

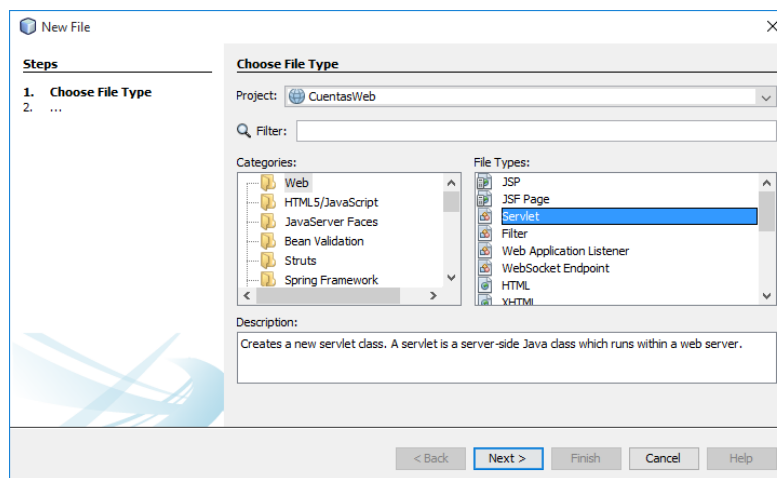


Arquitectura de Software - Universidad de Antioquia

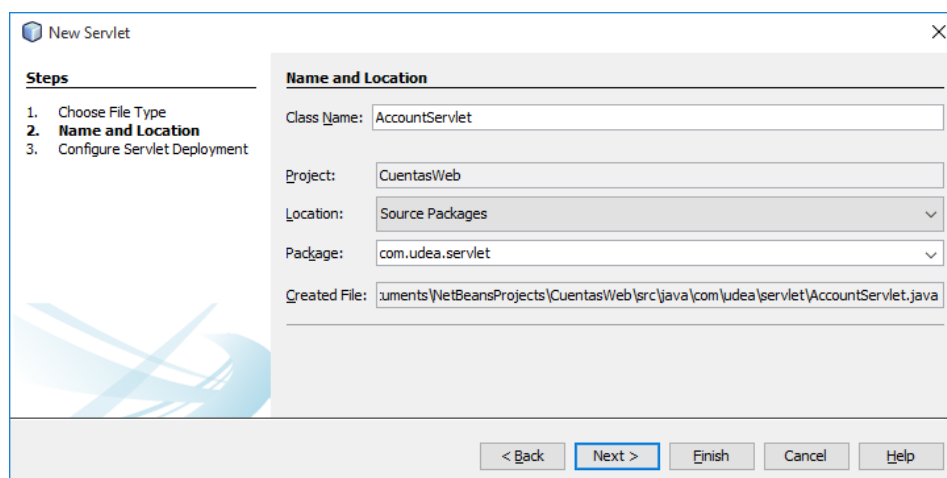
Creación del Servlet.

Se creará una clase Servlet llamado **AccountServlet**, que actuará como un controlador de las vistas. Realizaremos el siguiente procedimiento:

- 1- Haga click derecho sobre el respectivo nodo del proyecto. Seleccione **New → Other → Web → Servlet**



2. Coloque el nombre de la clase en nuestro caso **AccountServlet** y el nombre del paquete por ejemplo **com.udea.servlet**. Haga click en el botón siguiente:



3. En la siguiente ventana verifique que esté activado la opción de usar el descriptor de despliegue (web.xml), para poder registrar y mapear el servlet en el context path del proyecto.



1803

Arquitectura de Software - Universidad de Antioquia

New Servlet

Steps

1. Choose File Type
2. Name and Location
3. **Configure Servlet Deployment**

Configure Servlet Deployment

Register the Servlet with the application by giving the Servlet an internal name (Servlet Name). Then specify patterns that identify the URLs that invoke the Servlet. Separate multiple patterns with commas.

