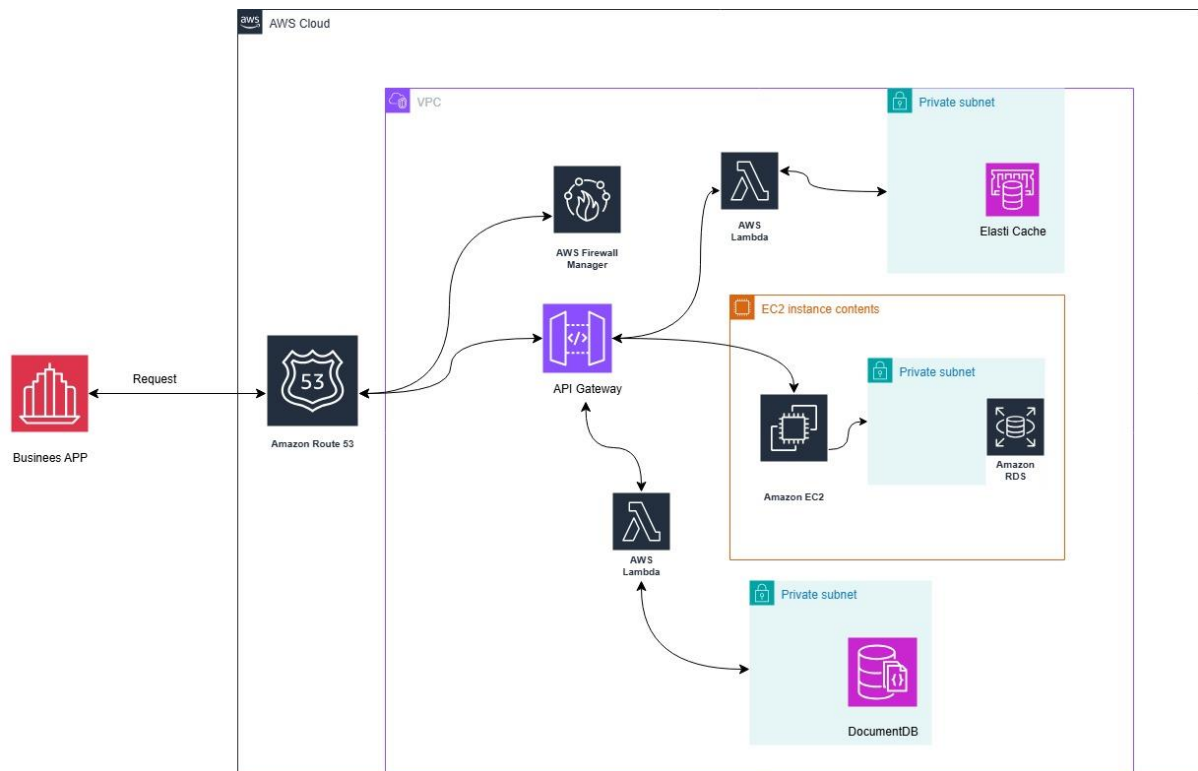


## Introducción al Proyecto: *Base de Datos Geoespacial*

En un mundo cada vez más interconectado, la **inteligencia de ubicación** se ha vuelto clave para aplicaciones modernas que requieren entender no solo **qué** es algo, sino **dónde** se encuentra y cómo se relaciona con su entorno.

Las bases de datos geoespaciales representan una evolución de los sistemas tradicionales de almacenamiento de datos (RDBMS), integrando tipos de datos espaciales (como puntos, líneas y polígonos), índices especializados y funciones geoespaciales avanzadas para consultar, analizar y visualizar información geográfica con gran eficiencia.

Este proyecto de **“Base de Datos Geoespacial”** tiene como objetivo diseñar una solución robusta y escalable para el manejo de jerarquías territoriales, desde provincias hasta barrios, así como entidades como hospitales. El enfoque combina tecnologías de alto desempeño: bases relacionales con PostGIS para consultas espaciales complejas, almacenamientos NoSQL para estructuras JSON jerárquicas y caché en memoria para acceso instantáneo.



