

# WILOBU - SOS Alert System

Sistema wearable de emergencia con GPS + notificaciones automáticas

## ⚡ Características

- **Botón SOS** con geolocalización automática
- **Notificaciones Push** a contactos de emergencia
- **Dispositivo Autónomo** (LTE + GPS)
- **App Móvil** para monitoreo en tiempo real
- **Seguridad** HTTPS/TLS end-to-end

## 📁 Componentes

wilobu_app/	→ App Flutter (iOS/Android) ★ TESTEA ESTO
wilobu_firmware/	→ Firmware C++ (ESP32)
functions/	→ Cloud Functions (FCM Notifications)
cloudflare-worker/	→ Security Proxy

## ⚡ Inicio Rápido (PARA EVALUAR EN MÓVIL)

### Requisitos Mínimos

- Flutter 3.38+
- Dispositivo Android/iOS conectado (o emulador)
- Conexión a Internet (para Firebase)

## ⚡ Opción 1: Automático (Recomendado)

### Windows:

```
start_app.bat
```

### macOS/Linux:

```
bash start_app.sh
```

## ⚡ Opción 2: Precompilación + Ejecución

### Windows:



```
wilobu_app\precompile.bat
```

#### macOS/Linux:

```
bash wilobu_app/precompile.sh
```

Luego ejecuta:

```
flutter run
```

#### ⚡ Opción 3: Manual Paso a Paso

```
cd wilobu_app

# Limpiar proyecto (opcional pero recomendado)
flutter clean

# Instalar dependencias
flutter pub get

# Ejecutar en dispositivo
flutter run

# Para ver logs detallados:
flutter run -v
```

#### ⌚ Qué Esperar

##### 1. Primera pantalla: Login

- Email: Cualquier correo (ej: test@example.com)
- Contraseña: Cualquier contraseña (sin validación)
- Botón: "Conectar"

##### 2. Dashboard: Lista de dispositivos

- Botón flotante "+" para agregar dispositivo
- Botón "Contactos" en la esquina superior
- Selector de tema (Claro/Oscuro)

##### 3. Flujo Completo:

- Login → Dashboard → Agregar Dispositivo (BLE) → Manage Contacts → SOS Alert

# 🔧 Detalles Técnicos

## App Tech Stack:

- Flutter 3.38+
- Riverpod (State Management)
- Firebase Auth + Firestore
- GoRouter (Navigation)

## Firmware:

- ESP32 + PlatformIO
- Soporta 3 hardware variants
- Máquina de estados (7 estados)
- GPS + LTE + BLE

## Cloud:

- Cloud Functions (FCM Multicast)
- Cloudflare Worker (Security Proxy)
- Firestore (Real-time Database)

## 📊 Flujo SOS

```
Usuario presiona botón
↓
GPS obtiene ubicación
↓
Envía a Firebase (LTE)
↓
Cloud Function dispara
↓
Busca contactos de emergencia
↓
FCM multicast a contactos
↓
Contacto recibe notificación + mapa
```

## ☑ Testing Checklist

- App inicia sin errores
- Login funciona
- Dashboard muestra estado
- Puedo agregar un dispositivo
- Gestión de contactos funciona
- Notificaciones se reciben (con Cloud Functions)

## 🔗 Pruebas Automatizadas

- `flutter test test/features/auth/login_flow_test.dart` → valida que el formulario de login autentica, crea el perfil en Firestore falso y redirige al dashboard.

## ⌚ Código Minimalista

- Sin documentación innecesaria
- Sin comentarios excesivos
- Máximo 3000 líneas totales en Flutter
- Máximo 500 líneas en Cloud Functions
- Máximo 280 líneas en Cloudflare Worker

## 📞 Soporte

Consulta el código comentado en:

- `wilobu_app/lib/main.dart` - Punto de entrada
  - `wilobu_app/lib/features/auth/` - Autenticación
  - `wilobu_app/lib/features/home/` - Dashboard
  - `functions/index.js` - Notificaciones FCM
  - `cloudflare-worker/worker.js` - Proxy seguro
- 

**Autor:** Osvaldo Díaz

**Estado:**  Funcional y Listo para Evaluar

**v2.0**