

Taller Caso de Estudio: Gestión de Alquiler de Inmuebles

Yhonier Arias

ADSO 2825192

SENA Centro de comercio y turismo armenia Quindío

2024

INTRODUCCION

El patrón MVC (Modelo-Vista-Controlador) es un enfoque arquitectónico ampliamente utilizado en el desarrollo de software para separar la lógica de negocio de la interfaz de usuario. Este patrón promueve una estructura organizada y mantenible al dividir una aplicación en tres componentes principales:

MODELO, VISTA, CONTROLADOR.

La importancia del patrón MVC radica en su capacidad para mejorar la modularidad y la reutilización del código, ya que cada componente tiene responsabilidades específicas y bien definidas. Esto facilita el desarrollo colaborativo, donde los equipos pueden trabajar de manera independiente en cada capa sin interferir en las demás. Además, MVC facilita el mantenimiento del software al permitir cambios en la interfaz de usuario o en la lógica de negocio sin afectar al resto del sistema.

En resumen, MVC no solo organiza el código de una manera que facilita su comprensión y mantenimiento, sino que también mejora la escalabilidad y la flexibilidad de las aplicaciones, haciendo que sean más robustas y adaptables a medida que evolucionan los requerimientos del negocio y del usuario.

Paso 1: Crea la base de datos en MySQL

Crea una base de datos llamada *alquilerinmueble* en MySQL, por ejemplo, el código SQL con el cual se generó la Base de Datos podría ser algo así:

```
SQL
Copiar
CREATE DATABASE IF NOT EXISTS `alquilerinmueble` DEFAULT CHARACTER SET utf8mb4
COLLATE utf8mb4_general_ci;
USE `alquilerinmueble`;
```

Paso 2: Crea la tabla alquiler

Crea una tabla llamada *alquiler* en la base de datos llamada tienda con los siguientes campos: id, nombre. Por ejemplo, el código SQL con el cual se generó la tabla podría ser algo así:

```
SQL

CREATE TABLE alquiler(
  idalquiler int(11) NOT NULL,
  idarreantarrío int(11) NOT NULL,
  idpropiedades int(11) NOT NULL,
  nombre varchar(45) NOT NULL,
  nombrePropiedad varchar(45) NOT NULL,
  tipoDePropiedad varchar(45) NOT NULL,
  precio int(11) NOT NULL,
  disponibilidad varchar(45) NOT NULL,
  PRIMARY KEY (idalquiler),
  KEY fkarrendatario_idx (idarreantarrío),
  KEY fk_propiedades_idx (idpropiedades),
  CONSTRAINT fk_propiedades FOREIGN KEY (idpropiedades) REFERENCES propiedades
(idpropiedades) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT fkarrendatario FOREIGN KEY (idarreantarrío) REFERENCES arrendatario
(identificacion) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci
```

Paso 3: Crea la tabla arrendatario

Crea una tabla llamada *productos* en la base de datos llamada tienda con los siguientes campos: *identificacion*, *nombre*, *apellido*, *contacto*, *referencias*. Por ejemplo, el código SQL con el cual se generó la tabla podría ser algo así:

```
CREATE TABLE `arrendatario` (  
  `identificacion` int(11) NOT NULL,  
  `nombre` varchar(45) NOT NULL,  
  `apellido` varchar(45) NOT NULL,  
  `contacto` int(11) NOT NULL,  
  `referencias` varchar(45) NOT NULL,  
  PRIMARY KEY (`identificacion`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_swedish_ci
```

- Crea la relación entre las tablas alquiler y arrendatario

Paso 4: Crea la tabla propiedades

Crea una tabla llamada *productos* en la base de datos llamada tienda con los siguientes campos: *idpropiedades*, *nombrePropiedad*, *direccion*, *tipoDePropiedad*, *tamano*, *numhabitaciones*, *precio*, *disponibilidad*, *referencias*. Por ejemplo, el código SQL con el cual se generó la tabla podría ser algo así:

```
CREATE TABLE `propiedades` (  
  
  `idpropiedades` int(11) NOT NULL AUTO_INCREMENT,  
  
  `nombrePropiedad` varchar(250) NOT NULL,  
  
  `direccion` varchar(200) NOT NULL,  
  
  `tipoDePropiedad` varchar(45) NOT NULL,  
  
  `tamano` varchar(80) NOT NULL,  
  
  `numhabitaciones` int(11) NOT NULL,  
  
  `precio` int(11) NOT NULL,  
  
  `disponibilidad` varchar(45) NOT NULL,  
  
  PRIMARY KEY (`idpropiedades`)  
  
) ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1  
COLLATE=latin1_swedish_ci
```

- Crea la relación entre las tablas alquiler y propiedades

Paso 5: Crea la clase para la configuración de la conexión a la base de datos

Crea un proyecto de JAVA en tu IDE de preferencia y crea un paquete nuevo llamado **CONEXION**. Luego cree una clase llamada **ConexionBD**; esta clase contiene los datos para realizar la conexión con la base de datos.

```
package CONEXION;

import javax.swing.*;
import java.sql.DriverManager;
import java.sql.SQLException;

public class ConexionDB {

    public static java.sql.Connection getConexion() {

        String url = "jdbc:mysql://localhost:3306/alquilerinmueble";

        String usuario = "root";

        String clave = "";
        java.sql.Connection conexion = null;

        try {
            System.out.println("Estableciendo conexion....");
            /*Eliminar -> Connection*/
            conexion = DriverManager.getConnection(url, usuario, clave);
            System.out.println(";Conexión exitosa!");
            //conexion.close();
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Error al conectar a la
                base de datos: " + e.getMessage());
        }
        return conexion;
    }

    public void closeConexion(java.sql.Connection conexion) {

        try {
            if (conexion != null) {
                conexion.close();
                System.out.println("Conexión cerrada.");
            }
        } catch (SQLException e) {
            System.err.println("Error al cerrar la conexión: " +
                e.getMessage());
        }
    }
}
```

Paso 6: Crea la clase para manejar la tabla alquiler de base de datos tienda

Crea un paquete nuevo llamado **MODELO**. Luego cree una clase llamada **alquiler**; esta clase representa la **tabla alquiler** en la base de datos.

```
package MODELO;

public class alquiler {

    private int idalquiler;
    private int idarrentarrio;
    private int idpropiedades;
    private String Nombre;
    private String nombrePropiedad;
    private String tipoDePropiedad;
    private int precio ;
    private String disponibilidad;

    public alquiler(int idalquiler, int idarrentarrio, int idpropiedades,
String nombre, String nombrePropiedad, String tipoDePropiedad, int
precio, String disponibilidad) {
        this.idalquiler = idalquiler;
        this.idarrentarrio = idarrentarrio;
        this.idpropiedades = idpropiedades;
        Nombre = nombre;
        this.nombrePropiedad = nombrePropiedad;
        this.tipoDePropiedad = tipoDePropiedad;
        this.precio = precio;
        this.disponibilidad = disponibilidad;
    }

    public int getIdalquiler() {
        return idalquiler;
    }

    public void setIdalquiler(int idalquiler) {
        this.idalquiler = idalquiler;
    }

    public int getIdarrentarrio() {
        return idarrentarrio;
    }

    public void setIdarrentarrio(int idarrentarrio) {
        this.idarrentarrio = idarrentarrio;
    }

    public int getIdpropiedades() {
        return idpropiedades;
    }
}
```

```
public void setIdpropiedades(int idpropiedades) {
    this.idpropiedades = idpropiedades;
}

public String getNombre() {
    return Nombre;
}

public void setNombre(String nombre) {
    Nombre = nombre;
}

public String getNombrePropiedad() {
    return nombrePropiedad;
}

public void setNombrePropiedad(String nombrePropiedad) {
    this.nombrePropiedad = nombrePropiedad;
}

public String getTipoDePropiedad() {
    return tipoDePropiedad;
}

public void setTipoDePropiedad(String tipoDePropiedad) {
    this.tipoDePropiedad = tipoDePropiedad;
}

public int getPrecio() {
    return precio;
}

public void setPrecio(int precio) {
    this.precio = precio;
}

public String getDisponibilidad() {
    return disponibilidad;
}

public void setDisponibilidad(String disponibilidad) {
    this.disponibilidad = disponibilidad;
}
}
```


Paso 6: Crea la clase para manejar la tabla arrendatario de base de datos alquilerinmueble

