

Los servlets son programas Java que utilizan la interfaz de programación de aplicaciones (API) de servlet Java. Debe empaquetar los servlets en un archivo WAR (archivador de aplicación web) o un módulo web para su despliegue en el servidor de aplicaciones. Los servlets, que se ejecutan en un servidor web habilitado para Java, amplían las funciones de un servidor web, de forma similar a como se ejecutan applets en un navegador, y amplían las funciones de un navegador.

Los Servlets pueden dar soporte al contenido dinámico de páginas web, proporcionar acceso a bases de datos, dar servicio a varios clientes al mismo tiempo y filtrar datos.

En el servidor de aplicaciones, las referencias a los servlets se centran en los servlets HTTP, que dan servicio a clientes basados en web.

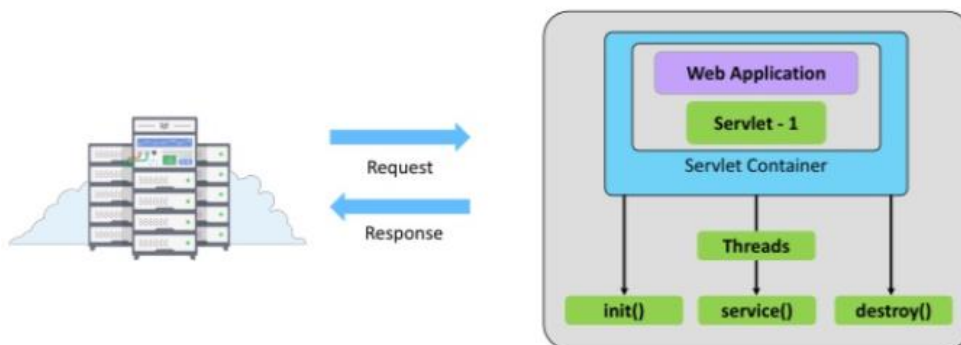
Puede definir servlets como archivos de bienvenida. Los recursos que no son de servlet sólo se sirven cuando el atributo `fileServingEnabled` está establecido en true en el archivo XMI de ampliaciones IBM, `ibm-web-ext.xmi`, que se encuentra en el directorio WEB-INF del módulo web o mediante una herramienta de ensamblaje para establecer la propiedad en el archivo .war de origen. El servicio de archivos de bienvenida está conectado al contenido estático del servicio. Por lo tanto, se establece `fileServingEnabled` en el módulo web.

## Parámetros de contexto

Un contexto de servlet define la vista del servidor de la aplicación web en la que se ejecuta el servlet. El contexto también da soporte a un servlet para acceder a sus recursos disponibles. Mediante el contexto del servlet, un servlet puede registrar sucesos, obtener referencias de URL a recursos, y establecer y almacenar atributos para que las utilicen otros servlets del contexto. Estas propiedades declaran los parámetros para el contexto de una aplicación web. Las propiedades ofrecen información de configuración como, por ejemplo, la dirección de correo electrónico del administrador de web o el nombre de un sistema con datos críticos.

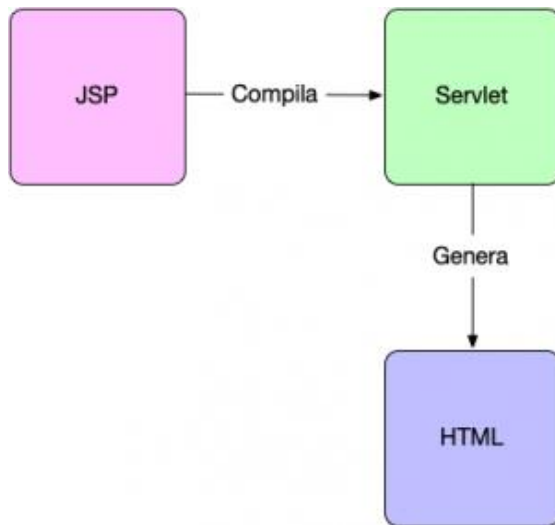
## Correlaciones de servlets

Una correlación de servlets es una correspondencia entre una solicitud de cliente y un servlet. Los contenedores Web utilizan vías de acceso de URL para correlacionar solicitudes de clientes con servlets y siguen las reglas de correlación de vías de acceso de URL tal como se indica en la especificación Java Servlet. El contenedor utiliza el identificador universal de recursos (URI) de la solicitud, menos la vía de acceso de contexto, como vía de acceso para correlacionarse con un servlet. El contenedor elige la vía de acceso de contexto más larga posible que coincida de la lista de aplicaciones web que alberga.



