- Toda actividad o reto se deberá realizar utilizando el estilo **MarkDown con extensión .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, y debe ser nombrado con la nomenclatura **A4.2\_NombreApellido\_Equipo.pdf**.
- Es requisito que el .md contenga una etiqueta del enlace al repositorio de su documento en GITHUB, por ejemplo **Enlace a mi GitHub** y al concluir el reto se deberá subir a github.
- Desde el archivo **.md** exporte un archivo **.pdf** que deberá subirse a classroom dentro de su apartado correspondiente, sirviendo como evidencia de su entrega, ya que siendo la plataforma **oficial** aquí se recibirá la calificación de su actividad.
- Considerando que el archivo **.PDF**, el cual fue obtenido desde archivo **.MD**, ambos deben ser idénticos.
- 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 índice, 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
  - 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
```

### Fuentes de apoyo para desarrollar la actividad

- [ESP32 Web Server](#)
- [LDR con ESP32](#)

## Desarrollo

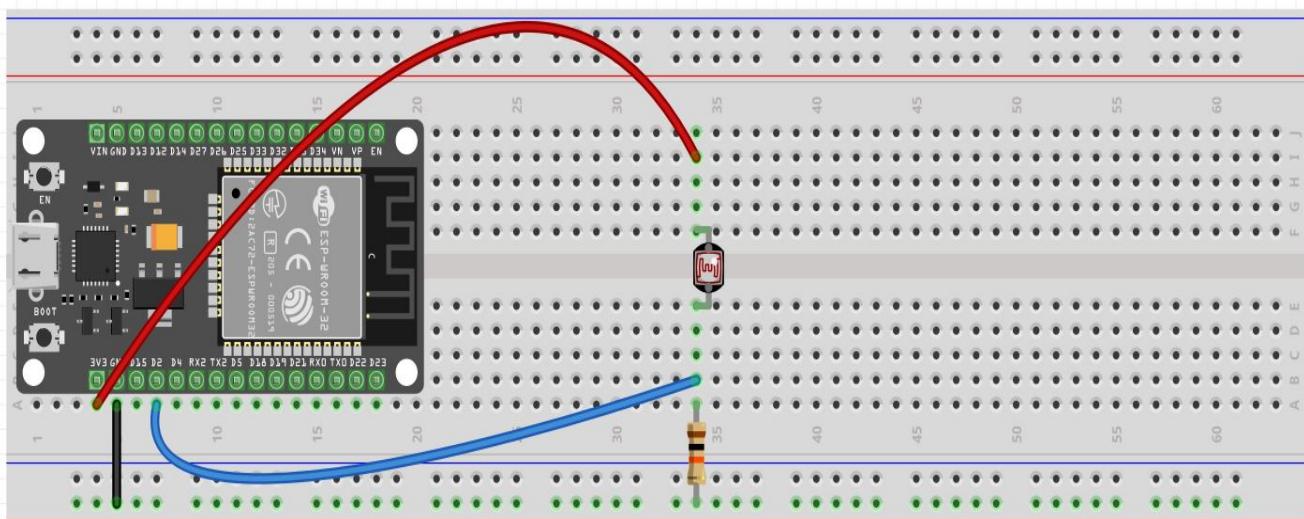
1. Utilizar el siguiente listado de materiales para la elaboración de la actividad

Cantidad	Descripción
1	Modulo Sensor LDR
1	Rele de 3.3v
1	Bombilla
1	Fuente de voltaje de 5V
1	NodeMCU ESP32
1	BreadBoard
1	Jumpers M/M