En el mismo paquete llamado **MODELO**. Luego cree una clase llamada **arrendatario**, esta clase representa la tabla **arrendatario** en la base de datos.

```
package MODELO;

public class arrendatario {

    private int identificacion;
    private String nombre;
    private String apellido;
    private int contacto;
    private String referencias;
    private int propiedadAlquilada;

    public arrendatario(int identificacion, String nombre, String
apellido, int contacto, String referencias ) {
        this.identificacion = identificacion;
        this.nombre = nombre;
        this.apellido = apellido;
        this.contacto = contacto;
        this.referencias = referencias;
    }

    public arrendatario(String nombre, String apellido, int contacto,
String referencias) {
        this.nombre = nombre;
        this.apellido = apellido;
        this.contacto = contacto;
        this.referencias = referencias;
    }

    public String getApellido() {
        return apellido;
    }

    public void setApellido(String apellido) {
        this.apellido = apellido;
    }

    public int getContacto() {
        return contacto;
    }
}
```

```
public void setContacto(int contacto) {
    this.contacto = contacto;
}

public int getIdentificacion() {
    return identificacion;
}

public void setIdentificacion(int identificacion) {
    this.identificacion = identificacion;
}

public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public int getPropiedadAlquilada() {
    return propiedadAlquilada;
}

public void setPropiedadAlquilada(int propiedadAlquilada) {
    this.propiedadAlquilada = propiedadAlquilada;
}

public String getReferencias() {
    return referencias;
}

public void setReferencias(String referencias) {
    this.referencias = referencias;
}
}
```

Paso 7: Crea la clase para manejar la tabla propiedades de base de datos alquilerinmueble

En el mismo paquete llamado **MODELO**. Luego cree una clase llamada *propiedades*, esta clase representa la tabla *propiedades* en la base de datos.

```
package MODELO;

public class Propiedades {

    private int idpropiedades;
    private String nombrePropiedad;
    private String direccion;
    private String tipoDePropiedad;
    private String tamano;
    private int numhabitaciones;
    private int precio;
    private String disponibilidad;

    public Propiedades(int idpropiedades, String nombrePropiedad, String
direccion, String tipoDePropiedad, String tamano, int numhabitaciones,
int precio, String disponibilidad) {
        this.idpropiedades = idpropiedades;
        this.nombrePropiedad = nombrePropiedad;
        this.direccion = direccion;
        this.tipoDePropiedad = tipoDePropiedad;
        this.tamano = tamano;
        this.numhabitaciones = numhabitaciones;
        this.precio = precio;
        this.disponibilidad = disponibilidad;
    }

    public Propiedades(String nombrePropiedad, String direccion, String
tipoDePropiedad, String tamano, int numhabitaciones, int precio, String
disponibilidad) {
        this.nombrePropiedad = nombrePropiedad;
        this.direccion = direccion;
        this.tipoDePropiedad = tipoDePropiedad;
        this.tamano = tamano;
        this.numhabitaciones = numhabitaciones;
        this.precio = precio;
        this.disponibilidad = disponibilidad;
    }
}
```

```
public int getIdpropiedades() {
    return idpropiedades;
}

public void setIdpropiedades(int idpropiedades) {
    this.idpropiedades = idpropiedades;
}

public String getNombrePropiedad() {
    return nombrePropiedad;
}

public void setNombrePropiedad(String nombrePropiedad) {
    this.nombrePropiedad = nombrePropiedad;
}

public String getDireccion() {
    return direccion;
}

public void setDireccion(String direccion) {
    this.direccion = direccion;
}

public String getTipoDePropiedad() {
    return tipoDePropiedad;
}

public void setTipoDePropiedad(String tipoDePropiedad) {
    this.tipoDePropiedad = tipoDePropiedad;
}

public String getTamano() {
    return tamano;
}

public void setTamano(String tamano) {
    this.tamano = tamano;
}

public int getNumhabitaciones() {
    return numhabitaciones;
}

public void setNumhabitaciones(int numhabitaciones) {
    this.numhabitaciones = numhabitaciones;
}

public int getPrecio() {
    return precio;
}

public void setPrecio(int precio) {
    this.precio = precio;
}
```

```

public String getDisponibilidad() {
    return disponibilidad;
}

public void setDisponibilidad(String disponibilidad) {
    this.disponibilidad = disponibilidad;
}
}

```

Paso 9: Crea los metodos para usar las consultas SQL con las tablas en la tabla

Crea un paquete llamado **LOGICAMODELO**. Luego cree una clase llamada **alquiler**. Esta clase contendrá las consultas SQL que se enviarán a la base de datos para gestionar la **tabla alquiler**.

```

package LOGICAMODELO;

import CONEXION.ConexionDB;
import MODELO.alquiler;
import MODELO.arrendatario;

import javax.swing.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class logicaAlquiler {
    private final Connection conexion;
    private final ConexionDB conexionDB;

    public logicaAlquiler() {
        conexionDB = new ConexionDB();
        conexion = ConexionDB.getConnection();
    }

    public void registrarAlquiler (alquiler cliente) {
        String sql = "INSERT INTO arrendatario
(idalquiler, idarrendatario, idpropiedades, nombre, nombrePropiedad, tipoDePro
piedad, precio, disponibilidad) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";
        try (PreparedStatement preparedStatement =
conexion.prepareStatement(sql)) {

```

```
        preparedStatement.setInt(1, cliente.getIdalquiler());
        preparedStatement.setInt(2, cliente.getIdarrentarrio());
        preparedStatement.setInt(3, cliente.getIdpropiedades());
        preparedStatement.setString(4, cliente.getNombre());
        preparedStatement.setString(5, cliente.getNombrePropiedad());
        preparedStatement.setString(6, cliente.getTipoDePropiedad());
        preparedStatement.setInt(7, cliente.getPrecio());
        preparedStatement.setString(8, cliente.getDisponibilidad());
        preparedStatement.executeUpdate();

        JOptionPane.showMessageDialog(null, "Registro Etixoso");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error: " +
e.getMessage());
    }
    conexionDB.closeConexion(conexion);
}
}
```

En el mismo paquete llamado **LOGICAMODELO**, cree una clase llamada **logicaarrendatario**. Esta clase contendrá las consultas SQL que se enviarán a la base de datos para gestionar la **tabla arrendatario**.

```
package LOGICAMODELO;

import CONEXION.ConexionDB;
import MODELO.arrendatario;

import javax.swing.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

/**
public class logicaArrendatario {

    private final Connection conexion;
    private final ConexionDB conexionDB;

    public logicaArrendatario() {
        conexionDB = new ConexionDB();
        conexion = ConexionDB.getConnection();
    }

    public void registrarArrendatario(arrendatario cliente) {
        String sql = "INSERT INTO arrendatario
(identificacion,nombre,apellido, contacto, referencias) VALUES (?, ?,
?, ?, ?)";
        try (PreparedStatement preparedStatement =
conexion.prepareStatement(sql)) {

            preparedStatement.setInt(1, cliente.getIdentificacion());
            preparedStatement.setString(2, cliente.getNombre());
            preparedStatement.setString(3, cliente.getApellido());
            preparedStatement.setInt(4, cliente.getContacto());
            preparedStatement.setString(5, cliente.getReferencias());
            preparedStatement.executeUpdate();
            JOptionPane.showMessageDialog(null,"Registro Exitoso");
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null,"Error: " +
e.getMessage());
        }
        conexionDB.closeConexion(conexion);
    }

    public void actualizarArrendatario(arrendatario cliente, int
identificacion) {
        String sql = "UPDATE arrendatario SET nombre = ?, apellido = ?,
contacto = ?, referencias = ? WHERE identificacion = "+identificacion+"
";

        try (PreparedStatement preparedStatement =
conexion.prepareStatement(sql)) {
```

```

        preparedStatement.setString(1, cliente.getNombre());
        preparedStatement.setString(2, cliente.getApellido());
        preparedStatement.setInt(3, cliente.getContacto());
        preparedStatement.setString(4, cliente.getReferencias());
        preparedStatement.executeUpdate();
        JOptionPane.showMessageDialog(null, "! Actualizado !");
    } catch (SQLException e) {
        System.out.println("error: "+e.getMessage());
        JOptionPane.showMessageDialog(null, "Error:
"+e.getMessage());
    }
    conexionDB.closeConexion(conexion);
}

    public void eliminarArrendatario(int id) {
        String sql = "DELETE FROM arrendatario WHERE identificacion =
"+id+" ";
        try (PreparedStatement preparedStatement =
conexion.prepareStatement(sql)) {
            preparedStatement.executeUpdate();
            JOptionPane.showMessageDialog(null, "ELIMINADO CON EXITO.");
        } catch (SQLException e) {
            System.out.println("Error al eliminar los datos: " +
e.getMessage());
        }
        conexionDB.closeConexion(conexion);
    }
}
}

```

En el mismo paquete llamado **LOGICAMODELO**, cree una clase llamada **logicapropiedades**. Esta clase contendrá las consultas SQL que se enviarán a la base de datos para gestionar la **tabla propiedades**.

```

package LOGICAMODELO;

import CONEXION.ConexionDB;
import MODELO.Propiedades;

import javax.swing.*;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class logicaPropiedades {
    private final Connection conexion;
    private final ConexionDB conexionDB;

    public logicaPropiedades() {