## JSP y Servlet

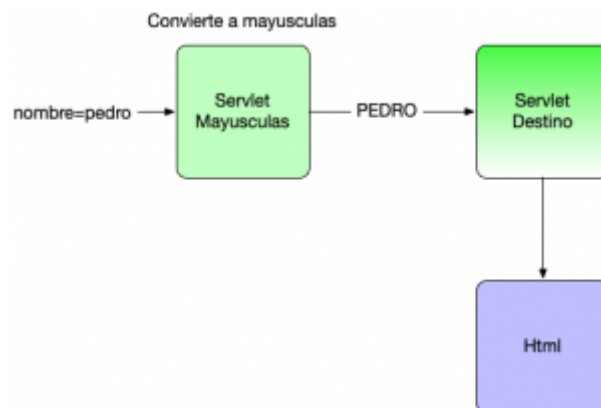
Cuando cualquier persona construye una página JSP esta página realmente no existe a nivel de Java ya que Java lo único que entiende son clases. Por lo tanto, el fichero JSP ha de convertirse por medio de un compilador en un Servlets.



cualquier fichero JSP que hayas creado en algún momento es un Servlet. Esto hace que la tecnología hoy por hoy siga estando muy viva. Aun así, muchas personas piensan que los Servlets son únicamente utilizados en aplicaciones antiguas con JSP. Las cosas no son como parecen ya que los Servlets tienen un diseño muy flexible a nivel de Arquitectura y permiten pasar información entre ellos de una forma transparente con un modelo de delegación.

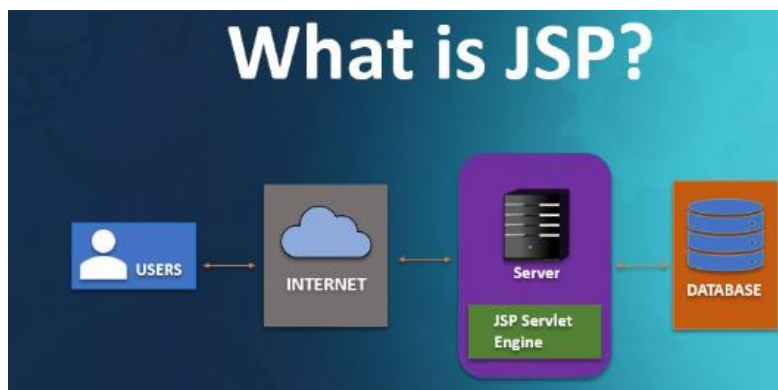
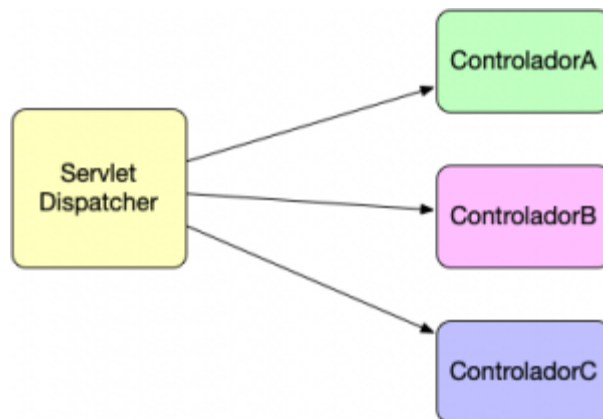
## Servlet y Despachadores

Por ejemplo, nosotros podríamos tener un Servlet que reciba un variable nombre, convierta esta variable a mayúsculas y delegue en otro Servlets que sea el encargado de mostrar la información.



## FrontController y Servlets

Puede parecer poco importante este sistema de delegación, pero es la base de todos los frameworks modernos web de Java ya que por ejemplo Spring MVC usa un Servlet Principal o FrontController para recibir todas las peticiones web y redirigirlas a cada una de los controladores que tiene. Este Servlet se denomina DispatcherServlet a nivel de Spring y puede que en algún momento nos veamos en la obligación de tocar su configuración de alguna forma.



La tecnología JavaServer Pages permite generar contenido Web dinámico como, por ejemplo, archivos HTML, DHTML, XHTML y XML, para incluirlos en una aplicación Web. Los archivos JSP son una forma de implementar contenido de

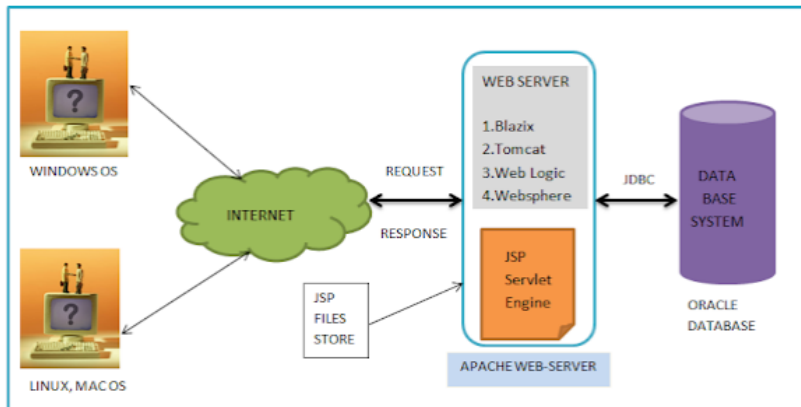
páginas dinámico del lado del servidor. Los archivos JSP permiten a un servidor Web como, por ejemplo, Apache Tomcat, añadir contenido dinámicamente a las páginas HTML antes de enviarlas al navegador que las solicita.

