

UNIVERSIDAD MARIANO GALVEZ DE GUATEMALA

FACULTAD DE INGENIERIA EN SISTEMAS

MODALIDAD VIRTUAL

“PROGRAMACION 2”



CARLOS ALFREDO SANDOVAL SAGUI

7690 – 22 – 2568

JavaServer Faces

Es una infraestructura de interfaz de usuario o una interfaz de programación de aplicaciones (API) que facilita el desarrollo de aplicaciones web Java™.

WebSphere® Application Server da soporte a JavaServer Faces 2.0 en un nivel de tiempo de ejecución. El tiempo de ejecución de JSF también:

- Facilita la construcción de una interfaz de usuario a partir de un conjunto de componentes de interfaz de usuario reutilizables
- Simplifica la migración de datos de aplicación hacia la interfaz de usuario o desde ésta.
- Ayuda a gestionar el estado de la interfaz de usuario durante las peticiones de servidor
- Proporciona un modelo sencillo para enviar sucesos generados por el cliente al código de aplicación del lado del servidor.
- Permite crear y reutilizar con facilidad los componentes de una interfaz de usuario personalizada

La implementación JSF de Apache MyFaces proporciona la base del código utilizado para el soporte JSF en WebSphere Application Server. La versión del tiempo de ejecución de JSF proporcionado por el servidor de aplicaciones reside en la ubicación de la biblioteca de tiempo de ejecución normal y está disponible para todas las aplicaciones Web que utilizan la API de JSF. La carga del servlet JSF funciona de la misma forma que si el tiempo de ejecución se empaquetara con la aplicación Web. La versión empaquetada incluye mejoras para una mejor integración con la exploración de anotaciones incorporada y otros componentes de tiempo de ejecución de WebSphere Application Server.

Normalmente, las aplicaciones web que utilizan esta API/infraestructura incorporan la API JSF y los archivos JAR (Java Archive) de implementación en su archivo WAR (Web Application Archive). Esta práctica no es necesaria cuando estas aplicaciones web se despliegan y ejecutan en WebSphere Application Server. Sólo es necesaria la eliminación de estos JAR junto con los archivos JAR JSTL del archivo WAR. Sin embargo, puesto que JavaServer Faces 2.0 forma parte de la plataforma Java Platform, Enterprise Edition (Java EE), no es necesario que una aplicación web empaquete una implementación JavaServer Faces cuando se ejecuta en un contenedor web compatible con la tecnología Java EE. Si una implementación de JavaServer Faces se empaqueta con una aplicación web.

El tiempo de ejecución de JSF para WebSphere Application Server no da soporte al uso de un único cargador de clases para toda la aplicación. Este soporte no está disponible cuando la aplicación contiene varios módulos web y uno de esos módulos es un módulo JSF. No se soporta un cargador de clases individual para la aplicación entera porque la inicialización FacesConfig requiere una clase de cargador individual para cada módulo JSF para realizar la inicialización. Por consiguiente, debe utilizar varios cargadores de clases cuando la aplicación contiene varios módulos web y al menos un módulo JSF.

Para utilizar diferentes implementaciones de JSF, el motor JSF de WebSphere Application Server determina si se utiliza SUN RI o Apache MyFaces desde el tiempo de ejecución del servidor de aplicaciones. Una vez que el motor JSF determina la implementación que debe utilizarse, la clase de escucha adecuada se registrará con el contenedor web. No es necesario añadir `com.sun.faces.ConfigureListener` u `org.apache.myfaces.StartupConfigureListener` al archivo `web.xml`.

La tecnología JavaServer Faces es un framework de interfaz de componentes de usuarios del lado del servidor para las aplicaciones web basadas en la tecnología Java. Los principales componentes de la tecnología JSF son:

- Una API para: Representar componentes de Interfaz de Usuario (UI) y gestionar su estado.
 - Manejar eventos, validar en el servidor y conversión de datos.
 - Definir la navegación de páginas.
 - Soporte de internacionalización y accesibilidad.
- Dos librerías de etiquetas JSP personalizadas para expresar componentes en una página JSP y enlazar los componentes a objetos del servidor.

El modelo de programación bien definido y las librerías de etiquetas facilitan la construcción y mantenimiento de las aplicaciones web con Interfaces de Usuario (UI) de servidor. Con un mínimo esfuerzo se podría:

- Poner componentes en una página mediante etiquetas de componentes.
- Enlazar eventos generados por componentes con código de la aplicación en el servidor.
- Relacionar componentes UI en una página con datos del servidor.

- Construir una UI con componentes reutilizables y extensibles.
- Salvar y restaurar el estado de la UI más allá de la vida de las peticiones.
- Una de las grandes ventajas de la tecnología JavaServer Faces es que ofrece una clara separación entre el comportamiento y la presentación. Las aplicaciones Web construidas con tecnología JSP conseguían parcialmente esta separación. Sin embargo, una aplicación JSP no puede mapear peticiones HTTP al manejo de eventos específicos de los componentes o manejar elementos UI como objetos con estado en el servidor.
- La tecnología JavaServer Faces permite construir aplicaciones Web que implementan una separación entre el comportamiento y la presentación tradicionalmente ofrecidas por arquitectura UI del lado del cliente. JSF se hace fácil de usar al aislar al desarrollador del API de Servlet.
- La separación de la lógica de la presentación también le permite a cada miembro del equipo de desarrollo de una aplicación Web enfocarse en su parte del proceso de desarrollo, y proporciona un sencillo modelo de programación para enlazar todas las piezas.

Otro objetivo importante de la tecnología JavaServer Faces es mejorar los conceptos familiares Disadvantage de componente-UI y capa-Web sin limitar a una tecnología de script particular o un lenguaje de marcas. Aunque la tecnología JavaServer Faces incluye una librería de etiquetas JSP personalizadas para representar componentes en una página JSP, los APIs de la tecnología JavaServer Faces se han creado directamente sobre el API JavaServlet.

- JavaServer Faces ofrece una gran cantidad de componentes opensource para las funcionalidades que se necesiten. Los componentes Tomahawk de MyFaces y ADFFaces de [Oracle](#) son un ejemplo. Además, también existe una gran cantidad de herramientas para el desarrollo IDE en JSF al ser el estándar de JAVA.

PAGINA WEB

<https://www.uv.mx/personal/lizhernandez/files/2012/05/jsf.pdf>

LIBRO

[Copyright \(c\) \(2001, February 10\) Yadira Jarvio Hernández](#)