

Capítulo V

DISEÑO E IMPLEMENTACION DEL SISTEMA DE RECUPERACION DE INFORMACION CON JINI

En este capítulo se analiza a detalle la forma en la cual se llevo a cabo el desarrollo del sistema para la recuperación de información visual, en especial de imágenes y documento de texto utilizando la tecnología Jini en su contexto cliente/servidor. Se explicaran las herramientas que se ocupan para lograr los objetivos planteados para el estudio de esta tecnología. Ver sección de Apéndice C y D para el análisis y diseño del sistema.

A continuación se explica la forma en que se dividen los módulos que proporcionaran las características esenciales en la recuperación de las imágenes y documentos de texto:

5.1 Diseño

5.1.1Módulo de Servidor:

En este módulo se desarrolla la interfaz del servidor, el cual nos proporciona los servicios requeridos para las imágenes y los textos. De esta manera, cumpliendo con las especificaciones de Jini, se definen los componentes necesarios para que pueda ofrecer sus servicios, esto es, definir sus especificaciones. Para esto se pondrá en marcha el gestor de seguridad logrando tener los permisos de transferencia con la ayuda del RMI, paso seguido, se da de alta el servidor para identificarse con el *Lookup*. Como se trató en el capítulo I, para que un servidor pueda ser identificado por la red Jini es necesario que este se de de alta con el *Lookup* y esto lo logrará mediante la llamada a los protocolos *Join* y *Discovery*.

Estos pasos son los que se llevan a cabo para poder levantar los servicios de nuestro servidor. Por ultimo, y lo más importante para el usuario, será definir los servicios que este servidor proporcionará.

Para nuestro diseño, se define dos métodos, un método que contiene una serie de imágenes que podrán ser compartidas al usuario y un método que contiene una búsqueda de archivos en el servidor, estos métodos recibirán información sobre que imagen o documentos deberán de proporcionarle al usuario. Cabe mencionar que en este sistema no se implementará ningún algoritmo complejo para reconocimiento de patrones de imagen en su recuperación.

5.1.2 Módulo de Servicio

Este módulo solo contendrá el método que proporcionara las imágenes al usuario y el método de recuperación de documentos de texto. Como se mencionó en el módulo anterior se describirá las características de estos métodos en el servidor, en este módulo solo se pondrá el nombre del método que nos proporcionara el servicio, esto es, el servicio *Lookup* lo encontrará y nos dirá en que servidor se encuentra, cuando nosotros pedimos el servicio, *Lookup* nos dará una copia de ese servicio mediante el servicio *proxy*.

5.1.3 Módulo de Cliente

Este módulo va a contener la interfaz para el usuario. Aquí el usuario podrá interactuar con los diferentes servicios de información visual, los cuales estarán disponibles en la red federada de Jini .

Al igual que la interfaz del servidor, se tienen que definir diferentes aspectos para la interfase del cliente, se define el gestor de seguridad para poder acceder los métodos

remotos, paso seguido será necesario preguntar al servicio *Lookup* sobre los servicios que existe de imágenes. Esto es, mandar una clave sobre el servicio que se necesita, la cual, el servidor que la contenga la recibirá y contestará con el servicio.

Una vez definidos los servicios necesarios, se podrá acceder directamente al servicio ofrecido, este se descargara en el servidor local, y será manejado como tal.

Esta interfase recibirá un objeto que contendrá la imagen o documentos de texto requerido por el usuario y se le mostrara en su interfase. Para poder lograr todo esto se desarrollara una interfaz grafica, en la cual el usuario podrá definir el servicio y la imagen o documento requerido por el, recuperarla de manera transparente, ofrecérsela visiblemente y guardarla en su servidor local.

Esta interfase obtendrá los tiempos de ejecución del sistema, los cuales serán útiles en los capítulos posteriores para obtener el desempeño de cada tecnología y poder compararlas.

