

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

**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**

Monografía de Trabajo de Grado

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

## DEDICATORIA

*A Dios, por haberme permitido llegar hasta este punto y por darme siempre su infinita bondad y amor.*

*A mis Padres, Roque Bolaños y Orfa Nery Delgado, por ser el pilar fundamental en todo lo que soy, por su gran esfuerzo, sacrificio, apoyo y dedicación; ya que gracias a ellos pude sacar mi carrera adelante y pude alcanzar todas las metas que me propuse a lo largo de esta.*

*A mi hermano Fredy Bolaños y mis hermanas Yuli Bolaños y Deisy Bolaños por su apoyo incondicional y comprensión; por haberme enseñado a nunca desfallecer y siempre seguir adelante. También a mis sobrinos Julieth y Fabián por brindarme siempre su amor incondicional.*

Jenny Bolaños Delgado

*Quiero dar gracias principalmente a DIOS, quien con su infinita grandeza me guía durante el transcurso de la vida.*

*A mi hermana Magda por su apoyo incondicional, paciencia y comprensión, sin ella no hubiera sido posible.*

*A mi madre, Oliva y a mis hermanos Alejandro y Manuel, quienes siempre creyeron en mí.*

Héctor Gentil Ordoñez Delgado

## **AGRADECIMIENTOS**

*Gracias a la Universidad del Cauca, a los Ingenieros y Profesores, que nos impartieron sus conocimientos, los cuales nos formaron como Ingenieros y como personas de bien, gracias a ellos seremos excelentes profesionales.*

*Gracias a todos los que nos apoyaron a lo largo de nuestra carrera, cada uno de ellos aportó su granito de arena para que alcanzáramos este gran logro de nuestras vidas.*

*Gracias a nuestro director de trabajo de grado Pablo Augusto Magé Imbachi, quien fue el gestor y guía para llevar a cabo satisfactoriamente esta tesis.*

Jenny Bolaños Delgado

Héctor Gentil Ordoñez Delgado

## I. Tabla de Contenido

CAPÍTULO 1. INTRODUCCIÓN.....	1
CAPITULO 2. MARCO CONCEPTUAL.....	4
<b>2.1. Redes superpuestas P2P .....</b>	<b>4</b>
2.1.1. Definición de Peer.....	4
2.1.2. Definición de red superpuesta P2P.....	4
2.1.3. Clasificación de Redes superpuestas P2P .....	4
2.1.4. Tipos de arquitecturas P2P .....	5
<b>2.2. Técnica de disponibilidad en una red superpuesta P2P .....</b>	<b>8</b>
<b>2.3. Prestación de servicios .....</b>	<b>8</b>
2.3.1. Servicio .....	8
2.3.2. Prestar servicios.....	9
2.3.3. Tipos de servicio .....	9
<b>2.4. Mecanismos de disponibilidad de servicios en redes superpuestas P2P .....</b>	<b>9</b>
2.4.1. Replicación de datos.....	10
2.4.2. Rotación.....	10
2.4.3. Tolerancia a fallos .....	10
<b>2.5. Conceptos de las tecnologías a utilizar .....</b>	<b>10</b>
2.5.1. JXTA .....	10
2.5.2. VLC .....	11
2.5.3. FTP .....	11
2.5.4. Sockets .....	11
2.5.5. Procesos e hilos.....	11
<b>2.6. Metodología de desarrollo .....</b>	<b>12</b>
2.6.1. Programación Extrema (XP) .....	12
2.6.2. Principios básicos .....	12
CAPITULO 3. DISPONIBILIDAD EN REDES SUPERPUESTAS P2P.....	13
<b>3.1. Exploración de tecnicas de disponibilidad .....</b>	<b>13</b>
<b>3.2. Selección de la tecnica de disponibilidad .....</b>	<b>13</b>
<b>3.3. Caracterización de la técnica de disponibilidad .....</b>	<b>15</b>
CAPITULO 4. TECNICA PARA SOPORTE DE LA DISPONIBILIDAD DE SERVICIOS.....	21

<b>4.1. Arquitectura del sistema .....</b>	<b>21</b>
4.1.1. Sistema P2P .....	22
4.1.2. Representante.....	22
4.1.3. Despliegue servicios .....	22
4.1.4. Perfiles .....	23
4.1.5. Replicación de servicios.....	23
4.1.6. Consumidor.....	23
4.1.7. Observador .....	23
<b>4.2. Descripción de la técnica para soporte de la disponibilidad de servicios .....</b>	<b>24</b>
4.2.1. Acciones de los nodos .....	24
4.2.2. Comunicación entre los nodos cliente y servidor.....	25
4.2.3. Servicios a desplegar.....	25
4.2.4. Conexión.....	26
4.2.5. Mensajes.....	26
4.2.6. Envío de mensajes.....	26
4.2.7. Búsqueda, recuperación y disponibilidad de los servicios .....	26
4.2.8. Observador .....	26
<b>4.3. Prestación de servicios .....</b>	<b>27</b>
4.3.1. Publicación de los servicios .....	27
4.3.2. Búsqueda de los servicios .....	28
4.3.3. Recuperación del servicio .....	29
4.3.4. Replicación de los servicios .....	30
<b>CAPITULO 5. TECNOLOGIAS PARA EL DESARROLLO DE REDES SUPERPUSTAS P2P .....</b>	<b>31</b>
<b>5.1. Tecnología para el desarrollo de la red superpuesta P2P .....</b>	<b>31</b>
5.1.1. JXTA .....	32
5.1.2. ¿Por qué JXTA como tecnología de desarrollo? .....	32
5.1.3. Arquitectura JXTA.....	33
5.1.4. Protocolos .....	34
<b>5.2. Tecnología para el servidor fuente del servicio de media streaming .....</b>	<b>34</b>
5.2.1. ¿Qué es VLC? .....	34
5.2.2. ¿Por qué VLC? .....	35

5.2.3. LibVLC .....	35
5.2.4. VLCj .....	35
<b>5.3. Tecnología para el servidor fuente del servicio de contenidos .....</b>	<b>36</b>
5.3.1. ¿Qué es FTP?.....	36
5.3.2. ¿Por qué FTP?.....	36
5.3.3. Servidor FTP .....	36
5.3.4. Cliente FTP .....	37
<b>5.4. Tecnologías para el servidor fuente del servicio de cálculos computacionales .....</b>	<b>37</b>
5.4.1. ¿Qué es un socket? .....	37
5.4.2. ¿Por qué sockets? .....	37
5.4.3. ¿Qué es un Hilo? .....	38
5.4.4. ¿Por qué hilos? .....	38
Capítulo 6. DESARROLLO DEL PROTOTIPO SOFTWARE .....	39
<b>6.1. Especificación de requisitos.....</b>	<b>39</b>
6.1.1. Alcance .....	39
6.1.2. Funciones del producto.....	39
6.1.3. Características del usuario.....	40
6.1.4. Requerimientos funcionales.....	40
6.1.5. Requerimientos no funcionales.....	41
<b>6.2. Implementación.....</b>	<b>41</b>
6.2.1. Metodología de Desarrollo .....	42
6.2.2. Herramientas.....	42
6.2.3. Análisis y diseño .....	43
6.2.4. Planificación de entregas .....	56
Capítulo 7. EVALUACIÓN Y ANÁLISIS DEL PROTOTIPO SOFTWARE .....	58
<b>7.1. Diseño experimental.....</b>	<b>58</b>
7.1.1. Estudio de caso: Evaluación prototipo software .....	58
7.1.1.1. Pregunta de investigación:.....	58
7.1.1.2. Objetivo del estudio de caso:.....	58
7.1.1.3. Selección del caso de estudio:.....	58
7.1.1.4. Contexto del caso: .....	58

7.1.1.5. Diseño del estudio.....	59
7.1.1.6. Desarrollo del caso: .....	64
7.1.1.7. Resultados: .....	65
7.1.1.8. Análisis de resultados: .....	79
Capítulo 8. CONCLUSIONES Y TRABAJOS FUTUROS .....	85
<b>8.1. Conclusiones.....</b>	<b>85</b>
<b>8.2. Trabajos futuros y recomendaciones .....</b>	<b>86</b>
REFERENCIAS BIBLIOGRÁFICAS .....	87



## II. LISTA DE FIGURAS

Figura 1. Arquitectura P2P descentralizada. ....	6
Figura 2. Arquitectura P2P centralizada. ....	7
Figura 3. Arquitectura P2P parcialmente centralizada. ....	8
Figura 4. Arquitectura del Sistema .....	21
Figura 5. Sistema P2P .....	22
Figura 6. Diagrama de flujo para publicar un servicio. ....	27
Figura 7. Diagrama de flujo para búsqueda de servicios. ....	28
Figura 8. Diagrama de flujo para recuperar servicios. ....	29
Figura 9. Diagrama de flujo para la replicación de servicios. ....	30
Figura 10. Arquitectura JXTA. ....	33
Figura 11. Diagrama de casos de uso. ....	46
Figura 12. Diagrama de clases del sistema.....	48
Figura 13. Diagrama de secuencia crear nodo y conectar a la red. ....	52
Figura 14. Diagrama de secuencia publicar servicios. ....	53
Figura 15. Diagrama de secuencia replicar servicios. ....	53
Figura 16. Diagrama de secuencia prestar servicios.....	54
Figura 17. Diagrama de secuencia consumir servicios. ....	55
Figura 18. Diagrama de secuencia buscar servicios. ....	55
Figura 19. Diagrama de secuencia recuperar servicios. ....	56
Figura 20. Estructura de escenarios de prueba.....	65
Figura 21. Representación del escenario de prueba número 1.....	66
Figura 22. Representación del escenario de prueba número 2.....	67
Figura 23. Representación del escenario de prueba número 3.....	68
Figura 24. Representación del escenario de prueba número 4.....	69
Figura 25. Representación del escenario de prueba número 5.....	70
Figura 26. Representación del escenario de prueba número 6.....	71
Figura 27. Representación del escenario de prueba número 7.....	72
Figura 28. Representación del escenario de prueba número 8.....	73
Figura 29. Representación del escenario de prueba número 9.....	74
Figura 30. Representación del escenario de prueba número. 10.....	75
Figura 31. Representación del escenario de prueba número. 11.....	76
Figura 32. Representación del escenario de prueba número. 12.....	77
Figura 33. Representación del escenario de prueba número. 13.....	78
Figura 34. Representación del escenario de prueba número. 14.....	79
Figura 35: Grafica de tiempo vs número de réplicas. ....	80
Figura 36: Grafica de tiempo vs número de fallos. ....	81
Figura 37. Aproximación para una estabilidad del cincuenta por ciento y 500 réplicas. ....	84

### III. LISTA DE TABLAS

Tabla 1. Técnicas y soluciones propuestas para mejora la disponibilidad en redes superpuestas P2P. ....	13
Tabla 2. Parámetros establecidos para seleccionar la técnica.....	14
Tabla 3. Caracterización de las soluciones encontradas. ....	18
Tabla 4. Características comunes entre las diferentes soluciones de replicación.....	18
Tabla 5. Comparación de algunas aplicaciones estudiadas.....	31
Tabla 6. Priorización de historias de usuario. ....	45
Tabla 7. Definición de escenarios de pruebas.....	63
Tabla 8. Formato de registro de resultados.....	63
Tabla 9. Tabla de resultados escenarios de pruebas número. 1.....	66
Tabla 10. Tabla de resultados escenarios de pruebas número. 2.....	67
Tabla 11. Tabla de resultados escenarios de pruebas número. 3.....	68
Tabla 12. Tabla de resultados escenarios de pruebas número. 4.....	69
Tabla 13. Tabla de resultados escenarios de pruebas número. 5.....	70
Tabla 14. Tabla de resultados escenarios de pruebas número. 6.....	71
Tabla 15. Tabla de resultados escenarios de pruebas número. 7.....	72
Tabla 16. Tabla de resultados escenarios de pruebas número. 8.....	73
Tabla 17. Tabla de resultados escenarios de pruebas número. 9.....	74
Tabla 18. Tabla de resultados escenarios de pruebas número. 10.....	75
Tabla 19. Tabla de resultados escenarios de pruebas número. 11.....	76
Tabla 20. Tabla de resultados escenarios de pruebas número. 12.....	77
Tabla 21. Tabla de resultados escenarios de pruebas número. 13.....	78
Tabla 22. Tabla de resultados escenarios de pruebas número. 14.....	79
Tabla 23. Disponibilidad en la red para una estabilidad del diez por ciento (10%). ....	82
Tabla 24. Disponibilidad en la red para una estabilidad del treinta por ciento (30%). ....	82
Tabla 25. Disponibilidad en la red para una estabilidad del setenta por ciento (70%). ....	83
Tabla 26. Disponibilidad en la red para una estabilidad del noventa por ciento (90%). ....	83
Tabla 27. Aproximación en el sistema para una estabilidad del cincuenta por ciento y 500 réplicas. ....	84

## CAPÍTULO 1. INTRODUCCIÓN

Las redes superpuestas P2P<sup>1</sup> (Peer-to-Peer) han emergido como una importante alternativa al paradigma tradicional de comunicación cliente/servidor para construir sistemas distribuidos a gran escala. Las arquitecturas P2P difieren de las tradicionales arquitecturas cliente/servidor en el hecho de que todos sus participantes establecen relaciones entre iguales; es decir, todos ellos pueden comportarse tanto como clientes y como servidores. Por lo tanto, cualquier computador se puede convertir en un elemento activo de la red superpuesta P2P permitiendo intercambiar información entre los usuarios; y además aumentar su capacidad de procesamiento, pues permite que los recursos de cada nodo trabajen en forma conjunta para mejorar el procesamiento y almacenamiento de datos, además de garantizar la creación, propagación y el acceso a la información a un bajo costo y sin la necesidad de coordinar entidades dedicadas [1].

En los últimos años las redes superpuestas P2P han venido obteniendo mayor importancia, ya que se han realizado muchos estudios de gran relevancia sobre este tema; esto debido al gran crecimiento de usuarios en las redes superpuestas P2P [7]. La gran acogida que han tomado las aplicaciones desarrolladas sobre redes superpuestas P2P, han llamado la atención tanto de las comunidades académicas como también industriales. Actualmente se puede encontrar una gran variedad de aplicaciones P2P, como: media streaming, videoconferencia, vigilancia, difusión multimedia, e-learning y de almacenamiento, entre otros [9]. Mediante las cuales millones de usuarios comparten contenidos tales como: imágenes, archivos de audio, software, música, programas de televisión, juegos, películas y otros documentos [8]. Todo esto es posible mediante el desarrollo de grandes plataformas que permiten compartir todas estas clases de archivos a través de esta tecnología; como por ejemplo BitTorrent y Gnutella, las cuales son plataformas famosas que se dedican a prestar el servicio de compartir contenido. Con todos estos avances de los dispositivos de red, las redes superpuestas P2P se han convertido en una tecnología muy importante alrededor del mundo [10].

Muchas redes superpuestas P2P son construidas para proveer un único tipo de servicio. Por ejemplo, Napster brinda intercambio de archivos musicales, Gnutella brinda intercambio de archivos de cualquier tipo y AIM brinda mensajería instantánea. Dadas las diversas características de estos distintos servicios y la falta de una estructura subyacente común P2P, cada empresa desarrolladora tiende a crear sistemas incompatibles, ya que ninguno de ellos es capaz de operar con alguno de los otros. Esto significa que cada empresa crea su propia comunidad de usuarios P2P haciendo un esfuerzo mayor al desarrollar software y bases comunes a todos los sistemas P2P. Es más, para que un nodo pueda participar en múltiples comunidades organizadas por diferentes plataformas P2P, debe soportar varias implementaciones, cada una para cada sistema o comunidad [32].

A pesar de la gran acogida que han tenido las redes superpuestas P2P, también presentan varias dificultades; algunas de estas son la fiabilidad, la disponibilidad de datos

---

<sup>1</sup> P2P: Acrónimo para redes peer to peer.

y la confidencialidad [14][2]. En este trabajo nos enfocaremos en el problema de la disponibilidad, el cual se presenta debido al alto comportamiento dinámico de las redes superpuestas P2P; ya que en este tipo de redes, los nodos pueden abandonar la red en cualquier momento con mucha frecuencia [13], esto debido a que no son entidades dedicadas, si no nodos de usuarios comunes que pueden ingresar y salir de la red en cualquier momento. Estos nodos pueden ser proveedores y consumidores de recursos al mismo tiempo [14].

Actualmente y desde hace muchos años atrás se ha venido estudiando el problema de la disponibilidad de datos en redes superpuesta P2P, con el fin de buscar soluciones que ayuden a mejorarla. Una de las técnicas utilizadas para mejorar la disponibilidad es la replicación de datos, la cual ayuda a aumentar la probabilidad de que un dato esté disponible en cualquier momento [12].

Por consiguiente, es indispensable establecer una forma de prestar disponibilidad de múltiples servicios, bajo un entorno inestable, aprovechando los dispositivos ubicados en la red; de tal forma que éstos puedan interactuar entre sí, prestando con alta disponibilidad algunos de los servicios que son ofrecidos en las redes superpuestas P2P.

Lo anterior genera preguntas como:

¿Cómo se puede apoyar la disponibilidad de servicios en redes superpuestas P2P no estructuradas? Con el objetivo de dar respuesta a la anterior interrogante, se propone desarrollar una técnica que integre algunos servicios P2P y que ayude a mejorar la disponibilidad de estos, para lo cual se definió el siguiente objetivo general y cuatro objetivos específicos:

**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.

**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.

En este trabajo de grado se realizó una contribución en el área de las redes superpuestas P2P, con la especificación de una técnica de disponibilidad para la prestación de servicios sobre redes superpuestas P2P no estructuradas. En la primera parte, se desarrolló la descripción de la información relevante con respecto a las redes superpuestas P2P y las técnicas que se han utilizado para proveer disponibilidad de servicios, lo que permitió definir la técnica y las características que se tuvieron en cuenta para nuestro trabajo de grado; seguidamente se llevó a cabo la selección de la arquitectura, la cual se adecuó con la técnica, para finalmente proponer la nueva técnica que soporte la disponibilidad de servicios en redes superpuesta P2P no estructuradas.

A continuación se describe la estructura que compone este documento:

**Capítulo I. Introducción:** se presenta al lector un preámbulo del contenido del documento.

**Capítulo II. Marco Conceptual:** se encuentran las definiciones relacionadas con el contexto del problema y la investigación realizada.

**Capítulo III. Disponibilidad en redes superpuestas P2P:** se describen las diferentes técnicas de disponibilidad de servicios sobre redes superpuestas P2P, además del proceso de selección y caracterización de una de ellas.

**Capítulo IV. Técnica para soporte de la disponibilidad de servicios:** se realiza la selección de la arquitectura para redes superpuestas P2P no estructuradas y su adecuación con la técnica de disponibilidad.

**Capítulo V. Tecnologías para el desarrollo de redes superpuestas P2P:** se describen las tecnologías escogidas para la implementación de la red superpuesta P2P.

**Capítulo VI. Desarrollo del Prototipo Software:** se presenta el proceso seguido durante el desarrollo del prototipo software para la verificación de la técnica propuesta.

**Capítulo VII. Evaluación y análisis de la simulación:** se muestra la evaluación y análisis de los resultados obtenidos durante la simulación.

**Capítulo VIII. Conclusiones y trabajos futuros:** se dan las conclusiones obtenidas en el desarrollo del trabajo de grado y algunas recomendaciones para realizar trabajos a futuro.

## **CAPITULO 2. MARCO CONCEPTUAL**

A continuación, se presentan los conceptos necesarios involucrados para el soporte de la disponibilidad de servicios en redes superpuestas P2P no estructuradas, con el fin de que haya una mejor comprensión del problema.

### **2.1. Redes superpuestas P2P**

#### **2.1.1. Definición de Peer**

Un peer es un nodo que está presente en una red superpuesta P2P y que forma la unidad fundamental de procesamiento de cualquier solución P2P. Un peer es una entidad capaz de realizar algún trabajo útil y comunicar los resultados de este trabajo a otra entidad sobre la red, ya sea directa o indirectamente [35].

#### **2.1.2. Definición de red superpuesta P2P**

Es una arquitectura de red distribuida en la cual todos los participantes de la red actúan como clientes y servidores al mismo tiempo; donde los participantes comparten una parte de sus propios recursos de hardware (potencia de procesamiento, capacidad de almacenamiento, ancho de banda, entre otros.). Estos recursos compartidos son necesarios para proporcionar el servicio y el contenido ofrecido por la red (por ejemplo, el intercambio de archivos o espacios de trabajo compartidos para la colaboración). Dichos recursos son accesibles por otros Peers<sup>2</sup> directamente, sin pasar por entidades intermediarias. Los participantes de una red superpuesta P2P de este tipo pueden solicitar recursos (servicios y contenidos) como proveedores, así como también pueden ser solicitantes de recursos (servicios y contenidos) como consumidores [15].

#### **2.1.3. Clasificación de Redes superpuestas P2P**

Las redes superpuestas P2P se pueden clasificar de acuerdo a los siguientes criterios:

- En primer lugar, las redes superpuestas P2P se pueden distinguir entre las que necesitan de una entidad central para proporcionar los servicios ofrecidos y las que no lo necesitan de la siguiente manera:

##### **Red superpuesta P2P pura**

Una arquitectura de red distribuida puede ser clasificada como una red superpuesta P2P pura, si es en primer lugar una red superpuesta P2P según la primera definición;

---

<sup>2</sup> Peer: hace referencia a todos los equipos participantes dentro de una red superpuesta P2P.

y en segundo lugar, si cualquier entidad escogida aleatoriamente se puede quitar de la red sin que la red sufra ninguna pérdida de servicio [15].

### **Red superpuesta P2P centralizada**

Este tipo de red está formada por nodos distribuidos que tienen la misma función dentro de la red donde existe un elemento servidor necesario para el funcionamiento de la red. La transferencia de información se realiza entre los clientes finales, no teniendo que intervenir dicho servidor central para realizar la transferencia. La diferencia radica en la forma de búsqueda de nodos activos y contenidos en la red [1].

### **Red superpuesta P2P híbrida (Parcialmente centralizada)**

Una arquitectura de red distribuida puede ser clasificada como una red superpuesta P2P híbrida, si es en primer lugar una red superpuesta P2P según la primera definición; y en segundo lugar, una entidad central es necesaria para proporcionar parte de los servicios de red ofrecidos [15].

- En segundo lugar, las redes superpuestas P2P se pueden distinguir de acuerdo a su topología de red de la siguiente manera:

### **Redes superpuestas P2P estructuradas**

Son redes que no solo implican una topología ya definida; si no también un conjunto de estrictas acciones del protocolo para que cada nodo lleve a cabo con el fin de mantener la red [3]. La topología es estrechamente controlada y la información no es colocada al azar en los nodos, si no en lugares específicos; con lo cual se podrán realizar consultas más eficientes [19].

### **Redes superpuestas P2P no estructuradas**

Son redes que se componen de pares que se unen a la red sin ninguna regla, sin ningún conocimiento previo de la topología. La red utiliza el mecanismo de inundación para enviar consultas a través de la superposición de nodos con un alcance limitado, cuando un nodo que tiene la información recibe la solicitud responde a la consulta [19]. En este tipo de redes las conexiones de cada nodo son autónomas; decidiendo automáticamente quién y cuándo conectarse a otros nodos. Por lo cual no hay un diseño formal de la red [3].

## **2.1.4. Tipos de arquitecturas P2P**

Las redes superpuestas P2P tienen la capacidad para aprovechar diversos recursos (computacionales, almacenamiento y ancho de banda) con menos coste de propiedad y también pueden disfrutar de diversas características deseables como la escalabilidad, la autonomía, entre otros [31]. Ya que la arquitectura de cualquier sistema es la columna principal para la construcción de las aplicaciones que se

implementan sobre él, es crucial el entendimiento de la arquitectura P2P. Tal estudio es muy beneficioso porque: (a) Ayuda a los investigadores, desarrolladores y usuarios para entender mejor las relaciones y diferencias entre redes superpuestas P2P y otros paradigmas de computación distribuida (por ejemplo, cliente-servidor y computación distribuida). (b) Permite ser conscientes de las ventajas potenciales de la computación P2P para demandas de las aplicaciones emergentes, y para determinar la arquitectura más adecuada para ellos. (c) Permite determinar los factores arquitectónicos que son críticos para el desempeño de una red superpuesta P2P como la escalabilidad, confiabilidad, y otras características importantes [30].

Las arquitecturas P2P permiten crear redes superpuestas P2P puras, híbridas y centralizadas. En las puras todos los nodos tienen las mismas funciones y ningún nodo de la red es indispensable para el funcionamiento de la red superpuesta P2P; en las híbridas algunos nodos tienen funciones especializadas y son necesarios para el funcionamiento de la red superpuesta P2P y en cambio en las redes centralizadas existe un servidor que es totalmente necesario para el funcionamiento de la red superpuesta P2P.

#### 2.1.4.1. Arquitecturas P2P puras

En este tipo de arquitecturas todos los nodos tienen las mismas funciones y responsabilidades dentro de la red superpuesta P2P, además no hay ningún servidor centralizado, en este tipo de red el sistema puede funcionar sin problemas cuando algún nodo sale de la red [31]. Los nodos pueden realizar peticiones de información a otros nodos y al mismo tiempo resolver y contestar peticiones de sus compañeros, consiguiendo así una comunicación asimétrica y totalmente descentralizada. De este modo los nodos en este tipo de red pueden cumplir tres roles dentro de la red: el de cliente, cuando realizan solicitud de información; el de servidor, cuando resuelve y da respuesta a peticiones de sus compañeros y el de encaminador, cuando se encuentra de intermediario entre dos nodos y transmite la información de uno a otro [30].

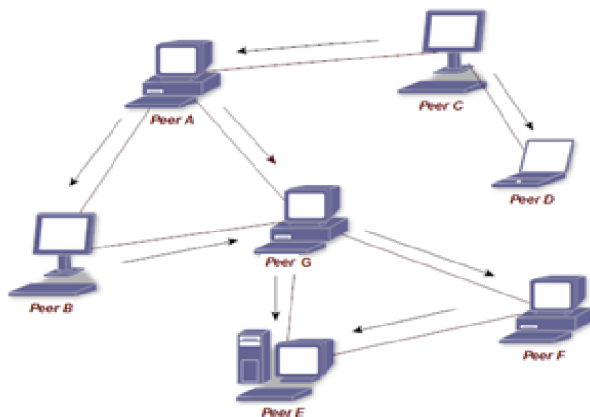


Figura 1. Arquitectura P2P descentralizada.



### 2.1.4.2. Arquitecturas P2P centralizadas

En este tipo de arquitectura se hace uso de un servidor central quien es el encargado de proporcionar los servicios; no todos los nodos son de la misma categoría ni desempeñan las mismas funciones. Estas arquitecturas se pueden considerar P2P, puesto que para ciertas funciones (como por ejemplo: pedido y transferencia de contenidos), los nodos se comunican entre sí de forma directa [30].

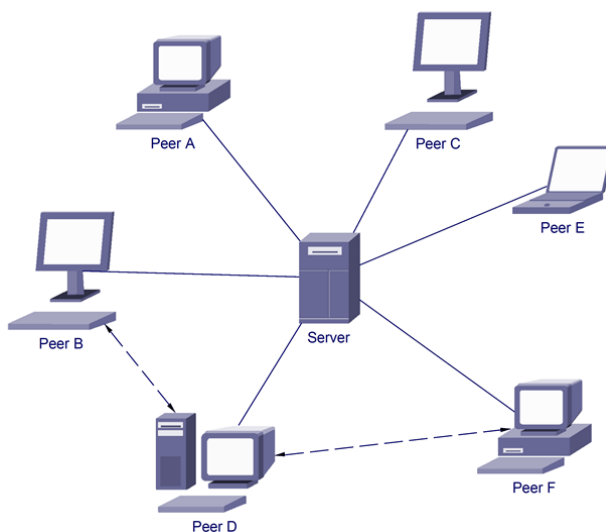


Figura 2. Arquitectura P2P centralizada.

### 2.1.4.3. Arquitecturas P2P híbridas

Este tipo de arquitectura es similar a la arquitectura descentralizada, pero con la diferencia que existen una serie de nodos llamados “supernodos”, que tienen más capacidad de procesamiento y ancho de banda. Estos “supernodos” transmiten las búsquedas, que son pedidas por los nodos, a otros “supernodos”. En este modelo las peticiones son enviadas a los supernodos, pues se considera que tienen mayores posibilidades de encontrar la petición. Aquellos clientes con menos ancho de banda serán los clientes de los “supernodos”. Este tipo de red superpuesta P2P utiliza un algoritmo de control de flujo de envío de peticiones y respuestas así como un esquema de prioridades en caso de necesitar desechar mensajes [30].

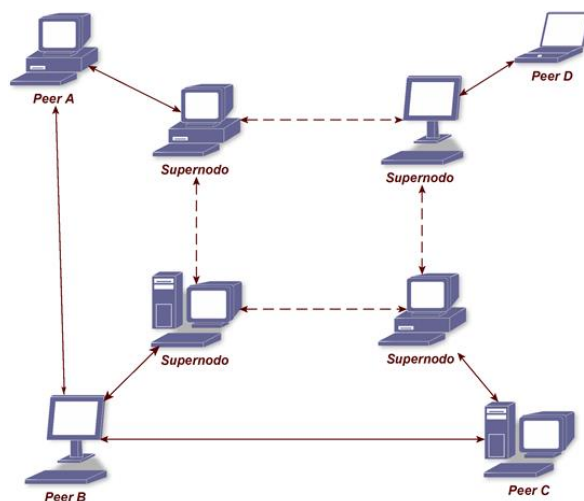


Figura 3. Arquitectura P2P parcialmente centralizada.

## 2.2. Técnica de disponibilidad en una red superpuesta P2P

Se debe entender como técnica, un procedimiento o conjunto, que sirven en el campo de la ciencia, de la tecnología, del arte, de la educación o en cualquier otra actividad [36].

La disponibilidad dentro de una red superpuesta P2P es una de las características más importantes que se debe garantizar debido a la inestabilidad de red. La disponibilidad se define como la facilidad de obtener acceso a un recurso que se necesita, el cual puede ser un dato o un nodo particular; usualmente es definida como una probabilidad [3].

## 2.3. Prestación de servicios

### 2.3.1. Servicio

El término servicio, debe entenderse como una parte diferente de un sistema de computadores, que gestiona una colección de recursos relacionados y presenta su funcionalidad a los usuarios y aplicaciones. El único acceso que tenemos al servicio es mediante un conjunto de operaciones que él ofrece; por ejemplo, un servicio de archivos proporciona las operaciones de lectura, escritura y borrado en los archivos. Los servicios pueden proporcionar información, realizar una acción, o controlar un recurso en nombre de otra entidad. Un servicio puede ser implementado como software, hardware, o una combinación de hardware y software. Además, se define como una funcionalidad predefinida que puede ser utilizada por los nodos que están conectados a una red superpuesta P2P [39].

### **2.3.2. Prestar servicios**

El término prestar, significa entregar algo a alguien para que lo utilice durante algún tiempo y después lo devuelva.

De acuerdo a la definición anterior, se puede concluir que el término prestación de servicios, significa entregar una funcionalidad y los recursos asociados a ella, para que pueda ser utilizada durante algún tiempo, mediante un conjunto de operaciones por los nodos conectados a una red superpuesta P2P [35].

### **2.3.3. Tipos de servicio**

Para el desarrollo de este trabajo de grado se seleccionaron tres tipos de servicios, los cuales fueron elegidos de acuerdo a la investigación bibliográfica realizada, que más adelante en la sección 3.3. se presenta con mas detalle. A continuación se presenta la definición de estos tipos de servicio.

- **Media streaming**

El término se refiere a los productos y técnicas utilizadas para la transmisión de contenidos multimedia tales como audio y video a través de una red. El nombre se refiere al método de envío, más que al medio como tal. El método se caracteriza por la posibilidad de reproducir los contenidos en el cliente sin la necesidad de descargar la totalidad del contenido o incluso mientras este se está generando en tiempo real; por ejemplo, usando una cámara web. Esto es posible gracias a que una vez iniciada la transferencia, el programa cliente puede empezar a desplegar contenido al usuario y al mismo tiempo continuar recibiendo el flujo de datos [26].

- **Contenido**

Se refiere a archivos con distintas características que se pueden compartir en internet tal como: imágenes, audio, software, música, programas de televisión, juegos, películas y entre otros [1][8].

- **Cálculos computacionales**

Se refiere a la capacidad de compartir la carga de procesamiento para realizar cálculos entre muchas máquinas descentralizadas, además de distribuir dichos procesos en términos de los participantes [3].

## **2.4. Mecanismos de disponibilidad de servicios en redes superpuestas P2P**

A continuación, se presentan las definiciones de los mecanismos o técnicas de disponibilidad de servicios sobre redes superpuestas P2P encontradas durante la revisión bibliográfica.

### **2.4.1. Replicación de datos**

Es una técnica de creación de copias idénticas de datos (archivos, bases de datos, entre otros.) en sitios distribuidos geográficamente, cada copia es llamada una réplica [16, 17]. Esta técnica es utilizada para mejorar la disponibilidad de los datos en redes superpuestas P2P, en la cual existen propuestas que intentan mejorarla cada vez más utilizando nuevos métodos de replicación [12].

### **2.4.2. Rotación**

Es una propiedad inherente de las redes superpuestas P2P; se define como el efecto colectivo de una conducta no coordinada de los usuarios, que se produce cuando un gran porcentaje de los nodos se unen y abandonan con frecuencia la red superpuesta P2P [2]. La rotación ha sido un tema de investigación y un problema complicado de resolver; por lo tanto, se han realizado distintas investigaciones alrededor de este tema, pero todas han sido enfocadas hacia alguna arquitectura de red o contexto específico, lo que ha llevado a la falta de versatilidad en las redes superpuestas P2P.

### **2.4.3. Tolerancia a fallos**

Es la capacidad de un sistema para ofrecer servicios, incluso cuando se presentan fallos. Es uno de los requisitos de las redes superpuestas P2P, con el fin de evitar la pérdida de datos y apoyar la entrega de mensajes correctos. La tolerancia a fallos es otra de las técnicas utilizadas para mejorar la disponibilidad de los datos en una red superpuesta P2P [11].

## **2.5. Conceptos de las tecnologías a utilizar**

A continuación, se presentan los principales conceptos de las tecnologías utilizadas. En el capítulo 5 se describe con más detalle cada una de las tecnologías y por qué fueron seleccionadas.

### **2.5.1. JXTA**

Es una tecnología que consiste en un grupo de protocolos abiertos (open source) para construir redes virtuales P2P. Estos protocolos JXTA definen los requerimientos de red mínimos para que los pares se formen y se unan a una red virtual mediante una capa de red genérica. La tecnología JXTA permite a los dispositivos de la red comunicarse, colaborar y compartir recursos [40].

### **2.5.1.1. Servicios JXTA**

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 nodos pueden descubrir éste servicio y hacer uso de él. Cada servicio tiene una ID única y un nombre que consiste en un string nombre y una serie de claves descriptivas que identifican unívocamente al servicio. Los servicios JXTA expanden las propiedades del núcleo y facilitan el desarrollo de aplicaciones. En esta capa se proveen mecanismos de búsqueda, intercambio, indexado, obtención de código y contenido [32].

### **2.5.2. VLC**

Es un programa simple, escrito entorno a la librería libvlc. Este 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 [39].

### **2.5.3. 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 [45].

### **2.5.4. Sockets**

Es un punto final de un enlace de comunicación bidireccional entre dos programas que se ejecutan en la red. Este está unido a un número de puerto de modo que la capa de TCP puede identificar la aplicación y los datos están destinados a ser enviados [47].

### **2.5.5. Procesos e hilos**

Un proceso es un mecanismo en un sistema operativo que puede ejecutar una serie de pasos. Algunos sistemas operativos utilizan los términos trabajo o tarea. Este normalmente tiene su propia área de memoria privada en la que se ejecuta. Un hilo 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 [48].

## 2.6. Metodología de desarrollo

En esta sección se describe la metodología de desarrollo utilizada para la implementación de este proyecto.

### 2.6.1. Programación Extrema (XP)

Es una metodología de desarrollo ligera (o ágil) basada en una serie de valores y de prácticas de buenas maneras que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas [50].

La programación extrema se basa en una serie de reglas y principios que se han ido gestando a lo largo de toda la historia de la ingeniería del software. Usadas conjuntamente proporcionan una nueva metodología de desarrollo software que se puede englobar dentro de las metodologías ligeras, que son aquéllas en la que se da prioridad a las tareas que dan resultados directos [49].

### 2.6.2. Principios básicos

La Programación Extrema se basa en 12 principios básicos agrupados en cuatro categorías [50]:

1. **Retroalimentación a escala fina.**
  - El principio de pruebas
  - Proceso de planificación
  - El cliente en el sitio
  - Programación en parejas
2. **Proceso continuo en lugar de por lotes**
  - Integración continua
  - Refactorización
  - Entregas pequeñas
3. **Entendimiento compartido**
  - Diseño simple
  - Metáfora
  - Propiedad colectiva del código
  - Estándar de codificación
4. **Bienestar del programador**
  - La semana de 40 horas

Este proyecto para trabajo de grado se desarrollara en el contexto de las redes superpuestas P2P no estructuradas y descentralizadas.

## CAPITULO 3. DISPONIBILIDAD EN REDES SUPERPUSTAS P2P

En este capítulo se presenta la descripción de las diferentes técnicas de disponibilidad de servicios sobre redes superpuestas P2P no estructuradas; además, el proceso de selección y caracterización de una de ellas.

### 3.1. Exploración de técnicas de disponibilidad

Mediante la técnica de búsqueda sistemática se realizó la búsqueda de las investigaciones acerca de la disponibilidad sobre redes superpuestas P2P, por medio de la cual se pudo identificar diferentes investigaciones que proponen soluciones para tratar de mejorarla; en las cuales las técnicas que más interés han despertado a la comunidad investigativa y científica son: la replicación, la tolerancia a fallos y la rotación. En la Tabla 1 se presentan las técnicas encontradas, junto con algunas de las soluciones que se han propuesto en las diferentes investigaciones estudiadas:

Técnicas	Soluciones
Tolerancia a fallos	<ul style="list-style-type: none"><li>• Reparación de ancho de banda [18][30][43].</li><li>• Fallas en el disco o nodo [2].</li><li>• Errores de lectura [2].</li><li>• Fallo en la transmisión de los datos [2].</li><li>• Redundancia de software [11].</li></ul>
Replicación	<ul style="list-style-type: none"><li>• DHT<sup>3</sup> y esquemas de redundancia [16][42].</li><li>• Borrado de código [14].</li><li>• Reparación proactiva [41][12][4].</li><li>• Utilización de supernodos [43][30][4][18].</li><li>• Creación de réplicas automáticamente [12].</li></ul>
Rotación	<ul style="list-style-type: none"><li>• Mecanismos basados en incentivos [21].</li><li>• Introducción de una capa lógica llamada Dechurn [7].</li></ul>

Tabla 1. Técnicas y soluciones propuestas para mejorar la disponibilidad en redes superpuestas P2P.

### 3.2. Selección de la técnica de disponibilidad

Con el objetivo de seleccionar la técnica apropiada para ser implementada en el desarrollo de este trabajo, se realizó una búsqueda bibliográfica exhaustiva en las bases de datos disponibles en la página de la biblioteca de la Universidad del Cauca (<http://biblio.unicauca.edu.co>), como por ejemplo, ProQuest y ScienceDirect, además de la página de internet [www.sci-hub.org](http://www.sci-hub.org). Para esta búsqueda se utilizaron palabras

---

<sup>3</sup> DHT: Acrónimo de Distributed Hash Tables (Tablas hash distribuidas): es una estructura para localizar recursos donde se almacena su clave y valor.

claves relacionadas con el tema de investigación: disponibilidad P2P, mecanismos de disponibilidad Peer-to-Peer, técnica de disponibilidad redes P2P, disponibilidad redes P2P, disponibilidad redes Peer-to-Peer, entre otras. También se tuvieron en cuenta dos filtros:

- Año de publicación, el cual se estableció en el rango del año 2011 hasta el año actual.
- Autores reconocidos, para este fin se realizó el estudio bibliográfico de cada uno de los autores, con el fin de seleccionar estudio de relevancia.

Con esta revisión bibliográfica se pudo identificar las tres técnicas ya mencionadas como: rotación, tolerancia a fallos y replicación. De esta búsqueda bibliográfica se establecieron dos parámetros (productividad, tipo de servicio) a tener en cuenta para poder seleccionar la técnica más adecuada; en la cual la productividad fue calculada como un porcentaje del total de investigaciones seleccionadas con respecto al número en donde exponían soluciones utilizando dicha técnica y los tipos de servicio hacia los cuales estaba enfocada cada técnica. En la Tabla 2 se muestran los parámetros establecidos para este fin. Para más información en el CD, en el documento de Anexos en el Anexo H se presenta un resumen de la literatura encontrada producto mediante el cual se obtuvo la Tabla 2.

<b>Técnicas</b>	<b>Productividad</b>	<b>Tipo servicio</b>
Tolerancia a fallos	10%	Contenidos. Gestión de redes.
Replicación	70%	Contenidos. Media streaming.
Rotación	20%	Contenidos. Media streaming. Cálculos computacionales.

Tabla 2. Parámetros establecidos para seleccionar la técnica.

De acuerdo con la tabla anterior se puede observar que la técnica más utilizada para tratar de mejorar la disponibilidad en redes superpuestas P2P no estructuradas, es la replicación; ya que es la técnica donde más producción investigativa se ha llevado a cabo; además es la que mejores resultados a dado en las diferentes investigaciones estudiadas y esta se enfoca hacia tres tipos de servicios muy populares. Con base en estos criterios de selección, se escogió esta técnica para ser implementada en este trabajo de grado.



### 3.3. Caracterización de la técnica de disponibilidad

Para caracterizar la técnica de disponibilidad se llevó a cabo el estudio de los trabajos relacionados con la técnica escogida. En la Tabla 3 se presenta el resumen de algunas soluciones que se han propuesto para esta técnica en las diferentes investigaciones encontradas durante la revisión bibliográfica.

Id	Título Solución	Tipo servicio	Objetivo	Descripción de la Topología propuesta	Año (publicación)	Características principales
S1	Vida y disponibilidad de los datos almacenados en un sistema P2P: Evaluación de esquemas de redundancia y recuperación.	Contenidos, aplicaciones de voz IP, copia de seguridad y almacenamiento.	El objetivo de este trabajo es el desarrollo de modelos matemáticos para evaluar las métricas de rendimiento fundamentales (vida y disponibilidad de datos) de P2PSS (sistemas de copia de seguridad).	Utiliza una topología de red Centralizada ya que necesita de un único servidor quien realiza la recuperación de múltiples pérdidas a la vez.	2014	<ul style="list-style-type: none"> <li>– Utilización de borrado de código y redundancia de datos.</li> <li>– Utilización de sistemas de recuperación centralizada.</li> </ul>
S2	Gestión de la replicación de datos y la ubicación en base a la disponibilidad.	No especificado	El objetivo de este trabajo es proponer 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.	Se utiliza una estructura jerárquica donde el nodo raíz llamado cluster-head es quien tiene los metadatos y la información de las replicas de cada grupo y es el encargado de gestionar cada grupo de nodos normales.	2013	<ul style="list-style-type: none"> <li>– Utiliza la distribución geográfica de los nodos y la popularidad de los datos.</li> <li>– Ubicación de réplicas.</li> <li>– Tabla de enrutamiento.</li> <li>– Comportamiento predictivo de los nodos (fracasos).</li> <li>– Creación del mínimo de réplicas posible</li> <li>– Utilización de un algoritmo para determinar el mejor candidato para colocar replicas.</li> </ul>
S3	Replicación de video adaptativo asistido por servidor para P2P VoD.	Media streaming.	En este trabajo se presenta un enfoque híbrido, donde se considera el problema de replicación como un problema de control adaptativo.	Topología centralizada ya que utiliza un servidor central que controla la replicación.	2012	<ul style="list-style-type: none"> <li>– Se describen tres modelos: un modelo de solicitud de video, un modelo de programación de servicios y un modelo de control adaptativo.</li> <li>– Algoritmo servidor: calcula el número de copias aumentar y disminuir.</li> <li>– Algoritmo de replicación homogénea: define el ancho de banda necesita para la replicación</li> <li>– Algoritmo de replicación Heterogénea: calcula el número de réplicas para cada nodo.</li> </ul>

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

Id	Título Solución	Tipo servicio	Objetivo	Descripción de la Topología propuesta	Año (publicación)	Características principales
S4	Un nuevo mecanismo de replicación de datos en el sistema P2P VoD.	Media streaming.	Se propone un mecanismo de replicación de datos proactiva del lado del cliente.	La topología esta compuesta por varios servidores dedicados, un rastreador, un servidor web y los nodos que se encuentran en red.	2011	<ul style="list-style-type: none"> <li>- utilización de servidores dedicados.</li> <li>- Los nodos utilizan replicación proactiva que pueden subir a los servidores dedicados.</li> <li>- Utilización de rastreador quien es el encargado de indexar toda la información de los nodos en red.</li> </ul>
S5	Un esquema de replicación proactiva baja sobrecarga de archivos para redes de distribución de contenidos P2P estructuradas.	Contenidos.	Se presenta un esquema de replicación de archivos de baja sobrecarga proactiva.	La topología consiste en la agrupación de supernodos (alta capacidad y rápida conexión) en forma de acorde y nodos regulares los cuales son gestionados por los supernodos.	2009	<ul style="list-style-type: none"> <li>- Replicación proactiva.</li> <li>- Replicación entre nodos físicamente cercanos.</li> <li>- Utiliza algoritmo para la redirección de consultas de archivos para nivelar las cargas.</li> <li>- Estudio de la capacidad de los nodos en red.</li> </ul>
S6	Mejorar la disponibilidad de datos a través de la replicación dirigido por modelos dinámicos en grandes comunidades Peer-to-Peer.	Cálculos computacionales.	El objetivo es mantener un nivel de umbral de la disponibilidad en todo momento.	La topología consiste en una red descentralizada y no estructurada donde todos los nodos son iguales y tienen las mismas funciones.	2002	<ul style="list-style-type: none"> <li>- Replicación automática de los nodos.</li> <li>- Predicción del número necesario de réplicas.</li> <li>- Todos los nodos tiene incorporado un modelo guía para la creación de réplicas.</li> <li>- Enfoque basado en modelos.</li> </ul>
S7	Reparaciones proactivas e incentivos para la disponibilidad de contenido en redes P2P superpuestas.	Contenidos.	El objetivo es promover la disponibilidad, escalabilidad y la auto-organización de una red superpuesta P2P.	La topología propuesta consiste en cuatro tipos de nodos que pueden ser: titular, índice, objetivo, solicitante. El nodo titular hospeda completa o parcialmente copias de los contenidos; los nodos índice son los encargados de evaluar la disponibilidad de estos contenidos y desencadenan las reparaciones necesarias; los nodos objetivo son nodos que estén dispuestos a acoger nuevas réplicas o fragmentos para los contenidos que no se almacena ninguna información; los nodos solicitantes son los nodos que deseen descargar un contenido de la red.	2012	<ul style="list-style-type: none"> <li>- Utilización de épocas para llevar a cabo la replicación.</li> <li>- Se propone un esquema de redundancia llamado reparación proactiva junto con un modelo analítico para disponibilidad de fragmentos.</li> <li>- Se propone un mecanismo de incentivos.</li> <li>- Utiliza un entorno P2P basado en DHT.</li> <li>- Proponen un algoritmo distribuido de auto organización.</li> </ul>

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

Id	Título Solución	Tipo servicio	Objetivo	Descripción de la Topología propuesta	Año (publicación)	Características principales
S8	Arquitectura de interconexión de redes P2P parcialmente centralizadas.	Redes P2P de compartición de contenidos, de computación distribuida.	Diseñar una arquitectura que permita tanto el intercambio de información entre todas las redes interconectadas a esta arquitectura como la compartición de recursos, o ficheros (dependiendo del caso), entre nodos de diferentes redes P2P parcialmente centralizadas.	La topología propuesta consiste en una arquitectura P2P parcialmente centralizada. La topología de la arquitectura diseñada y desarrollada es dinámica, es decir, cambia en función de la disponibilidad de conexiones de los nodos hacia otros nodos, según la capacidad de procesamiento que tenga disponible y el ancho de banda del nodo.	2006	<ul style="list-style-type: none"> <li>- Usa identificadores en vez de direcciones IP.</li> <li>- La arquitectura de interconexión se implementa sobre redes P2P parcialmente centralizadas que comparten el mismo tipo de recursos.</li> <li>- Se define una arquitectura propia para el sistema además de algoritmos y métodos para disponibilidad de servicios.</li> <li>- Utiliza algunos o todos los supernodos de las otras redes.</li> <li>- La arquitectura se plantea usando tres capas de organización, distribución y acceso.</li> </ul>
S9	Un marco integral de redundancia basado en incentivos para mejorar la disponibilidad de contenidos en las redes peer-to-peer.	Contenidos.	El objetivo es proponer un marco de trabajo que combina replicación e incentivos para promover la disponibilidad de contenidos en una red superpuesta P2P a larga escala.	La arquitectura es modelada en tres capas que son: intercambio de datos, manejo de redundancia y núcleo.  Los roles de los nodos pueden ser: índice, titular y candidato.	2012	<ul style="list-style-type: none"> <li>- 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.</li> <li>- Se utiliza DHT.</li> <li>- Utiliza un modelo basado en agentes para la interacción de las capas.</li> <li>- Se incluye un algoritmo distribuido de auto organización.</li> <li>- Promueve un mecanismo basado en incentivos para promover la participación de los nodos.</li> <li>- Replicación proactiva.</li> <li>- Fragmentación de réplicas.</li> </ul>
S10	Rendimiento de los métodos de redundancia en redes P2P Bajo rotación.	Contenidos.	El objetivo de esta investigación es en 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.	La topología obedece a redes ya implementadas tales como Modelnet y Bamboo.	2012	<ul style="list-style-type: none"> <li>- Fragmentación de réplicas.</li> <li>- Se utilizan 3 estrategias de borrado de código.</li> <li>- Máxima distancia separable (MDS), Mínimo ancho de banda de regeneración exacta (e-MBR), redundancia proactiva (PR).</li> <li>- Emplean una estructura de red emulada.</li> <li>- Utilizan una arquitectura DHT con extensiones para manejo de rotación.</li> </ul>

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

Id	Título Solución	Tipo servicio	Objetivo	Descripción de la Topología propuesta	Año (publicación)	Características principales
S11	Replicación de datos en sistemas P2P.	Contenidos.	El objetivo es proveer una solución de reconciliación escalable y altamente disponible para aplicaciones colaborativas P2P desarrollando un protocolo de reconciliación que garantice la consistencia de réplicas y tenga en cuenta el costo de acceso de datos, la escalabilidad y alta disponibilidad.	Tienen en cuenta la distancia entre nodos en términos de tiempos de latencia para el establecimiento de la topología de la red. Como resultado, los mensajes pueden ser enviados de manera más eficiente desde nodos vecinos que están físicamente cerca.	2007	<ul style="list-style-type: none"> <li>- Se basa en la arquitectura APPA (Atlas Peer-to-Peer Architecture)</li> <li>- Elaboran un algoritmo para reconciliación semántica llamado DSR.</li> <li>- Mecanismo que garantiza la consistencia entre replicas.</li> <li>- Persistencia de datos se utiliza PDM (Persistent Data Management) y KSR (Key-based Storage and Retrieval) con el fin de almacenar y recuperar objetos de datos.</li> <li>- Utiliza supernodos y DHT's</li> <li>- Replicación completa.</li> </ul>

Tabla 3. Caracterización de las soluciones encontradas.

Al analizar las soluciones caracterizadas en la Tabla 3, se puede evidenciar en la columna denominada "tipos de servicios", que la mayor productividad investigativa encontrada está orientada a tres tipos de servicio principales: contenidos, cálculos computacionales y media streaming; criterio por el cual la técnica para el soporte de disponibilidad de servicios propuesta en este trabajo de grado, será orientada hacia dichos servicios ya mencionados.

En la Tabla 4 se presentan las características comunes más relevantes encontradas en las diferentes soluciones de replicación. Con esta tabla se pretende seleccionar una de las soluciones o una mezcla de características, con el fin de diseñar la técnica para garantizar el soporte de disponibilidad de servicios en redes superpuestas P2P no estructuradas. El criterio para esta selección será el hecho de que la solución o conjunto de características sean comunes o se puedan adaptar a los servicios seleccionados.

Id Solución	Borrado de código	Replicación completa	Reparación proactiva	Supernodos	DHT	Algoritmos de automatización	Ancho de banda	Costo de reparación	Servidores centralizados
S1	✓		✓						
S2					✓	✓			
S3					✓				✓
S4			✓						✓
S5			✓	✓		✓	✓		
S6			✓						
S7			✓	✓	✓	✓			
S8				✓			✓	✓	
S9			✓	✓	✓	✓		✓	
S10	✓				✓	✓	✓	✓	
S11		✓		✓	✓	✓	✓	✓	

Tabla 4. Características comunes entre las diferentes soluciones de replicación.

En la Tabla 4, se puede evidenciar que no es posible seleccionar una de las soluciones, ya que cada una de estas presenta diferentes características las cuales no son comunes, ni se pueden adaptar a todos los tipos de servicios seleccionados; por lo tanto, se elegirá una mezcla de las características de esta tabla, de la siguiente manera:

- **Borrado de código:** consiste en la fragmentación de contenidos con el fin de almacenar fragmentos en diferentes nodos, para que estos estén disponibles en la red superpuesta P2P y haya múltiples fuentes del contenido [14]. Esta característica no será tomada en cuenta para el trabajo de grado, ya que está diseñada para dar soporte a un solo tipo de servicio, en este caso contenidos.
- **Replicación completa:** consiste en almacenar múltiples copias de contenidos en diferentes nodos [18]. Esta característica si será tomada en cuenta para nuestro trabajo de grado adecuándola para brindar disponibilidad de múltiples servicios.
- **Reparación proactiva:** en esta característica los clientes deciden que contenido, cuantas réplicas y en donde replicar [12][41][14][43]. Esta no será tomada en cuenta para nuestro trabajo de grado ya que se sale del alcance.
- **Supernodos:** consiste en utilizar algunos nodos con diferentes funciones a los otros nodos para llevar a cabo búsquedas, enrutamiento, optimizaciones, entre otras [43]. Esta no será tomada en cuenta para nuestro trabajo de grado ya que trabajaremos sobre una red superpuesta P2P descentralizada.
- **Uso de mecanismos de apoyo a la búsqueda:** consiste en utilizar tablas hash distribuidas para conocer la estructura de la red superpuesta P2P y la ubicación de los servicios que cada nodo presta [42][16]. Esta característica no será tomada en cuenta para nuestro trabajo de grado ya que trabajaremos sobre una red superpuesta P2P no estructurada.
- **Algoritmos de automatización:** consiste en utilizar algoritmos que automatizan tareas, tales como: uso de ancho de banda, enrutamiento, búsquedas, balanceo de carga, consistencia, entre otras [16][43]. Esta característica no será tomada en cuenta para nuestro trabajo de grado ya que se sale del alcance.
- **Ancho de banda:** consiste en seleccionar el nodo con mayor velocidad de transferencia de datos para llevar a cabo funciones como: búsquedas, replicación, enrutamiento, prestación de servicios, entre otras [5][43]. Esta característica no será tomada en cuenta para nuestro trabajo de grado ya que se sale del alcance.
- **Costo de reparación:** es usada para calcular el costo del proceso de reparar un archivo o un fragmento perdido en la red superpuesta P2P; el costo puede ser en tiempo, ancho de banda y procesamiento [1][4][18]. Esta característica no será tomada en cuenta para nuestro trabajo de grado ya que se sale del alcance.

- **Servidores centralizados:** en esta característica se hace uso de un servidor central para realizar algunas tareas dentro de la red superpuesta P2P, tales como: búsqueda de servicios y enrutamiento de los nodos [41][42]. Esta no será tomada en cuenta para nuestro trabajo de grado ya que trabajaremos sobre una red superpuesta P2P descentralizada.

En este capítulo se seleccionó la técnica de replicación que se implementó en este trabajo de grado y se llevó a cabo la caracterización de la técnica, se eligió la característica denominada replicación completa. Además se seleccionaron los tipos de servicio a prestar sobre la red superpuesta P2P: Contenidos, Cálculos computacionales y Media streaming.

## CAPITULO 4. TECNICA PARA SOPORTE DE LA DISPONIBILIDAD DE SERVICIOS

En este capítulo se describe el proceso para la selección de la arquitectura para redes superpuestas P2P no estructuradas y su adecuación con la técnica de disponibilidad seleccionada en el capítulo anterior.

Luego de haber llevado a cabo la investigación de las características y los beneficios de las redes superpuestas P2P; además de las diferentes técnicas utilizadas para proveer disponibilidad de servicios y tipos de servicios; y finalmente habiendo caracterizado la técnica de replicación para definir las características más importantes y cuáles de estas serán tenidas en cuenta para el desarrollo del trabajo de grado, se procede a describir la adecuación de la técnica para soporte de la disponibilidad de los servicios seleccionados.

### 4.1. Arquitectura del sistema

Para plantear la forma en que los distintos componentes dentro de un sistema se comunican, procedemos a establecer como se llevó a cabo la interacción en cuanto a publicar, localizar y consumir múltiples servicios y en caso de fallas aplicar una técnica de recuperación apoyada por la replicación y monitorización.

La arquitectura del sistema que se describe a continuación fue adecuada de la tesis de grado titulada “*Técnica de Recuperación y Ejecución de Servicios en Redes Superpuestas Peer to Peer No Estructuradas*”, presentada por Iván Ordoñez y Gonzalo Potosí [39]. En la Figura 4 se tiene una visión general del sistema, lo resaltado en azul son las adecuaciones realizadas sobre esta.

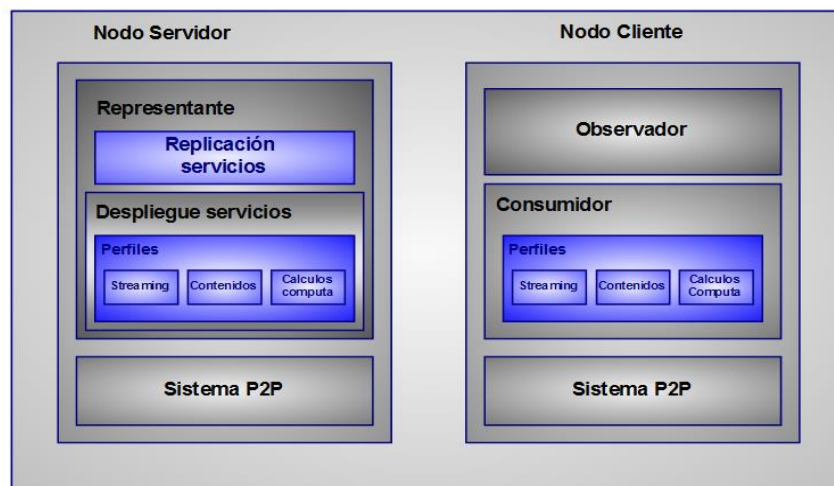


Figura 4. Arquitectura del Sistema

### 4.1.1. Sistema P2P

El sistema P2P proporciona un conjunto común de protocolos para el desarrollo de aplicaciones P2P. En la Figura 5, se muestran los componentes del sistema.



Figura 5. Sistema P2P

Este sistema es el encargado de crear, publicar y buscar los respectivos advertisements<sup>4</sup>, que representan el servicio en la red superpuesta P2P. También se encarga de las comunicaciones y del envío y recepción de mensajes entre cada nodo de la red superpuesta P2P, que se realizan por medio de los diferentes protocolos P2P [39].

Para la búsqueda y recuperación de servicios se utilizara una memoria cache la cual optimizara este proceso [39].

Para manejar la disponibilidad de servicios se hizo uso de un archivo log en el cliente, en la cual se almacenó el estado de la prestación para cada servicio, con el fin de que si se presenta una eventual falla del nodo servidor, los servicios se sigan prestando desde donde estaba la transferencia antes de la desconexión, sin pérdida de datos.

### 4.1.2. Representante

El representante del servicio a desplegar se encarga de crear el mensaje con el perfil del servicio a publicar, especificando los parámetros necesarios para establecer la comunicación con los mismos [39].

### 4.1.3. Despliegue servicios

Es el encargado de dar a conocer los distintos tipos de servicios a prestar en la red superpuesta P2P.

---

<sup>4</sup> Son documentos XML, que son utilizados para describir recursos de la red superpuesta P2P.



#### **4.1.4. Perfiles**

Este componente es el encargado de manejar las unidades funcionales de cada tipo de servicio, las entradas y salidas de datos y la forma en que se debe prestar o consumir dicho tipo de servicio. A continuación se presenta la descripción de cada uno de los perfiles para cada tipo de servicio:

##### **4.1.4.1. Perfil streaming**

Es un perfil para el manejo del servicio de media streaming, donde la unidad funcional es el flujo de bits; la entrada es el nombre del video; la salida es la transmisión del flujo de bits del video, el cual se debe consumir por medio de un reproductor de video.

##### **4.1.4.2. Perfil contenido**

Es un perfil para el manejo del servicio de contenido, donde la unidad funcional es un archivo; la entrada es el nombre del archivo; la salida es la transmisión del archivo, que se debe consumir almacenándolo en un espacio de disco.

##### **4.1.4.3. Perfil cálculos computacionales**

Es un perfil para el manejo del servicio de cálculos computacionales, donde la unidad funcional son los datos de entrada; un archivo con los datos para el cálculo representa la entrada; la salida es un archivo con las soluciones y se debe consumir por medio de una interfaz. Para el desarrollo del prototipo se realizara un cálculo de una serie de operaciones básicas.

#### **4.1.5. Replicación de servicios**

Es el encargado de publicar réplicas de los servicios, para que otros nodos disponibles en la red superpuesta P2P, repliquen dichos servicios.

#### **4.1.6. Consumidor**

El consumidor del servicio a desplegar toma el perfil creado por el representante, usa los parámetros de comunicación del servicio de acuerdo al perfil y presenta una interfaz para el consumo del mismo [39].

#### **4.1.7. Observador**

El observador realiza el monitoreo de la condición del servicio del servidor actual y en caso de desconexión, dispara el mecanismo de recuperación para obtener otro servicio igual [39].

## 4.2. Descripción de la técnica para soporte de la disponibilidad de servicios

Para poder describir la técnica de soporte de la disponibilidad de servicios, se hace necesario en primer lugar dar la descripción detallada de las funciones que tienen los nodos cliente y servidor; además de describir en más detalle las funciones del representante, consumidor y observador. También cómo se lleva a cabo la comunicación entre los nodos, el despliegue de los servicios y el intercambio de los mensajes dentro de la red superpuesta P2P.

### 4.2.1. Acciones de los nodos

#### 4.2.1.1. Nodo cliente

- **Iniciar cliente:** en esta acción el nodo cliente realiza las siguientes tareas: crear nodo cliente, crear grupo P2P, guardar grupo P2P, buscar servicio, guardar nodo servidor, iniciar conexión al servidor y recuperar servicio. Para garantizar que el cliente se inicie de una manera correcta, las tareas crear nodo cliente hasta iniciar conexión al servidor, deben ejecutarse de manera secuencial.
- **Crear nodo cliente:** esta tarea consiste en crear un nodo con parámetros como tipo de nodo, nombre del nodo y puerto. Define un identificador único para el nodo.
- **Crear grupo P2P:** esta tarea consiste en crear un grupo P2P para los servicios; por lo tanto, cada nodo que quiera usar los servicios debe registrarse en un grupo. Se utilizara el grupo por defecto que crea JXTA sin agregar subgrupos dentro de este.
- **Guardar grupo P2P:** el proceso de creación de un grupo en una red superpuesta P2P, lleva consigo la asignación de un identificador único para cada grupo. Por lo tanto, cada nodo de la red debe guardar este identificador para un uso futuro de los servicios dentro del grupo al cual se conectó. Esta tarea guarda la identificación del grupo.
- **Buscar servicio:** esta tarea consiste en realizar la búsqueda de los nodos que prestan el servicio deseado. El servicio es buscado por su nombre (búsqueda sintáctica) en la red superpuesta P2P.
- **Guardar nodo servidor:** después de realizada la tarea de buscar servicio, esta tarea se encarga de guardar localmente los nodos encontrados y elige uno de ellos para realizar la conexión.
- **Iniciar conexión al servidor y consumir servicio:** consiste en iniciar la conexión de un nodo cliente, hacia un nodo servidor que este prestando un servicio determinado y lo consume.
- **Recuperar servicio:** consiste en vigilar la condición del servicio y en caso de pérdida, realiza una nueva búsqueda con el objetivo de encontrar y consumir el mismo servicio desde otro servidor disponible.

#### 4.2.1.2. Nodo servidor

- **Iniciar servidor:** en esta acción el nodo servidor realiza las siguientes tareas: crear nodo servidor, crear grupo P2P, guardar grupo P2P, publicar servicio y guardar nodo cliente. Estas tareas deben ejecutarse de manera secuencial y ordenada, para garantizar que el servidor se inicie de una manera correcta.
- **Crear nodo servidor:** consiste en crear un nodo con parámetros como: tipo de nodo, nombre del nodo y puerto. Además define un identificador único para el nodo.
- **Crear grupo P2P:** esta tarea crea un grupo P2P para los servicios. Por lo tanto, cada nodo que quiera usar los servicios, debe registrarse en un grupo. Se utilizara el grupo por defecto que crea JXTA sin agregar subgrupos dentro de este.
- **Guardar grupo P2P:** el proceso de creación de un grupo en una red superpuesta P2P lleva consigo la asignación de un identificador único para cada grupo. Por lo tanto, cada nodo de la red debe guardar este identificador, para un uso futuro de los servicios dentro del grupo al cual se conectó. Esta tarea guarda la identificación del grupo.
- **Publicar servicio:** consiste en publicar cada uno de los servicios en la red superpuesta P2P usando un anuncio de publicación de servicio. Gracias a esto, la red se entera de la existencia de los servicios prestados por el nodo.
- **Conectar nodo cliente:** cuando un nodo cliente se conecta a un nodo servidor, se procede al intercambio de mensajes y a la prestación de servicios.
- **Replicar los servicios:** consiste en publicar réplicas de los servicios que ofrece un nodo servidor hacia otro nodo dentro de la red superpuesta P2P.

#### 4.2.2. Comunicación entre los nodos cliente y servidor

Para la comunicación entre los nodos se utiliza los protocolos de transporte y comunicación soportados por las redes superpuestas P2P. Para mejorar la comunicación entre los nodos es necesario utilizar un lenguaje común para el intercambio de mensajes.

- **Anuncio:** mensaje que publica la existencia de recursos (servicios) en la red superpuesta P2P. Este debe estar representado por un identificador único [39].
- **Perfil:** elemento integrado al anuncio que especifica la descripción y parámetros necesarios para interactuar con un servicio [39].

#### 4.2.3. Servicios a desplegar

Representan los servicios a prestar o desplegar a través de la red superpuesta P2P. Los servicios deben estar representados por un perfil que especifica los parámetros,

perfiles, requisitos y formatos necesarios para interactuar con los servicios. En el nodo servidor el representante es el encargado de definir el perfil y este posteriormente es integrado al anuncio de publicación que es usado en el proceso de publicación de servicios por el nodo servidor. En el nodo cliente el consumidor usando el perfil, presenta una interfaz para el consumo de los servicios.

#### **4.2.4. Conexión**

En [39] se representan los dos escenarios de conexión que se dan entre los nodos en una red superpuesta P2P.

#### **4.2.5. Mensajes**

Los mensajes dentro del prototipo software serán representados mediante el formato de intercambio de datos XML.

#### **4.2.6. Envío de mensajes**

Se llevó a cabo entre los nodos dentro de la red superpuesta P2P, con el fin de publicar los servicios, realizar búsquedas, recuperar servicios, entre otros.

#### **4.2.7. Búsqueda, recuperación y disponibilidad de los servicios**

La búsqueda es el procedimiento mediante el cual los nodos clientes pueden ubicar y recuperar recursos en la red; para el caso de este trabajo: servicios. Cuando un nodo cliente se conecta a la red superpuesta P2P con el fin de consumir alguno de los servicios, realiza una búsqueda sintáctica sobre los anuncios de publicación, como resultado obtendrá un conjunto de servidores disponibles. Posteriormente, el nodo cliente puede establecer una conexión directa con uno de los nodos servidores encontrados y utilizar el servicio prestado usando el perfil del servicio. En caso de desconexión del nodo servidor (en el que actualmente el nodo cliente se encuentra consumiendo el servicio) el observador disparará el mecanismo de recuperación quien utiliza procedimientos de búsqueda local (listado local de índices) o remota, para dar con un nuevo servidor disponible y poder recobrar el servicio. Para garantizar la disponibilidad se dispone de la replicación de los servicios en otro nodo.

#### **4.2.8. Observador**

Es un componente encargado de realizar el monitoreo del estado de los servicios; y en caso de desconexión, este será también el encargado de disparar el mecanismo de recuperación para obtener otro servicio igual y que este le siga prestando el servicio desde donde estaba antes de la desconexión, para lo cual se utiliza un archivo log que contiene el registro del estado del servicio.

### 4.3. Prestación de servicios

A continuación se representa a través de diagramas de flujo los procesos de publicación de servicios, búsqueda de servicios, recuperación de servicios y replicación de servicios.

#### 4.3.1. Publicación de los servicios

En la Figura 6 se presenta el diagrama de flujo del proceso de publicación de servicios por parte del nodo servidor.



Figura 6. Diagrama de flujo para publicar un servicio.

En el diagrama de flujo anterior, cabe resaltar que después de crear y publicar el grupo en la creación del descriptor del servicio, se incluye el perfil del servicio donde se especifican los parámetros necesarios para la publicación de dicho servicio. En la sección 6.3.3.4 se especifican los diagramas de secuencia donde se describe con más detalle este proceso.

### 4.3.2. Búsqueda de los servicios

A continuación se presenta el diagrama de flujo del proceso de búsqueda de los servicios para el nodo cliente Figura 7.

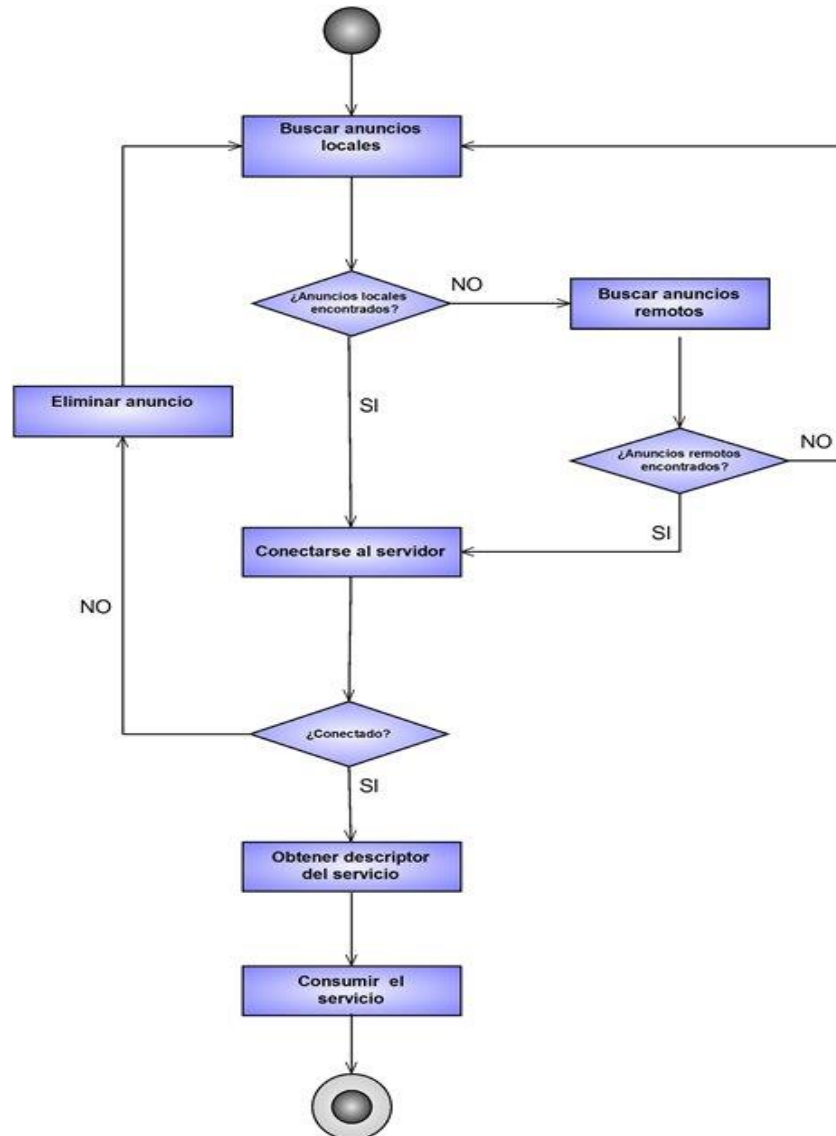


Figura 7. Diagrama de flujo para búsqueda de servicios.

En el anterior diagrama se realiza la búsqueda de servicios tanto locales como remotos y se toma como prioridad a los anuncios de servicios locales. Si no se encuentran servicios locales, se hace la búsqueda remota, para después hacer la conexión con el servidor que presta el servicio y así poder obtener el descriptor del servicio, donde se incluyen los parámetros necesarios para poder finalmente consumir el servicio requerido.

### 4.3.3. Recuperación del servicio

En la Figura 8 se presenta el diagrama de flujo del proceso para la recuperación de los servicios por parte del nodo cliente.

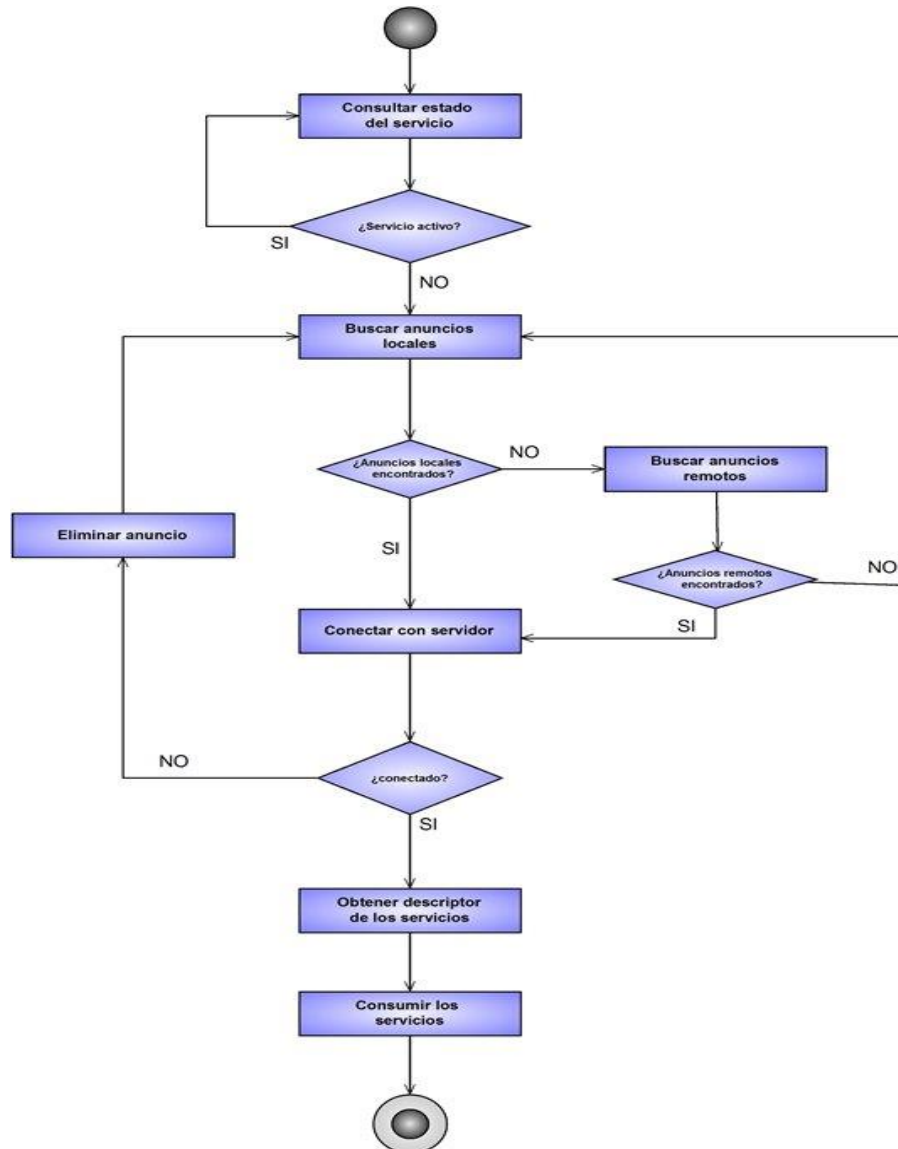


Figura 8. Diagrama de flujo para recuperar servicios.

En el diagrama anterior, para la recuperación de los servicios se tiene un mecanismo de monitoreo del consumo de los servicios, observando siempre el estado del servicio si está activo; en caso contrario se activa el mecanismo de recuperación de servicios para realizar una nueva búsqueda de otro servidor que preste el mismo servicio, para que el cliente lo pueda seguir consumiendo sin pérdida de datos.

#### 4.3.4. Replicación de los servicios

En el siguiente diagrama de flujo se presenta los pasos que realiza la red superpuesta P2P para la replicación de los servicios. Figura 9.



Figura 9. Diagrama de flujo para la replicación de servicios.

En el diagrama anterior, la replicación de servicios se lleva a cabo de la siguiente manera, un nodo servidor publica la intención de replicar por medio de un anuncio sobre la red; un nodo servidor con la intención de crear replicas realiza la búsqueda de anuncios de replicación publicados por otros servidores y publica el servicio.

En este capítulo se seleccionó la arquitectura del sistema, así como su adaptación con la técnica de replicación, además se realizó la descripción de todos sus componentes, sus funciones y sus interacciones. Finalmente por medio de diagramas de flujo se describió los procesos de publicación, búsqueda, recuperación y replicación de servicios.



## CAPITULO 5. TECNOLOGIAS PARA EL DESARROLLO DE REDES SUPERPUSTAS P2P

En este capítulo se describen las diferentes tecnologías seleccionadas para el desarrollo del prototipo. En el CD, en el documento de Anexos, en el Anexo C se puede encontrar más información sobre este capítulo.

### 5.1. Tecnología para el desarrollo de la red superpuesta P2P

Para seleccionar la tecnología que se utilizó en el desarrollo de la red superpuesta P2P; en primer lugar, se exploraron varias aplicaciones las cuales implementan diferentes protocolos. Después, se efectuó una comparación según: grado de centralización, uso de nodos especializados, protocolos utilizados, tipo de servicio y componentes. Tal como se muestra en la Tabla 5.

Aplicación	Grado centralización	Uso de nodos especializados	Protocolos utilizados	Tipo de servicio	Componentes
<b>Gnutella</b>	Descentralizada	Ultranodos	UDP TCP HTTP	Contenidos	Servents.
<b>BitTorrent</b>	Centralizada	Tracker	IP TCP	Contenidos	Tracker. Sedes. Lecheers.
<b>eDonkey</b>	Semi-centralizada	Múltiples servidores	MFTP TCP UDP	Contenidos	Servidores. Clientes.
<b>Napster</b>	Centralizada		HTTP	Contenidos	
<b>Kazaa</b>	Parcialmente centralizada	Super-nodos			Supernodos. Nodos.
<b>jxta</b>	Descentralizada	Super-nodos	TCP UDP HTTP	Múltiples servicios.	Nodos, Supernodos, Grupos.
<b>Can</b>	Centralizada	No aplica	TCP UDP	Contenidos	Servidor. Nodos.
<b>Chor</b>	Centralizada	No aplica	TCP UDP	Contenidos	Servidor. Nodos.
<b>Pastry</b>	Centralizada	No aplica	TCP UDP	Contenidos	Servidor. Nodos.
<b>MSN</b>	Centralizada	No aplica	No aplica	Chat	Servidor. Nodos.
<b>AOL</b>	Centralizada	No aplica	No aplica	Chat , video conferencia	Servidor. Nodos.
<b>SETI@home</b>	Centralizada	No	HTTP	Computación distribuida	Servidor. Nodos.

Tabla 5. Comparación de algunas aplicaciones estudiadas.

Los protocolos que se acaban de estudiar presentan características arquitecturales muy diferentes, y además solo brindan los mecanismos necesarios para el desarrollo de

aplicaciones P2P de propósito específico; aplicaciones que aunque son conceptualmente análogas, difieren profundamente en los detalles de implementación y también algunos son aplicaciones privativas. Para el desarrollo del trabajo de grado usaremos la tecnología JXTA que por medio de sus protocolos es la que más se acerca a los intereses de trabajo de grado; así mismo, este trabajo de grado tiene la base en dos tesis anteriores: uno de búsqueda [35] y otro de recuperación de servicios [39]; los cuales también utilizaron esta tecnología.

A continuación, se presentan los principales conceptos de esta tecnología, así como también las razones por las cuales fue seleccionada.

### 5.1.1. JXTA

En esta red virtual JXTA cualquier nodo puede interactuar con otros nodos a pesar de su ubicación, tipo de dispositivo o sistema operativo; inclusive cuando algunos nodos y recursos están ubicados detrás de un firewall o trabajan sobre un tipo de transporte de red diferente. Así el acceso a los recursos de red no está limitado por incompatibilidades de plataforma o los permisos de una arquitectura jerárquica cliente servidor. Esta tecnología puede correr sobre cualquier dispositivo dentro de la red, incluyendo teléfonos celulares, PDAs, sensores electrónicos, computadoras de escritorio y servidores. Al estar basado en tecnologías probadas y estandarizadas tales como: HTTP, TCP/IP, y XML; la tecnología JXTA 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 [40].

### 5.1.2. ¿Por qué JXTA como tecnología de desarrollo?

La tecnología JXTA provee un conjunto de funciones básicas para la implementación de redes superpuestas P2P, esta tecnología está diseñada para ser independiente de la plataforma de desarrollo, lenguaje de programación, definición de servicios y protocolos de red. El proyecto JXTA trae consigo la superación de varios defectos de muchas redes superpuestas P2P existentes o en desarrollo, tales como [40]:

- **Interoperabilidad.** La tecnología JXTA está diseñada para permitir que los nodos interconectados se ubiquen entre sí, se comuniquen, participen en actividades cooperativas y se ofrezcan servicios entre diferentes redes superpuestas P2P y distintas comunidades.
- **Independencia de plataforma.** La tecnología JXTA está diseñada para ser independiente del lenguaje de programación, del sistema operativo, de la definición de servicios (RMI o WSDL) o de protocolos de red.
- **Ubicuidad.** La tecnología JXTA está diseñada para ser implementada sobre cualquier dispositivo digital con acceso a la red.

### 5.1.3. Arquitectura JXTA

Se divide en tres capas lógicas como se muestra en la Figura 10. En seguida se describe cada una de ellas [40].

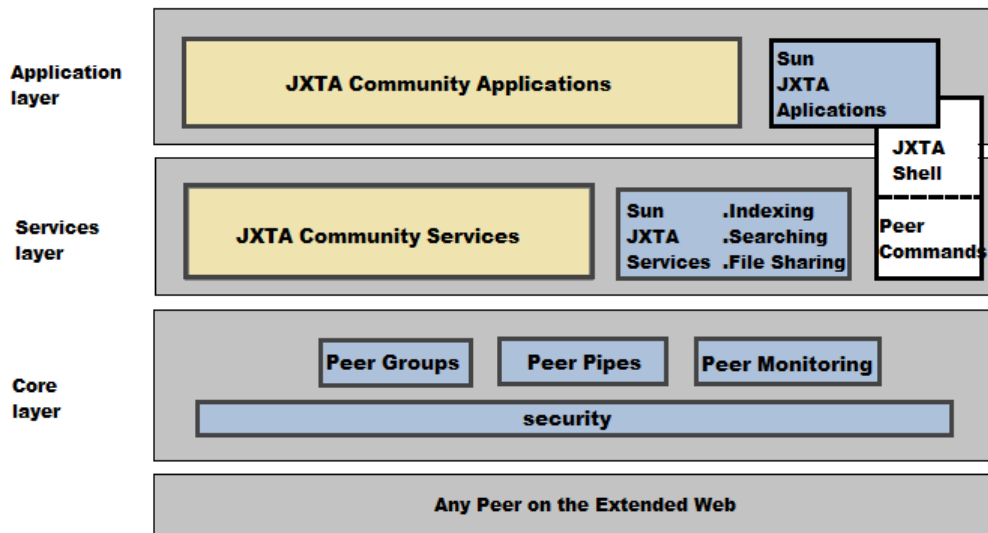


Figura 10. Arquitectura JXTA.

- **Núcleo JXTA**

El núcleo JXTA o core provee el soporte fundamental para los servicios P2P y las aplicaciones. Así mismo, los mecanismos de peer groups, peer pipes y peers monitoring en un ambiente de ejecución seguro multiplataforma [40].

- ✓ **Peer groups:** establece un grupo de nodos y los identifica nombrándolos dentro de un grupo de nodos, con mecanismos para crear políticas para la creación y borrado, membresía, publicidad y descubrimiento de otros grupos de nodos, comunicación, seguridad e intercambio de contenido.
- ✓ **Peer pipes:** provee canales de comunicación entre los nodos. Los mensajes enviados en peer pipes son estructurados con XML y soportan intercambio de datos, contenido y código en una forma independiente del protocolo.
- ✓ **Peers monitoring:** permite el control del comportamiento y las actividades de los nodos en un grupo. También puede ser utilizado para implementar funciones de administración de nodos, incluyendo control de acceso, medidas de tráfico y balance de ancho de banda.

- **Aplicaciones JXTA**

Son construidas usando, tanto los servicios de nodos, como la capa núcleo. La filosofía del proyecto es dar soporte a los niveles fundamentales ampliamente y confiar en la comunidad de desarrolladores P2P para proveer servicios y aplicaciones adicionales. Las aplicaciones de los nodos le permiten a las capas núcleo y servicios incluir subastas P2P que vinculan vendedores y compradores directamente. Los compradores son capaces de programar su estrategia de oferta escribiendo un simple script. Se facilita la comunicación y colaboración dentro de los grupos de nodos mediante servicios de mensajería instantánea y calendario, que son seguros e independientes de la necesidad de alojamiento en un proveedor de servicio [40].

#### **5.1.4. Protocolos**

Desde el nivel más alto de abstracción, la tecnología JXTA es un conjunto de protocolos. Cada protocolo está definido por uno o más mensajes intercambiados entre los participantes de la comunicación. JXTA define una serie de mensajes XML, conocidos como protocolos para la comunicación entre nodos. Los nodos usan estos protocolos para descubrir otros nodos, anunciar y descubrir recursos de red y para la comunicación y el enrutamiento de mensajes [40]. Estos protocolos son:

- Peer Discovery Protocol (Protocolo de Descubrimiento).
- Peer Resolver Protocol (Protocolo de Resolución).
- Peer Information Protocol (Protocolo de Información).
- Rendezvous Protocol (Protocolo de puntos encuentro).
- Pipe Binding Protocol (Protocolo de unión de tuberías).
- Endpoint Routing Protocol (Protocolo de Ruteo de Extremos).

## **5.2. Tecnología para el servidor fuente del servicio de media streaming**

Para el desarrollo del servidor fuente del servicio de media streaming, se seleccionó el reproductor de video VLC. A continuación, se presentan los principales conceptos de esta tecnología, así como también las razones por las cuales fue seleccionado.

### **5.2.1. ¿Qué es VLC?**

VLC Media Player es un reproductor multimedia, gratuito y multiplataforma, soporta un gran número de formatos de audio y vídeo sin necesidad de instalar códecs adicionales: MPEG-1, MPEG-2, MPEG-4, DivX, MP3, OGG, MOV, RAM, AVI, FLV, entre otros. Se puede utilizar como servidor de streaming en una red local o de banda ancha [51].

### 5.2.2. ¿Por qué VLC?

Esta selección se basó en el estudio realizado en [39], donde se mencionan algunas de las características que facilitan el uso de esta tecnología, las cuales 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 todos 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.

### 5.2.3. 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, los cuales son los encargados de todas sus funcionalidades [39].

### 5.2.4. VLCj

La librería VLCj puede ser usada 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 [44].

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 [44].

### 5.3. Tecnología para el servidor fuente del servicio de contenidos

Con el fin de simular el servidor fuente del servicio de contenidos, se seleccionó un servidor FTP. En seguida, se presentan los principales conceptos de esta tecnología, así como también las razones por las cuales fue seleccionado.

#### 5.3.1. ¿Qué es FTP?

FTP es el sistema de transferencia de archivos más estable y fiable que hay en Internet. Esto significa que la descarga y subida de archivos tendrán más opciones de completarse sin errores de transferencia, y quedarán intactos después del envío. El propio protocolo TCP/IP, provee de un sistema de control y corrección de paquetes al ser recibidos en el destino. Si un paquete en la secuencia de envío se pierde, el ordenador que recibe los datos hace una petición para el reenvío de datos. Esta es la razón de porque TCP/IP es tan fiable. Esto es una ventaja porque FTP funciona sobre el protocolo TCP/IP [52].

Existen normas o estándares que permiten a FTP funcionar en casi cualquier medio. Estas especificaciones son las que hacen que plataformas independientes dentro de Internet puedan comunicarse entre sí [52].

#### 5.3.2. ¿Por qué FTP?

Esta selección se basó en algunas de las características que facilitan el uso de esta tecnología, las cuales 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.

#### 5.3.3. 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 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 [45].

### 5.3.4. 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 [45].

## 5.4. Tecnologías para el servidor fuente del servicio de cálculos computacionales

Con el fin de simular el servidor fuente del servicio de cálculos computacionales, se seleccionó la tecnología de sockets. A continuación se presentan los principales conceptos de esta tecnología, así como también las razones por las cuales fue seleccionada.

### 5.4.1. ¿Qué es un socket?

Son un sistema de comunicación entre procesos de diferentes máquinas de una red. Más exactamente, un socket es un punto de comunicación por el cual un proceso puede emitir o recibir información. Utilizan una serie de primitivas para establecer el punto de comunicación, para conectarse a una máquina remota en un determinado puerto que esté disponible, para escuchar en él, para leer o escribir y publicar información en él, y finalmente para desconectarse [53].

Fueron popularizados por Berkeley Software Distribution, de la universidad norteamericana de Berkeley. Los sockets han de ser capaces de utilizar el protocolo de streams TCP (Transfer Control Protocol) y el de datagramas UDP (User Datagram Protocol) [53].

### 5.4.2. ¿Por qué sockets?

Esta selección se basó en algunas de las características que facilitan el uso de esta tecnología, las cuales 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.
- Cantidad de bibliografía existente.

### 5.4.3. ¿Qué es un Hilo?

Los hilos son útiles porque permiten que el flujo de un programa sea dividido en dos o más partes, cada una ocupándose de alguna tarea de forma independiente. En Java un Thread (hilo, tarea) es una clase base para definir hilos de ejecución concurrentes dentro de un mismo programa. Una vez que es iniciado, cada hilo vive de forma independiente de los otros, no existe ninguna relación entre ellos, ni tampoco ningún conflicto, dado que no comparten nada [54].

### 5.4.4. ¿Por qué hilos?

Esta selección se basó en algunas de las características que facilitan el uso de esta tecnología, las cuales se describen a continuación:

- Porque un nodo servidor cuando publica un servicio, necesita atender a múltiples clientes.
- Dan soporte a la concurrencia en cuanto a la prestación de servicios.

En este capítulo se seleccionó las tecnologías a utilizar en el desarrollo del prototipo software para la validación de la técnica propuesta. Además se describió cada una de ellas y por qué fue seleccionada.



## Capítulo 6. DESARROLLO DEL PROTOTIPO SOFTWARE

En este capítulo se describe el proceso seguido durante la implementación del prototipo software para la verificación de la técnica propuesta para el soporte de la disponibilidad de servicios en un entorno de redes superpuestas P2P no estructuradas.

### 6.1. Especificación de requisitos

Después de haber definido las características principales de la “*Técnica para soporte de la disponibilidad de servicios en un entorno de redes superpuestas P2P no estructuradas*”, se obtuvieron una serie de requerimientos, tanto funcionales como no funcionales, que se hacen necesarios para el desarrollo del prototipo software, estos requisitos se pueden ver con más detalle en el CD, en el documento de Anexos en el Anexo B: Documento de especificación de requisitos software – SRS.

#### 6.1.1. Alcance

Para verificar la “*Técnica para soporte de la disponibilidad de servicios en un entorno de redes superpuestas P2P no estructuradas*”, el prototipo software que será desarrollado utilizara tres tipos de servicio: media streaming, contenidos y cálculos computacionales. Por lo tanto, la mayoría de la funcionalidad del prototipo software se ejecutará de manera automática, con poca interacción con el usuario final del sistema. Este prototipo software permitirá ejecutar la funcionalidad tanto de cliente como de servidor en una sola interfaz de usuario.

El prototipo software integrará varias tecnologías para el desarrollo de redes superpuestas P2P: la primera JXTA, para el desarrollo de redes superpuestas P2P; la segunda VLC, para el desarrollo de sistemas de media streaming; la tercera FTP, para la transferencia de archivos y por último el uso de hilos y sockets para realizar los cálculos y transferir la respuesta.

#### 6.1.2. Funciones del producto

El prototipo software permitirá el despliegue de los servicios a través de la red superpuesta P2P, los cuales podrán ser media streaming, contenidos y cálculos computacionales, permitiendo conexiones de tipo punto a multipunto en el nivel de aplicación. Teniendo como base las fuentes principales de los servicios (servidor de media streaming, servidor FTP) ajenos a la red superpuesta P2P. Los nodos servidor que ingresen a la red superpuesta P2P prestarán el servicio de retransmisión de la fuente principal. Por otra parte, los nodos clientes que ingresen a la red superpuesta P2P buscarán los servicios 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 cualquiera de los servicios con 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 [39].

### 6.1.3. Características del usuario

Ya que la mayoría de las funcionalidades del prototipo software se ejecutarán de manera automática y con poca interacción con el usuario final del sistema, el usuario solo tendrá las siguientes funciones:

- 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.

### 6.1.4. Requerimientos funcionales

A continuación, se presenta la descripción de los requisitos funcionales del prototipo software:

- **Publicar servicios (R-1):** se debe crear y publicar los servicios de media streaming, contenidos y cálculos computacionales, no se publicaran servicios repetidos en el mismo servidor.
- **Buscar los servicios (R-2):** permitir la búsqueda de los servicios de media streaming, contenidos y cálculos computacionales, a través de las consultas sobre las publicaciones de los servicios de los nodos servidor.
- **Monitorear condición de los servicios (R-3):** 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 alguno de los servicios se debe disparar inmediatamente el mecanismo de la recuperación.
- **Recuperar los servicios (R-4):** se debe recuperar los servicios de media streaming, contenidos y cálculos computacionales. Un nodo cliente del sistema realiza la búsqueda en el cache local y/o búsqueda remota con el objetivo de encontrar nuevos servidores disponibles y recuperar los servicios.
- **Crear mensaje con el perfil para el servicio de media streaming (R-5):** es necesario 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.
- **Crear mensaje con el perfil para el servicio de contenidos (R-6):** se 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.

- **Crear mensaje con el perfil para el servicio de cálculos computacionales (R-7):** es necesario 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.
- **Obtener perfil de los servicios (R-8):** consultar las publicaciones de los servicios de los nodos servidores para poder obtener el perfil de cada servicio.
- **Consumir servicios (R-9):** permitir a través del perfil de cada servicio iniciar el consumo de los servicios que presta un nodo servidor.
- **Iniciar el nodo servidor del servicio de media streaming (R-10):** se deberá permitir al usuario iniciar el nodo servidor de media streaming.
- **Iniciar el nodo servidor del servicio de contenidos (R-11):** permitir al usuario iniciar el nodo servidor de contenidos.
- **Iniciar el nodo servidor del servicio de cálculos computacionales (R-12):** es necesario permitir al usuario iniciar el nodo servidor de cálculos computacionales.
- **Detener el nodo servidor de los servicios (R-13):** es necesario permitir al usuario detener el nodo servidor.
- **Iniciar el nodo cliente de los servicios (R-14):** permitir al usuario iniciar el nodo cliente.
- **Detener el nodo cliente de los servicios (R-15):** se deberá permitir al usuario detener el nodo cliente.

### 6.1.5. Requerimientos no funcionales

A continuación, se presenta la descripción de los requerimientos no funcionales del prototipo software:

- **Disponibilidad:** el prototipo debe brindar un alto nivel de disponibilidad a través de la técnica propuesta.
- **Portabilidad:** debe funcionar en diferentes sistemas operativos.
- **Reusabilidad:** permitir que su código pueda ser utilizado por otros sistemas.
- **Robustez:** ser capaz de recuperarse ante un eventual fallo.

## 6.2. Implementación

En esta sección vamos a describir el proceso que se llevó a cabo durante la implementación del prototipo software para la verificación de la técnica propuesta; teniendo en cuenta la metodología de desarrollo empleada, las herramientas utilizadas y los mecanismos de prueba.

### 6.2.1. Metodología de Desarrollo

Para el desarrollo del prototipo software se utilizó la metodología de programación extrema (Extreme Programming) [23]. Debido a que esta metodología se centra principalmente en satisfacer al cliente, y ya que no se cuenta con un cliente externo en nuestro trabajo de grado, hemos optado por incluir como cliente a nuestro director de trabajo de grado, el cual será el encargado de llevar a cabo todas las funciones que tiene el cliente dentro de esta.

Esta metodología presenta varias prácticas, de las cuales se van a tomar en cuenta las siguientes:

- **Programación por parejas:** el código es escrito por dos personas trabajando en el mismo computador.
- **El juego de la planificación:** la estrategia de esta práctica es poner en producción las características más importantes lo antes posible, las piezas claves son las historias de usuario.
- **Metáfora del sistema:** cada proyecto es guiado por una historia simple de cómo funciona el sistema en general y debe estar en un lenguaje común, entendible para todos.
- **Diseño simple:** el sistema se diseña con la máxima simplicidad posible.
- **Pruebas continuas:** los casos de prueba se escriben antes que el código. Los desarrolladores escriben pruebas unitarias y los clientes especifican pruebas funcionales.
- **Refactorización:** es posible reestructurar el sistema sin cambiar su comportamiento.
- **Poseción colectiva del código:** nadie es dueño de un módulo, cualquier programador puede cambiar cualquier parte del sistema en cualquier momento.
- **Integración continua:** los cambios se integran en el código base varias veces por día.
- **Semana laboral de 40 horas:** cada trabajador labora no más de 40 horas por semana.
- **Estándares de codificación:** todo el código debe estar escrito de acuerdo a un estándar de codificación.

### 6.2.2. Herramientas

Para la implementación del prototipo software se utilizaron las siguientes herramientas:

#### 6.2.2.1. Frameworks

Para la implementación del prototipo se utilizaron los siguientes frameworks:

- **Framework P2P:** para la implementación de la red superpuesta P2P se utilizó la tecnología JXTA.
- **Contenidos:** para la ejecución del servicio de contenidos a desplegar en la red superpuesta P2P, se utilizó un servidor FTP (File Transference Protocol).
- **Media streaming:** con el objetivo de prestar el servicio de media streaming a desplegar en la red superpuesta P2P, se utilizaron las librerías LibVLC (SDK Media Player VLC) y la librería de 'binding' VLCj.
- **Cálculos computacionales:** con el propósito de desarrollar el servicio de cálculos computacionales, se utilizaron las tecnologías de hilos, sockets y funciones en Java.

#### 6.2.2.2. Plataforma de desarrollo

El desarrollo del prototipo software se realizó utilizando el lenguaje de programación Java, con la versión del JDK 1.8.0\_25, el cual es un lenguaje de programación de propósito general e independiente de la plataforma. Este lenguaje fue escogido por las siguientes razones:

- ✓ Experiencia del equipo de desarrollo en este lenguaje.
- ✓ A pesar de que la plataforma para redes superpuestas P2P JXTA soporta la implementación en varios lenguajes tales como C Y C++, la implementación en Java es la más avanzada.
- ✓ La comunidad de desarrolladores en Java ofrecen una amplia variedad de información y documentación.

#### 6.2.2.3. Entorno de desarrollo integrado

El entorno de desarrollo integrado con el cual se desarrolló el prototipo software es Netbeans en su versión 8. Este entorno fue escogido por las siguientes razones:

- ✓ Experiencia del equipo de desarrollo en este entorno.
- ✓ Este entorno ofrece una gran usabilidad, ya que es muy fácil de entender y manejar.
- ✓ Este entorno también ofrece facilidad de integración, tanto con el código como con las librerías de JXTA, VLC, NET, Sockets e Hilos.

#### 6.2.3. Análisis y diseño

Para el análisis y diseño se realizaron las historias de usuario, los modelos de caso de uso en formato de alto nivel, diagramas de caso de uso, diagrama de clases y diagramas de secuencia. A continuación, se presenta una breve definición de estos; para más información se puede dirigir al CD, en el documento de Anexos, en el Anexo D: Historias de usuario.

### 6.2.3.1. Historias de usuario

A continuación, se presenta una breve descripción de las historias de usuario:

- **Solicitar un servicio de contenidos:** se presenta cuando el usuario desde la interfaz gráfica del cliente solicita el servicio de un contenido en la red superpuesta P2P.
- **Solicitar un servicio de media streaming:** se presenta cuando el usuario desde la interfaz gráfica del cliente solicita el servicio de media streaming en la red superpuesta P2P.
- **Solicitar un servicio de cálculos computacionales:** se presenta cuando el usuario desde la interfaz gráfica del cliente solicita el servicio de cálculos computacionales en la red superpuesta P2P.
- **Publicar un servicio de contenidos:** se presenta cuando el usuario desde la interfaz gráfica del servidor publica que tiene disponible el servicio de contenidos para prestar sobre la red superpuesta P2P.
- **Publicar un servicio de media streaming:** se presenta cuando el usuario desde la interfaz gráfica del servidor publica que tiene disponible el servicio de media streaming para prestar sobre la red superpuesta P2P.
- **Publicar un servicio de cálculos computacionales:** se presenta cuando el usuario desde la interfaz gráfica del servidor publica que tiene disponible el servicio de cálculos computacionales para prestar sobre la red superpuesta P2P.
- **Replicación de servicios:** se presenta cuando un nodo servidor publica un anuncio de replicación de sus servicios y este servicio es replicado por otro nodo servidor, con el fin de aumentar la disponibilidad de dichos servicios.

#### 6.2.3.1.1. Priorización de historias de usuario

A continuación, en la Tabla 6 se presenta la priorización de las historias de usuario, la cual fue definida en un rango de 1-5, siendo 1 la prioridad más baja y 5 la prioridad más alta. Igualmente se presentan las tareas que se despliegan de cada una de las historias de usuario.

NOMBRE HISTORIA	PRIORIDAD	TAREAS
Solicitar un servicio de contenidos	4	<ul style="list-style-type: none"><li>• Buscar servicio de contenido.</li><li>• Conectarse a un servidor que tenga el servicio.</li><li>• Consumir servicio de contenido.</li><li>• Monitorear el servicio.</li><li>• En caso de pérdida del servicio de contenido recuperar el servicio.</li></ul>
Solicitar un servicio de media streaming	2	<ul style="list-style-type: none"><li>• Buscar servicio de media streaming.</li></ul>

		<ul style="list-style-type: none"> <li>• Conectarse a un servidor que tenga el servicio.</li> <li>• Consumir servicio de media streaming.</li> <li>• Monitorear el servicio.</li> <li>• En caso de pérdida del servicio de media streaming recuperar el servicio.</li> </ul>
<b>Solicitar un servicio de cálculos computacionales</b>	4	<ul style="list-style-type: none"> <li>• Buscar servicio de cálculos computacionales.</li> <li>• Conectarse a un servidor que tenga el servicio.</li> <li>• Consumir servicio de cálculos computacionales.</li> <li>• Monitorear el servicio.</li> <li>• En caso de pérdida del servicio de cálculos computacionales recuperar el servicio.</li> </ul>
<b>Publicar un servicio de contenidos:</b>	3	<ul style="list-style-type: none"> <li>• Crear perfil del servicio de contenidos.</li> <li>• Agregar perfil del servicio de contenidos al anuncio.</li> <li>• Publicar el servicio de contenidos.</li> <li>• Prestar el servicio de contenidos.</li> </ul>
<b>Publicar un servicio de media streaming</b>	2	<ul style="list-style-type: none"> <li>• Crear perfil del servicio de media streaming.</li> <li>• Agregar perfil del servicio de media streaming al anuncio.</li> <li>• Publicar el servicio de media streaming.</li> <li>• Prestar el servicio de media streaming.</li> </ul>
<b>Publicar un servicio de cálculos computacionales</b>	3	<ul style="list-style-type: none"> <li>• Crear perfil del servicio de cálculos computacionales.</li> <li>• Agregar perfil del servicio de cálculos computacionales al anuncio.</li> <li>• Publicar el servicio de cálculos computacionales.</li> <li>• Prestar el servicio de cálculos computacionales.</li> </ul>
<b>Replicación de servicios</b>	5	<ul style="list-style-type: none"> <li>• Publicar un anuncio de replicación de servicios.</li> <li>• Replicar el servicio en otro nodo.</li> </ul>

Tabla 6. Priorización de historias de usuario.

### 6.2.3.2. Diagrama de casos de uso

En la Figura 11 se presenta el diagrama de casos de uso, mediante el cual se representan las funcionalidades del sistema relacionadas con los actores del mismo.

Debido a que este trabajo de grado corresponde a la fase II del trabajo de grado de Potosí y Ordoñez [39], los casos de uso resaltados en color azul, son los casos de uso en los cuales se realiza el aporte de este trabajo de grado, ya que es donde se modifica para aumentar la disponibilidad de servicios.

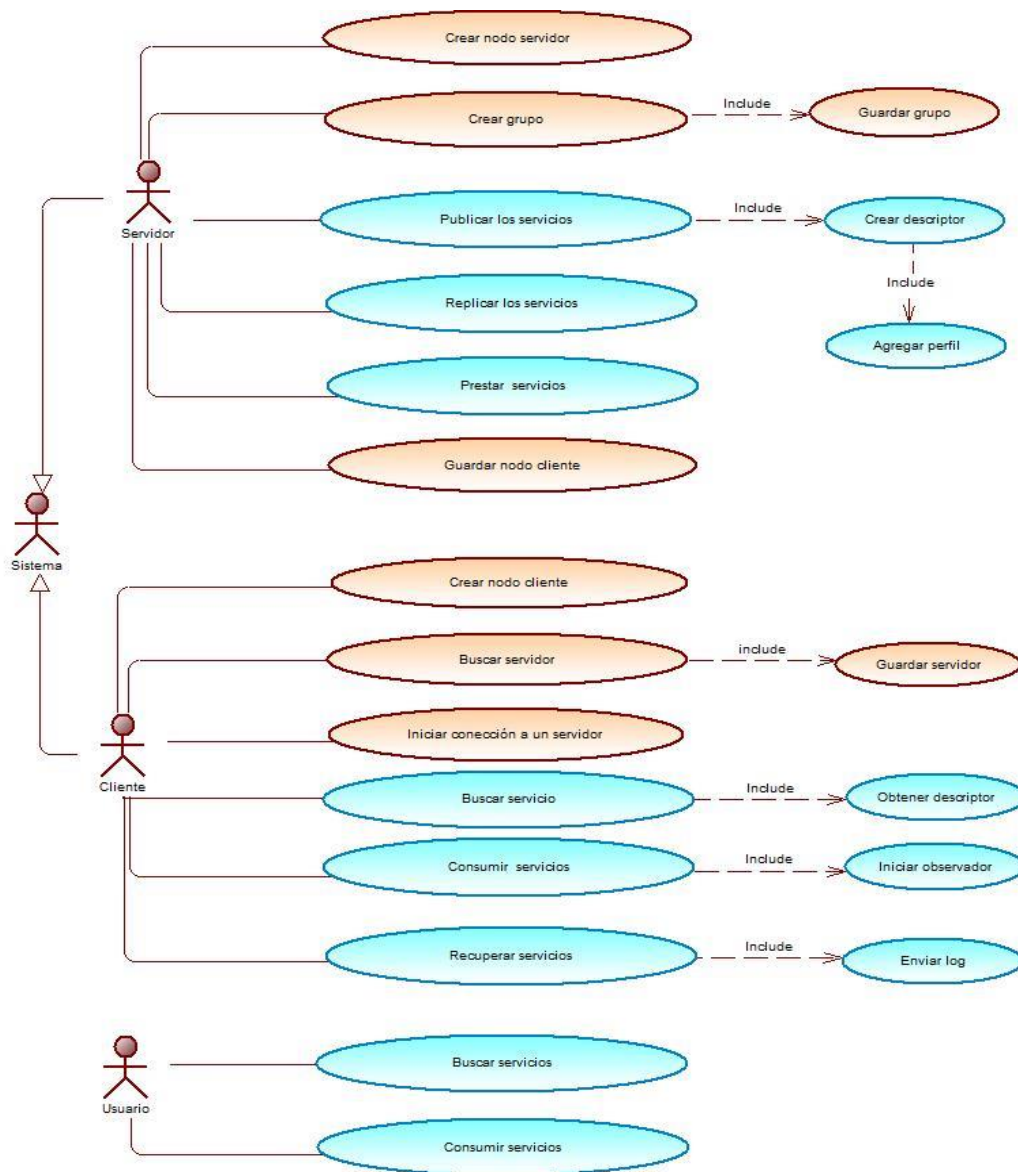


Figura 11. Diagrama de casos de uso.



#### 6.2.3.2.1. Actores del Sistema

A continuación, se presenta la definición de los actores que interactuarán con el prototipo software.

- ✓ **Actor sistema (ACT-1):** el sistema se puede comportar como cliente o como servidor:
  - Como servidor el sistema puede publicar, replicar y prestar servicios, guardar nodo cliente, crear grupo y crear nodo servidor.
  - Como cliente el sistema puede crear nodo cliente, iniciar conexión a un servidor, buscar, consumir y recuperar servicios.
- ✓ **Actor usuario (ACT-2):** el actor usuario por medio de una interfaz gráfica podrá publicar, buscar y consumir servicios.

#### 6.2.3.3. Diagrama de Clases

A continuación, se presenta el diagrama de clases del sistema.

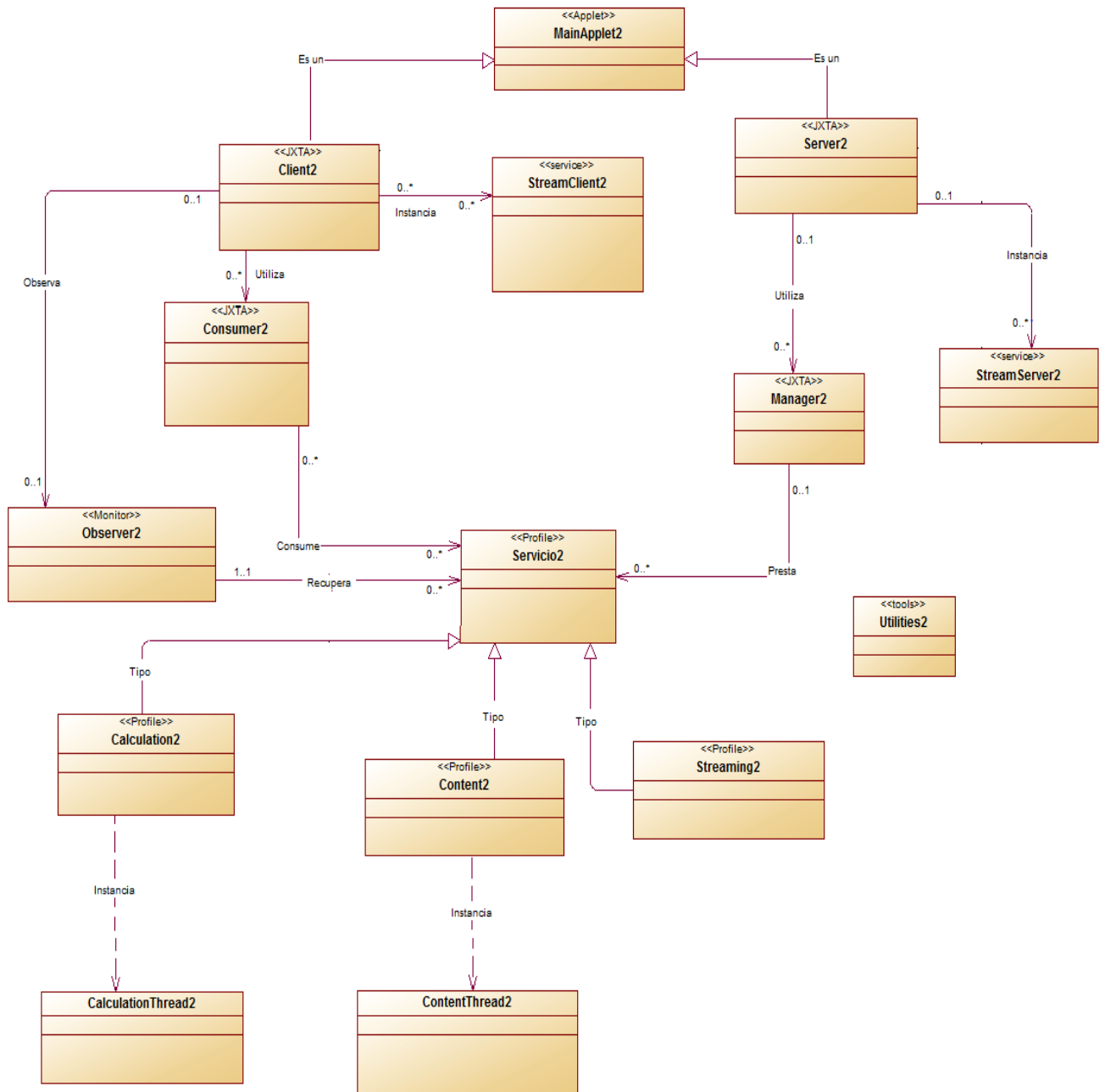


Figura 12. Diagrama de clases del sistema.

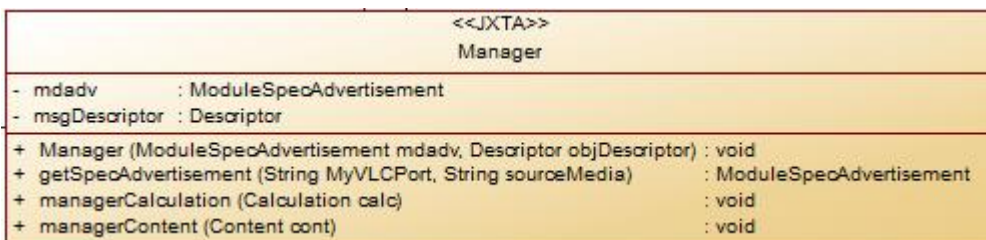
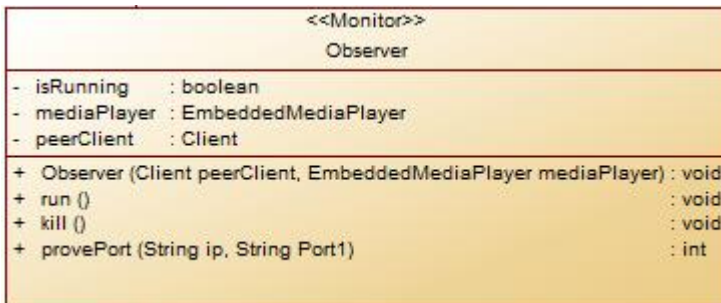
### 6.2.3.3.1. Descripción de clases

<<Applet>> MainApplet	
- nombre	: String
- StreamClient	: StreamClient
- mplayerComponent	: EmbeddedMediaPlayerComponent
- btnPublicar	: button
- btnConsumir	: button
- btnlimpiarCache	: button
- peerServer	: Server
- peerClient	: Client
- objObserver	: Odservar
- sc	: StreamClient
+ init () : int	

<<JXTA>> Client	
- netPeerGroup	: PeerGroup
- groupAdvertisement	: PeerGroupAdvertisement
- discovery	: DiscoveryService
- pipeService	: PipeService
- manager	: Networkmanager
- configurator	: NetworkConfigurator
- MyJXTAPort	: String
- nombre	: String
- mdsadvtmp	: ModuleSpecAdvertisement
- objConsumer	: Consumer
+ startJxta (String JXTAPort, String nombre) : void	
+ startClient () : Strir	
+ startContent (String nService) : Strir	
+ startCalculation (String nService, File file) : Strir	
+ stopJxta () : void	
+ recoverService (String nameService, File file) : int	
+ findRemote (String nService) : Strir	

<<JXTA>> Consumer	
- mdsadv	: ModuleSpecAdvertisement
- msgDescriptor	: Descriptor
+ Consumer (ModuleSpecAdvertisement mdsadv) : void	
+ setMdsadv (ModuleSpecAdvertisement mdsadv) : void	
+ getDescriptor () : Descriptor	
+ setDescriptor (Descriptor descriptor) : void	
+ loadContent (Content cont) : String	
+ loadCalculation (Calculation calc) : void	
+ clientCalculation (int n, int port) : Long	

<<service>> StreamClient	
- mediaPlayerComponent	: EmbeddedMediaPlayerComponent
+ StreamClient () : void	
+ getInstanceEMPC () : EmbeddedMediaPlayerComponent	
+ getMediaPlayerComponent () : EmbeddedMediaPlayerComponent	



<<service>> StreamServer	
- mediaPlayerComponent	: EmbeddedMediaPlayerComponent
- mediaPlayerFactory	: MediaPlayerFactory
- mediaPlayer	: EmbeddedMediaPlayer
+ StreamServer ()	: void
+ getInstanceEMPC ()	: EmbeddedMediaPlayerComponent
+ startStreaming (String media, String options)	: void
+ stopStreaming ()	: void

<<Profile>> Servicio	
- name	: String
- URL	: String
- port	: String
- enabled	: boolean
+ getName ()	: String
+ setName (String name)	: void
+ getURL ()	: String
+ setUrl (String URL)	: void
+ getPort ()	: String
+ setPort (String port)	: void
+ isEnabled ()	: boolean
+ setEnabled (boolean enabled)	: void

<<tools>> Utilities	
- JXTA_CACHE	: String
+ eliminarCache (String file)	: void
+ formatHttpStream (String serverAddress, int serverport)	: String
+ formatRtspStream (String serverAddress, int serverPort)	: String

<<Profile>> Calculation	
- file	: File
+ Calculation (String name, String URL, String port)	: void
+ getFile ()	: File
+ setFile (File file)	: void

<<Profile>> Content	
+ Content (String name, String URL, String port)	: void

<<Profile>> Streaming	
+ Streaming (String name, String URL, String port)	: void

CalculationThread	
- socket	: Socket
- dos	: DataOutputStream
- dis	: DataInputStream
- idSessio	: int
+ CalculationThread (Socket socket, int id)	: void
+ close ()	: void
+ run ()	: void

ContentThread	
- socket	: Socket
- dos	: DataOutputStream
- dis	: DataInputStream
- idSessio	: int
- cont	: Content
+ ContentThread (Socket socket, int id)	: void
+ close ()	: void
+ run ()	: void

#### 6.2.3.4. Diagramas de Secuencia

A continuación, se presentan los diagramas de secuencia para los procesos de crear nodo y conectar a la red superpuesta P2P, publicar, replicar, prestar, buscar, consumir y recuperar servicios.

- **Crear nodo y conectar a la red**

En la Figura 13 se describe la secuencia de pasos del proceso que lleva a cabo el sistema para crear un nodo y posteriormente conectarse a la red superpuesta P2P; este proceso se lleva a cabo de forma automática cuando el usuario ejecuta los procesos de publicación o consumo de servicios. Durante este proceso se da inicio a la red superpuesta P2P mediante el método startJxta (JXTAPort, VLCPort, FTPPort, CALPort, nombre) con todos los parámetros necesarios para este propósito.

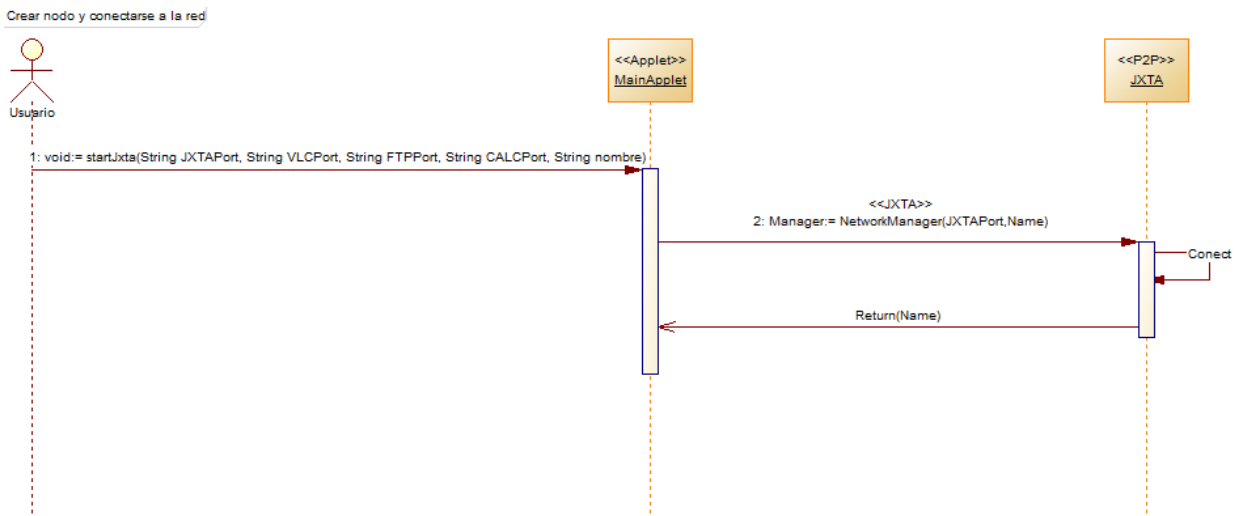


Figura 13. Diagrama de secuencia crear nodo y conectar a la red.

- **Publicar servicios**

En la Figura 14 se describe la secuencia de pasos para el proceso de publicar servicios, para este propósito el sistema utiliza el método Publish(Perfil) y mediante el parámetro Perfil se dan las especificaciones del tipo de servicio que se va a publicar sobre la red superpuesta P2P por parte del servidor.

Después de realizar la publicación de servicios, el nodo servidor ya queda disponible para prestar los servicios que publico sobre la red superpuesta P2P, quedando en capacidad para recibir solicitudes de clientes que deseen consumir cualquiera de los servicios publicados.

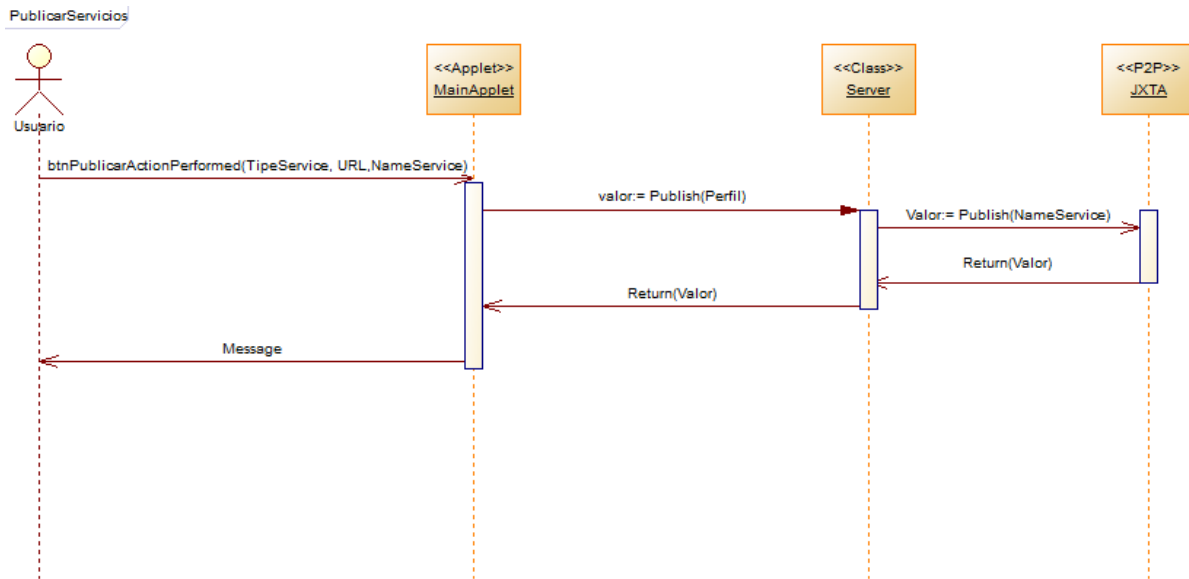


Figura 14. Diagrama de secuencia publicar servicios.

- **Replicar servicios**

En la Figura 15 se describe la secuencia de pasos para el proceso de replicación de servicios, este proceso se lleva a cabo de forma automática por parte del sistema; para este propósito el sistema utiliza el método `ReplicateService(Perfil, NameServer)`, donde el parámetro `Perfil` tiene las especificaciones del servicio que se va a replicar en la red superpuesta P2P y el parámetro `NameServer` especifica el nombre del servidor que desea replicar el servicio, el servidor publica un anuncio en la red solicitando a los otros nodos su disponibilidad para realizar la replicación.

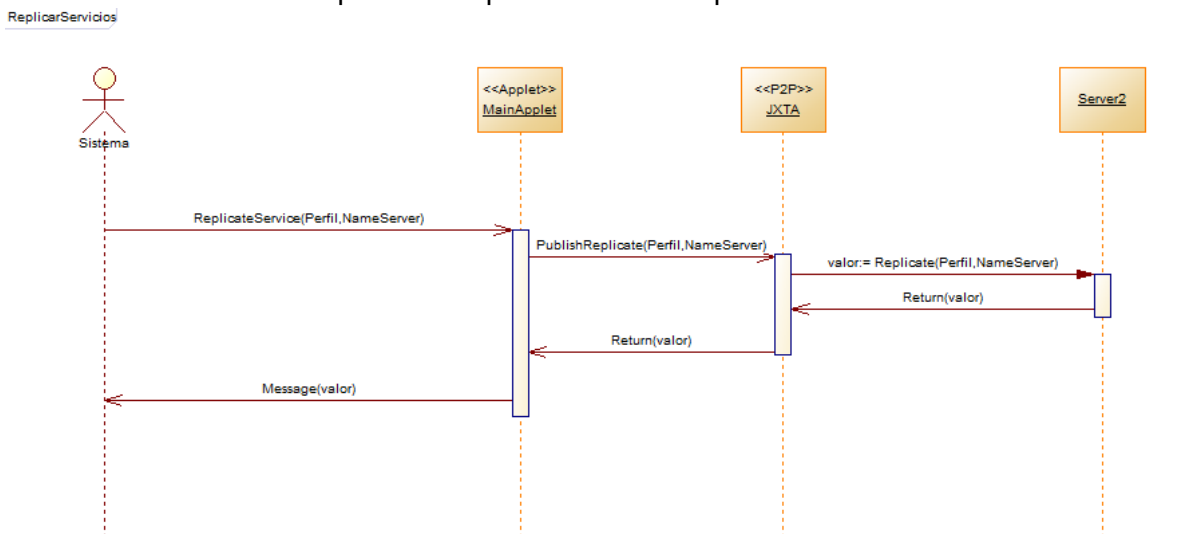


Figura 15. Diagrama de secuencia replicar servicios.

- **Prestar servicios**

En la Figura 16 se describen la secuencia de pasos para el proceso de prestar servicios sobre la red superpuesta P2P, este proceso es realizado de forma automática por parte del sistema mediante el rol de servidor. Para prestar cualquiera de los servicios, el servidor utiliza el método `ManagerService(Perfil)`, donde el parámetro `Perfil` especifica qué tipo de servicio se va a prestar.

En este proceso se utiliza el objeto `Manager` que es el encargado de implementar todas las funciones necesarias para prestar cada determinado servicio.

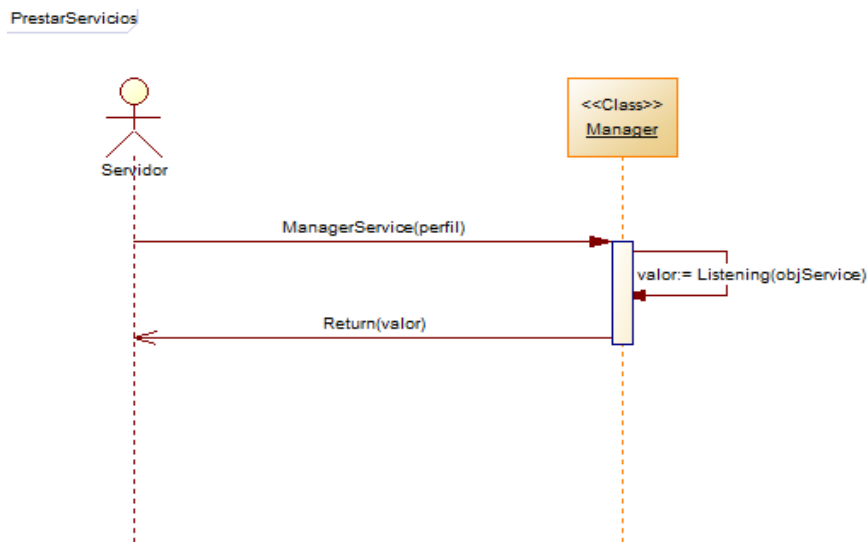


Figura 16. Diagrama de secuencia prestar servicios.

- **Consumir servicios.**

En la Figura 17 se muestra la secuencia de pasos para el proceso que realiza el sistema para consumir servicios; para este propósito el sistema mediante el rol de cliente utiliza el método `StartService (NameService)` donde el parámetro `NameService` especifica el nombre del servicio que se desea consumir; a través de este método se realiza la búsqueda sobre la red superpuesta P2P tanto de avisos locales, como también remotos, esto con el fin de encontrar los servidores que tienen disponible el servicio solicitado y obtener los parámetros necesarios para poder consumir dicho servicio.

En este proceso se utiliza el objeto `Consumer`, el cual es el encargado de llevar a cabo todas las funciones necesarias para cargar el servicio que el cliente solicitó para consumir.



ConsumirServicios

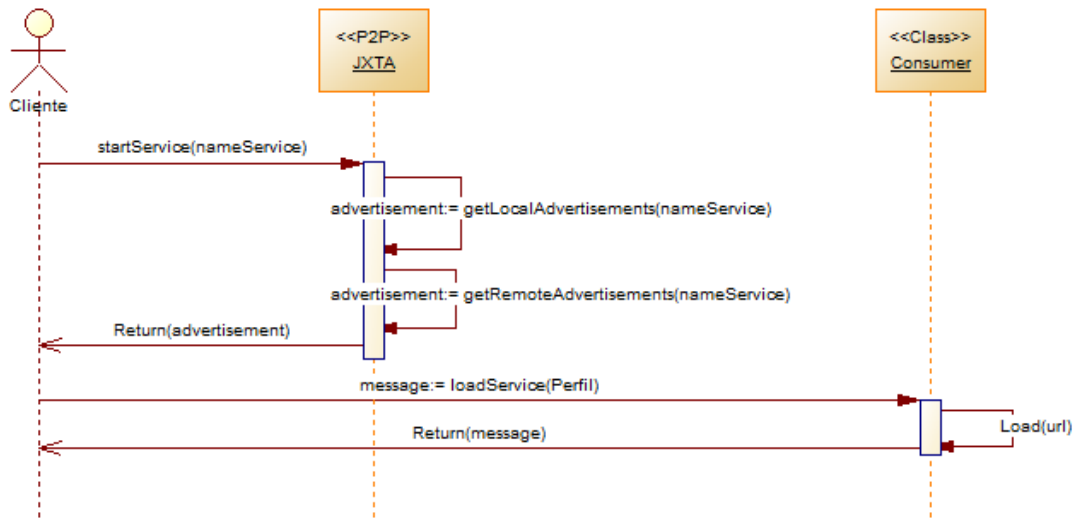


Figura 17. Diagrama de secuencia consumir servicios.

- **Buscar servicios**

Cuando un servidor publica sus servicios sobre la red superpuesta P2P, los nodos cliente pueden buscarlos y consumirlos. En la Figura 18 se presenta la secuencia de pasos para el proceso de buscar servicios sobre la red superpuesta P2P; para este propósito el sistema por medio del rol cliente utiliza el método StartService (objPerfil) para buscar anuncios de servicios, tanto locales como remotos sobre la red superpuesta P2P, el parámetro objPerfil especifica el tipo de servicio que se desea buscar sobre la red.

BuscarServicios

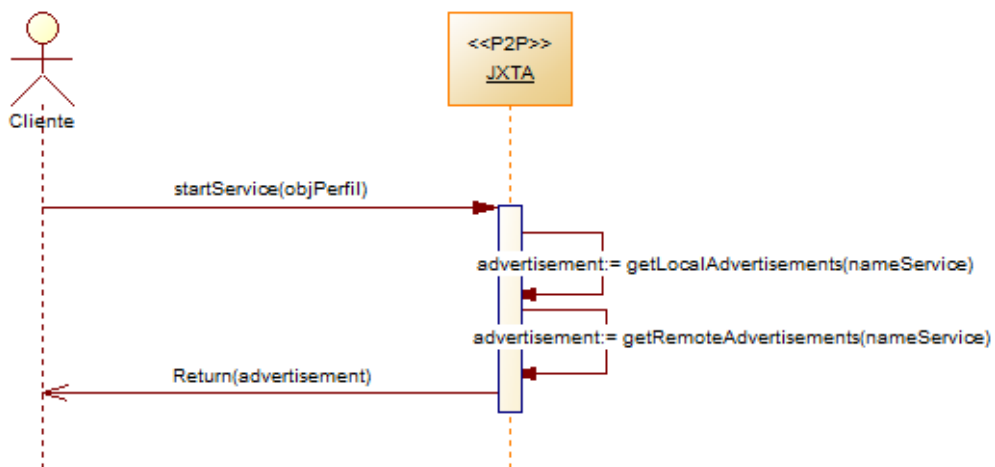


Figura 18. Diagrama de secuencia buscar servicios.

- **Recuperar servicios**

Cuando un servidor que está prestando un servicio se desconecta de forma voluntaria o involuntaria, los clientes que se encuentran consumiendo dicho servicio se ven afectados. Para solucionar este problema se implementó el objeto Observer, que es el encargado de monitorear la condición de los servicios, para que en caso de desconexión del servidor que presta el servicio dispare el mecanismo de recuperación del servicio.

En la Figura 19 se presenta la secuencia de pasos para el proceso de recuperar servicios sobre la red superpuesta P2P; para este propósito el sistema utiliza el método findService (Perfil) para buscar un nuevo servicio sobre la red superpuesta P2P y el parámetro Perfil especifica el tipo de servicio que se quiere recuperar.

1 RecuperarServicios

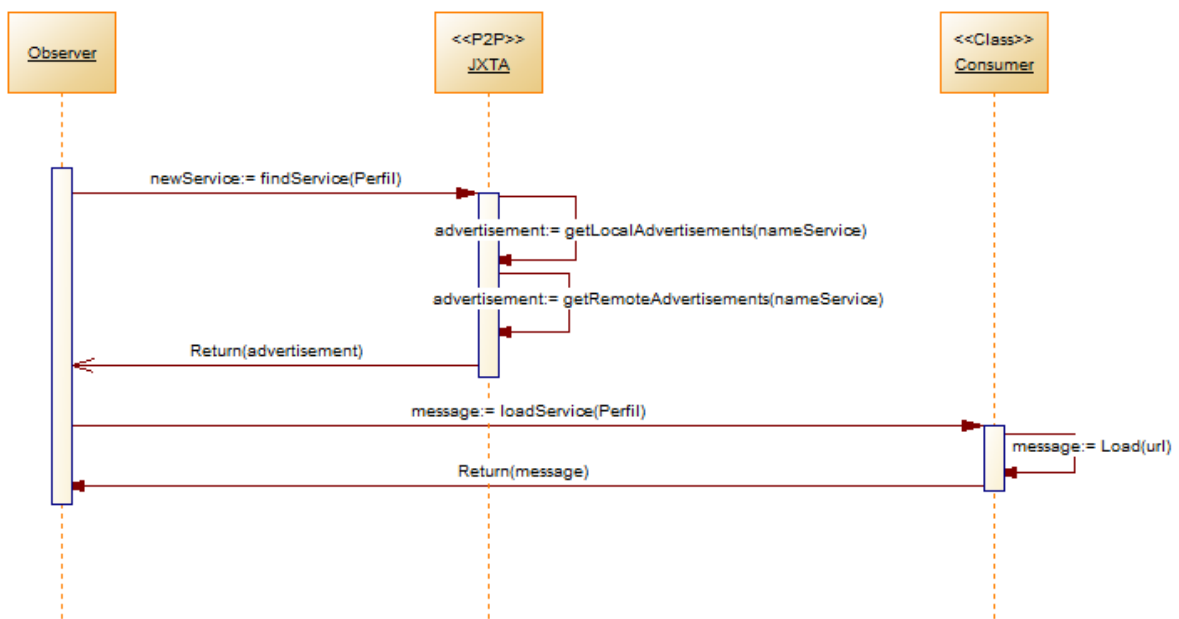


Figura 19. Diagrama de secuencia recuperar servicios.

#### 6.2.4. Planificación de entregas

Para la planificación de entregas se tuvo en cuenta la priorización de historias de usuario, descritos en la tabla 6. Con la cual se hizo la planificación de la siguiente manera:

- **Primera iteración:** se tuvieron en cuenta las historias de usuario: solicitar un servicio de media streaming, publicar un servicio de media streaming y replicar el servicio.

- **Segunda iteración:** se tuvieron presente las historias de usuario: solicitar un servicio de contenidos, publicar un servicio de contenidos y replicar el servicio.
- **Tercera iteración:** se tuvieron en cuenta las historias de usuario: solicitar un servicio de cálculos computacionales, publicar un servicio de cálculos computacionales y replicar el servicio.
- **Cuarta iteración:** se llevaron a cabo el despliegue del prototipo y las pruebas finales para la simulación.

En este capítulo se obtuvo la especificación de requisitos para el prototipo, así como también, el diseño del prototipo representado a través de diagramas de casos de uso, clases, secuencia e historias de usuario.

## **Capítulo 7. EVALUACIÓN Y ANÁLISIS DEL PROTOTIPO SOFTWARE**

En este capítulo se presenta la evaluación y análisis de los resultados obtenidos de las pruebas realizadas al prototipo software.

### **7.1. Diseño experimental**

#### **7.1.1. Estudio de caso: Evaluación prototipo software**

##### **7.1.1.1. Pregunta de investigación:**

A partir de la pregunta de investigación del proyecto se requirió evaluar el prototipo software con respecto a la disponibilidad de servicios en redes superpuestas P2P no estructuradas. La pregunta para este caso de estudio es: ¿Cómo se puede apoyar la disponibilidad de servicios en redes superpuestas P2P no estructuradas?

##### **7.1.1.2. Objetivo del estudio de caso:**

Verificar mediante el prototipo software, la aplicabilidad de la técnica de disponibilidad de servicios en redes superpuestas P2P no estructuradas propuesta en este trabajo de grado.

##### **7.1.1.3. Selección del caso de estudio:**

Para la realización de la evaluación se seleccionó la sala de computación Sala 4 de la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca, donde se montó la topología de red física propia.

##### **7.1.1.4. Contexto del caso:**

Para llevar a cabo la experimentación y simulación de la red P2P se utilizó una sala de laboratorio, sobre la que está montada la topología de red física, tipo LAN 10/100 y en donde se realizaron las pruebas de manera controlada. Está compuesta por 15 computadores, cada uno con diferentes características hardware y software; se utilizaron sistemas operativos instalados como máquinas virtuales para aumentar la cantidad de clientes en la red.

El laboratorio estaba dedicado sólo para la ejecución del prototipo software, no había usuarios consumiendo ancho de banda, ni capacidad de procesamiento en los equipos. La especificación de cada equipo utilizado en las pruebas se encuentra en el CD, en el documento de Anexos en el Anexo G: Especificación de los equipos utilizados en las pruebas.

### 7.1.1.5. Diseño del estudio

Para la realización el diseño del estudio, se definieron una serie de escenarios posibles, para probar las características de disponibilidad de servicios en la técnica para soporte de la disponibilidad de servicios.

Los escenarios para este tipo de red y para los diferentes tipos de servicios varían de acuerdo al número de servidores, clientes y los tipos de servicio. Los nodos servidor pueden ser replica de servicios de otros nodos, para cada tipo de servicio cada escenario se lleva a cabo con el mismo tipo de unidad de prueba; en el caso de cálculos se presta el servicio de descarga de un archivo con un tamaño medido de 4 megabytes, el de cálculos computacionales es un archivo con una cantidad de operaciones fija y para media streaming es la visualización de un flujo de streaming que se estará presentando en los clientes.

Las pruebas están diseñadas y son controladas para probar los escenarios de prestación, replicación y recuperación de servicios, para cada tipo de servicio. Los escenarios definidos para la verificación de la técnica son los que se describen en la Tabla 7:

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.

Numero escenario	Nodos servidor	Numero de fallos	Nodos cliente	Tipo servicio	Característica
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 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.

Numero escenario	Nodos servidor	Numero de fallos	Nodos cliente	Tipo servicio	Característica
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 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 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 los nodos servidores y antes que se consuma el servicio se presentarán ocho fallos de los nodos servidores que prestan el servicio.

Numero escenario	Nodos servidor	Numero de fallos	Nodos cliente	Tipo servicio	Característica
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 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.



Numero escenario	Nodos servidor	Numero de fallos	Nodos cliente	Tipo servicio	Característica
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 7. Definición de escenarios de pruebas

En la Tabla 8, se muestra el formato utilizado para registrar los resultados de los escenarios de prueba.

Escenario número: Descripción.					
Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de Recuperación Milisegundos
<b>Descripción de resultado:</b>					

Tabla 8. Formato de registro de resultados.

A continuación se describen los campos del formato de registro de resultados:

**Escenario número:** número y descripción del escenario especificado anteriormente.

**Número equipo:** corresponde al número del equipo que se le asignó previamente.

**Rol:** es el rol que desempeña cada nodo en la red. Puede ser un nodo cliente, un nodo servidor, o un servidor de réplica.

**IP:** es la dirección IP de cada nodo conectado en la red.

**Servidor consumo actual:** representa el nodo por el cual está consumiendo el servicio.

**Servidor consumo nuevo:** corresponde al nodo servidor al cual se conectó nuevamente después de un fallo en su servidor consumo actual.

**Tiempo de recuperación:** es el tiempo en milisegundos en que tarda el nodo cliente en recuperar el servicio y terminar de prestar dicho servicio.

**Descripción de resultado:** es la descripción de los resultados que se obtuvieron en el desarrollo del escenario de prueba.

#### 7.1.1.6. Desarrollo del caso:

En la Figura 20 se describe la estructura de los escenarios de prueba, en la cual los cuatro primeros escenarios (1, 2, 3, 4) corresponden a las pruebas realizadas al servicio de contenidos; los siguientes cuatro escenarios (5, 6, 7, 8) corresponden a las pruebas realizadas al servicio de cálculos computacionales; los siguientes cuatro escenarios (9, 10, 11, 12) corresponden a las pruebas realizadas al servicio de media streaming y finalmente los últimos dos escenarios (13, 14) corresponden a las pruebas realizadas con los tres tipos de servicio.

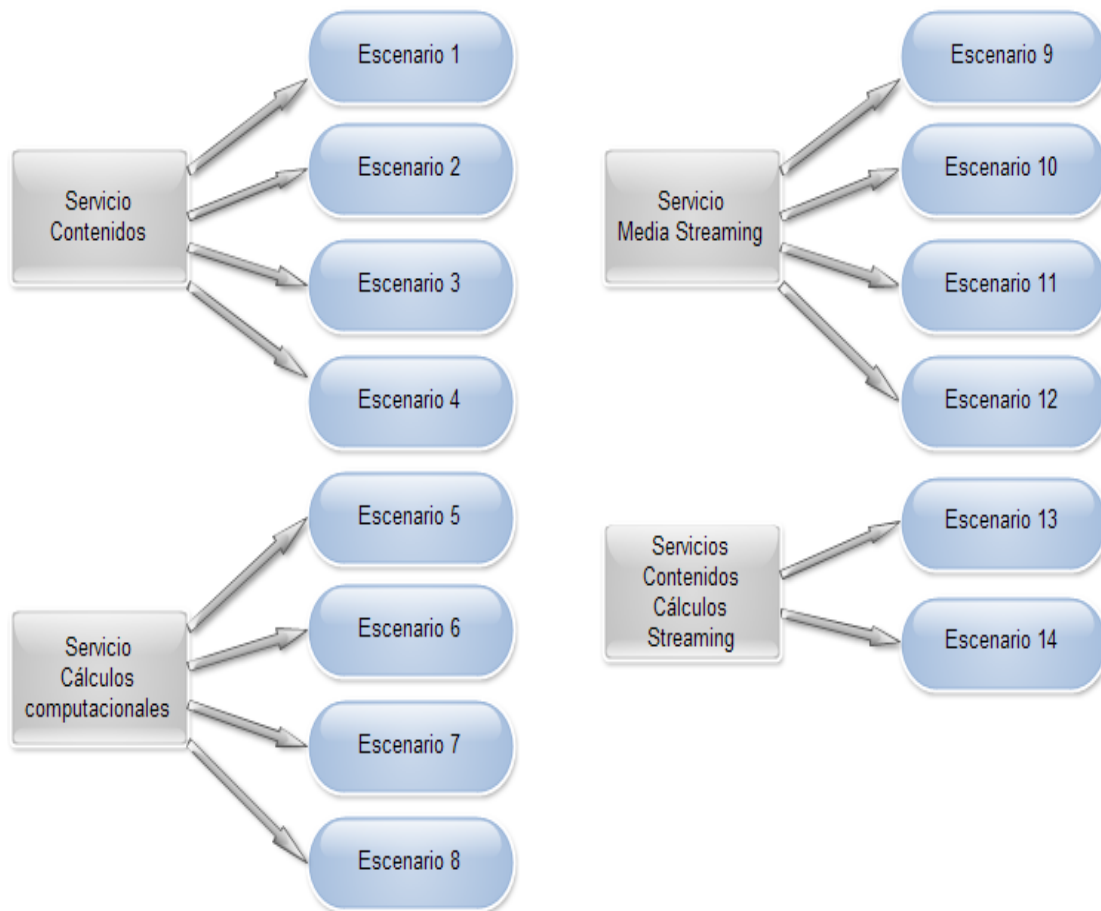


Figura 20. Estructura de escenarios de prueba.

Para cada escenario de prueba se presenta: una figura que describe gráficamente el escenario de prueba; y una tabla que presenta los resultados de las pruebas.

#### 7.1.1.7. Resultados:

A continuación se presentan los resultados obtenidos en el estudio de caso evaluación prototipo software con respecto a la técnica de disponibilidad de servicios en redes superpuesta P2P no estructuradas propuesta en este trabajo de grado. A continuación se muestran los resultados de cada uno de los escenarios de prueba:

- **Escenario de prueba número 1**

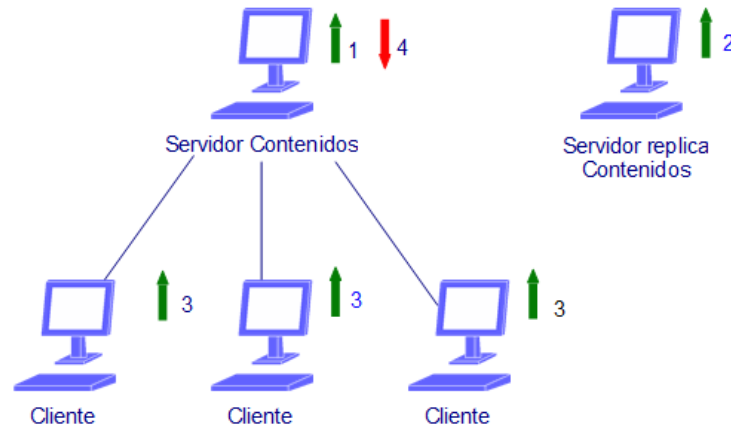


Figura 21. Representación del escenario de prueba número 1.

**Escenario número 1:** los nodos se conectarán a la red, el nodo servidor y replica serán iniciados primero, luego se inician los nodos clientes 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.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA	NA	
2	Replica	192.168.127.2	NA	NA	
3	Cliente	192.168.127.3	192.168.127.1	192.168.127.2	2466
4	Cliente	192.168.127.4	192.168.127.1	192.168.127.2	2171
5	Cliente	192.168.127.5	192.168.127.1	192.168.127.2	2660
6	Cliente	192.168.127.6	192.168.127.1	192.168.127.1	1720
7	Cliente	192.168.127.7	192.168.127.1	192.168.127.2	2366
8	Cliente	192.168.127.8	192.168.127.1	192.168.127.2	2450
9	Cliente	192.168.127.9	192.168.127.1	192.168.127.1	1670
10	Cliente	192.168.127.10	192.168.127.1	192.168.127.2	3240

**Descripción de resultado:** se puede observar que los tiempos de recuperación del servicio no son muy distantes el uno del otro, a pesar de que el servidor se inició primero y que algunos de los nodos cliente recuperaron el servicio del nodo replica; luego de detener el servidor, solo dos clientes consumieron el servicio del nodo servidor.

Tabla 9. Tabla de resultados escenarios de pruebas número. 1

- **Escenario de prueba número 2**

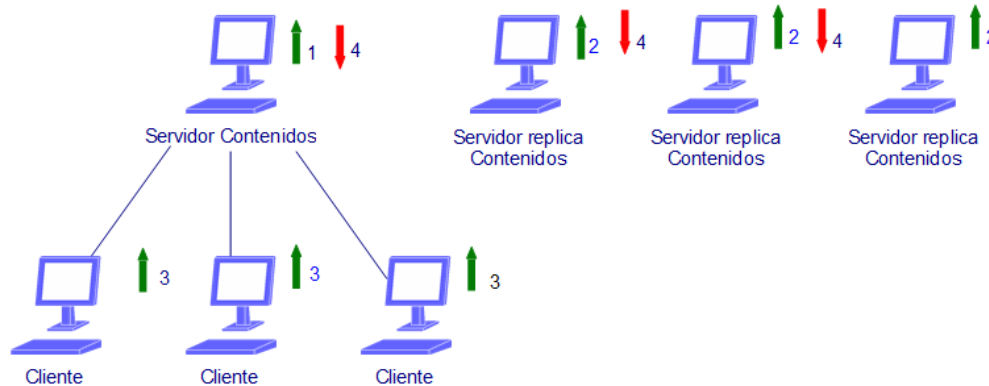


Figura 22. Representación del escenario de prueba número 2.

**Escenario número 2:** 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.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA	NA	NA
2	Replica	192.168.127.2	NA	NA	NA
3	Replica	192.168.127.3	NA	NA	NA
4	Replica	192.168.127.4	NA	NA	NA
5	Cliente	192.168.127.5	192.168.127.1	192.168.127.4	3405
6	Cliente	192.168.127.6	192.168.127.1	192.168.127.4	3606
7	Cliente	192.168.127.7	192.168.127.1	192.168.127.4	3566
8	Cliente	192.168.127.8	192.168.127.1	192.168.127.4	3590
9	Cliente	192.168.127.9	192.168.127.1	192.168.127.4	3478
10	Cliente	192.168.127.10	192.168.127.1	192.168.127.4	3698
11	Cliente	192.168.127.11	192.168.127.1	192.168.127.4	3567
12	Cliente	192.168.127.12	192.168.127.1	192.168.127.4	3700

**Descripción de resultado:** en este escenario se incrementaron los valores de los tiempos de recuperación y ningún cliente consumió el servicio del nodo servidor y sus dos primeras replicas, puesto que se presentaron fallos, todos terminan consumiendo el servicio de la tercera replica que es la que finalmente queda disponible.

Tabla 10. Tabla de resultados escenarios de pruebas número. 2

• **Escenario de prueba número 3**

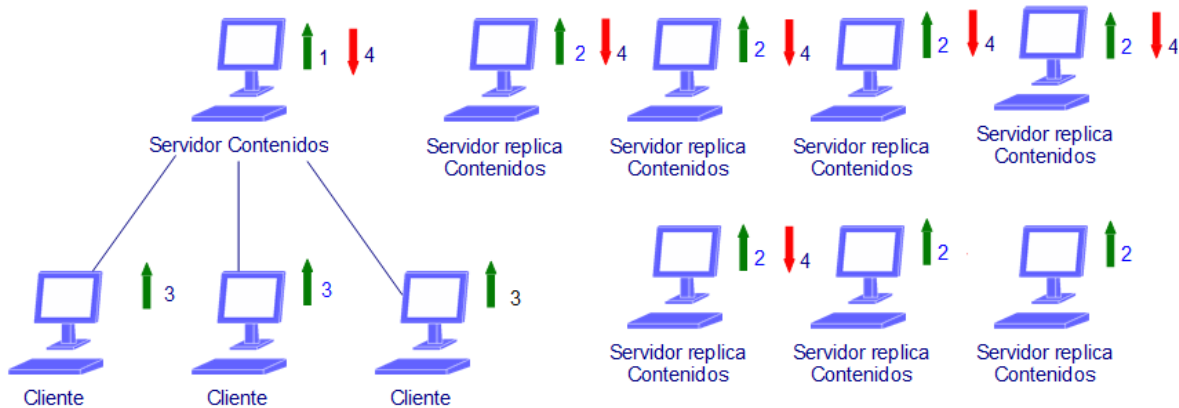


Figura 23. Representación del escenario de prueba número 3.

**Escenario número 3:** 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.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA	NA	NA
2	Replica	192.168.127.2	NA	NA	NA
3	Replica	192.168.127.3	NA	NA	NA
4	Replica	192.168.127.4	NA	NA	NA
5	Replica	192.168.127.5	NA	NA	NA
6	Replica	192.168.127.6	NA	NA	NA
7	Replica	192.168.127.7	NA	NA	NA
8	Replica	192.168.127.8	NA	NA	NA
9	Cliente	192.168.127.9	192.168.127.1	192.168.127.8	8905
10	Cliente	192.168.127.10	192.168.127.1	192.168.127.8	7990
11	Cliente	192.168.127.11	192.168.127.1	192.168.127.8	7560
12	Cliente	192.168.127.12	192.168.127.1	192.168.127.8	8067
13	Cliente	192.168.127.13	192.168.127.1	192.168.127.8	9056

**Descripción de resultado:** muestra la misma funcionalidad que el escenario anterior solo que este presente más número de fallos por consiguiente el tiempo de recuperación aumento, la prueba fue controlada para que los clientes terminaran consumiendo del último nodo que replicaba el servicio.

Tabla 11. Tabla de resultados escenarios de pruebas número. 3

- **Escenario de prueba número 4**

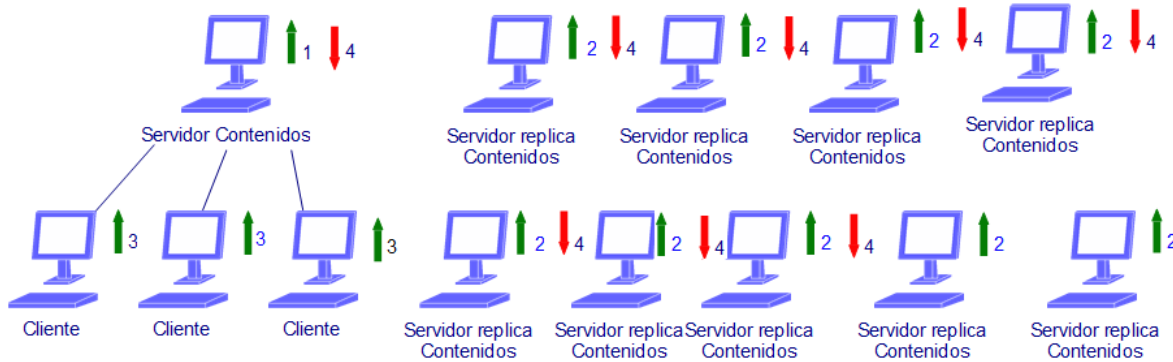


Figura 24. Representación del escenario de prueba número 4.

**Escenario número 4:** 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 contenidos de los nodos servidores y antes que se consuma el servicio se presentarán ocho fallos de los nodos servidores que prestan el servicio.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA	NA	NA
2	Replica	192.168.127.2	NA	NA	NA
3	Replica	192.168.127.3	NA	NA	NA
4	Replica	192.168.127.4	NA	NA	NA
5	Replica	192.168.127.5	NA	NA	NA
6	Replica	192.168.127.6	NA	NA	NA
7	Replica	192.168.127.7	NA	NA	NA
8	Replica	192.168.127.8	NA	NA	NA
9	Replica	192.168.127.9	NA	NA	NA
10	Replica	192.168.127.10	NA	NA	NA
11	Cliente	192.168.127.11	192.168.127.1	192.168.127.10	9870
12	Cliente	192.168.127.12	192.168.127.1	192.168.127.10	10067
13	Cliente	192.168.127.13	192.168.127.1	192.168.127.10	8900
14	Cliente	192.168.127.14	192.168.127.1	192.168.127.10	9876
15	Cliente	192.168.127.15	192.168.127.1	192.168.127.10	11069

**Descripción de resultado:** se aumentó el número de réplicas a un valor que representan un nivel de disponibilidad óptimo, como se explicara más adelante en el análisis y los tiempos aumentaron significativamente.

Tabla 12. Tabla de resultados escenarios de pruebas número. 4

- **Escenario de prueba número 5**

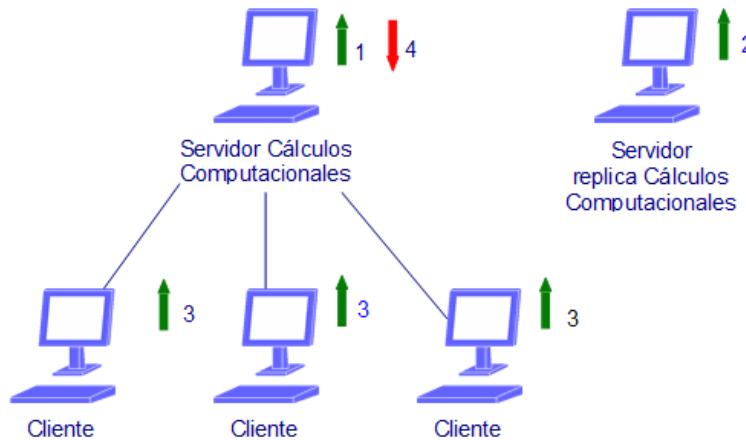


Figura 25. Representación del escenario de prueba número 5.

**Escenario número 5:** 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.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA	NA	
2	Replica	192.168.127.2	NA	NA	
3	Cliente	192.168.127.3	192.168.127.1	192.168.127.2	4466
4	Cliente	192.168.127.4	192.168.127.1	192.168.127.2	4171
5	Cliente	192.168.127.5	192.168.127.1	192.168.127.2	5660
6	Cliente	192.168.127.6	192.168.127.1	192.168.127.2	5320
7	Cliente	192.168.127.7	192.168.127.1	192.168.127.2	4766
8	Cliente	192.168.127.8	192.168.127.1	192.168.127.2	4890
9	Cliente	192.168.127.9	192.168.127.1	192.168.127.2	5270
10	Cliente	192.168.127.10	192.168.127.1	192.168.127.2	3900

**Descripción de resultado:** para este tipo de servicio los tiempos se aumentan debido a la cantidad que cálculos que realice el servidor para el cliente además no son muy distantes el uno del otro a pesar que el servidor se inició primero y que algunos de los nodos cliente recuperaron el servicio del nodo replica.

Tabla 13. Tabla de resultados escenarios de pruebas número. 5



- **Escenario de prueba número 6**

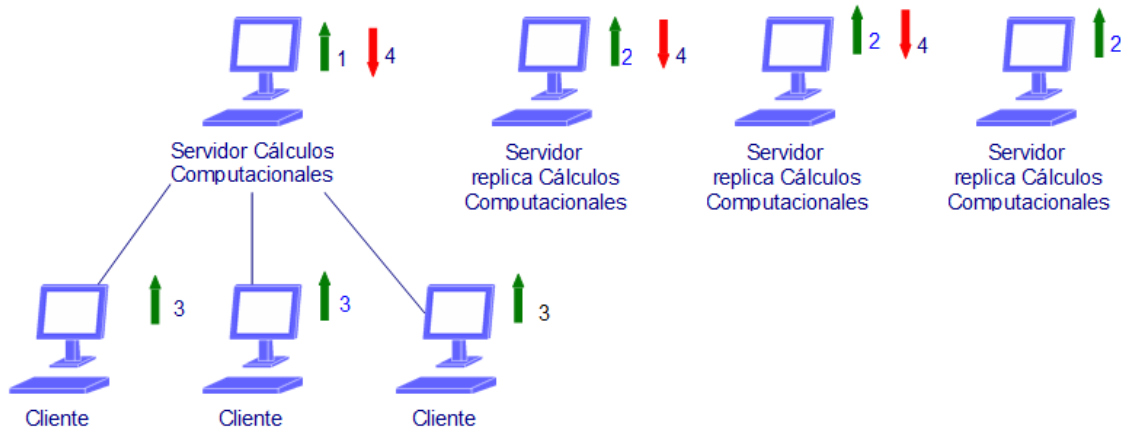


Figura 26. Representación del escenario de prueba número 6.

**Escenario número 6:** 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 de cálculos computacionales.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de Recuperación Milisegundos
1	Servidor	192.168.127.1	NA	NA	NA
2	Replica	192.168.127.2	NA	NA	NA
3	Replica	192.168.127.3	NA	NA	NA
4	Replica	192.168.127.4	NA	NA	NA
5	Cliente	192.168.127.5	192.168.127.1	192.168.127.4	7405
6	Cliente	192.168.127.6	192.168.127.1	192.168.127.4	6906
7	Cliente	192.168.127.7	192.168.127.1	192.168.127.4	8566
8	Cliente	192.168.127.8	192.168.127.1	192.168.127.4	6590
9	Cliente	192.168.127.9	192.168.127.1	192.168.127.4	7478
10	Cliente	192.168.127.10	192.168.127.1	192.168.127.4	8698
11	Cliente	192.168.127.11	192.168.127.1	192.168.127.4	6567
12	Cliente	192.168.127.12	192.168.127.1	192.168.127.4	6370

**Descripción de resultado:** en este escenario se incrementaron los valores de los tiempos de recuperación y ningún cliente consumió el servicio del nodo servidor y sus dos primeras réplicas; puesto que se presentaron fallos, todos terminan consumiendo el servicio de la tercera réplica, que es la que finalmente queda disponible.

Tabla 14. Tabla de resultados escenarios de pruebas número. 6

• **Escenario de prueba número 7**

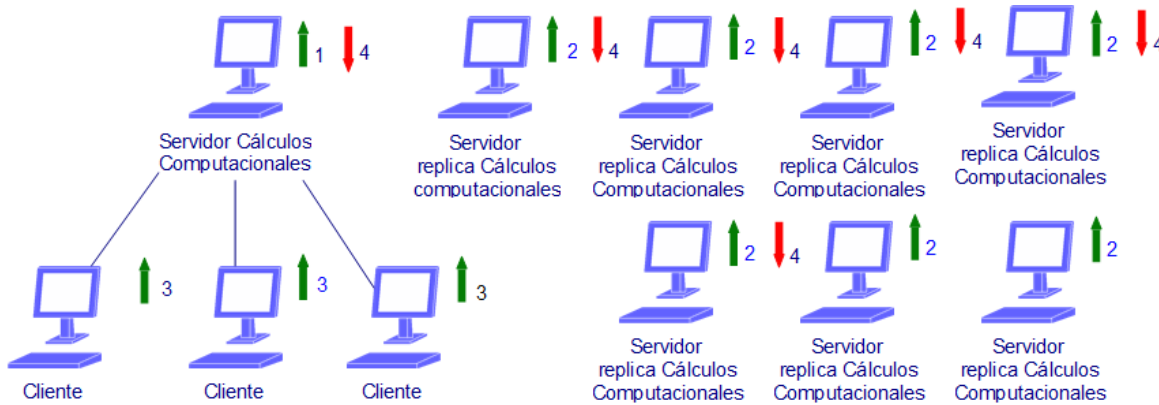


Figura 27. Representación del escenario de prueba número 7.

**Escenario número 7:** 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 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 de cálculos computacionales.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA	NA	NA
2	Replica	192.168.127.2	NA	NA	NA
3	Replica	192.168.127.3	NA	NA	NA
4	Replica	192.168.127.4	NA	NA	NA
5	Replica	192.168.127.5	NA	NA	NA
6	Replica	192.168.127.6	NA	NA	NA
7	Replica	192.168.127.7	NA	NA	NA
8	Replica	192.168.127.8	NA	NA	NA
9	Cliente	192.168.127.9	192.168.127.1	192.168.127.8	12905
10	Cliente	192.168.127.10	192.168.127.1	192.168.127.8	11990
11	Cliente	192.168.127.11	192.168.127.1	192.168.127.8	12905
12	Cliente	192.168.127.12	192.168.127.1	192.168.127.8	13080
13	Cliente	192.168.127.13	192.168.127.1	192.168.127.8	12790

**Descripción de resultado:** es el mismo escenario anterior, solo que en este se presento más número de fallos; por consiguiente, el tiempo de recuperación aumento, la prueba fue controlada para que los clientes terminaran consumiendo del último nodo que replicaba el servicio.

Tabla 15. Tabla de resultados escenarios de pruebas número. 7

- **Escenario de prueba número 8**

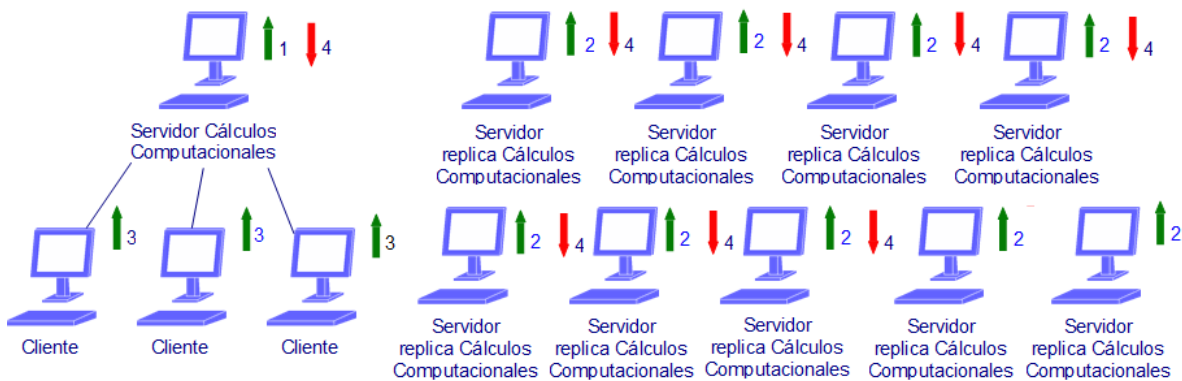


Figura 28. Representación del escenario de prueba número 8.

**Escenario número 8:** 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 de cálculos computacionales.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA	NA	NA
2	Replica	192.168.127.2	NA	NA	NA
3	Replica	192.168.127.3	NA	NA	NA
4	Replica	192.168.127.4	NA	NA	NA
5	Replica	192.168.127.5	NA	NA	NA
6	Replica	192.168.127.6	NA	NA	NA
7	Replica	192.168.127.7	NA	NA	NA
8	Replica	192.168.127.8	NA	NA	NA
9	Replica	192.168.127.9	NA	NA	NA
10	Replica	192.168.127.10	NA	NA	NA
11	Cliente	192.168.127.11	192.168.127.1	192.168.127.10	15870
12	Cliente	192.168.127.12	192.168.127.1	192.168.127.10	14967
13	Cliente	192.168.127.13	192.168.127.1	192.168.127.10	16900
14	Cliente	192.168.127.14	192.168.127.1	192.168.127.10	16876
15	Cliente	192.168.127.15	192.168.127.1	192.168.127.10	15969

**Descripción de resultado:** se aumentó el número de réplicas a un valor que representan un nivel de disponibilidad óptimo, como se explicara más adelante en el análisis y los tiempos aumentaron significativamente.

Tabla 16. Tabla de resultados escenarios de pruebas número. 8

- **Escenario de prueba número 9**

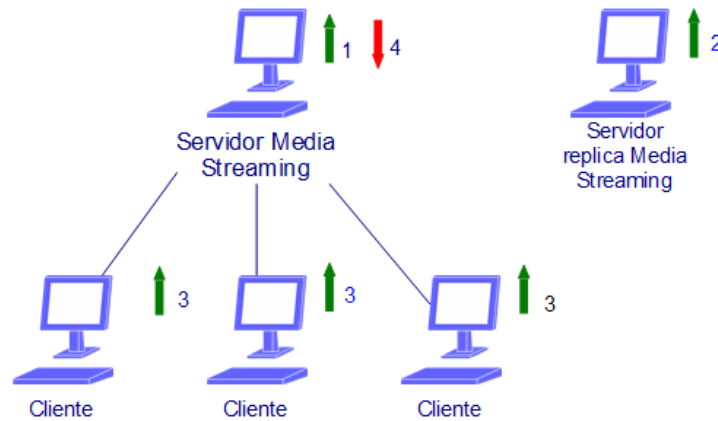


Figura 29. Representación del escenario de prueba número 9.

**Escenario número 9:** 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.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA	NA	
2	Replica	192.168.127.2	NA	NA	
3	Cliente	192.168.127.3	192.168.127.1	192.168.127.2	46470
4	Cliente	192.168.127.4	192.168.127.1	192.168.127.2	53451
5	Cliente	192.168.127.5	192.168.127.1	192.168.127.2	48845
6	Cliente	192.168.127.6	192.168.127.1	192.168.127.2	64685
7	Cliente	192.168.127.7	192.168.127.1	192.168.127.2	45800
8	Cliente	192.168.127.8	192.168.127.1	192.168.127.2	50546
9	Cliente	192.168.127.9	192.168.127.1	192.168.127.2	37778
10	Cliente	192.168.127.10	192.168.127.1	192.168.127.2	33597

**Descripción de resultado:** se puede observar que los tiempos de recuperación del servicio no son muy distantes el uno del otro, pero si se incrementan con respecto a los demás servicios; puesto que el servidor se inició primero y todos los nodos cliente recuperaron el servicio del nodo replica que quedo disponible luego de detener el servidor.

Tabla 17. Tabla de resultados escenarios de pruebas número. 9

• **Escenario de prueba número 10**

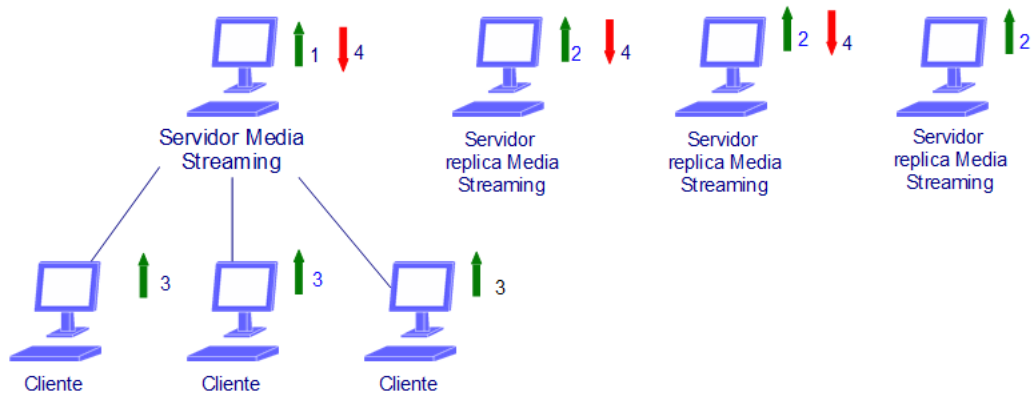


Figura 30. Representación del escenario de prueba número. 10.

**Escenario número 10:** los nodos se conectarán a la red, el nodo servidor y replica serán iniciados 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 de los nodos servidores que prestan el servicio.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA	NA	NA
2	Replica	192.168.127.2	NA	NA	NA
3	Replica	192.168.127.3	NA	NA	NA
4	Replica	192.168.127.4	NA	NA	NA
5	Cliente	192.168.127.5	192.168.127.1	192.168.127.4	54405
6	Cliente	192.168.127.6	192.168.127.1	192.168.127.4	44606
7	Cliente	192.168.127.7	192.168.127.1	192.168.127.4	44566
8	Cliente	192.168.127.8	192.168.127.1	192.168.127.4	53990
9	Cliente	192.168.127.9	192.168.127.1	192.168.127.4	54478
10	Cliente	192.168.127.10	192.168.127.1	192.168.127.4	43678
11	Cliente	192.168.127.11	192.168.127.1	192.168.127.4	54567
12	Cliente	192.168.127.12	192.168.127.1	192.168.127.4	43906

**Descripción de resultado:** en este escenario se incrementaron los valores de los tiempos de recuperación y ningún cliente consumió el servicio del nodo servidor y sus dos primeras réplicas; puesto que se presentaron fallos, todos terminan consumiendo el servicio de la tercera replica que es la que finalmente queda disponible, para este tipo de servicio los tiempos son muy grandes.

Tabla 18. Tabla de resultados escenarios de pruebas número. 10

- **Escenario de prueba número 11**

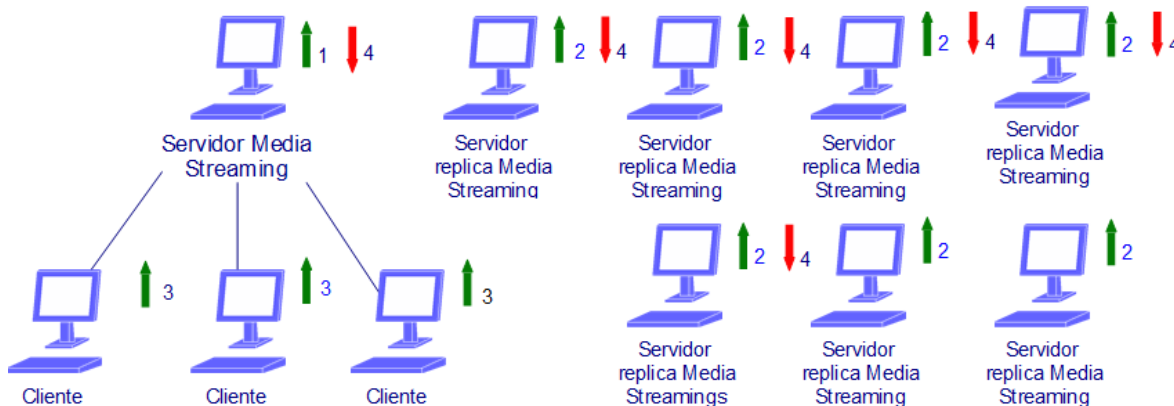


Figura 31. Representación del escenario de prueba número. 11.

**Escenario número 11:** 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.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA	NA	NA
2	Replica	192.168.127.2	NA	NA	NA
3	Replica	192.168.127.3	NA	NA	NA
4	Replica	192.168.127.4	NA	NA	NA
5	Replica	192.168.127.5	NA	NA	NA
6	Replica	192.168.127.6	NA	NA	NA
7	Replica	192.168.127.7	NA	NA	NA
8	Replica	192.168.127.8	NA	NA	NA
9	Cliente	192.168.127.9	192.168.127.1	192.168.127.8	61195
10	Cliente	192.168.127.10	192.168.127.1	192.168.127.8	81090
11	Cliente	192.168.127.11	192.168.127.1	192.168.127.8	71050
12	Cliente	192.168.127.12	192.168.127.1	192.168.127.8	69007
13	Cliente	192.168.127.13	192.168.127.1	192.168.127.8	81206

**Descripción de resultado:** presenta la misma funcionalidad que el escenario anterior, solo que en este se presentó más número de fallos; por consiguiente el tiempo de recuperación aumento, la prueba fue controlada para que los clientes terminaran consumiendo del último nodo que replicaba el servicio.

Tabla 19. Tabla de resultados escenarios de pruebas número. 11

- **Escenario de prueba número 12**

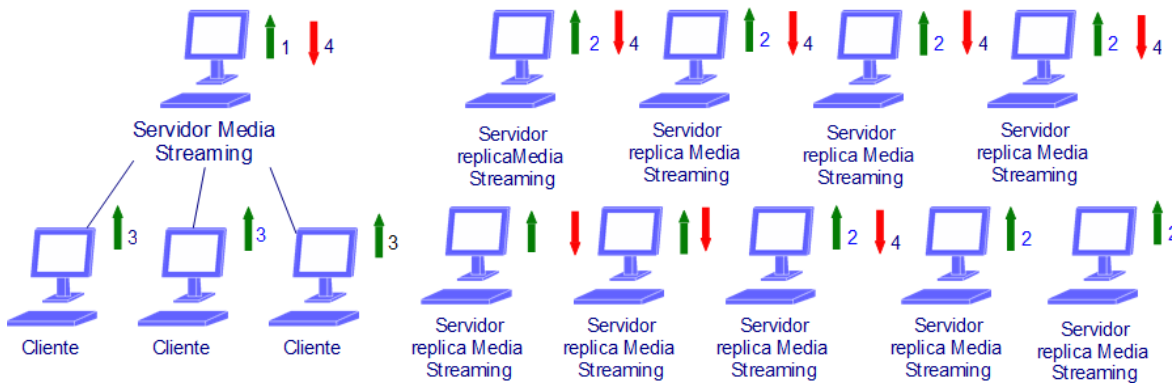


Figura 32. Representación del escenario de prueba número. 12.

**Escenario número 12:** 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.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA	NA	NA
2	Replica	192.168.127.2	NA	NA	NA
3	Replica	192.168.127.3	NA	NA	NA
4	Replica	192.168.127.4	NA	NA	NA
5	Replica	192.168.127.5	NA	NA	NA
6	Replica	192.168.127.6	NA	NA	NA
7	Replica	192.168.127.7	NA	NA	NA
8	Replica	192.168.127.8	NA	NA	NA
9	Replica	192.168.127.9	NA	NA	NA
10	Replica	192.168.127.10	NA	NA	NA
11	Cliente	192.168.127.11	192.168.127.1	192.168.127.10	99480
12	Cliente	192.168.127.12	192.168.127.1	192.168.127.10	89507
13	Cliente	192.168.127.13	192.168.127.1	192.168.127.10	76590
14	Cliente	192.168.127.14	192.168.127.1	192.168.127.10	87486
15	Cliente	192.168.127.15	192.168.127.1	192.168.127.10	99509

**Descripción de resultado:** se aumentó el número de réplicas a un valor que representan un nivel de disponibilidad óptimo como se explicara más adelante en el análisis y los tiempos aumentaron significativamente.

Tabla 20. Tabla de resultados escenarios de pruebas número. 12

• **Escenario de prueba número 13**

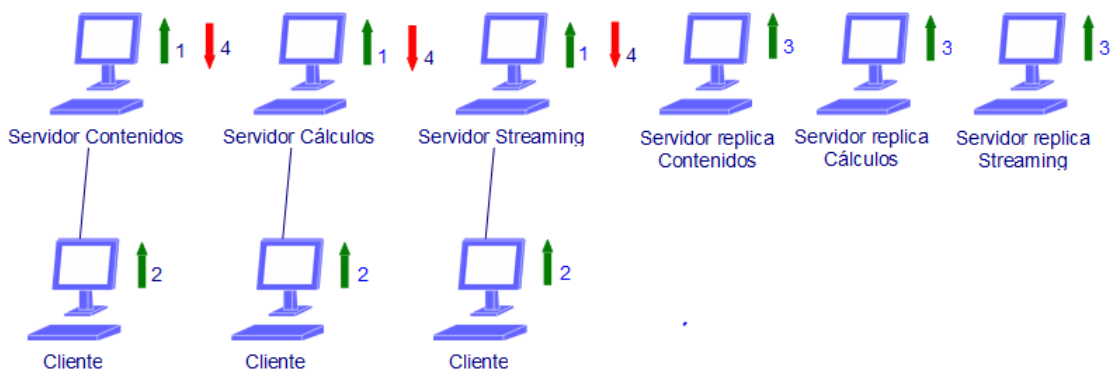


Figura 33. Representación del escenario de prueba número. 13.

**Escenario número 13:** se hace una mezcla de servidores con distintos servicios, se debe probar que se prestan, se replican y se recuperan los servicios.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA		NA
2	Replica	192.168.127.2	NA	NA	NA
3	Servidor	192.168.127.3	NA	NA	NA
4	Replica	192.168.127.4	NA	NA	NA
5	Servidor	192.168.127.5	NA	NA	NA
6	Replica	192.168.127.6	NA	NA	NA
7	Cliente	192.168.127.7	192.168.127.1	192.168.127.2	2098
8	Cliente	192.168.127.8	192.168.127.3	192.168.127.4	4073
9	Cliente	192.168.127.9	192.168.127.4	192.168.127.5	3099
10	Cliente	192.168.127.10	192.168.127.1	192.168.127.2	2006

**Descripción de resultado:** este escenario de prueba se llevó a cabo con el fin de demostrar que en la red superpuesta P2P se pueden prestar múltiples servicios con las características de disponibilidad propuestas en este trabajo, los tiempos varían de acuerdo al tipo de servicio.

Tabla 21. Tabla de resultados escenarios de pruebas número. 13



- **Escenario de prueba número 14**

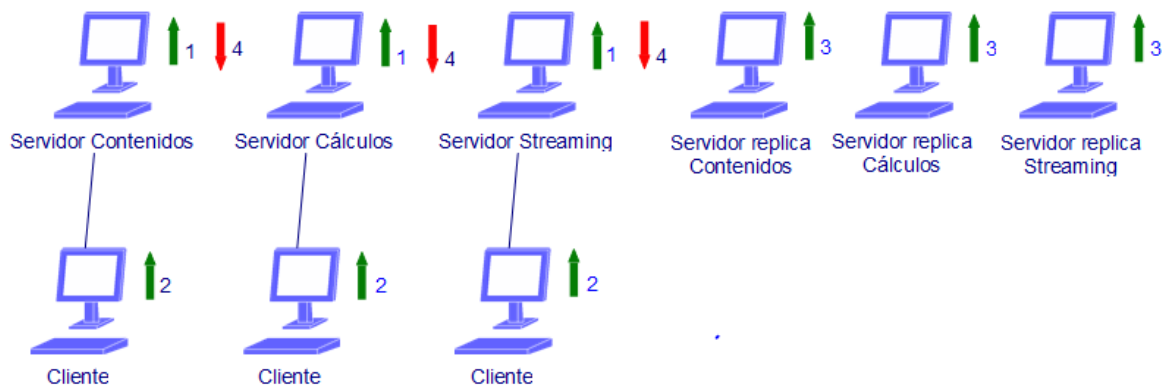


Figura 34. Representación del escenario de prueba número. 14.

**Escenario número 14:** se hace una mezcla de servidores con distintos servicios, se debe probar que se prestan, se replican y se recuperan los servicios.

Numero equipo	Rol	IP	Servidor consumo actual	Servidor consumo nuevo	Tiempo de recuperación milisegundos
1	Servidor	192.168.127.1	NA		NA
2	Servidor	192.168.127.2	NA	NA	NA
3	Servidor	192.168.127.3	NA	NA	NA
4	Replica	192.168.127.4	NA	NA	NA
5	Replica	192.168.127.5	NA	NA	NA
6	Replica	192.168.127.6	NA	NA	NA
7	Cliente	192.168.127.7	192.168.127.3	192.168.127.6	2598
8	Cliente	192.168.127.8	192.168.127.1	192.168.127.5	4673
9	Cliente	192.168.127.9	192.168.127.2	192.168.127.4	3665
10	Cliente	192.168.127.10	192.168.127.1	192.168.127.5	2876

**Descripción de resultado:** este escenario de prueba se llevó a cabo con el fin de demostrar que en la red superpuesta P2P se pueden prestar múltiples servicios con las características de disponibilidad propuestas en este trabajo, los tiempos varían de acuerdo al tipo de servicio.

Tabla 22. Tabla de resultados escenarios de pruebas número. 14

### 7.1.1.8. Análisis de resultados:

Para medir la disponibilidad de un servicio que ha sido publicado, vamos a visualizar y analizar una serie de variables como: tiempo de recuperación, número de réplicas, número de fallos y estabilidad, las cuales juegan un papel importante a la hora de validar los resultados.

Basados en [16], en la cual presentan una investigación y proponen una estrategia de réplicas que asegura la disponibilidad deseada con un mínimo de réplicas en el sistema a pesar que se presenten fallos. Por lo anterior, vamos a utilizar los métodos y las formulas allí propuestas con los datos recolectados en nuestra simulación.

- **Gráfica tiempo de recuperación vs número de réplicas**

La grafica se realiza con el fin de hacer una comparación entre el tiempo de recuperación del servicio de acuerdo al número de réplicas, para posteriormente mediante una aproximación lineal calcular cual sería el comportamiento de acuerdo a la escalabilidad en número de réplicas del sistema.

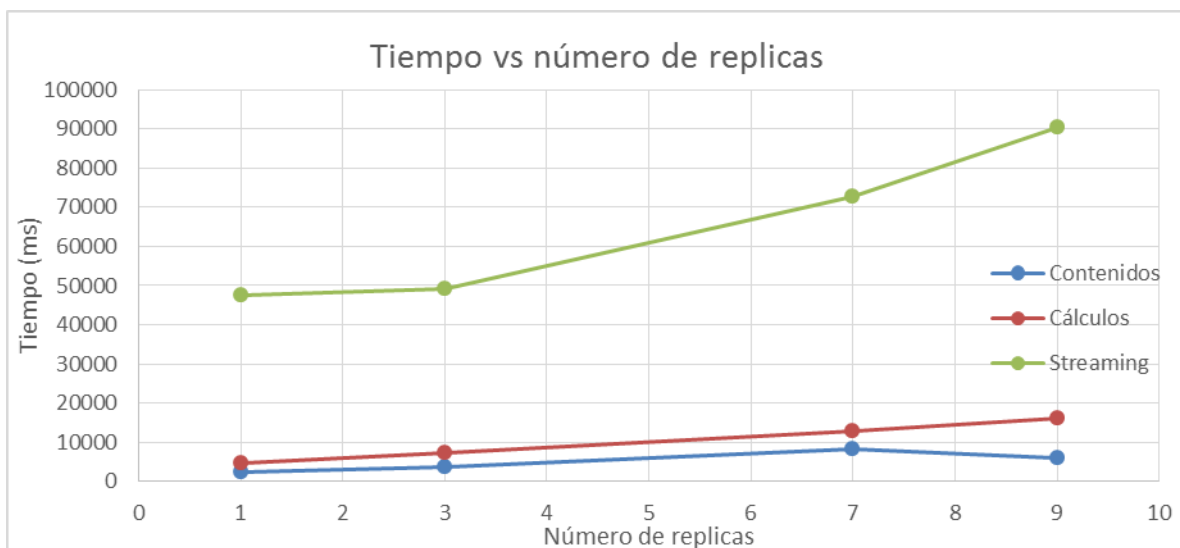


Figura 35: Gráfica de tiempo vs número de réplicas.