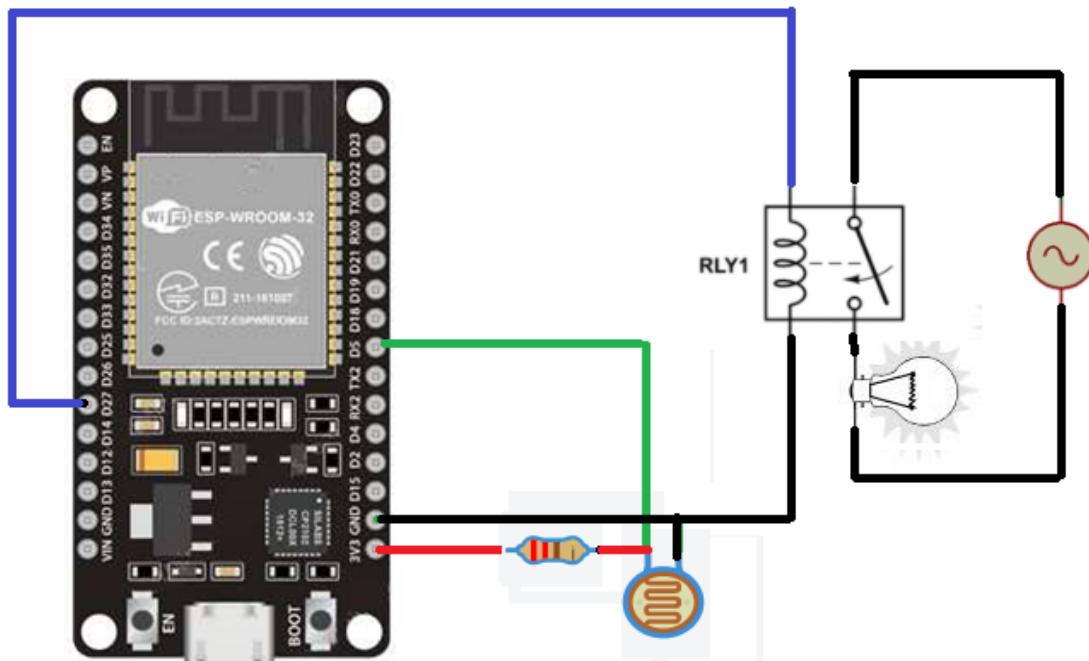
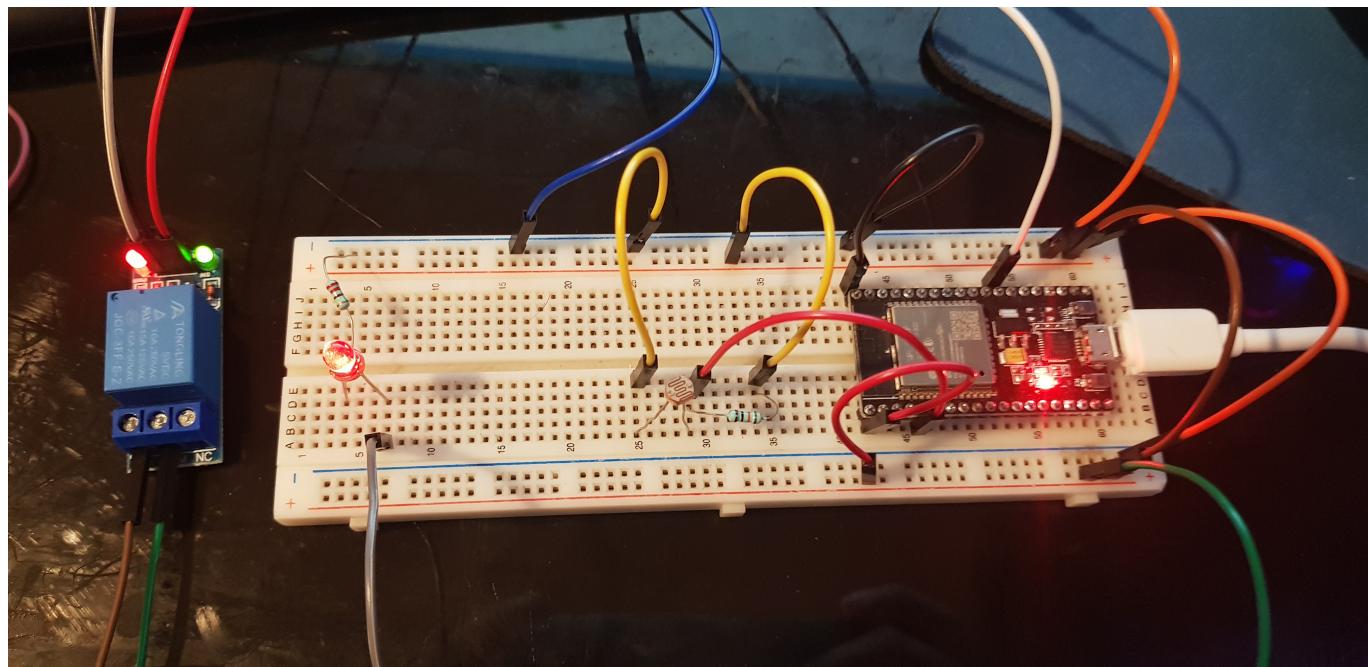
2. Basado en la imagen que se muestran en las **Figura 1**, ensamble un circuito electrónico y agregue un Led y un rele a una de las terminales del dispositivo, de tal manera que se pueda obtener un sistema capaz de cumplir con las instrucciones siguiente:

- Se deberá utilizar el NodeMCU como un **standalone Web server**, el cual deberá proveer una interface visual, que mostrara una imagen representando un comportamiento de "ON y OFF" dependiendo de la condición de iluminación ambiental.
- El sensor de iluminación deberá estar midiendo la cantidad de luz que existen en el ambiente.
- Al momento de detectar el sensor LDR, ausencia de luz deberá mostrar en la interface Web un estado "ON", y en caso contrario deberá representarse el estado "OFF".
- Al circuito de la figura 1 se le deberá agregar un relevador en serie con una bombilla, que sera activada en condición opuesta al estado "ON" y "OFF", es decir una vez que el sensor detecto poca o nula iluminación activara el rele y por consecuencia encenderá la bombilla, y en caso contrario lo apagara.

Fuente de apoyo Random Nerd Tutorial

**Figura 1 Circuito ESP32 Sensor LDR iluminacion**

3. Coloque aquí la imagen del circuito ensamblado



4. Coloque en este lugar el programa creado dentro del entorno de Arduino

