# Manual de usuario de Isocarta

*Isocarta* une *iso-* («igual», como en isolíneas) y *carta* («mapa»). Es un geoportal libre: una plataforma web para encontrar, ver, analizar, editar y compartir información geográfica. Este manual explica qué es un geoportal y cómo se usa cada función. También cuenta cómo enviar sugerencias y cómo crear plugins propios, ya sea para compartirlos como código libre o para comercializarlos.

# 1. ¿Qué es un geoportal?

Un **geoportal** es la puerta de entrada, en internet, a la información geográfica de un país, una región o una organización. Reúne en un solo sitio datos que suelen estar dispersos, por ejemplo:

- los límites administrativos;
- los ríos y las carreteras;
- las áreas protegidas;
- los servicios de salud y educación;
- las amenazas naturales.

Con un geoportal, esos datos se pueden ver juntos sobre un mapa.

Un geoportal no es sólo un visor de mapas. También sirve para:

- **Encontrar datos**: un catálogo con la ficha de cada capa (quién la produjo, de cuándo es, con qué licencia se puede usar).
- **Usarlos sin instalar nada**: medir, consultar atributos, cambiar la simbología o ver el relieve en 3D desde el navegador.
- **Integrarlos con otros sistemas**: publica los datos con **estándares abiertos** del Open Geospatial Consortium (OGC), como WMS, WFS u OGC API. Así los abren QGIS, ArcGIS, GDAL o el geoportal de otra institución.
- **Producir y compartir**: subir capas propias, dibujar, versionar los cambios, publicar mapas web y compartir enlaces.

Los geoportales son el corazón de una **Infraestructura de Datos Espaciales (IDE)**. Esa es la red de instituciones, normas y servicios que permite que la información geográfica pública se encuentre, se comparta y se reutilice.

# 2. Isocarta y su ejemplo con datos libres de Panamá

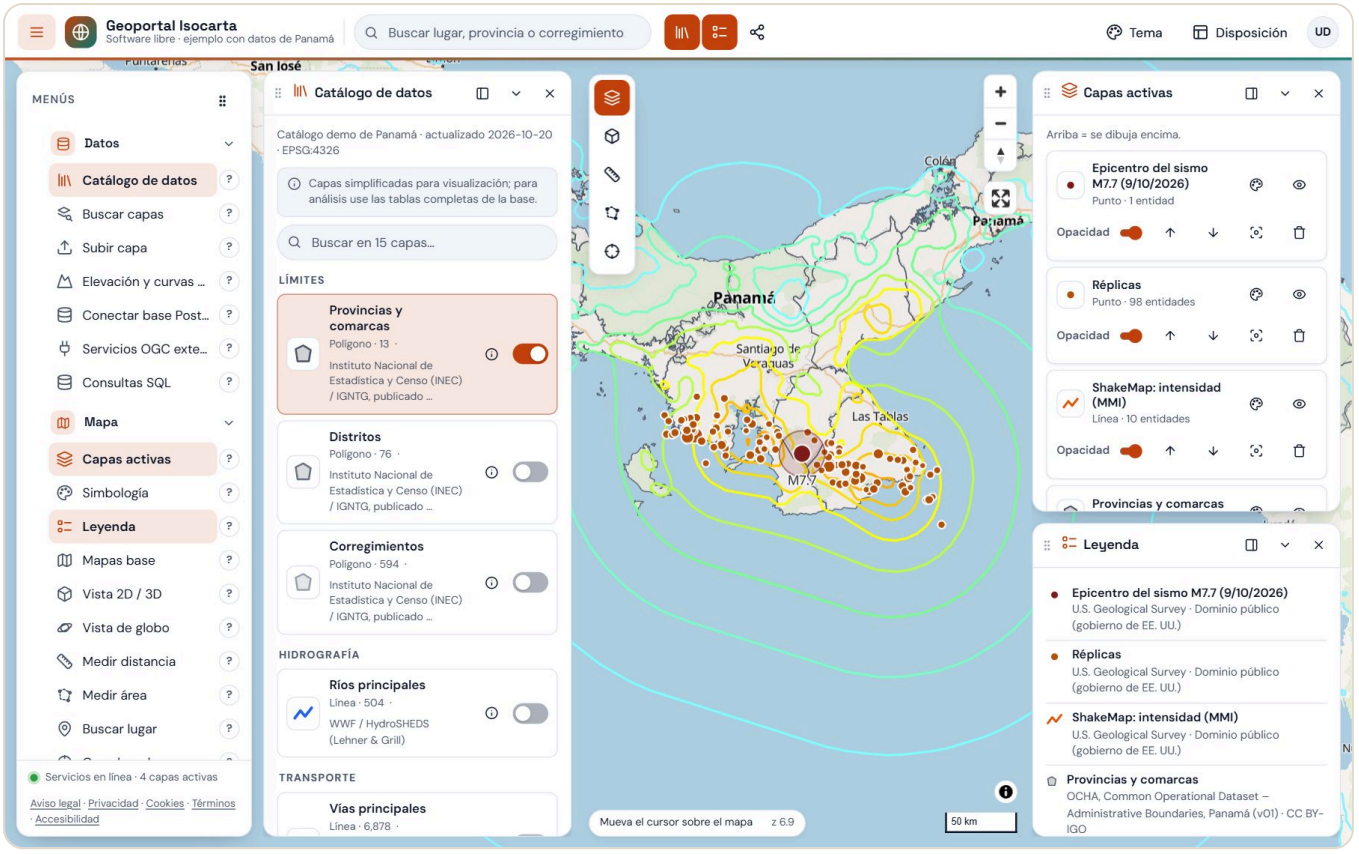
Isocarta se publica con un **ejemplo de datos libres de Panamá**, para probar todas sus funciones sin configurar nada. Son capas abiertas, simplificadas para verlas con fluidez en el navegador. Cada una conserva su fuente y su licencia en la ficha y en la leyenda.

| Tema | Capas | Fuente | Licencia |
|---|---|---|---|
| Límites | Provincias y comarcas (13), distritos (76), corregimientos (594) | INEC / IGNTG, publicado por OCHA (HDX) | CC BY-IGO |
| Hidrografía | Ríos principales (504) | WWF / HydroSHEDS | uso libre con atribución |
| Transporte | Vías principales (6 878) | colaboradores de OpenStreetMap (Geofabrik) | ODbL 1.0 |
| Ambiente | Áreas protegidas, SINAP 2025 (93) | Ministerio de Ambiente (MiAMBIENTE) | servicio público; licencia no declarada |
| Servicios esenciales | Establecimientos de salud (1 055), centros educativos (2 728) | Healthsites, HOT, a partir de OpenStreetMap | ODbL 1.0 |
| Sismo de Azuero 2026 | Epicentro M7.7, réplicas, ShakeMap (MMI y PGA), plano de ruptura | Servicio Geológico de EE. UU. (USGS) | dominio público |
| Amenazas | Fallas activas (74), sismos históricos M ≥ 6 (154) | GEM Foundation; USGS | CC BY-SA 4.0; dominio público |

El ejemplo gira en torno al **sismo de Azuero del 9 de octubre de 2026**: epicentro, réplicas, intensidades y fallas, junto a la población, las escuelas y los centros de salud. Así muestra para qué sirve un geoportal ante una emergencia.

> En el sitio de demostración se entra con una cuenta. Las cuentas, la bitácora de auditoría, las extensiones y las capas que usted suba se guardan **en su navegador**: otro navegador no las ve.

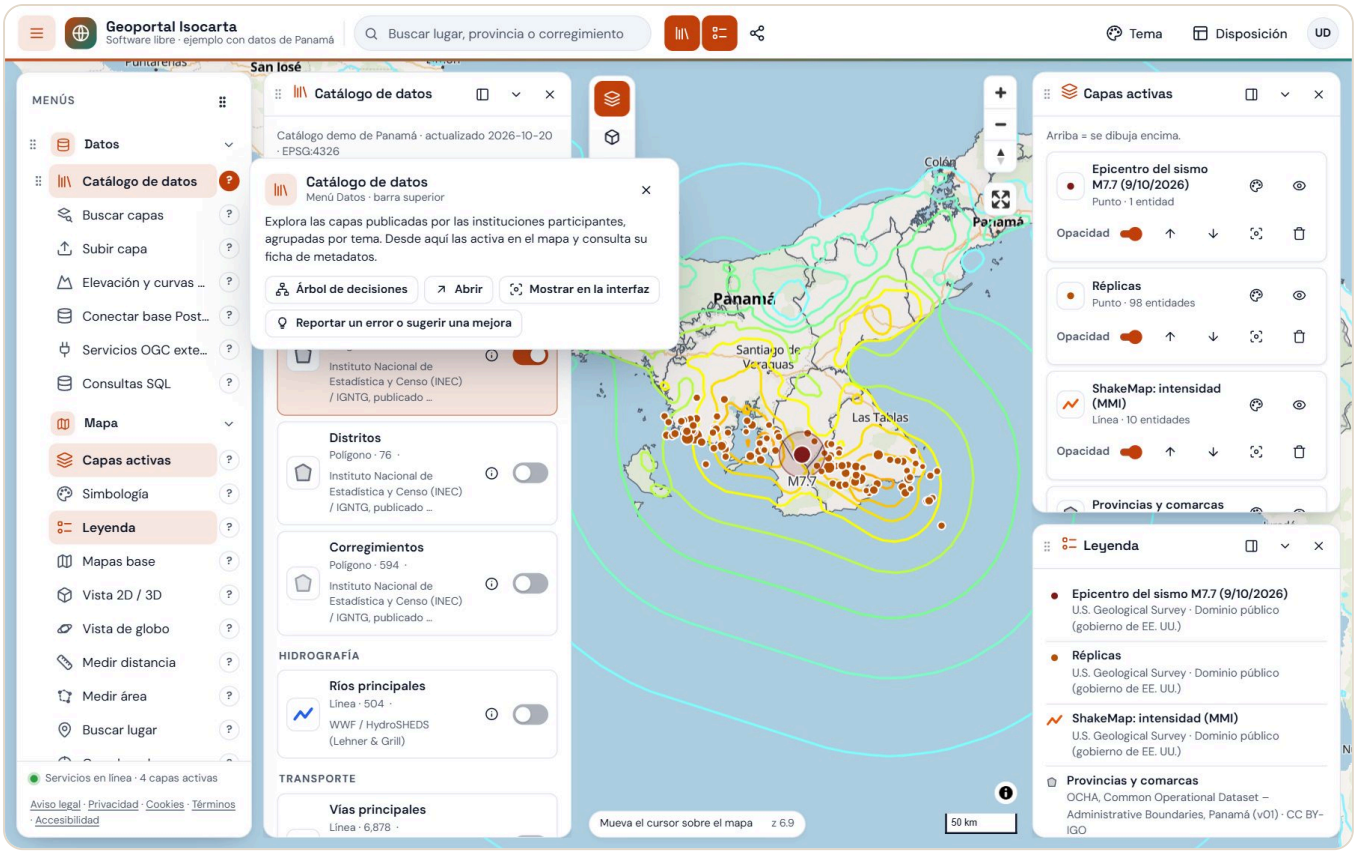
# 3. La pantalla




La pantalla de Isocarta: menús a la izquierda, el mapa al centro y los paneles a la derecha

- **Barra superior**:
- el buscador de lugares;
- accesos directos al Catálogo, la Leyenda y Compartir;
- **Tema** (colores y tipografía);
- **Disposición** (cómo se acomoda la interfaz);
- su cuenta.
- **Menús** (izquierda): las funciones, agrupadas en *Datos, Mapa, Editar, Publicar, Extensiones, Administrar* y *Ayuda*. El botón ☰ los oculta para dejar más espacio al mapa.
- **Barra del mapa**: capas, vista 3D, medir distancia y área, y coordenadas.
- **Paneles**: cada función abre un panel que se puede plegar, cerrar, ampliar o arrastrar a otro lado.
- **Pie del mapa**: la posición del cursor en grados y en UTM, el nivel de zoom y la escala.

**Ayuda contextual.** Junto a cada opción del menú hay un botón **?**. Explica para qué sirve la función y en qué menú está, y ofrece cuatro acciones: abrirla, mostrarla en la interfaz, recorrer el *árbol de decisiones* (qué necesita antes y qué habilita después) y **reportar un error o sugerir una mejora** sobre esa función.




La ayuda contextual del catálogo, con el acceso directo para reportar un error o sugerir una mejora

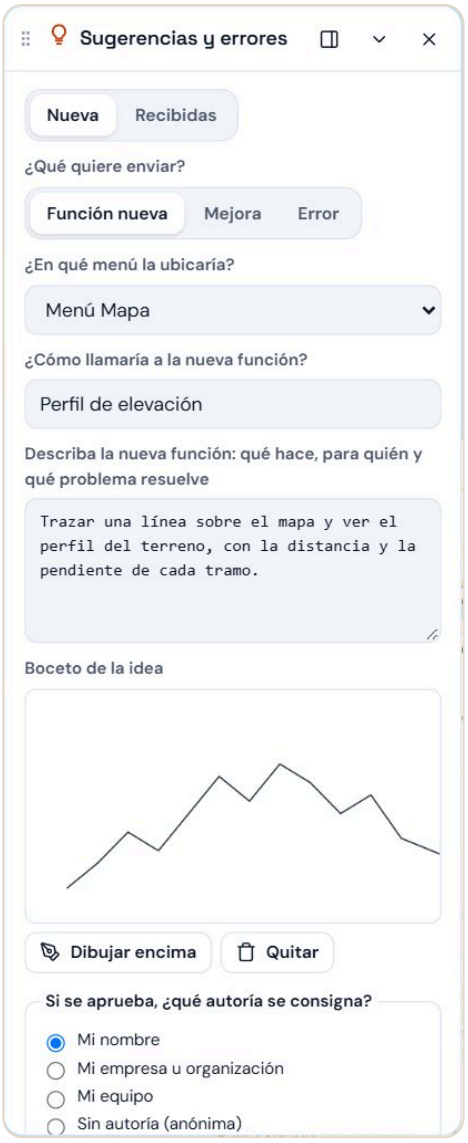
# 4. Cómo comunicar sugerencias, mejoras o errores

Isocarta mejora con lo que proponen quienes lo usan. Hay dos caminos:

1. **Menú Ayuda → Sugerencias y errores.**
2. Desde el botón **?** de cualquier función, con *Reportar un error o sugerir una mejora*. El formulario se abre con esa función ya elegida.

El formulario distingue tres tipos de envío:

| Tipo | Qué se pide |
|---|---|
| **Función nueva** | en qué menú la ubicaría, cómo la llamaría, qué hace y para quién, y un **boceto**: una foto de un dibujo en papel, una captura o un dibujo hecho ahí mismo |
| **Mejora** | la función que se mejora y qué cambiaría |
| **Error** | qué pasó, qué esperaba y los pasos para reproducirlo; puede adjuntar datos técnicos (navegador, pantalla y últimos errores de la página) |




Propuesta de una función nueva, con su boceto dibujado en el propio formulario

Si la propuesta se aprueba, usted elige **qué autoría se consigna**: su nombre, su empresa u organización, su equipo, o ninguna. También elige **cómo se desarrolla**:

- **Aporte a la plataforma (código abierto).** El equipo la desarrolla dentro de Isocarta y se publica con su licencia libre. Su autoría queda en el registro de cambios y en los créditos.
- **Extensión propia (plugin).** La desarrollan usted, su empresa o su equipo, no el equipo de Isocarta. El código y los derechos son suyos. En ese caso el formulario pide la **dirección donde los usuarios podrán adquirirla**.




La modalidad «Extensión propia», con la dirección donde se adquiere

En *Recibidas*, un administrador o publicador da seguimiento a cada envío (*Recibida, En estudio, Aprobada, En desarrollo, Resuelta* o *Descartada*), con notas. Cada envío se puede copiar como texto o descargar en JSON para hacerlo llegar al equipo.

# 5. Las funciones, una por una

## 5.1 Menú Datos

**Catálogo de datos.** Arriba están las **capas activas** (las que están en el mapa, con sus controles; ver *Capas activas* en el menú Mapa). Debajo, las demás capas por tema, con su tipo de geometría, el número de elementos y la fuente. El interruptor activa la capa en el mapa —pasa entonces a las capas activas— y el botón ⓘ abre su ficha de metadatos. Las capas que se encienden desde *Servicios OGC externos*, *Conectar base PostGIS* o *Elevación (DEM)*, y las derivadas de un DEM, se agregan al catálogo (grupos «Servicios OGC externos», «PostGIS» y «Elevación y derivadas») y siguen ahí aunque se quiten del mapa, también al volver a entrar. El botón ✕ las saca del catálogo.




