

# CURSO DE SERVLETS Y JSPS

# INCLUSIÓN DE ARCHIVOS CON JSP'S



Por el experto: Ing. Ubaldo Acosta



**CURSO DE SERVLETS Y JSPS**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Hola, te saluda Ubaldo Acosta. Bienvenida o bienvenido nuevamente. Espero que estés listo para comenzar con esta lección.

Vamos a estudiar el tema inclusión de archivos con JSP's.

¿Estás listo? Ok, ¡Vamos!

# INCLUSIÓN DE ARCHIVOS EN JSP'S

- Inclusión Estática (translation time):

```
<%@ include file="paginaRelativa.jsp" %>
```

- Inclusión Dinámica (request time):

```
<jsp:include page="paginaRelativa.jsp" />
```

**CURSO DE SERVLETS Y JSPS**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

En esta lección vamos a revisar el tema de la inclusión de archivos utilizando JSPs.

En términos simples la inclusión de archivos, como su nombre lo dice, es agregar un JSP dentro de otro, sin embargo existen dos formas de realizar esta acción cuando utilizamos JSPs.

En primer lugar tenemos la **inclusión estática** también conocida como inclusión en tiempo de traducción (translation time), por otro lado tenemos la **inclusión dinámica** también conocida como inclusión en tiempo de ejecución (request time). Vamos a revisar las diferencias entre cada tipo de inclusión de archivos.

Comencemos con la inclusión estática. Este concepto lo podemos aplicar utilizando la directiva **include** y según vimos anteriormente en el tema de directivas podemos reconocer a esta directiva por medio del símbolo **@**. Posteriormente lo que hacemos es especificar por medio del atributo **file** el nombre del JSP que vamos a incluir en el JSP principal.

En el caso de la inclusión dinámica lo que sucede es que vamos a utilizar, en lugar de una directiva, el concepto de una acción, posteriormente vamos a revisar más acciones, sobre todo para acceder a los JavaBeans, pero en este caso una acción también nos va a permitir hacer un include de manera dinámica de otro JSP. Posterior al nombre de la acción, especificamos el atributo **page** y como valor del atributo el JSP que queremos incluir en nuestro JSP principal.

Podemos observar que parte de las diferencias es que en nuestra directiva `<%@ include file="paginaRelativa.jsp" %>` estamos manejando el atributo de **file** y en la inclusión dinámica `<jsp:include page="paginaRelativa.jsp" />` estamos utilizando el atributo de **page**, estas son simplemente algunas de las diferencias y vamos a ir revisando estos dos conceptos a más detalle.

# INCLUSIÓN ESTÁTICA

## • Sintaxis:

```
<%@ include file="paginaRelativa.jsp" %>
```

## • Objetivo:

- Reutilizar código en texto plano, HTML o JSP's.
- En este caso el JSP que se incluye puede afectar al JSP incluyente

## • Notas:

- En este caso no se delega el control al JSP incluido, sino que se incluye el archivo JSP, generando un único JSP
- Por lo anterior, se pueden tener variables o métodos que se comparten entre los JSP's
- Para tener la versión más reciente, se recomienda compilar y construir nuevamente el proyecto

