

**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:

#### Mapa Colaborativo de Sitios Turísticos

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: Desarrollar una red social geo-referenciada donde los usuarios compartan sitios de interés, utilizando cálculos de distancia real y ubicación geoespacial.**

#### 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) Diseñar un esquema normalizado espacial:

- **sitios\_turisticos:** Información del lugar incluyendo su ubicación precisa mediante una columna **ubicacion** de tipo **GEOGRAPHY(POINT, 4326)** (para cálculos precisos en metros).
- **usuarios:** Información de perfil y, opcionalmente, **ultima\_ubicacion** (**GEOGRAPHY(POINT)**).
- **rutas\_sugeridas:** Trazados de recorridos turísticos (**GEOMETRY(LINESTRING)**).
- **Elementos:**
  - **Índices GIST:** Para optimizar búsquedas de "ceranos a mí".

**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. **Búsqueda por Radio:** Procedimiento que reciba las coordenadas del usuario y devuelva los sitios turísticos en un radio de 5km ordenados por distancia (**ST\_Distance**).
2. **Análisis de Proximidad entre Tipos:** Encontrar 'Restaurantes' que estén a menos de 100 metros de un 'Teatro' utilizando **ST\_DWithin**.
3. **Sitios en Zona Personalizada:** Permitir que el usuario defina un polígono arbitrario (dibujado en el mapa) y encontrar todos los sitios contenidos en él usando **ST\_Contains** o **ST\_Covers**.
4. **Longitud de Ruta Turística:** Si existen rutas guardadas, calcular la longitud total del recorrido en kilómetros usando **ST\_Length**.

## 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.