El catálogo de datos

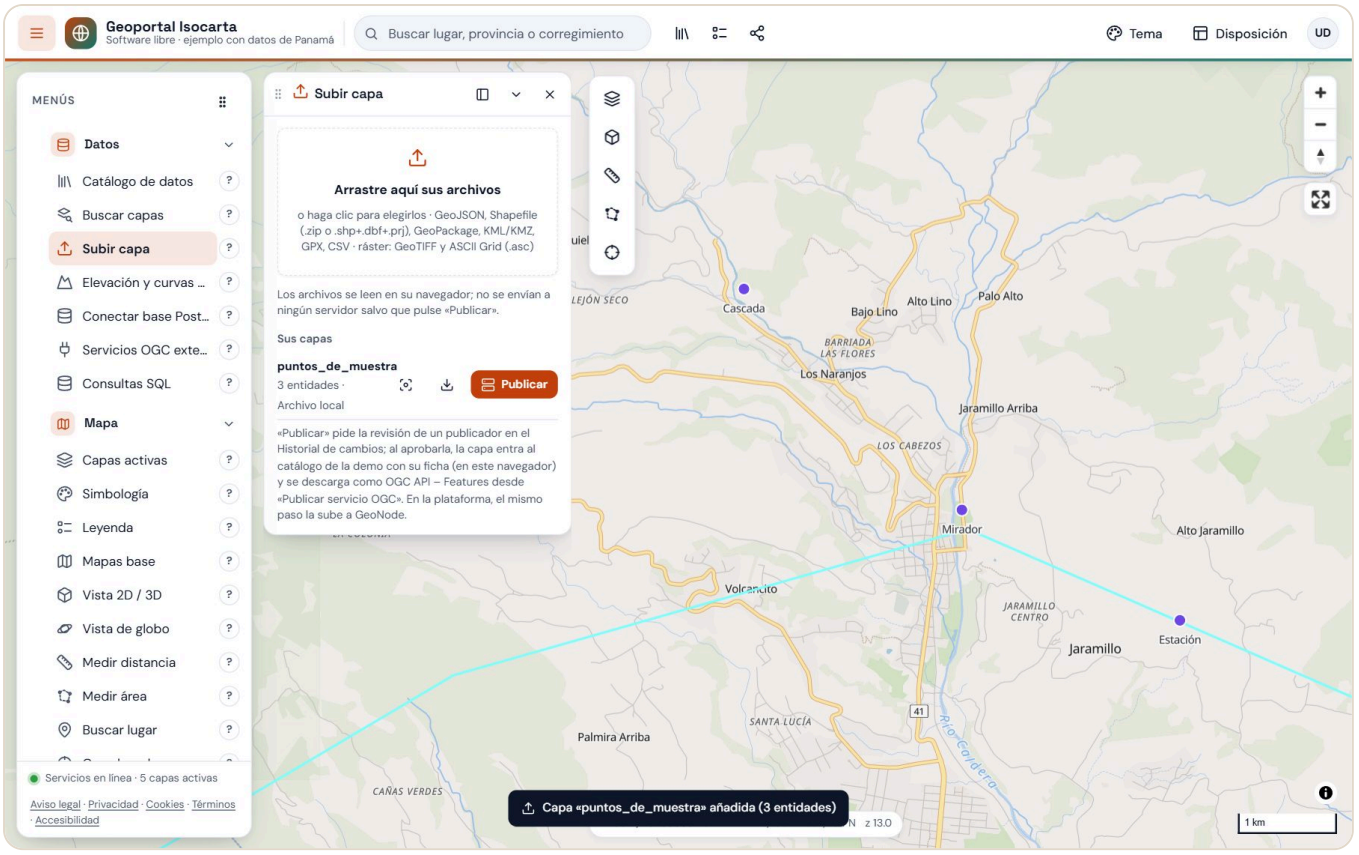
**Buscar capas.** Filtra el catálogo por palabras: título, institución o tema.




Búsqueda de capas con la palabra «sismo»

**Subir capa.** Arrastre sus archivos o elíjalos. Acepta GeoJSON, Shapefile (.zip o .shp + .dbf + .prj), GeoPackage, KML/KMZ, GPX y CSV, y también ráster: GeoTIFF y ASCII Grid.

- Los archivos **se leen en su navegador** y no se envían a ningún servidor salvo que usted pulse *Publicar*.
- Si el archivo trae otro sistema de coordenadas (por ejemplo UTM 17N o CRTM05), se reproyecta. Si no lo declara, Isocarta propone uno según la extensión de los datos.




Una capa propia recién subida

**Elevación (DEM).** Carga, para la zona que muestra el mapa, un modelo digital de elevación (DEM) global y gratuito. Se nombra con su cuadrante de 1° (como las teselas: *N08 W081*) y el lugar más céntrico de la zona, por ejemplo «GLO-30 · N08 W081 · Santiago», y queda en el Catálogo de datos, grupo «Elevación y derivadas».

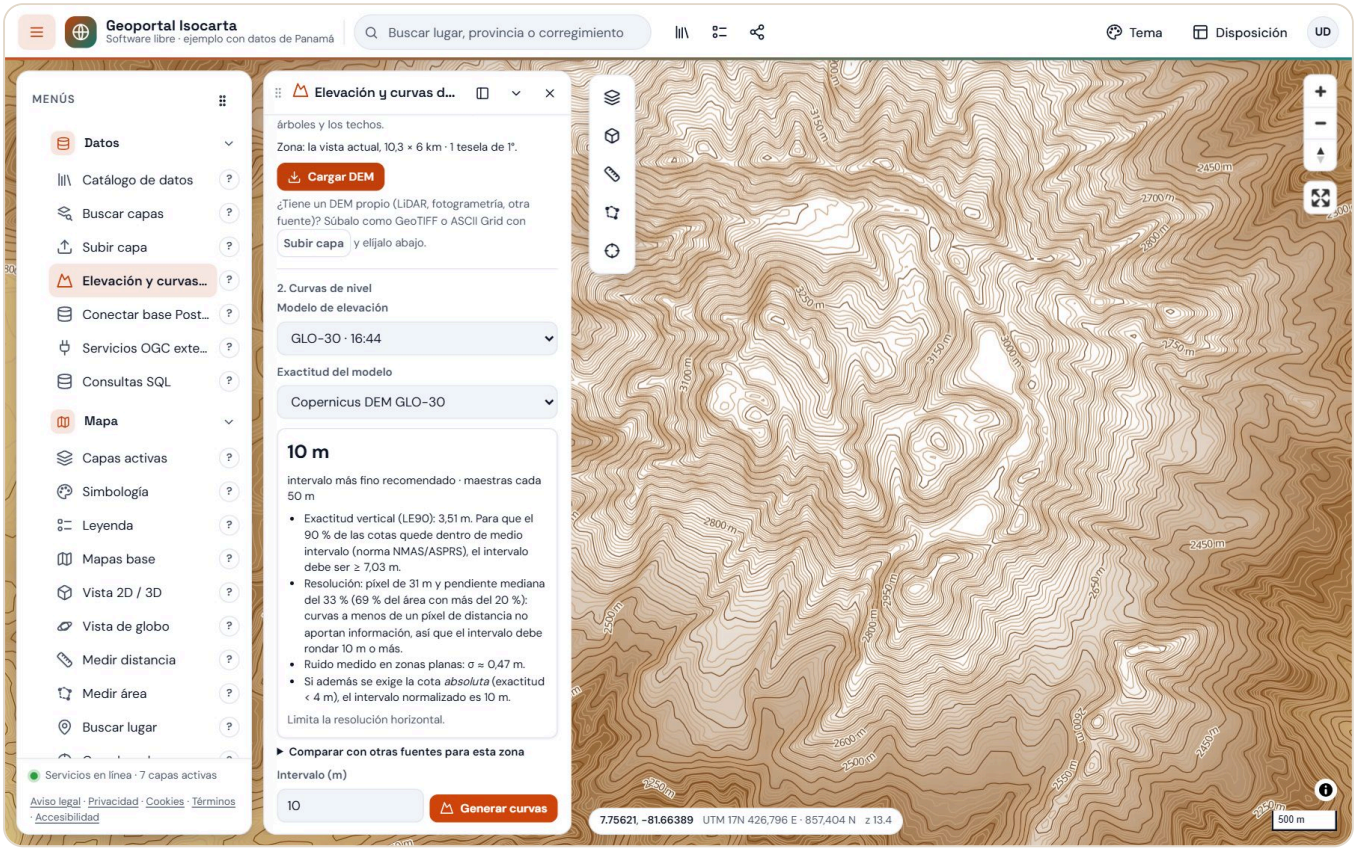
**Derivadas del DEM.** Con el botón de montaña de un DEM en el Catálogo de datos se generan sus derivadas: **curvas de nivel**, **sombreado**, **pendiente** (%) y **orientación** (°). Fuentes de DEM disponibles:

| Fuente | Píxel | Exactitud vertical (LE90) |
|---|---|---|
| Copernicus DEM GLO-30 | 30 m | < 4 m |
| Copernicus DEM GLO-90 | 90 m | < 4 m |
| ALOS AW3D30 (JAXA) | 30 m | ≈ 5 m |
| SRTM GL1 (NASA) | 30 m | ≈ 9 m |

- Acerque el mapa a la zona de interés: se cargan hasta 9 teselas de 1° × 1°. Los datos se leen directamente de OpenTopography y quedan sólo en su navegador.
- También sirve un DEM propio (LiDAR, fotogrametría con dron u otro): súbalo como GeoTIFF o ASCII Grid con *Subir capa*, elija su tipo e indique su exactitud (RMSE) si la conoce.

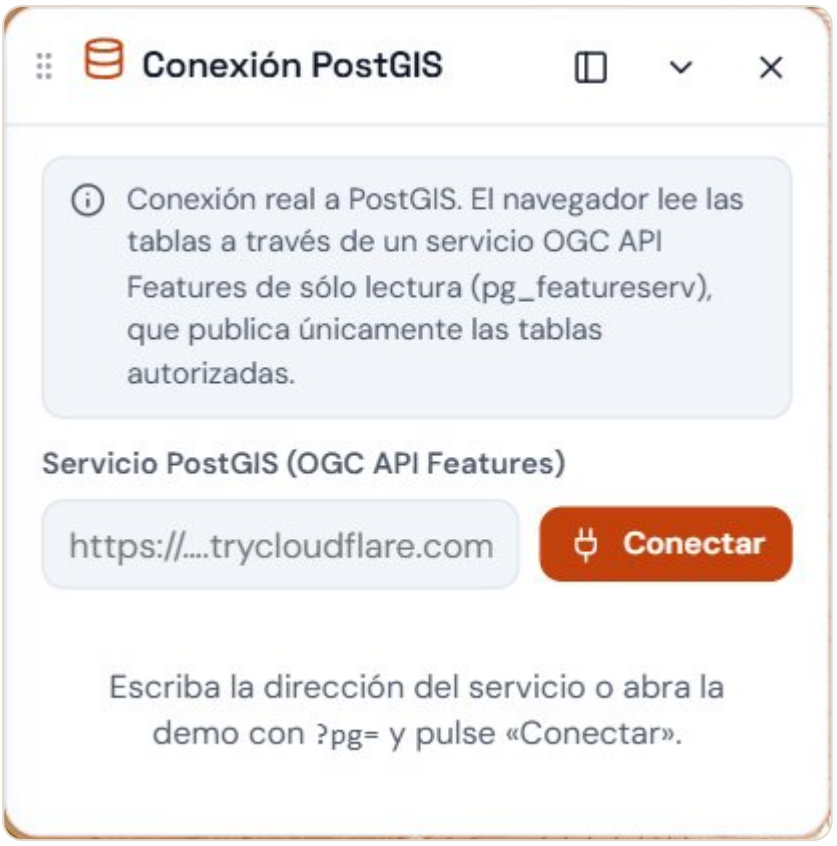
- **El intervalo se estima solo.** Isocarta propone el intervalo más fino que el modelo sostiene, con dos límites:
- la **exactitud**: el intervalo debe ser al menos el doble del error vertical (LE90), para que el 90 % de las cotas quede dentro de medio intervalo (criterio NMAS/ASPRS). El ruido se mide además en el propio modelo, en las zonas planas;
- la **resolución**: en terreno empinado las curvas no pueden estar más juntas que el tamaño del píxel.
- Se redondea a la serie habitual (1, 2, 2,5, 5, 10, 20, 25, 50 m…). Las curvas **maestras** van cada 5 intervalos (cada 4 si el intervalo es 2,5 o 25 m) y llevan su cota.
- Una tabla compara las cuatro fuentes para la misma zona y marca con ★ la que permite el intervalo más fino. Puede escribir otro intervalo; si es más fino que el recomendado, se le advierte.
- El DEM descargado, las curvas y las demás derivadas se agregan como **capas** al catálogo y a *Capas activas*. El botón **Simbología** de cada una abre su estilo: la rampa de colores del relieve, o el color, el grosor y las etiquetas de las curvas maestras e intermedias. También se pueden consultar, editar, exportar y publicar, y se conservan, con su estilo, al volver a entrar.

> Los modelos globales son de **superficie** (DSM): en bosques y ciudades miden la copa de los árboles o los techos, no el suelo. Para obras de detalle use un DEM de terreno (LiDAR).




Curvas de nivel cada 20 m sobre el Copernicus DEM GLO-30 en Boquete, con su intervalo recomendado

**Conectar base PostGIS.** Muestra como capas vivas las tablas de una base PostgreSQL/PostGIS, a través de un servicio OGC API Features de sólo lectura (pg_featureserv). Sólo se ven las tablas que ese servicio autoriza.




Conexión a PostGIS

**Servicios OGC externos.** Conecta servicios WMS u OGC API Features de otras instituciones (por ejemplo, las imágenes satelitales de NASA GIBS) y los muestra junto a las capas propias. El servidor externo tiene que permitir CORS.




Servicios OGC externos

**Consultas SQL.** Consultas espaciales en SQL (PostGIS) sobre las capas del catálogo y las suyas. Se ejecutan en una base que corre en su propio navegador, o en una base real a través de la pasarela de la plataforma. Los resultados con geometría se agregan al mapa. Es una extensión: se instala desde el panel Extensiones.




La extensión Consultas SQL

## 5.2 Menú Mapa

**Capas activas.** Las capas que están en el mapa, en la parte superior del Catálogo de datos (este comando abre el catálogo). Para cada una puede:

- mostrarla u ocultarla y cambiar su opacidad;
- cambiar el orden de dibujo (la de arriba se dibuja encima);
- acercar el mapa a su extensión;
- abrir su simbología o su ficha de metadatos, o quitarla.

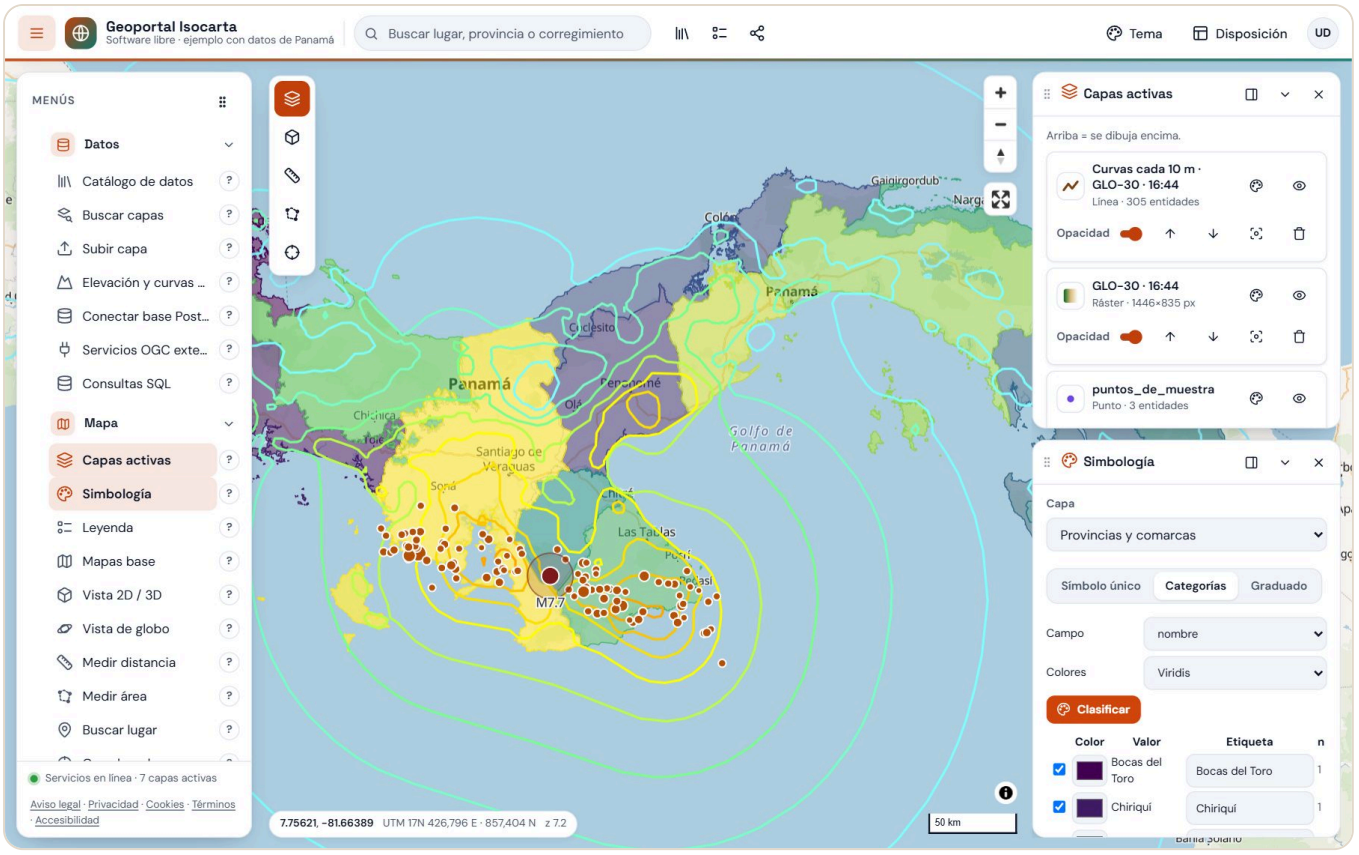



Capas activas

**Simbología.** Tematiza una capa, como en QGIS o GeoServer.

- **Vectores**:
- *símbolo único*;
- *categorías*, un color por valor de un campo;
- *graduado*, por intervalos iguales, cuantiles o cortes naturales de Jenks;
- además: rampas de color, borde, tamaño, etiquetas y la opción de ocultar clases.
- **Ráster**: gris, pseudocolor, paleta o RGB.

