



## A.3.2 Actividad de aprendizaje

Circuito sensor de tacto a través de un NodeMCU ESP32

### Instrucciones

- Basado en la figura 1, ensamblar un sistema, capaz de responder al tacto, a través de un circuito electrónico, utilizando un NodeMCU **ESP32**, un **Sensor de tacto capacitivo**.
- 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, y debe ser nombrado con la nomenclatura **A3.2\_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 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
  - C3.1_TituloActividad.md
  - C3.2_TituloActividad.md
  - C3.3_TituloActividad.md
  - C3.4_TituloActividad.md
  - C3.5_TituloActividad.md
  - C3.6_TituloActividad.md
  - C3.7_TituloActividad.md
  - C3.8_TituloActividad.md
- img
- docs
  - A3.1_TituloActividad.md
  - A3.2_TituloActividad.md
  - A3.3_TituloActividad.md
```

### Fuentes de apoyo para desarrollar la actividad

- [Sensor de tacto capacitivo](#)

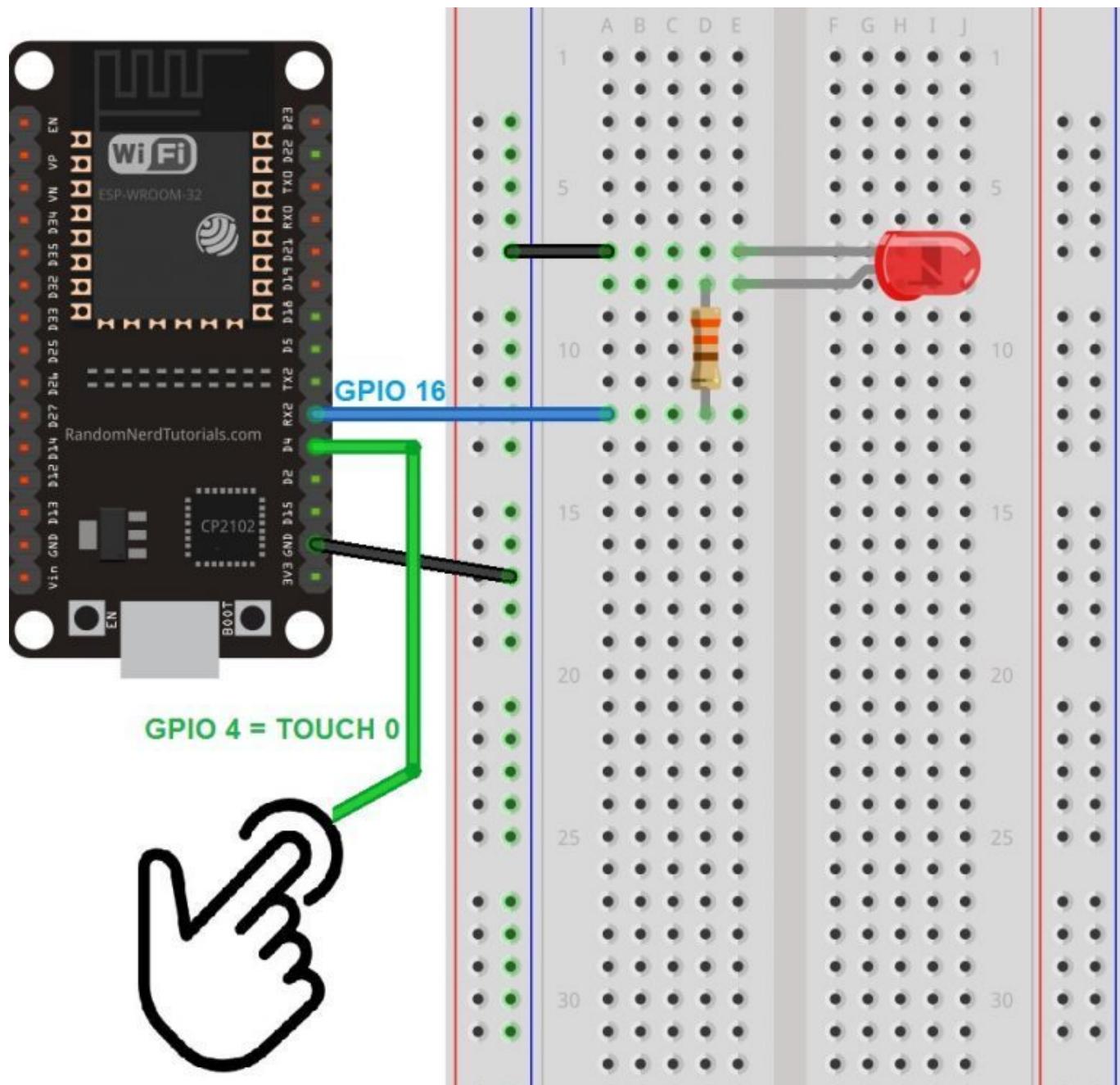
- Salida analoga PWM
- 

## Desarrollo

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

| Cantidad | Descripción             |
|----------|-------------------------|
| 1        | Diodo led Rojo          |
