

CRUD Y PRIMEFACES - Laboratorio 3

Fundamentos de Sistemas de Información

Luis David Ospina Chaparro
Manuel Guillermo Ochoa
Camilo Andres Sampedro
Sergio Andres Loaiza

Diego Jose Luis Botia

Universidad de Antioquia
Ingeniería de Sistemas
2015-1

Introducción

En este laboratorio, se busca crear un proyecto tipo CRUD(Create,Read, Update, Delete) basándose en la información de un concesionario de vehículos, con el objetivo de representar su venta general; usando primefaces, MySql DB, JSF para el backend y xhtml para la vista(Frontend).

Marco teórico

Java Server Pages:

En 1999, Sun Microsystems añadió un nuevo elemento a la colección de Enterprise Java tools: Java server pages(JSP), que están construidas en el tope de los Java Servlets y son diseñados para incrementar la eficiencia con la que los programadores y no programadores pueden crear contenido web. Se pueden combinar tecnologías de Servlets y JDBC(Java DataBase Connectivity) con el JSP de manera eficiente. [2]

Java Server Pages es una tecnología para desarrollar páginas web que incluyan contenido dinámico. A diferencia de una página HTML, la cual tiene contenido estático, es decir, que siempre permanece igual, una página JSP puede cambiar su contenido basado sobre cualquier número de variables de términos, incluyendo la identidad del usuario, el tipo de navegador, información proveída por el usuario y selecciones hechas por el usuario. Esta funcionalidad es la llave para que existan aplicaciones web como tiendas online y directorios de empleados, con contenido personalizado e internacionalizado. Una página JSP contiene elementos de marcado estándar, como tags HTML, como una página web regular, pero a su vez, también contiene elementos especiales del JSP que permite al servidor insertar contenido dinámico en la página. Los elementos JSP pueden ser usados para una variedad de propósitos, tal como buscar información de una base de datos o registrar preferencias del usuario. Cuando un usuario pregunta por una página JSP, el servidor ejecuta los elementos JSP, se unen los resultados con las partes estáticas de la página, y se envía la página compuesta dinámicamente al navegador. Generar contenido dinámico con JSP permite definir un número de elementos estándar que son útiles para cualquier aplicación, como acceder a componentes de JavaBeans, pasar el control entre páginas y compartir información. Los programadores también pueden extender la sintaxis JSP, implementando elementos con aplicaciones específicas que desarrollen tareas tales como acceder a bases de datos y Enterprise JavaBeans, enviar correos y generar HTML para presentar datos de una aplicación específica. Un conjunto común muy necesario son los elementos personalizados definidos por especificación relacionados a el JSP Standard Tag Library (JSTL). La combinación de elementos estándar y personalizados permiten la creación de poderosas aplicaciones.

JSP se utiliza como una solución eficiente al uso de interfaces no robustas como CGI(Common Gateway interface), la cual era la única herramienta para el desarrollo de contenido web, ya que para cada pedido que entra, el servidor web tiene que crear un nuevo proceso de sistema operativo, cargar un intérprete y un script, ejecutar el script, y luego hacer todo de nuevo. Esto es muy demandante para el servidor y no reacciona bien

cuando el tráfico de pedidos incrementa. Existen numerosas alternativas y soluciones como FastCGI, mod_perl from Apache, NSAPI from Netscape, ISAPI de Microsoft, and Java servlets de Sun Microsystems, que fueron creadas al pasar de los años, pero todas sufren el problema de trabajar con HTML directamente en el código del lenguaje de programación. Esto impulsó la creación de páginas webs dinámicas dentro del reino de los programadores, como el JSP que lo cambia todo.

Java Server Pages estan construidas en la cima del API de los Java Servlets, JSP tiene acceso a todo el poderoso emporio de las APIS de Java, incluyendo:

- JDBC
- Remote Method Invocation (RMI) and OMG CORBA support
- JNDI (Java Naming and Directory Interface)
- Enterprise JavaBeans (EJB)
- JMS (Java Message Service)
- JTA (Java Transaction API)
- JAXP (Java API for XML Processing)
- JavaMail

PrimeFaces

PrimeFaces es una suite de componentes open source de JSF con varias extensiones. Posee un set de componentes (HtmlEditor, Dialog, AutoComplete, Charts y mucho más), tiene Ajax empotrado basado en el estandar de las APIs de Ajax de JSF, es liviano, basta con tan solo un jar para ponerlo a funcionar, no tiene dependencias, viene con un kit para construir aplicaciones web móviles, trae un framework de skins con 35+ temas, amplia documentación, una comunidad de usuarios muy activa. [6]

Es una librería para JavaServer Faces que ofrece componentes de interfaz de usuario elegantes y sencillos que permiten desarrollar interfaces de usuario de una manera simple e intuitiva.