El estilo se **exporta e importa como SLD** (para GeoServer y GeoNode) **y como QML** (para QGIS), y queda guardado.



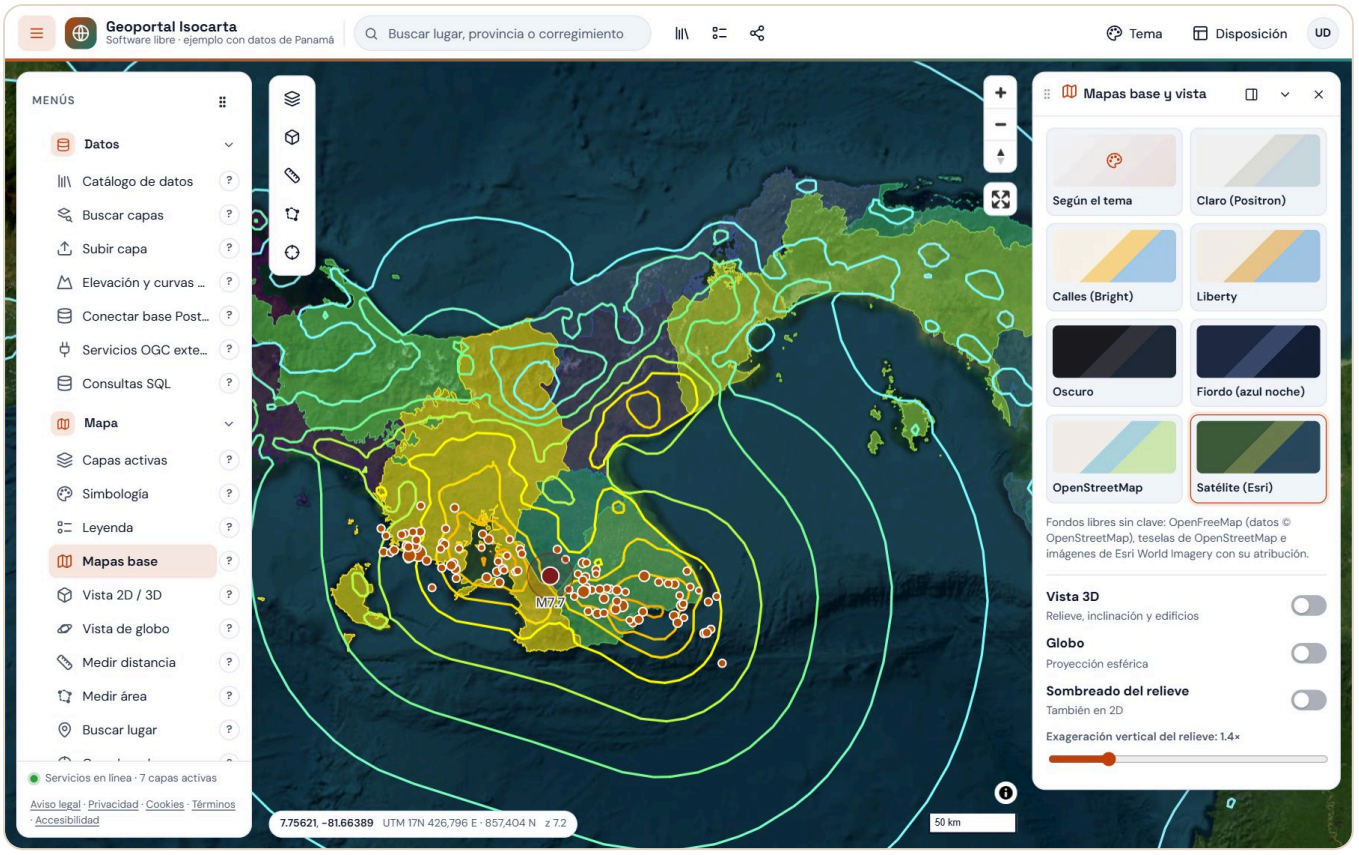

Provincias y comarcas por categorías del campo «nombre»

**Leyenda.** La simbología de las capas visibles, con la fuente y la licencia de cada una.




Leyenda

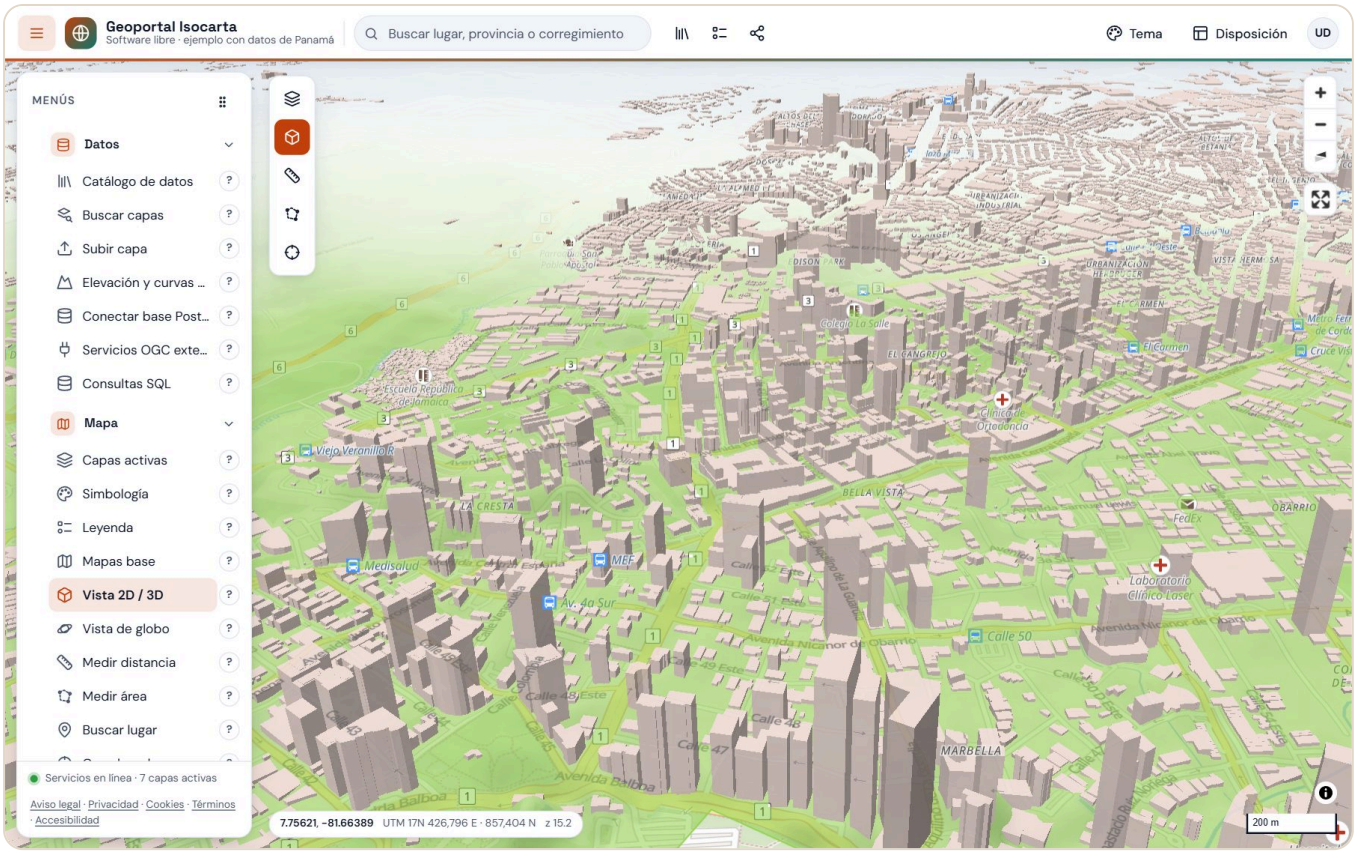
**Mapas base.** El fondo del mapa: claro, calles, Liberty, oscuro, fiordo, OpenStreetMap o satélite. Desde aquí también se activan el sombreado del relieve, la vista 3D y el globo.




Mapa base de satélite

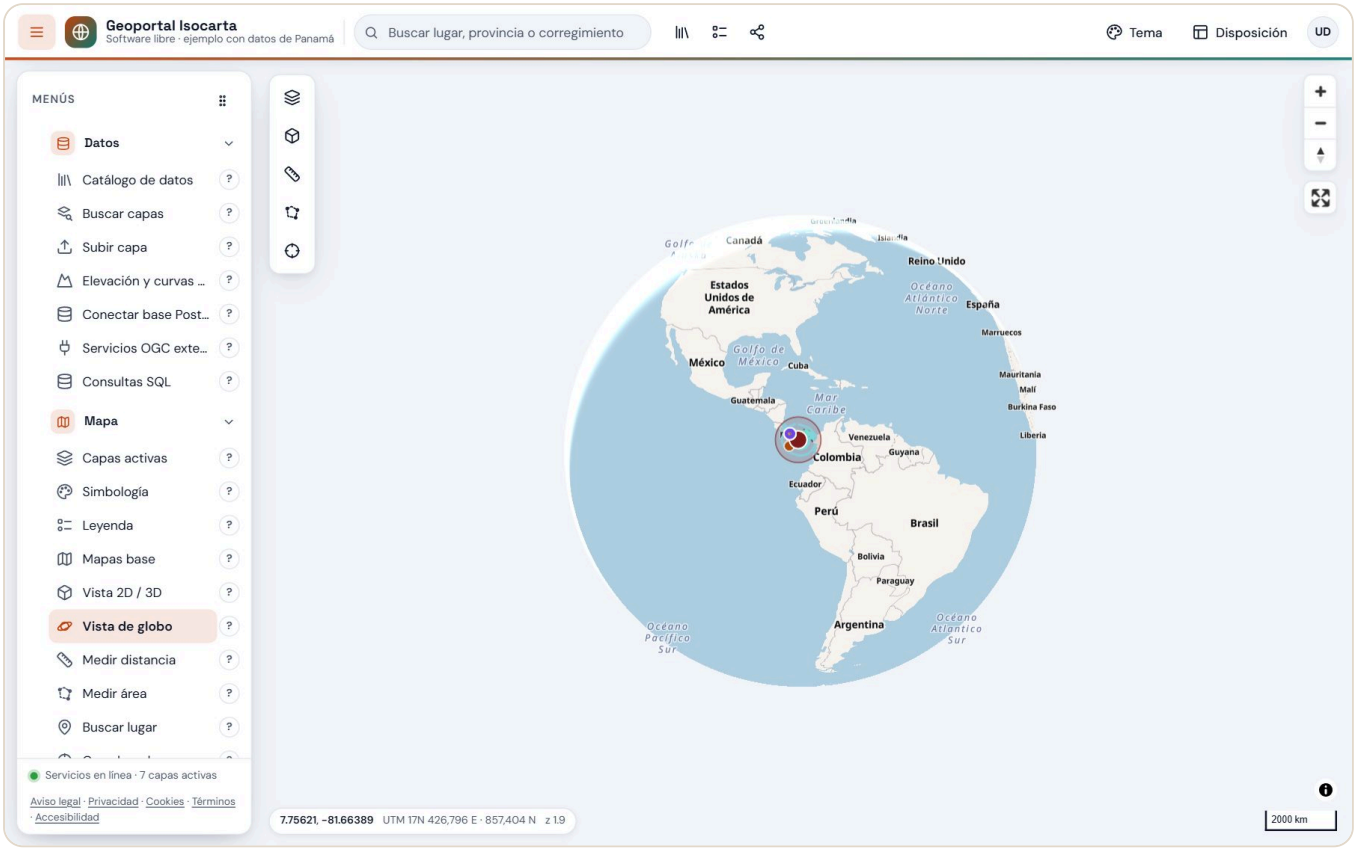
### Menú Tipo de vista

**Vista 2D / 3D.** El terreno con relieve real, inclinación y edificios extruidos. Acérquese a una ciudad para verlos.




Vista 3D de la ciudad de Panamá

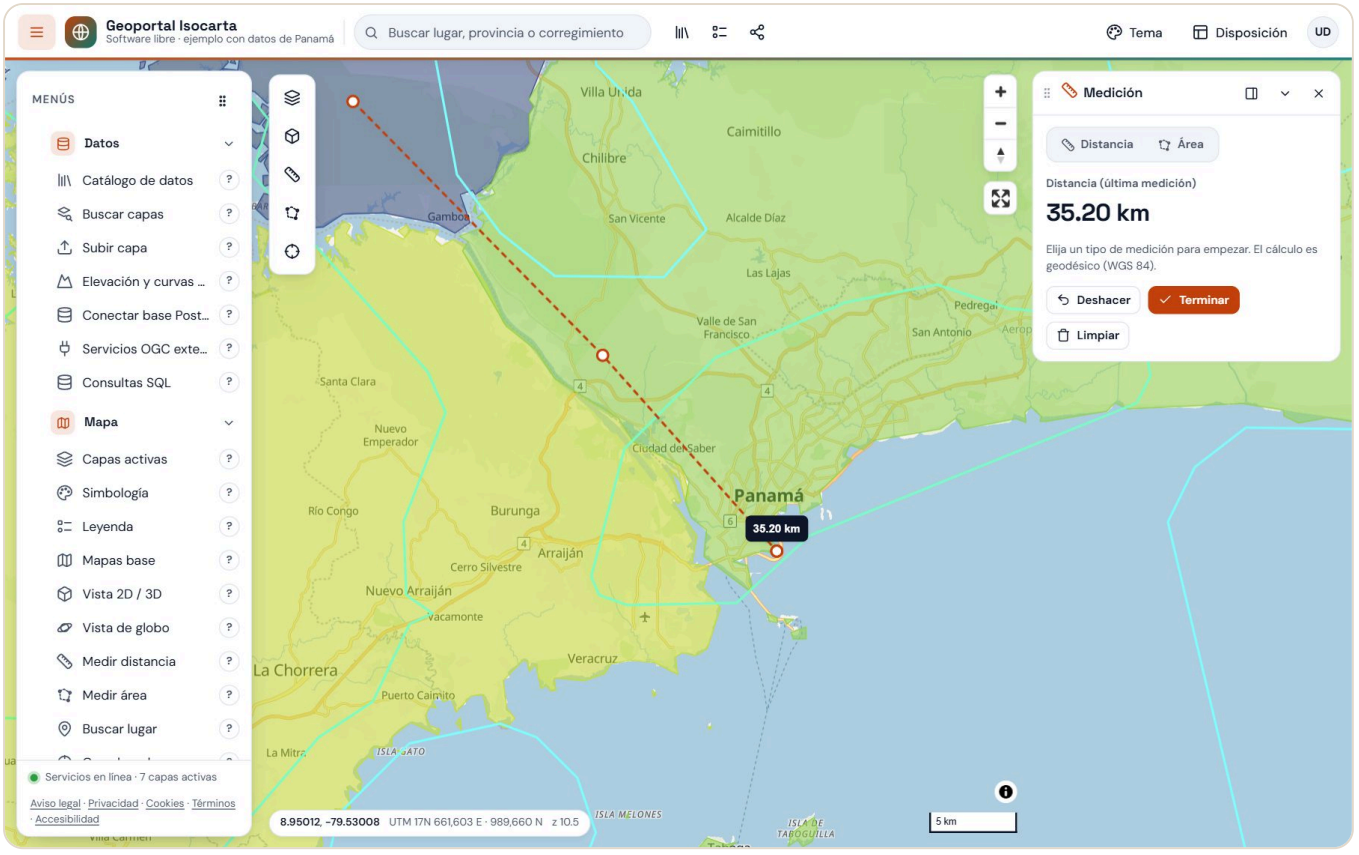
**Vista de globo.** Proyecta el mapa sobre un globo para ver Panamá en su contexto regional.



