

## Backend para AURA Sentinel

### Decisión estratégica: Backend híbrido, open-source y modular

El enfoque más competitivo y sostenible para AURA Sentinel es adoptar una arquitectura backend basada en software libre, modular y enfocado a "offline-first", para cumplir con las etapas del concurso y asegurar migración y expansión sin fricciones.

#### 1. Corto plazo (Concurso/MVP)

Tecnología recomendada: Appwrite

- Ventajas:
  - Plataforma open-source, fácilmente auto-hospedable o desplegable en nubes públicas/privadas.
  - Integración nativa con Flutter y SDKs robustos para mobile.
  - Módulos de autenticación, base de datos, funciones serverless y almacenamiento integrado.
  - Sincronización básica offline-local y control granular de datos y usuarios.
  - Cumplimiento sencillo de la LFPDPPP: puedes decidir la localización, resguardo y ciclo de vida de los datos desde el inicio.
- Motivo estratégico:
  - Permite iterar rápidamente el MVP, mantener bajo costo y controlar completamente los datos sensibles de usuarios, pieza esencial para competencia y confianza.

#### 2. Mediano plazo (Piloto real / primeros 1,000-10,000 usuarios)

Estrategia:

- Migración transparente a infraestructura más robusta:
  - Escalamiento a servidores dedicados (cloud propia, servidores nacionales) o proveedores como Back4App.

- Conservas lógica de frontend Flutter y migras backend sin reescribir ni perder lógica/módulos previos.
- Facilita integración de APIs avanzadas para IA (e.g., Grok/Claude) con funciones tipo proxy o microservicio, elevando capacidades analíticas conforme crece el uso real.

### 3. Largo plazo (escala nacional/global)

Estrategia:

- Posibilidad de migración a stacks enterprise (AWS, GCP, Azure) o adopción de capas avanzadas tipo Couchbase/Supabase para sincronización distribuida.
- Arquitectura modular desde inicio: soporta DevOps, integración continua, plugins de IA y paneles administrativos para actores públicos (911, protección civil).
- El stack nunca te ata a un proveedor específico y permite implementar nuevas normativas, métodos de cifrado o analítica avanzada sin rehacer la app mobile.

### Elección por fases – Resumen ilustrativo

Fase	Backend sugerido	Motivo	Escalabilidad
MVP/Concurso	Appwrite	Control, facilidad, privacidad, costo cero	Migrable
Piloto real	Appwrite/Back4App	Escala, soporte, integración APIs externas	Totalmente libre
Nacional	AWS, Couchbase, etc.	Alta disponibilidad, multi-región	Modular

## Justificación y links de referencia

- No te “encadenas” a proveedores cerrados como Firebase.
- Cumples legislación y requisitos técnicos desde el día 1 (LFPDPPP).
- Puedes empezar chico y crecer rápido, sin refactorizaciones masivas.
- Infraestructura lista para ambientes DevOps, migración y paneles IA.

Recursos:

- [appwrite.io - Comparativa BaaS 2025](#)
- [Alternativas a Firebase \(Back4App\)](#)

Conclusión:

Esta planeación te da flexibilidad, control, cumplimiento legal y velocidad, y prepara tu proyecto para competir, impactar y escalar profesionalmente, alineada al perfil y objetivos de AURA Sentinel.

## Diseño de Arquitectura, APIs y Módulos Backend – AURA Sentinel

### 1. Enfoque Estratégico

El backend de **AURA Sentinel** se diseña bajo una **arquitectura híbrida, modular y open-source**, alineada con la filosofía *offline-first* y los principios de seguridad, escalabilidad y soberanía de datos.

El objetivo es garantizar la continuidad operativa en entornos de baja conectividad, permitir una migración sin fricciones y cumplir con las normativas de protección de datos desde las primeras etapas del proyecto.

#### Decisión estratégica

- **Tipo:** Backend híbrido modular, autosuficiente y portable.
- **Principios:** Control total, escalabilidad por fases y cumplimiento legal (LFPDPPP).
- **Ventaja clave:** Crecimiento progresivo sin necesidad de refactorización masiva o dependencia de proveedores cerrados.

## 2. Arquitectura General

### 2.1. Vista por capas

La arquitectura propuesta se compone de cuatro capas principales:

#### 1. Capa de Dispositivo (Frontend/Edge)

- Aplicación móvil desarrollada en **Flutter**, con integración directa a los modelos **TFLite**.
- Procesamiento local de IA: clasificación de emergencias, guía emocional, rutas seguras, y activación por voz.
- Sincronización diferida con backend cuando la conexión es estable.

#### 2. Capa de Procesamiento Backend (Appwrite Stack)

- Núcleo de servicios alojado en Appwrite o infraestructura equivalente.
- Módulos principales:
  - Autenticación y control de acceso (JWT / OAuth).
  - Base de datos (Appwrite DB o MongoDB).
  - Funciones *serverless* (Node.js / Python).
  - Almacenamiento de mapas, fichas y logs de emergencia.
  - API REST/GraphQL nativa.
- Totalmente auto-hospedable y con control de localización de datos.

