**Sistema de Gestión de Flota**

PepsiCo Chile

MODELO DE DATOS

*Diagrama Entidad-Relación*

Fecha: 14 de octubre de 2025

Equipo: Joaquín Marín y Benjamin Vilches

Duoc UC - Proyecto de Título 2025

# MODELO DE DATOS - DIAGRAMA ENTIDAD-RELACIÓN

Proyecto: Plataforma de Gestión de Ingreso de Vehículos - PepsiCo Chile

Fecha: 13/10/2025

Responsables: Joaquín Marín y Benjamin Vilches

Versión: 1.0

Base de Datos: PostgreSQL 15+

1. DIAGRAMA ENTIDAD-RELACIÓN (FORMATO ASCII)

┌─────────────────────────────────────────────────────────────────────────┐
│ MODELO DE DATOS COMPLETO │
└─────────────────────────────────────────────────────────────────────────┘

┌──────────────┐ ┌──────────────────┐ ┌──────────────┐
│ USUARIOS │ 1 \* │ VEHICULOS │ 1 \* │ INGRESOS │
│──────────────│◄────────│──────────────────│◄────────│──────────────│
│ PK id │ │ PK id │ │ PK id │
│ email │ │ patente │ │ fecha\_ing │
│ password │ │ tipo │ │ km\_actual │
│ nombre │ │ marca │ │ estado │
│ rol │ │ modelo │ │ FK vehiculo │
│ activo │ │ año │ │ FK guardia │
│ ... │ │ FK region\_id │ │ FK recepcion │
└──────────────┘ │ activo │ │ ... │
│ └──────────────────┘ └──────────────┘
│ │
│ │ 1
│ │
│ 1 │ \*
│ ┌──────────────────┐ ┌──────────────┐
└────────────────►│ ORDENES\_TRABAJO │ 1 \* │ OT\_FOTOS │
│──────────────────│◄────────│──────────────│
│ PK id │ │ PK id │
│ numero\_ot │ │ url │
│ descripcion │ │ tipo │
│ prioridad │ │ FK ot\_id │
│ estado │ └──────────────┘
│ FK ingreso\_id │
│ FK mecanico\_id │
│ fecha\_inicio │ ┌──────────────┐
│ fecha\_fin │ │ OT\_REPUESTOS │
│ ... │ 1 \* │──────────────│
└──────────────────┘◄────────│ PK id │
│ │ FK ot\_id │
│ 1 │ FK repuesto │
│ │ cantidad │
│ \* │ precio │
┌──────────────────┐ └──────────────┘
│ OT\_ESTADOS │
│──────────────────│
│ PK id │ ┌──────────────┐
│ FK ot\_id │ │ REPUESTOS │
│ estado │ │──────────────│
│ fecha │ ┌───│ PK id │
│ observacion │ │ │ codigo │
│ FK usuario\_id │ │ │ nombre │
└──────────────────┘ │ │ marca │
│ │ categoria │
│ │ stock │
┌──────────────┐ ┌──────────────────┐ │ │ stock\_min │
│ REGIONES │ 1 \* │ TALLERES │ │ │ precio │
│──────────────│◄────────│──────────────────│ │ │ ubicacion │
│ PK id │ │ PK id │ │ └──────────────┘
│ nombre │ │ nombre │ │ │
│ codigo │ │ direccion │ │ │ \*
└──────────────┘ │ telefono │ │ │
│ │ FK region\_id │ │ ┌───────────────────┐
│ │ capacidad │ │ │ MOVIM\_REPUESTOS │
│ │ activo │ │ │───────────────────│
│ └──────────────────┘ └───│ PK id │
│ │ tipo │
│ 1 │ cantidad │
│ ┌──────────────────┐ │ fecha │
└────────────────►│ HORARIOS\_TALLER│ │ FK repuesto\_id │
│──────────────────│ │ FK usuario\_id │
│ PK id │ │ observacion │
│ FK taller\_id │ └───────────────────┘
│ dia\_semana │
│ hora\_inicio │
│ hora\_fin │ ┌───────────────────┐
│ activo │ │ CONTROL\_LLAVES │
└──────────────────┘ │───────────────────│
│ PK id │
│ FK vehiculo\_id │
┌──────────────┐ ┌──────────────────┐ │ numero\_llave │
│ DOCUMENTOS │ \* 1 │ INGRESOS │ │ estado │
│──────────────│────────►│──────────────────│ │ fecha\_entrega │
│ PK id │ │ │ │ fecha\_devol │
│ tipo │ │ (ver arriba) │ │ FK resp\_entrega │
│ nombre │ │ │ │ FK resp\_devol │
│ url │ │ │ │ observacion │
│ formato │ └──────────────────┘ └───────────────────┘
│ FK ingreso │
│ fecha │
└──────────────┘ ┌──────────────────┐ ┌───────────────────┐
│ NOTIFICACIONES │ │ PAUSAS │
│──────────────────│ │───────────────────│
│ PK id │ │ PK id │
│ tipo │ │ FK ot\_id │
│ titulo │ │ motivo │
│ mensaje │ │ fecha\_inicio │
│ leida │ │ fecha\_fin │
│ fecha │ │ duracion │
│ FK usuario\_id │ │ FK usuario\_id │
│ FK referencia │ │ observacion │
│ prioridad │ └───────────────────┘
└──────────────────┘