```

int LDR = 34;
int LDR_val = 0;
int relay = 16;

void setup() {
  Serial.begin(9600);
  pinMode(relay, OUTPUT);
}
  
```

```
void loop() {
    LDR_val = analogRead(LDR);
    Serial.println(LDR_val);

    if(LDR_val >= 500){
        digitalWrite(relay,LOW);
    }

    else{
        digitalWrite(relay,HIGH);
    }

    delay(1000);
}
```

5. Coloque aquí evidencias que considere importantes durante el desarrollo de la actividad.

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

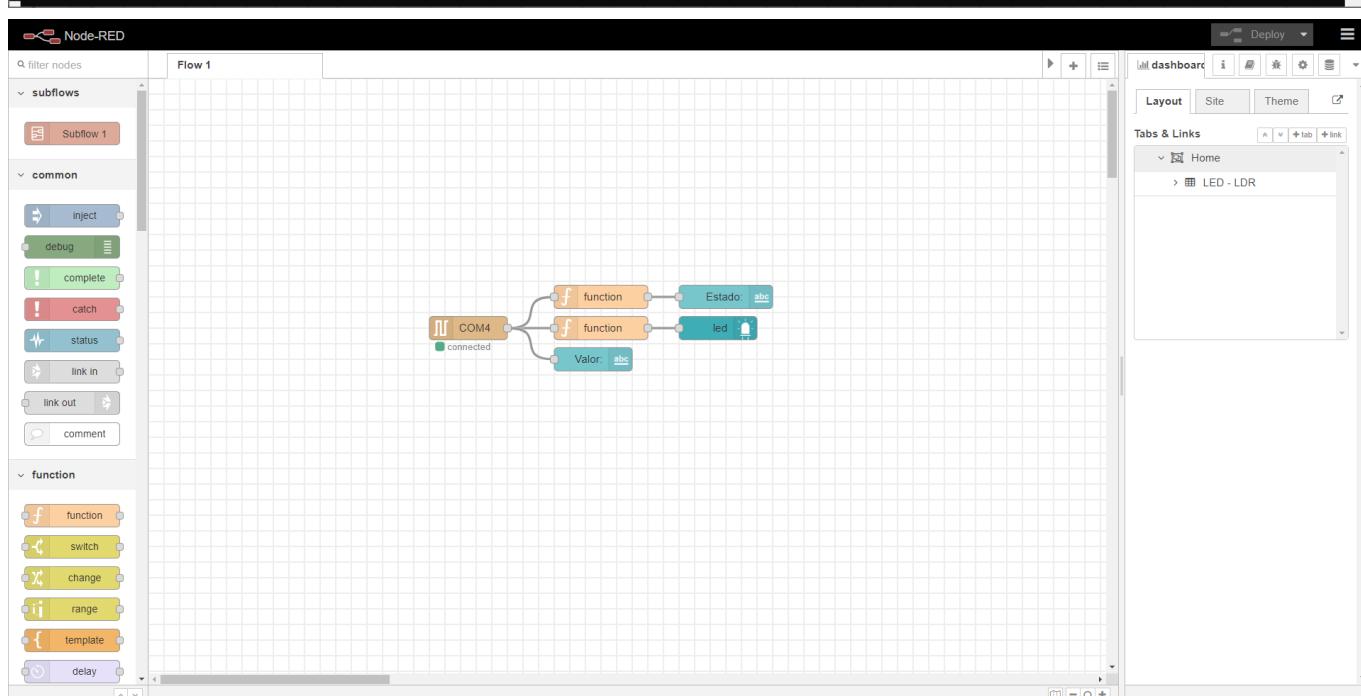
```
node-red
13 Dec 15:19:48 - [info] Node-RED version: v1.2.6
13 Dec 15:19:48 - [info] Node.js version: v14.15.1
13 Dec 15:19:48 - [info] Windows_NT 10.0.18363 x64 LE
13 Dec 15:19:48 - [info] Loading palette nodes
13 Dec 15:19:49 - [info] Dashboard version 2.25.0 started at /ui
13 Dec 15:19:49 - [info] Settings file : C:\Users\PIMPO\.node-red\settings.js
13 Dec 15:19:49 - [info] Context store : 'default' [module=memory]
13 Dec 15:19:49 - [info] User directory : \Users\PIMPO\.node-red
13 Dec 15:19:49 - [warn] Projects disabled : editorTheme.projects.enabled=false
13 Dec 15:19:49 - [info] Flows file : \Users\PIMPO\.node-red\flows_DESKTOP-KQ957L7.json
13 Dec 15:19:49 - [info] Server now running at http://127.0.0.1:1880/
13 Dec 15:19:49 - [warn]