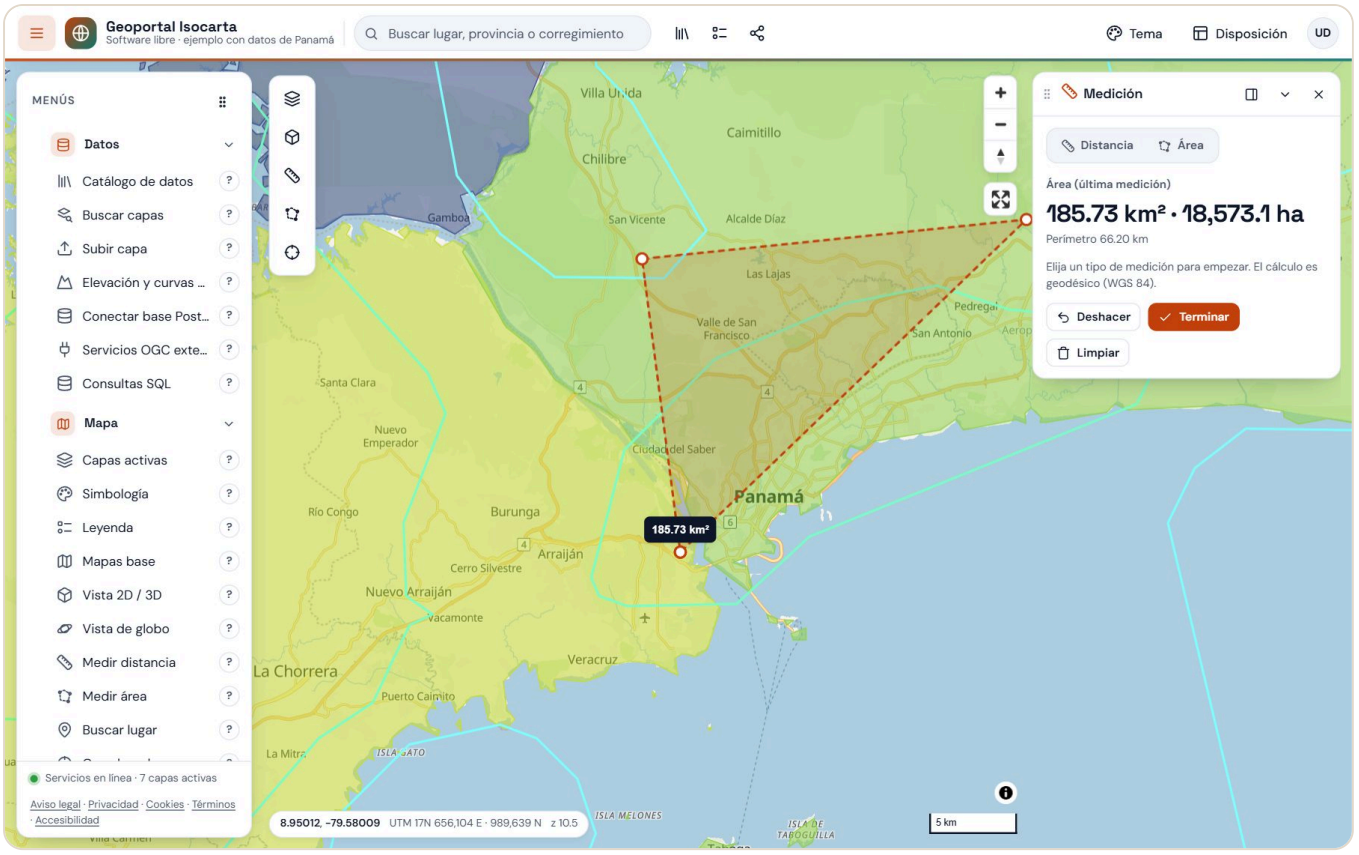

Vista de globo

### Menú Métricas

**Medir distancia y Medir área.** Haga clic en el mapa para añadir vértices y doble clic para terminar. El cálculo es geodésico, sobre el elipsoide WGS 84: la distancia en metros o kilómetros, y el área en hectáreas y km² con su perímetro. *Deshacer* quita el último vértice.

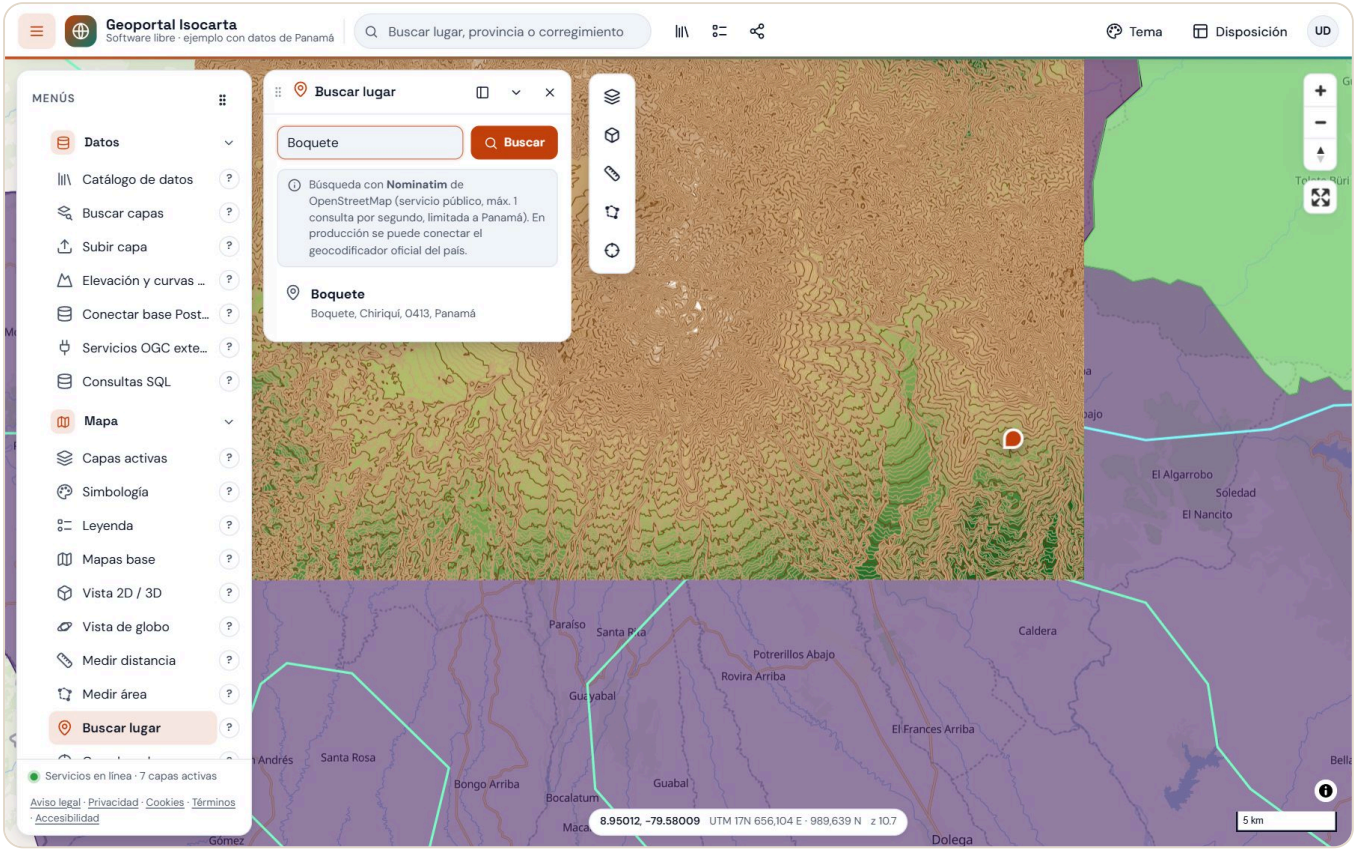



Medición de una distancia




Medición de un área

**Buscar lugar.** Busca poblados, direcciones y accidentes geográficos en el servicio Nominatim de OpenStreetMap, y lleva el mapa al resultado.




Búsqueda de «Boquete»

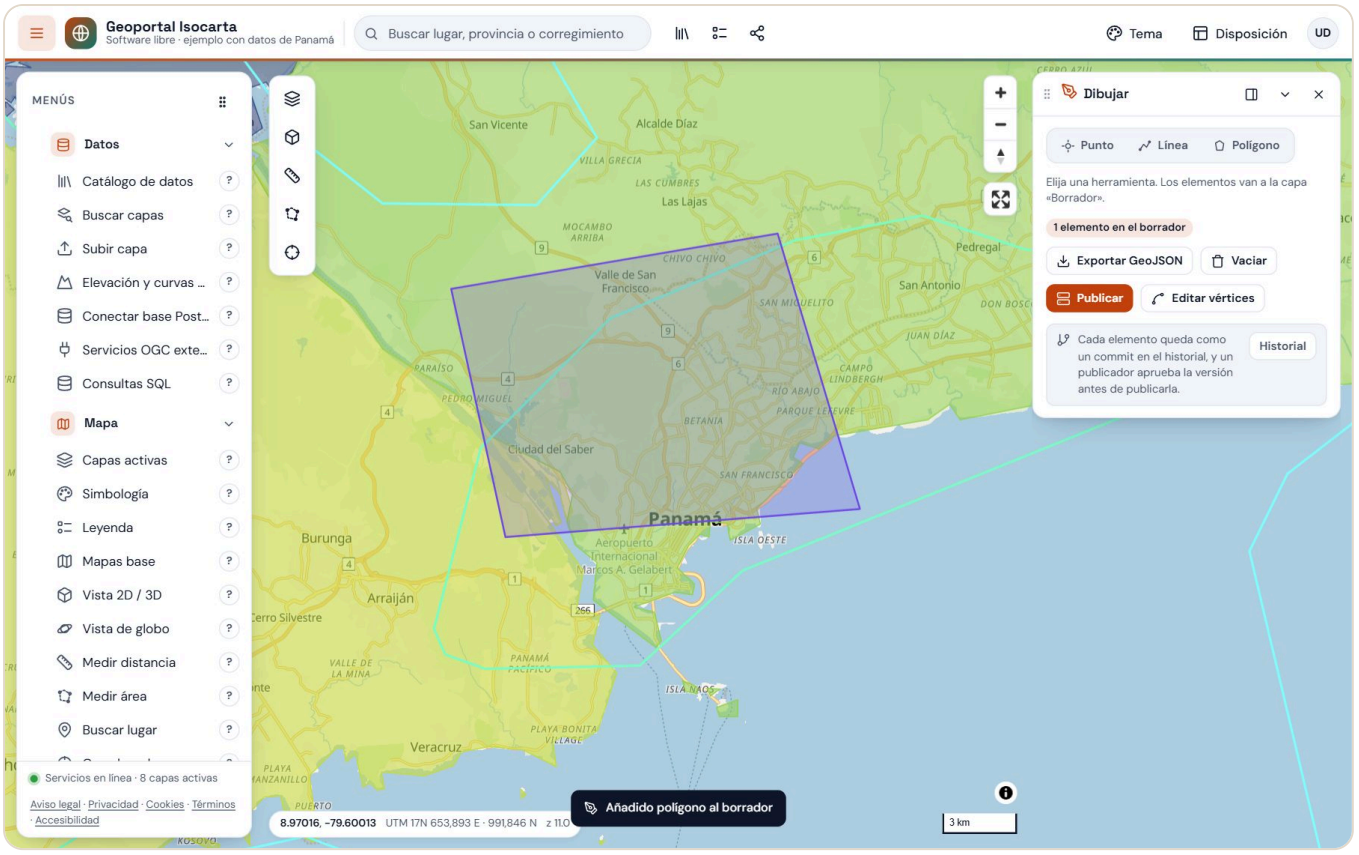
**Coordenadas.** La posición del cursor y del centro del mapa en grados, en grados-minutos-segundos y en UTM 17N. También lleva el mapa a unas coordenadas que usted escriba.




Coordenadas

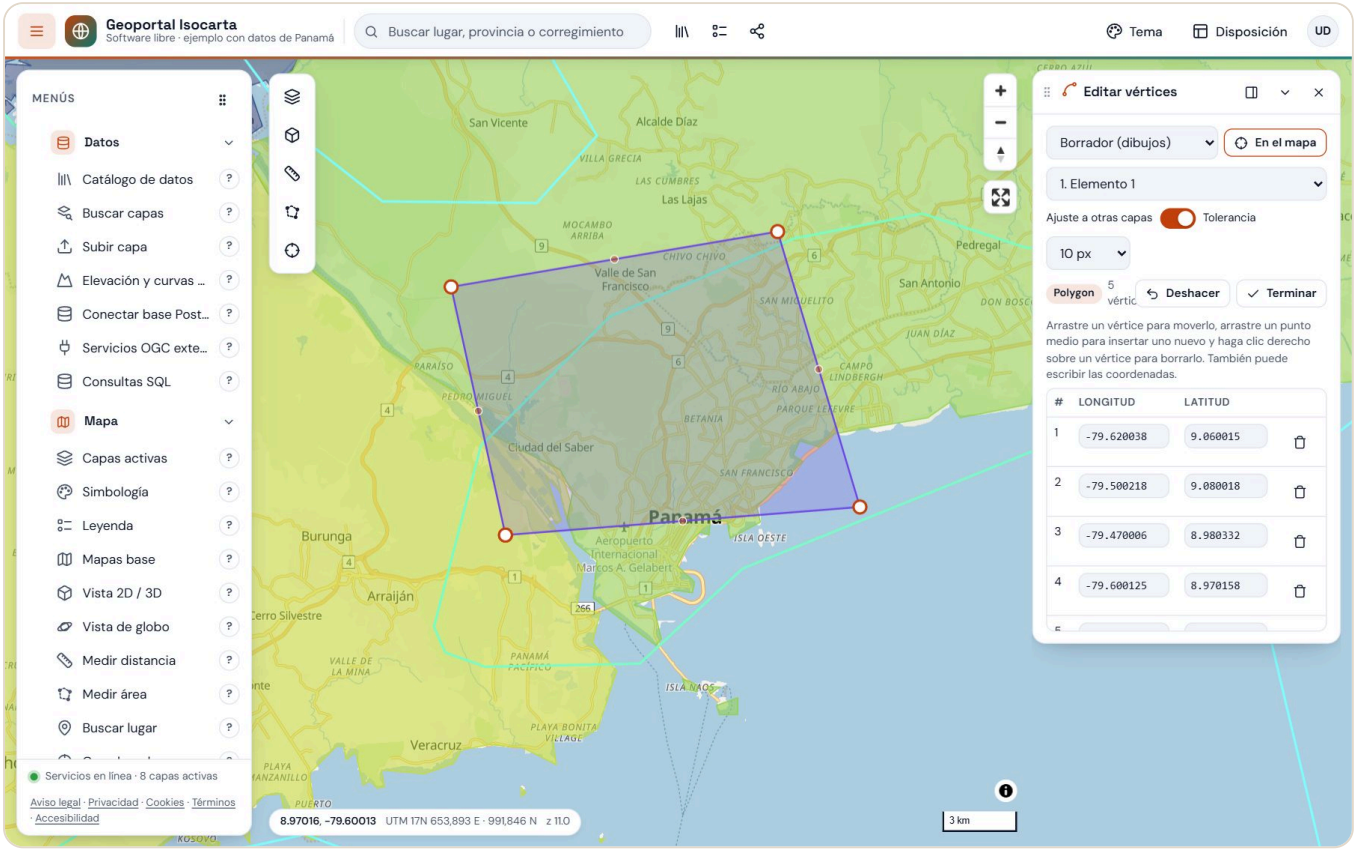
## 5.3 Menú Editar

**Dibujar.** Crea puntos, líneas y polígonos en la capa *Borrador*. Cada elemento queda como una versión en el Historial y se puede exportar en GeoJSON o enviar a revisión para publicarlo.




Un polígono dibujado

**Editar vértices.** Mueve, añade o elimina vértices de sus capas: las subidas o las dibujadas. Puede ajustar los vértices a otras capas con una tolerancia en píxeles. Las capas publicadas no se editan aquí.




Edición de vértices

**Tabla de atributos.** Los atributos de cualquier capa activa, con un renglón por elemento. Desde un renglón puede acercar el mapa a ese elemento.

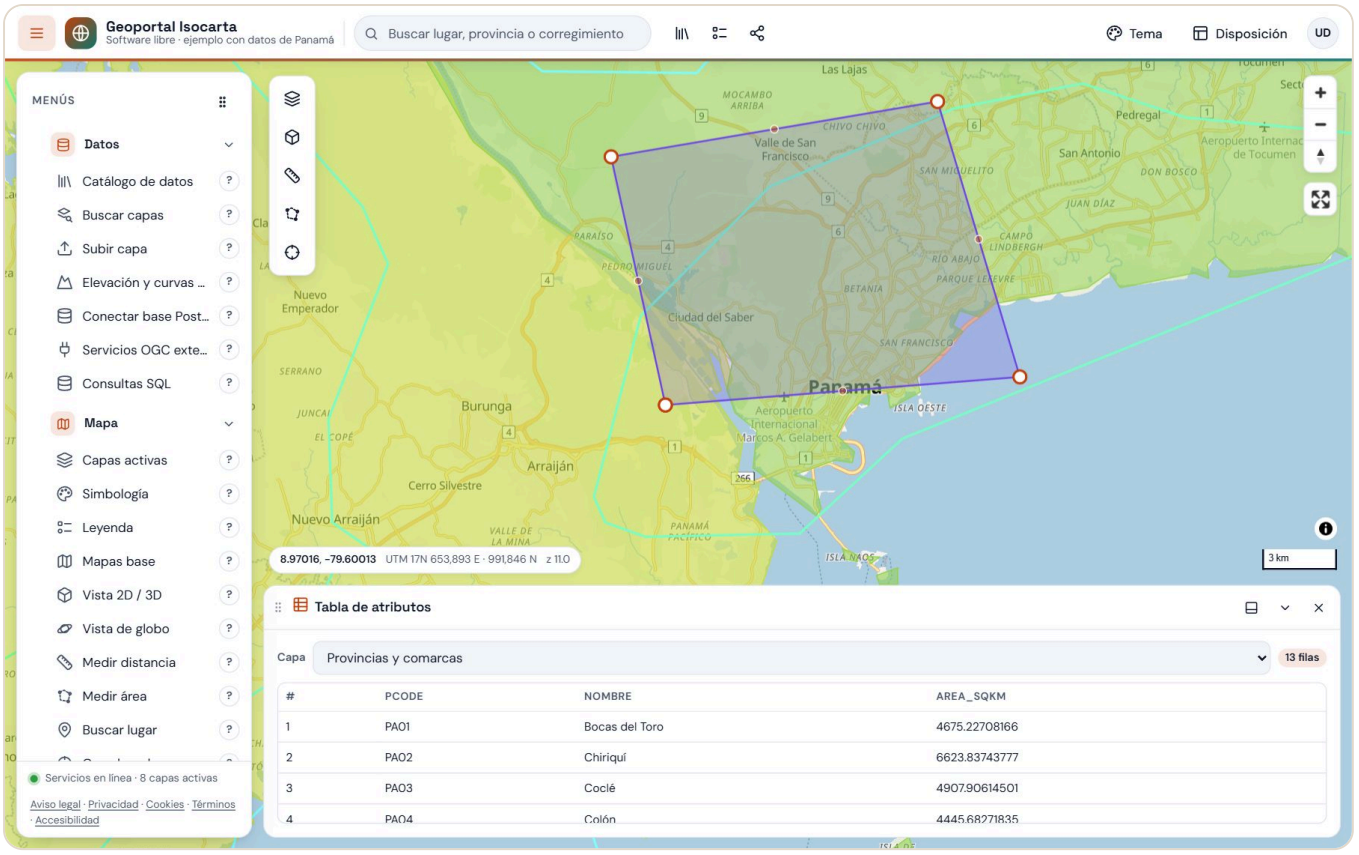
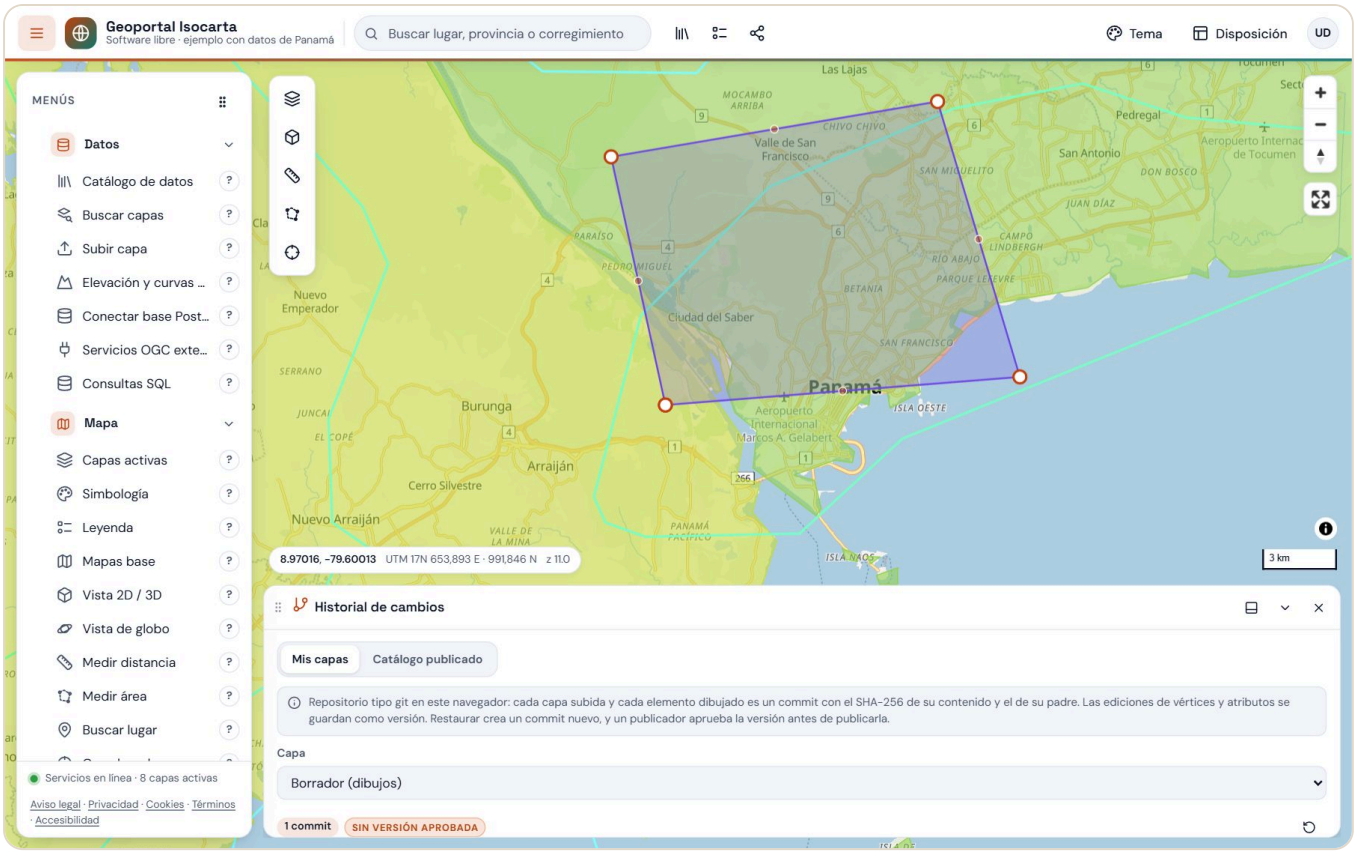



Tabla de atributos de las provincias

**Historial (git).** Cada capa subida y cada elemento dibujado quedan como un *commit*, con la huella SHA-256 de su contenido. El historial muestra quién cambió qué y cuándo. Restaurar una versión crea un commit nuevo, y un publicador aprueba la versión antes de publicarla.



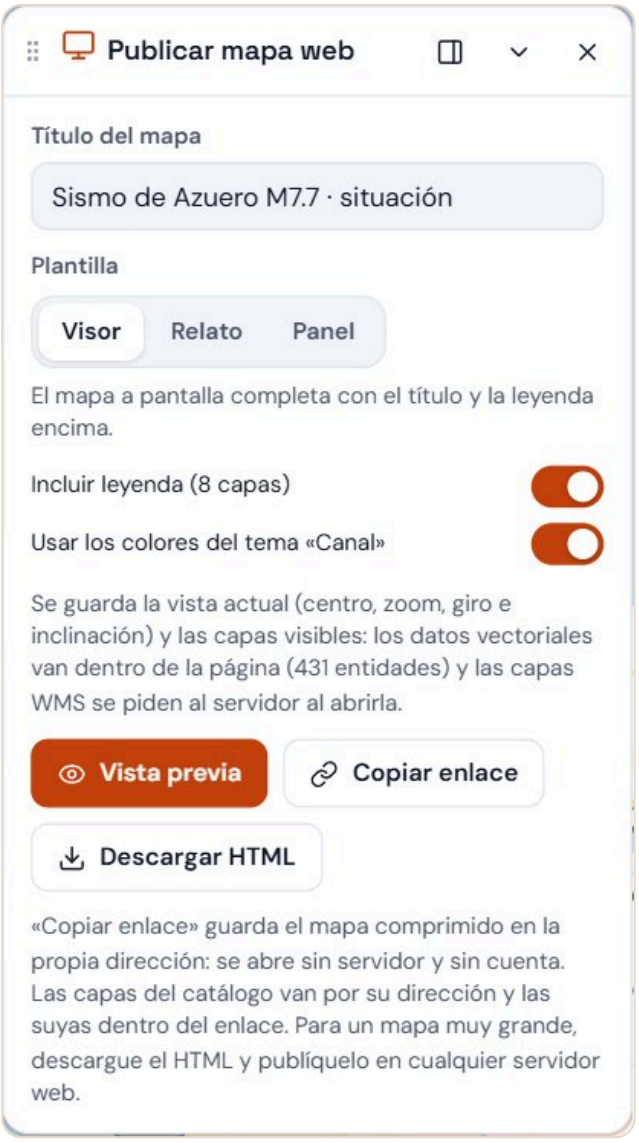

Historial de cambios

## 5.4 Menú Publicar

**Mapa web.** Convierte la vista actual (capas, leyenda y tema) en un mapa web, con cinco plantillas:

- *Visor*: el mapa a pantalla completa, con el título y la leyenda;
- *Relato*: un texto a la izquierda que cuenta lo que muestra el mapa;
- *Panel*: un panel lateral para encender y apagar cada capa;
- *Formal*: una lámina a escala fija, con coordenadas en los bordes, para PDF o para imprimir;
- *Infografía*: una página que cuenta una historia con el mapa, cifras y gráficos.

*Copiar enlace* guarda el mapa comprimido en la propia dirección, así que se abre sin servidor y sin cuenta. Con el servidor del geoportal, el enlace es corto ( `#m=…` ) y queda registrado en *Publicado*. *Descargar HTML* da una página para publicar en cualquier servidor web.

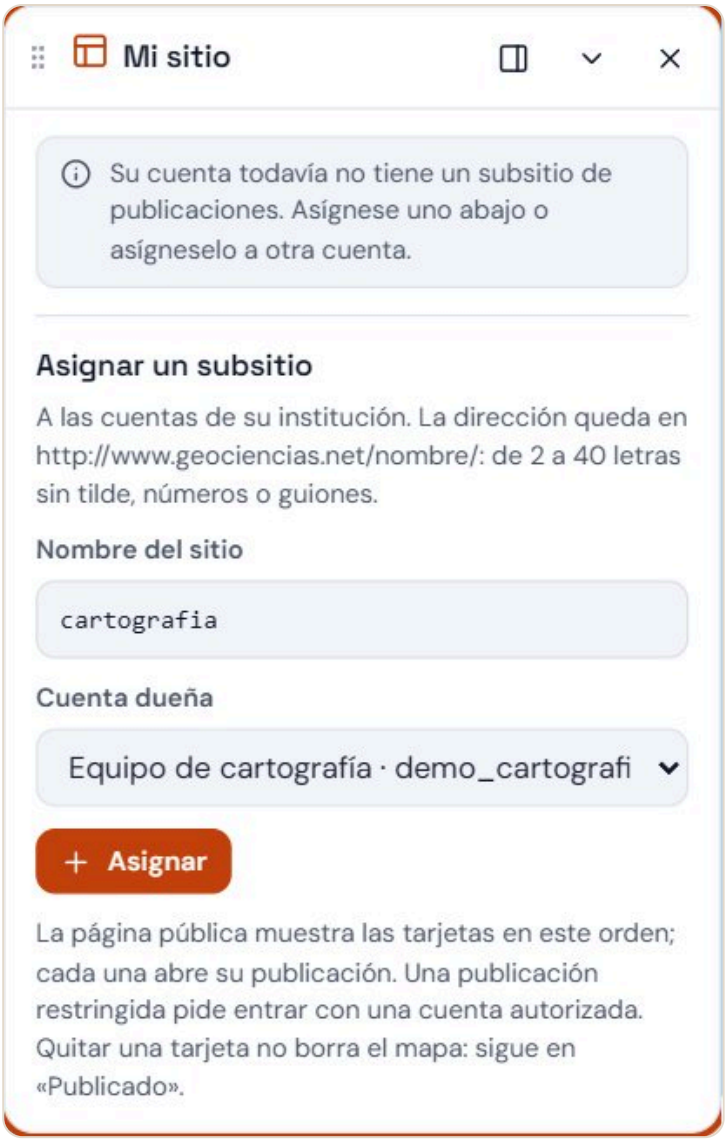



Publicar un mapa web

**Publicado.** Los mapas que usted publicó con enlace corto, con sus visitas y quién puede verlos: cualquiera con el enlace, cualquier cuenta del portal, o sólo las instituciones y cuentas que elija. El botón *A mi sitio* lleva cualquiera de ellos a su subsitio.

**Mi sitio (subsitio de publicaciones).** Una página pública propia, con una tarjeta por cada publicación, como la portada de los territorios indígenas en geociencias.net/territorios. Necesita el servidor del geoportal, que guarda los sitios y sus tarjetas. Funciona así:

**1. Un administrador asigna el sitio.** En *Publicar* → *Mi sitio*, el administrador de la institución (o el superadministrador) escribe el nombre del sitio y elige la cuenta dueña. El nombre va en la dirección: `www.geociencias.net/cartografia/` . Lleva de 2 a 40 letras sin tilde, números o guiones, y no puede ser una carpeta del geoportal (como `territorios` u `ogc` ). Una cuenta puede tener varios sitios.




El administrador asigna un subsitio a una cuenta

**2. Al publicar, se agrega la tarjeta.** Después de *Copiar enlace* en *Mapa web*, el botón *Agregar a mi sitio* abre la tarjeta ya llena: el título del mapa, su descripción y una miniatura tomada del mapa en pantalla. La miniatura se puede cambiar por otra imagen o quitar; sin ella, la tarjeta muestra la inicial del título sobre su color. Sirve para las cinco plantillas.



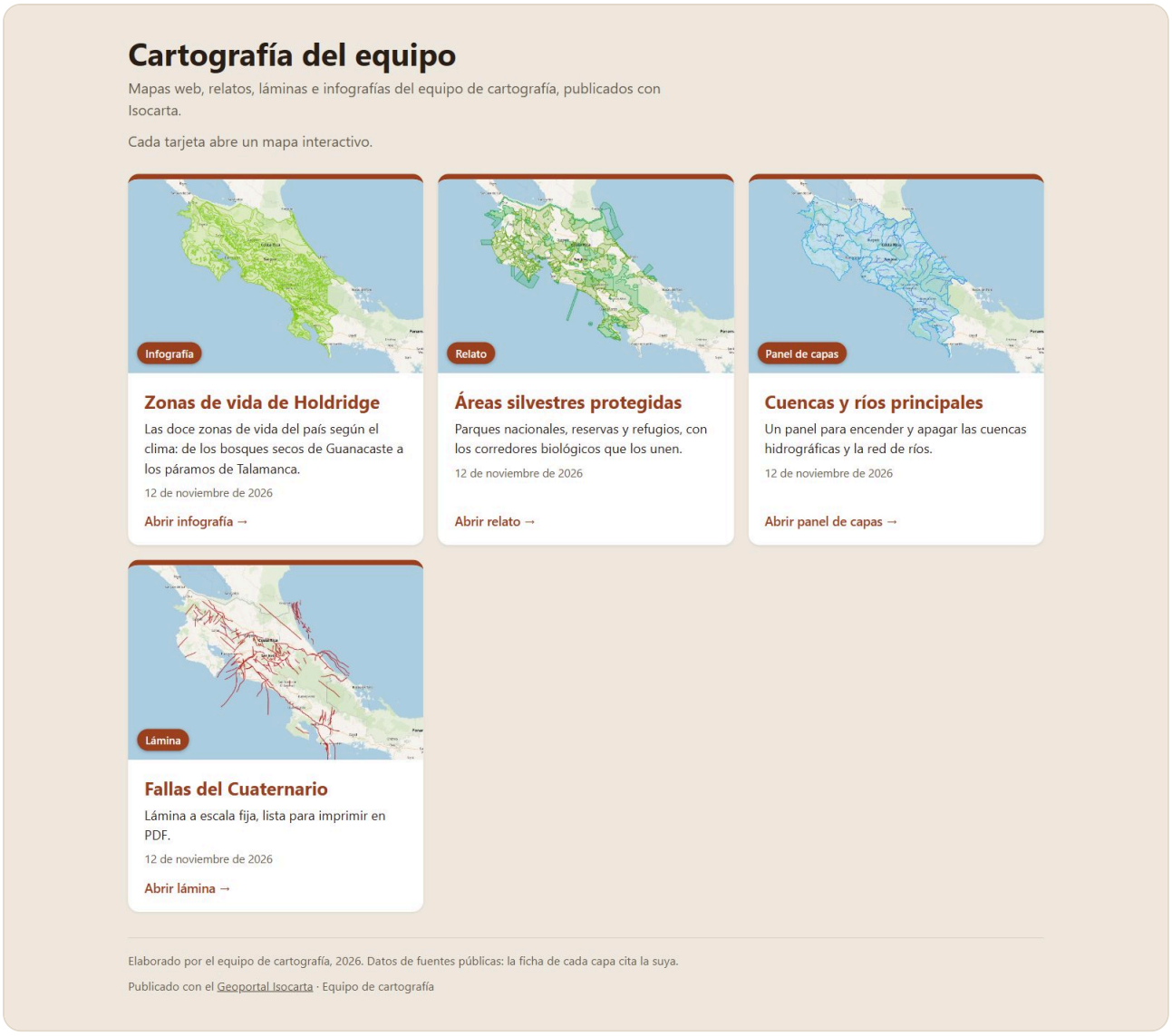

Agregar a mi sitio, al publicar un mapa web

**3. En *Mi sitio* se arma la portada.** Ahí están la dirección pública y sus visitas; el título, la presentación, el pie con la autoría y el color del sitio; y la lista de tarjetas, que se ordenan con las flechas, se editan o se quitan. *Agregar a mano* pone una tarjeta con cualquier otra dirección (un informe, otra página). Quitar una tarjeta no borra el mapa: sigue en *Publicado*. Con *Público* apagado, el sitio deja de verse.