┌──────────────┐ ┌──────────────────┐ ┌───────────────────┐
│ ROLES │ 1 \* │ PERMISOS │ │ ROLES\_PERMISOS │
│──────────────│ │──────────────────│ \* \* │───────────────────│
│ PK id │ ┌───►│ PK id │◄───────│ PK id │
│ nombre │ │ │ modulo │ │ FK rol\_id │
│ codigo │ │ │ accion │ │ FK permiso\_id │
│ desc │ │ │ descripcion │ │ activo │
└──────────────┘ │ └──────────────────┘ └───────────────────┘
│
└────────────────────────────────────────────┐
│
┌──────────────────────────────────────────────────────────────┐ │
│ AUDITORIA / LOGS │ │
│──────────────────────────────────────────────────────────────│ │
│ PK id │ │
│ FK usuario\_id │◄┘
│ accion │
│ tabla\_afectada │
│ registro\_id │
│ datos\_anteriores │
│ datos\_nuevos │
│ ip\_address │
│ user\_agent │
│ fecha\_hora │
└──────────────────────────────────────────────────────────────┘

2. CATÁLOGO DE ENTIDADES

## 2.1. USUARIOS (users)

────────────────────────────────────────────────────────────────

Descripción: Almacena la información de todos los usuarios del sistema

Tipo: Maestra

Relaciones: 1:N con ingresos, órdenes de trabajo, notificaciones

## 2.2. ROLES (roles)

────────────────────────────────────────────────────────────────

Descripción: Define los 10 perfiles de usuario del sistema

Tipo: Catálogo

Relaciones: 1:N con usuarios, N:M con permisos

## 2.3. PERMISOS (permissions)

────────────────────────────────────────────────────────────────

Descripción: Define las acciones permitidas en cada módulo

Tipo: Catálogo

Relaciones: N:M con roles

## 2.4. ROLES\_PERMISOS (role\_permissions)

────────────────────────────────────────────────────────────────

Descripción: Tabla intermedia para relacionar roles con permisos

Tipo: Relacional

Relaciones: N:M entre roles y permisos

## 2.5. REGIONES (regions)

────────────────────────────────────────────────────────────────

Descripción: Regiones operativas de PepsiCo (Norte, Centro, Sur, etc.)

Tipo: Catálogo

Relaciones: 1:N con talleres, vehículos

## 2.6. TALLERES (workshops)

────────────────────────────────────────────────────────────────

Descripción: Talleres de mantenimiento de la flota

Tipo: Maestra

Relaciones: N:1 con regiones, 1:N con horarios

## 2.7. HORARIOS\_TALLER (workshop\_schedules)

────────────────────────────────────────────────────────────────

Descripción: Horarios de operación de cada taller

Tipo: Detalle

Relaciones: N:1 con talleres

## 2.8. VEHICULOS (vehicles)

────────────────────────────────────────────────────────────────

Descripción: Información de cada vehículo de la flota

Tipo: Maestra

Relaciones: N:1 con regiones, 1:N con ingresos

## 2.9. INGRESOS (vehicle\_entries)

────────────────────────────────────────────────────────────────

Descripción: Registro de cada ingreso de vehículo al taller

Tipo: Transaccional

Relaciones: N:1 con vehículos, usuarios, 1:N con órdenes de trabajo

## 2.10. ORDENES\_TRABAJO (work\_orders)

────────────────────────────────────────────────────────────────

Descripción: Órdenes de trabajo generadas para reparaciones

Tipo: Transaccional

Relaciones: N:1 con ingresos, usuarios (mecánico), 1:N con estados, fotos

## 2.11. OT\_ESTADOS (work\_order\_statuses)

────────────────────────────────────────────────────────────────

Descripción: Historial de cambios de estado de cada orden de trabajo

Tipo: Histórico

Relaciones: N:1 con órdenes de trabajo, usuarios

## 2.12. OT\_FOTOS (work\_order\_photos)

────────────────────────────────────────────────────────────────

Descripción: Fotografías asociadas a órdenes de trabajo

Tipo: Detalle

Relaciones: N:1 con órdenes de trabajo

## 2.13. PAUSAS (work\_pauses)

────────────────────────────────────────────────────────────────

Descripción: Pausas en el trabajo de las órdenes (espera repuestos, etc.)

Tipo: Transaccional

Relaciones: N:1 con órdenes de trabajo, usuarios

## 2.14. REPUESTOS (spare\_parts)

────────────────────────────────────────────────────────────────

Descripción: Catálogo de repuestos disponibles en inventario

Tipo: Maestra

Relaciones: 1:N con movimientos de repuestos, OT\_repuestos

## 2.15. OT\_REPUESTOS (work\_order\_spare\_parts)

────────────────────────────────────────────────────────────────

Descripción: Repuestos utilizados en cada orden de trabajo

Tipo: Detalle

Relaciones: N:1 con órdenes de trabajo, repuestos

## 2.16. MOVIM\_REPUESTOS (spare\_part\_movements)

────────────────────────────────────────────────────────────────

Descripción: Movimientos de inventario (entradas/salidas)

Tipo: Transaccional

Relaciones: N:1 con repuestos, usuarios

## 2.17. CONTROL\_LLAVES (key\_control)

────────────────────────────────────────────────────────────────

Descripción: Control de entrega y devolución de llaves de vehículos

Tipo: Transaccional

Relaciones: N:1 con vehículos, usuarios

## 2.18. DOCUMENTOS (documents)

────────────────────────────────────────────────────────────────

Descripción: Documentos adjuntos a ingresos (permisos, informes, etc.)

