Para realizar la práctica 2, donde se configura el **ESP32 como un Punto de Acceso (AP)**, te voy a explicar paso a paso cómo implementarlo. A continuación te detallo el código necesario, con las explicaciones correspondientes para que entiendas cada parte del proceso.

**Código Explicado Paso a Paso:**

**1. Incluir la librería WiFi.h:**

* Como en la práctica anterior, esta librería te permitirá manejar la conexión Wi-Fi, pero en este caso usaremos el modo AP (Access Point).

cpp

Copiar código

#include <WiFi.h>

**2. Configurar las credenciales de la red:**

* En este paso, defines el **SSID** (nombre de la red) y la **contraseña** que creará el ESP32.

cpp

Copiar código

const char\* ssid = "MiRedESP32"; // Nombre de la red Wi-Fi

const char\* password = "12345678"; // Contraseña de la red

**Recomendación**: El SSID puede ser cualquier nombre que desees, y asegúrate de que la contraseña tenga al menos 8 caracteres para evitar errores.

**3. Configurar el ESP32 como Punto de Acceso:**

* En el **setup()**, configuras el ESP32 para que funcione en modo **Access Point** (AP) utilizando el método **WiFi.softAP()**.

cpp

Copiar código

void setup() {

Serial.begin(115200); // Inicializa el puerto serie para monitorización

// Configura el ESP32 en modo AP con SSID y contraseña

WiFi.softAP(ssid, password);

// Imprime la dirección IP del ESP32

Serial.println("Punto de Acceso Iniciado.");

Serial.print("Dirección IP del AP: ");

Serial.println(WiFi.softAPIP());

}

* **WiFi.softAP(ssid, password)**: Este método inicia el punto de acceso con el nombre (SSID) y contraseña que definiste.
* **WiFi.softAPIP()**: Obtiene y muestra la dirección IP local del ESP32 cuando está en modo AP.

**4. Monitorear las conexiones de los dispositivos:**

* Es importante que puedas ver cuántos dispositivos están conectados a la red Wi-Fi del ESP32. Para ello, puedes usar el método **WiFi.softAPgetStationNum()** en un bucle.

cpp

Copiar código

void loop() {

// Monitorea cuántos dispositivos están conectados al AP

int numDispositivosConectados = WiFi.softAPgetStationNum();

Serial.print("Dispositivos conectados: ");

Serial.println(numDispositivosConectados);

delay(5000); // Actualiza cada 5 segundos

}

* **WiFi.softAPgetStationNum()**: Devuelve el número de dispositivos actualmente conectados al punto de acceso.

**Código completo:**

cpp

Copiar código

#include <WiFi.h>

const char\* ssid = "MiRedESP32"; // Nombre de la red Wi-Fi

const char\* password = "12345678"; // Contraseña de la red

void setup() {

Serial.begin(115200); // Inicializa el puerto serie para monitorización

// Configura el ESP32 en modo AP con SSID y contraseña

WiFi.softAP(ssid, password);

// Imprime la dirección IP del ESP32

Serial.println("Punto de Acceso Iniciado.");

Serial.print("Dirección IP del AP: ");

Serial.println(WiFi.softAPIP());

}

void loop() {

// Monitorea cuántos dispositivos están conectados al AP

int numDispositivosConectados = WiFi.softAPgetStationNum();

Serial.print("Dispositivos conectados: ");

Serial.println(numDispositivosConectados);

delay(5000); // Actualiza cada 5 segundos

}

**Explicación de los pasos:**

1. **Configuración del Modo AP**: El ESP32 se inicializa como punto de acceso con un SSID y una contraseña.
2. **Iniciar el Punto de Acceso**: Una vez que el AP está activo, se imprime la dirección IP que el ESP32 ha recibido para su red.
3. **Monitorear Conexiones**: El ESP32 cuenta y muestra cuántos dispositivos están conectados al punto de acceso.
4. **Prueba de Funcionamiento**: Puedes conectarte a la red que crea el ESP32 desde tu smartphone o PC. Luego, en el monitor serie de VSCode, verás el número de dispositivos conectados actualizado cada 5 segundos.

**Recomendaciones Adicionales:**

* **Conexiones no autorizadas**: Usa una contraseña segura para evitar que personas no deseadas se conecten a tu AP.
* **Monitor Serie**: Utiliza el Monitor Serie en VSCode para verificar la cantidad de dispositivos conectados y la dirección IP del AP.

Este código te permitirá configurar el ESP32 como un punto de acceso y monitorear las conexiones de los dispositivos en tiempo real.