```



```

        conexionDB = new ConexionDB();
        conexion = ConexionDB.getConexion();
    }

    public void registrarPropiedades(Propiedades valor) {
        String sql = "INSERT INTO propiedades
(nombrePropiedad,direccion,tipoDePropiedad,tamano,numhabitaciones,precio,
disponibilidad) VALUES (?, ?, ?, ?, ?, ?, ?)";

        try (PreparedStatement preparedStatement =
conexion.prepareStatement(sql)) {

            preparedStatement.setString(1, valor.getNombrePropiedad());
            preparedStatement.setString(2, valor.getDireccion());
            preparedStatement.setString(3, valor.getTipoDePropiedad());
            preparedStatement.setString(4, valor.getTamano());
            preparedStatement.setInt(5, valor.getNumhabitaciones());
            preparedStatement.setInt(6, valor.getPrecio());
            preparedStatement.setString(7, valor.getDisponibilidad());

            preparedStatement.executeUpdate();

            JOptionPane.showMessageDialog(null, "Registro Exitoso");

        } catch (SQLException e) {
            System.out.println("error : "+e.getMessage());
        }
        conexionDB.closeConexion(conexion);
    }

    public void actualizarPropiedad(Propiedades propiedad, int id) {
        String sql = "UPDATE propiedades SET nombrePropiedad = ? ,
direccion = ?, tipoDePropiedad = ?, " +
            " tamano = ?, numhabitaciones =?, precio =?,
disponibilidad = ? WHERE idpropiedades = "+id+" ";

        try (PreparedStatement preparedStatement =
conexion.prepareStatement(sql)) {

preparedStatement.setString(1, propiedad.getNombrePropiedad());
preparedStatement.setString(2, propiedad.getDireccion());

preparedStatement.setString(3, propiedad.getTipoDePropiedad());
preparedStatement.setString(4, propiedad.getTamano());
preparedStatement.setInt(5, propiedad.getNumhabitaciones());
preparedStatement.setInt(6, propiedad.getPrecio());
preparedStatement.setString(7, propiedad.getDisponibilidad());

            preparedStatement.executeUpdate();
            JOptionPane.showMessageDialog(null, "! Actualizado !");
        } catch (SQLException e) {
            System.out.println("error: "+e.getMessage());
        }
        // JOptionPane.showMessageDialog(null, "Error:
"+e.getMessage());
    }
        conexionDB.closeConexion(conexion);
    }
}

```

```

        public void eliminarPropiedad(int id) {
            String sql = "DELETE FROM propiedades WHERE idpropiedades =
"+id+" ";
            try (PreparedStatement preparedStatement =
conexion.prepareStatement(sql)) {
                preparedStatement.executeUpdate();
                JOptionPane.showMessageDialog(null, "ELIMINADO CON EXITO.");
            } catch (SQLException e) {
                System.out.println("Error al eliminar los datos: " +
e.getMessage());
            } conexionDB.closeConexion(conexion);
        }
    }
}

```

En el mismo paquete llamado **LOGICAMODELO**, cree una clase llamada **tablas**. Esta clase contendrá las consultas SQL ejemplo: “select * from alquiler;” que se enviarán a la base de datos para gestionar los metodos consultar;

```

package LOGICAMODELO;

import CONEXION.ConexionDB;

import java.sql.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

public class tablas {
    private final Connection conexion;
    private final ConexionDB conexionDB;

    public tablas() {
        conexionDB = new ConexionDB();
        conexion = ConexionDB.getConexion();
    }

    public void MostrarArrendatario(JTable tabla) {
        DefaultTableModel modelo = new DefaultTableModel();
        modelo.addColumn("Identificacion");
        modelo.addColumn("Nombre");
        modelo.addColumn("Apellido");
        modelo.addColumn("Contacto");
        modelo.addColumn("Referencias");

        String sql = "SELECT * FROM arrendatario";
        try (PreparedStatement preparedStatement =
conexion.prepareStatement(sql)) {
            ResultSet resultSet = preparedStatement.executeQuery();

```

```

        while (resultSet.next()) {
            Object[] lista = {
                resultSet.getInt(1),
                resultSet.getString(2),
                resultSet.getString(3),
                resultSet.getInt(4),
                resultSet.getString(5),
            };
            modelo.addRow(lista);
        }
        tabla.setModel(modelo);
    } catch (SQLException e) {
        System.out.println("Error " + e.getMessage());
    }
}

public void BuscarArrendatario(String valor, JTable tabla) {
    DefaultTableModel modelo = new DefaultTableModel();
    modelo.addColumn("Identificacion");
    modelo.addColumn("Nombre");
    modelo.addColumn("Apellido");
    modelo.addColumn("Contacto");
    modelo.addColumn("Referencias");

    String sql = "select * from arrendatario where concat(nombre,identificacion) like ?";

    try (PreparedStatement statement =
conexion.prepareStatement(sql)) {
        statement.setString(1, "%" + valor + "%");
        ResultSet resultSet = statement.executeQuery();
        while (resultSet.next()) {
            Object[] lista = {
                resultSet.getInt(1),
                resultSet.getString(2),
                resultSet.getString(3),
                resultSet.getInt(4),
                resultSet.getString(5),
            };

            modelo.addRow(lista);
        }
        tabla.setModel(modelo);
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error " +
e.getMessage());
    }
}

public void MostrarPropiedades(JTable tabla) {
    DefaultTableModel modelo = new DefaultTableModel();
    modelo.addColumn("Id");
    modelo.addColumn("Nombre Propiedad");
    modelo.addColumn("Direccion");
    modelo.addColumn("Tipo Propiedad");
    modelo.addColumn("Tamaño");
}

```

```

        modelo.addColumn("Habitaciones");
        modelo.addColumn("Precio");
        modelo.addColumn("Disponibilidad");

        String sql = "SELECT * FROM propiedades";
        try (PreparedStatement preparedStatement =
conexion.prepareStatement(sql)) {
            ResultSet resultSet = preparedStatement.executeQuery();
            while (resultSet.next()) {
                Object[] lista = {
                    resultSet.getInt(1),
                    resultSet.getString(2),
                    resultSet.getString(3),
                    resultSet.getString(4),
                    resultSet.getString(5),
                    resultSet.getInt(6),
                    resultSet.getInt(7),
                    resultSet.getString(8)
                };
                modelo.addRow(lista);
            }
            tabla.setModel(modelo);
        } catch (SQLException e) {
            System.out.println("Error " + e.getMessage());
        }
    }

    public void BuscarPropiedad(String valor, JTable tabla) {
        DefaultTableModel modelo = new DefaultTableModel();
        modelo.addColumn("Id");
        modelo.addColumn("Nombre Propiedad");
        modelo.addColumn("Direccion");
        modelo.addColumn("Tipo Propiedad");
        modelo.addColumn("Tamaño");
        modelo.addColumn("Habitaciones");
        modelo.addColumn("Precio");
        modelo.addColumn("Disponibilidad");
        String sql = "select * from propiedades where concat(
direccion, disponibilidad) like ?";

        try (PreparedStatement statement =
conexion.prepareStatement(sql)) {
            statement.setString(1, "%" + valor + "%");
            ResultSet resultSet = statement.executeQuery();
            while (resultSet.next()) {
                Object[] lista = {
                    resultSet.getInt(1),
                    resultSet.getString(2),
                    resultSet.getString(3),
                    resultSet.getString(4),
                    resultSet.getString(5),
                    resultSet.getInt(6),
                    resultSet.getInt(7),
                    resultSet.getString(8)
                };
                modelo.addRow(lista);
            }
        }
    }

```

```

        tabla.setModel(modelo);
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "Error " +
e.getMessage());
    }
}

public void MostrarAlquiler(JTable tabla) {
    DefaultTableModel modelo = new DefaultTableModel();
    modelo.addColumn("Id");
    modelo.addColumn("IDENTIFICACION");
    modelo.addColumn("NOMBRE");
    modelo.addColumn("NOMBRE PROPIEDAD");
    modelo.addColumn("TIPO DE PROPIEDAD");
    modelo.addColumn("PRECIO");
    modelo.addColumn("ESTADO");

    String sql = "SELECT * FROM alquilerinmueble.vistaalquiler";
//    SELECT * FROM alquilerinmueble.vista_alquiler
    try (PreparedStatement preparedStatement =
conexion.prepareStatement(sql)) {
        ResultSet resultSet = preparedStatement.executeQuery();
        while (resultSet.next()) {
            Object[] lista = {
                resultSet.getInt(1),
                resultSet.getInt(2),
                resultSet.getString(3),
                resultSet.getString(4),
                resultSet.getString(5),
                resultSet.getString(6),
                resultSet.getString(7)
            };
            modelo.addRow(lista);
        }
        tabla.setModel(modelo);
    } catch (SQLException e) {
        System.out.println("Error " + e.getMessage());
    }
}

public void BuscarAlquiler(String valor, JTable tabla) {
    DefaultTableModel modelo = new DefaultTableModel();
    modelo.addColumn("Id");
    modelo.addColumn("DNI");
    modelo.addColumn("NOMBRE");
    modelo.addColumn("PROPIEDAD");
    modelo.addColumn("TIPO DE PROPIEDAD");
    modelo.addColumn("PRECIO");
    modelo.addColumn("ESTADO");
    String sql = "select * from propiedades where concat( DNI,
propiedad) like ?";

    try (PreparedStatement statement =
conexion.prepareStatement(sql)) {
        statement.setString(1, "%" + valor + "%");
        ResultSet resultSet = statement.executeQuery();
        while (resultSet.next()) {