A continuación se muestra figura 5.1 que representa la interacción de cada uno de estos módulos, formando en si, el sistema de recuperación de imágenes con Jini:

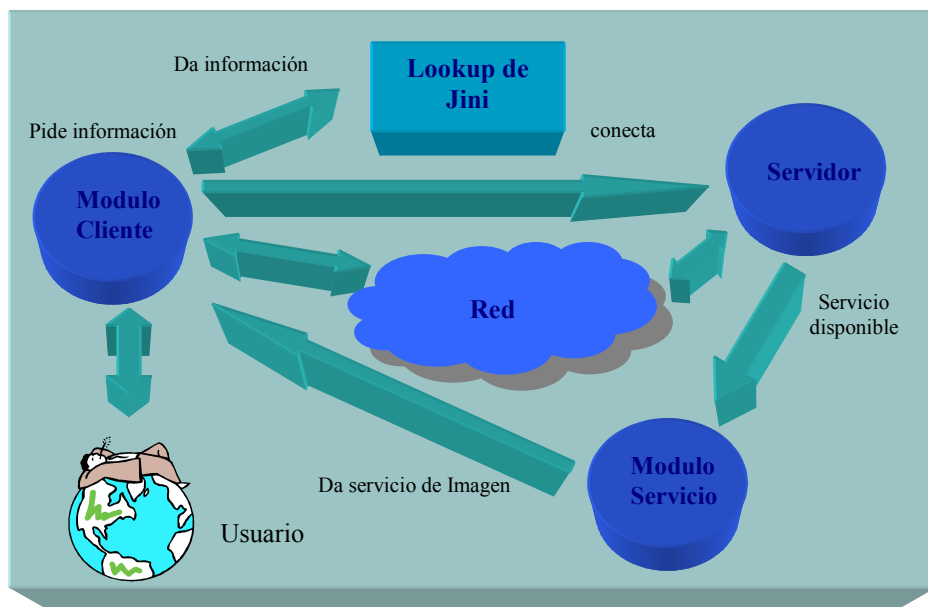


Figura 5.1 Diagrama de bloques del sistema de recuperación de imágenes con Jini

Se presentara a continuación detalladamente la forma de cómo se implemento el sistema de recuperación de imágenes y documentos con Jini.

5.2 Implementación

Como se describió anteriormente, se necesitaran de tres módulos, los cuales en conjunto lograrán para el sistema los objetivos planteados.

Existen algunas sugerencias por parte de la comunidad de Jini para el desarrollo de implementaciones con esta tecnología. Una de ellas, es crear diferentes directorios de trabajo para los módulos del sistema. Para aplicaciones sobre plataformas Unix, sugieren construir cuatro directorios dentro del directorio principal donde se encuentra instalado el kit de Jini, en este caso en el directorio “files”, uno de ellos es “service” el cual contendrá las clases del servidor utilizadas en cada implementación, el segundo directorio es “client” que contiene las clases de las interfaces de los clientes, el tercero es “service-dl” que contiene las clases de los servicios que ofrece cada servidor, y el ultimo “client-dl”.

Y para plataformas Windows se recomienda poner cada uno de estos directorios en raíz.

Cabe mencionar que estos nombres son sugerencias, y que pueden tomar el nombre que más les convenga según sea la aplicación que este desarrollando.

Antes de comenzar con la implementación primero se describirán la forma en que se llevo a cabo la instalación de Jini para poder lograr lo antes mencionado.

5.2.1 Herramientas básicas para el desarrollo Jini

A continuación se presentan las herramientas necesarias para lograr el modulo de aplicación de Jini en la recuperación personalizada de información visual:

5.2.1.1 Jini Starter Kit v1.1

Esta tecnología fue desarrollada por Sun Microsystems y al igual que el compilador de Java JDK, este paquete es puesto a disposición del público para desarrollar código Java orientado a Jini, y es otorgado de manera gratuita en la página oficial de Sun Microsystems [jiniProduct99].

Para este proyecto en especial se bajo el Kit de Jini con la versión 1.1. Este paquete viene en un *archivo.tar* o *.zip*, tanto para Solaris como para PC.

5.2.1.2 Instalación de Jini Starter Kit v1.1

El archivo *jini1.1-src.tar* se instaló en la cuenta específica de la Universidad de las Américas Puebla, bajo el sistema operativo Solaris, creando una pequeña red federada en la cual se lleva a cabo las pruebas de recuperación de información visual.

Cabe mencionar que antes de instalar el Kit de Jini, es necesario crear un directorio llamado “files”, dentro del cual se deberá instalar el Kit de Jini, esto es por especificaciones de la compañía creadora de esta tecnología.

Para la adecuada instalación del Kit de Jini se configuraron los CLASSPATH del paquete dentro del archivo del sistema *.cshrc*, agregando las rutas de las clases requeridas por el Kit de Jini para su buen funcionamiento. Los detalles sobre la adecuada configuración de los archivos del sistema, para la instalación correcta de Jini, serán definidos dentro del Apéndice B.

