Facultad de Ingeniería

Desarrollo web Integrado - Sección 18186

TITULO

Aplicación Web para la Simulación de Financiamiento y Cotización de Automóviles Toyota

Integrantes

Alarcon Oroncuy, Claudia Isabel U17200025

Bendezú Cucho, Abel Angel U22222206

Jaramillo Rojas Moisés Alonso U19220767

Melgarejo Vásquez, Maddiel Iroska U22240907

Camargo Pumahuacre, Aaron Gustavo U20307837

Docente

Chicana Aspajo Jorge

Lima, Perú – 2025

Introducción

El presente informe describe el desarrollo de una aplicación web orientada a la simulación de financiamiento para la adquisición de vehículos Toyota.
La solución busca responder a la creciente necesidad de los clientes de contar con herramientas digitales que permitan calcular de manera rápida y confiable sus cuotas, considerando inicial, plazo y tasa de interés.
Este proyecto se enmarca en el curso de Diseño Web Integrado, donde se aplican principios de programación web, arquitectura MVC, bases de datos relacionales y generación de reportes

Capítulo 1: Aspectos generales

Global cars – Plataforma de Financiamiento.

En la actualidad, gran parte de los concesionarios de autos en el país operan de manera tradicional, mediante locales físicos donde los asesores gestionan cotizaciones y financiamientos. Sin embargo, la tendencia digital ha impulsado la creación de plataformas online que permiten al cliente informarse y realizar procesos de manera remota.

El negocio pertenece al rubro automotriz, específicamente en la venta y financiamiento de vehículos de la marca Toyota.

Misión y visión

- Misión: Brindar a los clientes una plataforma digital confiable para simular y obtener cotizaciones de financiamiento de vehículos Toyota, garantizando transparencia y accesibilidad.
- Visión: Convertirse en la principal plataforma web de financiamiento automotriz del país, ampliando sus servicios hacia una concesionaria digital completa.

Problemática que afronta

Los clientes no cuentan con herramientas digitales que les permitan calcular de manera clara y precisa las cuotas de financiamiento. Esto genera retrasos, sobrecarga de consultas al área de asesores y pérdida de potenciales ventas.

Alternativas de solución

- Desarrollar una plataforma web con simulador de financiamiento y generación de cotizaciones en PDF.

1.7 Solución elegida y justificación del proyecto

Se optó por la tercera alternativa: una plataforma web con simulador.
La decisión se justifica porque permite:

- Brindar información inmediata y transparente al cliente.
- Reducir la carga de trabajo en asesores.
- Incrementar la tasa de conversión de consultas en ventas.
- Posicionar a la empresa en el mercado digital.

Capítulo 2: Alcance

2.1 Requerimientos funcionales

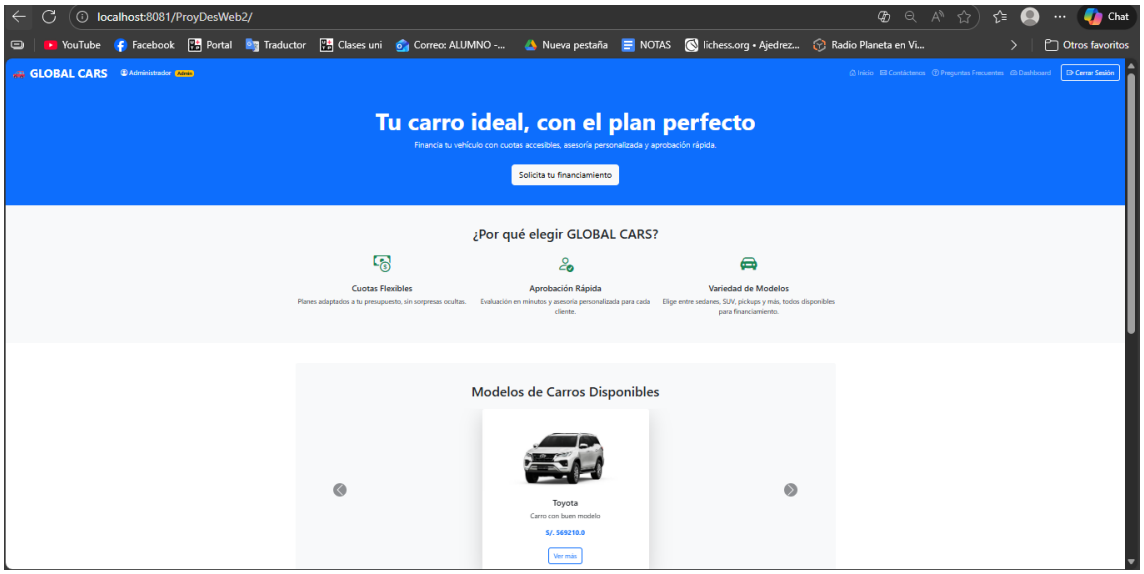
- Simular cuotas en función de precio, inicial, plazo y tasa.
- Mostrar catálogo básico de vehículos Toyota.
- Generar pre-cotización en formato PDF.
- Permitir gestión de modelos y tasas por parte del asesor.
- Registrar simulaciones en la base de datos.

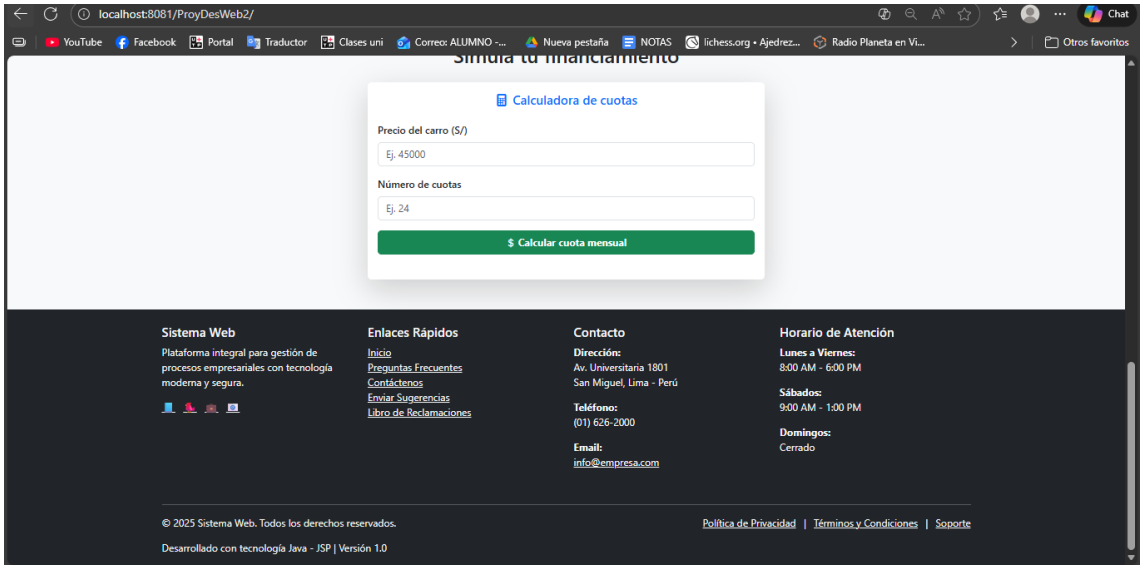
2.2 Requerimientos no funcionales

- La aplicación debe ser responsiva (usable en PC, tablet y móvil).
- El sistema debe utilizar el patrón MVC.
- La base de datos debe ser MySQL.
- El código debe ser mantenible y documentado.

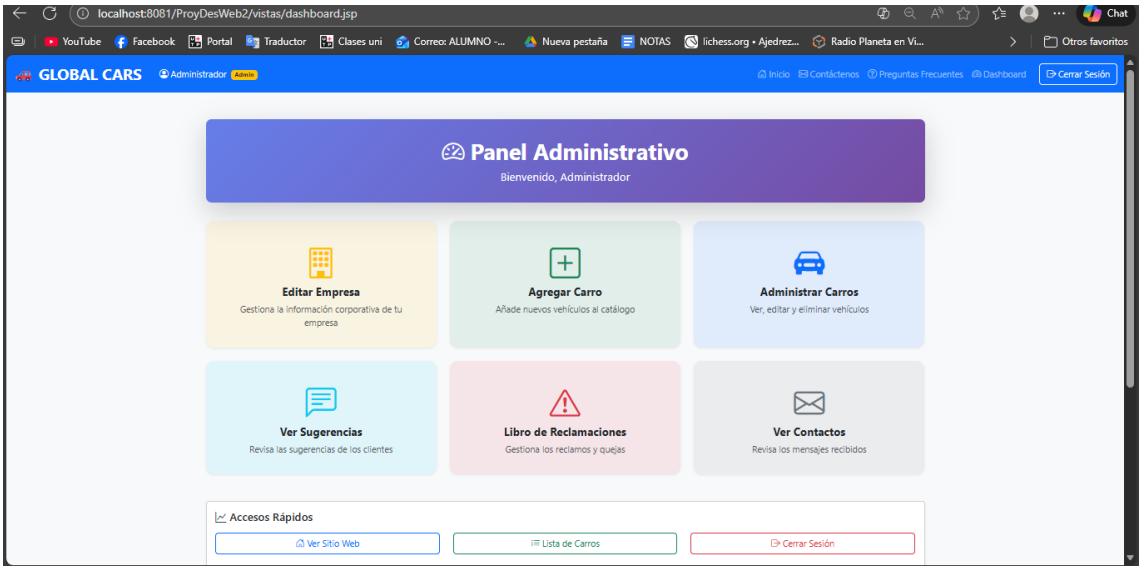
2.3 Funcionalidades adicionales exigidas

- Layout del sitio web: cabecera, menú de navegación, pie de página.

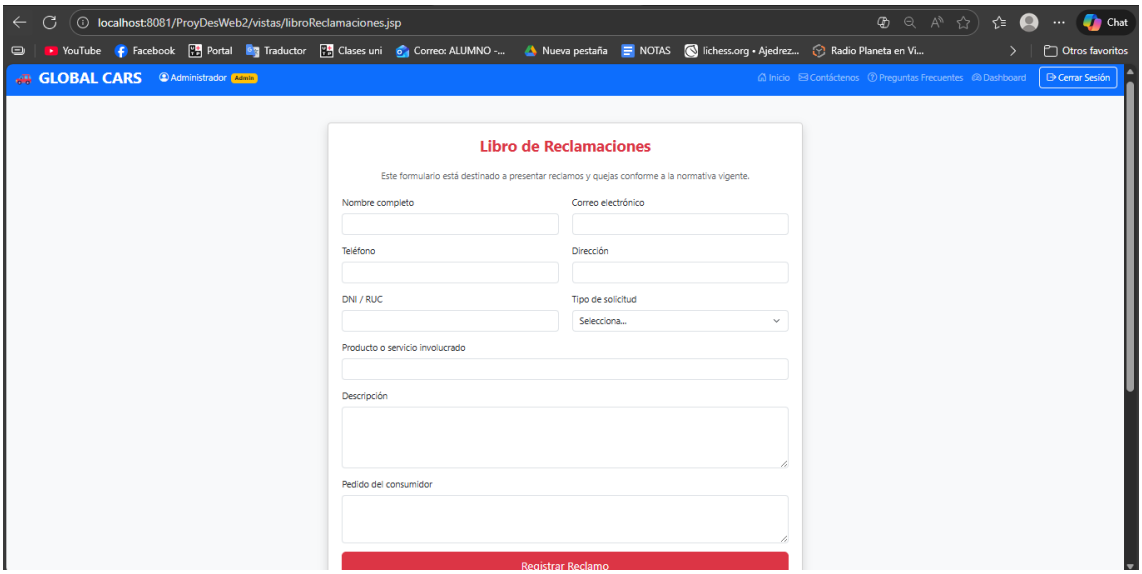




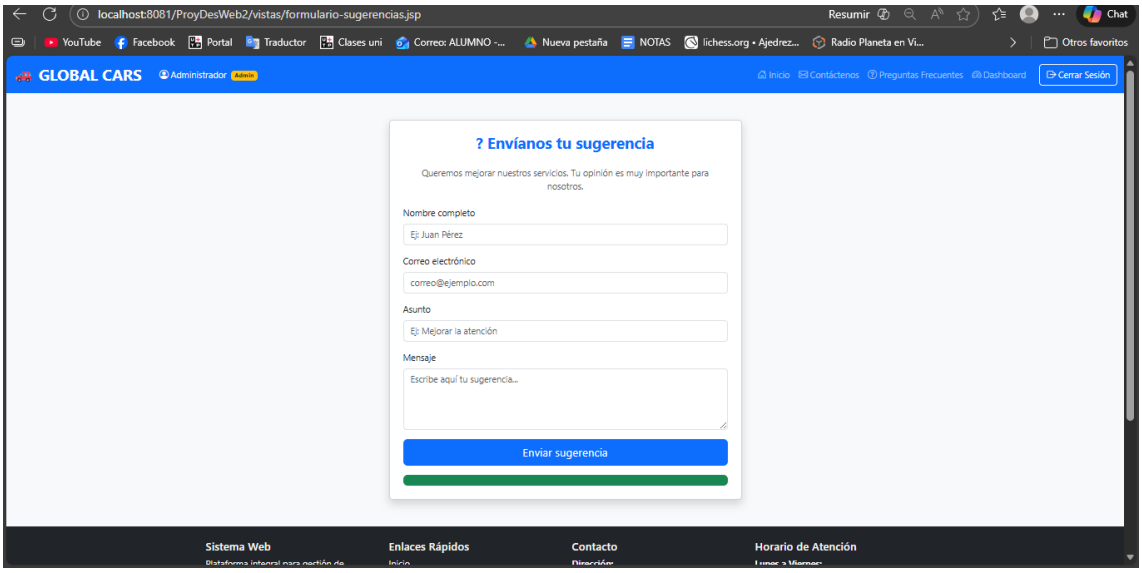
- **Dashboard Administrador:**



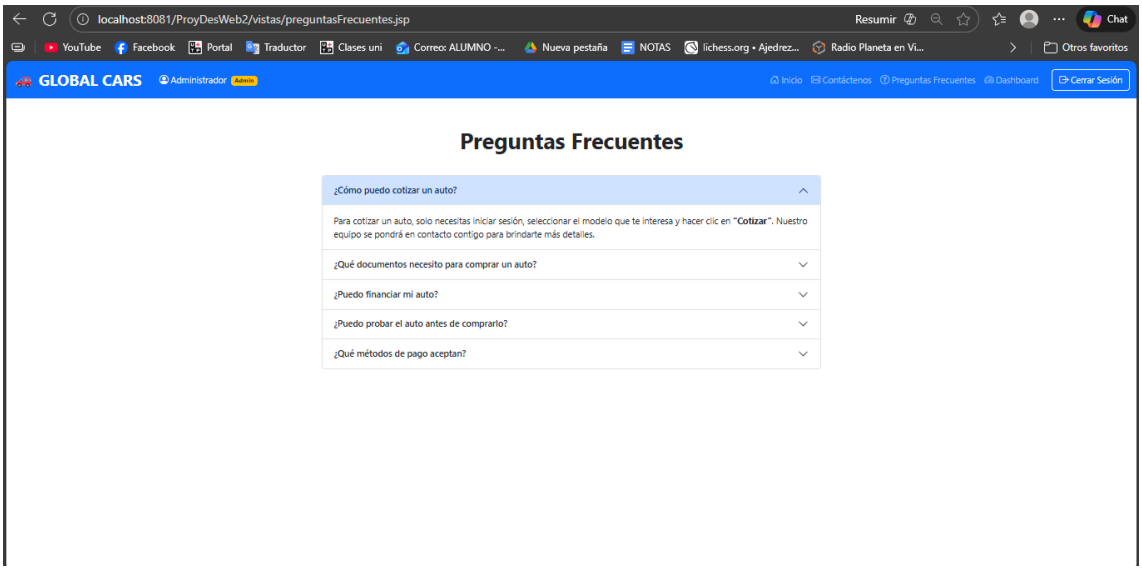
- **Formulario para libro de reclamaciones:**



