

UNIVERSIDAD AUTÓNOMA “GABRIEL RENE MORENO”

*FACULTAD DE INGENIERÍA Y CIENCIAS DE LA
COMPUTACIÓN Y TELECOMUNICACIONES*

INF552-SA



Primer Parcial

Materia: Arquitectura de Software

Semestre: I-2025

Docente: Ing. Veizaga Gonzales Josué Obed

Estudiante:

Montaño Claros Wilson Brandon	220001911	Ing. Sistemas
-------------------------------	-----------	---------------

Fecha: 23/04/2025

Contenido

1.	CAPTURA DE REQUISITOS.....	3
1.1.	REQUISITOS FUNCIONALES	3
1.1.1	IDENTIFICAR ACTORES Y CASOS DE USO.....	3
1.1.2	DETALLE DE CASOS DE USO.....	3
1.1.3	DIAGRAMA GENERAL DE CASOS DE USO.....	8
1.2.	REQUISITOS NO FUNCIONALES	8
2.	ANALISIS	9
2.1.	IDENTIFICION DE PAQUETES.....	9
2.2.	VISTA DE MODULOS	9
2.3.	ENCAPSULAMIENTO DE CASOS DE USO	10
3.	DISEÑO	11
3.1.	DISEÑO DE LA ARQUITECTURA LOGICA	11
3.2.	DISEÑO DE LA BASE DE DATOS	12
3.2.1	DISEÑO CONCEPTUAL	12
3.2.2	DISEÑO LOGICO	12
3.2.3	DISEÑO FISICO.....	13
3.3.	DISEÑO DE LA INTERFAZ	15
3.4.	DISEÑO DE DETALLE PROCEDIMENTAL.....	17

1. CAPTURA DE REQUISITOS

1.1. REQUISITOS FUNCIONALES

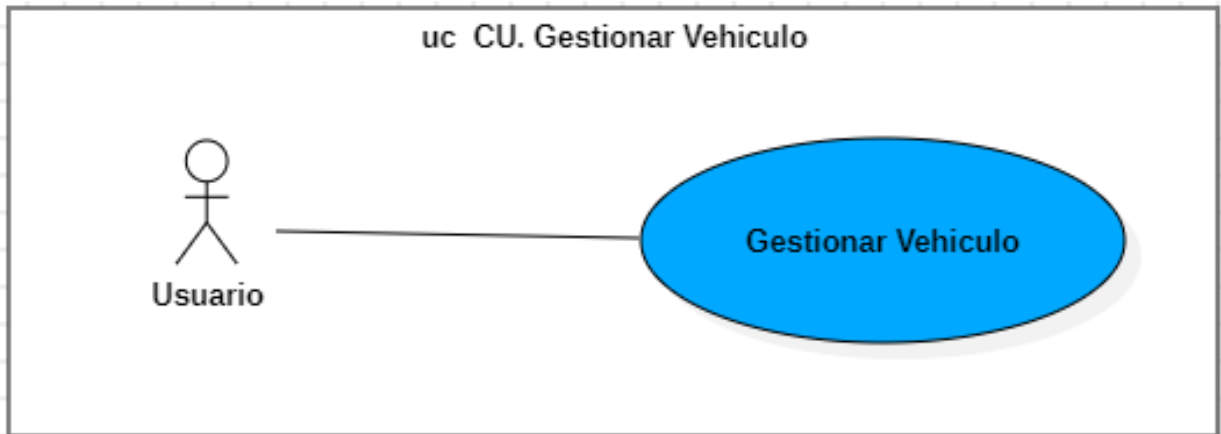
1.1.1 IDENTIFICAR ACTORES Y CASOS DE USO

Actores: Usuario

Casos de uso:

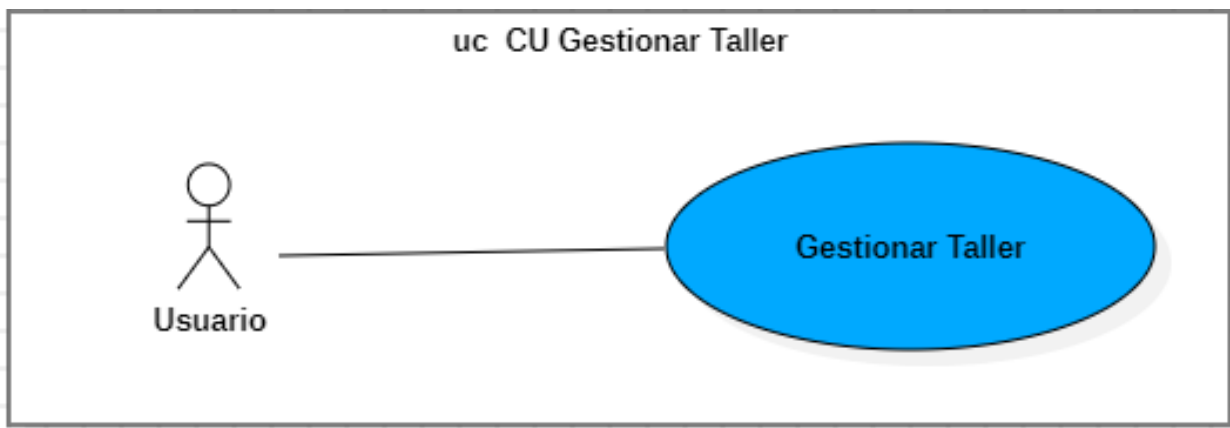
- UC1 Gestionar Vehículo
- UC2 Gestionar Taller
- UC3 Gestionar Tipo Mantenimiento
- UC4 Gestionar Mantenimiento
- UC5 Gestionar Gasto

1.1.2 DETALLE DE CASOS DE USO



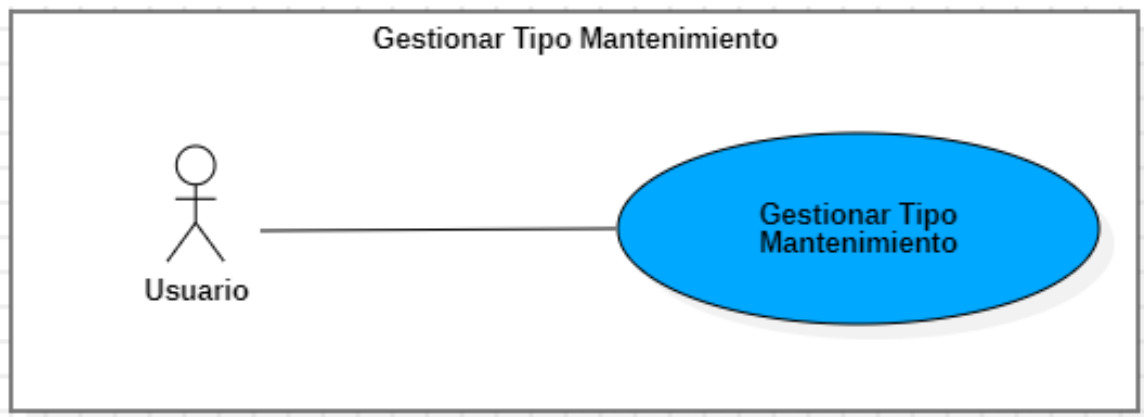
Caso de Uso		CU1. Gestionar Vehículo
Propósito	Gestionar Vehículo	
Descripcion	Permitirá gestionar los vehículos que se almacenaran en el dispositivo	
Actores	Usuario	
Actor Iniciador	Usuario	
Precondición	Ninguna	

Proceso	<ol style="list-style-type: none"> 1. Agregar Vehículo <ol style="list-style-type: none"> 1.1 Pulsar botón “Agregar” 1.2 Insertar datos 1.3 Guardar datos 2. Editar Vehículo <ol style="list-style-type: none"> 2.1 Pulsar botón “Editar” 2.2 Modificar datos 2.3 Actualizar datos 3. Eliminar Vehículo <ol style="list-style-type: none"> 3.1 Pulsar botón “Eliminar” 3.2 Eliminar vehículo
----------------	---