```

```

        Object[] lista = {
            resultSet.getInt(1),
            resultSet.getInt(2),
            resultSet.getString(3),
            resultSet.getString(4),
            resultSet.getString(5),
            resultSet.getString(6),
            resultSet.getString(7)
        };
        modelo.addRow(lista);
    }
    tabla.setModel(modelo);
} catch (SQLException e) {
    JOptionPane.showMessageDialog(null, "Error " +
e.getMessage());
}
}
}

```

Paso 10 : Crea las clases que controlan la llamada a los métodos que gestionaran las tablas en la base de datos

Crea un paquete llamado **CONTROLADOR** Luego cree una clase llamada **controladoralquiler**. Esta clase contendrá los métodos que controlan lo que se enviará a la tabla *alquiler*

```

package CONTROLADOR;

import Vistas.vistaAlquiler;
import LOGICAMODELO.tablas;

import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import javax.swing.table.TableModel;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

```

```

public class controladorAlquiler implements ActionListener,
ListSelectionListener, DocumentListener{

    vistaAlquiler vistaAlquiler;
    public static tablas tablas;;

    public controladorAlquiler(Vistas.vistaAlquiler vistaAlquiler) {
        this.vistaAlquiler = vistaAlquiler;
    }
    //
    this.vistaAlquiler.txtBuscar.getDocument().addDocumentListener(this);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource()== vistaAlquiler.btnRegistrarAlquiler){

        }

    }

    //
    public void buscar(){
    //
        String valor = vistaAlquiler.txtBuscar.getText();
    //
        tablas= new tablas();
    //
        tablas.BuscarAlquiler(valor,vistaAlquiler.tablaAlquiler);
    //
    }

    @Override
    public void insertUpdate(DocumentEvent e) {

    }

    @Override
    public void removeUpdate(DocumentEvent e) {

    }

    @Override
    public void changedUpdate(DocumentEvent e) {

    }

    @Override
    public void valueChanged(ListSelectionEvent e) {

        if (!e.getValueIsAdjusting()){
            e.getSource();
            vistaAlquiler.tablaAlquiler.getSelectionModel();
            int filasObtenidad
=vistaAlquiler.tablaAlquiler.getSelectedRow();

            if (filasObtenidad >= 0){
                TableModel modelo =

```

```

vistaAlquiler.tablaAlquiler.getModel();

        Object idalquiler = modelo.getValueAt(filasObtenidad,0);
        Object idarrendatarrio =
modelo.getValueAt(filasObtenidad,1);
        Object idpropiedades =
modelo.getValueAt(filasObtenidad,2);
        Object nombre = modelo.getValueAt(filasObtenidad,3);
        Object nombrePropiedad =
modelo.getValueAt(filasObtenidad,4);
        Object tipoDePropiedad =
modelo.getValueAt(filasObtenidad,5);
        Object precio = modelo.getValueAt(filasObtenidad,6);
        Object disponibilidad = modelo.getValueAt(filasObtenidad,
7);

vistaAlquiler.txtIDalquiler.setText(idalquiler.toString());

vistaAlquiler.txtIDarrendatario.setText(idarrendatarrio.toString());

vistaAlquiler.txtxiIDpropiedades.setText(idpropiedades.toString());
        vistaAlquiler.txtNombre.setText(nombre.toString());

    }

}

}

}

```

En el mismo paquete llamado **CONTROLADOR**, cree una clase llamada **controladorarrendatario**. Esta clase contendrá los métodos que controlan lo que se enviará a la tabla **arrendatario**.

```

package CONTROLADOR;

import MODELO.arrendatario;
import Vistas.vistaArrendatarios;
import LOGICAMODELO.logicaArrendatario;
import LOGICAMODELO.tablas;

import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import javax.swing.table.TableModel;

```



```

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class controladorArrendatario implements ActionListener ,
ListSelectionListener, DocumentListener {

    vistaArrendatarios vistaArrendatarios;
    public static tablas tablas;

    public controladorArrendatario(Vistas.vistaArrendatarios
vistaArrendatarios) {

        this.vistaArrendatarios = vistaArrendatarios;
        this.vistaArrendatarios.btnRegistrarA.addActionListener(this);
        this.vistaArrendatarios.btnActualizarA.addActionListener(this);
        this.vistaArrendatarios.btnEliminarA.addActionListener(this);
        this.vistaArrendatarios.btnCancelA.addActionListener(this);

        this.vistaArrendatarios.txtbuscarArrendatario.getDocument().addDocumentLi
stener(this);

        this.vistaArrendatarios.tablaArrendatarios.getSelectionModel().addListSel
ectionListener(this);

    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource()== vistaArrendatarios.btnRegistrarA){
            int identificacion =
Integer.parseInt(vistaArrendatarios.txtidentificacion.getText());
            String nombre =
vistaArrendatarios.txtNombreArrendatario.getText();
            String apellido = vistaArrendatarios.txtApellidoA.getText();
            int contacto =
Integer.parseInt(vistaArrendatarios.txtContactoA.getText());
            String referencias =
vistaArrendatarios.txtReferencias.getText();

            arrendatario arrendatario = new
arrendatario(identificacion,nombre,apellido,contacto,referencias);
            logicaArrendatario registrar = new logicaArrendatario();
            registrar.registrarArrendatario(arrendatario);

            tablas tablas = new tablas();

            tablas.MostrarArrendatario(vistaArrendatarios.tablaArrendatarios);
            limpiarEntradasARren();

        }

        if (e.getSource()== vistaArrendatarios.btnActualizarA){

            String nombre =

```

```

vistaArrendatarios.txtNombreArrendatario.getText();
    String apellido = vistaArrendatarios.txtApellidoA.getText();
    int contacto =
Integer.parseInt(vistaArrendatarios.txtContactoA.getText());
    String referencias =
vistaArrendatarios.txtReferencias.getText();
    int identificacion =
Integer.parseInt(vistaArrendatarios.txtidentificacion.getText());

    arrendatario actulizarARREn = new
arrendatario(identificacion,nombre,apellido,contacto,referencias);
    logicaArrendatario logicaArrendatario = new
logicaArrendatario();

logicaArrendatario.actualizarArrendatario(actulizarARREn,identificacion);
    tablas tablas = new tablas();

tablas.MostrarArrendatario(vistaArrendatarios.tablaArrendatarios);

        vistaArrendatarios.btnRegistrarA.setEnabled(false);
        vistaArrendatarios.btnEliminarA.setEnabled(false);

    }

    if (e.getSource()== vistaArrendatarios.btnEliminarA) {
        int id =
Integer.parseInt(vistaArrendatarios.txtidentificacion.getText());
        logicaArrendatario logicaArrendatario = new
logicaArrendatario();
        logicaArrendatario.eliminarArrendatario(id);

        tablas tablas = new tablas();

tablas.MostrarArrendatario(vistaArrendatarios.tablaArrendatarios);
        limpiarEntradasARREn();
    }
    if (e.getSource()== vistaArrendatarios.btnCancelarA) {
        limpiarEntradasARREn();
        vistaArrendatarios.btnActualizarA.setEnabled(true);
        vistaArrendatarios.btnEliminarA.setEnabled(true);
        vistaArrendatarios.btnRegistrarA.setEnabled(true);
    }
}

private void limpiarEntradasARREn()
{
    vistaArrendatarios.txtidentificacion.setText("");
    vistaArrendatarios.txtNombreArrendatario.setText("");
    vistaArrendatarios.txtApellidoA.setText("");
    vistaArrendatarios.txtContactoA.setText("");
    vistaArrendatarios.txtReferencias.setText("");
    vistaArrendatarios.txtbuscarArrendatario.setText("");
}

@Override
public void valueChanged(ListSelectionEvent e) {
    if (!e.getValueIsAdjusting()) {
        e.getSource();
        vistaArrendatarios.tablaArrendatarios.getSelectionModel();
    }
}

```

```

        int filasObtenidas =
vistaArrendatarios.tablaArrendatarios.getSelectedRow();

        if (filasObtenidas >= 0){
            TableModel modelo =
vistaArrendatarios.tablaArrendatarios.getModel();

            Object identificacion =
modelo.getValueAt(filasObtenidas,0);
            Object nombre = modelo.getValueAt(filasObtenidas,1);
            Object apellido = modelo.getValueAt(filasObtenidas,2);
            Object contacto = modelo.getValueAt(filasObtenidas,3);
            Object referencias = modelo.getValueAt(filasObtenidas,4)

vistaArrendatarios.txtidentificacion.setText(identificacion.toString());
vistaArrendatarios.txtNombreArrendatario.setText(nombre.toString());
vistaArrendatarios.txtApellidoA.setText(apellido.toString());
vistaArrendatarios.txtContactoA.setText(contacto.toString());
vistaArrendatarios.txtReferencias.setText(referencias.toString());

            System.out.println("fila: " + filasObtenidas);
        }
    }

    public void buscar(){
        String valor =
vistaArrendatarios.txtbuscarArrendatario.getText();
        tablas= new tablas();
        tablas.BuscarArrendatario(valor,vistaArrendatarios.tablaArrendatarios);
    }

    @Override
    public void insertUpdate(DocumentEvent e) {
        buscar();
    }

    @Override
    public void removeUpdate(DocumentEvent e) {
        buscar();
    }

    @Override
    public void changedUpdate(DocumentEvent e) {
        buscar();
    }
}