## Flujo de Solicitud (Request)

### 1. Business APP

Punto de origen de las solicitudes. Puede ser una aplicación web o móvil que requiere datos geoespaciales o información de negocio.

### 2. Amazon Route 53

Sistema de nombres de dominio (DNS) que recibe la solicitud, ubica la mejor región AWS por latencia y la redirige dentro del VPC. Se puede configurar con balanceo según ubicación.

## Control de Seguridad de Entrada

### **AWS Firewall Manager**

Aplica reglas y políticas centralizadas para proteger los recursos: bloquea tráfico no deseado, filtra ataques y administra configuraciones de seguridad en API y VPC.

## Entrada a Servicios REST

### **4. API Gateway**

- a. Recibe tráfico HTTP/HTTPS
- b. Maneja autenticación (JWT, API keys), límites (throttling), CORS y validaciones
- c. Dirige las solicitudes a funciones Lambda o instancias EC2 según el endpoint

## Cómputo Serverless

### **5. AWS Lambda (varias funciones)**

- a. **Lambda superior derecha:** consulta rápida en ElastiCache (Redis), devuelve respuestas geoespaciales cacheadas
- b. **Lambda inferior:** ingresa o consulta documentos JSON en DocumentDB
- c. Escala automáticamente según demanda

## Almacenamiento en Caché

### **6. Amazon ElastiCache (Redis)**

- a. Desplegado en subred privada
- b. Guarda en memoria respuestas frecuentes en GeoJSON para acceso rápido y reducir latencia
- c. Redis se consulta primero antes de otras fuentes de datos

## Cómputo Tradicional

### 7. Amazon EC2

- a. Ejecuta procesos o lógica de negocio más compleja
- b. Accede directamente a base de datos relacional (RDS)
- c. Podría usarse para pipelines, transformaciones o conexiones de larga duración

## Base de Datos Relacional

### 8. Amazon RDS (PostgreSQL + PostGIS)

- a. Instalado en subred privada
- b. Maneja transacciones, relaciones complejas e integridad referencial
- c. Incluye capacidades geoespaciales nativas para manipular datos georreferenciados

## Almacenamiento NoSQL

### 9. Amazon DocumentDB

- a. Subred privada también
- b. Aloja datos jerárquicos en JSON (sin necesidad de joins) para lecturas rápidas por parte de Lambdas

## Flujo Completo — Desde Request hasta Respuesta

1. **Solicitud entra** por Route 53 al VPC
2. Firewall Manager verifica la solicitud
3. API Gateway enruta la petición a Lambda o EC2
4. **Lambda > Redis**: si hay cache, devuelve GeoJSON
5. Si no está en cache, Lambda consulta DocumentDB o EC2 → RDS
6. El resultado se **almacena en Redis** para futuras solicitudes
7. Respuesta generada y devuelta al Business App

## Resumen de Roles por Componentes

Componente	Rol Principal
Route 53	Enrutamiento global por DNS y latencia
Firewall Manager	Protección y control de tráfico de red
API Gateway	Interfaz REST RESTful con seguridad, límites y validación
AWS Lambda	Lógica ligera, aislada, escalable y sin servidor
EC2	Procesamiento pesado, transaccional o duradero
ElastiCache	Capa de cache ultrarrápida para datos geoespaciales frecuentes
DocumentDB	Almacenamiento eficiente de estructuras JSON jerárquicas
RDS (PostgreSQL+PGIS)	Base relacional robusta con soporte espacial nativo

## Flujo de redirección por tipo de solicitud

- 1. Datos de uso constante (ej. provincias y ubicación)**
  - a. Destino: ElastiCache (Redis)**
  - b. Motivo:** Redis mantiene en memoria respuestas pre-cargadas (GeoJSON), lo que garantiza acceso en microsegundos [Towards](#). Se consultan primero en caché, evitando sobrecarga y reduciendo latencia.
- 2. Datos anidados (“consulta completa” como municipios, distritos, etc.)**
  - a. Destino: DocumentDB**
  - b. Motivo:** Almacena estructuras JSON jerárquicas, permitiendo lectura eficiente sin joins, ideal para datos anidados
- 3. Consultas relacionales o transaccionales (joins, integridad, actualizaciones)**
  - a. Destino: RDS (PostgreSQL + PostGIS)**
  - b. Motivo:** RDS ofrece integridad referencial, transacciones ACID y soporte nativo geoespacial; ideal para operaciones complejas

