



SISTEMA DE GESTIÓN DE TURNOS
PARA EMPRESA
SUPERCHARGER S.R.L.
Trabajo Práctico 2

DISEÑO ORIENTADO A OBJETOS

BOZZANO, Felipe – Legajo: 102050

OLIVA, Benjamín – Legajo: 100508

OLMOS, Francisco – Legajo: 102369

Profesor: Prof. Esp. Ing. Agustín Fernandez

INDICE

Introducción	3
Objetivos	3
Requerimientos Funcionales	4
Alcance.....	4
Diagrama de Clases	5
Diagrama de Casos de Uso	6
Diagrama de Colaboración	7
Diagrama de Secuencia	8
Diagrama de Estados.....	9
Clase Turno.....	9
Clase Ficha Mecánica	10
Clase Encargado Recepción.....	11
Interfaces del Sistema	12
Función: Turnos.....	12
Función: Registrar Turno.....	13
Función: Ficha Mecánica.....	14
Función: Nuevo Empleado	15
Función: Nuevo Cliente	16
Interfaces del Tests	17
Tests para Turnos	17
Tests para Vehículos	18
Resultado Turnos	18
Resultados Vehículos	19
Patrones de Diseño Localizados / Implementados.....	20

Introducción

El sistema de gestión de turnos para la empresa Supercharger S.R.L. es un programa diseñado para computarizar el flujo de los turnos, con el fin de monitorear y registrar la línea de vida de los mismos para mejorar y evaluar la eficiencia en cada proceso, facilitando a los usuarios un sistema amigable y seguro que les sirva en la realización de sus tareas.

Objetivo

Sistematizar los procesos que forman parte de la línea de vida de los turnos, y brindar reportes relevantes respecto al trabajo y los procesos realizados sobre los mismos.

Requerimientos funcionales

- Registro y cancelación de turnos.
- Generación de comprobante de turno.
- Generación de informe de servicios por compañía de seguro.
- Registro de cobranza.
- Registro de asistencia y nuevo cliente.
- Generación y registro de ficha mecánica.
- Generación de informe por actividad diaria de mecánico.
- Generación y registro de constancia de conformidad.
- Registro de agenda de turnos.

Alcance

El alcance del sistema comienza en la generación del turno y termina en el cobro del servicio a la compañía de seguro del titular del vehículo, incluyendo de por medio la realización de todos los requerimientos funcionales.

