



IDERA

Infraestructura de Datos Espaciales
de la República Argentina

Publicá mapas (y más servicios) con MVT Server

PostGIS · MVT Server · QGIS · MapLibre

José Jachuf — Helena Sangroniz

00

Apertura y contexto

¿QUÉ ES MVT SERVER?

- › **Servidor de Vector Tiles escrito en Rust**
Alta performance, bajo consumo de memoria, binario único sin dependencias
- › **Fuente de datos: PostgreSQL + PostGIS**
Lee directamente de tablas y vistas espaciales. Sin intermediarios.
- › **Produce teselas .pbf (Mapbox Vector Tile)**
Formato binario comprimido, estándar de facto en la industria
- › **Todo en un solo ejecutable**
Admin UI · estilos · leyendas · caché · monitor · filtros dinámicos · plugins



¿POR QUÉ VECTOR TILES?

Vector Tiles

- › Formato binario comprimido (protobuf)
- › Renderizado en el cliente
- › Un endpoint para todas las escalas
- › Simbología sin tocar el servidor
- › Compatible con QGIS, MapLibre, Leaflet

WMS / WFS

- › Imagen rasterizada por request
- › Descarga el dataset completo
- › Cada cambio de simbología = nuevo request
- › Escala mal con muchos usuarios concurrentes

01

Base de datos PostgreSQL + PostGIS

01 · CARGAR DATOS CON OGR2OGR bash

```
# Cargar geojson a PostgreSQL
ogr2ogr -overwrite -f "PostgreSQL" \
  PG:"host=localhost port=5432 dbname=mi_base_datos
  user=tu_usuario password=tu_contraseña" \
  "departamentos.json" -nln departamentos \
  -lco GEOMETRY_NAME=geom -nlt PROMOTE_TO_MULTI
```

02

Instalación

02 · DOS FORMAS DE INSTALAR

A) Binario precompilado

- › Descargar desde GitHub releases
- › Sin Rust ni compilador
- › Windows, MacOS, FreeBSD y Linux
- › Recomendado para el taller

B) Compilar desde fuente

- › Requiere Rust vía rustup
- › git clone + cargo build --release
- › Binario en target/release/mvt-rs
- › Para contribuir o modificar el código



02 · DESCARGAR EL BINARIO

A) Binario precompilado

- › Descargar desde: github.com/mvt-proj/mvt-rs/releases/latest
- › Ubicación: Guardar el archivo en un directorio llamado mvt-server (o similar).
- › Nota: Si la descarga se realiza automáticamente en la carpeta Descargas, recomendamos mover el binario a una carpeta dedicada para el proyecto (ej: mvt-server).

03

Configuración config.yaml

03 · DESCARGAR LA CONFIGURACIÓN DE EJEMPLO

A) Binario precompilado

› Descargar este archivo: <https://raw.githubusercontent.com/mvt-proj/mvt-rs/refs/heads/main/config.example.yaml>

› Ubicación: ¿Dónde guardarlo?

Abre la carpeta mvt-server que creaste antes.

Crea una nueva carpeta llamada config dentro de ella.

Guarda el archivo descargado dentro de esa carpeta config.

Ruta final esperada: **mvt-server\config\config.yaml**

03 · PUNTOS CLAVE DEL CONFIG.YAML

› **postgres_databases**

Admite múltiples entradas. Cada una con alias y connection string.

› **connection_string**

postgres://usuario:clave@host:5432/nombre_db — estándar PostgreSQL, soporta SSL.

› **security.secret_key**

Cambiar siempre en producción. Generar con: openssl rand -base64 32

› **Cómo cargar la config**

Por defecto: config/config.yaml · Argumento: --config ruta · Variable: MVT_CONFIG_PATH



Partir del config.example.yaml del repositorio y editar solo lo necesario.

