

## Configuración del archivo web.xml:

### Incluir header.jsp y footer.jsp en todas las páginas necesarias:

Al principio y final de cada página

```
<jsp:include page="header.jsp"></jsp:include>    código de la página ...
```

```
<jsp:include page="header.jsp"></jsp:include>
```

O bien en el archivo web.xml en la pestaña pages vamos a jsp group y en url patterns añadimos las páginas en las que queremos incluir el header y footer que se deben incluir en su casilla, si vamos a la pestaña code el código resultante sería algo así

```
<jsp-config>
  <jsp-property-group>
    <url-pattern>/index.jsp</url-pattern>
    <url-pattern>/WEB-INF/view/category.jsp</url-pattern>
    <url-pattern>/WEB-INF/view/checkout.jsp</url-pattern>
    <url-pattern>/WEB-INF/view/confirmation.jsp</url-pattern>
    <url-pattern>/WEB-INF/view/cart.jsp</url-pattern>
    <include-prelude>/WEB-INF/jspf/header.jsp</include-prelude>
    <include-coda>/WEB-INF/jspf/footer.jsp</include-coda>
  </jsp-property-group>
</jsp-config>
```

- Algunas versiones de tomcat pueden solo admitir un parámetro url-pattern, en este caso se podría hacer de la siguiente manera:

```
<jsp-config>
  <jsp-property-group>
    <url-pattern>*.jsp</url-pattern>
    <include-prelude>/WEB-INF/jspf/header.jsp</include-prelude>
    <include-coda>/WEB-INF/jspf/footer.jsp</include-coda>
  </jsp-property-group>
</jsp-config>
```

### Flujo de la aplicación:

Se le da un Alias “ControllerServlet” a la clase “ControllerServlet” del paquete “controller” y load on start up lo ponemos en 1 (true) para que ejecute el método init ( ) al principio.

```
<servlet>
  <servlet-name>ControllerServlet</servlet-name>
  <servlet-class>controller.ControllerServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```

Se crea un mapeo entre las paginas de la aplicación que llamen todas al Servlet

```
<servlet-mapping>
  <servlet-name>ControllerServlet</servlet-name>
  <url-pattern>/category</url-pattern>
  <url-pattern>/addToCart</url-pattern>
  <url-pattern>/viewCart</url-pattern>
  <url-pattern>/updateCart</url-pattern>
  <url-pattern>/checkout</url-pattern>
  <url-pattern>/purchase</url-pattern>
</servlet-mapping>
```

Despues en el servlet se controla desde donde le han llamado, colocamos el codigo en el metodo doPost o doGet del servlet dependiendo de como se haga la llamada.

```
String userPath= request.getServletPath();
String url=null;

if(userPath.equals("/category")){

  //Fijamos la url que queremos visualizar despues de ejecutar nuestro codigo
  url="/WEB-INF/view/category.jsp";
  //hacemos lo que sea
}
//redireccionamos a la pagina de visualizacion
request.getRequestDispatcher(url).forward(request, response);
```

### **Rutas a nuestras carpetas de imagenes;**

Se pueden crear alias para las rutas

```
<context-param>
  <param-name>productImagePath</param-name>
  <param-value>images/products</param-value>
</context-param>
<context-param>
  <param-name>categoryImagePath</param-name>
  <param-value>images/category</param-value>
</context-param>
```

Despues para incluirlas en los archivos.jsp con este codigo accedemos , initParam se refiere al contexto de la aplicación ...

```

```