☒ Add information to deployment descriptor (web.xml)

Class Name:

Servlet Name:

URL Pattern(s):

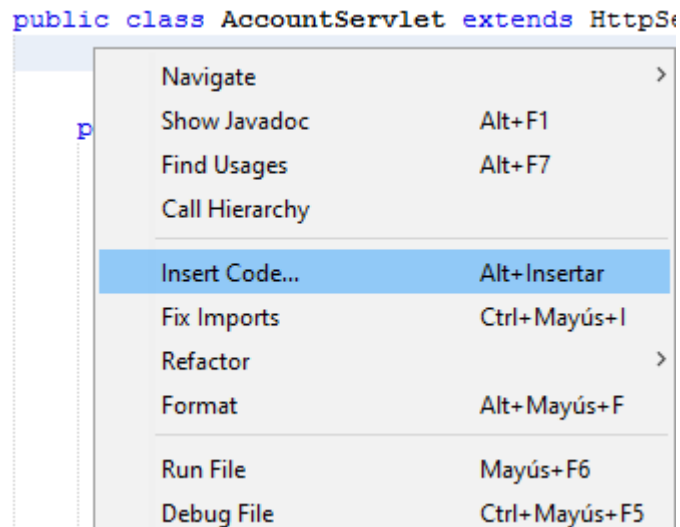
Initialization Parameters:

Name	Value
------	-------

New Edit... Delete

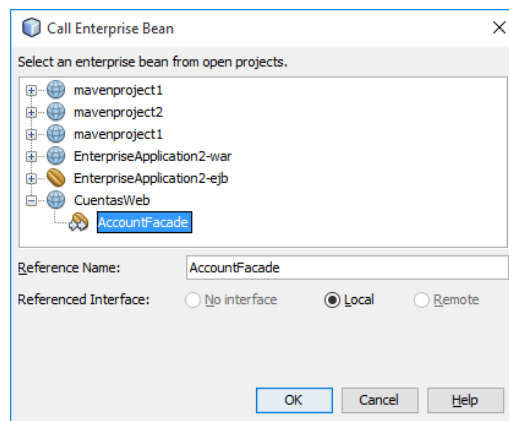
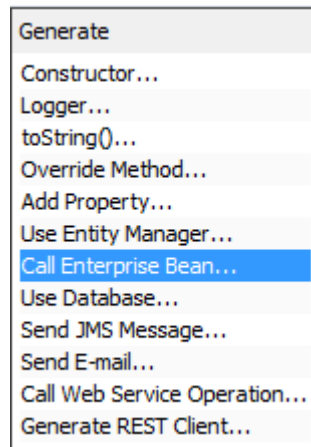
< Back Next > **Finish** Cancel Help

4. Sobre el editor código de la clase, haga click derecho y seleccione **Insert Code** luego seleccione la opción **Call Enterprise Bean** y agregue la Interfaz Local de la fachada de AccountFacade.





Arquitectura de Software - Universidad de Antioquia



Asegúrese que este referenciado el acceso a los métodos del negocio con la interfaz a través de la llamada con la anotación **@EJB**



1803

Arquitectura de Software - Universidad de Antioquia

```
import javax.servlet.http.HttpServletResponse;

/**...4 lines */
public class AccountServlet extends HttpServlet {

    @EJB
    private AccountFacadeLocal accountFacade;

    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        try (PrintWriter out = response.getWriter()) {
            /* TODO output your page here. You may use following sample code. */
            out.println("<!DOCTYPE html>");
            out.println("<html>");
        }
    }
}
```

5. Agregue sobre el método principal del servlet **processRequest**, el siguiente código que permitirá generar el control de los eventos de las vistas JSP.

```
29     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
30         throws ServletException, IOException {
31         response.setContentType("text/html;charset=UTF-8");
32         PrintWriter out = response.getWriter();
33     try{
34         String action = request.getParameter("action");
35         String url = "index.jsp";
36         if ("list".equals(action)) {
37             List<Account> findAll = accountFacade.findAll();
38             request.getSession().setAttribute("accounts", findAll);
39             url = "listAccounts.jsp";
40         } else if ("login".equals(action)) {
41             String u = request.getParameter("username");
42             String p = request.getParameter("password");
43             boolean checklogin = accountFacade.checkLogin(u, p);
44             if (checklogin) {
45                 request.getSession().setAttribute("login", u);
46                 url = "manager.jsp";
47             } else {
48                 url = "login.jsp?error=1";
49             }
50         } else if ("insert".equals(action)) {
51             Account a = new Account();
52             // String idu = request.getParameter("iduser");
53             // a.setId(Integer.parseInt(idu));
54             a.setUsername(request.getParameter("username"));
55             a.setPassword(request.getParameter("password"));
56             a.setEmail(request.getParameter("email"));
57             accountFacade.create(a);
58             url = "login.jsp";
59         }
60     } catch (Exception e) {
61         url = "login.jsp?error=1";
62     }
63     out.println("<html>");
64     out.println("</html>");
65 }
```



