	INGENIERÍA EN INFORMÁTICA – PLAN 2003 DISEÑO AVANZADO SOFTWARE – 10º CUATRIMESTRE	
	APUNTE SOBRE ARQUITECTURA MODELO 1	VERSIÓN: 1.3 VIGENCIA: 06-08-2009

DESARROLLAR APLICACIONES WEB UTILIZANDO LA ARQUITECTURA MODELO 1

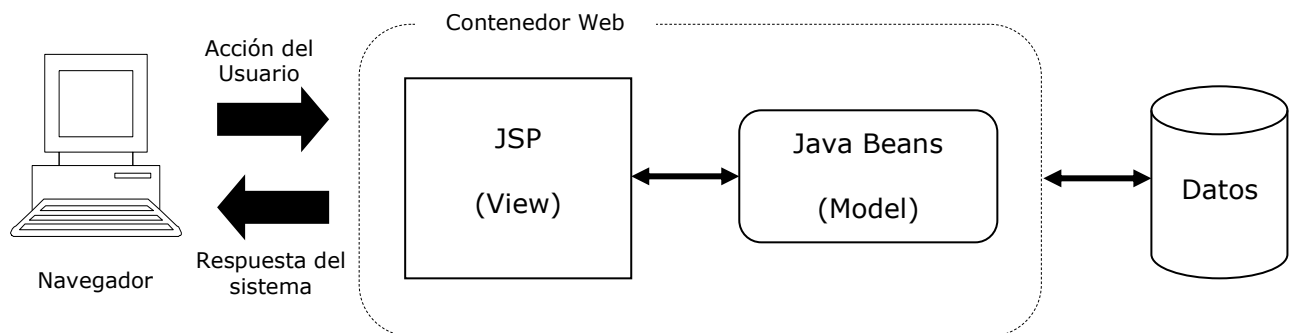
Observando cualquier aplicación Web debería reconocerse la presencia de algún esquema de navegación Web que dicta cómo fluyen las páginas junto con los datos que deben procesarse con ellas. Algunas aplicaciones emplean prácticas de Arquitectura Modelo 1 que codifican por hardware esos flujos de las páginas que componen las aplicaciones Web. Otros utilizan prácticas de Modelo 2 que incrustan los atributos de flujo en un archivo externo para que el mantenimiento y las rutas de navegación puedan manejarse fuera del propio código. Inevitablemente, durante la fase de diseño de un programa, se tendrá que hacer suposiciones sobre cuál de los métodos será puesto en práctica antes de desplegar el software. En la toma de decisión se podrían utilizar experiencias externas, como la experimentación y las lecciones aprendidas en trabajos anteriores, pero lo más probable es que los plazos de entrega, la madurez del personal y los plazos afecten a la decisión.

¿Qué es Modelo 1? ¿Por qué usarlo?

La Arquitectura Modelo 1 es un enfoque "centrado en páginas" en el que distintos componentes Web manejan, individualmente, los flujos de páginas. Esto significa que el procesamiento de solicitudes y respuestas se codifica por hardware en las páginas para acomodar la navegación del usuario en una aplicación Web.

Por supuesto, esto supone problemas de mantenimiento cuando se necesitan modificaciones lógicas para acomodar cambios de requisitos y necesidades del usuario final. Esos cambios obligan a los desarrolladores a hacer malabarismos por el código para garantizar que todos los flujos lógicos se manejan adecuadamente cuando los usuarios navegan por una aplicación Web. Junto con las responsabilidades propias del mantenimiento del flujo de navegación en los despliegues de Modelo 1, es necesario gestionar puntos relativos a la seguridad y al estado de la aplicación.

Los problemas a los que se enfrenta la Arquitectura Modelo 1 son difíciles decisiones de diseño que hacen frente al comienzo de un proyecto, pero las limitaciones de la experiencia en desarrollo de un equipo, el alcance de una aplicación y el tiempo de entrega pueden persuadir de adoptar esta filosofía de desarrollo. Adoptar la filosofía de Modelo 1 no es, necesariamente, una mala decisión, dependiendo de la estimación de lo que entregará un equipo, y como lo hará en un plazo de entrega apretado.

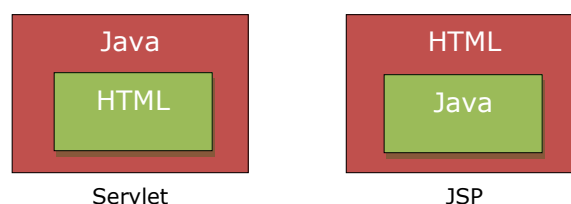


En resumen, los problemas principales de esta arquitectura son:

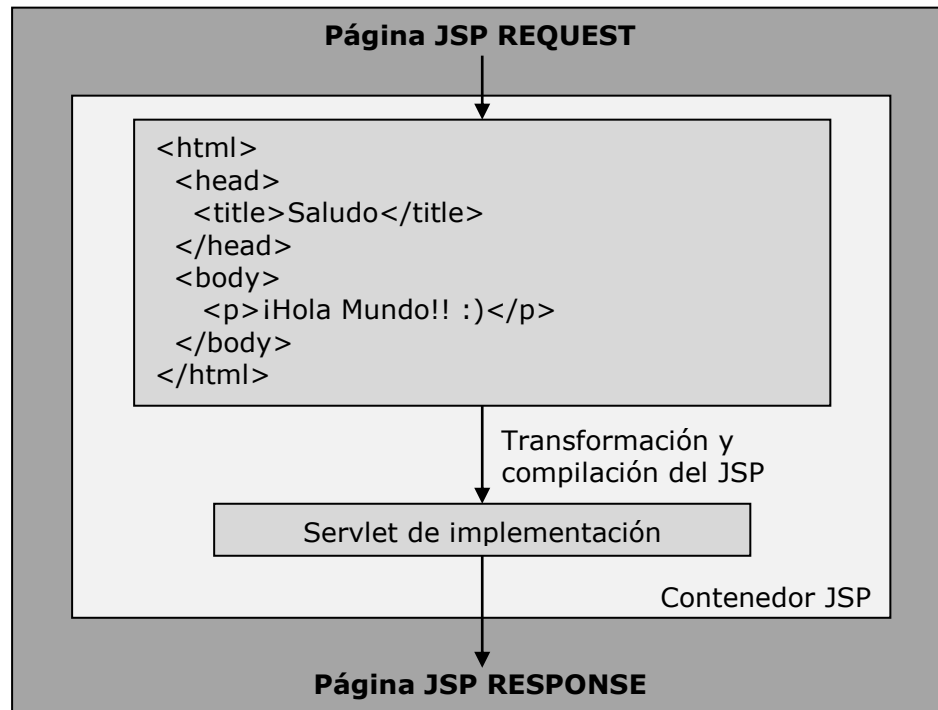
- (1) *Problemas de mantenimiento:* En general contiene varias páginas JSP que contienen numerosos scriptlets.
- (2) *Problemas de reusabilidad:* En general la lógica está embebida en las páginas JSP y torna complicado reusar la lógica de negocio (en general se corta y se pega).
- (3) *Problemas de seguridad:* Debido a que las páginas JSP son responsable por el manejo de todo el procesamiento, muy probablemente la página maneje funciones de logueo y lógica en forma embebida.

JSP (Java Server Page)

La tecnología JSP combina HTML, XML y Java Servlet, facilitando la creación de contenido dinámico. Si programáramos un Servlet y una página JSP que generaran el mismo código HTML, veríamos que en un Servlet el HTML es generado desde el código Java, mientras que en un JSP el código Java es embebido en el código HTML.



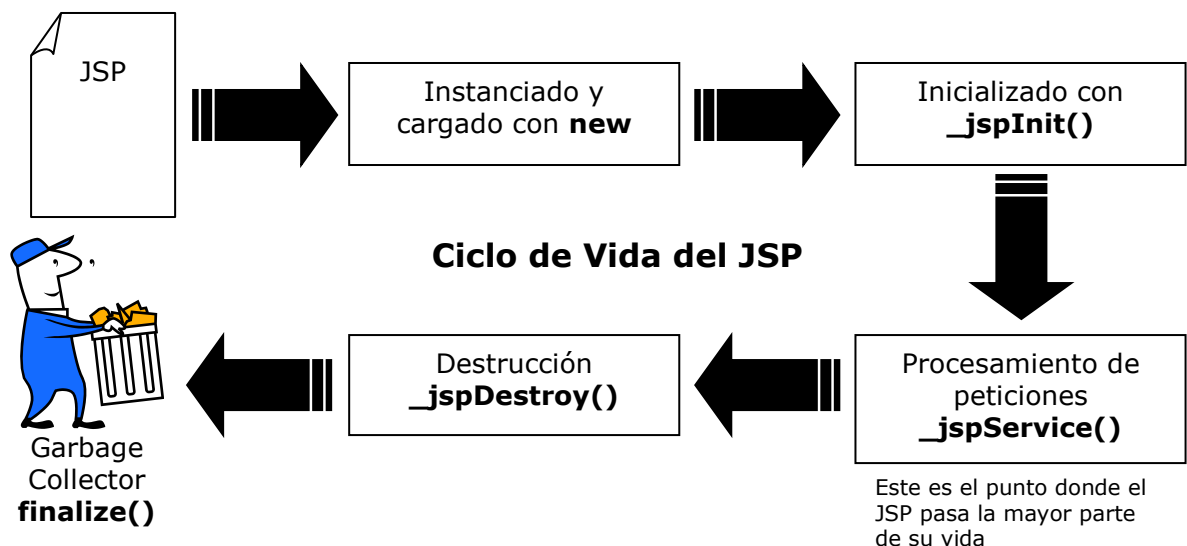
JSP es una extensión de Servlet. La página JSP es implementada a través de la clase `javax.servlet.Servlet` de acuerdo a la especificación Servlet 2.5