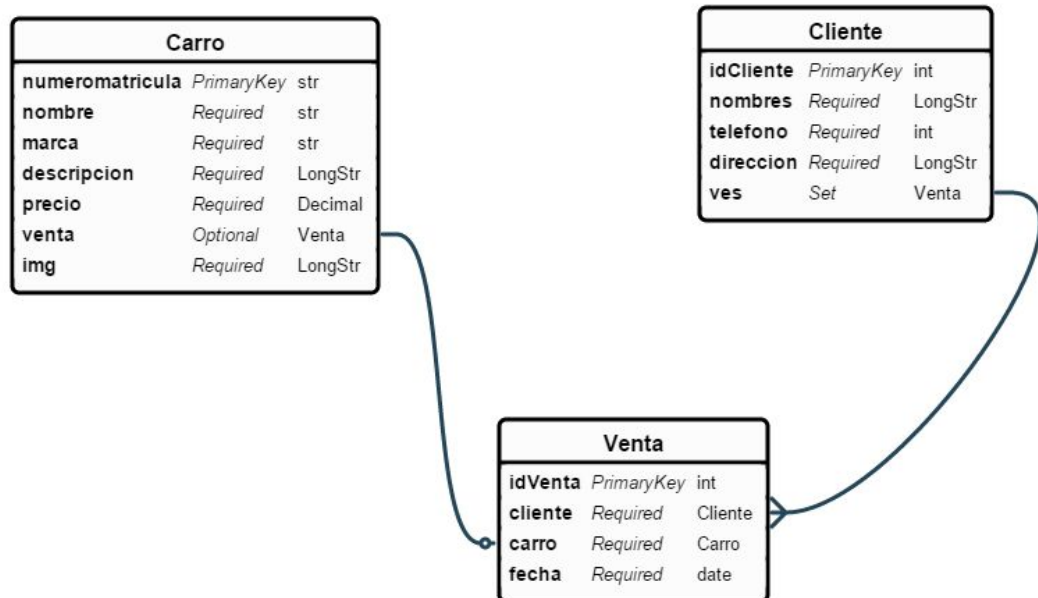
Los **principales características** de está librería de componentes son:

- Es en si mismo un gran conjunto de componentes de interfaz de usuario.
- Soporte AJAX para actualización de la interfaz de usuario de una manera selectiva.
- Posee temas prediseñados para utilizarlos sin mayor esfuerzo en nuestras aplicaciones.
- Posee componentes para desarrollar aplicaciones web para dispositivos móviles.

Procedimiento

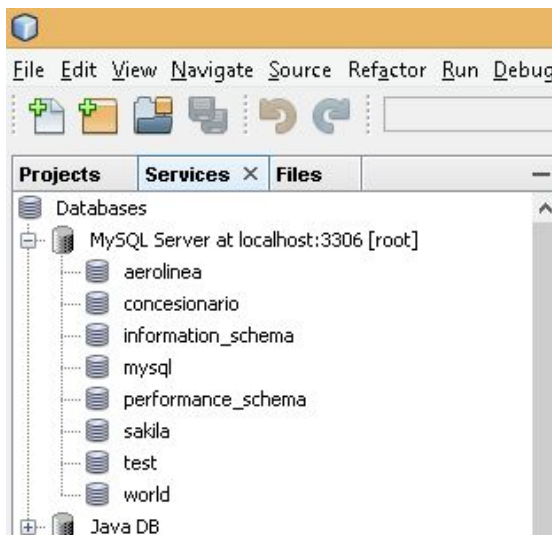
Diagrama entidad-relación

Primero vamos a crear el diagrama entidad relación, según lo especificado en el documento.



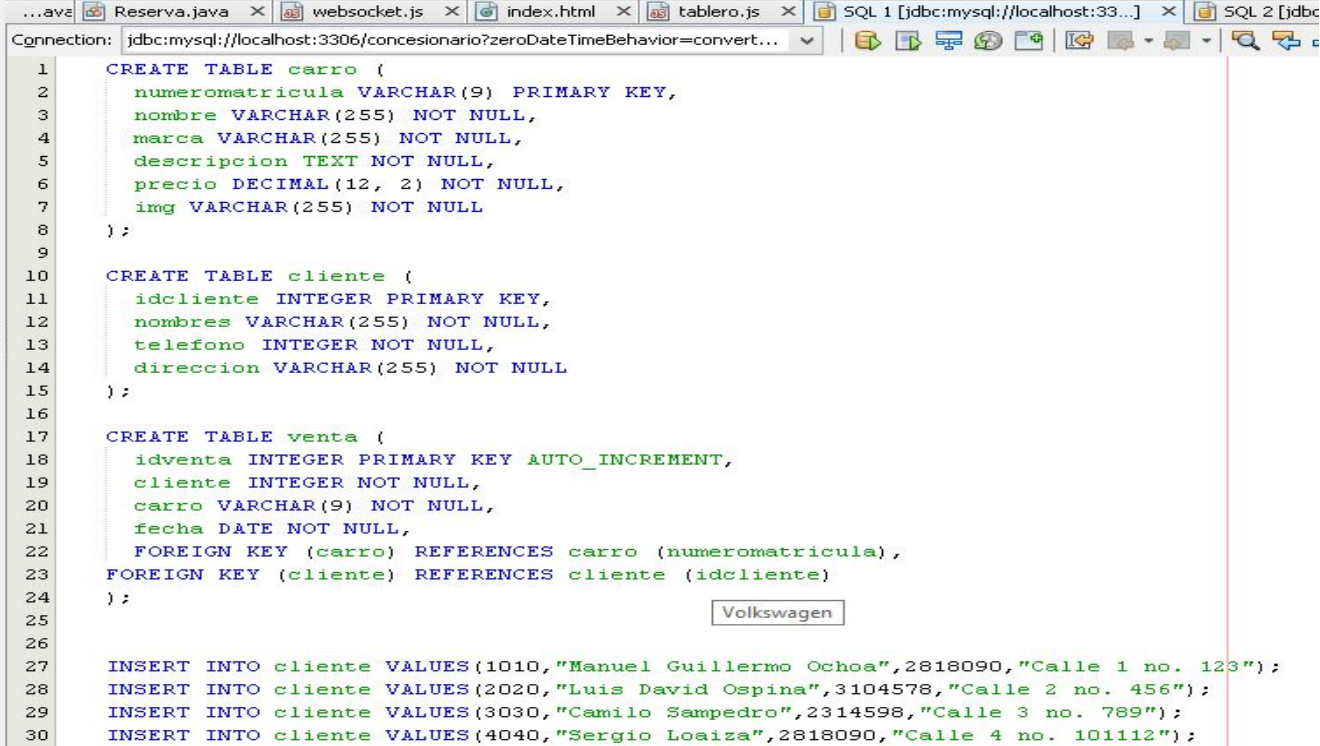
BD MySQL:

Luego creamos una base de datos en MySQL en Netbeans, que se va a llamar concesionario.



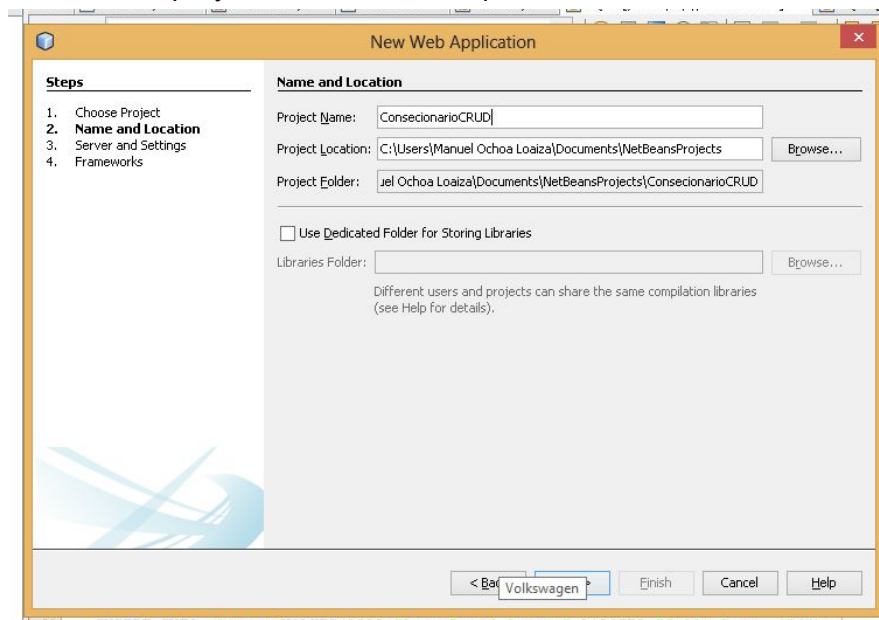
Script MySql:

Luego creamos un script sql, que represente la base de datos y agregamos algunos datos

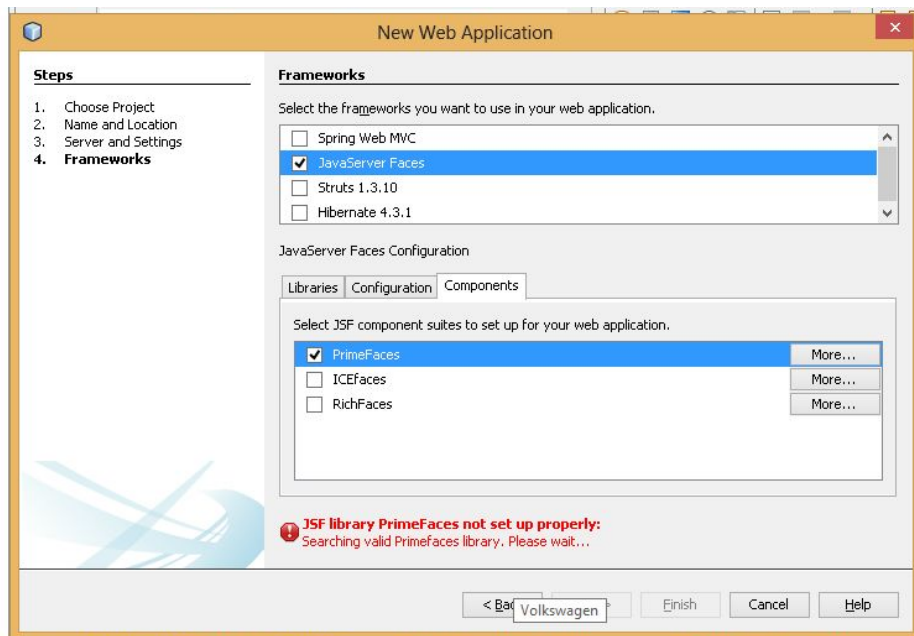


```
1 CREATE TABLE carro (
2     numeromatrícula VARCHAR(9) PRIMARY KEY,
3     nombre VARCHAR(255) NOT NULL,
4     marca VARCHAR(255) NOT NULL,
5     descripción TEXT NOT NULL,
6     precio DECIMAL(12, 2) NOT NULL,
7     img VARCHAR(255) NOT NULL
8 );
9
10 CREATE TABLE cliente (
11     idcliente INTEGER PRIMARY KEY,
12     nombres VARCHAR(255) NOT NULL,
13     telefono INTEGER NOT NULL,
14     direccion VARCHAR(255) NOT NULL
15 );
16
17 CREATE TABLE venta (
18     idventa INTEGER PRIMARY KEY AUTO_INCREMENT,
19     cliente INTEGER NOT NULL,
20     carro VARCHAR(9) NOT NULL,
21     fecha DATE NOT NULL,
22     FOREIGN KEY (carro) REFERENCES carro (numeromatrícula),
23     FOREIGN KEY (cliente) REFERENCES cliente (idcliente)
24 );
25
26
27 INSERT INTO cliente VALUES (1010, "Manuel Guillermo Ochoa", 2818090, "Calle 1 no. 123");
28 INSERT INTO cliente VALUES (2020, "Luis David Ospina", 3104578, "Calle 2 no. 456");
29 INSERT INTO cliente VALUES (3030, "Camilo Sampedro", 2314598, "Calle 3 no. 789");
30 INSERT INTO cliente VALUES (4040, "Sergio Loaiza", 2818090, "Calle 4 no. 101112");
31
```