Diagrama Casos de Uso

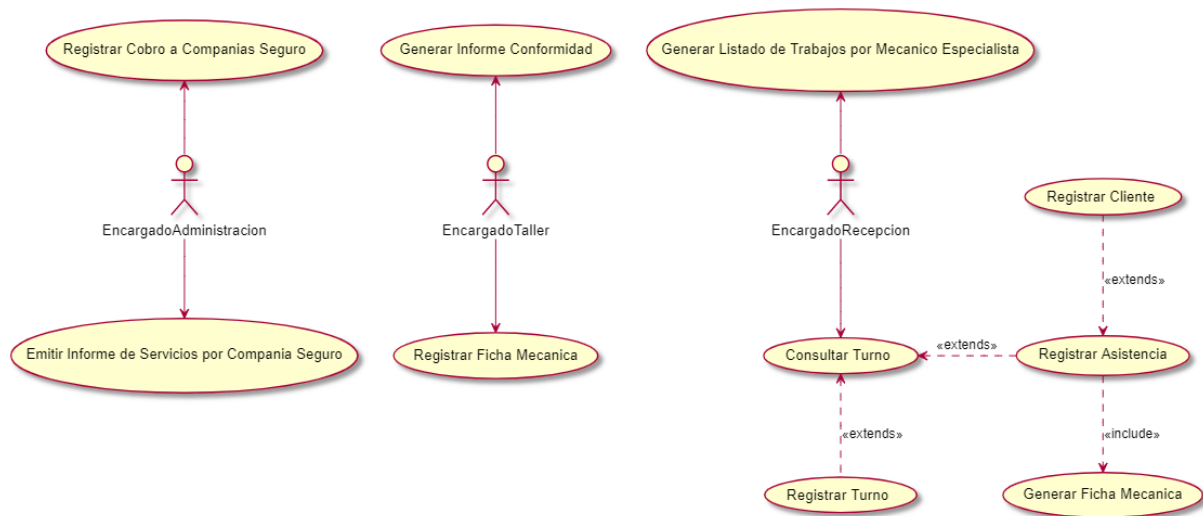


Diagrama de Colaboración

CU: Consultar Turno

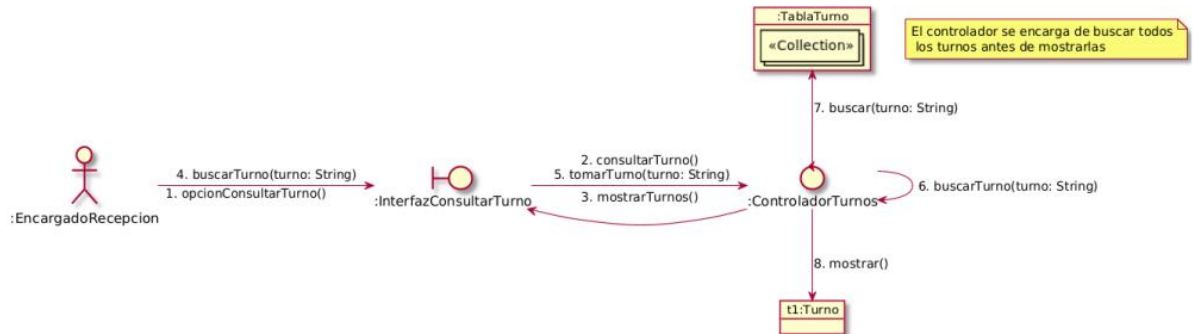


Diagrama de Secuencia

CU: Registrar Turno

