

PRÁCTICA # 01. CONOZCA SUS HERRAMIENTAS		Revisión:
<p>Objetivo: Comprender los principios básicos de la electricidad y el uso de una placa de pruebas para construir circuitos electrónicos simples.</p> <p>Materiales:</p> <ul style="list-style-type: none"> • Microcontrolador ESP32 • <i>Protoboard</i> • <i>LED</i> • Resistencias (220Ω) • Pulsador • Cables de conexión • Fuente de alimentación (USB o batería de 9V) <p>Desarrollo paso a paso:</p> <ol style="list-style-type: none"> 1. Desconectar la alimentación del ESP32 antes de conectar cualquier componente. 2. Conectar un cable desde el pin 3.3V del ESP32 a la línea de alimentación (+) del protoboard. 3. Conectar un cable desde GND del ESP32 a la línea de tierra (-) del protoboard. 4. Colocar el pulsador en el centro del protoboard. 5. Conectar una resistencia de 220Ω entre la línea de alimentación (+) y uno de los terminales del pulsador. 		<ol style="list-style-type: none"> 6. Conectar el ánodo (pata larga) del LED al otro lado del pulsador. 7. Conectar el cátodo (pata corta) del LED a tierra (-). 8. Alimentar el ESP32. 9. Presionar el pulsador y verificar que el LED enciende. <p>Conclusiones:</p> <ul style="list-style-type: none"> • La electricidad necesita un circuito cerrado para fluir. • La resistencia protege el LED de sobrecarga. • Un pulsador en serie controla el paso de corriente de forma manual. <p>Pregunta para el alumno: <i>¿Qué sucede si conectas el LED sin una resistencia en serie?</i> <i>Explica el motivo de tu respuesta.</i></p> <div style="border: 1px solid black; height: 60px; width: 100%;"></div>

PRÁCTICA # 02. INTERFAZ DE NAVE ESPACIAL		Revisión:
<p>Objetivo: Construir un panel de control con luces que se activan al presionar un botón, aprendiendo sobre entradas y salidas digitales.</p> <p>Materiales:</p> <ul style="list-style-type: none"> • ESP32 • Placa de pruebas • 1 LED verde • 2 LEDs rojos • 3 resistencias de 220Ω • Pulsador • Resistencia de 10KΩ • Cables de conexión <p>Desarrollo paso a paso:</p> <ol style="list-style-type: none"> 1. Conectar el protoboard a los pines de 3.3V y GND del ESP32. 2. Colocar el LED verde y conectarlo en serie con una resistencia de 220Ω al pin digital GPIO18 del ESP32. 3. Colocar los LEDs rojos y conectarlos en serie con resistencias de 220Ω a los pines digitales GPIO19 y GPIO21. 4. Colocar el pulsador en el protoboard. 		<ol style="list-style-type: none"> 5. Conectar un extremo del pulsador a la alimentación y el otro al pin digital GPIO23 del ESP32. 6. Agregar una resistencia de 10KΩ entre el pulsador y GND (para evitar lecturas erróneas). 7. Subir el código al ESP32 y probar el funcionamiento del panel de control. 8. Presionar el botón y verificar que los LEDs responden según lo programado. <p>Conclusiones:</p> <ul style="list-style-type: none"> • Se ha aprendido a usar pines digitales para controlar LEDs con una señal de entrada. • La resistencia de 10KΩ ayuda a evitar falsos positivos en la lectura del pulsador. • La programación permite que el LED verde se apague y los LEDs rojos parpadeen al presionar el botón. <p>Pregunta para el alumno: <i>¿Cómo podrías modificar el código para que los LEDs rojos parpadeen más rápido o más lento?</i></p> <div style="border: 1px solid black; height: 60px; width: 100%;"></div>

