



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Tarea 4 - Resumen: Desarrollo de Aplicaciones Web de Alta Calidad en la Plataforma Java EE

Unidad de aprendizaje: Web Application Development

Grupo: 3CM6

Alumnos(a):

Ramos Diaz Enrique

Profesor(a):

Montes Casiano Hermes Francisco

23 de febrero 2019

Índice

1	Desarrollo de Aplicaciones Web de Alta Calidad en la Plataforma Java EE	1
1.1	Introducción	1
1.2	La Arquitectura de Aplicaciones Web en la Plataforma Java EE utilizando el Patrón de Diseño MVC	1
1.2.1	El Patrón de Diseño MVC	1
1.2.2	Arquitectura de Aplicaciones Web en la Plataforma Java EE	2
1.3	Proceso de Desarrollo de Aplicaciones Web en la Plataforma Java EE	3
1.3.1	Proceso de desarrollo de aplicaciones web	3
1.3.2	Soporte en el Proceso de Desarrollo	3
1.4	Características Importantes para Aplicaciones Web en la Plataforma Java EE	3
1.5	Conclusiones	4
2	Bibliografía	4

1. Desarrollo de Aplicaciones Web de Alta Calidad en la Plataforma Java EE

1.1. Introducción

La gran competencia en el negocio de aplicaciones web obligan a los desarrolladores de aplicaciones web a ser más conscientes de la calidad. Será de gran valor si una aplicación web pudiese sobrevivir a los exigentes y cambiantes requisitos de los clientes y los cambiantes requisitos empresariales.

Java EE es una plataforma abierta, estandarizada, e independiente del hardware y/o sistema operativo en la que se pueden desarrollar y ejecutar aplicaciones empresariales distribuidas.

Como las aplicaciones dirigidas a la plataforma Java EE son neutrales con respecto a los proveedores, la organización no se enfrenta al problema del bloqueo de proveedores. Las aplicaciones basadas en la plataforma Java EE utilizan el patrón de diseño Modelo - Vista - Controlador para tres componentes arquitectónicos: lógica de presentación, lógica de control y lógica de negocios.

1.2. La Arquitectura de Aplicaciones Web en la Plataforma Java EE utilizando el Patrón de Diseño MVC

1.2.1. El Patrón de Diseño MVC

El patrón de diseño MVC consiste en tres tipos de objetos: Modelo, Vista y Controlador, que manejan tres tipos de responsabilidades básicas: entidad (datos), barrera (presentación), y control (comportamiento) respectivamente. El modelo encapsula los datos de la aplicación y la lógica de negocios; la vista maneja la representación de datos de la aplicación y la interfaz visual al usuario, y el controlador se encarga de la interacción del usuario con la aplicación.

El patrón de diseño MVC separa las vistas y los modelos estableciendo un protocolo de convenio y notificación entre ellos. Un objeto de vista debe asegurar que su apariencia refleje el estado del modelo. El objeto de modelo es independiente de ambos objetos de vista y de control, así que es posible tener múltiples vistas del mismo modelo. Todas las vistas asociadas pueden convenir con el modelo y el modelo les notifica sobre su cambio de estado.

1.2.2. Arquitectura de Aplicaciones Web en la Plataforma Java EE

En la arquitectura de aplicaciones web basadas en la plataforma Java EE, el servlet es el componente utilizado como controladores; el componente JavaBean como modelo; y los Java Server Pages (JSP) como plantilla para la vista. El Enterprise Java Bean (EJB) puede ser utilizado como modelo, el cuál se puede aplicar en el entorno distribuido, a diferencia de JavaBean. La ejecución de una JSP page genera una vista basada en contenido HTML.

Java EE ofrece soporte para seguridad, autenticaciones, autorizaciones y transacciones, así como manejo de conexión a bases de datos, que es configurable de forma externa en el deployment descriptor. En consecuencia, la responsabilidad del componente modelo es solo manejar únicamente los datos y la lógica de negocios. Los servicios de autenticación y autorización son proporcionados por Java EE al servlet y también son configurables de forma externa en el deployment descriptor.

