Instituto Tecnológico®
de Aguascalientes


**INSTITUTO TECNOLOGICO DE AGUASCALIENTES**

**Integrante:**

**Manuel Humberto Reyes Salas**

**María José Guerrero Rios**

**Jesús Eli Sánchez Ruvalcaba**


**Actividad:** Practica 1

**Materia:** Tecnologías inalámbricas

**Docente:** RICARDO ALEJANDRO RODRIGUEZ JIMENEZ

**Fecha:** 19 de Febrero del 2026

# Contenido

# Información

## Que es BLE?

Bluetooth Low Energy (BLE) es una tecnología inalámbrica de bajo consumo, que se usa para la transferencia de pequeñas cantidades de datos entre dispositivos compatibles a corta distancia. Es una versión más moderna y eficiente del protocolo Bluetooth tradicional, con el que se pueden lograr velocidades similares pero con menor gasto energético.

Esta tecnología inalámbrica se ha ido imponiendo en el mercado y su uso se ha extendido a muchos dispositivos, desde relojes inteligentes hasta cascos inalámbricos

## Principales Características del Bluetooth y el BLE

- Alcance . El alcance del Bluetooth clásico es mayor que el de BLE. El alcance del Bluetooth clásico puede llegar hasta los 100 metros. Mientras que el de BLE está limitado a solo 10 metros.
- Velocidad . La velocidad del Bluetooth clásico es mucho mayor que la del BLE. El Bluetooth clásico puede transferir datos a velocidades de hasta 3 Mbps, mientras que el BLE transmite datos a velocidades mucho más bajas, generalmente entre 100 Kbps y 1 Mbps.
- Energía . El consumo de energía también varía entre Bluetooth clásico y BLE. El consumo energético del Bluetooth clásico es mucho mayor que el del BLE porque consume más energía para conectar y mantener la conexión con los dispositivos remotos.
- El uso principal del Bluetooth clásico se centra en la transmisión de audio (compartir archivos, intercambiar archivos multimedia). Por otro lado, el uso principal del BLE se centra en soluciones IoT ( Internet of Things) tales como sensores inalámbricos, dispositivos wearables y otros dispositivos domésticos inteligentes.

## Placa ESP-32

Es un microcontrolador creado por Espressif Systems. Es muy usado en proyectos de IoT porque integra WiFi y Bluetooth, además de ser potente y económico.

El ESP32 funciona como el "cerebro" de un sistema electrónico. Ejecuta un programa (firmware) que tú cargas desde tu computadora (usualmente con Arduino IDE o ESP-IDF).

¿Cómo funciona?

Se alimenta (3.3V o 5V según la placa).

Ejecuta un programa cargado desde tu computadora.

Lee entradas (sensores, botones).

Procesa la información.

Controla salidas (LEDs, motores, WiFi, Bluetooth).

Partes principales: 1. Procesador: Doble núcleo hasta 240 MHz, 2. Memoria: Flash (programa), SRAM (trabajo), ROM (arranque), 3. Conectividad: WiFi + Bluetooth (BLE), 3. Periféricos: ADC, DAC, PWM, SPI, I2C, UART, timers, 4. Modos de energía: Activo, Sleep y Deep Sleep (bajo consumo).

El ESP32 tiene aproximadamente 30-36 pines programables, llamados GPIO.

¿Para qué se usa?

- Domótica
- Monitoreo remoto
- Robots
- Control de motores
- Servidores web
- Proyectos IoT

En pocas palabras

El ESP32 es un microcontrolador potente, económico y con conexión inalámbrica integrada, ideal para proyectos inteligentes y conectados a internet.

# Practica

## Explicación de la practica

En esta práctica desarrollamos un sistema de control inalámbrico utilizando tecnología Bluetooth Low Energy (BLE), cuyo propósito fue controlar el encendido y apagado de un LED desde un teléfono móvil mediante una placa **ESP32**.

**Programación en Arduino (ESP32)**

Posteriormente realizamos la programación del ESP32 en el entorno de Arduino IDE. En el código configuramos el pin del LED como salida y activamos la función de Bluetooth del microcontrolador, asignándole un nombre para que pudiera ser identificado desde el celular.

También programamos el ESP32 para que permaneciera en espera de datos enviados por Bluetooth. Cuando recibe un mensaje, el programa verifica su contenido:

Si recibe la palabra **ON**, activa el pin y enciende el LED.

Si recibe **OFF**, desactiva el pin y apaga el LED.

De esta manera, el ESP32 actúa como receptor de los comandos enviados desde la aplicación móvil y ejecuta las acciones correspondientes.

**Desarrollo de la aplicación móvil**

Después diseñamos una aplicación en App Inventor. En ella configuramos los permisos necesarios para utilizar el Bluetooth y agregamos un botón para buscar dispositivos cercanos. Los dispositivos detectados se muestran en una lista, permitiendo seleccionar el ESP32 para establecer la conexión.

Una vez conectado, la aplicación cambia su estado a “conectado” y habilita los botones que envían los comandos “ON” y “OFF”. También incluimos un botón para desconectar el dispositivo cuando fuera necesario.

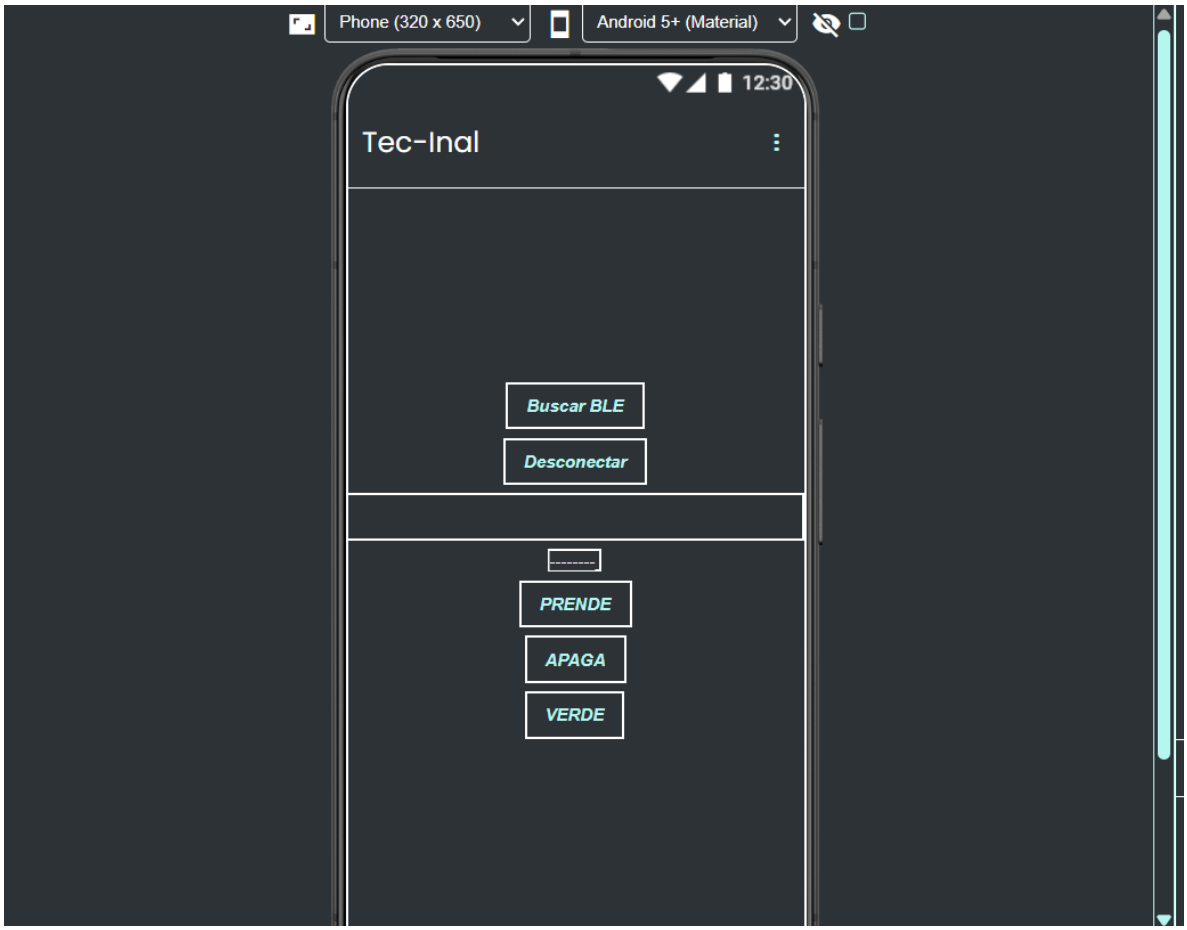
**Pruebas realizadas**

Finalmente realizamos varias pruebas para comprobar que el sistema funcionara correctamente. Confirmamos que al presionar el botón correspondiente el LED respondía de manera inmediata, encendiéndose o apagándose según el comando enviado.

Con esta práctica logramos comprobar el correcto funcionamiento de la comunicación inalámbrica entre el teléfono y el ESP32.

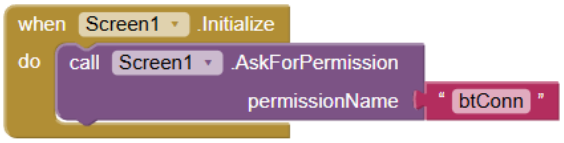
## Código de bloques de app inventor

## Diseño:

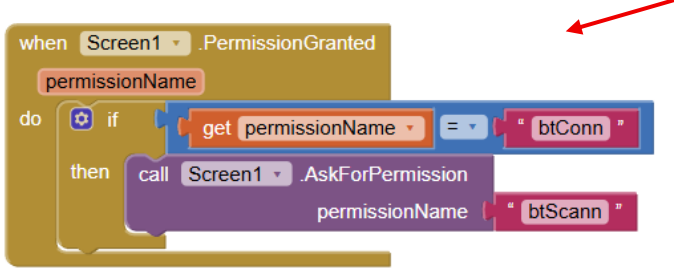


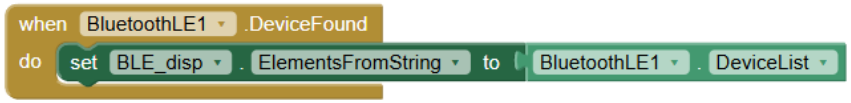

## Código en bloques

En estos primeros bloques lo que estamos haciendo es que cuando se inicie la aplicación pida permiso para poder usar el bluetooth y después pedir permiso para conectarse a dispositivos bluettoth cercanos.

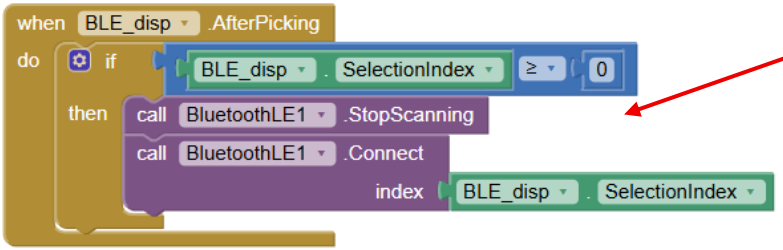



En esta parte lo que realiza es que todos los dispositivos bluetooth que encuentre los va agregando a una lista

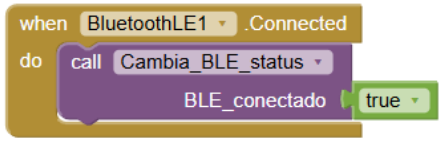


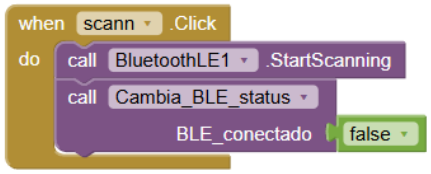

Lo que realiza es que verifica si se selecciono algo, si esto es verdadero deja de escanear y se conecta al dispositivo seleccionado



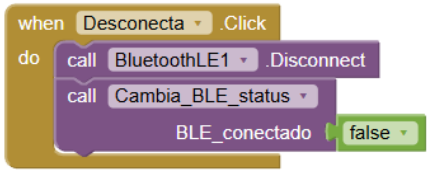

Cuando la conexión fue exitosa el estado cambia para indicar que esta conectado y manda llamar un método

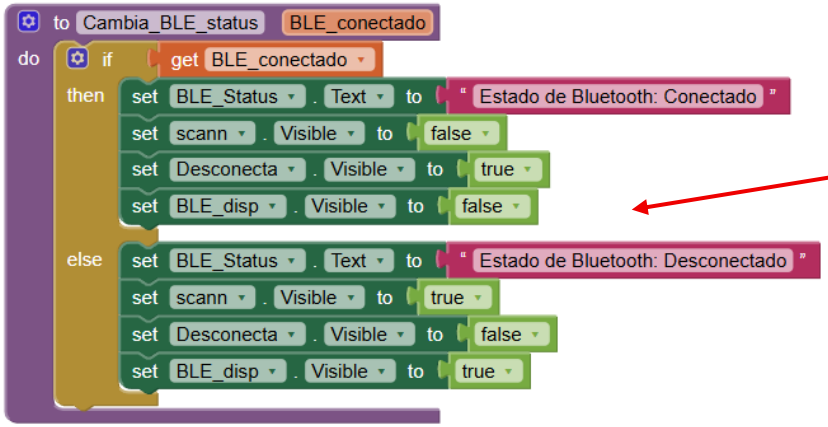


Cuando se presiona el botón de desconecta, este cierra la conexión bluetooth y lo pone en estado de desconectado



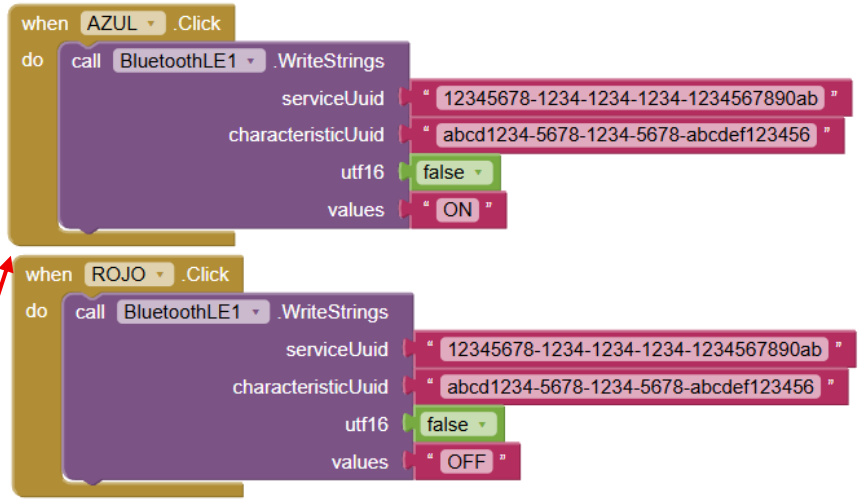

En estos bloques lo que realiza es que al momento de presionar el botón de scann, empieza a buscar dispositivos bluetooth cercanos y pone el estado en desconectado mientras busca



