

Capítulo 3

OGCE, *Open Grid Computing Environment*

3.1. Descripción general

OGCE es un portal Grid que proporciona un punto de acceso sencillo y transparente a los diferentes recursos de un grid computacional que un usuario esté autorizado a utilizar. Según Li, Maozhen et al [7] este portal proporciona a los usuarios de grid una interfaz personalizada de los recursos de software y hardware específicos al dominio particular del usuario de grid.

3.1.1. Arquitectura de los portales Grid

Los portales grid están compuestos de tres capas que se ilustran en la figura 3.1

1. La primera capa consiste de navegadores web.
2. La segunda capa de los servidores web.
3. La tercer capa la conforman los recursos y servicios finales del grid computacional, como pueden ser bases de datos, equipos de alto rendimiento, dispositivos de almacenamiento.

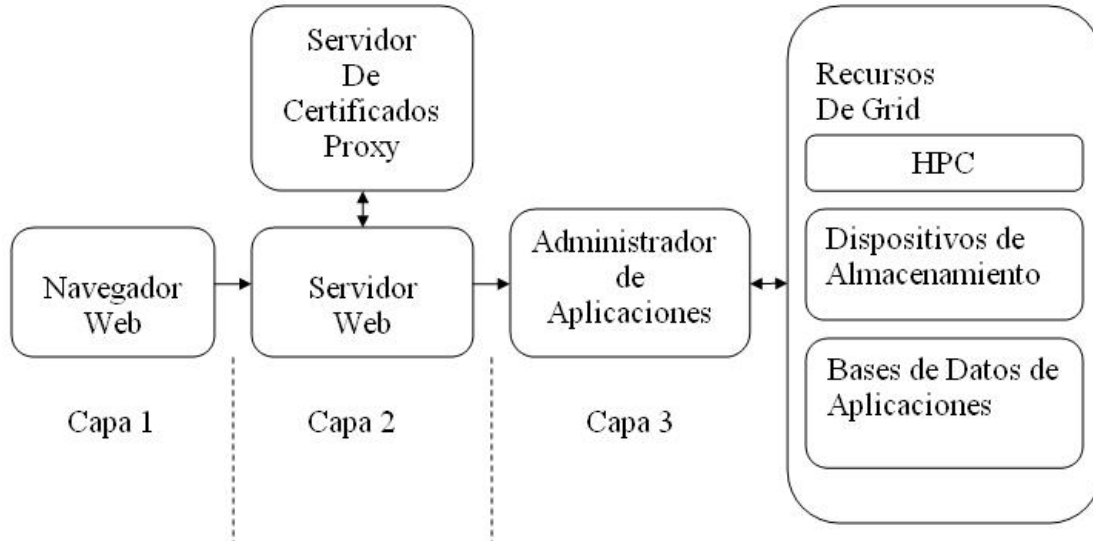


Figura 3.1: Arquitectura de un portal Grid

3.1.2. Funcionamiento de un portal grid

Un usuario de grid utiliza el navegador web para conectarse de forma segura con el servidor web. Posteriormente el servidor web obtiene el certificado proxy del usuario de un servidor de certificados proxy y utiliza este certificado para autentificar al usuario con los recursos del grid computacional.

Una vez que el portal grid ha obtenido el proxy, el usuario puede interactuar con los recursos de grid. Para utilizar estos recursos, el usuario mediante la interfaz del portal grid puede definir los parámetros necesarios para ejecutar un trabajo, ante esto el servidor web ejecuta el administrador de aplicaciones (*application manager*); el cual controla y monitorea la ejecución de las tareas en un grid computacional. El servidor web delega el certificado proxy del usuario al administrador de aplicaciones para que éste pueda actuar en nombre del usuario ante los recursos del grid.

Algunos de los servicios que proporciona un portal grid son:

- Autenticación:

Este servicio se encarga de autentificar a los usuarios utilizando su certificado proxy ante un grid computacional mediante un portal grid. Mediante este servicio, el usuario puede pedir al portal grid acceder a los recursos del grid a nombre suyo.

- Administración de Trabajos

Este servicio permite a los usuarios de un grid administrar sus trabajos de una forma cómoda y sencilla. Así los usuarios pueden mandar a ejecución sus trabajos a través de un navegador web (de forma confiable y segura), monitorear el estado de sus trabajos y cancelar trabajos cuando es necesario.

- Transferencia de información

Este servicio permite a los usuarios del portal grid transferir información en los recursos de grid desde sus máquinas locales. La información que se puede transferir puede ser de diversa índole, como archivos de configuración, archivos de entrada para la ejecución de sus aplicaciones o archivos de salidas generadas por las mismas aplicaciones.

- Servicios de información.

Este servicio muestra información de los diferentes recursos de un grid. Esta información es recabada de los diferentes recursos de cómputo de un grid y es publicada para el conocimiento de los usuarios. La información recabada puede ser de dos tipos: estática o dinámica. Estática se refiere a información como tipo de Sistema Operativo o CPU; dinámica es la carga que tiene el CPU, memoria libre, espacio en disco.

3.2. Portlets

Los portlets son importantes porque permiten la personalización de los portales grid, hace a los portales poco acoplados con los middleware de grid, etc.

3.2.1. Definición de Portlet

Un portlet gráficamente es una ventana que ofrece un servicio dentro de un portal grid. Es decir, un portlet visualmente tiene su propia ventana, título (de portlet), contenido (de portlet) y acciones (maximizar, minimizar).

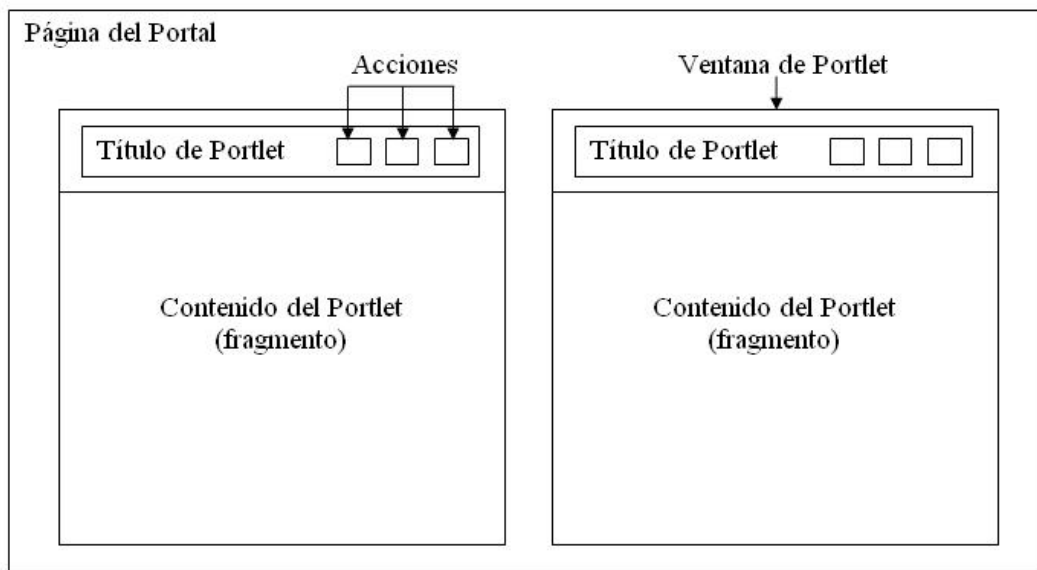


Figura 3.2: Portlet

Específicamente, un portlet es un componente de software administrado por un contenedor de portlet, que maneja las peticiones hechas por un usuario (de un portal grid) y genera contenido dinámico. Así los portlets como componentes de la interfaz de usuario, pasan información a la capa de presentación de un portal.

El contenido que genera un portlet es llamado fragmento. Ese fragmento es un conjunto de código HTML. La totalidad de los fragmentos generados por los portlets forman un documento completo de una página web.

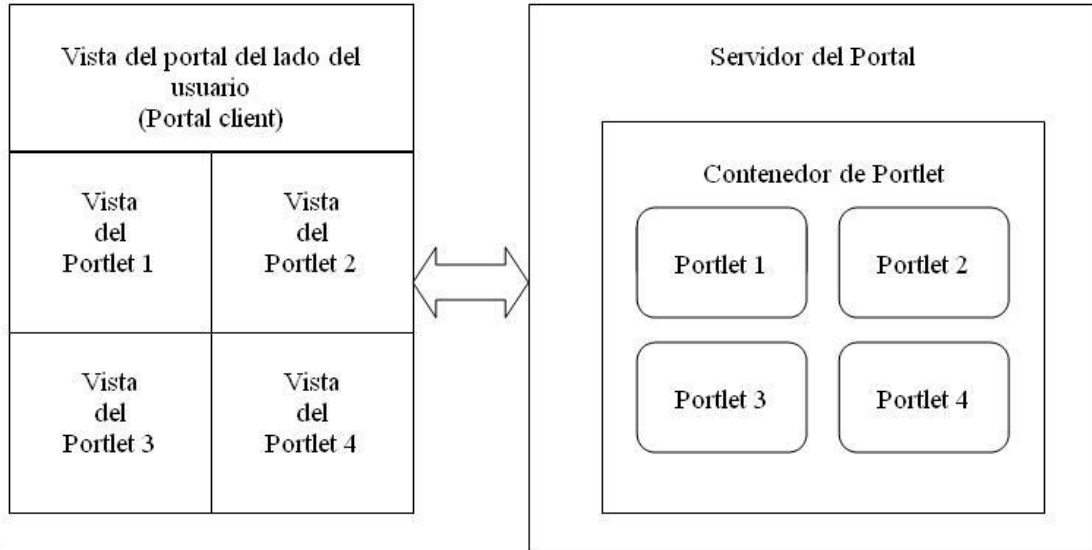


Figura 3.3: Portal con portlets

Contenedor de portlets

Un contenedor de portlets administra el ciclo de vida de los portlets en un portal. Así un contenedor de portlets proporciona un ambiente de ejecución para que los portlets sean invocados, ejecutados y destruidos. Los portlets se basan en la infraestructura del portal para interactuar con otros portlets, buscar certificados, almacenar información, acceder a contenido e información del usuario, etc.

Específicamente el contenedor de portlets está implementado en una capa superior a un contenedor de servlet, lo que le permite reutilizar la funcionalidad que proporciona los contenedores de servlet.

Portlets y Servlets

Los portlets al igual que los servlets procesan peticiones HTTP, realizan acciones y generan salida HTML. En el caso de un portlet su salida es sólo una parte de la página web. El servidor del portal se encarga de reunir toda información generada por los portlets y darles formato para la creación de la página web.

Las características de los portlets hacen que sean más flexibles y dinámicos que los servlets. De esta manera se pueden realizar los siguientes cambios en un portal sin necesidad de reiniciar el servidor del portal:

- Los portlets pueden ser creados y borrados dinámicamente.
- El administrador del portal puede cambiar los parámetros de configuración de un portlet.
- Una aplicación de portlets puede ser instalada o removida de la interfaz del usuario del portal.

Particularidades de un Portlet

La ventana de un portlet está compuesta de los siguientes elementos:

- Barra de título: Se muestra el título del Portlet.
- Decoraciones: Botones para cambiar el estado de la ventana del portlet (maximizar, minimizar); cambio en el modo del portlet, como por ejemplo, editar parámetros de configuración del portlet o solicitar ayuda del funcionamiento del portlet.
- Contenido producido por el portlet.

El ciclo de vida de un portlet es el siguiente:

- Invocación (*initialization*): Se utiliza la clase que inicializa al portlet y lo pone en servicio.
- Manejo de peticiones: Se reciben peticiones de la interacción de un usuario con un portlet en un portal y posteriormente se hace un procesamiento de acciones, se genera y actualiza (*rendering*) el contenido.
 - En el procesamiento de acciones se realizan las acciones solicitadas por un usuario, como pueden ser un click en una liga de un portlet. El procesamiento de acciones debe finalizar antes de que se actualicen(*rendering*) los otros portlets. En esta fase, el portlet puede cambiar el estado del portal.

- Generación y actualización de contenido (*rendering content*): En esta fase cada portlet produce su contenido (de *markup*) que será enviado al usuario del portal. Esta actualización no cambia el estado del portlet, sólo muestra las actualizaciones de la página web sin modificar el estado del portlet. La actualización de portlets puede ser desarrollada de forma paralela.
- Finalización: Se remueve un portlet de la página del portal utilizando la clase correspondiente.

Accediendo web services a través de portlets

Según Li, Maozhen et al [7] un portal web recibe una petición de un usuario, lo cual genera y despacha eventos al portlet utilizando los parámetros de la solicitud y entonces se invoca al portlet a ser desplegado mediante la Interfaz de Invocación de Portlet *Portlet Invocation Interface*, PII. El diseño interno de un portlet es similar al modelo Modelo Vista Controlador (MVC): en la parte del controlador se reciben las peticiones entrantes de PII; mientras que en la parte del modelo (encapsula la información y la lógica de la aplicación que accede a las aplicaciones o contenido web) se realizan las operaciones necesarias sobre los recursos web para obtener la respuesta a la petición del usuario; y finalmente en la parte de la vista se presentan los resultados al usuario.

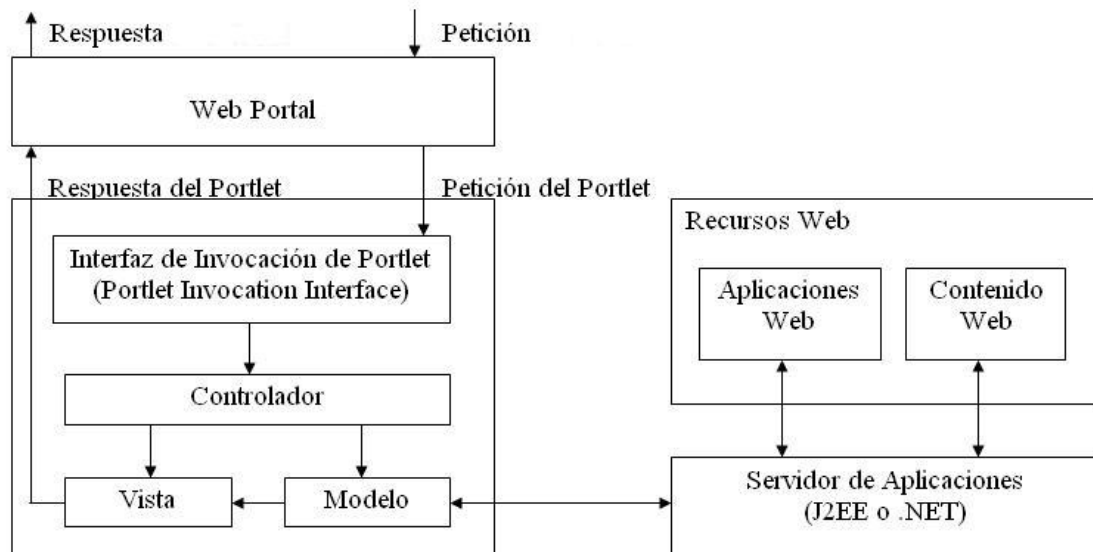


Figura 3.4: Estructura interna de un Portlet

Eventos para acceder a una página web con portlets

La secuencia de eventos para acceder a un a página web a través de portlets es mostrada a continuación:

- Un usuario desde un navegador web y después de haber sido autenticado, realiza una petición HTTP al portal.
- El portal recibe la solicitud
- El portal determina si la petición contiene una acción a ser realizada por alguno de los portlets asociados con la página del portal.
- Si la acción está asociada a un portlet, entonces el portal solicita al contenedor de portlets invocar al portlet para llevar a cabo el procesamiento de la acción.
- El portal invoca a los portlets restantes, a través del contenedor de portlet para obtener sus fragmentos que serán incluidos en la página del portal.
- El portal agrega los fragmentos en la página del portal y se manda esta al usuario.

Especificaciones de Portlets

Con el objetivo de tener interoperabilidad entre las diferentes implementaciones de portlets surgieron las siguientes especificaciones:

- WSRP (*Web Services for Remote Portlets*). Esta especificación proporciona una forma de fácil conexión (*plug and play*) de portlets, actúa como intermediario de aplicaciones de contenido e integra aplicaciones de diferentes orígenes. Así WSRP permite a las aplicaciones utilizar o producir *web services*. Estos web services incluyen elementos e información de presentación que permiten seleccionar y mostrar los portlets de diferentes proveedores sin la necesidad de integración de código. Los objetivos de WSRP son:
 - Permitir a los web services ser conectados dentro de portales que sigan el estándar.
 - Permitir crear y publicar contenidos y aplicaciones como web services.
 - Permitir la publicación de portlets de un portal para que puedan ser utilizados por otros portales sin complicaciones de integración.

- JSR 168. JSR (*Java Specification Request*) 168 es una especificación de portlet que define un conjunto de APIs de Java que permite la interoperabilidad entre portales y portlets. Define a un portlet como un componente web basado en Java, administrado por un contenedor de portlet que procesa peticiones de usuario y genera contenido dinámico. Los portales utilizan a los portlets como componentes de interfaz de usuario conectables que proporcionan una capa de presentación a los sistemas de información. Los objetivos de esta especificación son los siguientes:
 - Define un ambiente de ejecución (contenedor de portlets) para los portlets.
 - Define un API entre el contenedor de portlet y los portlets.
 - Proporciona mecanismos para almacenar información de los portlets.
 - Permite a los portlets incluir servlets y JSPs.
 - Define cómo empaquetar portlets para su fácil uso.
 - Permite que los portlets que cumplan con esta especificación puedan ser ejecutados como portlets remotos utilizando WSRP.

De acuerdo a Li, Maozhen et al [7] estas especificaciones son complementarias. La especificación JSR 168 define un API de Portlet estándar específica a los portales basados en Java, mientras que WSRP define un API general que permite a los portales utilizar portlets de cualquier tipo. De esta manera los portlets que siguen la especificación JSR 168 pueden ser encapsulados (*wrapped*) y presentados como servicios WSRP para su publicación como servicios; mientras que servicios WSRP pueden ser utilizados como portlets con la API de Portlet de Java para ser incluidos en los portales. Es decir, el JSR 168 define un conjunto de APIs de Java que permite a los portlets ejecutarse en portales que sigan esta especificación y WSRP permite a los web services ser mostrados como portlets en una forma de fácil conexión.

GridSphere

El portal Grid OGCE está basado en el *framework* de GridSphere. Este *framework* ayuda a la creación de portales web que utilizan portlets. GridSphere es un proyecto de código libre que proporciona un *framework* de portlet y una infraestructura para reutilizar portlets. De esta manera los desarrolladores pueden crear y empaquetar aplicaciones basadas en portlets de terceras partes y ejecutarlas dentro del contenedor de portlet de GridSphere.

GridSphere tiene un conjunto de portlets y servicios básicos que proporcionan la infraestructura necesaria para crear, desarrollar y administrar un portal web. Las características de GridSphere son las siguientes:

- Soporta portlets de terceras partes, lo que permite que estos sean integrados fácilmente en el contenedor de portlet de GridSphere.
- Portlets base (*core*) que proporcionan una funcionalidad básica para conexión al portal y administración de usuarios.

GridSphere proporciona específicamente un portal, un contenedor de portlets y un conjunto de portlets básicos. Entre estos portlets se encuentran portlets para la administración de usuarios o grupos y un portlet que permite agregar o quitar portlets del ambiente de trabajo del usuario.

Grid Portlet

Un Grid Portlet es un portlet dentro de un portal grid y que además está asociado con un servicio grid final (*backend grid service*). La figura 3.5 muestra como acceder a un servicio grid de un portal grid mediante un portlet:

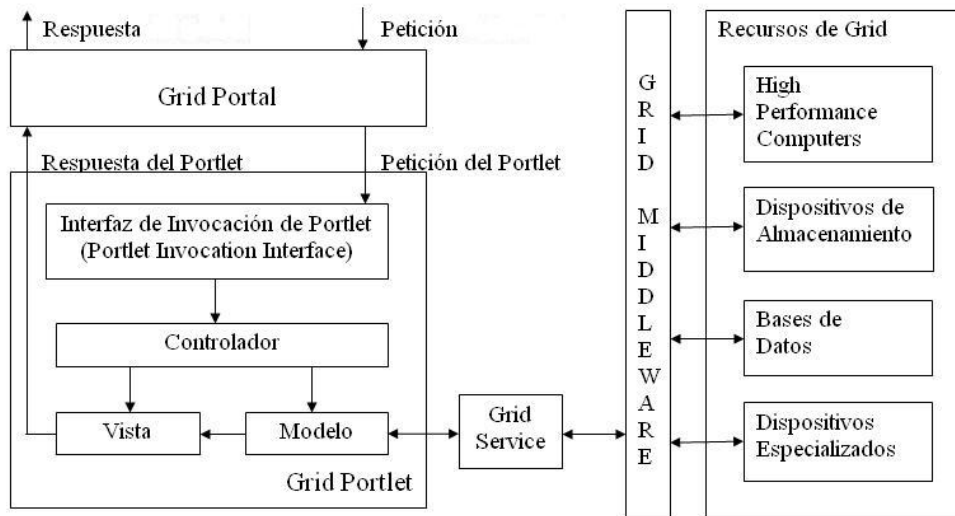


Figura 3.5: Grid Portlet

El Grid Portlet interactúa con un servicio de grid, este servicio es proporcionado por el middleware de grid para acceder a *fabric*, los recursos físicos de grid, (ver sección 2.1.3). Como los servicios grid son proporcionados por diferentes proveedores de servicio que pueden utilizar diferentes middlewares de grid, estos servicios pueden ser mostrados como portlets estándar, de esta forma los portales que utilizan portlets están desacoplados de los middlewares de grid.

3.2.2. Descripción de OGCE

OGCE fue creado con recursos de la *National Science Foundation Middleware Initiative* (NMI) de los EUA en 2003, con el fin de crear colaboraciones para los portales grid. Específicamente estableció un colaboración para portales grid que proporciona un foro en línea para desarrolladores de portales, un depósito de portlets y construyó un conjunto de componentes de portales reutilizables para ser integrados en un contenedor de un portal.

OGCE está vinculado con los siguientes proyectos:

- Java CoG
- CHEF Project
- GPIR y GridPort
- The Alliance Portal Expedition project.

Cabe mencionar que OGCE apoya tecnológicamente al portal de usuarios del Teragrid 2.2.3.

Específicamente, el portal OGCE está compuesto por el contenedor portlet de GridSphere y sus portlets cumplen con la especificación JSR 168.

La arquitectura de OGCE se muestra en la figura 3.6 :

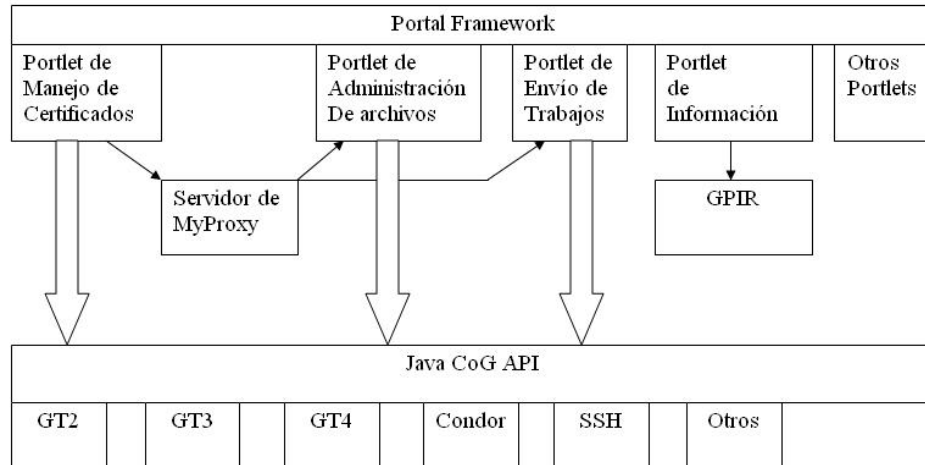


Figura 3.6: Open Grid Computing Environment

3.2.3. Portlet de Manejo de Certificados

Este portlet se encarga del servicio de autenticación de usuarios en un grid computacional. Un usuario dentro de un portal grid debe autenticarse ante un grid computacional mediante su certificado proxy. Terminada la autenticación, el usuario puede pedir al portal grid acceder a los recursos del grid en nombre del mismo usuario.