Mi sitio: portada y tarjetas

**4. La página pública.** Cualquiera la abre, sin cuenta. Cada tarjeta lleva su miniatura, el tipo de publicación (Infografía, Relato, Panel de capas, Lámina o Mapa interactivo), la descripción y la fecha, y abre la publicación. Si una publicación es restringida, pide entrar con una cuenta autorizada. *Descargar el sitio (HTML)* da una copia en un solo archivo, con las miniaturas dentro, para alojarla en otro servidor.




La página pública del subsitio

> Las capturas de *Mi sitio* son de la instalación con datos de Costa Rica (www.geociencias.net), que tiene el servidor con base de datos; la demostración estática con datos de Panamá no lo tiene.

**Servicio OGC.** Publica las capas como servicio OGC API – Features, que abren QGIS (*Capa → Añadir capa WFS / OGC API – Features*) y GDAL ( `ogrinfo OAPIF:…` ). En la edición con servidor, GeoServer las sirve además como WMS y WFS, con filtros y permisos.




Servicio OGC

**Compartir.** Un enlace a la vista actual y el código para incrustarla en otra página.




Compartir

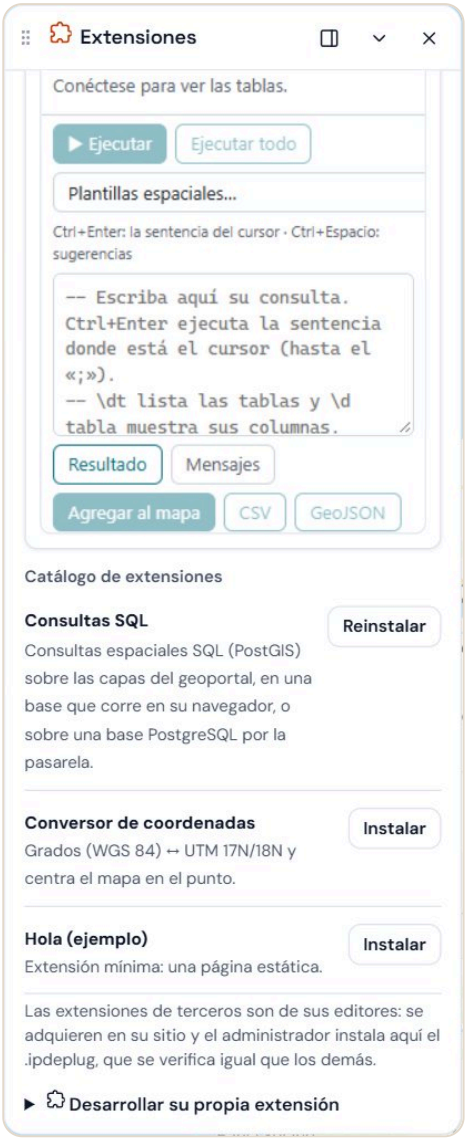
## 5.5 Menú Extensiones

Las extensiones (**plugins**) son paquetes firmados `.ipdeplug` . El menú tiene tres opciones:

**Extensiones instaladas.** Las que ya están en la plataforma. El administrador las habilita, las deshabilita o las quita; cualquiera puede abrir las habilitadas. De cada una se ven sus **permisos**, su **licencia**, su **precio** y el enlace al sitio del editor.

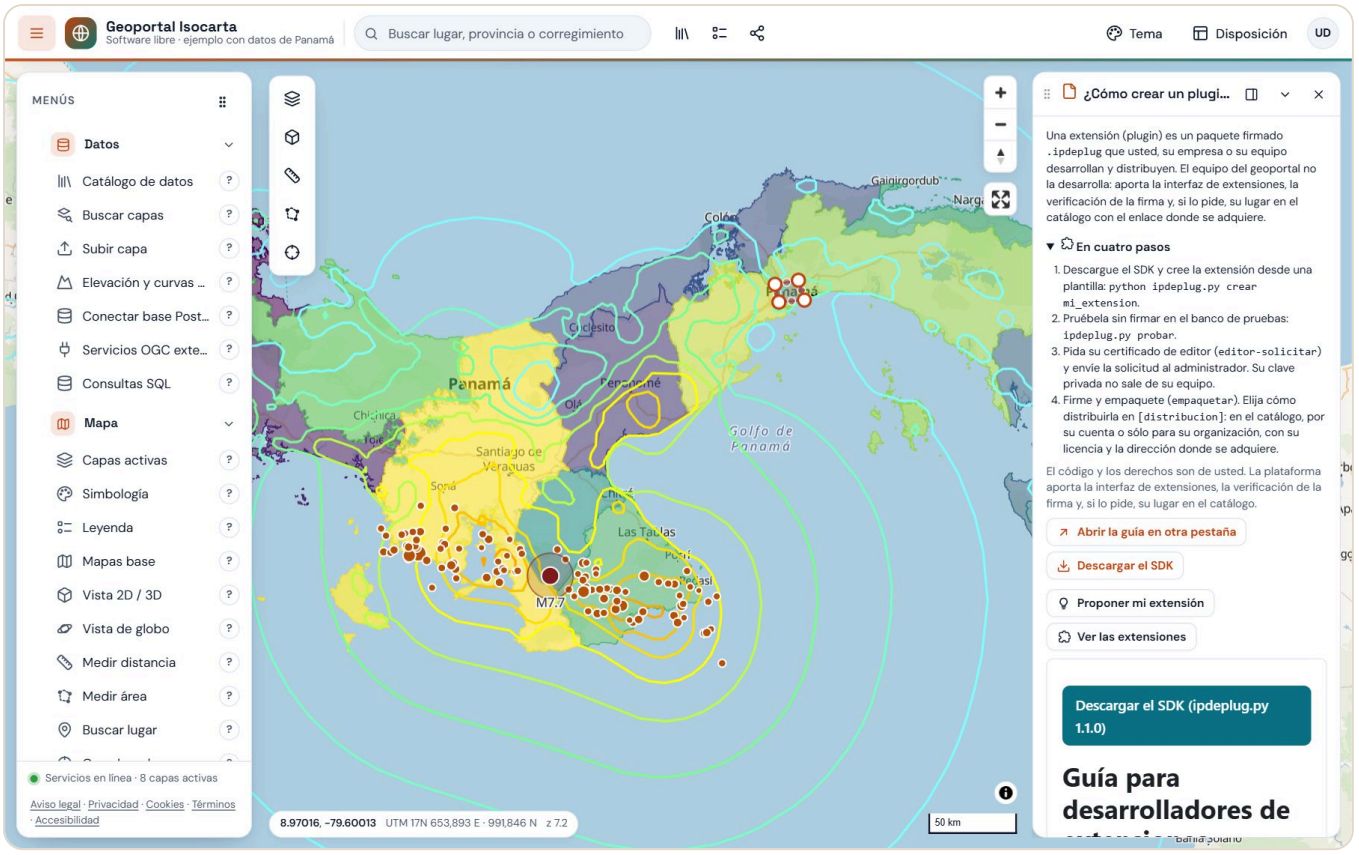
**Catálogo de extensiones.** Las que se pueden instalar, agrupadas por fuente, con un buscador. Cada una dice si es nueva, si ya está *instalada* o si hay una versión más reciente (*Actualizar*).

- **Fuentes.** La primera es el catálogo de la plataforma. Con *Agregar una fuente de extensiones* se añade la dirección (URL) del `indice.json` de otro editor o institución; tiene que ser https y permitir el acceso desde el navegador (CORS). Las fuentes agregadas se guardan en este navegador y se quitan con la ×.
- Antes de instalar, cada paquete se verifica de tres maneras: el certificado del editor, la firma y la huella de cada archivo. Venga de la fuente que venga, un paquete sin una firma válida se rechaza.
- Cada extensión se abre en un **marco aislado** y sólo puede hacer lo que declara su manifiesto.
- Sólo el administrador instala. Los demás ven el catálogo y pueden descargar el paquete.



Panel Extensiones con el catálogo

**¿Cómo crear un plugin propio?** Abre, dentro de la propia interfaz, la guía para desarrolladores y la descarga del SDK. La sección 6 lo explica en detalle.




La guía para crear un plugin propio

## 5.6 Menú Administrar

**Usuarios y roles.** Las cuentas, agrupadas por institución. Hay cinco roles: *superadministrador, administrador, publicador, editor* y *lector*.

- El **superadministrador** gestiona el portal entero: ve todas las instituciones, puede crearlas, editarlas y borrarlas, y es el único que cambia la seguridad, las extensiones, los temas y los permisos. En la demostración lo son Edgar, Gabriel y Oscar. Siempre debe quedar al menos uno activo.
- Cada cuenta sólo ve su institución. El **administrador** de una institución gestiona las cuentas de esa institución: invita, cambia roles, desactiva y desbloquea. No puede crear superadministradores ni tocar otras instituciones. En la auditoría ve sólo lo que hicieron las cuentas de su institución.
- Al abrir una institución se ven su ficha (nombre completo, sector, funciones y capas asignadas) y sus cuentas, ordenadas por rol: administrador, publicador, editor y lectores.
- *+ Institución* pide el alias (por ejemplo, MITRADEL), el nombre, el sector, las funciones y las capas asignadas. Puede crear de una vez sus cinco cuentas, con contraseñas temporales que se muestran una sola vez. Borrar una institución borra también sus cuentas.

- La demostración trae las 97 entidades públicas de Panamá que lista el Ministerio de Economía y Finanzas (informe del sector público a enero de 2026). Cada una tiene cinco cuentas: `ALIAS_ADMIN`, `ALIAS_PUBLICADOR`, `ALIAS_EDITOR`, `ALIAS_LECTOR01` y `ALIAS_LECTOR02`. La contraseña es el mismo usuario en mayúsculas; por ejemplo, `AMPYME_ADMIN` / `AMPYME_ADMIN`. La contraseña distingue mayúsculas. Las capas asignadas a la institución aparecen encendidas al entrar.
- Las contraseñas se guardan con PBKDF2 (310 000 iteraciones). Hay doble factor TOTP, y la cuenta se bloquea tras cinco intentos fallidos.
- En la plataforma con servidor, las cuentas las gestiona el directorio institucional (Keycloak).



Usuarios y roles

**Permisos por rol.** Una tabla de casillas con las funciones del menú y los cinco roles. Lo que un rol no puede usar desaparece de su menú y de sus barras, y su panel no se abre. Sólo el superadministrador la cambia, y no puede quitarse a sí mismo *Usuarios y roles* ni *Permisos por rol*. *Restablecer* vuelve a los permisos de partida:

- el lector consulta, mide y comparte;

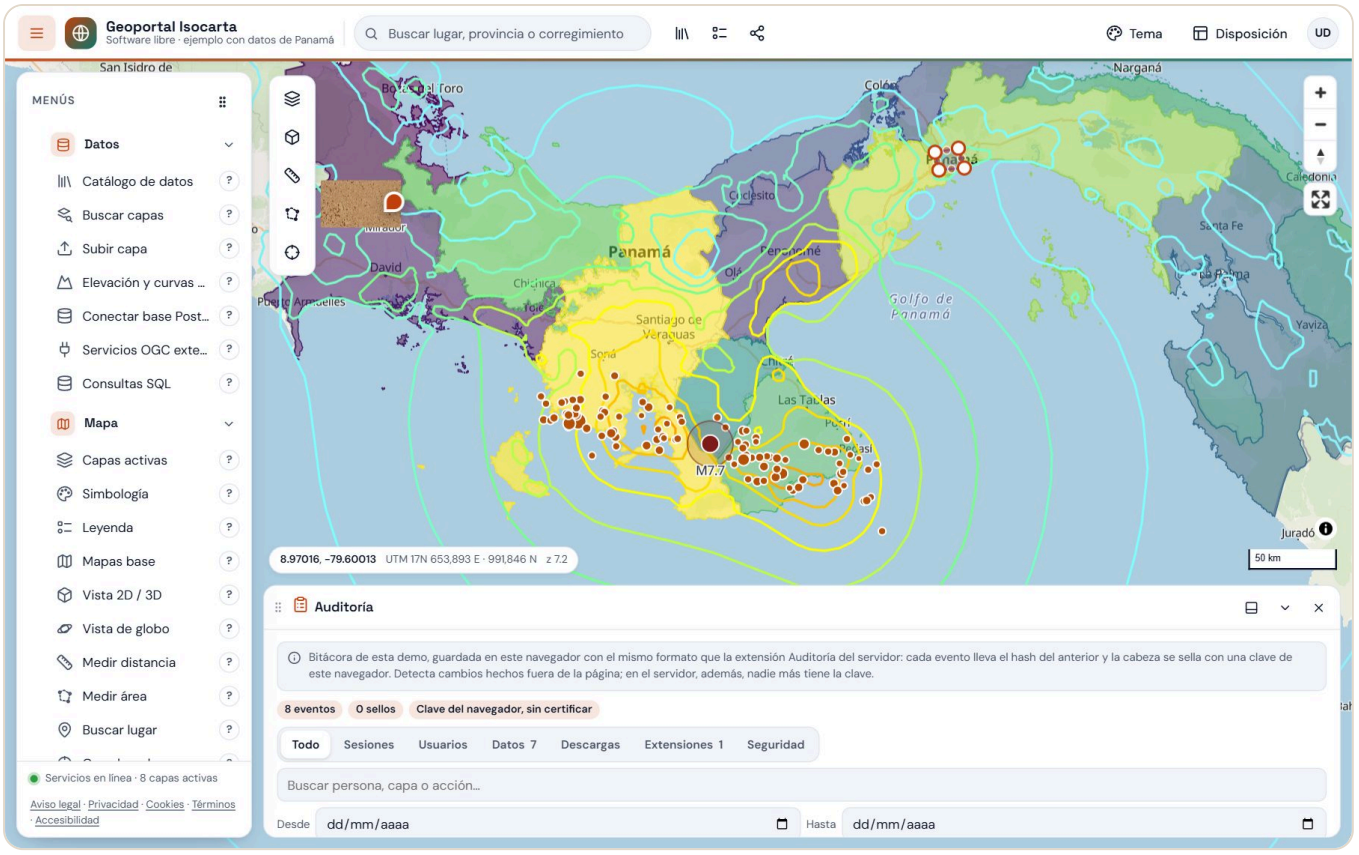
- el editor además sube capas, dibuja, edita y lleva el historial;
- el publicador además publica mapas web y servicios OGC;
- el administrador además gestiona su institución;
- el superadministrador puede usarlo todo.




Permisos por rol

**Auditoría.** Una bitácora encadenada: cada evento lleva la huella del anterior, así que una alteración se detecta.

- Registra sesiones, cambios de cuentas, capas subidas, descargas, mapas web, extensiones y cambios de seguridad.
- Se filtra por tipo y por fecha.
- Se exporta como **informe firmado**.



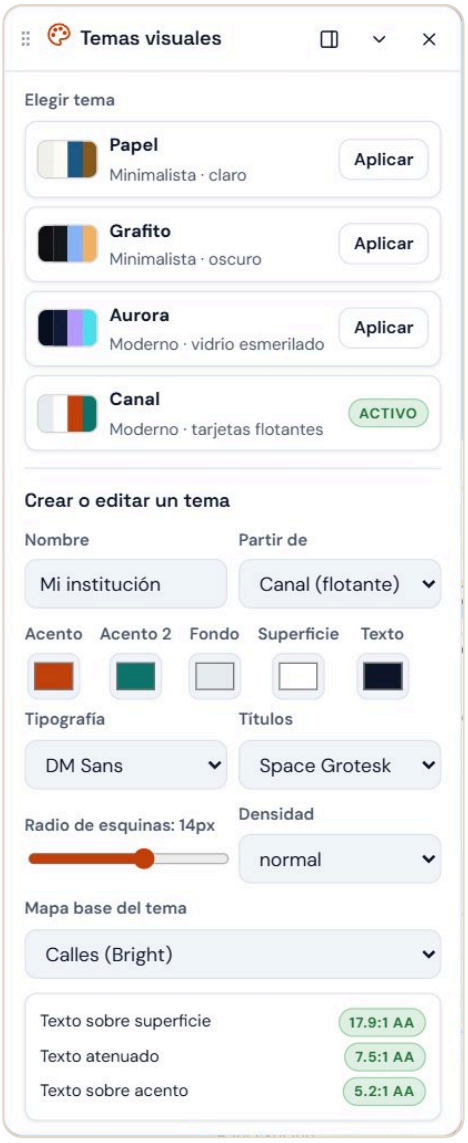

Auditoría

**Seguridad.** Un tablero que comprueba la configuración (conexión cifrada, cabeceras, contraseñas, doble factor, bloqueo, caducidad de sesión, auditoría, extensiones) y dice qué está bien y qué conviene revisar. Fija la política, por ejemplo exigir doble factor a los administradores.