## CURSO DE SERVLETS Y JSPS

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

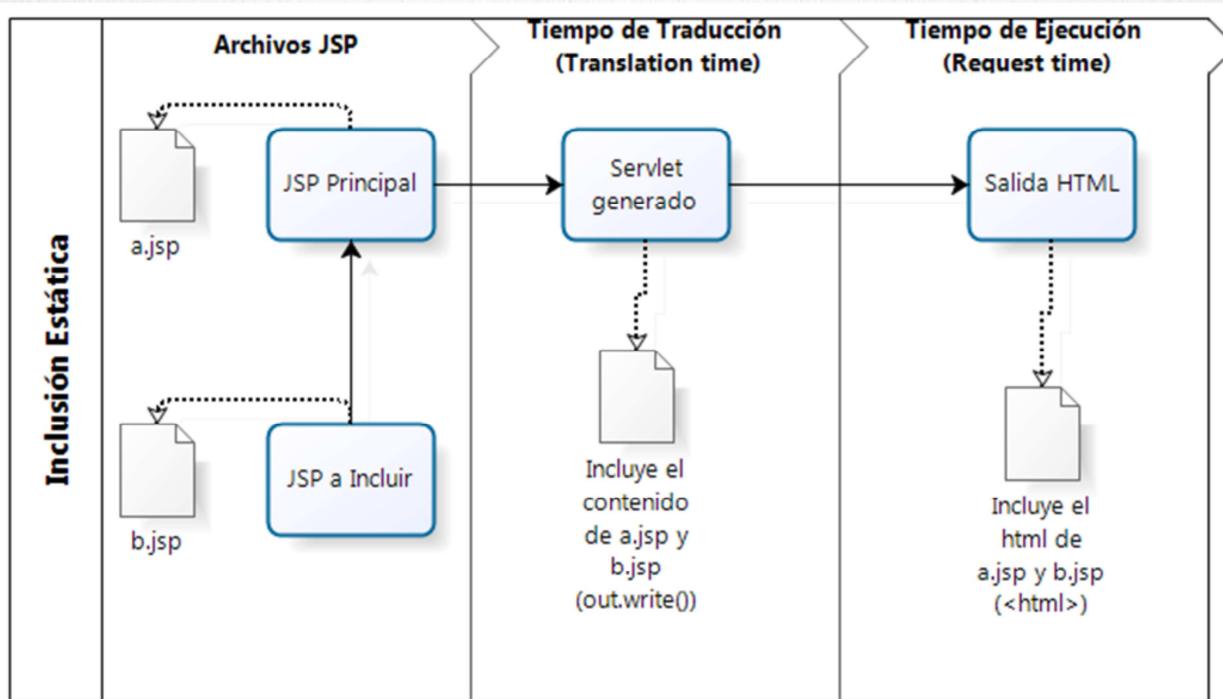
En la inclusión estática, según hemos visto, tenemos la siguiente sintaxis:

<%@ include file="paginaRelativa.jsp" %> vamos a utilizar una directiva, por lo tanto abrimos un picoparéntesis (<), seguido de un signo de porcentaje (%) y utilizamos el símbolo de arroba (@), posteriormente usamos la directiva **include** y finalmente especificamos el nombre del archivo que vamos a incluir en el JSP principal.

El objetivo de utilizar esta inclusión estática es reutilizar código que puede provenir de un texto plano de un HTML o de otro JSP. En el caso de la inclusión estática, el JSP que estamos incluyendo dentro del JSP principal puede afectar o compartir información con el JSP principal. En la inclusión estática no se delega el control al JSP que vamos a incluir, si no que se va a generar un único JSP que va a ser la combinación del JSP principal y del JSP que estamos incluyendo. Recordemos que un JSP se compila finalmente en un Servlet, ya que es necesario la clase Java para poder ejecutar el JSP asociado. Posteriormente veremos un ejemplo para tener más claro este concepto.

Una vez que tenemos ya un JSP o más bien un Servlet de manera única y que contiene a ambos JSPs, entonces podemos compartir variables y métodos entre los diferentes JSPs que estamos utilizando. Uno de los detalles es que, con la inclusión estática, para tener la versión más reciente del JSP que estamos incluyendo se recomienda compilar y reconstruir nuevamente el proyecto debido a que esta inclusión estática no garantiza que tengamos la última versión del JSP incluido. Esto se debe a que el contenido del JSP incluido se va a vaciar en el JSP principal; esto lo vamos a detallar más adelante.

# INCLUSIÓN ESTÁTICA



Vamos a revisar un diagrama de una inclusión estática. Supongamos que tenemos un JSP principal llamado a.jsp y un JSP que vamos a incluir llamado b.jsp. En el primer bloque del diagrama estamos mencionando los archivos jsp con los que vamos a trabajar, en el segundo bloque estamos trabajando ya en tiempo de traducción del JSP, es decir, cuando se genera un servlet y en el tercer bloque estamos especificando el tiempo de ejecución, es decir, cuando mandamos a llamar a nuestro JSP. Veamos a más detalle cada uno de los pasos.

