**Explicación del Funcionamiento de los Códigos**

El proyecto utiliza dos tipos de configuraciones con ESP32: **puntos de acceso (AP)** y un **cliente Wi-Fi**. A continuación, se detalla el funcionamiento de cada uno:

**Código 1: Configuración del ESP32 como Punto de Acceso (AP)**

Este código transforma un ESP32 en un punto de acceso Wi-Fi independiente con un SSID y contraseña definidos. Los pasos principales son:

1. **Configuración del AP:**
   * Se utiliza la función WiFi.softAP(ssid, password) para habilitar el modo AP. Esto convierte al ESP32 en un emisor Wi-Fi que otros dispositivos pueden detectar.
   * Cada ESP32 tiene un SSID único (como ESP32\_Miguel) para diferenciarse y permitir la identificación de su ubicación física.
2. **Comportamiento:**
   * El ESP32 emite señales constantemente, lo que permite que otros dispositivos midan su intensidad (RSSI).
   * La IP asignada al AP es impresa en el puerto serie para monitorear su configuración y estado.
3. **Escenario de Uso:**
   * Este AP se posiciona estratégicamente en un espacio físico (por ejemplo, cocina, sala, etc.) para que un cliente pueda determinar su ubicación en relación con este.

**Código 2: Configuración del ESP32 como Cliente Wi-Fi**

Este código conecta un ESP32 a una red Wi-Fi existente y lo utiliza para escanear redes cercanas, calcular distancias a los AP configurados y enviar datos a ThingSpeak.

1. **Escaneo de Redes:**
   * La función WiFi.scanNetworks() detecta todas las redes disponibles en el área.
   * A través de WiFi.RSSI() se obtiene la intensidad de la señal (RSSI) de cada SSID registrado.
2. **Cálculo de Distancias:**
   * El código utiliza una fórmula basada en el modelo de propagación de señales:

Donde:

* + - PTx: Potencia de transmisión (dBm).
    - RSSI: Intensidad de la señal recibida.
    - n: Factor ambiental (típicamente 2-4 para interiores).
  + A partir de estas distancias, se determina cuál AP está más cercano, asignando una "zona" específica.

1. **Conexión a ThingSpeak:**
   * Los datos calculados son enviados a ThingSpeak mediante un cliente HTTP. La API de ThingSpeak requiere una clave única (WriteAPIKey) para autenticar los datos del proyecto.
   * Si la conexión falla, el código implementa reintentos para asegurar la transmisión.

En el **Código 2**, se retorna un número en cada ubicación detectada, lo cual representa una **"zona"** específica

**Relación de Números y Ubicaciones**

En la función asignarZona, los números están asignados a las ubicaciones como sigue:

* **1:** Representa la **Cocina** (AP con SSID ESP32\_Cocina).
* **2:** Representa la **Sala** (AP con SSID ESP32\_Sala).
* **3:** Representa la ubicación de **Miguel** (AP con SSID ESP32\_Miguel).
* **4:** Representa el **3er Piso** (AP con SSID ESP32\_3Ro).
* **6:** Valor predeterminado si no se detecta ninguna red (o todas las distancias son demasiado altas).

**Razón para Retornar un Número**

1. **Identificación Simplificada:**
   * Los números representan de forma simplificada cada una de las ubicaciones o zonas posibles (por ejemplo, cocina, sala, etc.).
   * Esto es práctico para enviar datos compactos a ThingSpeak y facilita el procesamiento o visualización posterior.

**Algunas imágenes:**

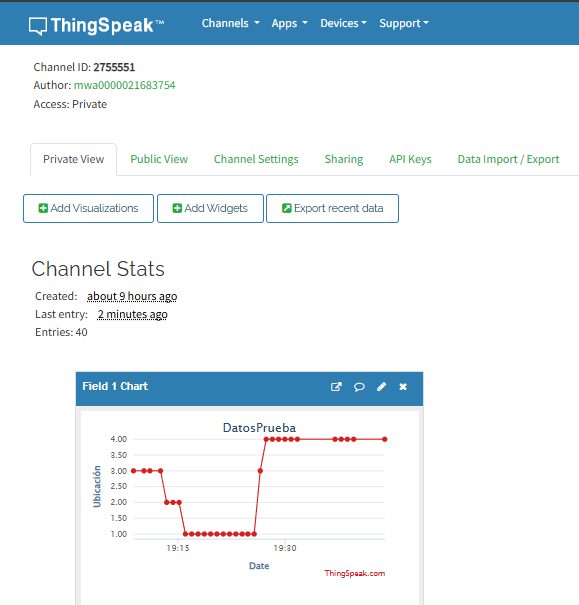
****

Ilustración Resultado de la subida de datos



Ilustración Dispositivos usados



Ilustración Ejemplo de distribucion de dispositivos

**Escalabilidad del Proyecto**

Para escalar el proyecto a entornos más complejos o aplicaciones más avanzadas, se pueden considerar las siguientes estrategias:

1. **Ampliación de la Red de Puntos de Acceso:**
   * Incrementar el número de ESP32 configurados como AP para cubrir áreas más grandes.
   * Utilizar un identificador único para cada AP, almacenando su ubicación en una base de datos centralizada.
2. **Optimización del Algoritmo de Localización:**
   * Implementar algoritmos más precisos como multilateración, utilizando las distancias calculadas a múltiples AP para obtener una ubicación en coordenadas (x, y).
   * Ajustar el factor ambiental (nnn) según las características físicas del entorno.
3. **Integración de Servicios Adicionales:**
   * Conectar la solución a plataformas como Google Maps para visualizar la ubicación en un mapa en tiempo real.
   * Incorporar sensores adicionales (como GPS) para mejorar la precisión en exteriores.
4. **Mejora de la Infraestructura de Datos:**
   * Configurar una base de datos en la nube para almacenar los datos históricos de localización y generar análisis detallados.
   * Utilizar plataformas más avanzadas que ThingSpeak, como AWS IoT o Azure IoT Hub, para manejar grandes volúmenes de datos.

**Temáticas Adicionales Relevantes**

1. **Impacto del Entorno en la Precisión:**
   * Factores como paredes, muebles y dispositivos electrónicos pueden interferir con las señales Wi-Fi, alterando los valores de RSSI y, por ende, las distancias calculadas.
   * Es crucial realizar pruebas en el entorno real para calibrar adecuadamente el sistema.
2. **Consumo de Energía:**
   * El modo AP y el escaneo continuo de redes consumen una cantidad significativa de energía. Para aplicaciones en dispositivos móviles, se debe optimizar el consumo energético o integrar baterías de mayor capacidad.
3. **Seguridad de la Información:**
   * Asegurar la transmisión de datos mediante cifrado (por ejemplo, HTTPS para la conexión con ThingSpeak).
   * Implementar mecanismos de autenticación más robustos si el sistema es escalado para su uso en producción.

**Conclusión**

Este proyecto aprovecha las capacidades del ESP32 para implementar una solución de localización basada en Wi-Fi. Con el diseño actual, es ideal para aplicaciones en interiores donde los GPS son menos efectivos. La escalabilidad y la precisión pueden mejorarse mediante ajustes en el algoritmo, expansión de la red de APs y una infraestructura de datos más robusta. La integración con servicios avanzados también abriría nuevas posibilidades, como el análisis en tiempo real y la generación de reportes históricos.