Arquitectura de Software - Universidad de Antioquia

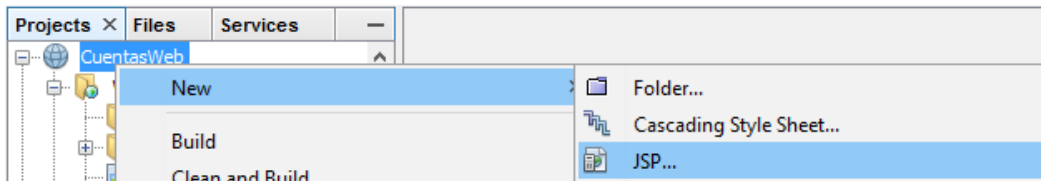
```
59
60         } else if ("delete".equals(action)) {
61             String id = request.getParameter("id");
62             Account a = accountFacade.find(Integer.valueOf(id));
63             accountFacade.remove(a);
64             url = "AccountServlet?action=list";
65
66         } else if ("logout".equals(action)) {
67             request.getSession().removeAttribute("login");
68             url = "login.jsp";
69
70         }
71         response.sendRedirect(url);
72     } finally {
73         out.close();
74     }
75 }
76
77 + HttpServlet methods. Click on the + sign on the left to edit the code.
115
116 }
```

Nota: Intente comprender el funcionamiento del código y si lo desea agregue nuevas funcionalidades.

Creación de Vistas JSP

El último paso será crear las distintas vistas de nuestra aplicación. Seguiremos los siguientes pasos:

- 1- Sobre el nodo principal del proyecto haga click derecho luego seleccione **New → Other → Web → JSP page**



- 2- Coloque el respectivo nombre a la vista. En nuestro caso tendremos las siguientes vistas:
 - **Index.jsp** → Página principal de la aplicación.
 - **Menú.jsp** → Contiene las diferentes opciones del menú que se carga en cada vista si se realizó un logueo exitoso.
 - **listAccounts.jsp** → Genera una lista sencilla con los datos de las cuentas registradas en la BD. Desde acá se podrán eliminar cuentas si el usuario lo desea.
 - **login.jsp** → Es el formulario para el logueo por usuario y contraseña de nuestros usuarios
 - **manager.jsp** → Sera la página de entrada a cada usuario siempre y cuando el logueo sea exitoso.
 - **newAccount.jsp** → General el formulario que permite crear nuevas cuentas a nuestros usuarios.

Todas las vistas tendrán enlace al framework de Twitter Bootstrap, para hacer un poco más usable la aplicación. Coloque el respectivo nombre a cada vista la cual quedará registrada en la carpeta Web Pages del proyecto.



Arquitectura de Software - Universidad de Antioquia

New JSP

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

File Name:

Project:

Location:

Folder:

Created File:

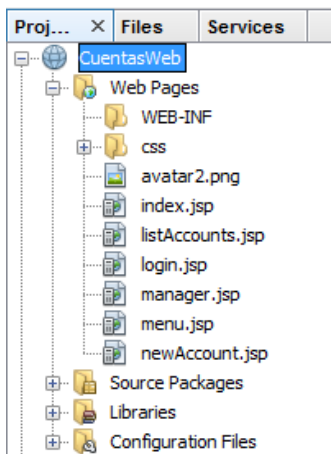
Options:

☒ JSP File (Standard Syntax) ☐ Create as a JSP Segment

☐ JSP Document (XML Syntax)

Description:

Las vistas serán las siguientes:





Arquitectura de Software - Universidad de Antioquia

3- Agregue el siguiente código JSP y HTML a cada vista:

Index.jsp

```
index.jsp x
Source History
... 5 lines
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>JSP Page</title>
13 </head>
14 <body>
15
16 <jsp:include page="menu.jsp"></jsp:include>
17 <h1>PAGINA PARA GESTION DE PERSONAL</h1>
18 </body>
19 </html>
```

Menu.jsp

```
menu.jsp x
Source History
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
2
3 <c:if test="${empty login}">
4 |<a href="login.jsp">Login</a>|
5 |<a href="newAccount.jsp">Register</a>|
6
7 </c:if>
8 <c:if test="${not empty login}">
9 |<a href="AccountServlet?action=logout">Logout</a>|
10 |<a href="AccountServlet?action=list">List Accounts</a>|
11
12 </c:if>
13 |<a href="about.jsp">About</a>|
14 <hr/>
```



1803

Arquitectura de Software - Universidad de Antioquia

listAccounts.jsp

```
Source History
...5 lines
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
9
10 <!DOCTYPE html>
11 <html>
12 <head>
13 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
14 <title>JSP Page</title>
15 </head>
16 <body>
17 <jsp:include page="menu.jsp"></jsp:include>
18 <h1>Hello World!</h1>
19 <c:forEach var="a" items="${accounts}">
20 | ${a.id}| | ${a.username}| | ${a.email}|
21 | <a onclick="return confirm('Esta seguro?')" href="AccountServlet?action=delete&id=${a.id}">Delete</a>
22 | <hr/>
23 </c:forEach>
24
25 </body>
26 </html>
```

newAccount.jsp

```
6 <%@page contentType="text/html" pageEncoding="UTF-8"%>
7 <!DOCTYPE html>
8 <html>
9 <head>
10 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
11 <title>JSP Page</title>
12 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
13 <link rel="stylesheet" href="style.css">
14 <!-- Optional theme -->
15 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">
16 <!-- Latest compiled and minified JavaScript -->
17 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
18 </head>
19 <body>
20 <jsp:include page="menu.jsp"></jsp:include>
21 <h1>Hello World!</h1>
22 <div class="container well">
23 | <form action="AccountServlet?action=insert" method="post">
24 |
25 | <table>
26 | | <tr>
27 | | | <th> <label><b>Username:</b></label> </th>
28 | | | <th>
29 | | | | <input type="text" placeholder="Enter Username" class="form-control" name="username" required=""/>
30 | | | </th>
31 | | </tr>
32 | | <tr>
33 | | | <th> <label><b>Password:</b></label></th>
34 | | | <th><input type="password" placeholder="Enter Password" class="form-control" name="password"
35 | | | | required=""/> </th>
```




1803

Arquitectura de Software - Universidad de Antioquia

```
6 </tr>
7 <tr>
8   <th> <label><b>Password:</b></label></th>
9   <th><input type="email" placeholder="Enter Email" class="form-control" name="email"
0     required=""/> </th>
1 </tr>
2
3 <div class="break"></div>
4 </div>
5 <tr>
6   <td colspan="2">
7     <input class="btn icon-btn btn-success" type="submit" name="action" value="Insertar">
8     <span class="glyphicon glyphicon-ok-sign"></span>
9
10    <!-- <span class="glyphicons glyphicon-user-add img-circle te:
11    <input class="btn btn-info btn-lg" type="reset" name="action" value="Reset">
12    <span class="glyphicon glyphicon-remove"></span>
13  </td>
14 </tr>
15 </table>
16 </form>
17 <br>
18 </div>
19 </body>
20 </html>
```