El JSP principal (a.jsp) debe agregar la directiva según hemos visto en la sintaxis de la inclusión estática y va a agregar el contenido del JSP incluido (b.jsp). Una vez que solicitamos nuestro recurso se hace una traducción del JSP, aquí la gran diferencia de la inclusión estática es que únicamente se va a generar un solo Servlet, y este Servlet contendrá el código de los dos JSPs (a.jsp y de b.jsp) y el código HTML generado por cada uno de los JSPs los va a mezclar en un solo Servlet utilizando el método `out.write()`. Este html ya incluye el resultado de los dos JSPs.

Más adelante vamos a ver un ejemplo de este tipo de inclusión.

## INCLUSIÓN DINÁMICA

**• Sintaxis:**

```
<jsp:include page="paginaRelativa.jsp" />
```

**• Objetivo:**

- Reutilizar código en texto plano, HTML o JSP's
- Permite actualizar el contenido sin alterar al JSP que incluye el contenido a reutilizar

**• Notas:**

- El JSP principal, delega el control al JSP incluido y la salida del JSP se agrega al JSP principal.
- No se comparten variables o métodos entre los JSPs, pero garantiza tener siempre la versión más reciente
- La página debe ser relativa a la aplicación Web (uso de / )
- Se pueden incluir archivos privados (desde la carpeta WEB-INF)

### CURSO DE SERVLETS Y JSPS

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Vamos a revisar ahora la inclusión dinámica.

La sintaxis según hemos comentado es una acción de los JSPs, una acción tiene la siguiente sintaxis:

<jsp:include page="paginaRelativa.jsp" /> abrir el picoparéntesis (<) posteriormente ponemos el prefijo jsp y utilizamos dos puntos (:) seguido del nombre include, esta es la acción que estamos utilizando la acción de include y posteriormente especificamos por medio del atributo page la página jsp que vamos a incluir en nuestro jsp principal.

El objetivo de esta inclusión dinámica es la misma que la inclusión estática, lo que queremos conseguir es reutilizar código ya sea que provenga de texto plano de HTML o de alguna otra fuente como puede ser un JSP.

La ventaja de utilizar la inclusión dinámica es que nos va a permitir actualizar el contenido del JSP principal sin alterar al mismo. El JSP principal en lugar de incluir al otro JSP lo que va hacer es delegar el control al JSP incluido y la salida que es el resultado de ejecutar al JSP incluido se va agregar al JSP principal.

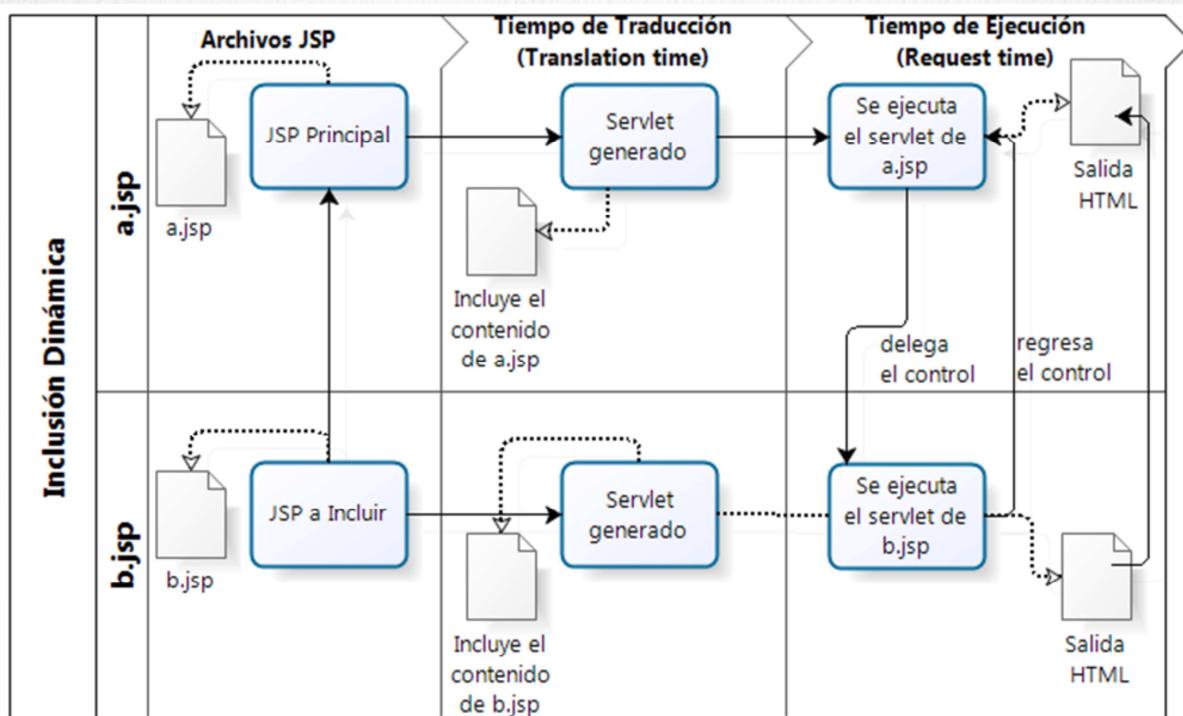
Debido a esta situación se generan 2 servlets, uno por cada JSP, ya que cada JSP se va a ejecutar de manera independiente. Esto implica que no vamos a poder compartir variables o métodos entre los JSPs sin embargo esto nos permite garantizar que siempre vamos a tener la versión más reciente del JSP que estamos incluyendo, ya que cada vez que mandemos llamar una inclusión dinámica se va a ejecutar el JSP incluido de manera independiente debido a que se delega el control al JSP incluido. Una vez que termine de ejecutarse ese JSP incluido regresa el control al JSP principal.