-----
Your 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.

13 Dec 15:19:49 - [info] Starting flows
13 Dec 15:19:49 - [info] Started flows
13 Dec 15:19:49 - [info] serial port COM4 opened at 9600 baud 8N1
```



Setup      Function      Close

```

1 if(msg.payload == "Apagado" || msg.payload < 500)
2 {
3     msg.payload = false;
4 }
5
6 if(msg.payload == "Encendido" || msg.payload >= 500)
7 {
8     msg.payload = true;
9 }
10
11 return msg;

```

Setup      **Function**      Close

The screenshot shows a software interface with a tab bar at the top. The 'Function' tab is selected. Below the tabs is a code editor window containing the following JavaScript-like pseudocode:

```
1 if(msg.payload >= 500)
2 {
3     msg.payload = "Encendido";
4 }
5
6 if(msg.payload < 500)
7 {
8     msg.payload = "Apagado";
9 }
10
11 return msg;
```

To the right of the code editor is a preview area with a blue header labeled 'Home'. Inside the preview area, there is a section titled 'LED - LDR' with a green circular icon. Below the icon, the text 'Estado:' is followed by the word 'Encendido' in bold. Underneath that, the text 'Valor:' is followed by the number '2074'.

Home

LED - LDR



Estado:

**Encendido**

Valor:

**2074**

Home

LED - LDR



Estado:

**Apagado**

Valor:

**80**

- Evidencia de reuniones:**

meet.google.com/xiu-hijn-for?pli=1&authuser=0

Jose Venegas está presentando

int LDR = 34;  
int LDR\_val = 0;  
int relay = 16;

```
void setup() {  
    Serial.begin(9600);  
    pinMode(relay, OUTPUT);  
}  
  
void loop() {  
    LDR_val = analogRead(LDR);  
    Serial.println(LDR_val);  
  
    if(LDR_val >= 500){  
        digitalWrite(relay, LOW);  
    }  
  
    else{  
        digitalWrite(relay, HIGH);  
    }  
  
    delay(1000);  
}
```

15:36 Tú

CESAR MANUEL RAMIREZ CERVANTES

Jose Venegas

DULCE JASMIN VILLALOBOS PEREZ

Type here to search

meet.google.com/xiu-hijn-for?pli=1&authuser=0

Jose Venegas está presentando

int LDR = 34;  
int LDR\_val = 0;  
int relay = 16;