1 8 0 3

Arquitectura de Software - Universidad de Antioquia

Login.jsp

```
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
9 <!DOCTYPE html>
10 <html>
11 <head>
12 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
13 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap.min.css">
14 <link rel="stylesheet" href="style.css">
15 <!-- Optional theme -->
16 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/css/bootstrap-theme.min.css">
17 <!-- Latest compiled and minified JavaScript -->
18 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.2.0/js/bootstrap.min.js"></script>
19
20 <title>JSP Page</title>
21 </head>
22 <body>
23 <jsp:include page="menu.jsp"></jsp:include>
24 <h1>Hello World!</h1>
25 <c:if test="${param.error==1}">
26 <font color="red">Usuario Invalido. Intentelo de nuevo</font>
27 </c:if>
28 <div class="container well">
29 <h1>Contact Information</h1>
30 <form action="AccountServlet?action=login" method="post">
31 <table>
32 <tr>
33 <th><label><b>Username:</b></label></th>
34 <th>
35 <input type="text" placeholder="Enter Username" class="form-control" name="username" required=""/>
36 </th>
37 </tr>
38 <tr>
39 <th><label><b>Password:</b></label></th>
40 <th><input type="password" placeholder="Enter Password" class="form-control" name="password"
41 <required=""/></th>
42 </tr>
43
44 <div class="break"></div>
45 </div>
46
47 <tr>
48 <td colspan="2">
49 <input class="btn btn-info btn-lg" type="submit" name="action" value="Login">
50 <span class="glyphicon glyphicon-lock"></span>
51
52 <!-- <span class="glyphicons glyphicons-user-add img-circle text--
53
54 <input class="btn btn-info btn-lg" type="reset" name="action" value="Reset">
55 <span class="glyphicon glyphicon-remove"></span>
56 </td>
57 </tr>
58 </table>
59 </form>
60 <br>
61 </div>
62
63 </body>
64 </html>
```



Arquitectura de Software - Universidad de Antioquia

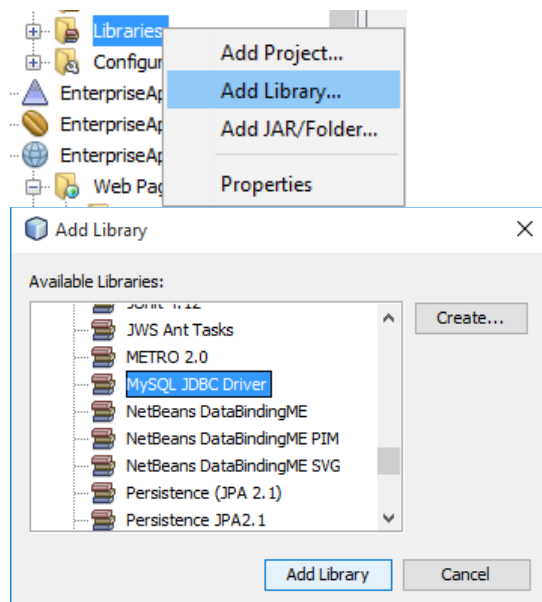


Arquitectura de Software - Universidad de Antioquia

Manager.jsp

```
manager.jsp x
Source History
...5 lines
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE html>
9 <html>
10 <head>
11 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 <title>JSP Page</title>
13 </head>
14 <body>
15 <jsp:include page="menu.jsp"></jsp:include>
16 <h1>BIENVENIDO A LA PAGINA ${login}</h1>
17 </body>
18 </html>
```

NOTA: Antes de realizar el despliegue de la aplicación asegúrese de tener cargado el driver de MySQL para JAVA dentro de la carpeta **Libraries** del proyecto. Si no la tiene solo debe adicionar el respectivo Jar.