Así también, se configuraron los *path* de Java *jdk1.2* otorgándole el apoyo requerido a Jini, los cuales de igual manera serán explicados a mayor detalle dentro del apéndice B.

Como nota importante, para el buen funcionamiento de Jini, es necesario estar trabajando bajo las versiones de Java jdk1.2 en adelante. Esto por las diferentes tecnologías y comandos que presentan estas versiones, tales como java -cp, -Djava, soporte de RMI, etc, las cuales son de gran apoyo dentro de la tecnología de Jini.

También se instaló Jini en una pequeña red de PC's, para el estudio de la tecnología dentro de este ambiente de trabajo, para ello se hizo uso del archivo jini1.1-src.zip, el cual fue descomprimido e instalado dentro de esta pequeña red, otorgándole a esta un federación de red para el estudio de resultados en recuperación de información visual, especialmente de imágenes y documentos de texto.

Al igual que en la red del centro de cómputo del *campus* universitario, en esta red, se configuraron los archivos necesarios del sistema para la definición de *paths* y CLASSPATH necesarios por el Kit de Jini para su buen funcionamiento e instalación exitosa. En la sección del apéndice B, se describirá a detalle la forma y los archivos del sistema que se configuraron para la instalación adecuada del sistema.

Al instalar Jini Starter Kit v1.1, genera una serie de directorios, los cuales contienen información importante del kit, tanto como las clase principales, como documentación técnica. Todo esto dentro del directorio especificado "files" por la empresa creadora del Kit de Jini. A continuación se muestra en la figura 3.5 la ruta de directorios que contendrá el Kit de Jini , los cuales serán descritos con mayor detalle en la parte del apéndice B:

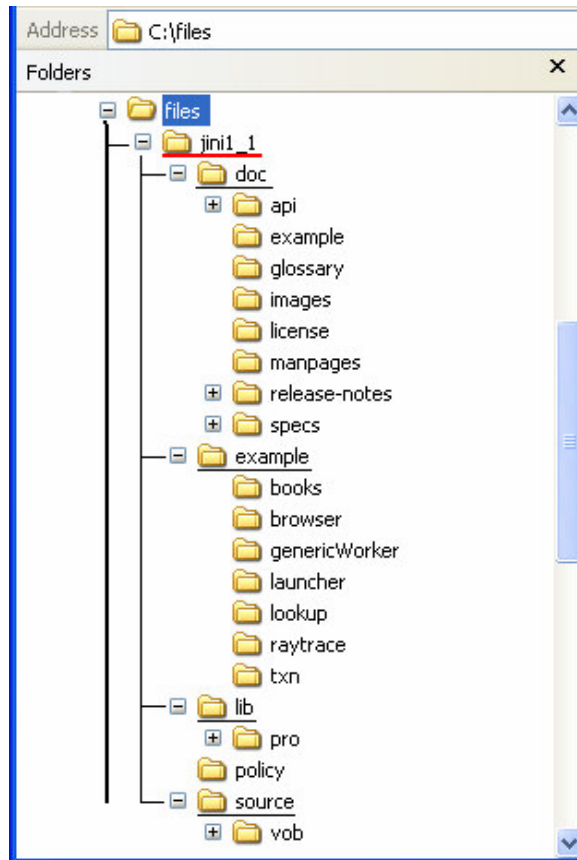


Figura 5.2 Directorio del contenido de Jini1.1

En la figura 3.5 se muestran las carpetas principales, las cuales son las subrayadas con negro, donde la carpeta subrayada con rojo es la carpeta principal del Kit de Jini1.1 que se encuentra dentro del directorio “files”.

Una vez instalado el Kit de Jini se procederá a explicar la forma en la cual se implementaron cada uno de los módulos que en conjunto forman el sistema de recuperación de información visual, especialmente en objetivos de imágenes y documentos.

5.2.2 Módulo de Servidor

Aquí se define un módulo llamado *Servicio*, este modulo es el Servidor del servicio y como se comento anteriormente se encuentra dentro del directorio “service”.

Este modulo se definió de la siguiente manera:

Como principio se tiene que hacer referencia al gestor de seguridad para poder tener la funcionalidad de RMI. Este método es sencillo, solo se requiere hacer referencia al método: *System.setSecurityManager* y crearle, dándole el manejador de seguridad de RMI, esto se llevo acabo de la siguiente manera:

```
System.setSecurityManager (new RMISecurityManager ());
```

Esto es todo lo que se necesita para tener los permisos de RMI y poder transferir métodos remotamente.

