



REPORTE DE PRÁCTICA NO. 2

NOMBRE DE LA PRÁCTICA: Flotilla

ALUMNO:

Janett Hernández
Adán Villeda Trejo



1. Introducción

La gestión eficiente de una flotilla vehicular requiere un control de distintos aspectos como el mantenimiento, documentación de cada unidad, los conductores asignados y las rutas recorridas. Para cumplir con este propósito, se diseñó una base de datos que integra las entidades principales relacionadas con la operación de una flotilla de autos.

El modelo propuesto incluye tablas para registrar mantenimientos, documentación vehicular, autos, conductores y rutas. Con la implementación de llaves primarias y foráneas, se asegura la gestión de los datos y se facilita la consulta de información relevante, como el historial de un vehículo, los documentos vigentes de cada unidad o las rutas asignadas a un conductor específico.

Marco Teorico

Procedimientos almacenados (Procedure)

Un procedimiento almacenado es un bloque de código PL/SQL o SQL que se almacena en la base de datos y puede ser ejecutado posteriormente mediante una llamada específica. Sirve para automatizar tareas recurrentes en la gestión de datos, aceptar parámetros de entrada y definir operaciones complejas que combinan varias instrucciones SQL. Los procedimientos pueden ser invocados desde aplicaciones externas o internamente desde otros procedimientos, facilitando la reutilización y centralización de la lógica de negocio.

Funciones (Function)

Las funciones son similares a los procedimientos almacenados pero se distinguen por siempre retornar un valor como resultado de su ejecución. Se utilizan para realizar cálculos, transformar datos o validar información, y su valor de retorno puede ser empleado directamente en consultas SQL, expresiones o incluso dentro de otros procedimientos y funciones. Además, las funciones aceptan parámetros y pueden tener lógica condicional, repetitiva y operaciones complejas.

Estructuras de control condicionales y repetitivas

Las estructuras de control condicionales permiten tomar decisiones en el flujo del programa, ejecutando diferentes bloques de código según el resultado de una condición lógica (por ejemplo, IF, CASE). Las estructuras repetitivas, por otro lado, facilitan la ejecución de instrucciones múltiples veces, ya sea un número fijo de iteraciones o mientras se cumpla una condición (por ejemplo, LOOP, WHILE, FOR). Estas estructuras son esenciales para procesar lotes de datos y controlar el flujo de ejecución en procedimientos y funciones.

Disparadores (Triggers)

Los disparadores son bloques de código que se ejecutan automáticamente en respuesta a ciertos eventos en la base de datos, como inserciones, actualizaciones o eliminaciones en una tabla. Permiten automatizar tareas de auditoría, validación o actualización sin necesidad de intervención manual, asegurando la integridad y coherencia de los datos. Los triggers se definen para actuar antes o después de los eventos especificados, y pueden incluir lógica condicional y llamadas a procedimientos o funciones.

2. Modelo Entidad - Relación

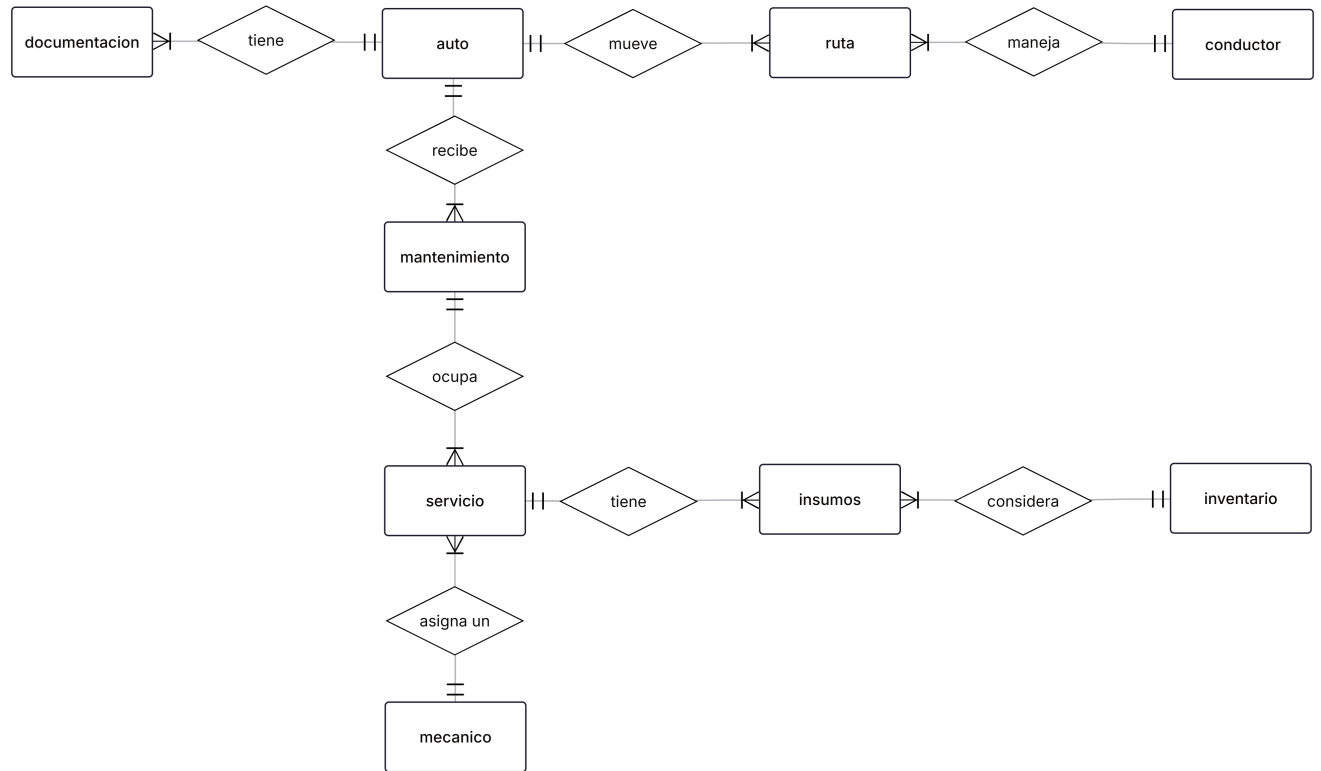


Figure 1: MER

3. Modelo Relacional

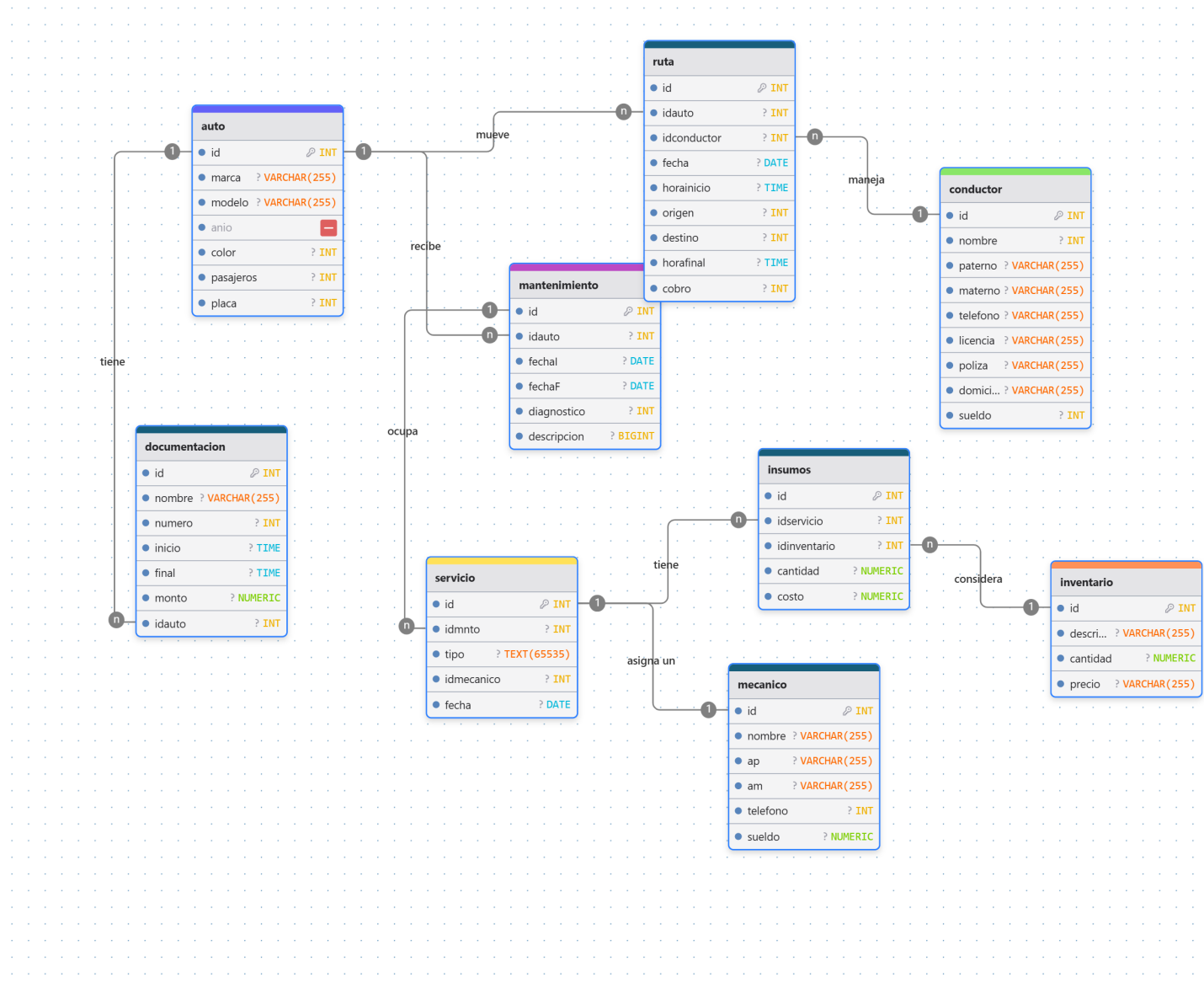


Figure 2: MR

4. Sentencias SQL(create table)

```
create database flotilla;  
use flotilla;
```

```
CREATE TABLE 'auto' (  
  'id' INT NOT NULL,  
  'marca' VARCHAR(255),  
  'modelo' VARCHAR(255),  
  'anio' INT,  
  'color' VARCHAR(255),  
  'pasajeros' INT,  
  'placa' VARCHAR(255),  
  PRIMARY KEY('id')  
);
```

```
CREATE TABLE 'conductor' (  
  'id' INT NOT NULL,  
  'nombre' VARCHAR(255),  
  'apellidoPaterno' VARCHAR(255),  
  'apellidoMaterno' VARCHAR(255),  
  'telefono' VARCHAR(255),  
  'licencia' VARCHAR(255),  
  'poliza' VARCHAR(255),  
  'domicilio' VARCHAR(255),  
  'sueldo' DECIMAL,  
  PRIMARY KEY('id')  
);
```

```
CREATE TABLE 'inventario' (  
  'id' INT NOT NULL,  
  'descripcion' VARCHAR(255),  
  'cantidad' INT,  
  'precio' DECIMAL,  
  PRIMARY KEY('id')  
);
```

```
CREATE TABLE 'mecanico' ( 'id' INT NOT NULL,  
  'nombre' VARCHAR(255),  
  'apellidoPaterno' VARCHAR(255),  
  'apellidoMaterno' VARCHAR(255),  
  'telefono' VARCHAR(255),  
  'domicilio' VARCHAR(255),  
  'sueldo' NUMERIC,  
  PRIMARY KEY('id') );
```

```
CREATE TABLE 'mantenimiento' ( 'id' INT NOT NULL,  
  'fechaInicio' DATE,  
  'fechaFinal' DATE,  
  'diagnostico' VARCHAR(255),  
  'descripcion' VARCHAR(255),  
  'idAuto' INT,  
  PRIMARY KEY('id'),
```

```
foreign key(idAuto) references auto(id) );
```

```
CREATE TABLE 'servicio' (  
  'id' INT NOT NULL,  
  'tipo' VARCHAR(255),  
  'fecha' DATE,  
  'idMecanico' INT,  
  'idMantenimiento' INT,  
  PRIMARY KEY('id'),  
  foreign key(idMecanico) references mecanico(id),  
  foreign key(idMantenimiento) references mantenimiento(id) );
```

```
CREATE TABLE 'ruta' (  
  'id' INT NOT NULL,  
  'fecha' DATE,  
  'horaInicio' TIME,  
  'origen' VARCHAR(255),  
  'destino' VARCHAR(255),  
  'horaFinal' TIME,  
  'cobro' DECIMAL,  
  'idConductor' INT,  
  'idAuto' INT,  
  PRIMARY KEY('id'),  
  foreign key(idConductor) references conductor(id),  
  foreign key(idAuto) references auto(id) );
```

```
CREATE TABLE 'insumos' (  
  'id' INT NOT NULL,  
  'cantidad' INT,  
  'costo' DECIMAL,  
  'idInventario' INT,  
  'idServicio' INT,  
  PRIMARY KEY('id'),  
  foreign key(idInventario) references inventario(id),  
  foreign key(idServicio) references servicio(id) );
```

```
CREATE TABLE 'documentacion' (  
  'id' INT NOT NULL,  
  'nombre' VARCHAR(255),  
  'numero' INT,  
  'inicio' DATE,  
  'final' DATE,  
  'monto' NUMERIC,  
  'idAuto' INT,  
  PRIMARY KEY('id'),  
  foreign key(idAuto) references auto(id) );
```

5. Sentencias SQL(insert into)

– AUTO

INSERT INTO auto (id, marca, modelo, anio, color, pasajeros, placa) VALUES

```
(1, 'Toyota', 'Corolla', 2018, 'Rojo', 5, 'ABC123'),
(2, 'Nissan', 'Versa', 2020, 'Azul', 5, 'BCD234'),
(3, 'Honda', 'Civic', 2019, 'Negro', 5, 'CDE345'),
(4, 'Chevrolet', 'Aveo', 2017, 'Blanco', 5, 'DEF456'),
(5, 'Ford', 'Focus', 2016, 'Gris', 5, 'EFG567'),
(6, 'Hyundai', 'Elantra', 2021, 'Rojo', 5, 'FGH678'),
(7, 'Kia', 'Rio', 2019, 'Azul', 5, 'GHI789'),
(8, 'Volkswagen', 'Jetta', 2015, 'Negro', 5, 'HIJ890'),
(9, 'Mazda', '3', 2022, 'Blanco', 5, 'IJK901'),
(10, 'Renault', 'Logan', 2018, 'Gris', 5, 'JKL012');
```

– CONDUCTOR

INSERT INTO conductor (id, nombre, apellidoPaterno, apellidoMaterno, telefono, licencia, poliza, domicilio, sueldo) VALUES

```
(1, 'Juan', 'Perez', 'Lopez', '5551111111', 'LIC123', 'POL123', 'Calle 1', 12000.50),
(2, 'Pedro', 'Martinez', 'Garcia', '5552222222', 'LIC124', 'POL124', 'Calle 2', 11000.00),
(3, 'Luis', 'Hernandez', 'Sanchez', '5553333333', 'LIC125', 'POL125', 'Calle 3', 11500.75),
(4, 'Carlos', 'Diaz', 'Fernandez', '5554444444', 'LIC126', 'POL126', 'Calle 4', 11800.20),
(5, 'Miguel', 'Gutierrez', 'Torres', '5555555555', 'LIC127', 'POL127', 'Calle 5', 13000.00),
(6, 'Antonio', 'Ramirez', 'Jimenez', '5556666666', 'LIC128', 'POL128', 'Calle 6', 12500.40),
(7, 'Jorge', 'Flores', 'Mendoza', '5557777777', 'LIC129', 'POL129', 'Calle 7', 12800.00),
(8, 'Ricardo', 'Cruz', 'Ortiz', '5558888888', 'LIC130', 'POL130', 'Calle 8', 13500.00),
(9, 'Manuel', 'Vargas', 'Moreno', '5559999999', 'LIC131', 'POL131', 'Calle 9', 14000.00),
(10, 'Francisco', 'Soto', 'Navarro', '5551010101', 'LIC132', 'POL132', 'Calle 10', 14500.60);
```

– INVENTARIO

INSERT INTO inventario (id, descripcion, cantidad, precio) VALUES

```
(1, 'Aceite 10W40', 50, 150.00),
(2, 'Filtro de aceite', 30, 100.00),
(3, 'Filtro de aire', 20, 120.00),
(4, 'Bujía', 100, 60.00),
(5, 'Llanta', 40, 1200.00),
(6, 'Batería', 15, 2500.00),
(7, 'Pastillas de freno', 60, 400.00),
(8, 'Amortiguador', 25, 1800.00),
(9, 'Anticongelante', 45, 90.00),
(10, 'Balatas', 70, 350.00);
```

– MECÁNICO

INSERT INTO mecanico (id, nombre, apellidoPaterno, apellidoMaterno, telefono, domicilio, sueldo) VALUES

```
(1, 'Jose', 'Ruiz', 'Hernandez', '5551112222', 'Taller 1', 9000.00),
(2, 'Ramon', 'Lara', 'Perez', '5552223333', 'Taller 2', 8500.50),
(3, 'Mario', 'Campos', 'Ortiz', '5553334444', 'Taller 3', 8700.00),
(4, 'Raul', 'Mejia', 'Solis', '5554445555', 'Taller 4', 9200.00),
(5, 'Hugo', 'Aguilar', 'Diaz', '5555556666', 'Taller 5', 8800.00),
(6, 'Julio', 'Santos', 'Garcia', '5556667777', 'Taller 6', 9500.00),
(7, 'Victor', 'Dominguez', 'Lopez', '5557778888', 'Taller 7', 9700.00),
(8, 'Oscar', 'Reyes', 'Mendez', '5558889999', 'Taller 8', 9300.00),
(9, 'Felipe', 'Paredes', 'Cruz', '5559990000', 'Taller 9', 9100.00),
```


(10, 'Eduardo', 'Castro', 'Navarro', '5551010102', 'Taller 10', 9400.00);

– MANTENIMIENTO

INSERT INTO mantenimiento (id, fechaInicio, fechaFinal, diagnostico, descripcion, idAuto) VALUES

(1, '2023-01-10', '2023-01-12', 'Cambio aceite', 'Mantenimiento preventivo', 1),
(2, '2023-02-05', '2023-02-07', 'Cambio frenos', 'Reemplazo balatas', 2),
(3, '2023-03-15', '2023-03-16', 'Batería descargada', 'Cambio de batería', 3),
(4, '2023-04-20', '2023-04-22', 'Revisión suspensión', 'Cambio amortiguadores', 4),
(5, '2023-05-01', '2023-05-02', 'Filtro sucio', 'Cambio de filtros', 5),
(6, '2023-06-10', '2023-06-11', 'Aceite bajo', 'Relleno aceite', 6),
(7, '2023-07-12', '2023-07-14', 'Problemas eléctricos', 'Revisión sistema eléctrico', 7),
(8, '2023-08-01', '2023-08-03', 'Llanta dañada', 'Cambio de llanta', 8),
(9, '2023-09-05', '2023-09-06', 'Frenos desgastados', 'Reemplazo frenos', 9),
(10, '2023-10-12', '2023-10-13', 'Revisión general', 'Servicio completo', 10);

– SERVICIO

INSERT INTO servicio (id, tipo, fecha, idMecanico, idMantenimiento) VALUES

(1, 'Cambio aceite', '2023-01-11', 1, 1),
(2, 'Cambio frenos', '2023-02-06', 2, 2),
(3, 'Cambio batería', '2023-03-15', 3, 3),
(4, 'Cambio amortiguadores', '2023-04-21', 4, 4),
(5, 'Cambio filtros', '2023-05-01', 5, 5),
(6, 'Relleno aceite', '2023-06-10', 6, 6),
(7, 'Sistema eléctrico', '2023-07-13', 7, 7),
(8, 'Cambio llanta', '2023-08-02', 8, 8),
(9, 'Reemplazo frenos', '2023-09-05', 9, 9),
(10, 'Servicio completo', '2023-10-12', 10, 10);

– RUTA

INSERT INTO ruta (id, fecha, horaInicio, origen, destino, horaFinal, cobro, idConductor, idAuto) VALUES

(1, '2023-01-01', '08:00:00', 'CDMX', 'Toluca', '09:30:00', 500.00, 1, 1),
(2, '2023-01-02', '09:00:00', 'CDMX', 'Puebla', '11:00:00', 600.00, 2, 2),
(3, '2023-01-03', '07:30:00', 'CDMX', 'Cuernavaca', '08:45:00', 400.00, 3, 3),
(4, '2023-01-04', '10:00:00', 'Toluca', 'Querétaro', '12:30:00', 700.00, 4, 4),
(5, '2023-01-05', '11:00:00', 'CDMX', 'Toluca', '12:30:00', 500.00, 5, 5),
(6, '2023-01-06', '12:00:00', 'CDMX', 'Puebla', '14:00:00', 600.00, 6, 6),
(7, '2023-01-07', '13:00:00', 'CDMX', 'Toluca', '14:30:00', 500.00, 7, 7),
(8, '2023-01-08', '14:00:00', 'Toluca', 'Querétaro', '16:30:00', 700.00, 8, 8),
(9, '2023-01-09', '15:00:00', 'CDMX', 'Cuernavaca', '16:15:00', 400.00, 9, 9),
(10, '2023-01-10', '16:00:00', 'CDMX', 'Toluca', '17:30:00', 500.00, 10, 10);

– INSUMOS

INSERT INTO insumos (id, cantidad, costo, idInventario, idServicio) VALUES

(1, 2, 300.00, 1, 1),
(2, 1, 100.00, 2, 2),
(3, 1, 2500.00, 6, 3),
(4, 2, 3600.00, 8, 4),
(5, 3, 360.00, 3, 5),
(6, 1, 150.00, 1, 6),
(7, 2, 800.00, 7, 7),
(8, 1, 1200.00, 5, 8),
(9, 4, 1400.00, 10, 9),
(10, 5, 450.00, 9, 10);

– DOCUMENTACION

INSERT INTO documentacion (id, nombre, numero, inicio, final, monto, idAuto) VALUES

(1, 'Tarjeta Circulación', 12345, '2022-01-01', '2023-01-01', 500.00, 1),
(2, 'Seguro', 54321, '2022-02-01', '2023-02-01', 3000.00, 2),
(3, 'Verificación', 67890, '2022-03-01', '2023-03-01', 400.00, 3),
(4, 'Tenencia', 98765, '2022-04-01', '2023-04-01', 1500.00, 4),
(5, 'Tarjeta Circulación', 19283, '2022-05-01', '2023-05-01', 500.00, 5),
(6, 'Seguro', 29384, '2022-06-01', '2023-06-01', 3200.00, 6),
(7, 'Verificación', 39485, '2022-07-01', '2023-07-01', 450.00, 7),
(8, 'Tenencia', 49586, '2022-08-01', '2023-08-01', 1600.00, 8),
(9, 'Seguro', 59687, '2022-09-01', '2023-09-01', 3500.00, 9),
(10, 'Verificación', 69788, '2022-10-01', '2023-10-01', 480.00, 10);

6. Sentencias Fragmentos

-Fragmento de vista para propietario

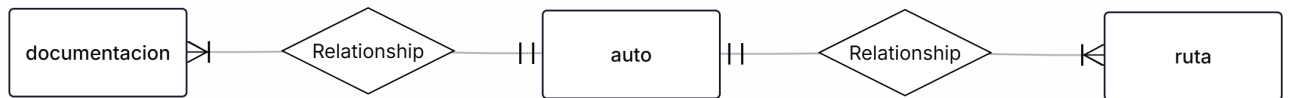


Figure 3: PC en el lugar del dueño

```
CREATE VIEW vistapropietario AS SELECT
auto.id,
auto.marca,
auto.modelo,
auto.anio,
auto.color,
auto.pasajeros,
auto.placa,
documentacion.nombre,
documentacion.numero,
documentacion.inicio,
documentacion.final,
documentacion.monto,
ruta.fecha,
ruta.horaInicio,
ruta.origen,
ruta.destino,
ruta.horaFinal,
ruta.cobro
FROM auto
LEFT JOIN documentacion ON auto.id = documentacion.idAuto
LEFT JOIN ruta ON auto.id = ruta.idAuto
ORDER BY auto.id, ruta.fecha;
```

-Fragmento de vista para mecanico

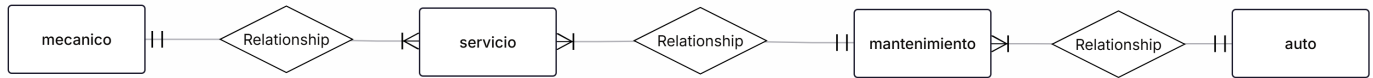


Figure 4: PC en el taller

```
CREATE VIEW vistamecanico AS SELECT
mecanico.id,
mecanico.nombre,
mecanico.apellidoPaterno,
mecanico.apellidoMaterno,
mecanico.telefono,
mecanico.domicilio,
mecanico.sueldo,
servicio.tipo,
servicio.fecha,
mantenimiento.fechaInicio,
mantenimiento.fechaFinal,
mantenimiento.diagnostico,
mantenimiento.descripcion,
auto.marca,
auto.modelo,
auto.anio,
auto.color,
auto.pasajeros,
auto.placa
FROM mecanico
LEFT JOIN servicio ON mecanico.id = servicio.idMecanico
LEFT JOIN mantenimiento ON servicio.idMantenimiento = mantenimiento.id
LEFT JOIN auto ON mantenimiento.idAuto = auto.id
ORDER BY mecanico.id, servicio.fecha;
SELECT * FROM vistamecanico;
SELECT * FROM vistamecanico WHERE id = 3;
```

-Fragmento de vista para chofer



Figure 5: PC a bordo de la unidad

```
CREATE VIEW vistachofer AS SELECT
conductor.id,
conductor.nombre,
conductor.apellidoPaterno,
conductor.apellidoMaterno,
conductor.telefono,
conductor.licencia,
conductor.poliza,
conductor.domicilio,
conductor.sueldo,
ruta.fecha,
ruta.horaInicio,
ruta.origen,
ruta.destino,
ruta.horaFinal,
ruta.cobro,
auto.marca,
auto.modelo,
auto.anio,
auto.color,
auto.pasajeros,
auto.placa
FROM conductor
LEFT JOIN ruta ON conductor.id = ruta.idConductor
LEFT JOIN auto ON ruta.idAuto = auto.id
ORDER BY conductor.id, ruta.fecha;
SELECT * FROM vistachofer WHERE id = 3;
```

