****

TRABAJO PRÁCTICO INTEGRADOR

“Sistema de Reservas”

JAVA

Docentes: Tabacman, Ricardo – Meca, Adrián

Comisión:

Integrantes:

* Miño, Mauricio – 41790
* Rueda, Juan - 41841

Índice

[Introducción 4](#_Toc499325491)

[Diagrama de Clases 5](#_Toc499325492)

[Modelo de Datos 6](#_Toc499325493)

[CU resumen re-estructurado 7](#_Toc499325494)

[CU de usuario principales sin reestructurar 9](#_Toc499325495)

[CUU1 - Registrar nueva reserva 9](#_Toc499325496)

[CUU2 - Cancelar Reserva 11](#_Toc499325497)

[Capturas de Pantalla del Sistema 12](#_Toc499325498)

[Login 12](#_Toc499325499)

[Pantalla Principal 13](#_Toc499325500)

[Admin 13](#_Toc499325501)

[Usuario 13](#_Toc499325502)

[Personas 14](#_Toc499325503)

[Alta de Persona 14](#_Toc499325504)

[Lista de Personas 14](#_Toc499325505)

[Baja y Modificación de Persona 15](#_Toc499325506)

[Elementos 16](#_Toc499325507)

[Alta de Elemento 16](#_Toc499325508)

[Lista de Elementos 16](#_Toc499325509)

[Baja y Modificación de Elementos 17](#_Toc499325510)

[Tipo de Elemento 18](#_Toc499325511)

[Alta de Tipo de Elemento 18](#_Toc499325512)

[Lista de Tipos de Elementos 18](#_Toc499325513)

[Baja y Modificación de Tipos de Elementos 19](#_Toc499325514)

[Reservas 20](#_Toc499325515)

[Alta Reservas I 20](#_Toc499325516)

[Alta Reservas II 20](#_Toc499325517)

[Lista Reservas – Admin 21](#_Toc499325518)

[Lista Reservas – Usuario 21](#_Toc499325519)

[Fragmentos del Código Fuente 22](#_Toc499325520)

[CUU1 – Registrar nueva reserva 22](#_Toc499325521)

[BookableItems.java 22](#_Toc499325522)

[CtrlBookableItems.java 23](#_Toc499325523)

[DataBookableItems.java 25](#_Toc499325524)

[SeleccionTipoEle.jsp 31](#_Toc499325525)

[SeleccionarTipoElemento.java 33](#_Toc499325526)

[altaReserva.jsp 34](#_Toc499325527)

[AltaReserva.java 36](#_Toc499325528)

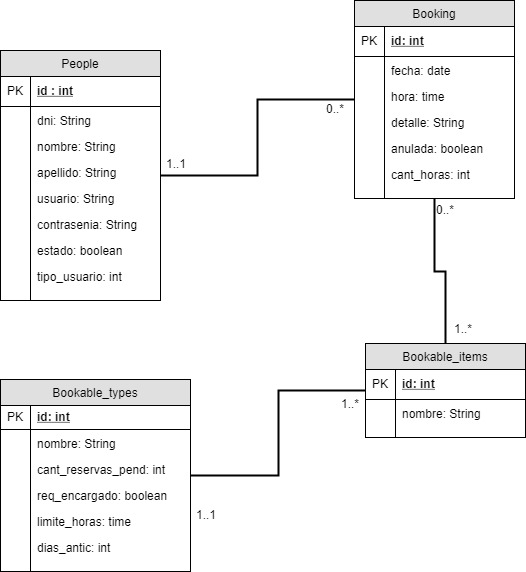
[reservas.jsp 38](#_Toc499325529)

# Introducción

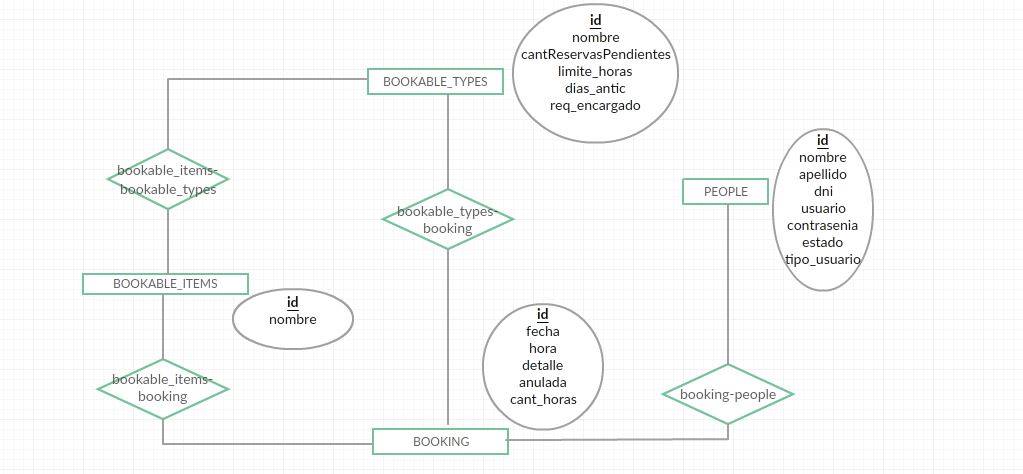
El siguiente informe corresponde al *Trabajo Práctico Integrador*, desarrollado en la cátedra JAVA, basado en un *Sistema de Reservas.* A continuación, se detallarán diversos artefactos como ser: diagrama de clases, casos de uso de usuario,entre otros.

Posteriormente, presentaremos un fragmento de código para poder explicarlo y las correspondientes capturas de pantalla del sistema en su totalidad.

# Diagrama de Clases



# Modelo de Datos



# CU resumen re-estructurado

**Código y Nombre del CASO DE USO:** CUR Realizar reserva

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nivel** | **Estructura** | **Alcance** | **Caja** | **Instanciación** | **Interacción** |
| **Resumen** | Re-estructurado | Negocio | Negra | Real | Semántico |

**Meta del CASO DE USO:** Reservar un elemento

**ACTORES**

**Primario:** Persona

**Otros:** Administrador

**PRECONDICIONES** **(de negocio):** Existen elementos disponibles para reservar

**DISPARADOR:** Persona desea realizar una reserva

**FLUJO DE SUCESOS**

**CAMINO BÁSICO:**

1. El administrador registra el alta de una persona invocando <CUU1 Agregar personas>
2. El administrador agrega un nuevo elemento invocando <CUU3 Agregar elemento>
3. El administrador agrega un nuevo tipo de elemento invocando <CUU6 Agregar tipo de elemento>
4. La persona realiza una reserva invocando <CUU9 Registrar nueva reserva>

**CAMINOS ALTERNATIVOS:**

1.a <reemplaza> La persona solicita ser eliminada del sistema

1.a.1 El administrador realiza baja de una persona invocando <CUU2 Eliminar personas>

1.b <reemplaza> Es necesario cambiar algún dato de la persona

1.b.1 El administrador modifica los datos de una persona invocando <CUU3 Modificar personas>

2.a <reemplaza> Se solicita eliminar un elemento existente

2.a.1 El administrador elimina un elemento existente invocando <CUU4 Eliminar elemento>

2.b <reemplaza> Es necesario cambiar algún dato del elemento

2.b.1 El administrador modifica un elemento existente invocando <CUU5 Modificar elemento>

3.a <reemplaza> Se solicita eliminar un tipo de elemento existente

3.a.1 El administrador elimina un tipo de elemento existente invocando <CUU7 Eliminar tipo de elemento>

3.b <reemplaza> Es necesario cambiar algún dato del tipo de elemento

3.b.1 El administrador modifica un tipo de elemento existente invocando <CUU8 Modificar tipo de elemento>

4.a <posterior> La persona desea cancelar una reserva

4.a.1 La persona cancela una reserva invocando <CUU10 Cancelar reserva>

4.b <posterior> Se solicita la anulación de la reserva

4.b.1 El administrador realiza la anulación de la reserva invocando <CUU11 Anular reserva>

**POSTCONDICIONES (de negocio)**

**Éxito:** Reserva realizada

**Fracaso:** Reserva no realizada

**Éxito alternativo:** <vacío>

# CU de usuario principales sin reestructurar

## CUU1 - Registrar nueva reserva

**Código y Nombre del CASO DE USO:** CUU1 - Registrar nueva reserva

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nivel** | **Estructura** | **Alcance** | **Caja** | **Instanciación** | **Interacción** |
| Usuario | Sin-Estructurar | Sistema | Negra | Real | Semántico |

**Meta del CASO DE USO:** Reservar un elemento.

**ACTORES**

**Primario:** Persona (es decir, Usuario y Encargado).

**Otros:** Sistema.

**PRECONDICIONES** **(de sistema):**

La persona está logueada en el sistema.

La persona está habilitada para realizar reservas.

**DISPARADOR:** Persona solicita realizar una nueva reserva.

**FLUJO DE SUCESOS**

**CAMINO BÁSICO:**

1. El sistema informa los tipos de elementos disponibles para la Persona.
2. La Persona selecciona un tipo de elemento. El sistema valida la opción seleccionada, la registra y muestra los elementos disponibles para ese tipo de elemento, los días y los horarios disponibles.
3. La Persona selecciona un elemento a reservar, elige una fecha y horario a reservar y la cantidad de horas. El sistema valida los datos seleccionados, registra la reserva y lo informa por email.

**CAMINOS ALTERNATIVOS:**

3.a <durante> La fecha y hora seleccionados superan el límite máximo para reservar.

3.a.1 El sistema lo informa

3.a.2 La persona selecciona un período válido. El sistema valida la selección, registra la reserva y lo informa por email.

3.b <durante> No se pueden realizar más cantidad de reservas del tipo de elemento seleccionado

3.b.1 El sistema lo informa. FCU

3.c <durante> La cantidad de horas ingresadas supera la máxima o la mínima para ese tipo de elemento.

3.c.1 El sistema informa la situación.

3.c.2 La Persona ingresa una cantidad de horas válidas. El sistema registra la reserva y lo informa por email.

**POSTCONDICIONES (de negocio)**

**Éxito:** Reserva registrada en el sistema

**Fracaso:** Reserva no registrada

**Éxito alternativo:** <vacío>

## CUU2 - Cancelar Reserva

**Código y Nombre del CASO DE USO:** CUU2 Cancelar Reserva

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Nivel** | **Estructura** | **Alcance** | **Caja** | **Instanciación** | **Interacción** |
| **Usuario** | Sin-estructurar | Sistema | Negra | Real | Semántico |

**Meta del CASO DE USO:** Cancelar una reserva pendiente.

**ACTORES**

**Primario:** Persona (es decir, Usuario y Encargado).

**Otros:** Sistema.

**PRECONDICIONES** **(de sistema):**

Persona habilitada.

Persona logueada en el sistema.

**DISPARADOR:** Persona desea cancelar su reserva.

**FLUJO DE SUCESOS**

**CAMINO BÁSICO:**

1. La Persona solicita sus reservas pendientes. El sistema muestra las reservas pendientes para ese usuario.
2. La Persona selecciona una de las reservas y la elimina. El sistema lo registra, elimina la reserva y lo informa por email a la Persona.