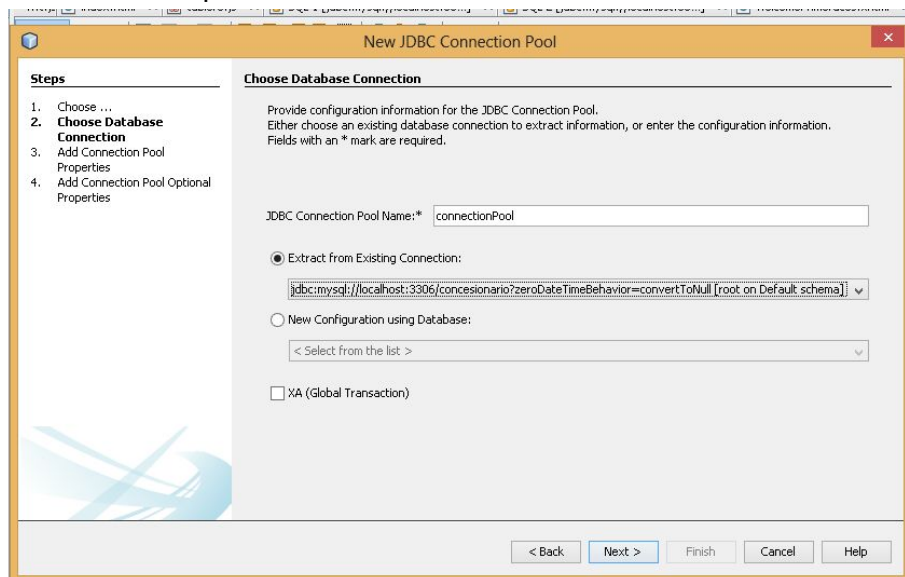
Creamos un proyecto en Java web, que utilice el servidor GlassFish.



Especificamos que vamos a usar PrimeFaces, para la parte de los JSF



Creamos el respectivo JDBC Pool Connection a la base de datos.



Y luego su JDBC Resource.

New JDBC Resource

Steps

1. Choose ...
2. **General Attributes - JDBC Resource**
3. Properties

General Attributes

Provide configuration information for the JDBC Resource.
Either choose an existing JDBC Connection Pool, or create a new JDBC Connection Pool.
Fields with an * mark are required.

☒ Use Existing JDBC Connection Pool

connectionPool

☐ Create New JDBC Connection Pool

JNDI Name:* jdbc/myDatasource

Object Type: user

Enabled: true

Description:

< Back Next > Finish Cancel Help

Procedemos a crear la unidad de persistencia(Entity class from DB).

New Entity Classes from Database

Steps

1. Choose File Type
2. **Database Tables**
3. Entity Classes
4. Mapping Options

Database Tables

Data Source: jdbc/myDatasource

Available Tables:

Selected Tables:

carro
cliente
venta

Add >

< Remove

Add All >>

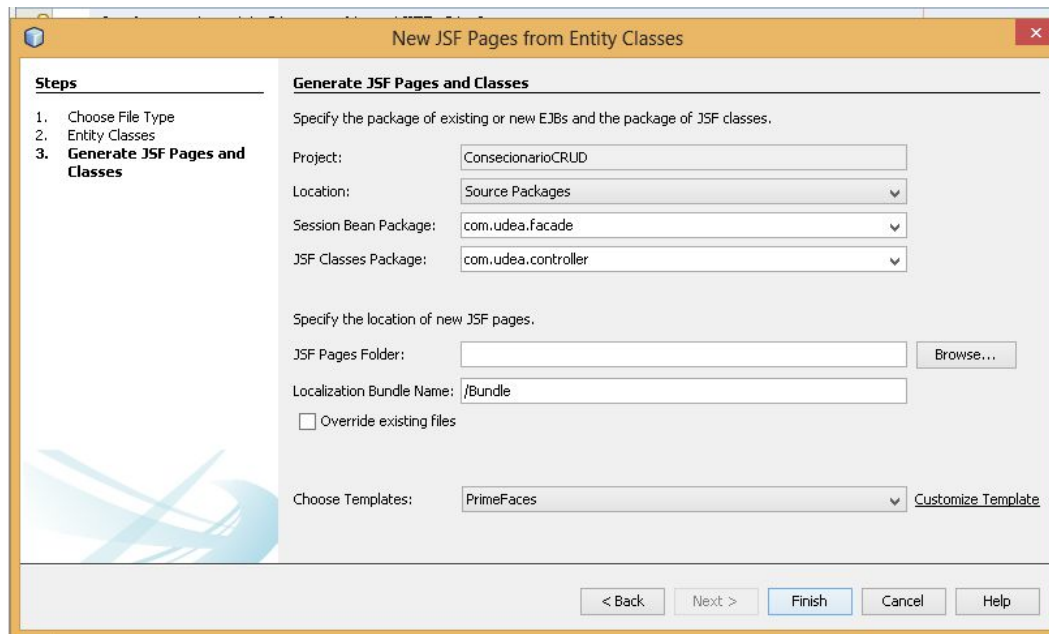
<< Remove All

Any

☒ Include Related Tables

< Back Next > Finish Cancel Help

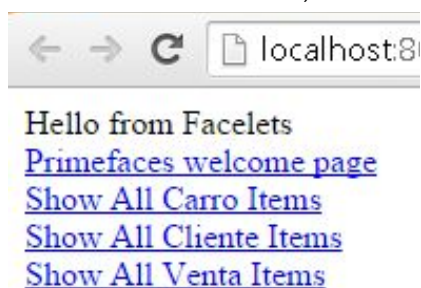
Luego creamos los EJB y los JSF(Facade y controller respectivamente) con la unidad de persistencia ya dada. Y especificamos en *Templates* que usaremos Primefaces.



Doy Clean-Buid y luego en Deploy al proyecto.

```
ava DB Database Process x GlassFish Server x ConseccionarioCRUD (run) x
ant -f "C:\\Users\\Manuel Ochoa Loaiza\\Documents\\NetBeansProjects\\ConseccionarioCRUD\\build.xml"
init:
deps-module-jar:
deps-ear-jar:
deps-jar:
library-inclusion-in-archive:
library-inclusion-in-manifest:
compile:
compile-jsp:
Incrementally deploying ConseccionarioCRUD
Completed incremental distribution of ConseccionarioCRUD
run-deploy:
Browsing: http://localhost:8080/ConseccionarioCRUD/faces/index.xhtml
run-display-browser:
run:
BUILD SUCCESSFUL (total time: 3 seconds)
```

Al correr el index.xhtml, observamos esto.



En la vista de carros tenemos

localhost:8080/ConcesionarioCRUD/faces/carro/List.xhtml

CONCESIONARIOCRUD

Home Maintenance




LIST

Numeromatrícula	Nombre	Marca	Descripción	Precio	Img
DER927	Spark Life	Chevrolet	Chevrolet presenta el nuevo Spark LIFE, un vehículo que llega a complementar el portafolio de productos Chevrolet, con el precio más accesible del segmento de los más pequeños y con el que busca democratizar el carro nuevo en Colombia.	20390000.00	C:\Users\Manuel Ochoa\LoaizaDesktop\imagenes\chevrolet.jpg
GTA820	Mustang GT Premium	Ford	El auto deportivo más emblemático de todos los tiempos, llegó por fin a Colombia para continuar escribiendo su leyenda que ya cumple 50 años, y demostrar por qué su estilo, potencia y tecnología continúan conquistando fanáticos en todos los mercados donde hace presencia. Ahora, tú también podrás comprobarlo.	127490000.00	C:\Users\Manuel Ochoa\LoaizaDesktop\imagenes\ford.jpg
LAS123	Mégane III Hatchback	Renault	Mégane III tiene un diseño enamorado, verdadera promesa de diseño y robustez. Faros delanteros perfilados, líneas fluidas y propulsoras, equilibrio de las proporciones... No esconde nada de sus excelentes cualidades de ruta. Sorprende tanto por la precisión de sus líneas como por la calidad de sus acabados. Con Mégane III, el diseño es una forma	55990000.00	C:\Users\Manuel Ochoa\LoaizaDesktop\imagenes\mazda.jpg




En este último y en el resto, debemos hacer que la imagen no sea un enlace, si no que se vea como una imagen como tal, para esto en el xml, cambiamos la manera como se muestra.



Queda aclarar que las imágenes, están en la carpeta *imagenes*, dentro de la carpeta *web*, del proyecto, y por tanto la base de datos lo único que guarda es un string, redireccionando las páginas xhtml a las carpetas dentro del proyecto. Para verlo a manera de servidor, las páginas se guardarían en el servidor y lo único que se haría sería llamarlos. Aquí se ve la página List.xhtml de los carros, con sus respectivas imágenes

CONCESIONARIOCRUD					
Home Maintenance					
LIST					
1 10					
Numeromatrícula	Nombre	Marca	Descripción	Precio	Imagen
DER927	Spark Life	Chevrolet	Chevrolet presenta el nuevo Spark LIFE, un vehículo que llega a complementar el portafolio de productos Chevrolet, con el precio más accesible del segmento de los más pequeños y con el que busca democratizar el carro nuevo en Colombia.	20390000.00	
GTA820	Mustang GT Premium	Ford	El auto deportivo más emblemático de todos los tiempos, llegó por fin a Colombia para continuar escribiendo su leyenda que ya cumple 50 años, y demostrar por qué su estilo, potencia y tecnología continúan conquistando fanáticos en todos los mercados donde hace presencia. Ahora, tú también podrás comprobarlo.	127490000.00	
			Mégane III tiene un diseño enamorado, verdadera promesa de diseño y robustez. Faros delanteros perfilados, líneas fluidas y propulsoras,		

Estas serían las ventas por fecha.

CONCESIONARIOCRUD					
Home Maintenance					
LIST					
1 10					
Idventa	Fecha	Carro	Cliente	Idcliente	Imagen
1209	03/12/2015	DER927	Sergio Loaiza	4040	
1345	01/10/2015	PER789	Luis David Ospina	2020	
					

Agregamos un filtro en la parte de matrículas, cuando listamos los carros. Así se podrán buscar por su número de matrícula o parte de ella.


localhost8080/ConcesionarioCRUD/faces/carro/List.xhtml

CONCESIONARIOCRUD

Home Maintenance

LIST

1 10

Numeromatrícula	Nombre	Marca	Descripcion	Precio	Imagen
> DER927	Spark Life	Chevrolet	Chevrolet presenta el nuevo Spark LIFE, un vehículo que llega a complementar el portafolio de productos Chevrolet, con el precio más accesible del segmento de los más pequeños y con el que busca democratizar el carro nuevo en Colombia.	20390000.00	

1 10

Create View Edit Delete


localhost8080/ConcesionarioCRUD/faces/carro/List.xhtml

CONCESIONARIOCRUD

Home Maintenance

LIST

1 10

Numeromatrícula	Nombre	Marca	Descripcion	Precio	Imagen
> GTA820	Mustang GT Premium	Ford	El auto deportivo más emblemático de todos los tiempos, llegó por fin a Colombia para continuar escribiendo su leyenda que ya cumple 50 años, y demostrar por qué su estilo, potencia y tecnología continúan conquistando fanáticos en todos los mercados donde hace presencia. Ahora, tu también podrás comprobarlo.	127490000.00	

1 10

Create View Edit Delete

Mediante, una expresión de la clase search de primefaces, expresada de la siguiente manera en el.xhtml.

Projects Services Files

Source History

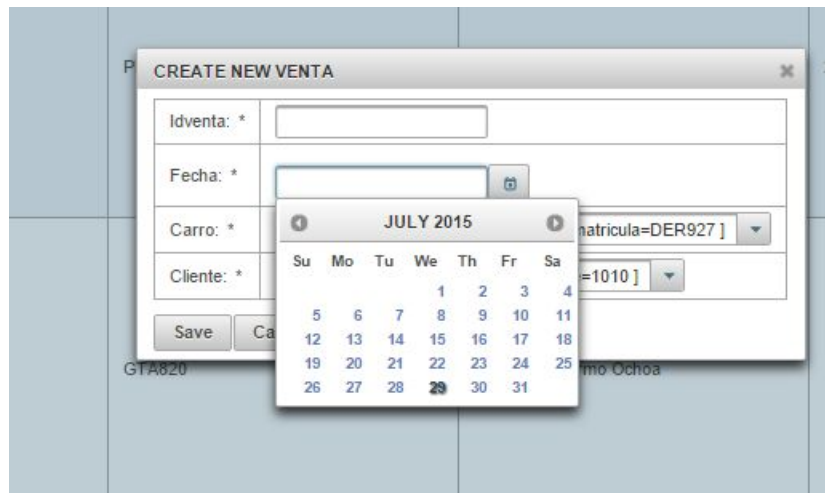
```

24
25 rows="10"
26 rowsPerPageTemplate="10,20,30,40,50"
27
28
29 <p:ajax event="rowSelect" update="createButton viewButton editButton deleteButton"/>
30 <p:ajax event="rowUnselect" update="createButton viewButton editButton deleteButton"/>
31
32 <p:column filterBy="#{item.numeromatrícula}" headerText="Matrícula" filterMatchMode="contains">>
33     <f:facet name="header">
34         <h:outputText value="#{bundle.ListCarroTitle_numeromatrícula}"/>
35     </f:facet>
36     <h:outputText value="#{item.numeromatrícula}"/>
37 </p:column>
38 <p:column>
39     <f:facet name="header">
40         <h:outputText value="#{bundle.ListCarroTitle_nombre}"/>

```

Output

En la parte de crear una venta, el primefaces tiene por defecto implementado sus gadgets correspondientes para la fecha.