Cuando se despliega un archivo JSP en un servidor Web que proporciona un motor de servlets, se procesa previamente en un servlet que se ejecuta en el servidor. Esto contrasta con JavaScript™ en el lado del cliente (dentro de códigos <SCRIPT>), que se ejecuta en un navegador. Una página JSP resulta ideal para tareas cuya ejecución es más adecuada en el servidor como, por ejemplo, acceder a bases de datos o llamar a beans de Enterprise Java™.

Puede crear y editar un archivo JSP en el editor HTML añadiendo su propio texto e imágenes utilizando HTML, códigos JSP o JavaScript, incluyendo código fuente Java dentro de los códigos de scriptlet. Normalmente, los archivos JSP tienen la extensión de archivo. jsp. Asimismo, la especificación JSP recomienda que los archivos de fragmento JSP tengan la extensión de archivo. jspf. Si no se sigue este convenio, el validador de JSP tratará los fragmentos JSP como archivos JSP autónomos normales y se pueden generar errores de compilación.

La Especificación Sun Microsystems JSP 1.2 ofrece la posibilidad de crear códigos JSP personalizados. Los códigos personalizados simplifican las acciones complejas y ofrecen a los desarrolladores un mayor control del contenido de las páginas. Los códigos personalizados se recopilan en una biblioteca (taglib). Un archivo descriptor de biblioteca de códigos (taglib.tld) es un documento XML que proporciona información sobre la biblioteca de códigos, incluido el nombre abreviado de la taglib, la descripción de la biblioteca y las descripciones de código. Consulte la Especificación Sun Microsystems JSP 1.2 para obtener más información.

Para utilizar bibliotecas de códigos personalizadas de JSP 1.2, puede importar la biblioteca de códigos .tld y los archivos .jar al proyecto para utilizarlos o asociarlos como proyectos de biblioteca Web. También puede hacer referencia a un archivo TDL utilizando un URI.



| Elemento JSP    | Síntaxis                             | Interpretación   | Notas   |
|-----------------|--------------------------------------|--|---|
| Expresión JSP   | <code>&lt;%= expression %&gt;</code> | La Expresión es evaluada y situada en la salida.   | El equivalente XML es <b><code>&lt;jsp:expression&gt; expression &lt;/jsp:expression&gt;</code></b> . Las variables predefinidas son <b>request, response, out, session, application, config, y pageContext</b> . |
| Scriptlet JSP   | <code>&lt;% code %&gt;</code>        | El código se inserta en el método <b>service</b> .   | El equivalente XML es: <b><code>&lt;jsp:scriptlet&gt; code &lt;/jsp:scriptlet&gt;</code></b> .  |
| Declaración JSP | <code>&lt;%! code %&gt;</code>       | El código se inserta en el cuerpo de la clase del servlet, fuera del método <b>service</b> . | El equivalente XML es: <b><code>&lt;jsp:declaration&gt; code &lt;/jsp:declaration&gt;</code></b> .  |

|                                  |  |   |  |
|----------------------------------|--|---|--|
| Acción<br><b>jsp:useBean</b>     | <pre>&lt;jsp:useBean att=val*/&gt;       o &lt;jsp:useBean att=val*&gt;   ... &lt;/jsp:useBean&gt;</pre> | Encuentra o construye un Java Bean.   | Los posibles atributos son: <ul style="list-style-type: none"> <li>• id="name"</li> <li>• scope="page request session application"</li> <li>• class="package.class"</li> <li>• type="package.class"</li> <li>• beanName="package.class"</li> </ul> |
| Acción<br><b>jsp:setProperty</b> | <pre>&lt;jsp:setProperty att=val*/&gt;</pre>   | Selecciona las propiedades del bean, bien directamente o designando el valor que viene desde un parámetro de la petición.                 | Los atributos legales son: <ul style="list-style-type: none"> <li>• name="beanName"</li> <li>• property="propertyName *"</li> <li>• param="parameterName"</li> <li>• value="val"</li> </ul>  |
| Acción<br><b>jsp:getProperty</b> | <pre>&lt;jsp:getProperty name="propertyName" value="val"/&gt;</pre>                                      | Recupera y saca las propiedades del Bean.   |  |
| Acción<br><b>jsp:forward</b>     | <pre>&lt;jsp:forward page="relative URL"/&gt;</pre>  | Reenvía la petición a otra página.  |  |
| Acción<br><b>jsp:plugin</b>      | <pre>&lt;jsp:plugin attribute="value"*&gt;   ... &lt;/jsp:plugin&gt;</pre>                               | Genera etiquetas <b>OBJECT</b> o <b>EMBED</b> , apropiadas al tipo de navegador, pidiendo que se ejecute un applet usando el Java Plugin. |  |