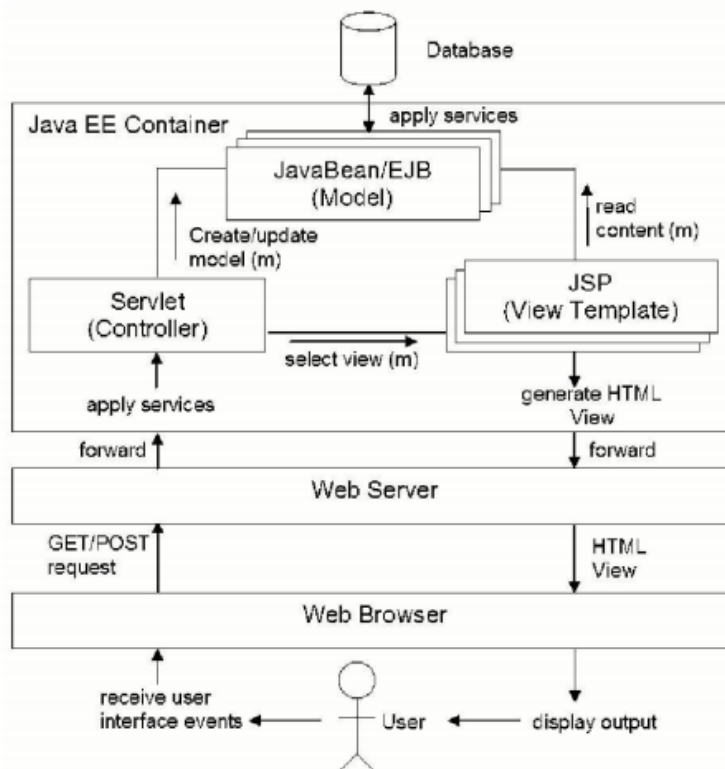


Figure 2. Use of the MVC Design Pattern in the Java EE Platform based web-application architecture

1.3. Proceso de Desarrollo de Aplicaciones Web en la Plataforma Java EE

1.3.1. Proceso de desarrollo de aplicaciones web

El desarrollo de aplicaciones web en Java EE consiste en: diseño, codificación, creación del deployment descriptor, empaquetado, ensamblaje y despliegue.

- **Proveedores de componentes de aplicación:** Crear componentes EJB y web.
- **Ensamblador de aplicaciones:** Ensambla diferentes componentes en una aplicación desplegable.
- **Desplegador:** Despliega la aplicación ensamblada en un entorno de operación.
- **Administrador del sistema:** Da mantenimiento y monitorea la aplicación. Balanceo de cargas.

1.3.2. Soporte en el Proceso de Desarrollo

1. **Soporte en Codificación:** El entorno de desarrollo debe proporcionar facilidades para evitar errores tipográficos, requerir el mínimo esfuerzo al añadir nuevas funcionalidades y proporcionar soporte para tecnologías y herramientas basadas en estándares.
2. **Soporte en Pruebas:** Buena depuración y soporte de pruebas unitarias son requisitos básicos en el proceso de pruebas. JUnit proporciona un framework de prueba de regresión para pruebas unitarias.
3. **Soporte en Integración y Despliegue:** El despliegue debe ser independiente de distintos servidores de aplicaciones de alojamiento. Las aplicaciones web que utilicen la tecnología Java podrán ejecutarse en cualquier compilación de Java EE sin modificar el código.
4. **Soporte en el Mantenimiento:** Los tres componentes de Java EE: servlets, JSPs y componentes JavaBean/EJB puede ser implementados y mantenidos independientemente por sus respectivos desarrolladores (roles).

1.4. Características Importantes para Aplicaciones Web en la Plataforma Java EE

Atributos como la escalabilidad, portabilidad, reusabilidad, seguridad, alto desempeño y flexibilidad están inherentes en las clases y componentes de Java, como lo son el servlet y los componentes JavaBean.

- **Seguridad:** La comunicación segura es proporcionada a través del soporte SSL. El contenedor SSL debe estar configurado en el contenedor Java EE y el certificado CA del servidor instalado en el mismo.

- **Procesamiento de transacciones:** Java EE soporta transacciones gestionadas por el contenedor y gestionadas por bean para sesiones bean y mensajes manejados por bean. El contenedor comienza una transacción antes de que un método de negocios inicie. Si existe una excepción en éste método, el contenedor anula automáticamente la transacción.
- **Soporte para manejo de sesiones:** La mayoría de aplicaciones web en Internet manejan las sesiones utilizando cookies. El mecanismo de codificación URL en Java EE automáticamente determina si el navegador de un cliente soporta o no cookies, y decide cómo debe almacenarse la información sobre la identificación de la sesión en la máquina cliente.
- **Páginas de error personalizadas:** Una buena aplicación web no debe mostrar mensajes de error generados por el servidor web o de aplicaciones directamente al usuario. En Java EE, las páginas de error pueden especificarse manualmente para los mensajes de error definidos por el usuario.
- **Internacionalización:** Si la información es proporcionada a usuarios en un lenguaje que entiendan y usen, será mucho más fácil para ellos entender y usar la aplicación. El soporte Unicode para cadenas de caracteres esta inherente en el lenguaje Java.

1.5. Conclusiones

La plataforma Java EE posee los siguientes atributos de calidad: usabilidad, funcionalidad, confiabilidad, eficiencia y mantenibilidad. Java EE simplifica los procesos de diseño, desarrollo, despliegue, integración y pruebas de las aplicaciones web sin comprometer la alta calidad. Además, enriquece al desarrollo de aplicaciones web con escalabilidad, portabilidad, interoperabilidad, reusabilidad, flexibilidad y seguridad.

Muchos frameworks para el desarrollo de aplicaciones web están disponibles para Java EE, pero ninguno se encuentra proporcionando soporte para aplicaciones web distribuidas de alta calidad.

2. Bibliografía

[1] H. B. Prajapati, V. K. Dabhi, *High Quality Web-Application Development on Java EE Platform*. 2009 IEE International Advance Computing Conference (IACC 2009). India. 2009.