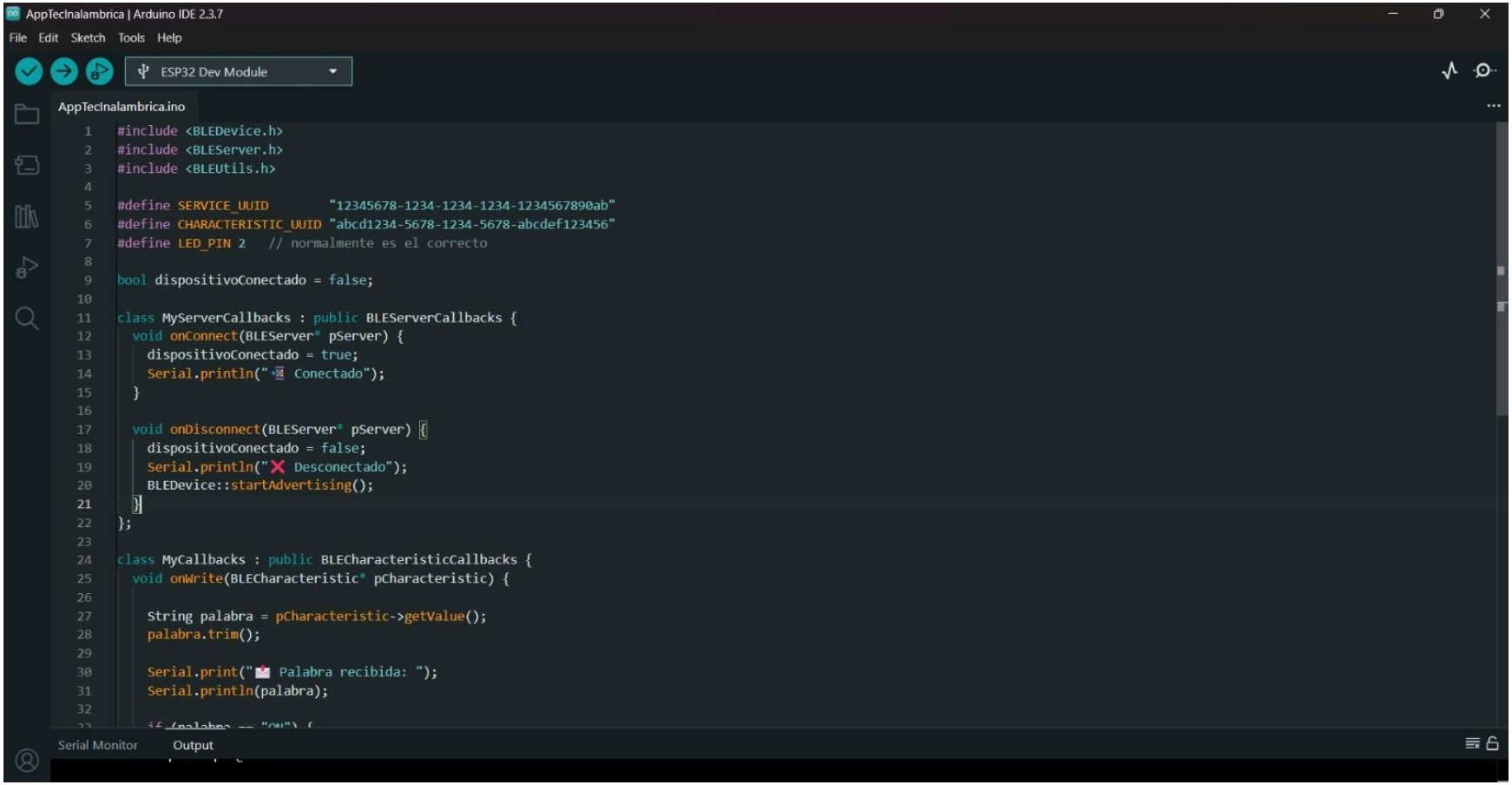

Este es el bloque mas importante, lo primero que realiza es que recibe el parámetro si esta conectado o no si está conectado lo que realiza es que muestra un mensaje de conectado y oculta el botón de scann y la lista de dispositivos cercanos, además de que muestra el boton para desconectar.

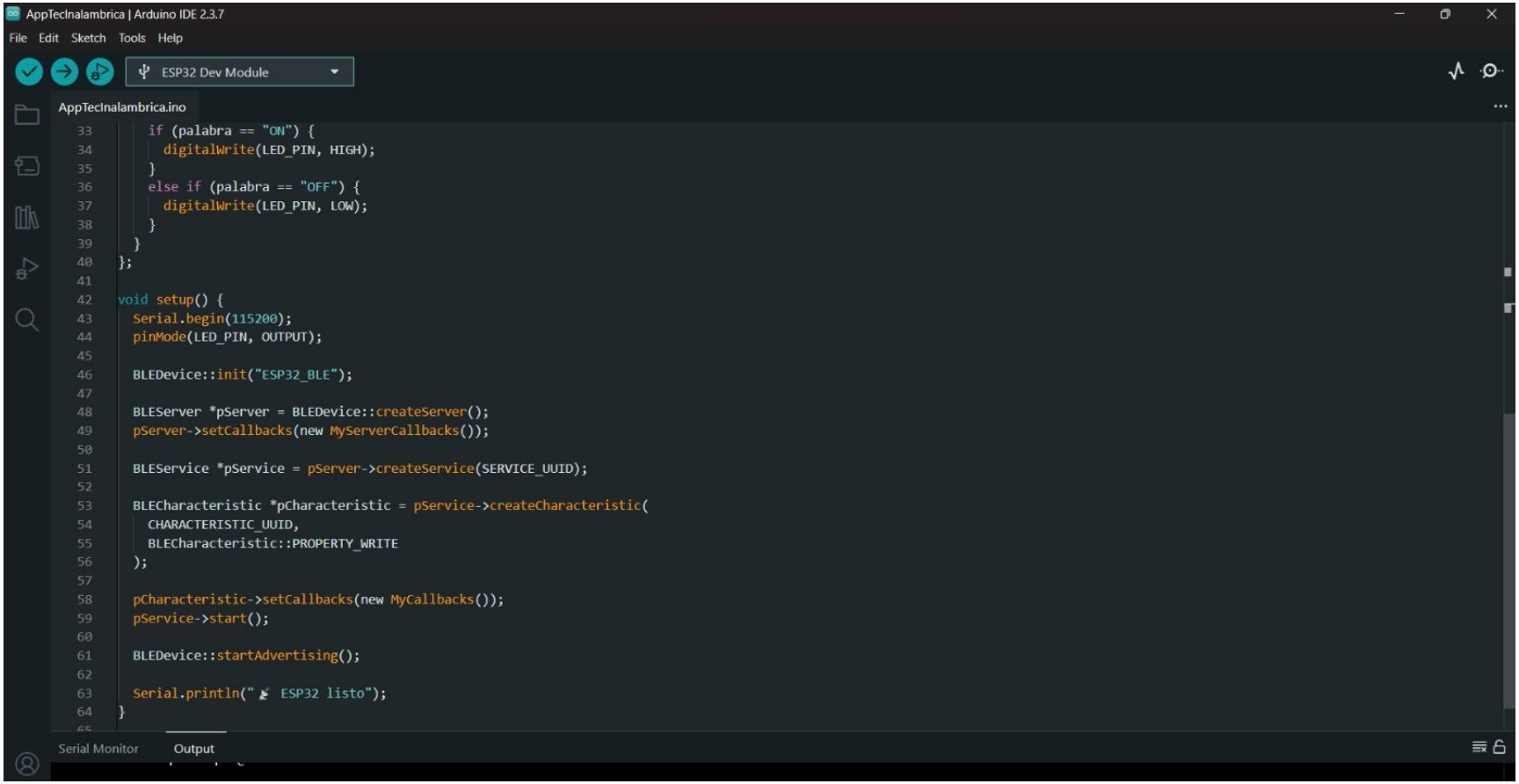
Si recibe el parámetro que esta desconectado lo que realiza es que muestra el mensaje de desconectado y permite volver a escanear dispositivos y oculta el botón de desconectar.