Tipo: Detalle

Relaciones: N:1 con ingresos

## 2.19. NOTIFICACIONES (notifications)

────────────────────────────────────────────────────────────────

Descripción: Notificaciones enviadas a usuarios

Tipo: Transaccional

Relaciones: N:1 con usuarios

## 2.20. AUDITORIA (audit\_logs)

────────────────────────────────────────────────────────────────

Descripción: Registro de todas las acciones realizadas en el sistema

Tipo: Log

Relaciones: N:1 con usuarios

3. DEFINICIÓN DETALLADA DE TABLAS

## 3.1. USUARIOS (users)

────────────────────────────────────────────────────────────────

CREATE TABLE users (

id SERIAL PRIMARY KEY,
email VARCHAR(255) UNIQUE NOT NULL,
password\_hash VARCHAR(255) NOT NULL,
nombre VARCHAR(100) NOT NULL,
apellido VARCHAR(100) NOT NULL,
rut VARCHAR(12) UNIQUE NOT NULL,
telefono VARCHAR(20),
rol\_id INTEGER NOT NULL REFERENCES roles(id),
region\_id INTEGER REFERENCES regions(id),
taller\_id INTEGER REFERENCES workshops(id),
avatar\_url VARCHAR(500),
activo BOOLEAN DEFAULT true,
ultimo\_acceso TIMESTAMP,
fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
fecha\_modificacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
creado\_por INTEGER REFERENCES users(id),
modificado\_por INTEGER REFERENCES users(id)

);

CREATE INDEX idx\_users\_email ON users(email);

CREATE INDEX idx\_users\_rut ON users(rut);

CREATE INDEX idx\_users\_rol ON users(rol\_id);

CREATE INDEX idx\_users\_activo ON users(activo);

## 3.2. ROLES (roles)

────────────────────────────────────────────────────────────────

CREATE TABLE roles (

id SERIAL PRIMARY KEY,
nombre VARCHAR(50) UNIQUE NOT NULL,
codigo VARCHAR(30) UNIQUE NOT NULL,
descripcion TEXT,
nivel\_acceso INTEGER DEFAULT 1,
activo BOOLEAN DEFAULT true,
fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Datos iniciales (10 perfiles)

INSERT INTO roles (nombre, codigo, descripcion, nivel\_acceso) VALUES

('Administrador del Sistema', 'ADMIN', 'Control total del sistema', 10),

('Supervisor de Flota', 'SUPERVISOR\_FLOTA', 'Auditoría y control de políticas', 9),

('Coordinador de Zona', 'COORDINADOR\_ZONA', 'Gestión regional, inventario y reportes', 8),

('Jefe de Taller', 'JEFE\_TALLER', 'Asignación de mecánicos, control de agenda', 7),

('Mecánico', 'MECANICO', 'Actualización de trabajos y reparaciones', 5),

('Recepcionista', 'RECEPCIONISTA', 'Validación documentos, generación OT', 5),

('Guardia de Acceso', 'GUARDIA', 'Registro ingreso/salida, fotos', 4),

('Asistente de Repuestos', 'ASIST\_REPUESTOS', 'Gestión inventario y entrega', 5),

('Encargado de Llaves', 'ENCARGADO\_LLAVES', 'Control de llaves y documentos', 4),

('Ventas Terreno', 'VENTAS\_TERRENO', 'Recepción y devolución de vehículos', 3);

## 3.3. PERMISOS (permissions)

────────────────────────────────────────────────────────────────

CREATE TABLE permissions (

id SERIAL PRIMARY KEY,
modulo VARCHAR(50) NOT NULL,
accion VARCHAR(50) NOT NULL,
codigo VARCHAR(100) UNIQUE NOT NULL,
descripcion TEXT,
activo BOOLEAN DEFAULT true,
fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
UNIQUE(modulo, accion)

);

-- Ejemplos de permisos

INSERT INTO permissions (modulo, accion, codigo, descripcion) VALUES

('usuarios', 'crear', 'users.create', 'Crear nuevos usuarios'),

('usuarios', 'leer', 'users.read', 'Ver información de usuarios'),

('usuarios', 'actualizar', 'users.update', 'Modificar usuarios'),

('usuarios', 'eliminar', 'users.delete', 'Eliminar usuarios'),

('vehiculos', 'crear', 'vehicles.create', 'Registrar nuevos vehículos'),

('vehiculos', 'leer', 'vehicles.read', 'Ver información de vehículos'),

('ingresos', 'crear', 'entries.create', 'Registrar ingreso de vehículos'),

('ingresos', 'leer', 'entries.read', 'Ver ingresos registrados'),

('ordenes', 'crear', 'orders.create', 'Crear órdenes de trabajo'),

('ordenes', 'leer', 'orders.read', 'Ver órdenes de trabajo'),

('ordenes', 'actualizar', 'orders.update', 'Actualizar estado de órdenes'),

('ordenes', 'asignar', 'orders.assign', 'Asignar mecánicos a órdenes'),

('repuestos', 'leer', 'parts.read', 'Ver inventario de repuestos'),

('repuestos', 'actualizar', 'parts.update', 'Actualizar stock de repuestos'),

('reportes', 'generar', 'reports.generate', 'Generar reportes'),

('llaves', 'gestionar', 'keys.manage', 'Gestionar control de llaves');

## 3.4. ROLES\_PERMISOS (role\_permissions)

────────────────────────────────────────────────────────────────

CREATE TABLE role\_permissions (

id SERIAL PRIMARY KEY,
rol\_id INTEGER NOT NULL REFERENCES roles(id) ON DELETE CASCADE,
permiso\_id INTEGER NOT NULL REFERENCES permissions(id) ON DELETE CASCADE,
activo BOOLEAN DEFAULT true,
fecha\_asignacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
UNIQUE(rol\_id, permiso\_id)

);

CREATE INDEX idx\_role\_permissions\_rol ON role\_permissions(rol\_id);

CREATE INDEX idx\_role\_permissions\_permiso ON role\_permissions(permiso\_id);

## 3.5. REGIONES (regions)

────────────────────────────────────────────────────────────────

CREATE TABLE regions (

id SERIAL PRIMARY KEY,
nombre VARCHAR(100) UNIQUE NOT NULL,
codigo VARCHAR(20) UNIQUE NOT NULL,
descripcion TEXT,
activo BOOLEAN DEFAULT true,
fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

