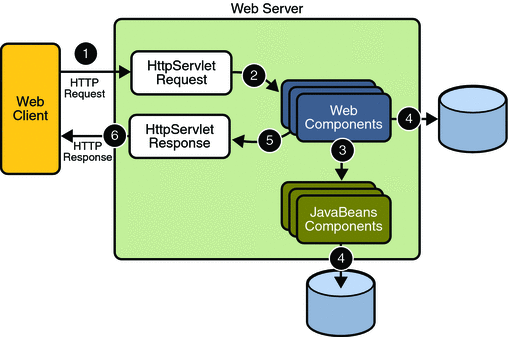
¿Que es Servlet en Java?

Los **servlets** son objetos que corren dentro y fuera del contexto de un contenedor de servlets (ej: [Tomcat](http://es.wikipedia.org/wiki/Tomcat)) y extienden su funcionalidad.

La palabra *servlet* deriva de otra anterior, [*applet*](http://es.wikipedia.org/wiki/Applet), que se refería a pequeños programas que se ejecutan en el contexto de un navegador web. Por contraposición.

El uso más común de los *servlets* es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.



Es una tecnología que nos permite crear aplicaciones web interactivas (dinámicas), es decir, le permite al usuario interactuar con la aplicación (hacer consultas, insertar y eliminar datos)

Un Servlet es un objeto java que pertenece a una clase que extiende de javax.servlet.http.HttpServlet

Son pequeños programas escritos en Java que admiten peticiones a través del protocolo HTTP. Los servlets reciben peticiones desde un navegador web, las procesan y devuelven una respuesta al navegador, normalmente en HTML. Para realizar estas tareas podrán utilizar las clases incluidas en el lenguaje Java. Estos programas son los intermediarios entre el cliente (casi siempre navegador web) y los datos (BBDD)

http://tutorials.jenkov.com/images/java-servlets/overview.png

**¿Qué es un contenedor de Servlets?**

Un contenedor de Servlet es un programa capaz de recibir peticiones de páginas web y redireccionar estas peticiones a un objeto Servlet. Ejemplo: Apache Tomcat

**¿Cómo funciona un contenedor de Servlets?**

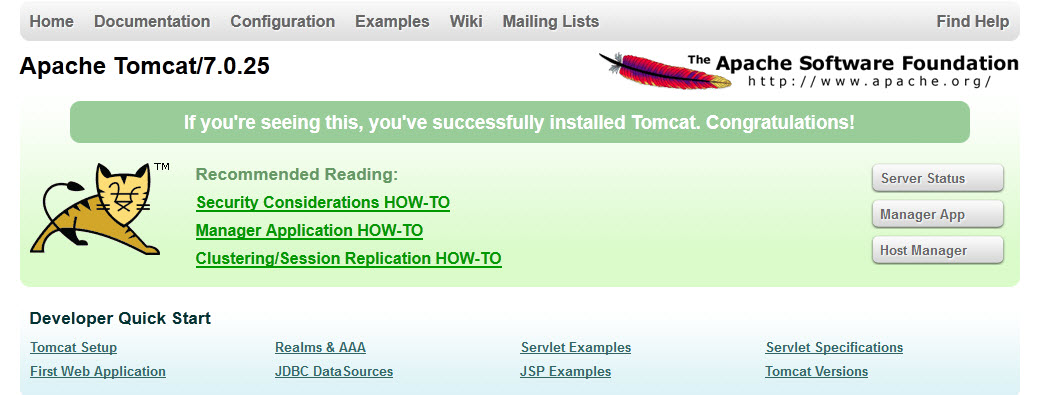
1. El navegador (cliente) pide una página al servidor HTTP que es un contenedor de Servlets.

2. El servlet procesa los argumentos de la petición, es decir, el contenedor de Servlets delega la petición a un Servlet en particular elegido de entre los Servlets que contiene.

3. El Servlet, que es una objeto java, se encarga de generar el texto de la página web que se entrega al contenedor.

4. El contenedor devuelve la página web al navegador (cliente) que la solicitó, normalmente en HTML. Por lo tanto nos encontramos en una arquitectura Cliente-Servidor. Lo normal para esto es utilizar Apache Tomcat como contenedor de servlets. Recordar que apache es un servidor HTTP.

**¿Qué es Apache Tomcat?**



Es un contenedor de servlets, no es un servidor de aplicaciones. La diferencia principal radica en que un contenedor de servlets está pensado únicamente para tecnología web (acceso vía HTTP), mientas que en un servidor de aplicaciones se pueden hacer sistemas más complejos y multicapa. Por ejemplo, un sistema que poseerá soluciones de escritorio, web y móvil deberá alojarse en un servidor de aplicaciones, ya que posee capas distintas que requerirán mayor complejidad en el lado servidor.

Antes de empezar con el desarrollo del artículo, veremos algunas definiciones interesantes que nos vendrán bien para entenderlo.

1. **Desplegar una aplicación (deployment):** significa ponerla en producción.
2. **Servidor de aplicaciones:** un servidor de aplicaciones consiste en un contenedor que abarca la lógica de negocio de un sistema (según el patrón MVC, sería el Modelo), y que provee de respuestas a las peticiones de distintos dispositivos que tienen acceso a ella. Son un claro ejemplo del modelo **cliente-servidor**, cuyo lado cliente ejecuta requerimientos de procesamiento al otro lado, donde el servidor se encarga de procesar y responder.

**Servidores de aplicaciones J2EE:**

* 1. **JBoss:** libre, es el más utilizado.
  2. **WebSphere:** conjunto de aplicaciones desarrolladas por IBM. La más importante es WebSphere Aplication Server. También se incluyen aplicaciones para diseñar modelos de negocio, ejecutar y monitorizar procesos.
  3. **GlassFish:** libre, desarrollado por Sun. Incorpora un componente llamado Grizzly, que aumenta la escalabilidad y velocidad del servidor.
  4. **WebLogic**

**Servidores de aplicaciones .NET:**

* 1. **Internet Information Server (IIS)**
  2. **Base4 Server**
  3. **Zope**