De acuerdo a la *Figura 35: Gráfica de tiempo vs número de réplicas*. Figura 35 se puede observar un comportamiento creciente del tiempo necesario para recuperar cada tipo de servicio de acuerdo al número de réplicas que se presentaron, se evidencia claramente una diferencia en el tiempo de acuerdo a cada tipo de servicio.

- **Gráfica tiempo vs número de fallos**

La grafica se realiza con el fin de hacer una comparación entre el tiempo de recuperación del servicio de acuerdo al número de fallos, para posteriormente mediante una aproximación lineal calcular cual sería el comportamiento de acuerdo a la escalabilidad en número de réplicas del sistema.

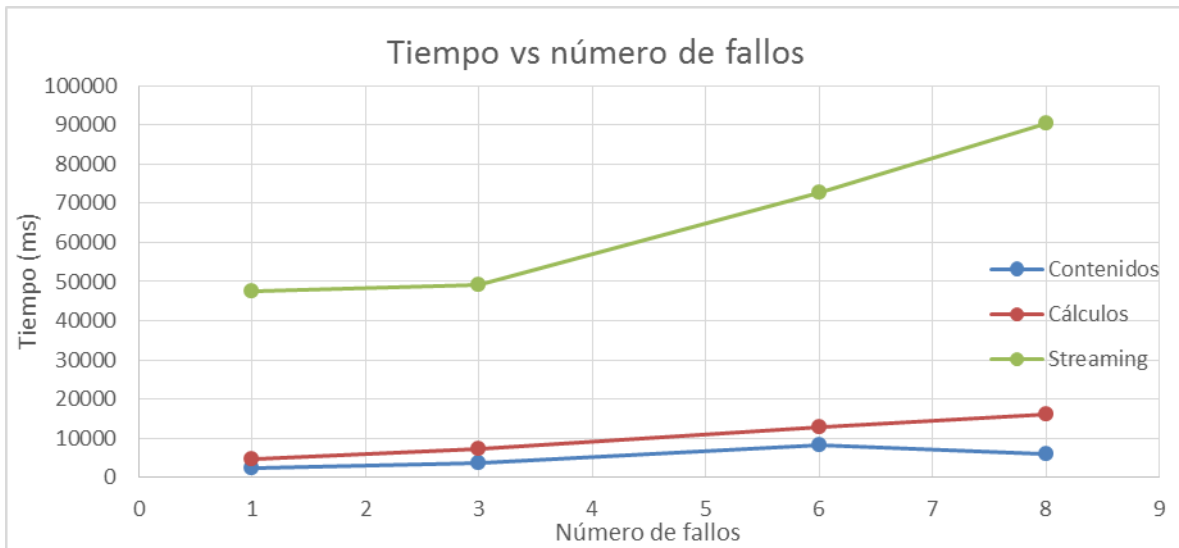


Figura 36: Gráfica de tiempo vs número de fallos.

De acuerdo a la Figura 36, se puede observar un comportamiento creciente del tiempo necesario para recuperar cada tipo de servicio de acuerdo al número de fallos que se presentaron, se evidencia claramente una diferencia en el tiempo de acuerdo a cada tipo de servicio.

- **Cálculo de la disponibilidad a partir de la estabilidad de los nodos**

Para calcular la disponibilidad de un servicio “Availj” en la red, utilizamos la fórmula propuesta por Meroufel y Belalem [16].

$$Avail_j = 1 - \prod_{i=1}^{\alpha} (1 - p_i)$$

En la cual:

**Availj** = disponibilidad (porcentaje valor entre 0 y 1).

**j** = servicio.

**α** = número de réplicas.

**Pi** = estabilidad del nodo donde se replica el servicio (porcentaje valor entre 0 y 1).

Se debe entender que tanto para la disponibilidad como para la estabilidad del nodo a medida que su valor se acerca más a uno, estas características mejoran en el sistema. Estabilidad es la probabilidad de que el nodo o los nodos en donde se prestan o se replican los servicios permanezcan disponibles en la red.

La relación entre las variables de disponibilidad, estabilidad y número de réplicas es directamente proporcional.

- **Resultados del cálculo de Disponibilidad con diferentes valores para la estabilidad.**

A continuación, se analizará la disponibilidad de un servicio a partir de la estabilidad de las réplicas, la información se encuentra en tablas, que además contienen un gráfico con el comportamiento.

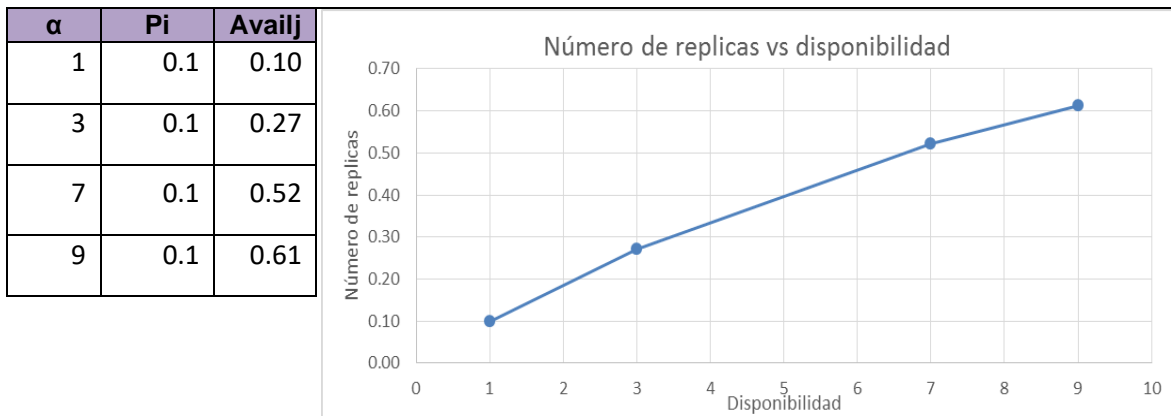


Tabla 23. Disponibilidad en la red para una estabilidad del diez por ciento (10%).

Para una estabilidad del diez por ciento (10%) la disponibilidad muestra un comportamiento creciente a medida que incrementa el número de réplicas.

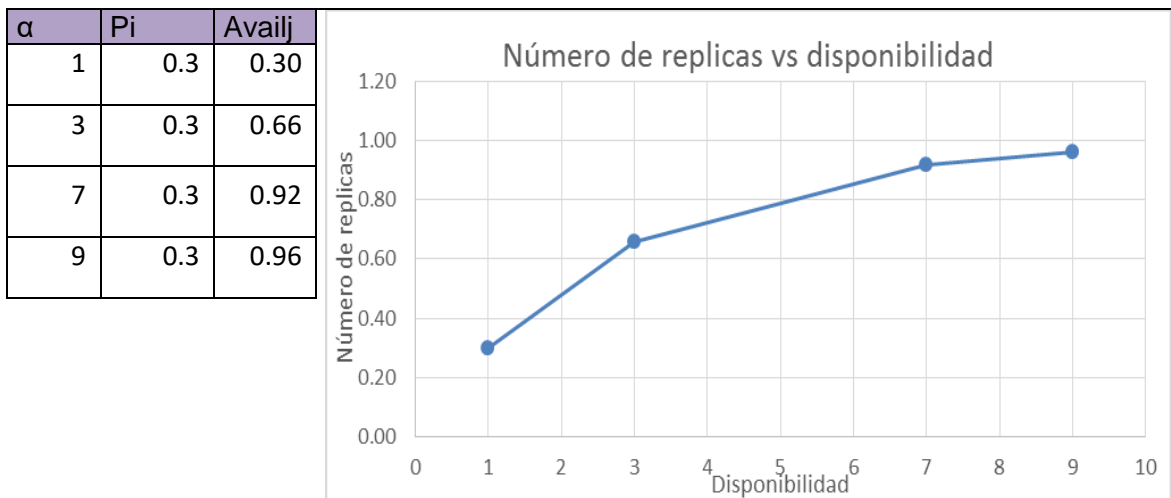


Tabla 24. Disponibilidad en la red para una estabilidad del treinta por ciento (30%).

Para una estabilidad del treinta por ciento (30%) la disponibilidad muestra un comportamiento creciente a medida que incrementa el número de réplicas, mejorando con respecto al anterior cálculo y creciendo más rápidamente.

$\alpha$	$P_i$	Availj
1	0.7	0.70
3	0.7	0.97
7	0.7	0.99
9	0.7	0.999

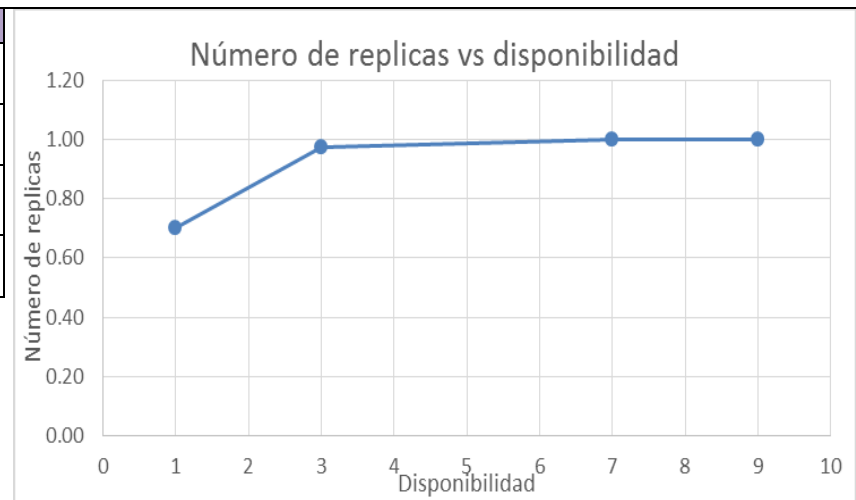


Tabla 25. Disponibilidad en la red para una estabilidad del setenta por ciento (70%).

Para una estabilidad del setenta por ciento (70%) la disponibilidad muestra un comportamiento creciente con valores muy cercanos a uno.

$\alpha$	$P_i$	Availj
1	0.9	0.90
3	0.9	0.99
7	0.9	0.999
9	0.9	0.9999

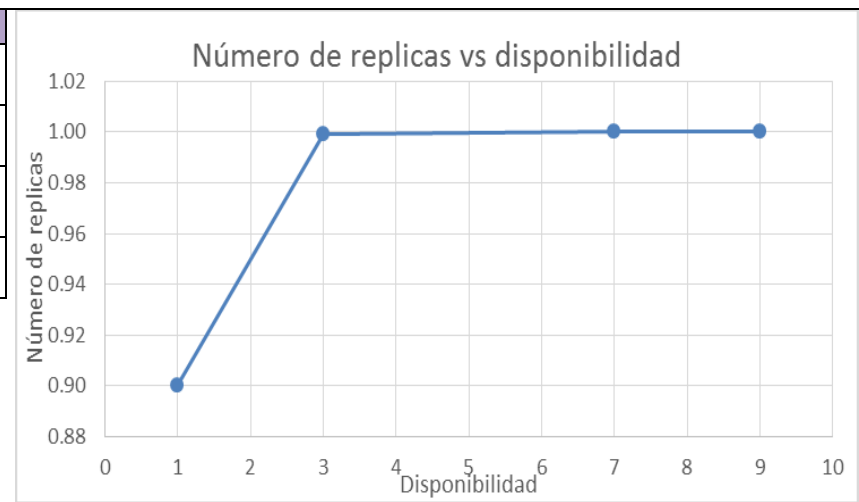


Tabla 26. Disponibilidad en la red para una estabilidad del noventa por ciento (90%).

En la Tabla 26 se observa que a medida que el número de réplicas aumenta y se incrementa la estabilidad de los nodos, la gráfica tiende a converger a valores que se acercan a uno, que es un valor óptimo para la red.

Para un número de réplicas mayor a nueve y con una estabilidad mayor del cincuenta por ciento (50%), el sistema tiende a tener un comportamiento casi lineal con valores cercanos a uno.

- **Aproximación para un gran número de réplicas y con un nivel medio de estabilidad de los nodos en la red.**

La aproximación se realiza con el fin de proyectar los resultados obtenidos hacia un entorno de ejecución real, en la cual millones de usuarios se unen a una red para compartir cualquier tipo de servicio.

$\alpha$	$P_i$	Avail <sub>j</sub>
1	0.5	0.5000000000
3	0.5	0.8750000000
7	0.5	0.9921875000
9	0.5	0.9980468750
100	0.5	0.9999900000
200	0.5	0.9999990000
500	0.5	0.9999999000

Tabla 27. Aproximación en el sistema para una estabilidad del cincuenta por ciento y 500 réplicas.

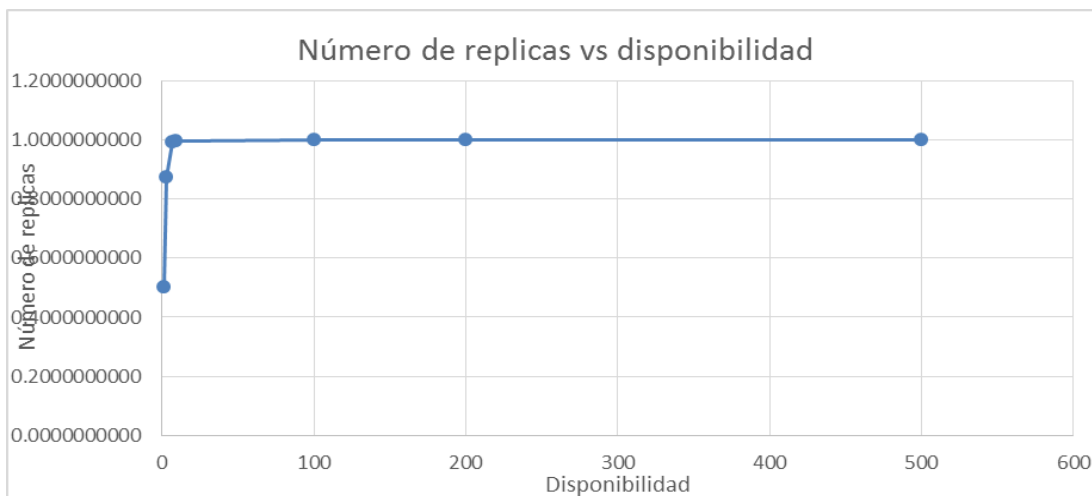


Figura 37. Aproximación para una estabilidad del cincuenta por ciento y 500 réplicas.

La Figura 37 muestra el comportamiento para un servicio en la red cuando se crean grandes cantidades de réplicas de este servicio.

En este capítulo con la realización de las pruebas y la validación realizada al prototipo software, se logró evidenciar que la técnica bajo la que fue construido el prototipo software cumple con el soporte de la disponibilidad en redes superpuestas P2P no estructuradas y brinda las características para el soporte de múltiples servicios en la red.

## Capítulo 8. CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se presentan las conclusiones obtenidas durante el desarrollo del trabajo de grado y algunas recomendaciones para realizar trabajos a futuro.

### 8.1. Conclusiones

- De acuerdo con la investigación realizada, se pudo evidenciar que aunque existen diferentes técnicas para mejorar la disponibilidad de servicios sobre redes superpuestas P2P, todas están enfocadas en la prestación de un determinado servicio, lo cual conlleva a la falta de versatilidad entre las diferentes redes superpuestas desarrolladas de hoy en día. En este trabajo se hizo un aporte para la solución de este problema, con la gestión de tres tipos de servicios.
- De acuerdo al estudio realizado sobre la técnica de disponibilidad seleccionada, la cual fue replicación, se logró demostrar que está es la que más producción investigativa y mejores resultados tiene; por lo tanto, esta es la mejor técnica con la cual se podrían realizar más investigaciones para su mejoramiento con el fin de alcanzar un buen nivel de disponibilidad.
- El estudio realizado sobre las dos investigaciones en la cuales se basó este trabajo, técnica de búsqueda y técnica de recuperación, fueron de gran ayuda para contextualizarnos con el problema y para el planteamiento de la solución propuesta.
- Los beneficios aportados por la plataforma JXTA, facilitó el cumplimiento de los requisitos del trabajo de grado, ya que permite de forma intuitiva el desarrollo de este tipo de redes, como son las redes superpuestas P2P.
- La integración de las tecnologías seleccionadas en este trabajo (VLC para el servicio de media streaming, FTP para el servicio de contenidos y Sockets e Hilos para el servicio de cálculos computacionales) con la plataforma JXTA, permitió comprobar las características de la interoperabilidad y la independencia de plataforma que esta posee.
- En este trabajo se realizó una investigación acerca de las redes superpuestas P2P, mediante la cual se logró evidenciar la inestabilidad inherente de los nodos en este tipo de redes. Además con la propuesta presentada en este trabajo de grado se contribuyó a disminuir la inestabilidad de estos nodos en la prestación de servicios.
- Se desarrolló un prototipo software con el cual se logró evaluar la técnica de disponibilidad de servicios propuesta en este trabajo de grado; lo que nos permitió gestionar tres tipos de servicios: contenidos, cálculos computacionales y media streaming.

## 8.2. Trabajos futuros y recomendaciones

- Se propone realizar una clasificación más amplia de los tipos de servicios soportados por las redes superpuestas P2P. Puesto que en el desarrollo de este trabajo de grado se implementó un módulo llamado “Perfiles”, en el cual se especifican las características o parámetros necesarios para prestar y consumir distintos tipos de servicios, se podrían agregar más perfiles para poder gestionar muchos más servicios.
- El estudio realizado sobre las técnicas utilizadas para mejorar la disponibilidad de servicios en redes superpuestas P2P, evidencio que la mejor técnica para proveer disponibilidad es la replicación, la cual fue implementada en este trabajo de grado, explorando solo una de sus características (replicación completa). Por lo tanto, se propone implementar esta técnica, explorando las demás características que esta posee, como por ejemplo: replicación a trozos y la consistencia entre replicas; con lo cual se podrá obtener un nivel más alto de disponibilidad.
- Se propone la implementación de una técnica de búsqueda más avanzada, en la cual se realicen búsquedas semánticas, con lo que el sistema pueda llevar a cabo un procesamiento más razonable de los servicios disponibles en la red superpuesta P2P, y así mejorar el descubrimiento de dichos servicios.
- Implementar la filosofía de los grupos de nodos de JXTA, para que cada usuario que pertenece a cada grupo preste y consuma diferentes servicios dentro de este; puesto que en este trabajo de grado solo se trabajó sobre el grupo global que se crea al publicar servicios.



## REFERENCIAS BIBLIOGRÁFICAS

- [1] Lloret, J. (2006). Arquitectura de interconexión de redes P2P parcialmente centralizadas.
- [2] Herrera O. & Znati, T. (2012). Proactive Repairs and Incentives for Content Availability in P2P Overlay Networks. *International Journal of Computer Science & Network Security*, 12(4).
- [3] Kwok, Y. K. R. (2011). *Peer-to-Peer Computing: Applications, Architecture, Protocols, and Challenges*. CRC Press.
- [4] 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).
- [5] 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.
- [6] Merino, L. R., Fernández, A., López, L., & Cholvi, V. Topologías dinámicas en redes P2P no estructuradas.
- [7] 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.
- [8] Argan, M. T., Argan, M., Ozer, A., & Kose, H. (2013). A Study of Motivational Factors Associated With Peer-to-Peer (P2P) File-Sharing. *Procedia-Social and Behavioral Sciences*, 99, 180-188.
- [9] Ramzan, N., Park, H., & Izquierdo, E. (2012). Video streaming over P2P networks: Challenges and opportunities. *Signal Processing: Image Communication*, 27(5), 401-411.
- [10] Wu, T. Y., Lee, W. T., Guizani, N., & Wang, T. M. (2014). Incentive mechanism for P2P file sharing based on social network and game theory. *Journal of Network and Computer Applications*, 41, 47-55.
- [11] 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.
- [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] 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.
- [14] 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>
- [15] Schollmeier, R. (2001, August). A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications. In *Peer-to-Peer Computing*, IEEE International Conference on (pp. 0101-0101). IEEE Computer Society.
- [16] Meroufel, B., & Belalem, G. (2013). Managing Data Replication and Placement based on Availability. *AASRI Procedia*, 5, 147-155.
- [17] Spaho, E., Barolli, L., Xhafa, F., Biberaj, A., & Shurdi, O. (2013). P2P data replication and trustworthiness for a JXTA-Overlay P2P system using fuzzy logic. *Applied Soft Computing*, 13(1), 321-328.
- [18] Dimakopoulos, V., Margariti, S., Ntetsika, M., & Pitoura, E. Data Replication in P2P Systems.
- [19] Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., & Lim, S. (2005). A survey and comparison of Peer-to-Peer overlay network schemes. *IEEE Communications Surveys and Tutorials*, 7(1-4), 72-93.
- [20] Muñoz, G. R., Quinchua, L. (2009). Extensión de una plataforma que soporte un sistema para la provisión de servicios en un entorno de redes superpuestas P2P.
- [21] Antoniadis, P., Courcoubetis, C., & Strulo, B. (2005). Incentives for content availability in memory-less peer-to-peer file sharing systems. *ACM SIGecom Exchanges*, 5(4), 11-20.
- [22] 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.
- [23] “Metodología XP” [Online]: Available: [http://fi.ort.edu.uy/innovaportal/file/2021/1/metodologia\\_xp.pdf](http://fi.ort.edu.uy/innovaportal/file/2021/1/metodologia_xp.pdf).

- [24] Sachs, M., & Piccard, P. (2005). Securing IM and P2P Applications for the Enterprise. Syngress, Chapter 9 - Gnutella Architecture, Pages 239–266.
- [25] Sachs, M., & Piccard, P. (2005). Securing IM and P2P Applications for the Enterprise. Syngress, Chapter 11 – BitTorrent, Pages 285-317.
- [26] Cerón M. A., Magé P. A., (2010, sept), Pixtream : Sistema de media Streaming basado en redes P2P, IX Ing. Telemática (JITEL 2010), Valladolid, España.
- [27] Sachs, M., & Piccard, P. (2005). Securing IM and P2P Applications for the Enterprise. Syngress, Capitulo 10 - eDonkey and eMule, Pages 267–283.
- [28] A. Banerjee, (December 2008), Measuring and Modeling Applications for Content Distribution in the Internet.
- [29] Saroiu, S., Gummadi, K. P., & Gribble, S. D. (2003). Measuring and analyzing the characteristics of Napster and Gnutella hosts. *Multimedia systems*, 9(2), 170-184.
- [30] Mauri, J. L. (2006). Arquitectura de interconexión de redes p2p parcialmente centralizadas (Doctoral dissertation, Universitat Politècnica de València).
- [31] Vu, Q. H., Lupu, M., & Ooi, B. C. (2010). Architecture of peer-to-peer systems. In *Peer-to-Peer Computing* (pp. 11-37). Springer Berlin Heidelberg.
- [32] Mariano Disanzo. (2006). Redes Peer to Peer y Tecnología JXTA.
- [33] Brookshier, D., Govoni, D., Krishnan, N., Soto, J. C., (2002, May). JXTA: Java™ P2P Programming.
- [34] Li Gong - Sun Microsystems Inc. (2001, Jun). JXTA: A Network Programming Environment.
- [35] Muñoz Andrade, A., Muñoz Luna, D. E., (2010, Nov), Técnica de Búsqueda para la Prestación de Servicios sobre Redes Superpuestas P2P No Estructuradas, Universidad del Cauca.
- [36] RAE. (2014), Real Academia Española. Diccionario de la Lengua Española. Disponible: <http://www.rae.es/>.
- [37] Traversat, B., Arora, A., Abdelaziz, M., Duigou, M., Haywood, C., Hugly, J. C., & Yeager, B. (2003). Project JXTA 2.0 super-peer virtual network.
- [38] Oaks, S., Traversat, B., & Gong, L. (2002). JXTA in a Nutshell. " O'Reilly Media, Inc."

- [39] 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.
- [40] Simón, J. L., Disanzo, M., (2006), Redes Peer to Peer y Tecnología JXTA, Escuela de Ingeniería Electrónica.
- [41] Liao, X., Jin, H., & Yu, L. (2012). A novel data replication mechanism in P2P VoD system. *Future Generation Computer Systems*,28(6), 930-939.
- [42] 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.
- [43] 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.
- [44] "Caprica." [Online]. Available: <http://caprica.github.io/vlcj/javadoc/2.1.0/> [Accessed: 20-Mar-2015].
- [45] "Informatica" [Online]. Available: <http://www.informatica-hoy.com.ar/aprender-informatica/Que-es-FTP-como-funciona.php> [Accessed: 20-Mar-2015].
- [46] Anderson, D. P., Cobb, J., Korpela, E., Lebofsky, M., & Werthimer, D. (2002). SETI@ home: an experiment in public-resource computing. *Communications of the ACM*, 45(11), 56-61.
- [47] "Oracle" [Online]. Available: <https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html> [Accessed: 09-Junio-2015].
- [48] "Oracle" [Online]. Available: <https://docs.oracle.com/javase/tutorial/essential/concurrency/> [Accessed: 09-Junio-2015].
- [49] "Is Design Dead" [Online]. Available: <http://www.martinfowler.com/articles/designDead.html>.
- [50] "Campusvirtual.univalle.edu.co" [Online]. Available: [https://campusvirtual.univalle.edu.co/moodle/pluginfile.php/77045/mod\\_resource/content/0/Lecturas/xp.pdf](https://campusvirtual.univalle.edu.co/moodle/pluginfile.php/77045/mod_resource/content/0/Lecturas/xp.pdf).
- [51] "ITE" [Online]. Available: <http://www.ite.educacion.es/formacion/materiales/107/cd/video/video0201.html>

- [52] “Ordenadores” [Oline]. Available: <http://www.ordenadores-y-portatiles.com/ftp.html>
- [53] “DLSI” [Online]. Available: <http://www.dlsi.ua.es/asignaturas/sid/J.Sockets.pdf>
- [54] “INFO” [Online]. Available: <http://www.infor.uva.es/~fdiaz/sd/doc/hilos>