-- Datos iniciales

INSERT INTO regions (nombre, codigo) VALUES

('Región Metropolitana', 'RM'),

('Región de Valparaíso', 'V'),

('Región del Biobío', 'VIII'),

('Región de Antofagasta', 'II'),

('Región de La Araucanía', 'IX');

## 3.6. TALLERES (workshops)

────────────────────────────────────────────────────────────────

CREATE TABLE workshops (

id SERIAL PRIMARY KEY,
nombre VARCHAR(100) NOT NULL,
codigo VARCHAR(20) UNIQUE NOT NULL,
tipo VARCHAR(20) NOT NULL, -- 'interno' o 'externo'
direccion TEXT,
comuna VARCHAR(100),
telefono VARCHAR(20),
email VARCHAR(100),
region\_id INTEGER NOT NULL REFERENCES regions(id),
capacidad\_diaria INTEGER DEFAULT 10,
activo BOOLEAN DEFAULT true,
fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
CHECK (tipo IN ('interno', 'externo'))

);

CREATE INDEX idx\_workshops\_region ON workshops(region\_id);

CREATE INDEX idx\_workshops\_activo ON workshops(activo);

## 3.7. HORARIOS\_TALLER (workshop\_schedules)

────────────────────────────────────────────────────────────────

CREATE TABLE workshop\_schedules (

id SERIAL PRIMARY KEY,
taller\_id INTEGER NOT NULL REFERENCES workshops(id) ON DELETE CASCADE,
dia\_semana INTEGER NOT NULL, -- 1=Lunes, 7=Domingo
hora\_inicio TIME NOT NULL,
hora\_fin TIME NOT NULL,
activo BOOLEAN DEFAULT true,
fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
CHECK (dia\_semana BETWEEN 1 AND 7),
UNIQUE(taller\_id, dia\_semana)

);

CREATE INDEX idx\_schedules\_taller ON workshop\_schedules(taller\_id);

## 3.8. VEHICULOS (vehicles)

────────────────────────────────────────────────────────────────

CREATE TABLE vehicles (

id SERIAL PRIMARY KEY,
patente VARCHAR(10) UNIQUE NOT NULL,
tipo VARCHAR(50) NOT NULL, -- 'camión', 'furgón', 'pickup', etc.
marca VARCHAR(50) NOT NULL,
modelo VARCHAR(50) NOT NULL,
año INTEGER NOT NULL,
color VARCHAR(30),
vin VARCHAR(17) UNIQUE,
kilometraje\_actual INTEGER DEFAULT 0,
capacidad\_carga DECIMAL(10,2), -- En toneladas
region\_id INTEGER NOT NULL REFERENCES regions(id),
estado\_general VARCHAR(20) DEFAULT 'operativo',
fecha\_adquisicion DATE,
activo BOOLEAN DEFAULT true,
observaciones TEXT,
fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
fecha\_modificacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
CHECK (estado\_general IN ('operativo', 'mantenimiento', 'fuera\_servicio')),
CHECK (año BETWEEN 1990 AND 2100)

);

CREATE INDEX idx\_vehicles\_patente ON vehicles(patente);

CREATE INDEX idx\_vehicles\_tipo ON vehicles(tipo);

CREATE INDEX idx\_vehicles\_region ON vehicles(region\_id);

CREATE INDEX idx\_vehicles\_estado ON vehicles(estado\_general);

## 3.9. INGRESOS (vehicle\_entries)

────────────────────────────────────────────────────────────────

CREATE TABLE vehicle\_entries (

id SERIAL PRIMARY KEY,
codigo\_ingreso VARCHAR(20) UNIQUE NOT NULL,
vehiculo\_id INTEGER NOT NULL REFERENCES vehicles(id),
taller\_id INTEGER NOT NULL REFERENCES workshops(id),
guardia\_ingreso\_id INTEGER NOT NULL REFERENCES users(id),
recepcionista\_id INTEGER REFERENCES users(id),
conductor\_nombre VARCHAR(100),
conductor\_rut VARCHAR(12),
conductor\_telefono VARCHAR(20),
fecha\_hora\_ingreso TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,
fecha\_hora\_salida TIMESTAMP,
kilometraje\_ingreso INTEGER NOT NULL,
kilometraje\_salida INTEGER,
nivel\_combustible VARCHAR(20), -- '1/4', '1/2', '3/4', 'lleno'
estado\_ingreso VARCHAR(20) NOT NULL DEFAULT 'registrado',
tipo\_servicio VARCHAR(50) NOT NULL, -- 'mantenimiento', 'reparacion', 'revision'
prioridad VARCHAR(20) DEFAULT 'normal',
observaciones\_ingreso TEXT,
observaciones\_salida TEXT,
documentos\_entregados JSONB, -- Array de documentos: ['permiso\_circulacion', 'seguro', etc.]
items\_verificados JSONB, -- Checklist de verificación de ingreso
duracion\_total INTERVAL, -- Calculado automáticamente
fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
fecha\_modificacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
CHECK (estado\_ingreso IN ('registrado', 'en\_recepcion', 'en\_taller',
'pausado', 'completado', 'entregado')),
CHECK (prioridad IN ('baja', 'normal', 'alta', 'urgente')),
CHECK (tipo\_servicio IN ('mantenimiento\_preventivo', 'mantenimiento\_correctivo',
'reparacion', 'revision\_tecnica', 'diagnostico'))

);

CREATE INDEX idx\_entries\_vehiculo ON vehicle\_entries(vehiculo\_id);

CREATE INDEX idx\_entries\_taller ON vehicle\_entries(taller\_id);

CREATE INDEX idx\_entries\_estado ON vehicle\_entries(estado\_ingreso);

