



## 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

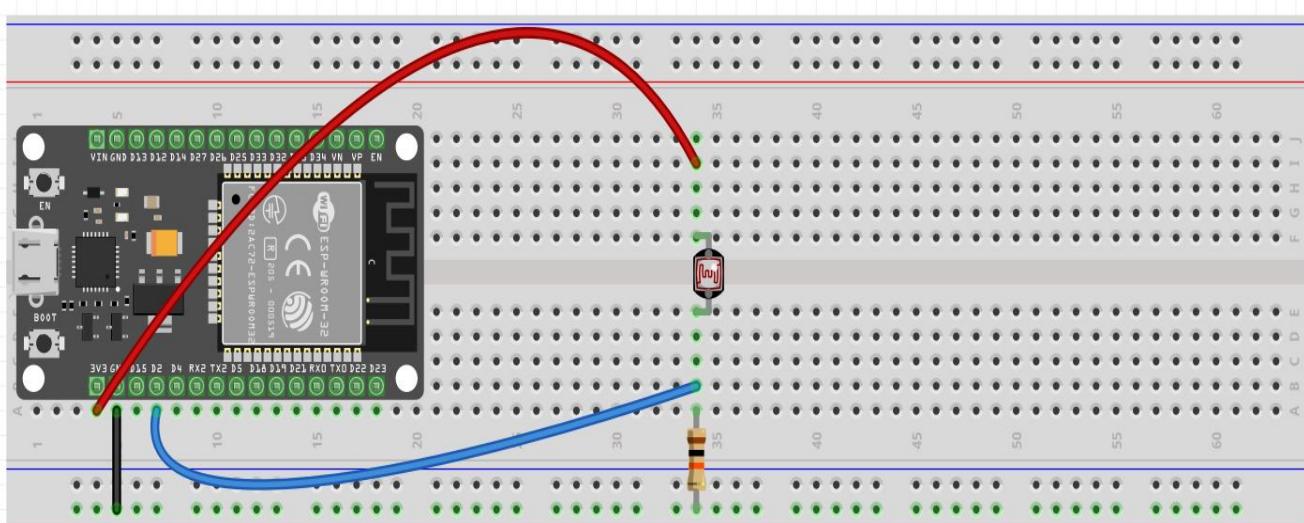
1. Utilizar el siguiente listado de materiales para la elaboración de la actividad