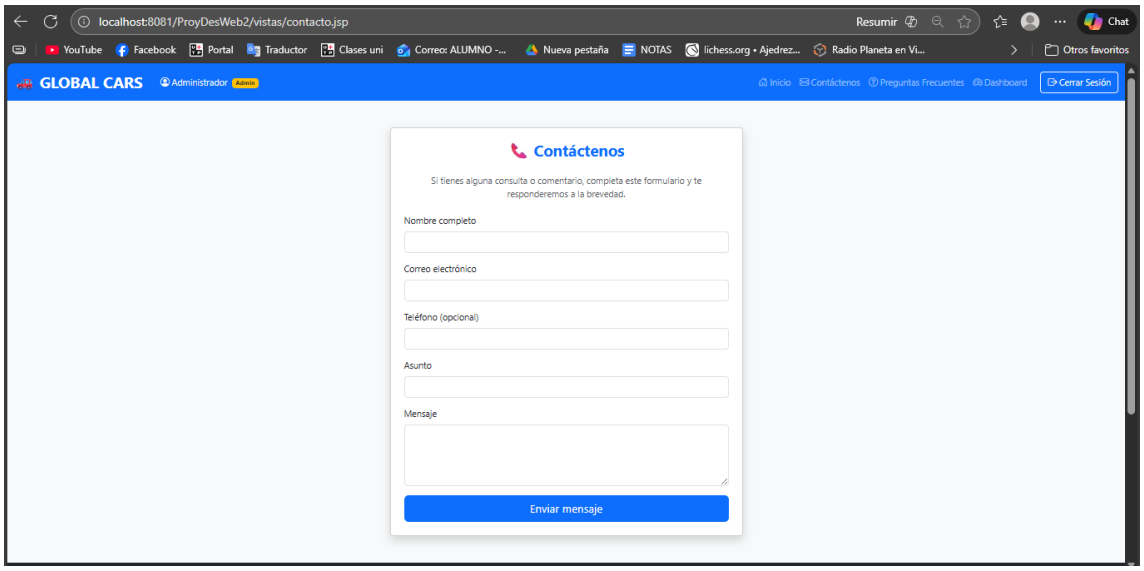
- **Formulario de sugerencias:**



- **Preguntas frecuentes (FAQ)**

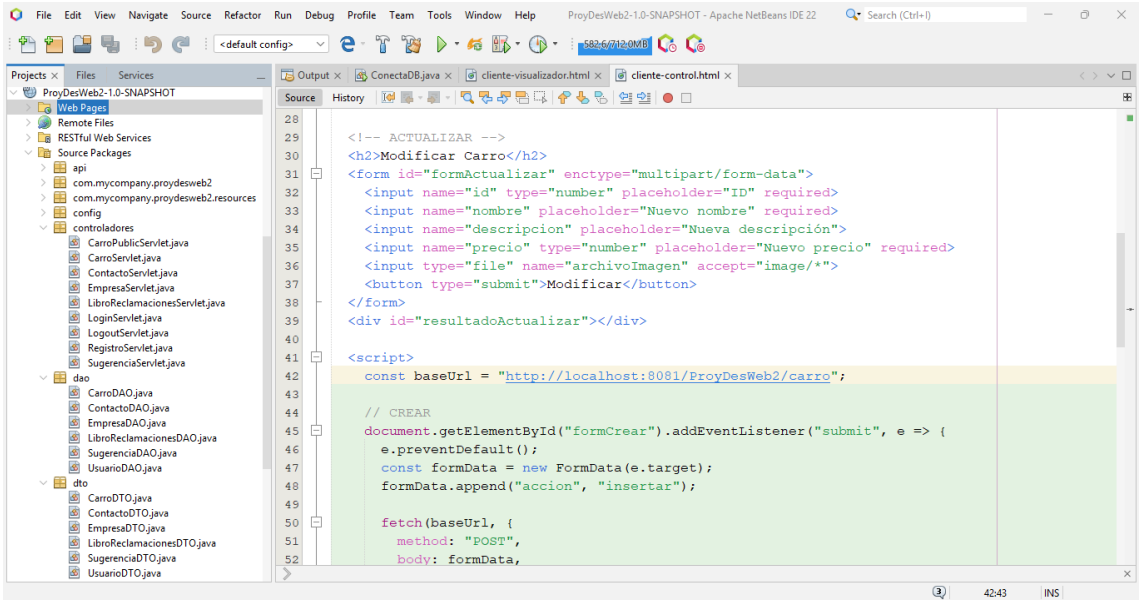


- **Contáctenos / Ubícanos: formulario**



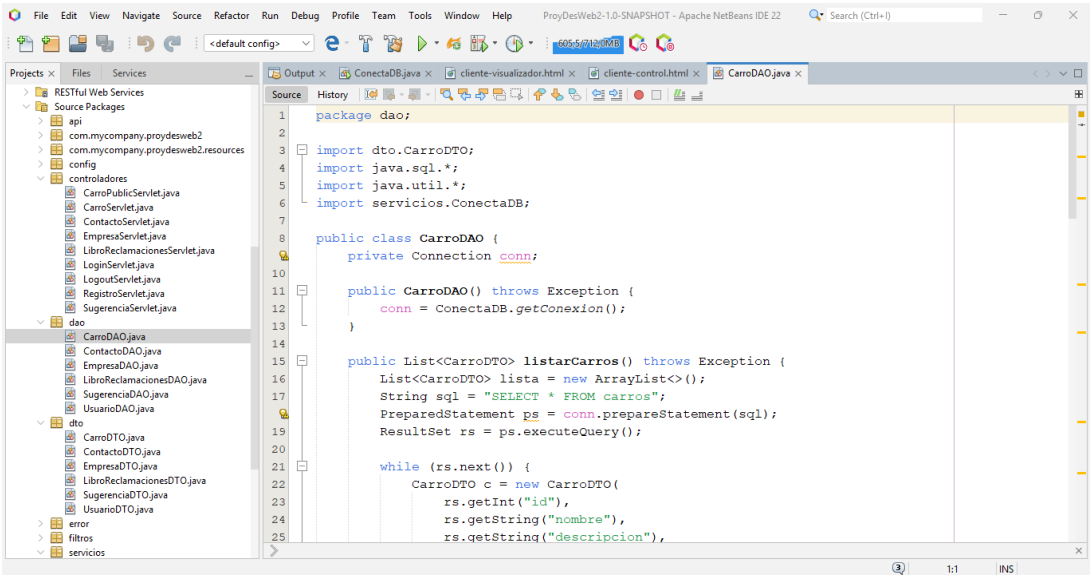
Capítulo 3:

3.1 Implementación de patrón MVC

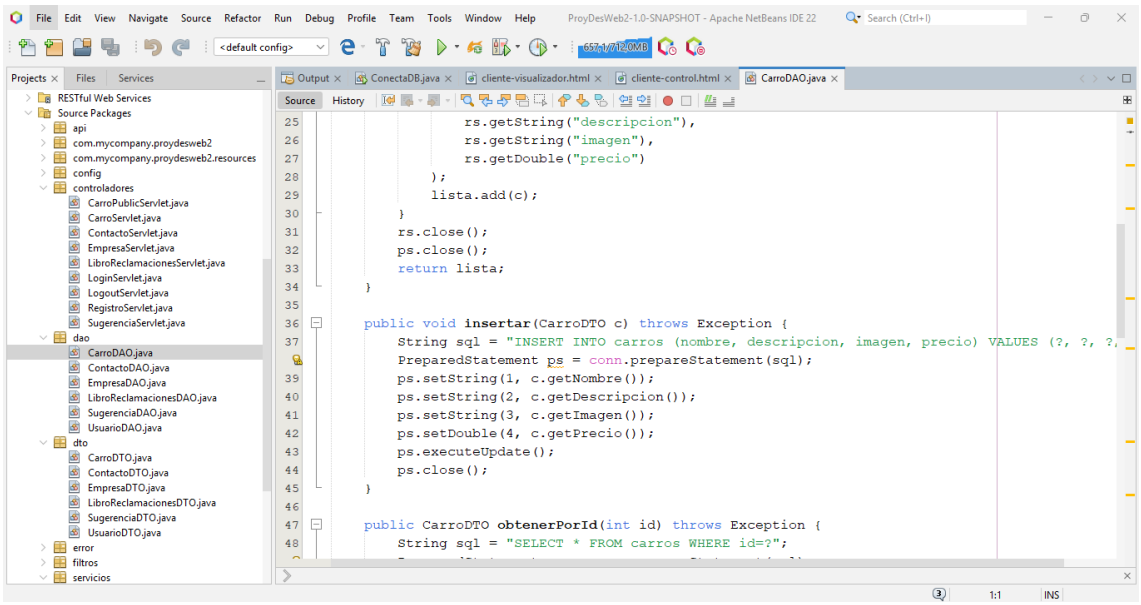


3.2 Funcionalidad del sistema

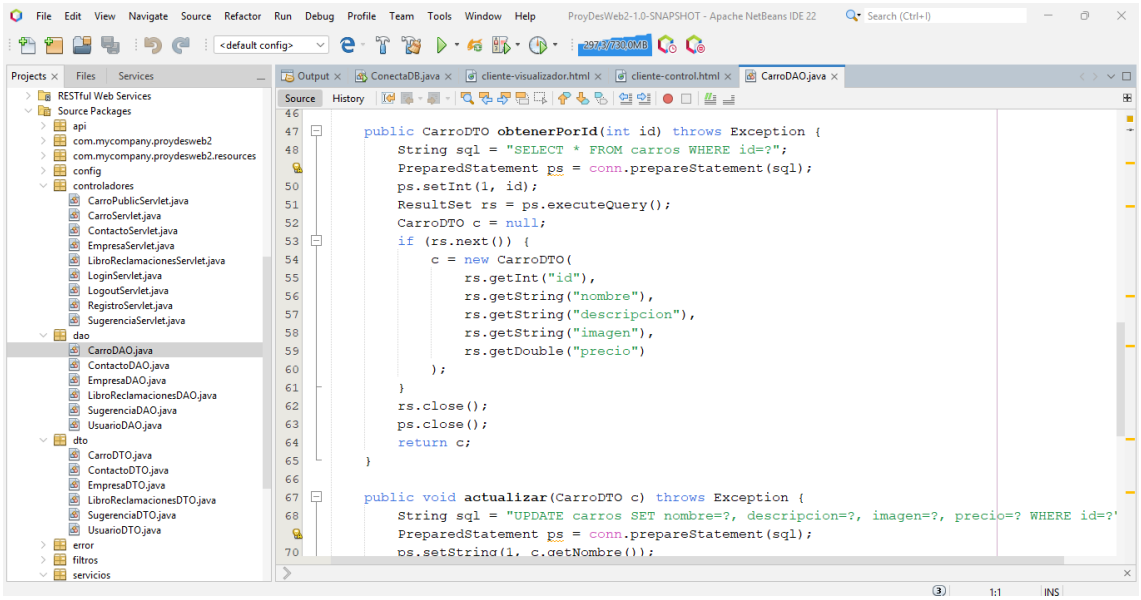
- **CarroDAO**

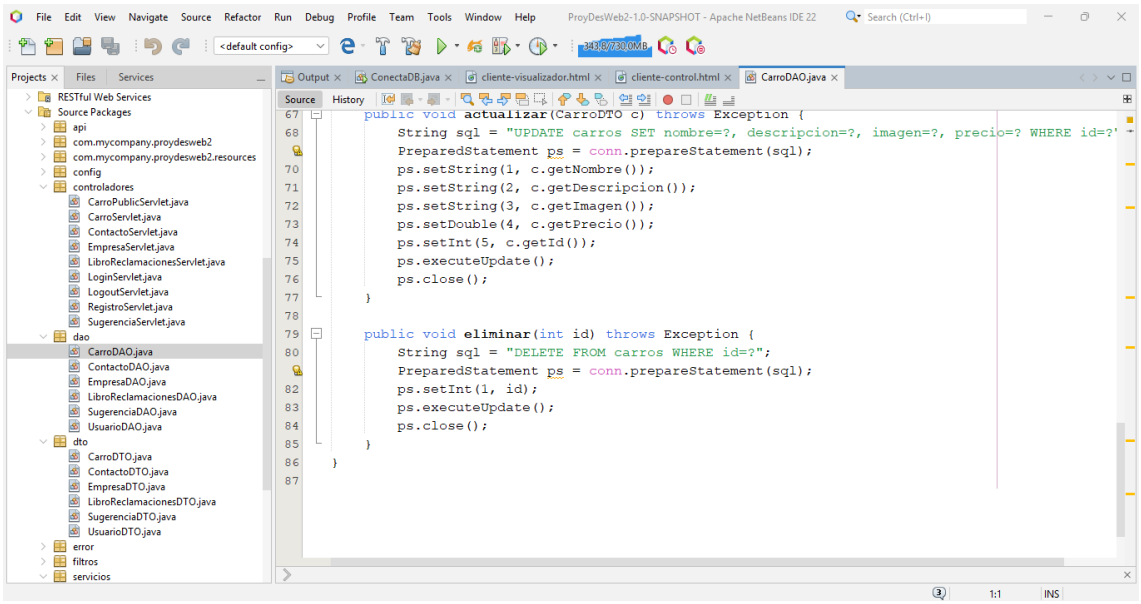


```
package dao;

import dto.CarroDTO;
import java.sql.*;
import java.util.*;
import servicios.ConectaDB;

public class CarroDAO {
    private Connection conn;

    public CarroDAO() throws Exception {
        conn = ConectaDB.getConexion();
    }

    public List<CarroDTO> listarCarros() throws Exception {
        List<CarroDTO> lista = new ArrayList<>();
        String sql = "SELECT * FROM carros";
        PreparedStatement ps = conn.prepareStatement(sql);
        ResultSet rs = ps.executeQuery();

        while (rs.next()) {
            CarroDTO c = new CarroDTO(
                rs.getInt("id"),
                rs.getString("nombre"),
                rs.getString("descripcion"),
```

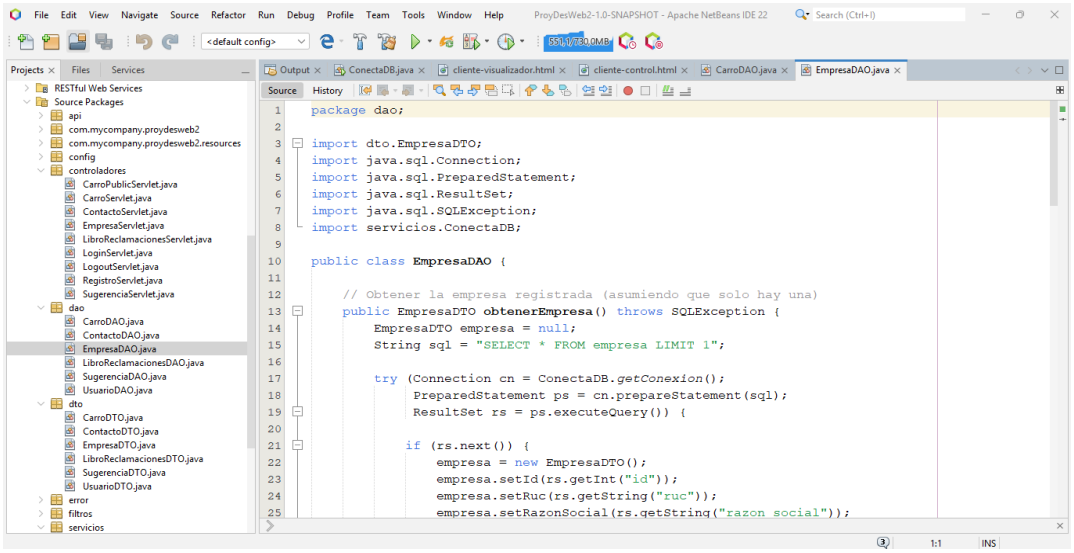


```
                rs.getString("descripcion"),
                rs.getString("imagen"),
                rs.getDouble("precio")
            );
            lista.add(c);
        }
        rs.close();
        ps.close();
        return lista;
    }

    public void insertar(CarroDTO c) throws Exception {
        String sql = "INSERT INTO carros (nombre, descripcion, imagen, precio) VALUES (?, ?, ?,
        PreparedStatement ps = conn.prepareStatement(sql);
        ps.setString(1, c.getNombre());
        ps.setString(2, c.getDescripcion());
        ps.setString(3, c.getImagen());
        ps.setDouble(4, c.getPrecio());
        ps.executeUpdate();
        ps.close();
    }

    public CarroDTO obtenerPorId(int id) throws Exception {
        String sql = "SELECT * FROM carros WHERE id=?";
```

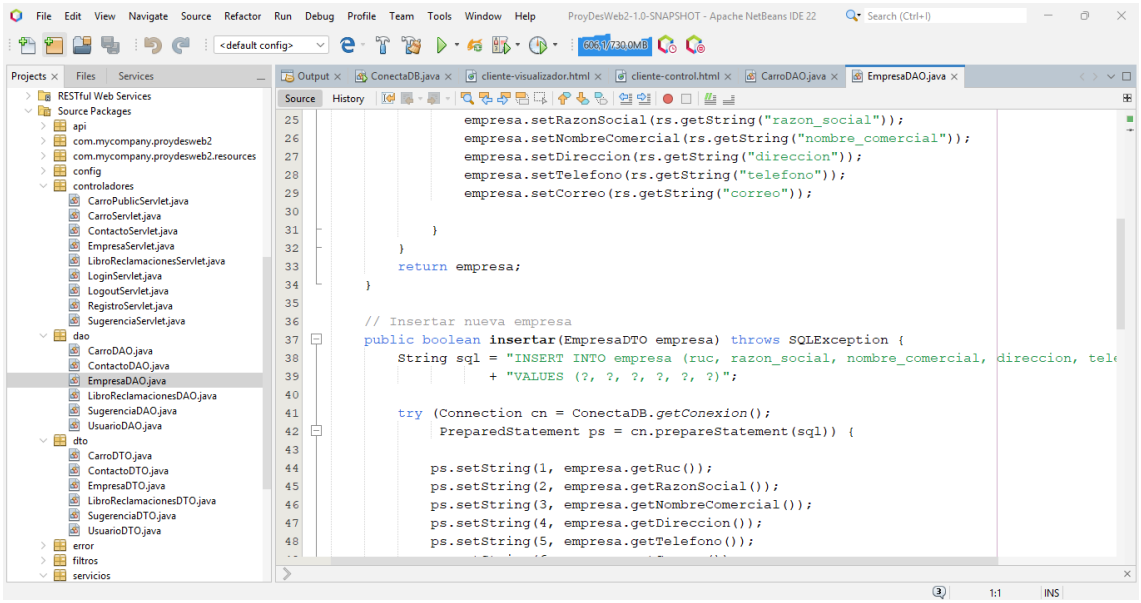

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
Output
ConectaDB.java
cliente-visualizador.html
cliente-control.html
CarroDAO.java
Source
History
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
public CarroDTO obtenerPorId(int id) throws Exception {
String sql = "SELECT * FROM carros WHERE id=?";
PreparedStatement ps = conn.prepareStatement(sql);
ps.setInt(1, id);
ResultSet rs = ps.executeQuery();
CarroDTO c = null;
if (rs.next()) {
c = new CarroDTO(
rs.getInt("id"),
rs.getString("nombre"),
rs.getString("descripcion"),
rs.getString("imagen"),
rs.getDouble("precio")
);
}
rs.close();
ps.close();
return c;
}
public void actualizar(CarroDTO c) throws Exception {
String sql = "UPDATE carros SET nombre=?, descripcion=?, imagen=?, precio=? WHERE id=?"
PreparedStatement ps = conn.prepareStatement(sql);
ps.setString(1, c.getNombre());
1:1
INS