| 1        | Resistencia 330 ohms    |
| 1        | Fuente de voltaje de 5V |
| 1        | NodeMCU ESP32           |
| 1        | BreadBoard              |
| 1        | Jumpers M/M             |
| 1        | Hoja de aluminio        |

**Figura 1 Circuito ESP32 IC L293 Motor DC**



1. Una vez ensamblado el circuito anterior, realice un programa que permita al ensamble cumplir con las siguientes condiciones:

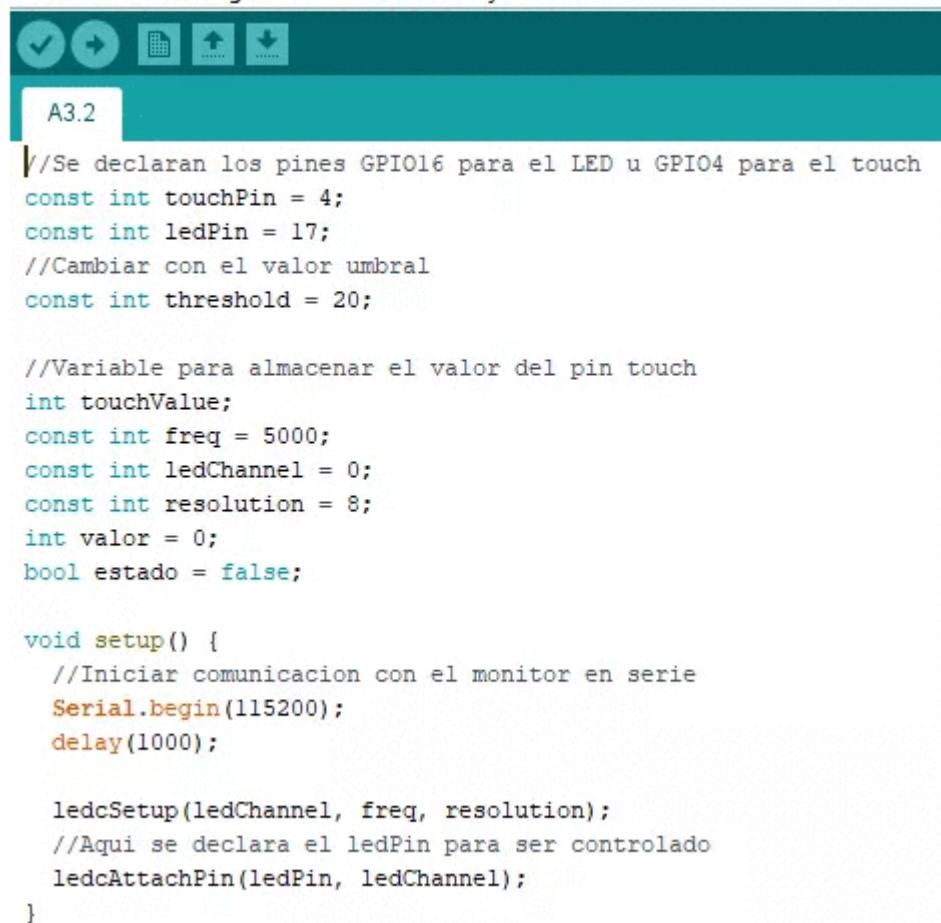
- El sistema deberá ser capaz de encender y apagar **un led** al presionar el sensor de tacto.
- El sistema deberá tener la característica que si el Led esta encendido, al tocar el sensor de tacto, este iniciara un secuencia de intermitencia de 3 segundos (es decir se apaga 1 segundo, se enciende un segundo y se apaga finalmente).
- El sistema deberá contar con la característica que si el Led esta apagado, al tocar el sensor de tacto, este se encenderá poco a poco hasta llegar a su nivel máximo de iluminación.

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

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

 A3.2 Arduino 1.8.12

Archivo Editar Programa Herramientas Ayuda