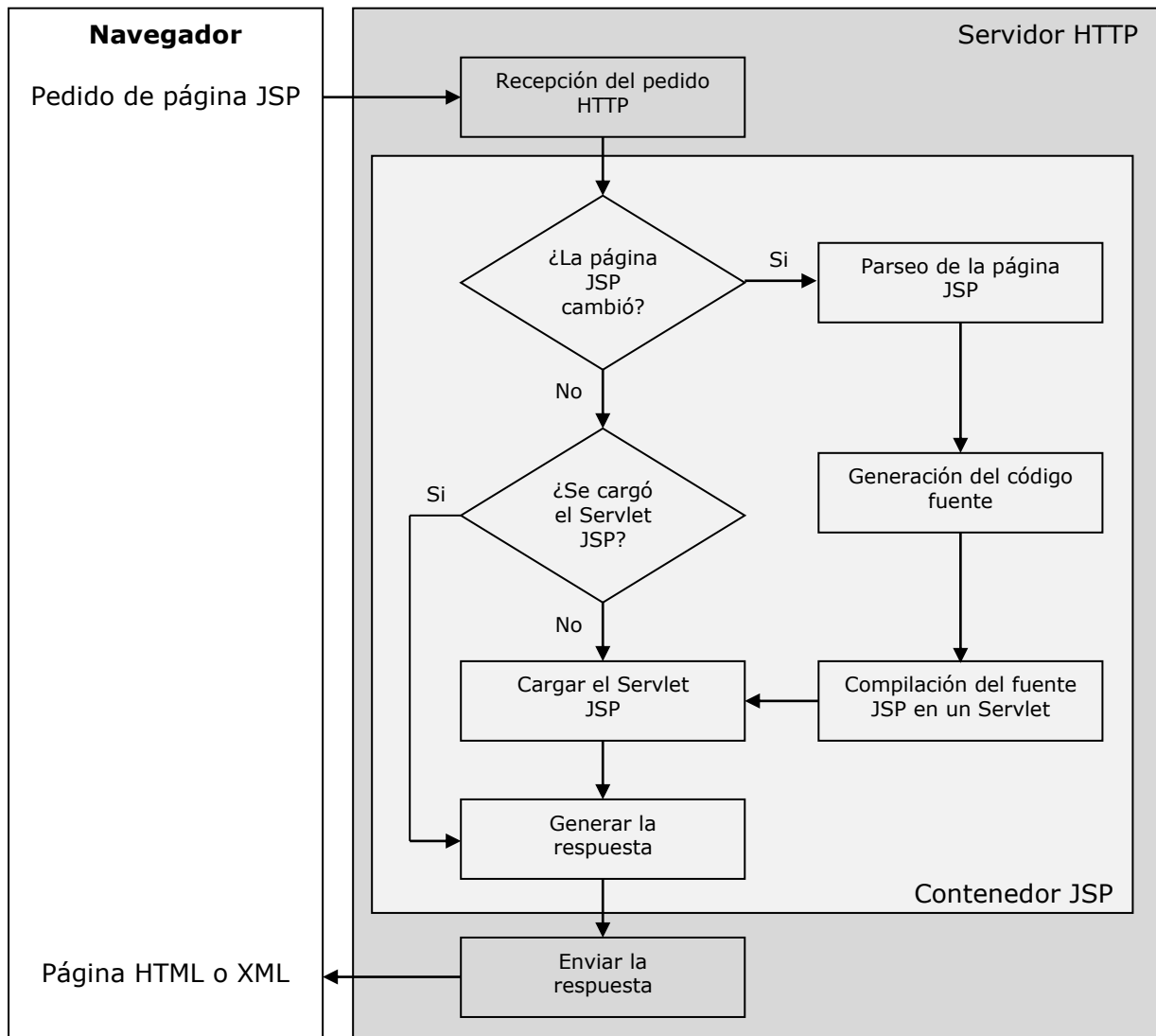
Ciclo de Vida del JSP

El contenedor JSP es el encargado de regular y controlar el ciclo de vida de un JSP. El ciclo de vida consiste de cuatro fases:

- *Traslación:* Si la sintaxis es correcta se traslada al Servlet de implementación.
- *Inicialización:* El contenedor JSP carga el Servlet de implementación y crea una instancia del Servlet para procesar el pedido. Se invoca al método `_jspInit()`
- *Ejecución:* Después de que el contenedor cargó e inicializó el Servlet de implementación, el pedido es procesado. Para resolver el pedido se llama al método `_jspService()`
- *Finalización:* El contenedor invoca el método `_jspDestroy()`. Después de que este método es invocado el Servlet de implementación no puede responder a ningún otro pedido.



¿Cómo trabaja una página JSP?



JSP (Java Server Page) y JSTL (JSP Standard Tag Library), son importantes componentes de aplicación Web para construir contenido dinámico en plataformas J2EE. Los scripts JSP pueden construir fácilmente contenido HTML y acceder a propiedades JavaBean a través de bibliotecas EL (Expression Language). Los componentes JSTL encapsulan funciones que permiten a los desarrolladores repetir a través de datos, realizar operaciones de transformación XSLT y acceder a datos de objeto y base de datos. Las dos tecnologías pueden combinarse para dar formas a componentes a nivel de presentación que mostrar y con los que interactuar con modelos de datos de back-end.