PRÁCTICA # 03. MEDIDOR DE ENAMORAMIENTO		Revisión:
Objetivo: Utilizar un sensor de temperatura para medir la temperatura de la piel y visualizar el resultado con LEDs, aprendiendo sobre entradas analógicas en ESP32.		
Materiales: <ul style="list-style-type: none">• ESP32• Protoboard• Sensor de temperatura TMP36• 3 LEDs (rojo, amarillo y verde)• 3 resistencias de 220Ω• Cables de conexión		
Desarrollo paso a paso: <ol style="list-style-type: none">1. Conectar la alimentación y GND del ESP32 al protoboard.2. Instalar el sensor de temperatura en el protoboard (con su lado plano orientado correctamente).3. Conectar el pin izquierdo del sensor a 3.3V, el derecho a GND y el pin central al pin GPIO34 del ESP32.4. Conectar los LEDs a los pines digitales GPIO25, GPIO26 y GPIO27, cada uno en serie con una resistencia de 220Ω.	<ol style="list-style-type: none">5. Subir el código al ESP32 para que lea el sensor y encienda los LEDs según la temperatura detectada.6. Probar el sistema: Tocar el sensor con la mano y verificar que los LEDs cambian según la temperatura.7. Utilizar el monitor serie para visualizar las lecturas del sensor.	
	Conclusiones: <ul style="list-style-type: none">• Se ha aprendido a usar un sensor analógico y convertir su señal en datos digitales.• La temperatura de la piel afecta la lectura del sensor y enciende diferentes LEDs según su valor.• El monitor serie permite visualizar en tiempo real la información del sensor.	
	Pregunta para el alumno: <i>¿Cómo podrías modificar el código para que en lugar de LEDs, se muestre la temperatura en una pantalla OLED?</i>	

PRÁCTICA # 04. LÁMPARA DE MEZCLA DE COLORES		Revisión:
Objetivo: Construir una lámpara con LEDs de colores que cambia según la cantidad de luz detectada.		
Materiales: <ul style="list-style-type: none">• ESP32• Protoboard• 3 LEDs (rojo, verde y azul)• 3 resistencias de 220Ω• Fotoresistencia (LDR)• Resistencia de 10KΩ• Cables de conexión		
Desarrollo paso a paso: <ol style="list-style-type: none">1. Conectar la alimentación y GND del ESP32 al protoboard.2. Instalar la fotoresistencia en el protoboard y conectarla en serie con la resistencia de 10KΩ entre 3.3V y GND.3. Conectar el punto medio entre la fotoresistencia y la resistencia de 10KΩ al pin GPIO35 del ESP32.4. Conectar los LEDs a los pines digitales GPIO16, GPIO17 y GPIO5 con resistencias de 220Ω.5. Subir el código al ESP32 y probar el comportamiento de los LEDs al variar la luz ambiente.		
<ol style="list-style-type: none">6. Probar el funcionamiento: Cubrir la fotoresistencia y verificar que los colores cambian. Exponer la fotoresistencia a más luz y observar la variación en los LEDs.		
Aspectos a revisar en el circuito: <ul style="list-style-type: none">• Verificar que la LDR está correctamente conectada en serie con la resistencia de 10KΩ.• Asegurar que los LEDs responden correctamente a los cambios de luz.• Comprobar que los pines GPIO del ESP32 están correctamente asignados.• Revisar en el monitor serie los valores de la fotoresistencia para confirmar la lectura.		
Conclusiones: <ul style="list-style-type: none">• Se ha aprendido a usar una fotoresistencia para medir la cantidad de luz ambiente.• Los LEDs de colores permiten crear diferentes tonalidades según la intensidad de la luz.• Se puede modificar el código para ajustar la sensibilidad de la fotoresistencia.		
Pregunta para el alumno: <i>¿Cómo podrías modificar el código para que los colores cambien de forma más gradual en función de la luz?</i>		

PRÁCTICA # 05. INDICADOR DEL ESTADO DE ÁNIMO		Revisión:
Objetivo: Construir un sistema que cambia el color de un LED RGB en función de la presión ejercida sobre un sensor.	Aspectos a revisar en el circuito: <ul style="list-style-type: none">• Comprobar que el sensor de presión varía su salida al ser presionado.• Asegurar que el LED RGB cambia de color correctamente.• Revisar en el monitor serie los valores del sensor para confirmar que el ESP32 los está leyendo correctamente.• Verificar que los GPIOs asignados al LED y al sensor son los correctos.	
Materiales: <ul style="list-style-type: none">• ESP32• Protoboard• LED RGB• Sensor de presión• 3 resistencias de 220Ω• Cables de conexión	Conclusiones: <ul style="list-style-type: none">• Se ha aprendido a usar un sensor de presión como entrada analógica.• Los LEDs RGB permiten mezclar colores para generar diferentes tonalidades.• La presión ejercida sobre el sensor controla el color del LED, lo que puede aplicarse a interfaces interactivas.	
Desarrollo paso a paso: <ol style="list-style-type: none">1. Conectar la alimentación y GND del ESP32 al protoboard.2. Conectar los tres pines del LED RGB a los pines digitales GPIO13, GPIO14 y GPIO15 del ESP32 con resistencias de 220Ω.3. Conectar el sensor de presión a 3.3V y GND, y su salida al pin GPIO36 del ESP32.4. Subir el código al ESP32 para leer el sensor y cambiar el color del LED en función de la presión ejercida.5. Probar el funcionamiento y observar cómo cambia el color según la presión aplicada.	Pregunta para el alumno: <i>¿Cómo podrías modificar el código para que el LED RGB también cambie de color en función del tiempo?</i>	

PRÁCTICA # 06. THEREMIN CONTROLADO POR LUZ		Revisión:
Objetivo: Construir un instrumento musical electrónico que cambia el tono según la cantidad de luz detectada por un sensor.		
Materiales: <ul style="list-style-type: none">• ESP32• Protoboard• Zumbador piezoeléctrico• Fotorresistencia (LDR)• Resistencia de 10KΩ• Cables de conexión		
Desarrollo paso a paso: <ol style="list-style-type: none">1. Conectar la alimentación (3.3V) y GND del ESP32 al protoboard.2. Conectar la fotorresistencia en serie con la resistencia de 10KΩ entre 3.3V y GND.3. Conectar el punto medio entre la fotorresistencia y la resistencia de 10KΩ al pin GPIO34 del ESP32.4. Conectar el zumbador al pin GPIO25 con su otro terminal a GND.5. Subir el código al ESP32: leer la señal de la fotorresistencia desde GPIO34. Ajustar la frecuencia del zumbador en función de la cantidad de luz detectada.		
<ol style="list-style-type: none">6. Probar el sistema: Encender el ESP32 y abrir el monitor serie para verificar los valores de luz detectados. Tapar parcialmente la fotorresistencia con la mano y observar cómo cambia la frecuencia del sonido. Exponer la fotorresistencia a más luz y verificar si la frecuencia del zumbador aumenta o disminuye.		
Aspectos a revisar en el circuito: <ul style="list-style-type: none">• Verificar que la fotorresistencia está correctamente conectada en serie con la resistencia de 10KΩ.• Asegurar que el zumbador piezoeléctrico está en el pin correcto y tiene buena conexión a GND.• Comprobar que la salida de voltaje del sensor varía en función de la luz ambiente usando el monitor serie.		
Conclusiones: <ul style="list-style-type: none">• Se ha aprendido a usar la fotorresistencia como entrada analógica para controlar una señal de salida de audio.• El ESP32 puede generar tonos variables según la cantidad de luz recibida.		
Pregunta para el alumno: <i>¿Cómo podrías modificar el código para generar diferentes escalas musicales?</i>		

PRÁCTICA # 07. TECLADO MUSICAL		Revisión:
<p>Objetivo: Construir un teclado simple que emite sonidos diferentes según el botón presionado.</p> <p>Materiales:</p> <ul style="list-style-type: none"> • ESP32 • Protoboard • 4 pulsadores • Zumbador piezoeléctrico • Resistencias de 10KΩ (una por cada pulsador) • Cables de conexión <p>Desarrollo paso a paso:</p> <ol style="list-style-type: none"> 1. Conectar la alimentación y GND del ESP32 al protoboard. 2. Conectar cada pulsador en un extremo a 3.3V, y el otro extremo a los pines GPIO32, GPIO33, GPIO26, y GPIO27. 3. Conectar una resistencia de 10KΩ entre cada pulsador y GND. 4. Conectar el zumbador al pin GPIO25 con su otro terminal a GND. 5. Subir el código al ESP32: configurar para detectar qué pulsador ha sido presionado. Generar una frecuencia específica en el zumbador dependiendo del botón presionado. 		<p>6. Probar el sistema: abrir el monitor serie para verificar que los botones se detectan. Presionar cada botón individualmente y comprobar si el zumbador emite un sonido diferente. Presionar dos o más botones al mismo tiempo y observar si la combinación de tonos. Ajustar el código para probar diferentes escalas musicales.</p> <p>Aspectos a revisar en el circuito:</p> <ul style="list-style-type: none"> • Cada pulsador tiene su respectiva resistencia de 10KΩ para evitar lecturas incorrectas. • Asegurar que el zumbador está bien conectado y responde a la presión de los botones. • Verificar en el monitor serie que el ESP32 detecta correctamente cada pulsador. <p>Conclusiones:</p> <ul style="list-style-type: none"> • Se ha aprendido a usar múltiples entradas digitales para generar diferentes respuestas sonoras. • Los pulsadores permiten generar un control interactivo sobre los sonidos. • El ESP32 puede procesar varias entradas al mismo tiempo. <p>Pregunta para el alumno: <i>¿Cómo podrías agregar más notas al teclado sin usar más pines digitales?</i></p>

PRÁCTICA # 08. RELOJ DE ARENA DIGITAL		Revisión:
<p>Objetivo: Construir un reloj de arena electrónico con LEDs que se apagan progresivamente al inclinarlo.</p> <p>Materiales:</p> <ul style="list-style-type: none"> • ESP32 • Protoboard • 5 LEDs • 5 resistencias de 220Ω • Sensor de inclinación • Cables de conexión <p>Desarrollo paso a paso:</p> <ol style="list-style-type: none"> 1. Conectar la alimentación y GND del ESP32 al protoboard. 2. Conectar los LEDs a los pines GPIO13, GPIO14, GPIO15, GPIO16 y GPIO17, cada uno con una resistencia de 220Ω. 3. Conectar el sensor de inclinación entre 3.3V y GPIO35, con una resistencia de 10KΩ a GND. 4. Subir el código al ESP32: configurar para que detecte cuando el sensor de inclinación cambia de estado. Programar el apagado progresivo de los LEDs cuando el circuito se inclina. Reiniciar la secuencia de encendido de LEDs cuando el circuito regrese a su posición original. 		<p>5. Probar el sistema: todos los LEDs están encendidos al inicio. Al inclinar el circuito los LEDs se apagan uno por uno. Volver a la posición original y verificar que los LEDs se encienden nuevamente. Observar en el monitor serie si se están detectando correctamente los cambios en el sensor de inclinación.</p> <p>Aspectos a revisar en el circuito:</p> <ul style="list-style-type: none"> • Comprobar que los LEDs están conectados en los pines correctos y en el orden adecuado. • Asegurar que el sensor de inclinación cambia su estado cuando se mueve el circuito. • Verificar en el monitor serie la respuesta del sensor al inclinarlo. <p>Conclusiones:</p> <ul style="list-style-type: none"> • Se ha aprendido a usar un sensor de inclinación como entrada digital. • Los LEDs pueden actuar como una representación visual del paso del tiempo. • Se pueden crear variaciones agregando diferentes tiempos de apagado. <p>Pregunta para el alumno: <i>¿Cómo podrías modificar el código para que el reloj de arena tenga una duración ajustable?</i></p>

