



ESTANDARES DE DESARROLLO WEB

Nombrado de Clases y Páginas JSP

Edición: V 2.0

**Autor: Unidad de Informática Corporativa
Área de Desarrollo y Mantenimiento**

Fecha: Abril 2005

Historia del Documento			
Versión: beta	<i>Descripción:</i> Documento inicial sobre la política de nombrado estandar para las clases y páginas de una aplicación Web J2EE		
	<i>Elaborado por:</i>	Juan A. Barras García	<i>Fecha:</i> Octubre 2002
	<i>Revisado por:</i>		<i>Fecha:</i>
			<i>Fecha:</i>
			<i>Fecha:</i>
	<i>Aprobado por:</i>		<i>Fecha:</i>
Versión: 1.0	<i>Descripción:</i> Documento revisado y ampliado sobre la política de nombrado estándar para las clases y páginas de una aplicación Web J2EE		
	<i>Revisado por:</i>	Diego García Carrera	<i>Fecha:</i> Enero 2003
		Juan A. Barras García	<i>Fecha:</i> Enero 2003
			<i>Fecha:</i>
			<i>Fecha:</i>
	<i>Aprobado por:</i>	Juan A. Barras García	<i>Fecha:</i> Enero 2003
Versión: 2.0	<i>Descripción:</i> Documento revisado y ampliado sobre la política de nombrado estándar para las clases y páginas de una aplicación Web J2EE		
	<i>Elaborado por:</i>	Diego García Carrera	<i>Fecha:</i> Marzo 2005
	<i>Revisado por:</i>	Juan A. Barras García	<i>Fecha:</i> Abril 2005
			<i>Fecha:</i>
			<i>Fecha:</i>
	<i>Aprobado por:</i>	Juan A. Barras García	<i>Fecha:</i> Abril 2005

Lista de distribución del documento

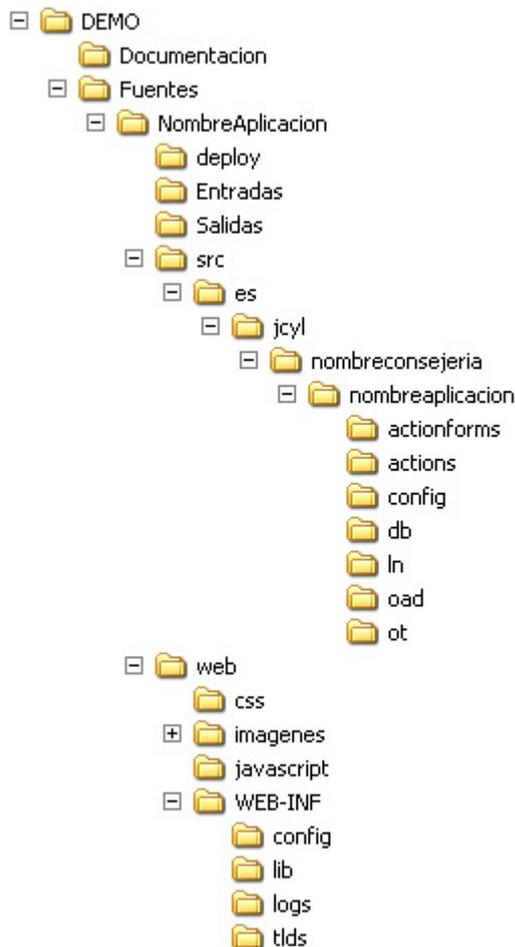
Nombre	Localización
	PUBLICO

Índice

1	Estructura de directorios/paquetes de una aplicación Web	4
2	Nombrado de Clases	4
2.1	Clases de la capa de presentación: Struts	5
2.2	Clases de la capas de integración	6
3	Nombrado de Páginas JSP	7
3.1	Ubicación de las páginas JSP	7
3.2	Nombrado de las páginas	7

1 Estructura de directorios/paquetes de una aplicación Web

La estructura de directorios / paquetes en las que debe estar divididas las clases y páginas de una aplicación Web son las siguientes:



nombreconsejeria : Se trata de un acrónimo de 2 o 3 caracteres donde se especifica la consejería o unidad propietaria de ese código.