```

En el mismo paquete llamado **CONTROLADOR**, cree una clase llamada **controladorpropiedades**. Esta clase contendrá los métodos que controlan lo que se enviará a la tabla **propiedades**.

```
package CONTROLADOR;

import MODELO.Propiedades;
import Vistas.vistaPropiedades;
import LOGICAMODELO.logicaPropiedades;
import LOGICAMODELO.tablas;

import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;

import javax.swing.table.TableModel;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class controladorPropiedades implements ActionListener,
ListSelectionListener, DocumentListener {

    vistaPropiedades vistaPro;
    public static tablas tablas;

    public controladorPropiedades(vistaPropiedades propiedades) {
        this.vistaPro = propiedades;
        this.vistaPro.btnREGISTRAR.addActionListener(this);
        this.vistaPro.btnActualizar.addActionListener(this);
        this.vistaPro.btnEliminarPRO.addActionListener(this);

this.vistaPro.txtBuscarPropiedad.getDocument().addDocumentListener(this);

this.vistaPro.tablaPropiedades.getSelectionModel().addListSelectionListen
er(this);
        this.vistaPro.btnCancelarPRO.addActionListener(this);

    }

    @Override
    public void actionPerformed(ActionEvent e) {

        if (e.getSource() == vistaPro.btnREGISTRAR) {

            String nombrePropiedad = vistaPro.txtNOMBRE.getText();
            String direccion = vistaPro.txtDIRECCION.getText();
            String tipoDePropiedad=
vistaPro.boxTIPOPRO.getSelectedItem().toString();
            String tamano = vistaPro.txtTAMANO.getText();
            int numhabitaciones =
Integer.parseInt(vistaPro.txtHABITACIONES.getText());
```

```

        int precio = Integer.parseInt(vistaPro.txtPRECIO.getText());
        String disponibilidad =
vistaPro.boXESTADO.getSelectedItem().toString();

        Propiedades nuevaPRO = new
Propiedades(nombrePropiedad,direccion,tipoDePropiedad, tamano,
numhabitaciones, precio, disponibilidad);
        logicaPropiedades nuevaLogica = new logicaPropiedades();
        nuevaLogica.registrarPropiedades(nuevaPRO);

        tablas = new tablas();
        tablas.MostrarPropiedades(vistaPro.tablaPropiedades);

        limpiar();
    }

    if (e.getSource() == vistaPro.btnActualizar) {

        String nombrePropiedad = vistaPro.txtNOMBRE.getText();
        String direccion = vistaPro.txtDIRECCION.getText();
        String tipoDePropiedad =
vistaPro.boxTIPOPRO.getSelectedItem().toString();
        String tamano = vistaPro.txtTAMANO.getText();
        int numhabitaciones =
Integer.parseInt(vistaPro.txtHABITACIONES.getText());
        int precio = Integer.parseInt(vistaPro.txtPRECIO.getText());
        String disponibilidad =
vistaPro.boXESTADO.getSelectedItem().toString();
        int id = Integer.parseInt(vistaPro.txtID.getText());

        Propiedades propiedades = new Propiedades(id,nombrePropiedad,
direccion, tipoDePropiedad, tamano, numhabitaciones, precio,
disponibilidad);
        logicaPropiedades logicaPropiedades = new
logicaPropiedades();
        logicaPropiedades.actualizarPropiedad(propiedades, id);

        tablas = new tablas();
        tablas.MostrarPropiedades(vistaPro.tablaPropiedades);
        limpiar();
    }

    if (e.getSource() == vistaPro.btnEliminarPRO) {
        int idpropiedades =
Integer.parseInt(vistaPro.txtID.getText());
        logicaPropiedades logicaPropiedades = new
logicaPropiedades();
        logicaPropiedades.eliminarPropiedad(idpropiedades);

        tablas = new tablas();
        tablas.MostrarPropiedades(vistaPro.tablaPropiedades);
        limpiar();
    }

    if (e.getSource() == vistaPro.btnCancelarPRO) {

```

```

        limpiar();
    }

}

@Override
public void valueChanged(ListSelectionEvent e) {
    if (!e.getValueIsAdjusting()) {
        e.getSource();
        vistaPro.tablaPropiedades.getSelectionModel();
        int filasObtenidas =
vistaPro.tablaPropiedades.getSelectedRow();

        if (filasObtenidas >= 0) {
            TableModel modelo =
vistaPro.tablaPropiedades.getModel();

            Object idpropiedades =
modelo.getValueAt(filasObtenidas, 0);
            Object nombrePropiedad =
modelo.getValueAt(filasObtenidas, 1);
            Object direccion = modelo.getValueAt(filasObtenidas, 2);
            Object tipoDePropiedad =
modelo.getValueAt(filasObtenidas, 3);
            Object tamaño = modelo.getValueAt(filasObtenidas, 4);
            Object numhabitaciones =
modelo.getValueAt(filasObtenidas, 5);
            Object precio = modelo.getValueAt(filasObtenidas, 6);
            Object disponibilidad =
modelo.getValueAt(filasObtenidas, 7);

            vistaPro.txtID.setText(idpropiedades.toString());
            vistaPro.txtNOMBRE.setText(nombrePropiedad.toString());
            vistaPro.txtDIRECCION.setText(direccion.toString());
            vistaPro.boxTIPOPRO.getSelectedItem().toString();
            vistaPro.txtTAMANO.setText(tamaño.toString());

vistaPro.txtHABITACIONES.setText(numhabitaciones.toString());
            vistaPro.txtPRECIO.setText(precio.toString());
            vistaPro.boXESTADO.getSelectedItem().toString();

            System.out.println("fila: " + filasObtenidas);
        }
    }
}

public void buscar() {
    String valor = vistaPro.txtBuscarPropiedad.getText();
    tablas = new tablas();
    tablas.BuscarPropiedad(valor, vistaPro.tablaPropiedades);
}

@Override
public void insertUpdate(DocumentEvent e) {
    buscar();
}

```

```
}

@Override
public void removeUpdate(DocumentEvent e) {
    buscar();
}

@Override
public void changedUpdate(DocumentEvent e) {
    buscar();
}

private void limpiar()
{
    vistaPro.txtID.setText("");
    vistaPro.txtNOMBRE.setText("");
    vistaPro.txtPRECIO.setText("");
    vistaPro.txtTAMANO.setText("");
    vistaPro.txtHABITACIONES.setText("");
    vistaPro.txtBuscarPropiedad.setText("");
    vistaPro.txtDIRECCION.setText("");
}
}
```

Paso 11: Crear la vista que recibe los datos del usuario

Creas un paquete llamado *vistas*. Luego creas una clase llamada *vistaAlquiler*. Esta clase recibe los datos que se enviará a la **tabla alquiler**.

```
package Vistas;

import CONTROLADOR.controladorArrendatario;
import CONTROLADOR.controladorPropiedades;

import LOGICAMODELO.tablas;

import javax.swing.JFrame;
public class vistaAlquiler extends JFrame {

    public static tablas tablas;

    public static vistaArrendatarios vistalarrendatarios;
    public static controladorArrendatario controladorArren;

    public static vistaPropiedades vistaPropiedades;
    public static controladorPropiedades pro;

    public vistaAlquiler() {

        initComponents();
        this.setTitle("PANEL PRINCIPAL");
        this.setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        tablaAlquiler = new javax.swing.JTable();
        jLabel1 = new javax.swing.JLabel();
        btnEliminarAlquiler = new javax.swing.JButton();
        btnLlamarArrendatario = new javax.swing.JButton();
        btnLlamarPropiedades = new javax.swing.JButton();
        txtBuscar = new javax.swing.JTextField();
        jLabel2 = new javax.swing.JLabel();
        btnRegistrarAlquiler = new javax.swing.JButton();
        btnActualizarAlquiler = new javax.swing.JButton();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        txtIDalquiler = new javax.swing.JTextField();
        txtPrecioPROPIEDAD = new javax.swing.JTextField();
        btnCancelar = new javax.swing.JButton();
        jLabel6 = new javax.swing.JLabel();
        jLabel7 = new javax.swing.JLabel();
    }
}
```



```

jLabel8 = new javax.swing.JLabel();
jLabel9 = new javax.swing.JLabel();
txtIDarrendatario = new javax.swing.JTextField();
txtNombre = new javax.swing.JTextField();
txtxiIDpropiedades = new javax.swing.JTextField();
txtTIPOPROPIEDAD = new javax.swing.JTextField();
txtEstadoPROPIEDAD = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setMinimumSize(new java.awt.Dimension(980, 450));
setPreferredSize(new java.awt.Dimension(980, 450));
getContentPane().setLayout(new
org.netbeans.lib.awtextra.AbsoluteLayout());

        tablaAlquiler.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
jScrollPane1.setViewportView(tablaAlquiler);

        getContentPane().add(jScrollPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(270, 90, 680, 290));

        jLabel11.setFont(new java.awt.Font("Segoe Print", 3, 24)); //
NOI18N
        jLabel11.setText("ALQUILER");
        getContentPane().add(jLabel11, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 20, 161, 41));

        btnEliminarAlquiler.setFont(new java.awt.Font("Segoe UI", 1,
14)); // NOI18N
        btnEliminarAlquiler.setText("ELIMINAR");
        getContentPane().add(btnEliminarAlquiler, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 340, 100, -1));

        btnLlamarArrendatario.setFont(new java.awt.Font("Segoe UI", 1,
14)); // NOI18N
        btnLlamarArrendatario.setText("NUEVO ARRENDATARIO");
        btnLlamarArrendatario.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnLlamarArrendatarioActionPerformed(evt);
            }
        });
        getContentPane().add(btnLlamarArrendatario, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 10, 240, 30));

        btnLlamarPropiedades.setFont(new java.awt.Font("Segoe UI", 1,
14)); // NOI18N