PRÁCTICA # 09. RUEDA DE COLORES MOTORIZADA		Revisión:
Objetivo: Construir un sistema que gira un disco de colores utilizando un motor, creando efectos visuales.	<div>7. Fijar el disco de colores al motor: Cortar un disco de cartón con diferentes colores. Montar el disco en el eje del motor con pegamento o un soporte adecuado.</div> <div>8. Probar el sistema: verificar que el motor comienza a girar. Ajustar la velocidad del motor y observar cómo cambia el efecto visual. Comprobar que el motor responde.</div>	
Materiales: <ul style="list-style-type: none">• ESP32• Motor DC• Transistor NPN 2N2222• Diodo 1N4007• Resistencia de 1KΩ• Fuente de alimentación externa (opcional)		
Desarrollo paso a paso: <div><div>1. Conectar la alimentación y GND del ESP32 al protoboard.</div><div>2. Conectar la base del transistor a GPIO19 a través de una resistencia de 1KΩ.</div><div>3. Conectar el colector del transistor al motor y el emisor a GND.</div><div>4. Conectar el otro terminal del motor a 3.3V o a una fuente externa si se requiere más potencia.</div><div>5. Agregar un diodo en paralelo al motor para proteger el circuito.</div><div>6. Subir el código y programar para activar y desactivar el motor. Ajustar la velocidad de rotación usando modulación por ancho de pulso.</div></div>	Aspectos a revisar en el circuito: <ul style="list-style-type: none">• Comprobar que el transistor está bien conectado para controlar la corriente del motor.• Verificar que el diodo está en la dirección correcta para evitar daños al circuito.• Asegurar que el motor gira correctamente y responde a los comandos del código. Conclusiones: <ul style="list-style-type: none">• Se ha aprendido a controlar un motor DC con un transistor.• El ESP32 puede manejar cargas más grandes utilizando transistores. Pregunta para el alumno: ¿Cómo podrías modificar el código para que la velocidad del motor sea ajustable?	

PRÁCTICA # 10.		Revisión:
Objetivo: Construir un zoótropo que muestra una animación en un disco giratorio mediante el control de un motor DC con ESP32.		
Materiales: <ul style="list-style-type: none">• ESP32• Protoboard• Motor DC• Transistor NPN• Diodo 1N4007• Resistencia de 1KΩ• Cables de conexión• Cartón con dibujos animados en un disco		
Desarrollo paso a paso: <ol style="list-style-type: none">1. Conectar la alimentación y GND del ESP32 al protoboard.2. Conectar la base del transistor NPN al pin GPIO19 del ESP32 a través de una resistencia de 1KΩ.3. Conectar el colector del transistor al motor y el emisor a GND.4. Conectar el otro terminal del motor a 3.3V o a una fuente externa si se requiere más potencia.5. Agregar un diodo en paralelo al motor para proteger el circuito.		
<ol style="list-style-type: none">6. Fijar el disco con dibujos en el eje del motor.7. Ajustar la velocidad del motor en el código para optimizar la animación.8. Subir el código al ESP32 y probar el efecto visual.		
Aspectos a revisar en el circuito: <ul style="list-style-type: none">• Verificar la conexión del motor y su respuesta al código.• Asegurar que el transistor está bien colocado y controla correctamente la corriente del motor.• Ajustar la velocidad del motor para que la animación se vea fluida.• Comprobar que el disco está bien sujeto y no se desbalancea al girar.		
Conclusiones: <ul style="list-style-type: none">• Ajustando la velocidad del motor, se puede mejorar la fluidez de la animación en el zoótropo.• Un diodo en paralelo al motor es fundamental para proteger el circuito de picos de voltaje generados por la bobina del motor.		
Pregunta para el alumno: <i>¿Cómo podrías sincronizar la velocidad del zoótropo con un LED estroboscópico para mejorar la percepción de la animación?</i>		