```
//Se declaran los pines GPIO16 para el LED u GPIO4 para el touch
const int touchPin = 4;
const int ledPin = 17;
//Cambiar con el valor umbral
const int threshold = 20;

//Variable para almacenar el valor del pin touch
int touchValue;
const int freq = 5000;
const int ledChannel = 0;
const int resolution = 8;
int valor = 0;
bool estado = false;

void setup() {
    //Iniciar comunicacion con el monitor en serie
    Serial.begin(115200);
    delay(1000);

    ledcSetup(ledChannel, freq, resolution);
    //Aqui se declara el ledPin para ser controlado
    ledcAttachPin(ledPin, ledChannel);
}
```

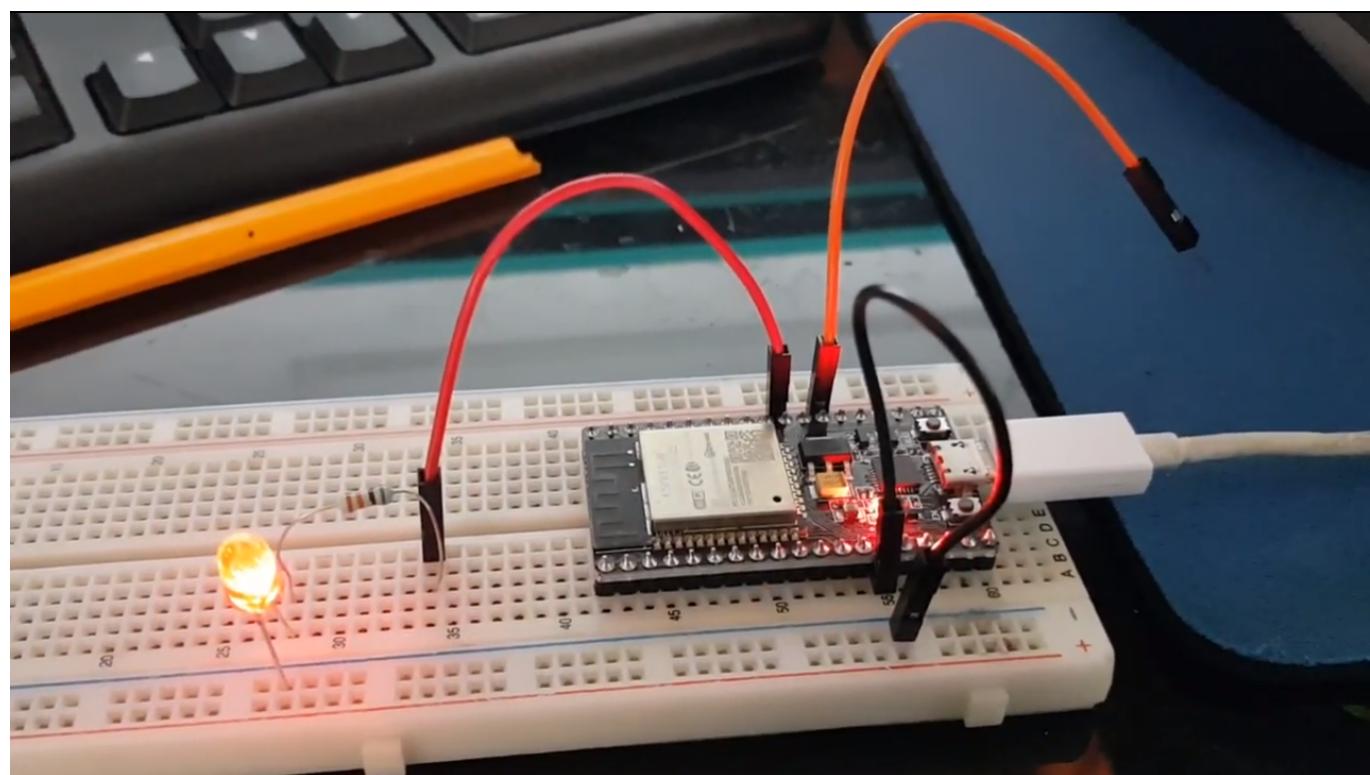
```
void loop() {
    //Lee el estado del pin touch
    touchValue = 21;
    touchValue = touchRead(touchPin);
    //Se imprime el valor obtenido del pin touch
    Serial.print(touchValue);

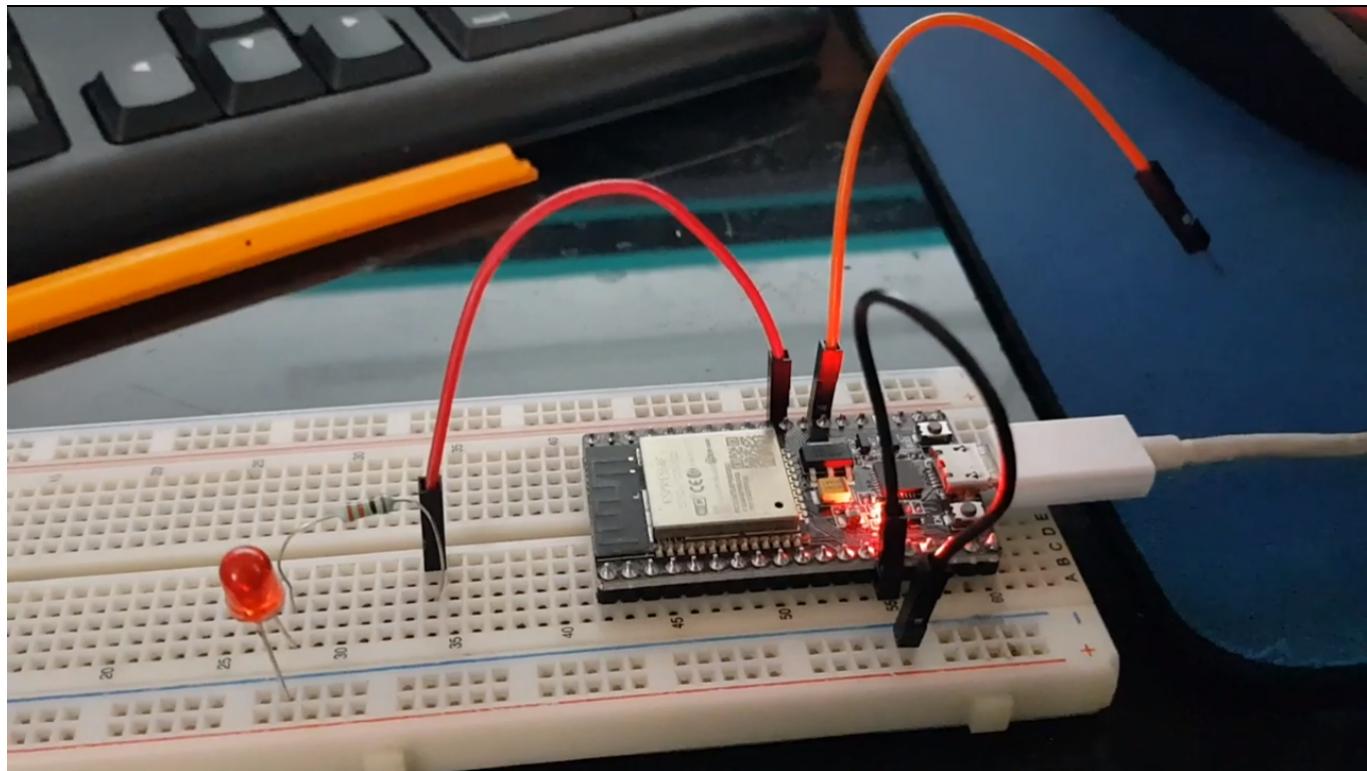
    //Si esta apagado, se prende el LED poco a poco
    if((touchValue < threshold) && (estado == false)){
        for(int dutyCycle = 0; dutyCycle <= 255; dutyCycle++){
            ledcWrite(ledChannel, dutyCycle);
            delay(25);
        }