Los valores usados son: cf, cee, gss, csbs, cma, etc Ese valor suele coincidir con el valor de NombreAplicacion de la jerarquía superior, aunque esta suele estar en Mayúsculas.

nombreaplicacion : Nombre de 4 caracteres con el identificador de la aplicación a la que pertenecen las clases generadas. Ejemplos : reti, coni, cfio, etc

El directorio con las fuentes se organiza en un subdirectorío con la versión consolidada, denominado NombreAplicación, y otros tantos subdirectoríos, uno por programador, en los que éstos podrán tener sus versiones de trabajo.

2 Nombrado de Clases

El objetivo es tener un criterio uniforme a la hora de nombrar elementos de programación. Los nombres deben ser lo suficientemente descriptivos como para que podamos asociar qué elementos intervienen en el mismo caso de uso, en aras de facilitar el mantenimiento de la aplicación, y de ser lo suficientemente autoexplicativos como para inferir qué hace cada elemento de la aplicación.

El nombrado de una clase sigue el formato genérico *propósito_de_la_clase+tipo_de_clase*.

Por supuesto, las clases siguen el estándar Java de nombrado que exige que la letra inicial sea mayúscula. La cadena *propósito_de_la_clase*, no debería exceder los 12 caracteres, pero pueden utilizarse más si ello hace más identificable la clase. Asimismo, si contiene varias palabras, éstas estarán concatenadas y sus iniciales irán en mayúsculas, por ejemplo *AltaUsuario*

En función de en qué capa se esté programando, tenemos:

2.1 Clases de la capa de presentación:

A estar basada esta capa en Struts, los elementos a nombrar son los establecidos por dicho FW, básicamente acciones, actionforms y el fichero de configuración

Estas acciones son las URIs de la *request* que se envía al servidor Web donde reside la aplicación, y que vienen especificadas en el fichero Struts-Config.xml asociado a la aplicación.

Estas acciones se invocan desde elementos *submit* de Javascript, o desde *actions* dentro de elementos Form de una Html.

La clase action asociadas deben tener las siguientes características:

paquete: es.jcyl.nombreconsejeria.nombreaplicacion.actions

Formato de nombrado : <Numero limitado caracteres que indique su funcionalidad>+Action

Las clases ActionForms :

paquete: es.jcyl.nombreconsejeria.nombreaplicacion.actionforms

Formato de nombrado : <Numero limitado caracteres que indique el formulario>+Form

NOTA: En el caso en el que el numero de elementos del paquete principal sea grande y la aplicación posea subsistemas entonces se aconseja crear subpaquetes a partir del paquete indicado

El fichero de configuración tendrá como nombre aplicación-config.xml. En caso de que la aplicación exija particionar este fichero en varios de tamaño más manejable, el nombrado será una combinación entre el nombre de la aplicación y el del subsistema.

2.2 Clases de la capa de Lógica de Negocio

Son las clases del tipo

LN: Clases de lógica de negocio

Clases de apoyo para publicar servicios como EJBs

Clases de apoyo para publicar servicios como WebServices

LN: Clases de lógica de negocio

paquete: es.jcyl.nombreconsejeria.nombreaplicacion.ln

Formato de nombrado: <Numero limitado caracteres que indique su funcionalidad>+LN

Clases para publicación de EJBs

paquete: es.jcyl.nombreconsejeria.nombreaplicacion.ejb.nombredeIEJB

Formato de nombrado: <Numero limitado caracteres que indique su funcionalidad>+[' | Home | Bean] en función de si la clase es el interface Remote, el Home, o es el Bean propiamente dicho.

Clases para publicación de WebServices

Aunque el contenedor de WebServices estándar, Axis, puede usar directamente los métodos de una clase de Lógica de Negocio para publicarles como servicios, en ocasiones puede ser interesante crear una clase *proxy* intermedia para restringir dichos servicios. La clases de apoyo tendrán entonces el **Formato de nombrado:** <Numero limitado caracteres que indique su funcionalidad>+Service

2.3 Clases de la capas de integración

Son las clases del tipo

OAD : Objetos de acceso a datos

OT: Objetos transferencia

Otras clases de Utilidades, Auxiliares, etc

Para cada uno de los siguientes elementos se deben ubicar y nombrar de la siguiente manera:

OAD : Objetos de acceso a datos

paquete: es.jcyl.nombreconsejeria.nombreaplicacion.oad

Formato de nombrado: <Numero limitado caracteres que indique la entidad de datos>+OAD

OT: Objetos transferencia

paquete: es.jcyl.nombreconsejeria.nombreaplicacion.ot

Formato de nombrado: <Numero limitado caracteres que indique la entidad de datos>+OAD

Otras clases de Utilidades, Auxiliares, etc

paquete: es.jcyl.nombreconsejeria.nombreaplicacion.util

Formato de nombrado: <Numero limitado caracteres que indique su funcionalidad>

NOTA: Al igual que antes, en el caso en el que el numero de elementos del paquete principal sea grande y la aplicación posea subsistemas entonces se aconseja crear subpaquetes a partir del paquete indicado

3 Nombrado de Páginas JSP

3.1 Ubicación de las páginas JSP

Las páginas JSP de nuestro proyecto deben estar ubicadas todas a partir del directorio Fuentes\NombreAplicacion\web. En el caso de que el número sea grande o existan subsistemas entonces se aconseja crear subdirectorios a partir del mismo donde ubicar tales páginas.

3.2 Nombrado de las páginas

El nombrado de páginas JSP de nuestro proyecto debe componerse de los siguientes apartados:

Formato de nombrado: <Apli><Caso de Uso o Funcionalidad><Tipo de Acción>

Donde:

<Apli> = 4 alfanuméricos con el código de la aplicación a la que pertenece.

<Caso de Uso o Funcionalidad> = Numero limitado de caracteres (8 preferentemente) con el Caso de Uso en el que se utiliza la JSP.

<Tipo de Accion> = 2 Con el código de la acción asociada a esta JSP, de acuerdo a esta tabla, mas 2 Numéricos opcionales que contienen una secuencia, en caso de que sea necesario encadenar JSPs del mismo tipo (altas encadenadas, por ejemplo)

Tipo Pagina	Código	Descripción del tipo de página
Alta	Al	Denota páginas preparadas para el alta de datos
Baja	Bj	Denota páginas preparadas para dar de baja datos
Modificación	Mf	Denota páginas preparadas para la modificación de datos
Consulta	Cn	Introducción de datos para realizar una búsqueda o consulta
Datos	Dt	Presentación de datos resultado de una operación
Listado	Ls	Páginas JSP que son susceptibles de ser impresas (listados de punteo, pequeños informes, páginas con documentos PDF embebidos)

Opcionalmente, se puede detallar más la función de una página empleando estos sufijos:

Tipo Pagina	Código	Descripción del tipo de página
LookUps	Lk	Pantallas 'hijas' flotantes de tipo 'código/descripción', que a partir de una búsqueda/navegación por el campo 'descripción', pueden enviar el campo 'código' a la JSP 'padre'
Navegación entre listas de resultados	Nv	Permite iterar entre los items de una lista presentada en bloques de líneas
Maestro/detalle	Md	Muestra datos Maestro y navegación entre datos detalle
Consulta Maestro/detalle	Cd	Permite introducir información en los campos de la parte 'maestro' y ver los resultados en la parte 'detalle'
Lanzamiento de proceso Batch	Bt	Permite introducir datos necesarios para lanzar un proceso Batch en concreto
Lanzamiento de proceso Background	Bk	Permite introducir datos necesarios para lanzar un proceso Background en concreto
Menú	Mn	Páginas que albergan menús especiales.

Ejemplo:

Supongamos que tenemos una aplicación cuyo fin sea la Asignación y Reserva de Salas de Conferencia de un departamento. Su código es ARSC.

En ella hay una funcionalidad que nos permita el mantenimiento de personas de ese departamento. Su identificación es MtoPerso

Desde una página en la que hay un menú, pedimos otra página donde podamos dar de alta un registro en una tabla. Una vez introducidos los datos, presionamos el botón 'grabar' y el sistema nos devuelve una página con los datos grabados, a modo de confirmación.

Página de alta: ARSCMtoPersoAl.jsp

Página devuelta con los datos: ARSCMtoPersoDt.jsp

En caso de que los datos de la persona fueran demasiados para que cupieran de forma legible en una única JSP, y fuera necesario emplear una segunda JSP, su nombre sería ARSCMtoPersoA102.jsp