|  |  |
| --- | --- |
|  | **DAW II** |
|  | Desarrollo de aplicaciones web en entorno servidor  Aitor Sagastagoitia Sáenz |

|  |
| --- |
| **[Java Servlets]** |
|  |



**Java Servlets:**

Servlet, sencillamente, es una clase Java, que se usa para ampliar las capacidades de los servidores. Estos, los servlets, pueden responder a cualquier solicitud, normalmente se usan para extender las aplicaciones del servidor web, de modo que pueden ser vistos como applets de Java que se ejecutan en servidores, en vez de navegadores. Estos son la contrapartida Java de otras tecnologías de contenido dinámico, como PHP. Estudiaremos dos de estos Servlets en profundidad:

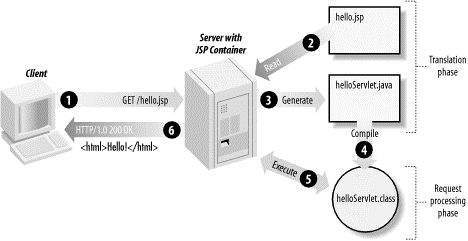
**Java Server Pages:**

JSP es una tecnología que contribuye a la creación de páginas web dinámicas, JSP es parecido a PHP pero usa el lenguaje de programación Java. Para que funcione JSP se necesita un servidor web compatible con servlet.

El rendimiento de JSP es el mismo que cualquier servlet equivalente, porque la compilación del código es idéntica a la de cualquier aplicación Java. La principal ventaja de JSP es que el lenguaje Java es un lenguaje de propósito general, es decir no está orientado al mundo web. Esto permite estructurar las aplicaciones web, de modo que la parte encargada de generar el HTML, este en el archivo JSP. Otra ventaja es la portabilidad de Java, ya que es posible ejecutar las aplicaciones en múltiples plataformas sin realizar ningún cambio. Los servlets y JSP, como hemos dicho antes son dos métodos de creación de páginas dinámicas similares a PHP, ASP… sin embargo tienen algunas diferencias.

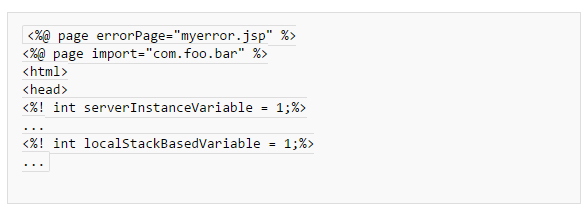
En primer lugar, JSP y servlets se ejecutan en una máquina virtual Java, lo que permite, en principio, que se pueda usar en cualquier ordenador. Estos se ejecutan en su propio contexto, pero no inicia la ejecución cada vez 1que recibe una petición, sino que va de una petición a otro, de modo que no se pierde tiempo de invocación.

Arquitectura:



JSP en el fondo, es una forma alternativa de crear servlets ya que el código JSP se traduce a código servlet la primera vez, y a partir de esta es el código servlet el que produce el código HTML que forma la página web de respuesta.

Ejemplo de una pagina JSP:



Ejemplo de una compilación JSP:



**Enterprise Java Beans:**

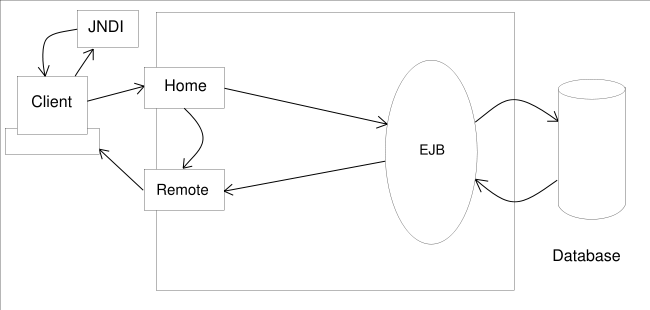
Las Enterprise JavaBeans (EJB) son una de las interfaces de programación de aplicaciones (API), que forman parte del estándar de construcción de aplicaciones empresariales JEE de Oracle.

Los EJB proporcionan un modelo de componentes distribuido del lado del servidor. El objetivo de EJB es dotar al programador de un modelo que le permita abstraer los problemas generales de una aplicación empresarial, para centrarse en el desarrollo de la lógica del negocio en si. El hecho de estar basado en componentes permite que estos sean flexibles y sobre todo reutilizables.

Existen tres tipos de EJB:

* EJB de Entidad: Su objetivo es encapsular los objetos del servidor que almacenan los datos.
* EJB de Sesión: Este tipo, gestiona el flujo de la información en el servidor.
* EJB dirigidos por mensajes: Estos son los únicos beans asíncronos. Usando JMS (Java Messaging System), se suscriben a un tema o a una cola y se activan al recibir un mensaje dirigido a dicho tema o cola.

Arquitectura:



Ejemplos EJB:

