

# **Técnica de Recuperación y Ejecución de Servicios en Redes Superpuestas Peer to Peer No Estructuradas**

**Gerardo Iván Ordoñez Zambrano  
Gonzalo Ernesto Potosí Gutiérrez**

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones  
Programa en Ingeniería de Sistemas  
Departamento de Sistemas  
Grupo IDIS - Investigación y Desarrollo en Ingeniería de Software  
Línea de Investigación en Tecnología de Soporte  
Popayán, febrero de 2015**

# **Técnica de Recuperación y Ejecución de Servicios en Redes Superpuestas Peer to Peer No Estructuradas**



**Gerardo Iván Ordoñez Zambrano**  
**Gonzalo Ernesto Potosí Gutiérrez**

Monografía de Trabajo de Grado

Director  
Ing. Pablo Augusto Magé Imbachí

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones**  
**Programa en Ingeniería de Sistemas**  
**Departamento de Sistemas**  
**Grupo IDIS - Investigación y Desarrollo en Ingeniería de Software**  
**Línea de Investigación en Tecnología de Soporte**  
Popayán, febrero de 2015

Técnica de Recuperación y Ejecución de Servicios en  
Redes Superpuestas Peer to Peer No Estructuradas

---

## TABLA DE CONTENIDO

|   |           |
|---|-----------|
| <b>INTRODUCCIÓN .....</b>   | <b>9</b>  |
| <b>CAPÍTULO I. MARCO CONCEPTUAL.....</b>  | <b>12</b> |
| 1.1. Redes Superpuestas.....  | 12        |
| 1.2. Redes Cliente-Servidor .....   | 12        |
| 1.3. Redes Peer-to-Peer .....   | 12        |
| 1.3.1. Definición de Red Superpuesta Peer-to-Peer .....   | 13        |
| 1.3.1.1. Redes Peer-to-Peer Centralizadas .....   | 14        |
| 1.3.1.2. Redes Peer-to-Peer Distribuidas o Puras .....  | 15        |
| 1.3.1.3. Redes Peer-to-Peer Híbridas .....  | 15        |
| 1.3.2. Redes Peer-to-Peer Estructuradas y No Estructuradas.....   | 16        |
| 1.4. Técnica .....  | 17        |
| 1.5. Recuperación .....   | 17        |
| 1.6. Servicios .....  | 17        |
| 1.6.1. Servicio .....   | 17        |
| 1.6.2. Clases de servicios .....  | 18        |
| 1.7. Sistemas de media streaming .....  | 19        |
| 1.7.1. Definición de media streaming .....  | 19        |
| 1.7.1.1 Cuello de botella en el ancho de banda .....  | 19        |
| <b>CAPÍTULO II. BÚSQUEDA Y RECUPERACIÓN DE SERVICIOS.....</b>   | <b>21</b> |
| 2.1. Descubrimiento de servicios .....  | 21        |
| 2.2. Búsqueda de servicios en Redes Superpuestas Peer-to-Peer No Estructuradas.....   | 21        |
| 2.2.1. Métodos de búsqueda de servicios.....  | 22        |
| 2.3. Características y selección de los métodos de búsqueda.....  | 24        |
| 2.4. Recuperación de servicios .....  | 26        |
| 2.5. Recuperación de servicios en Redes Superpuestas Peer-to-Peer No Estructuradas.....   | 27        |
| 2.7. Priorización de las características del sistema Peer-to-Peer y de la técnica de recuperación y ejecución de servicios..... | 28        |
| 2.8. Selección del perfil del servicio .....  | 30        |

|   |           |
|---|-----------|
| <b>CAPÍTULO III. TÉCNICA DE RECUPERACIÓN Y EJECUCIÓN DE SERVICIOS EN REDES PEER-TO-PEER .....</b>   | <b>31</b> |
| 3.1. Introducción.....  | 31        |
| 3.2. Arquitectura del Sistema .....   | 31        |
| 3.2.1. Sistema Peer-to-Peer .....   | 32        |
| 3.2.2. Representante .....  | 33        |
| 3.2.3. Consumidor .....   | 33        |
| 3.2.4. Observador .....   | 33        |
| 3.3. Descripción de la Técnica de Recuperación y Ejecución de Servicios .....   | 33        |
| 3.3.1. Acciones de los nodos .....  | 33        |
| 3.3.1.1. Acciones del nodo Servidor .....   | 33        |
| 3.3.1.2. Acciones del nodo Cliente .....  | 34        |
| 3.3.2. Comunicación entre nodos (Clientes y Servidores) .....   | 35        |
| 3.3.3. Servicio a Desplegar.....  | 35        |
| 3.3.4. Conexión.....  | 36        |
| 3.3.5. Mensajes .....   | 37        |
| 3.3.6. Envío de Mensajes .....  | 37        |
| 3.3.7. Mensajes en el Servicio a Desplegar.....   | 37        |
| 3.3.8. Búsqueda y Recuperación de Servicio .....  | 38        |
| 3.3.9. Observador .....   | 38        |
| 3.4. Prestación de Servicio .....   | 38        |
| 3.4.1. Publicación del Servicio .....   | 39        |
| 3.4.2. Búsqueda y Consumo del Servicio .....  | 44        |
| 3.4.3. Recuperación del Servicio .....  | 48        |
| <br>  |           |
| <b>CAPÍTULO IV. TECNOLOGÍAS PARA LA IMPLEMENTACIÓN DE REDES SUPERPUESTAS PEER-TO-PEER Y EL DESPLIEGUE DEL SERVICIO DE MEDIA STREAMING .....</b> | <b>50</b> |
| 4.1. JXTA .....   | 50        |
| 4.1.1. ¿Por qué JXTA como tecnología utilizada? .....   | 50        |
| 4.1.2. Arquitectura JXTA.....   | 51        |
| 4.1.3. Protocolos.....  | 52        |
| 4.1.4. Servicios JXTA.....  | 52        |
| 4.2. VLC .....  | 53        |

|   |           |
|---|-----------|
| 4.2.1. ¿Por qué VLC como sistema de media streaming?.....                                 | 53        |
| 4.2.2. libVLC .....   | 54        |
| 4.3. VLCJ .....   | 54        |
| <b>CAPÍTULO V. IMPLEMENTACIÓN DE PROTOTIPO SOFTWARE.....</b>                              | <b>55</b> |
| 5.1. Especificación de Requerimientos .....   | 55        |
| 5.1.1. Alcance .....  | 55        |
| 5.1.2. Funciones del Producto .....   | 55        |
| 5.1.3. Características del Usuario .....  | 56        |
| 5.1.4. Priorizar los Requisitos .....   | 56        |
| 5.1.5. Requerimientos Funcionales .....   | 56        |
| 5.1.6. Requerimientos Funcionales: .....  | 57        |
| 5.2. Implementación.....  | 58        |
| 5.2.2. Herramientas .....   | 59        |
| 5.2.2.1. Frameworks .....   | 59        |
| 5.2.2.2. Plataforma de Desarrollo .....   | 59        |
| 5.2.2.3. Entorno de Desarrollo Integrado .....  | 60        |
| 5.2.3. Pruebas .....  | 60        |
| <b>CAPÍTULO VI. EXPERIMENTOS Y RESULTADOS .....</b>                                       | <b>61</b> |
| 6.1. Definición de escenarios de prueba .....   | 61        |
| 6.1.1 Descripción del marco experimental .....  | 65        |
| 6.2. Desarrollo de escenarios de prueba .....   | 65        |
| 6.3. Pruebas en características de desempeño, escalabilidad y precisión de búsqueda ..... | 80        |
| <b>CAPÍTULO VII. CONCLUSIONES Y TRABAJOS FUTUROS.....</b>                                 | <b>82</b> |
| 7.1. Conclusiones.....  | 82        |
| 7.2. Trabajos futuros y recomendaciones .....   | 83        |
| <b>REFERENCIAS BIBLIOGRÁFICAS .....</b>   | <b>85</b> |

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1. Sistema peer-to-peer centralizado. ....                        | 14 |
| Figura 2. Sistema peer-to-peer puro o distribuido. ....                  | 15 |
| Figura 3. Sistema peer-to-peer hibrido. ....                             | 16 |
| Figura 4. Caracterización de la técnica de recuperación y ejecución .... | 30 |
| Figura 5. Arquitectura del sistema.....                                  | 31 |
| Figura 6. Sistema peer-to-peer.....                                      | 32 |
| Figura 7. Descripción modular del nodo servidor .....                    | 36 |
| Figura 8. Conexión aleatoria de nodos.....                               | 36 |
| Figura 9. Conexión directa entre nodos .....                             | 37 |
| Figura 10. Diagrama de flujo publicación del servicio .....              | 40 |
| Figura 11. Diagrama de secuencia crear nodo servidor.....                | 41 |
| Figura 12. Diagrama de secuencia crear nodo cliente .....                | 42 |
| Figura 13. Diagrama de secuencia crear grupo nodo servidor.....          | 42 |
| Figura 14. Diagrama de secuencia crear grupo nodo cliente .....          | 43 |
| Figura 15. Diagrama de secuencia publicar servicio .....                 | 44 |
| Figura 16. Diagrama de flujo búsqueda servicio .....                     | 45 |
| Figura 17. Diagrama secuencial búsqueda servicio .....                   | 47 |
| Figura 18. Diagrama de secuencia consumir servicio .....                 | 47 |
| Figura 19. Diagrama de secuencia recuperar servicio .....                | 48 |
| Figura 20. Diagrama de flujo recuperar servicio.....                     | 49 |
| Figura 21. Arquitectura plataforma JXTA .....                            | 51 |
| Figura 22. Representación del escenario de prueba No. 1 .....            | 66 |
| Figura 23. Representación del escenario de prueba No. 2 .....            | 67 |
| Figura 24. Representación del escenario de prueba No. 3 .....            | 68 |
| Figura 25. Representación del escenario de prueba No. 4 .....            | 69 |
| Figura 26. Representación del escenario de prueba No. 5 .....            | 70 |
| Figura 27. Representación del escenario de prueba No. 6 .....            | 71 |
| Figura 28. Representación del escenario de prueba No. 7 .....            | 72 |
| Figura 29. Representación del escenario de prueba No. 8 .....            | 73 |
| Figura 30. Representación del escenario de prueba No. 9 .....            | 74 |
| Figura 31. Representación del escenario de prueba No. 10 .....           | 75 |
| Figura 32. Representación del escenario de prueba No. 11 .....           | 76 |
| Figura 33. Representación del escenario de prueba No. 12 .....           | 77 |
| Figura 34. Representación del escenario de prueba No. 13 .....           | 78 |
| Figura 35. Representación del escenario de prueba No. 14 .....           | 79 |
| Figura 36. Gráfica recuperación y ejecución del servicio .....           | 80 |
| Figura 37. Gráfica precisión de búsqueda.....                            | 81 |
| Figura 38. Gráfica recuperación de servicio .....                        | 81 |

## LISTA DE TABLAS

|  |    |
|--|----|
| Tabla 1. Descripción de métodos de búsqueda .....              | 23 |
| Tabla 2. Métodos de búsqueda seleccionados .....               | 26 |
| Tabla 3. Descripción de escenarios de prueba .....             | 64 |
| Tabla 4. Formato de registro de resultados.....                | 64 |
| Tabla 5. Tabla de resultados escenario de prueba Nro. 1.....   | 66 |
| Tabla 6. Tabla de resultados escenario de prueba Nro. 2.....   | 67 |
| Tabla 7. Tabla de resultados escenario de prueba Nro. 3.....   | 68 |
| Tabla 8. Tabla de resultados escenario de prueba Nro. 4.....   | 69 |
| Tabla 9. Tabla de resultados escenario de prueba Nro. 5.....   | 70 |
| Tabla 10. Tabla de resultados escenario de prueba Nro. 6.....  | 71 |
| Tabla 11. Tabla de resultados escenario de prueba Nro. 7.....  | 72 |
| Tabla 12. Tabla de resultados escenario de prueba Nro. 8.....  | 73 |
| Tabla 13. Tabla de resultados escenario de prueba Nro. 9.....  | 74 |
| Tabla 14. Tabla de resultados escenario de prueba Nro. 10..... | 75 |
| Tabla 15. Tabla de resultados escenario de prueba Nro. 11..... | 76 |
| Tabla 16. Tabla de resultados escenario de prueba Nro. 12..... | 77 |
| Tabla 17. Tabla de resultados escenario de prueba Nro. 13..... | 78 |
| Tabla 18. Tabla de resultados escenario de prueba Nro. 14..... | 79 |



## INTRODUCCIÓN

Hoy en día debido al cambio favorable en las condiciones en los bordes de Internet, tales como el aumento del número de equipos de cómputo seguido de un notorio avance en cuanto a su potencia de hardware y un veloz crecimiento del ancho de banda disponible, las redes peer-to-peer cobran cada vez más importancia e interés motivando a que los recursos desplegados en la Internet sean explotados. Lo anterior facilita que las máquinas clientes sean capaces de prestar y consumir servicios o contenidos de manera directa sin depender de entidades centrales. Las redes peer-to-peer nacen como una alternativa al popular modelo Cliente-Servidor [1] y pueden ser de tipo estructuradas o no estructuradas [2]. El presente trabajo se enfocará en las redes peer-to-peer no estructuradas y el problema a resolver se considerará solo en este tipo de redes.

Es necesario destacar por otra parte el crecimiento e importancia que han cobrado los servicios de comunicación, puesto que han sido pensados en el usuario final y presentan la gran ventaja de ser usados en cualquier lugar y momento. La computación distribuida ha sido un ejemplo de infraestructura de soporte para comunicación y prestación de servicios [3]. Las redes peer-to-peer han demostrado ser una infraestructura apropiada para la prestación de servicios como por ejemplo para distribución o intercambio de contenidos, mensajería instantánea, voz y video sobre IP, juegos y ambientes colaborativos [4].

Los servicios de comunicación seguirán en auge dada su gran utilidad y disponibilidad por medio de los diversos dispositivos que conocemos actualmente. Además de su avance cuenta con la aparición de nuevos enfoques como agentes de software, redes neuronales artificiales y programación genética [3].

Sin embargo las redes peer-to-peer también cuentan con algunos problemas, como por ejemplo la interoperabilidad entre servicios [5][6][7][8], debido a la gran variedad de servicios desplegados en distintos tipos de sistema peer-to-peer, cada uno implementa sus propias interfaces o serie de protocolos de red, lo cual dificulta que dos o más plataformas puedan interactuar entre sí. Por otro parte, los sistemas peer-to-peer suelen formar sus redes sobre la información de la capa de aplicación sin tener en cuenta la infraestructura de red y políticas de la capa subyacente. Como resultado de esto, nodos que son locales en una red LAN no se enlazan de manera directa, utilizando enlaces de otra u otras LAN, lo que conduce a un uso ineficiente de los recursos de red.[9][10][11][12][13][14]

Cabe resaltar que las redes peer-to-peer puras no disponen de un nodo central que gobierne a los demás como en la arquitectura Cliente-Servidor, por lo tanto la búsqueda determinista de recursos dada la carencia de una entidad central, presenta un gran desafío.[14][15][16].

Por último, uno de los mayores problemas de las redes peer-to-peer, es la disponibilidad en la prestación de servicios debido a que, por la naturaleza de este tipo de redes, un nodo puede salir de la red en cualquier momento de manera voluntaria o involuntaria, impidiendo así la prestación del servicio a otros nodos de la red. [17][18][19][20][21]

Dada la necesidad de establecer una forma de prestar servicios donde se aprovechen los dispositivos ubicados en la red, y puedan relacionarse entre sí, buscando, localizando y recuperando contenidos y servicios que son ofrecidos en las redes superpuestas peer-to-peer no estructuradas.

Por lo tanto este proyecto pretende contribuir en el problema de la disponibilidad de prestación de servicios en las redes superpuestas peer-to-peer no estructuradas. La primera fase incluye la información relevante sobre la contextualización de las redes superpuestas peer-to-peer, y que a su vez permitió tomar la decisión de cómo recuperar un servicio sobre éstas redes superpuestas peer-to-peer no estructuradas. Se incluyeron los sistemas Multiplataforma aprovechando los beneficios de los protocolos y las comunicaciones de las redes superpuestas peer-to-peer. De este modo y con una agrupación de sistemas adecuada, se pretende construir una técnica de recuperación y ejecución de un servicio dentro de las redes superpuestas peer-to-peer no estructuradas.

A continuación se describe la estructura que compone este documento:

**Capítulo I. Marco conceptual:** En este capítulo se encuentran las definiciones relacionadas con el contexto del problema y la investigación realizada.

**Capítulo II. Búsqueda y recuperación de servicios:** En este capítulo se describen las diferentes técnicas, algoritmos y características de búsqueda de servicios sobre redes superpuestas peer-to-peer como parte esencial del proceso de recuperación de servicios, además de describir las diferentes características de recuperación de servicios en redes peer-to-peer.

**Capítulo III. Técnica de recuperación y ejecución de servicios en redes peer-to-peer:** En este capítulo se presenta la construcción de la técnica, la cual está basada en redes superpuestas peer-to-peer.

**Capítulo IV. Tecnologías para la implementación de redes superpuestas peer-to-peer y el despliegue del servicio de media streaming:** En este capítulo se describen las tecnologías escogidas para la implementación de la red superpuesta peer-to-peer y la plataforma de despliegue del servicio escogido.

**Capítulo V. Implementación de prototipo software:** En este capítulo se presenta el proceso de desarrollo del prototipo software para la verificación de la técnica propuesta en el Capítulo III.

**Capítulo VI. Experimentos y resultados:** En este capítulo se describen los experimentos, casos de prueba y los resultados obtenidos a través del prototipo software.

**Capítulo VII. Conclusiones y trabajos futuros:** En este capítulo se presentan las conclusiones finales, recomendaciones y trabajos futuros.

## **CAPÍTULO I. MARCO CONCEPTUAL**

A continuación se describen los conceptos necesarios involucrados en la recuperación y ejecución de servicios en redes superpuestas peer-to-peer no estructuradas, para lograr una mejor comprensión del problema. A continuación se presentan estos conceptos.

### **1.1. Redes Superpuestas**

Una red Superpuesta es una red virtual construida en la parte superior de otra red. Los nodos que se encuentran en la red superpuesta forman enlaces lógicos que corresponden a un camino, tal vez a través de muchos enlaces físicos, en la red subyacente. Este tipo de redes surge dada la necesidad de ofrecer servicios que no se encuentran disponibles en la red subyacente. Internet es un ejemplo de red superpuesta ya que se encuentra construida como una superposición a la red telefónica [4].

### **1.2. Redes Cliente-Servidor**

Una red cliente-servidor es una red distribuida superpuesta construida sobre Internet, la cual consta de un sistema de alto desempeño, el Servidor, y varios sistemas de bajo desempeño, los Clientes. El servidor es la unidad de registro, así como el único proveedor de contenidos y servicios. Un cliente únicamente hace peticiones de contenidos o ejecución de servicios, sin compartir alguno de sus propios recursos [22].

### **1.3. Redes Peer-to-Peer**

Una red peer-to-peer también conocida como red de pares, red entre iguales, red entre pares, red punto a punto o simplemente P2P, es una red de computadores cuya arquitectura de red distribuida normalmente es implementada como una red superpuesta construida en la capa de aplicación de Internet [4].

Se dice que es una red entre iguales ya que los nodos participantes pueden actuar como clientes y servidores, el intercambio de información es directo y sin un control centralizado. Las aplicaciones peer-to-peer explotan los recursos hardware tales como poder de procesamiento CPU, espacio en disco, ancho de banda, impresoras, etc. Estos recursos compartidos por los usuarios en los bordes de Internet son necesarios para proporcionar el servicio y contenido ofrecido por la red (p. ej. distribución de archivos o espacios de trabajo colaborativos compartidos). Los nodos participantes de una red peer-to-peer son, pues, los proveedores de recursos (servicio y contenido), así como los solicitantes de recursos (servicio y contenidos) [22].

Debido a su naturaleza, los sistemas peer-to-peer en general poseen una serie de características inherentes a su condición descentralizada, y que son las que los hacen realmente atractivos: [23]

- **Escalabilidad:** En general, lo deseable es que cuantos más nodos estén conectados a una red peer-to-peer mejor será su funcionamiento. Así, cuando más nodos se unen a la red y comparten sus propios recursos, los recursos totales del sistema aumentarán. Esta característica difiere bastante a la arquitectura Cliente-Servidor con un sistema fijo de servidores, en los cuales la adición de más clientes podría significar una transferencia de información más lenta para los usuarios.
- **Robustez:** La naturaleza distribuida de los sistemas peer-to-peer también incrementa la robustez en caso de haber fallos en la réplica excesiva de los datos hacia múltiples destinos, y en los sistemas peer-to-peer puros permite a los nodos encontrar la información sin hacer peticiones a ningún servidor centralizado.
- **Descentralizado:** Estas redes por definición son descentralizadas y todos los nodos son iguales. No existen nodos con funciones especiales, lo cual indica que ninguno es imprescindible para el funcionamiento de la red. Sin embargo algunos sistemas peer-to-peer no cumplen con esta característica.
- **Anonimato:** Se quiere que en estas redes que anónimo el autor de un contenido, el editor, el lector, el servidor y la petición para encontrarlo siempre que así los usuarios lo requieran.
- **Costes Compartidos:** Entre los usuarios del sistema se comparten recursos a cambio de otros recursos. Según la aplicación de red, los recursos pueden ser archivos, ancho de banda, ciclos de procesamiento en disco.

Las redes peer-to-peer también son clasificadas en: **Híbridas y Puras** dada la carencia o no de una entidad central para la gestión del tráfico entre los nodos. Y en **Estructuradas y No Estructuradas** dependiendo de cómo se encuentran organizados o enlazados los nodos. [23]

### 1.3.1. Definición de Red Superpuesta Peer-to-Peer

Una red superpuesta peer-to-peer es una red lógica, construida sobre una red física, que facilita la localización de los recursos distribuidos sin un control centralizado. Los sistemas peer-to-peer son una clase de aplicaciones distribuidas, que explotan los recursos proporcionados por los usuarios en los bordes de Internet, para su beneficio. Estos sistemas han surgido gracias al crecimiento generalizado de conexiones a Internet de banda ancha [22].

### 1.3.1.1. Redes Peer-to-Peer Centralizadas

Este enfoque se basa en una arquitectura donde todas las operaciones y transacciones se hacen a través de un servidor que es el punto de enlace entre todos los nodos de la red, y que a la vez almacena y distribuye los nodos donde se almacenan los contenidos. Una red de este tipo posee las siguientes características: (Figura 1) [23]

- Se rige bajo un único servidor cuya función es ser punto de enlace entre nodos y como servidor de acceso al contenido, el cual distribuye a petición de los nodos de la red.
- Todas las solicitudes y enrutamientos entre nodos dependen exclusivamente del servidor.
- Un sistema simple y rápido.
- Soporta búsquedas complejas de contenidos al tener sólo un servidor que rige la red.

La principal desventaja de un sistema centralizado es que no cumple con una de las características generales de las redes peer-to-peer como es la escalabilidad. Además de esto, el servidor central necesita de varios recursos computacionales y ancho de banda para soportar a un gran número de nodos. [23]

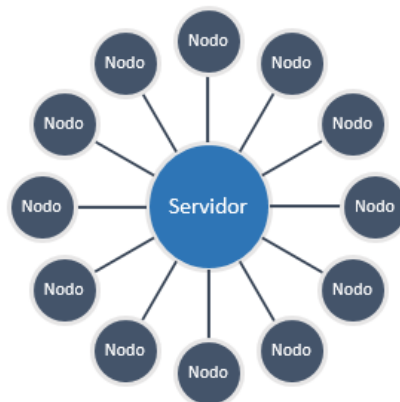


Figura 1. Sistema peer-to-peer centralizado.