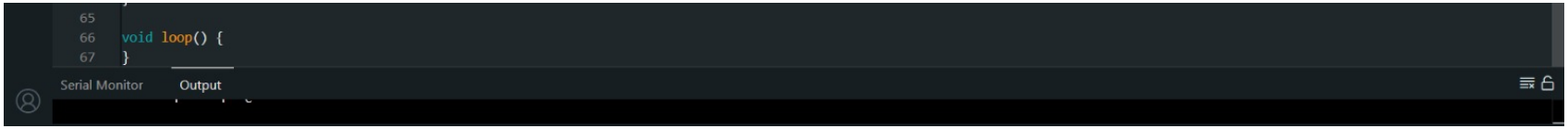



En estos bloques lo que estamos realizando es que estamos enviando comandos al dispositivo bluetooth, cuando presionamos el botón Azul la aplicación envia el texto ON a el dispositivo y este comando funciona para prender un led, mientras que por el otro lado si presionamos el botón Rojo realiza lo contrario y la aplicación manda el texto OFF a el dispositivo y este comando funciona para apagar el led.

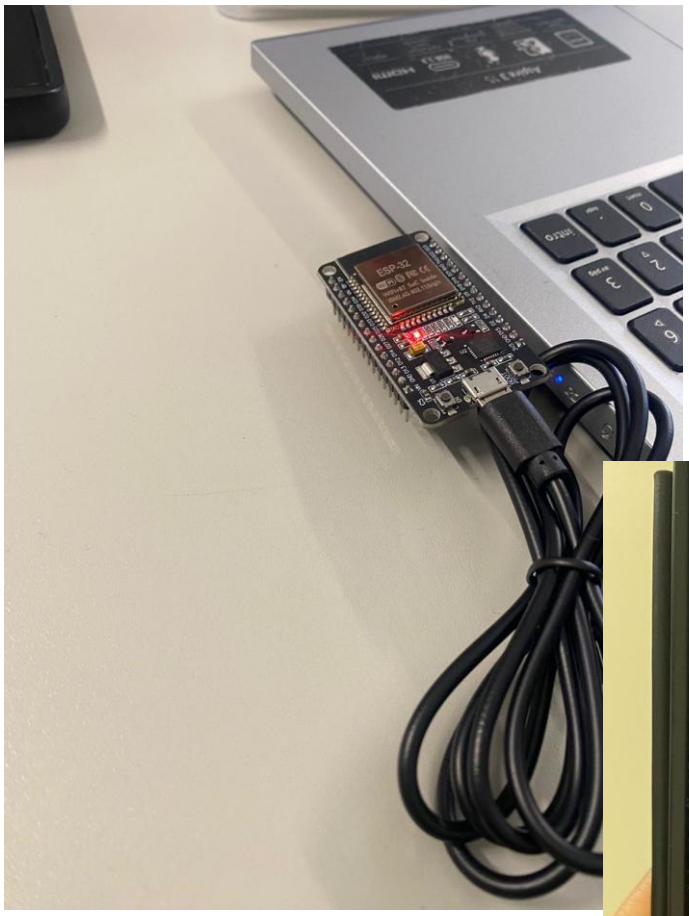
## Código de Arduino

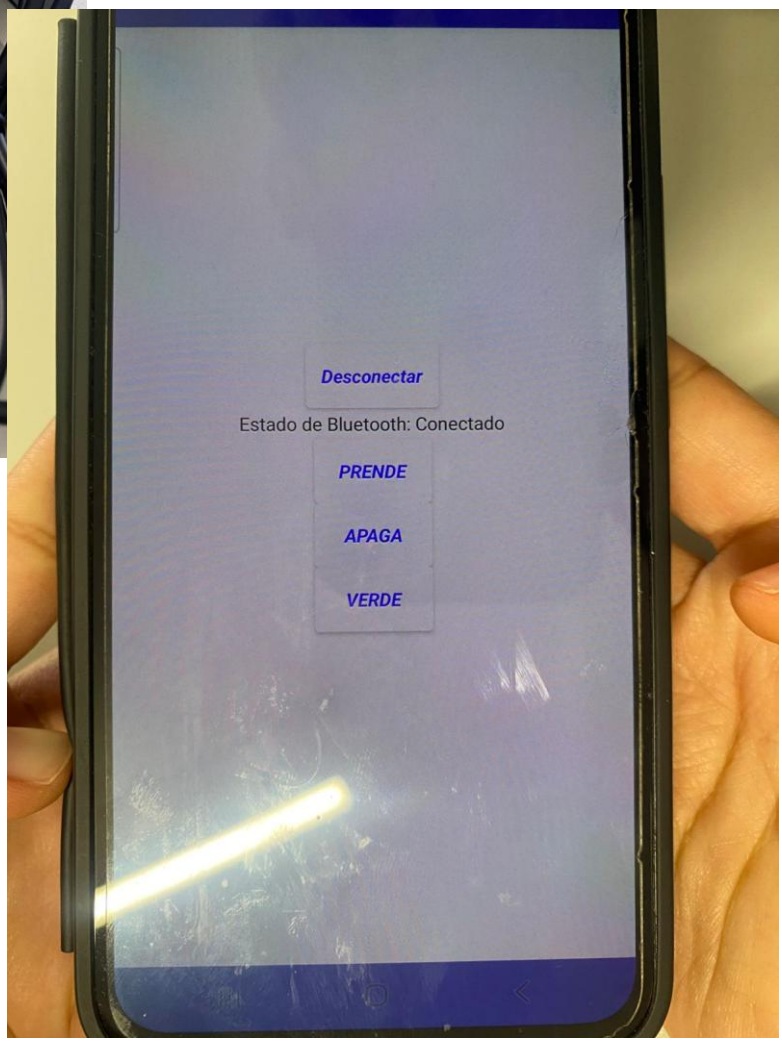


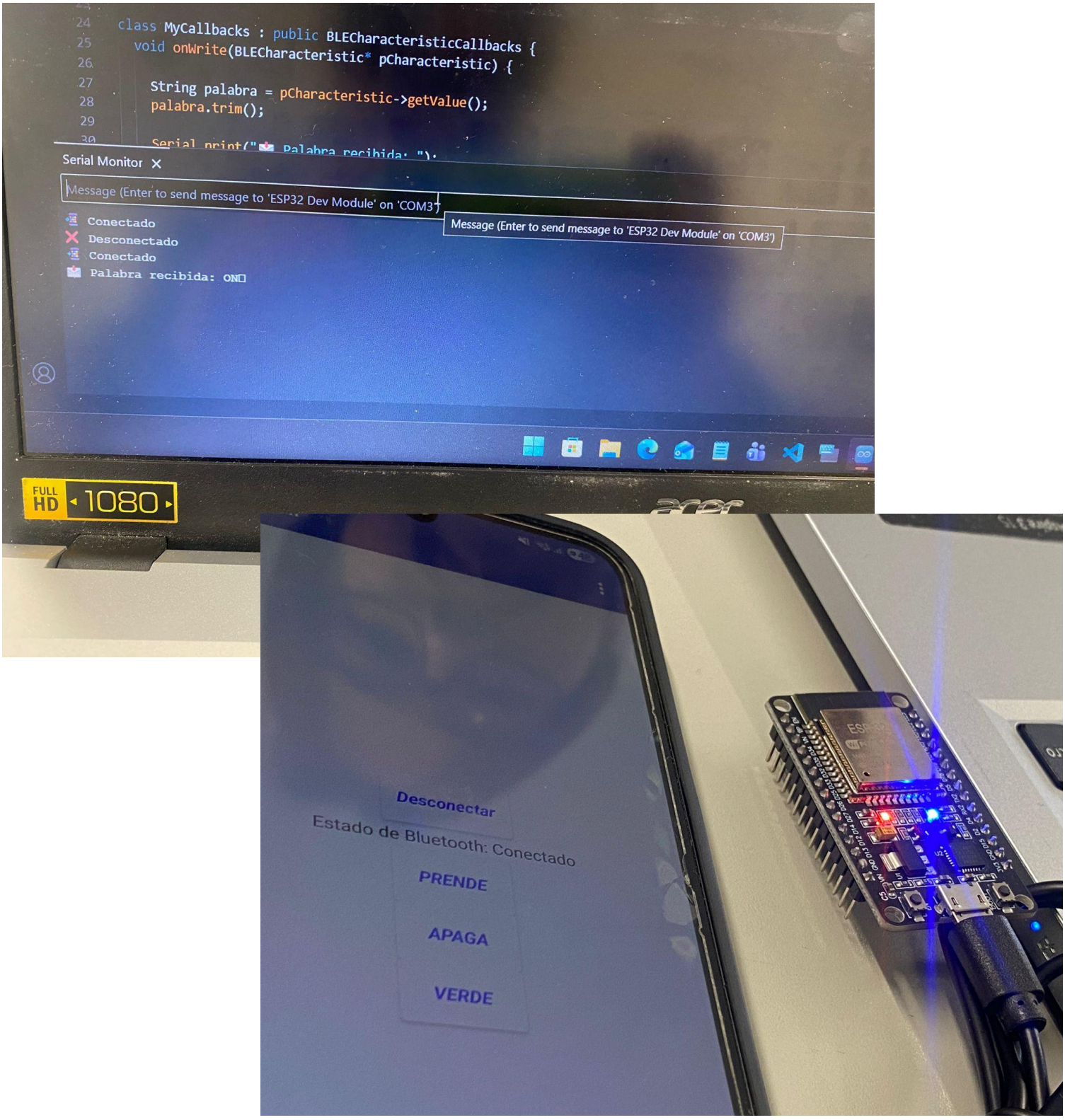

AppTecInalambrica | Arduino IDE 2.3.7
File Edit Sketch Tools Help
ESP32 Dev Module
AppTecInalambrica.ino
if (palabra == "ON") {
digitalWrite(LED_PIN, HIGH);
}
else if (palabra == "OFF") {
digitalWrite(LED_PIN, LOW);
}
}
};
void setup() {
Serial.begin(115200);
pinMode(LED_PIN, OUTPUT);
BLEDevice::init("ESP32_BLE");
BLEServer *pServer = BLEDevice::createServer();
pServer->setCallbacks(new MyServerCallbacks());
BLEService *pService = pServer->createService(SERVICE_UUID);
BLECharacteristic *pCharacteristic = pService->createCharacteristic(
CHARACTERISTIC_UUID,
BLECharacteristic::PROPERTY_WRITE
);
pCharacteristic->setCallbacks(new MyCallbacks());