```

```

        btnLlamarPropiedades.setText("NUEVA PROPIEDAD");
        btnLlamarPropiedades.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                btnLlamarPropiedadesActionPerformed(evt);
            }
        });
        getContentPane().add(btnLlamarPropiedades, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 50, 240, -1));
        getContentPane().add(txtBuscar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(570, 60, 370, -1));

        jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
        jLabel2.setText("buscar");
        getContentPane().add(jLabel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(570, 20, 49, 66));

        btnRegistrarAlquiler.setFont(new java.awt.Font("Segoe UI", 1,
12)); // NOI18N
        btnRegistrarAlquiler.setText("REGISTRAR");
        getContentPane().add(btnRegistrarAlquiler, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 300, -1, -1));

        btnActualizarAlquiler.setFont(new java.awt.Font("Segoe UI", 1,
12)); // NOI18N
        btnActualizarAlquiler.setText("ACTUALIZAR ");
        getContentPane().add(btnActualizarAlquiler, new
org.netbeans.lib.awtextra.AbsoluteConstraints(130, 300, 110, 30));

        jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel3.setText("ID");
        getContentPane().add(jLabel3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 80, -1, -1));

        jLabel4.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel4.setText("PRECIO");
        getContentPane().add(jLabel4, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 240, -1, -1));

        jLabel5.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel5.setText("ESTADO");
        getContentPane().add(jLabel5, new
org.netbeans.lib.awtextra.AbsoluteConstraints(40, 270, -1, -1));
        getContentPane().add(txtIDalquiler, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 80, 80, -1));

        txtPrecioPROPIEDAD.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txtPrecioPROPIEDADActionPerformed(evt);
            }
        });
        getContentPane().add(txtPrecioPROPIEDAD, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 240, 150, -1));

        btnCancelar.setFont(new java.awt.Font("Segoe UI", 1, 12)); //
NOI18N

```

```

        btnCancelar.setText("Cancelar");
        getContentPane().add(btnCancelar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(130, 340, 110, 30));

        jLabel6.setText("IDENTIFICACION");
        getContentPane().add(jLabel6, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 120, -1, -1));

        jLabel7.setText("NOMBRE");
        getContentPane().add(jLabel7, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 150, -1, -1));

        jLabel8.setText("PROPIEDAD");
        getContentPane().add(jLabel8, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 180, -1, -1));

        jLabel9.setText("TIPO DE PROPIEDAD");
        getContentPane().add(jLabel9, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 210, -1, -1));

        txtIDarrendatario.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                txtIDarrendatarioActionPerformed(evt);
            }
        });
        getContentPane().add(txtIDarrendatario, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 120, 140, -1));
        getContentPane().add(txtNombre, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 150, 150, -1));
        getContentPane().add(txtxiIDpropiedades, new
org.netbeans.lib.awtextra.AbsoluteConstraints(100, 180, 140, -1));
        getContentPane().add(txtTIPOPROPIEDAD, new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 210, 90, -1));
        getContentPane().add(txtEstadoPROPIEDAD, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 270, 150, -1));

        getAccessibleContext().setAccessibleName("ALQUILER");

        pack();
    } // </editor-fold> //GEN-END: initComponents

    private void
btnLlamarArrendatarioActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_btnLlamarArrendatarioActionPerformed

        vistalarrendatarios = new vistaArrendatarios();
        vistalarrendatarios.setVisible(true);
        vistalarrendatarios.setLocationRelativeTo(null);
        controladorArren = new
controladorArrendatario(vistalarrendatarios);

        tablas = new tablas();

        tablas.MostrarArrendatario(vistalarrendatarios.tablaArrendatarios);

```

```

        } //GEN-LAST:event_btnLlamarArrendatarioActionPerformed

        private void
txtPrecioPROPIEDADActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_txtPrecioPROPIEDADActionPerformed
        // TODO add your handling code here:
        } //GEN-LAST:event_txtPrecioPROPIEDADActionPerformed

        private void
btnLlamarPropiedadesActionPerformed(java.awt.event.ActionEvent evt)
{ //GEN-FIRST:event_btnLlamarPropiedadesActionPerformed

        vistaPropiedades = new vistaPropiedades();
        vistaPropiedades.setVisible(true);
        vistaPropiedades.setLocationRelativeTo(null);
        pro = new controladorPropiedades(vistaPropiedades);
        tablas = new tablas();

        tablas.MostrarPropiedades(vistaPropiedades.tablaPropiedades);
        } //GEN-LAST:event_btnLlamarPropiedadesActionPerformed

        private void
txtIDarrendatarioActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_txtIDarrendatarioActionPerformed
        // TODO add your handling code here:
        } //GEN-LAST:event_txtIDarrendatarioActionPerformed

        /**
         * @param args the command line arguments
         */
        public static void main(String args[]) {

            /* Set the Nimbus look and feel */
            //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
            /* If Nimbus (introduced in Java SE 6) is not available, stay
with the default look and feel.
             * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
             */
            try {
                for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                    if ("Nimbus".equals(info.getName())) {
                        javax.swing.UIManager.setLookAndFeel(info.getClassName());
                        break;
                    }
                }
            } catch (ClassNotFoundException ex) {

                java.util.logging.Logger.getLogger(vistaAlquiler.class.getName()).log(jav
a.util.logging.Level.SEVERE, null, ex);
            } catch (InstantiationException ex) {

```

```

java.util.logging.Logger.getLogger(vistaAlquiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(vistaAlquiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(vistaAlquiler.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {

        }

    });
}

// Variables declaration - do not modify//GEN-BEGIN:variables
public javax.swing.JButton btnActulizarAlquiler;
public javax.swing.JButton btnCancelar;
public javax.swing.JButton btnEliminarAlquiler;
public javax.swing.JButton btnLlamarArrendatario;
public javax.swing.JButton btnLlamarPropiedades;
public javax.swing.JButton btnRegistrarAlquiler;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JScrollPane jScrollPane1;
public javax.swing.JTable tablaAlquiler;
public javax.swing.JTextField txtBuscar;
public javax.swing.JTextField txtEstadoPROPIEDAD;
public javax.swing.JTextField txtIDalquiler;
public javax.swing.JTextField txtIDarrendatario;
public javax.swing.JTextField txtNombre;
public javax.swing.JTextField txtPrecioPROPIEDAD;
public javax.swing.JTextField txtTIPOPROPIEDAD;
public javax.swing.JTextField txtxiIDpropiedades;
// End of variables declaration//GEN-END:variables
}

```


PANEL PRINCIPAL

ALQUILER

ID

IDENTIFICACION

NOMBRE

PROPIEDAD

TIPO DE PROPIEDAD

PRECIO

ESTADO

REGISTRAR

ELIMINAR

ACTUALIZAR

Cancelar

NUEVO ARRENDATARIO

NUEVA PROPIEDAD

buscar

Id	IDENTIFICACION	NOMBRE	NOMBRE PRO...	TIPO DE PROP...	PRECIO	ESTADO
1	1096038399	santiago	jardinn	Apartamento	150000	DISPONIBLE