```
void setup() {  
    Serial.begin(9600);  
    pinMode(relay, OUTPUT);  
}  
  
void loop() {  
    LDR_val = analogRead(LDR);  
    Serial.println(LDR_val);  
  
    if(LDR_val >= 500){  
        digitalWrite(relay, LOW);  
    }  
  
    else{  
        digitalWrite(relay, HIGH);  
    }  
  
    delay(1000);  
}
```

15:37 Tú

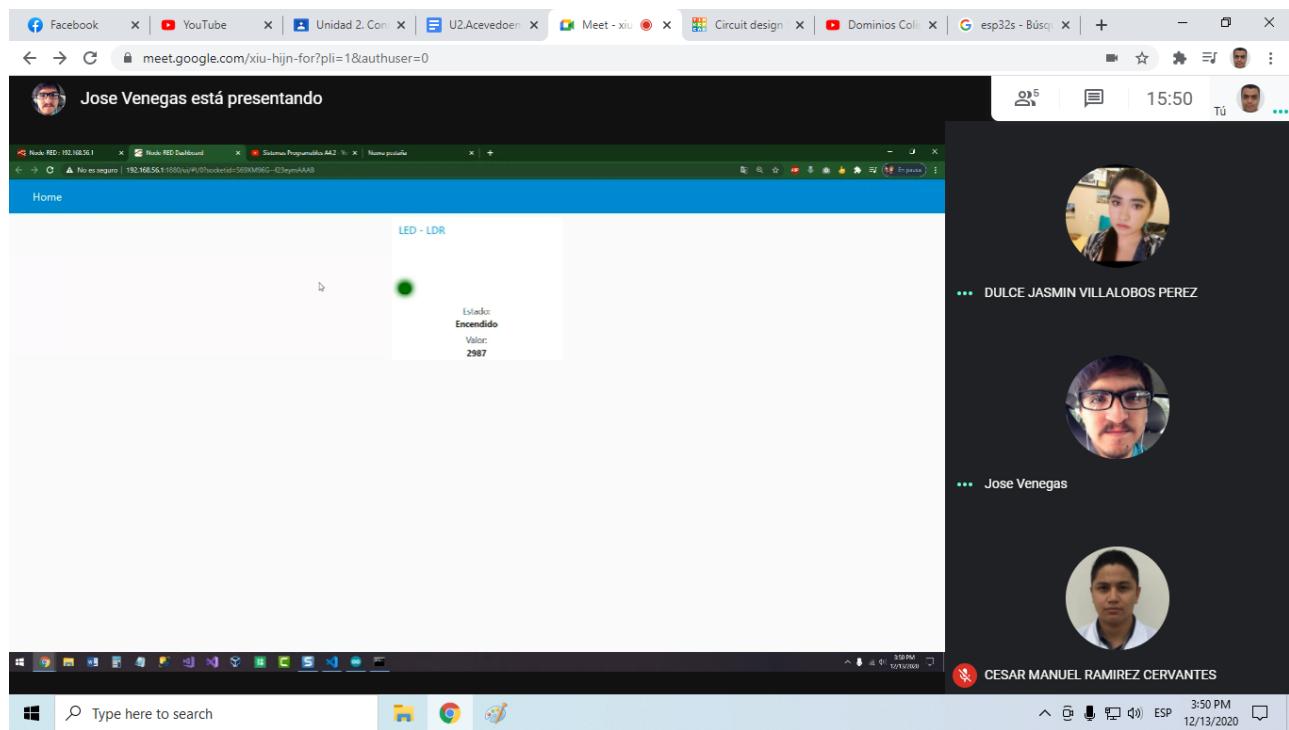
CESAR MANUEL RAMIREZ CERVANTES

Jose Venegas

DULCE JASMIN VILLALOBOS PEREZ

Type here to search

The screenshot shows a Microsoft Teams meeting interface. On the right, three participant profiles are visible: CESAR MANUEL RAMIREZ CERVANTES (top), Jose Venegas (middle), and DULCE JASMIN VILLALOBOS PEREZ (bottom). The main window displays a Node-RED flow diagram with various nodes like 'inject', 'function', 'COMA', 'function', 'test', and 'DOM'. To the right of the flow is a screenshot of a web application with a blue header 'LED - LDR' and a status box showing 'Estado Encendido' with a value of '2074'. Below this is another screenshot of a web application showing an LED status with a red light indicator.



## 6. Conclusiones.

- Acevedo Ensiso Pedro Gabriel:**

Dentro de la realización de esta práctica me di cuenta de que existen muchas aplicaciones reales a las que le podemos implementar un circuito con este funcionamiento, desde sistemas de detección hasta sistemas de iluminación las aplicaciones son muy extensas, con los conocimientos adquiridos también pude comprender mejor las ventajas que existen con el diseño de un servidor web con el cual podamos visualizar o monitorear procesos en tiempo real desde la comodidad de un dispositivo inteligente. Tales procesos monitoreados van desde los procesos de manufactura hasta los procesos de vigilancia, existen muchas aplicaciones para sistemas así y este es solo la punta del "iceberg" en cuanto a este tema concierne, podemos aplicar también este tipo de aplicaciones al área doméstica con sistemas de iluminación automatizados con los cuales podemos controlar la iluminación de una manera automática y además podemos monitorear su funcionamiento.

- Ramirez Cervantes Cesar Manuel:**

Para esta práctica se utilizó un sensor Ldr junto con un relevador para elaborar un circuito sobre un ESP232 que es capaz de encender un led cuando el Ldr se presenta ausencia de luz y se apaga cuando el Ldr recibe más luz, este comportamiento también es gracias al relevador el cual al momento de que el Ldr recibe menos luz presenta más resistencia por lo tanto el voltaje se desvía al relevador y de esta forma se acciona para darle el voltaje necesario al led para encenderse. Lo interesante de esta práctica es que se utilizó el NodeMCU como un standalone web server para una visualización del estado actual del led, es decir, cuando el led se enciende por ausencia de luz se mostrará en la interfaz web el mensaje del estado del led como encendido, en caso contrario se mostrará el mensaje en apagado.

- Venegas Medina Jose Alfredo:**