pService->start();
BLEDevice::startAdvertising();
Serial.println(" ESP32 listo");
}
Serial Monitor
Output


void loop() {
}
Serial Monitor
Output

## Fotos de la practica






class MyCallbacks : public BLECharacteristicCallbacks {
void onWrite(BLECharacteristic* pCharacteristic) {
String palabra = pCharacteristic->getValue();
palabra.trim();
Serial Monitor
Message (Enter to send message to 'ESP32 Dev Module' on 'COM3')
Conectado
Desconectado
Conectado
Palabra recibida: ON
FULL HD 1080
Desconectar
Estado de Bluetooth: Conectado
PRENDE
APAGA
VERDE

## Conclusiones

**Manuel Humberto Reyes Salas**

En esta práctica pude entender de manera más clara cómo funciona la comunicación inalámbrica mediante Bluetooth Low Energy y cómo se puede aplicar en un proyecto real. Me ayudó a comprender la importancia del ESP32 como herramienta para desarrollar sistemas IoT, ya que permite conectar dispositivos de forma sencilla y eficiente.

También considero que fue muy útil trabajar tanto la parte de programación en Arduino IDE como el diseño de la aplicación en App Inventor, porque pude ver cómo se complementan el hardware y el software para lograr que un sistema funcioné correctamente. Finalmente, esta práctica me permitió reforzar mis conocimientos y darme cuenta de cómo tecnologías como BLE pueden utilizarse en aplicaciones reales de control y automatización.

**María José Guerrero Rios**

En esta práctica logramos aplicar los conocimientos sobre comunicación inalámbrica utilizando BLE, estableciendo una conexión exitosa entre una aplicación móvil y la placa ESP32. A través de este proceso entendimos cómo enviar y recibir datos de forma inalámbrica para controlar un dispositivo electrónico.

Además, reforzamos habilidades tanto en la programación del ESP32 desde Arduino IDE como en el desarrollo de aplicaciones móviles con programación por bloques en App Inventor.

**Jesús Eli Sánchez Ruvalcaba**

Gracias a esta practica pudimos comprender un nuevo mundo en microprocesadores, pudimos aplicar técnicas inalámbricas las cuales nos dan muy buenos fundamentos para futuros proyectos, en este caso pudimos realizar una conexión de BLE desde un teléfono hasta el ESP32 para prender un led de la misma placa.

Al realizar esta practica no tuvimos mucha complicación, ya que la manera de codificar es similar a Arduino que lo trabajamos en el semestre anterior, y la aplicación con la que se conecta del teléfono a la placa fue fácil de realizar.