CREATE INDEX idx\_entries\_fecha\_ingreso ON vehicle\_entries(fecha\_hora\_ingreso);

CREATE INDEX idx\_entries\_codigo ON vehicle\_entries(codigo\_ingreso);

## 3.10. ORDENES\_TRABAJO (work\_orders)

────────────────────────────────────────────────────────────────

CREATE TABLE work\_orders (

id SERIAL PRIMARY KEY,
numero\_ot VARCHAR(20) UNIQUE NOT NULL,
ingreso\_id INTEGER NOT NULL REFERENCES vehicle\_entries(id),
mecanico\_asignado\_id INTEGER REFERENCES users(id),
jefe\_taller\_id INTEGER NOT NULL REFERENCES users(id),
tipo\_trabajo VARCHAR(50) NOT NULL,
descripcion TEXT NOT NULL,
diagnostico TEXT,
trabajo\_realizado TEXT,
prioridad VARCHAR(20) DEFAULT 'normal',
estado VARCHAR(30) NOT NULL DEFAULT 'pendiente',
fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
fecha\_asignacion TIMESTAMP,
fecha\_inicio\_real TIMESTAMP,
fecha\_fin\_estimada TIMESTAMP,
fecha\_fin\_real TIMESTAMP,
horas\_estimadas DECIMAL(5,2),
horas\_reales DECIMAL(5,2),
costo\_repuestos DECIMAL(10,2) DEFAULT 0,
costo\_mano\_obra DECIMAL(10,2) DEFAULT 0,
costo\_total DECIMAL(10,2) GENERATED ALWAYS AS
(costo\_repuestos + costo\_mano\_obra) STORED,
observaciones TEXT,
requiere\_aprobacion BOOLEAN DEFAULT false,
aprobada BOOLEAN DEFAULT false,
aprobada\_por INTEGER REFERENCES users(id),
fecha\_aprobacion TIMESTAMP,
fecha\_modificacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
CHECK (estado IN ('pendiente', 'asignada', 'en\_progreso', 'pausada',
'completada', 'cancelada', 'requiere\_aprobacion')),
CHECK (prioridad IN ('baja', 'normal', 'alta', 'urgente'))

);

CREATE INDEX idx\_orders\_numero ON work\_orders(numero\_ot);

CREATE INDEX idx\_orders\_ingreso ON work\_orders(ingreso\_id);

CREATE INDEX idx\_orders\_mecanico ON work\_orders(mecanico\_asignado\_id);

CREATE INDEX idx\_orders\_estado ON work\_orders(estado);

CREATE INDEX idx\_orders\_fecha\_creacion ON work\_orders(fecha\_creacion);

## 3.11. OT\_ESTADOS (work\_order\_statuses)

────────────────────────────────────────────────────────────────

CREATE TABLE work\_order\_statuses (

id SERIAL PRIMARY KEY,
ot\_id INTEGER NOT NULL REFERENCES work\_orders(id) ON DELETE CASCADE,
estado\_anterior VARCHAR(30),
estado\_nuevo VARCHAR(30) NOT NULL,
usuario\_id INTEGER NOT NULL REFERENCES users(id),
observacion TEXT,
fecha\_cambio TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE INDEX idx\_ot\_statuses\_ot ON work\_order\_statuses(ot\_id);

CREATE INDEX idx\_ot\_statuses\_fecha ON work\_order\_statuses(fecha\_cambio);

## 3.12. OT\_FOTOS (work\_order\_photos)

────────────────────────────────────────────────────────────────

CREATE TABLE work\_order\_photos (

id SERIAL PRIMARY KEY,
ot\_id INTEGER REFERENCES work\_orders(id) ON DELETE CASCADE,
ingreso\_id INTEGER REFERENCES vehicle\_entries(id) ON DELETE CASCADE,
tipo VARCHAR(30) NOT NULL, -- 'ingreso', 'proceso', 'salida', 'daño'
url VARCHAR(500) NOT NULL,
thumbnail\_url VARCHAR(500),
descripcion TEXT,
tomada\_por INTEGER NOT NULL REFERENCES users(id),
fecha\_captura TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
metadata JSONB, -- Info adicional: ubicación, dimensiones, etc.
CHECK (tipo IN ('ingreso', 'proceso', 'salida', 'daño', 'repuesto', 'diagnostico'))

);

CREATE INDEX idx\_photos\_ot ON work\_order\_photos(ot\_id);

CREATE INDEX idx\_photos\_ingreso ON work\_order\_photos(ingreso\_id);

CREATE INDEX idx\_photos\_tipo ON work\_order\_photos(tipo);

## 3.13. PAUSAS (work\_pauses)

────────────────────────────────────────────────────────────────

CREATE TABLE work\_pauses (

id SERIAL PRIMARY KEY,
ot\_id INTEGER NOT NULL REFERENCES work\_orders(id) ON DELETE CASCADE,
motivo VARCHAR(100) NOT NULL,
descripcion TEXT,
fecha\_inicio TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,
fecha\_fin TIMESTAMP,
duracion\_minutos INTEGER GENERATED ALWAYS AS
(EXTRACT(EPOCH FROM (fecha\_fin - fecha\_inicio)) / 60) STORED,
registrado\_por INTEGER NOT NULL REFERENCES users(id),
observaciones TEXT,
CHECK (motivo IN ('espera\_repuesto', 'espera\_autorizacion', 'falta\_herramienta',
'problema\_tecnico', 'otro'))

);

CREATE INDEX idx\_pauses\_ot ON work\_pauses(ot\_id);

CREATE INDEX idx\_pauses\_fecha\_inicio ON work\_pauses(fecha\_inicio);

