

Técnica de Búsqueda para la Prestación de Servicios sobre Redes Superpuestas P2P No Estructuradas

**Alejandro Muñoz Andrade
Diego Eryk Muñoz Luna**

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 Ingeniería de la Colaboración e Ingeniería del
Software Basada en la Colaboración**

Popayán, Noviembre de 2010

Técnica de Búsqueda para la Prestación de Servicios sobre Redes Superpuestas P2P No Estructuradas



Alejandro Muñoz Andrade
Diego Eryk Muñoz Luna

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 Ingeniería de la Colaboración e Ingeniería del Software Basada en la Colaboración
Popayán, Noviembre de 2010

DEDICATORIA

A mis Padres, Alvaro y Alma Lyda, por su gran esfuerzo, sacrificio y apoyo; gracias a ellos pude sacar mi carrera adelante y pude alcanzar todas las metas que me propuse a lo largo de ella.

A mis hermanos, Felipe y Daniel, por su apoyo incondicional y su comprensión; me fortalecieron en los momentos más difíciles.

A mi abuelo, Israel, por su forma de salir adelante en los momentos difíciles de la vida; me enseñó a nunca desfallecer y seguir siempre adelante.

A Dios, por su amor, su infinita grandeza y sabiduría; gracias por darme la vida y por ayudarme a cumplir con éxito esta etapa de mi vida.

Alejandro Muñoz Andrade

Cuando te fijas metas a largo plazo y estas se cumplen, siempre existen personas que te dieron su mano y te apoyaron incondicionalmente, a estas personas, mis Padres, Luz Marina y Laureano, por creer en mí y poner cada escalón para poder llegar donde estoy.

A mi familia, Angela y Diego Alejandro, que llegaron a mi vida para complementar mi felicidad, y que me dieron las fuerzas y ánimos para alcanzar mis sueños, los amo.

A mi familia porque aunque parecía una carrera eterna, hoy pueden decir, al fin; gracias por sus ánimos incondicionales.

Principalmente, quiero dedicar este triunfo a Dios, quien siempre estuvo a mi espalda para darme ese empujoncito de fe que necesitaba.

Diego Eryk Muñoz Luna

AGRADECIMIENTOS

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

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

Gracias a todas las personas, que en el transcurso de nuestra carrera, nos hicieron sus amigos y compartieron cada momento para cumplir esta meta que hoy se hace realidad.

Diego Eryk Muñoz Luna
Alejandro Muñoz Andrade

TABLA DE CONTENIDO

INTRODUCCIÓN	11
CAPÍTULO I. MARCO CONCEPTUAL.....	15
1.1. Técnica de Búsqueda	15
1.2. Prestación de Servicios.....	15
1.3. Redes Superpuestas P2P	16
1.3.1. Definición de Red Superpuesta P2P.....	16
1.3.2. Clasificación de Redes Superpuestas P2P.....	16
1.3.4. Comunicación en Redes Superpuestas P2P	18
1.4. Agentes y Sistemas Multi-Agente	19
1.4.1. Agentes.....	19
1.4.2. Sistemas Multi-Agente	21
1.4.3. Comunicación y Coordinación	22
1.4.4. FIPA.....	23
1.4.5. Ontologías	25
1.5. Sistemas Multi-Agente en Entornos P2P	26
CAPÍTULO II. BÚSQUEDA DE SERVICIOS	27
2.1. Búsqueda y Descubrimiento de Servicios	27
2.2. Búsqueda en Redes Superpuestas P2P No Estructuradas	28
2.3. Búsqueda de Servicios en Sistemas Multi-Agente.....	32
2.4. Características de los Métodos de Búsqueda.....	32
2.5. Comparación de los Métodos de Búsqueda	35
2.6. Selección del Método de Búsqueda.....	38
2.7. Priorización de las Características de los Métodos de Búsqueda.....	39
CAPÍTULO III. TÉCNICA DE BÚSQUEDA PARA LA PRESTACIÓN DE SERVICIOS.....	40
3.1. Introducción.....	40
3.2. Arquitectura del Sistema	40
3.2.1. Sistema Multi-Agente.....	41

3.2.2. Sistema P2P	42
3.2.3. Comunicador	42
3.3. Descripción de la Técnica de Búsqueda	43
3.3.1. Comportamientos de los Agentes	43
3.3.2. Comunicación entre Agentes y Ontologías	46
3.3.3. Envío de Mensajes	50
3.3.4. Interacción entre los Agentes y el Mundo Exterior	50
3.3.5. Conversión de Mensajes	51
3.4. Prestación de Servicios	52
3.4.1. Publicación del Servicio	53
3.4.2. Búsqueda del Servicio	56
3.4.3. Recuperación del Servicio	61
CAPÍTULO IV. TECNOLOGÍAS PARA EL DESARROLLO DE REDES SUPERPUESTAS P2P Y SISTEMAS MULTI-AGENTE.....	68
4.1. JXTA	68
4.1.1. ¿Qué es JXTA?	68
4.1.2. ¿Por qué JXTA?	68
4.1.3. Arquitectura JXTA.....	69
4.1.4. Protocolos.....	69
4.1.5. Servicios JXTA.....	69
4.2. JADE.....	70
4.2.1. ¿Qué es JADE?	70
4.2.2. ¿Por qué JADE?	70
4.2.3. Arquitectura JADE	70
4.2.4. Comportamientos en JADE	71
4.2.5. Servicios en JADE	71
CAPÍTULO V. DESARROLLO DEL PROTOTIPO SOFTWARE	73
5.1. Especificación de Requerimientos	73
5.1.1. Alcance	73
5.1.2. Funciones del Producto	73
5.1.3. Características del Usuario.....	74