#### 3. Capa Híbrida de Integración

- Conexión con APIs externas y servicios avanzados:
  - **IA Proxy** (Grok, Claude) para guías médicas complejas.
  - **Servicios de emergencia (911 / Protección Civil)**.
  - **Pasarelas SMS** (Twilio, etc.) para comunicación redundante.
- Microservicios para sincronización offline-cloud, notificaciones y registro de eventos.

#### 4. Capa de Datos y Escalabilidad

- Infraestructura de datos distribuida con tecnologías como **Couchbase**, **Supabase** o equivalentes.
- Registro y monitoreo mediante **Grafana**, **Loki** y **Prometheus**.
- Panel administrativo *Sentinel* para autoridades y personal operativo.
- Integración DevOps / CI-CD mediante contenedores **Docker** o **Kubernetes**.

### 3. Módulos Backend Principales

Módulo	Tecnología base	Función principal	Fase recomendada
<b>API Gateway</b>	Appwrite Functions / Nginx	Enrutamiento de peticiones y control de acceso	Todas
<b>Auth / Users</b>	Appwrite Auth	Gestión de identidades, roles y permisos	MVP
<b>Database Core</b>	Appwrite DB / MongoDB / Supabase	Almacenamiento de fichas médicas, logs y mapas	MVP → Nacional
<b>Serverless Functions</b>	Node.js / Python	Ejecución de lógica IA y microservicios	MVP
<b>Sync Manager</b>	Custom Appwrite Function	Sincronización offline → nube	Piloto
<b>Emergency Notifier</b>	Twilio / API 911	Notificaciones automáticas y alertas SMS/voz	Piloto → Nacional
<b>IA Proxy Service</b>	Python (FastAPI)	Enlace con Grok / Claude para guía avanzada	Piloto
<b>Monitoring / Logging</b>	Grafana / Prometheus / Loki	Registro de métricas y eventos	Nacional
<b>Admin Panel Sentinel</b>	Flutter Web / React	Panel de control para instituciones	Nacional

#### 4. Diseño de APIs



El backend expone un conjunto de APIs RESTful, diseñadas bajo principios de interoperabilidad, versionamiento y simplicidad.

#### 4.1. Nivel interno (Appwrite / Core)

- [/account/login](#) – Autenticación de usuario.
- [/account/register](#) – Creación de cuenta.
- [/database/{collection}/list](#) – Consulta de datos.
- [/functions/{functionId}/execute](#) – Ejecución de funciones *serverless*.

#### 4.2. Nivel de API AURA Sentinel

Endpoint	Método	Descripción
<a href="#">/emergency/report</a>	POST	Reporte de emergencia y prioridad.
<a href="#">/geo/refuges</a>	GET	Consulta de refugios cercanos.
<a href="#">/user/ficha</a>	GET / PUT	Consulta o actualización de ficha médica.
<a href="#">/alert/911</a>	POST	Envío de alerta a servicio de emergencia.
<a href="#">/sync</a>	POST	Sincronización de datos locales.
<a href="#">/ai/guide</a>	POST	Solicitud de guía médica avanzada (IA Proxy).

#### 4.3. Nivel de integraciones externas

- **IA Proxy (Grok, Claude)**: soporte cognitivo avanzado.
- **API 911 / Protección Civil**: envío autenticado de reportes.
- **Servicios de mensajería (Twilio / SMS Gateway)**: respaldo de comunicación.

### 5. Escalabilidad por Fases

Fase	Backend sugerido	Enfoque principal	Beneficio técnico
------	------------------	-------------------	-------------------



		AURA Sentinel	investigación backend
<b>Corto plazo (MVP/Concurso)</b>	<b>Appwrite</b>	Control, privacidad, bajo costo	Auto-hospedable, rápido para iterar
<b>Mediano plazo (Piloto real)</b>	<b>Appwrite + Back4App</b>	Integración de APIs externas y escalamiento moderado	Migración sin reescritura
<b>Largo plazo (Nacional / Global)</b>	<b>AWS / Couchbase / Supabase</b>	Alta disponibilidad y redundancia multi-región	Escalabilidad total y compatibilidad con IA avanzada

## 6. Seguridad y Cumplimiento

- Cumplimiento integral con la **Ley Federal de Protección de Datos Personales (LFPDPPP)**.
- Cifrado de datos en reposo (AES-256) y en tránsito (TLS 1.3).
- Autenticación segura mediante **JWT** y control granular de roles.
- Localización controlada de datos conforme a políticas nacionales.

## 7. Resumen Visual Conceptual

[ Integraciones IA + APIs 911 ]

[ Backend Modular Appwrite / Supabase ]

[ Serverless Functions + Sync Manager + Data Core ]

[ App Móvil Flutter + Modelos (AURA Core) ]

## 8. Conclusión

El diseño backend de **AURA Sentinel** ofrece una **arquitectura sólida, escalable y legalmente conforme**, enfocada en garantizar:



- Soberanía total sobre los datos de los usuarios.
- Operatividad constante incluso sin conexión.
- Integración flexible con servicios de inteligencia artificial y autoridades.
- Evolución natural hacia entornos nacionales o globales sin reescritura del sistema.

Este enfoque modular y open-source permite al proyecto **competir, escalar e impactar socialmente**, manteniendo la ética, la privacidad y la sostenibilidad tecnológica como pilares de la innovación.

Buffalos