PRÁCTICA # 11. LA BOLA DE CRISTAL		Revisión:
<p>Objetivo:</p> <p>Construir un dispositivo que muestra respuestas aleatorias en una pantalla OLED SSD1306 cuando se agita.</p> <p>Materiales:</p> <ul style="list-style-type: none">• ESP32• Protoboard• Pantalla OLED SSD1306 (comunicación I2C)• Sensor de inclinación• Resistencia de 10KΩ• Cables de conexión <p>Desarrollo paso a paso:</p> <ol style="list-style-type: none">1. Conectar la alimentación (3.3V) y GND del ESP32 al protoboard.2. Conectar el sensor de inclinación entre 3.3V y GPIO35, con una resistencia de 10KΩ a GND.3. Conectar la pantalla OLED SSD1306 de la siguiente manera:<ul style="list-style-type: none">○ VCC → 3.3V○ GND → GND○ SDA → GPIO21○ SCL → GPIO224. Subir el código al ESP32 y probar cómo se generan respuestas aleatorias en la pantalla al agitar el dispositivo.	<p>Aspectos a revisar en el circuito:</p> <ul style="list-style-type: none">• Verificar que el sensor de inclinación responde correctamente al movimiento.• Asegurar que la pantalla OLED SSD1306 se comunica correctamente con el ESP32 a través de I2C.• Comprobar en el monitor serie que los valores generados por el sensor cambian al agitar el circuito. <p>Conclusiones:</p> <ul style="list-style-type: none">• Se ha aprendido a utilizar un sensor de inclinación para generar eventos aleatorios.• La pantalla OLED SSD1306 permite visualizar el resultado en lugar de usar LEDs.• Se pueden modificar las respuestas almacenadas en el código para personalizar el dispositivo. <p>Pregunta para el alumno:</p> <p><i>¿Cómo podrías modificar el código para agregar diferentes categorías de respuestas?</i></p>	

PRÁCTICA # 12. MECANISMO DE BLOQUEO SECRETO		Revisión:
<p>Objetivo:</p> <p>Crear un sistema de seguridad que solo permite el acceso con un código correcto, usando pulsadores y una pantalla OLED SSD1306 con ESP32.</p> <p>Materiales:</p> <ul style="list-style-type: none">• ESP32• Protoboard• 4 pulsadores• 4 resistencias de 10KΩ• Pantalla OLED SSD1306 (I2C)• Cables de conexión <p>Desarrollo paso a paso:</p> <ol style="list-style-type: none">1. Conectar la alimentación del ESP32:2. Conectar los pulsadores: Colocar cuatro pulsadores en el protoboard.<ul style="list-style-type: none">• Conectar un lado de cada pulsador a 3.3V.• Conectar el otro lado de cada pulsador a los pines GPIO32, GPIO33, GPIO26 y GPIO27 del ESP32.• Conectar una resistencia de 10KΩ entre cada pulsador y GND (pull-down).3. Conectar la pantalla OLED SSD 1306.4. Subir el código al ESP32.	<p>Aspectos a revisar en el circuito:</p> <ul style="list-style-type: none">• Asegurar que los pulsadores detectan correctamente la presión.• Verificar que la pantalla OLED SSD1306 muestra los intentos de acceso.• Comprobar en el monitor serie si el código ingresado coincide con el esperado. <p>Conclusiones:</p> <ul style="list-style-type: none">• Se ha aprendido a manejar múltiples entradas digitales con ESP32.• La pantalla OLED SSD1306 permite una mejor interfaz de usuario.• Se pueden agregar funciones como tiempo de espera o intentos limitados. <p>Pregunta para el alumno:</p> <p><i>¿Cómo podrías modificar el código para que el sistema bloquee el acceso después de tres intentos fallidos?</i></p>	

