

Manual Técnico

Proyecto Final – Parqueo

Versión 1.0

1. Descripción General del Sistema

El sistema “Proyecto Final – Parqueo” es una aplicación de escritorio desarrollada en **Java Swing** que permite:

- Registrar usuarios y vehículos.
- Registrar **entrada** y **salida** de vehículos.
- Gestionar cobros por modalidad **plano** o **variable**.
- Administrar disponibilidad de **spots** de estacionamiento.
- Registrar transacciones en una base de datos **SQLite**.

2. Tecnologías Utilizadas

Componente	Tecnología
Lenguaje	Java SE 8+
Interfaz Gráfica	Java Swing
Base de Datos	SQLite
Driver	JDBC SQLite
IDE recomendado	NetBeans

3. Estructura del Proyecto

```
src/  
├── com.mycompany.proyectofinal/  
│   ├── PROYECTOFINAL.java  
│   ├── Spot.java  
│   ├── Usuariomolde.java  
│   ├── conexiondatos.java  
│   ├── cobro.java  
│   ├── usuariodatos.java  
│   ├── vehiculo.java  
│   ├── mapa.java  
│   ├── vista.java  
└── mostrardatos.java
```

4. Descripción de Clases

4.1 PROYECTOFINAL.java

Clase principal que inicializa el JFrame principal `mostrardatos`.

4.2 mostrardatos.java

- Ventana principal del sistema.
 - Implementa dos pestañas:
 - Registro de usuarios y vehículos.
 - Registro de entrada/salida de vehículos.
 - Interactúa directamente con las clases:
 - `Spot` (entradas/salidas)
 - `usuariodatos` (registro en BD)
-

4.3 Spot.java

Clase central de la lógica del parqueo.

Funciones principales:

- `registrarEntrada()` → asigna spot, genera ticket y marca spot ocupado.
- `registrarSalida()` → calcula monto, registra salida y libera spot.

Incluye clase interna:

Cobro

- `cobroPlano()` → Q15
 - `cobroVariable(minutos)` → Q10 por hora redondeada hacia arriba.
-

4.4 Usuariomolde.java

Molde de datos del usuario:

- Nombre
 - Carnet
 - Vehículo asociado
-

4.5 cobro.java

Clase sencilla alterna para manejo básico de cobros:

- Admítela como legacy, no principal.
-

4.6 conexiondatos.java

Clase de conexión con la base SQLite mediante JDBC.

4.7 usuariodatos.java

- Inserta usuarios en la tabla `usuario`.
 - Obtiene ID autogenerado.
 - Inserta vehículo asociado en `vehiculo`.
-

4.8 vehiculo.java

Molde de datos del vehículo:

- placa
 - modelo
 - rol (Estudiante/Catedrático)
-

4.9 mapa.java

Panel gráfico que carga imágenes para mostrar el mapa del parqueo.

4.10 vista.java

JFrame que contiene el panel `mapa`.

5. Base de Datos

5.1 Esquema

Tabla: Area

- Define zonas según rol y tipo de vehículo.

Tabla: Usuario

- Registra usuarios con ID autoincremental.

Tabla: Vehiculo

- Registra vehículos y su asociación a un usuario.

Tabla: spot

- Spots disponibles.
- Estado: libre, ocupado, reservado.

Tabla: tickets

- Registra ingresos, salidas, cobros y metodo_pago.
-

6. Flujo General del Sistema

6.1 Registro de Usuario

1. El usuario ingresa datos.
 2. Se crean objetos `usuariomolde` y `vehiculo`.
 3. `usuariodatos.agregarUsuario()` inserta en BD.
 4. Vehículo se relaciona con el usuario.
-

6.2 Registro de Entrada

1. Se recibe placa y modo.
 2. Se valida vehículo en BD.
 3. Se identifica área compatible.
 4. Se busca spot libre.
 5. Se registra ticket.
 6. Se marca spot como ocupado.
-

6.3 Registro de Salida

1. Se consulta ticket activo.
 2. Si modo = variable, calcula minutos.
 3. Se genera cobro correspondiente.
 4. Se actualiza ticket.
 5. Se libera spot.
-

7. Requerimientos

- JDK 8 o superior
 - SQLite JDBC Driver (incluido en NetBeans si usas `sqlite-jdbc.jar`)
 - Carpeta `Basededatos/`
 - Archivo: `basededatos.db`
-

8. Instalación y Ejecución

1. Descargar proyecto.
2. Abrir en NetBeans.
3. Verificar ruta de base:

`jdbc:sqlite:Basededatos/basededatos.db`

4. Ejecutar la clase `PROYECTOFINAL.java`.
-

9. Manejo de Errores Comunes

Error	Causa	Solución
Vehículo no encontrado	Placa no existe	Registrar vehículo
No hay spots libres	Área llena	Esperar o revisar BD
Billete insuficiente	cobro > billete	Ingresar monto correcto
DB no encontrada	Ruta incorrecta	Revisar carpeta Basededatos

10. Créditos

Proyecto Final – Parqueo
Desarrollado para fines académicos.