**CAMINOS ALTERNATIVOS:**

1.a <durante> La persona no posee reservas pendientes. FCU

**POSTCONDICIONES (de negocio)**

**Éxito:** Cancelación de reserva registrada.

**Fracaso:** Cancelación de reserva no registrada.

**Éxito alternativo:** <vacío>

# Capturas de Pantalla del Sistema

## Login





## Pantalla Principal

### Admin

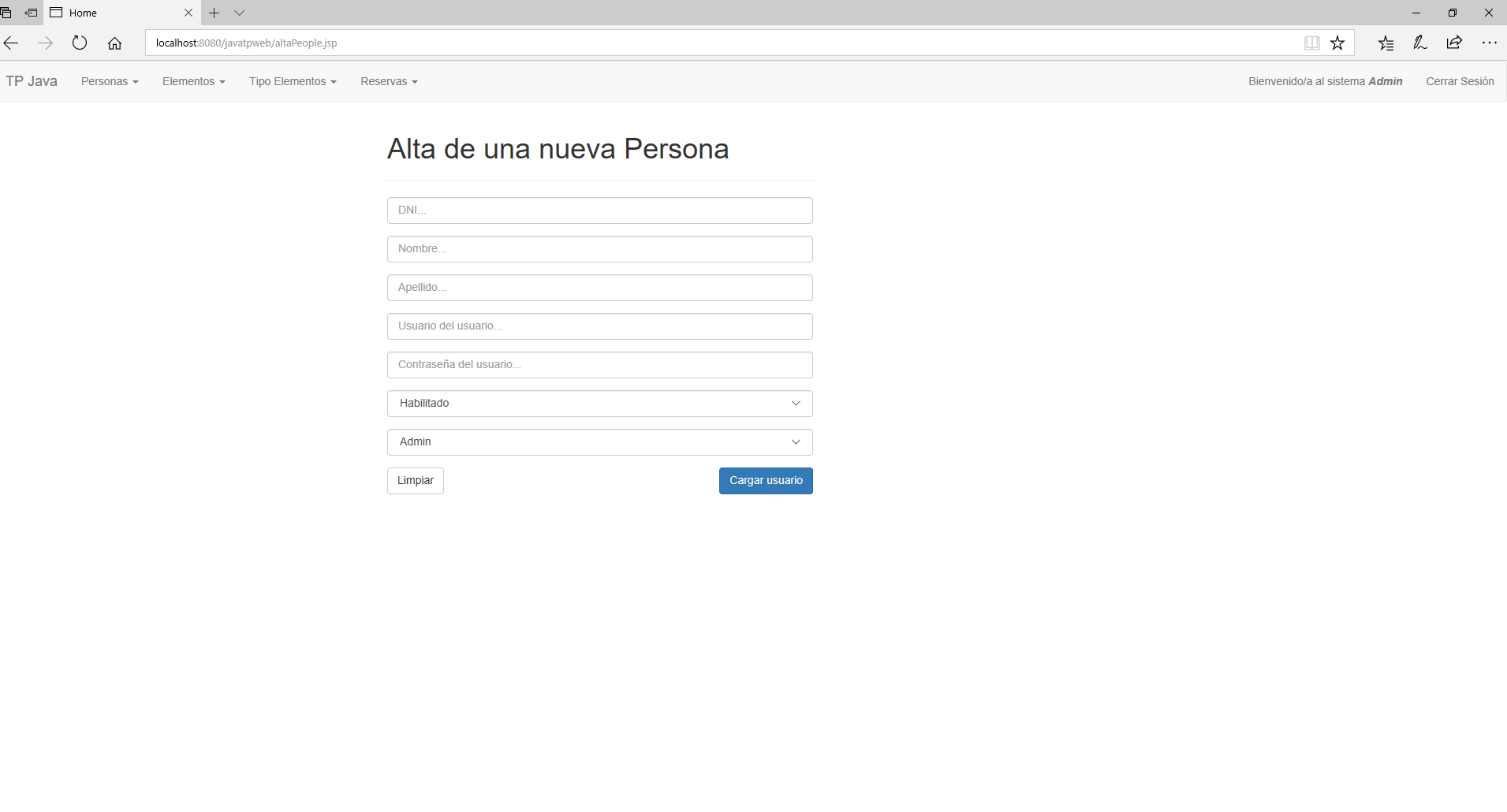


### Usuario

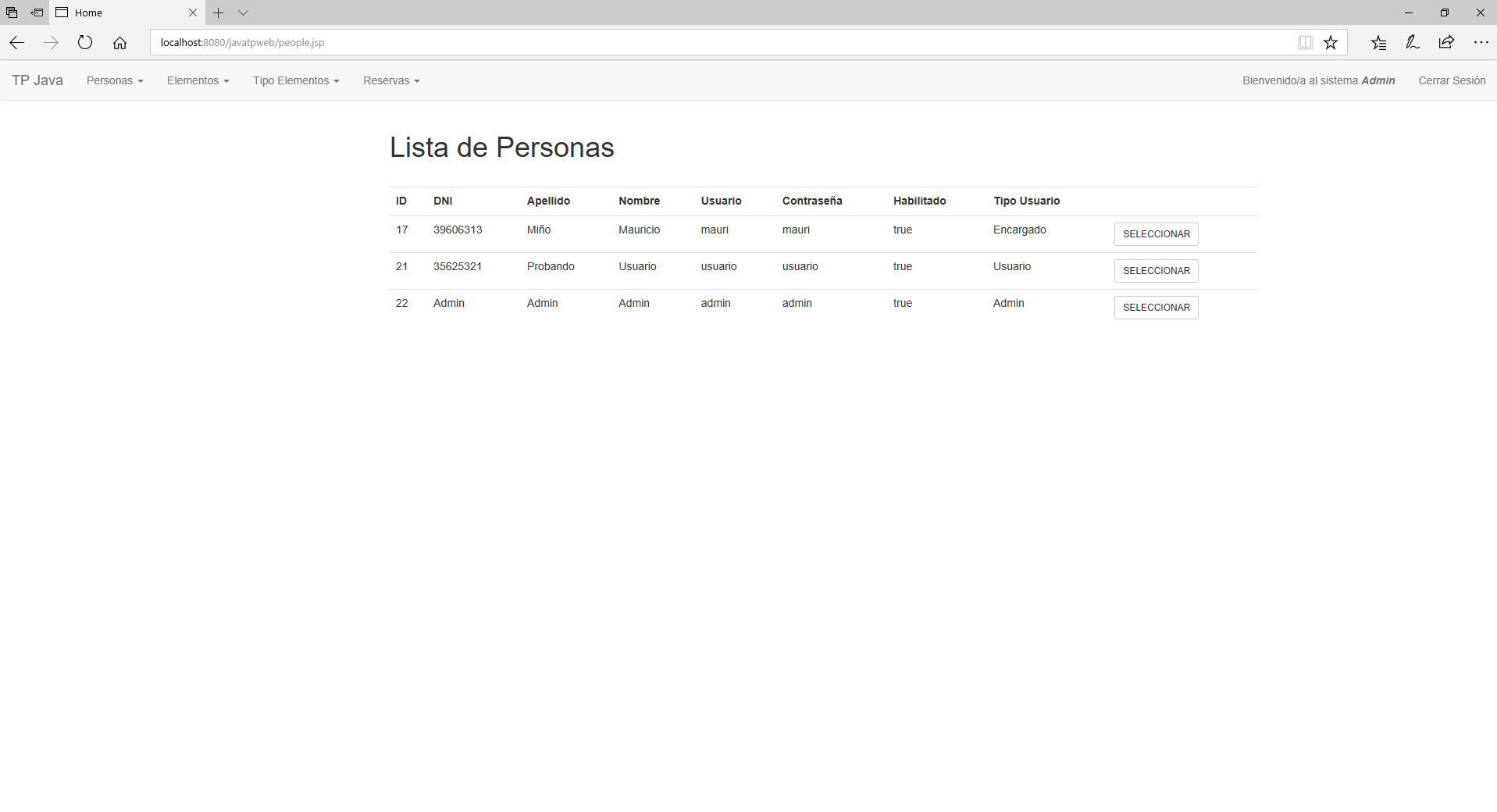


## Personas

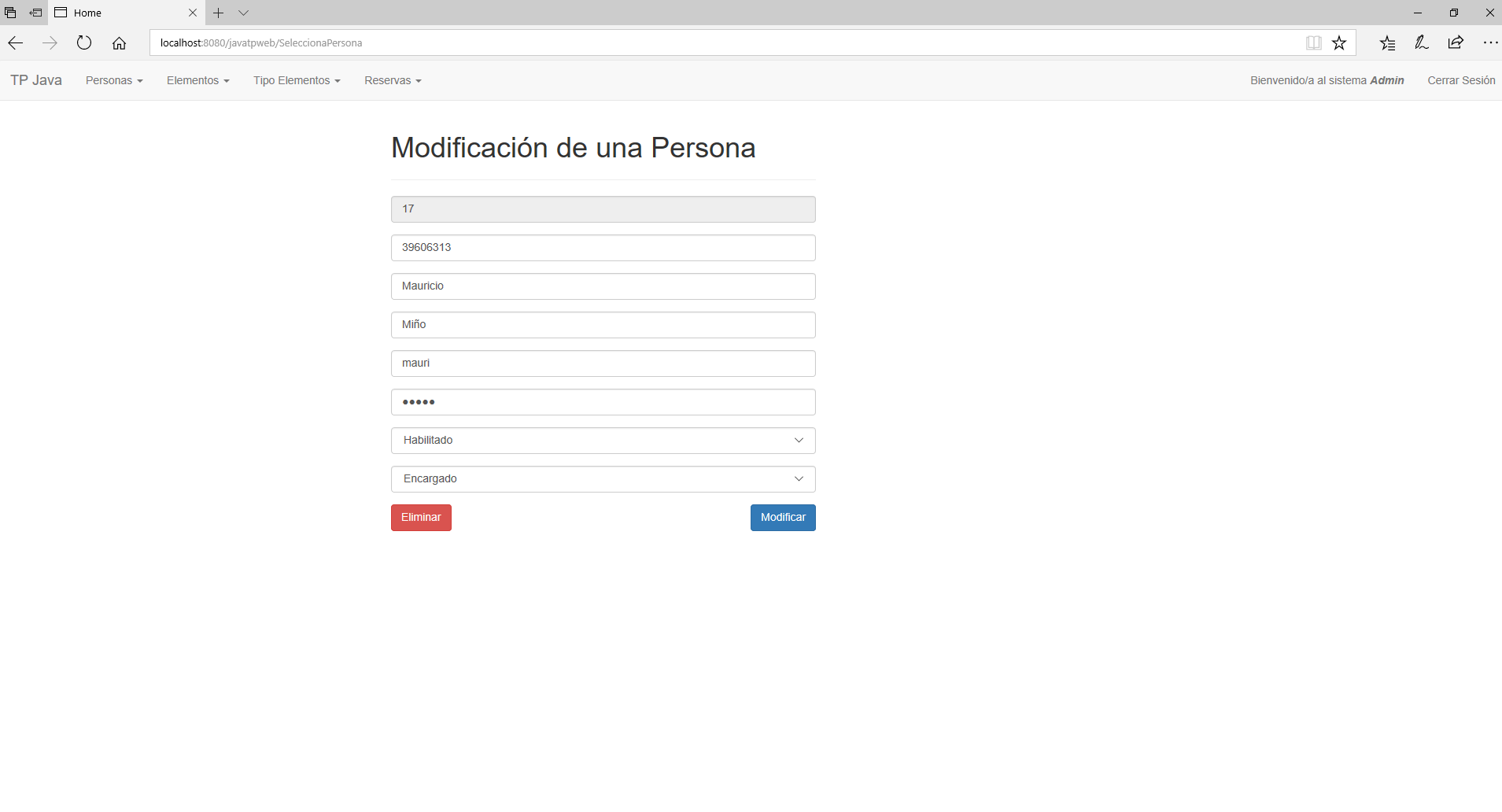
### Alta de Persona



### Lista de Personas

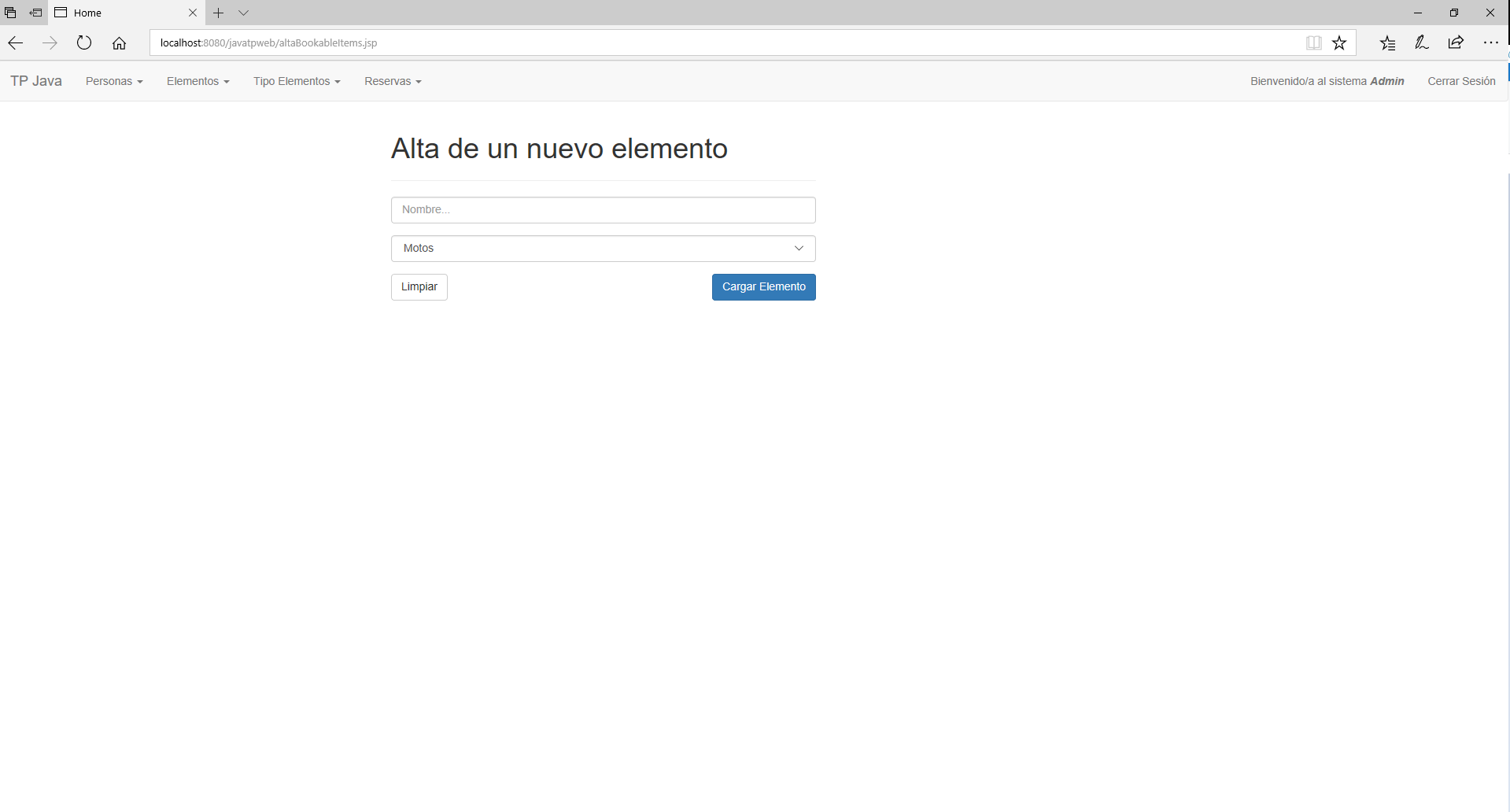


### Baja y Modificación de Persona

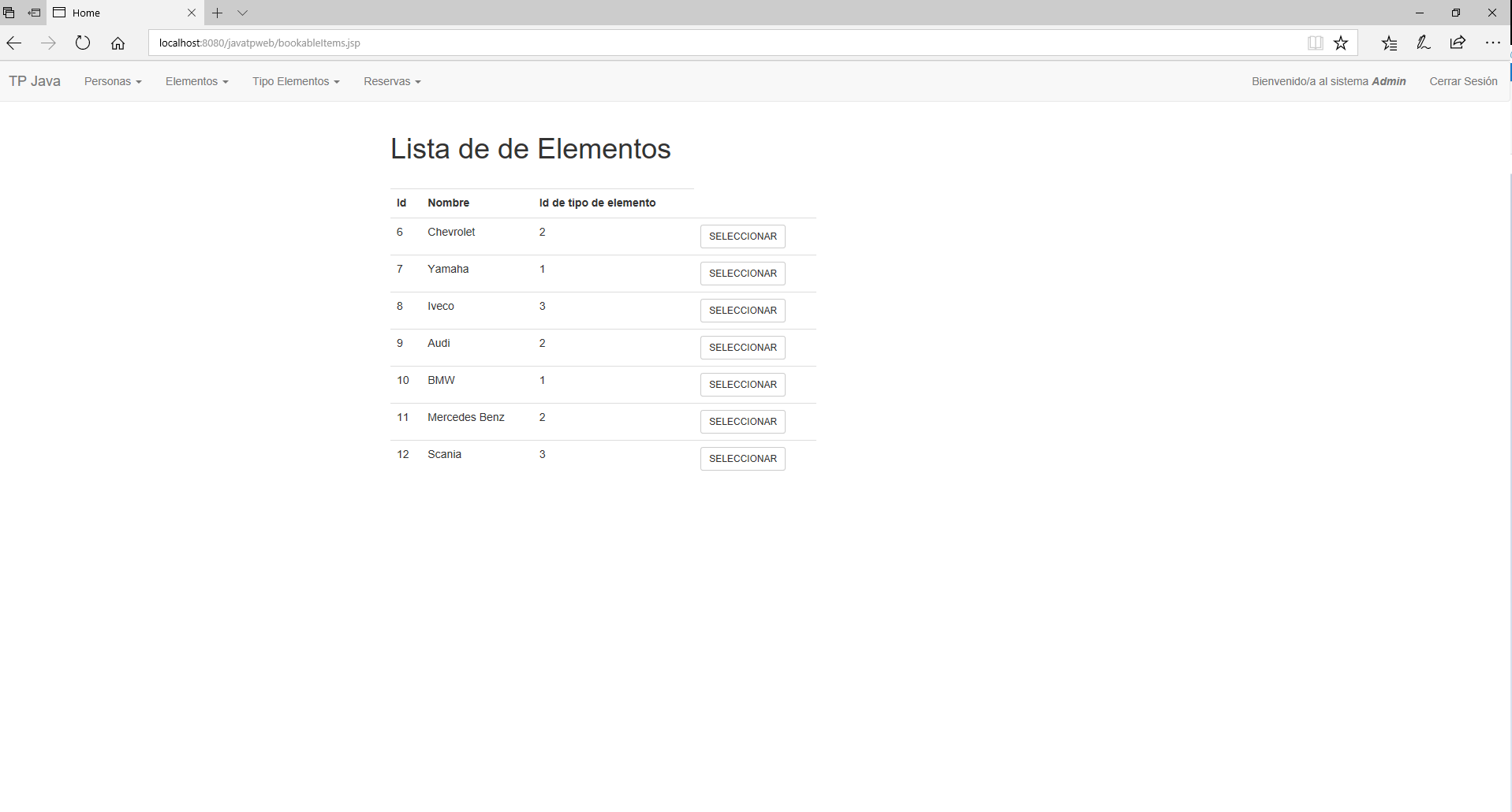


## Elementos

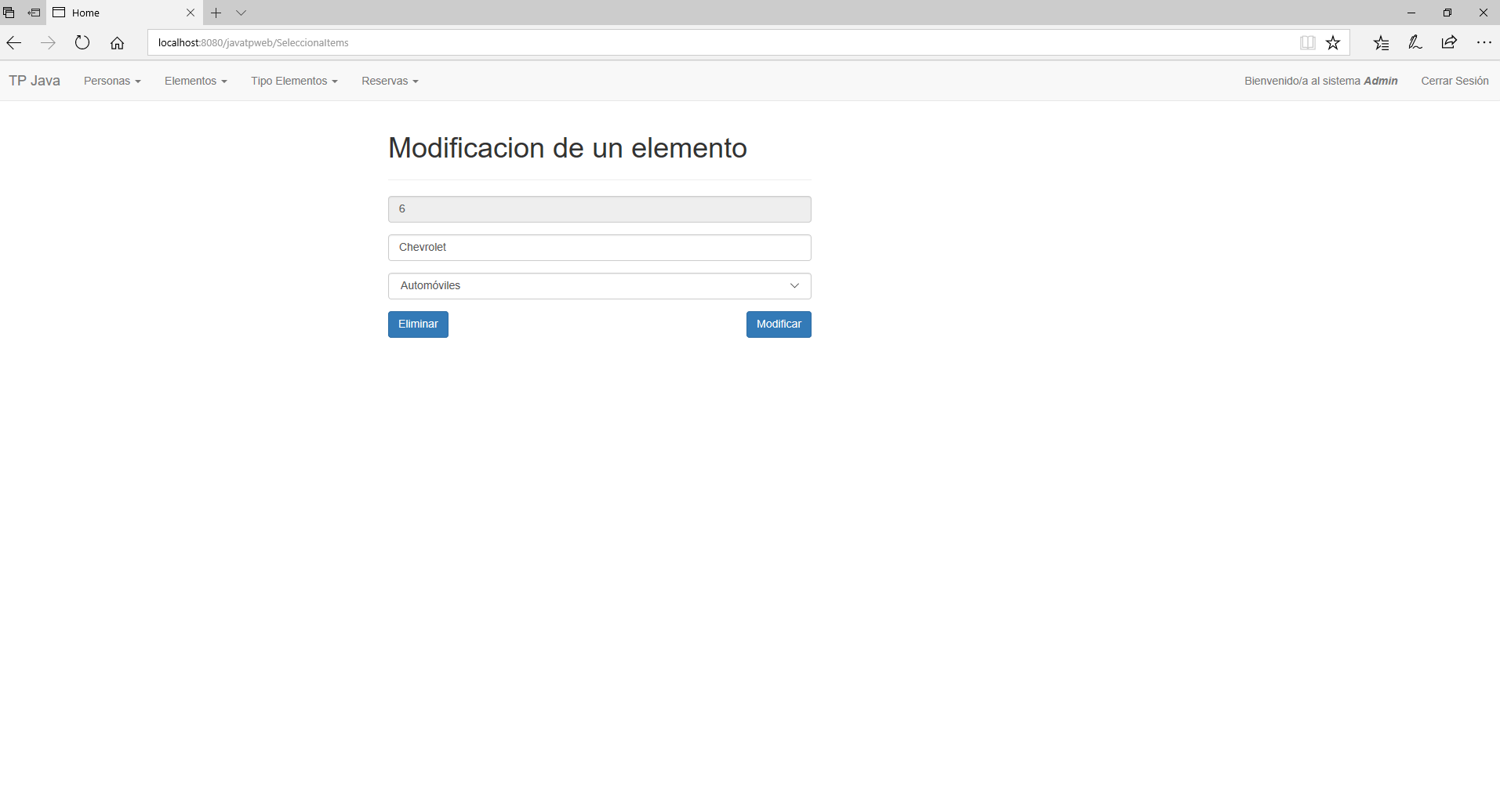
### Alta de Elemento



### Lista de Elementos

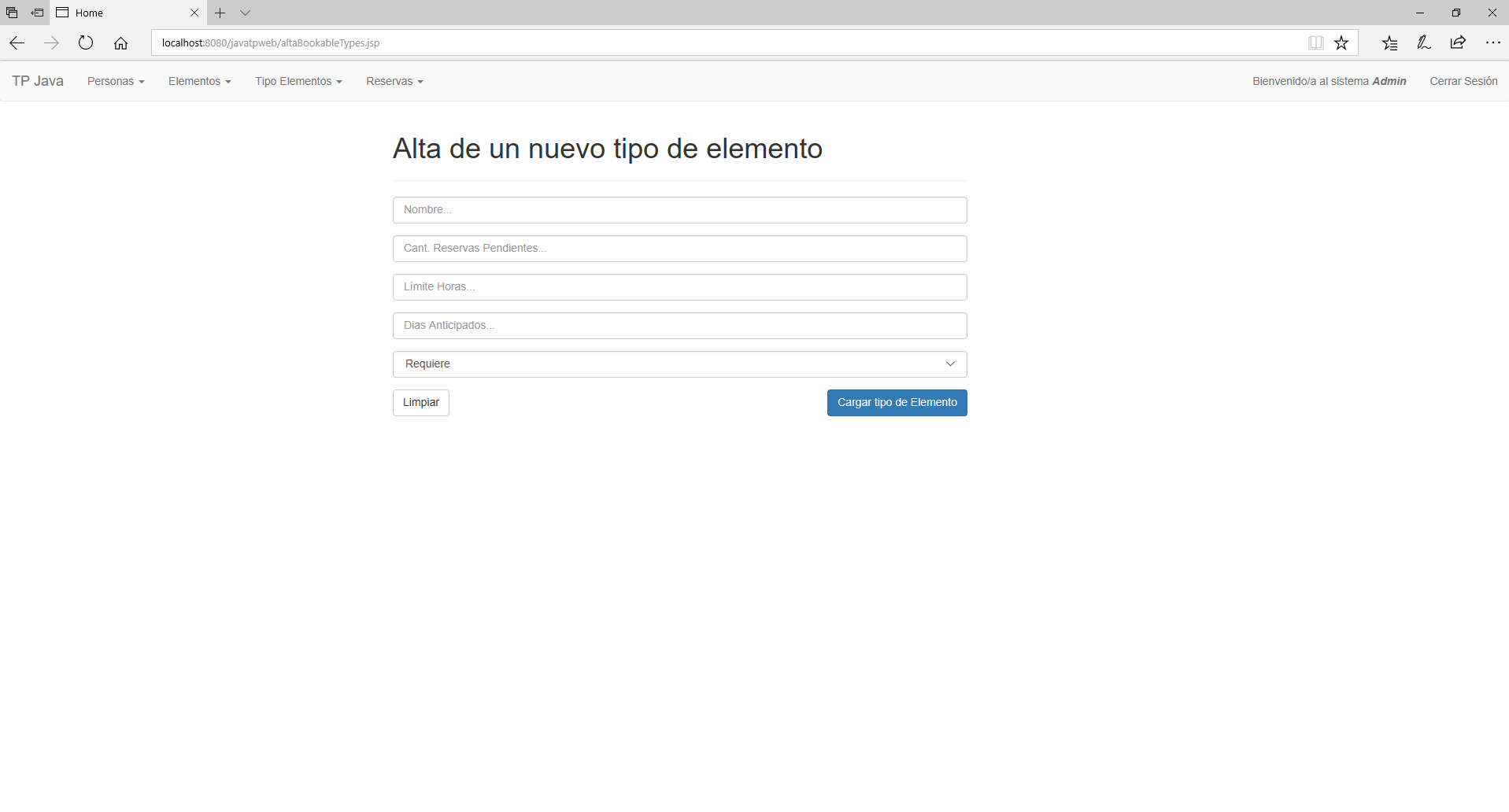


### Baja y Modificación de Elementos

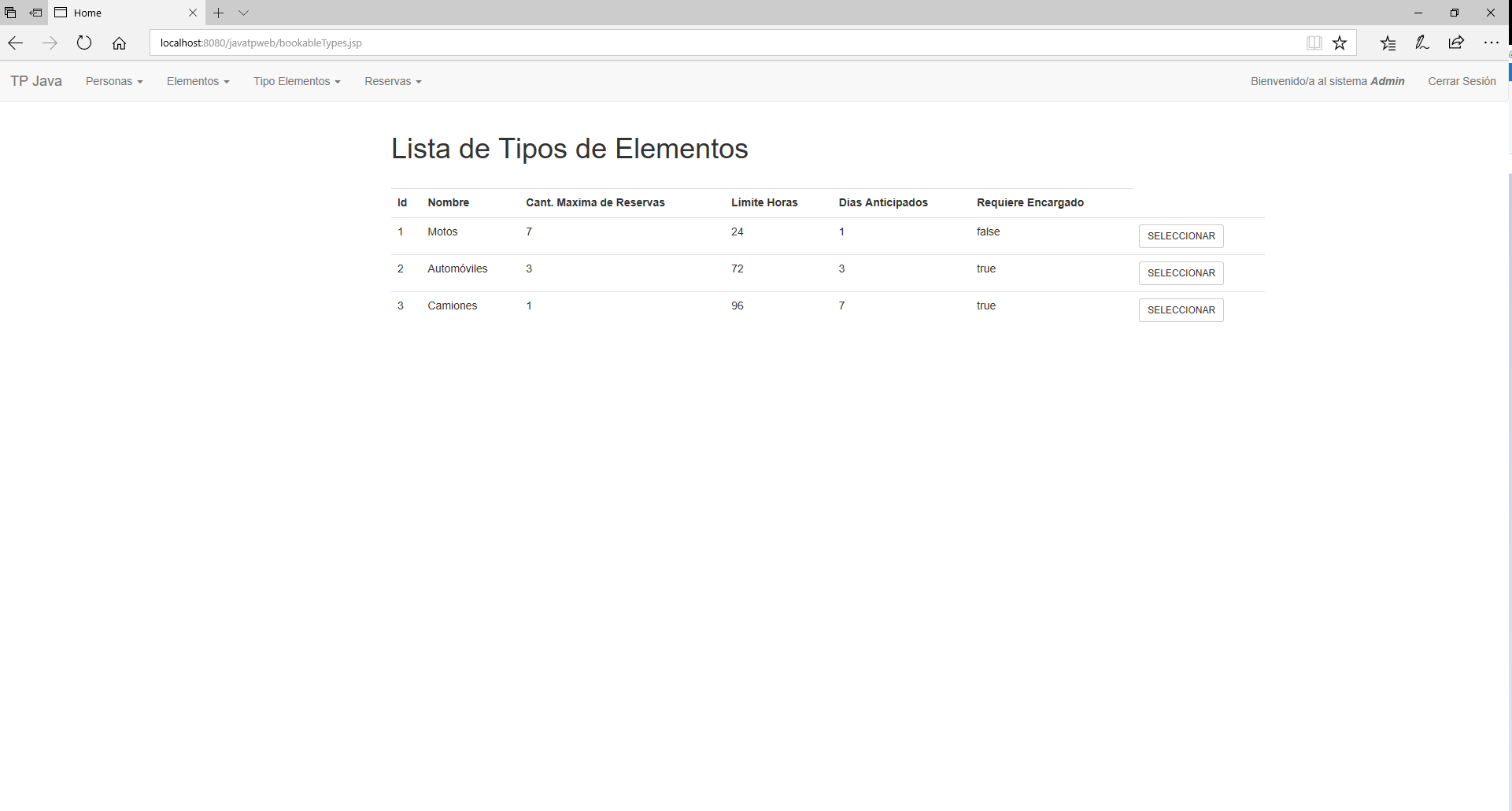


## Tipo de Elemento

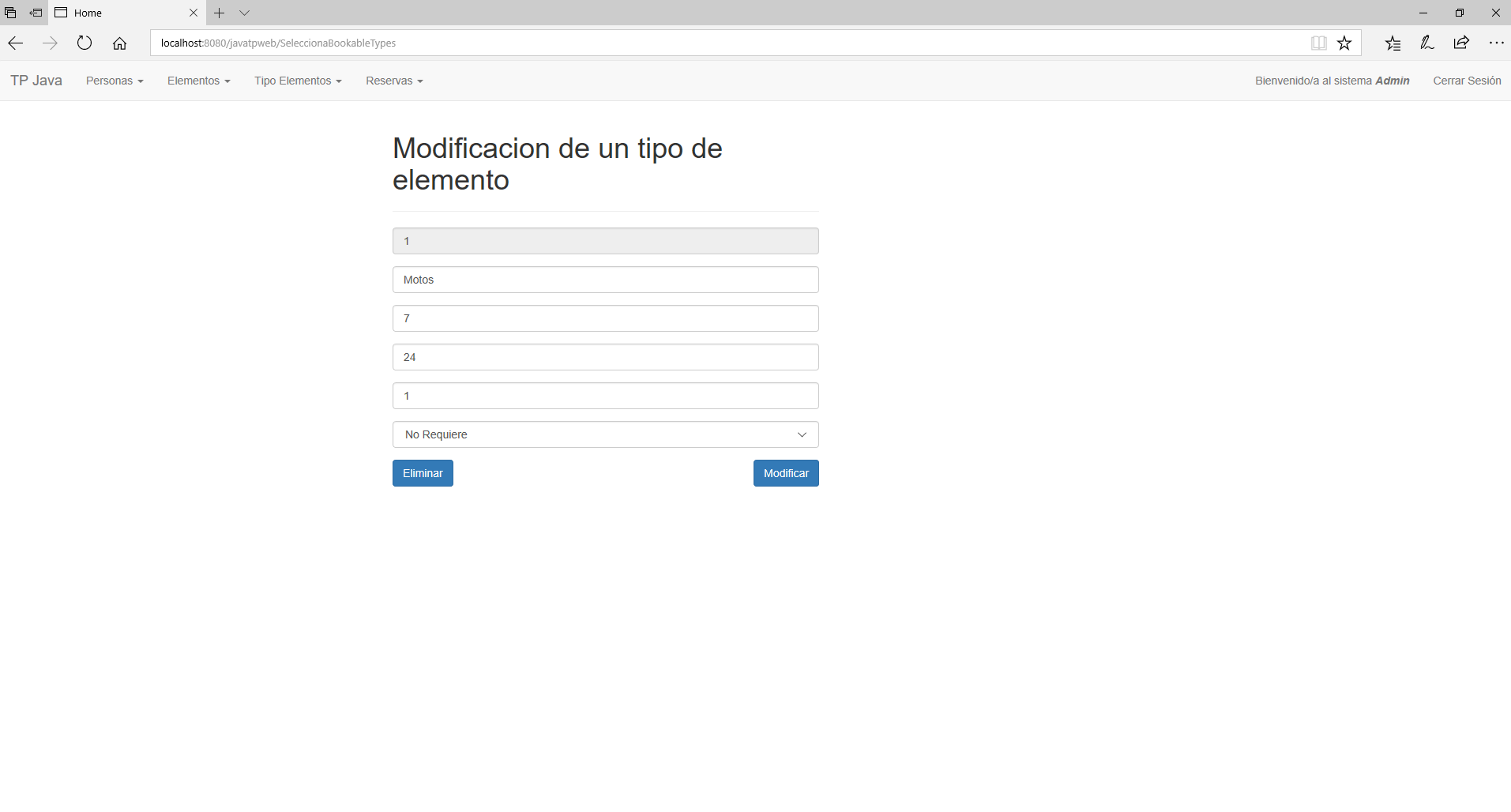
### Alta de Tipo de Elemento



### Lista de Tipos de Elementos

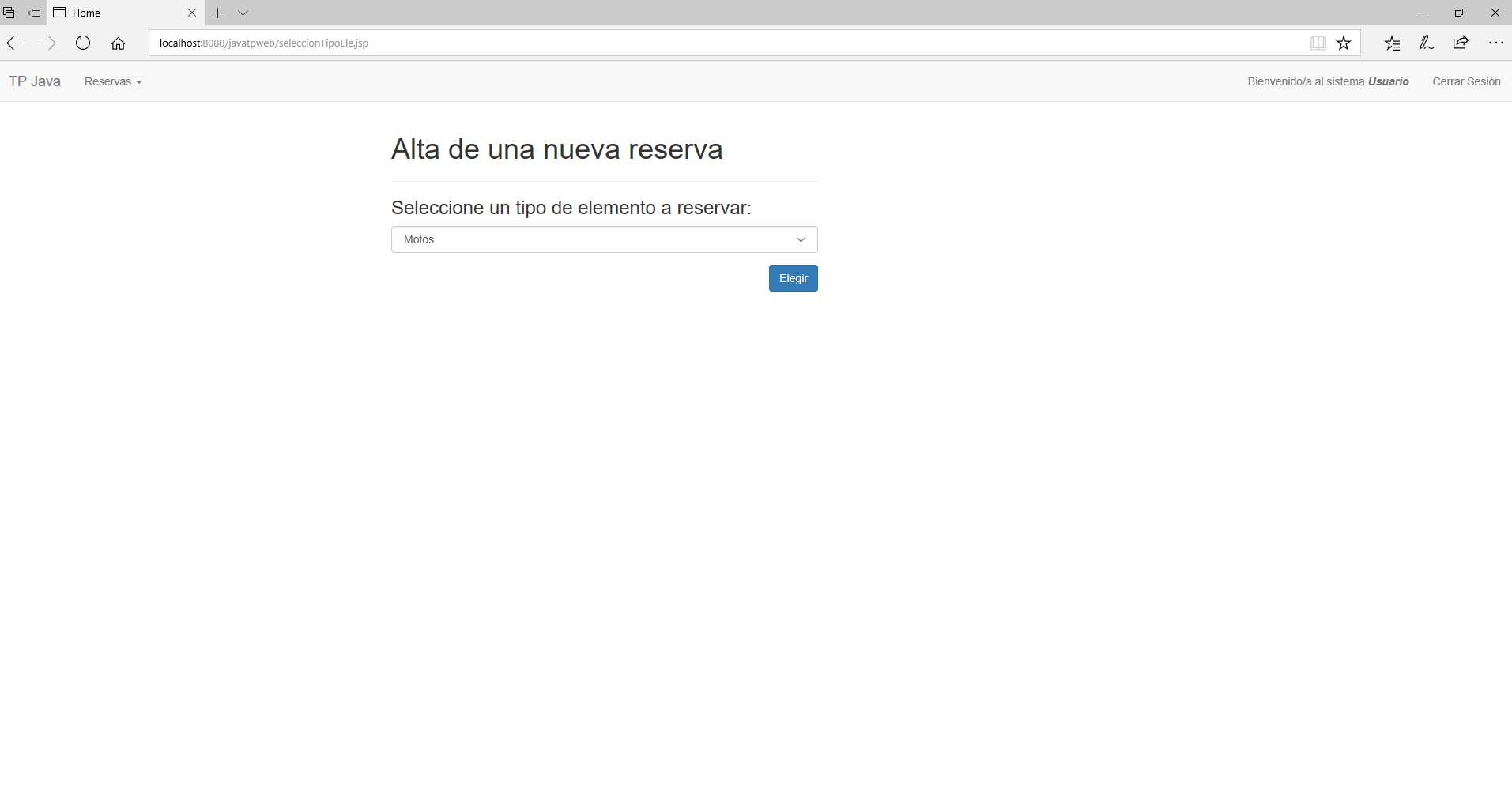


