



# LA ARQUITECTURA MODELO VISTA CONTROLADOR

## LA ARQUITECTURA MODELO VISTA CONTROLADOR

### Introducción a los Framework MVC

#### Framework MVC

El patrón MVC constituye la arquitectura más adecuada de cara a crear aplicaciones para la Web. Hemos visto sus beneficios y características, así como la manera de ponerlo en práctica utilizando componentes Java EE.

No obstante, la creación de aplicaciones Java EE extensas basadas en el patrón MVC, puede resultar bastante pesada y tediosa utilizando directamente los componentes Java EE.

Para simplificar la construcción de aplicaciones MVC, diferentes comunidades de desarrolladores han creado lo que conocemos como frameworks, que no son más que conjuntos de utilidades (librerías de clases, archivos de configuración, acciones JSP, etc.) especialmente pensadas para facilitar la construcción de aplicaciones Web Java EE, siguiendo el patrón MVC.

#### Facilidades proporcionadas por los frameworks MVC.

Entre las facilidades que nos proporcionan en general los frameworks MVC para la construcción de aplicaciones destacamos:

- Implementación de Front controller. La mayoría de los frameworks ya incorporan un front-controller para el control de peticiones, lo que nos ahorra el tema de los parámetros en la URL y los incómodos if dentro de un servlet
- Reglas de navegación declarativas. El control del flujo de peticiones suele realizarse en un archivo de configuración xml, en vez de tener que definirlo dentro del propio código de la aplicación
- Simplificación en la recogida de parámetros. La mayoría de los frameworks incorporan alguna facilidad que simplifica la recogida de parámetros en JavaBeans, sin necesidad de recurrir a las expresiones `request.getParameter`.
- Librerías de acciones JSP. Todos los frameworks MVC incluyen librerías de acciones JSP propias que nos van a facilitar la creación de las interfaces gráficas de las vistas

#### Frameworks MVC más populares

A lo largo de los últimos años han surgido numerosos frameworks MVC, pero solo unos cuantos han sido lo suficientemente maduros como para que hayan sido ampliamente aceptados por la comunidad de desarrolladores Java EE

## LA ARQUITECTURA MODELO VISTA CONTROLADOR

Entre los más populares destacamos:

- Struts
- JSF
- Sping

### Struts

Struts es un framework desarrollado por el grupo Apache en el año 2002. Fue el framework que tuvo más aceptación en la época, por lo que empezó a ser utilizado masivamente por la comunidad de desarrolladores Java. Actualmente, es uno de los frameworks MVC más utilizados

Su principal aportación es la estructuración y simplificación del controlador, aunque también proporciona un conjunto de librerías de acciones para ayudar a la implementación de las vistas. Struts no incluye ninguna utilidad para la implementación del modelo ni tampoco establece ningún condicionante sobre cómo ha de ser desarrollado éste, integrándose perfectamente con las tecnologías utilizadas de forma habitual para la creación de la lógica de negocio de una aplicación, como son las clases estándares Java o los Enterprise Java Beans

Incorpora un archivo de configuración struts-config.xml, en el que se declaran los controladores de acción y se definen las reglas de navegación. También incorpora un API entre cuyas clases se incluye un front controller, y otra serie de clases para simplificar la construcción del controlador. Dispone también de cuatro librerías de acciones para la creación de vistas.

Desde hace algunos años, el framework ha evolucionado poco, pero dado su grado de penetración en las empresas, se sigue utilizando mucho.

En el año 2008, apareció una nueva versión del framework, conocida como Struts 2, que muy poco tenía que ver con el clásico Struts. Struts 2 incorpora muchas ventajas respecto a Struts 1, pero actualmente no se utiliza tanto como éste

### JSF

Java Server Faces (JSF) es un framework para la construcción de aplicaciones Web siguiendo el patrón MVC, y que fue incorporado por Sun Microsystems a la especificación Java EE en el año 2007.

Al igual que Struts, JSF se centra en las capas controlador y vista de una aplicación, permitiendo que el modelo pueda ser desarrollado con cualquier tecnología Java específica de esta capa, como por ejemplo EJB.

## LA ARQUITECTURA MODELO VISTA CONTROLADOR

Al tratarse de un framework más moderno que Struts, JSF simplifica aún más las tareas de gestión y tratamiento de peticiones y, especialmente, la generación de las vistas.

JSF también incorpora un archivo de configuración `faces-config.xml`, en el que se definen las reglas de navegación, así como los llamados managed bean, que uno de los elementos más importantes que incluye JSF. Un managed bean no es más que una clase POJO que hace las veces de controlador de acción, y de JavaBean para la recogida de parámetros de un formulario.

Como se ha indicado, JSF es un framework muy orientado a la vista. En el API JSF se incluye toda una jerarquía de componentes gráficos que permiten la construcción de interfaces gráficas enriquecidas con poco esfuerzo para el programador.

La filosofía de creación de aplicaciones JSF está pensada para que el programador no tenga que preocuparse de muchos detalles relacionados con la programación Web.

A partir de la versión JSF 2, el framework incorporó una serie de novedades que simplificaban aún más su uso. Entre ellas, la posibilidad de utilizar anotaciones o la integración de AJAX, sin necesidad de realizar ningún tipo de programación JavaScript en la capa cliente

### Spring

Spring es un framework de propósito general surgido a principios del año 2003, que tiene como objetivo facilitar la implantación en las aplicaciones Java de las buenas prácticas definidas en el catálogo de diseño de patrones.

Spring no es un frameworks más, pues a diferencia de otros que se centran únicamente en resolver determinadas capas de una aplicación Web, Spring puede ser utilizado con todos los tipos de desarrollos en Java y en todas las capas en las que se organiza la aplicación.

Spring hace tremendamente sencilla la construcción de aplicaciones empresariales en Java. En este sentido, uno de los calificativos que mejor define a Spring es el de ser un framework no invasivo, lo que realmente significa que puede ser utilizado sobre clases POJO no dependientes de ninguna clase o interfaz del API Spring.

El hecho de que Spring pueda utilizarse en numerosas partes de una aplicación se traduce en una organización modular del propio framework. En la siguiente figura se muestran los módulos que constituyen el framework Spring. Esta organización permite que solamente se utilicen en los desarrollos aquellas partes de Spring que se necesiten.