## 3.14. REPUESTOS (spare\_parts)

────────────────────────────────────────────────────────────────

CREATE TABLE spare\_parts (

id SERIAL PRIMARY KEY,
codigo VARCHAR(50) UNIQUE NOT NULL,
nombre VARCHAR(150) NOT NULL,
descripcion TEXT,
marca VARCHAR(50),
categoria VARCHAR(50) NOT NULL,
subcategoria VARCHAR(50),
unidad\_medida VARCHAR(20) DEFAULT 'unidad',
stock\_actual INTEGER DEFAULT 0,
stock\_minimo INTEGER DEFAULT 5,
stock\_maximo INTEGER DEFAULT 100,
precio\_unitario DECIMAL(10,2) NOT NULL,
ubicacion\_fisica VARCHAR(100),
proveedor VARCHAR(100),
activo BOOLEAN DEFAULT true,
fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
fecha\_modificacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
CHECK (stock\_actual >= 0),
CHECK (categoria IN ('motor', 'frenos', 'suspension', 'electrico',
'transmision', 'neumaticos', 'lubricantes', 'filtros', 'otros'))

);

CREATE INDEX idx\_parts\_codigo ON spare\_parts(codigo);

CREATE INDEX idx\_parts\_categoria ON spare\_parts(categoria);

CREATE INDEX idx\_parts\_stock ON spare\_parts(stock\_actual);

## 3.15. OT\_REPUESTOS (work\_order\_spare\_parts)

────────────────────────────────────────────────────────────────

CREATE TABLE work\_order\_spare\_parts (

id SERIAL PRIMARY KEY,
ot\_id INTEGER NOT NULL REFERENCES work\_orders(id) ON DELETE CASCADE,
repuesto\_id INTEGER NOT NULL REFERENCES spare\_parts(id),
cantidad INTEGER NOT NULL,
precio\_unitario DECIMAL(10,2) NOT NULL,
precio\_total DECIMAL(10,2) GENERATED ALWAYS AS
(cantidad \* precio\_unitario) STORED,
aplicado BOOLEAN DEFAULT false,
fecha\_solicitud TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
solicitado\_por INTEGER NOT NULL REFERENCES users(id),
entregado\_por INTEGER REFERENCES users(id),
fecha\_entrega TIMESTAMP,
observaciones TEXT,
CHECK (cantidad > 0)

);

CREATE INDEX idx\_ot\_parts\_ot ON work\_order\_spare\_parts(ot\_id);

CREATE INDEX idx\_ot\_parts\_repuesto ON work\_order\_spare\_parts(repuesto\_id);

## 3.16. MOVIM\_REPUESTOS (spare\_part\_movements)

────────────────────────────────────────────────────────────────

CREATE TABLE spare\_part\_movements (

id SERIAL PRIMARY KEY,
repuesto\_id INTEGER NOT NULL REFERENCES spare\_parts(id),
tipo\_movimiento VARCHAR(20) NOT NULL,
cantidad INTEGER NOT NULL,
stock\_anterior INTEGER NOT NULL,
stock\_nuevo INTEGER NOT NULL,
precio\_unitario DECIMAL(10,2),
costo\_total DECIMAL(10,2) GENERATED ALWAYS AS
(cantidad \* precio\_unitario) STORED,
ot\_id INTEGER REFERENCES work\_orders(id),
usuario\_id INTEGER NOT NULL REFERENCES users(id),
documento\_referencia VARCHAR(50),
observaciones TEXT,
fecha\_movimiento TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
CHECK (tipo\_movimiento IN ('entrada', 'salida', 'ajuste', 'devolucion')),
CHECK (cantidad != 0)

);

CREATE INDEX idx\_movements\_repuesto ON spare\_part\_movements(repuesto\_id);

CREATE INDEX idx\_movements\_tipo ON spare\_part\_movements(tipo\_movimiento);

CREATE INDEX idx\_movements\_fecha ON spare\_part\_movements(fecha\_movimiento);

## 3.17. CONTROL\_LLAVES (key\_control)

────────────────────────────────────────────────────────────────

CREATE TABLE key\_control (

id SERIAL PRIMARY KEY,
vehiculo\_id INTEGER NOT NULL REFERENCES vehicles(id),
ingreso\_id INTEGER REFERENCES vehicle\_entries(id),
numero\_llave VARCHAR(20),
estado VARCHAR(20) NOT NULL DEFAULT 'disponible',
ubicacion\_actual VARCHAR(100),
fecha\_entrega TIMESTAMP,
responsable\_entrega INTEGER REFERENCES users(id),
fecha\_devolucion TIMESTAMP,
responsable\_devolucion INTEGER REFERENCES users(id),
observaciones TEXT,
fecha\_registro TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
CHECK (estado IN ('disponible', 'entregada', 'en\_uso', 'extraviada'))

);

CREATE INDEX idx\_keys\_vehiculo ON key\_control(vehiculo\_id);

CREATE INDEX idx\_keys\_estado ON key\_control(estado);

CREATE INDEX idx\_keys\_ingreso ON key\_control(ingreso\_id);

## 3.18. DOCUMENTOS (documents)

────────────────────────────────────────────────────────────────

CREATE TABLE documents (

id SERIAL PRIMARY KEY,
ingreso\_id INTEGER NOT NULL REFERENCES vehicle\_entries(id) ON DELETE CASCADE,
ot\_id INTEGER REFERENCES work\_orders(id) ON DELETE CASCADE,
tipo\_documento VARCHAR(50) NOT NULL,
nombre\_archivo VARCHAR(255) NOT NULL,
url VARCHAR(500) NOT NULL,
formato VARCHAR(10) NOT NULL,
tamaño\_bytes INTEGER,
descripcion TEXT,
subido\_por INTEGER NOT NULL REFERENCES users(id),
fecha\_subida TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
CHECK (tipo\_documento IN ('permiso\_circulacion', 'seguro', 'revision\_tecnica',
'informe\_mecanico', 'cotizacion', 'factura',
'autorización', 'otro')),
CHECK (formato IN ('pdf', 'jpg', 'png', 'docx', 'xlsx'))