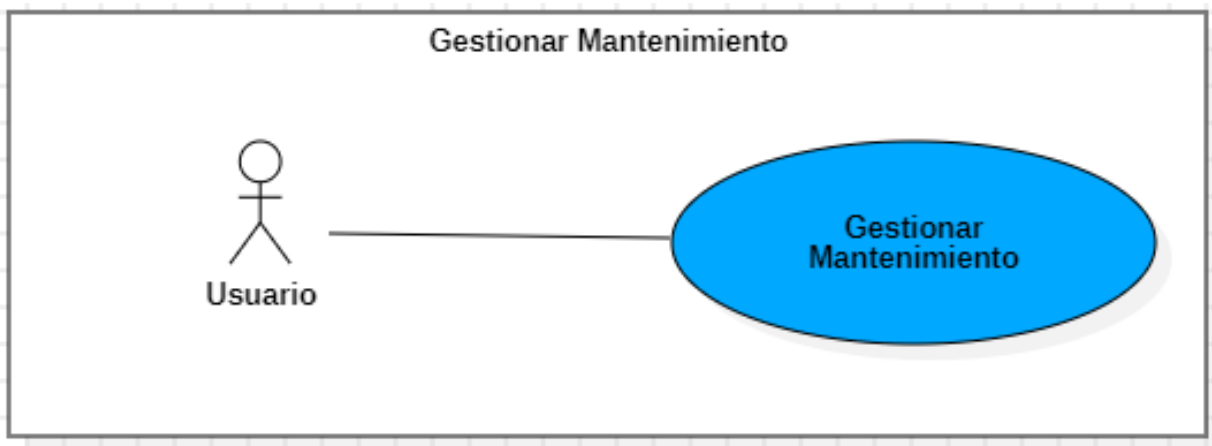
Caso de Uso	CU2. Gestionar Taller
Propósito	Gestionar Taller
Descripcion	Permitirá gestionar los talleres registrados en el sistema
Actores	Usuario
Actor Iniciador	Usuario
Precondición	Ninguna

Proceso	<ol style="list-style-type: none"> 1. Agregar Taller <ol style="list-style-type: none"> 1.1 Pulsar botón “Agregar” 1.2 Insertar datos 1.3 Guardar datos 2. Editar Taller <ol style="list-style-type: none"> 2.1 Pulsar botón “Editar” 2.2 Modificar datos 2.3 Actualizar datos 3. Eliminar Taller <ol style="list-style-type: none"> 3.1 Pulsar botón “Eliminar” 3.2 Eliminar vehículo
----------------	---



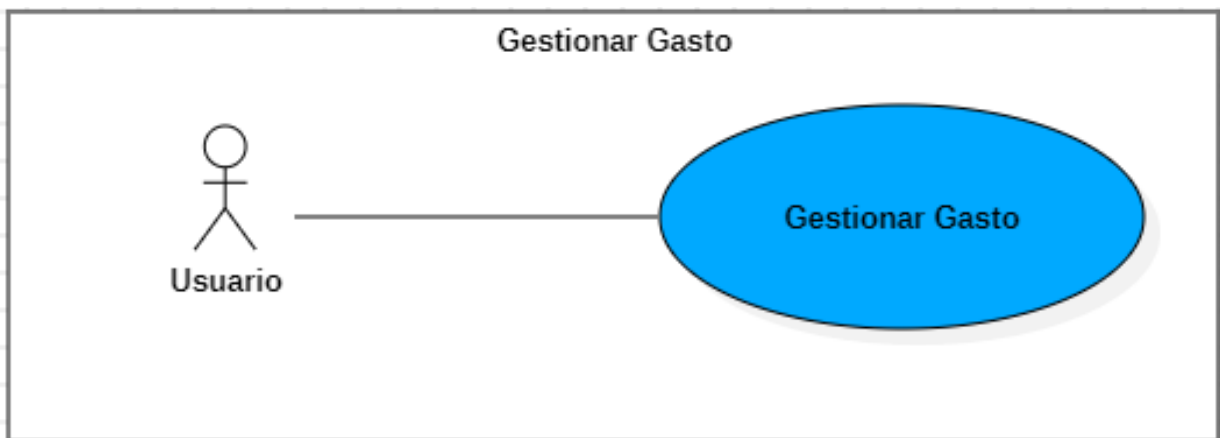
Caso de Uso	CU3. Gestionar Tipo Mantenimiento
Propósito	Gestionar Tipo Mantenimiento
Descripcion	Permitirá gestionar los tipos de mantenimiento disponibles
Actores	Usuario
Actor Iniciador	Usuario
Precondición	Ninguna

Proceso	<ol style="list-style-type: none"> 1. Agregar Tipo Mantenimiento <ol style="list-style-type: none"> 1.1 Pulsar botón “Agregar” 1.2 Insertar datos 1.3 Guardar datos 2. Editar Tipo Mantenimiento <ol style="list-style-type: none"> 2.1 Pulsar botón “Editar” 2.2 Modificar datos 2.3 Actualizar datos 3. Eliminar Tipo Mantenimiento <ol style="list-style-type: none"> 3.1 Pulsar botón “Eliminar” 3.2 Eliminar vehículo
----------------	---



Caso de Uso	CU4. Gestionar Mantenimiento
Propósito	Gestionar Mantenimiento
Descripcion	Permitirá registrar, modificar y eliminar mantenimientos asociados a vehículos
Actores	Usuario
Actor Iniciador	Usuario
Precondición	Ninguna

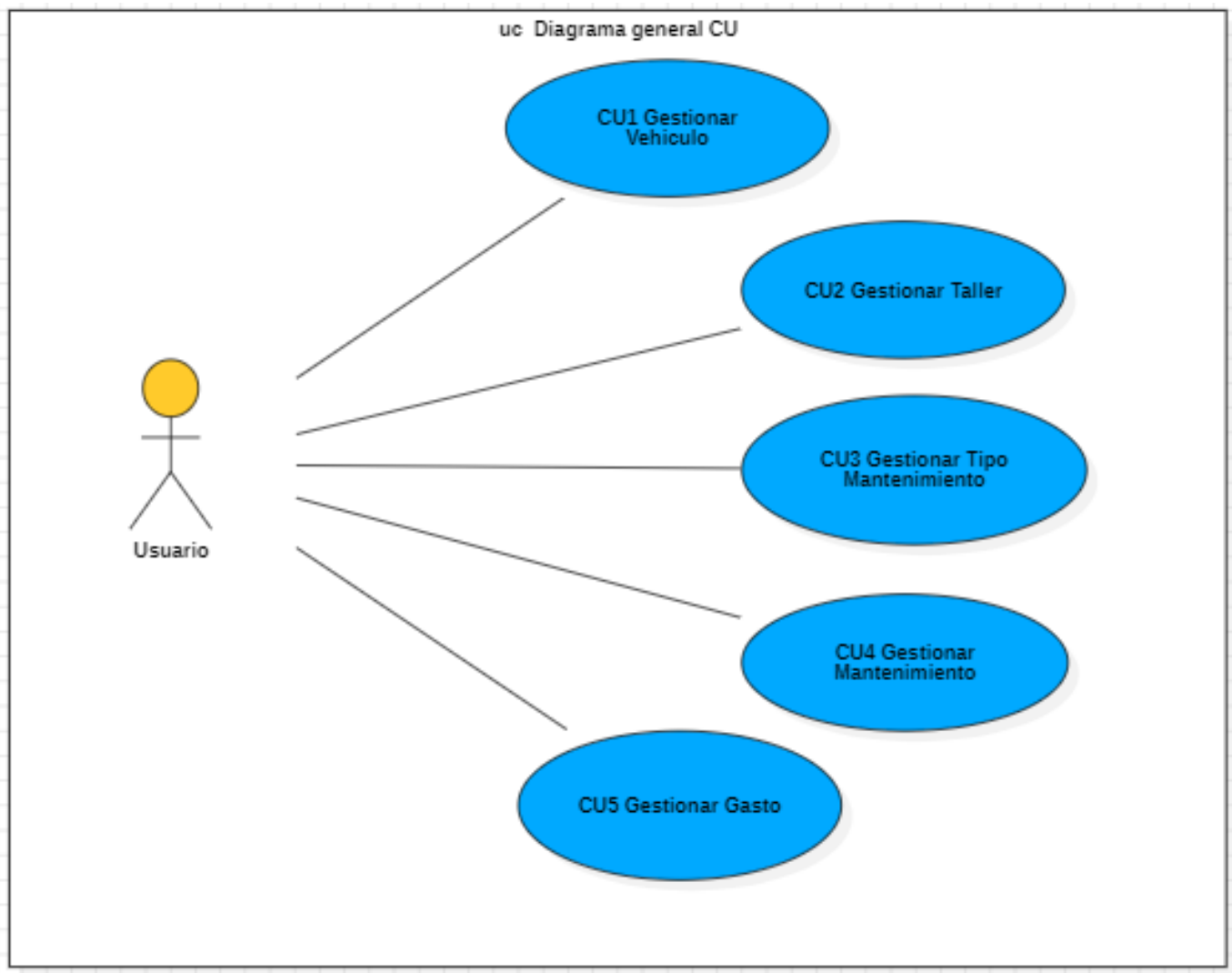
Proceso	1. Agregar Mantenimiento 1.1 Pulsar botón “Agregar” 1.2 Insertar datos 1.3 Guardar datos 2. Editar Mantenimiento 2.1 Pulsar botón “Editar” 2.2 Modificar datos 2.3 Actualizar datos 3. Eliminar Mantenimiento 3.1 Pulsar botón “Eliminar” 3.2 Eliminar mantenimiento 4. Ver Detalles 4.1 Pulsar botón 'Ver Detalles' 4.2 Visualizar detalles del mantenimiento y sus gastos
----------------	--



Caso de Uso CU5. Gestionar Gasto	
Propósito	Gestionar Gastos de un mantenimiento
Descripcion	Permitirá añadir, listar y eliminar gastos asociados a un mantenimiento
Actores	Usuario
Actor Iniciador	Usuario
Precondición	Debe existir al menos un mantenimiento
Proceso	1. Seleccionar Vehículo 1.1 Elegir un vehículo desde el desplegable 2. Seleccionar Mantenimiento 2.1 Elegir mantenimiento asociado al vehículo

	3. Agregar Gasto 3.1 Insertar nombre y monto 3.2 Guardar gasto 4. Eliminar Gasto 4.1 Pulsar ícono de eliminar 4.2 Eliminar gasto seleccionado
--	--

1.1.3 DIAGRAMA GENERAL DE CASOS DE USO

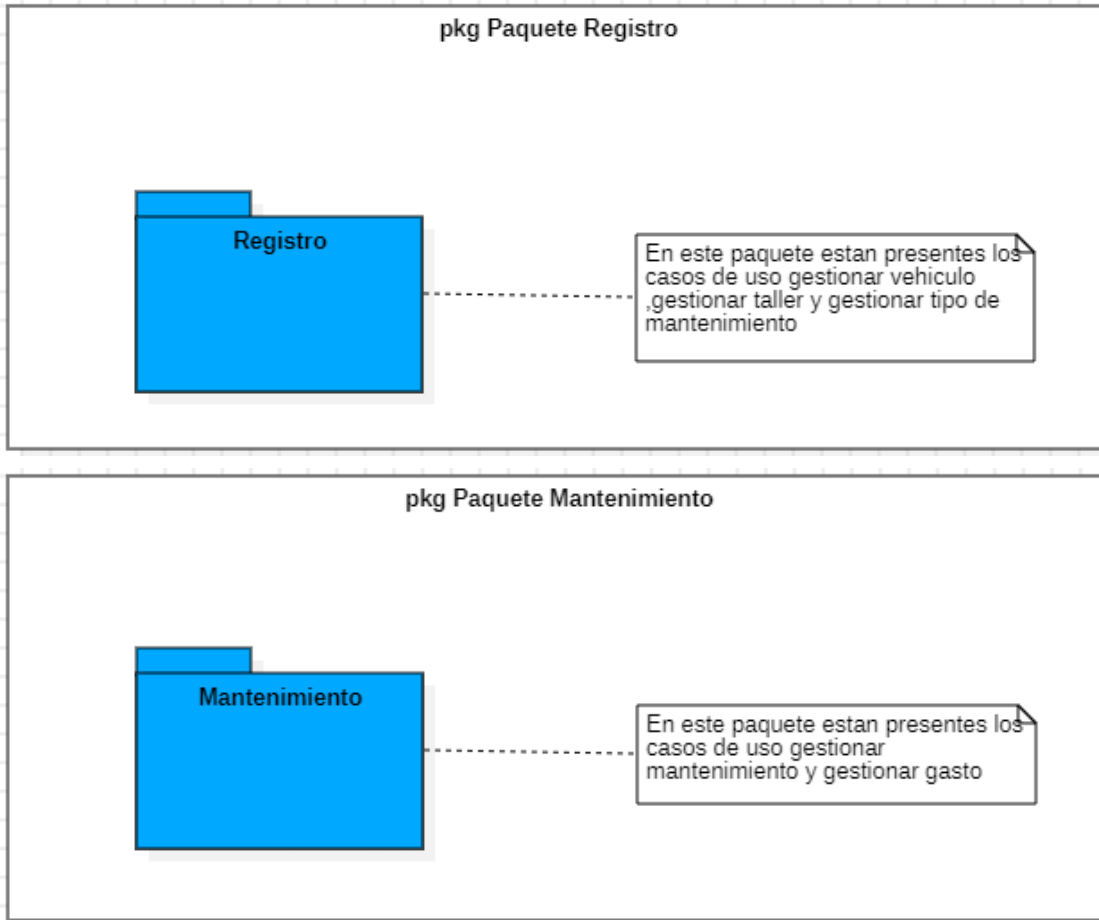


1.2. REQUISITOS NO FUNCIONALES

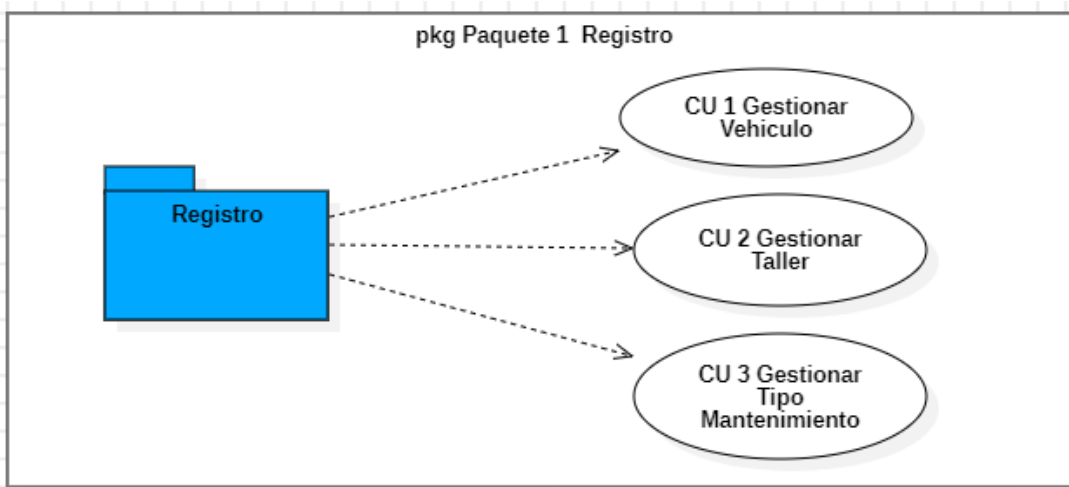
- **Usabilidad:** La app debe ser fácil de entender y utilizar.
- **Rendimiento:** Debe responder de forma rápida al realizar acciones como registrar mantenimientos, vehículos, talleres y tipos de mantenimiento.

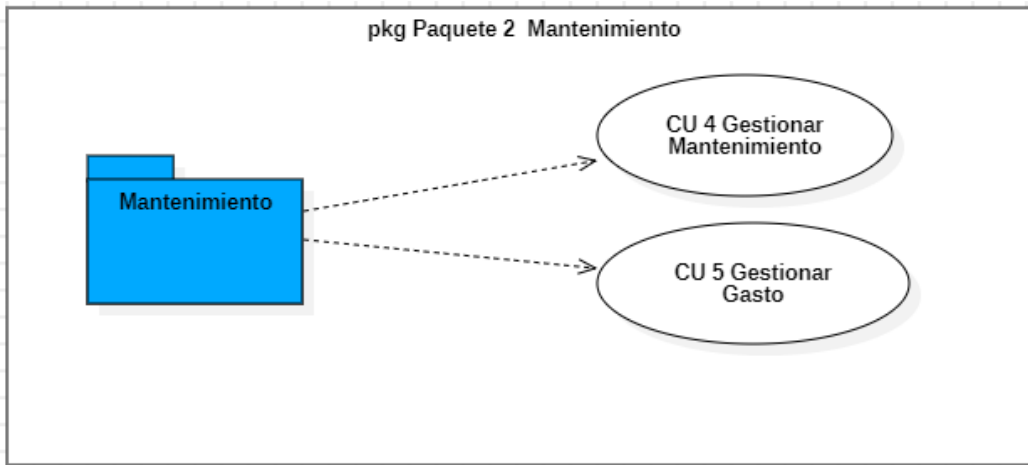
2. ANALISIS

2.1. IDENTIFICACION DE PAQUETES

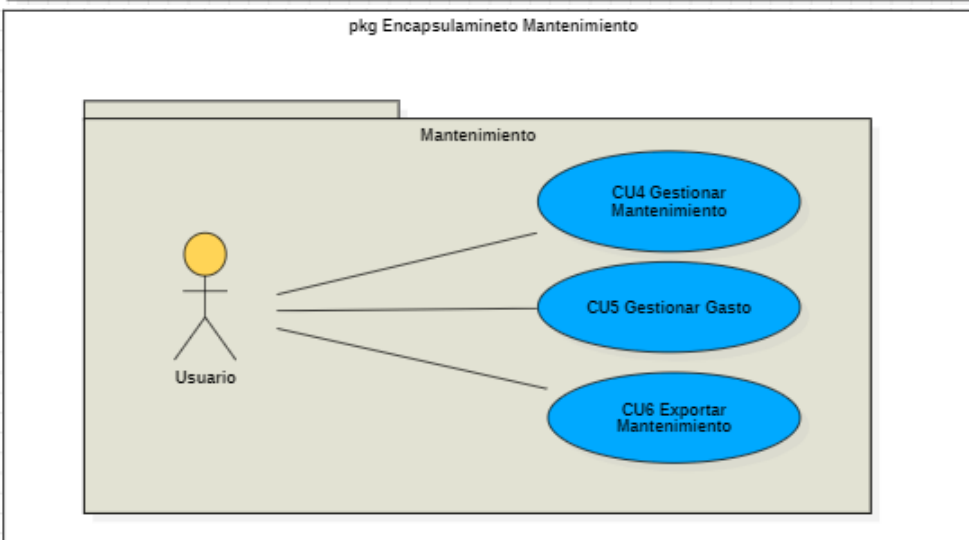
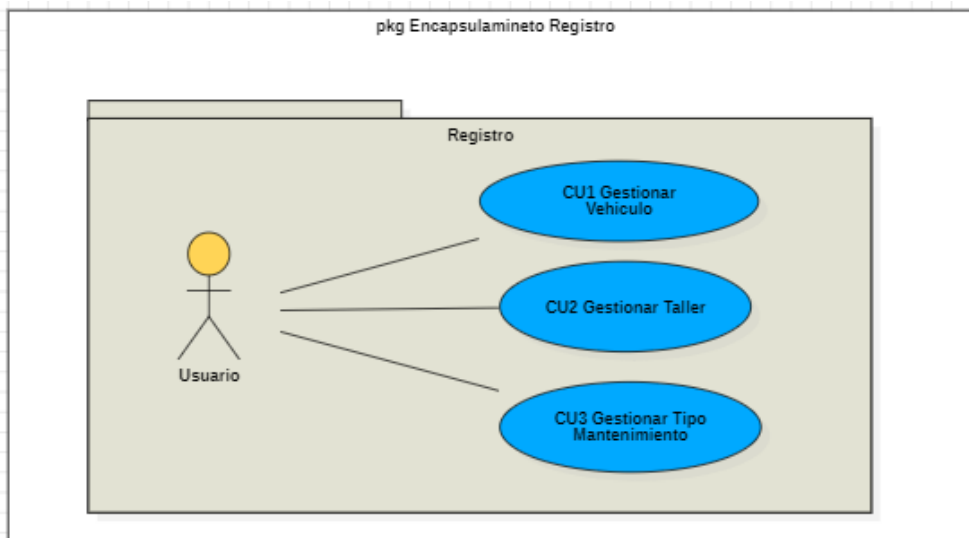


2.2. VISTA DE MODULOS





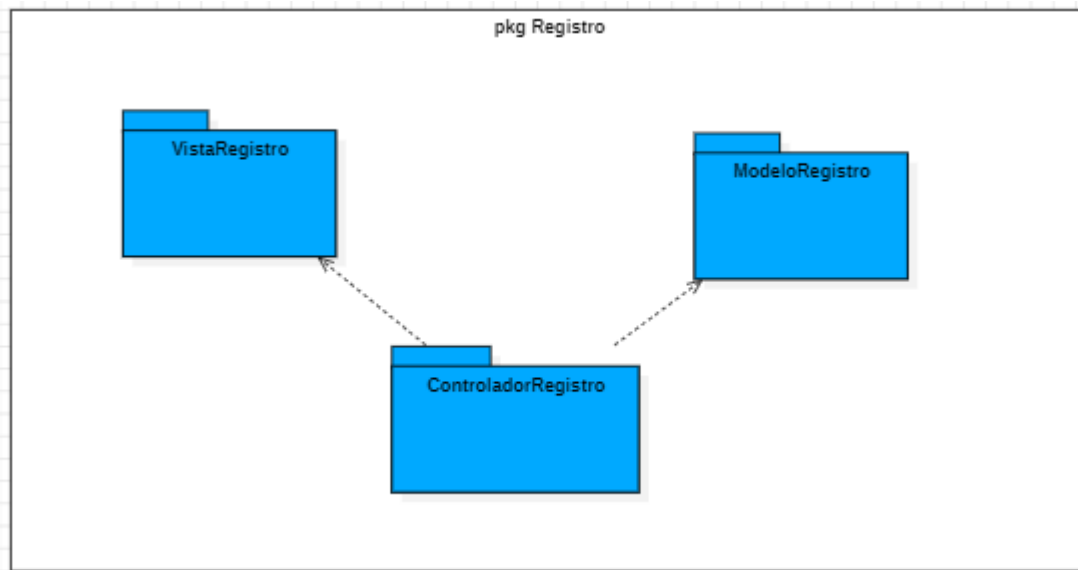
2.3. ENCAPSULAMIENTO DE CASOS DE USO



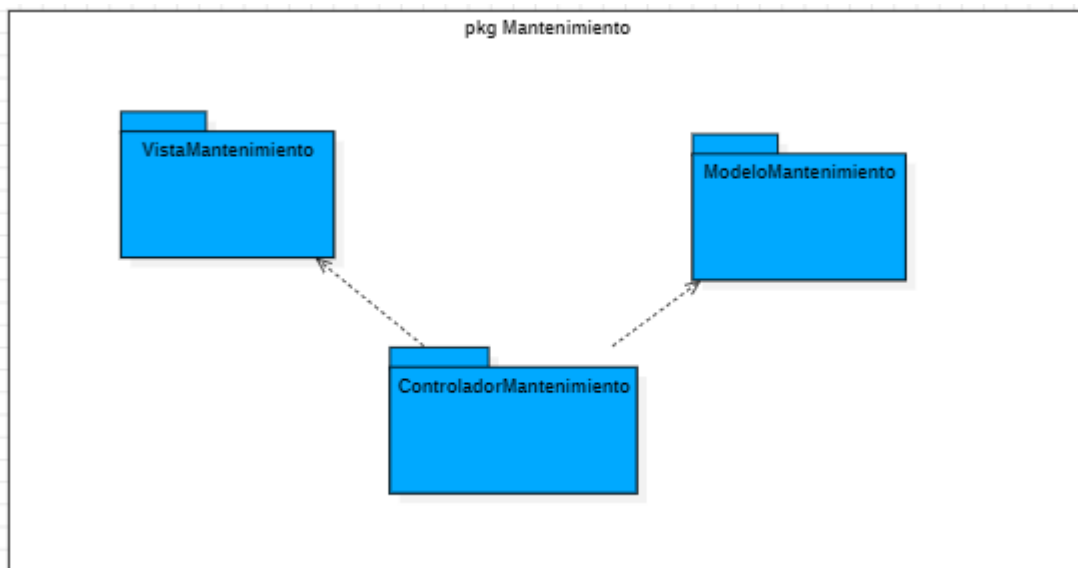
3. DISEÑO

3.1. DISEÑO DE LA ARQUITECTURA LOGICA

Paquete1: Registro:

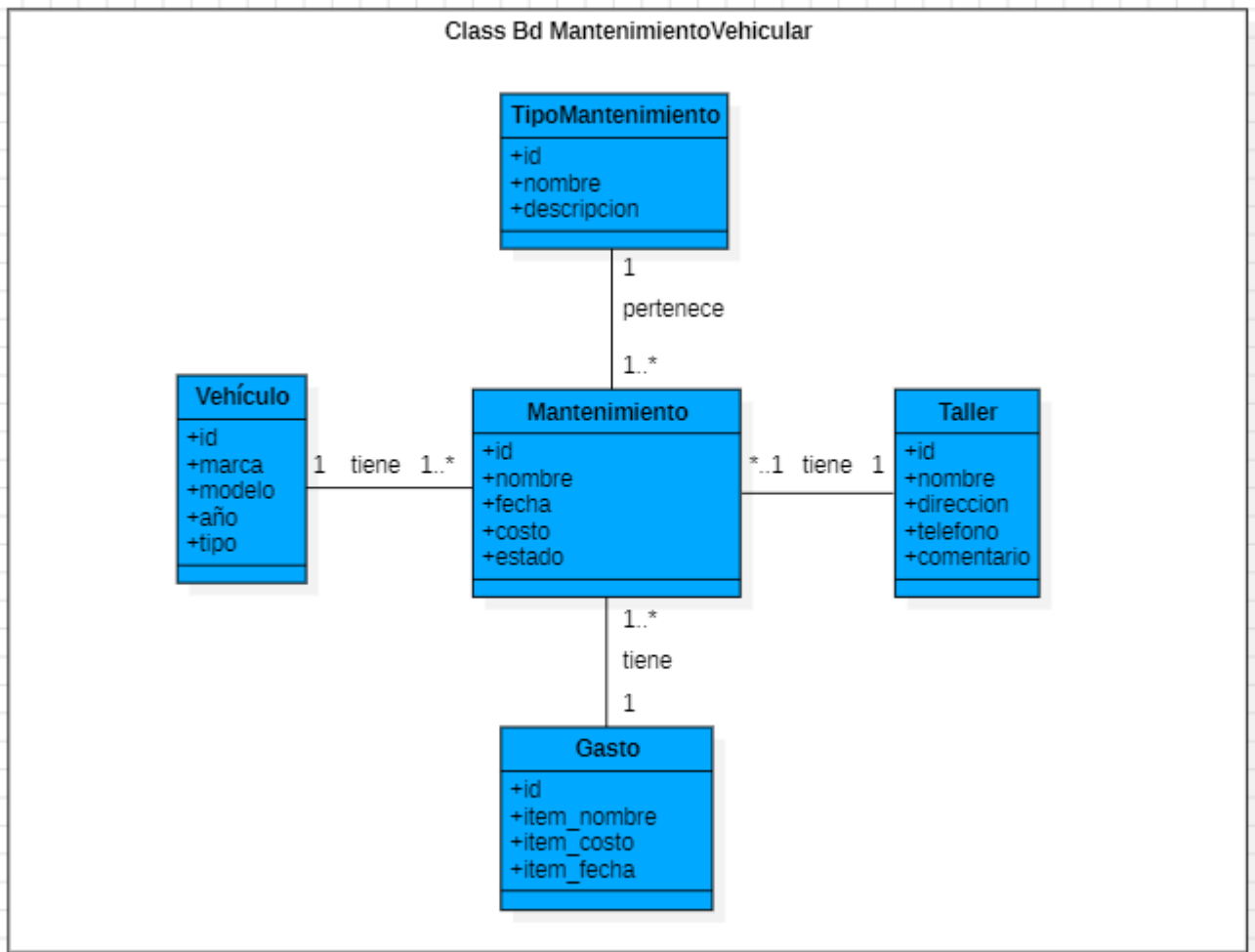


Paquete 2: Mantenimiento



3.2. DISEÑO DE LA BASE DE DATOS

3.2.1 DISEÑO CONCEPTUAL



3.2.2 DISEÑO LOGICO

TABLA_VEHICULO

Pk				
<u>Id</u>	marca	modelo	año	tipo

TABLA_TALLER

Pk				
<u>Id</u>	nombre	direccion	telefono	comentario

TABLA_TIPOMANTENIMIENTO

Pk		
<u>Id</u>	nombre	descripcion

TABLA_MANTENIMIENTO

Pk					Fk	Fk	Fk
<u>Id</u>	nombre	fecha	costo	estado	id_vehiculo	id_taller	id_tipomantenimiento

TABLA_GASTO

Pk				Fk
<u>Id</u>	item_nombre	item_costo	item_costo	id_mantenimiento

3.2.3 DISEÑO FISICO

```
CREATE TABLE
VEHICULO (
id INTEGER PRIMARY
KEY,          marca
TEXT,
modelo TEXT,
año INTEGER,
tipo TEXT
);
```

```
CREATE TABLE TALLER (
id INTEGER PRIMARY KEY
AUTOINCREMENT,      nombre
TEXT,               direccion TEXT,
telefono TEXT,      comentario
TEXT
);
```

```
CREATE TABLE TIPOMANTENIMIENTO (
id INTEGER PRIMARY KEY
AUTOINCREMENT,      nombre
TEXT,               descripcion TEXT
```

);

```
CREATE TABLE MANTENIMIENTO (  
  id INTEGER PRIMARY KEY  
  AUTOINCREMENT,  
  id_vehiculo INTEGER,  
  id_taller INTEGER,  
  id_tipomantenimiento INTEGER,  
  nombre TEXT,          fecha TEXT,  
  costo REAL,           estado TEXT,  
          FOREIGN KEY(id_vehiculo) REFERENCES VEHICULO (id),  
          FOREIGN KEY(id_taller) REFERENCES TALLER (id)          FOREIGN  
  KEY(id_tipomantenimiento) REFERENCES TIPOMANTENIMIENTO (id) ON DELETE  
  CASCADE  
  );
```

```
CREATE TABLE GASTO (          id  
  INTEGER PRIMARY KEY  
  AUTOINCREMENT,  
  id_mantenimiento INTEGER,  
  item_nombre TEXT,  
  item_costo REAL,  
  item_fecha TEXT,  
          FOREIGN KEY(id_mantenimiento) REFERENCES MANTENIMIENTO (id)  
  );
```

3.3. DISEÑO DE LA INTERFAZ

Gestionar vehiculo

Lista de Vehículos

[Agregar Vehículo](#)

Marca: Toyota
Modelo: Corolla
Año: 2000
Tipo: Auto

[Editar](#) [Eliminar](#)

Marca: ZUsuki
Modelo: Vitara
Año: 2010
Tipo: Auto

[Editar](#) [Eliminar](#)

Registrar Vehículo

Gestionar Taller

Lista de Talleres

[Agregar Taller](#)

Nombre: El cumpa
Dirección: 22
Teléfono: 22
Comentario: 22

[Editar](#) [Eliminar](#)

Nombre: Bolivar
Dirección: 4to Anillo
Teléfono: 12345
Comentario: Buen servicio

[Editar](#) [Eliminar](#)

Registrar Taller

Gestionar Tipo Mantenimiento

Lista de Tipos de Mantenimiento

Agregar Tipo de Mantenimiento

Nombre: A/C
Descripción: D

EditarEliminar

Nombre: Cambio de Aceite
Descripción: General

EditarEliminar

Nombre: Preventivo
Descripción: En general

EditarEliminar

Registrar Tipo de Mantenimiento

Nombre

Descripción

GuardarCancelar

Gestionar Mantenimiento




Lista de Mantenimientos

Filtrar por vehículo




Todos

Registrar Mantenimiento

Tipo: A/C
Vehículo: Toyota Corolla
Estado: programado



Tipo: Cambio de Aceite
Vehículo: Toyota Corolla
Estado: terminado



Editar Mantenimiento

Nombre

Alerta aceite en pantalla

Fecha

2025-04-22

Costo (Bs)

300.0

Vehículo

Toyota Corolla

Taller


Bolívar


Tipo de Mantenimiento


Cambio de Aceite


GuardarCancelar


Detalles del Mantenimiento


 Nombre: Alerta aceite en pantalla


 Fecha: 2025-04-22


 Costo base: 300.00 Bs

 Vehículo: Toyota Corolla

 Taller: Bolívar


 Tipo de Mantenimiento: Cambio de Aceite


 Estado: terminado

 Detalle de Gastos:

• Mano de obra: 150.0 Bs

• Aceite marca CleanMotor: 120.0 Bs

 Total gastos: 270.00 Bs

 Total final: 570.00 Bs

Volver

Gestionar Gasto

Gastos por Mantenimiento

Filtrar por Vehículo

Todos

Mantenimiento: XD

Mantenimiento: Alerta aceite en pantalla

Formulario de Gastos

Nombre del gasto

Costo (Bs)

Añadir Gasto

Item: Mano de obra
Costo: 150.0 Bs
Fecha: 2025-04-23

Item: Aceite marca CleanMotor
Costo: 120.0 Bs
Fecha: 2025-04-23

Volver

3.4. DISEÑO DE DETALLE PROCEDIMENTAL

CU1: Gestionar vehículo

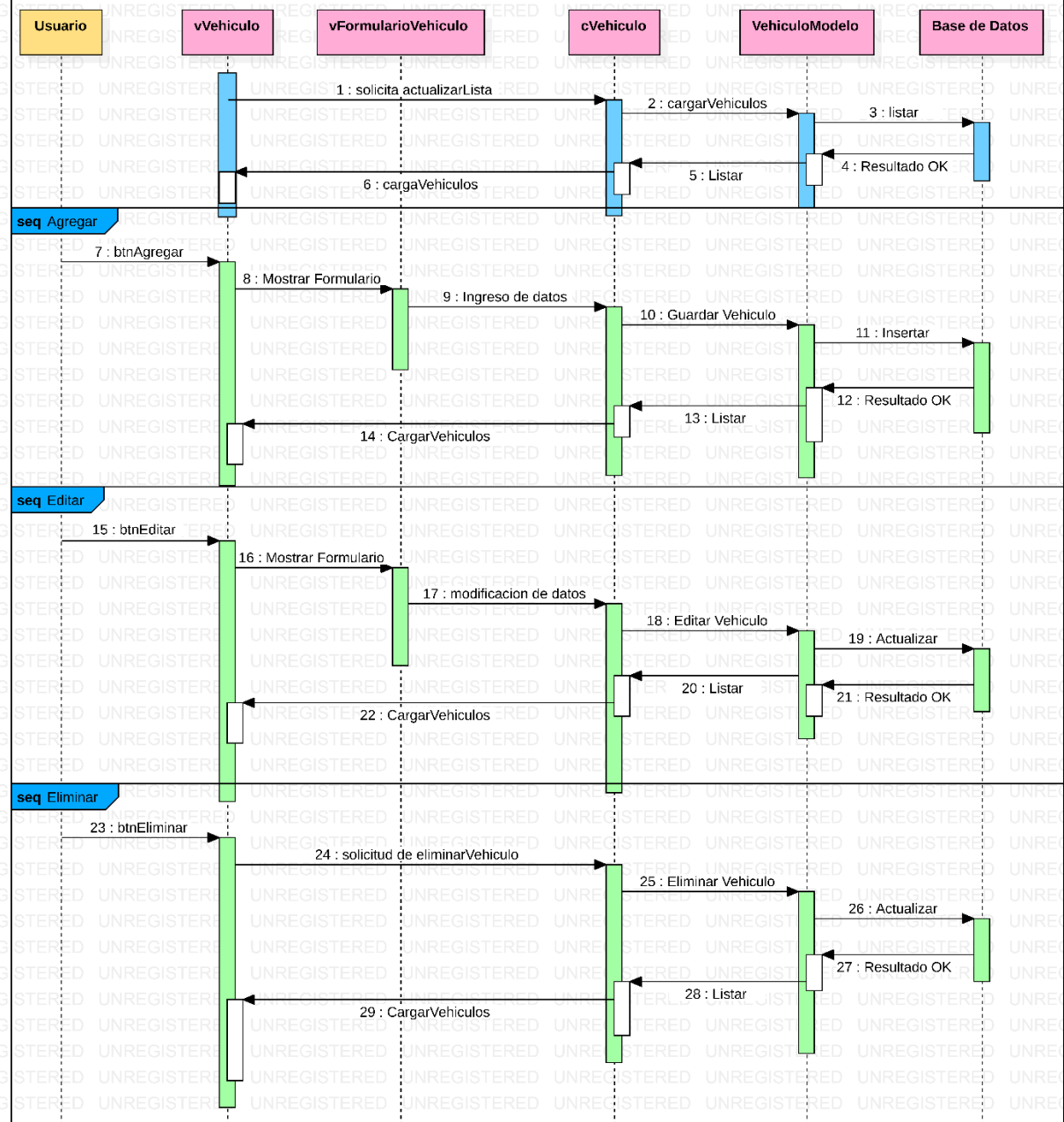
Class CU 1 Gestionar Vehiculo

```
classDiagram
    class VistaVehiculo {
        +btnAgregar: Button
        +btnEditar: Button
        +btnEliminar: Button
        +actualizarLista(): fun
        +mostrarMensaje(): fun
    }
    class VistaFormularioVehiculo {
        +btnAgregar: Button
        +btnCancelar: Button
        +actualizarLista(): fun
        +mostrarMensaje(): fun
    }
    class VehiculoControlador {
        +cargarVehiculo(): fun
        +guardarVehiculo(): fun
        +editarVehiculo(): fun
        +eliminarVehiculo(): fun
    }
    class VehiculoModelo {
        -id: Int
        -marca: String
        -modelo: String
        -año: Int
        -tipo: String
        +insertar(): fun
        +actualizar(): fun
        +eliminar(): fun
        +listar(): fun
        +buscarPorId(): fun
        +tieneMantenimientos(): fun
    }
    class MantenimientoVehicular {
        +MantenimientoVehicular: SQLiteOpenHelper
        +onCreate(): SQLiteDatabase
        +onUpgrade(): SQLiteDatabase
    }
    VistaVehiculo --> VehiculoControlador
    VistaFormularioVehiculo --> VehiculoControlador
    VehiculoControlador --> VehiculoModelo
    VehiculoModelo --> MantenimientoVehicular
```