        Serial.println(" - LED on");
        estado = true;
        touchValue = 21;
        delay(1000);
    }

    //Si esta prendido, el led apagara, prendera y apagara
    if((touchValue < 20) && (estado == true)){
        ledcWrite(ledChannel, 0);
        delay(1000);
        ledcWrite(ledChannel, 255);
        delay(1000);
        ledcWrite(ledChannel, 0);
        delay(1000);

        Serial.println(" - LED off");
        estado = false;
        touchValue = 21;
        delay(1000);
    }
}
```





### 3. Evidencias reuniones.

PEDRO GABRIEL ACEVEDO ENSISO está presentando

**A3.2.md** - Repository sistemas\_programables - Visual Studio Code

REPOSITORIO\_SISTEMAS\_P...

```

> A3.2.md
blog > A3.2.md > # dropby:A3.2 Actividad de aprendizaje > ## Desarrollo
| 1 | [Diodo led Rojo](https://www.sternen.com.mx/led-ultrabrilante-de-5-mm-color-rojo.html)
| 1 | [Resistencia 330 ohms](https://www.sternen.com.mx/resistencia-de-carbon-de-1-2-watt-al-5-de-tolerancia-de-330-ohms.html)
| 1 | [Fuente de voltaje de 5V](https://www.amazon.com.mx/AC-DC-fuente-alimentacionX3X3n-Black/dp/B078RXZMC/ref-asc_df_B078RXZMC/?tag-gledskshopmx-20&linkCode=df0&hvaddid=450926494212&hvpos=&hvnetw=g&hvrand=1295240476675663068&hvpcne=&hvptwo=&hvqmt=&hvdev=c&hvdevmdl=&hvlocint=&hvlocphy=9073855&hvargld=pla-43641810450&psc=1)
| 1 | [NodeMCU ESP32](https://www.amazon.com.mx/ESP-32-ESP-32S-ESP-WROOM-32-ESP32-5-desarrollo/dp/B07TBFCT5Z/ref=sr_1_2?mk=es_MX=%C3%88%C3%88%C3%88%C3%88%C3%88%C3%88&dchild=1&refinements=)

```

PREVIEW A3.2.md

Hoja de aluminio

Figura 1 Circuito ESP32 IC L293 Motor DC

1. Una vez ensamblado el circuito anterior, realice un programa que

CESAR MANUEL RAMIREZ CERVANTES

**PEDRO GABRIEL ACEVEDO ENSISO** está presentando

A3.2.md - Repositorio\_sistemas\_programables - Visual Studio Code

REPOSITORIO\_SISTEMAS\_P...

blog > A3.2.md > # trophy: A3.2 Actividad de aprendizaje > ## Desarrollo

PREVIEW A3.2.md

3. Evidencias reuniones.

4. Conclusiones:

- \*\*Acevedo Ensiso Pedro Gabriel:\*\*  
Dentro de esta práctica aprendimos cómo manejar mejor el sensor de tacto para realizar una función específica, en este caso nos referimos al encendido de un LED con diferentes especificaciones. Las especificaciones en este caso son que cuando toquemos el pin el LED se encienda lentamente hasta llegar a su máxima luminosidad y cuando lo toquemos de nuevo este se apague, prenda y apague en ese orden. La importancia de esta práctica es que nosotros podemos aplicar esta salida para algo más importante como un actuador, sistema de seguridad, etc. sistemas que quizás queramos que actúen de una manera
- \*\*Ramirez Cervantes Cesar Manuel:\*\*  
aaa
- \*\*Venegas Medina Jose Alfredo:\*\*  
aaa
- \*\*Villalobos Perez Dulce Jasmin:\*\*  
Para la realización de esta práctica tenímos que hacer el circuito dado en la actividad, aunque si fue algo diferente la forma de organizarnos ya qué nuestro compañero qué está a cargo de los materiales, comenzó armar el circuito mientras observábamos como se iban

3. Evidencias reuniones.

4. Conclusiones:

- **Acevedo Ensiso Pedro Gabriel:**  
Dentro de esta práctica aprendimos cómo manejar mejor el sensor de tacto para realizar una función específica, en este caso nos referimos al encendido de un LED con diferentes especificaciones. Las especificaciones en este caso son que cuando toquemos el pin el LED se encienda lentamente hasta llegar a su máxima luminosidad y cuando lo toquemos de nuevo este se apague, prenda y apague en ese orden. La importancia de esta práctica es que nosotros podemos aplicar esta salida para algo más importante como un actuador, sistema de seguridad, etc. sistemas que quizás queramos que actúen de una manera
- **Ramirez Cervantes Cesar Manuel:**  
aaa
- **Venegas Medina Jose Alfredo:**  
aaa

CEAR MANUEL RAMIREZ CERVANTES

**CESAR MANUEL RAMIREZ CERVANTES** está presentando

YouTube MX

Buscar

https://www.youtube.com/watch?v=Vc1HeNsraIA&feature=youtu.be&fbclid=IwAR1Zki8YAoYgICUgnvJyjWQAZ\_aBt0M5dP\_9IyEIN959Q7kjwTDkEc&ab\_channel=JoseVenegas

TÚ

CESAR MANUEL RAMIREZ CERVANTES

DULCE JASMIN VILLALOBOS PEREZ

CESAR MANUEL RAMIREZ CERVANTES

The screenshot displays a Google Meet interface with four participants. On the right, there are circular profile pictures for each participant. The main window shows two code editors side-by-side.

**Top Editor (CESAR MANUEL RAMIREZ CERVANTES):**