### Baja y Modificación de Tipos de Elementos

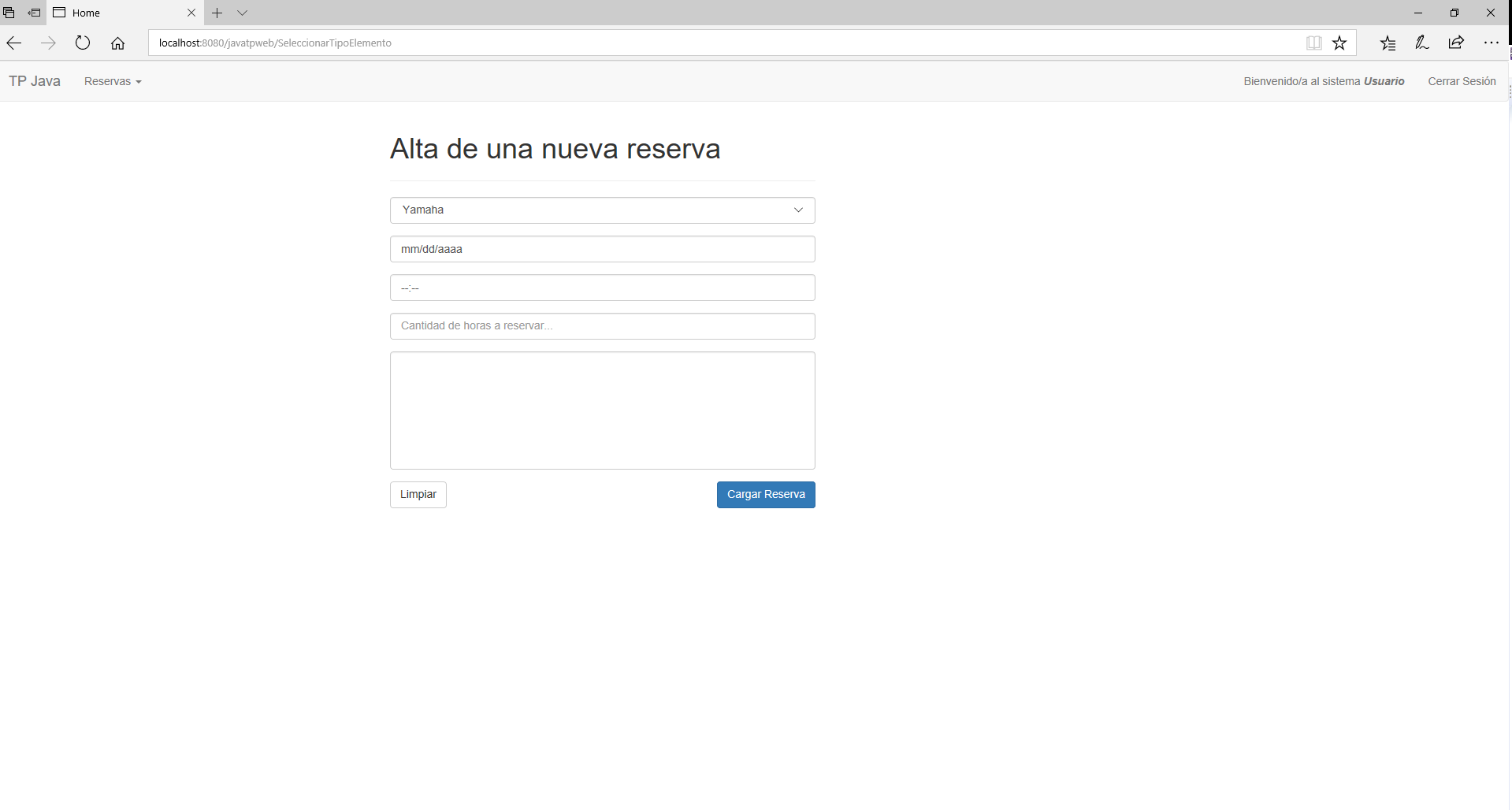


## Reservas

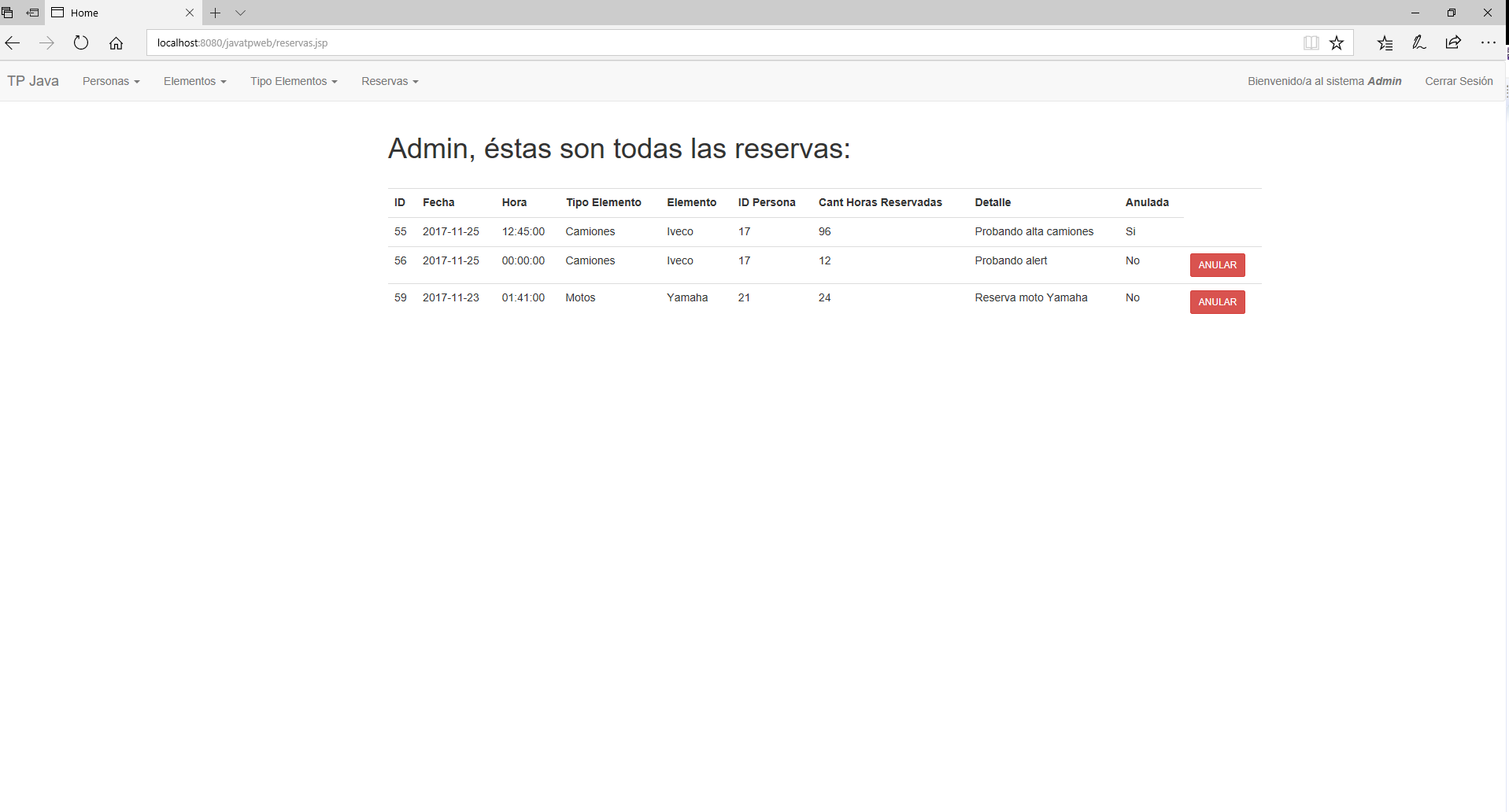
### Alta Reservas I



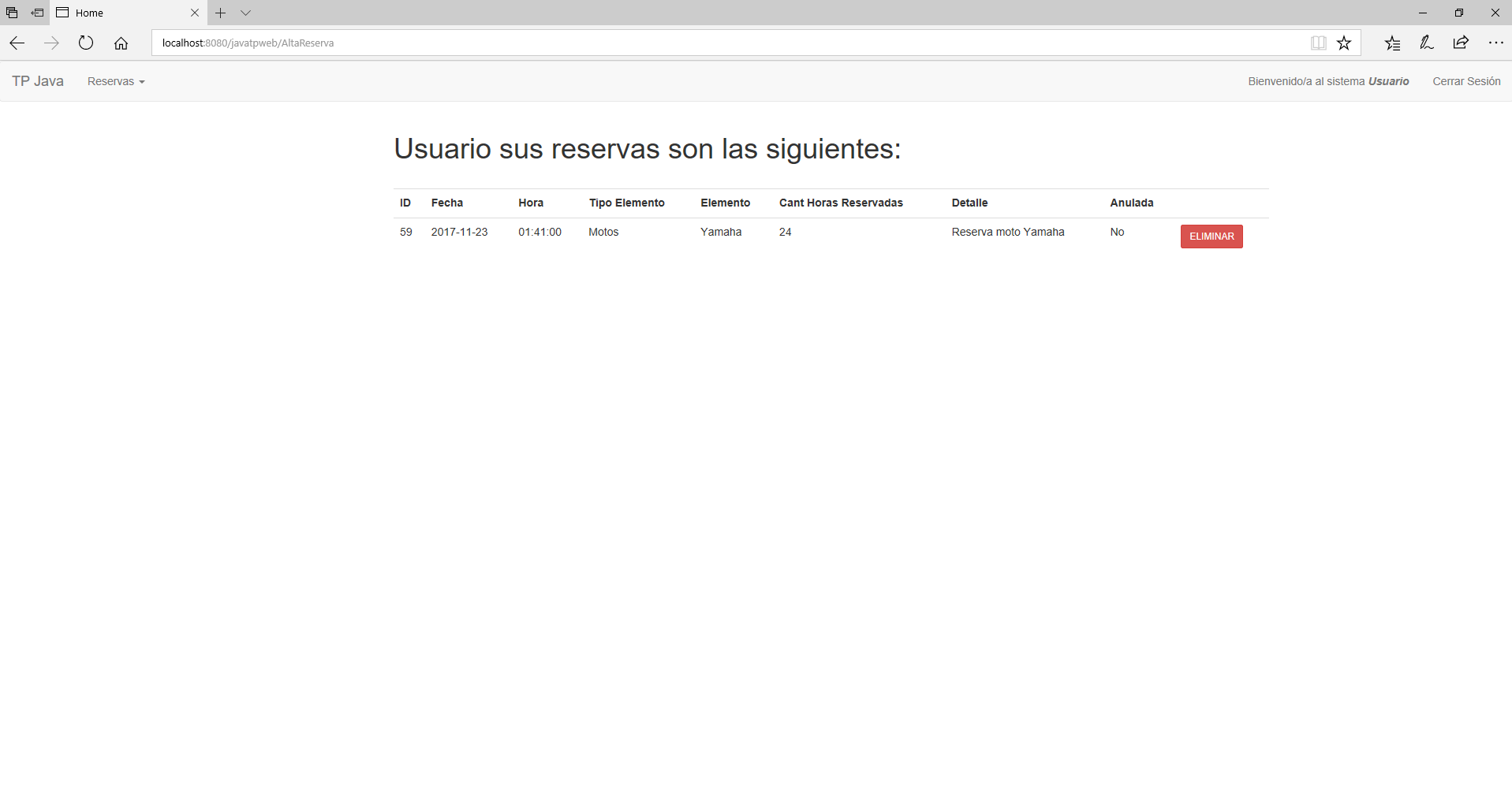
### Alta Reservas II



### Lista Reservas – Admin



### Lista Reservas – Usuario



# Fragmentos del Código Fuente

## CUU1 – Registrar nueva reserva

*Se transcribirán únicamente una* ***entidad, controlador y datos*** *para entender el funcionamiento de los mismos ya que son todos similares.*

### BookableItems.java

**package** entity;

**import** java.io.Serializable;

**public** **class** BookableItems **implements** Serializable {

**private** **int** id;

**private** String nombre;

**private** **int** id\_tipoElemento;

**public** BookableItems() {

nombre = **null**;

}

**public** **void** setBookableItems(**int** id, String nom, **int** id\_tipo) {

**this**.id = id;

nombre = nom;

id\_tipoElemento = id\_tipo;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getNombre() {

**return** nombre;

}