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

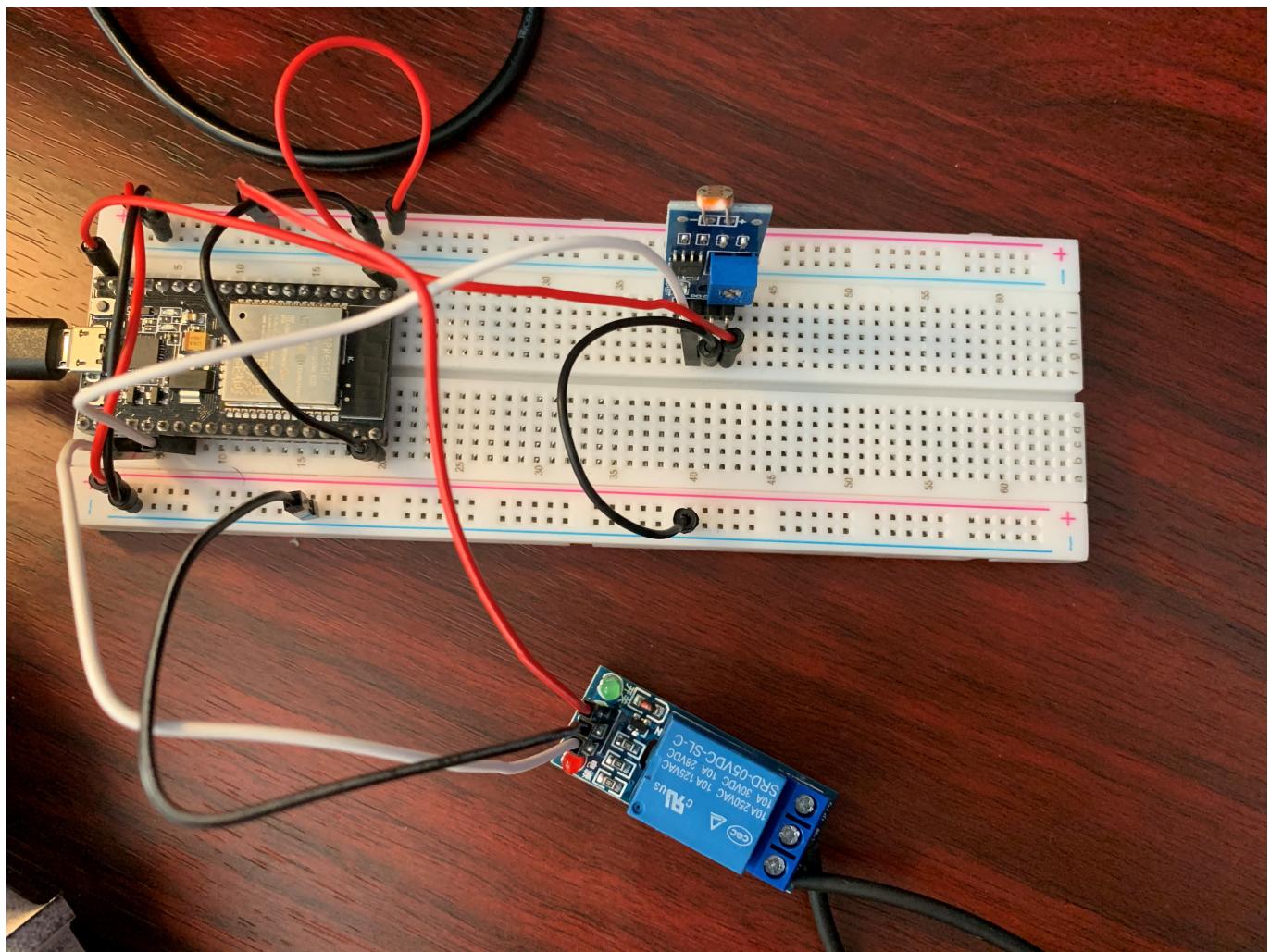
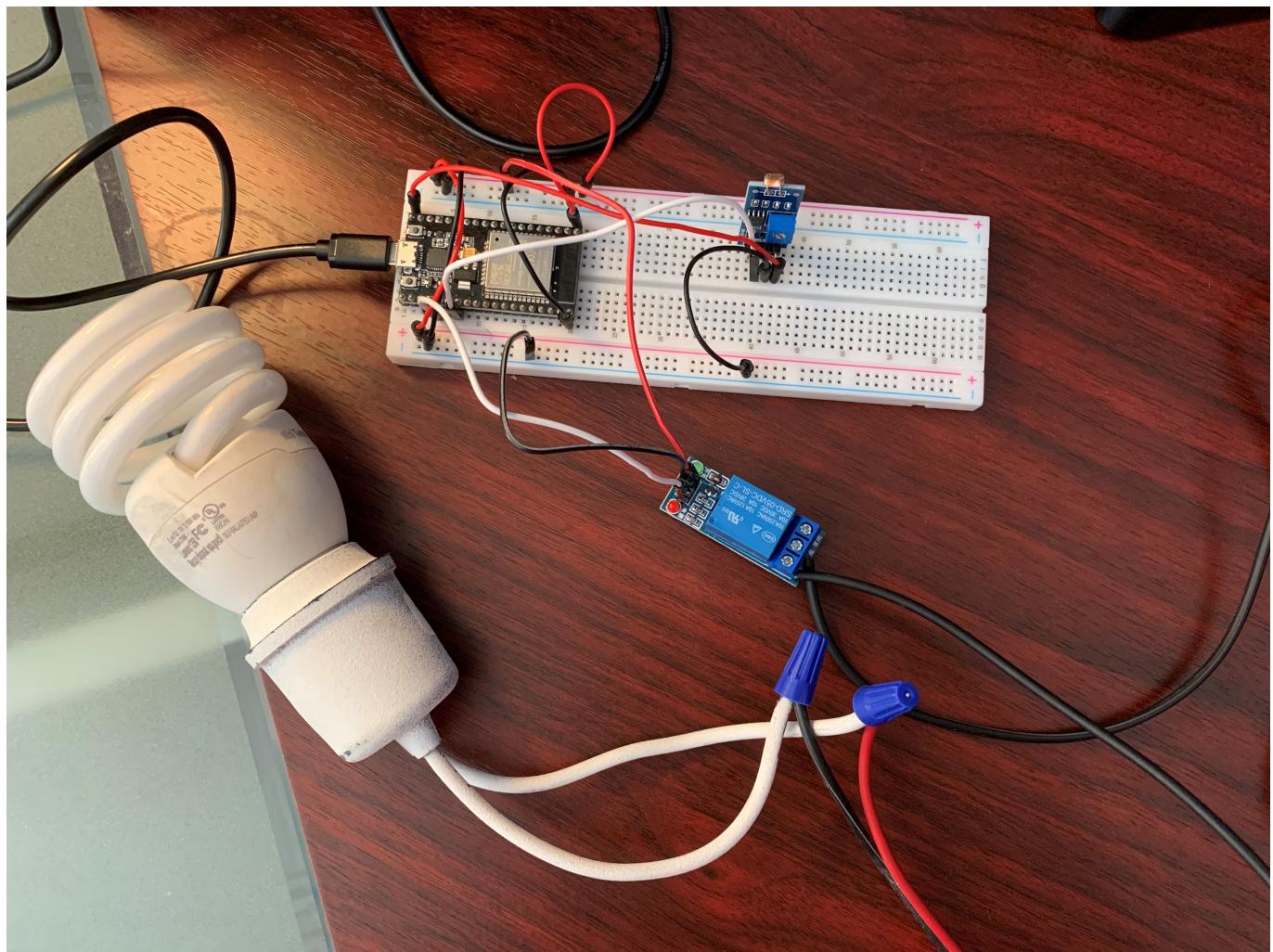
2. Basado en la imagen que se muestran en las **Figura 1**, ensamble un circuito electrónico y agregue un Led y un rele a una de las terminales del dispositivo, de tal manera que se pueda obtener un sistema capaz de cumplir con las instrucciones siguiente:

- 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

```
#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = "sfwvkHTFaToHkwtucNyb6oaOTv0MH-Cc"; // Es el token por el que la aplicacion se conectara
char ssid[] = "100 Euros";
char pass[] = "Enrollados En Tu boca";

WidgetLCD lcd(V1); // Se inicia la pantalla lcd de blynk

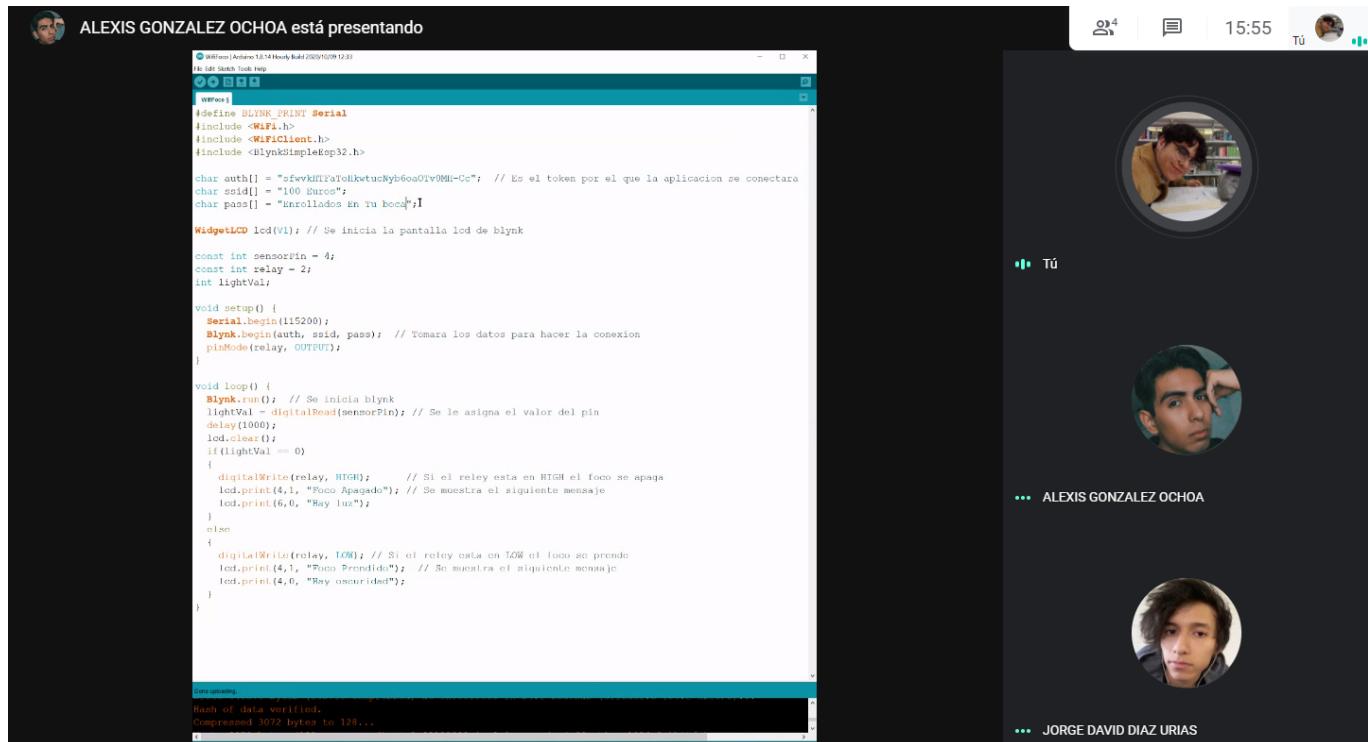
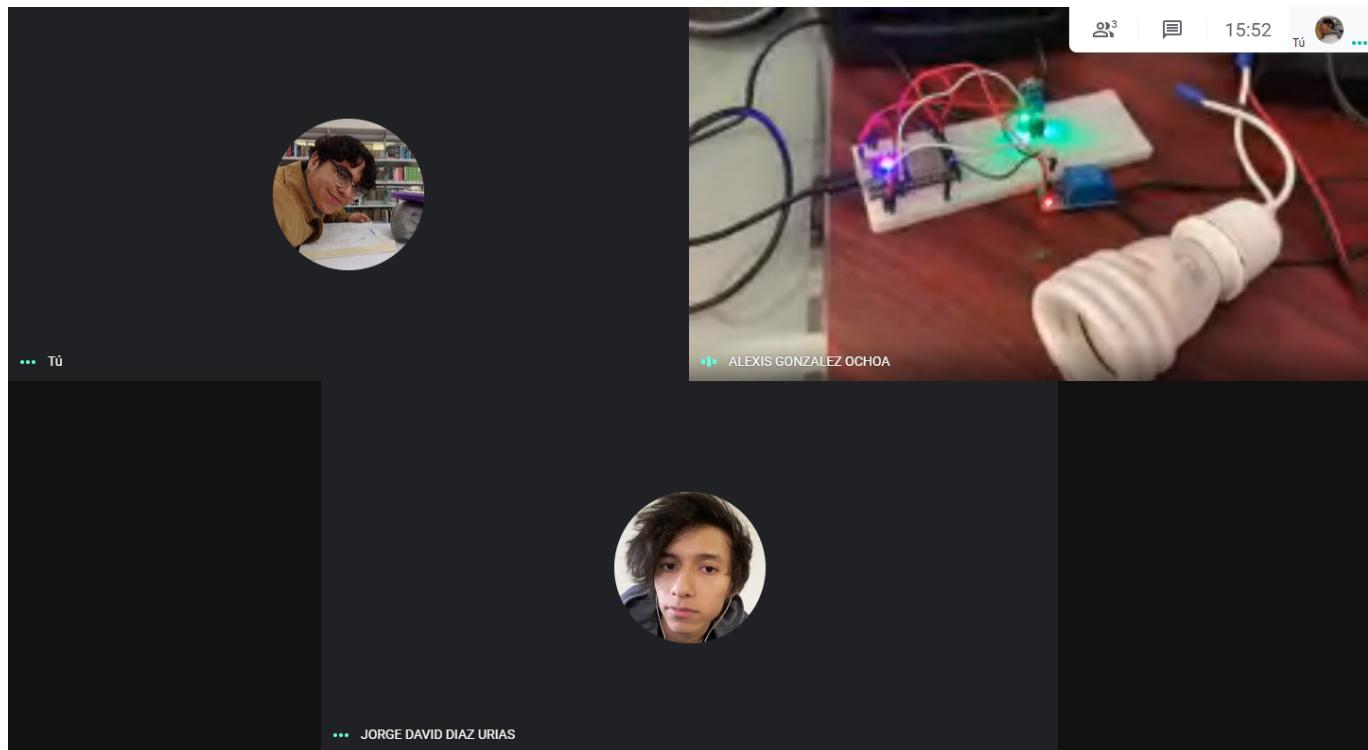
const int sensorPin = 4;
const int relay = 2;
int lightVal;

void setup() {
    Serial.begin(115200);
    Blynk.begin(auth, ssid, pass); // Tomara los datos para hacer la conexion
    pinMode(relay, OUTPUT);
}

void loop() {
    Blynk.run(); // Se inicia blynk
    lightVal = digitalRead(sensorPin); // Se le asigna el valor del pin
    delay(1000);
    lcd.clear();
    if(lightVal == 0)
    {
        digitalWrite(relay, HIGH); // Si el reley esta en HIGH el foco se apaga
        lcd.print(4,1, "Foco Apagado"); // Se muestra el siguiente mensaje
        lcd.print(6,0, "Hay luz");
    }
    else
    {
        digitalWrite(relay, LOW); // Si el reley esta en LOW el foco se prende
        lcd.print(4,1, "Foco Prendido"); // Se muestra el siguiente mensaje
        lcd.print(4,0, "Hay oscuridad");
    }
}
```