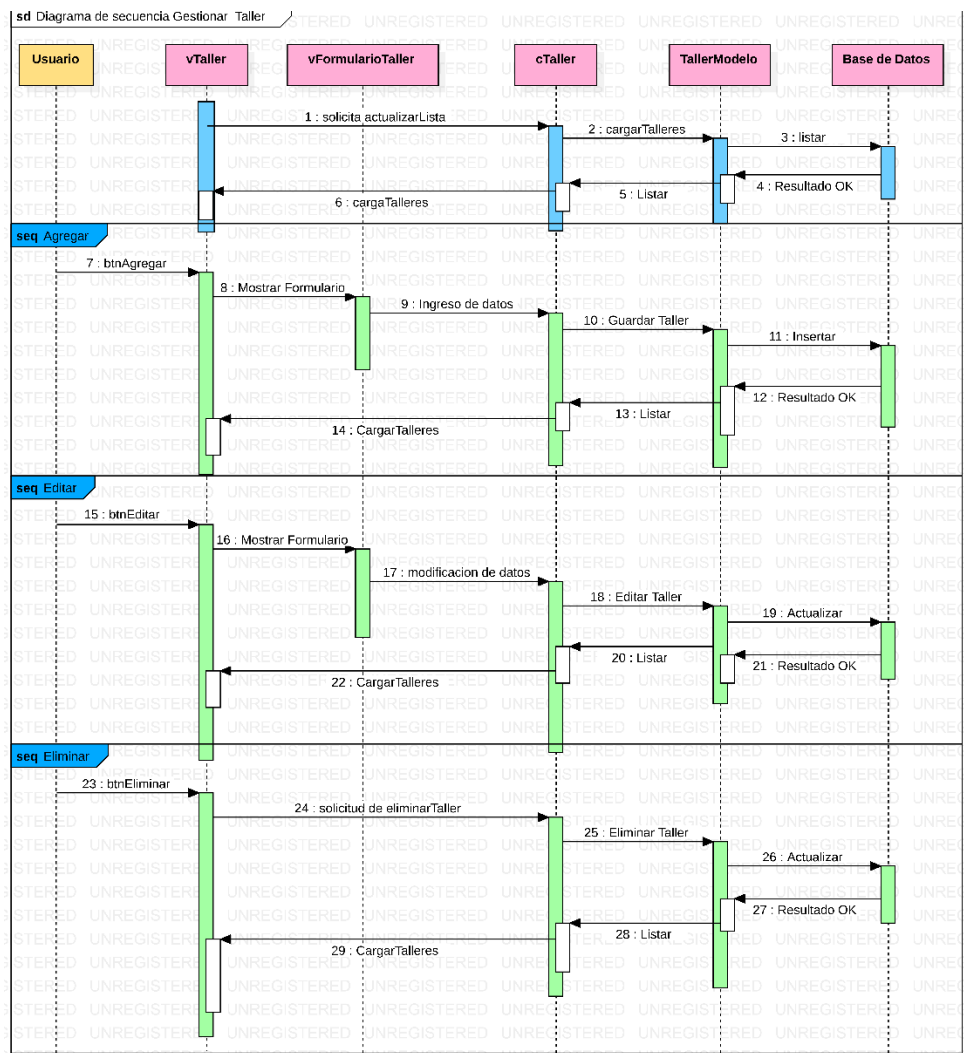
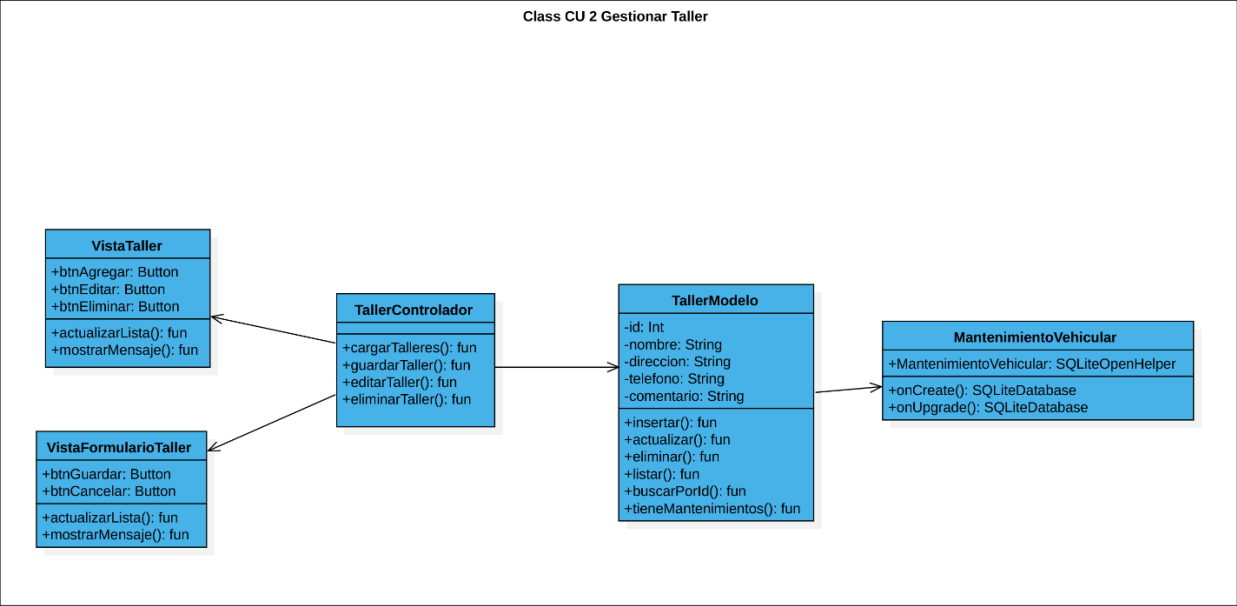
The diagram illustrates the procedural design for managing vehicles (CU1). It features five main components: **VistaVehiculo**, **VistaFormularioVehiculo**, **VehiculoControlador**, **VehiculoModelo**, and **MantenimientoVehicular**. **VistaVehiculo** and **VistaFormularioVehiculo** are the user interfaces, each containing buttons for adding, editing, and deleting vehicles, as well as methods for updating the list and showing messages. **VehiculoControlador** acts as the central logic layer, receiving input from the views and interacting with the **VehiculoModelo**. **VehiculoModelo** is the data model, containing attributes like ID, brand, model, year, and type, and methods for inserting, updating, deleting, listing, and searching for vehicles, as well as checking for maintenance. **MantenimientoVehicular** is a helper class for database operations, including SQLiteOpenHelper, onCreate, and onUpgrade methods.

17

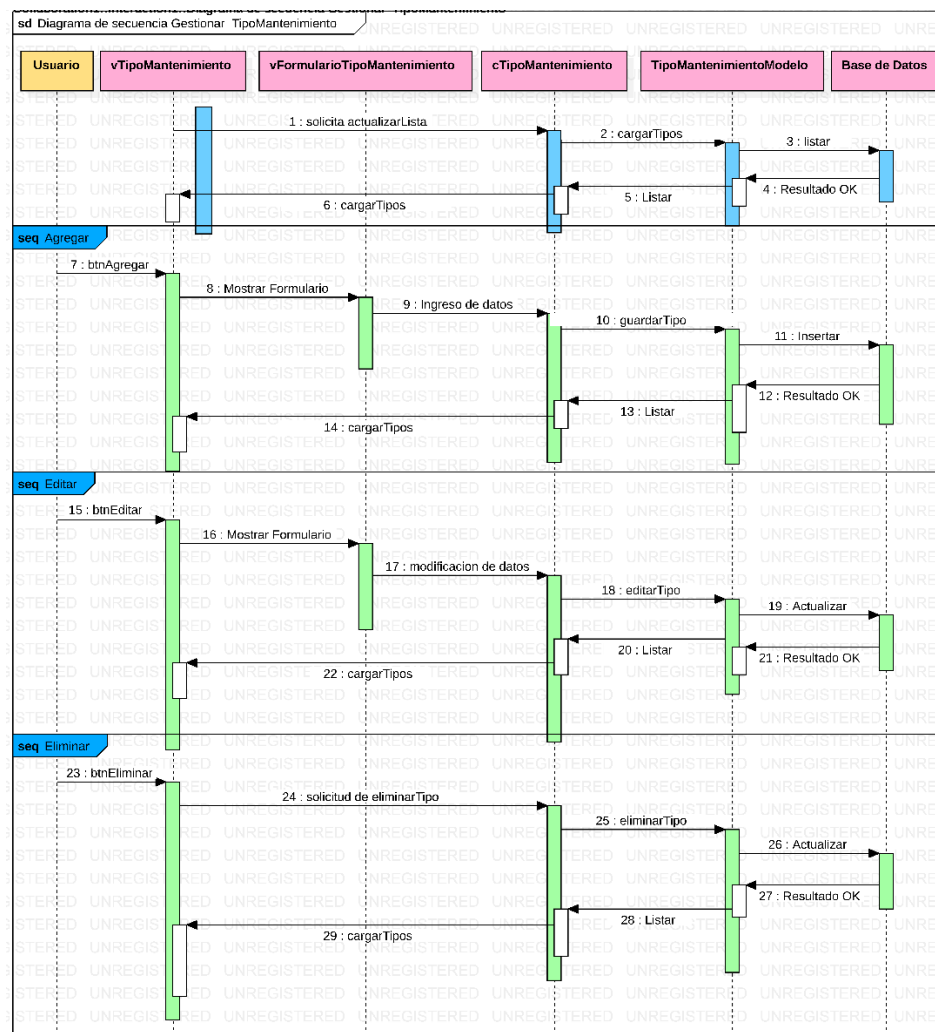
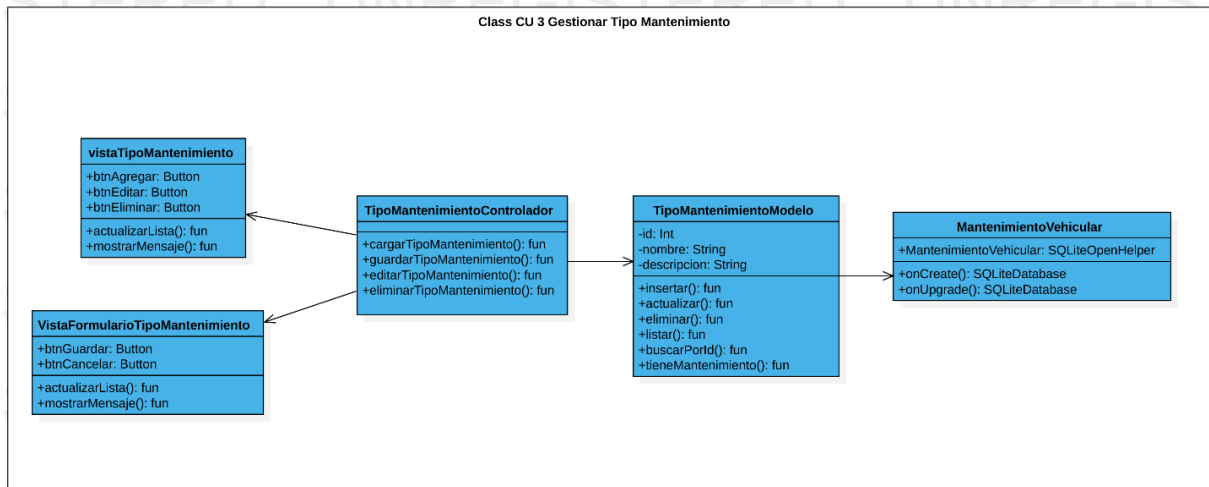
sd Diagrama de secuencia Gestionar Vehiculo