03 · ESTRUCTURA DEL CONFIG.YAML yaml

```
server:
  host: 0.0.0.0
  port: 5800

postgres_databases:
  pool_min: 2
  pool_max: 5
  default: "postgres://user1:password1@host:port/db_gis1"
  catastro: "postgres://user2:password2@host2:5432/db_catastro"

database:
  sqlite_path: "mvtrs.db"
  #redis_url: "redis://localhost:6379"
  # omit to use disk cache instead
```

03 · ESTRUCTURA DEL CONFIG.YAML yaml

```
security:
  jwt_secret: "change-me-to-a-random-secret-at-least-64-chars-long"
  session_secret: "change-me-to-a-random-secret-at-least-64-chars-long"
  session_duration_minutes: 20

paths:
  config: "config"
  cache: "cache"
  assets: "map_assets"
  plugins: "plugins"
```

04

Arrancar el servidor

04 · ARRANCAR MVT SERVER bash

```
# Con config en la ruta por defecto
```

```
mvt-rs
```

```
# Con ruta explícita
```

```
mvt-rs --config /etc/mvt/config.yaml
```

```
# Log esperado:
```

```
Database initialized successfully.
```

```
2026-06-25T13:07:23Z INFO mvt_server::plugins: Plugins directory
```

```
'plugins' not found – no plugins load
```

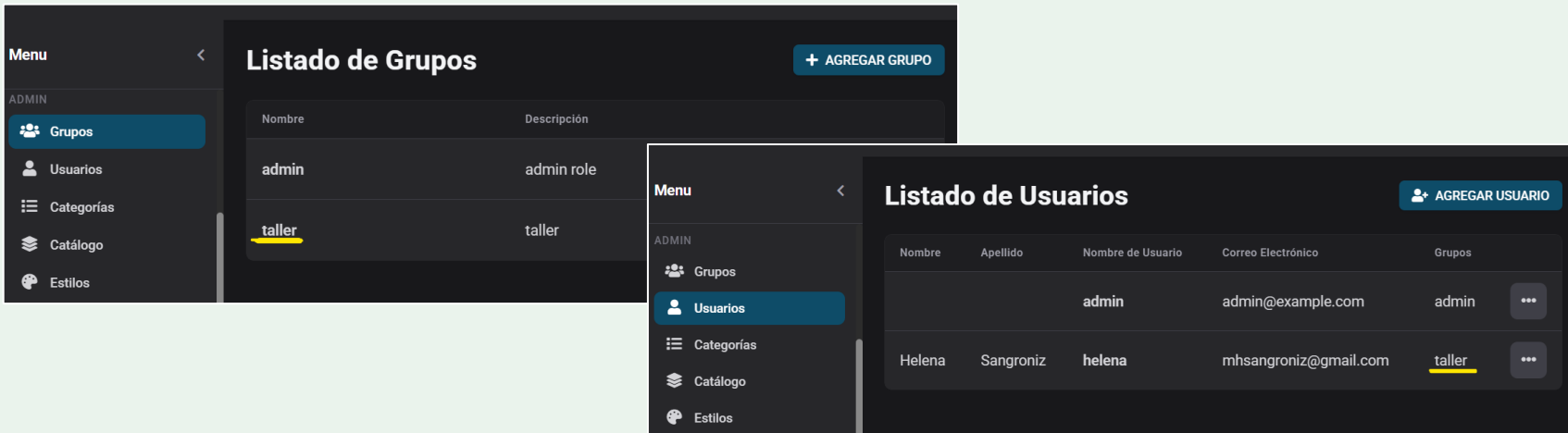
05

Interfaz de administración **Recorrida rápida**

05 · USUARIOS Y GRUPOS

Los Grupos definen roles y permisos: quién puede ver o editar cada capa y estilo, y quiénes pueden consumir los geoservicios.

Los Usuarios se asignan a un grupo. Solo quienes pertenecen al grupo 'admin' pueden administrar usuarios, categorías, catálogo y estilos.