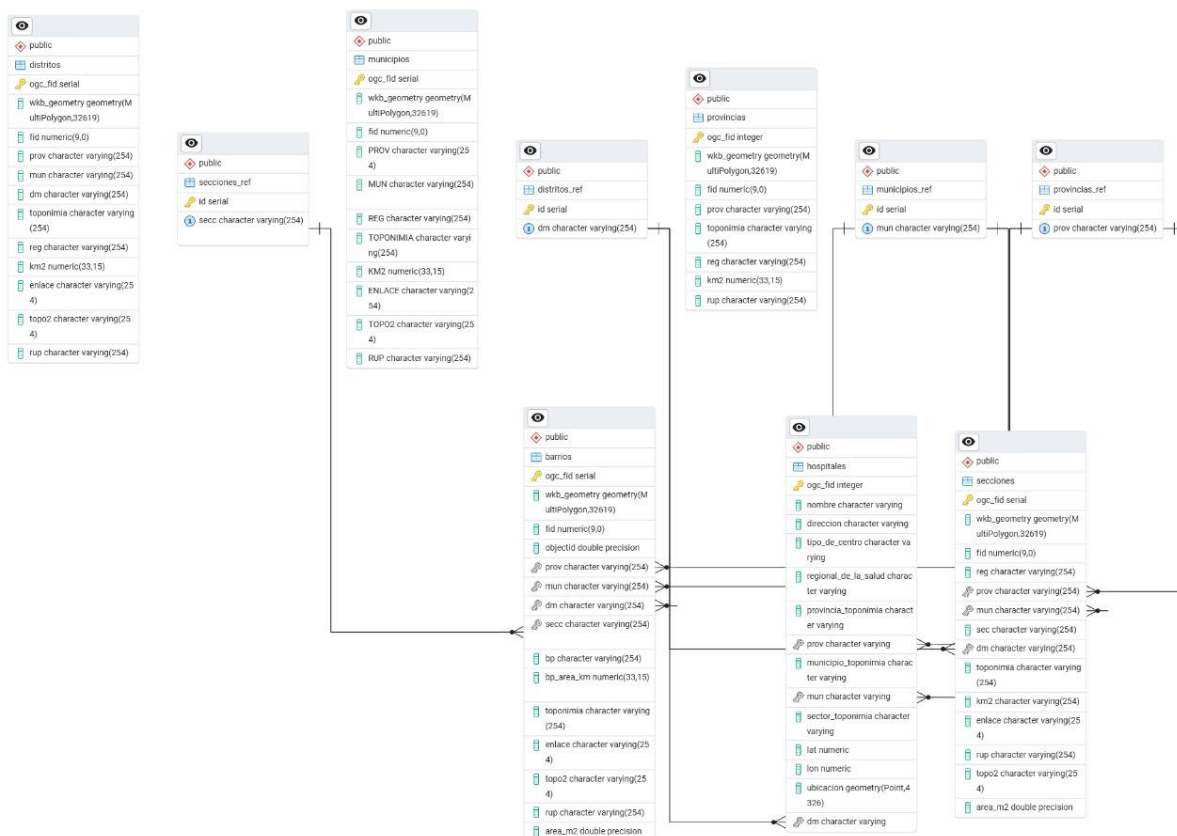
## Resumen del flujo de solicitud

- **API Gateway** recibe la petición.
- **Lambda o EC2** evalúa el tipo de consulta:
  - Si es un dato común (ej. provincias), va a **Redis**.
  - Si devuelve datos jerárquicos (municipios/distritos), va a **DocumentDB**.
  - Si la solicitud implica transacciones, relaciones complejas o integridad, se usa **RDS**.
- Si la respuesta no está en Redis, tras consultar DocumentDB o RDS, se almacena en caché para futuras consultas.