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
Output
ConectaDB.java
cliente-visualizador.html
cliente-control.html
CarroDAO.java
Source
History
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
public void actualizar(CarroDTO c) throws Exception {
String sql = "UPDATE carros SET nombre=?, descripcion=?, imagen=?, precio=? WHERE id=?"
PreparedStatement ps = conn.prepareStatement(sql);
ps.setString(1, c.getNombre());
ps.setString(2, c.getDescripcion());
ps.setString(3, c.getImagen());
ps.setDouble(4, c.getPrecio());
ps.setInt(5, c.getId());
ps.executeUpdate();
ps.close();
}
public void eliminar(int id) throws Exception {
String sql = "DELETE FROM carros WHERE id=?";
PreparedStatement ps = conn.prepareStatement(sql);
ps.setInt(1, id);
ps.executeUpdate();
ps.close();
}
}
1:1
INS

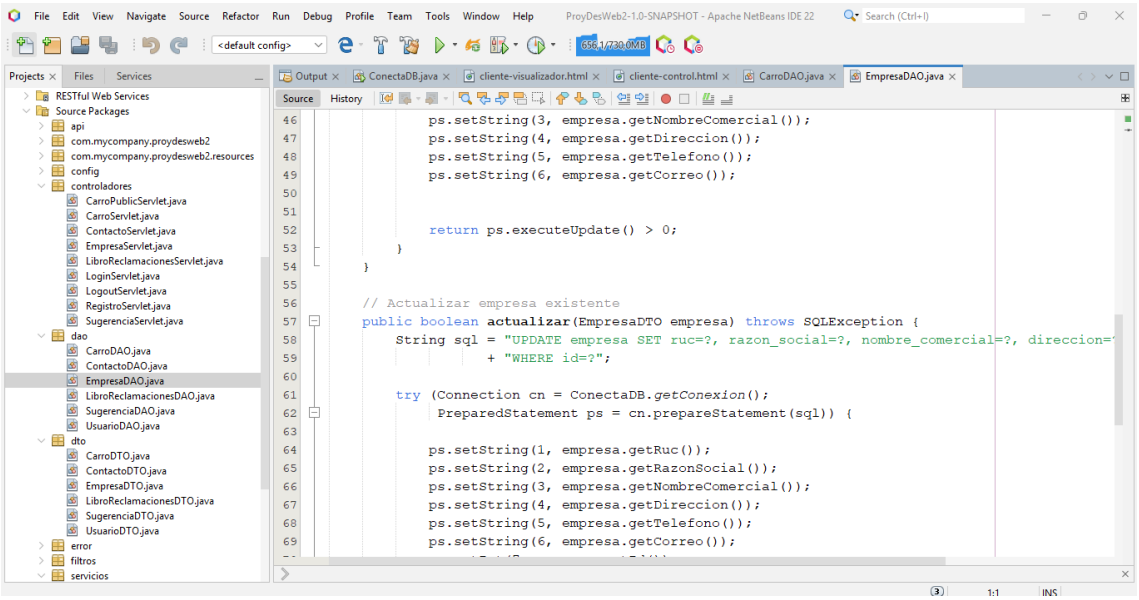
- **EmpresaDAO**

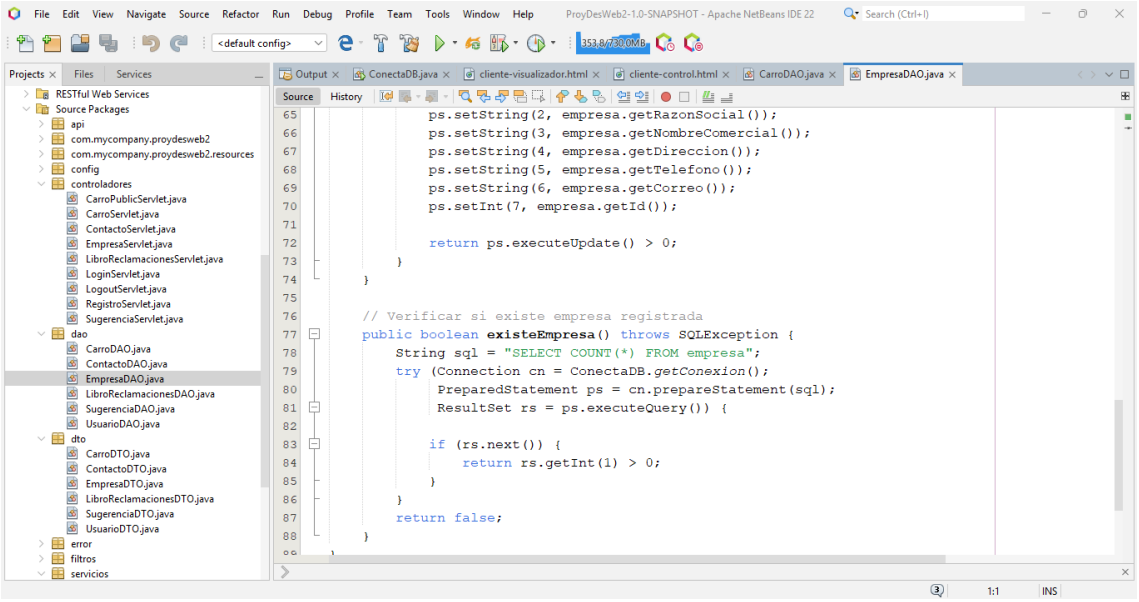


```
package dao;

import dto.EmpresaDTO;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import servicios.ConectaDB;

public class EmpresaDAO {

    // Obtener la empresa registrada (asumiendo que solo hay una)
    public EmpresaDTO obtenerEmpresa() throws SQLException {
        EmpresaDTO empresa = null;
        String sql = "SELECT * FROM empresa LIMIT 1";

        try (Connection cn = ConectaDB.getConexion();
             PreparedStatement ps = cn.prepareStatement(sql);
             ResultSet rs = ps.executeQuery()) {

            if (rs.next()) {
                empresa = new EmpresaDTO();
                empresa.setId(rs.getInt("id"));
                empresa.setRuc(rs.getString("ruc"));
                empresa.setRazonSocial(rs.getString("razon_social"));
```



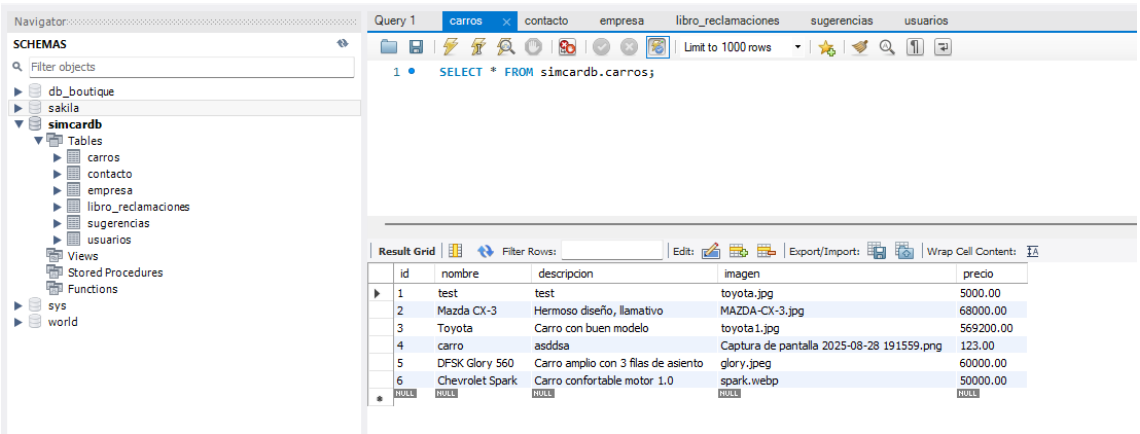
```
                empresa.setRazonSocial(rs.getString("razon_social"));
                empresa.setNombreComercial(rs.getString("nombre_comercial"));
                empresa.setDireccion(rs.getString("direccion"));
                empresa.setTelefono(rs.getString("telefono"));
                empresa.setCorreo(rs.getString("correo"));

            }
        }
        return empresa;
    }

    // Insertar nueva empresa
    public boolean insertar(EmpresaDTO empresa) throws SQLException {
        String sql = "INSERT INTO empresa (ruc, razon_social, nombre_comercial, direccion, tele
                   + "VALUES (?, ?, ?, ?, ?, ?)";

        try (Connection cn = ConectaDB.getConexion();
             PreparedStatement ps = cn.prepareStatement(sql)) {

            ps.setString(1, empresa.getRuc());
            ps.setString(2, empresa.getRazonSocial());
            ps.setString(3, empresa.getNombreComercial());
            ps.setString(4, empresa.getDireccion());
            ps.setString(5, empresa.getTelefono());
```