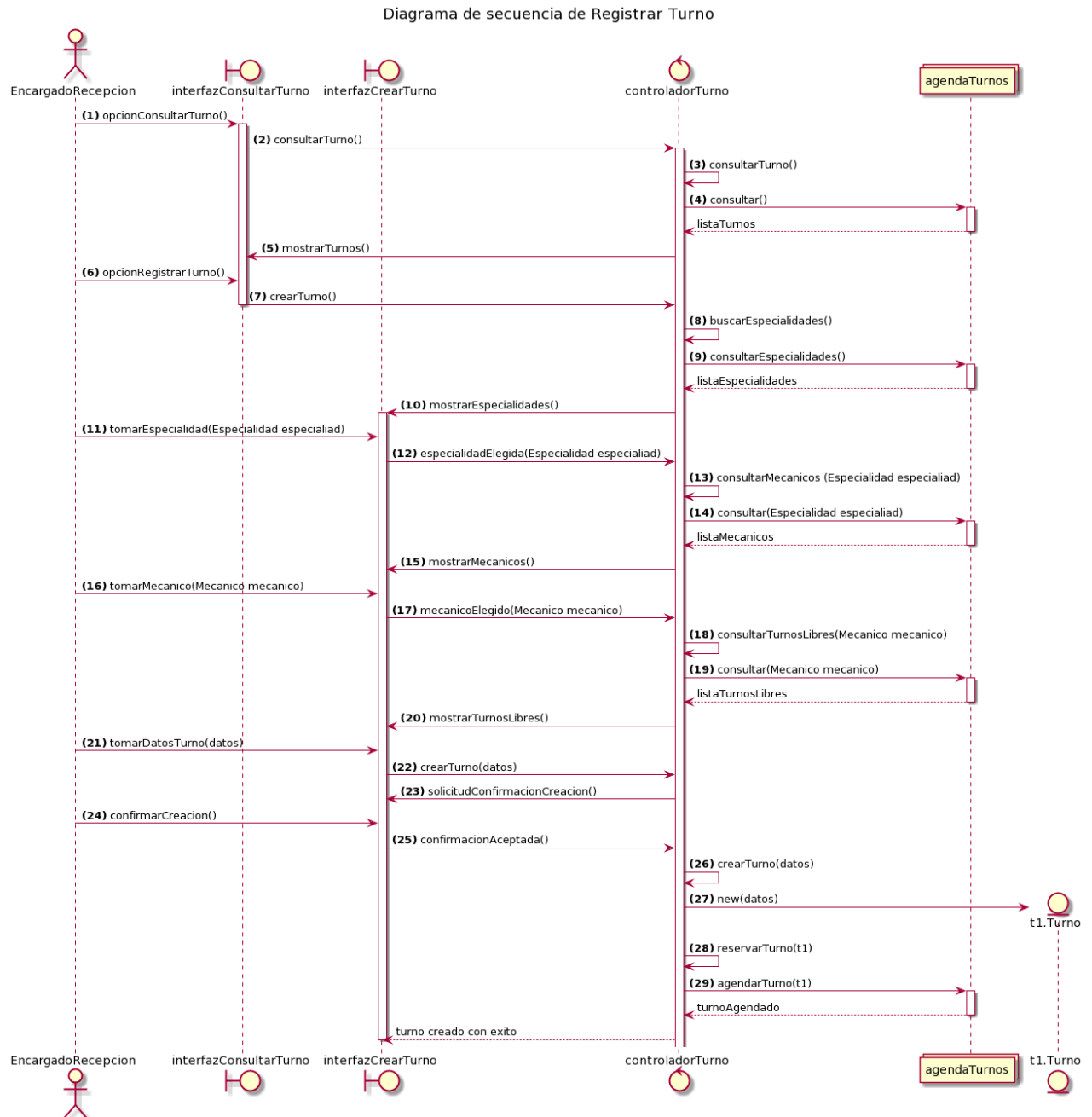
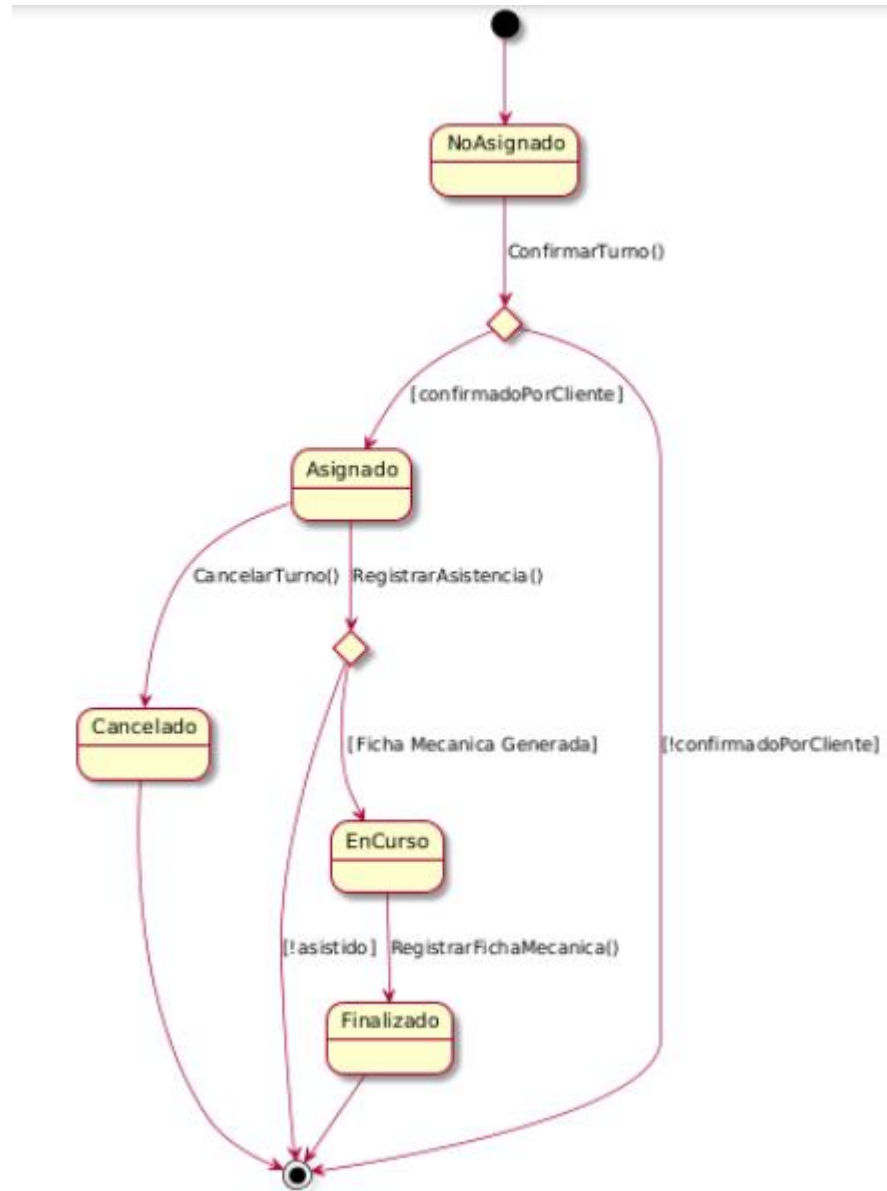
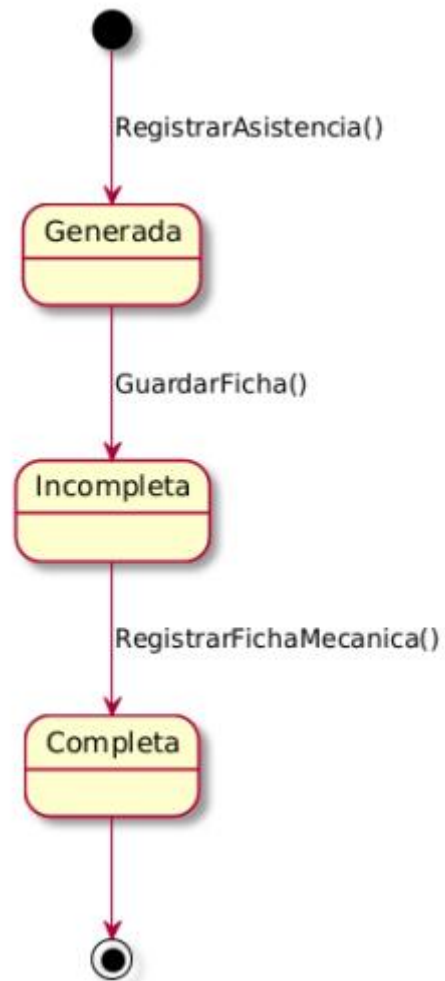