MyProxy

Un sistema que realiza parte fundamental de la autenticación de los usuarios de un grid computacional (ya sea que utilicen una interfaz web o no) es MyProxy. MyProxy es un sistema que administra certificados para los grids computacionales. MyProxy delega el certificado proxy de un usuario de grid a un Portal Grid, el portal a su vez utiliza este certificado para acceder a los recursos del mismo (grid computacional) en nombre del usuario. Una de las ventajas de MyProxy es que permite recuperar un certificado proxy independientemente de lugar donde se encuentre el usuario y en el momento que lo necesite gracias a su depósito (*repository*) de certificados. Otra característica importante de MyProxy es que al utilizarlo, se puede renovar los certificados proxy de un usuario automáticamente (sin la intervención del usuario) y evitar que trabajos que llevan largo tiempo ejecutándose sean interrumpidos.

Considerando que el certificado del usuario se encuentra almacenado en el servidor de

Myproxy y la creación de dicho certificado se realizó mediante el comando `myproxy-admin-adduser`, se muestra el proceso de autenticación utilizando Myproxy en la siguiente figura:

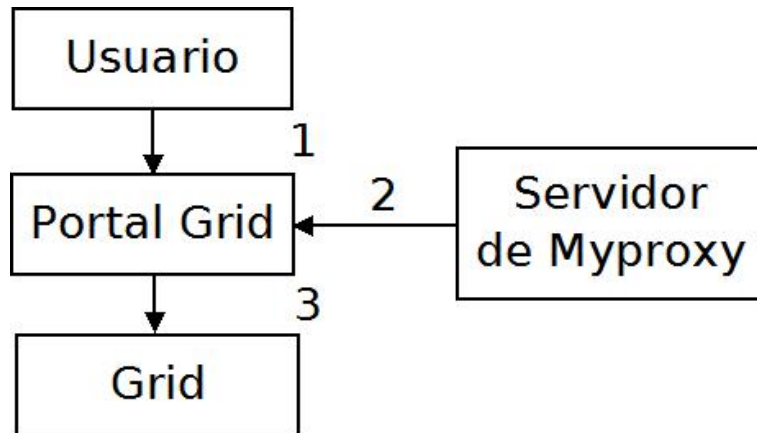


Figura 3.7: Servidor de Certificados

1. Se accede al portal Grid.
2. El portal grid solicitará la información necesaria para obtener el certificado proxy. En el caso del portal Grid OGCE esto se realiza mediante el portlet de MyProxy. En este paso el portal grid ejecutará el comando `myproxy_get_delegation` para recuperar el certificado proxy del servidor MyProxy utilizando el nombre de usuario y contraseña.
3. Una vez que el portal grid obtiene el certificado proxy del usuario, el portal accederá a los recursos del grid computacional con el certificado proxy a su nombre.

3.2.4. Portlet de Administración de Archivos

El Portlet de administración de archivos permite al usuario de un portal grid:

- Listar archivos en recursos remotos
- Cargar archivos a recursos remotos
- Transferir archivos entre recursos remotos

Este portlet puede mostrar los directorios uno al lado de otro de diferentes recursos de grid, esto permite al usuario listar los archivos de los diferentes recursos para su transferencia.

Específicamente este portlet proporciona funciones básicas del cliente GridFTP en una interfaz amigable. Las versiones de Globus toolkit que soporta este portlet son la 2.4, 3.2.1, 4.0.1 y 5.0 Este portlet realiza la administración de archivos directamente utilizando Java CoG.

3.2.5. Portlet de Envío de Jobs

Este portlet permite a un usuario enviar sus trabajos a un grid computacional utilizando el protocolo GRAM de Globus. Asimismo, en este portlet se pueden definir los parámetros para la ejecución de un trabajo, enviar a ejecución a un trabajo y ver el estado en que un trabajo se encuentra. Este portlet soporta las versiones de Globus 2.4, 3.2.1, 4 y 5.0. Para el envío de trabajos este portlet se comunica directamente con el middleware de grid utilizando el API de Java CoG.