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

#### Video demostrativo



## 6. Conclusiones

Alexis Gonzalez

Con esta practica pude experimentar el uso de un sensor de luz y de un revelador. Teniendo problemas con la lectura del sensor LDR al momento de enviar los datos, ya que el valor dado se elevaba hasta el maximo y tuvimos que cambiar la lectura analogica a digital dando como resultados "0" para apagado y "1" para prendido. Tambien descubrimos la utilizacion de una nueva app llamada blynk la cual se puede utilizar para proyectos IOT y que funciona a la perfeccion con el ESP32.

Jorge Diaz Con la realización de esta práctica pudimos ver la funcionalidad del wifi con el esp32 junto con sensor de lectura LDR y el descubrimiento de una aplicación muy útil llamada blynk, la cual nos permite controlar el hardware de manera remota. Además de que proporciona diferentes widgets que pueden ser

usados para distintas aplicaciones como fue el uso del widget LCD para está práctica y es que además esto nos permite añadir más funciones demasiado interesantes.

Julio Jimenez En conclusión, la práctica nuevamente nos enseña una de las habilidades del ESP32, en este caso se utilizó mediante wifi, pero además del conocimiento adquirido sobre cómo utilizar un poco más la conexión wifi para poder encender y apagar la bombilla, también aprendimos cuento apoyo ha tenido el ESP32 en si, como la aplicación blynk, nos permitió realizar la practica de manera sencilla, ésta nos permitió controlar de manera remota la aplicación.

---

### 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

 [Ir a GitHub de Julio Jimenez](#)

 [Ir a GitHub de Jorge Diaz](#)

 [Ir a GitHub de Alexis Gonzalez](#)