Diagrama de Estados

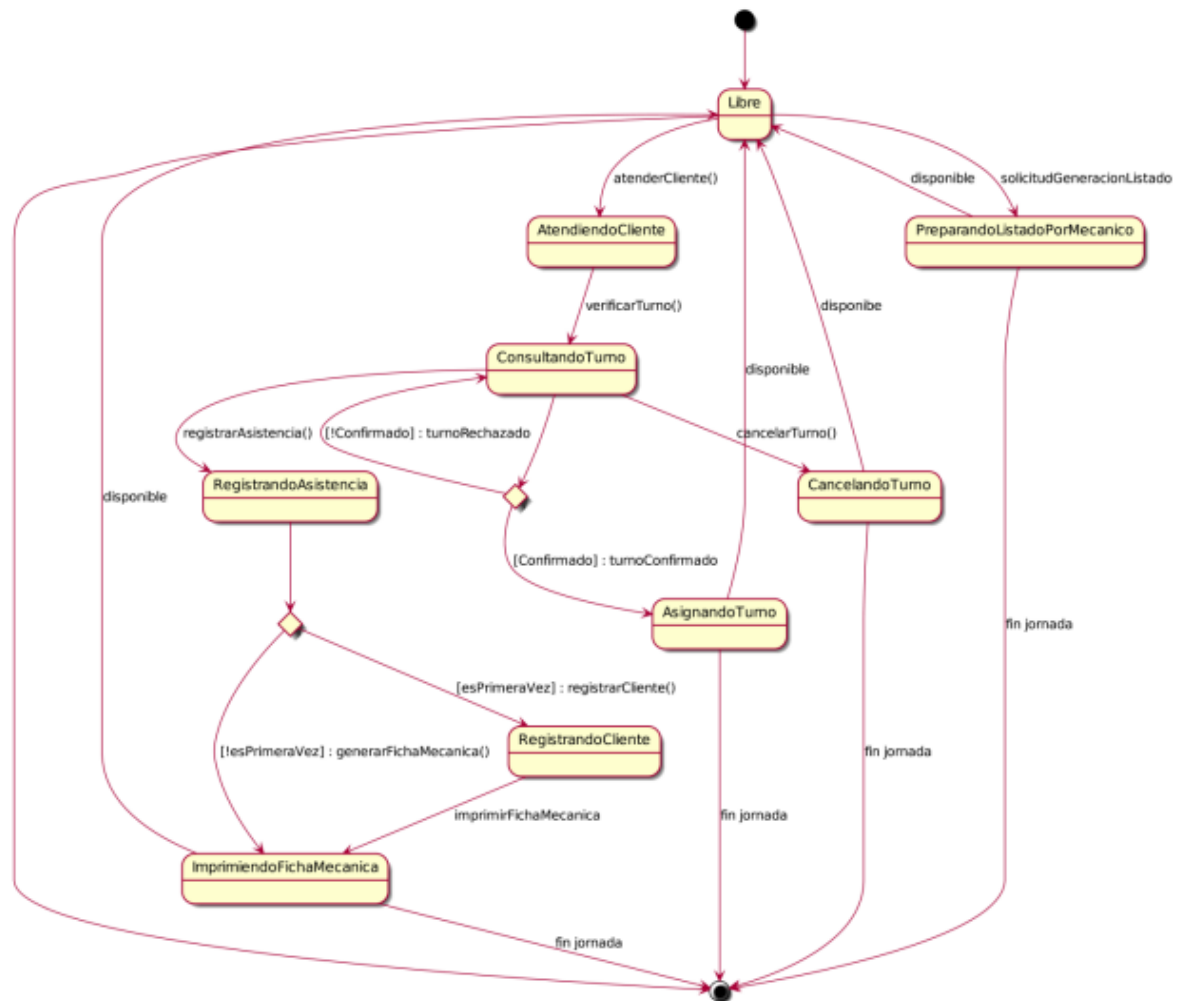
Clase: Turno



Clase: Ficha Mecánica




Clase: Encargado Recepción



Interfaces del Sistema

Función: Turnos – vistaHome - Implementada


— □ ×

SuperCharger S.R.L.

Nuevo Turno

Consultar Turno

Nro	Dia	Hora	Mecanico	Vehiculo	Titular	Compania	Estado	FichaMeca...
1	2021-06-20	23:25:00	Pepillo	Tiggo	Francisco	La Segunda	Asignado	No creada
2	2012-01-28	12:00:00	Fernando	Amarok	Benja	NN	Asignado	No creada
3	2015-12-01	15:00:05	Patricio	Vento	Matias	NN	Asignado	No creada
4	2020-07-16	12:00:00	Panchito	Camaro	Felipe	Zurich	Asignado	No creada
5	2010-07-16	00:00:00	Carlos	Camaro	Felipe	Zurich	Asignado	No creada
6	2002-02-20	00:00:00	Alberto	Camaro	Felipe	Zurich	Asignado	No creada
8	1990-10-04	06:25:32	Mecanico1	Auto1	Titular1	Compania1	Asignado	No creada
9	1995-07-05	23:50:50	Mecanico	Vehiculo	Titular	Compania	Asignado	No creada

Función: Registrar Turno – FrmNuevoTurno - Implementada

Formulario de Turno

Especialidad: Frenos

Mecanico: Hector

Datos del Vehiculo

nro Poliza: 7 Marca: Marca7 Modelo: Modelo7

Titular

Nombre: Titular7 Apellido: Apellido7

Tipo: DNI Número: 77777777

Telefono: Cod. 7 Caract. 77 Numero: 777

Compania de Seguro: La Segunda

Fecha del Turno: 01

Hora del Turno: 09:00

Cancelar Guardar

Función: Ficha Mecanica – vistaFichaMecanica – Sin Implementar

Ficha Mecanica

Numero de Ficha:

NRO

Titular

Nombre:

NOMBRE

Apellido:

APELLIDO

D.N.I.:

NUMERO

Telefono:

NUMERO

Mecanico

Nombre:

NOMBRE

Legajo:

NUMERO

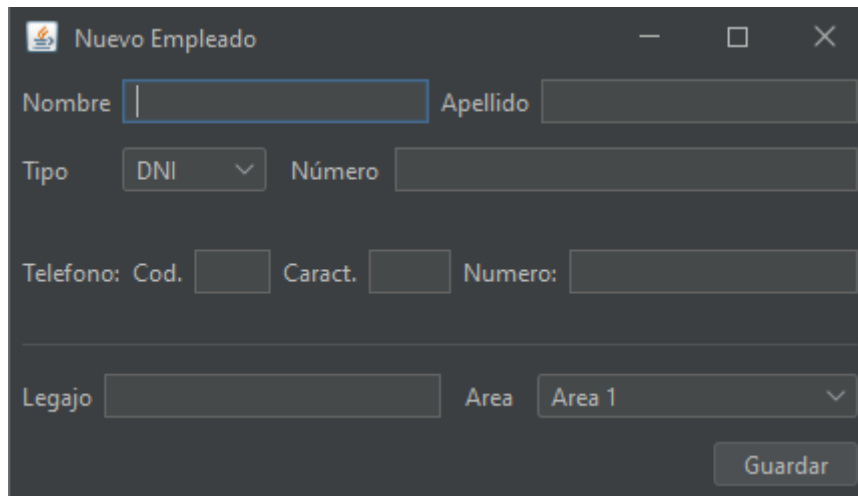
Detalle

nroDetalle 1

Ver

TOTAL:

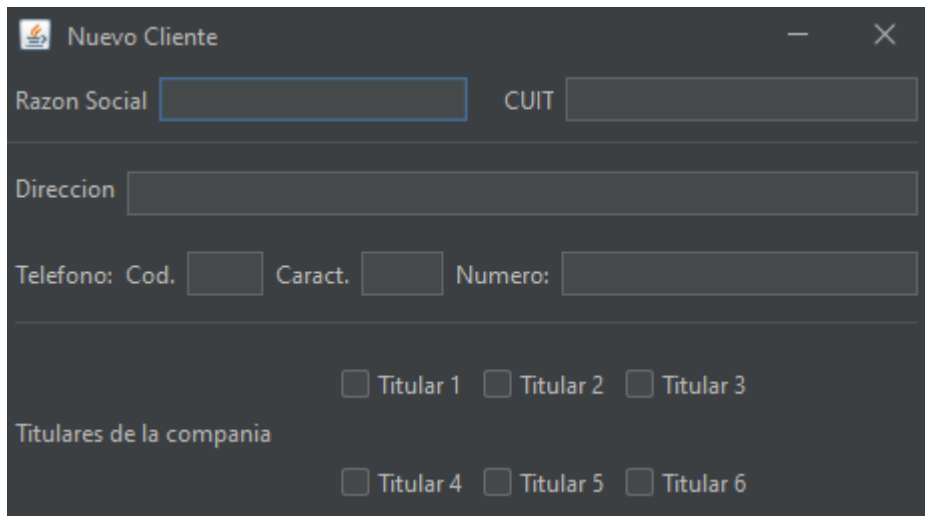
Función: Nuevo Empleado – vistaNuevoEmpleado – Sin Implementar



The image shows a software window titled "Nuevo Empleado" with standard Windows window controls (minimize, maximize, close). The form contains the following fields and controls:

- Nombre:** A text input field.
- Apellido:** A text input field.
- Tipo:** A dropdown menu currently showing "DNI".
- Número:** A text input field.
- Telefono:** A section containing three sub-fields:
 - Cod.:** A small text input field.
 - Caract.:** A small text input field.
 - Numero:** A text input field.
- Legajo:** A text input field.
- Area:** A dropdown menu currently showing "Area 1".
- Guardar:** A button located at the bottom right of the form.

Función: Nuevo Cliente – vistaNuevoCliente – Sin Implementar



The screenshot shows a web form titled "Nuevo Cliente" with a dark theme. The form contains the following fields and controls:

- Razon Social:** A text input field.
- CUIT:** A text input field.
- Direccion:** A text input field.
- Telefono:** A section containing three input fields: "Cod." (short), "Caract." (medium), and "Numero:" (long).
- Titulares de la compania:** A section with six checkboxes labeled "Titular 1", "Titular 2", "Titular 3", "Titular 4", "Titular 5", and "Titular 6".

