

**TECNICA PARA SOPORTE DE LA DISPONIBILIDAD DE SERVICIOS EN UN
ENTORNO DE REDES SUPERPUESTAS P2P NO ESTRUCTURADAS.**

ANEXOS

**Jenny Bolaños Delgado
Héctor Gentil Ordoñez Delgado**

Universidad del Cauca
**Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo IDIS - Investigación y Desarrollo en Ingeniería de Software
Línea de Investigación en Tecnologías de Soporte
Popayán, Octubre de 2015**

**TECNICA PARA SOPORTE DE LA DISPONIBILIDAD DE SERVICIOS EN UN
ENTORNO DE REDES SUPERPUESTAS P2P NO ESTRUCTURADAS.**



**Jenny Bolaños Delgado
Héctor Gentil Ordoñez Delgado**

**Director
Esp. Pablo Augusto Magé Imbachí**

Universidad del Cauca
**Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Sistemas
Grupo IDIS - Investigación y Desarrollo en Ingeniería de Software
Línea de Investigación en Tecnologías de Soporte
Popayán, Octubre de 2015**

I.TABLA DE CONTENIDO

ANEXO A	1
ALCANCE DEL PROYECTO	1
1. Objetivos del proyecto	2
1.1. Objetivo general.....	2
1.2. Objetivos específicos	2
2. Justificación del proyecto.....	2
3. Metodología de trabajo	2
4. Resumen de entregables.....	5
4.1. Relacionados con la gestión del trabajo de grado.....	5
4.2. Relacionados con el Prototipo Software.....	5
5. Criterios de éxito del Proyecto	5
ANEXO B	6
DOCUMENTO DE ESPECIFICACION DE REQUERIMIENTOS	6
1. Introducción	7
1.1. Propósito.....	7
1.2. Alcance	7
1.3. Definiciones, siglas y abreviaciones	7
2. Descripción Global.....	7
2.1. Perspectiva del producto	7
2.1.1. Interfaces con el sistema.....	8
2.1.2. Interfaces con el usuario	8
2.1.3. Interfaces con el hardware	8
2.1.4. Interfaces con el software.....	8
2.1.5. Interfaces de comunicación.....	8
2.1.6. Restricciones de memoria	8
2.1.7. Requerimientos de adaptación.....	9
2.2. Funciones del producto	9
2.3. Características del usuario	9

3.	Requerimientos específicos	9
3.1.	Requerimientos funcionales	10
3.2.	Requerimientos no funcionales	15
4.	Restricciones de diseño	15
ANEXO C		16
TECNOLOGIAS UTILIZADAS PARA EL DESARROLLO DEL PROTOTIPO SOFTWARE		16
1.	Tecnología JXTA	17
1.1.	Servicios JXTA	17
1.2.	Anuncios	17
1.3.	Protocolos	18
1.3.1.	Core Specification Protocols	18
1.3.2.	Standard Service Protocols	19
2.	VLC	20
2.1.	LibVLC	20
2.2.	VLCj	20
3.	FTP	21
3.1.	Servidor FTP	21
3.2.	Ciente FTP	22
4.	Sockets	22
5.	Hilos	22
ANEXO D		23
HISTORIAS DE USUARIO		23
1.	Historias de usuario	24
ANEXO E		29
MANUAL TECNICO Y DE INSTALACION		29
1.	Requerimientos software del prototipo	30
1.1.	Instalación de Java	30
1.2.	Lista de excepciones Java	30
1.3.	Instalación de VLC	30
1.4.	Crear servidor FTP	31

2. Instalación de la aplicación	31
3. Ejecución del prototipo.....	32
ANEXO F.....	33
MANUAL DE USUARIO.....	33
1. Ejecutar la aplicación	34
1.1. Funcionamiento de la interfaz del servidor	34
1.2. Funcionamiento de la interfaz del cliente	36
2. Salir de la aplicación	40
ANEXO G	41
PRUEBAS DEL PROTOTIPO SOFTWARE	41
1.1 Especificación de los equipos utilizados en las pruebas	42
1.2 Pruebas manuales	44
ANEXO H	50
RESUMEN BIBLIOGRAFIA ESTUDIADA PARA LA SELECCIÓN DE LA TECNICA DE DISPONIBILIDAD	50
1. Replicación	51
2. Rotación.....	52
3. Tolerancia a fallos.....	53
BIBLIOGRAFIA.....	55

II. LISTA DE FIGURAS

Figura 1. Protocolos JXTA.....	19
Figura 2. Estructura de directorios.....	32
Figura 3. Ejecutar prototipo	32
Figura 4. Interfaz del prototipo.....	34
Figura 5. Campos requeridos para publicar cada servicio	35
Figura 6. Mensaje de servicio publicado.	36
Figura 7. Parte superior de la interfaz del cliente	37
Figura 8. Parte inferior de la interfaz del cliente	37
Figura 9. Imagen consumo de servicio en el cliente.....	38
Figura 10. Cargar archivo de prueba para el servicio de cálculos computacionales.....	39
Figura 11. Resultado de consumir el servicio de cálculos computacionales.....	40

III. LISTA DE TABLAS

Tabla 1. R1 - Publicar servicios.....	10
Tabla 2. R2 - Buscar servicios.....	10
Tabla 3. R3 – Monitorear condición de los servicios.	11
Tabla 4. R4 – Recuperar los servicios.....	11
Tabla 5. R5 – Crear mensaje descriptor para el servicio de media streaming.	12
Tabla 6. R6 – Crear mensaje descriptor para el servicio de contenidos.	12
Tabla 7. R7 – Crear mensaje descriptor para el servicio de cálculos computacionales.....	13
Tabla 8. R8 – Obtener mensaje descriptor de los servicios.	13
Tabla 9. R9 – Consumir servicios.....	13
Tabla 10. R10 – Iniciar el nodo servidor del servicio de media streaming.	13
Tabla 11. R11 – Iniciar el nodo servidor del servicio de contenidos.....	14
Tabla 12. R12 – Iniciar el nodo servidor del servicio de cálculos computacionales.	14
Tabla 13. R13 – Detener el nodo servidor.....	14
Tabla 14. R14 – Iniciar el nodo cliente.	15
Tabla 15. R15 – Detener el nodo cliente.	15
Tabla 16. HU - Solicitar un servicio de contenidos.....	24
Tabla 17. HU - Solicitar un servicio de media streaming.....	25
Tabla 18. HU - Solicitar un servicio de cálculos computacionales.	26
Tabla 19. HU – Publicar un servicio de contenidos.	27
Tabla 20. HU – Publicar un servicio de media streaming.....	27
Tabla 21. HU – Publicar un servicio de cálculos computacionales.	28
Tabla 22. HU – Replicación de servicios.....	28
Tabla 23. Equipo 1.....	42
Tabla 24. Equipo 2.....	42
Tabla 25. Equipo 3.....	42
Tabla 26. Equipo 4.....	42
Tabla 27. Equipo 5.....	42
Tabla 28. Equipo 6.....	43
Tabla 29. Equipo 7.....	43
Tabla 30. Equipo 8.....	43
Tabla 31. Equipo 9.....	43
Tabla 32. Equipo 10.....	43
Tabla 33. Equipo 11.....	43
Tabla 34. Equipo 12.....	44
Tabla 35. Equipo 13.....	44
Tabla 36. Equipo 14.....	44
Tabla 37. Equipo 15.....	44
Tabla 38. Definición de escenarios de pruebas.....	48
Tabla 39. Formato de registro de otros resultados.....	49

ANEXO A

ALCANCE DEL PROYECTO

1. Objetivos del proyecto

1.1. Objetivo general

Diseñar una técnica que apoye la disponibilidad de servicios en una red superpuesta P2P no estructurada la cual permita gestionar algunos tipos de servicios.

1.2. Objetivos específicos

- Seleccionar las principales técnicas utilizadas para apoyar la disponibilidad de servicios, con el fin de caracterizar una de estas para ser aplicada en la técnica propuesta.
- Adecuar una arquitectura con la técnica seleccionada para redes superpuestas P2P no estructuradas capaz de gestionar los servicios seleccionados.
- Diseñar e implementar un prototipo software para gestionar los servicios seleccionados mediante una red superpuesta P2P.
- Evaluar el prototipo software con respecto a la disponibilidad de los servicios en un entorno de redes superpuestas P2P.

2. Justificación del proyecto

Debido a la gran acogida y desarrollo que han venido teniendo las redes superpuestas P2P; además de todas las investigaciones y desarrollos realizados que se han llevado a cabo de forma independiente de acuerdo al servicio prestado. Este proyecto de investigación pretende aportar en el problema de la disponibilidad de servicios en redes superpuestas P2P no estructuradas, con el desarrollo de una técnica para soporte de disponibilidad de servicios en un entorno de redes superpuestas P2P no estructuradas. Entre los aportes se encuentran:

- Documento de caracterización con la técnica de disponibilidad seleccionada.
- Documento de diseño de la arquitectura con la técnica seleccionada.
- Prototipo software de la aplicación adecuada para una red superpuesta P2P no estructurada.
- Documento de resultados de la evaluación.

3. Metodología de trabajo

Para el desarrollo de este trabajo de grado se aplicó la metodología de investigación apoyada sobre los métodos de investigación de ingeniería Finkelstein

[22] y para la implementación del software se siguió la metodología de desarrollo denominada Programación Extrema (XP) adecuándola a las necesidades de nuestro proyecto.

La metodología de desarrollo del trabajo de grado consta de las siguientes etapas:

- **Etapas 1. Investigación y caracterización.**

Exploración bibliográfica sobre técnicas de disponibilidad: esta actividad consiste en realizar una investigación bibliográfica a través de libros, artículos, tesis y demás fuentes de información acerca de las técnicas utilizadas para apoyar la disponibilidad en redes superpuestas P2P no estructuradas.

Selección y Caracterización de una de las técnicas: esta actividad consiste en identificar las técnicas utilizadas para mejorar la disponibilidad dentro de las redes superpuestas P2P no estructuradas y caracterizar la más adecuada para apoyar la disponibilidad de servicios en este tipo de redes.

Construcción del marco teórico: esta actividad consiste en realizar la descripción de los conceptos relacionados con las redes superpuestas P2P no estructuradas.

- **Etapas 2. Adaptación de la arquitectura.**

Búsqueda de las arquitecturas y selección de una de ellas: esta actividad consiste en realizar la búsqueda y estudio de las arquitecturas ya diseñadas para redes superpuestas P2P no estructuradas y seleccionar la más apropiada para las necesidades del proyecto.

Diseño de la técnica para apoyar la disponibilidad de servicios en redes P2P: esta actividad consiste en realizar el diseño de una técnica para apoyar la disponibilidad de servicios en redes superpuestas P2P no estructuradas.

Adecuación de la técnica a la arquitectura seleccionada: esta actividad consiste en hacer la adecuación de la técnica diseñada con la arquitectura seleccionada sobre una red superpuesta P2P no estructurada.

- **Etapas 3. Diseño e implementación del prototipo.**

- ✓ **Fase de exploración:** en esta fase se llevarán a cabo las siguientes actividades:

El planteamiento de historias de usuario y la familiarización con las tecnologías: en esta actividad, con cada historia de usuario se describirá parte de las funcionalidades que el prototipo tendrá, mientras que al mismo tiempo el equipo se debe familiarizar con las tecnologías a utilizar.

Diseño de prototipo no funcional: en esta actividad se realizara el diseño no funcional del prototipo con el fin de explorar algunos aspectos de la arquitectura a ser implementada.

- ✓ **Fase de planificación de la entrega:** en esta fase se llevara a cabo la siguiente actividad:

Priorización de las historias de usuario: en esta actividad se fija la prioridad de cada una de las historias de usuario, los desarrolladores estiman cuanto esfuerzo requiere cada una de ellas y se establece cual va a ser el contenido de la primera entrega.

