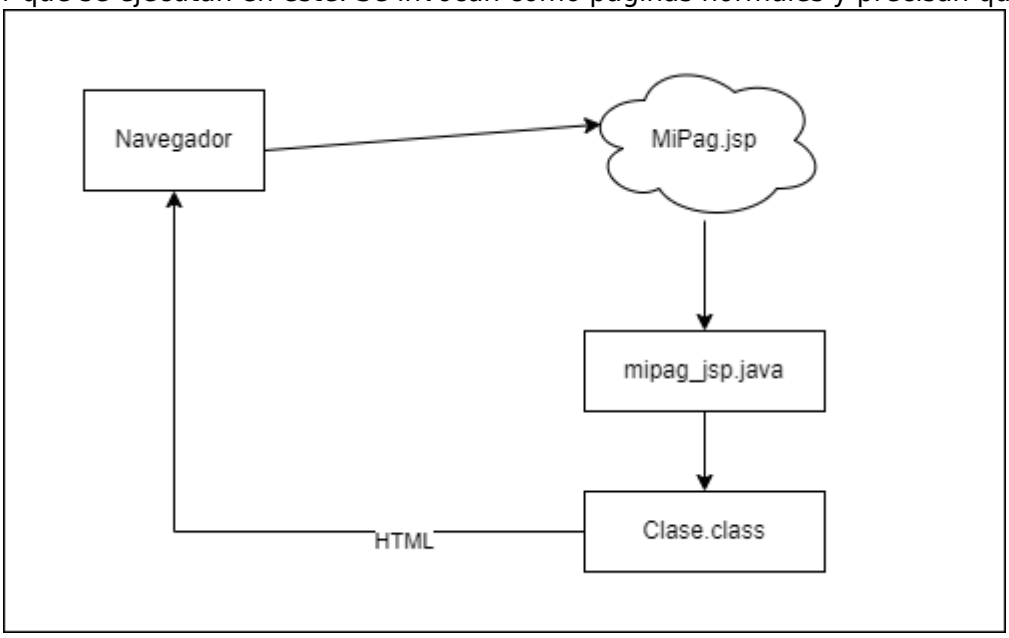
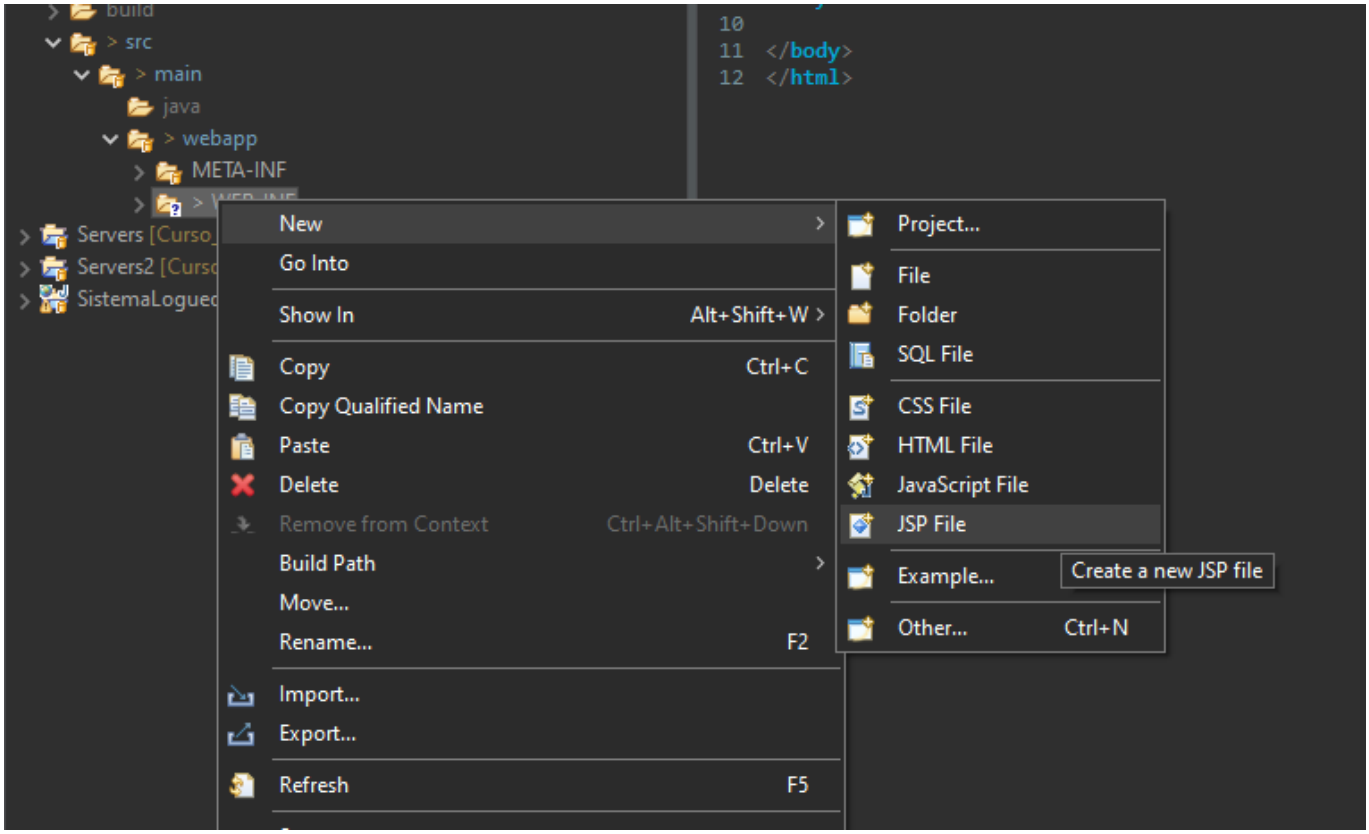
JSP: Java Services Page

Busca separar la lógica de la aplicación de la lógica de presentación. Tenemos páginas compiladas por el servidor que se ejecutan en este. Se invocan como páginas normales y precisan que el servidor entienda JSP.



Implementación

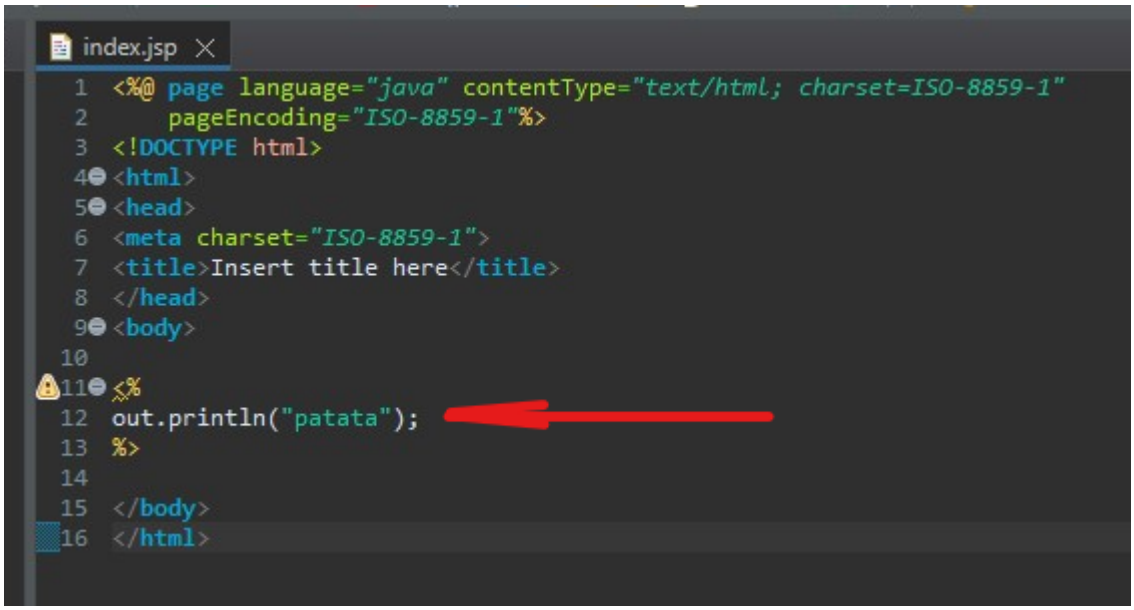
En la ruta src/main/webapp/WEB-INF crearemos el archivo Index.jsp:



Tras eso se nos generará el siguiente archivo:



A este archivo podemos añadir código de java introduciéndolo entre “<% %>”:




```
<%
out.println("patata");
%>
```

Otros Ejemplos

ejemplo.jsp

```
<body>
  El resultado de 5*6 es: <h1><%=5*6%></h1>
```

```
  <%for (int i=0; i<=4;i++)%>
    <%=i%> * 10 = <% i*10%><br>
</body>
```

Etiquetas de JSP

```
<%@ Directiva de configuración de la página%>
  <%@ page errorPage = "error.jsp"%>

<%! Declaración de variables y funciones%>
  <%! int i=0; %>

<%Bloque de código Java%>
  <%for(i=0;i<5;i++){%> <%}%>
```

Modificaciones del web.xml para declaración de servlet

web.xml

```
<Servlet>
  <servlet-name>Servlet<servlet-name>//nombre del servlet
  <servlet-
class>paquete.paqueton.servlets.ImplementacionServlet<servlet-
class>//Clase que lo implementa
</servlet>
<servlet-mapping>
  <servlet-name>Servlet</servlet-name>
  <url-pattern>/servlet</url-pattern>
</servlet-mapping>
```

Modelo Vista Controlador (MVC) y Paso de Atributos

Esta parte se centra en la capa de negocio.

Control

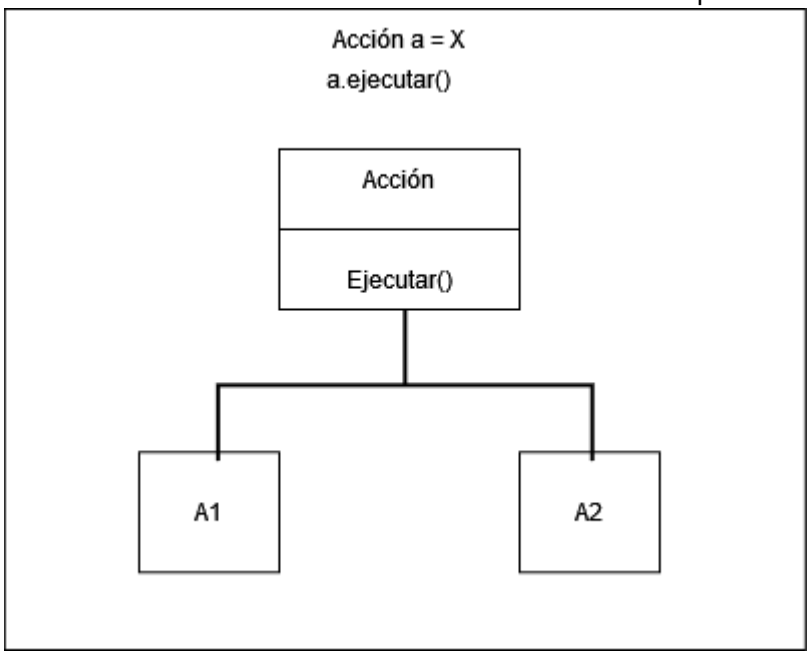
Es un servlet único que recibirá todas las peticiones y las redirigirá a otra clase. Este servlet comprueba los permisos e invoca las funciones o modelos correspondientes. Para interactuar con el control se usa la URL seguida de un “?” y un “ACTION_ID” que indique la acción a realizar:

```
http://localhost:8080/contexto/control?ACTION_ID=LEER_CORREO
```

En esta URL sería Control el Servlet que controla todas las peticiones y LEER_CORREO el identificador de modelo o acción a ejecutar.

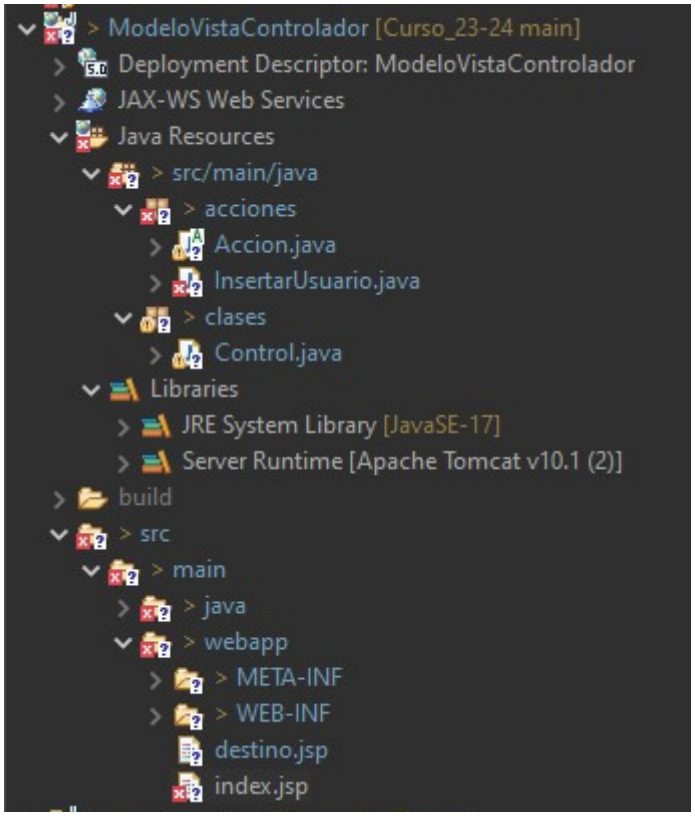
Modelo

Realiza toda la funcionalidad, se podría decir que es una clase padre de la que heredan todas las demás. Suele haber varios modelos dentro de una aplicación.



Implementación

El proyecto seguirá la siguiente estructura en la que separaremos por un lado el servlet control y por otro lado las clases:



Servlet Control

Control.java

```
package clases;

import jakarta.servlet.ServletConfig;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.Hashtable;

import acciones.Accion;
import acciones.InsertarUsuario;

/**
 * Servlet implementation class Control
 */
public class Control extends HttpServlet {
    private static final long serialVersionUID = 1L;
    private Hashtable<String, Accion> acciones; //Tabla hash

    public static final String PARAM_ACTION_ID = "ID_ACCION";

    /**
```

```
     * @see HttpServlet#HttpServlet()
     */
    public Control() {
        super();
        // TODO Auto-generated constructor stub
    }


    @Override
    public void init(ServletConfig config) throws ServletException {
        // TODO Auto-generated method stub

        super.init(config);
        this.acciones = new Hashtable<String, Accion>();//Tabla de
acciones
        this.acciones.put("INSERTAR_USUARIO", new
InsertarUsuario());//insertamos accion en tabla hash
    }


    /**
     * @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doGet(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        String jsp = "index.jps";
        String paramIdAccion =
request.getParameter("ID_ACCION");//recibimos como parametro el ID de
accion
        Accion accion = this.acciones.get(paramIdAccion);//Obtenemos la
accion a realizar de la tabla hash
        jsp = accion.ejecutar(request, response);//Nos devuelve la JSP
a la que hay que ir para realizar la acción
        request.getRequestDispatcher(jsp).forward(request, response);
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request,
HttpServletResponse response)
     */
    protected void doPost(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException {
        // TODO Auto-generated method stub
        doGet(request, response);
    }

}
```

Clase Accion

Accion.java

```
package acciones;

import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public abstract class Accion {
    //Esta es la clase Padre
    public abstract String ejecutar(HttpServletRequest request,
HttpServletResponse response) throws ServletException, IOException;

}
```

Clase InsertarUsuario

InsertarUsuario.java

```
package acciones;
import java.io.IOException;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

public class InsertarUsuario extends Accion {

    @Override
    public String ejecutar(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {

        System.out.println("Toda la funcionalidad de insertar
usuario");

        return "destino.jsp";
    }

}
```

index.jsp

index.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<a href="control?<%= Control.PARAM_ACTION_ID
%>>=INSERTAR_USUARIO">Insertar</a>
</body>
</html>
```

destino.jsp

destino.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
Destinooo
</body>
</html>
```

From:
https://knoppia.net/ - **Knoppia**

Permanent link:
https://knoppia.net/doku.php?id=dad2:jsp&rev=1708517106

Last update: **2024/02/21 12:05**