Seguridad

**Temas visuales.** Cuatro temas listos (Papel, Grafito, Aurora y Canal), o uno propio: colores, tipografía, esquinas, densidad y mapa base. Comprueba el contraste para que el tema sea accesible y se exporta en JSON.




Temas visuales

**Información legal del sitio.** Cada sitio que use el geoportal debe identificar a su titular y explicar qué hace con los datos de las personas. Este formulario llega **prellenado**:

- con lo que ya está publicado en `config.json` (bloque `legal`);
- con lo que la plataforma sabe de sí misma: qué guarda en el navegador, a qué servicios de terceros llama (mapas, búsqueda de lugares, modelos de elevación, tipografías) y si hay cuentas en un servidor;
- con la ley de datos del país elegido (Costa Rica, Panamá, Colombia, España, México, El Salvador u otro): la ley, la autoridad ante la que se reclama y los derechos que se pueden ejercer.

Basta con completar el titular: nombre o razón social, identificación (cédula jurídica, RUC, NIF…), domicilio, correo y, si es una empresa, su inscripción registral. El panel dice cuántos datos faltan y hace una **revisión de cumplimiento**: aviso legal, privacidad, cookies, términos, comercio, accesibilidad (idioma, enlace para saltar al contenido, textos alternativos, contraste del tema), patrones oscuros, licencias, seguridad (HTTPS, contraseñas, contacto para vulnerabilidades), contenido de usuarios, publicidad y menores. Lo que puede comprobarse desde la página se comprueba solo.

- **Vista previa** muestra los textos con lo que falta resaltado. Sólo la ve quien la pide.
- **config.json** descarga la configuración con el bloque `legal`, lista para el servidor.
- **Textos (HTML)** descarga los nueve documentos en una página suelta, para un sitio web que no use el geoportal.
- **security.txt** descarga el archivo que va en `/.well-known/` con el contacto para reportar vulnerabilidades.
- Si el sitio usa analítica, se declara aquí. Sólo se carga después de que la persona la acepte, y el aviso ofrece «Rechazar» con el mismo peso que «Aceptar».

> Los textos son plantillas que se adaptan a cada sitio. Antes de publicarlos, hágalos revisar por asesoría legal.

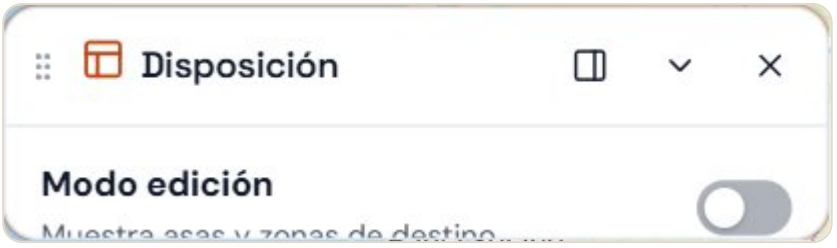



Información legal del sitio: datos del titular y revisión de cumplimiento

**Disposición.** Reorganiza la interfaz arrastrando:

- opciones entre menús;
- opciones hacia las barras;
- menús enteros;
- paneles de un lado a otro.

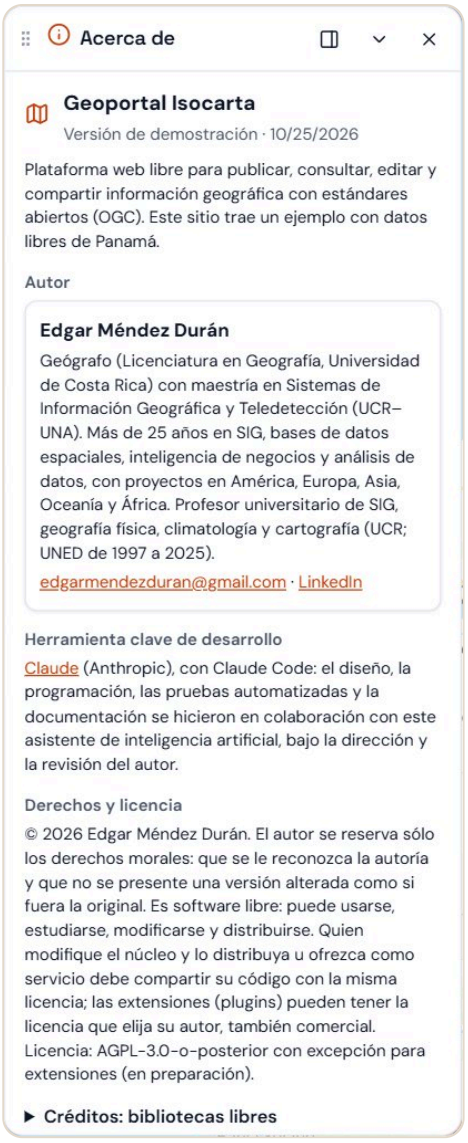
Cada disposición se guarda con un nombre, se restablece o se exporta.




Disposición

## 5.7 Menú Ayuda

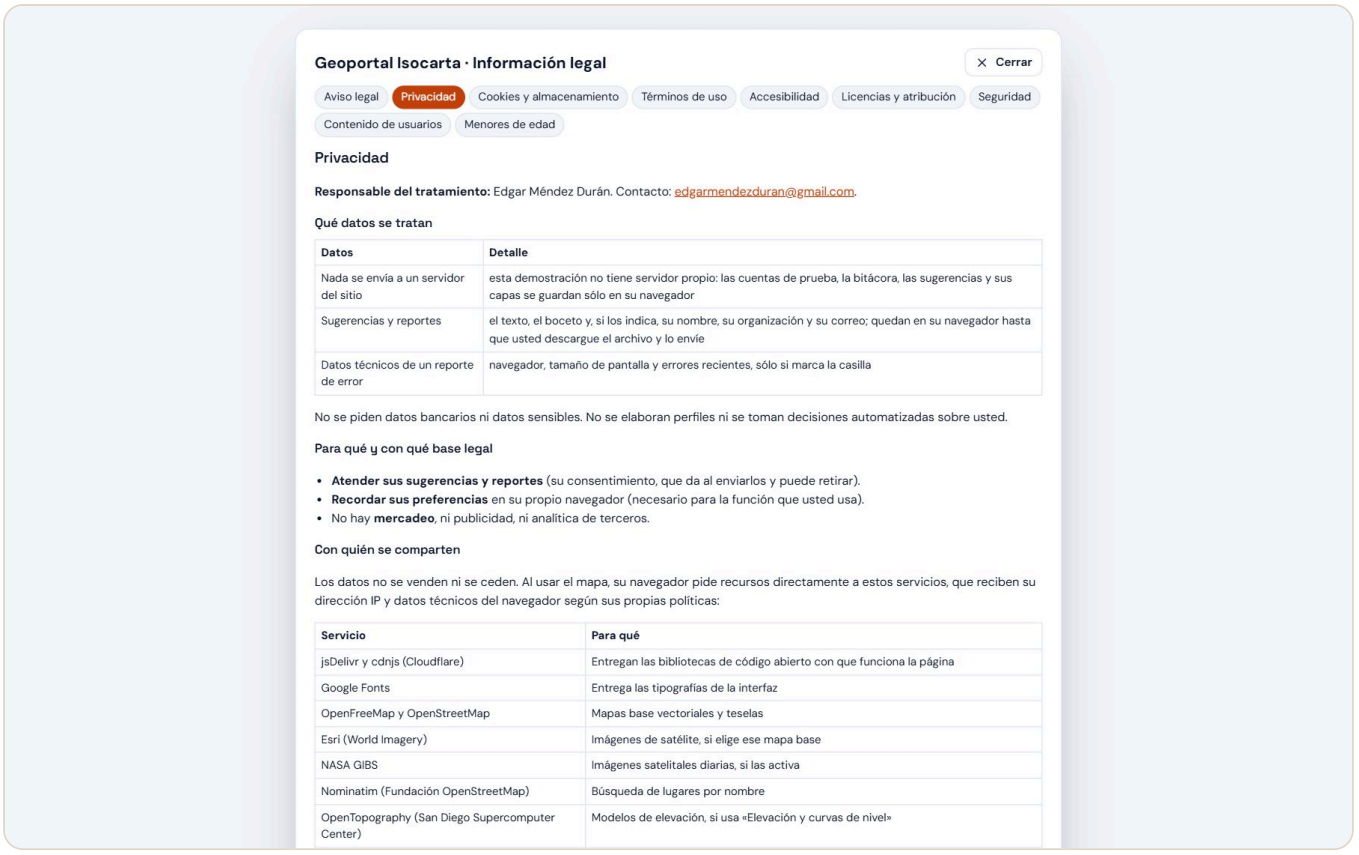
- **Sugerencias y errores**: la sección 4.
- **Manual de usuario**: este documento, con el video. Es público: también se abre sin cuenta desde el botón *Manual* de la portada, junto a *Inicio*, *Acerca del proyecto* y *Acerca del software*.
- **Información legal**: aviso legal, privacidad, cookies y almacenamiento, términos de uso, accesibilidad, licencias y atribución, seguridad, normas para el contenido de usuarios y menores de edad. Se abren también sin iniciar sesión: desde los enlaces al pie de los menús y de la pantalla de acceso, desde los mapas web publicados y desde la dirección `?legal=privacidad` (o `aviso`, `cookies`, `terminos` …).
- **Acerca de**: el autor, la herramienta con que se desarrolló, la licencia y los créditos de las bibliotecas y los datos.




Acerca de




El botón Manual en la portada pública




La política de privacidad, abierta sin iniciar sesión

La primera vez, un aviso al pie explica qué guarda el sitio en su navegador. Este geoportal no usa cookies de publicidad ni de analítica. Si un sitio las agrega, el aviso le pide permiso antes y puede cambiar su elección en *Información legal → Cookies*.

# 6. ¿Qué es un plugin?

Un **plugin** (o *extensión*) es un complemento que añade una función a Isocarta sin tocar su núcleo. Por ejemplo:

- una calculadora topográfica;
- un análisis de cuencas;
- un conector con un sistema de su institución;
- un reporte especializado.

En Isocarta, un plugin es un paquete `.ipdeplug` que lleva:

- su página web;
- un **manifiesto** que declara qué permisos pide: centrar el mapa, leer capas, agregar capas o descargar archivos;
- la **firma** de su editor.

La plataforma lo verifica antes de instalarlo y lo ejecuta aislado: un plugin no ve la sesión ni los datos que no pidió.

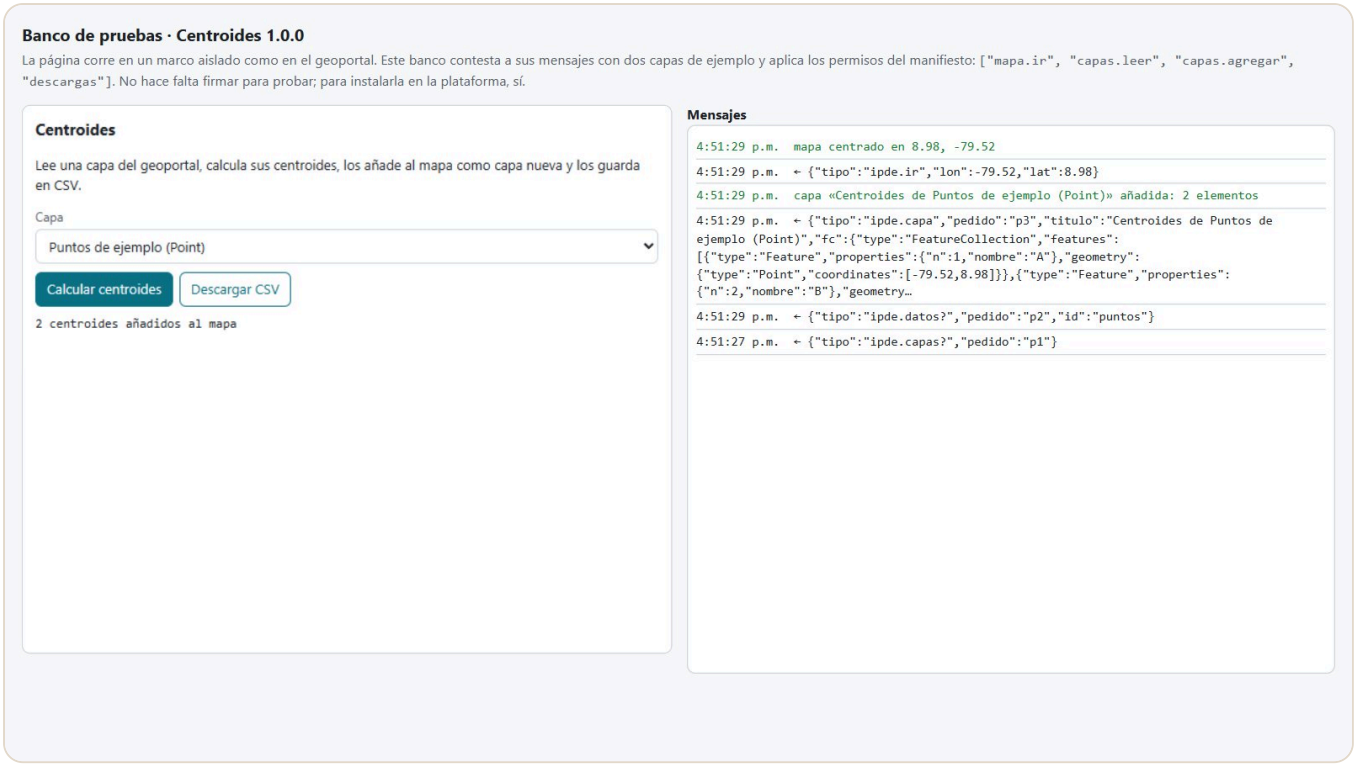
**Usted decide cómo distribuirlo.** Al empaquetarlo, elige en `[distribucion]`:

| Modo | Para qué | Qué hace Isocarta |
|---|---|---|
| **Código libre en el catálogo** | compartirlo con todos, con una licencia libre (MIT, Apache-2.0, GPL…) | lo anuncia en el catálogo de extensiones y, si es gratuito, puede alojarlo para instalarlo con un clic |
| **Comercial** | venderlo o cobrar su uso ( `pago` , `suscripcion` ), con licencia propietaria | lo anuncia, si usted quiere, con el botón **Adquirir** que lleva a su tienda o su sitio; la venta ocurre allí |
| **Privado** | uso interno de su organización o de un cliente | sólo verifica la firma al instalarlo |

El código y los derechos de un plugin son de quien lo desarrolla. La licencia de Isocarta **no obliga a abrir el código de un plugin**: sólo obliga a compartir los cambios que se hagan al núcleo (ver la sección 9).

**Descargar el SDK.** El kit de desarrollo (SDK) está en *Extensiones → ¿Cómo crear un plugin propio? → Descargar el SDK*. Es un solo programa en Python, `ipdeplug.py` , con la guía completa. Con él puede:

```
python ipdeplug.py crear mi_plugin --plantilla capas    crea el proyecto desde una
python ipdeplug.py probar mi_plugin                     lo prueba sin firmar en ur
python ipdeplug.py editor-solicitar --id ... --nombre "..."   pide su certificado
python ipdeplug.py empaquetar mi_plugin --editor ...    lo firma y crea el .ipdepl
python ipdeplug.py verificar mi_plugin-1.0.0.ipdeplug   lo comprueba como lo hará
```




El banco de pruebas del SDK: el plugin funciona sin firmar, con capas de ejemplo y el registro de mensajes

Su clave privada de editor **nunca sale de su equipo**: la plataforma sólo recibe su clave pública y le devuelve un certificado.

Las secciones 6.1 a 6.9 son la referencia técnica para quien escribe el plugin: el manifiesto, los mensajes con el mapa, la firma y la verificación.

## 6.1 Requisitos y proyecto

- **Python 3.11** o posterior (con 3.9 o 3.10 hace falta además `pip install tomli`) y el paquete `cryptography` (`pip install cryptography`).
- El SDK: `ipdeplug.py` y `raiz_publica.txt`, en la misma carpeta.
- Un navegador para el banco de pruebas.

Todas las órdenes tienen ayuda: `python ipdeplug.py --help` o `python ipdeplug.py ORDEN --help`.

`crear` tiene dos plantillas: `pagina`, el mínimo, centra el mapa en unas coordenadas; `capas` lee una capa, calcula sus centroides, los añade al mapa y los descarga en CSV, y usa los cuatro permisos. El

proyecto queda así:

```
mi_plugin/                  abra esta carpeta en VS Code (code mi_plugin)
  AGENTS.md                 reglas del puente para asistentes de IA; CLAUDE.md la i
  CLAUDE.md
  .vscode/extensions.json   extensiones recomendadas del editor
  .gitignore                deja fuera la clave del editor y los paquetes generados
  LEEME.md
  paquete/                  lo único que se firma y se distribuye
    extension.toml          el manifiesto
    web/index.html          la página que se abre en el geoportal
```

En `paquete/` va sólo lo que irá en el paquete: la firma cubre cada archivo y la verificación rechaza cualquier archivo añadido después. `empaquetar` se niega a firmar si encuentra ahí `.git`, `__pycache__`, `node_modules` o archivos del sistema. El control de versiones (git) va en la carpeta del proyecto.