Paso seguido se definió los métodos, los cuales serán el servicio que nuestro servidor ofrecerá en la red Jini, estos métodos son:

```
public Image obtenImagen(int i);
```

```
public File obtenDocument(String s);
```

Estos método son públicos y regresan un objeto de tipo *Image* y *File* que contiene la imagen y documento requerido por el usuario, así como también reciben un valor entero y tipo *String* los cuales le indicaran que tipo de imagen y documento deberán regresar. Estos métodos contienen un arreglo de archivos de imágenes y archivos de texto, los cuales estarán ofrecidos en la red.

Uno de los aspectos importantes en la definición del servidor, es la definición de sus atributos, estos atributos servirán para que el servicio *Lookup* con las especificaciones del cliente, busque y compare con los atributos dados requeridos, de esta manera se contacta el cliente con el servidor que ofrece los servicios requeridos. Para lograr esto, se define dentro de un arreglo de objetos las características del servidor de imagen, en este caso la más

importante es llamada “*Servicio*” y es la que hará que nuestro cliente realice peticiones al servicio que esta ofrece. La forma en que se llevo acabo es con el servicio “*Entry*” que se encuentra en el paquete *net.jini.core.entry*. *

Claro que nada de esto seria posible si no se registra el Servidor, para esto se necesitarán los atributos definidos anteriormente. La manera como se realiza, es con la clase *JoinManager* a la cual le pasamos los atributos antes mencionados, instancia del módulo del servidor, y el gestor *Leasing* (para rentar el servicio). Esto es, se registra el servidor con sus atributos, y el arrendamiento de su servicio. Una vez establecida esta configuración se busca el servicio *Lookup*, enviándole el nombre del servidor en el cual se encuentra el servidor.

El siguiente paso es interactuar con el servicio Jini, esto se logra colocando el método *ServiceRegistrar*, interactuándolo con *ServiceMatches*.

Con esto se consigue definir el Servidor junto con los servicios que se ofrecen y darlos de alta en el servicio Jini.

En el Apéndice B se explicarán los pasos necesarios para tener compilado el modulo del servidor y tenerlo dentro de este directorio de trabajo. Como adelanto se informa que Jini maneja las clases y demás archivos necesarios dentro archivos JAR para un mejor manejo en la transacción de servicios. Y por lo cual requiere de compilación y puesta en marcha de la clase con instrucciones especiales que se detallaran en el siguiente capitulo.

5.2.3 Módulo del Servicio.

En este módulo se definen los servicios que serán disponibles dentro de la red, en particular se definió solo uno, y que ofrece el servicio de imágenes. Este módulo está definido de la siguiente manera:

```

public interface ServiciodeImagen extends Remote
{
    public Image obtenImagen(int i) throws RemoteException;
    public Image obtenDocument(String i) throws RemoteException;
}

```

Como se puede observar, es un método que se usara de manera remota, obteniendo un valor que le indicara el archivo de imagen o documento que tendrá que regresar.

Este módulo es un archivo que al igual que el Servidor, estará almacenado dentro de una carpeta especial llamada “service-dl”, y como se comento anteriormente requerirá de comandos especiales explicados a detalle más adelante

5.2.4 Módulo de Cliente.

Aquí se lleva acabo el módulo del cliente, en este proyecto llevará el nombre de *ImagenCliente*. Este módulo esta dividido en una interfaz grafica de usuario, el módulo de registro y petición de servicios a Jini.

El módulo de componentes necesarios para interactuar con los servicios de la red Jini esta definido de la siguiente manera:

Al igual que el Servidor, el cual lleva el nombre de *ImagenServicio*, requiere la definición del gestor de seguridad para poder cargar las clases remotas, y se define de la siguiente manera:

```

System.setSecurityManager (new RMISecurityManager ());

```

Paso seguido es buscar el servicio *Lookup* del entorno, esto se logra con la obtención de la definición del *Lookup service*, como se muestra a continuación:

```

lookup = new LookupLocator ("jini://sp205236");

```

Una vez interactuados con el *Lookup* se piden los servicios que necesitamos conforme a los atributos que definimos como servicios requeridos, estos, como se comentaron en el servidor, serán comparados por el servicio *Lookup* con el servidor y si coinciden se enlazaran y se proporcionara el servicio que ese servidor ofrece.

