

Paquetes

- Beans → Clases
- Controllers → Controladores
- Model → Lógica de negocio y conexión
- Utils →

Cosas importantes

- “**Instalar**” el Apache Tomcat, después crear proyecto y correrlo.

```
@WebServlet(name = "LibrosController", urlPatterns = { "/LibrosController" })  
public class LibrosController extends HttpServlet
```

- Id autoincrementable

HTML

- ```
<select>
<option selected> Open this menu </option>
<option value="1"> One </option>
<option value="2"> Two </option>
<option value="3"> Three </option>
</select>
```

•

## Validaciones

- ```
pattern="[/A-Za-zÑñáéíóúÁÉÍÓÚ\st]+"
```
- ```
pattern="[/0-9]{8}$"
```
- ```
pattern="\d{4}-(0[1-9]|1[0-2])-(0[1-9]|1[2][0-9]|3[01])"
```
- ```
pattern="[/A-Za-zÑñáéíóúÁÉÍÓÚ0-9.#\st]+"
```

## Bean

Como una clase normal

## Conexion

Dentro del model

```
public class Conexion {

 private static String url = "jdbc:mysql://localhost:3306/****cambia****";
 private static String user = "root";
 private static String password = "123456";
 protected static Conexion conexion;

 public static Conexion openConexion() {
 try {
 Class.forName("com.mysql.cj.jdbc.Driver");
 conexion = DriverManager.getConnection(url, user, password);
 System.out.println("conexion open.");
 } catch (Exception e) {
 System.out.println("openConexion() " + e.getMessage());
 }
 return conexion;
 }

 public static Conexion closeConexion() {
 try {
 if (conexion != null && !conexion.isClosed()) {
 conexion.close();
 System.out.println("conexion close.");
 }
 } catch (SQLException e) {
 System.out.println("closeConexion() " + e.getMessage());
 }
 return conexion;
 }
}
```

## BeanModel

Como los que solían hacerse en POO-I.

```
public class EditorialesModel {

 private CallableStatement cs;
 private Conexion conexion;
 private ResultSet rs;
 private Statement st;

 public List<Editorial> listarEditoriales(){
 ArrayList<Editorial> editoriales = null;
 try {
 String sql = "CALL splistarEditoriales()";
 editoriales = new ArrayList<>();
 conexion = Conexion.abrirConexion();
 cs = conexion.prepareCall(sql);
 rs = cs.executeQuery();

 while (rs.next()) {
 Editorial editorial = new Editorial();
 editorial.setIdeditorial(rs.getInt("ideditorial"));
 editorial.setNombre(rs.getString("nombre"));
 editorial.setContacto(rs.getString("contacto"));
 editorial.setTelefono(rs.getString("telefono"));
 }
 }
 }
}
```

```

<a href="#"<%=url%>AutoresController?op=listar" class="btn btn-outline-primary"> Volver

</form>
</div>
</div>

```

```

<input type="number" name="precio" placeholder="Precio"> <p></p>
<section class="section">
<select class="form-select" name="genero" class="select">
<option selected>Generos</option>
<%=
if(listaGeneros != null){
for(int i=0; i<listaGeneros.size(); i++){
<option><%= listaGeneros.get(i) %></option>
<%=
}
}
</select>
<%=--
<a href="#"<%=url%>GenerosController?op=nuevo" >Nuevo genero <p></p> --%>
</section>

```

```

 editoriales.add(editorial);
}
conexion = Conexion.cerrarConexion();
} catch (Exception e) {
 System.out.println("ListarEditoriales() " + e.getMessage());
}
return editoriales;
}

public int insertarEditorial(Editorial editorial) {
 int filasAfectadas = 0;
 try {
 String sql = "CALL spInsertarEditorial(?, ?, ?)";
 conexion = Conexion.abrirConexion();
 cs = conexion.prepareStatement(sql);

 cs.setString(1, editorial.getNombre());
 cs.setString(2, editorial.getContacto());
 cs.setString(3, editorial.getTelefono());
 filasAfectadas = cs.executeUpdate();
 if (filasAfectadas == 0) {
 System.out.println("Editorial NO insertada en model");
 }
 } catch (SQLException ex) {
 System.out.println("Error en insertarEditorial() " + ex.getMessage());
 }
 conexion = Conexion.cerrarConexion();
 return filasAfectadas;
}

public Editorial obtenerEditorial(int idEditorial) throws SQLException {
 Editorial editorial = null;
 try {
 String sql = "CALL spObtenerEditorial(?)";
 conexion = Conexion.abrirConexion();
 cs = conexion.prepareStatement(sql);
 cs.setInt(1, idEditorial);
 rs = cs.executeQuery();

 if (rs.next()) {
 editorial = new Editorial();
 editorial.setIdEditorial(rs.getInt("idEditorial"));
 editorial.setNombre(rs.getString("nombre"));
 editorial.setContacto(rs.getString("contacto"));
 editorial.setTelefono(rs.getString("telefono"));
 }
 } catch (SQLException ex) {
 System.out.println("obtenerEditorial() " + ex.getMessage());
 }
 conexion = Conexion.cerrarConexion();
 return editorial;
}

public int modificarEditorial(Editorial editorial) {
 int filasAfectadas = 0;
 try {
 String sql = "CALL spModificarEditorial(?, ?, ?, ?)";
 conexion = Conexion.abrirConexion();
 cs = conexion.prepareStatement(sql);

 cs.setInt(1, editorial.getIdEditorial());
 cs.setString(2, editorial.getNombre());
 cs.setString(3, editorial.getContacto());
 cs.setString(4, editorial.getTelefono());
 filasAfectadas = cs.executeUpdate();
 if (filasAfectadas == 0) {
 System.out.println("Editorial NO modificada en model");
 }
 }
}

```

```

 conexion = Conexion.cerrarConexion();
 } catch (SQLException ex) {
 System.out.println("ModificarEditorial() " + ex.getMessage());
 conexion = Conexion.cerrarConexion();
 }
 return filasAfectadas;
}

public int eliminarEditorial(int ideditorial) {
 int filasAfectadas = 0;
 try {
 String sql = "CALL spEliminarEditorial(?)";
 conexion = Conexion.abrirConexion();
 cs = conexion.prepareStatement(sql);
 cs.setInt(1, ideditorial);
 filasAfectadas = cs.executeUpdate();
 if (filasAfectadas == 0) {
 System.out.println("Editorial NO eliminada en model");
 }
 } catch (SQLException ex) {
 System.out.println("eliminarEditorial() " + ex.getMessage());
 }

 conexion = Conexion.cerrarConexion();
 return filasAfectadas;
}

public List<String> listarNombresEditoriales() {
 List<String> editoriales = new ArrayList<>();
 try {
 String sql = "SELECT e.nombre FROM editorial e";
 conexion = Conexion.abrirConexion();
 st = conexion.createStatement();
 rs = st.executeQuery(sql);

 while(rs.next()) {
 editoriales.add(rs.getString(1));
 }

 conexion = Conexion.cerrarConexion();
 } catch (Exception e) {
 System.out.println("listarNombresEditoriales() " + e.getMessage());
 }

 return editoriales;
}
}

```

## BeansController

```

@WebServlet(name = "LibrosController", urlPatterns = { "/LibrosController" })
public class LibrosController extends HttpServlet {

 private static final long serialVersionUID = 1L;
 private GenerosModel generosModel = new GenerosModel();
 private LibrosModel librosModel = new LibrosModel();

 public LibrosController() {
 super();
 }

 protected void processRequest(HttpServletRequest request, HttpServletResponse response)
 throws ServletException, IOException {
 response.setContentType("text/html;charset=UTF-8");
 try (PrintWriter out = response.getWriter()) {

```

```

<input type="text" name="nombre" <%= (nombre == null) ? "placeholder=Nombre" :
"value=" + nombre + " " %> id="nombre" class="form-control" > <p></p>
<input type="text" name="nacionalidad" <%= (nacionalidad == null) ?
"placeholder='Nacionalidad'" : "value=" + nacionalidad + " " %> id="nacionalidad" class="form-
control" > <p></p>

<input type="submit" value="Guardar" name="Guardar" class="btn btn-primary">
<a href="<%=url%>AutoresController?op=listar" class="btn btn-outline-primary">
Volver </form>
</div>
</div>

```

## editarBeans

```

<script>
function validarO {
 let resp = true;
 resp = validarCampo('nombre', 'Ingrese el nombre del autor.') && resp;
 resp = validarCampo('nacionalidad', 'Ingrese la nacionalidad del autor.') && resp;

 return resp;
}

function validarCampo(parametro, mensaje){
 let value = true;
 const aux = document.getElementById(parametro).value.trim();
 if(aux == null || aux == ''){
 alert(mensaje);
 document.getElementById(parametro).focus();
 value = false;
 }
 return value;
}
</script>
</head>
<body>
<%
 Autor autor;
 HttpSession sesion = request.getSession();
 if (request.getAttribute("autor") == null) {
 autor = new Autor();
 } else {
 autor = (Autor) request.getAttribute("autor");
 }

 String nombre = (String) request.getAttribute("nombre");
 System.out.println(nombre==null? "nombre null" : nombre+"!!!");
 String nacionalidad = (String) request.getAttribute("nacionalidad");
 System.out.println(nacionalidad==null? "nacionalidad null" : nacionalidad+"!!!");
%>
<%@ include file="/cabeceraMenu.jsp" %>
<div class="container">

<h3>Editar autor</h3> <p></p>
<div class="form-group">
<form role="form" action="<%=url %>AutoresController" method="POST" onsubmit="return
validar()">
 <input type="hidden" name="op" value="modificar">
 <input type="hidden" name="idautor" value="<%=autor.getIdAutor() %>" >
 <input type="text" name="nombre" value="<%= (nombre == null) ? autor.getNombre() :
nombre %>" class="form-control" id="nombre"> <p></p>
 <input type="text" name="nacionalidad" value="<%= (nacionalidad == null) ?
autor.getNacionalidad() : nacionalidad %>" class="form-control" id="nacionalidad"> <p></p>

 <input type="submit" value="Guardar" name="Guardar" class="btn btn-primary">

```

```

<th>Nombre del autor</th>
<th>Nacionalidad del autor</th>
<th>Operaciones</th>
</tr>
</thead>
<tbody>
<%
if (listaAutores != null) {
for (Autor autor : listaAutores) {
%>
<tr>
<td><%=autor.getIdAutor() %></td>
<td><%=autor.getNombre() %></td>
<td><%=autor.getNacionalidad() %></td>
<td>
<a href="#"javascript:modificar(<%=autor.getIdAutor() %>)" class="btn btn-outline-warning"> Modificar
<%=--
href="#"<%=url %>EditorialesController?op=obtener&ideditorial=<%=editorial.getIdEditorial() %>" --%>
btn-outline-danger"> Eliminar
<a href="#"javascript:eliminar(<%=autor.getIdAutor() %>)" class="btn
</td>
</tr>
<%
}
}
</tbody>
</table>
</div>

```

#### nuevoBean

```

<script>
function validar() {
let resp = true;
resp = validarCampo('nombre', 'Ingrese el nombre del autor.') && resp;
resp = validarCampo('nacionalidad', 'Ingrese la nacionalidad del autor.') && resp;
return resp;
}
function validarCampo(parametro, mensaje){
let value = true;
const aux = document.getElementById(parametro).value.trim();
if(aux == '' || aux == null){
alert(mensaje);
document.getElementById(parametro).focus();
value = false;
}
return value;
}
</script>
<%
String nombre = (String) request.getAttribute("nombre");
String nacionalidad = (String) request.getAttribute("nacionalidad");

<%@ include file="/cabeceraMenu.jsp" %>
<div class="container">

<h3> Nuevo Autor </h3>
<div class="form-group">
<form role="form" action="#"<%=url %>AutoresController" method="POST" onsubmit="return
validar()">
<input type="hidden" name="op" value="insertar">

```

```

if (request.getParameter("op") == null) {
listar(request, response);
return;
}

String operacion = request.getParameter("op");
switch (operacion) {
case "listar":
listar(request, response);
break;
case "nuevo":
listarSelects(request, response);
break;
case "insertar":
insertar(request, response);
break;
case "obtener":
obtener(request, response);
break;
case "modificar":
modificar(request, response);
break;
case "eliminar":
eliminar(request, response);
break;
}
} catch (Exception e) {
System.out.println("processRequest() " + e.getMessage());
}
}

protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
}

protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
processRequest(request, response);
}

protected void listarSelects(HttpServletRequest request, HttpServletResponse response) {
try {
request.setAttribute("nombresEditoriales",
editorialesModel.listarNombresEditoriales());
request.setAttribute("nombresGeneros", generosModel.listarNombresGeneros());
request.setAttribute("nombresAutores", autoresModel.listarNombresAutores());
request.getRequestDispatcher("/libros/nuevoLibro.jsp").forward(request, response);
} catch (Exception e) {
System.out.println("Error en listarSelects() " + e.getMessage());
}
}

protected void validar(HttpServletRequest request, HttpServletResponse response) {
boolean hayErrores = false;
List<String> listaErrores = new ArrayList<String>();
try {
validarParametro(request, response, "nombre", listaErrores,
Mensajes.AUTOR_NOMBRE_ERROR);
validarParametro(request, response, "nacionalidad", listaErrores,
Mensajes.AUTOR_NACIONALIDAD_ERROR);
if (listaErrores.size() > 0) {
hayErrores = true;
}
request.setAttribute("respuesta", hayErrores);
request.setAttribute("listaErrores", listaErrores);
} catch (Exception e) {
System.out.println("validar() " + e.getMessage());
}
}

```

```

 }
 return hayErrores;
}

protected void validarParametro(HttpServletRequest request, HttpServletResponse response,
String parametro, List<String> listaErrores, String mensaje) {
 try {
 String aux = request.getParameter(parametro);
 if(aux.isEmpty() || aux == null) {
 listaErrores.add(mensaje);
 request.setAttribute(parametro, null);
 }else {
 request.setAttribute(parametro, request.getParameter(parametro));
 }
 } catch (Exception e) {
 System.out.println("validarParametro " + e.getMessage());
 }
}

protected void listar(HttpServletRequest request, HttpServletResponse response) {
 try {
 request.setAttribute("listarLibros", librosModel.listarLibros());
 request.getRequestDispatcher("/libros/listaLibros.jsp").forward(request,
response);
 } catch (Exception e) {
 System.out.println("listar " + e.getMessage());
 }
}

protected void insertar(HttpServletRequest request, HttpServletResponse response) {
 try {
 if(!validar(request, response)) {
 Autor autor = new Autor();
 autor.setNombre(request.getParameter("nombre"));
 autor.setNacionalidad(request.getParameter("nacionalidad"));

 if (autorModelo.insertarAutor(autor) > 0) {
 request.getSession().setAttribute("exito", "autor creado");
 } else {
 request.getSession().setAttribute("fracaso", "autor NO creado");
 }
 response.sendRedirect(request.getContextPath() +
"/AutoresController?op=listar");
 }else {
 request.getRequestDispatcher("/autores/nuevoAutor.jsp").forward(request,
response);
 }
 } catch (Exception e) {
 System.out.println("insertar " + e.getMessage());
 }
}

private void obtener(HttpServletRequest request, HttpServletResponse response) {
 try {
 if(!validar(request, response)) {
 int idautor = Integer.parseInt(request.getParameter("idautor"));
 Autor autor = autorModelo.obtenerAutor(idautor);

 if (autor != null) {
 request.setAttribute("autor", autor);
 }
 request.getRequestDispatcher("/autores/editarAutor.jsp").forward(request, response);
 }
 } catch (Exception e) {
 System.out.println("obtener " + e.getMessage());
 }
}

```

6

```

 }

 private void modificar(HttpServletRequest request, HttpServletResponse response) {
 try {
 if(!validar(request, response)) {
 Autor autor = new Autor();
 autor.setIdAutor(Integer.parseInt(request.getParameter("idautor")));
 autor.setNombre(request.getParameter("nombre"));
 autor.setNacionalidad(request.getParameter("nacionalidad"));
 request.setAttribute("autor", autor);

 if (autorModelo.modificarAutor(autor) > 0) {
 request.getSession().setAttribute("exito", "autor modificado ");
 } else {
 request.getSession().setAttribute("fracaso", "autor NO modificado");
 }
 response.sendRedirect(request.getContextPath() +
"/AutoresController?op=listar");
 }else {
 request.getRequestDispatcher("/autores/editarAutor.jsp").forward(request,
response);
 }
 } catch (Exception e) {
 System.out.println("modificar " + e.getMessage());
 }
 }

 protected void eliminar(HttpServletRequest request, HttpServletResponse response) {
 try {
 int idlibro = Integer.parseInt(request.getParameter("idlibro"));

 if(librosModel.eliminarLibro(idlibro) > 0) {
 request.getSession().setAttribute("exito", "libro eliminado");
 } else {
 request.getSession().setAttribute("fracaso", "libro NO eliminado");
 }
 response.sendRedirect(request.getContextPath() + "/LibrosController?op=listar");
 } catch (Exception e) {
 System.out.println("eliminar " + e.getMessage());
 }
 }
}

```

## Jsp

### listarBeans

```

<script>
function modificar(id) {
 if(confirm("&Desea modificar el registro?") == true){
 location.href = "AutoresController?op=obtener&idautor=" + id;
 }
}
</script>
</head>
<body>

<% List<Autor> listaAutores = (List<Autor>) request.getAttribute("ListaAutores"); %>
<%@ include file="/cabeceraMenu.jsp" %>
<div class="container">

 <a href="<%=url%>AutoresController?op=nuevo" class="btn btn-primary"> Nuevo Autor

 <table id="tabla" class="table table-bordered">
 <thead>
 <tr>
 <th>Id del autor</th>

```

7