### 1.3.1.2. Redes Peer-to-Peer Distribuidas o Puras

Un sistema peer-to-peer distribuido o puro consiste en una red de nodos que no dependen de una unidad o servidor central. Cada nodo de la red tiene las mismas características que los demás y cualquier nodo puede iniciar una comunicación en la red. Las redes de este tipo poseen las siguientes características: (Figura 2) [23]

- No existe una entidad que actúe como servidor central que administre la red.
- Cada nodo de la red actúa como cliente y servidor.
- Cualquier nodo podrá ser removido sin afectar la funcionalidad del sistema.
- Los mecanismos de búsqueda son proporcionados por la cooperación de todos los nodos de la red.

Ya que los nodos están enlazados directamente pueden enviar mensajes de búsqueda a sus vecinos, así, cualquier nodo puede enviar una petición de búsqueda, y posteriormente cada nodo que recibe dicha petición buscará localmente y retornará un resultado hacia sus demás nodos vecinos (dependiendo del tipo de protocolo implementado en la red), sin embargo al no contar con un servidor central, las búsquedas son más lentas y de mayor coste.

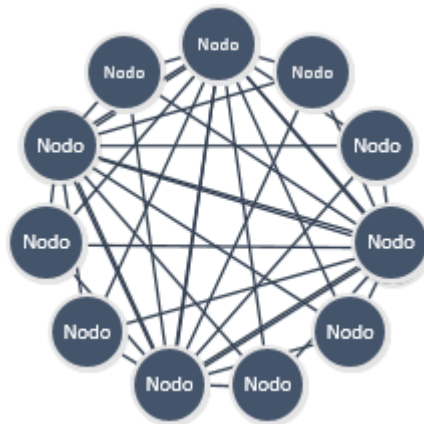


Figura 2. Sistema peer-to-peer puro o distribuido.

### 1.3.1.3. Redes Peer-to-Peer Híbridas

Actualmente la mayoría de aplicaciones peer-to-peer emplean un modelo mixto o híbrido. Este enfoque consiste en que algunos nodos proporcionan algunas funcionalidades extras como el enrutamiento para facilitar la interconexión entre nodos. Los nodos que tienen esta funcionalidad actúan como catálogo de direcciones y ayudan a gestionar el tráfico dirigido hacia otros nodos.

Generalmente los nodos que cuentan con esta funcionalidad son llamados supernodos, que brindan el servicio de descubrimiento de otros nodos normales de la red. Este sistema posee algunas características como: [23]

- Supernodos que ofrecen servicios de forma centralizada.
- Permite una mayor frecuencia en las búsquedas y descargas de contenido.
- Tiene mayor grado de escalabilidad que las redes centralizadas.
- Cualquier nodo podrá ser removido sin afectar la funcionalidad del sistema.
- Los mecanismos de búsqueda son proporcionados por la cooperación de los supernodos.

Los supernodos tienen un alto ancho de banda y un gran tamaño de almacenamiento, que pueden procesar eficientemente una gran cantidad de consultas de los pares ordinarios, por lo tanto, hace un buen uso de la heterogeneidad entre pares. Sin embargo la escogencia de un supernodo necesita y depende de algoritmos para seleccionar un nodo fiable.

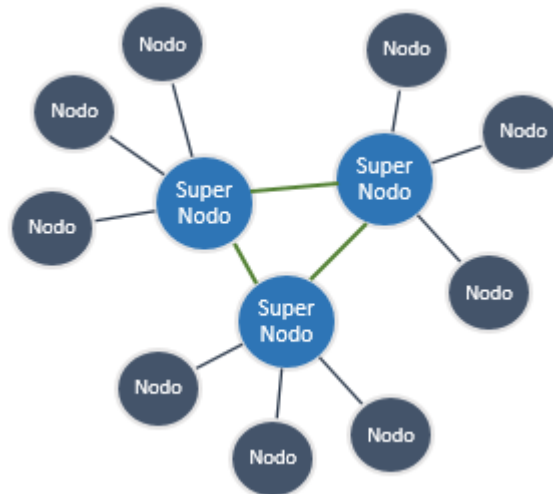


Figura 3. Sistema peer-to-peer híbrido.

### 1.3.2. Redes Peer-to-Peer Estructuradas y No Estructuradas

- **Las redes peer-to-peer estructuradas** mantienen una tabla hash o DHT (Distributed Hash Table). A todo contenido se le da un valor y a uno de los nodos se le hace responsable de una parte específica del contenido. Cuando se recibe una petición por un contenido en particular, un protocolo de carácter global determina qué nodo es responsable de proveer ese contenido, y la búsqueda se dirige a esa dirección [22].
- **Las redes peer-to-peer no estructuradas** se forman cuando los nodos están enlazados de una manera arbitraria. Cuando un nuevo nodo se une a



la red, puede copiar algunos enlaces existentes y crear los suyos propios más tarde. Cuando un nodo busca un contenido, puede encontrarlo con los enlaces que tiene. La desventaja con este sistema, es que si el contenido que se solicita es raro, poco común o no es muy popular, es posible que la búsqueda no obtenga buenos resultados o incluso ninguno, esto quiere decir que los nodos cercanos ya no tienen ese contenido [22].

#### **1.4. Técnica**

Se debe entender como técnica, un procedimiento o conjunto de estos (reglas, normas o protocolos), que tienen como objetivo obtener un resultado determinado, ya sea en el campo de la ciencia, de la tecnología, del arte, de la educación o en cualquier otra actividad [24].

#### **1.5. Recuperación**

Según la definición de la RAE respecto a “recuperar” (1) volver a tomar o adquirir lo que se había perdido (2) Volver a la normalidad después de una crisis [25].

En el contexto de los servicios de comunicación recuperación de un servicio indica retomar un proceso tras la ausencia voluntaria o involuntaria del ente prestador.

#### **1.6. Servicios**

##### **1.6.1. 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. Un servicio se define como una funcionalidad predefinida que puede ser utilizada por los nodos que están conectados a una red superpuesta P2P [26].

De acuerdo a las definiciones anteriores, 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.

### 1.6.2. Clases de servicios

La tecnología peer-to-peer durante los últimos años está abarcando muchas categorías y han revolucionado la forma en la cual se pueden compartir y aprovechar los recursos de forma eficaz. A diferencia de las arquitecturas tradicionales de cliente-servidor, los sistemas peer to peer son medios colaborativos que se desempeñan juntos para realizar ciertas tareas [26].

Los diferentes sistemas peer-to-peer han sido utilizados en diversas aplicaciones ofreciendo diferentes servicios entre los cuales se encuentran principalmente:

- **Difusión de contenido:** La mayoría de los sistemas peer to peer actuales están en esta categoría, ya que tienen una estructura específicamente diseñada para compartir diferentes tipos de medios digitales entre usuarios. En esta categoría se incluyen aplicaciones que van desde la distribución simple de archivos, hasta sistemas más robustos que poseen medios de almacenamiento distribuido para buscar, publicar y recuperar información de manera eficaz y segura [27]. Uno de los subconjuntos más populares de estas tecnologías en la actualidad y que están desempeñando un puesto imprescindible en la transmisión eficiente de video en la red [28], son los sistemas de media streaming peer to peer, los cuales contrastan enormemente con los sistemas de media streaming tradicional, basados en la arquitectura cliente-servidor, contrarrestando así el creciente uso de banda ancha y los costos que esto acarrea [29]. Algunos ejemplos son: Pixtream [30], Gnutella [31], sistemas de live streaming o P2PTV como PULSE [11], sistemas de intercambio de archivos como BitTorrent [32].
- **Comunicación:** Este conjunto de sistemas ofrece una estructura y configuración que facilita la comunicación directa, usualmente en tiempo real. Aquí se pueden incluir los chat, sistemas de mensajería instantánea y aplicaciones que son aún más populares [30]; por ejemplo una de las grandes ventajas de la telefonía peer to peer frente al modelo tradicional de voz sobre IP, es que el flujo de datos, en este caso el flujo de audio de la voz, circula entre extremos intervinientes en la comunicación como cualquier otro tráfico de datos, sin la mediación de ningún servidor, tarea que implementa el popular sistema peer to peer Skype [33] a lo que se refiere a compartición de contenidos como flujos de audio.

- **Computación distribuida:** Esta categoría incluye sistemas cuyo objetivo es aprovechar la potencia de procesamiento de información de cada par disponible en la red. Esto se logra dividiendo las tareas intensivas de computación en unidades de trabajo pequeñas y distribuyéndolas entre pares diferentes que ejecutan su unidad de trabajo correspondiente y devuelven los resultados [27]. Ejemplos de estos sistemas tenemos proyectos como SetiAtHome [34] y GenomeAtHome [35].
- **Bases de Datos:** Se ha realizado un trabajo considerable en el diseño de sistemas de bases de datos distribuidas basadas en infraestructuras peer to peer que difieren un poco de la filosofía tradicional. Ejemplos como PIER [36] un motor de búsquedas distribuido escalable construido sobre una red peer to peer superpuesta, que permite consultas relacionales ejecutadas a través de miles de computadores. PIAZZA [37] sistema que proporciona datos (por ejemplo de una base de datos relacional), esquemas (u ontologías) o ambos. Y Edutella [38], un proyecto que provee una infraestructura de metadatos y la capacidad de búsqueda de aplicaciones peer to peer.

## 1.7. Sistemas de media streaming

### 1.7.1. Definición de media streaming

El término media streaming se refiere a los productos y técnicas utilizados 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 [30]. 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 (stream) [30]

#### 1.7.1.1 Cuello de botella en el ancho de banda

El contenido multimedia, en especial el video, suele necesitar muchos bytes para ser representado. Es por esta razón que el principal problema para el media streaming es al ancho de banda requerido para poder realizar la transmisión. Esto es especialmente importante si se tiene en cuenta que en la mayoría de los casos, no es uno sino muchos los receptores de contenido en un sistema de streaming.

Este proceso se conoce como broadcasting [30]. Es decir la transmisión de contenido desde una sólo fuente hasta muchos receptores. Dado que el ancho de banda suele ser un recurso limitado y costoso, puede decirse que es un factor determinante para el media streaming, toda vez que define el número de clientes hacia los cuales se puede transmitir el contenido.

La arquitectura de streaming más simple es la de múltiples conexiones unicast. Una transferencia en red tradicional es llamada unicast. Se realiza de un computador a otro. A la hora de realizar streaming, esto significa que, para lograr transmitir, el emisor de contenidos deberá establecer una conexión unicast con cada cliente receptor. El uso de múltiples conexiones implica que el ancho de banda necesario para poder transmitir es equivalente al ancho de banda requerido por una conexión unicast multiplicado por todos los clientes que se desea servir. Debido a esto se genera un cuello de botella en cuanto al ancho de banda que se requiere en el lado del servidor de contenidos. Configurar un servidor de media streaming con esta tecnología puede resultar muy costoso si se tienen muchos clientes [30].

## **CAPÍTULO II. BÚSQUEDA Y RECUPERACIÓN DE SERVICIOS**

En este capítulo se describen las diferentes técnicas, algoritmos y características de búsqueda de servicios sobre redes superpuestas peer-to-peer como parte esencial del proceso de recuperación de servicios, y a continuación se describen diferentes características de recuperación de servicios en redes peer-to-peer.

### **2.1. Descubrimiento de servicios**

El descubrimiento de servicios puede ser definido como la capacidad de buscar, encontrar y utilizar posteriormente un servicio basado en alguna descripción publicada de su funcionalidad y parámetros operacionales. El descubrimiento de servicios puede ser abordado bajo dos enfoques principalmente: Descubrimiento sintáctico y semántico [26].

El descubrimiento sintáctico se basa en técnicas de comparación de interfaces (por ejemplo: WSDL, IDL, Interfaces RMI, etc.) o palabras clave para realizar la búsqueda de servicios, requiriendo coincidencias exactas a nivel sintáctico entre las descripciones de los servicios y los parámetros empleados [26].

La descripción semántica de servicios busca la posibilidad de que las máquinas puedan realizar un mayor procesamiento y razonamiento sobre los servicios. La principal diferencia con las descripciones sintácticas es la forma como se representa la información. Mientras la sintáctica se enfoca en definir los servicios a partir de los mensajes de entrada y salida, tipos y partes de los mensajes, la semántica busca ofrecer información acerca de la funcionalidad del servicio. La representación semántica del contenido de las descripciones de un servicio permiten a las máquinas entender y procesar su contenido, soportando el descubrimiento e integración dinámica de los servicios [26].

En el presente trabajo de investigación, se utilizará un enfoque sintáctico, debido a que éste permite realizar búsqueda de servicios con mayor facilidad sin disminuir la calidad de los resultados.

### **2.2. Búsqueda de servicios en Redes Superpuestas Peer-to-Peer No Estructuradas**

Un método de búsqueda debe tener en cuenta aspectos tales como el envío de consultas, el conjunto de nodos que reciben los mensajes relacionados con la consulta, la forma de los mensajes, el procesamiento local, el almacenamiento de índices y su mantenimiento, entre otros [26].

Los métodos de búsqueda de servicios y recursos, utilizados en las redes superpuestas P2P, pueden utilizar el esquema ciego o el informado, así como la búsqueda basada en la inundación o sin ella, y con o sin la utilización de índices en los nodos. Todas estas características, de los métodos de búsqueda, brindan un punto de partida para escoger un método eficiente de búsqueda, o la combinación de varios de ellos, para ser utilizado en la técnica de recuperación y ejecución de servicios.

### 2.2.1. Métodos de búsqueda de servicios

Basados en el trabajo de investigación [26] se describen a continuación una variedad de métodos de búsqueda de servicios y recursos, para redes superpuestas P2P no estructuradas.

| Método de Búsqueda                             | Descripción  |
|--|--|
| <b>Gnutella</b>                                | El algoritmo original Gnutella, también conocido como esquema de inundación <sup>1</sup> , contacta todos los nodos accesibles dentro del rango TTL. Sus características fundamentales son su simplicidad y la enorme sobrecarga que produce al contactar muchos nodos.        |
| <b>Modified-Breadth First Search (Mod-Bfs)</b> | Este método utiliza una variación del esquema de inundación, donde los nodos escogen aleatoriamente solo una proporción de sus vecinos para continuar con la búsqueda.   |
| <b>Iterative Deepening</b>                     | En este método se inician múltiples búsquedas en anchura (Breadth First Search), con límites de profundidad cada vez más largos, hasta que la consulta se realice de manera satisfactoria o hasta que la profundidad máxima $D^2$ sea alcanzada.                               |
| <b>Random Walks (Rwalks)</b>                   | El nodo solicitante envía una cantidad $k$ de mensajes de búsqueda a un igual número de nodos vecinos, elegidos aleatoriamente. Cada uno de estos mensajes sigue su propia ruta, teniendo nodos intermedios que eligen sus propios nodos vecinos aleatoriamente, en cada paso. |
| <b>Super-Peer Approaches - Gnutella2 (G2)</b>  | En el método de súper nodos, cuando un súper nodo <sup>3</sup> recibe una consulta de un nodo normal <sup>4</sup> , el súper nodo transmite la consulta a sus hojas correspondientes y a sus súper nodos vecinos. Estos  |

<sup>1</sup> Flooding Scheme: Esquema de Inundación o Búsqueda Ciega

<sup>2</sup> Variable de profundidad

<sup>3</sup> Un súper nodo también es llamado centro, eje o hub.

<sup>4</sup> Un nodo normal también es llamado hoja o leaf.

|  |  |
|--|--|
|  | súper nodos procesan la consulta a nivel local y la transmiten a sus hojas correspondientes. Ningún otro nodo es visitado en este algoritmo.   |
| <b>Intelligent-Bfs (Int-Bfs)</b>           | Este método es una versión informada del método Modified-BFS. Los nodos almacenan la información de sus nodos vecinos en pares de la forma "NodoID-Consulta" para solicitudes recientemente contestadas desde o hacia sus nodos vecinos con el fin de clasificarlos.   |
| <b>Adaptive Probabilistic Search (Aps)</b> | En este método, cada nodo mantiene un índice local, que consiste en una entrada para cada objeto que ha sido solicitado por cada nodo vecino. El valor de esta entrada refleja la posibilidad relativa de que el vecino de este nodo sea elegido como el siguiente salto en una futura petición para el objeto especificado.                             |
| <b>Swapping-Aps</b>                        | Este método utiliza la estrategia de seleccionar un ancho de banda más eficiente para minimizar la cantidad de mensajes de actualización a lo largo de la ruta inversa.  |
| <b>Local Indices (Li)</b>                  | En este método cada nodo indexa los objetos almacenados en todos los nodos dentro de un cierto radio $r$ , y puede responder a las peticiones en nombre de todos ellos. La búsqueda es realizada de una manera semejante a BFS, pero solo los nodos que están a una cierta profundidad del nodo solicitante, procesan la consulta.                       |
| <b>Gia</b>                                 | En este método, los nodos solicitantes despliegan caminantes parciales, con el fin de descubrir varios objetos. Cada nodo decide transmitir la consulta al nodo vecino que tenga la mayor capacidad anunciada. Esta capacidad es una métrica definida por el usuario que refleja el poder de procesamiento de un nodo dentro del sistema.                |
| <b>Gnutella With Shortcuts (Gs)</b>        | En este método, se propone la adición de accesos directos o shortcuts, al algoritmo original Gnutella. Un acceso directo puede entenderse como una conexión o enlace directo a un nodo que recientemente ha respondido las peticiones de una manera satisfactoria. El sistema de inundación original se utiliza para localizar inicialmente los objetos. |
| <b>Freenet</b>                             | Es un sistema peer-to-peer descentralizado de almacenamiento y recuperación de información, que permite a los usuarios compartir el espacio en disco no usado. FreeNet usa un método de búsqueda similar a DFS, pero con enrutamiento inteligente. En este método, el nodo envía la consulta solo a un nodo vecino a la vez.                             |

Tabla 1. Descripción de métodos de búsqueda

### 2.3. Características y selección de los métodos de búsqueda.

En [26] se describen las siguientes características que deben tenerse en cuenta para la búsqueda y el descubrimiento de servicios y recursos sobre redes superpuestas P2P:

- **Desempeño:** Mide el tiempo que tarda una operación de búsqueda para retornar un conjunto de resultados. Una aplicación descentralizada, tiene que iniciar una búsqueda, a lo largo de la red, para determinar donde se encuentra un recurso; por lo tanto, en adición al tiempo requerido por cada nodo para procesar la consulta, un tiempo extra es requerido para establecer la conexión al siguiente nodo y propagar la consulta. Debido a esta propagación, el tiempo para una operación de búsqueda se incrementa con el incremento del número de nodos en el sistema.
- **Tolerancia a fallos:** Una de las características principales de las redes superpuestas P2P es el cambio constante de su topología. Este cambio es debido a que cualquier nodo puede entrar y salir del sistema, en cualquier momento. Se espera que el sistema se recupere, de la pérdida de datos y de la pérdida de información de enrutamiento, cuando los nodos entren o dejen la red, por medio de replicación de datos o rutas alternas, respectivamente. También, se espera que el sistema continúe suministrando conectividad y confiabilidad del servicio, a todos los nodos que se encuentran conectados, sin que ellos sean conscientes de los cambios en la topología de red. La tolerancia a fallos denota la extensión de un mecanismo de búsqueda, que pueda encontrar una solución a los problemas que se presenten
- **Costo de la búsqueda:** Esta característica indica, el costo que necesitan los nodos para procesar una consulta y puede ser representada por el número de nodos que procesan la consulta.
- **Precisión de la búsqueda:** Esta característica mide la eficiencia del método de búsqueda. La precisión de la búsqueda representa la proporción entre el número de resultados que coinciden con la consulta y el número total de resultados devueltos por la consulta.
- **Confiabilidad:** La confiabilidad denota el éxito del mecanismo de búsqueda. La combinación de la tolerancia a fallos y la eficiencia del método de búsqueda determinan su confiabilidad.
- **Control local:** Conceptualmente, las aplicaciones descentralizadas están compuestas de nodos que son autónomos y tienen un control local completo sobre sus datos. Sin embargo, en ciertos casos, un nodo puede almacenar y mantener datos de otros nodos. La característica de control local, ayuda a distinguir los sistemas que permiten a sus nodos tener un control local de sus datos y los que no lo permiten.



- **Escalabilidad:** La escalabilidad se refiere, a la habilidad de un mecanismo de búsqueda para soportar un gran número de nodos, un gran número de interacciones entre los nodos, o ambos. Las ventajas de la escalabilidad, incluyen un incremento en la probabilidad de encontrar un recurso y un posible incremento en la redundancia del recurso a través de los nodos, debido a que más nodos pueden almacenar múltiples copias del recurso. La habilidad de los nodos para tener múltiples conexiones con otros nodos, también afecta la escalabilidad, debido a que los nodos pueden tener mecanismos para mantener la calidad de las conexiones, cuando el número de estas se incrementa.
- **Flexibilidad de la búsqueda:** Existen algunas técnicas de búsqueda que solo permiten que los recursos sean buscados por su nombre. Esta característica se refiere a que la técnica de búsqueda debe permitir a los usuarios buscar un recurso, no solo por su nombre, sino por otros conceptos
- **Costo de banda ancha:** El consumo de ancho de banda, en aplicaciones P2P, es medido generalmente en términos del número de mensajes intercambiados entre nodos. Los nodos en una red superpuesta P2P, generalmente, tienen restricciones de memoria y conectividad. Dependiendo del método de búsqueda, los nodos deben procesar una cantidad de mensajes, en algunas ocasiones el número de mensajes se incrementa demasiado y algunos mensajes pueden ser borrados, debido a la falta de capacidad del nodo de procesarlo, afectando el rendimiento del sistema. La reducción del ancho de banda, es uno de los objetivos más importantes de cualquier mecanismo de búsqueda.
- **Costo de almacenamiento:** Ciertas técnicas de búsqueda, se basan en algún tipo de información de enrutamiento, para dirigir las consultas a los nodos que pueden responder mejor a ellas. Los nodos implicados, mantienen tablas de enrutamiento y punteros a los datos y a otros nodos en el sistema. La memoria extra, requerida para almacenar esta información de enrutamiento, es llamada costo de almacenamiento. El costo de almacenamiento, también depende del número y el tamaño de los recursos almacenados en uno o varios nodos.

En [26] a partir de las características mencionadas anteriormente se estudian y evalúan los métodos de búsqueda con el objetivo de determinar cuáles sobresalen en términos de eficiencia. Para esto las características son abordadas y agrupadas balanceando costos y calidad de resultados.

Finalmente en el trabajo de investigación que se ha venido mencionando se concluye escogiendo los métodos *Local Indices*, *Gnutella with Shortcuts (GS)* y *Super-Peer Approaches - Gnutella2 (G2)* Ver Tabla 2, gracias a que estos logran

una buena calidad en los resultados además de un número alto de hits producidos sacrificando el costo de ancho de banda y procesamiento.

| <b>Métodos de Búsqueda Seleccionados</b>      |  |
|---|--|
| <b>Local Indices</b>                          | <p>Alto número de hits (resultados).</p> <p>Alto porcentaje de éxito de la búsqueda.</p> <p>Deficiencias debido alto número de mensajes producidos y su tolerancia a fallos.</p>   |
| <b>Gnutella with Shortcuts (GS)</b>           | <p>Número de hits (resultados), superior en relación a Local Índices y G2.</p> <p>Porcentaje de éxito superior en relación a Local Índices y G2.</p> <p>Deficiencia en el alto número de mensajes producidos.</p>                        |
| <b>Super-Peer Approaches - Gnutella2 (G2)</b> | <p>A pesar de que el número de hits obtenidos es muy bajo, el número de mensajes se reduce considerablemente.</p> <p>Método utilizado en el descubrimiento de recursos por muchas tecnologías P2P, entre las cuáles se incluye JXTA.</p> |

Tabla 2. Métodos de búsqueda seleccionados

## 2.4. Recuperación de servicios

La recuperación de servicio puede entenderse como la capacidad de poder utilizar de nuevo un servicio que previamente había sido perdido a causa de un fallo ocurrido en el cliente, servidor o red.

A continuación se revisan propuestas donde se identifican tipos de fallos comunes y algunas propuestas para manejarlos o mitigarlos, en el ámbito de la recuperación de servicios.

En [39] se realiza un análisis sobre reportes detallados de fallos de servicios de gran escala con el objetivo de responder a la pregunta cómo y porqué los servicios fallan. Este trabajo evidencia que las principales causas son los errores humanos y de red, y que, para mantener disponibilidad en el servicio una de las técnicas más usadas para mitigar este tipo de errores es la redundancia o replicación.

En [40] se define redundancia como el hecho de establecer una o múltiples réplicas que están capacitadas para desplegar un mismo servicio, algunas de ellas actúan como réplicas activas, otras actúan como réplicas respaldo de las anteriormente nombradas réplicas activas. En caso de fallo, un algoritmo de selección de réplica segura, selecciona un reemplazo que garantice que el servicio continuará siendo desplegado de manera apropiada.

La redundancia para garantizar la alta disponibilidad de un servicio ha sido un tema que ha sido discutido durante décadas [41][42][43]. A nivel industrial se sigue notando que para minimizar el impacto de un desastre, en sus ambientes físicos tradicionales la tendencia es tener duplicados de su infraestructura tanto de hardware y como también de software para asegurar la disponibilidad. [44][45][46][47]

La redundancia es una práctica que también ha sido aplicada en sistemas embebidos [48] [49].

Las soluciones tradicionales en las últimas décadas sugieren usar la redundancia física y/o de software para mantener la alta disponibilidad en los servicios. Estas soluciones implementan generalmente mecanismos de monitorización cuya tarea consiste en vigilar el estado del servicio para garantizar su correcto funcionamiento. [40]

## **2.5. Recuperación de servicios en Redes Superpuestas Peer-to-Peer No Estructuradas**

A continuación se referencian algunas propuestas en relación a la temática del manejo de fallos en la prestación de servicios en las redes superpuestas peer-to-peer.

En [50] se propone un esquema de tolerancia a fallos para un sistema de video en demanda (Video on Demanda VoD) desplegado en una red peer-to-peer. Para garantizar la disponibilidad y calidad del servicio (Quality of Service QoS) el esquema explota un conjunto de nodos para almacenar estáticamente réplicas de porciones de los archivos multimedia. Cabe destacar que en esta propuesta la red peer-to-peer se utiliza totalmente para desplegar el servicio de video en demanda y algunos nodos son usados para la redundancia de contenido.

En [51] se presenta un sistema de video en demanda (VoD) que aprovecha la red peer-to-peer para transmitir sobre ella y enfrentar el problema de ancho de banda. La solución explota técnicas de reordenamiento de nodos y el uso de réplicas en caché.

En [52] se propone un framework para computación distribuida tipo cycle-stealing usando una red peer-to-peer. Para la recuperación de un proceso llevado a cabo por un nodo que desaparece de la red superpuesta se explota la migración de objetos y el envío de mensajes (que contienen puntos de control 'checkpoints' del proceso) a los demás nodos de la red.

En [53] se implementa un simulador para eventos discretos distribuidos sobre una red peer-to-peer. En esta propuesta se explota el concepto de redundancia para hacer rollback a un estado anterior del sistema.

En [54] se analiza el problema de estabilidad en los juegos multiplayer masivos en línea en las redes peer-to-peer, propone la implementación de árboles multicast y extiende el concepto de 'cluster' con el fin de buscar estabilidad. Cabe destacar que este trabajo solo concluye en términos de planteamiento la solución para la tolerancia a fallas.

En [55] se propone un sistemas peer-to-peer para video en demanda hibrido, para el problema de fallos de red se apuesta por una técnica de migración de servicios que varía el punto de origen de las transmisiones esto a través de rastreadores inteligentes (smart trackers) ubicados en la red.

Para migración de servicios en redes superpuestas peer-to-peer también se tienen propuestas como [56], que, para la detección de fallos proponen técnicas basadas en el retardo de paquetes para sistemas de streaming.

En las propuestas [57] y [58] para sistemas de streaming la detección de fallos se realiza teniendo en cuenta el factor ancho de banda. Los nodos servidores realizan transmisiones dinámicas de una parte de la tasa total a los clientes. Este trabajo no tiene en cuenta técnicas para recuperar o migrar el servicio en caso de caídas de los nodos servidores.

## **2.7. Priorización de las características del sistema Peer-to-Peer y de la técnica de recuperación y ejecución de servicios**

Los trabajos anteriormente mencionados tratan sobre la prestación de servicios, tipos de fallos comunes y algunas propuestas para manejarlos o mitigarlos. Con base en lo anterior el presente trabajo "Técnica de Recuperación y Ejecución de Servicios en Redes Superpuestas Peer-to-Peer No Estructuradas" se enfocará en la prestación de servicios usando la red superpuesta peer-to-peer como medio de descubrimiento, búsqueda y recuperación de servicio. La red peer-to-peer en el despliegue de servicios debe cumplir en lo posible con características de Escalabilidad, Disponibilidad y Tolerancia a Fallos.

Teniendo en cuenta que un sistema tolerante a fallos debe considerar todos los tipos de fallos en el sistema (cliente, servidor, red) el mecanismo de recuperación en este trabajo se centrará en los siguientes tipos de fallas:

- El nodo que presta el servicio ha dejado de permanecer en la red superpuesta peer-to-peer a causa de un problema de red.
- El nodo que presta el servicio ha sido desconectado de la red de manera voluntaria o involuntaria por parte del usuario.

Por otra parte, para los tipos de fallas a tratar en el presente trabajo y teniendo como referencia las consideraciones encontradas en los trabajos relacionados [39] [40][44][46][47]. El mecanismo de recuperación debe priorizar las siguientes características:

- **Monitorización:** Entidad que debe estar encargada de vigilar la condición del servicio.
- **Redundancia:** Para garantizar la disponibilidad del servicio a través de nodos replica.

Es importante destacar que para poder encontrar un servicio deseado y, en caso de fallas, posteriormente poder recuperarlo, los métodos seleccionados en la sección 2.3 cumplen con las siguientes características, que se describen en la misma sección: [26]

- **Desempeño**
- **Tolerancia a Fallos**
- **Costo de la Búsqueda**
- **Precisión de la Búsqueda**
- **Confiabilidad**
- **Control Local**
- **Escalabilidad**

Estas características forman parte de la técnica de recuperación y ejecución de servicios en redes superpuestas peer-to-peer no estructuradas. En la siguiente figura se representa la caracterización de la técnica de recuperación y ejecución de servicios en redes superpuestas peer-to-peer no estructuradas:

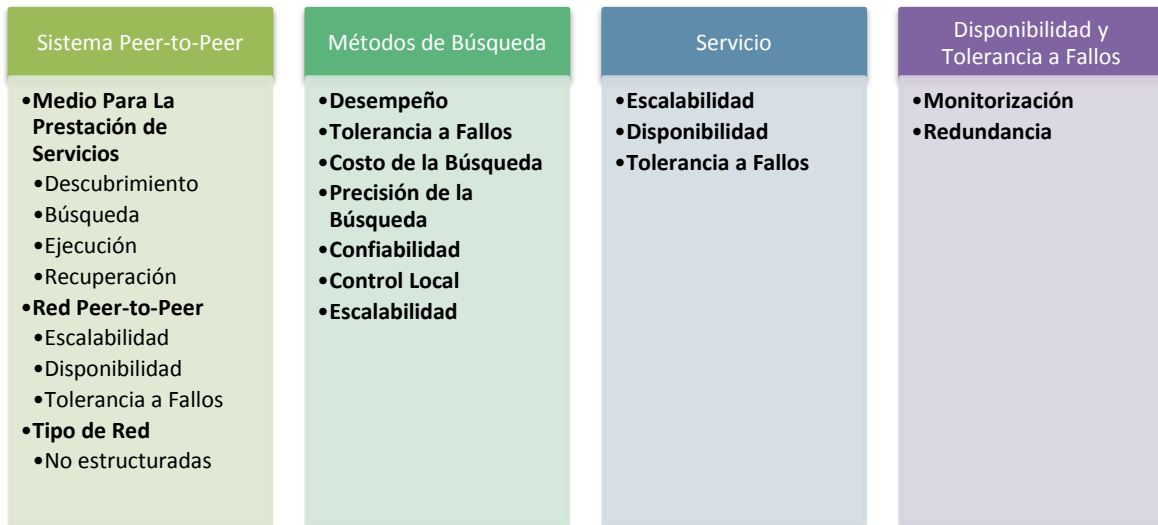


Figura 4. Caracterización de la técnica de recuperación y ejecución

## 2.8. Selección del perfil del servicio

En base a las características que un mecanismo de recuperación debe priorizar, mencionadas en la sección 2.7; el perfil de servicio seleccionado para verificar dichas características y que mejor evidencia el comportamiento de la técnica de recuperación y ejecución de servicios, es un servicio de media streaming en vivo, ya que en este tipo de servicio el usuario visualiza el video o audio en paralelo mientras se descarga.

## CAPÍTULO III. TÉCNICA DE RECUPERACIÓN Y EJECUCIÓN DE SERVICIOS EN REDES PEER-TO-PEER

### 3.1. Introducción

Una vez estudiado y definidas las características más importantes de los métodos de búsqueda y recuperación de servicios, se procede a describir la adecuación de la técnica de recuperación y ejecución de servicios.

La solución abarca en gran medida el aprovechamiento de las características de comunicación y transporte de las redes superpuestas peer-to-peer y por otra parte la aplicación de conceptos de búsqueda, redundancia y monitorización.

### 3.2. Arquitectura del Sistema

Después de describir los beneficios ofrecidos por los sistemas peer-to-peer se puede pensar en publicar, localizar y consumir servicios y, en caso de fallas, aplicar una técnica de recuperación apoyada por la redundancia y monitorización.

A continuación en la figura 5 se tiene una visión general del sistema.

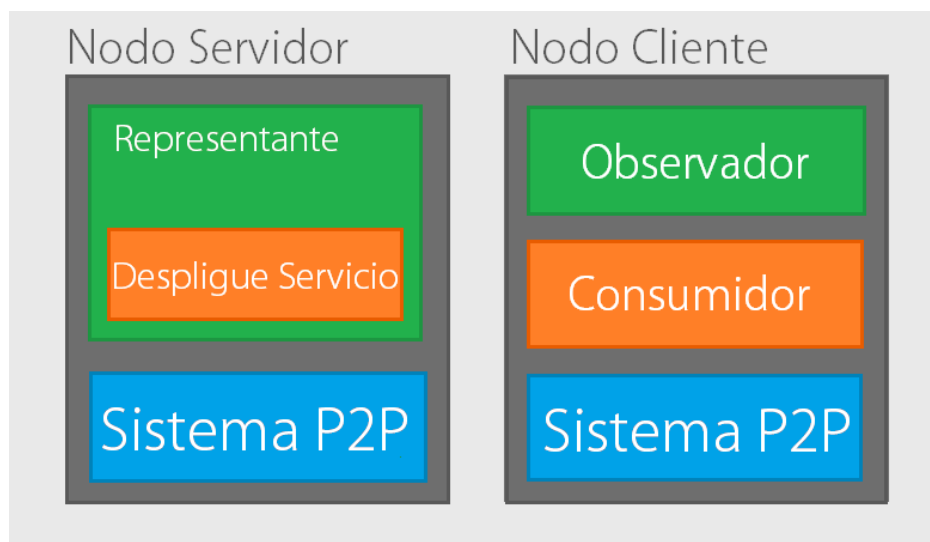


Figura 5. Arquitectura del sistema

### 3.2.1. Sistema Peer-to-Peer

El sistema peer-to-peer proporciona un conjunto común de protocolos para el desarrollo de aplicaciones peer-to-peer. En la figura 6, se muestran los componentes del sistema peer-to-peer.

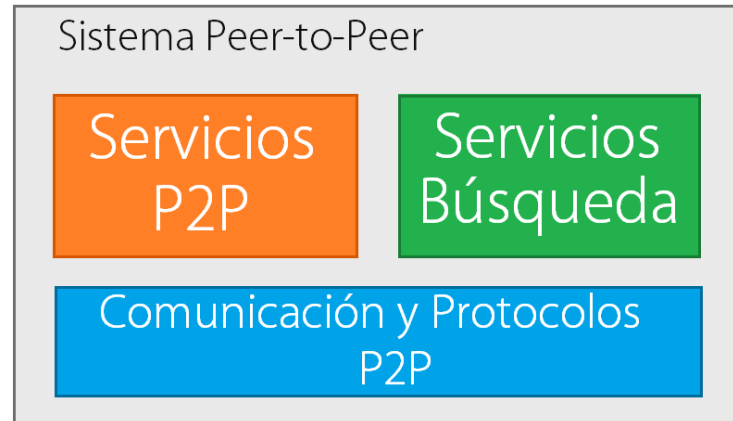


Figura 6. Sistema peer-to-peer

El sistema peer-to-peer se encarga de crear, publicar y buscar los respectivos avisos (advertisements<sup>5</sup>), que representan los recursos ofrecidos en la red superpuesta peer-to-peer (ej. servicios). También se encarga de las comunicaciones y del envío y recepción de mensajes entre cada nodo de la red superpuesta peer-to-peer, a través de diferentes protocolos peer-to-peer.

La propiedad caché en el componente servicios de búsqueda es explotada para lograr buen desempeño en los procesos de búsqueda y recuperación de servicios. El caché permite almacenar de manera local en el nodo cliente el listado de los nodos servidor disponibles que son resultado de la búsqueda del servicio realizada. En caso de que el nodo cliente pierda el servicio, el algoritmo de recuperación consultará en primera instancia, el listado de posibles nodos servidor disponibles en el caché, en caso de éxito utiliza aleatoriamente un servidor del listado, por el contrario la red será consultada a través de una nueva búsqueda del servicio lo que resulta más costoso en el sistema.

---

<sup>5</sup> Son documentos XML, que son utilizados para describir recursos de la red superpuesta P2P



### 3.2.2. Representante

El Representante del servicio a desplegar se encarga de crear el mensaje descriptor del servicio a publicar, especificando los parámetros necesarios para establecer la comunicación con el mismo.

### 3.2.3. Consumidor

El Consumidor del Servicio a Desplegar toma el descriptor creado por el representante y usa los parámetros de comunicación del servicio y presenta una interfaz para consumo del mismo.

### 3.2.4. 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.

## 3.3. Descripción de la Técnica de Recuperación y Ejecución de Servicios

Para describir la técnica de recuperación y ejecución de servicios es importante describir en detalle las tareas de los nodos cliente y servidor presentes en el sistema peer-to-peer. También se hace necesario precisar la labor del representante, el consumidor, el observador, el cómo se realiza la comunicación entre nodos, el despliegue del servicio, el intercambio de mensajes y sus interacciones dentro de la red superpuesta.

### 3.3.1. Acciones de los nodos

#### 3.3.1.1. Acciones del nodo Servidor

- **Iniciar Servidor:** Realiza las siguientes tareas: Crear Nodo Servidor, Crear Grupo peer-to-peer, Guardar Grupo Peer-to-Peer, 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:** Esta tarea crea un nodo teniendo en cuenta parámetros como tipo de nodo, nombre del nodo y puerto. Define un identificador único para el nodo.
- **Crear Grupo Peer-to-Peer:** Esta tarea crea un grupo peer-to-peer específico para cada servicio. Por lo tanto, cada nodo que quiera usar el

servicio debe registrarse en el respectivo grupo. El nombre del grupo será asignado dependiendo del nombre o tipo del servicio prestado.

- **Guardar Grupo Peer-to-Peer:** El proceso de creación de un grupo en una red superpuesta peer-to-peer 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:** Esta acción publica el servicio en la red superpuesta peer-to-peer usando un aviso de publicación de servicio. Gracias a esto la red se entera de la existencia del servicio prestado por el nodo.
- **Guardar Nodo Cliente:** Cuando un nodo cliente se conecta a un nodo servidor, el nodo cliente y sus recursos, si tiene recursos disponibles, son almacenados en una lista a manera de índices locales, para facilitar la localización y envío de mensajes posteriormente.

### 3.3.1.2. Acciones del nodo Cliente

- **Iniciar Cliente:** Realiza las siguientes tareas: Crear Nodo Cliente, Crear Grupo Peer-to-Peer, Guardar Grupo Peer-to-Peer, 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 crea un nodo teniendo en cuenta parámetros como tipo de nodo, nombre del nodo y puerto. Define un identificador único para el nodo.
- **Crear Grupo Peer-to-Peer:** Esta tarea crea un grupo peer-to-peer específico para cada servicio. Por lo tanto, cada nodo que quiera usar el servicio debe registrarse en el respectivo grupo. El nombre del grupo será asignado dependiendo del nombre o tipo del servicio prestado.
- **Guardar Grupo Peer-to-Peer:** El proceso de creación de un grupo en una red superpuesta peer-to-peer 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:** Es la acción encargada de buscar los nodos que prestan el servicio deseado. El servicio es buscado por su nombre (búsqueda sintáctica) en la red superpuesta peer-to-peer.

- **Guardar Nodo Servidor:** Realizada la búsqueda de 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:** Esta acción inicia la conexión de un nodo cliente, hacia un nodo que está prestando un servicio y lo consume.
- **Recuperar Servicio:** Esta tarea se encarga de vigilar la condición del servicio y en caso de pérdida, realiza una nueva búsqueda con el objetivo de encontrar y consumir el mismo servicio.

### 3.3.2. Comunicación entre nodos (Clientes y Servidores)

La comunicación entre nodos se logra gracias a los protocolos de transporte y comunicación soportados por las redes superpuestas peer-to-peer, sin embargo para que los nodos se comuniquen de mejor manera, es necesario que cada uno de los mensajes enviados maneje un lenguaje común, que pueda ser usado para construir conversaciones más complejas.

- **Aviso:** Mensaje que anuncia la existencia de un recurso (servicio) en la red superpuesta peer-to-peer. Un aviso debe estar representado por un identificador único.
- **Descriptor:** Elemento integrado al aviso que especifica la descripción y parámetros necesarios para interactuar con un servicio.

### 3.3.3. Servicio a Desplegar

Representa el servicio a prestar o desplegar a través del sistema peer-to-peer. El servicio debe estar representado por un descriptor que especifica los parámetros, requisitos y formatos necesarios para interactuar con el servicio. En el nodo servidor el representante es el encargado de definir el descriptor y este posteriormente es integrado al aviso de publicación que es usado en el proceso de publicación de servicio por el nodo servidor. En el nodo cliente el consumidor usando el descriptor presenta una interfaz para el consumo del servicio. Ver figura 7.

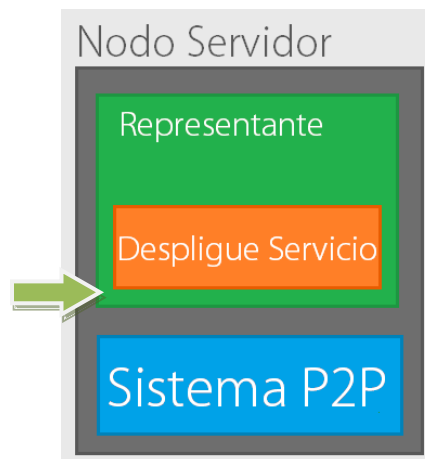


Figura 7. Descripción modular del nodo servidor

### 3.3.4. Conexión

En el sistema peer-to-peer la conexión se presenta en dos escenarios. El primero, cuando nodos clientes y servidores se unen a la red superpuesta y establecen enlaces entre ellos para luego poder realizar operaciones de búsqueda y recuperación de servicios. Ver Figura 8.

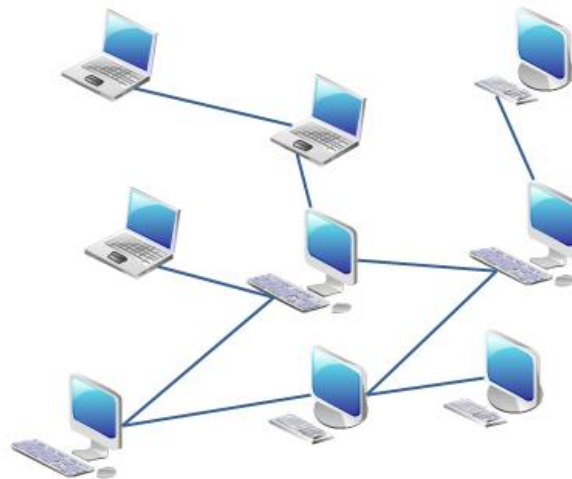
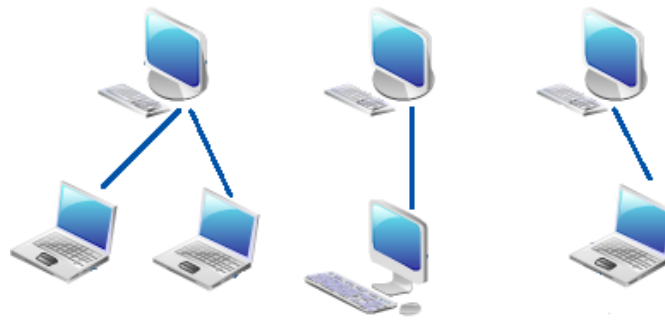


Figura 8. Conexión aleatoria de nodos

Una vez realizada la búsqueda y encontrado un servicio deseado el segundo escenario de conexión se presenta cuando un nodo cliente establece enlace directo con el servicio desplegado en un nodo servidor. Ver Figura 9.



**Figura 9. Conexión directa entre nodos**

### **3.3.5. Mensajes**

Para la comunicación (envío y recepción) y describir recursos (nodos, grupos, canales, servicios, etc.) en la red superpuesta es necesario el intercambio de mensajes los cuales pueden ser representados por formatos para el intercambio de datos como XML o JSON. Avisos y descriptores como se describió en 3.3.2.

### **3.3.6. Envío de Mensajes**

Cuando un nodo ingresa a la red es necesario que ante los demás nodos (clientes y servidores) describa sus recursos. Los recursos de la red pueden ser por ejemplo la existencia del nodo mismo, grupos, canales y servicios. Para llevar a cabo lo anterior un nodo realiza procedimientos de publicación en el cual se propagan avisos en la red para anunciar sobre la existencia de los recursos que ofrece.

Una de las desventajas de los sistemas peer-to-peer, es que no se han implementado estándares para la comunicación y el envío de mensajes, por lo tanto para este propósito cada una de las implementaciones posee su propia serie de protocolos Peer-to-Peer [59]. El desarrollo de este aspecto dependerá de la implementación de redes superpuestas a usar y de sus protocolos.

### **3.3.7. Mensajes en el Servicio a Desplegar**

Para el servicio a desplegar en la red superpuesta es necesario definir un tipo de mensaje descriptor en el cual se especificarán los parámetros, requisitos y formatos necesarios para interactuar con el servicio.

- **Representante:** En el nodo servidor el representante usando un lenguaje común es el encargado de crear el mensaje descriptor donde se especifica el servicio.
- **Consumidor:** En el nodo cliente el consumidor es el encargado de tomar el mensaje descriptor del servicio y presentar interfaz para el consumo.

Es importante destacar que tanto el representante como el consumidor deben ponerse de acuerdo sobre los nombres de las etiquetas a especificar en el descriptor del servicio.

### 3.3.8. Búsqueda y Recuperación de Servicio

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 con el deseo de consumir un servicio específico, realiza una búsqueda sintáctica sobre los avisos de publicación, como resultado obtendrá ninguno o un conjunto servidores disponibles. A continuación el nodo cliente puede establecer una conexión directa con uno de los nodos servidores encontrados y utilizar el servicio prestado usando el descriptor 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.

### 3.3.9. Observador

El observador es el encargado de realizar el monitoreo de la condición del servicio y, en caso de desconexión, disparará el mecanismo de recuperación para obtener otro servicio igual. El observador debido a sus características de monitorización puede ser montado aprovechando las ventajas ofrecidas por los agentes inteligentes o el desarrollo especializado de hilos de ejecución. Sin embargo lo anterior dependerá de la implementación de la red superpuesta y de las condiciones para la compatibilidad e integración.

## 3.4. Prestación de Servicio

A continuación se describen los procesos publicación del servicio, búsqueda del servicio y recuperación del servicio además representándolos respectivamente a través de diagramas de flujo.

### **3.4.1. Publicación del Servicio**

La creación del servicio, es el proceso mediante el cual, un nodo servidor publica el servicio en el sistema peer-to-peer.

En la figura 10 de la página 40 se muestra el diagrama de flujo del proceso de publicación del servicio por parte del nodo servidor.

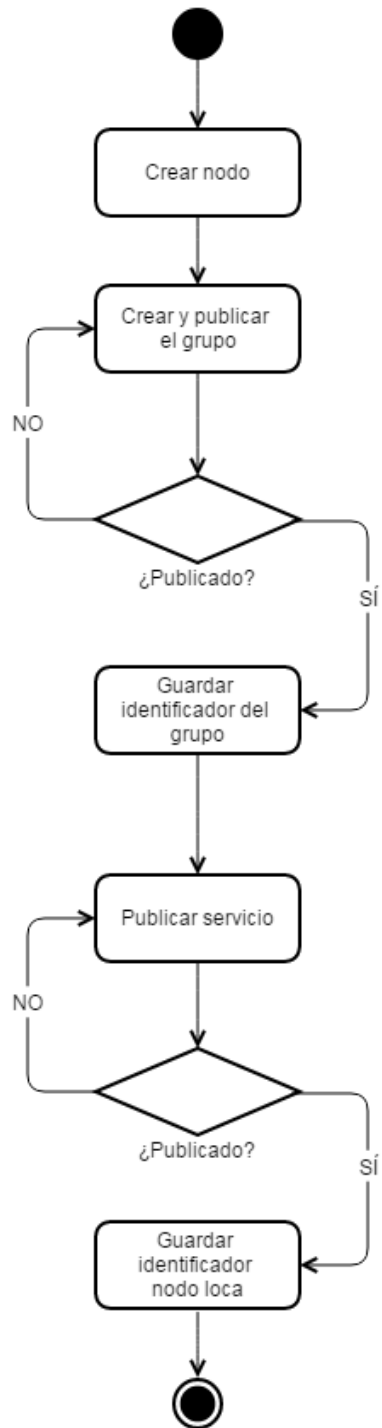


Figura 10. Diagrama de flujo publicación del servicio

Para describir más fácilmente el proceso de registro y publicación de un servicio dentro del sistema propuesto, se muestran los siguientes diagramas de secuencia,



los cuales muestran los pasos necesarios para completar el proceso satisfactoriamente.

La figura 11 y 12 muestra los pasos necesarios para crear un nodo teniendo en cuenta parámetros como tipo, nombre y caché (indexación local de avisos). Es vital la creación como tal de un nodo para poder participar en la red peer-to-peer. Lo anteriormente dicho define un nodo ya sea para propósitos de cliente o servidor.

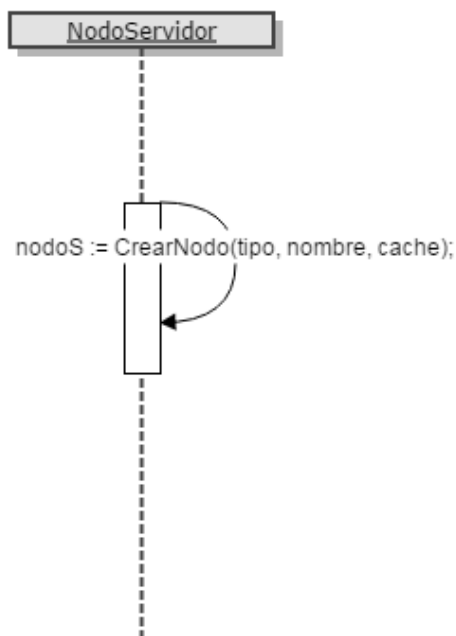


Figura 11. Diagrama de secuencia crear nodo servidor

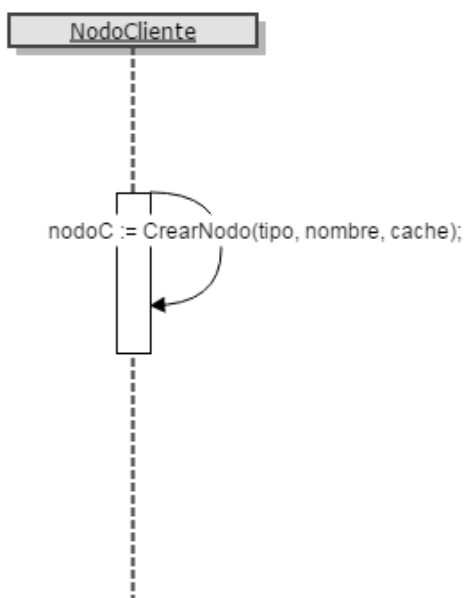


Figura 12. Diagrama de secuencia crear nodo cliente

Después de crear el nodo, en este caso un nodo para propósitos de servidor, en la red peer-to-peer cada nodo servidor debe crear un grupo específico del servicio que desea utilizar, si el grupo ya está creado, simplemente se une a él. La secuencia de pasos necesarios para crear el grupo se muestra en la figura 13 y 13. Es necesario guardar el identificador del grupo que se crea, por este motivo, luego de una publicación exitosa del grupo en la red superpuesta peer-to-peer, el nodo debe guardar el respectivo identificador para ser utilizado posteriormente.

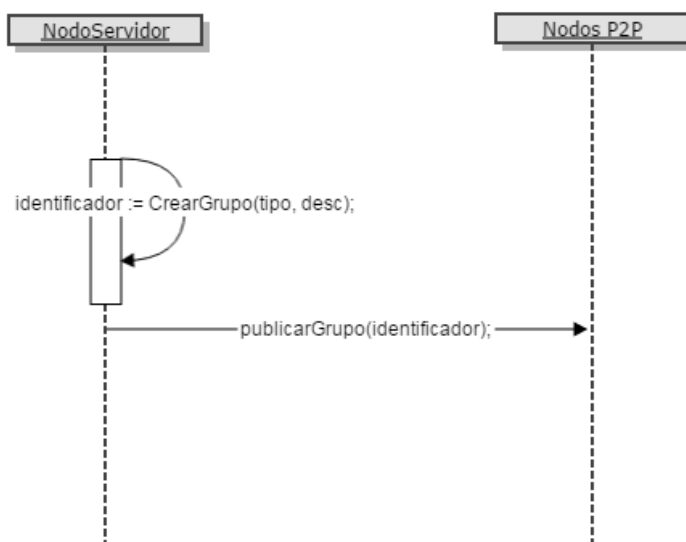


Figura 13. Diagrama de secuencia crear grupo nodo servidor

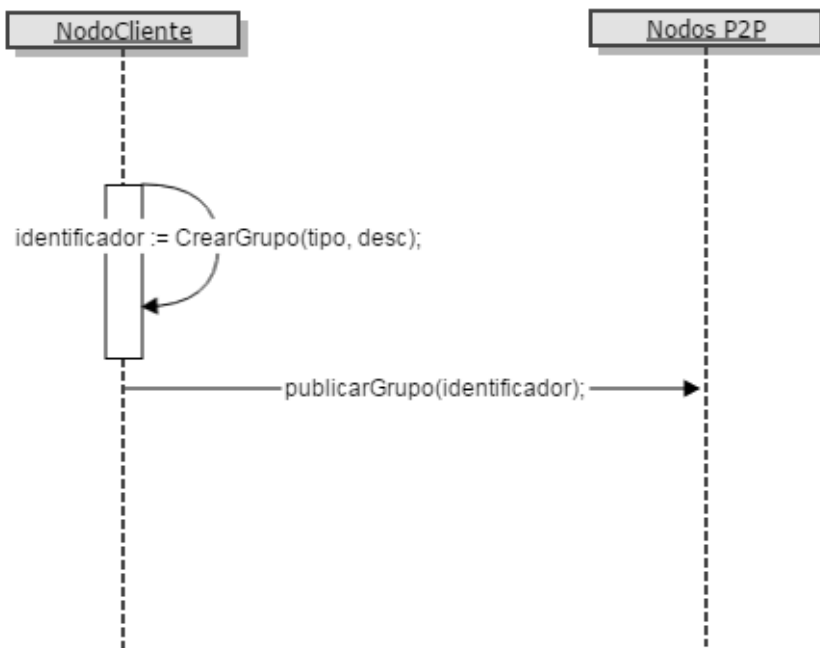


Figura 14. Diagrama de secuencia crear grupo nodo cliente

El siguiente paso consiste en la publicación del servicio en la red superpuesta peer-to-peer. Para este propósito se utiliza el grupo que se creó y guardó anteriormente. La Figura 15, muestra la secuencia de pasos necesarios para realizar la publicación del servicio. El servicio debe publicarse en la red superpuesta peer-to-peer, mediante el método *PublicarServicio(identificador)*. En la publicación del servicio se destaca la participación del Representante quien crea el mensaje descriptor que es integrado al aviso de publicación.

Después de la publicación del servicio en la red superpuesta peer-to-peer, el nodo servidor ya está en la capacidad de recibir conexiones desde los nodos cliente y prestar el servicio.

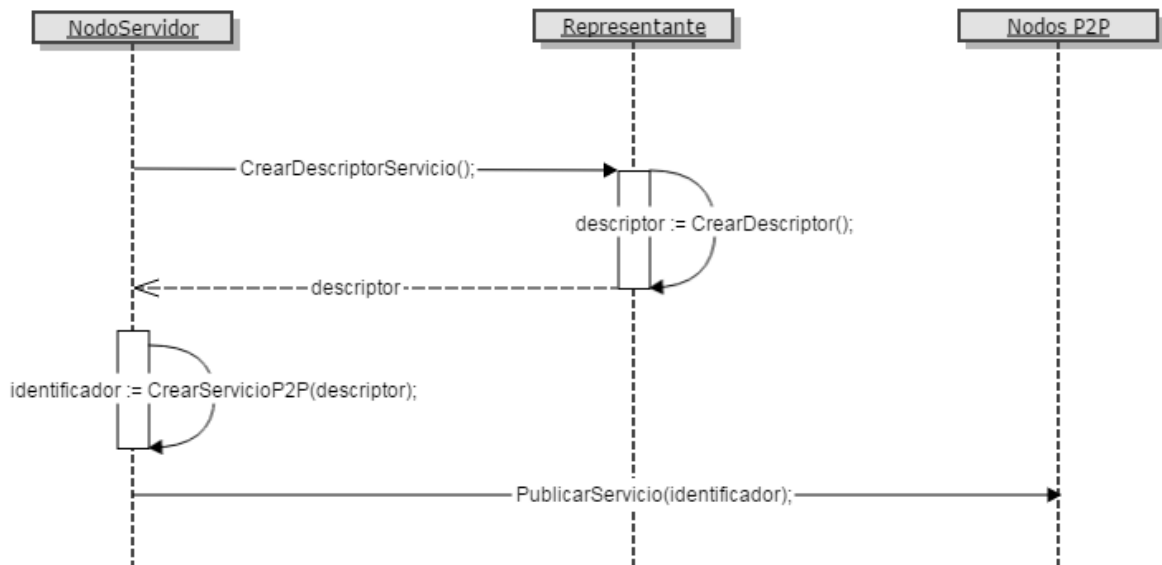


Figura 15. Diagrama de secuencia publicar servicio

### 3.4.2. Búsqueda y Consumo del Servicio

La búsqueda del servicio, es el proceso mediante el cual, un nodo cliente busca el servicio requerido en el sistema a través de los avisos de publicación. Una vez ubicado el servicio el nodo cliente puede proceder a consumir el servicio en el nodo servidor.

En la Figura 16, se muestra el diagrama de flujo del proceso de búsqueda.

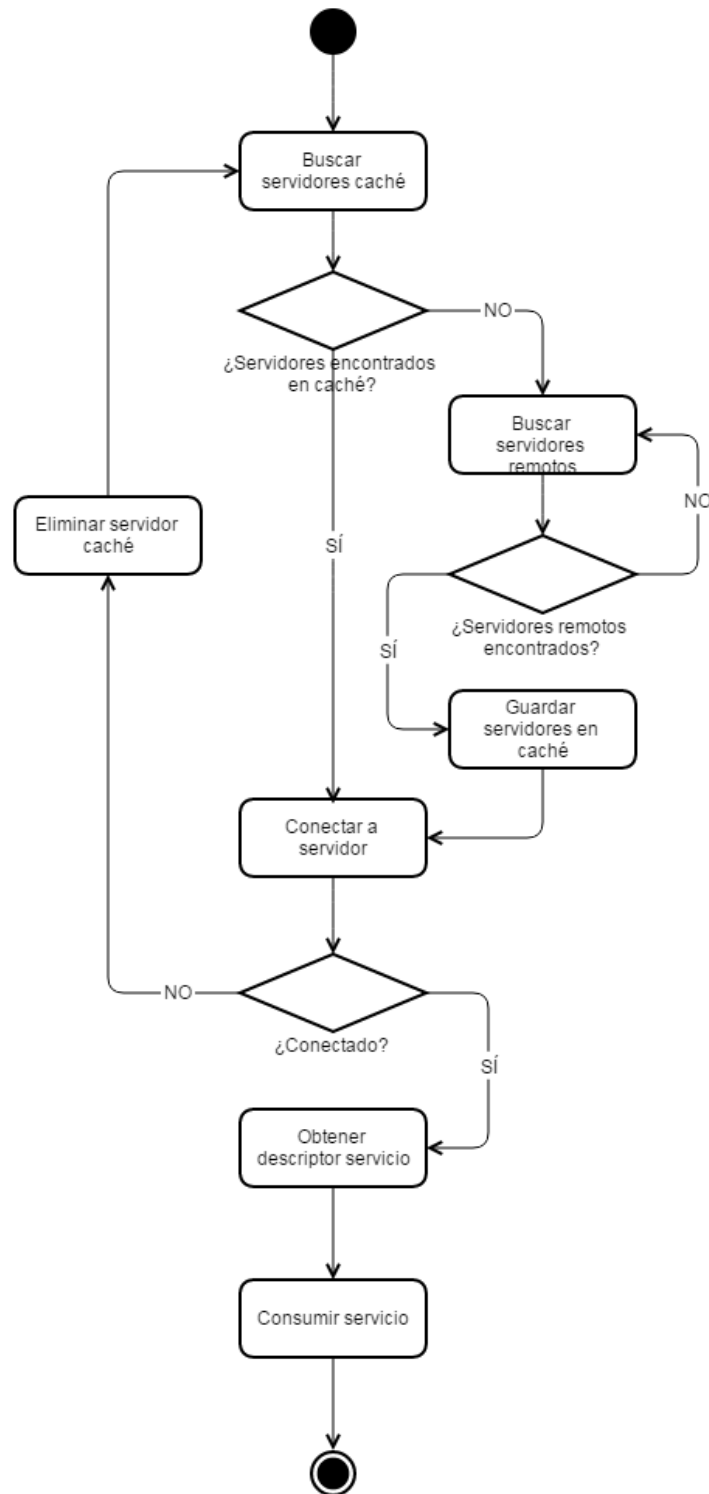


Figura 16. Diagrama de flujo búsqueda servicio

Cuando un servicio se publica en la red superpuesta peer-to-peer, un nodo cliente puede buscarlo y utilizarlo si lo necesita. La Figura 17, muestra la secuencia de pasos para buscar un nodo servidor que tenga a disposición un determinado servicio. El nodo cliente busca repetitivamente un nodo servidor usando un criterio (nombre del servicio) hasta encontrar uno. Para lograr lo anterior un nodo cliente primero realiza una búsqueda sobre su caché local y en caso de no tener aquí identificadores (avisos de publicación) de servidores disponibles realiza una búsqueda remota con el objetivo de encontrar nuevos servidores. Si encuentra uno o varios nodos servidor, el nodo cliente finalmente recibe mensajes identificadores, que representa cada uno de los nodos servidores encontrados y los guarda en su caché local en caso de necesitarlos en el futuro.

Es importante notar que la búsqueda del nodo servidor en la red superpuesta peer-to-peer se hace por medio de los protocolos de la implementación específica de la red superpuesta peer-to-peer que se esté usando. Sin embargo, es deseable que esta búsqueda se realice por un método más eficiente que el básico de inundación. El método de súper-nodos es el más deseable en este caso, debido a que es uno de los métodos que reduce considerablemente el número de mensajes y es uno de los más usados en las diferentes implementaciones de redes superpuestas peer-to-peer.

Se destaca nuevamente que el nodo cliente guarda localmente el identificador (aviso de publicación) de cada nodo servidor a manera de acceso directo, debido a que este puede dar una respuesta satisfactoria a las peticiones que se realicen posteriormente. El hecho de manejar un caché local en el nodo cliente facilita que cuando se realicen nuevas búsquedas se puedan encontrar servidores disponibles de manera mucho más rápida que cuando se hacen búsquedas remota en la red superpuesta peer-to-peer, lo cual sin duda también es costoso.

Si posteriormente existe un error en la conexión hacia un nodo servidor, el identificador que está guardado como un acceso directo en el caché local debe eliminarse y se debe volver a realizar el paso buscar servicio.

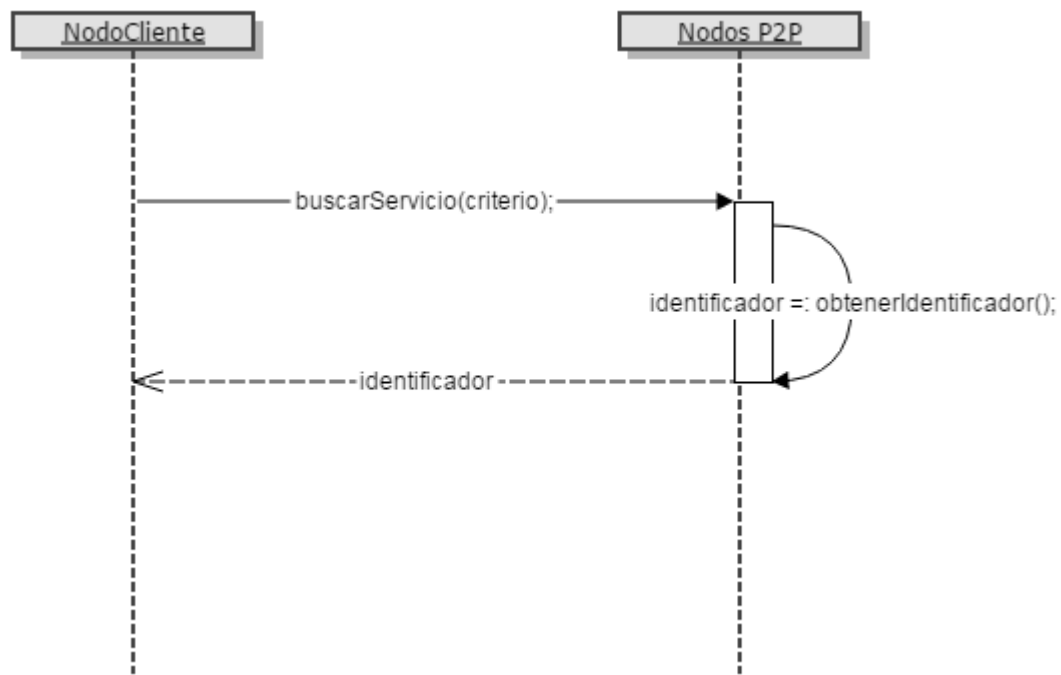


Figura 17. Diagrama secuencial búsqueda servicio

Cuando un nodo cliente encuentra un servicio ya sea a través de la búsqueda local (caché) o remota, el siguiente paso consiste en usar el identificador (que representa un nodo servidor) para poder realizar el consumo del servicio. En la Figura 18, se muestra la secuencia de pasos básicos a realizar. En el consumo del servicio se destaca la participación del Consumidor quien usa el mensaje descriptor del servicio para presentar una interfaz de consumo.

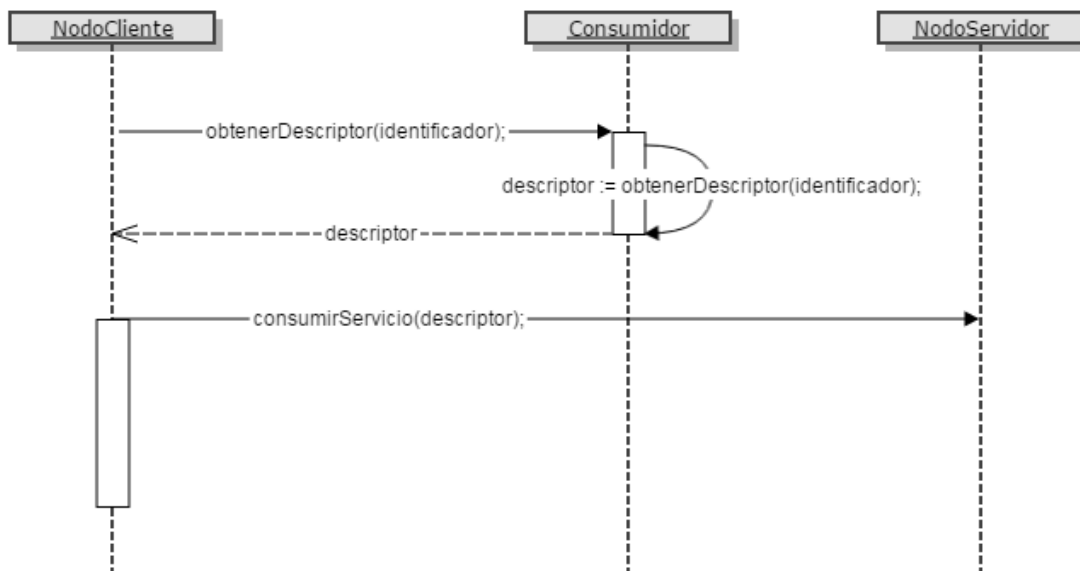


Figura 18. Diagrama de secuencia consumir servicio

### 3.4.3. Recuperación del Servicio

Una de las características de las redes superpuestas peer-to-peer es que los nodos dentro de la red pueden conectarse y desconectarse de la red de manera arbitraria.

En la Figura 20, se muestra el diagrama de flujo del proceso de recuperación. Si un nodo que está realizando la provisión del servicio se desconecta de la red, los nodos clientes que estaban utilizando ese servicio se verán afectados. Para solucionar este problema, el nodo cliente posee un mecanismo de recuperación a través de un Observador quien es el encargado de realizar el monitoreo de la condición del servicio y, en caso de desconexión del nodo servidor, disparará el mecanismo de recuperación para obtener otro servicio igual. De esta manera, el servicio es recuperado, mejorando la tolerancia a fallos a través del mecanismo de monitoreo y la características de redundancia apoyada por los nodo servidor que prestan un mismo servicio.

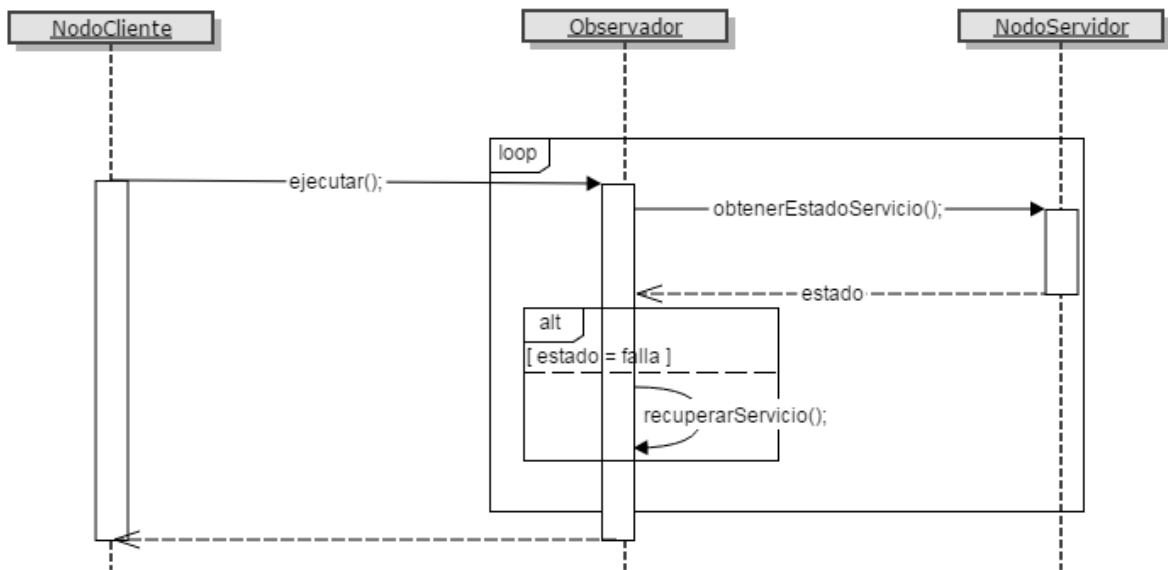


Figura 19. Diagrama de secuencia recuperar servicio



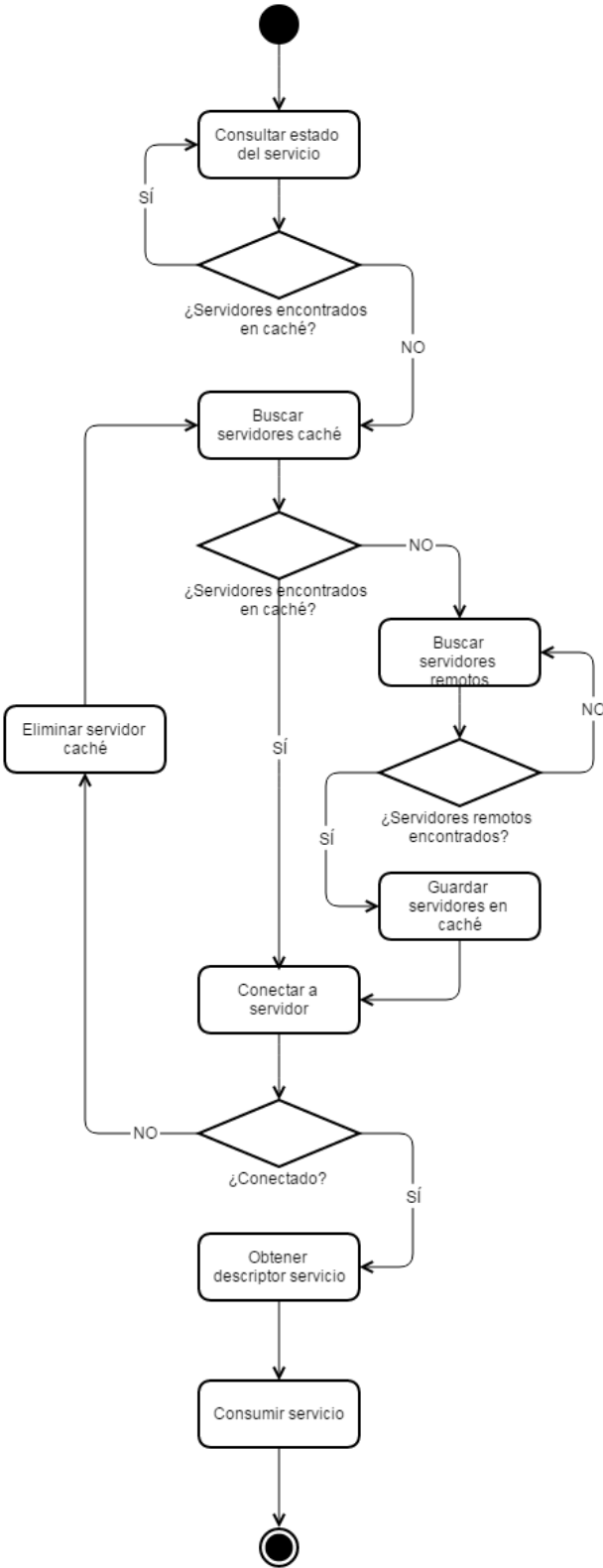


Figura 20. Diagrama de flujo recuperar servicio

## CAPÍTULO IV. TECNOLOGÍAS PARA LA IMPLEMENTACIÓN DE REDES SUPERPUESTAS PEER-TO-PEER Y EL DESPLIEGUE DEL SERVICIO DE MEDIA STREAMING

A continuación se describen las tecnologías seleccionadas para la implementación de la red superpuesta peer-to-peer y el despliegue del servicio de media streaming.

### 4.1. JXTA

JXTA refleja las operaciones entre dos nodos, estableciendo temporalmente una asociación para formar una red peer-to-peer. JXTA no es una filosofía de diseño de software y tampoco es una aplicación software. JXTA es un conjunto de protocolos que los desarrolladores de software pueden implementar usando su propia tecnología, y así establecer conexiones peer-to-peer entre nodos utilizando las mismas tecnologías o diferentes implementaciones de JXTA [60].

Por ejemplo, un grupo de desarrolladores pueden implementar JXTA en Visual Basic bajo Windows XP. Otro grupo podría hacer lo mismo en C++ en Linux, y un tercer grupo podría implementar la capa de JXTA en un dispositivo móvil en JAVA. Todos ellos serían capaces de encontrarse unos a otros en la red y comenzar a intercambiar cualquier tipo de información o servicios entre ellos, a pesar de la utilización de diferentes tecnologías subyacentes [60].

#### 4.1.1. ¿Por qué JXTA como tecnología utilizada?

Como se mencionó anteriormente, JXTA provee una plataforma que contiene y refleja las funciones básicas de las redes superpuestas peer-to-peer. JXTA supera los principales defectos potenciales de muchas plataformas peer-to-peer ya existentes: [26]

- **Interoperabilidad:** La tecnología JXTA, está diseñada para que un nodo pueda proveer servicios peer-to-peer, para localizar y comunicarse el uno con otro independientemente de direcciones de red y protocolos físicos.
- **Independiente de la Plataforma:** La tecnología JXTA, está diseñada para ser independiente de la tecnología o lenguaje de programación, protocolos de transporte de red y plataformas de despliegue.
- **Ubicuidad<sup>6</sup>:** La tecnología JXTA, está diseñada para ser accesible por cualquier dispositivo digital y en cualquier plataforma de despliegue.

---

<sup>6</sup> Ubicuidad significa que puede estar en todas partes.

#### 4.1.2. Arquitectura JXTA

La arquitectura de JXTA está dividida en tres capas lógicas que se describen a continuación: [60]



Figura 21. Arquitectura plataforma JXTA

- **Plataforma:** Esta capa es la base de JXTA y contiene la implementación de las funcionalidades mínimas y esenciales que son requeridas para la creación de una red peer-to-peer. Inicialmente, todos los nodos habilitados implementarán todas las funcionalidades de JXTA a pesar de no estar obligados a hacerlo. A esta capa se le conoce también como la capa núcleo.
- **Servicios:** Esta capa contiene los servicios adicionales que no son absolutamente necesarios para que un sistema peer-to-peer opere, pero que podrían ser útiles. Por ejemplo: intercambio de archivos, sistema de archivos distribuidos, etc. Estos servicios no son parte del conjunto de servicios definidos por JXTA.
- **Aplicación:** Las aplicaciones peer-to-peer son construidas en la parte superior de la capa de servicios. Sin embargo, si yo desarrollo una aplicación de intercambio de archivos y dejo que otras aplicaciones basadas en JXTA hagan peticiones a mi aplicación, las otras aplicaciones podrán ser percibidas como un servicio. Por lo tanto, la frontera entre un servicio y una aplicación depende de la perspectiva.

Uno de los puntos importantes de la arquitectura JXTA, es que la línea entre las capas de la arquitectura no es rígida. Una aplicación de un cliente puede considerarse como un servicio para otro cliente. Si se desarrolla un nodo que provee alguna funcionalidad, otro nodo puede usar parte de esa funcionalidad como un servicio que se ajuste a lo que él necesita y le puede agregar otros

módulos de software para complementarlo; pero un tercer nodo puede ver esa funcionalidad como una aplicación completa, sin necesidad de agregarle otros módulos. JXTA está diseñada para ser modular, permitiendo a los desarrolladores seleccionar y escoger una colección de servicios y aplicaciones de acuerdo a sus necesidades [26].

Existen cuatro aspectos esenciales de la arquitectura JXTA que la distinguen de otros modelos de red distribuida:

- El uso de documentos XML, llamados advertisements, para describir recursos de red.
- La abstracción de pipes a nodos, y de nodos a endpoints, sin depender de una autoridad central de nombramiento/direccionamiento, como es el caso de DNS.
- Un esquema de direccionamiento uniforme de nodos.
- Una infraestructura de búsqueda descentralizada basada sobre DHT, para indexación de recursos.

#### 4.1.3. Protocolos

JXTA define una serie de mensajes XML, también 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 .[26] Los protocolos JXTA son: Peer Discovery Protocol (PDP), Peer Information Protocol (PIP), Peer Resolver Protocol (PRP), Pipe Binding Protocol (PBP), Endpoint Routing Protocol (ERP), Rendezvous Protocol (RVP).[26]

#### 4.1.4. Servicios JXTA

Todos los protocolos estándar de JXTA son implementados como servicios, llamados servicios del núcleo o *core services* [26]. Los servicios JXTA son: Discovery Service, Peer Info Service, Resolver Service, Pipe Service, Endpoint Service, Rendezvous Service.[26]

Los protocolos y servicios de la plataforma JXTA, se describen más extensamente en el ANEXO C: tecnologías para la implementación de redes superpuestas peer-to-peer y el despliegue del servicio de media streaming.

## 4.2. VLC

VLC Media Player, inicialmente llamado VideoLAN Client, es un reproductor multimedia y framework multimedia del proyecto VideoLAN. Es un software libre distribuido bajo licencia GPL. Soporta múltiples códecs de audio y video, así como diferentes tipos de archivos y varios protocolos de streaming. También puede ser utilizado como servidor en unicast o multicast, en IPv4 o IPv6, en una red de banda ancha [61].

La distribución por defecto de VLC incluye un gran número de decodificadores libre y librerías para codificar, evitando la necesidad de buscar y/o calibrar plugins propietarios. Algunos de los códecs de VLC están provistos de la librería libavcodec del proyecto FFmpeg, pero usan principalmente sus propias implementaciones. Además, VLC es el primer reproductor que soporta playback de DVDs cifrados en Linux y OS X usando la librería de descifrado de DVDs llamada libdvdcss [61].

### 4.2.1. ¿Por qué VLC como sistema de media streaming?

La selección del VLC Media Player como sistema de media streaming se debe a que éste se encuentra en constante evolución y desarrollo por parte de investigadores y desarrolladores de todas partes del mundo. Además posee como núcleo una librería llamada libVLC, la cual provee a interfaces como el VLC, una gran cantidad de funcionalidades, tales como: el acceso a una publicación de tipo streaming, salidas de video y audio, uso de plugins, entre otros. Algunas de las características que facilitan el uso de este sistema se describen a continuación: [61]

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

#### 4.2.2. libVLC

libVLC es el motor principal y la interfaz del entorno multimedia en el que se basa el VLC Media Player [62]. 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 pligins, entre otras funciones. Posee subcomponentes como *interface*, *playlist*, *input*, *video\_output*, *audio\_output*, *streaming\_output*, en cargados de todas sus funcionalidades [63].

#### 4.3. VLCJ

VLCJ es un proyecto de código abierto que brinda la conexión y enlace por medio de Java para el reproductor VLC Media Player [64].

Ya que la librería libVLC, el motor principal del VLC Media Player, está escrita en C++, no puede ser usada directamente desde código JAVA, por esta razón el acceso y uso de la librería libVLC se hace por medio de VLCJ. La librería VLCJ implementa Java Native Access para tener acceso a código nativo (en este caso C++) directamente desde Java. Por lo tanto, este enlace a bajo nivel facilita la construcción de aplicaciones multimedia clientes o servidores usando Java [63].

La librería VLCJ además de proveer los enlaces por medio de Java a la funcionalidad nativa proporcionada por libVLC, también ofrece un entorno a un modelo de programación de alto nivel más simple que libVLC. El entorno de VLCJ brinda en tanto como sea posible, protección en caso del uso incorrecto de la librería nativa, evitando bloqueos o caídas fatales de aplicación [64].

VLCJ encapsula completamente el manejo asíncrono de eventos de reproducción de medios de forma transparente y son enviados de una forma serializada que es más familiar para Java. El manejo incorrecto de estos eventos también pueden causar bloqueos, el VLCJ sabe de ello y se encarga de esto, y proporciona un modelo de programación de tal manera que las aplicaciones no padecen de este problema [64].

Ya que el proyecto VLCJ está en constante evolución y mucho más maduro, proporcionando acceso a la mayoría de características brindadas por el libVLC, posibilidad de construcción de aplicaciones Java multimedia capaces de reproducir casi cualquier tipo de medios de comunicación; se ha escogido como tecnología para la implementación y puesta a prueba de la técnica de recuperación y ejecución de un servicio de media streaming [64].

## **CAPÍTULO V. IMPLEMENTACIÓN DE PROTOTIPO SOFTWARE**

En este capítulo se presenta el proceso de desarrollo del prototipo software, para la verificación de la técnica de búsqueda para la prestación de servicios, propuesta en el capítulo 3.

### **5.1. Especificación de Requerimientos**

Después de haber definido la técnica de recuperación y ejecución de servicios, se obtuvieron una serie de requerimientos para poder desarrollar el prototipo software, estos requerimientos se pueden ver con más detalle en el ANEXO B: Documento de Especificación de Requerimientos Software – SRS.

#### **5.1.1. Alcance**

El prototipo software que será desarrollado, pretende verificar la técnica de recuperación y ejecución de servicios propuesta utilizando un servicio de Media Streaming. Por lo tanto, la mayoría de la funcionalidad del prototipo software se ejecutará de manera automática, con reducida interacción con el usuario final del sistema. El prototipo software permitirá ejecutar la funcionalidad tanto de cliente (solicitante del servicio) como de servidor (prestador del servicio) en una sola interfaz de usuario.

El prototipo software integrará dos tecnologías para el desarrollo de redes peer-to-peer y el sistema de media streaming, la primera, JXTA, para el desarrollo de Redes peer-to-peer; y la segunda, VLC (VLCj), para el desarrollo de sistemas de Media Streaming.

#### **5.1.2. Funciones del Producto**

El prototipo software permitirá el despliegue del servicio de media streaming a través de la red peer-to-peer, permitiendo conexiones de tipo punto a multipunto en el nivel de aplicación. Teniendo como base una fuente principal de streaming (Servidor de Streaming) ajeno a la red peer-to-peer. Los nodos servidor que ingresen a la red peer-to-peer prestarán el servicio de retransmisión de la fuente principal. Por otra parte los nodos clientes que ingresen a la red peer-to-peer buscarán el servicio de streaming, y lo 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 el servicio de streaming 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.

### 5.1.3. Características del Usuario

La mayoría de la funcionalidad del prototipo software se ejecutará de manera automática, con reducida interacción con el usuario final del sistema. La única interacción del usuario final con el sistema, será la selección manual del tipo de funcionamiento del sistema, es decir el sistema puede comportarse como cliente, o como servidor.

### 5.1.4. Priorizar los Requisitos

- **Prioridad 1:** Por medio de la plataforma JXTA desarrollar la funcionalidad del sistema y de las comunicaciones para la red peer-to-peer del prototipo software e integrar la tecnología VLC a manera de servicio de media streaming en los nodos servidor.
- **Prioridad 2:** En el sistema peer-to-peer implementar las técnicas de búsqueda, recuperación y ejecución del servicio de media streaming.

### 5.1.5. Requerimientos Funcionales

Los requerimientos funcionales del prototipo software se describen a continuación:

- **Publicar El Servicio De Media Streaming (R-1):** El sistema debe crear y publicar el servicio de media streaming.
- **Buscar El Servicio De Media Streaming (R-2):** El sistema debe buscar el servicio de media streaming a través de la consulta sobre las publicaciones de servicio de los nodos servidor.
- **Monitorear Condición Servicio De Media Streaming (R-3):** El sistema debe permitir el monitoreo de la condición del servicio de media streaming que un nodo cliente se encuentra consumiendo. En caso de detectar desconexión o pérdida del servicio se debe disparar inmediatamente el mecanismo de la recuperación. Esta funcionalidad pertenece a un nodo cliente.



- **Recuperar El Servicio De Media Streaming (R-4):** El sistema debe recuperar el servicio de media streaming. Un nodo cliente del sistema realiza búsqueda en el cache local y/o búsqueda remota con el objetivo de encontrar nuevos servidores disponibles y recuperar el servicio.
- **Crear Mensaje Descriptor Para El Servicio De Media Streaming (R-5):** El sistema debe crear un mensaje descriptor del servicio de media streaming a integrar en la publicación del servicio. Esta funcionalidad pertenece a un nodo servidor.
- **Obtener Mensaje Descriptor Del Servicio De Media Streaming (R-6):** El sistema debe consultar una publicación de servicio de un nodo servidor para poder obtener el descriptor del servicio. Esta funcionalidad pertenece a un nodo cliente.
- **Consumir Servicio De Media Streaming (R-7):** El sistema debe permitirle a través del descriptor del servicio iniciar el consumo del servicio de media streaming que presta un nodo servidor. Esta funcionalidad pertenece a un nodo cliente.
- **Iniciar El Nodo Servidor Del Servicio De Media Streaming (R-8):** El sistema debe permitir al usuario iniciar el nodo servidor.
- **Detener El Nodo Servidor Del Servicio De Media Streaming (R-9):** El sistema debe permitir al usuario detener el nodo servidor.
- **Iniciar El Nodo Cliente Del Servicio De Media Streaming (R-10):** El sistema debe permitir al usuario iniciar el nodo cliente.
- **Detener El Nodo Cliente Del Servicio De Media Streaming (R-11):** El sistema debe permitir al usuario detener el nodo cliente.

#### 5.1.6. Requerimientos Funcionales:

- **Fiabilidad:** El prototipo software debe ser tolerante a fallos en especial a fallos de red o desconexión voluntaria o involuntaria por parte del usuario. Debe poder recuperarse de los diferentes errores que puedan presentarse. Es importante decir que los fallos por componentes hardware pueden estar presentes y no se tienen en cuenta en el desarrollo de este prototipo, así como también es importante decir que los fallos de software siempre están latentes en una aplicación y algunas veces pueden ser inesperados.
- **Disponibilidad:** Una de las características de las redes peer-to-peer es que los nodos pueden entrar y salir de esta arbitrariamente, por lo tanto el grado de disponibilidad de la aplicación depende de la disponibilidad del servicio en la red peer-to-peer.

- **Seguridad:** El prototipo software no tendrá en cuenta aspectos de seguridad, tales como nombre de usuarios o contraseñas, o sistemas de autenticación.
- **Consistencia:** La consistencia es un punto importante en las aplicaciones peer-to-peer, sin ella, el sistema no funcionará de una manera adecuada. Por lo tanto, la aplicación debe garantizar consistencia en el envío y recepción de mensajes, y en actualización de interfaz de usuario.
- **Mantenimiento:** El desarrollo del prototipo software no incluye mantenimiento. Sin embargo, si hay una mejora en las actualizaciones de las herramientas de desarrollo o las plataformas utilizadas, se incluirán esas nuevas actualizaciones, siempre y cuando el prototipo no se vea afectado de una manera negativa.
- **Portabilidad:** El prototipo software se podrá ejecutar en computadores personales o de escritorio, que tengan una conexión a Internet. El prototipo software deberá poder ejecutarse en cualquier sistema operativo a través de un Applet Viewer o un Navegador Web, que tenga instalada una Máquina Virtual Java. Se recomienda la versión 1.7.0\_21 del JDK o la v7 del JRE de Oracle.

## 5.2. Implementación

A continuación describe cómo se desarrolló la implementación del prototipo de software para la verificación de la técnica de recuperación y ejecución de servicios propuesta, teniendo en cuenta la metodología empleada, las herramientas utilizadas y los mecanismos de prueba.

### 5.2.1. Metodología de Desarrollo

Para el desarrollo del prototipo se utilizó la metodología Extreme Hacking [65] que representa un subconjunto de la metodología Programación Extrema [66]. Extreme Hacking ha sido adaptado para proyectos en donde no se cuenta con un cliente externo, condición que se cumple en el caso de este trabajo.

Extreme Hacking propone la utilización de algunas prácticas conocidas de programación extrema:

- Pruebas unitarias.
- Refactorización sin piedad.

- Principio "No lo vas a necesitar" (YAGNI).
- Metáfora del sistema.
- Lanzamientos frecuentes.
- Estándares de codificación.

## 5.2.2. Herramientas

### 5.2.2.1. Frameworks

- **Framework Peer-to-Peer:** Para la implementación de la red superpuesta peer-to-peer se utilizó JXTA.
- **Media Streaming:** Para la implementación del servicio de media streaming a desplegar en la red superpuesta peer-to-peer se utilizaron las librerías nativas de LibVLC (SDK Media Player VLC) y la librería de 'binding' VLCj.

### 5.2.2.2. Plataforma de Desarrollo

La aplicación fue desarrollada usando el lenguaje de programación Java con el JDK versión 1.7.0\_21. Java es conocido por ser un lenguaje estricto y el manejo de excepciones.

Las razones por las que este lenguaje de programación fue escogido para el desarrollo son:

- El Framework JXTA en su versión Java para el desarrollo de aplicaciones peer-to-peer fue escogido porque es la más avanzada respecto a las implementaciones en Lenguaje C (JXTA-C) y Lenguaje C++ (JXTA-C++).
- Experiencia en desarrollo de aplicaciones de Java.
- Amplia documentación, comunidades Java tanto del lenguaje como de JXTA y VLCj.

### 5.2.2.3. Entorno de Desarrollo Integrado

Para el desarrollo del prototipo software se utilizó el entorno de desarrollo integrado Netbeans en su versión 7. Este entorno fue escogido por las siguientes razones:

- Experiencia de uso.
- Usabilidad.
- Facilidad de integrar documentación de kits de desarrollo integrado (SDK) y librerías.
- Integración con GIT para el control de versiones.

### 5.2.3. Pruebas

Las pruebas que se realizaron sobre el prototipo se separaron en dos categorías:

- **Pruebas de software:** Para validar que el programa funcione como debería funcionar.
- **Pruebas de desempeño:** Como las descritas en el capítulo 6, sirven para determinar qué tan efectivas son los procedimientos y técnicas usadas dentro del prototipo.

Dentro de las pruebas de software se destacan:

- **Pruebas unitarias:** Pruebas automáticas escritas en el mismo lenguaje de la aplicación y pueden ejecutarse cuantas veces se desee. Este tipo de pruebas ayudan a verificar cada vez que un método o función sea modificado; Para este propósito se usó JUnit (librería para pruebas unitarias en Java). Las pruebas se encuentran en el directorio `\Aplicacion\Codigo Fuente\test` del CD entregado.
- **Pruebas manuales:** Para validar el software y verificar su desempeño, se realizaron varias pruebas manuales sobre nodos peer-to-peer en diferentes escenarios. En el transcurso del desarrollo del prototipo estas pruebas se realizaron de manera local en un mismo equipo de cómputo diferenciando los nodos a través de la definición de un puerto JXTA y también otro para el servicio media de streaming (puerto HTTP o RTSP). En el ANEXO H se encuentra detallado los escenarios definidos y el formato usado para las pruebas locales realizadas.

## CAPÍTULO VI. EXPERIMENTOS Y RESULTADOS

### 6.1. Definición de escenarios de prueba

Para la realización del plan de pruebas, se definieron una serie de escenarios posibles, para probar las características de recuperación de servicios en la técnica de ejecución y recuperación propuesta. En este caso el perfil del servicio media streaming, se definió en la sección 2.8.

Como se describió en el capítulo I, las redes superpuestas peer-to-peer no estructuradas, se forman cuando los nodos que la componen están enlazados de una manera arbitraria, y los mismos pueden entrar y salir de la red constantemente. Además de esto, cada nodo, en la red superpuesta peer-to-peer, puede comportarse como un cliente o un servidor.

Los escenarios definidos para la verificación de la técnica son los que se describen a continuación:

| Nro. de Escenario | Nodos Servidores | Nodos Clientes | Característica   |
|-------------------|------------------|----------------|--|
| 1                 | Uno              | Muchos         | Los nodos se conectarán a la red y los nodos clientes serán iniciados primero, luego se inicia el nodo servidor proveedor del servicio de streaming y que será consumido por los nodos cliente ya conectados.  |
| 2                 | Uno              | Muchos         | Los nodos se conectarán a la red y el nodo será servidor iniciado primero, luego se inician los nodos clientes que consumirán el servicio de streaming que provee el primer nodo servidor lanzado.   |
| 3                 | Dos              | Muchos         | Primero se inicia un nodo servidor que proveerá el servicio de streaming, luego se inician los nodos clientes que empezarán a consumir el servicio, para luego conectar y lanzar el segundo servidor. Por último se detiene el primer servidor y verificar que los nodos clientes recuperan el servicio conectándose al segundo nodo servidor. |
| 4                 | Dos              | Muchos         | Aplica la secuencia del escenario anterior, para luego iniciar de nuevo el primer  |

|   |        |        |   |
|---|--------|--------|---|
|   |        |        | servidor y detener el segundo servidor.   |
| 5 | Dos    | Muchos | Primero se inician todos los nodos servidores que proveerán el servicio de streaming; luego se inician los nodos clientes que empezarán a consumir el servicio.   |
| 6 | Dos    | Muchos | Primero se inician todos los nodos clientes; luego se inician los nodos servidores que empezarán a prestar el servicio de streaming. Una vez los clientes estén conectados al primer servidor consumiendo el servicio, se conecta y lanza el segundo nodo servidor.   |
| 7 | Tres   | Muchos | Primero se inician todos los nodos servidores que proveerá el servicio de streaming; luego se inician los nodos clientes que empezarán a consumir el servicio conectándose de forma aleatoria a los nodos servidores. Por último se detiene un nodo servidor aleatorio y verificar que los nodos clientes recuperan el servicio conectándose al cualquier nodo servidor que siga conectado.   |
| 8 | Tres   | Muchos | Primero se inician dos nodos servidores que proveerá el servicio de streaming; luego se inician todos los nodos clientes que empezarán a consumir el servicio conectándose de forma aleatoria a los nodos servidores que están conectados. Por último se conecta el tercer nodo servidor y se detienen los dos primeros iniciados anteriormente y verificar que los nodos clientes recuperan el servicio conectándose al último nodo servidor que se conectó. |
| 9 | Cuatro | Muchos | Primero se inician todos los nodos servidores que proveerá el servicio de streaming; luego se inician los nodos clientes que empezarán a consumir el servicio conectándose de forma aleatoria a los nodos servidores. Por último se detiene un nodo servidor aleatorio y verificar que los nodos clientes recuperan el servicio conectándose al cualquier nodo servidor que siga conectado.   |

|    |        |        |   |
|----|--------|--------|---|
| 10 | Cuatro | Muchos | Primero se inician tres nodos servidores que proveerá el servicio de streaming; luego se inician todos los nodos clientes que empezarán a consumir el servicio conectándose de forma aleatoria a los nodos servidores que están conectados. Por último se conecta el cuarto nodo servidor y se detienen los tres primeros iniciados anteriormente y verificar que los nodos clientes recuperan el servicio conectándose al último nodo servidor que se conectó.     |
| 11 | Cinco  | Muchos | Primero se inician todos los nodos servidores que proveerá el servicio de streaming; luego se inician los nodos clientes que empezarán a consumir el servicio conectándose de forma aleatoria a los nodos servidores. Por último se detiene un nodo servidor aleatorio y verificar que los nodos clientes recuperan el servicio conectándose al cualquier nodo servidor que siga conectado.   |
| 12 | Cinco  | Muchos | Primero se inician cuatro nodos servidores que proveerá el servicio de streaming; luego se inician todos los nodos clientes que empezarán a consumir el servicio conectándose de forma aleatoria a los nodos servidores que están conectados. Por último se conecta el quinto nodo servidor y se detienen los cuatro primeros iniciados anteriormente y verificar que los nodos clientes recuperan el servicio conectándose al último nodo servidor que se conectó. |
| 13 | Seis   | Muchos | Primero se inician todos los nodos servidores que proveerá el servicio de streaming; luego se inician los nodos clientes que empezarán a consumir el servicio conectándose de forma aleatoria a los nodos servidores. Por último se detiene un nodo servidor aleatorio y verificar que los nodos clientes recuperan el servicio conectándose al cualquier nodo servidor que siga conectado.   |

|    |      |        |  |
|----|------|--------|--|
| 14 | Seis | Muchos | Primero se inician cinco nodos servidores que proveerá el servicio de streaming; luego se inician todos los nodos clientes que empezarán a consumir el servicio conectándose de forma aleatoria a los nodos servidores que están conectados. Por último se conecta el sexto nodo servidor y se detienen los cinco primeros iniciados anteriormente y verificar que los nodos clientes recuperan el servicio conectándose al último nodo servidor que se conectó. |
|----|------|--------|--|

Tabla 3. Descripción de escenarios de prueba

En la tabla 4, se muestra el formato utilizado para registrar los resultados de los escenarios de prueba.

| Escenario:                       |     |    |                         |                        |                                     |
|----------------------------------|-----|----|-------------------------|------------------------|-------------------------------------|
| Equipo                           | Rol | IP | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
| <b>Descripción de resultado:</b> |     |    |                         |                        |                                     |

Tabla 4. Formato de registro de resultados

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

- **Escenario:** Número y descripción del escenario especificado anteriormente.
- **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 o un nodo servidor.
- **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:** Representa el tiempo en milisegundos que tarda el nodo cliente en recuperar el servicio y ejecutarlo nuevamente.



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

### 6.1.1 Descripción del marco experimental

Para la verificación de la técnica de recuperación y ejecución y el despliegue del prototipo, se realizó el marco experimental sobre una red LAN 10/100, conectada por medio de una topología física de red tipo BUS.

La sala de laboratorio, sobre la que está montada la topología de red física y en donde se realizaron las pruebas, está compuesta por 12 computadores, cada uno con diferentes características hardware y software. 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 ANEXO D: Especificación de los equipos utilizados en las pruebas.

### 6.2. Desarrollo 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. A continuación se muestran los resultados de cada uno de los escenarios de prueba:

### 6.2.1. Escenario de prueba Nro. 1

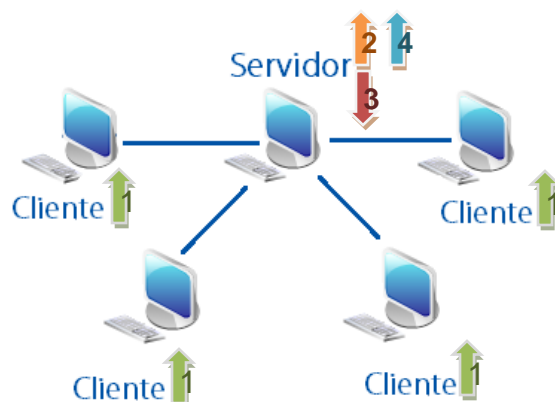


Figura 22. Representación del escenario de prueba No. 1

**Escenario Nro. 1:** Primero son iniciados todos los nodos cliente y seguido a eso se inicia el único servidor . Luego el servidor se detiene y se inicia nuevamente.

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Cliente  | 192.168.127.33 | 192.168.127.51          | 192.168.127.51         | 3030                                |
| 3      | Cliente  | 192.168.127.31 | 192.168.127.51          | 192.168.127.51         | 990                                 |
| 4      | Cliente  | 192.168.127.8  | 192.168.127.51          | 192.168.127.51         | 1050                                |
| 5      | Cliente  | 192.168.127.9  | 192.168.127.51          | 192.168.127.51         | 3020                                |
| 6      | Cliente  | 192.168.127.10 | 192.168.127.51          | 192.168.127.51         | 2995                                |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.51          | 192.168.127.51         | 2995                                |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.51          | 192.168.127.51         | 990                                 |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.51          | 192.168.127.51         | 3045                                |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.51          | 192.168.127.51         | 990                                 |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.51          | 192.168.127.51         | 3015                                |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.51          | 192.168.127.51         | 998                                 |

**Descripción de resultado:** Se puede observar que los tiempos de recuperación del servicio no son muy distantes el uno del otro y que todos los nodos cliente recuperaron el servicio.

Tabla 5. Tabla de resultados escenario de prueba Nro. 1

### 6.2.2. Escenario de prueba Nro. 2

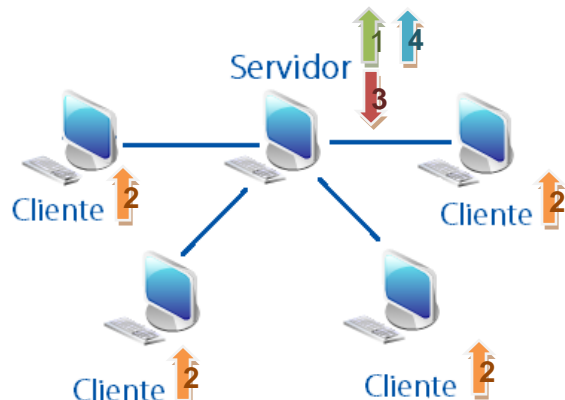


Figura 23. Representación del escenario de prueba No. 2

**Escenario Nro. 2:** Primero es iniciado nodo servidor y seguido a eso se inician los nodos clientes . Luego el servidor se detiene y se inicia nuevamente.

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Cliente  | 192.168.127.33 | 192.168.127.51          | 192.168.127.51         | 3010                                |
| 3      | Cliente  | 192.168.127.31 | 192.168.127.51          | 192.168.127.51         | 1050                                |
| 4      | Cliente  | 192.168.127.8  | 192.168.127.51          | 192.168.127.51         | 1080                                |
| 5      | Cliente  | 192.168.127.9  | 192.168.127.51          | 192.168.127.51         | 3040                                |
| 6      | Cliente  | 192.168.127.10 | 192.168.127.51          | 192.168.127.51         | 2995                                |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.51          | 192.168.127.51         | 1075                                |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.51          | 192.168.127.51         | 3020                                |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.51          | 192.168.127.51         | 3045                                |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.51          | 192.168.127.51         | 1025                                |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.51          | 192.168.127.51         | 3015                                |
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |

**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 todos los nodos cliente recuperaron el servicio luego de detener e iniciar de nuevo el servidor.

Tabla 6. Tabla de resultados escenario de prueba Nro. 2

### 6.2.3. Escenario de prueba Nro. 3

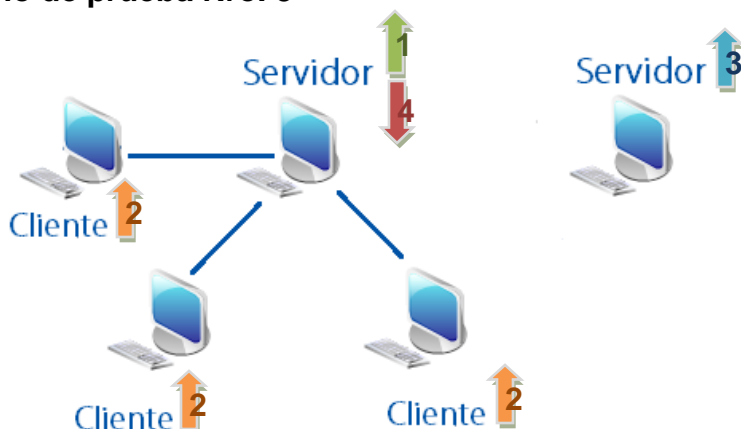


Figura 24. Representación del escenario de prueba No. 3

**Escenario Nro. 3:** Primero es iniciado el primer nodo servidor y seguido a eso se inician los nodos clientes . Luego el segundo servidor es iniciado y el primer servidor es detenido .

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Servidor | 192.168.127.33 | FUENTE                  | No aplica              | No aplica                           |
| 3      | Cliente  | 192.168.127.31 | 192.168.127.51          | 192.168.127.33         | 16470                               |
| 4      | Cliente  | 192.168.127.8  | 192.168.127.51          | 192.168.127.33         | 63451                               |
| 5      | Cliente  | 192.168.127.9  | 192.168.127.51          | 192.168.127.33         | 58845                               |
| 6      | Cliente  | 192.168.127.10 | 192.168.127.51          | 192.168.127.33         | 44685                               |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.51          | 192.168.127.33         | 65800                               |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.51          | 192.168.127.33         | 30595                               |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.51          | 192.168.127.33         | 37795                               |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.51          | 192.168.127.33         | 23585                               |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.51          | 192.168.127.33         | 44685                               |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.51          | 192.168.127.33         | 11547                               |

**Descripción de resultado:** Se observa que los tiempos de recuperación se aumentaron considerablemente, ya que los nodos clientes no tienen en su caché local el segundo servidor que se inició luego de la conexión de los clientes, sin embargo todos los nodos lograron conectarse al nuevo servidor y recuperar el servicio.

Tabla 7. Tabla de resultados escenario de prueba Nro. 3

#### 6.2.4. Escenario de prueba Nro. 4

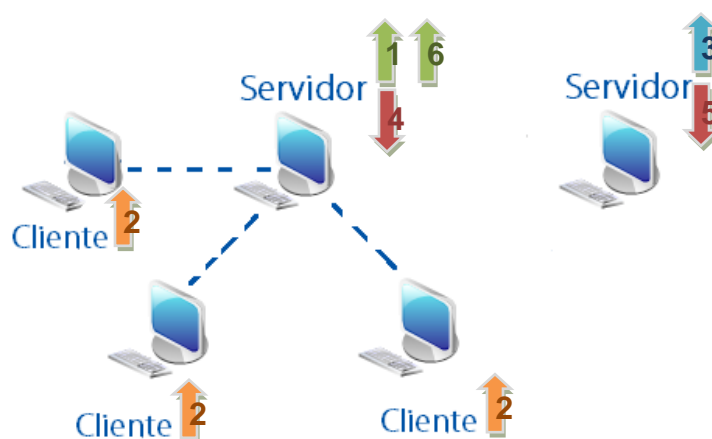


Figura 25. Representación del escenario de prueba No. 4

**Escenario Nro. 4:** Primero es iniciado el primer nodo servidor y seguido a eso se inician los nodos clientes . Luego el segundo servidor es iniciado y el primer servidor es detenido , seguido a eso se detiene al segundo servidor y se inicia de nuevo el primer servidor .

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Servidor | 192.168.127.33 | FUENTE                  | No aplica              | No aplica                           |
| 3      | Cliente  | 192.168.127.31 | 192.168.127.33          | 192.168.127.51         | 10350                               |
| 4      | Cliente  | 192.168.127.8  | 192.168.127.33          | 192.168.127.51         | 2435                                |
| 5      | Cliente  | 192.168.127.9  | 192.168.127.33          | 192.168.127.51         | 10385                               |
| 6      | Cliente  | 192.168.127.10 | 192.168.127.33          | 192.168.127.51         | 10385                               |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.33          | 192.168.127.51         | 10390                               |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.33          | 192.168.127.51         | 12390                               |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.33          | 192.168.127.51         | 10375                               |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.33          | 192.168.127.51         | 10365                               |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.33          | 192.168.127.51         | 12385                               |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.33          | 192.168.127.51         | 10344                               |

**Descripción de resultado:** Aquí el tiempo disminuye nuevamente en consideración al escenario anterior, ya que los nodos clientes ya tienen en su caché al primer servidor detenido, por lo tanto al recuperar de nuevo el servicio por segunda vez, el tiempo disminuye y todos los nodos clientes recuperan el servicio.

Tabla 8. Tabla de resultados escenario de prueba Nro. 4

### 6.2.5. Escenario de prueba Nro. 5

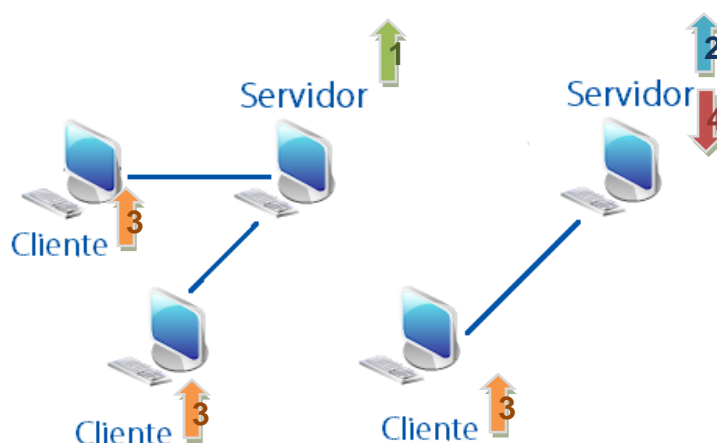


Figura 26. Representación del escenario de prueba No. 5

**Escenario Nro. 5:** Primero es iniciado el primer nodo servidor y luego el segundo nodo servidor es iniciado , seguido a eso los nodos clientes son iniciados y el segundo servidor es detenido .

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Servidor | 192.168.127.33 | FUENTE                  | No aplica              | No aplica                           |
| 3      | Cliente  | 192.168.127.31 | 192.168.127.33          |                        |                                     |
| 4      | Cliente  | 192.168.127.8  | 192.168.127.51          | 192.168.127.33         | 410                                 |
| 5      | Cliente  | 192.168.127.9  | 192.168.127.33          |                        |                                     |
| 6      | Cliente  | 192.168.127.10 | 192.168.127.33          |                        |                                     |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.33          |                        |                                     |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.51          | 192.168.127.33         | 355                                 |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.33          |                        |                                     |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.51          | 192.168.127.33         | 400                                 |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.51          | 192.168.127.33         | 480                                 |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.51          | 192.168.127.33         | 438                                 |

**Descripción de resultado:** Se observa que, cuando los nodos servidores son iniciados primero, el tiempo de recuperación en caso de detención de un servidor, es supremamente bajo ya que los nodos clientes ya conocen a los nodos servidores y no tardan en encontrar otro nodo servidor para consumir el servicio.

Tabla 9. Tabla de resultados escenario de prueba Nro. 5

### 6.2.6. Escenario de prueba Nro. 6

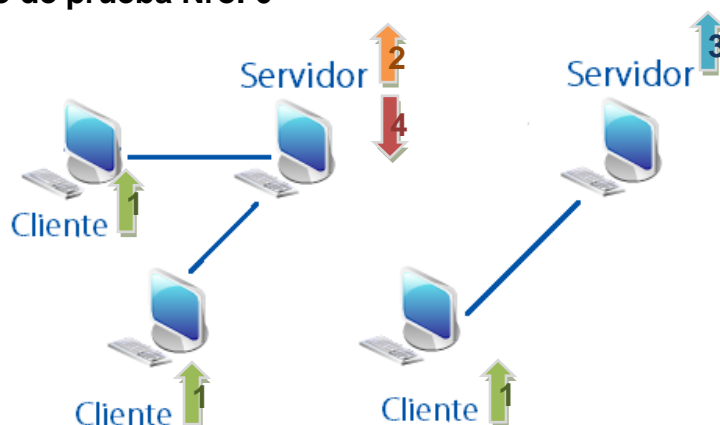


Figura 27. Representación del escenario de prueba No. 6

**Escenario Nro. 6:** Primero son iniciados todos los nodos cliente y seguido a eso se inicia el primer nodo servidor . Luego de que los nodos clientes hayan hecho conexión, se inicia el segundo nodo servidor y se detiene al primer nodo servidor .

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Servidor | 192.168.127.33 | FUENTE                  | No aplica              | No aplica                           |
| 3      | Cliente  | 192.168.127.31 | 192.168.127.33          | 192.168.127.51         | 32435                               |
| 4      | Cliente  | 192.168.127.8  | 192.168.127.33          | 192.168.127.51         | 465                                 |
| 5      | Cliente  | 192.168.127.9  | 192.168.127.33          | 192.168.127.51         | 425                                 |
| 6      | Cliente  | 192.168.127.10 | 192.168.127.33          | 192.168.127.51         | 10380                               |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.33          | 192.168.127.51         | 390                                 |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.33          | 192.168.127.51         | 10400                               |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.33          | 192.168.127.51         | 30420                               |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.33          | 192.168.127.51         | 10380                               |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.33          | 192.168.127.51         | 10390                               |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.33          | 192.168.127.51         | 46640                               |

**Descripción de resultado:** En este escenario se puede observar una variedad de tiempos de recuperación similares, a excepción de algunos nodos clientes, y en este caso al ser iniciados primero los clientes y luego los servidores, el tiempo promedio de recuperación del servicio es un poco más alto cuando el nodo servidor al que están conectados es detenido.

Tabla 10. Tabla de resultados escenario de prueba Nro. 6

### 6.2.7. Escenario de prueba Nro. 7

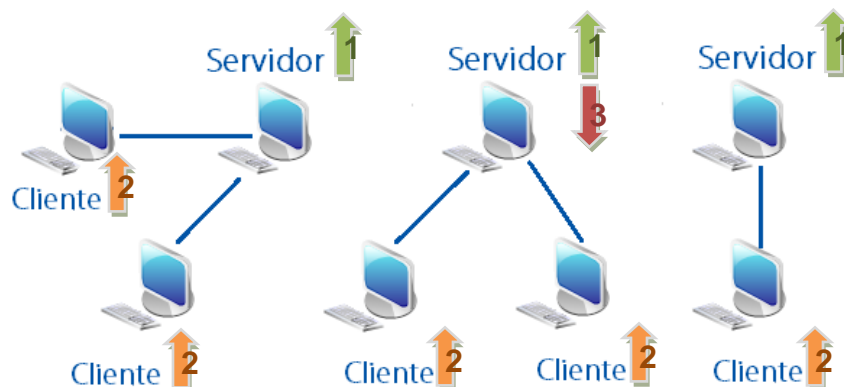


Figura 28. Representación del escenario de prueba No. 7

**Escenario Nro. 7:** Primero son iniciados los nodos servidores y seguido a eso se inician los nodos clientes . Luego se detiene un nodo servidor aleatorio con conexiones .