## Tabla de redirección y razones

Tipo de solicitud	Servicio destino	Justificación
Lista de provincias, coordenadas, datos constantes	<b>Redis</b> <b>(ElastiCache)</b>	Acceso ultrarrápido, frecuentes, bajo overhead
Datos anidados (municipios, distritos...)	<b>DocumentDB</b>	JSON jerárquico, lectura sin joins, consultas rápidas
Consultas relacionales, transacciones, integridad	<b>RDS</b> <b>(PostgreSQL+PostGIS)</b>	Soporte ACID, transacciones, geoespacial

Diagrama ER



## Tablas principales y campos clave

### 1. provincias

- ogc\_fid (PK) — identificador único.
  - fid — número de registro.
- prov, toponimia, reg, km2, rup — atributos relacionados con nombre, región, área, y código geográfico.
- Clave externa: ninguna directa aquí; referencias hacia distritos, municipios, secciones.

### 2. municipios

- ogc\_fid (PK).
- fid, PROV — identifica y asocia municipio con provincia.
- MUN, KM2, ENLACE, TOPO2, RUP — atributos del municipio.
- Relación: **municipios** → **provincias** (a través de PROV).

### 3. distritos

- ogc\_fid (PK).
- fid, prov, mun, dm, toponimia, reg, km2, enlace, topo2, rup.
  - Llaves:
    - prov → provincias
    - mun → municipios

### 4. secciones

- ogc\_fid (PK).
- fid, prov, mun, sec, dm, toponimia, km2, enlace, rup, area\_m2.
  - Relacionada con provincias y municipios (campo prov, mun).

### 5. barrios

- ogc\_fid (PK).
- Campos de localización: prov, mun, dm, secc (sección que lo agrupa).
- objectid, bp, bp\_area\_km, toponimia, enlace, topo2, rup, area\_m2.
- Relaciona niveles geográficos: provincia → municipio → sección → barrio.

### 6. hospitales

- ogc\_fid (PK).
- Atributos: nombre, dirección, tipo\_centro, regional\_de\_la\_salud, provincia\_toponimia, prov, municipio\_toponimia, mun, sector\_toponimia, lat, lon, ubicacion, dm.
- Relaciona con geografía: prov, mun conectan a **provincias** y **municipios**.

### 7. Tablas\_ref\* (referenciales)

- provincias\_ref, municipios\_ref, distritos\_ref, secciones\_ref

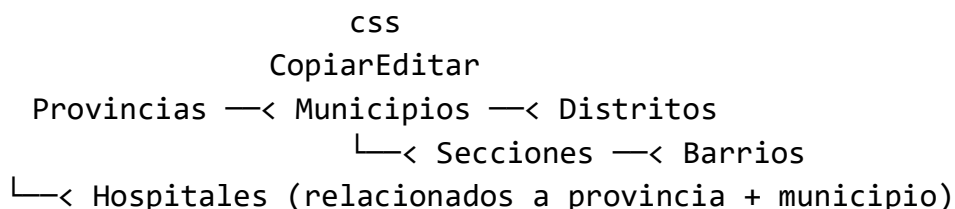


- Contienen claves primarias (ogc\_fid serial) y un campo con nombre referencial (prov, mun, dm, secc).
- Están referenciadas por las tablas principales para mantener integridad de los valores.

## Relaciones entre tablas

1. **provincias**  $\leftarrow$  **municipios** (PROV)
2. **municipios**  $\leftarrow$  **distritos** (prov, mun)
3. **municipios**  $\leftarrow$  **secciones** (prov, mun)
4. **secciones**  $\leftarrow$  **barrios** (prov, mun, secc)
5. **provincias** / **municipios**  $\leftarrow$  **hospitales** (prov, mun)
6. Referenciales (\*\_ref) están ligados a sus tablas principales correspondientes, usadas para integridad de dimensión geográfica.

## Jerarquía espacial resumida



- Tablas \*\_ref definen dominios válidos de valores (prov, mun, dm, secc).

## Propósitos generales

- Soporta consultas geográficas jerárquicas:
  - “¿Qué municipios hay en X provincia?”
  - “¿Qué barrios hay en una sección dada?”
- Permite filtrar y referenciar hospitales según niveles territoriales (provincia/municipio).
- Usa tablas referenciales para mantener limpieza e integridad de datos.

