1. Descripción del Sistema

Aplicación de escritorio desarrollada en Intellij java (JDK 17+) utilizando JavaFX para la interfaz gráfica y MySQL como gestor de base de datos. El sistema gestiona la relación entre Estacionamientos, Lugares, Personas y Vehículos.

2. Arquitectura de Software

El proyecto sigue una arquitectura MVC (Modelo-Vista-Controlador) estricta para garantizar la mantenibilidad y escalabilidad.

- **com.example.Models:** Contiene las clases POJO (Persona, Vehiculo, Lugar).
  - *Patrón Builder:* Se utiliza para instanciar objetos complejos de manera limpia y legible (ej. new PersonaBuilder(id).setNombre("Luis").build()).
- **com.example.Views:** Archivos .fxml que definen la interfaz gráfica (separación total de la lógica).
- **com.example.Controllers:** Clases Java que gestionan la interacción del usuario y conectan las Vistas con los DAOs.
- **com.example.DataAccess (DAO):** Capa de acceso a datos.
  - *Patrón Singleton:* La clase DatabaseConnection asegura una única instancia de conexión a MySQL para optimizar recursos.
  - *Patrón DAO:* Clases como AuthDAO y OperacionesDAO encapsulan todas las consultas SQL.
- **com.example.Utils:** Utilidades transversales.
  - *Genéricos:* Se implementó AlertaUtils usando Programación Genérica (<T>) para mostrar alertas de cualquier tipo de dato (String, Int, Exception) sin duplicar código.

3. Base de Datos (Relacional)

El esquema consta de 5 tablas principales en la base de datos AdminEstacionamiento:

Tabla	Descripción	Clave Primaria (PK)
Estacionamientos	Catálogo de zonas (A, B, C).	id_estacionamiento
Lugares	Espacios físicos. Usa PK Compuesta.	(id_lugar, id_estacionamiento)
Personas	Usuarios del sistema con hashPass (SHA-1).	noCont
Vehiculos	Autos registrados. FK a Personas.	placa
Registros	Historial de entradas/salidas (relación entre lugares y vehículos)	id_registro