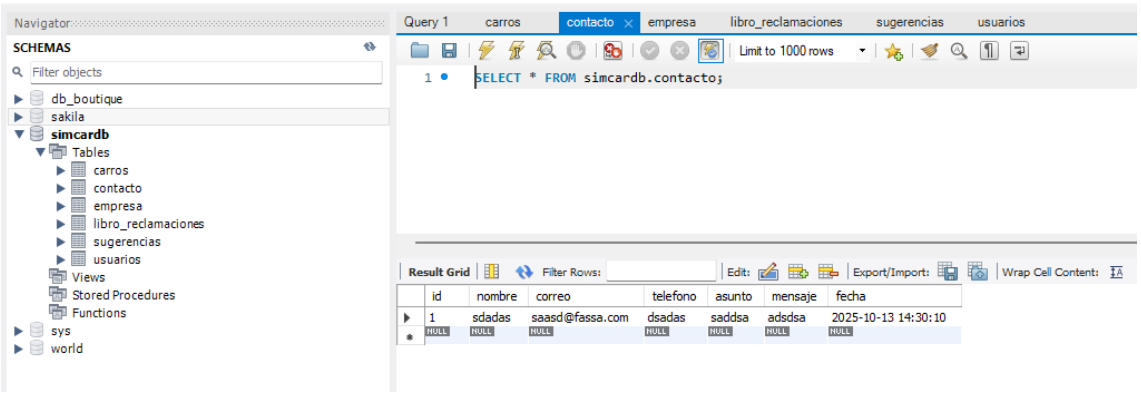


3.3 Diseño de la base de datos con al menos 6 tablas relacionadas

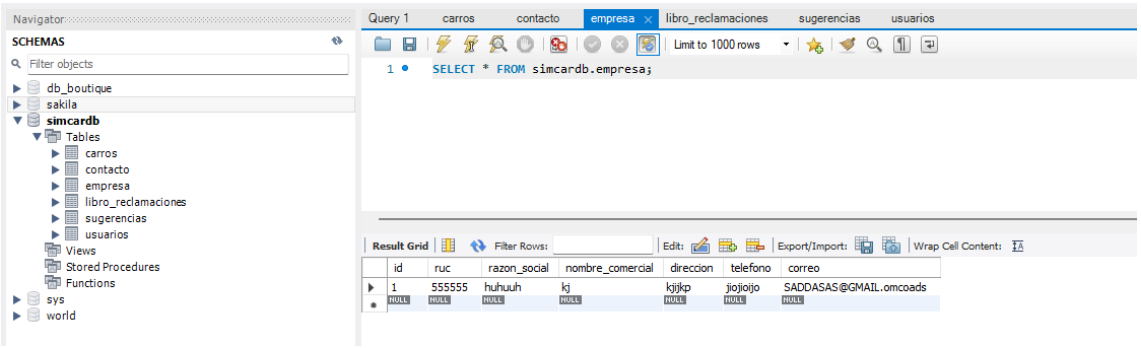
- **Tabla carros**



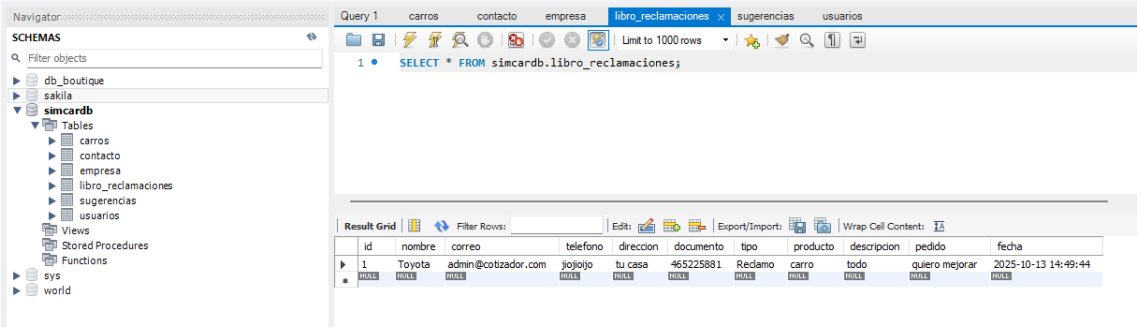
- **Tabla Contacto**



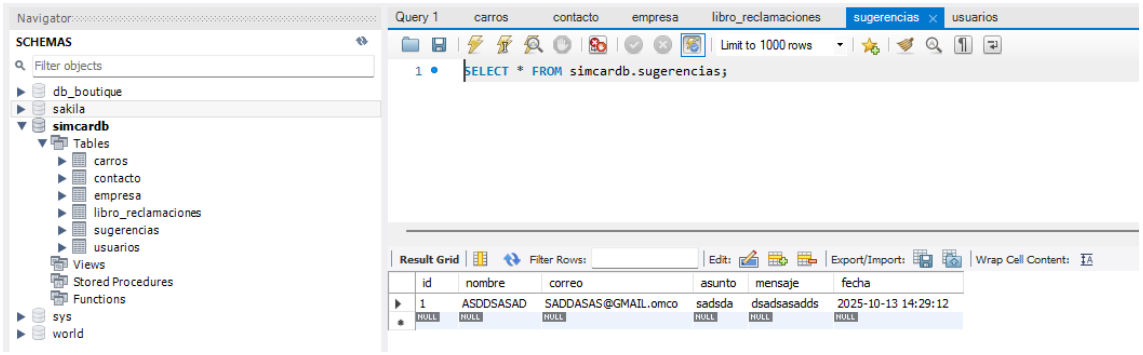
- **Tabla Empresa**



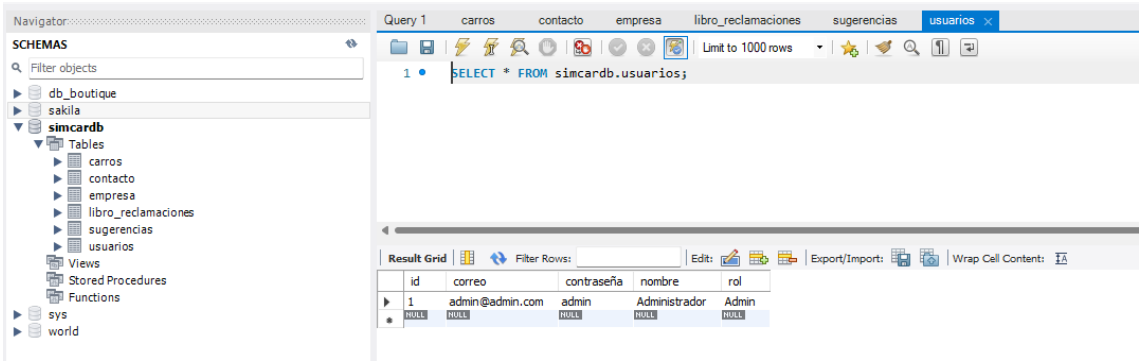
- **Tabla Libro_reclamaciones**



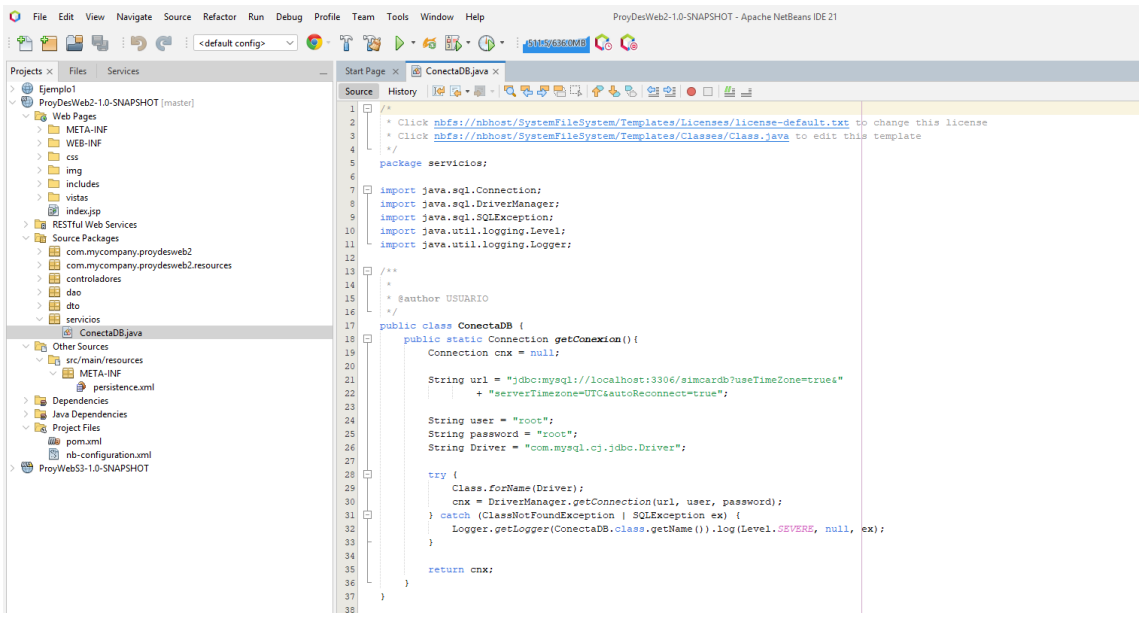
- **Tabla Sugerencias**



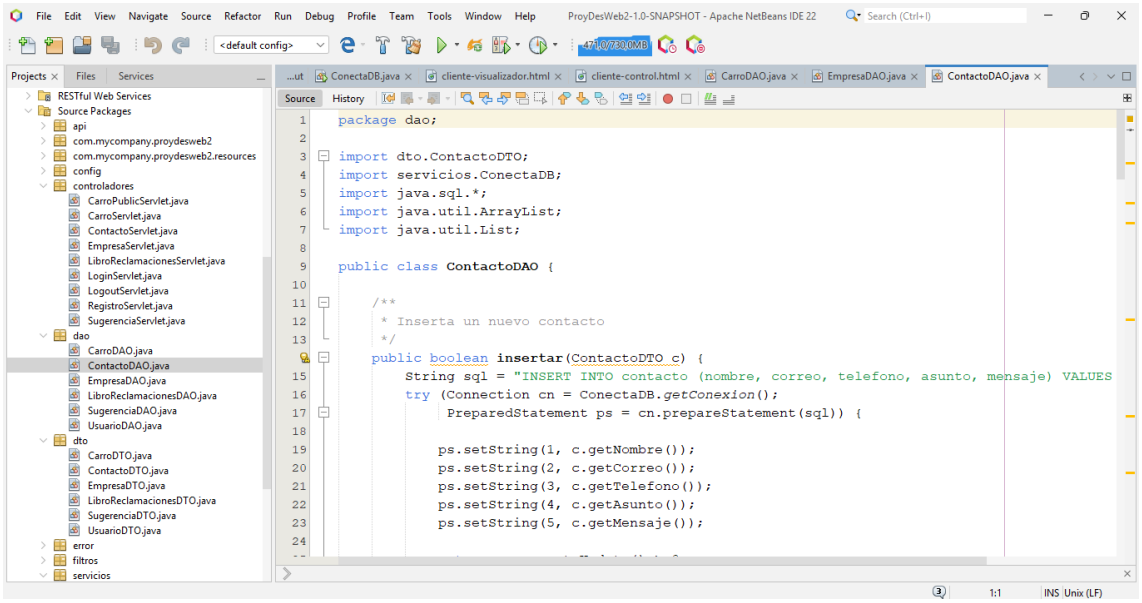
- **Tabla Usuarios**

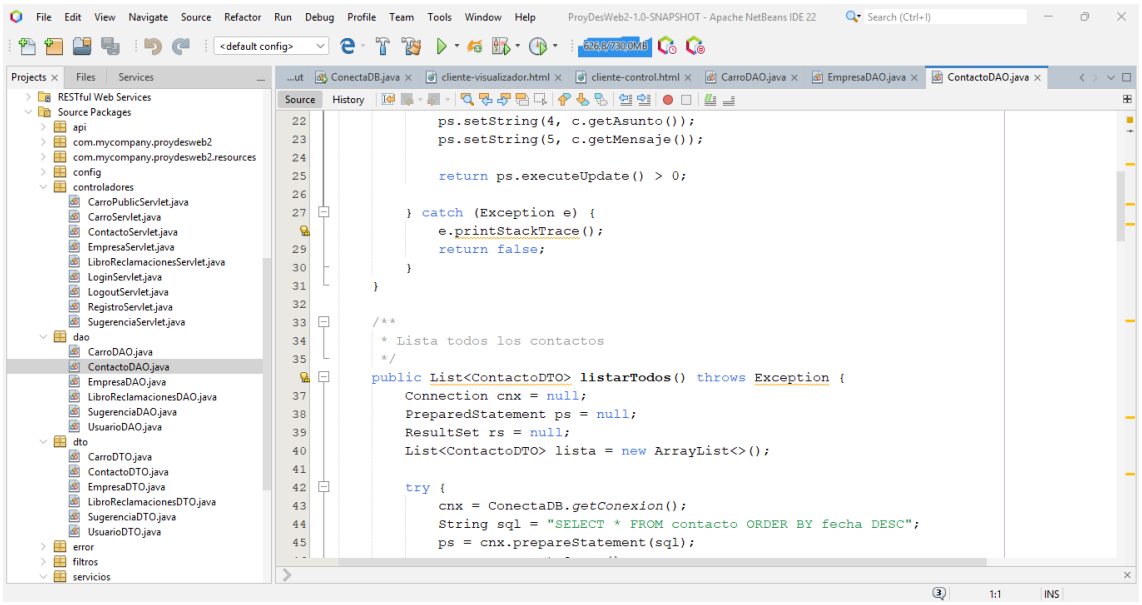


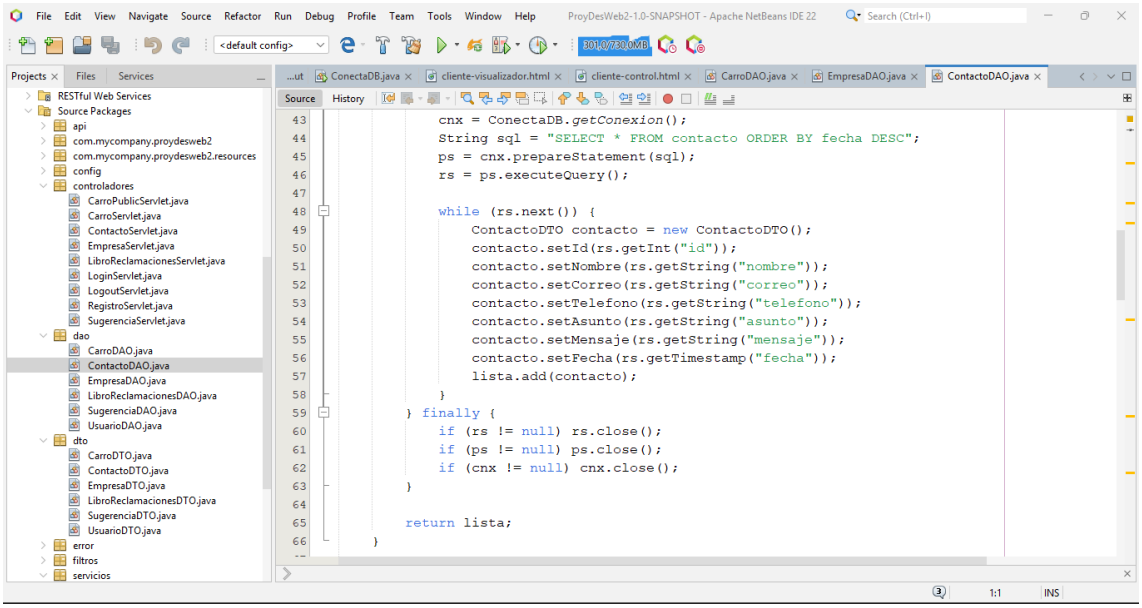
- **Conexión ConectaDB**

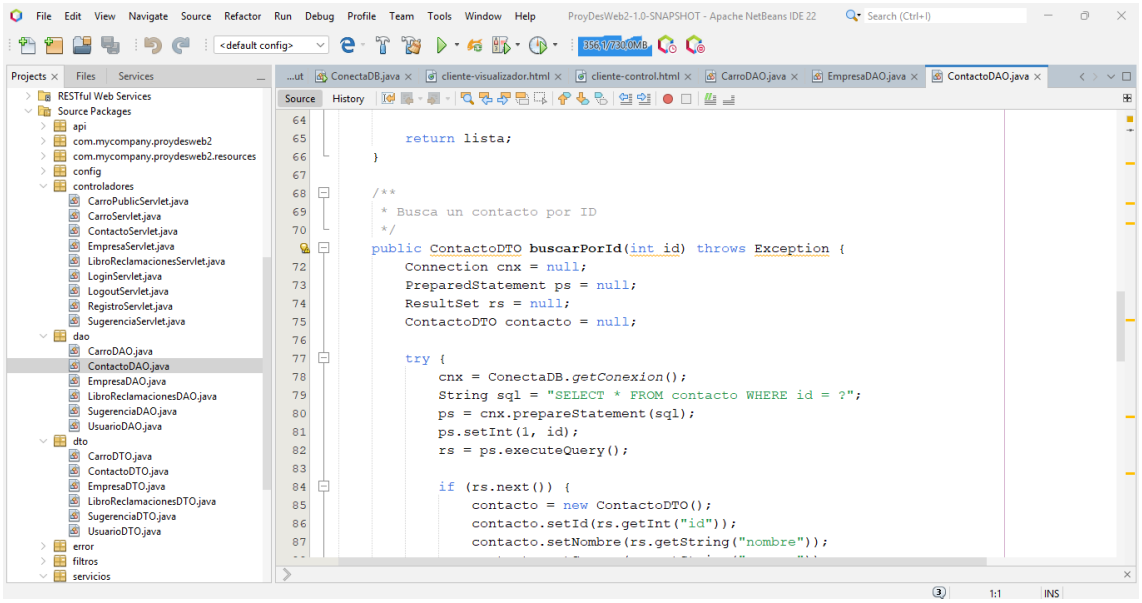


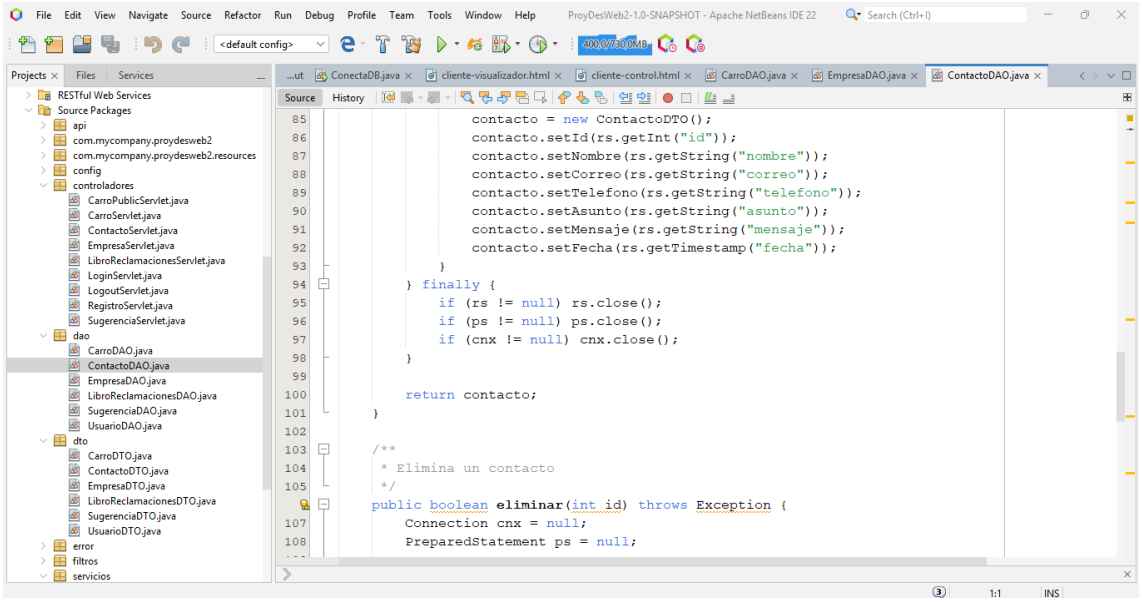
- **ContactoDAO**

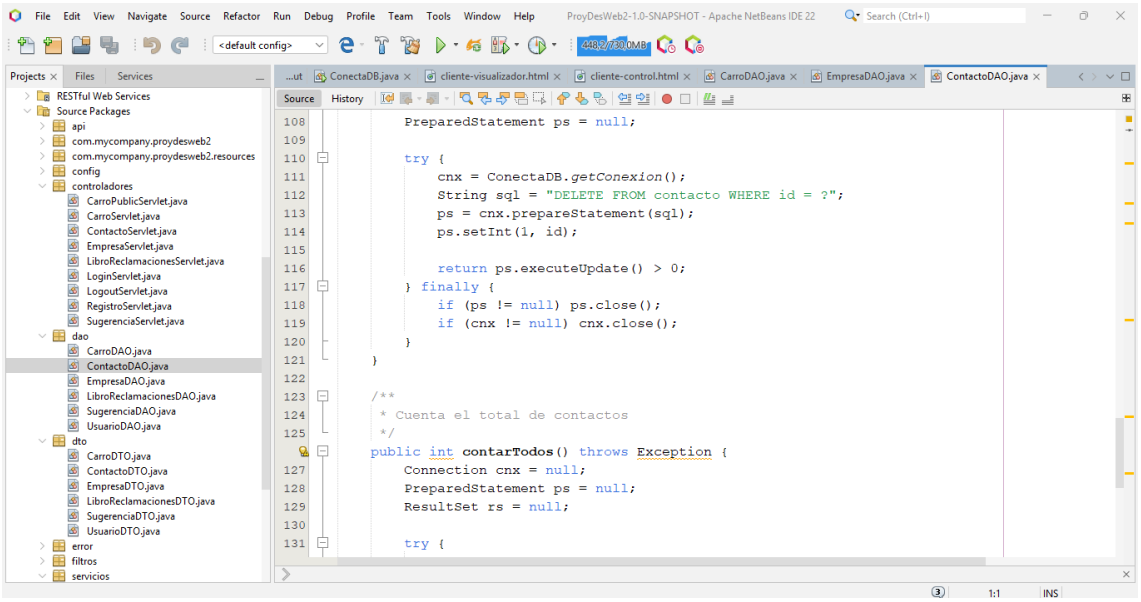


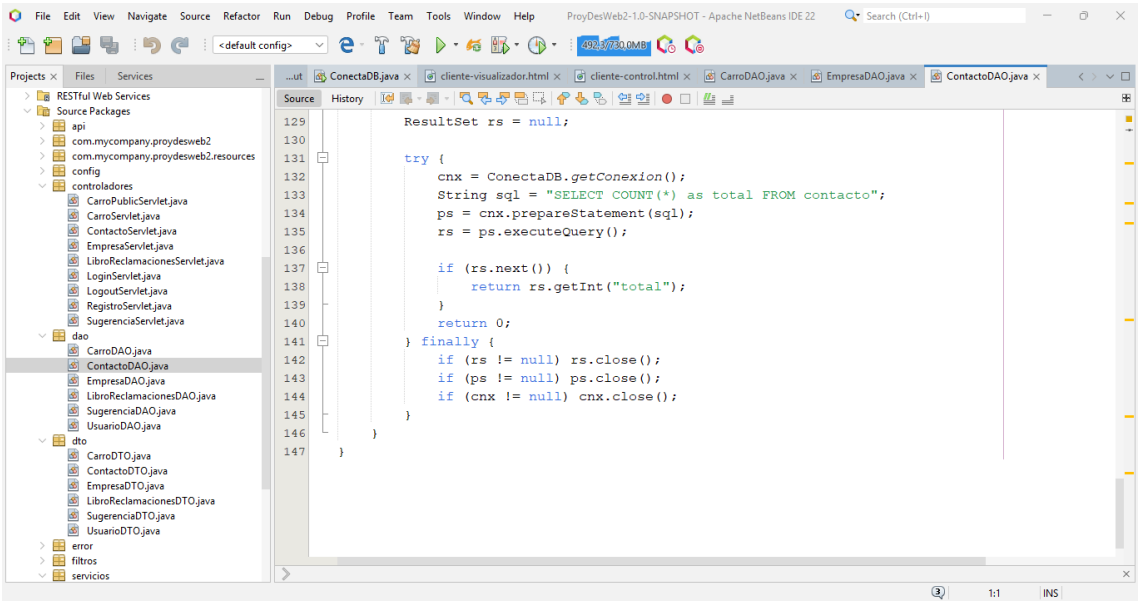



ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
ContactoDAO.java
cnx = ConectaDB.getConexion();
String sql = "SELECT * FROM contacto ORDER BY fecha DESC";
ps = cnx.prepareStatement(sql);
rs = ps.executeQuery();
while (rs.next()) {
ContactoDTO contacto = new ContactoDTO();
contacto.setId(rs.getInt("id"));
contacto.setNombre(rs.getString("nombre"));
contacto.setCorreo(rs.getString("correo"));
contacto.setTelefono(rs.getString("telefono"));
contacto.setAsunto(rs.getString("asunto"));
contacto.setMensaje(rs.getString("mensaje"));
contacto.setFecha(rs.getTimestamp("fecha"));
lista.add(contacto);
} finally {
if (rs != null) rs.close();
if (ps != null) ps.close();
if (cnx != null) cnx.close();
return lista;