5.1.4. Priorizar los Requisitos	74
5.1.5. Requerimientos Funcionales	74
5.1.6. Requerimientos No Funcionales	75
5.2. Análisis y Diseño	76
5.2.1. Actores del Sistema	76
5.2.2. Casos de Uso en Formato de Alto Nivel	76
5.2.3. Casos de Uso en Formato Extendido	77
5.2.4. Arquitectura del Prototipo Software	79
5.2.5. Diagrama de Clases y Diagramas de Secuencia.....	82
5.3. Implementación.....	83
5.4. Pruebas y Verificación de la Técnica de Búsqueda	83
5.4.1. Definición de los Escenarios de Prueba	83
5.4.2. Descripción del Marco Experimental.....	85
5.4.3. Desarrollo de los Escenarios de Prueba	86
5.4.4. Características Probadas.....	92
CAPÍTULO VI. CONCLUSIONES.....	98
CAPÍTULO VII. RECOMENDACIONES Y TRABAJO FUTURO	101
REFERENCIAS BIBLIOGRÁFICAS	103

LISTA DE FIGURAS

Figura 1. Modelo de Gestión de Agentes	24
Figura 2. Arquitectura del Sistema	40
Figura 3. Componentes del Sistema Multi-Agente	41
Figura 4. Componentes del Sistema P2P	42
Figura 5. Componentes del Comunicador	43
Figura 6. Ontología Servicio	47
Figura 7. Ontología Conexión.....	48
Figura 8. Ontología Mensaje	48
Figura 9. Ontología Búsqueda.....	49
Figura 10. Interacción entre los Agentes y el Comunicador	51
Figura 11. Diagrama de Flujo - Publicación del Servicio	53
Figura 12. Registrar un Servicio	54
Figura 13. Crear un Grupo para el Servicio.....	55
Figura 14. Publicar un Servicio.....	56
Figura 15. Diagrama de Flujo - Búsqueda y Recuperación del Servicio	57
Figura 16. Buscar un Servicio	59
Figura 17. Conectar el Cliente al Servidor (a)	60
Figura 18. Conectar el Cliente al Servidor (b)	61
Figura 19. Diagrama de Flujo - Prestación del Servicio	62
Figura 20. Diagrama de Flujo - Utilización del Servicio	63
Figura 21. Enviar Mensajes.....	64
Figura 22. Recibir Mensajes.....	65
Figura 23. Enviar Directorio.....	66
Figura 24. Recibir Directorio.....	67
Figura 25. Arquitectura JXTA	69
Figura 26. Arquitectura JADE.....	71
Figura 27. Diagrama de Casos de Uso – Paquetes	77
Figura 28. Diagrama de Casos de Uso Extendido - Paquete Servicio	78
Figura 29. Diagrama de Casos de Uso Extendido - Paquete Mensajes.....	78
Figura 30. Diagrama de Casos de Uso Extendido - Paquete Opciones.....	79
Figura 31. Arquitectura General del Prototipo	79
Figura 32. Módulo JADE	80
Figura 33. Módulo JXTA.....	81
Figura 34. Módulo Comunicador	81
Figura 35. Diagrama de Clases del Prototipo.....	82
Figura 36. Sala de Informática utilizada para las Pruebas	85
Figura 37. Representación del Escenario de Prueba No. 1	86
Figura 38. Representación del Escenario de Prueba No. 2	86
Figura 39. Representación del Escenario de Prueba No. 3	87
Figura 40. Representación del Escenario de Prueba No. 4	87
Figura 41. Representación del Escenario de Prueba No. 5	88
Figura 42. Representación del Escenario de Prueba No. 6	89

Figura 43. Activando y Desactivando Nodos Servidor	91
Figura 44. Activando y Desactivando Nodos Cliente.....	91
Figura 45. Gráfica de Desempeño de la Búsqueda.....	92
Figura 46. Gráfica de Tolerancia a Fallos (a)	93
Figura 47. Gráfica de Tolerancia a Fallos (b)	94
Figura 48. Gráfica de Costo de la Búsqueda.....	94
Figura 49. Escenario de Escalabilidad	96
Figura 50. Gráfica de Escalabilidad y Desempeño de la Búsqueda	97

