

Esta es la documentación técnica oficial del **Sistema Táctico de Monitoreo WiFi (Mesh-Sniffer)**. Está diseñada para que un equipo técnico pueda entender la arquitectura, replicar el hardware y continuar con el desarrollo del software.

## 1. Visión General del Proyecto

El sistema es una red de sensores inalámbricos diseñada para detectar la presencia de dispositivos móviles en un área geográfica (San José de Mayo) mediante la captura de paquetes **Probe Request**.

### Finalidad

- Mapeo de Densidad:** Identificar zonas de mayor concurrencia.
- Rastreo No Invasivo:** Utiliza las señales que los teléfonos emiten buscando redes WiFi, sin necesidad de conectarse a ellos.
- Infraestructura Resiliente:** Los nodos se comunican entre sí formando una malla (Mesh), permitiendo cubrir áreas donde no hay cobertura WiFi centralizada.

## 2. Arquitectura del Sistema

El sistema se divide en tres capas jerárquicas:

### A. Nodos Esclavos (Sensores)

Son dispositivos **Wemos D1 Mini (ESP8266)** alimentados por baterías 18650.

- Función:** Alternan entre dos modos de operación.
  - Modo Sniffer:** Capturan direcciones MAC y potencia de señal (RSSI) del aire.
  - Modo Malla (Mesh):** Se conectan a la red interna para enviar los datos recolectados al Maestro.

### B. Nodo Maestro (Gateway)

Un ESP8266 conectado permanentemente a una PC vía USB.

- Función:** Actúa como el concentrador de la red. Recibe los paquetes de todos los esclavos y los vuelca al puerto serie en formato CSV.

### C. Dashboard (Interfaz de Usuario)

Una aplicación web alojada en Vercel que utiliza la **Web Serial API**.

- Función:** Visualiza en tiempo real las detecciones, identifica fabricantes y permite exportar reportes.

## 3. Implementación del Software

### Lógica del Esclavo: "Hard-Switching" de Radio

El mayor desafío técnico resuelto es la alternancia de la radio del ESP8266. Un solo chip no puede "escanear" el aire y "estar conectado" a la red Mesh simultáneamente de forma estable.

### Flujo de ejecución:

- Captura (20s):** La radio entra en modo promiscuo y salta por los 13 canales WiFi.
- Reinicio:** Se detiene el sniffer y se limpia la pila TCP/IP.
- Transmisión:** Se inicia `painlessMesh`, se busca al Maestro y se envía la ráfaga de datos.

4. **Confirmación:** El nodo no vuelve a capturar hasta que la red Mesh confirma la entrega ( `sendBroadcast` ).

### Lógica del Maestro: Concentrador Pasivo

El Maestro es un nodo "siempre disponible". No realiza sniffing para no perder paquetes de los esclavos. Su único trabajo es ejecutar el `receivedCallback` y enviar el String recibido directamente al `Serial.println()` .

## 4. Guía de Hardware

Componente	Función
<b>Wemos D1 Mini</b>	Microcontrolador principal (ESP8266).
<b>Batería 18650</b>	Autonomía para los nodos esclavos.
<b>TP4056</b>	Módulo de carga para la batería.
<b>Antena (Opcional)</b>	Algunos modelos permiten antena externa para mayor alcance.

## 5. Especificaciones del Protocolo de Datos

Los datos viajan por la red Mesh y el puerto serie siguiendo este estándar: `NODO_ID`, `MAC`, `RSSI`, `FINGERPRINT`

- **NODO\_ID:** Identificador único (ej: `N-001` ).
- **MAC:** Dirección física del dispositivo detectado.
- **RSSI:** Nivel de señal (0 a -100).
- **FINGERPRINT:** Hash único basado en los parámetros de la trama para evitar duplicados.

## 6. Próximos Pasos para el Desarrollo

Para el equipo que retome este proyecto, se sugieren las siguientes mejoras:

1. **Persistencia en Nube:** Implementar un endpoint API en Vercel para guardar los datos en una base de datos (PostgreSQL/Redis) y no perderlos al cerrar el navegador.
2. **Georreferenciación:** Vincular cada `NODO_ID` a coordenadas GPS fijas en el `mapa.html` .
3. **Optimización de Batería:** Implementar `Deep Sleep` en los esclavos después de cada transmisión exitosa para extender la duración de la batería de días a meses.

**Estado Actual:** El sistema es funcional en modo local. Los nodos logran conmutar sus radios y el Maestro entrega datos limpios al Dashboard.