|  |
| --- |
| **Dev90++** |
| Arquitectura de Software |
| Sistema de Transporte y Planificación |
|  |
|  |
| **09/12/2012** |

|  |
| --- |
| En este documento indicaremos las características de las tecnologías que se utilizan en el desarrollo de este software. |

# Introducción

El uso de un marco en una aplicación presenta muchas ventajas durante la ejecución de una aplicación de software.

Un marco de trabajo proporciona una infraestructura reutilizable que aborda muchos detalles de implementación comunes a través de las aplicaciones y ayuda a evitar volver a descubrir y re-implementación de soluciones a los problemas comunes de implementación. Además, un marco a menudo se ajusta a un modelo de programación que representa una solución eficaz a los problemas de la construcción de un tipo específico de aplicación.

Para Java 2 Enterprise Edition (J2EE) aplicaciones web, los marcos basados ​​en el patrón de diseño Model-View-Controller (MVC) son considerados los mejores de su clase, y entre ellos, Struts ha sido el más popular.

Más recientemente, el comité de estándares J2EE lanzó el JavaServer Faces (JSF) especificación ofrece un framework MVC con beneficios adicionales.  
En este documento se describen las ventajas del patrón de diseño MVC y proporciona una visión general de la JSF.

Se discute su propósito, se enumeran sus ventajas y desventajas, y proporciona directrices sobre cómo elegir entre ellos.

Model-View-Controller

Model-View-Controller (MVC) es un patrón de diseño que separa interfaz de usuario de una aplicación a partir de su lógica de negocio.

Lo hace por capas de la arquitectura de la aplicación en tres partes: Modelo, Vista y Controlador. La figura 1 muestra la arquitectura MVC como se aplica a aplicaciones Web.

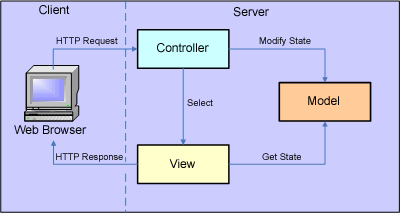


Figura 1: Arquitectura MVC para aplicaciones Web

Modelo  
El modelo representa el estado de la aplicación y define las acciones comerciales que lo modifican (datos persistentes y la lógica de negocios). Puede ser consultado acerca de su estado (por lo general por la vista) y se le pedirá que lo cambie (por lo general por el controlador). No sabe nada acerca de la vista o al controlador.

Vista  
La vista proporciona la presentación del Modelo. Se representa el aspecto de la aplicación, es decir, su interfaz de usuario. Es responsable de la entrada y salida de datos hacia y desde el usuario. La vista puede obtener el modelo de estado, pero no puede modificarlo.

Controlador  
El controlador reacciona a la entrada del usuario e informa al Modelo para cambiar su estado en consecuencia. Específicamente, procesa las solicitudes de entrada de usuario mediante el envío de ellos a las funciones de lógica de negocio apropiados (en el modelo) y la selección de la respuesta al usuario (la vista) basándose en el resultado.

Beneficios  
El patrón de diseño MVC separación de la lógica de negocio a partir de los resultados de la presentación de los siguientes beneficios:

* **Mantenibilidad mejorada**  
  Debido a la Vista y el Modelo de capas están desconectados, puede cambiar la interfaz de usuario sin afectar a las reglas de negocio y vice-versa. El impacto de los cambios es, por lo tanto, reduce al mínimo.
* **Modelo de Reutilización**  
  Se pueden crear múltiples vistas del mismo modelo. Por ejemplo, si su aplicación debe ser compatible con diferentes tipos de dispositivos de cliente (por ejemplo, teléfonos celulares y PDAs), puede crear nuevos puntos de vista específicos para cada tecnología y volver a utilizar el mismo modelo.
* **Separación de Responsabilidades**  
  Funciones de desarrollo pueden ser separados permitiendo que los diferentes miembros del equipo de desarrollo para centrarse en su área de especialización. Diseñadores de páginas web, por ejemplo, puede ser responsable de la capa de la Vista y trabajar de forma independiente de los desarrolladores de Java que se pueden concentrar en la aplicación del controlador y las capas del modelo.

JavaServer Faces

JavaServer Faces (JSF) es un framework de aplicaciones que permite la construcción de aplicaciones Java Web utilizando un enfoque de interfaz de usuario impulsada.

La tecnología proporciona una infraestructura para la creación de aplicaciones Web mediante una interfaz de usuario estándar de modelo de componente que se ejecuta en el servidor.

Su meta es hacer más fácil el desarrollo de aplicaciones web mediante la simplificación del modelo de programación y promoción de la interfaz de usuario orientada, orientada a eventos de desarrollo Web.

Arquitectura

JSF es un framework de aplicaciones basado en MVC. Proporciona una rica arquitectura para la definición de los componentes de interfaz de usuario, la gestión de su estado en el servidor, y el manejo de los eventos generados por el cliente.

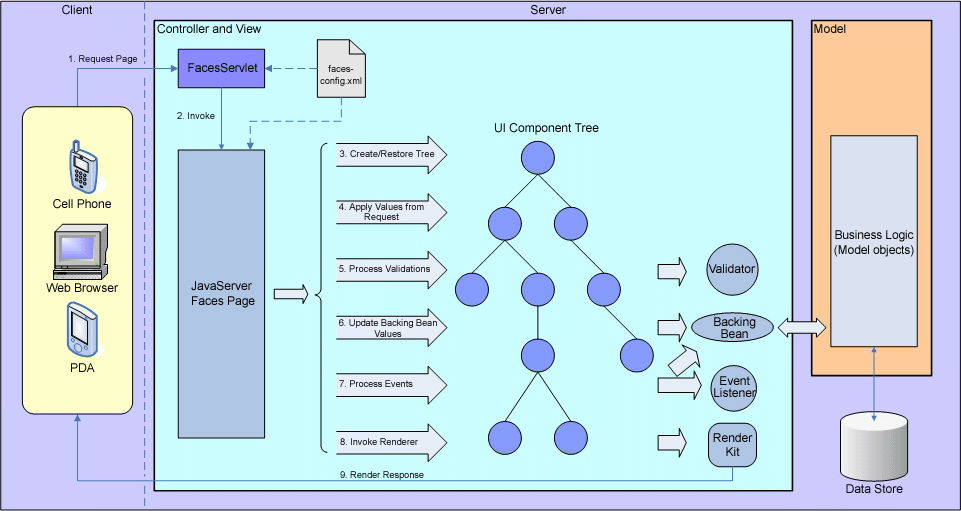
También proporciona soporte para validar la entrada del usuario y el control de navegación de la página. La Figura 3 muestra los componentes principales de la arquitectura de JSF y representa el flujo de procesamiento de una solicitud de cliente.

Figura 3: Arquitectura de componentes JSF y flujo de página Solicitud de Procesamiento

# FacesServlet

El FacesServlet es el punto de entrada para todas las solicitudes de una página JavaServer Faces. Se inicializa los recursos que necesita el marco para el control del ciclo de vida de la página y, posteriormente, invoca la página para procesar la solicitud. Sirve como un controlador frontal a la aplicación.  
Página JavaServer Faces

Una página JavaServer Faces es una página JSP que incluye etiquetas de JavaServer Faces para expresar todos los componentes de la interfaz de usuario que contiene. Cada componente declara su asociación con el valor de una propiedad de bean de respaldo y se especifican los detectores de eventos, validadores y conversores que requiere.

El marco se sincronizará automáticamente los datos de un componente con una propiedad enlazada en su bean de respaldo correspondiente.

Interfaz de usuario “Componente Tree”  
Los componentes de la interfaz de usuario en la página JSF se representan en el servidor mediante un árbol de componentes (también llamado "View"). Cuando se procesa una solicitud por el marco, este árbol se crea (para una solicitud de página inicial) o restaurado a partir de su estado guardado (para peticiones de página siguientes).

Validator  
Un validador se utiliza para validar la entrada del usuario. JSF incluye muchas clases de validación estándar y es compatible con la creación de las personalizadas.

Copia de Bean  
Un bean de respaldo es un JavaBean que contiene los datos de los componentes de una página JSF interfaz de usuario e implementa métodos que permiten su comportamiento. Estos métodos suelen incluir lógica para realizar el control de eventos, validación y control de navegación. Un bean de respaldo por lo general llama a los métodos de un objeto de modelo para realizar la lógica empresarial. JSF permite declarar todos los granos de soporte usado por la página en el archivo de configuración de Faces (cara-config.xml) para que automáticamente se crea una instancia del contenedor Web en tiempo de inicio de la aplicación (estos frijoles se llaman beans gestionados). El archivo de configuración también especifica las reglas de navegación de páginas que funcionan en conjunción con la lógica de control de la navegación en los granos de respaldo.

Event Listener  
Un detector de eventos es una clase definida por el usuario diseñado para manejar un tipo específico de evento de componente generado. JSF es compatible con tres tipos de eventos: Se ha cambiado el valor eventos (por ejemplo, el usuario cambia el valor de un componente), eventos de acción (por ejemplo, el usuario ha hecho clic en un botón) y del modelo de datos eventos (usuario selecciona una fila en un conjunto de datos).

Renderer Kit  
Un Kit Renderer es un grupo de extracción de grasas para un tipo de cliente en particular. A Renderer es una clase que produce la salida adecuada para mostrar un componente de interfaz de usuario en un dispositivo cliente específico. JSF proporciona un kit estándar de HTML que genera render HTML para la visualización de los componentes y le permite crear su propio kit de render para otros tipos de clientes.

Convertidor  
Un convertidor se utiliza para trasladar un objeto de cadena para la exhibición y de la cadena en un objeto para su procesamiento. También se utiliza para aplicar opciones de formato y la localización.

Beneficios  
JSF es un marco basado en los estándares desarrollados a través del Java Community Process (JCP) y pasarán a formar parte de las futuras versiones de Java 2 Enterprise Edition (J2EE).

Como tal, establece la norma oficial para la construcción Java del lado del servidor interfaces de usuario. Sus puntos fuertes incluyen un modelo de programación simplificado que permite a los desarrolladores construir aplicaciones web con un esfuerzo mínimo y una arquitectura flexible que separa claramente la lógica de aplicación a partir de la presentación (en virtud de su naturaleza MVC).

Los siguientes beneficios se derivan de estas ventajas:

* **Desarrollo rápido de aplicaciones**  
  JSF permite el desarrollo rápido de aplicaciones web a través de su modelo de programación simplificado. Los desarrolladores pueden fácilmente reutilizables ensamblar componentes de la interfaz de usuario en una página, conectarlos a fuentes de datos de aplicaciones y alambre de eventos generados por el cliente a los controladores de eventos del lado del servidor. El marco se preocupa de tomar el estado del componente de sincronización automática y proporciona soporte completo para las tareas comunes de programación web como validar la entrada del usuario, la ejecución de la lógica de negocio y el control de navegación de la página. Además, los vendedores cada vez más el apoyo de herramientas para la creación de interfaces de usuario Web utilizando JSF de forma visual para mejorar aún más la productividad.
* **Mantenibilidad**  
  Separación limpia JSF de presentación y el comportamiento en los resultados de la interfaz de usuario de nivel de componentes de una separación de las preocupaciones y un mantenimiento mejorado a un nivel de grano fino:  
  o Un diseñador de páginas puede enfocar con una etiqueta para mostrar un componente mientras que un desarrollador de Java implementa su comportamiento. Cuando haya terminado, rápidamente se puede vincular las piezas utilizando el modelo de programación sencillo JSF o cambios en la presentación de un componente no afectan su comportamiento. Puede, por ejemplo, seleccionar una nueva etiqueta para mostrar un componente sin necesidad de cambiar el código "detrás de él".
* **Flexibilidad**  
  JSF no le limita a una tecnología determinada presentación o lenguaje de marcas:  
  modelo de representación flexibles o JSF permite que los componentes que el dispositivo cliente independiente. Los componentes se pueden mostrar de manera distinta a HTML utilizando renderizadores que generan un lenguaje de marcas diferentes. La lógica mismo componente puede, por lo tanto, ser reutilizado para múltiples tipos de cliente utilizando distintos procesadores.  
  o JSF proporciona una biblioteca de etiquetas JSP personalizadas para representar componentes en una página JSP, pero también permite el uso de una tecnología de otra presentación que JSP. Esto es posible porque la tecnología JSF está directamente en capas en la parte superior de la API Servlet.
* **Extensibilidad**  
  Interfaz de usuario JSF modelo de componentes se puede ampliar para crear componentes personalizados. Esto le permite crear sofisticados componentes, como los árboles y los menús y construir más ricas y más fáciles de usar interfaces de usuario para las aplicaciones Web.

Limitaciones

JSF limitaciones incluyen:

* Es una nueva tecnología que seguirá mejorando con el tiempo.
* JSF no tiene las características de administración de diseño para promover aspecto coherente y permitir que una página se divide en secciones individualmente elaboradas.
* JSF desarrollo es difícil cuando se hace a mano, sin ayuda de herramientas.
* Sus beneficios RAD se realiza efectivamente cuando se utiliza un IDE con herramientas de diseño visual.

# Mysql

Características:

* **Licencia**: Bajo la Licencia Pública General, MySQL es un sistema de código abierto. Sin gastar cualquier cantidad, un desarrollador puede comenzar a trabajar con este sistema. Sin embargo, una persona tiene que comprar una licencia comercial si quiere vender el software. Pero el caso es un poco diferente para MS SQL. Un desarrollador tiene que obtener una licencia. Con fines de desarrollo sencillo, un servicio gratuito de SQL Server es dado. Pero si se quiere usar para fines comerciales, entonces tiene que comprar por lo menos la edición estándar del servidor SQL.
* **Naturaleza**: El MS SQL es un código cerrado en comparación con MySQL. El último está disponible como una fuente abierta y es por lo tanto fácilmente accesibles. Para el MS SQL, uno está limitado en el motor de Sybase-derivada. Pero MySQL ofrece numerosas variantes como Pila, Berkeley DB, MyISAM, InnoDB, etc
* **Funcionalidad**: Si los dos sistemas se comparan sobre la base de su rendimiento, entonces se verá que MySQL seguramente tiene más ventajas. Se necesita un espacio de almacenamiento muy inferior en el disco y la base de datos MyISAM da un rendimiento notable. Además de Windows, este sistema también se desempeña bien en los sistemas UNIX y LINUX. Para MS SQL, su rendimiento es relativamente complejo. Es difícil trabajar con él sin conocimiento muy racional y eficiente del hardware. Trabajar con MS SQL ocupa más espacio en disco en comparación con MySQL.
* **Inconvenientes técnicos**: En comparación con MS SQL, la mayor parte de las claves externas no son compatibles con MySQL. Algunas de las características relacionales que se encuentran en MS SQL no están presentes en MySQL.
* **Recuperación**: Si los dos sistemas se consideran sobre la base del proceso de recuperación, luego de SQL tiene más beneficios que MySQL. Puesto que tiene una configuración con MyISAM, que a veces se convierte en un inconveniente para el sistema. Si hay una alteración o interrupción se lleva a cabo mientras se trabaja con MySQL, entonces puede dar lugar a la pérdida de datos. Incluso si el sistema se enfrenta a cualquier problema mientras trabaja con MS SQL los datos no se pierden.
* **Seguridad**: Tanto MySQL y MS SQL adoptar amplias medidas de seguridad desde el principio. Si un puerto se vuelve muy vulnerable a utilizar entonces el usuario puede cambiarlo. El usuario es necesario para actualizar el software de vez en cuando para protegerlos de los usuarios no deseados e intrusos.