LISTA DE TABLAS

Tabla 1. Comparación de las características de siete métodos de búsqueda.....	35
Tabla 2. Comparación de las métricas de nueve métodos de búsqueda	37
Tabla 3. Rendimiento relativo de tres métodos de búsqueda	38
Tabla 4. Escenarios de Prueba	84
Tabla 5. Formato de Registro para los Resultados de las Pruebas	84
Tabla 6. Escenario de Prueba No. 1	86
Tabla 7. Escenario de Prueba No. 2	87
Tabla 8. Escenario de Prueba No. 3	87
Tabla 9. Escenario de Prueba No. 4	88
Tabla 10. Escenario de Prueba No. 5	89
Tabla 11. Escenario de Prueba No. 6	90

INTRODUCCIÓN

En los últimos años se han venido implementando con gran notoriedad las Redes Superpuestas Peer-to-Peer (P2P) [1], un nuevo modelo a nivel de redes que surge como una alternativa al modelo cliente-servidor [2]. Esto se debe al gran crecimiento que ha tenido la red de redes, Internet, al evidente incremento de usuarios que hacen uso de ella y al aumento notorio en las capacidades de cómputo en las máquinas "cliente", las cuales son capaces de prestar y consumir servicios y recursos directamente, sin llegar a depender de una máquina central "servidor" encargada de prestarlos [3]. De acuerdo a la forma que están organizados sus nodos, las redes superpuestas P2P pueden ser estructuradas o no estructuradas [4], en el presente trabajo se estudiarán las redes superpuestas P2P no estructuradas y el problema a resolver se centrará solo en este tipo de clasificación.

Las redes superpuestas P2P son capaces de prestar una gran cantidad de servicios: Uso compartido de archivos, distribución de contenidos, sincronización de datos, mensajería instantánea, telefonía, videoconferencia IP y uso en técnicas de aprendizaje colaborativo [5].

Los potenciales usos de las redes superpuestas P2P son muchos, pero se encuentran varios problemas en los servicios que estas prestan, por ejemplo: Los equipos computacionales, en una red, permanecían ocultos para el resto de los usuarios, por lo tanto era prácticamente imposible nombrarlos y direccionarlos, bajo los esquemas clásicos de trabajo en red. En cuanto a la implementación de este tipo de redes, también existen algunos problemas a resolver, como, por ejemplo: Confianza mutua, seguridad de las aplicaciones e integridad de los datos. En algunos ambientes los inconvenientes relacionados con la seguridad y la privacidad resultan importantes barreras para su rápida adopción [6].

La carencia de interoperabilidad, entre las diferentes implementaciones de este tipo de redes, es otro de los obstáculos que se debe afrontar, debido a que cada una de estas implementaciones posee su propia serie de protocolos P2P, lo cual impide la comunicación entre ellas. Así, una aplicación diseñada para el uso compartido de archivos, no puede comunicarse con una de mensajería instantánea y viceversa [5]. Otro problema surge del hecho de que diferentes redes superpuestas P2P son definidas y desplegadas. Es deseable que los usuarios de una red superpuesta P2P puedan llegar a los usuarios de otras redes superpuestas P2P, sin la necesidad de aplicaciones que implementen los mecanismos y protocolos de todas las redes superpuestas P2P disponibles. Por lo tanto, se necesitan mecanismos para la interconexión y la interoperabilidad entre las redes superpuestas P2P [7].

Existen muchas aplicaciones alrededor de las redes superpuestas P2P, las cuales son utilizadas en gran cantidad por un número amplio de personas y compañías en todo el mundo, pero aún existen muchos aspectos de investigación sobre las redes superpuestas P2P que aún faltan por resolverse; por ejemplo: Las actuales redes superpuestas P2P suelen formarse basadas en la información de la capa de aplicación y hacen caso omiso de la estructura y las políticas de la infraestructura de red subyacente. Como resultado, los nodos a menudo se conectan con otros nodos en una red remota, aunque un nodo equivalente esté disponible en la red local. Esto puede conducir a un uso ineficiente de los recursos de red y generar conflictos entre las aplicaciones y el enrutamiento, a nivel de red [5].

En las redes superpuestas P2P de hoy en día, se están ofreciendo diferentes tipos de servicios heterogéneos, encontrar estos servicios de forma eficiente, se torna un problema de gran importancia [8]. Uno de los problemas más grandes de las redes superpuestas P2P, es que no siempre se tiene en cuenta una de las características esenciales para la prestación de servicios, la facilidad de uso, por eso debería pensarse en una interfaz común para la prestación de todos los servicios [9].

Muchas empresas, especializadas en el área de prestación de servicios, se están dando cuenta de la necesidad de generalizar el modo en que estos servicios se prestan. De hecho, el ideal de las redes ISDN¹ es crear redes convergentes, capaces de prestar gran variedad de servicios integrados, a través de una sola interfaz de comunicación, llevada por cualquier tipo de red. Otro de los inconvenientes al trabajar con redes superpuestas P2P es la búsqueda de los recursos, ya que los sistemas P2P puros carecen de una entidad central encargada de realizar dicha tarea, lo que dificulta a los nodos de la red, saber qué otros nodos contienen los servicios que se están buscando; este problema ataca la principal característica de las redes superpuestas P2P, que es la alta disponibilidad, gracias a la existencia de múltiples nodos en un grupo, cosa que facilita que al menos un nodo del grupo pueda satisfacer una petición de un usuario [10].