```

void loop() {
    //Lee el estado del pin touch
    touchValue = 21;
    touchValue = touchRead(touchPin);
    //Se imprime el valor obtenido del pin touch
    Serial.print(touchValue);

    //Si esta apagado, se prende el LED poco a poco
    if((touchValue < threshold) && (estado == false)){
        for(int dutyCycle = 0; dutyCycle < 255; dutyCycle++){
            ledWrite(ledChannel, dutyCycle);
            delay(25);
        }

        Serial.println(" - LED on");
        estado = true;
        touchValue = 21;
        delay(1000);
    }

    //Si esta prendido, el led apagara, prendera y apagara
    if((touchValue < 20) && (estado == true)){
        ledWrite(ledChannel, 0);
        delay(1000);
        ledWrite(ledChannel, 255);
        delay(1000);
        ledWrite(ledChannel, 0);
        delay(1000);

        Serial.println(" - LED off");
        estado = false;
        touchValue = 21;
        delay(1000);
    }
}

```

**Bottom Editor (Jose Venegas):**

```

void setup() {
    //Iniciar comunicacion con el monitor en serie
    Serial.begin(115200);
    delay(1000);

    ledWrite(ledChannel, freq_resolution);
    //No se declara el ledPin para ser controlado
    ledWrite(ledPin, ledChannel);
}

void loop() {
    //Lee el estado del pin touch
    touchValue = 21;
    touchValue = touchRead(touchPin);
    //Se imprime el valor obtenido del pin touch
    Serial.println(touchValue);

    //Si esta apagado, se prende el LED poco a poco
    if((touchValue < threshold) && (estado == false)){
        for(int dutyCycle = 0; dutyCycle < 255; dutyCycle++){
            ledWrite(ledChannel, dutyCycle);
            delay(25);
        }

        Serial.println(" - LED on");
        estado = true;
        touchValue = 21;
        delay(1000);
    }

    //Si este prendido, el led apaga, prendera y apaga
    if((touchValue < 20) && (estado == true)){
        ledWrite(ledChannel, 0);
        delay(1000);
        ledWrite(ledChannel, 255);
        delay(1000);
    }
}