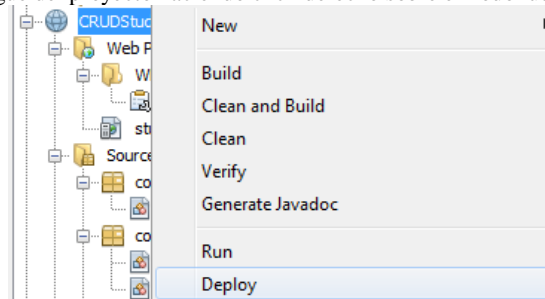




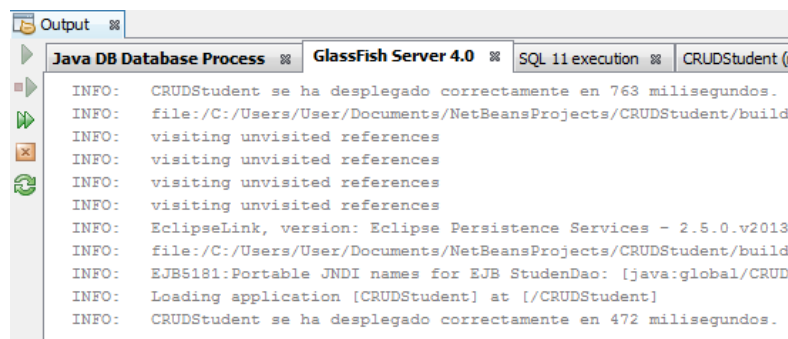
Arquitectura de Software - Universidad de Antioquia

Despliegue del Proyecto

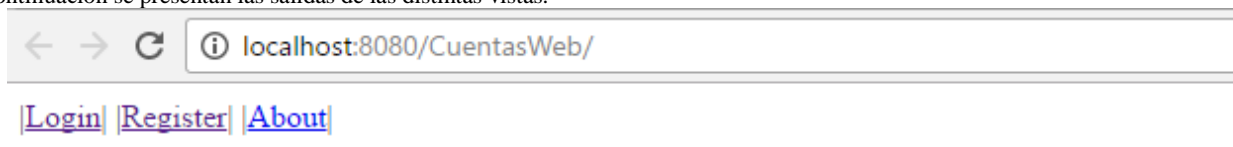
Ahora solo debemos realizar el despliegue del proyecto haciendo click derecho sobre el nodo luego seleccione la opción **Deploy**.



Una vez desplegado exitosamente el proyecto sobre el servidor de aplicaciones puede seleccionar la opción Run para abrir el Navegador.



A continuación se presentan las salidas de las distintas vistas.



PAGINA PARA GESTION DE PERSONAL



1803

Arquitectura de Software - Universidad de Antioquia

← → ↻ ⓘ localhost:8080/CuentasWeb/login.jsp
[|Login|](#) [|Register|](#) [|About|](#)

Hello World!

Contact Information

Username:

Password:

← → ↻ ⓘ localhost:8080/CuentasWeb/manager.jsp
[|Logout|](#) [|List Accounts|](#) [|About|](#)

BIENVENIDO A LA PAGINA dbotia

← → ↻ ⓘ localhost:8080/CuentasWeb/listAccounts.jsp
[|Logout|](#) [|List Accounts|](#) [|About|](#)

Hello World!

[1] |dbotia| |dbotia@gmail.com| [Delete](#)

[2] |jperez| |jperez@gmail.com| [Delete](#)



Arquitectura de Software - Universidad de Antioquia

← → ↻ ⓘ localhost:8080/CuentasWeb/newAccount.jsp

[|Login|](#) [|Register|](#) [|About|](#)

Hello World!

Username:

Password:

Email:

← → ↻ ⓘ localhost:8080/CuentasWeb/listAccounts.jsp

[|Logout|](#) [|List Accounts|](#) [|About|](#)

Hello World!

1	dbotia	dbotia@gmail.com	Delete
2	jperez	jperez@gmail.com	Delete
3	admin	admin@gmail.com	Delete

localhost:8080 dice:

Esta seguro?

I. Ejercicios Propuestos

FECHA DE ENTREGA: 12 de Julio de 2019

Realizar una aplicación CRUD completa (siguiendo los mismos pasos de este laboratorio) que permita manejar la información de un concesionario de vehículos. Deberá soportar la información de Clientes, Vehículos, y Ventas generales. Además deberá permitir subir las fotos de los vehículos y visualizarlos según se busque la información de cada uno por medio de la introducción de la matrícula.

Entregables: Informe de Laboratorio, Script SQL y Proyecto en GitHub.



Arquitectura de Software - Universidad de Antioquia

II. Bibliografía Propuesta - Webgrafia

<http://docs.oracle.com/javaee/1.2.1/devguide/html/DevGuideTOC.html>

<https://docs.oracle.com/javaee/7/tutorial/ejb-intro003.htm#GIPKO>

H. Bergesten, *Java Server Faces*, 1005 Gravenstein Highway North, Sebastopol: O'Reilly & Associates, 2002.