Al igual que la inclusión estática debemos de manejar una inclusión relativa a nuestra aplicación web, por ello será importante el uso de una diagonal (/) y posteriormente vamos a indicar el recurso web que vamos a utilizar.

Otra ventaja de la inclusión dinámica es que vamos a poder incluir también archivos privados, es decir, archivos o JSPs que estén dentro de la carpeta de WEB-INF. Por lo tanto podremos acceder tanto a recursos públicos que estén fuera de la carpeta WEB-INF o a recursos privados, es decir a JSPs o recursos que estén dentro de la carpeta de WEB-INF.

Más adelante realizaremos un ejercicio de este tipo de inclusión.

## INCLUSIÓN DINÁMICA



Vamos a observar ahora un diagrama de inclusión dinámica. Podemos observar que ahora tenemos los mismos bloques pero estamos separando ya a más detalle los archivos a.jsp y b.jsp.

También de igual manera observamos los archivos que vamos a utilizar (a.jsp y b.jsp), el tiempo de traducción y observamos bloque de tiempo de ejecución.

El JSP principal va a utilizar una acción de los JSPs para incluir el archivo b.jsp dentro del archivo a.jsp, posteriormente lo que sucede en tiempo de traducción es que se van a generar dos servleты, uno correspondiente al JSP principal (a.jsp) y el otro correspondiente al JSP a incluir (b.jsp).

Entonces podemos observar que se genera un JSP con el contenido del archivo a.jsp y se genera otro servlet con el contenido del archivo b.jsp, pero aquí viene lo interesante cuando ejecutamos este JSP. Lo que sucede si utilizamos una inclusión dinámica es que estamos solicitando el recurso de a.jsp y en cuanto este JSP detecta la inclusión del archivo que queremos agregar entonces delega el control al otro servlet en este caso es el servlet asociado con el jsp de b.jsp y en cuanto termina de ejecutarse este JSP va a regresar el control al jsp principal y finalmente lo que va hacer es generar una salida, esta salida del servlet incluido se va agregar dentro del JSP principal, por ello es que podemos estar seguros que este JSP que se está incluyendo siempre va a contener la última versión.

Finalmente la salida que se manda al cliente va a tener tanto la salida del JSP principal como la salida del JSP que se incluyó.

