

- [Guía— Geovisualizador de Áreas de Interés](#)
  - [Objetivo del sistema](#)
  - [Arquitectura](#)
  - [Flujo principal](#)
  - [Archivos y responsabilidades](#)
  - [Tomar en cuenta](#)
  - [Carga de dependencias](#)
  - [Seguridad y privacidad](#)
  - [Demo: <https://areasdeinteres.pages.dev/>](#)

# Guía— Geovisualizador de Áreas de Interés

---

Este documento resume, de forma breve, cómo está implementado el código

## Objetivo del sistema

---

Herramienta que permite cargar un KML con un área de interés y localizar, visualizar y listar las localidades que intersectan dicha área.

## Arquitectura

---

- Tipo: Sitio estático (sin backend).
- UI: Bootstrap 5 (mobile-first), componentes en un acordeón.
- Mapas: Leaflet 1.9.x + OpenStreetMap.
- Geoespacial: Turf v6 (buffer, intersección, centroides).
- Conversión de datos: togeojson.js (KML → GeoJSON) en el navegador.
- Datos externos: Localidades GeoJSON (cargado bajo demanda desde CDN).
- Cache-busting: APP\_VERSION en index.html para invalidar recursos estáticos.

## Flujo principal

---

1. Usuario carga un archivo KML.

- Se parsea como XML y se convierte a GeoJSON con `toGeoJSON.kml()`.
- Se selecciona el primer polígono válido (Polygon/MultiPolygon) y se dibuja en el mapa.

## 2. Selección del tipo de área.

- Núcleo: se genera un buffer de 500 m alrededor del polígono (`Turf.buffer`).
- Influencia directa/indirecta: se usa el polígono tal cual.

## 3. Preparación de datos base.

- Si aún no se cargó, se descarga una sola vez el GeoJSON de localidades desde el CDN.

## 4. Análisis de intersección.

- Se recorre cada feature de localidades y se evalúa `T.booleanIntersects(localidad, áreaRecorte)`.
- El proceso se trocea con `requestAnimationFrame` para mantener la UI fluida y la barra de progreso activa (es un arreglo meramente visual)

## 5. Resultados y visualización.

- Se dibujan las features que intersectan (color único por CVEGEO) y se generan etiquetas/popup.
- Se listan los CVEGEO en panel lateral con navegación al mapa.
- Se encuadra el mapa al conjunto: KML/buffer  $\cup$  resultados (`fitBounds` con `padding`).

# Archivos y responsabilidades

---

- `index.html`: Estructura de UI, links a CDNs, control de versión (`APP_VERSION`), preloader, modal genérico, acordeón de controles.
- `index.js`: Lógica de aplicación (inicialización de Leaflet, carga/parse de KML, `ensureTurf` con CDNs de respaldo, análisis de recorte, render, navegación lista  $\leftrightarrow$  mapa, modales, progreso, encuadre, limpiar).
- `style.css`: Estilos institucionales, mapa responsivo, preloader, lista de CVEGEO, etiquetas (`DivIcon`).
- `togeojson.js`: Librería local para convertir KML a GeoJSON (procesa en el cliente).

# Tomar en cuenta

---

- Coordenadas: Se asume EPSG:4326 (WGS84) tanto para KML como para localidades.
- Entrada KML: archivo .kml válido con al menos un Polygon/MultiPolygon; KMZ no soportado directamente.
- Área “núcleo”: buffer de 500 m (unidades en metros con Turf.buffer).
- Intersección: `T.booleanIntersects(localidad.geometry, area.geometry)` preservando la feature completa.
- Propiedades esperadas en localidades: `CVEGEO`, nombre `(NOM_LOC|NOMGEO|NOMBRE)` y `AMBITO`.
- Estado de UI: mapa principal, capa KML, capa buffer, capa resultados, mapa de referencias `featureLayersById`, `lastAreaBounds` para “Restaurar vista”.

## Carga de dependencias

---

- `ensureTurf()`: intenta varias URLs de CDN hasta disponer de Turf; todas las llamadas usan alias `T.*`.
- Preloader con barra de progreso y mensajes; análisis particionado para no bloquear el hilo UI.
- Cache-busting: `APP_VERSION` en `index.html` para `style.css`, `togeojson.js` e `index.js`.

## Seguridad y privacidad

---

- Sin backend ni envío de datos a terceros: el KML se procesa íntegramente en el navegador (cliente).
- El GeoJSON de localidades se obtiene de una URL pública:  
`https://cdn.sassoapps.com/Gabvy/localidades_4326.geojson`; considerar bajarlo de aquí mismo pesa 244 MB.

**Demo: <https://areasdeinteres.pages.dev/>**

---

- Prueba rápida: Cargar KML de muestra → seleccionar tipo de área → ejecutar recorte → verificar lista y popups.
- Fallos típicos: KML sin polígonos, indisponibilidad del CDN de localidades, fallo de carga de Turf (ver consola).