PRÁCTICA # 13. LÁMPARA SENSIBLE AL TACTO		Revisión:
<p>Objetivo: Construir una lámpara que enciende y cambia de intensidad al tocar un sensor capacitivo.</p> <p>Materiales:</p> <ul style="list-style-type: none">• ESP32• Protoboard• LED de alto brillo• Resistencia de 220Ω• Sensor capacitivo (placa de aluminio o cable expuesto)• Cables de conexión <p>Desarrollo paso a paso:</p> <ol style="list-style-type: none">1. Conectar la alimentación y GND del ESP32 al protoboard.2. Conectar el LED al pin GPIO16 con una resistencia de 220Ω.3. Conectar el sensor capacitivo al pin GPIO4 del ESP32.4. Subir el código al ESP32 y probar el efecto de encender y atenuar la luz con el tacto.	<p>Aspectos a revisar en el circuito:</p> <ul style="list-style-type: none">• Asegurar que el LED tiene una resistencia de 220Ω para evitar sobrecarga.• Verificar que el sensor capacitivo responde correctamente al tacto.• Comprobar en el monitor serie que el ESP32 detecta cambios de capacitancia. <p>Conclusiones:</p> <ul style="list-style-type: none">• Se ha aprendido a usar entradas capacitivas para interactuar con un LED.• El ESP32 permite ajustar la sensibilidad del sensor para mejorar la respuesta.• Se pueden agregar múltiples sensores para controlar más dispositivos. <p>Pregunta para el alumno: <i>¿Cómo podrías modificar el código para que el LED tenga diferentes niveles de brillo según la duración del toque?</i></p>	

PRÁCTICA # 14. RETOCAR EL LOGOTIPO DE ARDUINO		Revisión:
<p>Objetivo:</p> <p>Mostrar una imagen en la pantalla OLED SSD1306 y permitir su modificación con botones.</p> <p>Materiales:</p> <ul style="list-style-type: none">• ESP32• Protoboard• Pantalla OLED SSD1306 (I2C)• 2 pulsadores• Resistencia de 10KΩ para cada pulsador• Cables de conexión <p>Desarrollo paso a paso:</p> <ol style="list-style-type: none">1. Conectar la alimentación y GND del ESP32 al protoboard.2. Conectar los pulsadores a los pines GPIO32 y GPIO33, cada uno con una resistencia de 10KΩ a GND.3. Conectar la pantalla OLED SSD1306 como en los proyectos anteriores.4. Subir el código al ESP32 y probar cómo los botones modifican la imagen en la pantalla.	<p>Aspectos a revisar en el circuito:</p> <ul style="list-style-type: none">• Verificar que los pulsadores funcionan correctamente con sus respectivas resistencias.• Asegurar que la pantalla OLED SSD1306 carga correctamente la imagen.• Comprobar en el monitor serie que los botones generan eventos al ser presionados. <p>Conclusiones:</p> <ul style="list-style-type: none">• Se ha aprendido a manipular imágenes en una pantalla OLED SSD1306.• Los botones pueden usarse para modificar elementos visuales en una interfaz gráfica.• Se pueden agregar más botones o sensores para mejorar la interactividad. <p>Pregunta para el alumno:</p> <p><i>¿Cómo podrías modificar el código para que la imagen se mueva en la pantalla sin necesidad de botones?</i></p>	

PRÁCTICA # 15. HACKEAR BOTONES		Revisión:
<p>Objetivo: Simular la pulsación de un botón en otro dispositivo usando transistores.</p> <p>Materiales:</p> <ul style="list-style-type: none"> • ESP32 • Protoboard • Transistor NPN • Resistencia de 1KΩ • Cables de conexión <p>Desarrollo paso a paso:</p> <ol style="list-style-type: none"> 1. Conectar la alimentación y GND del ESP32 al protoboard. 2. Conectar la base del transistor al pin GPIO23 con una resistencia de 1KΩ. 3. Conectar el colector del transistor al botón que se quiere hackear. 4. Conectar el emisor del transistor a GND. 5. Subir el código al ESP32 y probar si el circuito simula la pulsación del botón. 		<p>Aspectos a revisar en el circuito:</p> <ul style="list-style-type: none"> • Asegurar que el transistor está correctamente conectado. • Verificar que el ESP32 activa el transistor cuando se ejecuta el código. • Comprobar que el dispositivo externo responde como si se presionara el botón físicamente. <p>Conclusiones:</p> <ul style="list-style-type: none"> • Se ha aprendido a controlar dispositivos externos mediante transistores. • Se puede usar este principio para automatizar pulsaciones en otros sistemas. <p>Pregunta para el alumno: <i>¿Cómo podrías agregar un temporizador para activar el botón en intervalos específicos?</i></p>

EXAMEN		Revisión: