

JSR 168 (Java Specification Requests for Java Portlet Specification)

ÍNDICE

ÍNDICE.....	2
1. Evidencia.....	3
1.1 Evidencia de la investigación	4
1.1.1 Tema de Investigación.....	4
1.1.2 Descripción del tema	4
1.1.3 Resultado de la Investigación	4

1. Evidencia

1.1 Evidencia de la investigación

1.1.1 Tema de Investigación

JSR 168 (Java Specification Requests for Java Portlet Specification).

1.1.2 Descripción del tema

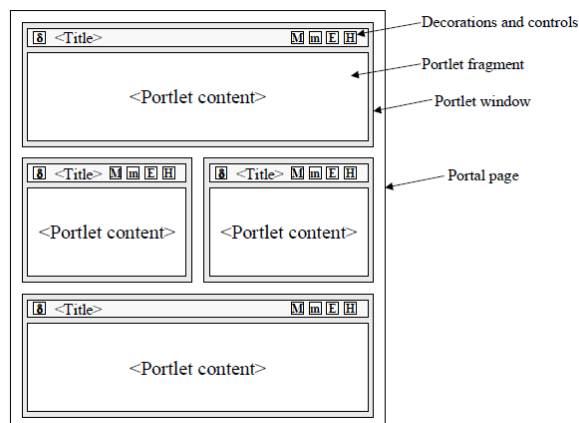
Estándar de Java acerca del uso y configuración de un Portlet.

1.1.3 Resultado de la Investigación

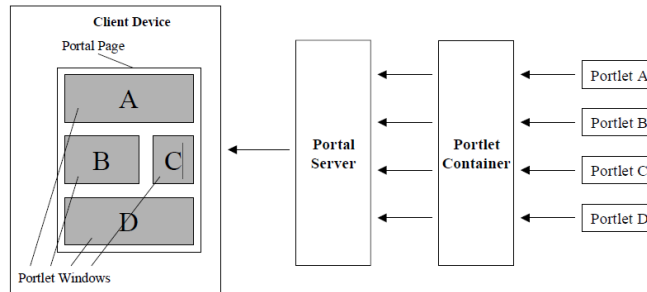
El documento especifica la relación que existe entre un Portlet y un Servlet, esto hizo que entendiera del tema más rápido, ya que el tema de Servlets es un tema básico en el curso de Plataforma de Desarrollo Java.

Las ideas más resaltantes acerca de los Portlets son las siguientes:

1. Los Portlets generan contenido dinámico como los Servlets.
2. Los Portlets son administrados por un contenedor especial.
3. El ciclo de vida de un Portlet es administrado por ese contenedor
4. Los Portlets interactúan con el cliente web mediante el paradigma request/response.
5. Los elementos de un Portlet son: Decoración y controles, Portlet Fragment, Portlet Window, Portal Page.



6. El contenedor Portlet recibe el contenido generado por los Portlets. El servidor del Portlet crea el portal con el contenido generado por los Portlets y los envía a un browser donde este se muestra al usuario.



7. El ciclo de vida del Portlet se define como: cargado, instanciado e inicializado. Es expresado a través del *init*, *processAction*, *render* y el método *destroy* de la interfaz del Portlet.
8. El método *processAction* recibe dos parámetros *ActionRequest* y *ActionResponse*.
9. El método *render* recibe dos parámetros *RenderRequest* y *RenderResponse*.
10. La clase abstracta *GenericPortlet* provee funcionalidad para manejar los requests. Siempre se invoca el método *doDispatch* (como en el caso de los Servlets). A su vez el método *doDispatch* implementa funcionalidad, estos métodos son: *doView* (para ver el request), *doEdit* (para editar el request), *doHelp* (para “ayudar” al request).
11. Los parámetros de inicialización son *getInitParameterNames* y *getInitParameter*.
12. El paquete de recursos de los portlets contiene elementos que pueden ser directamente incluidos en la definición del portlet en el descriptor de despliegue. Un ejemplo de un descriptor de despliegue definiendo la información del portlet puede ser:

```
<portlet>
...
<portlet-info>
  <title>Stock Quote Portlet</title>
  <short-title>Stock</short-title>
  <keywords>finance,stock market</keywords>
</portlet-info>
...
</portlet>
```

13. Si los recursos están definidos en un paquete, el portlet debe proveer el nombre de dicho paquete. Ejemplo:

```
<portlet>
...
<resource-bundle>com.foo.myApp.QuotePortlet</resource-bundle>
...
</portlet>
```

14. El API del Portlet define la interface *PortletURL*. Un Portlet crea objetos *PortletURL* invocando el *createActionURL* y *createRenderURL*.
15. Un ejemplo de cómo crear un Portlet URI puede ser:

```

...
PortletURL url = response.createRenderURL();
url.setParameter("customer","foo.com");
url.setParameter("show","summary");
writer.print("<A HREF=\"" + url.toString() + "\">Summary</A>");
...

```

16. Para poder incluirle un modo al Portlet o un Windows State los métodos son los siguientes: *setWindowState* y *setPortletMode*. Ejemplo:

```

...
PortletURL url = response.createActionURL();
url.setParameter("paymentMethod","creditCardInProfile");
url.setWindowState(WindowState.MAXIMIZED);
writer.print("<FORM METHOD=\"" POST "\" ACTION=\"" + url.toString() + "\">");
...

```

17. Con el método *setSecure* asegura que el URL del Portlet es seguro (HTTPS).

18. Modos de los portlets: View, edit, help. Además existen modos customizados. Ejemplo:

```

<portlet-app>
...
<custom-portlet-mode>
<description>Creates content for Cut and Paste</description>
<name>clipboard</name>
</custom-portlet-mode>

<custom-portlet-mode>
<description>Provides administration functions</description>
<name>config</name>
</custom-portlet-mode>
...
</portlet-app>

```

19. Se pueden definir modos de soporte de la siguiente manera:

```

...
<supports>
<mime-type>text/html</mime-type>
<portlet-mode>edit</portlet-mode>
<portlet-mode>help</portlet-mode>
...
</supports>
<supports>
<mime-type>text/vnd.wap.wml</mime-type>
<portlet-mode>help</portlet-mode>
...
</supports>
...

```

20. Estados de ventana: Normal, maximizado, minimizado. Además se pueden customizar las ventanas de la siguiente manera:

```

<portlet-app>
...
<custom-window-state>
<description>Occupies 50% of the portal page</description>
<name>half_page</name>
</custom-window-state>
...
</portlet-app>

```

21. Los parámetros iniciales accesibles a través del PortletContext debe ser la misma que los del ServletContext de la aplicación portal (en el web.xml).

22. Los métodos del PortletContext deben proveer la misma funcionalidad que los métodos del ServletContext de nombre similar: getAttribute,getAttributeNames, getInitParameter, getInitParameterNames, getMimeType,getRealPath, getResource, getResourcePaths, getResourceAsStream, log, removeAttribute y setAttribute.
23. Los parámetros del PortletsRequest son: getParameter, getParameterNames, getParameterValues, getParameterMap (se encuentran en la librería “javax.portlet”).
24. Los request attributes son: getAttribute, getAttributeNames, setAttribute, removeAttribute.
25. Los request properties son: getProperty, getProperties, getPropertyNames.
26. Los atributos de seguridad: getAuthType, getRemoteUser, getUserPrincipal, isUserInRole, isSecure.
27. El ActionRequest provee: getContentType, getCharacterEncoding, setCharacterEncoding, getContentLength.
28. Los Response Properties son: setProperty, addProperty.
29. Para setear la información del buffer: getBufferSize, setBufferSize, isCommitted, reset, resetBuffer, flushBuffer.
30. La interface PortletPreferences proveen los siguientes métodos: getNames, getValue, setValue, getValues, setValues, getMap, isReadOnly, reset y store.

Bibliografía:

<http://jcp.org/aboutJava/communityprocess/final/jsr168/index.html>