Implementación de Tests

- Test para Turnos

```
@Test
public void testInsertarTurno() {
    System.out.println("insertarTurno");
    String dia = "99";
    String hora = "99";
    String mecanico = "Mecanico99";
    String vehiculo = "Vehiculo99";
    String titular = "Titular99";
    String companiaSeguro = "Compania99";
    TurnoDAOImplSql instance = new TurnoDAOImplSql();
    boolean expResult = true;
    boolean result = instance.insertarTurno(dia, hora, mecanico, vehiculo, titular, companiaSeguro);
    assertEquals(expResult, result);
}

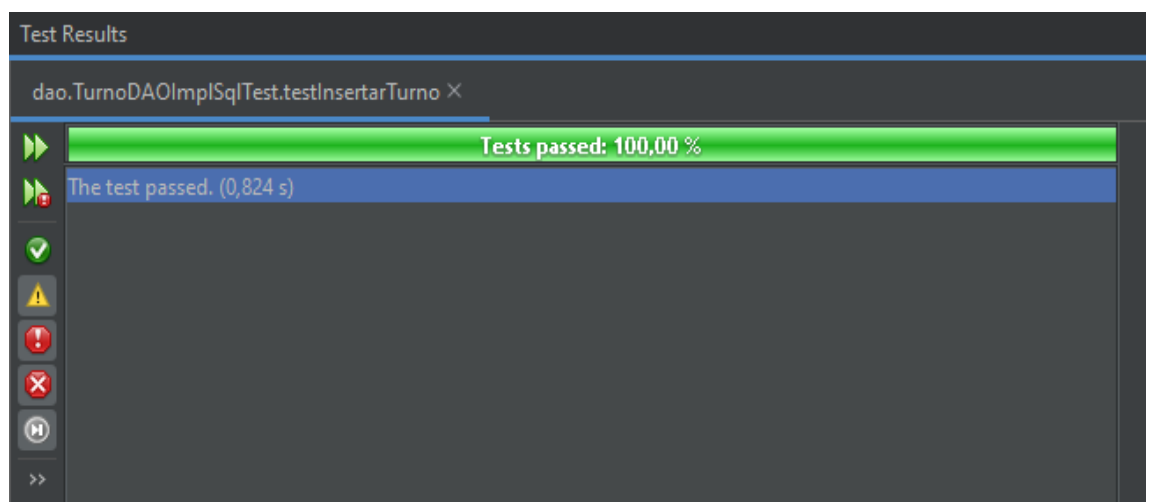
/**
 * Test of modificarTurno method, of class TurnoDAOImplSql.
 */
@Test
public void testModificarTurno() {
    System.out.println("modificarTurno");
    String nro = "10";
    String dia = "98";
    String hora = "98";
    String mecanico = "Mecanico98";
    TurnoDAOImplSql instance = new TurnoDAOImplSql();
    boolean expResult = true;
    boolean result = instance.modificarTurno(nro, dia, hora, mecanico);
    assertEquals(expResult, result);
}
```