Listado de Grupos

Nombre	Descripción
admin	admin role
taller	taller

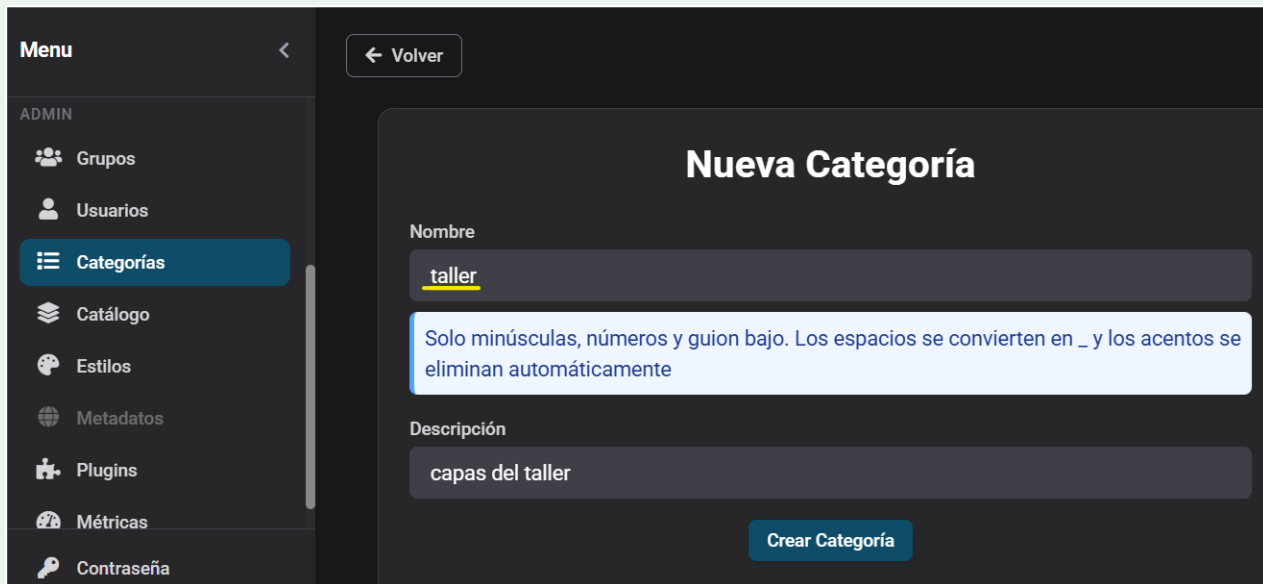
Listado de Usuarios

Nombre	Apellido	Nombre de Usuario	Correo Electrónico	Grupos
		admin	admin@example.com	admin
Helena	Sangroniz	helen	mhsangroniz@gmail.com	taller

05 · CATEGORÍAS

Las categorías funcionan como namespaces para organizar capas y estilos.

Aparecen en la URL del tile y facilitan mantener el catálogo ordenado cuando hay muchas capas publicadas.



The screenshot shows a web interface for creating a new category. On the left is a dark sidebar menu with the title 'Menu' and a back arrow. The menu items are: ADMIN, Grupos, Usuarios, Categorías (highlighted in blue), Catálogo, Estilos, Metadatos, Plugins, Métricas, and Contraseña. The main content area has a dark header with a 'Volver' button and a title 'Nueva Categoría'. Below the title are two input fields: 'Nombre' with the value 'taller' and a tooltip explaining the naming rules (lowercase, numbers, and underscores only; accents are removed), and 'Descripción' with the value 'capas del taller'. A blue 'Crear Categoría' button is at the bottom right.

Menu < Volver

ADMIN

- Grupos
- Usuarios
- Categorías**
- Catálogo
- Estilos
- Metadatos
- Plugins
- Métricas
- Contraseña

Nueva Categoría

Nombre

taller

Solo minúsculas, números y guion bajo. Los espacios se convierten en _ y los acentos se eliminan automáticamente

Descripción

capas del taller

Crear Categoría

05 · CATÁLOGO

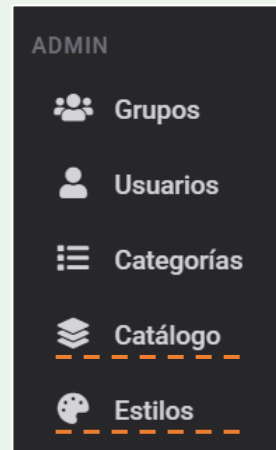
Es el corazón del panel: aquí se declaran y configuran las capas geográficas.

Se elige la base PostgreSQL, el schema y la tabla, y se ajustan parámetros

05 · ESTILOS

Se definen y publican estilos de renderizado para las capas.

Simples (una capa, pensados para QGIS) o proyectos MapLibre completos, con múltiples sources, layers y reglas por zoom.



06

Publicar la primera capa vector tile

06 · PASOS EN EL CATÁLOGO

1 Catálogo → Nueva capa

Elegir base PostgreSQL → schema → tabla

2 Seleccionar campos

Exponer sólo los atributos necesarios. Menos campos = tiles más livianos.