La mayoría de las investigaciones anteriores, sobre búsqueda de servicios en las redes superpuestas P2P, se ha centrado en el intercambio de archivos o almacenamiento distribuido de información. Como resultado, las redes superpuestas P2P han utilizado principalmente técnicas de búsqueda sencillas, basadas en nombres de documentos, identificadores, palabras clave o en términos de vocabulario, y proveen un soporte muy limitado para la búsqueda [11]. En la generación actual de redes superpuestas P2P, se pretende dar solución a este problema, de forma que la búsqueda de los recursos sea determinista; es decir, si un recurso en concreto se encuentra en la red superpuesta P2P, entonces, el recurso debería poder obtenerse para ser utilizado por cualquier nodo dentro de la

¹ Integrated Services Digital Network

red superpuesta P2P. Pero el más grande de los desafíos a la hora de trabajar con redes superpuestas P2P, tal vez sea la continua inestabilidad del entorno en el que se prestan los servicios, debido a que no existe una entidad permanente que realice la prestación de los servicios [5].

Por tanto, se considera que es indispensable establecer una forma de prestar los servicios, bajo un entorno inestable, aprovechando los dispositivos ubicados en la red, de tal forma que éstos puedan interactuar entre sí, buscando, localizando y recuperando los recursos y servicios que son ofrecidos en las redes superpuestas P2P no estructuradas.

Este proyecto realiza una contribución en el área de las redes superpuestas P2P, con la especificación de una Técnica de Búsqueda para la Prestación de Servicios sobre Redes Superpuestas P2P No Estructuradas. En la primera fase se incluye la información relevante sobre la contextualización de las redes superpuestas P2P, lo que permitió tomar una decisión importante sobre cómo realizar la búsqueda de servicios sobre estas redes superpuestas P2P no estructuradas. En la técnica de búsqueda se incluyeron los Sistemas Multi-Agente aprovechando los beneficios de los protocolos y las comunicaciones de las redes superpuestas P2P. Con esta asociación de sistemas se adecuó una técnica, la cual se enfocó en la creación, publicación y búsqueda de servicios dentro de las redes superpuestas P2P, utilizando las características y beneficios de los Sistemas Multi-Agente.

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 de Servicios: En este capítulo se describen las diferentes técnicas y algoritmos de búsqueda de servicios sobre redes superpuestas P2P y Sistemas Multi-Agente.

Capítulo III. Técnica de Búsqueda para la Prestación de Servicios: En este capítulo se presenta la construcción o adecuación de la técnica, la cual está basada en redes superpuestas P2P y Sistemas Multi-Agente.

Capítulo IV. Tecnologías para el Desarrollo de Redes Superpuestas P2P y Sistemas Multi-Agente: En este capítulo se describen las tecnologías escogidas para la implementación de la red superpuesta P2P y el Sistema Multi-Agente en el prototipo software.

Capítulo V. Desarrollo del 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. Conclusiones: En este capítulo se presentan las conclusiones obtenidas en este trabajo de grado.

Capítulo VII. Recomendaciones y Trabajo Futuro: En este capítulo se presentan algunas recomendaciones para realizar trabajos futuros en los campos relacionados con la investigación.

CAPÍTULO I. MARCO CONCEPTUAL

A continuación se describen los conceptos necesarios, que están involucrados en la búsqueda de servicios sobre redes superpuestas P2P. Se considera conveniente presentar las definiciones de los términos que se encuentran en el enunciado del problema y de los demás conceptos que giran alrededor de la solución del problema, para conseguir una mejor comprensión del problema. Dichos conceptos se presentan a continuación.

1.1. Técnica de Búsqueda

Se debe entender como técnica, un procedimiento o conjunto de estos, 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 [12].

El concepto de búsqueda se refiere a realizar alguna acción que conduzca a descubrir algo que hasta entonces era desconocido, y que además se pueda aprovechar y sea de utilidad para el propósito establecido [12]. En el caso de este trabajo de investigación ese "algo" se refiere a los servicios que las redes superpuestas P2P prestan en algún momento determinado. Por otro lado, la búsqueda en redes superpuestas P2P consiste en encontrar los nodos más cercanos, que prestan el servicio deseado, al nodo que está solicitando el servicio [13].

1.2. Prestación de Servicios

El término prestar, significa entregar algo a alguien para que lo utilice durante algún tiempo y después lo devuelva [12].

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 ficheros. 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 [14]. En [15] se define un servicio, como una funcionalidad predefinida que puede ser utilizada por los nodos que están conectados a una red superpuesta P2P.

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.

1.3. Redes Superpuestas P2P

1.3.1. Definición de Red Superpuesta P2P

Una red superpuesta P2P es una red lógica, construida sobre una red física, que facilita la localización de los recursos distribuidos sin un control centralizado. Los sistemas P2P son una clase de aplicaciones distribuidas, que explotan los recursos² 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 [4].

1.3.1.1. Peer

Un peer es un nodo que está presente en una red superpuesta P2P y que forma la unidad fundamental de procesamiento de cualquier solución P2P. Un peer es una entidad capaz de realizar algún trabajo útil y comunicar los resultados de este trabajo a otra entidad sobre la red, ya sea directa o indirectamente [16].

1.3.2. Clasificación de Redes Superpuestas P2P

1.3.2.1. Redes Superpuestas P2P Puras

Una red superpuesta P2P pura, puede ser descrita, como un sistema de redes descentralizado, en el que todos los equipos participantes, también conocidos como peers, pares o nodos, tienen deberes y responsabilidades simétricas. En términos simples, todos los equipos participantes actúan como clientes y servidores entre sí, dando lugar a un gran número de fuentes de información y poder de cómputo. Las principales características de las redes superpuestas P2P son la descentralización, la auto-organización, el dinamismo y la tolerancia a fallos, todas las cuales hacen el paradigma P2P muy atractivo para el almacenamiento y recuperación de información [4].

1.3.2.2. Redes Superpuestas P2P Centralizadas

Existe dentro de la arquitectura de redes superpuestas P2P, una variación de la red original o pura, denominada Red Superpuesta P2P Centralizada [15], el cual hace uso de un servidor encargado de recibir todas las peticiones y de informar a los usuarios en donde se encuentra la información requerida. Una vez el

² Los recursos proporcionados pueden ser: Nodos, Objetos, Servicios, Documentos, Espacio de almacenamiento, Ciclos de CPU, Contenidos, Ancho de banda, entre otros.

solicitante es informado de dónde puede encontrar la información requerida³, se establece una conexión extremo a extremo entre las dos máquinas, el cliente o solicitante del servicio, y el servidor o prestador del servicio. El inconveniente de este nuevo modelo de red, es que deja un único punto de fallo, por lo que cualquier tipo de ataque o caída del servidor central inhabilita a todo el sistema.

Por otro lado, dependiendo de cómo se organizan los nodos dentro de estas redes, las redes superpuestas P2P pueden ser clasificadas como estructuradas o no estructuradas [4].

1.3.2.3. Redes Superpuestas P2P Estructuradas

Las redes superpuestas P2P estructuradas mantienen una tabla hash o DHT⁴. En este tipo de redes, a todo contenido se le da un valor específico 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 global determina cuál nodo es responsable de proveer ese contenido, y la búsqueda se dirige hacia ese nodo en particular [4].

1.3.2.4. Redes Superpuestas P2P No Estructuradas

Las redes superpuestas P2P no estructuradas, se forman cuando los nodos que la componen 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. De esta forma cuando un nodo busca un contenido utiliza los enlaces que tiene para poder encontrarlo, si por alguna razón los enlaces que tiene ya no están disponibles en la red, el nodo realiza una nueva búsqueda para encontrar otros enlaces. La desventaja con este sistema, es que si el contenido que se solicita es raro o no es muy popular, es posible que la búsqueda no obtenga buenos resultados o incluso ninguno [4].

En las redes superpuestas P2P no estructuradas los recursos (nodos, objetos, servicios o documentos) tienen un único identificador, el cual se usa para especificar el destino de la consulta. Los nodos que están directamente conectados se conocen como vecinos. Cada nodo tiene solo un conocimiento local de la red⁵, y cada uno tiene una colección de servicios o documentos que servir y pueden hacer una serie de solicitudes de aquellos servicios o documentos que desean obtener. Por lo tanto, los algoritmos de búsqueda no pueden imponer la colocación de los servicios o documentos, ni tampoco la replicación en el sistema [17].

³ La información requerida por un nodo para realizar una conexión puede ser la dirección IP o el nombre de la máquina del servidor potencial.

⁴ Distributed Hash Table

⁵ Por ejemplo cada uno de los nodos solo conoce a sus nodos vecinos.

Los recursos tienen una popularidad diferente, que afecta el número de réplicas y solicitudes recibidas. La distribución de replicación determina el número y la identidad de los objetos almacenados en cada nodo. La distribución de consulta controla la frecuencia de las solicitudes de cada objeto. Una búsqueda tiene éxito si se descubre al menos una réplica del objeto solicitado. La proporción de éxito para las búsquedas totales realizadas se llama la tasa de éxito o precisión. Una búsqueda puede dar lugar a descubrimientos múltiples o aciertos (hits), que son réplicas de un mismo objeto almacenado en diferentes nodos. Un parámetro global TTL⁶ representa la distancia máxima, en saltos, que una consulta puede alcanzar antes de que sea descartada [17].

En las redes superpuestas P2P no estructuradas, los nodos generalmente usan la técnica de inundación para resolver una consulta. Estas técnicas pueden ser usadas para búsquedas complejas debido a que no hay límite para indexar datos en la red. El problema principal de estos sistemas es que el costo de la búsqueda no es bien escalado, por lo tanto crece proporcionalmente con el tamaño de la red. La complejidad del mantenimiento de las redes superpuestas P2P no estructuradas es superior a la de las redes superpuestas P2P estructuradas. Como cada nodo puede seleccionar cualquier cantidad de nodos como un vecino, la meta, de la estrategia de mantenimiento, es mantener el número de vecinos igual a un valor que haya sido especificado previamente, lo que garantiza una economía en términos de comunicación [4].