);

CREATE INDEX idx\_documents\_ingreso ON documents(ingreso\_id);

CREATE INDEX idx\_documents\_ot ON documents(ot\_id);

CREATE INDEX idx\_documents\_tipo ON documents(tipo\_documento);

## 3.19. NOTIFICACIONES (notifications)

────────────────────────────────────────────────────────────────

CREATE TABLE notifications (

id SERIAL PRIMARY KEY,
usuario\_id INTEGER NOT NULL REFERENCES users(id) ON DELETE CASCADE,
tipo VARCHAR(30) NOT NULL,
titulo VARCHAR(150) NOT NULL,
mensaje TEXT NOT NULL,
enlace VARCHAR(200),
referencia\_tipo VARCHAR(50), -- 'ot', 'ingreso', 'vehiculo', etc.
referencia\_id INTEGER,
prioridad VARCHAR(20) DEFAULT 'normal',
leida BOOLEAN DEFAULT false,
fecha\_lectura TIMESTAMP,
enviada\_email BOOLEAN DEFAULT false,
fecha\_envio\_email TIMESTAMP,
fecha\_creacion TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
fecha\_expiracion TIMESTAMP,
CHECK (tipo IN ('info', 'advertencia', 'error', 'exito', 'tarea', 'recordatorio')),
CHECK (prioridad IN ('baja', 'normal', 'alta', 'urgente'))

);

CREATE INDEX idx\_notifications\_usuario ON notifications(usuario\_id);

CREATE INDEX idx\_notifications\_leida ON notifications(leida);

CREATE INDEX idx\_notifications\_fecha ON notifications(fecha\_creacion);

## 3.20. AUDITORIA (audit\_logs)

────────────────────────────────────────────────────────────────

CREATE TABLE audit\_logs (

id SERIAL PRIMARY KEY,
usuario\_id INTEGER REFERENCES users(id),
accion VARCHAR(50) NOT NULL,
tabla\_afectada VARCHAR(50) NOT NULL,
registro\_id INTEGER,
datos\_anteriores JSONB,
datos\_nuevos JSONB,
ip\_address VARCHAR(45),
user\_agent TEXT,
metodo\_http VARCHAR(10),
endpoint VARCHAR(200),
fecha\_hora TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,
CHECK (accion IN ('CREATE', 'UPDATE', 'DELETE', 'LOGIN', 'LOGOUT'))

);

CREATE INDEX idx\_audit\_usuario ON audit\_logs(usuario\_id);

CREATE INDEX idx\_audit\_tabla ON audit\_logs(tabla\_afectada);

CREATE INDEX idx\_audit\_fecha ON audit\_logs(fecha\_hora);

CREATE INDEX idx\_audit\_accion ON audit\_logs(accion);

4. RELACIONES Y CARDINALIDAD

┌─────────────────────────┬────────────┬─────────────────────────┬────────────┐
│ Tabla Origen │ Card. │ Tabla Destino │ Card. │
├─────────────────────────┼────────────┼─────────────────────────┼────────────┤
│ users │ 1 │ vehicle\_entries │ N │
│ users │ 1 │ work\_orders │ N │
│ users │ N │ roles │ 1 │
│ roles │ N │ permissions │ M │
│ regions │ 1 │ workshops │ N │
│ regions │ 1 │ vehicles │ N │
│ workshops │ 1 │ workshop\_schedules │ N │
│ workshops │ 1 │ vehicle\_entries │ N │
│ vehicles │ 1 │ vehicle\_entries │ N │
│ vehicles │ 1 │ key\_control │ N │
│ vehicle\_entries │ 1 │ work\_orders │ N │
│ vehicle\_entries │ 1 │ documents │ N │
│ vehicle\_entries │ 1 │ work\_order\_photos │ N │
│ work\_orders │ 1 │ work\_order\_statuses │ N │
│ work\_orders │ 1 │ work\_order\_photos │ N │
│ work\_orders │ 1 │ work\_pauses │ N │
│ work\_orders │ N │ spare\_parts │ M │
│ spare\_parts │ 1 │ spare\_part\_movements │ N │
│ users │ 1 │ notifications │ N │
│ users │ 1 │ audit\_logs │ N │
└─────────────────────────┴────────────┴─────────────────────────┴────────────┘

5. REGLAS DE NEGOCIO IMPLEMENTADAS

## 5.1. Integridad Referencial

✓ Todas las FK con ON DELETE configurado apropiadamente

✓ CASCADE para tablas de detalle/histórico

✓ RESTRICT para tablas maestras

## 5.2. Validaciones con CHECK

✓ Estados válidos (ENUM simulado)

✓ Rangos numéricos (años, cantidades)

✓ Fechas lógicas

## 5.3. Campos Calculados

✓ costo\_total en work\_orders

✓ precio\_total en work\_order\_spare\_parts

✓ duracion\_minutos en work\_pauses

✓ GENERATED ALWAYS AS (automático)

## 5.4. Auditoría Automática

✓ fecha\_creacion en todas las tablas

✓ fecha\_modificacion actualizable

✓ Registro de usuario creador/modificador

## 5.5. Soft Delete

✓ Campo 'activo' en tablas maestras

✓ No se eliminan registros físicamente

6. ÍNDICES Y OPTIMIZACIÓN

## 6.1. Índices Principales

✓ Primary Keys (SERIAL con índice automático)

✓ Foreign Keys (índices explícitos)

✓ Campos de búsqueda frecuente (email, rut, patente, código)