Para definir los atributos que necesitamos en nuestro cliente se definió lo siguiente:

```
Atributos = new Entry[1];
```

```
Atributos[0]=new Name("ImagenServicio");
```

Como se puede observar, es el atributo que nos enlazará con el servidor *ImagenServicio* y con el cual obtendremos el servicio requerido, llevándolo acabo de la siguiente manera:

```
plantilla = new ServiceTemplate(null,null,Atributos);
```

la variable *plantilla* es una instancia de la clase *ServiceTemplate* y nos sirve para buscar el servicio deseado en el *Lookup*. Como se puede observar se le pasan los atributos que requerimos para el servicio.

Una vez obtenido el servicio que requerimos, este es guardado dentro de un objeto *imagen*, y entonces ponemos en acción el modulo de interfaz grafica de usuario. La imagen es procesada en escala de su tamaño para el tamaño del área de trabajo como se muestra a continuación:

```
imagen=myServiciodeImagen.obtenImagen(espacio);
```

```
mt.addImage(imagen,0);
```

```
mt.waitForID(0);
```

```
int w=imagen.getWidth(this);
```

```
int h=imagen.getHeight(this);
```

```
g.drawImage(imagen,110,40,w/4,h/4,nc);
```

Este módulo será almacenado dentro del directorio “client”, y al igual que los demás módulos este requiere de instrucciones especiales para compilarse y poder llegar a este directorio de trabajo, las cuales serán descritas a mayor detalle en el Apéndice B

A continuación se presenta la descripción de la interfaz grafica de usuario, con la descripción de los diferentes componentes que esta presenta.

5.2.4.1 Interfaz del Cliente

En la siguiente figura se presenta la Interfaz Grafica de Usuario del modulo del cliente, en la cual podrá interactuar el usuario para obtener los servicios que requiera. En este caso solo será para la recuperación de Imágenes y documentos:

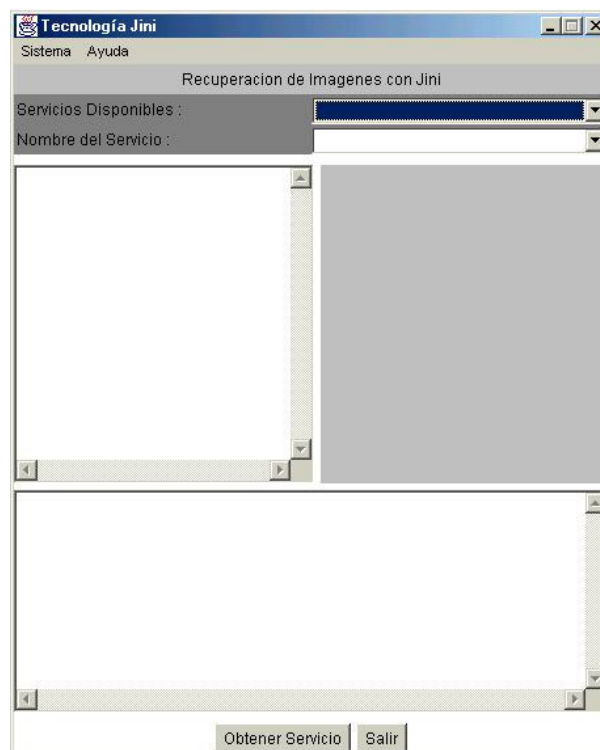


Figura 5.3 Interfaz del Cliente para la Recuperación de Imágenes con Jini

A continuación se describen los componentes de la interfaz gráfica de usuario:

El sistema cuenta con una barra de menú, la cual funciona de la siguiente manera:

En la etiqueta de Sistema aparecen los campos “*Acerca del Sistema*”, “*Limpiar Area*” y “*Salir*”. Al seleccionar el campo *Acerca del Sistema*, obtendrá la información de fabricación del sistema tal como el nombre del proyecto, e información personal del desarrollador del sistema.

En el campo de *Limpiar Area*, se lleva a cabo la limpieza del área de trabajo, limpiando con esto, todos los campos utilizados por el sistema en la recuperación de información, y dejando lista la interfaz para siguientes resultados de búsquedas. La opción *Salir* da por concluido la ejecución del sistema.

En la etiqueta de *Ayuda* se encuentra el campo “*Funcionamiento del Sistema*”, aquí se encuentra una guía del uso adecuado del sistema para la recuperación de información.