## 6.2 El manifiesto extension.toml

```
id = "mi_plugin"              # letras, números, - y _; no se puede cambiar después
nombre = "Mi plugin"
version = "1.0.0"             # X.Y.Z; súbala en cada entrega
descripcion = "Qué hace, en una frase."
editor = "su_id_de_editor"    # opcional; si lo pone, debe ser el de su certificad
nivel = "opcional"            # los de terceros, siempre "opcional"
requiere = []                 # ids de otros plugins que deben estar habilitados
permisos = ["mapa.ir", "capas.leer"]

[distribucion]
modo = "catalogo"             # catalogo | independiente | privada
licencia = "MIT"              # MIT, Apache-2.0, GPL-3.0-or-later, EUPL-1.2, Propi
precio = "gratuita"           # gratuita | pago | suscripcion | consultar
url = "https://su-sitio.com/mi_plugin"   # dónde se adquiere o se descarga
soporte = "soporte@su-sitio.com"         # opcional: correo o https://

[[instalar]]
accion = "mensaje"
texto = "Instalando Mi plugin"

[[habilitar]]
accion = "copiar"
desde = "web"
hacia = "{interfaz}/extensiones/mi_plugin"

[[deshabilitar]]
```

```
accion = "borrar"
ruta = "{interfaz}/extensiones/mi_plugin"

[[desinstalar]]
accion = "mensaje"
texto = "Mi plugin retirado"
```

`[distribucion]` es obligatoria. Los tres modos de la tabla de arriba se escriben `catalogo`, `independiente` (el comercial, que usted vende o entrega en su sitio) y `privada`. La `url` es obligatoria salvo en `privada` y empieza por `https://`.

**Acciones.** Al instalar se ejecuta `[[instalar]]` y luego `[[habilitar]]`. Los pasos posibles son:

| Paso | Campos | Uso |
|---|---|---|
| `copiar` | `desde`, `hacia` | publicar la página: copiar `web` a `{interfaz}/extensiones/<id>` |
| `borrar` | `ruta` | retirarla al deshabilitar |
| `mensaje` | `texto` | dejar constancia en el resultado |
| `sql` | `archivo` | ejecutar un `.sql` en la base de la plataforma (sólo en la edición con servidor) |
| `orden` | `linea` | desactivado en la plataforma; no lo use |

No se admiten rutas absolutas ni `..`. Un paso `sql` tiene todos los permisos sobre la base: el administrador lo revisa antes de instalar y puede rechazarlo si no hace falta. `python ipdeplug.py validar mi_plugin` revisa el manifiesto sin firmar.

## 6.3 La página y el puente con el mapa

La página es un solo `web/index.html` autocontenido. El geoportal la abre en un marco `sandbox="allow-scripts"`, así que:

- no ve los demás archivos del paquete: el CSS y el JavaScript van en línea, y las imágenes como `data:`;
- puede cargar bibliotecas desde un CDN con CORS (cdnjs.cloudflare.com, cdn.jsdelivr.net, unpkg.com);
- no tiene `localStorage`, cookies ni ventanas emergentes, y no descarga archivos por sí misma (para eso está el permiso `descargas`);
- se adapta al tema claro u oscuro con `prefers-color-scheme`.

Los permisos que puede declarar el manifiesto:

| Permiso | Qué puede hacer el plugin |
|---|---|
| `mapa.ir` | centrar el mapa en un punto (todos lo tienen) |
| `capas.leer` | ver la lista de capas del catálogo y del mapa, y leer sus datos en GeoJSON (WGS 84) |
| `capas.agregar` | añadir una capa propia al mapa; queda en el Historial como cualquier capa subida |
| `descargas` | guardar en el equipo del usuario un archivo que genera el plugin |

Las plantillas traen el cliente del puente (el objeto `ipde`). Cada pedido lleva un número y la interfaz contesta con el mismo número o con `ipde.error`:

```
const capas = await ipde.capas();                  // [{id, titulo, grupo, geometri
const fc = await ipde.datos(capas[0].id);          // FeatureCollection en WGS 84
await ipde.agregarCapa('Resultado', fc);           // {id, elementos}
await ipde.descargar('resultado.csv', texto, 'text/csv');
ipde.ir(-79.52, 8.98, 14);                         // lon, lat, zoom (sin respuesta
```

Por debajo son estos mensajes:

| Envía el plugin | Permiso | Responde la interfaz |
|---|---|---|
| `{tipo:'ipde.ir', lon, lat, zoom}` | `mapa.ir` | nada: centra el mapa y pone un marcador |
| `{tipo:'ipde.capas?', pedido}` | `capas.leer` | `{tipo:'ipde.capas', pedido, capas:[…]}` |
| `{tipo:'ipde.datos?', pedido, id}` | `capas.leer` | `{tipo:'ipde.datos', pedido, id, fc}` |
| `{tipo:'ipde.capa', pedido, titulo, fc}` | `capas.agregar` | `{tipo:'ipde.capa.ok', pedido, id, elementos}` |
| `{tipo:'ipde.descargar', pedido, nombre, texto, tipoMime}` | `descargas` | `{tipo:'ipde.descargar.ok', pedido}` |
| `{tipo:'ipde.tamano', grande}` | — | nada: amplía o reduce el marco |

Si falta el permiso o algo sale mal, la respuesta es `{tipo:'ipde.error', pedido, error}`. Límites: una capa añadida tiene hasta 100 000 elementos, con geometrías en grados; un archivo descargado y el paquete descomprimido pesan hasta 50 MB cada uno. Las capas que sólo se sirven como imagen (WMS) no tienen datos: `ipde.datos?` responde con un error.

## 6.4 El certificado de editor

Se pide una sola vez por editor, no por plugin. `editor-solicitar` crea la carpeta `editor_<id>/` con dos archivos:

- `publisher_private.pem`, la clave privada: no la envíe ni la suba a un repositorio, y guarde una copia de seguridad cifrada. Quien la tenga puede firmar en su nombre.
- `solicitud.json`: su identificador, su nombre, su clave pública y una prueba de que posee la privada. No tiene nada secreto.

Envíe `solicitud.json` al administrador de la plataforma; puede acompañarla de una propuesta desde *Sugerencias → Función nueva → Extensión propia (plugin)*, con la dirección donde se adquirirá. Recibirá `publisher.cert.json`: guárdelo junto a la clave privada. Dura lo que decida la autoridad, dos años por omisión.

- **Si vence,** sus paquetes dejan de instalarse: pida otro antes de esa fecha (con la misma clave basta reenviar `solicitud.json`) y vuelva a firmar.
- **Si pierde la clave o se la roban,** avise de inmediato a la plataforma, cree una clave nueva en otra carpeta y pida un certificado nuevo.

## 6.5 Firmar y publicar

`empaquetar` revisa el manifiesto, calcula el SHA-256 de cada archivo, firma la lista con su clave, escribe `META/` y crea el ZIP. Al final muestra el SHA-256 del paquete: publíquelo junto a la descarga para que se pueda comprobar. Después de firmar no toque la carpeta: cualquier cambio, aunque sea un espacio, invalida la firma. Para corregir algo, edite, suba la versión y vuelva a empaquetar.

**Precio.** La plataforma no cobra ni procesa pagos: la venta y las licencias de uso ocurren en su sitio. Si quiere limitar el uso a quien pagó, hágalo en su propio servicio, por ejemplo con una clave de licencia que el plugin compruebe contra su servidor.

**Entrar al catálogo.** Genere la ficha y envíela al administrador con el paquete:

```
python ipdeplug.py ficha mi_plugin-1.0.0.ipdeplug -o ficha.json
```

La ficha resume el paquete verificado (id, versión, editor, licencia, precio, permisos, url y SHA-256) y el administrador la añade a `extensiones/indice.json`. Si la plataforma no aloja el paquete, la ficha va

sin archivo y el catálogo sólo enlaza a su sitio.

## 6.6 Instalación, versiones y seguridad

El administrador instala el paquete en *Administrar → Extensiones → Instalar .ipdeplug* (el navegador verifica la firma, el certificado y cada archivo antes de guardarlo) o, en la edición con servidor, con `consola extension instalar mi_plugin-1.0.0.ipdeplug`. Si algo no cuadra, el paquete se rechaza con uno de los mensajes de la sección 6.8 y el intento queda en la auditoría.

- **Versiones:** 1.0.1 para correcciones, 1.1.0 para funciones nuevas y 2.0.0 para cambios que rompen. Una versión nueva reemplaza la anterior y la auditoría guarda el paso. El `id` no cambia nunca: con otro id sería otro plugin.
- Si cambia los permisos, dígalo en las notas de versión: el administrador verá los nuevos en el panel.
- **Qué garantiza la firma:** quién hizo el paquete y que nadie lo cambió después; no que el plugin sea correcto ni inocuo. Por eso la página corre aislada con sus permisos, los pasos `orden` están desactivados, los `sql` los revisa el administrador y se rechaza un certificado vencido o no firmado por la raíz de la plataforma. Pida sólo los permisos que use.

## 6.7 Para la autoridad de la plataforma

**Emitir un certificado.** Primero se comprueba la identidad del solicitante por un canal propio (correo de su dominio, contrato o llamada). Después:

```
python ipdeplug.py editor-emitir solicitud.json --raiz CARPETA_DE_LA_RAIZ --anios
```

Antes de firmar, la orden comprueba que la solicitud esté firmada con la clave que trae, que la raíz sea la de esta plataforma, que el id no lo use ya otro editor con otra clave (`--reemplazar` sólo si es la misma organización que perdió su clave) y que el nombre no tenga saltos de línea. Se devuelve `salida/publisher.cert.json`; la copia en `--registro` es el inventario de editores.

La clave raíz (`root_private.pem`) se guarda fuera de línea y fuera de todo repositorio. Si se compromete, hay que crear otra raíz y recompilar la consola y el geoportal con la nueva clave pública, y todos los certificados y paquetes anteriores dejan de valer.

**Retirar a un editor.** No hay lista de revocación: un certificado vale hasta que vence, por eso conviene emitirlos por uno o dos años. Ante un caso grave: quitar sus paquetes del catálogo, avisar a los administradores para que los desinstalen y no renovarle el certificado.

El catálogo es `extensiones/indice.json`. Una entrada con `archivo` (sólo plugins gratuitos alojados por la plataforma) se instala con un clic; sin `archivo`, enlaza a la `url` del editor:

```
{"paquetes": [
  {"archivo": "conversor.ipdeplug", "id": "conversor", "nombre": "Conversor de coc
  {"id": "mi_plugin", "nombre": "Mi plugin", "version": "1.0.0", "descripcion": "…
   "licencia": "Propietaria", "precio": "pago", "url": "https://su-empresa.com/mi_
]}
```

## 6.8 Mensajes de verificación

| Mensaje | Qué significa |
|---|---|
| SIN FIRMAR: falta META/… | el paquete no se firmó o se le quitó `META/` |
| el certificado del editor no está firmado por la raíz de esta plataforma | certificado de otra plataforma, de prueba o falsificado |
| el certificado del editor venció el … | pida un certificado nuevo y vuelva a firmar |
| la firma del paquete no valida con la clave del editor | se alteró `META/hashes.json` o se firmó con otra clave |
| «archivo» fue modificado después de firmar | se cambió un archivo tras empaquetar |
| archivo no declarado en la firma / falta un archivo declarado | se añadió o se quitó un archivo |
| id del manifiesto ≠ firma / versión del manifiesto ≠ firma | se editó `extension.toml` después de firmar |
| el manifiesto declara otro editor | el `editor` de `extension.toml` no es el del certificado |
| entrada de ZIP inválida | rutas absolutas o con `..` (se rechazan para no escribir fuera de la carpeta) |

## 6.9 Formato del paquete .ipdeplug

```
clave raíz        --firma⟶  META/publisher.cert.json   (certificado del editor)
clave del editor  --firma⟶  META/hashes.json           (bytes crudos del archivo
META/hashes.json  --cubre⟶  SHA-256 de cada archivo fuera de META/
```

- ECDSA P-256 con SHA-256. Las firmas van en DER y las claves públicas en SPKI DER, ambas en base64.

- El certificado firma los bytes UTF-8 de `publisher_id\nname\npubkey\nissued\nexpires`, no el JSON.
- `META/hashes.json` es `json.dumps({"alg":"SHA-256","files":{…}}, indent=2, sort_keys=True, ensure_ascii=False)`, sin salto de línea final. `META/signature.json` lleva `alg`, `publisher_id`, `plugin_id`, `version`, `signed_at` y `sig`.
- El ZIP lleva una carpeta raíz `<id>/`; el `id` y la `version` de `extension.toml` deben coincidir con `plugin_id` y `version` de la firma.
- `ipdeplug.py`, `consola firmar` y el panel Extensiones producen y verifican exactamente lo mismo.

# 7. Entorno de desarrollo recomendado

Recomendamos **Visual Studio Code** (VS Code): es gratuito, funciona en Windows, macOS y Linux, y edita HTML, JavaScript, Python y TOML en un mismo sitio. El proyecto que crea el SDK trae la lista de extensiones recomendadas; VS Code propone instalarlas al abrir la carpeta ( `code mi_plugin` ):

- **Python**: ejecutar y depurar el SDK.
- **Even Better TOML**: editar el manifiesto con colores y avisos de error.
- **ESLint** y **Prettier**: errores y formato del JavaScript.
- **Claude Code**: el asistente de IA dentro del editor.

Alternativas: **VSCodium** (el mismo editor, 100 % libre y sin telemetría), WebStorm o PyCharm. Para el ciclo de trabajo basta la terminal integrada:

1. Editar la página.
2. Correr `probar` .
3. Recargar el banco de pruebas.
4. Hacer el commit con git.

# 8. Usar la inteligencia artificial para crear plugins

Un asistente de IA acelera mucho el desarrollo: escribe la página, conecta el puente con el mapa, propone pruebas y explica errores.

El proyecto que crea el SDK ya viene preparado:

- `AGENTS.md` explica al asistente la estructura del proyecto, el marco aislado, los mensajes y permisos del puente, y las reglas de seguridad.
- `CLAUDE.md` lo importa para Claude Code.
- GitHub Copilot, Cursor y otros asistentes leen `AGENTS.md` directamente.

Abra el proyecto y pida lo que necesita en lenguaje natural, por ejemplo:

```
Quiero que el plugin calcule el área de cada polígono de la capa elegida en hectár
la muestre en una tabla ordenable y permita descargarla en CSV.
Usa sólo los permisos necesarios y pruébalo con «python ipdeplug.py probar .».
```

Buenas prácticas:

1. **Cambios pequeños y probados.** Pida un cambio y pruébelo en el banco antes de pedir el siguiente.
2. **Revise lo que propone.** Usted firma el paquete y responde por él.
3. **Nunca le entregue su clave privada.** Firme usted con `empaquetar` . El proyecto la deja fuera de git y `AGENTS.md` le prohíbe leerla.
4. **Seguridad de los datos.** Pida que escape los textos que vienen de las capas y que declare si envía datos a un servicio externo.
5. **Licencias.** Si la IA reutiliza código de terceros, incluya sus avisos de licencia.
6. **Memoria del proyecto.** Mantenga `AGENTS.md` al día con las reglas propias de su plugin.

# 9. Licencia y autoría

Isocarta es **software libre**: se puede usar, estudiar, modificar y distribuir. La licencia propuesta es **AGPL-3.0-o-posterior con una excepción para extensiones**:

- Quien **modifique el núcleo** y lo distribuya, o lo ofrezca como servicio en internet, debe **compartir su código** con la misma licencia.
- Los **plugins** que sólo usan la interfaz de extensiones pueden tener **cualquier licencia, también comercial**, sin abrir su código.
- El SDK y las plantillas son MIT.
- Se respetan las licencias de las bibliotecas y de los datos de terceros, que se listan en *Acerca de*.

**Autoría.** © 2026 Edgar Méndez Durán: desarrollador. José María Agüero: QA. Agricultura Científica S.A. (AgriCien): empresa desarrolladora. El autor se reserva únicamente los **derechos morales**: que se le reconozca la autoría y que no se presente una versión modificada como si fuera la original. No se reserva derechos patrimoniales más allá de lo que exige la licencia.

**Acerca del autor.** Edgar Méndez Durán es geógrafo (Licenciatura en Geografía, Universidad de Costa Rica), con maestría en Sistemas de Información Geográfica y Teledetección (UCR–UNA). Tiene más de 25 años de experiencia en SIG, bases de datos espaciales, inteligencia de negocios y análisis de datos, con proyectos en América, Europa, Asia, Oceanía y África. Es profesor universitario de SIG, geografía física, climatología y cartografía (UCR; UNED de 1997 a 2025). Contacto: edgarmendezduran@gmail.com.

**Acerca de José María Agüero (QA).** Fundador y presidente de Agricultura Científica S.A. (AgriCien) desde 1992, donde también fue director general y gerente general hasta 2024. Licenciado en Suelos (Universidad de Costa Rica), con formación en INCAE Business School. Tiene más de 34 años de experiencia en manejo del agua en la agricultura y la ganadería, nivelación de terrenos en 3D, producción de arroz en el trópico, agricultura de precisión y control de maquinaria, cosecha de agua, regeneración de suelos y desarrollo de proyectos agropecuarios. Contacto: jose.aguero@agricien.com.