- ✓ **Fase de iteraciones:** en esta fase se llevaran a cabo 4 iteraciones, cada una de ellas consta de diseño (realizar los modelos de diseño), implementación (programación de las funcionalidades) y pruebas (diseño y ejecución de pruebas).

- ✓ **Fase de producción:** en esta fase se realizara la siguiente actividad:
Despliegue del prototipo: en esta actividad se mantiene el sistema en ejecución y se realizan muchos más chequeos y pruebas.

- **Etapa 4. Evaluación del prototipo software.**

Realización de simulación y análisis de resultados: esta actividad consiste en definir algunas variables de disponibilidad para medir durante la realización de la simulación en una sala de cómputo lo que permitirá tener un ambiente controlado sobre los clientes de la red superpuesta P2P y posteriormente hacer el análisis de los resultados obtenidos con el fin de sacar conclusiones, ventajas y desventajas.

De forma paralela se elaborara la documentación correspondiente para cada etapa desde la etapa 1 hasta la etapa 4.

4. Resumen de entregables

4.1. Relacionados con la gestión del trabajo de grado

- Monografía del trabajo de grado
- Documento de anexos
- Artículo de investigación

4.2. Relacionados con el Prototipo Software

- Definición de la arquitectura del sistema
- Modelo de los diferentes diagramas
- Manual de instalación
- Manual de usuario
- Prototipo software que cumpla con la arquitectura propuesta en este trabajo y su código fuente.

5. Criterios de éxito del Proyecto

El proyecto será exitoso si como resultado genera:

- Los entregables relacionados con la gestión del trabajo de grado.
- Los entregables relacionados con el prototipo software.

ANEXO B

DOCUMENTO DE ESPECIFICACION DE REQUERIMIENTOS

1. Introducción

1.1. Propósito

En este documento se realiza la descripción de los requerimientos que han sido identificados para el desarrollo del prototipo software, con el cual se hará la verificación de la técnica para soporte de la disponibilidad de servicios en un entorno de redes superpuestas P2P no estructuradas.

1.2. Alcance

Para que el desarrollo del prototipo software logre verificar la técnica para soporte de la disponibilidad de servicios propuesta se utilizarán tres tipos de servicios, media streaming, contenidos y cálculos computacionales. También la mayoría de la funcionalidad del prototipo software se ejecutará de manera automática, con reducida interacción con el usuario final del sistema. Además el prototipo software desarrollado permitirá ejecutar la funcionalidad tanto de cliente como de servidor en una sola interfaz gráfica de usuario.

El prototipo software integrará varias tecnologías, para el desarrollo de redes P2P (JXTA), para proveer el servicio de media streaming (VLC (VLCj)), para proveer el servicio de contenidos (FTP) y para proveer el servicio de cálculos computacionales (Sockets, Hilos y funciones Java).

1.3. Definiciones, siglas y abreviaciones

- GUI: Interfaz Gráfica de Usuario.
- JDK: Java Development Kit.
- JRE: Java Runtime Environment.
- JVM: Java Virtual Machine.
- VLCj: Framework Java para el Media Player VLC.
- JXTA: Framework Desarrollo P2P.
- FTP: File Transfer Protocol.

2. Descripción Global

2.1. Perspectiva del producto

2.1.1. Interfaces con el sistema

El prototipo software deberá integrarse con JXTA, VLCj, FTP, Sockets e Hilos.

2.1.2. Interfaces con el usuario

- El usuario interactuará con el prototipo software a través de una aplicación de escritorio.
- El prototipo software tendrá la funcionalidad de cliente, de servidor y de mensajes de depuración en la misma interfaz gráfica de usuario.
- El usuario tendrá una interfaz gráfica la cual será simple e intuitiva, permitiendo al usuario un fácil manejo de la aplicación.
- La interfaz gráfica contará con un panel de notificaciones donde se mostrarán los mensajes de información, depuración y error.

2.1.3. Interfaces con el hardware

El prototipo software se podrá ejecutar en computadores personales o de escritorio, que tengan una conexión a Internet.

2.1.4. Interfaces con el software

El prototipo software estará implementado en el lenguaje de programación java, al igual que las tecnologías de desarrollo utilizadas como VLCj, JXTA, FTP, Sockets e Hilos. Por lo tanto, el computador donde se ejecute el prototipo deberá tener instalada una Máquina Virtual Java y las librerías de LibVLC (Media Player VLC) y un servidor FTP. Se recomienda la versión 1.8.0_05 del JDK.

2.1.5. Interfaces de comunicación

El prototipo software será soportado por los protocolos de comunicaciones JXTA y su respectiva arquitectura.

2.1.6. Restricciones de memoria

- Para espacio de memoria se necesitarán como mínimo 512 MB de memoria RAM y un procesador con una velocidad de 1.2 GHz para ejecutar la aplicación.
- Para tener un buen desempeño de la aplicación se recomienda una memoria RAM de 2GB o más y un procesador con una velocidad de 2.0 GHz o más.

2.1.7. Requerimientos de adaptación

El prototipo software no tendrá en cuenta aspectos de seguridad, tales como nombre de usuarios o contraseñas, o sistemas de autenticación.

2.2. Funciones del producto

El prototipo software permitirá el despliegue de los servicios de media streaming, contenidos y cálculos computacionales a través de la red superpuesta P2P permitiendo conexiones de tipo punto a multipunto en el nivel de aplicación. Teniendo como base una fuente principal de media streaming (Servidor de Streaming) y una fuente principal de contenidos (Servidor FTP) ajenos a la red P2P. Los nodos servidor que ingresen a la red superpuesta P2P prestarán los servicios por medio de la retransmisión de la fuente principal. Por otra parte, los nodos clientes que ingresen a la red P2P buscarán los servicios de media streaming, contenidos y cálculos computacionales, y los consumirán de un determinado nodo servidor elegido aleatoriamente. En caso de fallas de red o de desconexión en el nodo servidor, el nodo cliente estará en la capacidad de recuperar los servicios de media streaming, contenidos y cálculos computacionales por medio de otro nodo servidor. Además, el prototipo permitirá ejecutar la funcionalidad de cliente y servidor en una sola interfaz de usuario y mostrará mensajes de información y depuración del proceso que se está llevando a cabo por medio de un panel de notificación.

2.3. Características del usuario

La mayoría de la funcionalidad del prototipo software se ejecutará de manera automática, con reducida interacción con el usuario final del sistema. Las interacciones del usuario final con el sistema serán:

- La selección manual del tipo de funcionamiento del sistema, es decir el sistema puede comportarse como cliente, o como servidor.
- La selección manual del tipo de servicio que desea publicar o consumir, que puede ser media streaming, contenidos o calculos computacionales.
- Ingreso manual del nombre del servicio que desea publicar o consumir.

3. Requerimientos específicos

A continuación se presentan la descripción detallada de los requerimientos funcionales y no funcionales del prototipo software.

3.1. Requerimientos funcionales

REQUERIMIENTO: Publicar servicios	
Código	R-1
Nombre	Publicar servicios
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe crear y publicar los servicios de media streaming, contenidos y cálculos computacionales.
Prerrequisito	Fuentes principales de los servicios de media streaming y contenido disponibles (Servidores de media streaming y FTP ajenos a la red P2P). Se debe haber creado el anuncio descriptor de los servicios para integrarlo en la publicación de cada tipo de servicio.
Manejo de errores	E1: El servicio de media streaming no se puede publicar. El nodo servidor queda a la espera del levantamiento del servicio de media streaming en la fuente principal. El sistema muestra un mensaje de notificación. E2: El servicio de contenidos no se puede publicar. El nodo servidor queda a la espera del levantamiento del servicio de contenidos en la fuente principal. El sistema muestra un mensaje de notificación.

Tabla 1. R1 - Publicar servicios.

REQUERIMIENTO: Buscar los servicios	
Código	R-2
Nombre	Buscar los servicios
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe permitir buscar los servicios de media streaming, contenidos y cálculos computacionales a través de la consulta sobre las publicaciones de los servicios de los nodos servidores.
Prerrequisito	Seleccionar tipo de servicio y digitar nombre del servicio.
Manejo de errores	E1: Servicio no encontrado: el sistema muestra un mensaje de notificación de que el servicio no fue encontrado.

Tabla 2. R2 - Buscar servicios.

REQUERIMIENTO: Monitorear condición de los servicios	
Código	R-3
Nombre	Monitorear condición de los servicios
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe permitir el monitoreo de la condición de los servicios de media streaming, contenidos y cálculos computacionales que un nodo cliente se encuentra consumiendo. En caso de detectar desconexión o pérdida de cualquiera de los servicios se debe disparar inmediatamente el mecanismo de recuperación. Esta funcionalidad pertenece a un nodo cliente.
Prerrequisito	Nodo cliente estar consumiendo alguno de los servicios.
Manejo de errores	

Tabla 3. R3 – Monitorear condición de los servicios.

REQUERIMIENTO: Recuperar los servicios	
Código	R-4
Nombre	Recuperar los servicios
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe recuperar los servicios de media streaming, contenidos y cálculos computacionales. Un nodo cliente del sistema realiza una nueva búsqueda en el cache local y/o búsqueda remota con el objetivo de encontrar nuevos anuncios de servidores disponibles y recuperar los servicios.
Prerrequisito	Nodo cliente estar consumiendo alguno de los servicios.
Manejo de errores	

Tabla 4. R4 – Recuperar los servicios.

REQUERIMIENTO: Crear mensaje con el perfil para el servicio de media streaming	
Código	R-5
Nombre	Crear anuncio descriptor para el servicio de media streaming
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe crear un mensaje con el perfil del servicio de media streaming a integrar en la publicación del servicio, en este mensaje se especifica el perfil del

	servicio de media streaming con la información de la unidad funcional, entradas y salidas, además de la forma como se debe consumir el servicio.
Prerrequisito	La fuente principal de media streaming debe estar disponible (Servidor de media Streaming ajeno a la red P2P).
Manejo de errores	E1: La fuente principal de media streaming no está disponible. El sistema muestra un mensaje de notificación y espera hasta el restablecimiento de la fuente principal.

Tabla 5. R5 – Crear mensaje descriptor para el servicio de media streaming.

REQUERIMIENTO: • Crear mensaje con el perfil para el servicio de contenidos	
Código	R-6
Nombre	Crear perfil para el servicio de contenidos
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe crear un mensaje perfil del servicio de contenidos a integrar en la publicación del servicio, en este mensaje se especifica el perfil del servicio de contenidos con la información de la unidad funcional, entradas y salidas, además de la forma como se debe consumir el servicio.
Prerrequisito	La fuente principal de contenido debe estar disponible (Servidor de FTP ajeno a la red P2P).
Manejo de errores	E1: La fuente principal de contenido no está disponible. El sistema muestra un anuncio de notificación y espera hasta el restablecimiento de la fuente principal.

Tabla 6. R6 – Crear mensaje descriptor para el servicio de contenidos.

REQUERIMIENTO: Crear mensaje con el perfil para el servicio de cálculos computacionales	
Código	R-7
Nombre	Crear perfil para el servicio de cálculos computacionales
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe crear un mensaje con el perfil del servicio de cálculos computacionales a integrar en la publicación del servicio, en este mensaje se especifica el perfil del servicio de cálculos computacionales con la información de la unidad funcional, entradas y salidas, además de la forma como se debe consumir el servicio.

Técnica para soporte de la disponibilidad de servicios en un entorno de redes
superpuestas P2P no estructuradas

Prerrequisito	
Manejo de errores	

Tabla 7. R7 – Crear mensaje descriptor para el servicio de cálculos computacionales.

REQUERIMIENTO: Obtener perfil de los servicios	
Código	R-8
Nombre	Obtener anuncio descriptor de los servicios
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe permitir consultar las publicaciones de los servicios de los nodos servidores para poder obtener el descriptor de cada servicio.
Prerrequisito	Debe haber publicaciones de los servicios por parte de los nodos servidor.
Manejo de errores	

Tabla 8. R8 – Obtener mensaje descriptor de los servicios.

REQUERIMIENTO: Consumir servicios	
Código	R-9
Nombre	Consumir servicios
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe permitirle a través del descriptor de cada servicio iniciar el consumo de los servicios que presta un nodo servidor.
Prerrequisito	Debe haber publicaciones disponibles de los servicios por parte de los nodos servidores.
Manejo de errores	

Tabla 9. R9 – Consumir servicios.

REQUERIMIENTO: Iniciar el nodo servidor del servicio de media streaming	
Código	R-10
Nombre	Iniciar el nodo servidor del servicio de media streaming
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe permitir al usuario iniciar el nodo servidor de media streaming.
Prerrequisito	El nodo servidor de media streaming debe estar detenido.
Manejo de errores	

Tabla 10. R10 – Iniciar el nodo servidor del servicio de media streaming.

REQUERIMIENTO: Iniciar el nodo servidor del servicio de contenidos	
Código	R-11
Nombre	Iniciar el nodo servidor del servicio de contenidos
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe permitir al usuario iniciar el nodo servidor de contenidos.
Prerrequisito	El nodo servidor de contenido debe estar detenido.
Manejo de errores	

Tabla 11. R11 – Iniciar el nodo servidor del servicio de contenidos.

REQUERIMIENTO: Iniciar el nodo servidor del servicio de cálculos computacionales	
Código	R-12
Nombre	Iniciar el nodo servidor del servicio de cálculos computacionales
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe permitir al usuario iniciar el nodo servidor de cálculos computacionales.
Prerrequisito	El nodo servidor de cálculos computacionales debe estar detenido.
Manejo de errores	

Tabla 12. R12 – Iniciar el nodo servidor del servicio de cálculos computacionales.

REQUERIMIENTO: Detener el nodo servidor	
Código	R-13
Nombre	Detener el nodo servidor
Fuente	Información recolectada en la fase de investigación
Prioridad	Alta/Esencial X Media/Deseado Baja/Opcional
Descripción	El sistema debe permitir al usuario detener el nodo servidor de los servicios.
Prerrequisito	El nodo servidor de los servicios debe estar funcionando.
Manejo de errores	E1: El nodo servidor no se puede detener. El sistema muestra un mensaje de error.

Tabla 13. R13 – Detener el nodo servidor.

REQUERIMIENTO: Iniciar el nodo cliente	
Código	R-14

Nombre	Iniciar el nodo cliente
Fuente	Información recolectada en la fase de investigación
Prioridad	X Alta/Esencial Media/Deseado Baja/Opcional
Descripción	El sistema debe permitir al usuario iniciar el nodo cliente.
Prerrequisito	El nodo cliente debe estar detenido.
Manejo de errores	E1: El nodo cliente no se puede iniciar. El sistema muestra un mensaje de error.

Tabla 14. R14 – Iniciar el nodo cliente.

REQUERIMIENTO: Detener el nodo cliente	
Código	R-15
Nombre	Detener el nodo cliente
Fuente	Información recolectada en la fase de investigación
Prioridad	Alta/Esencial X Media/Deseado Baja/Opcional
Descripción	El sistema debe permitir al usuario detener el nodo cliente.
Prerrequisito	El nodo cliente debe estar funcionando.
Manejo de errores	E1: El nodo cliente no se puede detener. El sistema muestra un mensaje de error.

Tabla 15. R15 – Detener el nodo cliente.

3.2. Requerimientos no funcionales

- **Disponibilidad:** la disponibilidad es una de las características inmersas en las redes P2P, ya que los nodos pueden salir y entrar en la red de manera arbitraria de forma voluntaria o involuntaria, por lo tanto el sistema debe brindar un alto nivel de disponibilidad a través de la técnica de disponibilidad propuesta.
- **Portabilidad:** el prototipo software debe funcionar en diferentes sistemas operativos que tenga instalada una maquina virtual java, ademas que tenga acceso a internet.
- **Reusabilidad:** el prototipo software debe permitir que su código pueda ser utilizado por otros sistemas.
- **Robustez:** el prototipo software debe ser capaz de recuperarse ante un eventual fallo.

4. Restricciones de diseño

No se ha detectado restricciones de diseño.

ANEXO C

TECNOLOGIAS UTILIZADAS PARA EL DESARROLLO DEL PROTOTIPO SOFTWARE

1. Tecnología JXTA

JXTA es una plataforma de computación de red abierta, diseñada para computación P2P, que provee los bloques de construcción básica y servicios requeridos para permitir conectividad, a cualquier aplicación donde quiera que esta se encuentre [1].

El nombre “JXTA” no es un acrónimo. Es una contracción para yuxtapose, que significa, colocado junto a algo o en posición inmediata a algo. Es bien conocido que P2P es yuxtapuesto, a la computación cliente-servidor, o a la computación basada en Web, las cuales son modelos tradicionales de computación distribuida de hoy en día [1].

JXTA es un conjunto de protocolos P2P abiertos y generalizados, que permiten a cualquier dispositivo de red comunicarse y colaborar mutuamente como nodos. Los protocolos JXTA son independientes del lenguaje de programación, y existen múltiples implementaciones para diferentes entornos [1].

JXTA al estar basado en tecnologías probadas y estandarizadas tales como HTTP, TCP/IP y XML no depende de un lenguaje de programación en particular ni de la plataforma de red ni del sistema operativo y por eso puede trabajar con cualquier combinación de ellos [1].

1.1. Servicios JXTA

Un servicio es un conjunto de funciones que brinda un proveedor. Un nodo puede ofrecer un servicio por sí mismo o en cooperación con otros nodos. Un nodo proveedor de servicios publicita un anuncio del servicio, otros pares pueden descubrir éste servicio y hacer uso de él. Cada servicio tiene una ID única y un nombre que consiste en una cadena nombre y una serie de claves descriptivas que identifican unívocamente al servicio [1].

los servicios JXTA expanden las propiedades del núcleo y facilitan el desarrollo de aplicaciones. En esta capa se proveen mecanismos de búsqueda, intercambio, indexado y obtención de código y contenido para habilitar puentes entre aplicaciones y traducido de archivos [1].

1.2. Anuncios

Todos los recursos de red en el proyecto JXTA tales como nodos, grupos de nodos, pipes y servicios, son representados por anuncios (advertisements). Los

anuncios son estructuras de metadata escritas en un lenguaje neutral, son descriptores de recursos representados como documentos XML.

El proyecto JXTA estandariza los anuncios para los siguientes recursos del núcleo JXTA: nodos, grupos de nodos, pipes, servicios, medición, ruteo, contenido, rendezvous, endpoint y transporte [1].

1.3. Protocolos

Los protocolos de JXTA se componen de una serie de seis protocolos divididos en dos categorías:

1.3.1. Core Specification Protocols

Estos protocolos dan toda la funcionalidad requerida por todas las implementaciones. Las implementaciones que deseen ser compatibles con JXTA deben implementar todos los Core Specification Protocols, pero aun así la implementación del Core Specification Protocols no garantiza, ni siquiera provee interoperabilidad con otras implementaciones JXTA, por lo que se necesita la implementación de otros comportamientos. El Core Specification Protocols define los siguientes protocolos [1]:

- **El Endpoint Routing Protocol (ERP):** Es el protocolo por el cual un nodo puede descubrir rutas, para enviar un mensaje a otro nodo. El ERP es usado para administrar y determinar la información de ruteo. El enrutamiento se lleva a cabo por medio de una secuencia ordenada de IDs de los nodos intermediarios para llevar un mensaje de un nodo a otro, por lo tanto este protocolo permite a los nodos enviar mensajes sin que haya conexiones directas entre ellos.
- **El Peer Resolver Protocol (PRP):** Es el protocolo por el cual un nodo puede enviar una petición genérica a uno o más nodos y recibir una o muchas respuestas. El protocolo PRP permite la diseminación de peticiones genéricas a uno o más manejadores dentro del grupo. Una petición dada puede ser recibida por cualquier número de nodos dentro del grupo, posiblemente todos. Este también permite a los nodos y servicios definir e intercambiar cualquier información arbitraria que necesiten. Este protocolo usa el servicio rendezvous para diseminar una consulta a múltiples nodos y usa mensajes unicast para enviar consultas a nodos específicos.

1.3.2. Standard Service Protocols

Estos protocolos definen los componentes y comportamientos requeridos para todas las implementaciones JXTA, no se requiere que las implementaciones JXTA implementen estos servicios, pero es altamente recomendado que lo hagan, sin embargo, al implementar estos servicios se proveerá mayor interoperabilidad con otras implementaciones y más amplia funcionalidad. El Standard Service Protocols define los siguientes protocolos [1]:

- **El Rendezvous Protocol (RVP):** este protocolo es usado por los nodos para suscribirse o ser un suscriptor de un servicio. Dentro de un grupo, los nodos pueden ser rendezvous o nodos que escuchan a los rendezvous. RVP permite enviar mensajes a todos los que escuchan un servicio. RVP es usado por el Peer Resolver Protocol (PRP) para propagar los mensajes.
- **El Peer Discovery Protocol (PDP):** este protocolo es usado por los nodos para publicar sus propios anuncios y descubrir anuncios de otros nodos. PDP usa el Peer Resolver Protocol (PRP) para enviar y propagar peticiones de descubrimiento de anuncios.
- **El Peer Information Protocol (PIP):** este protocolo es usado por los nodos para obtener información del estado de otros nodos, tales como su estado, tiempo de actividad, carga de tráfico, capacidades, etc. PIP usa el Peer Resolver Protocol (PRP) para enviar y propagar peticiones de información.
- **El Pipe Binding Protocol (PBP):** este protocolo es usado por los nodos para establecer un canal de comunicación virtual o pipe entre uno o más nodos. El PBP es usado también para enlazar los dos o más extremos de conexión de un pipe. PBP usa el Peer Resolver Protocol (PRP) para enviar y propagar peticiones de enlazado de pipes.

En “Redes Peer to Peer y Tecnología JXTA” [1], se muestra cada uno de los protocolos JXTA y su relación con los otros protocolos. Figura 1.

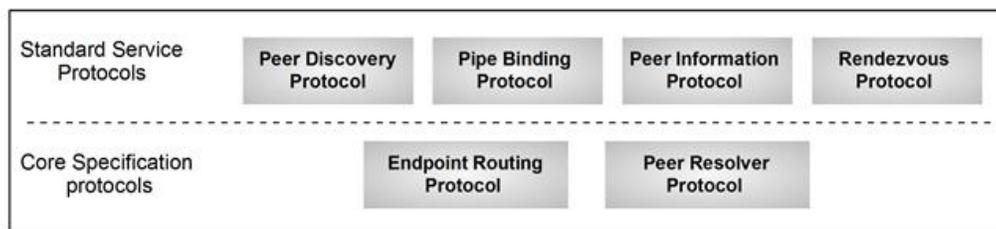


Figura 1. Protocolos JXTA

2. VLC

VLC es un programa simple escrito entorno a la librería libvlc, este programa es muy pequeño, pero es un reproductor multimedia con todas las funciones gracias al apoyo de la librería libvlc para módulos dinámicos. Libvlc es la parte central de VLC, ya que es una biblioteca que proporciona una interfaz para programas como VLC con una gran cantidad de funcionalidades, VLC es un software libre distribuido bajo licencia GPL inicialmente llamado VideoLAN Client [21].

Algunas de las características que facilitan el uso de esta tecnología, se describen a continuación:

- **Multiplataforma:** Debido a su naturaleza, el VLC puede operar en múltiples plataformas o en diferentes arquitecturas de sistemas operativos.
- **Modularidad:** VLC presenta un diseño modular. De esta manera se facilita incluir módulos, plugins, para el soporte de nuevos archivos, códecs o métodos de streaming.
- **Interfaces:** VLC tiene una interfaz estándar basada en módulos, por lo que tiene soporte para consola y múltiples máscaras para interfaz gráfica (Windows y Linux, OS X y BeOS).
- **Paquetes:** VLC es un reproductor basado en paquetes; reproduce casi todo los tipos de contenido de vídeo. Puede reproducir éstos, aunque estén dañados, incompletos o no terminados como, por ejemplo, archivos que todavía se están descargando vía redes superpuestas P2P.

2.1. LibVLC

libVLC es la parte central y la interfaz del entorno multimedia en el que se basa VLC Media Player. Es una librería nativa desarrollada en C++ para soportar difusión multimedia usando diversos tipos de protocolos (PE, HTTP, RTCP), salida de audio y video, manejo de plugins, entre otras funciones. Posee subcomponentes como interface, playlist, input, video_output, audio_output, streaming_output, encargados de todas sus funcionalidades [21].

2.2. VLCj

La librería VLCj puede ser usado para crear muchos tipos de aplicaciones de medios, incluyendo reproductores de audio, reproductores de películas con salida de vídeo nativo incrustado en un componente de Java, clientes de la red de

transmisión y servidores, clientes web-cam, aplicaciones de procesamiento de vídeo y así sucesivamente [22].

Debido a que libVLC está escrita en lenguaje C++, puede ser utilizada directamente con lenguaje java, por lo tanto la librería VLCj ofrece enlaces Java de bajo nivel a la biblioteca libVLC nativo, y también proporciona un marco de programación con las clases de nivel superior y las construcciones que más encapsulan, si no todas, de las dificultades que cuando se trabaja con libVLC [22].

Debido a que el proyecto VLCJ está siempre en constante evolución y mucho más maduro, proporcionando acceso a la mayoría de características brindadas por el libVLC, posibilidad de construcción de aplicaciones Java multimedia capaces de reproducir casi cualquier tipo de medios de comunicación; se ha escogido como tecnología para la implementación y puesta a prueba de la técnica de recuperación y ejecución de un servicio de media streaming [21].

3. FTP

Es un protocolo de red utilizado para la transferencia de archivos (File Transfer Protocol), los orígenes de FTP se remontan a 1971, cuando los ingenieros del MIT y otras instituciones académicas buscaban un método eficaz para la transferencia de archivos, este protocolo de red básicamente se encuentra diseñado en torno a una arquitectura del tipo cliente-servidor, es decir que el equipo o computadora cliente se debe conectar primero a un servidor para descargar o añadir archivos, este protocolo utiliza por defecto los puertos 20 y 21 [23].

Algunas de las características que facilitan el uso de esta tecnología, se describen a continuación:

- Permite que equipos remotos puedan compartir archivos.
- Permite la independencia entre los sistemas de archivo del equipo del cliente y del equipo del servidor.
- Permite una transferencia de datos eficaz.
- Se encuentra mucha información acerca de esta tecnología y su facilidad de utilización.

3.1. Servidor FTP

Un servidor FTP es un software que se encuentra instalado en una computadora servidor conectada a Internet, o en el caso de corporaciones, instituciones u otras también puede estar conectada a redes LAN o MAN. El principal propósito de este

tipo de software de servidor de FTP es permitir el acceso y el intercambio controlado de archivos contenidos en la computadora en la que se alojan, con otras computadoras que lo requieren. Es decir que el software de servidor FTP es el encargado de procesar las peticiones para la descarga de archivos [23].

3.2. Cliente FTP

La conexión con un servidor FTP se realiza mediante los programas llamados clientes de FTP, los clientes FTP básicos vienen integrados en algunos sistemas operativos, sin embargo hay disponibles varios clientes con más funcionalidades en internet gratuito y de pago, algunos ejemplos de clientes FTP más usados son FileZilla, CuteFTP y WS_FTP [23].

4. Sockets

Es un punto final de un enlace de comunicación bidireccional, por medio del cual dos programas pueden comunicarse e intercambiar cualquier flujo de datos, un socket está unido a un número de puerto de modo que la capa de comunicación de la familia de protocolos de internet (TCP, IP) puede identificar la aplicación y los datos a ser enviados [24].

Algunas de las características que facilitan el uso de esta tecnología, se describen a continuación:

- Es una forma fácil de comunicar procesos tanto locales como remotos.
- Para la transmisión de paquetes pueden utilizar UDP o TCP.
- Permiten implementar una arquitectura cliente-servidor.
- Facilidad de encontrar documentación.

5. Hilos

Es un programa en ejecución; a veces se llaman procesos ligeros. Ambos procesos e hilos ofrecen un entorno de ejecución, pero la creación de un nuevo hilo requiere menos recursos que la creación de un nuevo proceso. Los hilos existen dentro de un proceso, cada proceso tiene al menos un hilo [25].

ANEXO D

HISTORIAS DE USUARIO

1. Historias de usuario

A continuación se presenta la descripción de las historias de usuario para el prototipo software. La priorización de las historias de usuario, se definió en un rango de 1-5, siendo 1 la prioridad más baja y 5 la prioridad más alta.

Fecha	27/02/15			
Identificador de la historia de usuario	HU-01			
Nombre historia de usuario	Solicitar un servicio de contenidos.			
Rol	Cliente			
Prioridad	4			
Característica/funcionalidad	El cliente necesita descargar un contenido.			
Razón/Resultado	Con la finalidad de obtener un contenido.			
Escenarios				
Número	Criterio de aceptación	Contexto	Evento	Resultado/Comportamiento esperado
1	El contenido solicitado está disponible en la red superpuesta P2P.	En caso de que haya al menos un nodo en la red con el contenido solicitado.	Cuando el cliente recibe el contenido solicitado.	A continuación el cliente tendrá acceso al contenido.
2	El contenido solicitado no está disponible en la red superpuesta P2P.	En caso de que no haya un nodo en la red con el contenido solicitado.	Cuando el cliente no encuentra ese contenido disponible en la red superpuesta P2P.	El cliente no tiene acceso al archivo.

Tabla 16. HU - Solicitar un servicio de contenidos.

Fecha	27/02/15
Identificador de la historia de usuario	HU-02
Nombre historia de	Solicitar un servicio de media streaming.

Técnica para soporte de la disponibilidad de servicios en un entorno de redes superpuestas P2P no estructuradas

usuario				
Rol	Cliente			
Prioridad	2			
Característica/funcionalidad	El cliente necesita visualizar un servicio de media streaming.			
Razón/Resultado	Con la finalidad de visualizar un servicio de media streaming.			
Escenarios				
Número	Criterio de aceptación	Contexto	Evento	Resultado/Comportamiento esperado
1	El servicio de media streaming solicitado está disponible en la red superpuesta P2P.	En caso de que haya al menos un nodo en la red prestando el servicio de media streaming solicitado.	Cuando el cliente recibe el servicio de media streaming solicitado.	A continuación el cliente tendrá acceso al servicio de media streaming solicitado.
2	El servicio de media streaming solicitado no está disponible en la red superpuesta P2P.	En caso de que no haya un nodo en la red prestando el servicio de media streaming solicitado.	Cuando el cliente no recibe el servicio de media streaming porque no está disponible en la red superpuesta P2P.	El cliente no tiene acceso al servicio de media streaming solicitado.

Tabla 17. HU - Solicitar un servicio de media streaming.

Fecha	27/02/15
Identificador de la historia de usuario	HU-03
Nombre historia de usuario	Solicitar un servicio de cálculos computacionales.
Rol	Cliente
Prioridad	4
Característica/funcionalidad	El cliente necesita realizar un cálculo computacional.
Razón/Resultado	Con la finalidad de obtener un resultado.

Escenarios				
Número	Criterio de aceptación	Contexto	Evento	Resultado/Comportamiento esperado
1	El servicio de cálculos computacionales solicitado está disponible en la red superpuesta P2P.	En caso de que haya al menos un nodo en la red con el servicio de cálculos computacionales solicitado.	Cuando el cliente tiene acceso al servicio permitiéndole enviar los datos de entrada para el cálculo solicitado.	A continuación el cliente obtendrá el resultado del cálculo computacional solicitado.
2	El servicio de cálculos computacionales solicitado no está disponible en la red superpuesta P2P.	En caso de que no haya al menos un nodo en la red con el servicio de cálculos computacionales solicitado.	Cuando el cliente no encuentra el servicio de cálculos computacionales porque no está disponible en la red superpuesta P2P.	El cliente no tiene acceso al servicio.

Tabla 18. HU - Solicitar un servicio de cálculos computacionales.

Fecha	27/02/15			
Identificador de la historia de usuario	HU-04			
Nombre historia de usuario	Publicar un servicio de contenidos			
Rol	Servidor			
Prioridad	3			
Característica/funcionalidad	El servidor necesita publicar un contenido.			
Razón/Resultado	Con la finalidad de compartir un contenido.			
Escenarios				
Número	Criterio de aceptación	Contexto	Evento	Resultado/Comportamiento esperado
1	El servidor tiene	El servidor	El servidor	El contenido queda

Técnica para soporte de la disponibilidad de servicios en un entorno de redes superpuestas P2P no estructuradas

disponible el contenido para publicarlo.	está disponible.	realiza la publicación del contenido en la red superpuesta P2P.	disponible en la red superpuesta P2P.
--	------------------	---	---------------------------------------

Tabla 19. HU – Publicar un servicio de contenidos.

Fecha	27/02/15			
Identificador de la historia de usuario	HU-05			
Nombre historia de usuario	Publicar un servicio de media streaming.			
Rol	Servidor			
Prioridad	2			
Característica/funcionalidad	El servidor necesita publicar un servicio de media streaming.			
Razón/Resultado	Con la finalidad de compartir un servicio de media streaming.			
Escenarios				
Número	Criterio de aceptación	Contexto	Evento	Resultado/Comportamiento esperado
1	El servidor tiene disponible el servicio de media streaming para publicarlo.	El servidor está disponible.	El servidor realiza la publicación del servicio de media streaming en la red superpuesta P2P.	El servicio de media streaming queda disponible en la red superpuesta P2P.

Tabla 20. HU – Publicar un servicio de media streaming.

Fecha	27/02/15			
Identificador de la historia de usuario	HU-06			
Nombre historia de usuario	Publicar un servicio de cálculos computacionales.			
Rol	Servidor			
Prioridad	3			

Característica/funcionalidad	El servidor necesita publicar el servicio de un cálculo computacional.			
Razón/Resultado	Con la finalidad de compartir recursos computacionales.			
Escenarios				
Número	Criterio de aceptación	Contexto	Evento	Resultado/Comportamiento esperado
1	El servidor tiene disponible el servicio de cálculos computacionales para publicarlo.	El servidor está disponible.	El servidor realiza la publicación del servicio de cálculos computacionales en la red superpuesta P2P.	El servicio de cálculos computacional queda disponible en la red superpuesta P2P.

Tabla 21. HU – Publicar un servicio de cálculos computacionales.

Fecha	27/02/15			
Identificador de la historia de usuario	HU-07			
Nombre historia de usuario	Replicación de servicios.			
Rol	Servidor			
Prioridad	5			
Característica/funcionalidad	El servidor necesita replicar un servicio.			
Razón/Resultado	Con la finalidad de proveer disponibilidad.			
Escenarios				
Número	Criterio de aceptación	Contexto	Evento	Resultado/Comportamiento esperado
1	El servidor tiene al menos un servicio disponible para replicar.	Que haya al menos un nodo servidor en la red superpuesta P2P con la intención de replicar el servicio.	El servicio queda replicado en otro nodo.	El servicio queda disponible en otro nodo.

Tabla 22. HU – Replicación de servicios.

ANEXO E

MANUAL TECNICO Y DE INSTALACION

1. Requerimientos software del prototipo

1.1. Instalación de Java

Para el funcionamiento de la aplicación es necesario instalar el Entorno de Ejecución de Java (JRE), pero si se desea se puede instalar el Java Development Kit (JDK) en nuestro equipo tenemos la versión Java 8 Update 25 que se la puede descargar desde Oracle [2].

1.2. Lista de excepciones Java

Para activar y poder ejecutar aplicaciones Java se debe configurar el apartado de seguridad en el panel de control de Java. Para esto, abrir 'Panel de control', abrir "Programas" y en la sección 'Java' dirijase al apartado de seguridad.

- Para versiones anteriores a Java 8, coloque el nivel de seguridad más bajo.
- Para versiones iguales o superiores a Java 8, agregue en la sección 'Editar lista de sitios', la dirección URL que aparece en el navegador al iniciar la aplicación.

Para poder utilizar java se debe configurar las variables de entorno del equipo esto se hace para que se pueda dar la comunicación entre Java y el sistema operativo, lo más importante que se debe configurar son "JAVA_HOME" y "PATH" las cuales le indican al sistema operativo donde y como ubicar Java dentro del mismo.

En www.aprenderaprogramar.com [3] se puede encontrar una guía detallada para la configuración de estas dos variables.

1.3. Instalación de VLC

Para el funcionamiento de la aplicación también es necesario instalar el paquete de VLC Media Player en el equipo que se ejecutará la aplicación, ya que contiene la librería núcleo libVLC para interactuar con las funcionalidades de media streaming. Sólo es necesario instalar el software VLC Media Player en su versión 2.0.10 o superiores.

Desde VLC-media [4] se puede descargar VLC Media Player para distintos sistemas operativos.

1.4. Crear servidor FTP

Para el funcionamiento de la aplicación se debe crear un servidor FTP, el cual es necesario para simular el servicio de contenidos, en [6] se puede encontrar una guía detallada del proceso para configurar el equipo donde se va a ejecutar la aplicación, con el fin de montar el servidor de FTP, tanto para la LAN como para Internet, este servidor debe ser local a la máquina donde se ejecutará el cliente y debe tener la misma dirección del host, la dirección debe ser agregada a la variable ftpServer en la clase Utilities, además de crear una carpeta local donde se agregaran todos los contenidos que el cliente va a publicar en el caso de proveer el servicio de contenidos.

2. Instalación de la aplicación

La instalación de la aplicación no es necesaria, ya que solo se debe copiar en un disco que permita la lectura y escritura, esto porque JXTA y VLC crean archivos temporales en el directorio de ejecución de la aplicación. La estructura del directorio se muestra en la Figura 2.

Importante tener en cuenta las librerías que se encuentran en el paquete de librerías en la estructura de directorios de la Figura 2, ya que si estas no se encuentran agregadas la aplicación no funcionará.

:

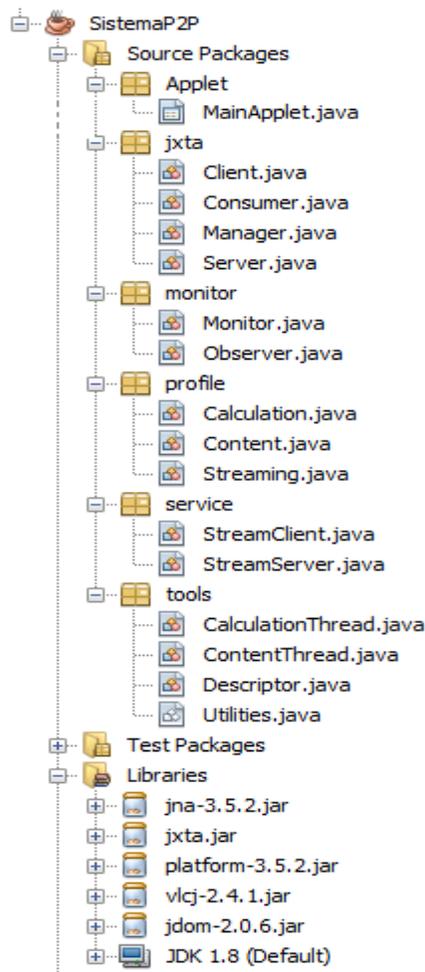


Figura 2. Estructura de directorios

3. Ejecución del prototipo

Para la ejecución del prototipo se debe abrir el proyecto con el entorno de desarrollo Netbeans, abrir el paquete Applet y correr el archivo MainApplet.java.

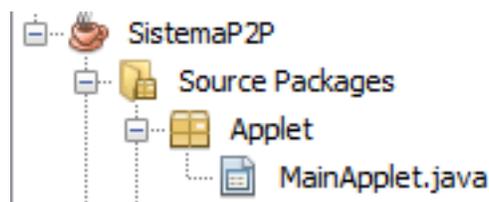


Figura 3. Ejecutar prototipo

ANEXO F

MANUAL DE USUARIO

1. Ejecutar la aplicación

Ejecute el archivo Applet.html para que se inicie la aplicación. Cuando se ejecute la aplicación se mostrara la Figura 4.

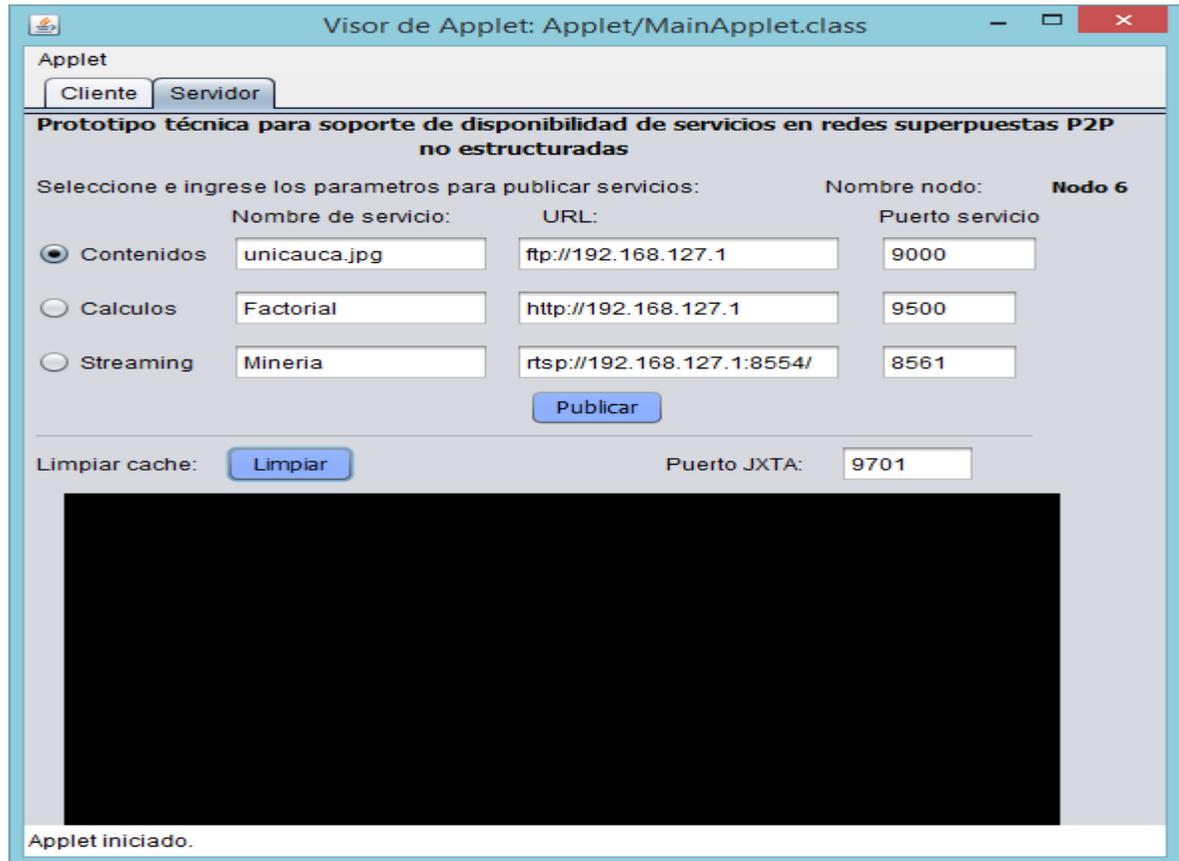


Figura 4. Interfaz del prototipo

El prototipo tiene dos interfaces una para rol de servidor desde donde se lleva a cabo la publicación de servicios y la otra para el rol de cliente desde donde se lleva a cabo el consumo de los servicios, a continuación se describe el funcionamiento para cada una de las interfaces del prototipo.

1.1. Funcionamiento de la interfaz del servidor

Esta interfaz cuenta con cuatro campos que especifican los requisitos que se necesitan para publicar cada uno de los tipos de servicios como son media

streaming, contenido y cálculos computacionales estos campos se muestran en la Figura 5:

Seleccione e ingrese los parámetros para publicar servicios: Nombre nodo: **Nodo 6**

	Nombre de servicio:	URL:	Puerto servicio
<input checked="" type="radio"/> Contenidos	<input type="text" value="unicauca.jpg"/>	<input type="text" value="ftp://192.168.127.1"/>	<input type="text" value="9000"/>
<input type="radio"/> Calculos	<input type="text" value="Factorial"/>	<input type="text" value="http://192.168.127.1"/>	<input type="text" value="9500"/>
<input type="radio"/> Streaming	<input type="text" value="Mineria"/>	<input type="text" value="rtsp://192.168.127.1:8554/"/>	<input type="text" value="8561"/>

Figura 5. Campos requeridos para publicar cada servicio

El primer campo es de tipo selección para seleccionar que tipo de servicio se desea publicar, el segundo campo es para ingresar el nombre del servicio, el tercer campo es para especificar la IP del servidor fuente desde donde se va a prestar el servicio y finalmente el cuarto campo es para ingresar el puerto por donde se va a establecer la comunicación para prestar el servicio que se va a publicar. Finalmente se encuentra el botón “Publicar” mediante el cual se envía la orden al sistema de publicar el servicio en la red superpuesta P2P con las especificaciones descritas en los campos. en la Figura 6 se muestra la imagen cuando se publica un servicio y este servidor queda a la espera de clientes.

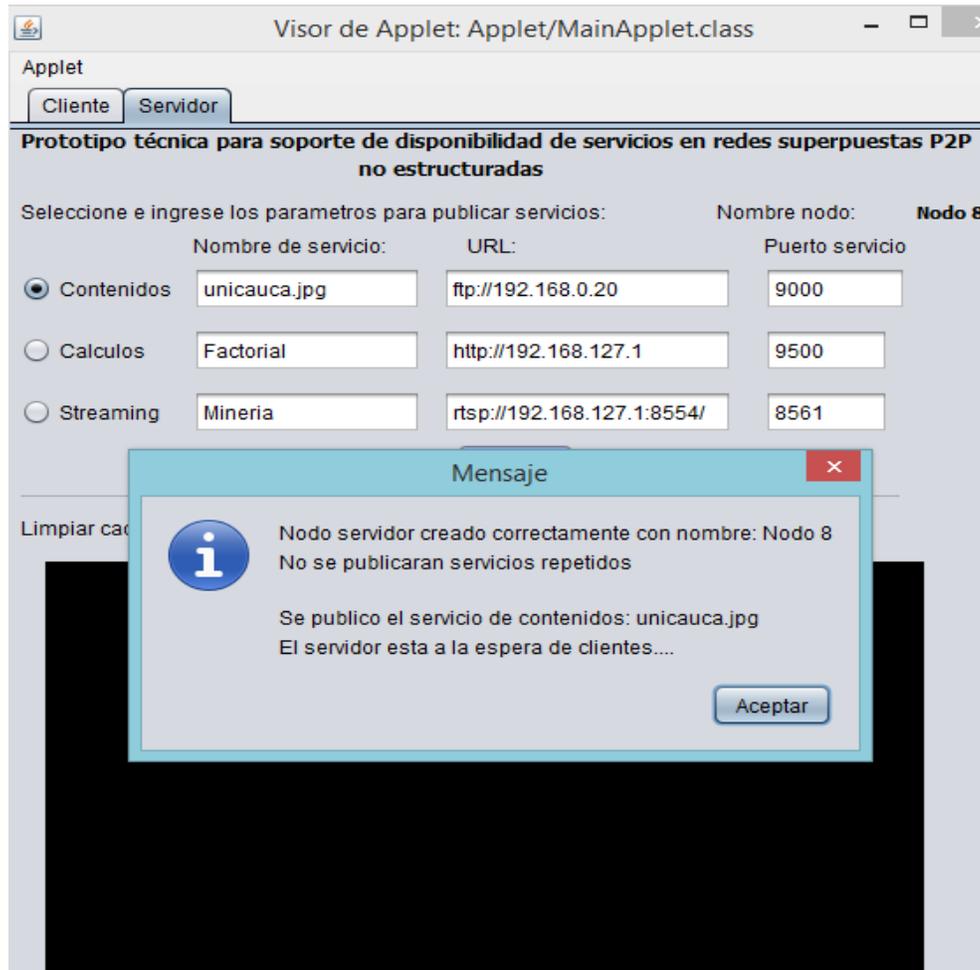


Figura 6. Mensaje de servicio publicado.

En la parte final de esta interfaz se presenta el boton “Limpiar”, mediante el cual se limpia la memoria cache del prototipo, donde se guardan datos de la ejecución de la aplicación, esto es necesario ya que cada vez que se ejecuta el prototipo se generan nuevos archivos lo que puede congestionar la memoria, por lo tanto se recomienda limpiar el cache cada vez que se ejecute el prototipo, también se encuentra un campo para ingresar el puerto para JXTA; este puerto es donde se ejecuta la red superpuesta P2P de JXTA.

1.2. Funcionamiento de la interfaz del cliente

Esta interfaz tambien cuenta en la parte superior con tres botones de tipo selección que especifican los tipos de servicios que el cliente puede consumir como son media streaming, contenido y cálculos computacionales, además hay

un campo denominado “Nombre de servicio” en el cual se debe ingresar el nombre del servicio que se desea consumir, también esta el boton “Consumir”, mediante el cual se envia la solicitud al sistema para buscar y consumir el servicio especificado en los campos mencionados Figura 7.

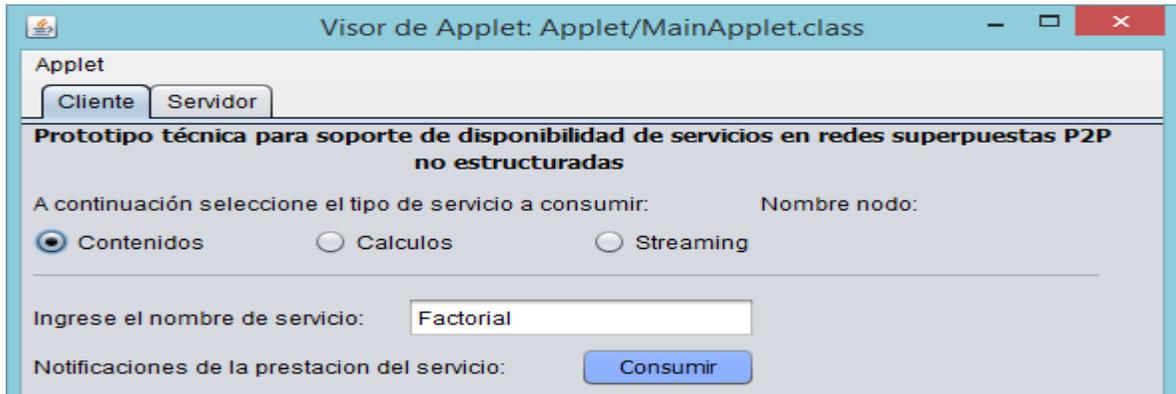


Figura 7. Parte superior de la interfaz del cliente

Finalmente en la parte inferior de esta interfaz Figura 8, se encuentra un panel de notificación donde se mostraran los mensajes del estado de la ejecución del prototipo y un componente de video para cuando el cliente desee consumir el servicio de media streaming.



Figura 8. Parte inferior de la interfaz del cliente

Al consumir un servicio si este es encontrado en la red P2P, los resultados se muestran automáticamente como se puede observar en la Figura 9.



Figura 9. Imagen consumo de servicio en el cliente.

Para consumir el servicio de calculos computacionales se debe adjuntar a la solicitud de servicio un archivo con los datos a calcular la Figura 10 muestra la ventana para cargar el archivo de pruebas.

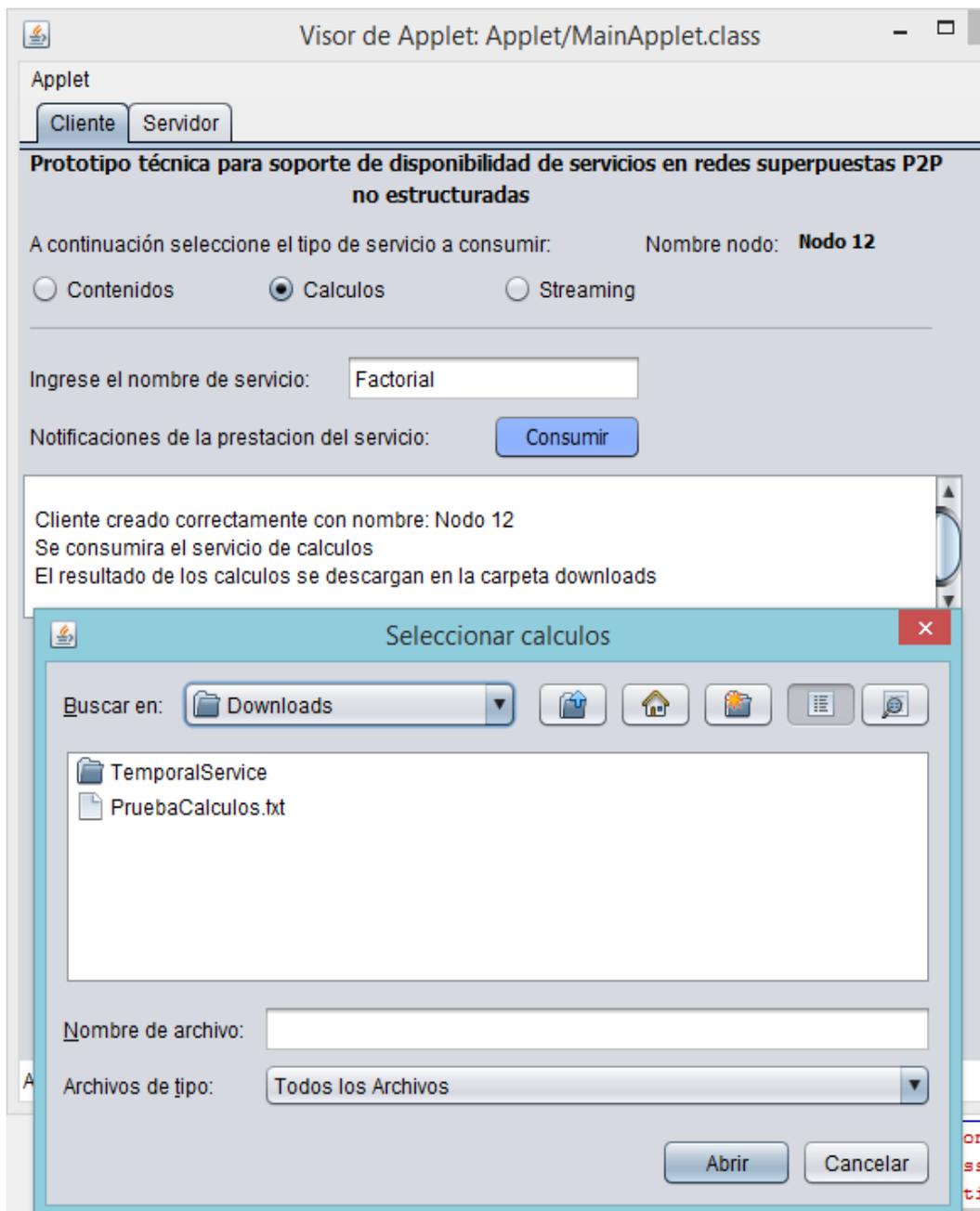


Figura 10. Cargar archivo de prueba para el servicio de cálculos computacionales.

Al terminar de consumir el servicio muestra un archivo con los resultados como se muestra en la Figura 11.

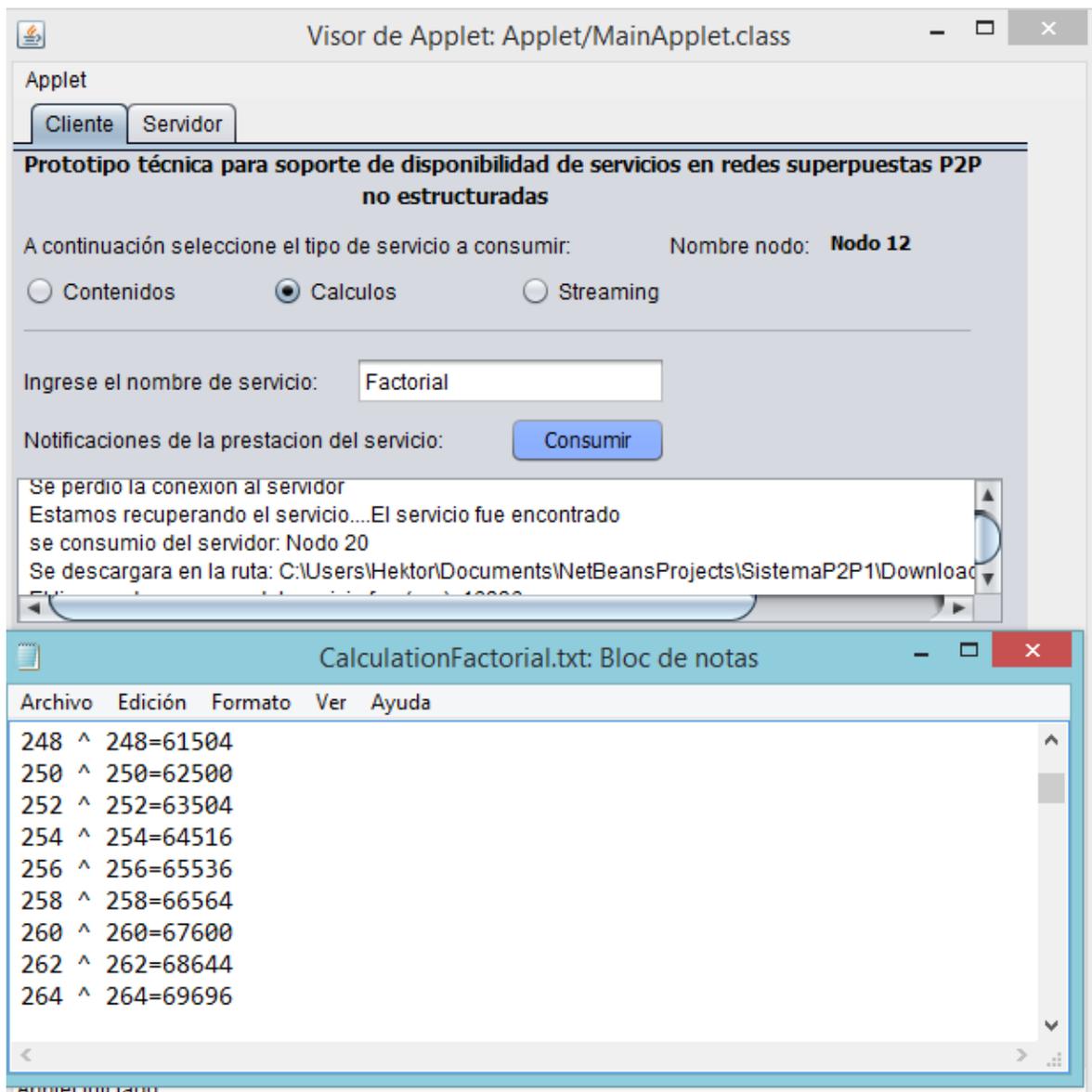


Figura 11. Resultado de consumir el servicio de cálculos computacionales.

Para el tema de media streaming en el proyecto de grado de Potosi y Ordoñez [21], se puede encontrar la forma de publicar, ejecutar, recuperar y consumir un servicio de este tipo.

2. Salir de la aplicación

Para salir de la aplicación, cerrar la ventana donde se esta ejecutando la aplicación.

ANEXO G

PRUEBAS DEL PROTOTIPO SOFTWARE

1.1 Especificación de los equipos utilizados en las pruebas

A continuación, mediante el uso de tablas se presentan las respectivas especificaciones de cada equipo usado en las pruebas con el fin de validar la técnica para soporte de disponibilidad de servicios.

EQUIPO			
SERIAL	PC-01		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	4GB
TIPO PROCESADOR	AMD E-450	VELOCIDAD	1.65GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	RADEON HD6320		

Tabla 23. Equipo 1

EQUIPO			
SERIAL	PC-02		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	8GB
TIPO PROCESADOR	Intel Core i5	VELOCIDAD	2.2GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	Integrada Intel		

Tabla 24. Equipo 2

EQUIPO			
SERIAL	PC-03		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	4GB
TIPO PROCESADOR	Intel Core i5	VELOCIDAD	2.2GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	Integrada Intel		

Tabla 25. Equipo 3

EQUIPO			
SERIAL	PC-04		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	4GB
TIPO PROCESADOR	Intel (R) Dual CPU	VELOCIDAD	2.6GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	nvidia geforce		

Tabla 26. Equipo 4

EQUIPO			
SERIAL	PC-05		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	4GB
TIPO PROCESADOR	Intel (R) core i3	VELOCIDAD	1.8GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	Integrada Intel		

Tabla 27. Equipo 5

EQUIPO			
SERIAL	PC-06		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	4GB
TIPO PROCESADOR	AMD Phenom x4	VELOCIDAD	2.2GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	Integrada Intel		

Tabla 28. Equipo 6

EQUIPO			
SERIAL	PC-07		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	4GB
TIPO PROCESADOR	Intel® Core™ i3	VELOCIDAD	2.4GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	Integrada Intel		

Tabla 29. Equipo 7

EQUIPO			
SERIAL	PC-08		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	2GB
TIPO PROCESADOR	Intel (R) Dual CPU	VELOCIDAD	2.1GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	Integrada Intel		

Tabla 30. Equipo 8

EQUIPO			
SERIAL	PC-09		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	2GB
TIPO PROCESADOR	Intel (R) Dual CPU	VELOCIDAD	2.1GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	Integrada Intel		

Tabla 31. Equipo 9

EQUIPO			
SERIAL	PC-10		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	2GB
TIPO PROCESADOR	Intel (R) Dual CPU	VELOCIDAD	2.1GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	Integrada Intel		

Tabla 32. Equipo 10

EQUIPO			
SERIAL	PC-11		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	2GB
TIPO PROCESADOR	Intel (R) Pentium (R) Dual CPU	VELOCIDAD	1.65GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits(Máquina virtual)		
TARJETA GRAFICA	NA		

Tabla 33. Equipo 11

EQUIPO			
SERIAL	PC-12		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	2GB
TIPO PROCESADOR	Intel (R) Pentium (R) Dual CPU	VELOCIDAD	1.8GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits(Máquina virtual)		
TARJETA GRAFICA	NA		

Tabla 34. Equipo 12

EQUIPO			
SERIAL	PC-13		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	2GB
TIPO PROCESADOR	Intel (R) Pentium (R) Dual CPU	VELOCIDAD	1.8GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	Integrada Intel		

Tabla 35. Equipo 13

EQUIPO			
SERIAL	PC-14		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	2GB
TIPO PROCESADOR	Intel (R) Pentium (R) Dual CPU	VELOCIDAD	1.8GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	Integrada Intel		

Tabla 36. Equipo 14

EQUIPO			
SERIAL	PC-15		
MEMORIA RAM	DDR-SDRAM	CAPACIDAD	2GB
TIPO PROCESADOR	Intel (R) Pentium (R) Dual CPU	VELOCIDAD	1.8GHz
SISTEMA OPERATIVO	Windows 7 Professional 32bits		
TARJETA GRAFICA	Integrada Intel		

Tabla 37. Equipo 15

1.2 Pruebas manuales

Se llevan a cabo para verificar la correcta interacción de los nodos en la red P2P para prestar y consumir diferentes tipos de servicios, creando múltiples instancias de nodos de redes P2P y sus respectivos escenarios.

A continuación se muestra la tabla de escenarios planteados para llevar a cabo las pruebas.

Técnica para soporte de la disponibilidad de servicios en un entorno de redes superpuestas P2P no estructuradas

Numero escenario	Nodos servidor	Numero de fallos	Nodos cliente	Tipo servicio	Característica
1	2	1	Muchos	Contenidos	Los nodos se conectarán a la red, el nodo servidor y replica serán iniciados primero, luego se inician los nodos cliente que consumirán el servicio de contenidos del nodo servidor y antes que se consuma el servicio se presentara un fallo del nodo servidor que presta el servicio.
2	4	3	Muchos	Contenidos	Los nodos se conectarán a la red y un nodo servidor será iniciado primero, luego se inician los tres nodos servidor que replicaran el servicio del primer nodo y por último los clientes que consumirán el servicio de contenidos de los nodos servidores y antes que se consuma el servicio se presentarán tres fallos de los nodos servidores que prestan el servicio.
3	8	6	Muchos	Contenidos	Los nodos se conectarán a la red y un nodo servidor será iniciado primero, luego se inician los siete nodos servidor que replicaran el servicio del primer nodo y por último los clientes que consumirán el servicio de contenidos de los nodos servidores y antes que se consuma el servicio se presentarán seis fallos de los nodos servidores que prestan el servicio.
4	10	8	Muchos	Contenidos	Los nodos se conectarán a la red y un nodo servidor será iniciado primero, luego

Técnica para soporte de la disponibilidad de servicios en un entorno de redes superpuestas P2P no estructuradas

					se inician los nueve nodos servidor que replicaran el servicio del primer nodo y por último los clientes que consumirán el servicio de contenidos de los nodos servidores y antes que se consuma el servicio se presentarán ocho fallos de los nodos servidores que prestan el servicio.
5	2	1	Muchos	Cálculos Computacionales	Los nodos se conectarán a la red, el nodo servidor y replica serán iniciados primero, luego se inician los nodos cliente que consumirán el servicio de cálculos computacionales del nodo servidor y antes que se consuma el servicio se presentara un fallo del nodo servidor que presta el servicio.
6	4	3	Muchos	Cálculos Computacionales	Los nodos se conectarán a la red y un nodo servidor será iniciado primero, luego se inician los tres nodos servidor que replicaran el servicio del primer nodo y por último los clientes que consumirán el servicio de cálculos computacionales de los nodos servidores y antes que se consuma el servicio se presentarán tres fallos de los nodos servidores que prestan el servicio.
7	8	6	Muchos	Cálculos Computacionales	Los nodos se conectarán a la red y un nodo servidor será iniciado primero, luego se inician los siete nodos servidor que replicaran el servicio del primer nodo y por último los clientes que consumirán el servicio de

Técnica para soporte de la disponibilidad de servicios en un entorno de redes
superpuestas P2P no estructuradas

					cálculos computacionales de los nodos servidores y antes que se consuma el servicio se presentarán seis fallos de los nodos servidores que prestan el servicio.
8	10	8	Muchos	Cálculos Computacionales	Los nodos se conectarán a la red y un nodo servidor será iniciado primero, luego se inician los nueve nodos servidor que replicaran el servicio del primer nodo y por último los clientes que consumirán el servicio de cálculos computacionales de los nodos servidores y antes que se consuma el servicio se presentarán ocho fallos de los nodos servidores que prestan el servicio.
9	2	1	Muchos	Media Streaming	Los nodos se conectarán a la red, el nodo servidor y replica serán iniciados primero, luego se inician los nodos cliente que consumirán el servicio de media streaming del nodo servidor y antes que se consuma el servicio se presentara un fallo del nodo servidor que presta el servicio.
10	4	3	Muchos	Media Streaming	Los nodos se conectarán a la red y un nodo servidor será iniciado primero, luego se inician los tres nodos servidor que replicaran el servicio del primer nodo y por último los clientes que consumirán el servicio de media streaming de los nodos servidores y antes que se consuma el servicio se presentarán tres fallos

Técnica para soporte de la disponibilidad de servicios en un entorno de redes superpuestas P2P no estructuradas

					de los nodos servidores que prestan el servicio.
11	8	6	Muchos	Media Streaming	Los nodos se conectarán a la red y un nodo servidor será iniciado primero, luego se inician los siete nodos servidor que replicaran el servicio del primer nodo y por último los clientes que consumirán el servicio de media streaming de los nodos servidores y antes que se consuma el servicio se presentarán seis fallos de los nodos servidores que prestan el servicio.
12	10	8	Muchos	Media Streaming	Los nodos se conectarán a la red y un nodo servidor será iniciado primero, luego se inician los nueve nodos servidor que replicaran el servicio del primer nodo y por último los clientes que consumirán el servicio de media streaming de los nodos servidores y antes que se consuma el servicio se presentarán ocho fallos de los nodos servidores que prestan el servicio.
13	6	3	Muchos	Contenidos, cálculos Computacionales, Media Streaming	Se hace una mezcla de servidores con distintos servicios, se debe probar que se prestan, se replican y se recuperan los servicios.
14	6	3	Muchos	Contenidos, cálculos computacionales, Media Streaming	Se hace una mezcla de servidores con distintos servicios, se debe probar que se prestan, se replican y se recuperan los servicios.

Tabla 38. Definición de escenarios de pruebas

En la tabla 34, se muestran otros campos que se tuvieron en cuenta para registrar los resultados de los escenarios de prueba. Estas pruebas se realizaron de

manera local en un mismo equipo de cómputo diferenciando los nodos a través de la definición de un puerto JXTA y también otro para cada servicio.

Escenario número: Descripción.					
Instancia local	Rol	Puerto servicio	Puerto JXTA servidor consumo actual	Puerto JXTA servidor consumo nuevo	Tiempo de Recuperación Milisegundos
Descripción de resultado:					

Tabla 39. Formato de registro de otros resultados

ANEXO H

RESUMEN BIBLIOGRAFIA ESTUDIADA PARA LA SELECCIÓN DE LA TECNICA DE DISPONIBILIDAD

Durante la revisión bibliográfica se obtuvieron tres mecanismos principales utilizados para mejorar la disponibilidad de servicios sobre redes superpuestas P2P, los cuales son rotación, replicación y tolerancia a fallos, a continuación se presenta el resumen de las diferentes referencias encontradas, las cuales fueron utilizadas para seleccionar la técnica de disponibilidad, bajo los criterios de mayor productividad investigativa y tipo de servicios.

1. Replicación

Dandoush, Alouf y Nain proponen en este trabajo el desarrollo de modelos matemáticos para evaluar las métricas de rendimiento fundamentales (vida y disponibilidad de datos) de P2PSS (sistemas de copia de seguridad). Orientado a servicios de contenidos, aplicaciones de voz IP, copia de seguridad y almacenamiento [7].

Meroufel y Belalem proponen en este trabajo dos estrategias una para la replicación basada en la disponibilidad y otra para la ubicación y reemplazo de réplicas asegurando la disponibilidad deseada con el mínimo de réplicas a pesar de la presencia de fallos de nodos y sin sobrecargar el sistema [8].

Zhou, Fu, y Chiu proponen un enfoque híbrido, donde se considera el problema de replicación como un problema de control adaptativo. Se describen tres modelos: un modelo de solicitud de video, un modelo de programación de servicios y un modelo de control adaptativo. Orientado al servicio de media streaming [9].

Liao, Jin y Yu, proponen un mecanismo de replicación de datos proactiva del lado del cliente. Con la utilización de servidores dedicados, los nodos utilizan replicación proactiva que pueden subir a estos servidores. También se utiliza un rastreador que es el encargado de indexar toda la información de los nodos en red. Orientado hacia el servicio de media streaming [10].

Shen y Zhu presentan un esquema de replicación de archivos de baja sobrecarga proactiva, Además la replicación entre nodos físicamente cercanos, la utilización del algoritmo para la redirección de consultas de archivos para nivelar las cargas y el estudio de la capacidad de los nodos en red. Orientado hacia el servicio de contenidos [11].

Ranganathan, Iamnitchi y Foster proponen mantener un nivel de umbral de la disponibilidad en todo momento. Mediante la replicación automática de los nodos y la predicción del número necesario de réplicas, donde todos los nodos tienen

incorporado un modelo guía para la creación de réplicas. Orientado hacia el servicio de cálculos computacionales [12].

Mauri propone el diseño de una arquitectura que permita tanto el intercambio de información entre todas las redes interconectadas a esta arquitectura como la compartición de recursos, contenidos o ficheros (dependiendo del caso), entre nodos de diferentes redes P2P parcialmente centralizadas. Orientado al servicio de contenidos y cálculos computacionales [13].

Herrera propone un marco de trabajo que combina replicación e incentivos para promover la disponibilidad de contenidos en sistemas P2P a larga escala. Donde el marco de disponibilidad de contenido tiene seis componentes que son el comportamiento del usuario, disponibilidad del nodo, estructura de la red, contenido, métricas de rendimiento, incentivo y redundancia. Orientado al servicio de contenidos [14].

Herrera y Znati proponen una investigación de replicación y estrategias de reparación para proveer disponibilidad de contenidos en redes P2P sobre la rotación, para proveer disponibilidad a partir de recuperar fragmentos perdidos de un contenido. Orientada hacia el servicio de contenidos [15].

Dimakopoulos, Margariti, Ntetsika y Pitoura proponen una solución de reconciliación escalable y altamente disponible para aplicaciones colaborativas P2P desarrollando un protocolo de reconciliación que garantice la consistencia eventual de las réplicas y tenga en cuenta el costo de acceso a los datos, además la escalabilidad y alta disponibilidad para el mecanismo de replicación. Orientado hacia el servicio de contenidos [16].

2. Rotación

Meng, Chen y Ding proponen un método cuantitativo para la rotación, donde se incluye una nueva capa, la cual no implicara ningún cambio en los algoritmos de enrutamiento utilizados, mediante esta capa se lleva a cabo un control sobre los nodos que se unen y abandonan la red, este control está basado en el tiempo de vida de cada nodo, para lo cual se establecen cuatro etapas basándose en el concepto de tiempo de latencia, además para mejorar la rotación se tiene una red llamada constelación en donde habrán grupos de constelaciones y cada nodo tendrá asignado un rol[17].

La disponibilidad de los recursos en una red superpuesta P2P se ve obstaculizada por la participación transitoria de los miembros y los comportamientos no colaborativos de estos, se propone un marco de trabajo eficiente centrado en

la redundancia de contenidos, actividades generales de mantenimiento bajos y reparación, además de los incentivos para mitigar el impacto de la pérdida de clientes. Con este fin, se propone un esquema de redundancia que requiere una reducida reparación de ancho de banda para mejorar la disponibilidad de contenidos. El esquema se complementa con un proceso de mantenimiento de la redundancia eficiente para automatizar las reparaciones. También se introduce un nuevo mecanismo basado en incentivos para garantizar una participación sostenida y equitativa de los nodos y un reparto justo de contenidos [18].

3. Tolerancia a fallos

Barshan, Fathy y Yousefi proponen una arquitectura basada en tres niveles, la cual provisiona la propiedad de tolerancia a fallos, ya que su principal objetivo es realizar un balance entre la seguridad, el costo, la escalabilidad y el rendimiento, esto mediante la gestión de los nodos de una red en forma de estrella, cada nodo en la red cumple un rol específico lo que ayuda a la gestión (nodo gestor, nodo gestionado o finales) [19].

Znati establece que la disponibilidad de contenido también puede ser impactada por fallos en el hardware cuando se presentan fallos en el disco, en los nodos o direccionar a información inaccesible [18].

De acuerdo a la búsqueda bibliográfica realizada no se encontraron trabajos relevantes que apoyen la disponibilidad a nivel de distintos tipos de servicios sobre redes superpuestas P2P. Aunque se han llevado a cabo muchos estudios sobre estos temas como en [20], donde se expone el problema que existe en las redes superpuestas P2P por la falta de interoperabilidad entre los diferentes tipos de redes, esto debido a que la gran mayoría de las redes superpuestas P2P se especializan en un determinado servicio, lo que hace que no sea posible que los usuarios de las diferentes redes superpuestas P2P se comuniquen entre ellas, esto repercute en la posibilidad de crecimiento de las mismas. En [17] también evidencian este problema de la versatilidad, ya que en esta investigación los autores exponen este problema pero enfocados en el problema de la rotación, dado que hay investigaciones sobre la rotación pero de forma independiente o en forma aislada dependiendo del servicio que presta cada red.

En [20] se define una arquitectura para lograr la interconexión de redes superpuestas P2P parcialmente centralizadas. Esta propuesta se basa en un nuevo sistema jerárquico para interconectar nodos de diferentes tipos de redes siempre y cuando todas ellas compartan el mismo tipo de recursos. Todas estas

investigaciones están enfocadas hacia el estudio de la versatilidad pero a nivel de la interoperabilidad de las redes superpuestas P2P, a diferencia de la presente propuesta que pretende apoyar la disponibilidad de servicios ofreciendo versatilidad a estos sobre redes superpuestas P2P no estructuradas.

BIBLIOGRAFIA

- [1] Simón, J. L., Disanzo, M., (2006), Redes Peer to Peer y Tecnología JXTA, Escuela de Ingeniería Electrónica.
- [2] "Oracle" [Online]. Available: <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html> [Accessed: 25-Mar-2015].
- [3] "Aprenderaprogramar" [Online]. Available: http://www.aprenderaprogramar.com/index.php?option=com_attachments&task=download&id=337 [Accessed: 25-Mar-2015].
- [4] "VLC-media" [Online]. Available: <http://vlc-media-player.uptodown.com/> [Accessed: 25-Mar-2015].
- [5] Finkelsteiin, A., & Kramer, J. (2000, May). Software engineering: a roadmap. In Proceedings of the conference on The future of Software Engineering (pp. 3-22). ACM.
- [6] "Ajpdsoft" [Online]. Available: <http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=461> [Accessed: 25-Mar-2015].
- [7] Dandoush, A., Alouf, S., & Nain, P. (2014). Lifetime and availability of data stored on a P2P system: Evaluation of redundancy and recovery schemes. Computer Networks, 64, 243. Retrieved from <http://search.proquest.com/docview/1519069279?accountid=41027>
- [8] Meroufel, B., & Belalem, G. (2013). Managing Data Replication and Placement based on Availability. AASRI Procedia, 5, 147-155.
- [9] Zhou, Y., Fu, T. Z., & Chiu, D. M. (2012). Server-assisted adaptive video replication for P2P VoD. Signal Processing: Image Communication, 27(5), 484-495.
- [10] Liao, X., Jin, H., & Yu, L. (2012). A novel data replication mechanism in P2P VoD system. Future Generation Computer Systems, 28(6), 930-939.

- [11] Shen, H., & Zhu, Y. (2009). A proactive low-overhead file replication scheme for structured p2p content delivery networks. *Journal of Parallel and Distributed Computing*, 69(5), 429-440.
- [12] Ranganathan, K., Iamnitchi, A., & Foster, I. (2002, May). Improving data availability through dynamic model-driven replication in large Peer-to-Peer communities. In *Cluster Computing and the Grid, 2002. 2nd IEEE/ACM International Symposium on* (pp. 376-376). IEEE.
- [13] Mauri, J. L. (2006). *Arquitectura de interconexión de redes p2p parcialmente centralizadas* (Doctoral dissertation, Universitat Politècnica de València).
- [14] Herrera-Ruiz, O. (2012). *A holistic redundancy-and incentive-based framework to improve content availability in Peer-to-Peer networks* (Doctoral dissertation, University of Pittsburgh).
- [15] Herrera-Ruiz, O., & Znati, T. (2012, January). Performance of redundancy methods in P2P networks under churn. In *Computing, Networking and Communications (ICNC), 2012 International Conference on* (pp. 327-331). IEEE.
- [16] Dimakopoulos, V., Margariti, S., Ntetsika, M., & Pitoura, E. *Data Replication in P2P Systems*.
- [17] Meng, X., Chen, X., & Ding, Y. (2013). Using the complementary nature of node joining and leaving to handle churn problem in P2P networks. *Computers & Electrical Engineering*, 39(2), 326-337.
- [18] Znati, T. (2012). *Proactive Repairs and Incentives for Content Availability in P2P Overlay Networks*. *International Journal of Computer Science & Network Security*, 12(4).
- [19] Barshan, M., Fathy, M., & Yousefi, S. (2012). Improving the availability of P2P-based network management systems by provisioning fault tolerance property. *The Journal of Supercomputing*, 61(3), 912-934.
- [20] Lloret, J. (2006). *Arquitectura de interconexión de redes P2P parcialmente centralizadas*.
- [21] Ordoñez Zambrano, J. I., Potosí Gutiérrez, G. E., Magé, P. A., (2015, Feb), *Técnica de Recuperación y Ejecución de Servicios en Redes Superpuestas Peer to Peer No Estructurada*, Universidad del Cauca.

- [22] “Caprica.” [Online]. Available: <http://caprica.github.io/vlcj/javadoc/2.1.0/> [Accessed: 20-Mar-2015].
- [23] “Informatica” [Online]. Available: <http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-FTP-como-funciona.php> [Accessed: 20-Mar-2015].
- [24] “Oracle” [Online]. Available: <https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html> [Accessed: 09-Junio-2015].
- [25] Oracle” [Online]. Available: <https://docs.oracle.com/javase/tutorial/essential/concurrency/> [Accessed: 09-Junio-2015].