-Fragmento de vista para almacen

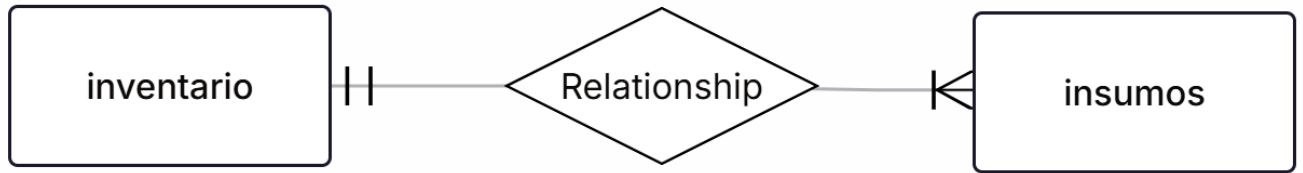


Figure 6: PC en el almacen

```
CREATE VIEW vistaalmacen AS SELECT
insumos.id,
insumos.cantidad,
insumos.costo,
inventario.descripcion,
inventario.cantidad AS cantidadInventario,
inventario.precio
FROM insumos
LEFT JOIN inventario ON insumos.idInventario = inventario.id
ORDER BY insumos.id;
SELECT * FROM vistaalmacen WHERE id = 3;
```

-Fragmento de vista administración

```
CREATE VIEW vistaadmin AS SELECT
auto.id,
auto.marca,
auto.modelo,
auto.anio,
auto.color,
auto.pasajeros,
auto.placa,
documentacion.nombre nombreDocumentacion,
documentacion.numero numeroDocumentacion,
documentacion.inicio inicioDocumentacion,
documentacion.final finalDocumentacion,
documentacion.monto montoDocumentacion,
ruta.fecha fechaRuta,
ruta.horaInicio,
ruta.origen,
ruta.destino,
ruta.horaFinal,
ruta.cobro cobroRuta,
conductor.id idConductor,
conductor.nombre nombreConductor,
conductor.apellidoPaterno apellidoPaternoConductor,
conductor.apellidoMaterno apellidoMaternoConductor,
conductor.telefono telefonoConductor,
conductor.licencia licenciaConductor,
conductor.poliza polizaConductor,
conductor.domicilio domicilioConductor,
conductor.sueldo sueldoConductor,
mantenimiento.fechaInicio fechaInicioMantenimiento,
mantenimiento.fechaFinal fechaFinalMantenimiento,
mantenimiento.diagnostico diagnosticoMantenimiento,
mantenimiento.descripcion descripcionMantenimiento,
servicio.tipo tipoServicio,
servicio.fecha fechaServicio,
mecanico.id idMecanico,
mecanico.nombre nombreMecanico,
mecanico.apellidoPaterno apellidoPaternoMecanico,
mecanico.apellidoMaterno apellidoMaternoMecanico,
mecanico.telefono telefonoMecanico,
mecanico.domicilio domicilioMecanico,
mecanico.sueldo sueldoMecanico,
insumos.id idInsumo,
insumos.cantidad cantidadInsumo,
insumos.costo costoInsumo,
inventario.descripcion descripcionInventario,
inventario.cantidad cantidadInventario,
inventario.precio precioInventario
FROM auto LEFT JOIN documentacion ON auto.id = documentacion.idAuto
LEFT JOIN ruta ON auto.id = ruta.idAuto
LEFT JOIN conductor ON ruta.idConductor = conductor.id
LEFT JOIN mantenimiento ON auto.id = mantenimiento.idAuto
LEFT JOIN servicio ON mantenimiento.id = servicio.idMantenimiento
```

```
LEFT JOIN mecanico ON servicio.idMecanico = mecanico.id
LEFT JOIN insumos ON servicio.id = insumos.idServicio
LEFT JOIN inventario ON insumos.idInventario = inventario.id
ORDER BY auto.id;
SELECT * FROM vistaadmin WHERE id = 3;
```


7. Conclusiones

La BD desarrollada constituye una solución para la administración de una flotilla vehicular, al integrar los registros de mantenimiento, documentación, vehículos, conductores y rutas. Su diseño facilita el almacenamiento y la consulta de información, al igual que también garantiza la coherencia e integridad de los datos.

Con esta estructura, se logra un mejor seguimiento de las operaciones de la flotilla, reduciendo la posibilidad de errores, agilizando la gestión de los recursos, brindando una base sólida para futuros proyectos, como reportes automatizados o sistemas de análisis de desempeño.