Por ejemplo, en [18], después de un periodo de tiempo pre-especificado, cada nodo inicia un mensaje de broadcast con un cierto TTL para descubrir nuevos nodos conectados a la red superpuesta P2P. Si un nodo que recibe el mensaje está todavía activo, este debe confirmarlo mediante el envío de un mensaje de respuesta, el cual es llamado mensaje pong. El nodo que envía la petición espera por un periodo de tiempo a que el mensaje pong llegue. El nodo puede seleccionar un nuevo nodo vecino desde el conjunto de nodos activos que enviaron el mensaje de respuesta, para reemplazar uno o varios nodos vecinos que se hayan desconectado de la red superpuesta P2P.

1.3.4. Comunicación en Redes Superpuestas P2P

1.3.4.1. Comunicación Punto a Punto

La comunicación punto a punto es la vía más obvia de comunicación entre nodos. Esta comunicación consiste en abrir un puerto de comunicación entre 2 nodos. La comunicación punto a punto, es utilizada principalmente para enviar mensajes que necesitan ser directamente direccionados a un nodo específico [19].

⁶ Time to Live

1.3.4.2. Comunicación Broadcast

El broadcasting se hace normalmente dentro de una sub-red, en la cual hay un nodo que envía la petición (broadcaster) y uno o más nodos que escuchan esta petición (listeners). El mensaje es enviado a través de la red por el broadcaster y los otros nodos lo reciben por un puerto específico. La comunicación mediante broadcast es muy económica porque hay solo un mensaje enviado por cada sub-red. Desafortunadamente, los mensajes de broadcast son muchas veces filtrados o bloqueados. El broadcasting es muy utilizado para el descubrimiento de nodos y recursos dentro de la red superpuesta P2P [19].

1.3.4.3. Comunicación Multi-Punto

La comunicación multi-punto es algunas veces asociada con la comunicación broadcast. Sin embargo, la diferencia es que las conexiones múltiples son hechas con transmisiones múltiples que contienen los mismos datos. En muchas situaciones el broadcasting no puede usarse, por lo tanto la alternativa es abrir una conexión multi-punto, la cual crea múltiples conexiones punto a punto [19].

Existen varios problemas con la comunicación multi-punto. Primero, no se garantiza que los mensajes sean recibidos simultáneamente. Es posible que el intervalo entre la transmisión y la recepción sea un poco más largo en cada punto, debido a los tipos de nodos de una red superpuesta P2P. En este tipo de comunicación, los mensajes pueden perderse debido a que algunos nodos se conectan solo un periodo de tiempo muy corto, tienen un ancho de banda restringido, o tienen recursos de hardware o software insuficientes. Sin embargo, la comunicación multi-punto es muy utilizada, buena y eficiente [19].

1.4. Agentes y Sistemas Multi-Agente

1.4.1. Agentes

Un agente es un componente especial de un sistema de computación que está situado en algún entorno, y que es capaz de actuar de forma autónoma y flexible en este entorno para lograr un objetivo. Un agente puede decidir cooperar con otros agentes para su beneficio mutuo o puede competir para cumplir sus propios objetivos [20] [21].

Un agente toma información de entrada desde su entorno y produce una acción de salida que afecta este entorno. En muchas ocasiones, un agente no tiene el control total de su entorno, pero si tiene un control parcial y puede influenciar el entorno. Desde este punto de vista, significa que si el agente realiza una misma acción, ejecutada varias veces y en las mismas circunstancias, puede provocar efectos diferentes en el entorno y por lo tanto no podría obtenerse el efecto deseado [21].

Generalmente un agente tiene un conjunto de acciones que tiene disponibles para poder ejecutar, llamadas comportamientos. Estos comportamientos representan la capacidad de los agentes de modificar su entorno. Es importante mencionar que no todos los comportamientos pueden ser ejecutados en todas las situaciones que se le presentan al agente. El principal problema que debe resolver un agente es decidir cuál de los comportamientos debe ejecutar en orden para satisfacer los objetivos que debe cumplir y para los que fue diseñado [21].

Un agente debe ser [20] [21] [22]:

- **Autónomo:** Los agentes operan sin la directa intervención de los humanos y tienen el control sobre sus acciones y su estado interno.
- **Reactivo:** Los agentes son capaces de percibir su entorno y responder a los cambios que ocurran en él, para satisfacer sus objetivos.
- **Proactivo:** Los agentes son capaces de tomar la iniciativa para cumplir sus propios planes y objetivos.
- **Social:** Los agentes deben tener la capacidad de interactuar, cooperar y comunicarse con otros agentes o con los humanos, para satisfacer sus objetivos.