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Servidor | 192.168.127.33 | FUENTE                  | No aplica              | No aplica                           |
| 3      | Servidor | 192.168.127.31 | FUENTE                  | No aplica              | No aplica                           |
| 4      | Cliente  | 192.168.127.8  | 192.168.127.31          | 192.168.127.33         | 425                                 |
| 5      | Cliente  | 192.168.127.9  | 192.168.127.33          |                        |                                     |
| 6      | Cliente  | 192.168.127.10 | 192.168.127.31          | 192.168.127.33         | 410                                 |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.31          | 192.168.127.33         | 435                                 |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.31          | 192.168.127.51         | 420                                 |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.51          |                        |                                     |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.31          | 192.168.127.51         | 415                                 |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.51          |                        |                                     |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.51          |                        |                                     |

**Descripción de resultado:** Se observa que, cuando los nodos servidores son iniciados primero, el tiempo de recuperación en caso de detención de un servidor, es supremamente bajo ya que los nodos clientes ya conocen a los nodos servidores y están almacenados en su caché local y no tardan en encontrar otro nodo servidor para consumir el servicio en caso de falla de uno de los servidores cliente.

Tabla 11. Tabla de resultados escenario de prueba Nro. 7



### 6.2.8. Escenario de prueba Nro. 8

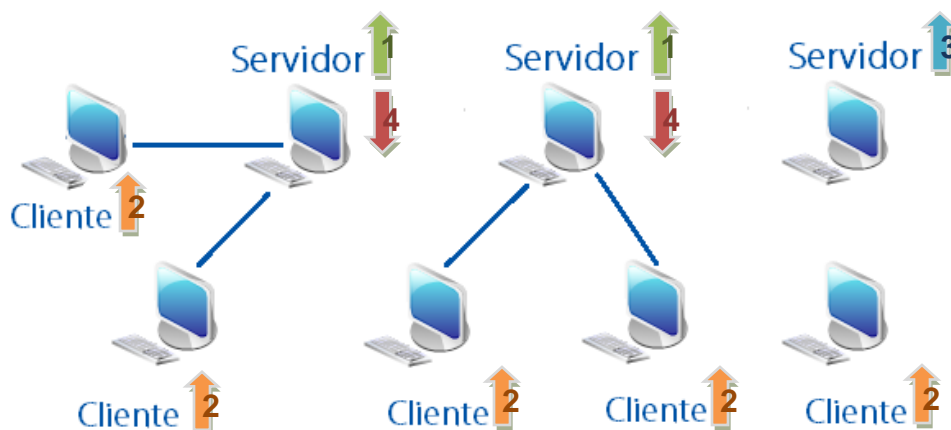


Figura 29. Representación del escenario de prueba No. 8

**Escenario Nro. 8:** Primero son iniciados los dos primeros nodos servidores y seguido a eso se inician los nodos clientes . Luego se inicia el último servidor. y se detiene a los primeros nodos servidores .

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Servidor | 192.168.127.33 | FUENTE                  | No aplica              | No aplica                           |
| 3      | Servidor | 192.168.127.31 | FUENTE                  | No aplica              | No aplica                           |
| 4      | Cliente  | 192.168.127.8  | 192.168.127.33          | 192.168.127.31         | 36435                               |
| 5      | Cliente  | 192.168.127.9  | 192.168.127.33          | 192.168.127.31         | 36415                               |
| 6      | Cliente  | 192.168.127.10 | 192.168.127.33          | 192.168.127.31         | 36336                               |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.33          | 192.168.127.31         | 46315                               |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.33          | 192.168.127.31         | 29435                               |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.33          | 192.168.127.31         | 46385                               |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.33          | 192.168.127.31         | 46395                               |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.33          | 192.168.127.31         | 46485                               |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.33          | 192.168.127.31         | 39719                               |

**Descripción de resultado:** De nuevo se observa que los tiempos de recuperación del servicio aumentan, ya que los nodos clientes al ser iniciados primero, no conocen al último servidor iniciado y primero buscarán recuperar el servicio con los nodos servidores que tienen en su caché local. Sin embargo se puede ver que todos los nodos cliente recuperaron el servicio.

Tabla 12. Tabla de resultados escenario de prueba Nro. 8

### 6.2.9. Escenario de prueba Nro. 9

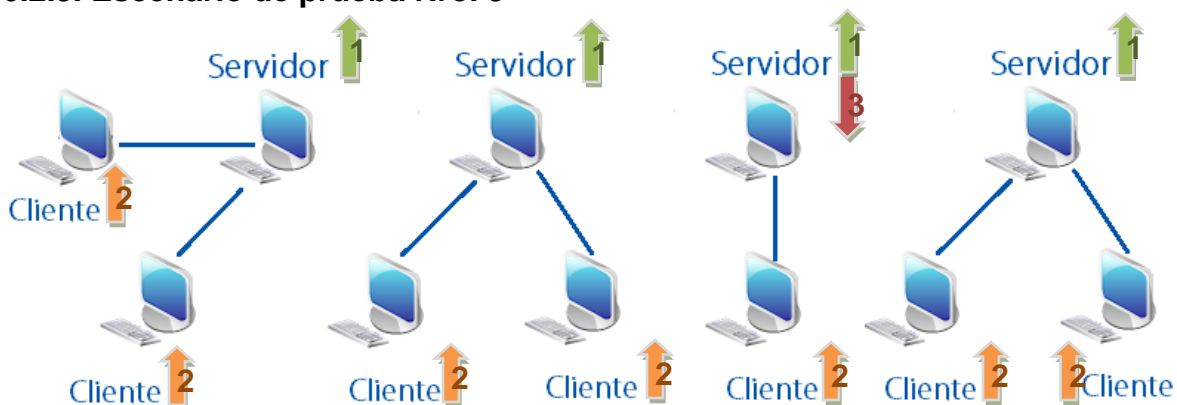


Figura 30. Representación del escenario de prueba No. 9

**Escenario Nro. 9:** Primero son iniciados los nodos servidores y seguido a eso se inician los nodos clientes . Luego se detiene un nodo servidor aleatorio con conexiones .

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Servidor | 192.168.127.33 | FUENTE                  | No aplica              | No aplica                           |
| 3      | Servidor | 192.168.127.31 | FUENTE                  | No aplica              | No aplica                           |
| 4      | Servidor | 192.168.127.8  | FUENTE                  | No aplica              | No aplica                           |
| 5      | Cliente  | 192.168.127.9  | 192.168.127.33          | 192.168.127.8          | 475                                 |
| 6      | Cliente  | 192.168.127.10 | 192.168.127.33          | 192.168.127.31         | 475                                 |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.31          |                        |                                     |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.33          | 192.168.127.51         | 440                                 |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.31          |                        |                                     |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.31          |                        |                                     |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.33          | 192.168.127.8          | 420                                 |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.33          | 192.168.127.31         | 516                                 |

**Descripción de resultado:** En este nuevo escenario se observa que, cuando los nodos servidores son iniciados primero, el tiempo de recuperación en caso de detención de un servidor aleatorio, es supremamente bajo ya que los nodos clientes ya conocen a los nodos servidores y están almacenados en su caché local y no tardan en encontrar otro nodo servidor para consumir el servicio en caso de falla de uno de los servidores cliente. Todos los nodos cliente recuperaron el servicio.

Tabla 13. Tabla de resultados escenario de prueba Nro. 9

### 6.2.10. Escenario de prueba Nro. 10

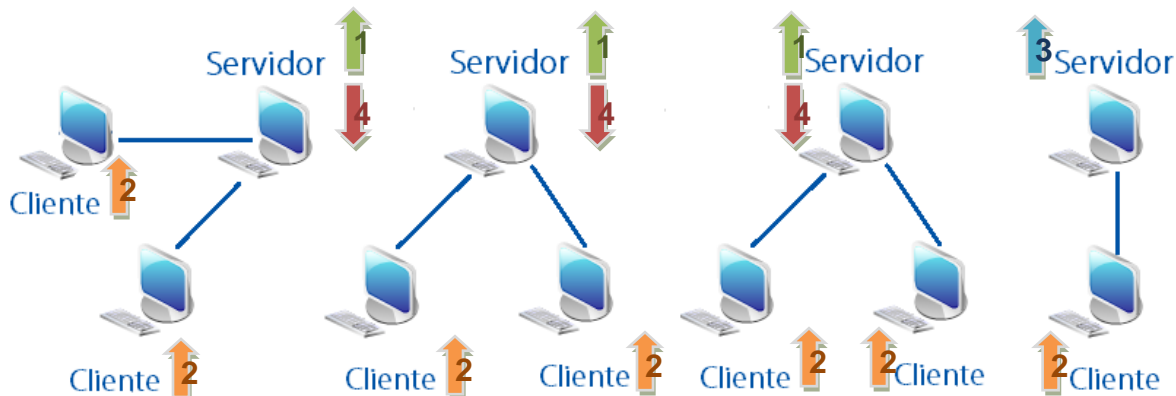


Figura 31. Representación del escenario de prueba No. 10

**Escenario Nro. 10:** Primero son iniciados los tres primeros nodos servidores y seguido a eso se inician los nodos clientes . Cuando los nodos clientes estén con una conexión se inicia el último servidor. y se detiene a los primeros nodos servidores .

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Servidor | 192.168.127.33 | FUENTE                  | No aplica              | No aplica                           |
| 3      | Servidor | 192.168.127.31 | FUENTE                  | No aplica              | No aplica                           |
| 4      | Servidor | 192.168.127.8  | FUENTE                  | No aplica              | No aplica                           |
| 5      | Cliente  | 192.168.127.9  | 192.168.127.51          | 192.168.127.8          | 72470                               |
| 6      | Cliente  | 192.168.127.10 | 192.168.127.33          | 192.168.127.8          | 106510                              |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.51          | 192.168.127.8          | 50450                               |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.31          | 192.168.127.8          | 47450                               |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.51          | 192.168.127.8          | 50445                               |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.33          | 192.168.127.8          | 47450                               |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.33          | 192.168.127.8          | 106500                              |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.31          | 192.168.127.8          | 99375                               |

**Descripción de resultado:** Aquí se nota que los tiempos de recuperación del servicio aumentan, ya que los nodos clientes al ser iniciados primero, se conectarán a los nodos servidores que inmediatamente se conecten, por lo tanto no conocen al último servidor iniciado y buscarán recuperar el servicio con los nodos servidores que tienen en su caché local. Sin embargo se puede ver que todos los nodos cliente recuperaron el servicio.

Tabla 14. Tabla de resultados escenario de prueba Nro. 10

### 6.2.11. Escenario de prueba Nro. 11

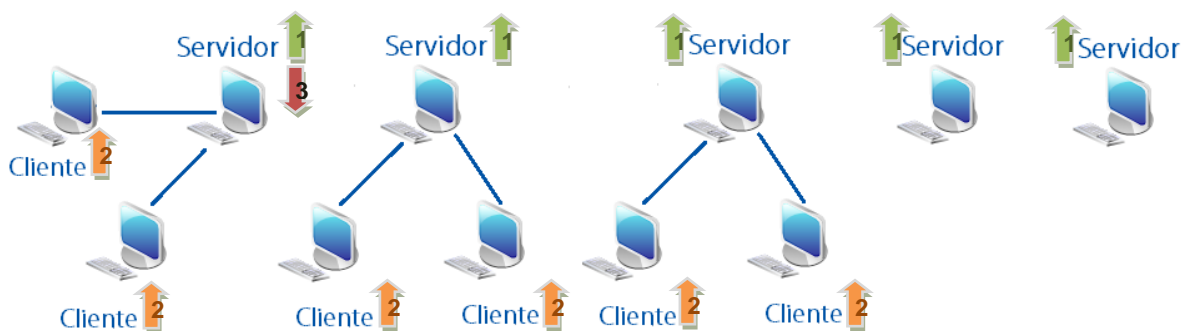


Figura 32. Representación del escenario de prueba No. 11

**Escenario Nro. 11:** Primero son iniciados los nodos servidores y seguido a eso se inician los nodos clientes . Luego se detiene un nodo servidor aleatorio con conexiones

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Servidor | 192.168.127.33 | FUENTE                  | No aplica              | No aplica                           |
| 3      | Servidor | 192.168.127.31 | FUENTE                  | No aplica              | No aplica                           |
| 4      | Servidor | 192.168.127.8  | FUENTE                  | No aplica              | No aplica                           |
| 5      | Servidor | 192.168.127.9  | FUENTE                  | No aplica              | No aplica                           |
| 6      | Cliente  | 192.168.127.10 | 192.168.127.51          | 192.168.127.9          | 430                                 |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.51          | 192.168.127.8          | 555                                 |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.51          | 192.168.127.33         | 425                                 |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.31          |                        |                                     |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.9           |                        |                                     |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.51          | 192.168.127.8          | 425                                 |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.8           |                        |                                     |

**Descripción de resultado:** Se observa que, cuando los nodos servidores son iniciados primero, el tiempo de recuperación en caso de detención de un servidor aleatorio, es supremamente bajo ya que los nodos clientes ya conocen a los nodos servidores y están almacenados en su caché local y no tardan en encontrar otro nodo servidor para consumir el servicio en caso de falla de uno de los servidores cliente. Todos los nodos cliente recuperaron el servicio.

Tabla 15. Tabla de resultados escenario de prueba Nro. 11

### 6.2.12. Escenario de prueba Nro. 12

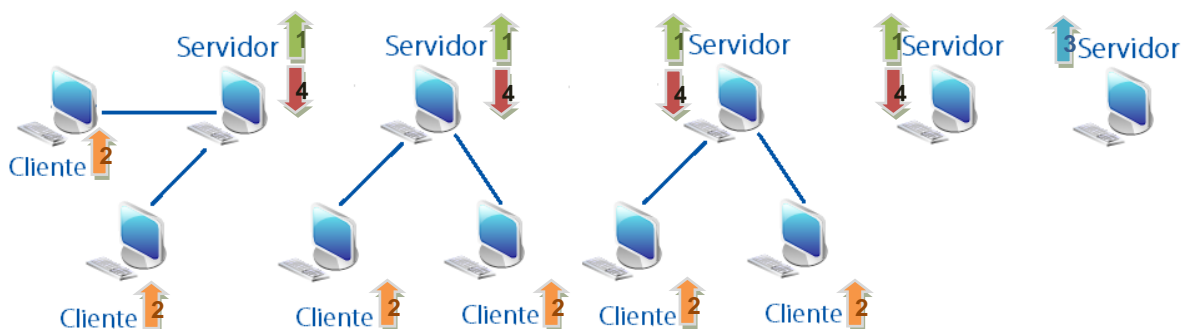


Figura 33. Representación del escenario de prueba No. 12

**Escenario Nro. 12:** Primero son iniciados los cuatro primeros nodos servidores y seguido a eso se inician los nodos clientes . Cuando los nodos clientes tengan una conexión se inicia el último servidor. y se detiene a los primeros nodos servidores .

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Servidor | 192.168.127.33 | FUENTE                  | No aplica              | No aplica                           |
| 3      | Servidor | 192.168.127.31 | FUENTE                  | No aplica              | No aplica                           |
| 4      | Servidor | 192.168.127.8  | FUENTE                  | No aplica              | No aplica                           |
| 5      | Servidor | 192.168.127.9  | FUENTE                  | No aplica              | No aplica                           |
| 6      | Cliente  | 192.168.127.10 | 192.168.127.51          | 192.168.127.8          | 112500                              |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.31          | 192.168.127.8          | 109460                              |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.51          | 192.168.127.8          | 112530                              |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.51          | 192.168.127.8          | 112480                              |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.51          | 192.168.127.8          | 112435                              |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.51          | 192.168.127.8          | 112435                              |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.33          | 192.168.127.8          | 47391                               |

**Descripción de resultado:** Aquí nuevamente se nota que los tiempos de recuperación del servicio aumentan considerablemente, ya que los nodos clientes al ser iniciados primero, se conectarán a los nodos servidores que inmediatamente se conecten, por lo tanto no conocen al último servidor iniciado y buscarán recuperar el servicio con los nodos servidores que tienen en su caché local. Sin embargo se puede ver que todos los nodos cliente recuperaron el servicio.

Tabla 16. Tabla de resultados escenario de prueba Nro. 12

### 6.2.13. Escenario de prueba Nro. 13

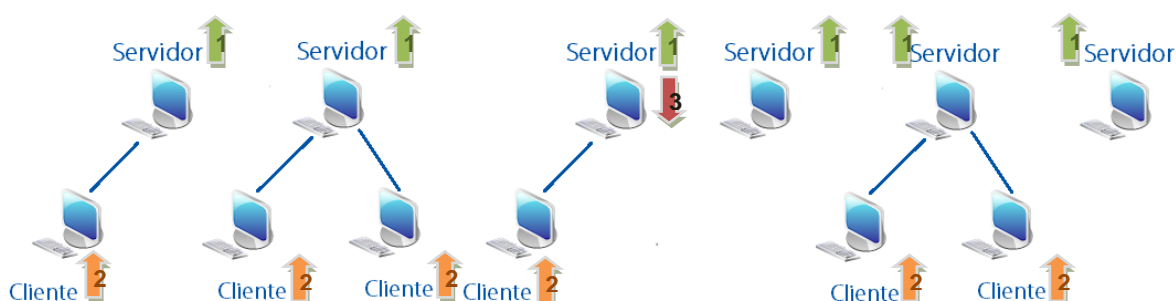


Figura 34. Representación del escenario de prueba No. 13

**Escenario Nro. 13:** Primero son iniciados los nodos servidores y seguido a eso se inician los nodos clientes . Luego se detiene un nodo servidor aleatorio con conexiones .

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Servidor | 192.168.127.33 | FUENTE                  | No aplica              | No aplica                           |
| 3      | Servidor | 192.168.127.31 | FUENTE                  | No aplica              | No aplica                           |
| 4      | Servidor | 192.168.127.8  | FUENTE                  | No aplica              | No aplica                           |
| 5      | Servidor | 192.168.127.9  | FUENTE                  | No aplica              | No aplica                           |
| 6      | Servidor | 192.168.127.10 | FUENTE                  | No aplica              | No aplica                           |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.33          |                        |                                     |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.51          |                        |                                     |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.31          |                        |                                     |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.8           | 192.168.127.51         | 540                                 |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.31          |                        |                                     |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.8           | 192.168.127.51         | 563                                 |

**Descripción de resultado:** En este escenario se observa que, cuando los nodos servidores son iniciados primero, el tiempo de recuperación en caso de detención de un servidor aleatorio, es supremamente bajo ya que los nodos clientes ya conocen a los nodos servidores y están almacenados en su caché local y no tardan en encontrar otro nodo servidor para consumir el servicio en caso de falla de uno de los servidores cliente. Todos los nodos cliente recuperaron el servicio.

Tabla 17. Tabla de resultados escenario de prueba Nro. 13

### 6.2.14. Escenario de prueba Nro. 14

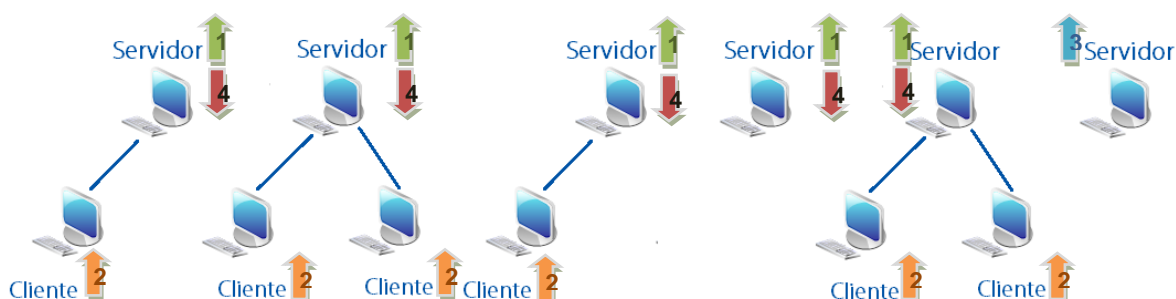


Figura 35. Representación del escenario de prueba No. 14

**Escenario Nro. 14:** Primero son iniciados los cinco primeros nodos servidores y seguido a eso se inician los nodos clientes . Cuando los nodos clientes tengan una conexión se inicia el último servidor. y se detiene a los primeros nodos servidores .

| Equipo | Rol      | IP             | Servidor Consumo Actual | Servidor Consumo Nuevo | Tiempo de Recuperación Milisegundos |
|--------|----------|----------------|-------------------------|------------------------|-------------------------------------|
| 0      | Fuente   | 192.168.127.25 | No aplica               | No aplica              | No aplica                           |
| 1      | Servidor | 192.168.127.51 | FUENTE                  | No aplica              | No aplica                           |
| 2      | Servidor | 192.168.127.33 | FUENTE                  | No aplica              | No aplica                           |
| 3      | Servidor | 192.168.127.31 | FUENTE                  | No aplica              | No aplica                           |
| 4      | Servidor | 192.168.127.8  | FUENTE                  | No aplica              | No aplica                           |
| 5      | Servidor | 192.168.127.9  | FUENTE                  | No aplica              | No aplica                           |
| 6      | Servidor | 192.168.127.10 | FUENTE                  | No aplica              | No aplica                           |
| 7      | Cliente  | 192.168.127.14 | 192.168.127.33          | 192.168.127.10         | 111230                              |
| 8      | Cliente  | 192.168.127.13 | 192.168.127.9           | 192.168.127.10         | 108460                              |
| 9      | Cliente  | 192.168.127.26 | 192.168.127.31          | 192.168.127.10         | 99333                               |
| 10     | Cliente  | 192.168.127.19 | 192.168.127.8           | 192.168.127.10         | 109460                              |
| 11     | Cliente  | 192.168.127.11 | 192.168.127.31          | 192.168.127.10         | 101064                              |
| 12     | Cliente  | 192.168.127.81 | 192.168.127.8           | 192.168.127.10         | 104326                              |

**Descripción de resultado:** Por último en este escenario se nota que los tiempos de recuperación del servicio aumentan con respecto al escenario anterior, ya que los nodos clientes al ser iniciados primero, se conectarán a los nodos servidores que inmediatamente se conecten, por lo tanto no conocen al último servidor iniciado y buscarán recuperar el servicio con los nodos servidores que tienen en su caché local. Sin embargo se puede ver que todos los nodos cliente recuperaron el servicio.

Tabla 18. Tabla de resultados escenario de prueba Nro. 14

### 6.3. Pruebas en características de desempeño, escalabilidad y precisión de búsqueda

Las pruebas, experimentos y resultados obtenidos y descritos en la sección anterior, confirman que la técnica de recuperación y ejecución de servicios, es efectiva debido al buen desempeño de las búsquedas: local (caché) y remota; de igual forma se evidencia que los nodos clientes siempre recuperan el servicio, siempre y cuando un nodo servidor esté conectado. Además el aumento de conexiones de nodos clientes y nodos servidores no afecta el rendimiento y demuestra claramente escalabilidad.