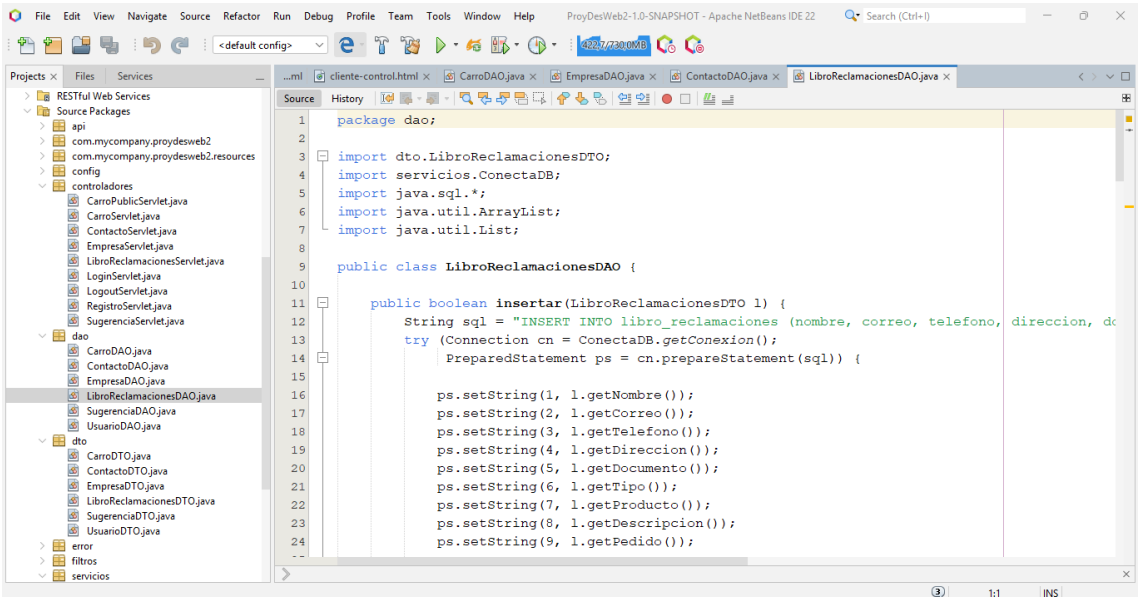

ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
ContactoDAO.java
return lista;
/**
* Busca un contacto por ID
*/
public ContactoDTO buscarPorId(int id) throws Exception {
Connection cnx = null;
PreparedStatement ps = null;
ResultSet rs = null;
ContactoDTO contacto = null;
try {
cnx = ConectaDB.getConexion();
String sql = "SELECT * FROM contacto WHERE id = ?";
ps = cnx.prepareStatement(sql);
ps.setInt(1, id);
rs = ps.executeQuery();
if (rs.next()) {
contacto = new ContactoDTO();
contacto.setId(rs.getInt("id"));
contacto.setNombre(rs.getString("nombre"));

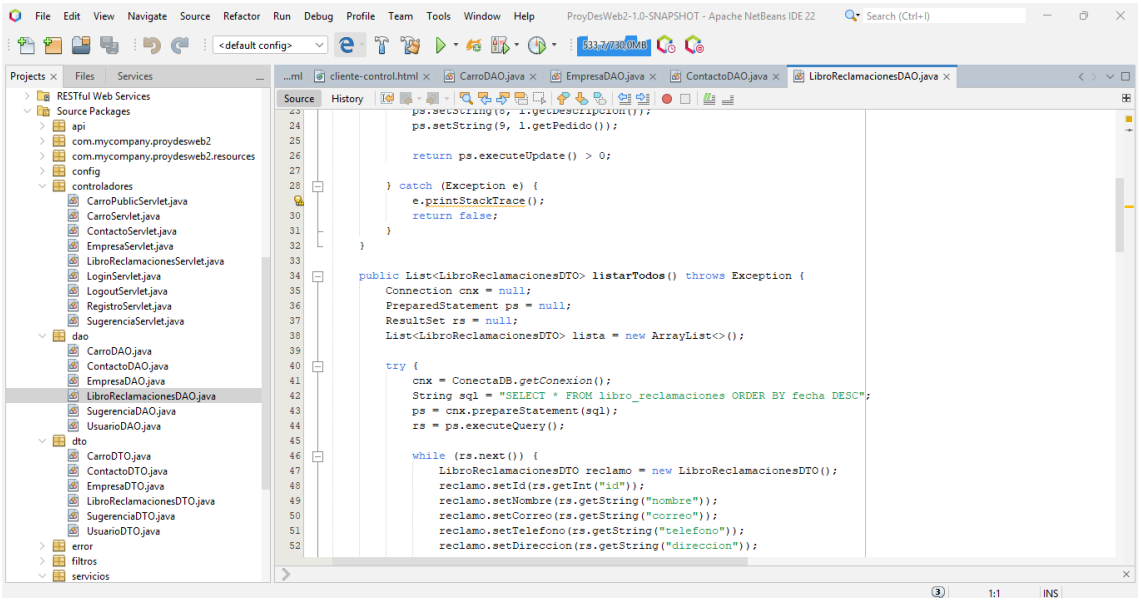

ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
ContactoDAO.java
contacto = new ContactoDTO();
contacto.setId(rs.getInt("id"));
contacto.setNombre(rs.getString("nombre"));
contacto.setCorreo(rs.getString("correo"));
contacto.setTelefono(rs.getString("telefono"));
contacto.setAsunto(rs.getString("asunto"));
contacto.setMensaje(rs.getString("mensaje"));
contacto.setFecha(rs.getTimestamp("fecha"));
} finally {
if (rs != null) rs.close();
if (ps != null) ps.close();
if (cnx != null) cnx.close();
return contacto;
/**
* Elimina un contacto
*/
public boolean eliminar(int id) throws Exception {
Connection cnx = null;
PreparedStatement ps = null;

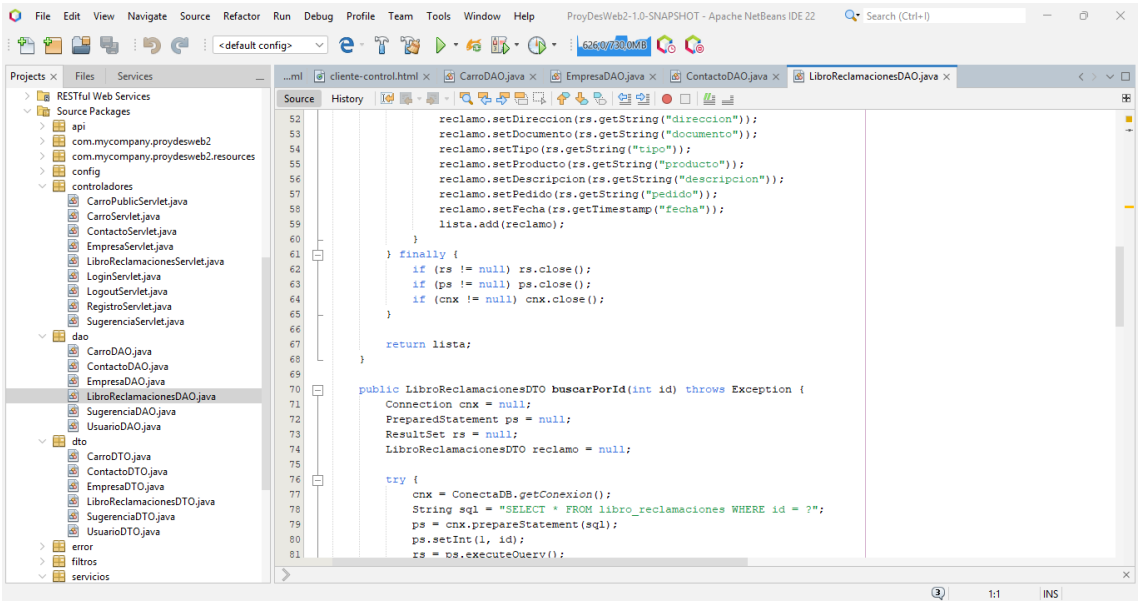

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
ConectaDB.java
cliente-visualizador.html
cliente-control.html
CarroDAO.java
EmpresaDAO.java
ContactoDAO.java
Source
History
PreparedStatement ps = null;
try {
cnx = ConectaDB.getConexion();
String sql = "DELETE FROM contacto WHERE id = ?";
ps = cnx.prepareStatement(sql);
ps.setInt(1, id);
return ps.executeUpdate() > 0;
} finally {
if (ps != null) ps.close();
if (cnx != null) cnx.close();
}
}
/**
* Cuenta el total de contactos
*/
public int contarTodos() throws Exception {
Connection cnx = null;
PreparedStatement ps = null;
ResultSet rs = null;
try {
1:1
INS

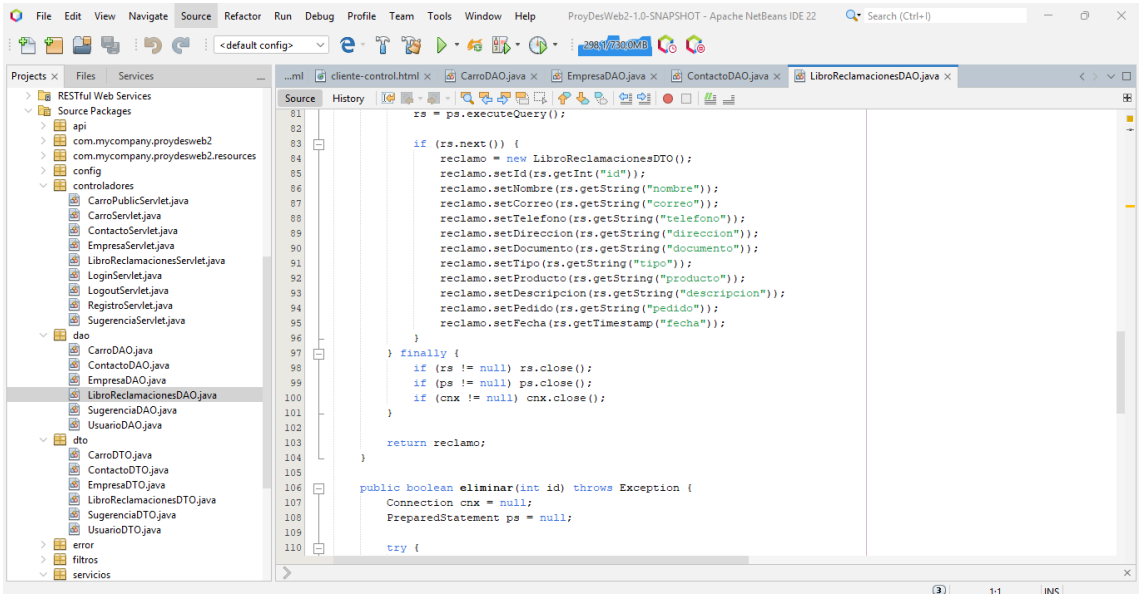

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
ConectaDB.java
cliente-visualizador.html
cliente-control.html
CarroDAO.java
EmpresaDAO.java
ContactoDAO.java
Source
History
ResultSet rs = null;
try {
cnx = ConectaDB.getConexion();
String sql = "SELECT COUNT(*) as total FROM contacto";
ps = cnx.prepareStatement(sql);
rs = ps.executeQuery();
if (rs.next()) {
return rs.getInt("total");
}
return 0;
} finally {
if (rs != null) rs.close();
if (ps != null) ps.close();
if (cnx != null) cnx.close();
}
}
}
1:1
INS

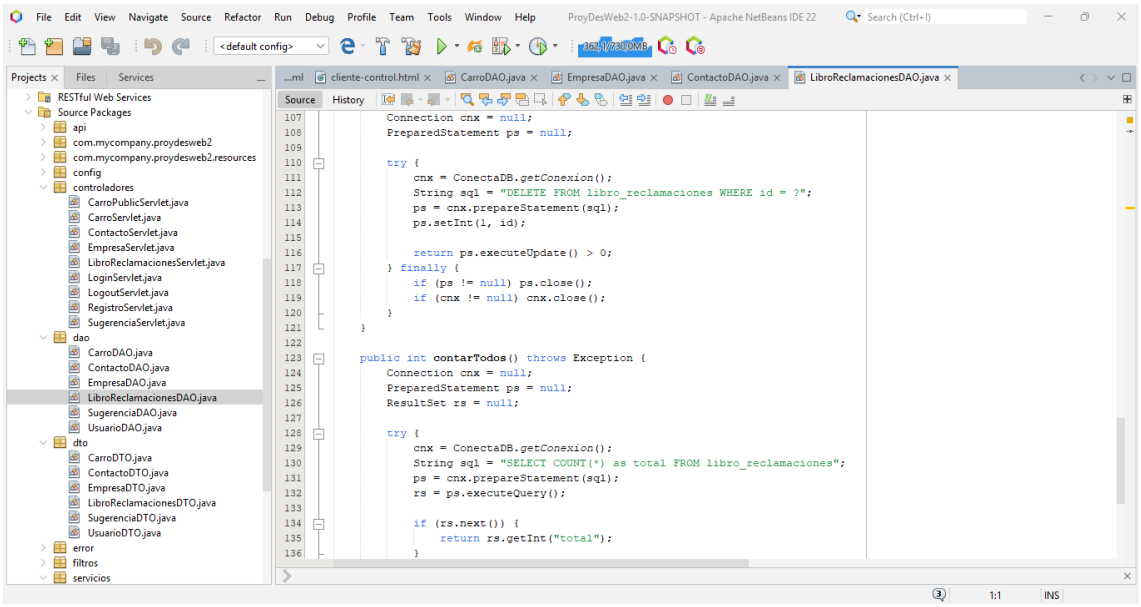
- **LibroReclamacionesDAO**

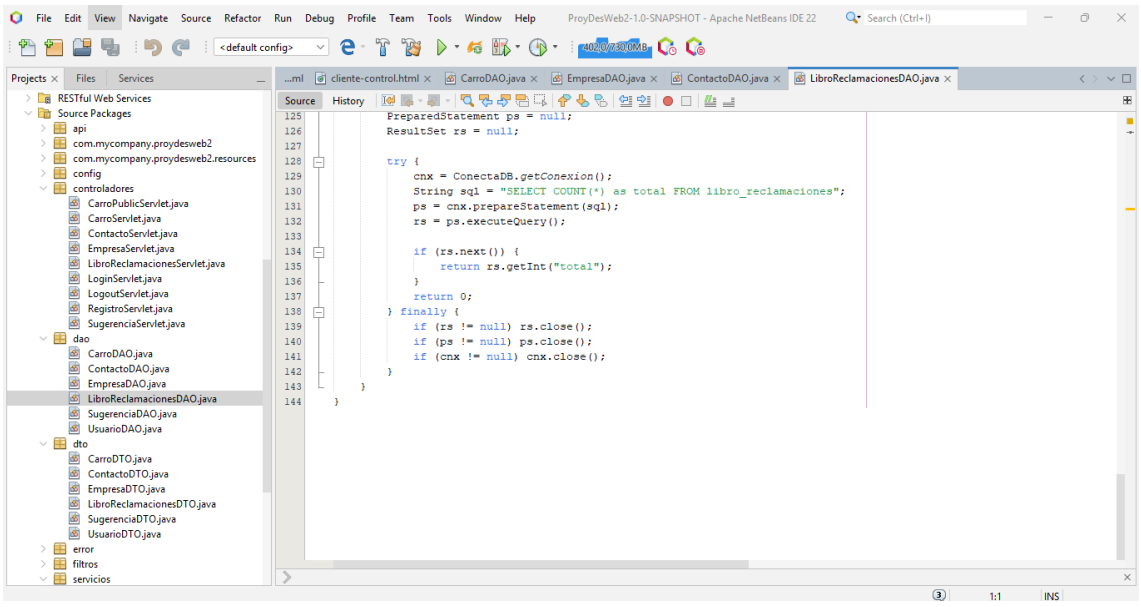



File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
cliente-control.html
CarroDAO.java
EmpresaDAO.java
ContactoDAO.java
LibroReclamacionesDAO.java
Source
History
ps.setString(9, l.getPedido());
return ps.executeUpdate() > 0;
} catch (Exception e) {
e.printStackTrace();
return false;
}
}
public List<LibroReclamacionesDTO> listarTodos() throws Exception {
Connection cnx = null;
PreparedStatement ps = null;
ResultSet rs = null;
List<LibroReclamacionesDTO> lista = new ArrayList<>();
try {
cnx = ConectaDB.getConexion();
String sql = "SELECT * FROM libro_reclamaciones ORDER BY fecha DESC";
ps = cnx.prepareStatement(sql);
rs = ps.executeQuery();
while (rs.next()) {
LibroReclamacionesDTO reclamo = new LibroReclamacionesDTO();
reclamo.setId(rs.getInt("id"));
reclamo.setNombre(rs.getString("nombre"));
reclamo.setCorreo(rs.getString("correo"));
reclamo.setTelefono(rs.getString("telefono"));
reclamo.setDireccion(rs.getString("direccion"));
1:1
INS

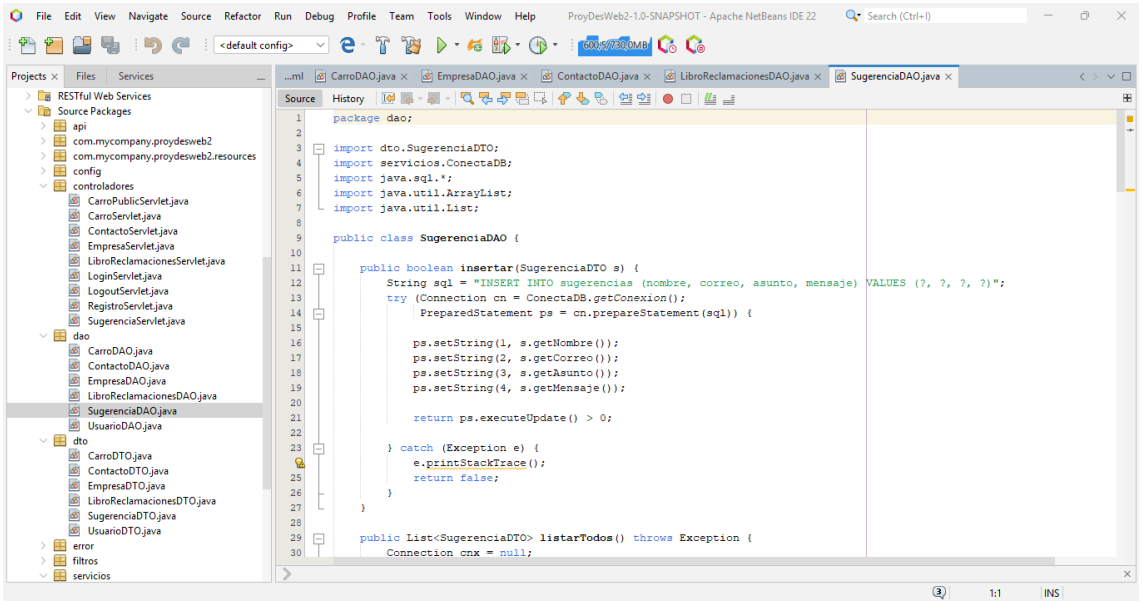

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
cliente-control.html
CarroDAO.java
EmpresaDAO.java
ContactoDAO.java
LibroReclamacionesDAO.java
Source
History
reclamo.setDireccion(rs.getString("direccion"));
reclamo.setDocumento(rs.getString("documento"));
reclamo.setTipo(rs.getString("tipo"));
reclamo.setProducto(rs.getString("producto"));
reclamo.setDescripcion(rs.getString("descripcion"));
reclamo.setPedido(rs.getString("pedido"));
reclamo.setFecha(rs.getTimestamp("fecha"));
lista.add(reclamo);
}
} finally {
if (rs != null) rs.close();
if (ps != null) ps.close();
if (cnx != null) cnx.close();
}
return lista;
}
public LibroReclamacionesDTO buscarPorId(int id) throws Exception {
Connection cnx = null;
PreparedStatement ps = null;
ResultSet rs = null;
LibroReclamacionesDTO reclamo = null;
try {
cnx = ConectaDB.getConexion();
String sql = "SELECT * FROM libro_reclamaciones WHERE id = ?";
ps = cnx.prepareStatement(sql);
ps.setInt(1, id);
1:1
INS

