

**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:

#### Sistema de Gestión de Flotas de Drones

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:** Monitoreo de drones utilizando coordenadas tridimensionales y trazado de rutas de vuelo.

#### 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)

- **registro\_vuelo (Telemetría):** Debe usar `GEOMETRY(POINTZ, 4326)` para almacenar Latitud, Longitud y **Altitud**.
- **misiones:** Campo `ruta_planificada` de tipo `GEOMETRY(LINESTRINGZ)` (Ruta en 3D).
- **zonas\_prohibidas:** Polígonos de zonas de exclusión aérea (`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. **Distancia 3D a Punto de Interés:** Calcular qué tan cerca pasó un dron de un punto de interés considerando la altura (`ST_3DDistance`), no solo la distancia en el suelo.
2. **Longitud Real de Vuelo:** Reconstruir la trayectoria volada uniendo los puntos de telemetría (`ST_MakeLine`) y calcular la longitud total recorrida (`ST_Length`).
3. **Infracción de Zonas Aéreas:** Detectar si alguna parte de la `ruta_planificada` o los puntos de `registro_vuelo` intersectan con `zonas_prohibidas` (`ST_Intersects`).
4. **Velocidad Media entre Puntos:** Calcular la distancia geográfica entre dos registros consecutivos de telemetría y dividir por la diferencia de tiempo para estimar velocidad.

## 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.