

UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

TALLER DE BASE DE DATOS DIURNO 2-2025
Enunciado 2

Ayudante: Pablo Macuada
Fernando Solis

Profesor: Matías Calderón

Enunciado Laboratorio 2

Entrega: 19 enero de 2026

Proyecto:

Visualizador de Datos de Cambio Climático

En el presente laboratorio se trabajará con herramientas de bases de datos espaciales aplicadas a un problema real de gestión logística. A través de este ejercicio, se busca que los estudiantes integren conceptos de modelado de datos, consultas espaciales y desarrollo de aplicaciones.

Objetivo: Plataforma científica para visualizar datos climáticos georreferenciados y analizar fenómenos espaciales.

Tecnologías y Herramientas Requeridas

- **Base de Datos:** PostgreSQL con extensión **PostGIS**.
- **Backend:** Spring Boot con Java.
- **Frontend:** Vue.js 3 con librerías de mapas (Leaflet o OpenLayers).
- **Comunicación:** Axios.
- **Seguridad:** JWT.

Requisitos Específicos

1. Requisitos de la Base de Datos (PostgreSQL + PostGIS)

- **puntos_medicion:** Campo `geometria` (`GEOMETRY(POINT, 4326)`).
- **areas_afectadas:** Polígonos que representan zonas de sequía, inundación, etc. (`GEOMETRY(POLYGON)`).

No se permite el uso de JPA/Hibernate. La comunicación entre la aplicación y la base de datos debe realizarse exclusivamente a través de sentencias SQL nativas.

Consultas SQL a desarrollar (Enfoque Espacial)

1. **Análisis de Correlación Espacial:** Encontrar puntos de medición de CO2 que estén en un radio de 50km de puntos de medición de temperatura (`ST_DWithin`) y devolver la distancia.
2. **Interpolación (Vecinos):** Para una ubicación sin sensores, estimar el valor climático promediando los valores de los 3 sensores más cercanos (K-NN con operador `<->`).
3. **Detección de Puntos en Zonas de Riesgo:** Identificar qué puntos de medición caen dentro de polígonos definidos como "Zonas de Riesgo Climático" (`ST_Intersects`).
4. **Validación de Geometrías:** Consulta para detectar puntos de medición con coordenadas inválidas o geometrías corruptas (`ST_IsValid` es falso).

Documentación y Entrega

Todo el proyecto debe ser subido a un repositorio de **GitHub**. La entrega debe incluir:

- **Documentación de la Base de Datos:** Un documento conteniendo lo previamente realizado incluyendo para esta entrega las modificaciones para adaptar al sistema con la base de datos georeferenciada.
- **Script de Creación y Carga de Datos:** Un archivo `.sql` que permita recrear la base de datos completa, incluyendo todo lo previo más las adaptaciones de una base de datos georeferenciada.
- **Código Fuente:** El código completo del backend (Spring Boot) y el frontend (Vue.js), subido al repositorio de GitHub.

- **README.md:** Un archivo **README.md** en el repositorio principal que contenga instrucciones previas, y además, se debe incluir en el caso de usar mapas vectoriales ya generados obtenidos de internet, las instrucciones del cómo configurar y cargar esta misma a la base de datos