Como se observa en la anterior, en la parte de cliente y carro, se veía una dirección de código que no debe ser visible para el usuario, por tanto cambiando el método `toString()`, de la unidad de persistencia en carro y cliente, podemos hacer que demuestre todo como se debe.

```

70      }
71
72      @Override
73      public String toString() {
74          return numeromatrícula;
75      }
76

```

```

template.xhtml
welcomePrimefaces.xhtml
ace Packages
<default package>
com.udea.controller
com.udea.controller.util
com.udea.facede
com.udea.persistencia
Carro.java
Cliente.java
Venta.java
aries
 Surprise Beans
figuration Files

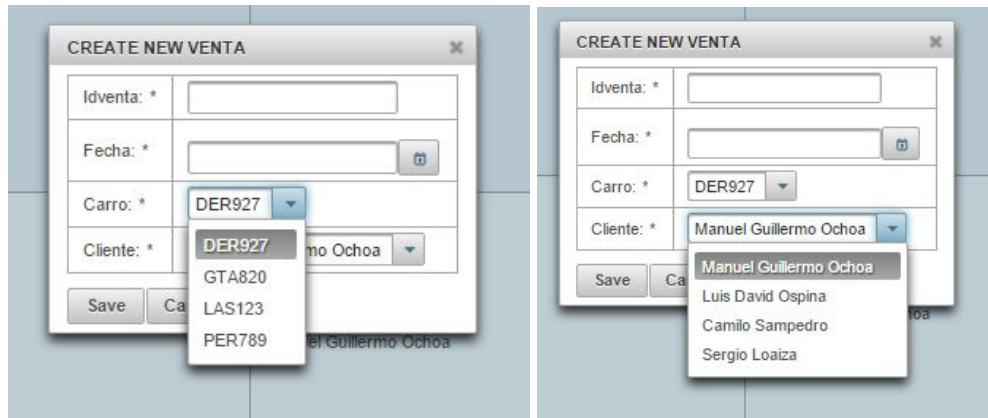
```

```

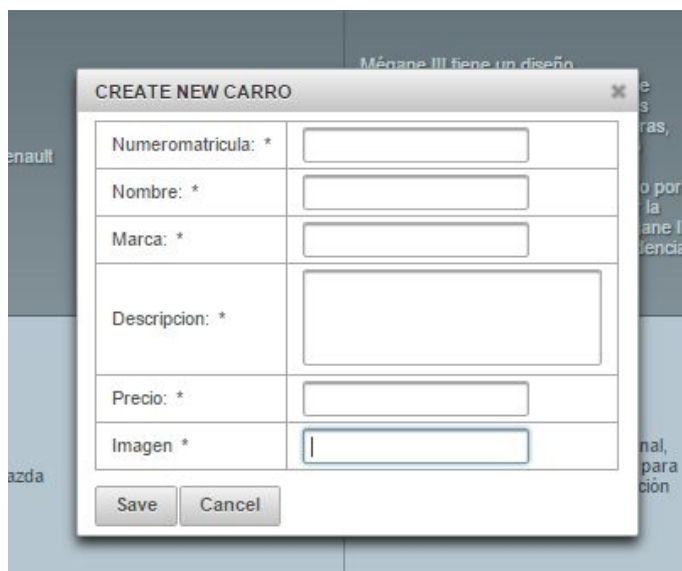
134
135
136
137      @Override
138      public String toString() {
139          return nombres;
140      }
141
142

```

Para así tener algo más presentable.



Solo queda verificar, que no fuimos capaces con el manejo del archivo y por lo tanto no se implementó. Aunque intentamos con la forma automatica, que tiene primefaces en su documentación. Pero no nos dio.



Conclusiones

- Es mucho más sencillo, crear una aplicación tipo CRUD, si se tiene la base de datos bien diseñada.
- El CRUD sirve para el manejo de información en general y cualquier tipo de información puede ser manejada allí.
- Primefaces, es un componente tipo JSF muy útil, ya que posee herramientas que permiten el fácil desarrollo del FrontEnd.
- No es recomendable manejar imágenes en bases de datos, es mejor tener las imágenes en el servidor y llamarlas con direcciones url.
- Se aprendió a la fuerza, que es mejor tener la carpeta de imágenes dentro del mismo proyecto y/o servidor.
- Queda para futuro implementar el manejo de subida de archivos. Y su respectivo manejo desde el frontend, hasta la base de datos.
- Primefaces tiene una amplia documentación, solo es probar y probar hasta que algo suceda.