

Práctica 1-Aplicación Web utilizando JSF, JSP y servlets de Java

M.I. Capel

ETS Ingenierías Informática y
Telecomunicación
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Granada
Email: manuelcapel@ugr.es

DDBCS
Máster Universitario en Ingeniería Informática



Índice

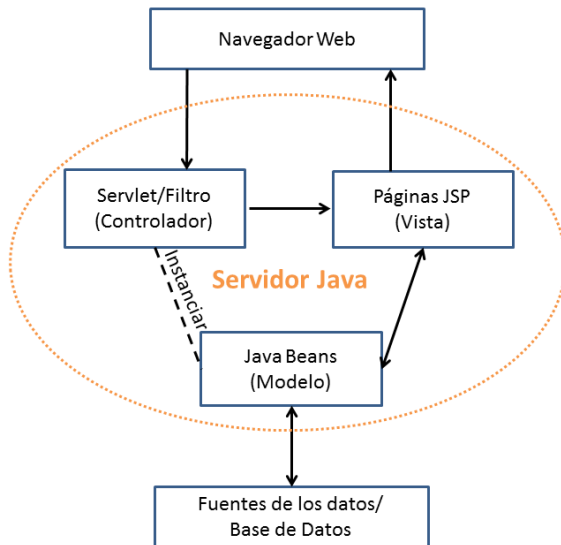
- 1 Práctica 1-Aplicación Web utilizando JSF, JSP y servlets de Java

Marco de Trabajo JSF

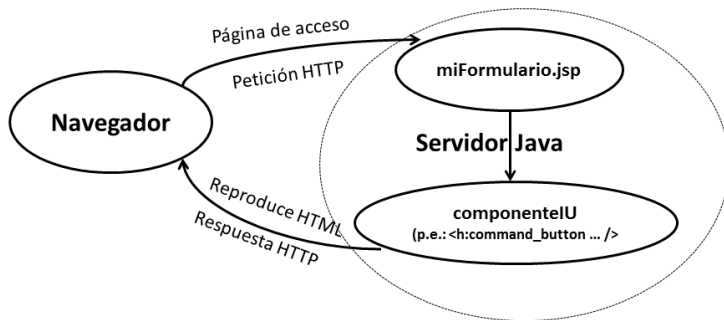
Definition

EL Sistema *Java Server Faces* es un marco de trabajo basado en el patrón arquitectónico MVC que simplifica la construcción de Interfaces de Usuario (IU) para aplicaciones basadas en servidores utilizando componentes-IU dentro de una página (JSP).

MVC de una aplicación que sólo usa páginas JSP



Arquitectura de una aplicación que usa la tecnología JSF



Marco de Trabajo JSF II

Beneficios

- Facilita la conexión entre “*widgets*” de IU con fuentes de datos y manejadores de eventos en la parte del servidor
- Proporciona un API estándar para desarrollar componentes software
- Permite la reutilización y extensión de los estándares de componentes IU actuales
- Facilita la transferencia de datos entre componentes IU
- Gestiona un estado de componente-IU pasando por múltiples servidores
- Conecta los eventos (parte–cliente) con el código de aplicación que ha de tratarlos (parte–servidor)

Ventajas de utilizar la tecnología JSF

- Con la tecnología JSF podemos conectar eventos generados en el cliente, con el código de la aplicación Web en el lado del servidor
- Hacer corresponder los *componentes IU* con los objetos del lado del servidor

Ventajas de utilizar la tecnología JSF-II

La página JSP gestiona todos los objetos que intervienen en la aplicación Web:

- objetos componentes: etiquetas del tipo
`< h : command_button... >`,
- los escuchadores de las acciones que se inician desde los objetos *validadores* y *conversores*, que suelen estar incluidos en las etiquetas de los componentes:
`< f : validate _longrange minimum="0"maximum="10"/>`
- los objetos del modelo que contienen los datos de la aplicación Web,
- el resto de la funcionalidad de los componentes de la aplicación.

Arquitectura Software de JSF

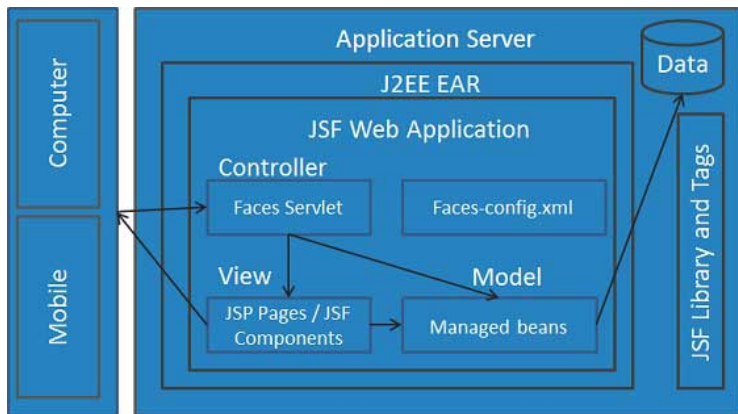


Figura: Arquitectura de JSF

“Controller”

Faces Servlet

- Se encarga de controlar todas la solicitudes que provienen del cliente
- Inicializa todos los componentes del marco de trabajo JSF antes de que se muestre la página de JSP
- Los servicios básicos para construirlo se importan de `javax.faces.webapp.FacesServlet`
- Hay que programarlo en el archivo `web.xml`: indica que busque entre los archivos acabados en `.jsf` y `.xhtml` para las vistas

Managed Beans de Java

Concepto:

Una clase bean de Java pero registrada dentro del marco de trabajo JSF.