3 ZMin / ZMax

Calibrar según escala de datos: nacional 0–8 · provincial 6–12 · local 10–18

4 Caché TTL

0 = sin expiración (datos estáticos) · valor en segundos para datos dinámicos

5 Guardar y verificar

URL generada: /services/tiles/{categoría}:{nombre}/{z}/{x}/{y}.pbf



Botón 'Inspeccionar capa' en el Catálogo para previsualizar antes de publicar.

06 · FORMULARIO DE NUEVA CAPA

En Catálogo → Nueva capa se elige la base PostgreSQL, el schema y la tabla a publicar.

El campo 'Name' debe ser una sola palabra (preferentemente en minúsculas); 'Alias' admite una etiqueta más descriptiva. Se recomienda exponer solo los campos necesarios.



Menu

- ADMIN
- Grupos
- Usuarios
- Categorías
- Catálogo**
- Estilos
- Metadatos
- Plugins
- Contraseña
- Cerrar sesión

Nueva Capa

Categoría
taller

Geometría
Polígonos

Nombre
cob_uso_suelo_urbano
Solo minúsculas, números y guion bajo. Los espacios se se eliminan automáticamente

Alias
Cobertura Uso del Suelo Urbano

Descripción
Uso del suelo urbano

Base de Datos
Default

Esquema
sc_land_urbano

Tabla
lu_cordoba

Campos
gid
valor
categoria

Publicada
☒ Sí ☐ No

Grupos Permitidos
admin
taller

06 · PROBAR LA CAPA PUBLICADA

El botón 'Inspeccionar capa' permite verificar que la capa se publicó correctamente antes de consumirla desde QGIS o MapLibre.

Es la forma más rápida de validar la geometría, el rango de zoom y el estilo por defecto.

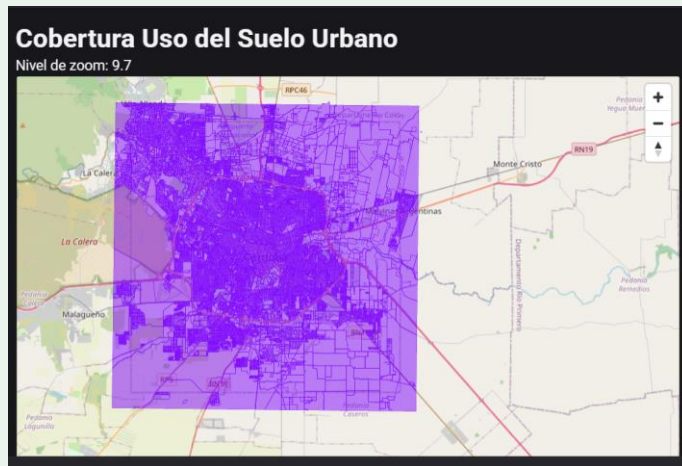
Catálogo de capas publicadas

Filtro... 🔍

PUBLICAR CAPA

Categoría	Base de Datos	Nombre de la Capa	Alias	Tabla		
	taller	Default	cob_uso_suelo_urbano	Cobertura Uso del Suelo Urbano	sc_land_urbano.lu_cordoba	⋮

