

Wilobu - Sistema IoT de Emergencia

Sistema completo de botón de emergencia IoT con monitoreo en tiempo real, notificaciones push y vinculación por Bluetooth.

🚀 Inicio Rápido para Evaluación

1. Probar la App Móvil (Recomendado)

Requisitos

- Android device/emulator (minSdk 24) o iOS device/simulator (minTarget 11.0)
- Flutter 3.10+
- Cuenta Firebase configurada (incluida en el proyecto)

Ejecutar

```
cd wilobu_app  
flutter pub get  
flutter run
```

Credenciales de prueba:

- Email: test@wilobu.com
- Password: Test1234!

2. Funcionalidades Principales

📱 App Móvil

1. **Registro/Login:** Firebase Authentication
2. **Vincular Dispositivo:**
 - Presionar botón SOS en hardware 5 segundos
 - Escanear dispositivo BLE "Wilobu-XXXXXX"
 - Vinculación automática
3. **Enviar Alerta SOS:** Presionar botón SOS 3 segundos
4. **Ver Ubicación:** Mapa en tiempo real con OpenStreetMap
5. **Gestionar Contactos:** Agregar contactos de emergencia

🔧 Hardware (Opcional)

```
cd wilobu_firmware  
python -m platformio run --target upload
```

Hardware: ESP32 + A7670SA modem

Pines: Definidos en `src/main.cpp`

3. Backend (Pre-configurado)

Firebase

- **Proyecto:** `wilobu-d21b2`
- **Firestore:** Reglas en `firebase.rules`
- **Functions:** Node.js functions en `functions/`

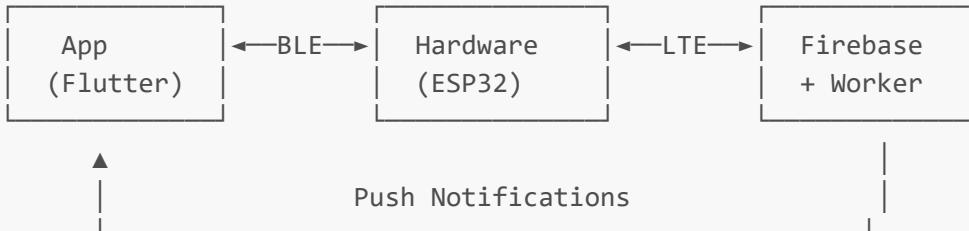
Para re-deployar:

```
firebase login  
firebase deploy --only firestore:rules,functions
```

Cloudflare Worker (Proxy HTTPS)

```
cd cloudflare-worker  
wrangler deploy
```

📋 Arquitectura



Stack Tecnológico

- **Frontend:** Flutter + Riverpod + GoRouter
- **Backend:** Firebase (Auth, Firestore, Functions)
- **Hardware:** ESP32 + NimBLE + A7670SA modem
- **Infraestructura:** Cloudflare Worker (proxy HTTPS)

⌚ Flujos de Uso

Vinculación de Dispositivo

1. Usuario crea cuenta en app
2. Presiona botón SOS en hardware por 5 segundos

3. App escanea BLE y encuentra "Wilobu-XXXXXX"
4. Vinculación automática (ownerUid enviado por BLE)
5. Dispositivo aparece en app con status online

Alerta SOS

1. Usuario presiona botón SOS en hardware 3 segundos
2. Dispositivo envía GPS + tipo de alerta a Firebase
3. Cloud Function notifica contactos de emergencia vía FCM
4. Contactos reciben push con ubicación y mapa

📁 Estructura del Proyecto

```
wilobu/
├── wilobu_app/          # App Flutter
│   ├── lib/features/     # Features (auth, devices, alerts, profile)
│   ├── lib/ble/           # Servicio BLE
│   └── lib/theme/         # Tema UI
├── wilobu_firmware/      # Firmware ESP32
│   └── src/               # main.cpp, ModemProxy, ModemHTTPS
├── functions/            # Cloud Functions
│   └── index.js           # heartbeat, SOS handler
├── cloudflare-worker/    # Worker proxy
│   └── worker.js
└── firestore.rules        # Reglas de seguridad
└── README.md
```

📝 Testing

Casos de Prueba Sugeridos

1. Registro de usuario nuevo
2. Vinculación de dispositivo por BLE
3. Envío de alerta SOS (General/Médica/Seguridad)
4. Visualización de ubicación en mapa
5. Agregar contacto de emergencia
6. Recepción de notificaciones push
7. Desvincular dispositivo

Usuario de Prueba

Ya existe en Firebase con dispositivo vinculado:

- **Email:** test@wilobu.com
- **Password:** Test1234!
- **Dispositivo:** 781C3CB994FC

🔑 Configuración (Solo si necesitas cambiar)

Firebase

- Proyecto ID: wilobu-d21b2
- Credenciales: wilobu_app/android/app/google-services.json

Cloudflare Worker

- Account ID y API Token en cloudflare-worker/wrangler.toml
- Secrets: FIREBASE_PROJECT_ID, FIREBASE_CLIENT_EMAIL, FIREBASE_PRIVATE_KEY

🔧 Troubleshooting

Problema	Solución
App no compila	<code>flutter clean && flutter pub get</code>
BLE no conecta	Verificar permisos Bluetooth y Location
Alertas no llegan	Verificar permisos de notificaciones
Firmware no flashea	Cerrar monitor serial (Ctrl+C)
410 en heartbeat	Verificar que documento existe en Firestore

📄 Licencia

Propietario - Todos los derechos reservados

Versión: 2.0

Última actualización: 8 de Diciembre, 2025

Estado: Producción