Otras características de los agentes pueden ser [20] [21] [22]:

- **Continuidad Temporal:** Un agente se considera un proceso sin fin, ejecutándose continuamente y desarrollando su función.
- **Autonomía:** Un agente es completamente autónomo si es capaz de actuar basándose en su experiencia. El agente es capaz de adaptarse aunque el entorno cambie severamente. Por otra parte, una definición menos estricta de autonomía sería cuando el agente percibe el entorno.
- **Racionalidad:** Un agente siempre realiza lo correcto, a partir de los datos que percibe del entorno.
- **Adaptabilidad:** Esta característica está relacionada con el aprendizaje que un agente es capaz de realizar, y si ese aprendizaje puede cambiar su comportamiento.
- **Movilidad:** Es la capacidad de un agente de trasladarse a través de una red telemática.

- **Veracidad:** Es la asunción de que un agente no comunica información falsa a propósito.
- **Benevolencia:** Es la asunción de que un agente está dispuesto a ayudar a otros agentes, si esto no entra en conflicto con sus propios objetivos.

Los agentes son considerados actualmente uno de los paradigmas más importantes en el mundo de los sistemas y el desarrollo de software. Por un lado, pueden proveer métodos para conceptualización, diseño e implementación de sistemas de software; por otro lado, pueden solucionar el problema legal de la integración de software [20].

1.4.2. Sistemas Multi-Agente

Un Sistema Multi-Agente o MAS⁷, es aquél que está compuesto por un conjunto de agentes que cooperan, coordinan y se comunican entre ellos para conseguir un objetivo común [22].

Las principales ventajas de la utilización de un Sistema Multi-Agente son:

- **Modularidad:** Se reduce la complejidad de la programación al trabajar con unidades más pequeñas, que permiten una programación más estructurada.
- **Eficiencia:** La programación distribuida permite repartir las tareas entre los agentes, consiguiendo paralelismo, esto quiere decir que hay agentes trabajando al mismo tiempo en diferentes máquinas.
- **Fiabilidad:** El hecho de que un elemento del sistema deje de funcionar no tiene que significar que el resto también deje de hacerlo; además, se puede conseguir más seguridad replicando servicios críticos para así conseguir redundancia cuando se necesite.
- **Flexibilidad:** Se pueden añadir y eliminar agentes dinámicamente al sistema.

Las tecnologías de agentes han sido el objeto de investigaciones y discusiones extensas dentro de la comunidad científica por varios años, pero es solo recientemente que se ha visto un significativo grado de explotación en las aplicaciones comerciales. Los Sistemas Multi-Agente están siendo utilizados en una amplia variedad de aplicaciones, desde sistemas sencillos para asistencia personal, hasta sistemas más complejos y críticos para aplicaciones industriales. Algunos ejemplos donde los Sistemas Multi-Agente han sido incluidos en la

⁷ Multi-Agent System

industria incluyen control de procesos, diagnósticos de sistemas, manufacturas, logística de transporte y gestión de redes [20].

1.4.3. Comunicación y Coordinación

Uno de los componentes principales de los Sistemas Multi-Agente es la comunicación. De hecho, los agentes necesitan ser capaces de comunicarse con los usuarios, con recursos del sistema y con otros agentes para cooperar, colaborar, negociar y realizar otras tareas que lo ameriten, para poder cumplir sus objetivos. En particular, los agentes interactúan con otros agentes utilizando un lenguaje de comunicación especial, llamado ACL (Agent Communication Language), que permite una separación entre la comunicación y el lenguaje del contenido [20].

El primer lenguaje de comunicación de agentes desarrollado fue KQML, en el año de 1990, como parte del proyecto ARPA Knowledge Sharing Effort del gobierno de los Estados Unidos. KQML es un lenguaje y protocolo para intercambiar información y conocimiento, que define un número de verbos representativos y permite que el contenido del mensaje sea representado como un lenguaje de orden lógico [20].

Actualmente el lenguaje de comunicación de agentes más estudiado y utilizado es FIPA-ACL, el cual incorpora muchos aspectos de KQML. Las principales características de FIPA-ACL son: la posibilidad de usar diferentes lenguajes de contenido y la capacidad de gestionar las conversaciones a través de protocolos de interacción predefinidos. La coordinación, es un proceso en el cuál los agentes interactúan, para ayudar a asegurar que una comunidad de agentes individuales funcione de forma coherente. Hay varias razones que explican por qué los agentes necesitan ser coordinados [20]:

- Los objetivos de los agentes pueden causar conflictos con las acciones de los agentes.
- Los objetivos de los agentes pueden ser interdependientes.
- Los agentes pueden tener diferentes capacidades y diferente conocimiento.
- Los objetivos de los agentes pueden alcanzarse más rápidamente si hay varios agentes que trabajen en ellos.

La coordinación entre agentes, puede ser alcanzada con una amplia variedad de enfoques incluyendo estructura organizacional, contratación, planeación multi-agente centralizada o distribuida y negociación competitiva o cooperativa.

1.4.4. FIPA

FIPA⁸ fue establecido en 1996, como una asociación internacional sin ánimo de lucro, para desarrollar una colección de estándares relacionados con la tecnología de agentes software. Los miembros iniciales, un conjunto de organizaciones académicas e industriales, deciden construir un conjunto de especificaciones estándar para las tecnologías de agentes software que puedan ser usados por una amplia variedad de aplicaciones [20] [23].