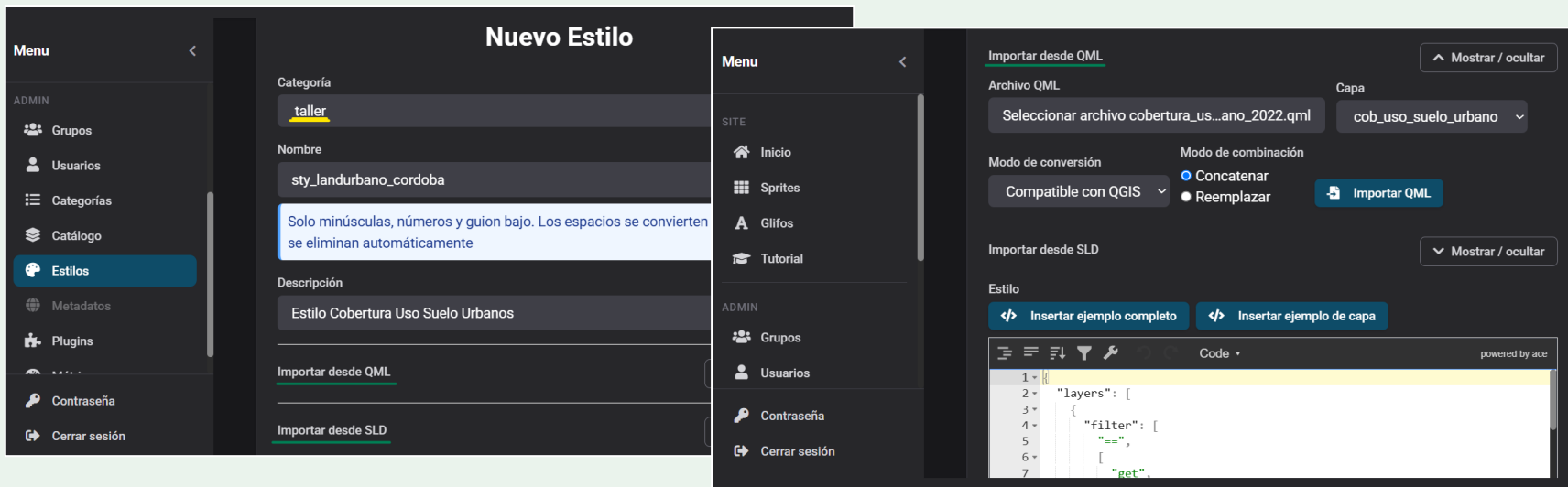
- Copiar
- Información
- Inspeccionar capa
- Publicar / Despublicar
- Eliminar Caché
- Editar
- Eliminar



06 · CARGAR Y ASIGNAR UN ESTILO

En la sección Estilos del panel se crea un estilo simple (una capa) o se sube un proyecto MapLibre completo, definiendo colores, opacidad y etiquetas.

Luego, en Catálogo → editar la capa, se asigna el estilo publicado en el campo correspondiente para que QGIS lo consuma automáticamente vía URL.



The image shows two overlapping screenshots of the IDERA web application interface.

Left Screenshot: 'Nuevo Estilo' (New Style) Form

- Menu:** ADMIN, Grupos, Usuarios, Categorías, Catálogo, Estilos (selected), Metadatos, Plugins, Contraseña, Cerrar sesión.
- Categoría:** taller
- Nombre:** sty_landurbano_cordoba
- Descripción:** Estilo Cobertura Uso Suelo Urbanos
- Importar desde QML:** Solo minúsculas, números y guion bajo. Los espacios se convierten se eliminan automáticamente
- Importar desde SLD:**

Right Screenshot: 'Importar desde QML' (Import from QML) Section

- Menu:** SITE (Inicio, Sprites, Glifos, Tutorial), ADMIN (Grupos, Usuarios, Contraseña, Cerrar sesión).
- Importar desde QML:** Archivo QML (Seleccionar archivo cobertura_us...ano_2022.qml), Capa (cob_uso_suelo_urbano), Modo de conversión (Compatible con QGIS), Modo de combinación (Concatenar, Reemplazar), Importar QML.
- Importar desde SLD:** Mostrar / ocultar.
- Estilo:** Insertar ejemplo completo, Insertar ejemplo de capa.
- Code Editor:** Code, powered by ace. The code shows a QML layer definition with a filter.

07

Consumir con QGIS

07 · AGREGAR CAPA VECTOR TILE EN QGIS

1 Panel Fuentes → Vector Tiles → Nueva conexión genérica

2 Pegar la URL de la capa publicada

`http://localhost:5800/services/tiles/taller:{z}/{x}/{y}.pbf`

3 Agregar al proyecto y ajustar simbología

Click derecho → Propiedades → Simbología. Colores, grosor, relleno.