**public** **void** setNombre(String nombre) {

**this**.nombre = nombre;

}

**public** **int** getId\_tipoElemento() {

**return** id\_tipoElemento;

}

**public** **void** setId\_tipoElemento(**int** id\_tipoElemento) {

**this**.id\_tipoElemento = id\_tipoElemento;

}

}

### CtrlBookableItems.java

**package** controllers;

**import** java.util.ArrayList;

**import** data.DataBookableItems;

**import** entity.BookableItems;

**import** util.DuplicatedException;

**public** **class** CtrlBookableItems {

**private** DataBookableItems databi;

**private** ArrayList<BookableItems> bookitems;

**public** CtrlBookableItems(){

databi = **new** DataBookableItems();

}

**public** **void** add(BookableItems bi) **throws** DuplicatedException, Exception{

**if**(databi.getByNombre(bi)!=**null**){

**throw** **new** DuplicatedException("El elemento ya existe");

} **else** {

databi.add(bi);

}

}

**public** **void** delete(BookableItems bi)**throws** Exception{

databi.delete(bi);

}

**public** **void** update(BookableItems bi)**throws** Exception{

databi.update(bi);

}

**public** ArrayList<String> getAllNames(**int** id\_tipoEle) **throws** Exception{

**return** databi.getAllNames(id\_tipoEle);

}

**public** **int** getItemId(String elem) **throws** Exception{

**return** databi.getItemId(elem);

}

**public** BookableItems getById(**int** id) **throws** Exception {

**return** databi.getById(id);

}

**public** ArrayList<BookableItems> getAll()**throws** Exception{

**return** databi.getAll();

}

**public** ArrayList<BookableItems> getAllByType(**int** id\_tipoEle)**throws** Exception{

**return** databi.getAllByType(id\_tipoEle);

}

**public** BookableItems getRow(**int** index) **throws** Exception{

bookitems = databi.getAll();

**return** bookitems.get(index);

}

}

### DataBookableItems.java

**package** data;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.util.ArrayList;

**import** entity.BookableItems;

**import** util.AppDataException;

**public** **class** DataBookableItems {

**public** ArrayList<BookableItems> getAll() **throws** Exception{

Statement stmt=**null**;

ResultSet rs=**null**;

ArrayList<BookableItems> bookItems= **new** ArrayList<BookableItems>();

**try** {

stmt = FactoryConexion.*getInstancia*()

.getConn().createStatement();

rs = stmt.executeQuery("select \* from bookable\_items");

**if**(rs!=**null**){

**while**(rs.next()){

BookableItems bi =**new** BookableItems();

bi.setId(Integer.*parseInt*(rs.getString("id")));

bi.setNombre(rs.getString("nombre"));

bi.setId\_tipoElemento(Integer.*parseInt*(rs.getString("id\_tipo\_elemento")));

bookItems.add(bi);

}

}

} **catch** (SQLException e) {

**throw** e;

} **catch** (AppDataException ade){

**throw** ade;

}

**try** {

**if**(rs!=**null**) rs.close();

**if**(stmt!=**null**) stmt.close();

FactoryConexion.*getInstancia*().releaseConn();

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** bookItems;

}

**public** ArrayList<BookableItems> getAllByType(**int** id\_tipoEle) **throws** Exception{

ResultSet rs=**null**;

PreparedStatement stmt=**null**;

ArrayList<BookableItems> bookitems= **new** ArrayList<BookableItems>();

**try** {

stmt=FactoryConexion.*getInstancia*().getConn().prepareStatement(

"select \* from bookable\_items where id\_tipo\_elemento=?");

stmt.setInt(1, id\_tipoEle);

rs=stmt.executeQuery();

**if**(rs!=**null**){

**while**(rs.next()){

BookableItems bi =**new** BookableItems();

bi.setId(Integer.*parseInt*(rs.getString("id")));

bi.setNombre(rs.getString("nombre"));

bi.setId\_tipoElemento(Integer.*parseInt*(rs.getString("id\_tipo\_elemento")));

bookitems.add(bi);

}

}

} **catch** (SQLException e) {

**throw** e;

} **catch** (AppDataException ade){

**throw** ade;

}

**try** {

**if**(rs!=**null**) rs.close();

**if**(stmt!=**null**) stmt.close();

FactoryConexion.*getInstancia*().releaseConn();

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** bookitems;

}

**public** ArrayList<String> getAllNames(**int** id\_tipo) **throws** Exception{

ResultSet rs=**null**;

PreparedStatement stmt=**null**;

ArrayList<String> names= **new** ArrayList<String>();

**try** {

stmt=FactoryConexion.*getInstancia*().getConn().prepareStatement(

"select \* from bookable\_items where id\_tipo\_elemento=?");

stmt.setInt(1, id\_tipo);

rs=stmt.executeQuery();

**if**(rs!=**null**){

**while**(rs.next()){

names.add(rs.getString("nombre"));

}

}

} **catch** (SQLException e) {

**throw** e;

} **catch** (AppDataException ade){

**throw** ade;

}

**try** {

**if**(rs!=**null**) rs.close();

**if**(stmt!=**null**) stmt.close();

FactoryConexion.*getInstancia*().releaseConn();

} **catch** (SQLException e) {

e.printStackTrace();

}

**return** names;

}

**public** BookableItems getById(**int** id) **throws** Exception{

BookableItems bi = **null**;

PreparedStatement stmt=**null**;

ResultSet rs=**null**;

**try** {

stmt=FactoryConexion.*getInstancia*().getConn().prepareStatement(

"select \* from bookable\_items where id=?");

stmt.setString(1, String.*valueOf*(id));

rs=stmt.executeQuery();

**if**(rs!=**null** && rs.next()){

bi = **new** BookableItems();

bi.setBookableItems(Integer.*parseInt*(rs.getString("id")), rs.getString("nombre"), Integer.*parseInt*(rs.getString("id\_tipo\_elemento")));

}

} **catch** (Exception e) {

**throw** e;

} **finally**{

**try** {

**if**(rs!=**null**)rs.close();

**if**(stmt!=**null**)stmt.close();

FactoryConexion.*getInstancia*().releaseConn();

} **catch** (SQLException e) {

**throw** e;

}

}

**return** bi;

}

**public** **int** getItemId(String elem) **throws** Exception{

Integer idElem = **null**;

PreparedStatement stmt=**null**;

ResultSet rs=**null**;

**try** {

stmt=FactoryConexion.*getInstancia*().getConn().prepareStatement(

"select \* from bookable\_items where nombre=?");

stmt.setString(1, elem);

rs=stmt.executeQuery();

**if**(rs!=**null** && rs.next()){

idElem = (Integer.*parseInt*(rs.getString("id")));

}

} **catch** (Exception e) {

**throw** e;

} **finally**{

**try** {

**if**(rs!=**null**)rs.close();

**if**(stmt!=**null**)stmt.close();

FactoryConexion.*getInstancia*().releaseConn();

} **catch** (SQLException e) {

**throw** e;

}

}

**return** idElem;

}

**public** **void** update(BookableItems bi) **throws** Exception {

PreparedStatement stmt=**null**;

**try** {

stmt=FactoryConexion.*getInstancia*().getConn()

.prepareStatement(

"UPDATE bookable\_items SET nombre=?, id\_tipo\_elemento=? where id=? "

);

stmt.setString(1, bi.getNombre());

stmt.setString(2, String.*valueOf*(bi.getId\_tipoElemento()));

stmt.setString(3, String.*valueOf*(bi.getId()));

stmt.execute();

} **catch** (SQLException | AppDataException e) {

**throw** e;

}

**try** {

**if**(stmt!=**null**)stmt.close();

FactoryConexion.*getInstancia*().releaseConn();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**public** **void** delete(BookableItems bi) **throws** Exception {

PreparedStatement stmt=**null**;

**try** {

stmt=FactoryConexion.*getInstancia*().getConn()

.prepareStatement(

"delete from bookable\_items where id=? "

);

stmt.setString(1, String.*valueOf*(bi.getId()));

stmt.execute();

} **catch** (SQLException | AppDataException e) {

**throw** e;

}

**try** {

**if**(stmt!=**null**)stmt.close();

FactoryConexion.*getInstancia*().releaseConn();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**public** **void** add(BookableItems bi) **throws** Exception{

PreparedStatement stmt=**null**;

ResultSet keyResultSet=**null**;

**try** {

stmt=FactoryConexion.*getInstancia*().getConn()

.prepareStatement(

"insert into bookable\_items(nombre, id\_tipo\_elemento) values (?,?)",

PreparedStatement.***RETURN\_GENERATED\_KEYS***

);

stmt.setString(1, bi.getNombre());

stmt.setString(2, String.*valueOf*((bi.getId\_tipoElemento())));

stmt.executeUpdate();

keyResultSet=stmt.getGeneratedKeys();

**if**(keyResultSet!=**null** && keyResultSet.next()){

bi.setId(keyResultSet.getInt(1));

}

} **catch** (SQLException | AppDataException e) {

**throw** e;

}

**try** {

**if**(keyResultSet!=**null**)keyResultSet.close();

**if**(stmt!=**null**)stmt.close();

FactoryConexion.*getInstancia*().releaseConn();

} **catch** (SQLException e) {

e.printStackTrace();

}

}

**public** BookableItems getByNombre(BookableItems bookitems) **throws** Exception{

BookableItems bi =**null**;

PreparedStatement stmt=**null**;

ResultSet rs=**null**;

**try** {

stmt=FactoryConexion.*getInstancia*().getConn().prepareStatement(

"select \* from bookable\_items where nombre=?");

stmt.setString(1, bookitems.getNombre());

rs=stmt.executeQuery();

**if**(rs!=**null** && rs.next()){

bi=**new** BookableItems();

bi.setId(Integer.*parseInt*(rs.getString("id")));

bi.setNombre(rs.getString("nombre"));

bi.setId\_tipoElemento(Integer.*parseInt*(rs.getString("id\_tipo\_elemento")));

}

} **catch** (Exception e) {

**throw** e;

} **finally**{

**try** {

**if**(rs!=**null**)rs.close();

**if**(stmt!=**null**)stmt.close();

FactoryConexion.*getInstancia*().releaseConn();

} **catch** (SQLException e) {

**throw** e;

}

}

**return** bi;

}

}

### SeleccionTipoEle.jsp

<%@page import=*"entity.People"*%>

<%@page import=*"entity.Booking"*%>

<%@page import=*"entity.BookableItems"*%>

<%@page import=*"entity.BookableTypes"*%>

<%@page import=*"java.util.ArrayList"*%>

<%@page import=*"controllers.CtrlBookingTypes"* %>

<%@page import=*"controllers.CtrlBookableItems"* %>

<%@ include file=*"navbar.jsp"* %>

<%@ include file=*"controlReserva.jsp"* %>

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Seleccion de Tipo de Elemento</title>

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/bootstrap.min.css"*>

<script src=*"https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"*></script>

<script src=*"js/bootstrap.min.js"*></script>

</head>

<body>

<div class=*"container"*>

<div class=*"col-md-6 col-md-offset-1"*>

<h1>Alta de una nueva reserva</h1>

<hr>

<h3>Seleccione un tipo de elemento a reservar: </h2>

<form class=*"form-group"* action=*"SeleccionarTipoElemento"* method=*"post"* accept-charset=*utf-8*>

<div class=*"form-group"*>

<select name=*"tipo\_elemento"* class=*"form-control"*>

<%

CtrlBookingTypes ctrl = **new** CtrlBookingTypes();

ArrayList<BookableTypes> listbt = ctrl.getAll();

ArrayList<BookableTypes> listaReduc = ctrl.getAllReduc();

**if**(((People)session.getAttribute("user")).getTipo\_usuario() == 1) {

**for**(BookableTypes bt : listbt) {

%>

<option value=*"*<%=bt.getId() %>*"*><%=bt.getNombre()%></option>

<%

}

}

**else** {

**for**(BookableTypes lr : listaReduc) {

%>

<option value=*"*<%=lr.getId() %>*"*><%=lr.getNombre()%></option>

<%

}

}

%>

</select>

</div>

<div class=*"form-group"*>

<input type=*"submit"* class=*"btn btn-primary pull-right"* name=*"submit"* value=*"Elegir"*>

</div>

</form>

</div>

</div>

</body>

</html>

### SeleccionarTipoElemento.java

**package** servlet;

**import** java.io.IOException;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

/\*\*

\* Servlet implementation class SeleccionarTipoElemento

\*/

@WebServlet("/SeleccionarTipoElemento")

**public** **class** SeleccionarTipoElemento **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

/\*\*

\* **@see** HttpServlet#HttpServlet()

\*/

**public** SeleccionarTipoElemento() {

**super**();

// **TODO** Auto-generated constructor stub

}

/\*\*

\* **@see** HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// **TODO** Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\*\*

\* **@see** HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**int** id\_tipo = Integer.*parseInt*(request.getParameter("tipo\_elemento"));

request.getSession().setAttribute("id\_tipoEle", id\_tipo);

request.getRequestDispatcher("altaReserva.jsp").forward(request, response);

}

}

### altaReserva.jsp

<%@page import=*"entity.People"*%>

<%@page import=*"entity.Booking"*%>

<%@page import=*"entity.BookableItems"*%>

<%@page import=*"entity.BookableTypes"*%>

<%@page import=*"java.util.ArrayList"*%>

<%@page import=*"java.util.Calendar"*%>

<%@page import=*"java.util.Date"*%>

<%@page import=*"controllers.CtrlBookingTypes"* %>

<%@page import=*"controllers.CtrlBookableItems"* %>

<%@ include file=*"navbar.jsp"* %>

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Insert title here</title>

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/bootstrap.min.css"*>

<script src=*"https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"*></script>

<script src=*"js/bootstrap.min.js"*></script>

</head>

<body>

<div class=*"container"*>

<div class=*"col-md-6 col-md-offset-1"*>

<h1>Alta de una nueva reserva</h1>

<hr>

<form class=*"form-group"* action=*"AltaReserva"* method=*"post"* >

<div class=*"form-group"*>

<input type=*"hidden"* class=*"form-control"* name=*"tipo\_elemento"* value=*"*<%=(Integer)session.getAttribute("id\_tipoEle")%>*"*>

</div>

<div class=*"form-group"*>

<select name=*"elemento"* class=*"form-control"*>

<%

Calendar calendar = Calendar.getInstance();

**int** year = calendar.get(Calendar.YEAR);

**int** month = calendar.get(Calendar.MONTH)+1;

**int** day = calendar.get(Calendar.DAY\_OF\_MONTH)+1;

BookableTypes bt = **new** BookableTypes();

CtrlBookingTypes ctrlTypes = **new** CtrlBookingTypes();

bt = ctrlTypes.getById((Integer)session.getAttribute("id\_tipoEle"));

calendar.add(Calendar.DAY\_OF\_MONTH, bt.getDias());

**int** yearMax = calendar.get(Calendar.YEAR);

**int** monthMax = calendar.get(Calendar.MONTH)+1;

**int** dayMax = calendar.get(Calendar.DAY\_OF\_MONTH);

CtrlBookableItems ctrlItems = **new** CtrlBookableItems();

ArrayList<BookableItems> listbi = ctrlItems.getAllByType((Integer)session.getAttribute("id\_tipoEle"));

**for**(BookableItems bi : listbi) {

%>

<option value=*"*<%=bi.getId() %>*"*><%=bi.getNombre()%></option>

<%

}

%>

</select>

</div>

<div class=*"form-group"*>

<input type=*"date"* class=*"form-control"* name=*"fecha"* min=*"*<%=year+"-"+month+"-"+day %>*"* max=*"*<%=yearMax+"-"+monthMax+"-"+dayMax %>*"*>

</div>

<div class=*"form-group"*>

<input type=*"time"* class=*"form-control"* name=*"hora"*>

</div>

<div class=*"form-group"*>

<input type=*"number"* class=*"form-control"* name=*"cant\_horas"* max=*"*<%=bt.getLimite() %>*"* min=*"1"* placeholder=*"Cantidad de horas a reservar..."*>

</div>

<div class=*"form-group"*>

<textarea style="resize:*none*; height:*150px*" type=*"text"* class=*"form-control"* name=*"detalle"* placeholder=*"Detalle..."* required> </textarea>

</div>

<div class=*"form-group"*>

<input type=*"hidden"* class=*"form-control"* name=*"id\_persona"* value=*"*<%=((People)session.getAttribute("user")).getId()%>*"*>

</div>

<div class=*"form-group"*>

<button type=*"reset"* value=*"Reset"* class=*"btn btn-default"* >Limpiar</button>

<input type=*"submit"* class=*"btn btn-primary pull-right"* name=*"submit"* value=*"Cargar Reserva"*>

</div>

</form>

</div>

</div>

</body>

</html>

### AltaReserva.java

**package** servlet;

**import** java.io.IOException;

**import** java.sql.Date;

**import** java.sql.Time;

**import** javax.servlet.ServletException;

**import** javax.servlet.annotation.WebServlet;

**import** javax.servlet.http.HttpServlet;

**import** javax.servlet.http.HttpServletRequest;

**import** javax.servlet.http.HttpServletResponse;

**import** controllers.CtrlBooking;

**import** entity.Booking;

**import** util.AppDataException;

**import** util.DuplicatedException;

**import** util.Emailer;

/\*\*

\* Servlet implementation class AltaReserva

\*/

@WebServlet("/AltaReserva")

**public** **class** AltaReserva **extends** HttpServlet {

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

/\*\*

\* **@see** HttpServlet#HttpServlet()

\*/

**public** AltaReserva() {

**super**();

// **TODO** Auto-generated constructor stub

}

/\*\*

\* **@see** HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)

\*/

**protected** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

// **TODO** Auto-generated method stub

response.getWriter().append("Served at: ").append(request.getContextPath());

}

/\*\*

\* **@see** HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)

\*/

**protected** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {

**try** {

String detalle = request.getParameter("detalle");

Date fecha = Date.*valueOf*(request.getParameter("fecha"));

Time hora = Time.*valueOf*(request.getParameter("hora")+":00");

**int** id\_tipoEle = Integer.*parseInt*(request.getParameter("tipo\_elemento"));

**int** id\_ele = Integer.*parseInt*(request.getParameter("elemento"));

**int** id\_persona = Integer.*parseInt*(request.getParameter("id\_persona"));

**int** cant\_horas = Integer.*parseInt*(request.getParameter("cant\_horas"));

CtrlBooking ctrl = **new** CtrlBooking();

Booking bok = **new** Booking();

bok.setDetalle(detalle);

bok.setFecha(fecha);

bok.setHora(hora);

bok.setId\_elemento(id\_ele);

bok.setId\_tipoElemento(id\_tipoEle);

bok.setId\_persona(id\_persona);

bok.setAnulada(**false**);

bok.setCant\_horas(cant\_horas);

**try** {

ctrl.add(bok);

Emailer.*getInstance*().send("tpjava2017@gmail.com","Alta de reserva","Su reserva " +detalle +" para el dia " +fecha +" a las " +hora +" fue realizada con satisfacción.");

request.getRequestDispatcher("/reservas.jsp").forward(request, response);

} **catch** (AppDataException ade) {

request.setAttribute("Error", ade.getMessage());

} **catch** (DuplicatedException de){

request.setAttribute("Errorlim", de.getMessage());

request.getRequestDispatcher("/WEB-INF/limiteError.jsp").forward(request, response);

} **catch** (Exception e) {

response.setStatus(502);

}

} **catch** (Exception e) {

e.printStackTrace();

}

}

}

### reservas.jsp

<%@page import=*"java.time.chrono.ChronoLocalDate"*%>

<%@page import=*"java.time.LocalDateTime"*%>

<%@page import=*"org.omg.CORBA.CurrentOperations"*%>

<%@page import=*"java.sql.Date"*%>

<%@page import=*"java.sql.Timestamp"*%>

<%@page import=*"entity.People"*%>

<%@page import=*"entity.Booking"*%>

<%@page import=*"entity.BookableItems"*%>

<%@page import=*"entity.BookableTypes"*%>

<%@page import=*"java.util.ArrayList"*%>

<%@page import=*"controllers.CtrlBooking"*%>

<%@page import=*"controllers.CtrlBookingTypes"*%>

<%@page import=*"controllers.CtrlBookableItems"*%>

<%@ include file=*"navbar.jsp"* %>

<%@ page language=*"java"* contentType=*"text/html; charset=ISO-8859-1"*

pageEncoding=*"ISO-8859-1"*%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv=*"Content-Type"* content=*"text/html; charset=ISO-8859-1"*>

<title>Reservas</title>

<link rel=*"stylesheet"* type=*"text/css"* href=*"css/bootstrap.min.css"*>

<script src=*"https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"*></script>

<script src=*"js/bootstrap.min.js"*></script>

</head>

<body>

<div class=*"container"*>

<div class=*"col-md-12 col-md-offset-1"*>

<% **if**(((People)session.getAttribute("user")).getTipo\_usuario() == 0)

{

%>

<h1><%=((People)session.getAttribute("user")).getNombre() %>, éstas son todas las reservas: </h1>

<br>

<table class=*"table table-hover"*>

<tr>

<th>ID</th>

<th>Fecha</th>

<th>Hora</th>

<th>Tipo Elemento</th>

<th>Elemento</th>

<th>ID Persona</th>

<th>Cant Horas Reservadas</th>

<th>Detalle</th>

<th>Anulada</th>

<th></th>

</tr>

<%

CtrlBooking ctrl= **new** CtrlBooking();

ArrayList<Booking> bookings = ctrl.getList();

CtrlBookingTypes ctrlType = **new** CtrlBookingTypes();

CtrlBookableItems ctrlItem = **new** CtrlBookableItems();

String nombreType;

String nombreItem;

**for**(Booking b : bookings){

nombreType = ctrlType.getById(b.getId\_tipoElemento()).getNombre();

nombreItem = ctrlItem.getById(b.getId\_elemento()).getNombre();

%>

<form method=*"POST"* action=*"ModificarReserva"*>

<tr>

<td><%=b.getId() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getId() %>*"*></td>

<td><%=b.getFecha() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getFecha() %>*"*></td>

<td><%=b.getHora() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getHora() %>*"*></td>

<td><%=nombreType %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getId\_tipoElemento() %>*"*></td>

<td><%=nombreItem %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getId\_elemento() %>*"*></td>

<td><%=b.getId\_persona() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getId\_persona() %>*"*></td>

<td><%=b.getCant\_horas() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getCant\_horas() %>*"*></td>

<td><%=b.getDetalle() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getDetalle() %>*"*></td>

<td><%=b.isAnuladaString() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.isAnulada() %>*"*></td>

<%

LocalDateTime fecha = LocalDateTime.now();

**if**(!b.isAnulada())

{

%>

<td><button type=*"submit"* class=*"btn btn-danger btn-sm"* name=*"anular"* value=*"Anular"*>ANULAR</button></td>

<%

}

%>

</tr>

</form>

<%

}

%>

</table>

<%

} **else** {

%>

<h1><%=((People)session.getAttribute("user")).getNombre() %> sus reservas son las siguientes: </h1>

<br>

<table class=*"table table-hover"*>

<tr>

<th>ID</th>

<th>Fecha</th>

<th>Hora</th>

<th>Tipo Elemento</th>

<th>Elemento</th>

<th>Cant Horas Reservadas</th>

<th>Detalle</th>

<th>Anulada</th>

<th></th>

</tr>

<%

CtrlBooking ctrl = **new** CtrlBooking();

CtrlBookingTypes ctrlType = **new** CtrlBookingTypes();

CtrlBookableItems ctrlItem = **new** CtrlBookableItems();

String nombreType;

String nombreItem;

**int** id\_pers = ((People)session.getAttribute("user")).getId();

ArrayList<Booking> bookings = ctrl.getReservasByPerson(id\_pers);

**for**(Booking b : bookings){

nombreType = ctrlType.getById(b.getId\_tipoElemento()).getNombre();

nombreItem = ctrlItem.getById(b.getId\_elemento()).getNombre();

%>

<form method=*"POST"* action=*"ModificarReserva"*>

<tr>

<td><%=b.getId() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getId() %>*"*></td>

<td><%=b.getFecha() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getFecha() %>*"*></td>

<td><%=b.getHora() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getHora() %>*"*></td>

<td><%=nombreType %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getId\_tipoElemento() %>*"*></td>

<td><%=nombreItem %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getId\_elemento() %>*"*></td>

<td><%=b.getCant\_horas() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getCant\_horas() %>*"*></td>

<td><%=b.getDetalle() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.getDetalle() %>*"*></td>

<td><%=b.isAnuladaString() %><input type=*"hidden"* name=*"id"* value=*"*<%=b.isAnulada() %>*"*></td>

<input type=*"hidden"* name=*"id"* value=*"*<%=b.getId\_persona() %>*"*>

<%**if**(!b.isAnulada())

{

%>

<td><button type=*"submit"* class=*"btn btn-danger btn-sm"* name=*"eliminar"* value=*"Eliminar"*>ELIMINAR</button></td>

<%

}

%>

</tr>

</form>

<%

}

%>

</table>

<%

}

%>

</div>

</div>

</body>

</html>