✓ Campos de filtrado (estado, fecha, activo)

## 6.2. Performance

✓ JSONB para datos semi-estructurados (más rápido que JSON)

✓ Índices compuestos donde sea necesario

✓ TIMESTAMP para búsquedas temporales eficientes

7. TAMAÑO ESTIMADO DE LA BASE DE DATOS

**Estimación para 1 año de operación:**

┌───────────────────────────┬──────────────┬──────────────┬──────────────┐
│ Tabla │ Registros │ KB/Registro │ Tamaño Total │
├───────────────────────────┼──────────────┼──────────────┼──────────────┤
│ users │ 100 │ 1 │ 100 KB │
│ vehicles │ 500 │ 1 │ 500 KB │
│ vehicle\_entries │ 12,000 │ 2 │ 24 MB │
│ work\_orders │ 15,000 │ 2 │ 30 MB │
│ work\_order\_statuses │ 60,000 │ 0.5 │ 30 MB │
│ work\_order\_photos │ 50,000 │ 0.5 │ 25 MB │
│ spare\_parts │ 1,000 │ 1 │ 1 MB │
│ spare\_part\_movements │ 30,000 │ 1 │ 30 MB │
│ notifications │ 100,000 │ 0.5 │ 50 MB │
│ audit\_logs │ 500,000 │ 1 │ 500 MB │
│ Otros │ - │ - │ 10 MB │
├───────────────────────────┼──────────────┼──────────────┼──────────────┤
│ TOTAL │ - │ - │ ~700 MB │
└───────────────────────────┴──────────────┴──────────────┴──────────────┘

Nota: Las imágenes se almacenan en Cloudinary (no en BD)

SCRIPT DE INICIALIZACIÓN (schema.prisma)

Este modelo se implementará usando Prisma ORM.

El archivo schema.prisma se creará en el siguiente entregable.

9. CONSULTAS FRECUENTES OPTIMIZADAS

## 9.1. Vehículos en taller por región

────────────────────────────────────────────────────────────────

SELECT

r.nombre as region,
COUNT(DISTINCT ve.vehiculo\_id) as vehiculos\_en\_taller

FROM vehicle\_entries ve

JOIN vehicles v ON ve.vehiculo\_id = v.id

JOIN regions r ON v.region\_id = r.id

WHERE ve.estado\_ingreso IN ('en\_recepcion', 'en\_taller', 'pausado')

GROUP BY r.nombre;

## 9.2. OT pendientes por mecánico

────────────────────────────────────────────────────────────────

SELECT

u.nombre || ' ' || u.apellido as mecanico,
COUNT(\*) as ot\_pendientes,
SUM(wo.horas\_estimadas) as horas\_totales

FROM work\_orders wo

JOIN users u ON wo.mecanico\_asignado\_id = u.id

WHERE wo.estado IN ('asignada', 'en\_progreso')

GROUP BY u.id, u.nombre, u.apellido;

## 9.3. Repuestos bajo stock mínimo

────────────────────────────────────────────────────────────────

SELECT

codigo,
nombre,
stock\_actual,
stock\_minimo,
(stock\_minimo - stock\_actual) as faltante

FROM spare\_parts

WHERE stock\_actual < stock\_minimo

AND activo = true

ORDER BY (stock\_minimo - stock\_actual) DESC;

## 9.4. Tiempo promedio de reparación por tipo

────────────────────────────────────────────────────────────────

SELECT

tipo\_servicio,
COUNT(\*) as total\_ingresos,
AVG(EXTRACT(EPOCH FROM duracion\_total)/3600) as horas\_promedio

FROM vehicle\_entries

WHERE estado\_ingreso = 'completado'

AND duracion\_total IS NOT NULL

GROUP BY tipo\_servicio;

10. RESPALDOS Y RECUPERACIÓN

## 10.1. Estrategia de Backup

• Daily full backup (automatizado por Neon/Supabase)

• Point-in-time recovery disponible

• Retención: 7 días en plan gratuito

## 10.2. Datos Críticos

• users (credenciales)

• vehicles (flota)

• vehicle\_entries (registros de ingreso)

• work\_orders (órdenes de trabajo)

• spare\_parts (inventario)

11. MIGRACIÓN DESDE EXCEL

Se debe crear un proceso de migración para datos existentes en Excel:

Vehículos de la flota → vehicles

Historial de ingresos → vehicle\_entries (últimos 6 meses)

Inventario actual → spare\_parts

Usuarios actuales → users (con roles asignados)

12. PRÓXIMOS PASOS

□ Crear archivo schema.prisma con este modelo

□ Implementar migraciones iniciales

□ Crear seeders para datos de catálogo (roles, permisos, regiones)

□ Documentar API endpoints basados en este modelo

□ Crear diagramas visuales (usando dbdiagram.io o similar)

VALIDACIÓN Y APROBACIÓN

Este modelo de datos ha sido diseñado para cumplir con todos los

requerimientos funcionales especificados en el documento ERS.

**Cubre:**

✓ RF-01: Registro de ingreso/salida con fotos

✓ RF-02: Validación de documentación

✓ RF-03: Generación de OT

✓ RF-04: Asignación de mecánicos y tiempos

✓ RF-05: Estados de reparación

✓ RF-06: Control de inventario

✓ RF-07: Control de llaves

✓ RF-08: Generación de reportes

✓ RF-09: Notificaciones

✓ RF-10: Autenticación y roles

**Elaborado por:**

Joaquín Marín - Gerente de Proyecto

Benjamin Vilches - Arquitecto de Software / Backend Developer

Fecha: 13/10/2025

**Aprobado por:**

[ ] Alexis González - Patrocinador

[ ] Fabián Álvarez - Docente

Firma: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Fecha: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_