

# Plataforma de Alquiler de Vehículos

## 1. Descripción General

Este proyecto implementa una base de datos relacional en **PostgreSQL** para la gestión integral de una empresa de alquiler de vehículos. La solución contempla la administración de flota, clientes, oficinas, reservas, contratos de alquiler, historial de mantenimiento, tarifas por temporada, roles de usuarios y auditorías.

## 2. Tecnologías Usadas

- **PostgreSQL** como sistema gestor de base de datos.
- **Docker** para la gestión de ambientes de base de datos.
- **DrawSQL (draw.db)** para el diseño del modelo relacional.
- **GitHub** como repositorio de entrega.
- **.sql** como script principal.
- **.xlsx** como diccionario de datos.

## 3. Modelo de Datos

El sistema cuenta con **16 tablas principales**, cada una relacionada con aspectos clave del negocio:

Tabla	Descripción
Marcas	Almacena las marcas de vehículos.
Modelos	Relaciona cada modelo con una marca.
TiposVehiculo	Categoriza los vehículos (Sedán, SUV, etc.).
EstadosVehiculo	Controla el estado actual del vehículo.
Oficinas	Registra oficinas de alquiler y devolución.
Vehiculos	Detalla información completa de cada vehículo.
Clientes	Contiene datos personales y de contacto.
Reservas	Almacena las solicitudes de reserva de vehículos.
ContratosAlquiler	Documenta el contrato entre cliente y empresa.
Seguros	Define tipos de seguros disponibles.
VehiculoSeguro	Tabla intermedia entre vehículos y seguros.
Tarifas	Tarifas por tipo de vehículo y temporada.
Mantenimientos	Historial de mantenimientos por vehículo.
RolesUsuario	Define los distintos roles del sistema.
Usuarios	Usuarios de la plataforma vinculados a roles.
Auditoria_Reservas	Audita nuevas reservas.

## 4. Datos de Ejemplo

Se insertaron **más de 5 registros** por cada una de las tablas principales para validación y pruebas, asegurando consistencia e integridad referencial.

## 5. Funcionalidades Avanzadas SQL

### a. Índices

- `idx_vehiculos_tipo` y `idx_vehiculos_oficina` para búsquedas rápidas.
- `idx_reservas_cliente_fecha` para consultas por cliente y fechas.
- `idx_contratos_activos` para contratos por fecha de finalización.

### b. Vistas

1. `vista_vehiculos_disponibles`: Vehículos por estado "Disponible".
2. `vista_clientes_alquileres_activos`: Clientes con alquiler en curso.
3. `vista_historial_alquileres_cliente`: Historial detallado por cliente.
4. `vista_vehiculos_mantenimiento`: Vehículos en mantenimiento.
5. `vista_detalle_contratos`: Contratos con detalles de cliente y vehículo.

### c. Procedimientos Almacenados (Stored Procedures)

- `crear_vehiculo()`: Inserta nuevo vehículo.
- `crear_cliente()`: Inserta nuevo cliente.
- `crear_reserva()`: Inserta nueva reserva.
- `crear_contrato()`: Genera un contrato de alquiler.

### d. Triggers

1. `trg_actualizar_estado_alquiler`: Al insertar contrato, cambia estado del vehículo a "Alquilado".
2. `trg_marcar_disponible`: Marca vehículo como "Disponible" al finalizar contrato.
3. `trg_auditoria_reserva`: Audita la creación de nuevas reservas.

### e. Funciones Definidas por el Usuario

- `verificar_disponibilidad(oficina_id, tipo_id)`: Devuelve número de vehículos disponibles.
- `calcular_costo_alquiler(dias, tarifa_dia)`: Calcula costo total del alquiler.

## 6. Consultas y Subconsultas

Para cada tabla principal se desarrollaron:

- 2 consultas directas con `JOIN`.
- 2 subconsultas con `INNER`, `LEFT` o `RIGHT JOIN`.

Ejemplos:

- Clientes con más alquileres.
- Vehículos disponibles en oficinas específicas.
- Contratos activos por fecha.

- Historial de mantenimientos con filtros por tipo.

## 7. Optimización

Se utilizaron expresiones comunes (CTE) en consultas complejas como:

- Agrupación de vehículos disponibles por tipo y oficina.
- Historial de alquileres por cliente con fechas ordenadas.

## 8. Seguridad

Se crearon **roles personalizados** con permisos específicos:

Rol	Descripción y Permisos Básicos
cliente	Lectura de reservas y contratos propios.
agente_alquiler	Inserción y actualización de reservas y contratos.
mecanico	Gestión de vehículos y mantenimientos.
admin_flota	Acceso total sobre las tablas.
admin_sistema	Rol superusuario con gestión de roles.

Usuarios de ejemplo fueron creados y se asignaron a los roles respectivos con contraseñas seguras.

## 9. Estructura del Repositorio

```
/proyecto-alquiler-vehiculos
|
├─ script_postgresql.sql      # Script completo de creación de tablas,
relaciones e inserciones
├─ modelo_relacional.drawdb   # Diagrama del modelo relacional generado en
DrawSQL
├─ diccionario_datos.xlsx     # Diccionario de datos: campos, tipos, claves,
restricciones
├─ presentacion.pptx         # Presentación con el resumen del sistema en 10
diapositivas
```

## 10. Conclusiones

Este sistema representa una implementación robusta de una base de datos transaccional orientada a la gestión de alquiler de vehículos. Se aplicaron buenas prácticas de diseño relacional, normalización, seguridad de roles, auditoría de operaciones, optimización de consultas y uso de funciones y triggers para automatizar comportamientos críticos.

Es una solución ideal para empresas del sector automotor que buscan digitalizar sus procesos de reserva, mantenimiento y control de flotas.

## 11. Referencias

- [PostgreSQL Official Documentation](#)
- [DrawSQL - Visual Database Schema Designer](#)
- [Docker PostgreSQL Images](#)