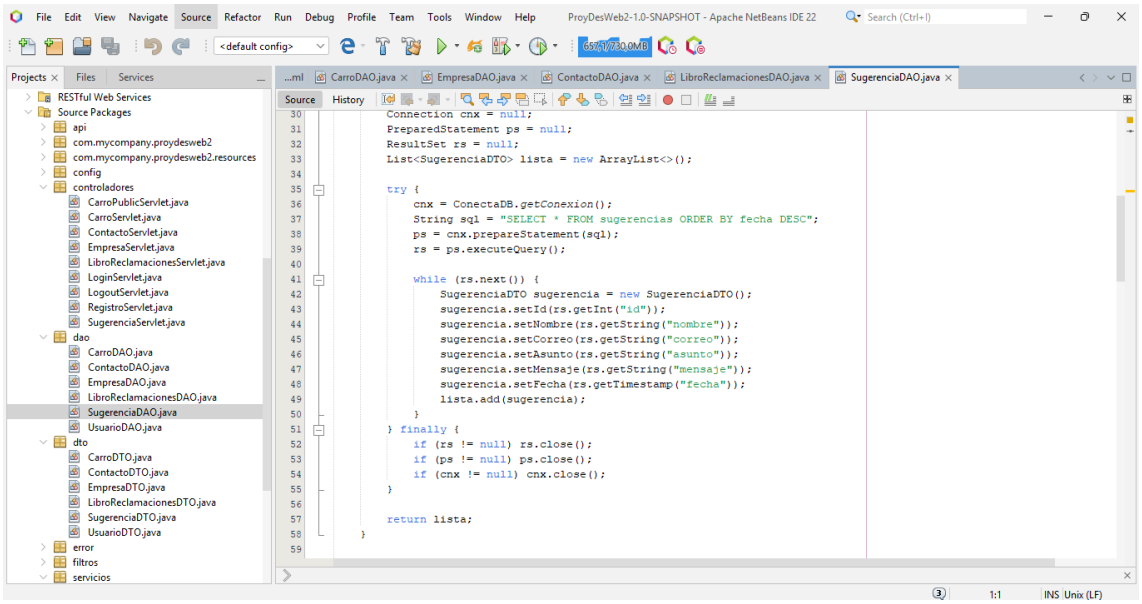

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
cliente-control.html
CarroDAO.java
EmpresaDAO.java
ContactoDAO.java
LibroReclamacionesDAO.java
Source
History
rs = ps.executeQuery();
if (rs.next()) {
reclamo = new LibroReclamacionesDTO();
reclamo.setId(rs.getInt("id"));
reclamo.setNombre(rs.getString("nombre"));
reclamo.setCorreo(rs.getString("correo"));
reclamo.setTelefono(rs.getString("telefono"));
reclamo.setDireccion(rs.getString("direccion"));
reclamo.setDocumento(rs.getString("documento"));
reclamo.setTipo(rs.getString("tipo"));
reclamo.setProducto(rs.getString("producto"));
reclamo.setDescripcion(rs.getString("descripcion"));
reclamo.setPedido(rs.getString("pedido"));
reclamo.setFecha(rs.getTimestamp("fecha"));
} finally {
if (rs != null) rs.close();
if (ps != null) ps.close();
if (cnx != null) cnx.close();
return reclamo;
public boolean eliminar(int id) throws Exception {
Connection cnx = null;
PreparedStatement ps = null;
try {
1:1
INS

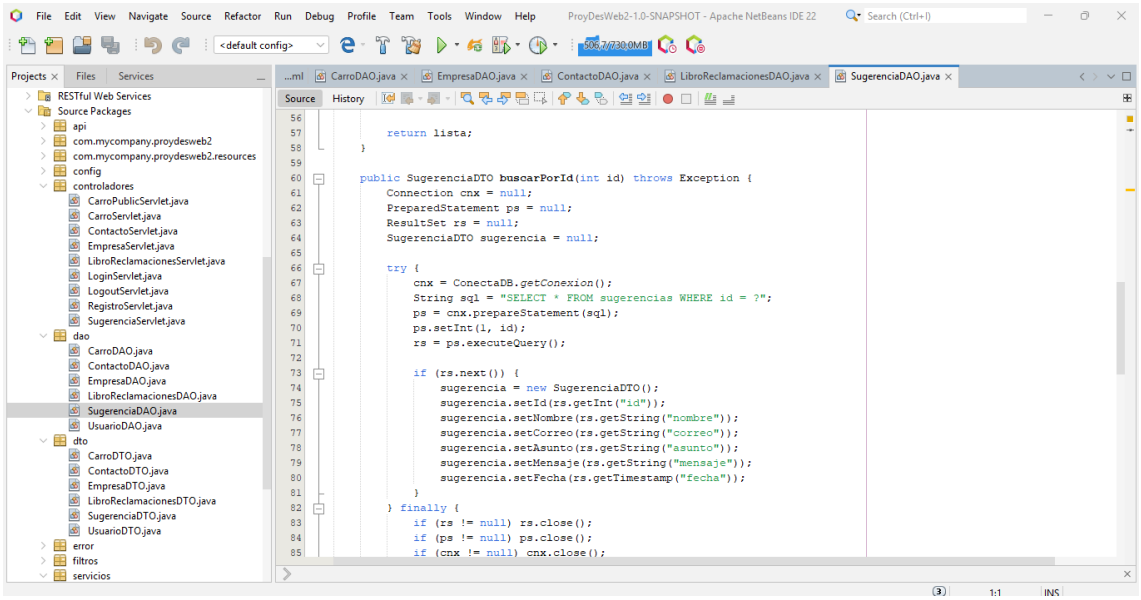

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
cliente-control.html
CarroDAO.java
EmpresaDAO.java
ContactoDAO.java
LibroReclamacionesDAO.java
Source
History
Connection cnx = null;
PreparedStatement ps = null;
try {
cnx = ConectaDB.getConexion();
String sql = "DELETE FROM libro_reclamaciones WHERE id = ?";
ps = cnx.prepareStatement(sql);
ps.setInt(1, id);
return ps.executeUpdate() > 0;
} finally {
if (ps != null) ps.close();
if (cnx != null) cnx.close();
public int contarTodos() throws Exception {
Connection cnx = null;
PreparedStatement ps = null;
ResultSet rs = null;
try {
cnx = ConectaDB.getConexion();
String sql = "SELECT COUNT(*) as total FROM libro_reclamaciones";
ps = cnx.prepareStatement(sql);
rs = ps.executeQuery();
if (rs.next()) {
return rs.getInt("total");
1:1
INS


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
cliente-control.html
CarroDAO.java
EmpresaDAO.java
ContactoDAO.java
LibroReclamacionesDAO.java
Source
History
PreparedStatement ps = null;
ResultSet rs = null;
try {
cnx = ConectaDB.getConexion();
String sql = "SELECT COUNT(*) as total FROM libro_reclamaciones";
ps = cnx.prepareStatement(sql);
rs = ps.executeQuery();
if (rs.next()) {
return rs.getInt("total");
return 0;
} finally {
if (rs != null) rs.close();
if (ps != null) ps.close();
if (cnx != null) cnx.close();
1:1
INS

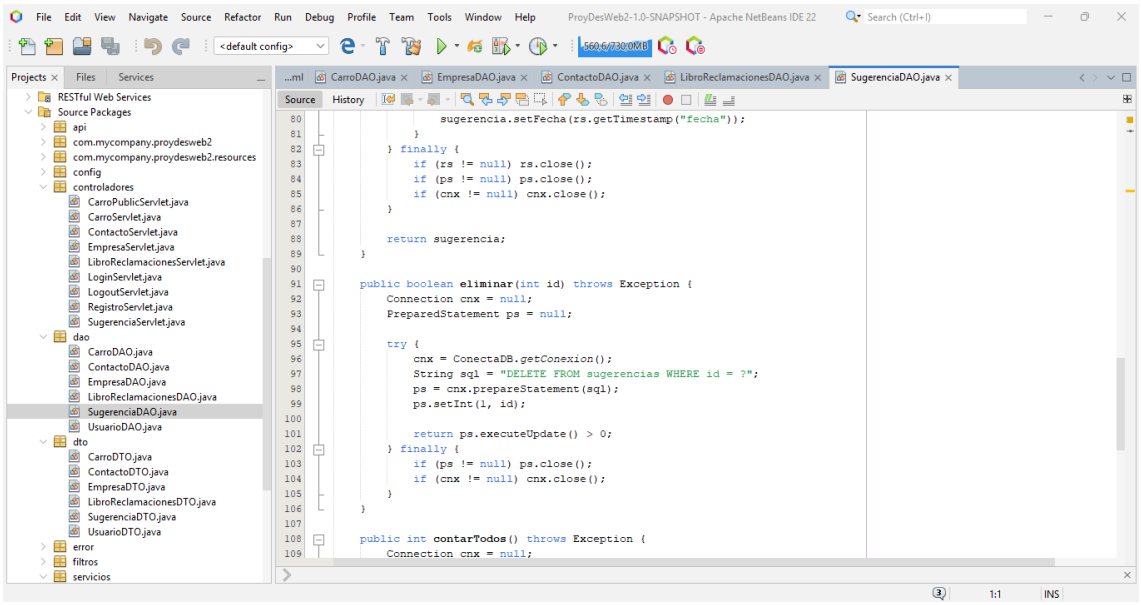
- **SugerenciaDAO**



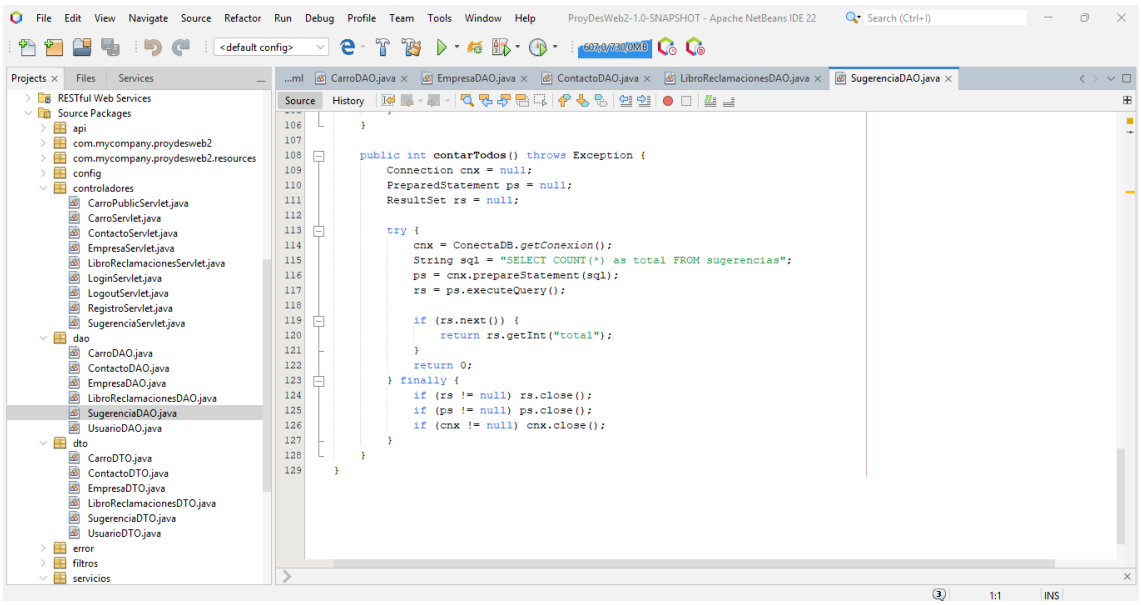




```
        return lista;
    }

    public SugerenciaDTO buscarPorId(int id) throws Exception {
        Connection cnx = null;
        PreparedStatement ps = null;
        ResultSet rs = null;
        SugerenciaDTO sugerencia = null;

        try {
            cnx = ConectaDB.getConexion();
            String sql = "SELECT * FROM sugerencias WHERE id = ?";
            ps = cnx.prepareStatement(sql);
            ps.setInt(1, id);
            rs = ps.executeQuery();

            if (rs.next()) {
                sugerencia = new SugerenciaDTO();
                sugerencia.setId(rs.getInt("id"));
                sugerencia.setNombre(rs.getString("nombre"));
                sugerencia.setCorreo(rs.getString("correo"));
                sugerencia.setAsunto(rs.getString("asunto"));
                sugerencia.setMensaje(rs.getString("mensaje"));
                sugerencia.setFecha(rs.getTimestamp("fecha"));
            }
        } finally {
            if (rs != null) rs.close();
            if (ps != null) ps.close();
            if (cnx != null) cnx.close();
```

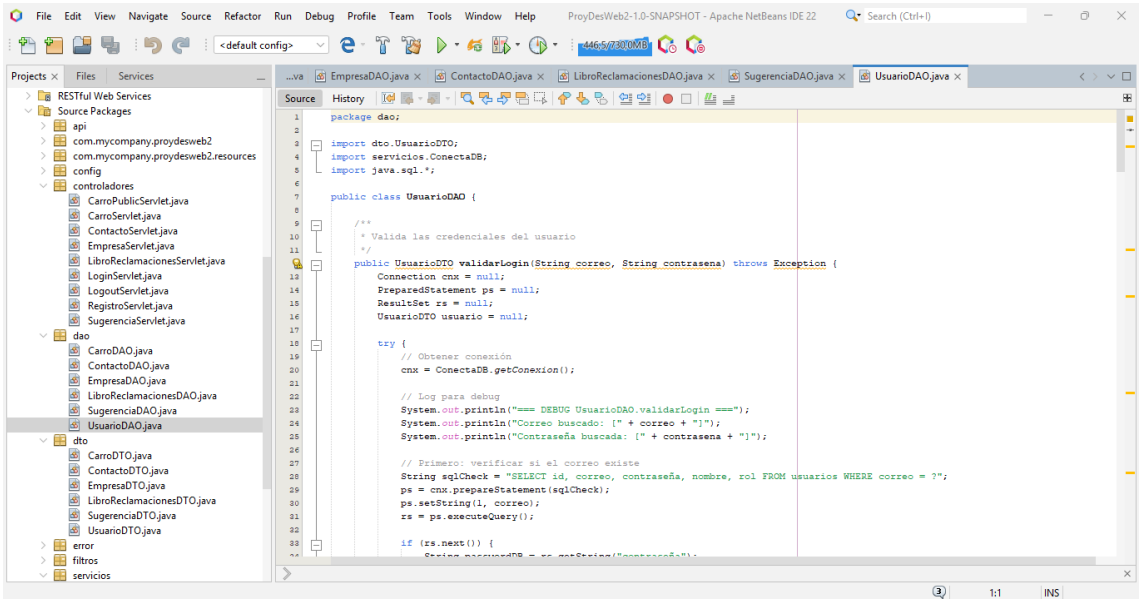


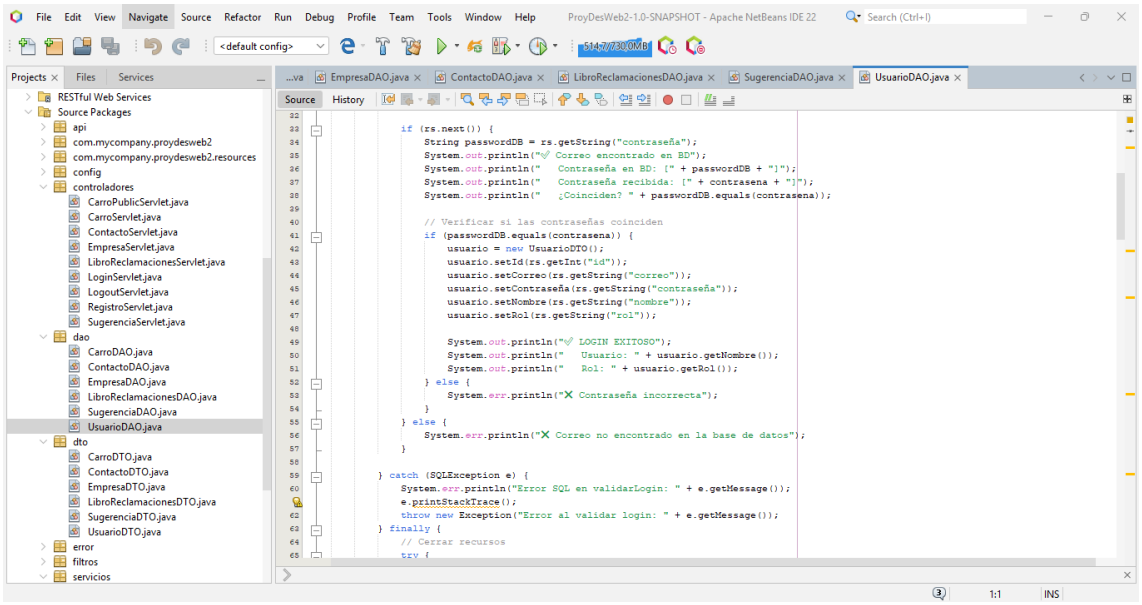
```
                sugerencia.setFecha(rs.getTimestamp("fecha"));
            }
        } finally {
            if (rs != null) rs.close();
            if (ps != null) ps.close();
            if (cnx != null) cnx.close();
        }

        return sugerencia;
    }

    public boolean eliminar(int id) throws Exception {
        Connection cnx = null;
        PreparedStatement ps = null;

        try {
            cnx = ConectaDB.getConexion();
            String sql = "DELETE FROM sugerencias WHERE id = ?";
            ps = cnx.prepareStatement(sql);
            ps.setInt(1, id);

            return ps.executeUpdate() > 0;
        } finally {
            if (ps != null) ps.close();
            if (cnx != null) cnx.close();
        }
    }

    public int contarTodos() throws Exception {
        Connection cnx = null;
```