Serial.println(" - Hard resetting via RTS pin...");

```

#### 4. Conclusiones:

- Acevedo Ensiso Pedro Gabriel:**

Dentro de esta práctica aprendimos cómo manejar mejor el sensor de tacto para realizar una función específica, en este caso nos referimos al encendido de un LED con diferentes especificaciones. Las especificaciones en este caso son que cuando toquemos el pin el LED se encienda lentamente hasta llegar a su máxima luminosidad y cuando lo toquemos de nuevo este se apague, prenda y apague en ese orden. La importancia de esta práctica es que nosotros podemos aplicar esta salida para algo más importante como un actuador, sistema de seguridad, etc. sistemas que quizás queramos que actúen de una manera específica y gracias a esta práctica aprendimos de mejor manera las distintas maneras en que podemos hacerlo, también dentro del desarrollo de la práctica aprendí que al manejar voltajes es muy importante siempre asegurar las conexiones antes de probarlas ya que un error puede averiar nuestro esp32 dejandolo inoperable.

- Ramirez Cervantes Cesar Manuel:**

En esta práctica se buscó realizar un circuito capaz de captar mediante el pin de un cable el tacto como si se tratase de un componente touch, con esto se debería enviar una señal al ESP32 para que este con ayuda del código desarrollado encendiera un led poco a poco aumentando su intensidad con un retraso de 25 milisegundos. Al llegar al punto de la máxima intensidad del led este debe permanecer encendido hasta que el ESP32 reciba nuevamente una señal del cable por medio del tacto, lo que sucede después es que el led se apagara, no sin antes como muestra de que el led se apagara el led parpadeará una vez para posteriormente apagarse y permanecer así hasta que se vuelva a tocar el pin del cable.

- Venegas Medina Jose Alfredo:**

En esta práctica con ayuda de NodeMCU ESP32 y su salida analógica PWM además de su sensor de tacto capacitivo, se pudo controlar el estado de un led, cuando este estuviera apagado, y se tocara el pin touch, la programación le indicaría al LED que se prendiera poco a poco, esto gracias a su salida analógica, ya que iría de un estado 0 a 255 con un retraso de 25 milisegundos por cada estado, y se mantendría prendido. Por otro lado, si el LED estuviera prendido y el pin touch se tocara, entonces el LED iniciaría una secuencia de apagado, prendido y apagado finalmente. La práctica se realizó con facilidad, ya que se adaptó de las actividades C3.7 y C3.8, no obstante, se desconocía cómo mantener el LED en un estado después de que el pin touch sea tocado. Así que se buscaron diferentes maneras para poder realizar el código de forma correcta, siguiendo las especificaciones solicitadas. La solución fue agregar una variable estado, para condicionar cada estado del led y no se ciclara.

- Villalobos Perez Dulce Jasmin:**

Para la realización de esta práctica teníamos qué hacer el circuito dado en la actividad, aunque si fue algo diferente la forma de organizarnos ya qué nuestro compañero qué está a cargo de los materiales, comenzó armar el circuito mientras observábamos como se iban conectando tanto teórico como en físico, en cuanto al código qué se realizó con base en la actividad y el enlace qué proporcionó, en si la actividad tenía qué hacer qué al tocar el touch el led se encendiera y al tocarlo de nuevo el cable qué enlaza a pin del 4 touch se apaga enciende y permanece apagado, esto con base en el código nos dice que usamos el pin 17 GPIO para Pin Led, la parte qué se encarga de qué el led se encienda poco a poco está dentro de la función Loop con el bloque de instrucción donde usamos un if y un ciclo dentro, donde usamos los valores de 0 a 255, y el estado es controlado con un booleano qué nos indica si es apagado en false y encendido en true , una vez terminado el ciclo y el Led se apague de nuevo este estado regresa a ser false, para poder evaluar la condición de nuevo al querer encender el Led de nuevo al tocar el pin touch.

 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      |

| Criterios    | Descripción   | Puntaje |
|--------------|---|---------|
| 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](#)