A continuación de la barra de herramientas, en la parte superior, se encuentra el área de *Servicios Disponibles*, aquí el usuario escogerá el servicio que deseara obtener. Entre los servicios disponibles se encuentran “*Imágenes*” y “*Documentos*”, Una vez seleccionada esta opción, en la parte inferior a ella se encuentra *Tipo de Servicio*, aquí el usuario seleccionara el tipo de servicio, por ejemplo, si el usuario escogió el servicio de Imágenes, entonces el tipo de servicio para ello es “*Delfines*”, “*Escudo UDLA*”, etc., que el usuario seleccionara para traerlo a su servidor.

En la parte central se encuentra dos áreas, una es el área de información útil al usuario y la otra es el área de Imágenes, en el área de texto se le presenta al usuario información de éxito en la búsqueda e información de los datos obtenidos, en el área de imágenes se presenta de manera grafica la imagen que el usuario selecciono y la cual fue obtenida de un servidor remoto.

En la parte inferior se encuentra el área de resultados, aquí aparecerán los mensajes de obtención de servicios, esto es, un mensaje si todo salio bien en la transferencia de la

imagen, un tiempo de salida y el tiempo de retraso al obtenerla, y el nombre del servidor desde donde se recuperó la información.

Para finalizar en la parte inferior se muestran dos botones, uno es llamado “Obtener Servicio”, este botón tiene la función de conectarse con *Lookup* y pedirle el servicio que el usuario seleccionó, una vez obtenidos los servicios, estos son presentados al usuario junto con los resultados de la recuperación. El botón Salir, tiene como función la de cerrar la sesión y salirse del sistema del Cliente.

A continuación se muestra en la figura 5.3 un ejemplo de un servicio recuperado al usuario, en éste ejemplo el usuario selecciono el servicio de “Imágenes”, paso seguido definió el tipo de imagen que deseaba obtener del servicio de Imagen, en este caso fue la opción “Delfín”.

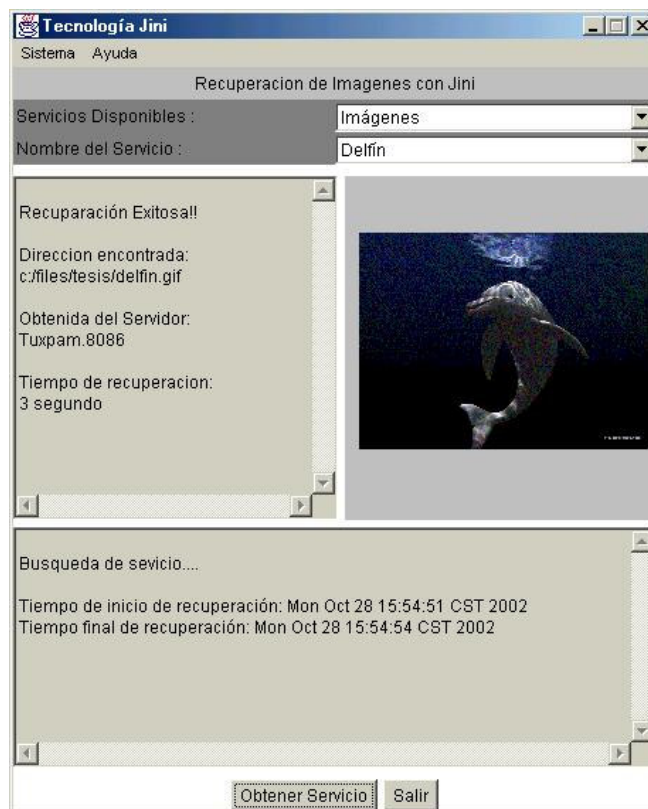


Figura 5.4 Ejemplo de recuperación del servicio de imagen

5.3 Conclusiones

Esta implementación para la recuperación de imágenes y documentos de texto es solo una herramienta y no una aplicación como producto, ya que solo nos ayudará a obtener datos que serán de gran utilidad en el estudio de las dos tecnologías.

Jini fue puesto en marcha exitosamente, aunque con un tiempo de retraso a lo previsto, ya que sus instrucciones para compilar los módulos descritos anteriormente son muy extensos y algo complicadas, debido a que es necesario al momento de compilar, comprimir las clases y los archivos utilizados en el servicio dentro de archivos JAR. Esto es uno de los inconvenientes de Jini, sin embargo una vez puesto en marcha su desempeño es favorable. Sin embargo existen diversos problemas de la puesta en marcha de la red federada Jini dentro de la red de la Universidad de las Américas, esto se debe a que Jini esta fabricado para trabajar sobre redes con un rendimiento optimo, y no sobre redes que requieran de bajar y subir a cada momento servidores para depurar la red.