En la realización de esta práctica se realizó un circuito en el cual se pudiera controlar el estado de un led por medio de un LDR, un relay y una página web creada en Node-RED. Primeramente, se realizó un circuito básico el cual pudiera leer los valores del LDR, una vez hecho esto, se procedió a realizar un programa en entorno Arduino para su implementación. Al tener el circuito se investigó como realizar

una página en Node-RED, para esto, se tendría que descargar y usaría la dirección IP de la computadora para conectarse al Node-RED y así poder realizar una página utilizando los bloques de programación. Luego se investigó una forma de desarrollar la página utilizando las distintas opciones que tiene Node-RED, una vez investigadas, se implementó un LED que se encendería verde si asignaba encendido y rojo si asignaba apagado, igualmente se mostraría un mensaje del estado del LED y el mostraría el valor del LDR. Una vez terminado esto, se implementó un relay para el control de un LED físico, inicialmente se pretendía utilizar una bombilla, pero se decidió cambiar por un LED. De modo que, al estar iluminado el LDR, el LED por medio del relay se apagaría, y se mostraría en la página un LED en rojo y su estado en "Apagado". Por otro lado, si el LDR no estaba iluminado el LED encendería y la página mostraría el mensaje "Encendido" y un LED en verde. La aplicación de esta práctica sería para tener un control total de los circuitos que existen en nuestra casa y ver su estado.

- **Villalobos Perez Dulce Jasmin:**

En esta práctica se utilizó un LDR para variar el valor de la resistencia eléctrica dependiendo de la cantidad de luz a la que se exponga, también un relé que se activara dependiendo del valor del LDR, cuando el valor del LDR es mayor a 500 entonces el Led enciende, ya que como se ve el circuito ya hecho, cuando el LDR se exponga a una luz este se apague, lo que quiere decir que solo se encenderá cuando este a una exposición muy baja de luz, en esta práctica se hizo necesario trabajar con una interfaz gráfica de página web, la cual se hizo desde node-Red, en el cual se vinculaba la esp32 y esta nos permitía ver de forma visual el comportamiento de LDR ya que imprime tanto el valor de la resistencia y el estado del led, si está apagado o encendido, y pues este servidor web nos permite acceder a él desde cualquier dispositivo, que se encuentre conectado a la red local, pues esto es muy interesante, ya que es una forma de vincular los aparatos electrónicos podrían ser unas luces de jardín funcionen con este sensor y puedan apagarse solas cuando sea necesario y pues revisar que están apagadas o encendidas desde cualquier dispositivo.

 **Rubrica**

Criterios	Descripción	Puntaje
Instrucciones	Se cumple con cada uno de los puntos indicados dentro del apartado Instrucciones?	10
Desarrollo	Se respondió a cada uno de los puntos solicitados dentro del desarrollo de la actividad?	60
Demostración	El alumno se presenta durante la explicación de la funcionalidad de la actividad?	20
Conclusiones	Se incluye una opinión personal de la actividad por cada uno de los integrantes del equipo?	10

**EQUIPO VERDE**
**Acevedo Ensiso Pedro Gabriel:**
 [Ir a mi Github](#)
**Ramirez Cervantes Cesar Manuel:**

 Enlace a mi repositorio 

**Venegas Medina Jose Alfredo:**

 Mi Github

**Villalobos Perez Dulce Jasmin:**

  ENLACE - MI GITHUB