

Capítulo 5

Resultados

Capítulo 5 Resultados

5.1 Resultados

Los resultados obtenidos fueron la implementación e incorporación del nuevo servidor “Conexion” bajo la Arquitectura Galaxy. Aunque el servidor se agregó, no se generaron las reglas para poder interactuar con los otros servidores.

Las herramientas desarrolladas para la transferencia de texto, imagen y video fueron probadas mediante la interacción con el servidor “Conexion”, aunque en el caso de imagen, texto y video la interacción fue simulada se realizó con éxito la comunicación.

5.1 Problemas presentados y soluciones ofrecidas

Uno de los principales problemas que se presentaron fue la versión de Java Development Kit. Debido a que la versión que incluye Linux Red Hat 7.2 es la 1.0.6, y era necesario utilizar la versión 1.3 o superior, esto porque el API que se utilizó para la captura de audio y video es soportado solo por esta versión. Así es que se actualizó la versión.

Otro de los problemas encontrados fue las restricciones de seguridad de los applets, debido a que para la captura de audio se realizó mediante un applet, y para poder leer o escribir un archivo desde un applet es necesario cambiar el archivo `java.policy` que se encuentra dentro del siguiente path:

`/jdk/lib/security`

Dentro del archivo java.policy se tendrá que agregar las siguientes líneas:

```
grant {  
    permission java.io.FilePermission "<<ALL FILES>>", "read, write";  
    permission javax.sound.sampled.AudioPermission "record";  
    permission java.util.PropertyPermission "user.dir", "read";  
};
```

Otro de los problemas encontrados fue que linux no acepta todo tipo de formatos para reproducir el video, así es que se tuvieron que realizar varias pruebas con los distintos formatos hasta que se encontró el formato con el cual fue posible reproducir el video.

Uno de los problemas que surgieron al iniciar la implementación fue contar con las herramientas necesarias como lo es el JMF (Java Media Framework) el cual se instaló en un principio en Windows, y no hubo ningún inconveniente. Después cuando se trato de instalar en linux hubo una serie de problemas debido a que no podía encontrar el path correcto del jdk1.4.1 es por ello que se tuvo que agregar la siguiente línea de comando dentro de los archivos jmfinit, jmfregistry, jmstudio:

```
exec /home/cucomm/j2sdk1.4.1/bin/java JMFRegistry
```

Se le debe agregar el comando exec para que ejecute el comando deseado, y después se especifica el path en donde se encuentra el jdk.

Por último otro de los principales problemas fue la falta de información que existe sobre la base del funcionamiento del CU Communicator, así es que se tuvo que realizar un análisis profundo de toda su infraestructura, y de la forma en como se comunicaban los servidores con el HUB.

5.2 Pruebas

Las pruebas de unidad y de integración fueron realizadas durante el desarrollo de cada módulo. Las pruebas de integración fueron basadas en hilos. Mediante esta prueba se probaron por separado las herramientas para la transferencia de texto, audio, video e imagen y conexión con el CU Communicator. Para la prueba del audio se utilizó la clase JavaSound y Capturaaudio y se probó desde 3 navegadores diferentes Explorer 5.0, Netscape, Hot Java. Aunque no se presentó ningún problema para la captura y reproducción del audio, el resultado no fue optimo en todos los navegadores (ver Tabla 5.1.).

Tabla 5.1 Tabla de prueba de implementación de audio

<i>Navegador</i>	<i>Tiempo de respuesta</i>	<i>Problemas y Soluciones</i>
Explorer	Optimo	ninguno
Netscape 6.2	Optimo	Se tuvieron que instalar los plugins para que pudiera soportar java
Hot Java	Malo	Aunque no se presentó ningún problema para la captura de video, el tiempo de respuesta con este navegador fue muy lento.

Es optimo o malo en base al tiempo de respuesta que se halla generado en los diferentes navegadores.

