

# Asistente geoespacial – Informe técnico

**Documento de apoyo:** herramientas, APIs, decisiones de diseño y limitaciones resueltas a nivel de implementación.

## 1) Herramientas utilizadas

- **Frontend:** Next.js (App Router), React, TypeScript, Tailwind CSS, shadcn/ui, lucide-react.
- **Mapa:** Mapbox (capas base, overlays raster y WMS).
- **Backend:** Route Handlers de Next.js (proxy de llamadas externas, normalización de respuestas y control de errores).
- **Build/Dev:** Turbopack en desarrollo y compilación de producción con chequeo estricto de TypeScript.

## 2) APIs consumidas

- **Geocodificación:** OpenStreetMap Nominatim (búsqueda de direcciones y retorno de coordenadas).
- **Contexto urbano:** Overpass API (POIs, viales y recuentos por categorías dentro de un radio).
- **Meteo:** OpenWeather (tiles de temperatura, precipitación, nubes y viento; y consulta puntual para mini-card e informe).
- **Riesgo Q100:** WMS oficial (SNCZI/IDEE) para cartografía de zona inundable estadística (T=100 años).
- **EFAS:** Copernicus EFAS (WMS) para capas de alerta temprana y lectura de capacidades (capas y tiempos disponibles).

### Diferencias entre Q100 y EFAS

- **Q100 (zona inundable estadística):** cartografía histórica/estadística por periodo de retorno; útil como línea base de planificación.
- **EFAS (alerta temprana):** modelización y previsión meteorológica-hidrológica a escala europea; orientado a vigilancia operativa y anticipación.
- **Uso combinado:** Q100 aporta contexto estructural del territorio; EFAS aporta señal temporal (situación prevista) para priorizar vigilancia y respuesta.

### 3) Decisiones de diseño

- **UI centrada en el mapa:** el mapa es el elemento principal; los controles se integran como overlays para no ocultar el contexto.
- **Separación por capas:** distinción clara entre mapa base, meteo, Q100 y EFAS, con control de opacidad para compatibilizar lectura y visibilidad.
- **Tolerancia a fallos:** si una fuente falla, el análisis continúa y se refleja como limitación, evitando bloquear el resto del flujo.
- **Trazabilidad:** el informe expone fuentes consultadas y limitaciones para facilitar auditoría y revisión.

### 5) Limitaciones y mitigaciones

- **Respuestas incompletas o no JSON:** se añadió validación de contenido y manejo robusto de errores para evitar rupturas en la interfaz cuando un proveedor devuelve HTML o respuestas parciales.
- **Campos opcionales en el contrato:** se incorporó normalización y valores por defecto para listas (fuentes/limitaciones) y para bloques de datos opcionales.
- **Dependencias de navegador en compilación:** el mapa se carga solo en cliente para evitar errores de renderizado en servidor durante el build.
- **WMS intermitente:** se implementó degradación controlada (marcar como limitación y continuar el análisis) cuando el servicio WMS responde con errores temporales.
- **Selección de capas EFAS:** se incorporó lectura de capacidades y una capa por defecto configurable para mantener un comportamiento consistente si la capa seleccionada no está disponible.