A continuación unas graficas en las cuales los resultados se ven claramente descritos:

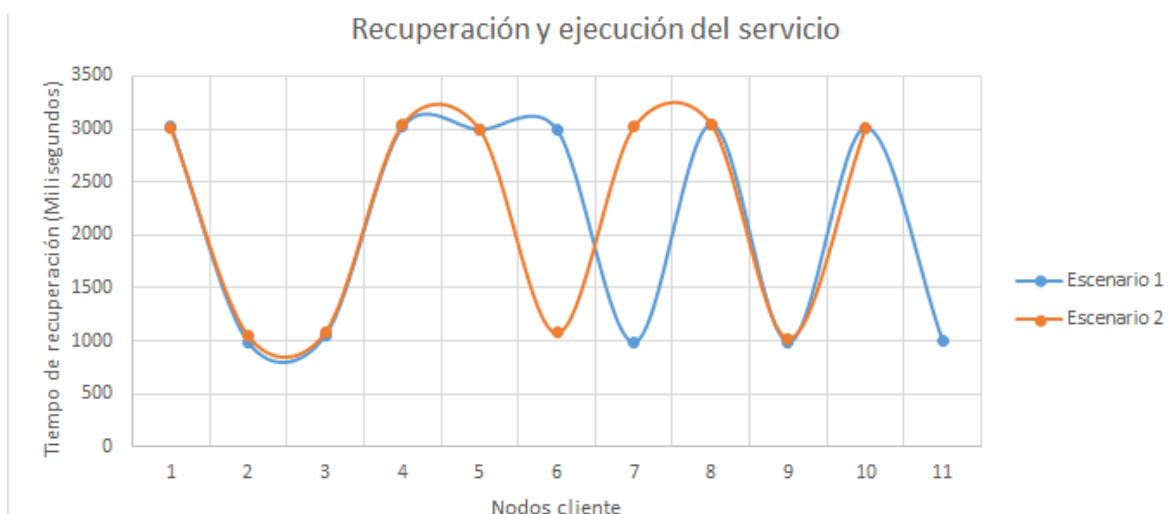


Figura 36. Gráfica recuperación y ejecución del servicio

En la figura 36 se puede observar que no hay una mayor diferencia entre los tiempos de recuperación del servicio cuando el único servidor se inicia primero, se detiene y se vuelve a iniciar, en este caso el escenario 1, y el segundo escenario donde los nodos clientes se inician primero.



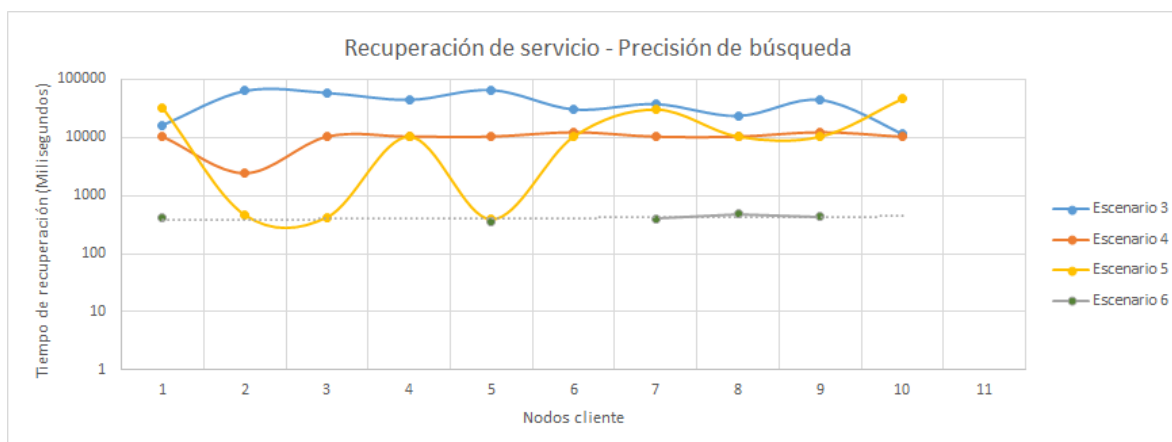


Figura 37. Gráfica precisión de búsqueda

En la figura 37 se aprecia claramente el tiempo de recuperación disminuye considerablemente, esto gracias también al caché que cada nodo cliente guarda localmente y que cada nodo siempre encuentra un nodo servidor para la recuperación del servicio y su nueva ejecución.

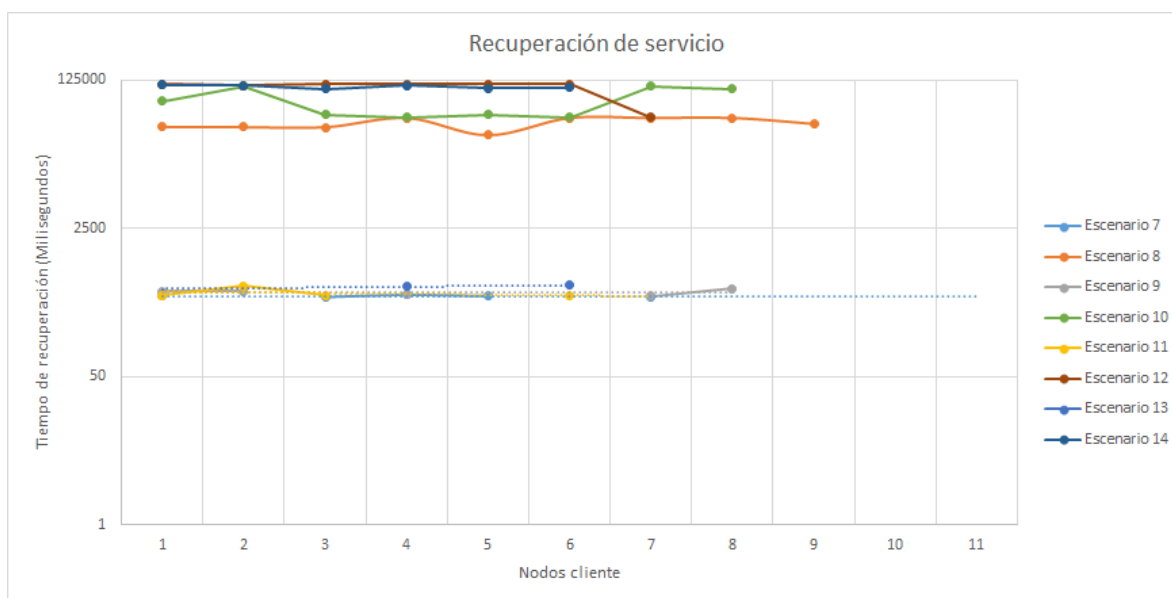


Figura 38. Gráfica recuperación de servicio

Por último se puede apreciar que a pesar de que los nodos servidores y nodos clientes siempre fueron aumentando o disminuyendo, el tiempo de recuperación se mantiene lineal en los escenarios que son similares en características y descripción. Es el caso de los escenarios 7, 9, 11, 13 y los escenarios 8, 10, 12, 14.

## CAPÍTULO VII. CONCLUSIONES Y TRABAJOS FUTUROS

### 7.1. Conclusiones.

- En este trabajo de investigación, se mencionó que las redes superpuestas peer-to-peer, tienen las capacidades para prestar una gran cantidad de servicios. Lamentablemente, se encuentran varios problemas cuando un nodo que está prestando un servicio deja arbitrariamente, lo que ocasiona una baja tolerancia a fallos. Por lo tanto, se necesitan mecanismos para la recuperación y ejecución de servicios en las redes superpuestas peer-to-peer.
- Se definieron y describieron los conceptos necesarios para conseguir un mejor entendimiento del problema. Se describieron las características de las redes superpuestas P2P, y la forma en que estas redes están clasificadas, centrándose en la definición y descripción de las redes superpuestas P2P no estructuradas, que son las de interés en este trabajo de grado.
- Se estudiaron las tecnologías de desarrollo, JXTA y VLCJ, para realizar la implementación de la red superpuesta P2P y el sistema de despliegue del servicio de media streaming en el prototipo software. JXTA, es una plataforma, diseñada para computación peer-to-peer, la cual contiene un conjunto de protocolos peer-to-peer, que permiten a cualquier dispositivo de red, comunicarse y colaborar mutuamente como nodos. VLCJ, es una librería de desarrollo implementada en Java que proporciona la conexión y enlaces necesarios para interactuar con el núcleo central de la aplicación VLC utilizada para el desarrollo del prototipo software.
- Las investigaciones realizadas sobre redes superpuestas peer-to-peer, JXTA y VLCJ, evidenciaron que pueden coexistir y formar una gran relación entre estos sistemas, lo que permite beneficiarse de su composición.
- Existe una gran variedad de métodos de búsqueda de servicios, en las redes superpuestas peer-to-peer no estructuradas. Cada uno de estos métodos, debe cumplir con un conjunto de características, que son deseadas, para la búsqueda de servicios sobre redes superpuestas peer-to-peer, las cuales se analizaron en detalle.
- Basándose en la investigación de recuperación de servicios y sus características, se analizaron los conceptos de redundancia y monitorización los cuales fueron esenciales para la construcción de la técnica de recuperación y ejecución de servicios.

- Se elaboró un prototipo de software basado en las plataformas de JXTA para la creación de la red peer-to-peer, VLC y VLCJ para la plataforma de media streaming. El prototipo verifica y pone a prueba la técnica de recuperación y ejecución del servicio de media streaming y fue liberado como software libre.
- Las pruebas y experimentos realizados con el prototipo desarrollado muestran las ventajas y desventajas de la técnica de recuperación y ejecución de servicios. Se pudo observar que un nodo cliente siempre recupera su servicio, en tanto haya un nodo servidor disponible. Por otro lado se observaron las desventajas en el aumento del tiempo de recuperación del servicio, ya que se depende de un caché local en el nodo cliente que no siempre tendrá todos los nodos servidores.
- Los resultados obtenidos, en la aplicación de las pruebas, para la verificación de la técnica de recuperación y ejecución de servicios, demostraron que la técnica realiza sus procesos de una forma correcta y se comporta de la manera esperada.

## 7.2. Trabajos futuros y recomendaciones

- Se propone la implementación en el nodo cliente de un mecanismo de monitoreo sobre el caché local (listados de nodos servidor) como bien se sabe es utilizado como primera opción por la técnica de recuperación cuando se realiza la búsqueda de nuevos servidores disponibles para poder recuperar el servicio. Este mecanismo implicaría realizar una constante verificación y actualización sobre el listado de servidores alternativos (caché local) según se encuentren disponibles. Aunque esto implicaría costos adicionales en la red ayudaría a mantener alta disponibilidad en la prestación del servicio.
- Se propone la implementación de nodos especializados o una entidad fija central para la indexación de nodos servidor alternativo que presten el mismo servicio. Un mecanismo de monitorización debería ser implementado sobre este tipo de entidades con el objeto de mantener listados de servidores disponibles actualizados. Estos nodos serían contactados por la técnica de recuperación para restablecer el servicio en los nodos cliente. Lo anterior también implicaría costos adicionales en la red, pero a diferencia de la propuesta anterior solo serían una o pocas entidades las responsables de ocasionar estos costos.
- Se propone extender la técnica de recuperación y ejecución de servicios considerando en el sistema un conjunto de fallos más amplio. Este

trabajo se enfocó solo en tipos de fallo de red que provocan desconexión en los nodos, sin embargo tipos de fallo a nivel de componentes software en el cliente y servidor deberían ser tenidos en cuenta.

- Se propone la implementación de un control sobre los nodos servidor que permita limitar el número de nodos cliente que consumen el servicio. Este número debería ser configurable o estar determinado por la cantidad de recursos a compartir por el nodo.
- Extender la definición y descripción de los servicios, mediante modelos, mecanismos y lenguajes de definición y ejecución de procesos de negocio, como, por ejemplo: BPEL. BPEL es un lenguaje para la ejecución de procesos de negocio, que permite definir una notación estándar, para especificar el comportamiento de un proceso basado en servicios.
- Se propone la implementación de un algoritmo que permita balancear el consumo del servicio, ya que en el presente trabajo la elección del nodo servidor por parte del cliente es aleatorio lo que puede causar que unos nodos servidor en ciertos momentos puedan estar sobrecargados.
- Se propone verificar la técnica de búsqueda propuesta, con un conjunto más amplio de servicios, lo que permitirá establecer con qué tipos de servicios esta propuesta logra mejores resultados teniendo en cuenta las características estudiadas y definidas para técnica pe. Desempeño.
- Se propone realizar la verificación de la técnica y las pruebas sobre una red más amplia, no solo en una intranet. Es decir, la red sobre la cual se probará la técnica de recuperación, tendrá todos los componentes propios de seguridad e infraestructura de las redes de hoy en día: Firewall, NAT's, Routers, etc.
- Extender y mejorar la técnica de búsqueda, para incluir descripción y búsqueda semántica de servicios. La descripción semántica de servicios busca la posibilidad de que las máquinas puedan realizar un mayor procesamiento y razonamiento sobre los servicios, soportando el descubrimiento e integración dinámica de los servicios.
- Extender la técnica de recuperación y ejecución de servicios a entornos de computación ubicua. La ubicuidad, está orientada a la disponibilidad de servicios, procesos e información en todo momento, en cualquier lugar e independientemente del dispositivo de acceso a dicha información "anytime, anywhere, any device". Gracias a que JXTA provee bloques de construcción básica y servicios requeridos, para permitir conectividad a cualquier aplicación donde quiera que está se encuentre, no debe ser difícil extender la técnica a este tipo de entornos.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] L. M. García Sanjuán, *Diseño e Implementación De Un Protocolo De Redes Peer-To-Peer*. 2009.
- [2] D. Doval, D. O. Mahony, and T. College, "Overlay Networks: A Scalable Alternative for P2P," *IEEE Internet Comput.*, 2003.
- [3] K. J. Turner, E. H. Magill, and D. J. Marples, *Service Provision: Technologies for Next Generation Communications*. Wiley, 2004.
- [4] R. Schollmeier, "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications," *Peer-to-Peer Comput. 2001. Proceedings. First Int. Conf.*, 2002.
- [5] D. Olmedilla and M. Palmer, "Interoperability for Peer-to-Peer Networks: Opening P2P to the Rest of the World," 2005.
- [6] A. Tsalgatidou and E. Koutrouli, "Interoperability and eServices," *Distrib. Appl. Interoper. Syst.*, 2005.
- [7] S. M. Lui and S. H. Kwok, "Interoperability of Peer-to-Peer File Sharing Protocols," *ACM SIGecom Exch.*, Jun. 2002.
- [8] L. Xu, S. Zhou, K. Zhao, W. Qian, and A. Zhou, "PeerBus : A Middleware Framework Towards Interoperability Among P2P Data Sharing Systems," *Grid Coop. Comput.*, 2004.
- [9] M. Lin, J. C. S. Lui, and D.-M. Chiu, "An ISP-Friendly File Distribution Protocol: Analysis, Design, and Implementation," *IEEE Trans. Parallel Distrib. Syst.*, Sep. 2010.
- [10] Z. Qin, F. Zhou, and L. Li, "Cooperative Approach between ISP and P2P Users to Reduce Inter-AS Traffic," *J. Electron. Sci. Technol.*, 2012.
- [11] F. Pianese, D. Perino, J. Keller, and E. W. Biersack, "PULSE An Adaptive Practical Live Streaming System," 2007.
- [12] R. Bindal, P. Cao, W. Chan, J. Medval, G. Suwala, T. Bates, and A. Zhang, "Improving Traffic Locality in BitTorrent via Biased Neighbor Selection," *Distrib. Comput. Syst.*, 2006.

- [13] Y. Liu, L. Guo, F. Li, and S. Chen, "A Case Study of Traffic Locality in Internet P2P Live Streaming Systems," *2009 29th IEEE Int. Conf. Distrib. Comput. Syst.*, Jun. 2009.
- [14] R. Mondéjar, P. García, and C. Pairet, "Bunshin: DHT para Aplicaciones Distribuidas," *J. Parallel Distrib. Comput.*, 2005.
- [15] F. Forster and H. De Meer, "Discovery of Web Services With a P2P Network," *Comput. Sci. 2004*, 2004.
- [16] W. Bradley and D. Maher, "The NEMO P2P Service Orchestration Framework," *Proc. 37th Annu. Hawaii Int. Conf. Syst. Sci.*, 2004.
- [17] C. Pérez-Miguel, M.-A. Jose, and A. Mendiburu, "Informe sobre Sistemas de Computación en Redes P2P," 2009.
- [18] R. Bhagwan, S. Savage, and G. Voelker, "Understanding Availability," *Peer-to-Peer Syst. II*, 2003.
- [19] R. Ahmed, N. Shahriar, and M. Sharmin, "Persistence Service for Non-Persistent P2P Systems," *CS-2012-18*, 2012.
- [20] J. Kangasharju, K. W. Ross, and D. a. Turner, "Optimizing File Availability in Peer-to-Peer Content Distribution," *IEEE INFOCOM 2007 - 26th IEEE Int. Conf. Comput. Commun.*, 2007.
- [21] D. Figueiredo, J. Shapiro, and D. Towsley, "Incentives to Promote Availability in Peer-to-Peer Anonymity Systems," *13TH IEEE Int. Conf. Netw. Protoc.*, 2005.
- [22] O. Riera, M. García López, and P. Sanchez Artigas, *A Hierarchical Framework for Peer-to-Peer Systems: Design and Optimizations*. Universitat Pompeu Fabra, 2008.
- [23] O. A. Gallardo Perez, "Especificación de un framework para la construcción de un sistema distribuido peer-to-peer (P2P)," 2008.
- [24] RAE, "Definición RAE: Técnica." [Online]. Available: <http://lema.rae.es/drae/srv/search?id=b696NokCiDXX2S50NiNV|kq4NayGwcDXX2ddIQcYi>. [Accessed: 23-Jun-2013].
- [25] RAE, "Definición RAE: Recuperación." [Online]. Available: <http://lema.rae.es/drae/?val=recuperar>. [Accessed: 23-Jun-2013].

- [26] A. Andrade, D. Luna, and P. Magé, "Técnica de Búsqueda para la Prestación de Servicios, basada en Sistemas Multi-Agente y Redes P2P," *Sist. Telemática*, pp. 65–75, 2012.
- [27] S. Androutsellis-Theotokis and D. Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies," *ACM Comput. Surv.*, Dec. 2004.
- [28] N. Ramzan, H. Park, and E. Izquierdo, "Video Streaming Over P2P Networks: Challenges and Opportunities," *Signal Process. Image Commun.*, May 2012.
- [29] G. Wen, H. Longshe, and F. Qiang, "Recent Advances in Peer-to-Peer Media Streaming Systems," *Featur. Artic. Multimed. Commun.*, 2006.
- [30] M. A. Cerón and P. A. Magé, "Pixtream : Sistema de Streaming P2P," *IX Jornadas Ing. Telemática (JITEL 2010), Valladolid, España, 29 septiembre 2010.*, 2010.
- [31] M. Ripeanu, "Peer-to-Peer Architecture Case Study: Gnutella Network," *Peer-to-Peer Comput. 2001. Proceedings. First Int. Conf.*, 2001.
- [32] J. Pouwelse, P. Garbacki, D. Epema, and H. Sips, "The Bittorrent P2P File-Sharing System: Measurements and Analysis," *4th Int. Work. Peer-to-Peer Syst. IPTPS*, 2005.
- [33] L. C. Martín, "Skype Que Lugar Quedan Operadoras Telefonía P2P," *Bit*, 2005.
- [34] "SETI@home." [Online]. Available: <http://setiathome.berkeley.edu/>. [Accessed: 23-Jun-2013].
- [35] "Genome@home." [Online]. Available: <http://genomeathome.stanford.edu/>. [Accessed: 23-Jun-2013].
- [36] R. Huebsch, *PIER: Internet Scale P2P Query Processing with Distributed Hash Tables*. 2008.
- [37] A. Halevy, Z. Ives, and J. Madhavan, "The Piazza Peer Data Management System," *Knowl. Data Eng. IEEE Trans.*, 2004.
- [38] "Edutella." [Online]. Available: <http://www.edutella.org/edutella.shtml>. [Accessed: 23-Jun-2013].

- [39] D. Oppenheimer and D. a. Patterson, "Studying and using failure data from large-scale internet services," *Proc. 10th Work. ACM SIGOPS Eur. Work. beyond PC - EW10*, p. 255, 2002.
- [40] P. N. Ayuso, "Tolerancia A Fallos Para La Alta Disponibilidad De Software De Red De Alto Rendimiento Con Estados," 2008.
- [41] A. Avizienis, "Toward systematic design of fault-tolerant systems," *Computer (Long. Beach. Calif.)*, vol. 30, no. 4, pp. 51–58, Apr. 1997.
- [42] R. Guerraoui and A. Schiper, "Software-based replication for fault tolerance," *Computer (Long. Beach. Calif.)*, vol. 30, no. 4, pp. 68–74, Apr. 1997.
- [43] S. M. Manish Marwah, "TCP server fault tolerance using connection migration to a backup server," *2003, IEEE Intl, San Francisco*, pp. 373–382, 2003.
- [44] G. Robinson, A. Narin, and C. Elleman, "Using Amazon Web Services for Disaster Recovery," no. October, 2014.
- [45] G. T. Santos and C. Montez, "FTWeb: A Fault Tolerant Infrastructure for Web Services," in *Ninth IEEE International EDOC Enterprise Computing Conference (EDOC'05)*, pp. 95–105.
- [46] D. Zagorodnov, K. Marzullo, L. Alvisi, and T. C. Bressoud, "Engineering fault-tolerant TCP/IP servers using FT-TCP," in *2003 International Conference on Dependable Systems and Networks, 2003. Proceedings.*, 2003, pp. 393–402.
- [47] R. Jhawar and V. Piuri, "Fault Tolerance and Resilience in Cloud Computing Environments Fault Tolerance and Resilience in Cloud Computing Environments," vol. 2, 2013.
- [48] H. Miyajima, "Fault tolerant server," *US Pat. App. 14/204,567*, pp. 1–15, 2014.
- [49] I. Meedeniya, A. Aleti, and B. Buhnova, "Redundancy allocation in automotive systems using multi-objective optimisation," *Symp. Avion. ...*, pp. 1–16, 2009.
- [50] R. Godoi, "P2P-VoD on Internet: Fault Tolerance and Control Architecture," 2009.



- [51] T. T. Do, K. A. Hua, and M. A. Tantaoui, "P2VoD: providing fault tolerant video-on-demand streaming in peer-to-peer environment," in *2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577)*, 2004, vol. 3, pp. 1467–1472 Vol.3.
- [52] F. R. Mason, "G2-P2P: A Fully Decentralised Fault-Tolerant Cycle-Stealing."
- [53] J. R. Ortiz and R. M. Jiménez, "Fault-tolerant distributed discrete event simulator based on a P2P architecture," *SIMUL 2011, Third ...*, pp. 21–26, 2011.
- [54] D. Tanvir Ahmed and S. Shirmohammadi, "A fault tolerance procedure for P2P online games," in *10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)*, 2010, pp. 614–617.
- [55] X. Yang, M. Gjoka, P. Chhabra, A. Markopoulou, and P. Rodriguez, "Kangaroo: video seeking in P2P systems," p. 6, Apr. 2009.
- [56] M. Karol, P. Krishnan, and J. J. Li, "VoIP network failure detection and user notification," in *Proceedings. 12th International Conference on Computer Communications and Networks (IEEE Cat. No.03EX712)*, 2003, pp. 511–516.
- [57] T. Nguyen and A. Zakhor, "Multiple Sender Distributed Video Streaming," *IEEE Trans. Multimed.*, vol. 6, no. 2, pp. 315–326, Apr. 2004.
- [58] N. Magharei and R. Rejaie, "Adaptive receiver-driven streaming from multiple senders," *Multimed. Syst.*, vol. 11, no. 6, pp. 550–567, Apr. 2006.
- [59] V. Räsänen, *Service Modelling: Principles and Applications*. Wiley, 2006, p. 296.
- [60] I. T. N. DawningStreams, *Practical JXTA II - Cracking the P2P puzzle*. Lulu Enterprises, Inc., 2010.
- [61] C. Massiot, *VLC media player API Documentation*. 2001.
- [62] "VideoLAN - libVLC media player, Open Source video framework for every OS!" [Online]. Available: <http://www.videolan.org/vlc/libvlc.html>. [Accessed: 21-Nov-2014].
- [63] P. Pale, J. Petrović, and B. Jeren, "LeCTo: a rich lecture capture solution," pp. 1037–1041, 2014.

- [64] “Caprica Software vlcj Project Page.” [Online]. Available: <http://capricasoftware.co.uk/legacy/projects/vlcj/index.html>. [Accessed: 21-Nov-2014].
- [65] A. Jewel, “Extreme Hacking.” Cunningham, Inc.
- [66] K. Beck, *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional, 1999, p. 224.