## COMPARACIÓN ENTRE LOS TIPOS DE INCLUSIÓN

	Inclusión Estática: <%@ include ... %>	Inclusión Dinámica: <jsp:include ...>
¿Cuando ocurre la inclusión?	En tiempo de traducción	En tiempo de ejecución
¿Qué se incluye?	El contenido del archivo	La salida de la página
Número de Servlets generados	1	2
Puede incluir cabeceros que afecten al JSP principal	Si	No
¿Puede incluir atributos o métodos en común?	Si	No
¿Se debe actualizar la página principal si se actualiza la página incluida?	Si	No
¿El JSP principal delega el control al JSP incluido?	No	Si

Por último vamos a revisar una comparación entre los diferentes tipos de inclusión.

La inclusión estática en la cual utilizamos una directiva y la inclusión dinámica en la cual utilizamos una acción de los JSPs, podemos observar cuando ocurre la inclusión, en el caso de una inclusión estática ocurre en tiempo de traducción y en una inclusión dinámica ocurre en tiempo de ejecución.

También podemos responder a la pregunta de ¿Qué se incluye en el JSP principal? En el caso de la inclusión estática se va a incluir directamente el contenido del archivo y en el caso de la inclusión dinámica se va a incluir la salida de la página JSP.

El número de Servlets generados en una inclusión estática únicamente es 1 y en el caso de una inclusión dinámica se generan dos Servlets, uno por cada JSP, el principal y el que se va a incluir.

También podemos responder a la pregunta ¿Podemos incluir cabeceros que afecten al JSP principal? En el caso de la inclusión estática únicamente se genera un Servlet, por lo que sí podemos modificar al JSP principal por medio de cabeceros desde el JSP que estamos incluyendo. En el caso de la inclusión dinámica esto no puede suceder debido a que los JSPs se ejecutan de manera independiente, por lo que no se comparte información entre ellos, y tampoco cabeceros.

Respecto a la pregunta ¿Podemos incluir atributos o métodos en común? En el caso de la inclusión estática es cierto y es debido a que se genera un único servlet por lo tanto vamos a poder compartir atributos o métodos que estén definidos en un Servlet o en otro y en el caso de la inclusión dinámica esto no es posible debido a que los JSPs nunca se mezclan simplemente se delega el control y por eso es falso.

Respecto a la pregunta ¿Debemos actualizar la página principal si se actualiza la página incluida? En el caso de la inclusión estática esto es cierto para garantizar que tenemos la última versión del JSP incluido. En el caso de la inclusión dinámica esto no es necesario debido a que como delegamos el control siempre se está ejecutando la última versión del JSP incluido y en el caso del JSP principal delega el control al JSP incluido.

A continuación vamos a revisar algunos ejercicios para poner en práctica el uso de las inclusiones estática y dinámica utilizando JSPs.

## EJERCICIOS CURSO SERVLETS Y JSPS

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO:** Ejercicio Inclusión Estática
- **EJERCICIO:** Ejercicio Inclusión Dinámica

**CURSO DE SERVLETS Y JSPS**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# CURSO ONLINE

# SERVLETS & JSPS

Por: Ing. Ubaldo Acosta



## CURSO DE SERVLETS Y JSPS

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

En Global Mentoring promovemos la Pasión por la Tecnología Java. Te invitamos a visitar nuestro sitio Web donde encontrarás cursos Java Online desde Niveles Básicos, Intermedios y Avanzados, y así te conviertas en un experto programador Java.

Además agregamos nuevos cursos para que continúes con tu preparación como programador Java profesional. A continuación te presentamos nuestro listado de cursos:

- |                          |                                     |
|--------------------------|-------------------------------------|
| ✓ Lógica de Programación | ✓ Hibernate Framework               |
| ✓ Fundamentos de Java    | ✓ Spring Framework                  |
| ✓ Programación con Java  | ✓ JavaServer Faces                  |
| ✓ Java con JDBC          | ✓ Java EE (EJB, JPA y Web Services) |
| ✓ HTML, CSS y JavaScript | ✓ JBoss Administration              |
| ✓ Servlets y JSP's       | ✓ Android con Java                  |
| ✓ Struts Framework       | ✓ HTML5 y CSS3                      |

### Datos de Contacto:

Sitio Web: [www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Email: [informes@globalmentoring.com.mx](mailto:informes@globalmentoring.com.mx)