En el paquete llamado **vistas**, cree una clase llamada **vistaArrendatario**. Esta clase recibe los datos que se enviará a la tabla **arrendatario**.

```
package Vistas;

import CONTROLADOR.controladorAlquiler;
import CONTROLADOR.controladorArrendatario;
import CONTROLADOR.controladorPropiedades;
import LOGICAMODELO.logicaArrendatario;
import LOGICAMODELO.tablas;

public class vistaArrendatarios extends javax.swing.JFrame {

    public vistaArrendatarios() {
        initComponents();
    }
    @SuppressWarnings("unchecked")

    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        tablaArrendatarios = new javax.swing.JTable();
        jLabel1 = new javax.swing.JLabel();
    }
}
```

```

jLabel2 = new javax.swing.JLabel();
jLabel3 = new javax.swing.JLabel();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
txtIdentificacion = new javax.swing.JTextField();
txtNombreArrendatario = new javax.swing.JTextField();
txtApellidoA = new javax.swing.JTextField();
txtReferencias = new javax.swing.JTextField();
txtContactoA = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
txtBuscarArrendatario = new javax.swing.JTextField();
btnRegistrarA = new javax.swing.JButton();
btnActualizarA = new javax.swing.JButton();
btnEliminarA = new javax.swing.JButton();
btnCancelarA = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setMinimumSize(new java.awt.Dimension(810, 410));
getContentPane().setLayout(new
org.netbeans.lib.awtextra.AbsoluteLayout());

    tablaArrendatarios.setModel(new
javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane1.setViewportViewView(tablaArrendatarios);

getContentPane().add(jScrollPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(223, 56, 564, 220));

jLabel1.setFont(new java.awt.Font("Segoe Script", 3, 18)); //
NOI18N
jLabel1.setText("ARRENDATARIOS");
getContentPane().add(jLabel1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(12, 13, 202, -1));

jLabel2.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel2.setText("Identifiacion");
getContentPane().add(jLabel2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(12, 59, -1, -1));

jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel3.setText("Nombre");
getContentPane().add(jLabel3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(12, 94, -1, -1));

jLabel4.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N

```

```

        jLabel4.setText("Apellido");
        getContentPane().add(jLabel4, new
org.netbeans.lib.awtextra.AbsoluteConstraints(12, 129, -1, -1));

        jLabel5.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel5.setText("Contacto");
        getContentPane().add(jLabel5, new
org.netbeans.lib.awtextra.AbsoluteConstraints(12, 169, -1, -1));

        jLabel6.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel6.setText("Referencias");
        getContentPane().add(jLabel6, new
org.netbeans.lib.awtextra.AbsoluteConstraints(12, 201, -1, -1));
        getContentPane().add(txtidentificacion, new
org.netbeans.lib.awtextra.AbsoluteConstraints(94, 56, 84, -1));
        getContentPane().add(txtNombreArrendatario, new
org.netbeans.lib.awtextra.AbsoluteConstraints(94, 91, 84, -1));
        getContentPane().add(txtApellidoA, new
org.netbeans.lib.awtextra.AbsoluteConstraints(94, 126, 84, -1));
        getContentPane().add(txtReferencias, new
org.netbeans.lib.awtextra.AbsoluteConstraints(94, 201, 84, -1));
        getContentPane().add(txtContactoA, new
org.netbeans.lib.awtextra.AbsoluteConstraints(94, 166, 84, -1));

        jLabel7.setText("CONSULTAR");
        getContentPane().add(jLabel7, new
org.netbeans.lib.awtextra.AbsoluteConstraints(352, 24, -1, -1));
        getContentPane().add(txtbuscarArrendatario, new
org.netbeans.lib.awtextra.AbsoluteConstraints(424, 21, 182, -1));

        btnRegistrarA.setFont(new java.awt.Font("Segoe UI", 1, 12)); //
NOI18N
        btnRegistrarA.setText("REGISTRAR");
        getContentPane().add(btnRegistrarA, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 230, -1, -1));

        btnActualizarA.setFont(new java.awt.Font("Segoe UI", 1, 12)); //
NOI18N
        btnActualizarA.setText("ACTUALIZAR");
        getContentPane().add(btnActualizarA, new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 230, -1, -1));

        btnEliminarA.setFont(new java.awt.Font("Segoe UI", 1, 12)); //
NOI18N
        btnEliminarA.setText("ELIMINAR");
        getContentPane().add(btnEliminarA, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 260, -1, -1));

        btnCancelarA.setFont(new java.awt.Font("Segoe UI", 1, 12)); //
NOI18N
        btnCancelarA.setText("CANCELAR");
        getContentPane().add(btnCancelarA, new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 260, 100, -1));

        pack();
    } // </editor-fold> // GEN-END: initComponents

```



```

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
    /* If Nimbus (introduced in Java SE 6) is not available, stay
with the default look and feel.
    * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
    */
    try {
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(vistaArrendatarios.class.getName()).lo
g(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(vistaArrendatarios.class.getName()).lo
g(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(vistaArrendatarios.class.getName()).lo
g(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(vistaArrendatarios.class.getName()).lo
g(java.util.logging.Level.SEVERE, null, ex);
    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {

        }

    });
}


// Variables declaration - do not modify//GEN-BEGIN:variables
public javax.swing.JButton btnActualizarA;
public javax.swing.JButton btnCancelarA;
public javax.swing.JButton btnEliminarA;
public javax.swing.JButton btnRegistrarA;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;

```

```

private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JScrollPane jScrollPane1;
public javax.swing.JTable tablaArrendatarios;
public javax.swing.JTextField txtApellidoA;
public javax.swing.JTextField txtContactoA;
public javax.swing.JTextField txtNombreArrendatario;
public javax.swing.JTextField txtReferencias;
public javax.swing.JTextField txtbuscarArrendatario;
public javax.swing.JTextField txtidentificacion;
// End of variables declaration//GEN-END:variables
}

```



ARRENDATARIOS

CONSULTAR

Identifiacion

Nombre

Apellido

Contacto

Referencias

REGISTRAR

ACTUALIZAR

ELIMINAR

CANCELAR

Identificacion	Nombre	Apellido	Contacto	Referencias
1	yhonier	arias	310719	filosofo
2	anderson	arias	31023	vigilante
3	miguel	morales	453	estudiante
4	william	rangel	32035	arquitecto
5	sebastian	dimate	2334	estudiante
1096038399	santiago	garcia	3200453	estudiante

En el paquete llamado *vistas*, cree una clase llamada **vistaPropiedades**. Esta clase recibe los datos que se enviará a la tabla *propiedades*.

```
package Vistas;

import CONTROLADOR.controladorAlquiler;
import CONTROLADOR.controladorArrendatario;
import CONTROLADOR.controladorPropiedades;
import LOGICAMODELO.logicaArrendatario;
import LOGICAMODELO.tablas;

public class vistaPropiedades extends javax.swing.JFrame {

    public vistaPropiedades() {
        initComponents();
    }

    @SuppressWarnings("unchecked")
    private void initComponents() {

        jScrollPane1 = new javax.swing.JScrollPane();
        tablaPropiedades = new javax.swing.JTable();
    }
}
```

```

jLabel11 = new javax.swing.JLabel();
jLabel12 = new javax.swing.JLabel();
jLabel13 = new javax.swing.JLabel();
jLabel14 = new javax.swing.JLabel();
jLabel15 = new javax.swing.JLabel();
jLabel16 = new javax.swing.JLabel();
jLabel17 = new javax.swing.JLabel();
jLabel18 = new javax.swing.JLabel();
txtID = new javax.swing.JTextField();
txtNOMBRE = new javax.swing.JTextField();
jLabel19 = new javax.swing.JLabel();
txtBuscarPropiedad = new javax.swing.JTextField();
txtDIRECCION = new javax.swing.JTextField();
txtPRECIO = new javax.swing.JTextField();
txtTAMANO = new javax.swing.JTextField();
btnREGISTRAR = new javax.swing.JButton();
btnActualizar = new javax.swing.JButton();
btnEliminarPRO = new javax.swing.JButton();
btnCancelarPRO = new javax.swing.JButton();
jLabel10 = new javax.swing.JLabel();
txtHABITACIONES = new javax.swing.JTextField();
boXESTADO = new javax.swing.JComboBox<>();
boxTIPOPRO = new javax.swing.JComboBox<>();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setMinimumSize(new java.awt.Dimension(1000, 450));
getContentPane().setLayout(new
org.netbeans.lib.awtextra.AbsoluteLayout());

    tablaPropiedades.setModel(new
javax.swing.table.DefaultTableModel(
    new Object [][] {
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null},
        {null, null, null, null}
    },
    new String [] {
        "Title 1", "Title 2", "Title 3", "Title 4"
    }
));
jScrollPane1.setViewportViewView(tablaPropiedades);

getContentPane().add(jScrollPane1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(260, 80, 740, 310));

jLabel11.setFont(new java.awt.Font("Times New Roman", 3, 24)); //
NOI18N
jLabel11.setText("PROPIEDADES");
getContentPane().add(jLabel11, new
org.netbeans.lib.awtextra.AbsoluteConstraints(12, 0, 202, 35));

jLabel12.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
jLabel12.setText("ID");
getContentPane().add(jLabel12, new
org.netbeans.lib.awtextra.AbsoluteConstraints(40, 40, -1, -1));

```

```

        jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel3.setText("Nombre Propiedad");
        getContentPane().add(jLabel3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 70, -1, -1));

        jLabel4.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel4.setText("Direccion");
        getContentPane().add(jLabel4, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 100, -1, -1));

        jLabel5.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel5.setText("Tipo de Propiedad");
        getContentPane().add(jLabel5, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 130, -1, -1));

        jLabel6.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel6.setText("Numero de Habitaciones");
        getContentPane().add(jLabel6, new
org.netbeans.lib.awtextra.AbsoluteConstraints(10, 200, 140, -1));

        jLabel7.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel7.setText("Tamaño");
        getContentPane().add(jLabel7, new
org.netbeans.lib.awtextra.AbsoluteConstraints(30, 170, -1, -1));

        jLabel8.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel8.setText("Precio");
        getContentPane().add(jLabel8, new
org.netbeans.lib.awtextra.AbsoluteConstraints(40, 230, -1, -1));
        getContentPane().add(txtID, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 40, 130, -1));
        getContentPane().add(txtNOMBRE, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 70, 132, -1));

        jLabel9.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel9.setText("buscar Propiedades Disponibles");
        getContentPane().add(jLabel9, new
org.netbeans.lib.awtextra.AbsoluteConstraints(270, 40, -1, -1));
        getContentPane().add(txtBuscarPropiedad, new
org.netbeans.lib.awtextra.AbsoluteConstraints(450, 40, 207, -1));
        getContentPane().add(txtDIRECCION, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 100, 130, -1));
        getContentPane().add(txtPRECIO, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 230, 129, -1));
        getContentPane().add(txtTAMANO, new
org.netbeans.lib.awtextra.AbsoluteConstraints(121, 160, 130, -1));

        btnREGISTRAR.setFont(new java.awt.Font("Segoe UI", 1, 12)); //
NOI18N
        btnREGISTRAR.setText("REGISTRAR");
        getContentPane().add(btnREGISTRAR, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 310, 100, -1));

        btnActualizar.setFont(new java.awt.Font("Segoe UI", 1, 12)); //
NOI18N
        btnActualizar.setText("ACTUALIZAR");

```

```

        getContentPane().add(btnActualizar, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 310, -1, -1));

        btnEliminarPRO.setFont(new java.awt.Font("Segoe UI", 1, 12)); //
NOI18N
        btnEliminarPRO.setText("ELIMINAR");
        getContentPane().add(btnEliminarPRO, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 350, 100, -1));

        btnCancelarPRO.setFont(new java.awt.Font("Segoe UI", 1, 12)); //
NOI18N
        btnCancelarPRO.setText("CANCELAR");
        getContentPane().add(btnCancelarPRO, new
org.netbeans.lib.awtextra.AbsoluteConstraints(140, 350, 100, -1));

        jLabel110.setFont(new java.awt.Font("Segoe UI", 1, 12)); // NOI18N
        jLabel110.setText("Estado");
        getContentPane().add(jLabel110, new
org.netbeans.lib.awtextra.AbsoluteConstraints(20, 270, -1, -1));
        getContentPane().add(txtHABITACIONES, new
org.netbeans.lib.awtextra.AbsoluteConstraints(150, 200, 60, -1));

        boXESTADO.setFont(new java.awt.Font("Segoe UI", 1, 12)); //
NOI18N
        boXESTADO.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "DISPONIBLE", "NO DISPONIBLE" }));
        getContentPane().add(boXESTADO, new
org.netbeans.lib.awtextra.AbsoluteConstraints(110, 270, -1, -1));

        boxTIPOPRO.setFont(new java.awt.Font("Segoe UI", 1, 12)); //
NOI18N
        boxTIPOPRO.setModel(new javax.swing.DefaultComboBoxModel<>(new
String[] { "Casa", "Apartamento", "Campestre", " " }));
        getContentPane().add(boxTIPOPRO, new
org.netbeans.lib.awtextra.AbsoluteConstraints(120, 130, 130, -1));

        pack();
    } // </editor-fold> // GEN-END: initComponents

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay
with the default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());

```

```

        break;
    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(vistaPropiedades.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(vistaPropiedades.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(vistaPropiedades.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(vistaPropiedades.class.getName()).log(
java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>


/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {

    }
});
}

// Variables declaration - do not modify//GEN-BEGIN:variables
public javax.swing.JComboBox<String> boXESTADO;
public javax.swing.JComboBox<String> boxTIPOPRO;
public javax.swing.JButton btnActualizar;
public javax.swing.JButton btnCancelarPRO;
public javax.swing.JButton btnEliminarPRO;
public javax.swing.JButton btnREGISTRAR;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JScrollPane jScrollPane1;
public javax.swing.JTable tablaPropiedades;
public javax.swing.JTextField txtBuscarPropiedad;
public javax.swing.JTextField txtDIRECCION;
public javax.swing.JTextField txtHABITACIONES;
public javax.swing.JTextField txtID;
public javax.swing.JTextField txtNOMBRE;
public javax.swing.JTextField txtPRECIO;
public javax.swing.JTextField txtTAMANO;

```

```
// End of variables declaration//GEN-END:variables
```



PROPIEDADES

ID

Nombre Propiedad

Direccion

Tipo de Propiedad

Casa

Tamaño

Numero de Habitaciones

Precio

Estado

DISPONIBLE

REGISTRAR

ACTUALIZAR

ELIMINAR

CANCELAR

buscar Propiedades Disponibles

Id	Nombre Propi...	Direccion	Tipo Propiedad	Tamaño	Habitaciones	Precio	Disponibilidad
1	jardinn	armenia	Apartamento	120mertos	4	150000	DISPONIBLE
2	amanecer	tebaida	Apartamento	60 metros	6	50000	DISPONIBLE
3	manantial	armenia	Casa	30 metros	6	200000	NO DISPONIB...
4	casa de mi ab...	cali	Casa	30 metros	5	1350000	NO DISPONIB...

Paso 10: Actualiza tu metodo main para que el usuario pueda ingresar datos

Coloca la llamada a las clases del paquete *vista*, las cuales se encargan de recibir los datos de parte del usuario.

```
package main;

import CONTROLADOR.controladorAlquiler;
import Vistas.vistaAlquiler;
import LOGICAMODELO.tablas;

public class main {

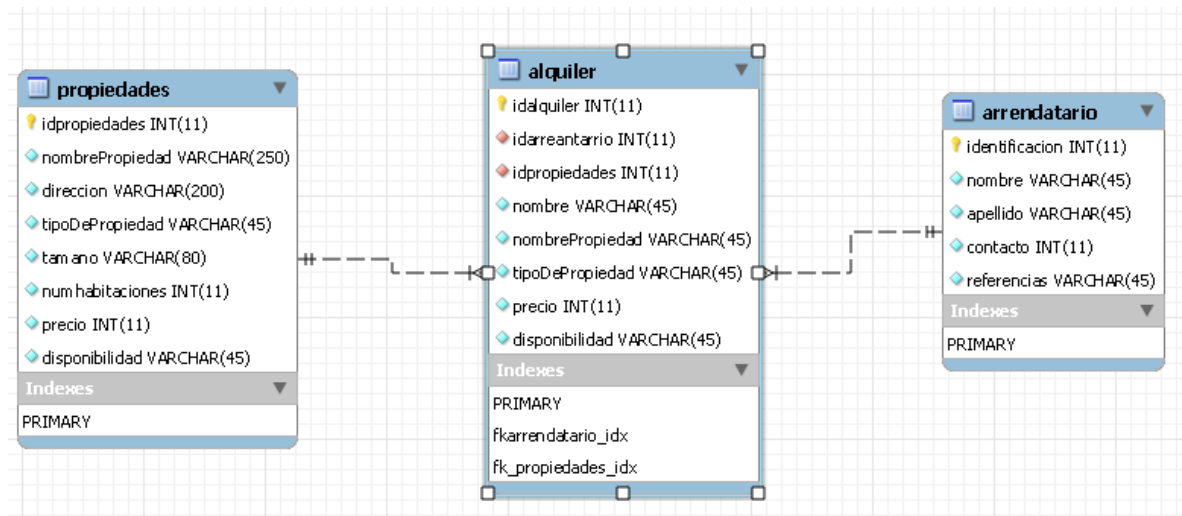
    public static vistaAlquiler vistaAlquiler;
    public static tablas tablas;
    public static controladorAlquiler contAlquiler;

    public static void main(String[] args) {

        vistaAlquiler = new vistaAlquiler();
        vistaAlquiler.setVisible(true);
        vistaAlquiler.setLocationRelativeTo(null);
        contAlquiler = new controladorAlquiler(vistaAlquiler);
        tablas = new tablas();
        tablas.MostrarAlquiler(vistaAlquiler.tablaAlquiler);

    }
}
```

Paso 11: Diagrama



Paso 12: Preguntas

¿Cuáles crees que son las principales ventajas de utilizar el patrón MVC en comparación con otros enfoques arquitectónicos?

- MVC ofrece una clara separación de responsabilidades, lo que mejora la organización del código y facilita la colaboración entre equipos. Además, permite una mayor modularidad y reutilización del código al dividir la aplicación en componentes interconectados pero independientes.

¿Por qué es importante separar la lógica de negocio de la interfaz de usuario en una aplicación?

- Separar la lógica de negocio de la interfaz de usuario facilita el mantenimiento y la escalabilidad del software. Permite que los desarrolladores trabajen de manera más eficiente en diferentes partes del sistema sin interferir entre sí. Además, promueve la reutilización del código, ya que los cambios en la lógica de negocio no necesariamente afectan a la interfaz de usuario y viceversa.