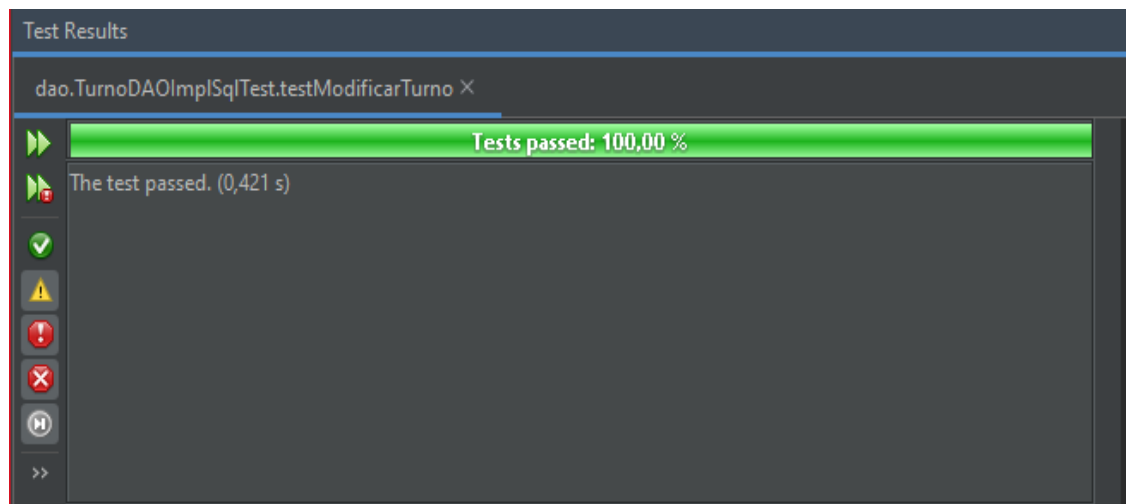
- Test para Vehículos

```
@Test
public void testConsultarVehiculo() {
    System.out.println("consultarVehiculo");
    int nroPoliza = 1;
    VehiculoDAOImplSql instance = new VehiculoDAOImplSql();
    VehiculoDTO expectedResult = new VehiculoDTO(1, "Modelo1", "Marca1", "11111111");
    VehiculoDTO result = instance.consultarVehiculo(nroPoliza);
    assertEquals(expectedResult.getNroPoliza(), result.getNroPoliza());
    assertEquals(expectedResult.getMarca(), result.getMarca());
    assertEquals(expectedResult.getModelo(), result.getModelo());
    assertEquals(expectedResult.getNroDNITitular(), result.getNroDNITitular());
}

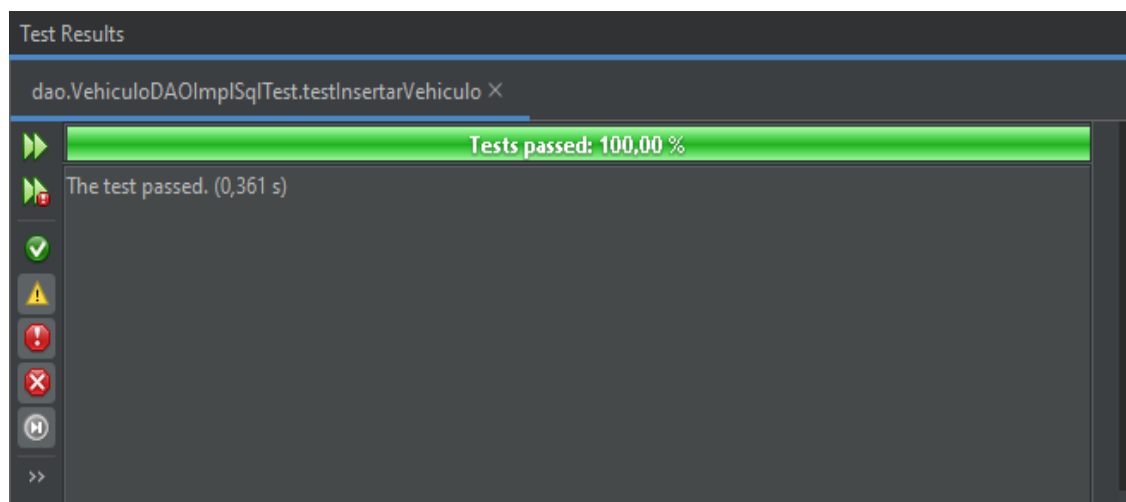
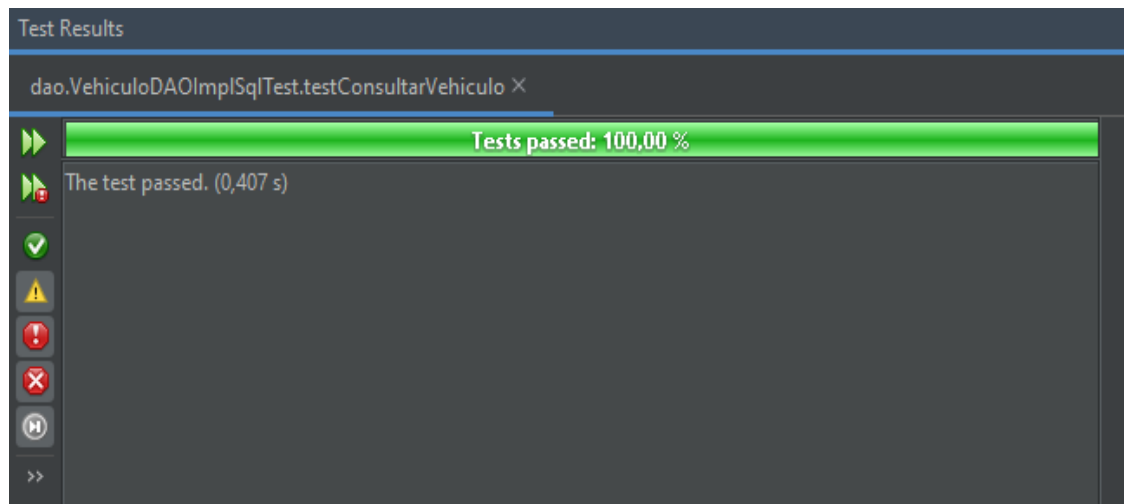
/**
 * Test of insertarVehiculo method, of class VehiculoDAOImplSql.
 */
@Test
public void testInsertarVehiculo() {
    System.out.println("insertarVehiculo");
    int nroPoliza = 98;
    String modelo = "Modelo99";
    String marca = "Marca99";
    String nroDNITitular = "999999999";
    VehiculoDAOImplSql instance = new VehiculoDAOImplSql();
    boolean expectedResult = true;
    boolean result = instance.insertarVehiculo(nroPoliza, modelo, marca, nroDNITitular);
    assertEquals(expectedResult, result);
}
```

- Resultados Turnos:





- Resultados Vehículos:



Patrones Localizados / Implementados:

- Patrón DAO
- Abstract Factory (Dentro de patrón DAO)
- Factory Method (Utilizado para la creación de Modelos)
- Singleton (Conexión con base de datos SQL)
- MVC (Modelo Vista Controlador)

Aclaraciones:

Los casos de herencia utilizados en distintos momentos se consideran otro Patrón de diseño implementado.