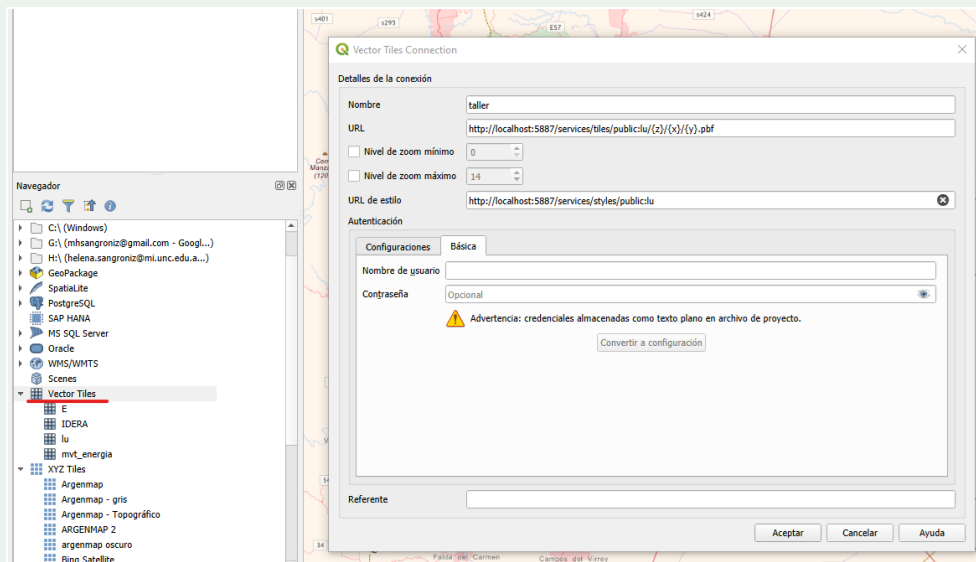
4 URL de estilo (opcional)

Si hay un estilo publicado en MVT Server, pegarlo en el campo 'Estilo'

07 · NUEVA CONEXIÓN VECTOR TILES EN QGIS

En el panel Fuentes de QGIS, clic derecho sobre 'Vector Tiles' → 'Nueva conexión genérica'.

Se asigna un nombre a la conexión, se pega la URL de la capa publicada y, opcionalmente, la URL del estilo. Con 'Aceptar' la conexión queda disponible para agregar al proyecto.



08

Consumir con MapLibre

08 · MAPLIBRE — HTML MÍNIMO json

```
map.on('load', () => {  
  map.addSource('src_departamentos', {  
    type: 'vector',  
    tiles:  
    ['http://127.0.0.1:5887/services/tiles/public:departamentos/{z}/{x}/{y}.pbf'],  
  });  
  map.addLayer({  
    id: 'departamentos-fill', type: 'fill',  
    source: 'src_departamentos', 'source-layer': 'src_departamentos',  
    paint: { 'fill-color': '#167025', 'fill-opacity': 0.7 }  
  });  
});
```

09

Crear y publicar estilos

09 · DOS TIPOS DE ESTILOS

Estilo simple (capa)

- › Para una sola capa
- › Color, opacidad, labels básicos
- › Se asigna en el Catálogo
- › QGIS lo consume vía URL
- › Ideal para compartir simbologías

Proyecto MapLibre (completo)

- › MapLibre Style Specification
- › Múltiples sources y layers
- › Reglas por zoom level
- › `map.setStyle(url)` en MapLibre
- › Glifos y sprites desde MVT Server

09 · STYLE MAPLIBRE (ESTRUCTURA) json

```
{
  "version": 8,
  "sources": {
    "capa": { "type": "vector",
      "tiles": ["http://localhost:5800/services/tiles/taller:nombre/{z}/{x}/{y}.pbf"] }
  },
  "layers": [
    { "id": "fill", "type": "fill", "source": "capa",
      "source-layer": "nombre",
      "paint": { "fill-color": "#167025", "fill-opacity": 0.7 } }
  ]
}
```



Casos de uso

+ CASOS DE USO

- › **Filtros dinámicos en la URL**

?estado=activo · ?tipo__in=1,2,3 — sin tocar la base de datos

- › **Leyendas SVG automáticas**

El servidor genera SVG a partir del estilo. Integrable en apps web.

- › **Control de acceso por capa**

Autenticación Basic o JWT. Cada capa puede ser pública o restringida.

- › **Monitor en tiempo real**

/admin/monitor/dashboard — RPS, latencia P95, cache hits/misses en vivo

- › **Nginx como reverse proxy + gzip**

Compresión de tiles .pbf: ahorro del 60–80% de ancho de banda



IDERA

Infraestructura de Datos Espaciales
de la República Argentina

¡Gracias!

github.com/mvt-proj/mvt-rs

Preguntas · Consultas · Demo libre