Elementos de un *managed bean*:

- Recordar que los *beans* con serializables
- Poseen métodos `getter()` y `setter()`
- La lógica del negocio: la información asociada a un *bean* contenida en formularios HTML de aplicación

“Model”

- Se trata de un paquete incluido en `src/main/java`, que contiene las clases `.java` como *Managed Beans* (MB)
- Los “beans” de Java pueden ser gestionados por el JSF utilizando una configuración XML o mediante **anotaciones**:
 - `@ManagedBean(name= , eager=)`
 - `@(Request|None|View|Session|Application|Custom) Scoped`
 - `@ManagedProperty`

“Model”—II

```
package prueba ;  
import javax.faces.bean.ManagedBean ;  
import javax.faces.bean.ManagedProperty ;  
import javax.faces.bean.RequestScoped ;  
  
@ManagedBean (name = "holaMundo", eager = true)  
@RequestScoped  
public class HolaMundo{  
    @ManagedProperty ( value ="#{mensaje}" )  
    //completar  
    public HolaMundo ( ){  
        System.out.println( "HolaMundo ha comenzado ! " ) ;  
    }  
    public String getMensaje( ){  
        //completar  
    }  
    public void setBeanMensaje ( Mensaje m ) {  
        //completar  
    }  
}
```

“View”

Creación de una página JSF

Se articula creando una página `home.xhtml` que puede ser procesada por un explorador de XML en `src/main/webapp/`

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
    Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.
        dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>JSF Tutorial!</title>
</head>
<body>
    #{holaMundo.mensaje}
</body>
</html>
```

Estructura en el navegador de proyectos Eclipse

The screenshot displays the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Design, Window, and Help. Below the menu is a toolbar with various icons for file operations and development tools.

The **Package Explorer** on the left shows the project structure for 'holamundo':

- holamundo
 - src/main/resources
 - JRE System Library [jdk1.8.0_05]
 - Referenced Libraries
 - src
 - main
 - java
 - prueba
 - HolaMundo.java
 - Mensaje.java
 - package-info.java
 - webapp
 - WEB-INF
 - home.xhtml
 - web.xml** (selected)
 - index.jsp
 - target
 - holamundo
 - META-INF
 - WEB-INF
 - classes
 - lib
 - jsf-api-2.1.7.jar
 - jsf-impl-2.1.7.jar
 - home.xhtml
 - web.xml** (selected)
 - index.jsp
 - maven-archiver
 - surefire
 - test-classes
 - holamundo.war
 - pom.xml

The **web.xml** file is open in the editor, showing the following XML content:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://java.sun.com/xml/ns/javaee"
  xmlns:web="http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
  id="WebApp_ID" version="2.5">
  <welcome-file-list>
    <welcome-file>faces/home.xhtml</welcome-file>
  </welcome-file-list>
  <!--
    FacesServlet es un servlet principal responsable de gestionar toda
    Actua como un controlador central.
    Este servlet inicializa los componentes de JSF antes de que se muestre
  -->
  <servlet>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <servlet-mapping>
    <servlet-name>Faces Servlet</servlet-name>
    <servlet-url-patterns>/faces/*</servlet-url-patterns>
  </servlet-mapping>
</web-app>
```

Below the editor, the **Problems** tab is active, showing the following output:

```
<terminated> C:\Program Files\Java\jdk1.8.0_05\bin\javaw.exe (15/10/2014 10:43:30)
[INFO] --- maven-compiler-plugin:2.3.2:testCompile (default-testCompile) @ 1
[INFO] No sources to compile
[INFO] --- maven-surefire-plugin:2.10:test (default-test) @ holamundo ---
```

Referencias

- Tutorial JSP : <http://www.jsptut.com/>
- Tutorial EJB:
<http://www.tutorialspoint.com/ejb/>
- Java Server Faces en Wikipedia : http://en.wikipedia.org/wiki/JavaServer_Faces
- Archivos Java WAR : <http://www.yolinux.com/TUTORIALS/Java-WAR-files.html>
- Maven Repository : <http://mvnrepository.com/artifact/org.apache.commons>
- Maven POM Reference :
<http://maven.apache.org/pom.html>