CU2: Gestionar taller

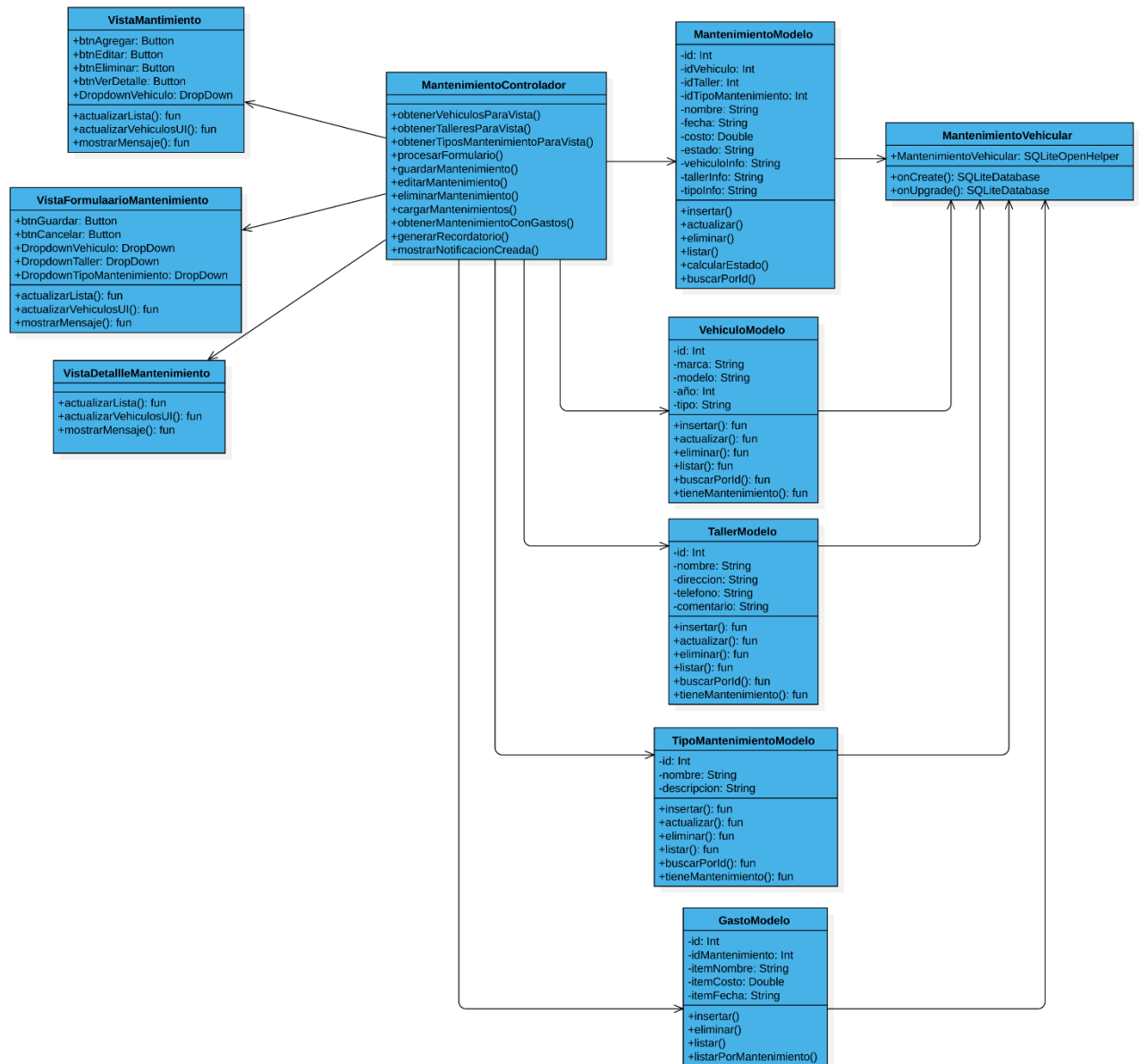


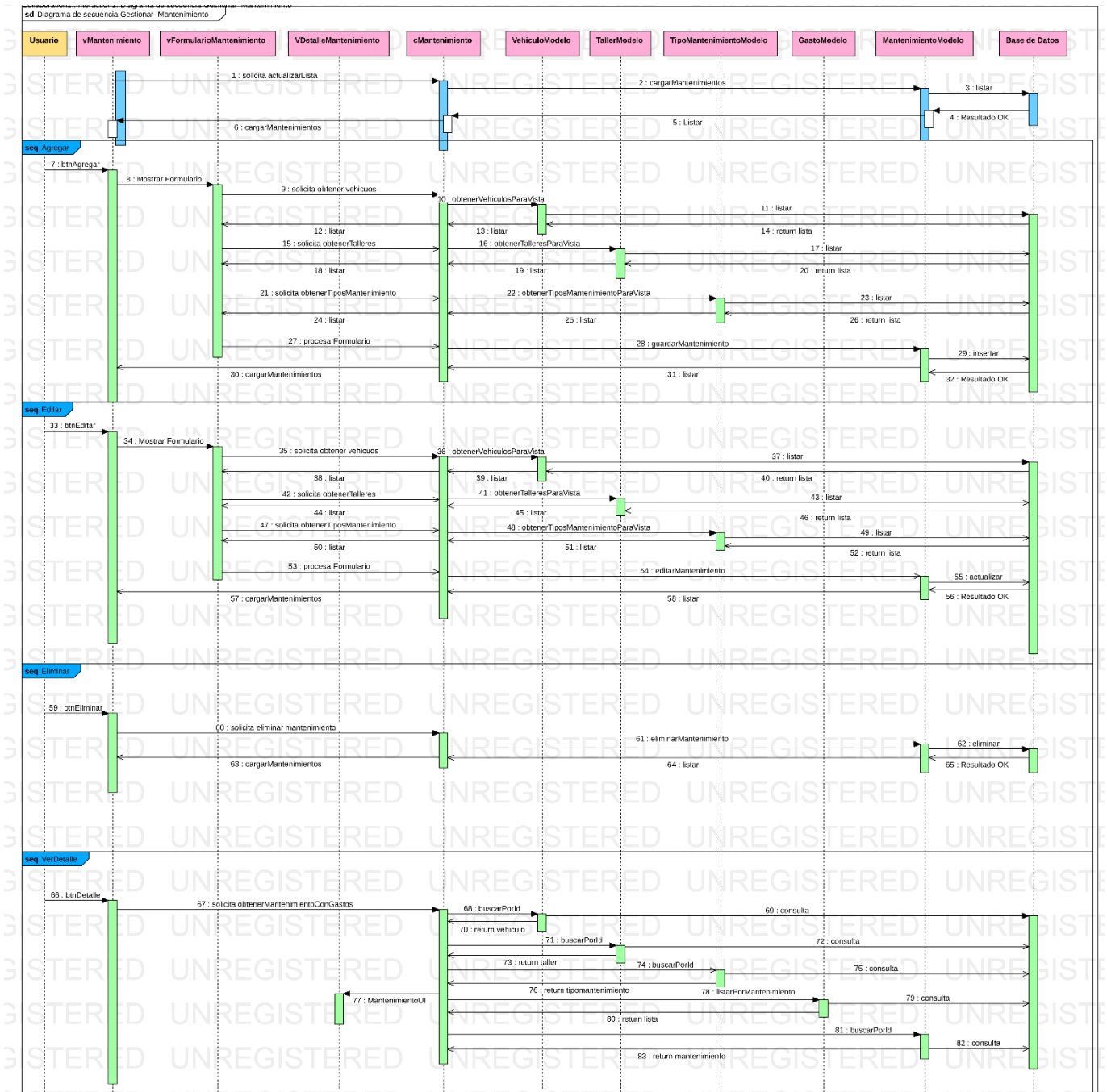
CU3: Gestionar tipo mantenimiento



CU4: Gestionar mantenimiento

Class CU 4 Gestionar Mantenimiento





CU5: Gestionar gasto