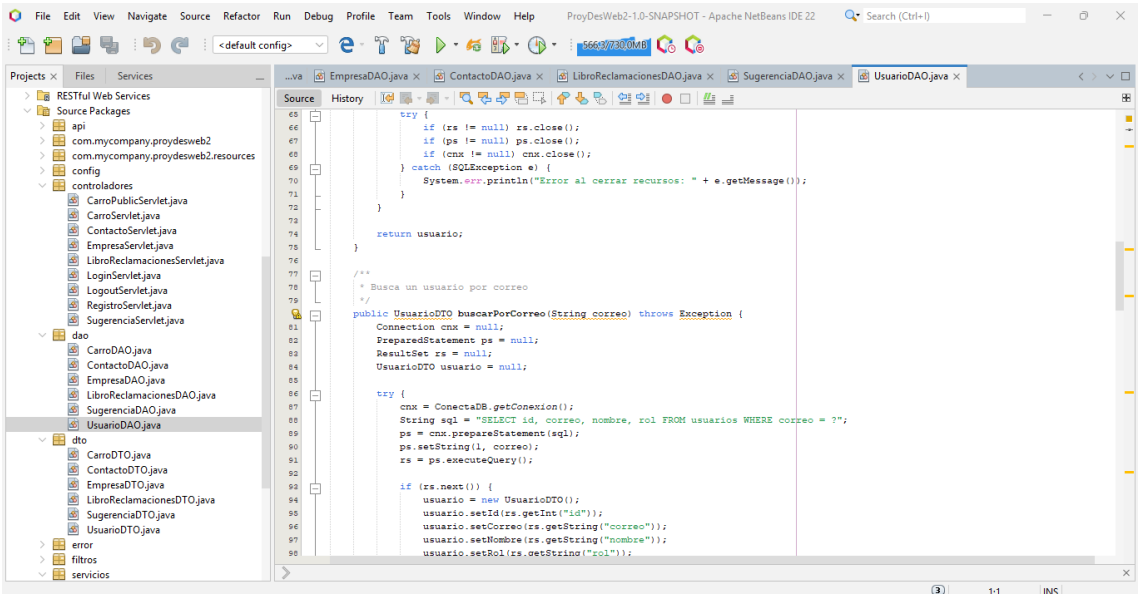


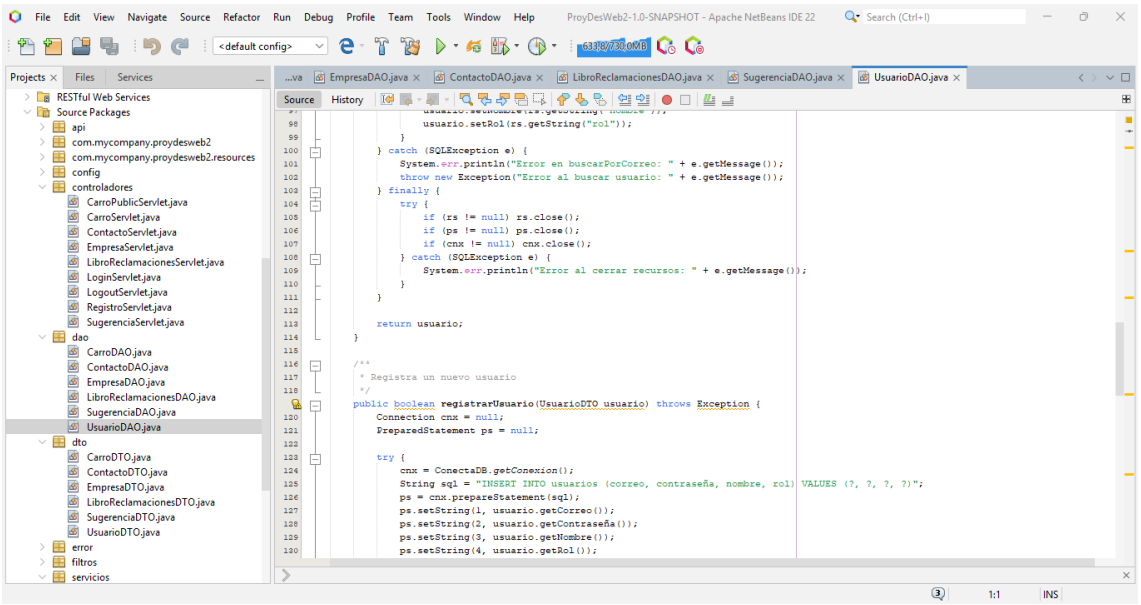
```
    }

    public int contarTodos() throws Exception {
        Connection cnx = null;
        PreparedStatement ps = null;
        ResultSet rs = null;

        try {
            cnx = ConectaDB.getConexion();
            String sql = "SELECT COUNT(*) as total FROM sugerencias";
            ps = cnx.prepareStatement(sql);
            rs = ps.executeQuery();

            if (rs.next()) {
                return rs.getInt("total");
            }
            return 0;
        } finally {
            if (rs != null) rs.close();
            if (ps != null) ps.close();
            if (cnx != null) cnx.close();
        }
    }
}
```

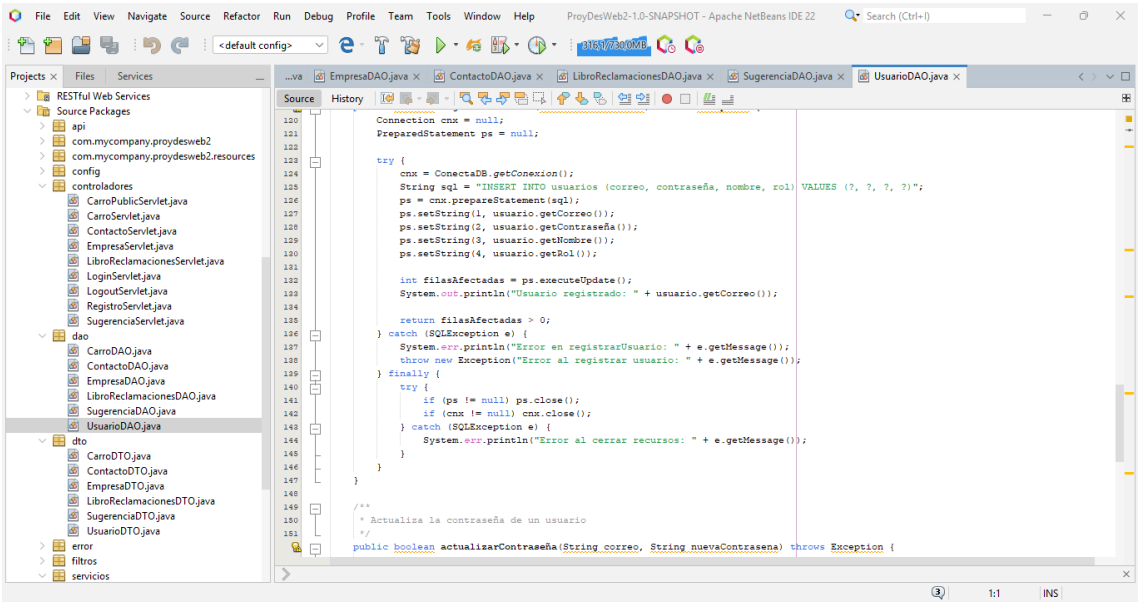
- **UsuarioDAO**

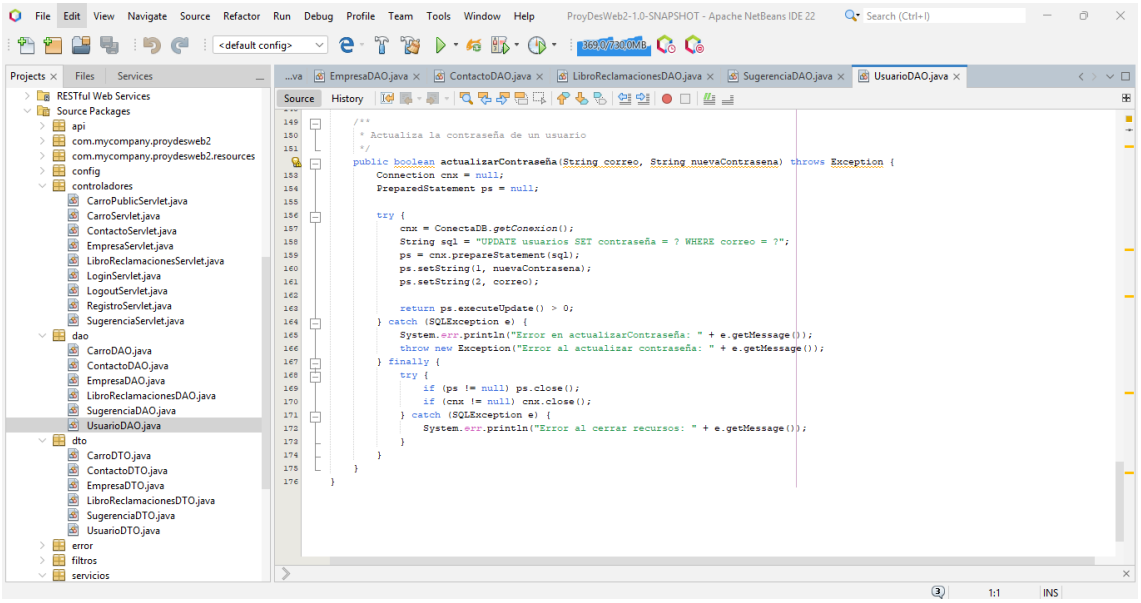

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
EmpresaDAO.java
ContactoDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
Source
History
package dao;
import dto.UsuarioDTO;
import servicios.ConectaDB;
import java.sql.*;
public class UsuarioDAO {
/**
* Valida las credenciales del usuario
*/
public UsuarioDTO validarLogin(String correo, String contrasena) throws Exception {
Connection cnx = null;
PreparedStatement ps = null;
ResultSet rs = null;
UsuarioDTO usuario = null;
try {
// Obtener conexión
cnx = ConectaDB.getConexion();
// Log para debug
System.out.println("=== DEBUG UsuarioDAO.validarLogin ===");
System.out.println("Correo buscado: [" + correo + "]");
System.out.println("Contraseña buscada: [" + contrasena + "]");
// Primero: verificar si el correo existe
String sqlCheck = "SELECT id, correo, contraseña, nombre, rol FROM usuarios WHERE correo = ?";
ps = cnx.prepareStatement(sqlCheck);
ps.setString(1, correo);
rs = ps.executeQuery();
if (rs.next()) {
1:1
INS


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
EmpresaDAO.java
ContactoDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
Source
History
if (rs.next()) {
String passwordDB = rs.getString("contraseña");
System.out.println("✓ Correo encontrado en BD");
System.out.println(" Contraseña en BD: [" + passwordDB + "]");
System.out.println(" Contraseña recibida: [" + contrasena + "]");
System.out.println(" ¿Coinciden? " + passwordDB.equals(contrasena));
// Verificar si las contraseñas coinciden
if (passwordDB.equals(contrasena)) {
usuario = new UsuarioDTO();
usuario.setId(rs.getInt("id"));
usuario.setCorreo(rs.getString("correo"));
usuario.setContraseña(rs.getString("contraseña"));
usuario.setNombre(rs.getString("nombre"));
usuario.setRol(rs.getString("rol"));
System.out.println("✓ LOGIN EXITOSO");
System.out.println(" Usuario: " + usuario.getNombre());
System.out.println(" Rol: " + usuario.getRol());
} else {
System.err.println("✗ Contraseña incorrecta");
}
} else {
System.err.println("✗ Correo no encontrado en la base de datos");
}
} catch (SQLException e) {
System.err.println("Error SQL en validarLogin: " + e.getMessage());
e.printStackTrace();
throw new Exception("Error al validar login: " + e.getMessage());
} finally {
// Cerrar recursos
try {
1:1
INS


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
EmpresaDAO.java
ContactoDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
Source
History
try {
if (rs != null) rs.close();
if (ps != null) ps.close();
if (cnx != null) cnx.close();
} catch (SQLException e) {
System.err.println("Error al cerrar recursos: " + e.getMessage());
}
}
return usuario;
}
/**
* Busca un usuario por correo
*/
public UsuarioDTO buscarPorCorreo(String correo) throws Exception {
Connection cnx = null;
PreparedStatement ps = null;
ResultSet rs = null;
UsuarioDTO usuario = null;
try {
cnx = ConectaDB.getConexion();
String sql = "SELECT id, correo, nombre, rol FROM usuarios WHERE correo = ?";
ps = cnx.prepareStatement(sql);
ps.setString(1, correo);
rs = ps.executeQuery();
if (rs.next()) {
usuario = new UsuarioDTO();
usuario.setId(rs.getInt("id"));
usuario.setCorreo(rs.getString("correo"));
usuario.setNombre(rs.getString("nombre"));
usuario.setRol(rs.getString("rol"));

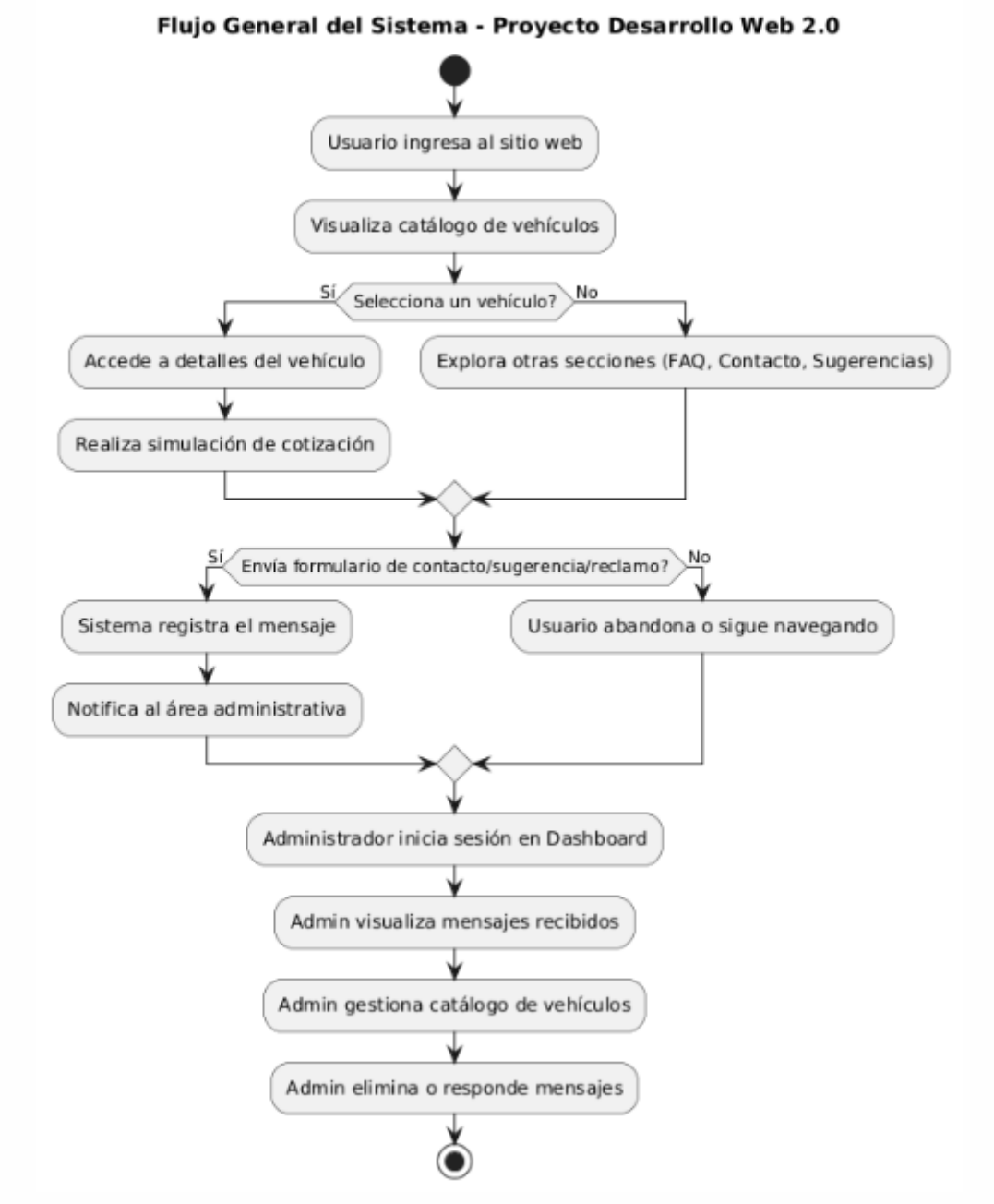

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
ProyDesWeb2-1.0-SNAPSHOT - Apache NetBeans IDE 22
Search (Ctrl+I)
<default config>
Projects
Files
Services
RESTful Web Services
Source Packages
api
com.mycompany.proydesweb2
com.mycompany.proydesweb2.resources
config
controladores
CarroPublicServlet.java
CarroServlet.java
ContactoServlet.java
EmpresaServlet.java
LibroReclamacionesServlet.java
LoginServlet.java
LogoutServlet.java
RegistroServlet.java
SugerenciaServlet.java
dao
CarroDAO.java
ContactoDAO.java
EmpresaDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
dto
CarroDTO.java
ContactoDTO.java
EmpresaDTO.java
LibroReclamacionesDTO.java
SugerenciaDTO.java
UsuarioDTO.java
error
filtros
servicios
EmpresaDAO.java
ContactoDAO.java
LibroReclamacionesDAO.java
SugerenciaDAO.java
UsuarioDAO.java
Source
History
usuario.setRol(rs.getString("rol"));
}
} catch (SQLException e) {
System.err.println("Error en buscarPorCorreo: " + e.getMessage());
throw new Exception("Error al buscar usuario: " + e.getMessage());
} finally {
try {
if (rs != null) rs.close();
if (ps != null) ps.close();
if (cnx != null) cnx.close();
} catch (SQLException e) {
System.err.println("Error al cerrar recursos: " + e.getMessage());
}
}
return usuario;
}
/**
* Registra un nuevo usuario
*/
public boolean registrarUsuario(UsuarioDTO usuario) throws Exception {
Connection cnx = null;
PreparedStatement ps = null;
try {
cnx = ConectaDB.getConexion();
String sql = "INSERT INTO usuarios (correo, contraseña, nombre, rol) VALUES (?, ?, ?, ?)";
ps = cnx.prepareStatement(sql);
ps.setString(1, usuario.getCorreo());
ps.setString(2, usuario.getContraseña());
ps.setString(3, usuario.getNombre());
ps.setString(4, usuario.getRol());



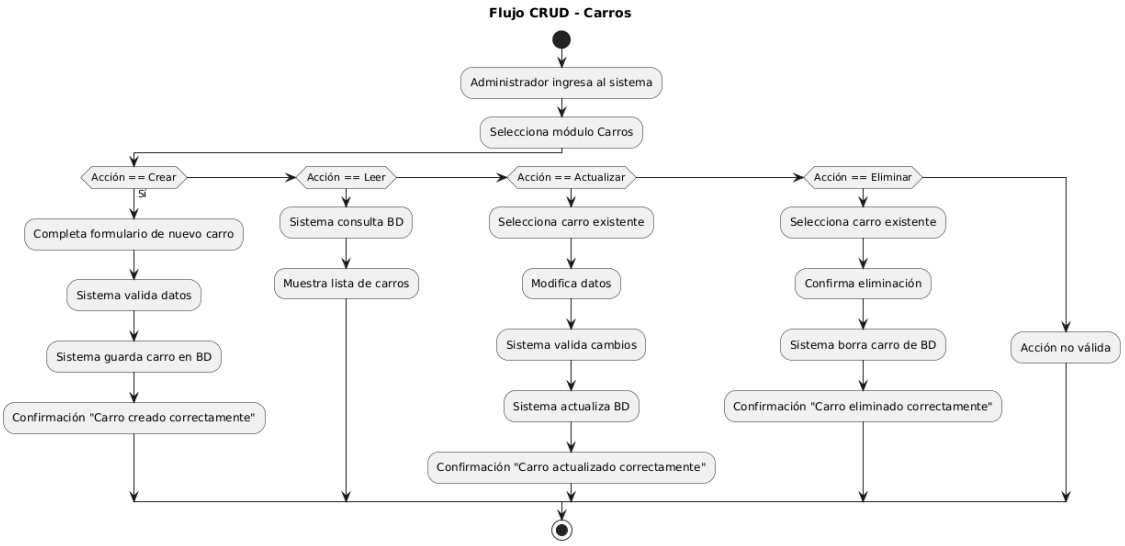


3.4 Diagramas de flujo del sistema:

- **Diagrama general del sistema:**



- **Diagrama del CRUD de carros:**



Capitulo 4:

Operaciones CRUD con Datatable y AJAX:

- **"listarAjax"**

```
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    HttpSession sesion = request.getSession(false);
    UsuarioDTO usuario = (sesion != null) ? (UsuarioDTO) sesion.getAttribute("usuario") : null;

    if (usuario == null) {
        response.sendRedirect("vistas/login.jsp");
        return;
    }

    String accion = request.getParameter("accion");
    try {
        if (accion == null) {
            List<CarroDTO> lista = dao.listarCarros();
            request.setAttribute("listaCarros", lista);
            request.getRequestDispatcher("vistas/carro/listar.jsp").forward(request, response);

        } else if ("listarAjax".equals(accion)) {
            List<CarroDTO> lista = dao.listarCarros();
            JSONArray arr = new JSONArray();
            for (CarroDTO c : lista) {
                JSONObject obj = new JSONObject();
                obj.put("id", c.getId());
                obj.put("nombre", c.getNombre());
                obj.put("descripcion", c.getDescripcion());
                obj.put("imagen", c.getImagen());
                obj.put("precio", c.getPrecio());
                arr.put(obj);
            }
            response.setContentType("application/json");
            response.setCharacterEncoding("UTF-8");
            response.getWriter().print(arr.toString());
```

- **“Insertar”**

```
if ("insertar".equals(accion)) {
    dao.insertar(c);
    ok = true;
    json.put("success", ok);
    json.put("message", "Carro registrado correctamente.");
} else if ("actualizar".equals(accion)) {
    c.setId(Integer.parseInt(request.getParameter("id")));
    dao.actualizar(c);
    ok = true;
    json.put("success", ok);
    json.put("message", "Carro actualizado correctamente.");
} else {
    json.put("success", false);
    json.put("message", "Acción inválida.");
}
```

- **“Editar”**

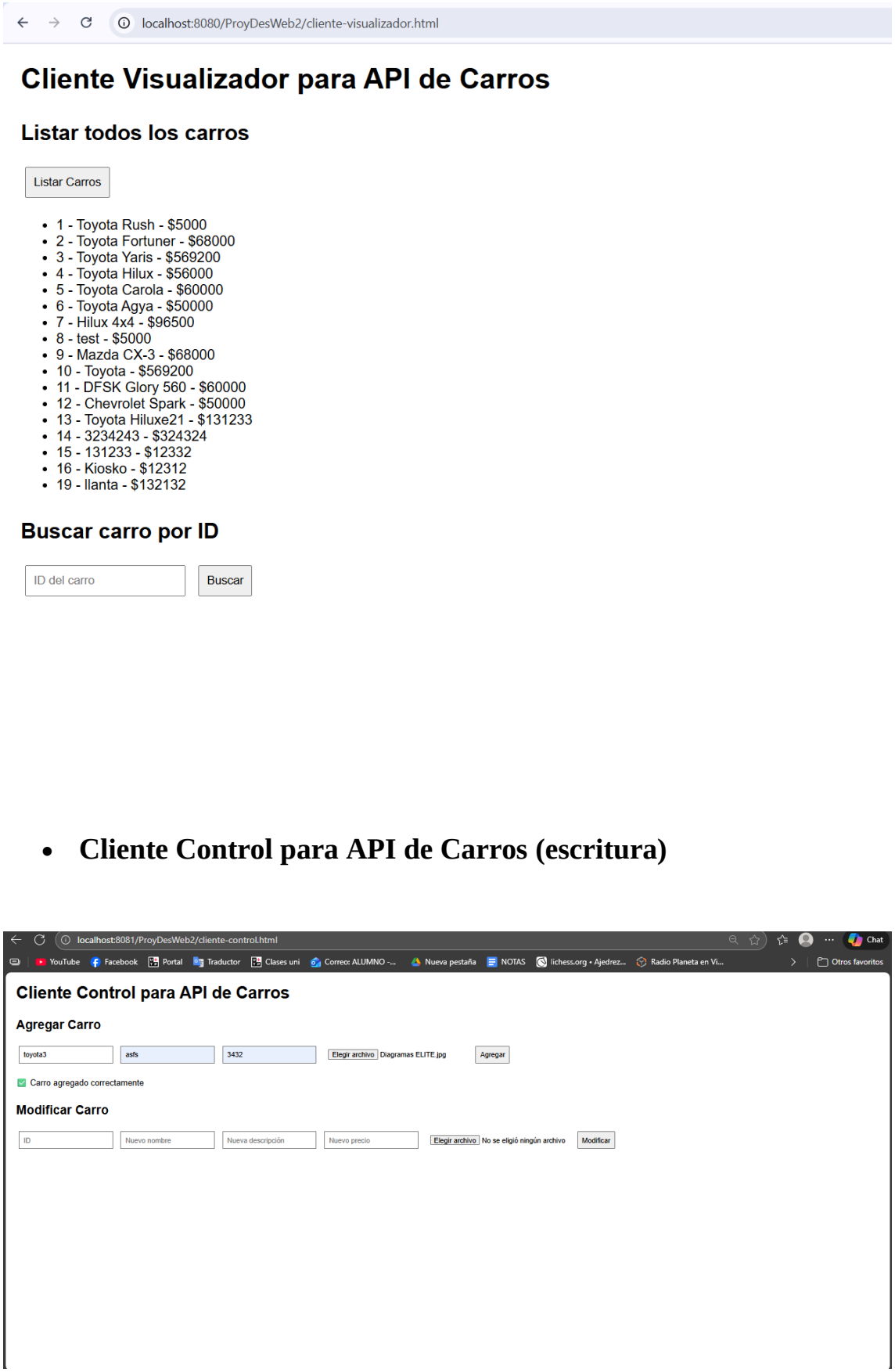
```
} else if ("editar".equals(accion)) {
    int id = Integer.parseInt(request.getParameter("id"));
    CarroDTO carro = dao.obtenerPorId(id);
    request.setAttribute("carroEditar", carro);
    request.getRequestDispatcher("vistas/carro/carroForm.jsp").forward(request, response);
```

- **“Eliminar”**

```
} else if ("eliminar".equals(accion)) {
    int id = Integer.parseInt(request.getParameter("id"));
    dao.eliminar(id);
    response.sendRedirect("carro");
}
```

Consumiendo Servicios Web

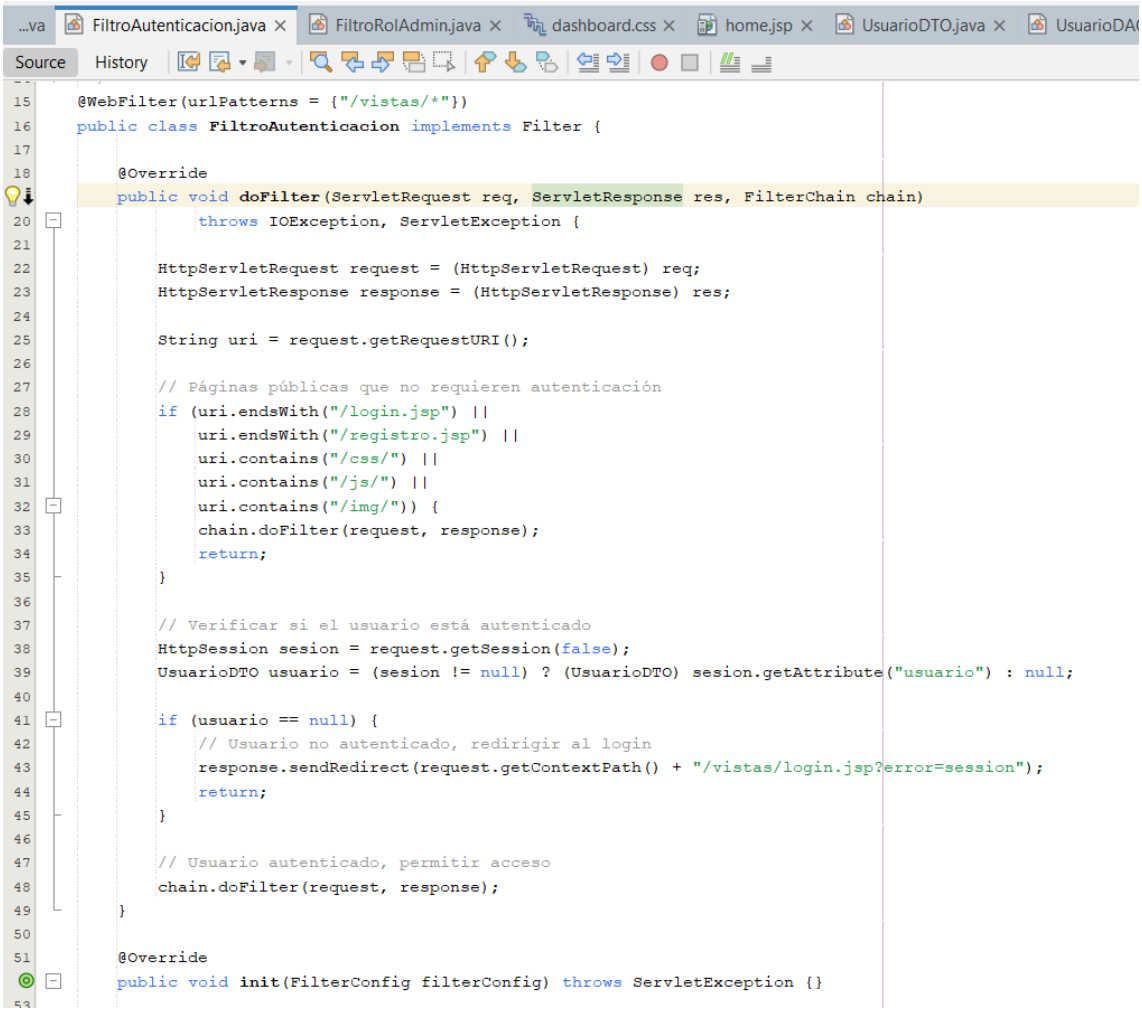
- **Cliente Visualizador (lectura)**



- **Cliente Control para API de Carros (escritura)**

Manejo de Sesiones

- ## Filtro de Autenticación




```java
@WebFilter(urlPatterns = {"/vistas/*"})
public class FiltroAutenticacion implements Filter {

    @Override
    public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain)
            throws IOException, ServletException {

        HttpServletRequest request = (HttpServletRequest) req;
        HttpServletResponse response = (HttpServletResponse) res;

        String uri = request.getRequestURI();

        // Páginas públicas que no requieren autenticación
        if (uri.endsWith("/login.jsp") ||
            uri.endsWith("/registro.jsp") ||
            uri.contains("/css/") ||
            uri.contains("/js/") ||
            uri.contains("/img/")) {
            chain.doFilter(request, response);
            return;
        }

        // Verificar si el usuario está autenticado
        HttpSession sesion = request.getSession(false);
        UsuarioDTO usuario = (sesion != null) ? (UsuarioDTO) sesion.getAttribute("usuario") : null;

        if (usuario == null) {
            // Usuario no autenticado, redirigir al login
            response.sendRedirect(request.getContextPath() + "/vistas/login.jsp?error=session");
            return;
        }

        // Usuario autenticado, permitir acceso
        chain.doFilter(request, response);
    }

    @Override
    public void init(FilterConfig filterConfig) throws ServletException {}
```

- ## Filtro de Roles

```java
public class FiltroRolAdmin implements Filter {

    @Override
    public void doFilter(ServletRequest req, ServletResponse res, FilterChain chain)
            throws IOException, ServletException {

        HttpServletRequest request = (HttpServletRequest) req;
        HttpServletResponse response = (HttpServletResponse) res;

        String uri = request.getRequestURI();
        String metodo = request.getMethod(); // Detectar si es POST

        HttpSession sesion = request.getSession(false);
        UsuarioDTO usuario = (sesion != null) ? (UsuarioDTO) sesion.getAttribute("usuario") : null;

        // Permitir acceso si es POST a /sugerencia, /libro o /contacto (envío de formularios)
        if (("POST".equalsIgnoreCase(metodo)) &&
            (uri.endsWith("/sugerencia") || uri.endsWith("/libro") || uri.endsWith("/contacto"))) {
            chain.doFilter(request, response);
            return;
        }

        // Verificar autenticación
        if (usuario == null) {
            response.sendRedirect(request.getContextPath() + "/vistas/login.jsp?error=session");
            return;
        }

        // Verificar rol
        if (!usuario.esAdmin()) {
            response.sendRedirect(request.getContextPath() + "/vistas/usuario/home.jsp?error=permission");
            return;
        }

        // Usuario es admin, permitir acceso
        chain.doFilter(request, response);
    }

    @Override
```

Glosario de términos

- MVC: Modelo-Vista-Controlador, patrón de arquitectura para aplicaciones web.
- TEA: Tasa Efectiva Anual.
- Simulación financiera: cálculo de cuotas en base a monto, plazo e interés.
- Pre-cotización: documento digital con el detalle del plan de financiamiento.

Bibliografía

- Documentación oficial de Spring Boot: https://spring.io/projects/spring-boot
- Documentación de MySQL: https://dev.mysql.com/doc

- Toyota Perú – Información de modelos: https://www.toyotaperu.com.pe