Para el video se tuvieron que realizar 2 pruebas diferentes: una para la reproducción, y la segunda para la captura y su reproducción. Esto debido a que no se contaba con una tarjeta de captura de video para que bajo linux pudiera detectar la webcam. Es por ello que una de las pruebas de captura y reproducción se realizaron bajo Windows. Primero se probó como una aplicación local para verificar que la codificación y transmisión de video pudiese lograrse correctamente. El problema que surgió fue que no reconocía el puerto en el que el servidor estaba recibiendo las peticiones para transferir el video. Esto fue debido a que cuando se terminaba de utilizar el puerto éste se dejaba activado y la siguiente vez que se intentara activarlo estaba ocupado. Esto se solucionó mediante la desactivación del puerto cada vez que se terminara de transmitir.

Una vez que se terminó de probar con una aplicación local se prosiguió a probar la transmisión de video vía WEB. No hubo ningún problema para transmitir el video, solamente se notó un desfase de 2 segundos en la visualización del video.

La siguiente prueba fue la transferencia de imagen, se probó con diversos formatos para revisar si no presentaba ningún error al visualizarlos, o el tiempo de respuesta. No surgió ningún error, y el tiempo de respuesta fue óptimo casi con la mayoría de los formatos, excepto con el formato .bmp, esto debido al tamaño.

La última prueba que se realizó fue la transferencia de texto. Desde la aplicación desarrollada se probó enviar texto y también el poder recibir y visualizarlo. En esta última prueba no hubo ningún problema.

5.3 Conclusiones

El CU Communicator es una aplicación recientemente desarrollada, su funcionamiento es de un alto grado de complejidad. Debido a que esta aplicación es nueva no existe mucha información acerca de su funcionamiento y esto dificultó mucho el poder llegar a las soluciones de los problemas que se presentaban, sin tomar en cuenta la complejidad de la aplicación misma.

En esta tesis se muestra un análisis específico del funcionamiento del CU Communicator y la forma en como se comunican los servidores con el Hub. También se describen las diferentes herramientas utilizadas para desarrollar las aplicaciones mediante las cuales se transfieren el texto, imagen, voz y video. Se describe también la forma para agregar un nuevo servidor al CU Communicator.

Los objetivos específicos planteados para esta tesis se cumplieron satisfactoriamente. Los objetivos fueron los siguientes:

- a) Desarrollar herramienta para la transferencia de texto.
- b) Desarrollar herramienta para la transferencia de imagen.
- c) Desarrollar herramienta para la transferencia de video.
- d) Desarrollar herramienta para la transferencia de audio.
- e) Modulo de cambios.

5.4 Perspectiva en el futuro de las herramientas desarrolladas

Las herramientas desarrolladas sientan la base para la transferencia de diversos tipos de datos. El objetivo de esta tesis fue el desarrollar las herramientas necesarias para poder

transferir el audio, video, texto, imagen tomando en cuenta la infraestructura del CU Communicator. Como se mencionó con anterioridad la interacción con el CU Communicator es mediante voz o texto. Uno de los trabajos que se puede desarrollar mas adelante es el agregar al CU Communicator la posibilidad de interactuar mediante video e imagen sobre la base de las herramientas desarrolladas, e integrarlas en aplicaciones.

Otro de las cosas sobre la que se podría trabajar es el acoplar estas herramientas al CU Communicator. Aunque en este trabajo de tesis se agregó un nuevo servidor al CU Communicator, solamente se generaron las reglas del Hub para que se pudiera probar el funcionamiento adecuado de las herramientas. Así es que uno de los trabajos a desarrollar podría ser el generar las reglas del Hub para poder interactuar con los otros servidores. Esto consistiría básicamente en realizar una serie de reglas para el servidor de audio, el parser y el sintetizador de voz.

Debido a que se pretenden desarrollar sistemas de enseñanza sería de gran importancia que se llevaran a cabo estos trabajos, esto ampliaría los posibles alcances de las aplicaciones desarrolladas.