FIPA sigue el siguiente conjunto de principios [23]:

1. Las tecnologías de agentes proveen un nuevo paradigma para solucionar viejos y nuevos problemas.
2. Algunas tecnologías de agentes tienen un considerable grado de madurez.
3. Algunas tecnologías de agentes requieren de estandarización para ser utilizadas.
4. La estandarización de tecnologías genéricas deben ser mostradas y proveer resultados efectivos por otras estandarizaciones.
5. La estandarización de los mecanismos internos de agentes no es el interés principal, pero sí lo es la infraestructura y el lenguaje requerido para la interoperación.

FIPA-ACL tiene un conjunto de estados para representar un mensaje, llamados actos comunicativos o performativos. FIPA-ACL tiene un conjunto de 22 actos comunicativos, donde cada acto es descrito usando una forma narrativa y una semántica formal, basadas en una lógica modal que especifica: los efectos de enviar el mensaje sobre las actitudes mentales de los agentes que envían y reciben el mensaje. Esta forma de lógica es consistente con el modelo de razonamiento BDI⁹ [20] [23].

Basada en estos actos comunicativos, FIPA define un conjunto de protocolos de interacción, los cuales consisten de una secuencia de actos comunicativos para coordinar acciones multi-mensajes, para establecer concordancia y para realizar otro tipo de acciones. Dentro de la estructura de cada mensaje, FIPA-ACL no exige el uso de un lenguaje particular, para expresar el contenido, aunque existen especificaciones para varias representaciones, incluyendo FIPA-SL, FIPA-KIF y FIPA-RDF. Solo FIPA-SL ha sido promovido como un estándar [20] [23].

⁸ The Foundation for Intelligent, Physical Agents

⁹ BDI: Belief, Desires, Intention (Opinión, Deseos, Intención)

1.4.4.1. Gestión de Agentes FIPA

El segundo aspecto fundamental de las tecnologías de agentes tratado por la FIPA es la gestión de agentes, un marco de trabajo normativo en el cual los agentes pueden existir, operar y ser gestionados. Este establece un modelo de referencia lógica para la creación, registro, localización, comunicación, migración y operación de agentes. El modelo de referencia contiene los componentes mostrados en la Figura 1 [20] [23].

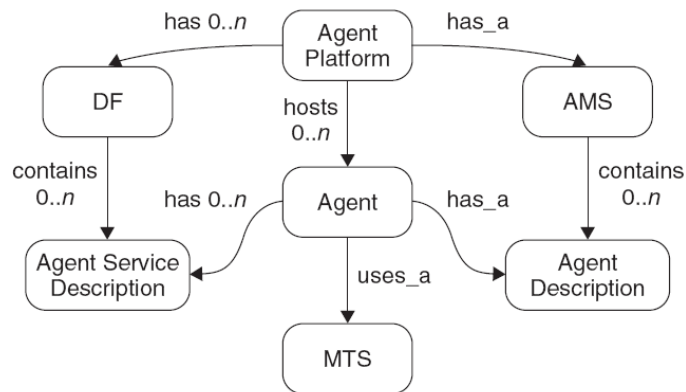


Figura 1. Modelo de Gestión de Agentes

A continuación se describen los elementos del modelo de gestión de agentes [23]:

- **Plataforma de Agentes (AP):** La plataforma de agentes provee la infraestructura física en la cual los agentes son desplegados. La plataforma de agentes está compuesta de las maquinas, sistemas operativos, componentes de la gestión de agentes FIPA, los agentes y cualquier software de soporte adicional. Una plataforma de agentes puede estar distribuida en múltiples computadores, en donde los agentes no necesariamente están localizados en un mismo computador.
- **Agente:** Un agente es un proceso computacional, que habita en una plataforma de agentes y ofrece uno a más servicios computacionales, que pueden ser publicados como una descripción del servicio. Un agente debe tener un propietario y debe soportar un concepto de identidad, que puede ser descrito usando un AID¹⁰, que identifica un agente para que sea diferenciado de una manera no ambigua. Un agente puede tener registradas varias direcciones de transporte para poder ser contactado.
- **Facilitador de Directorio (DF):** El DF provee el servicio de páginas amarillas a otros agentes. Es un componente opcional de la plataforma de agentes. El DF

¹⁰ Identificador de Agentes FIPA (FIPA Agent Identifier)

mantiene una lista precisa y completa de agentes, que provee la información actualizada acerca de los agentes, la descripción de sus servicios y su directorio de servicios, para todos los agentes autorizados en la plataforma. Una plataforma de agentes puede soportar cualquier número de DFs, los cuales pueden registrarse con otro DF para formar federaciones.

- **Sistema de Gestión de Agentes (AMS):** El AMS es el responsable de gestionar la operación de la plataforma de agentes; como por ejemplo: la creación y el borrado de agentes, y la migración de agentes desde y hacia la plataforma. Es un componente obligatorio de la plataforma de agentes. Cada agente debe registrarse con un AMS, para obtener su propio AID, que es almacenado por el AMS en un directorio que contiene el AID y el estado actual de todos los agentes dentro de la plataforma. Una plataforma de agentes solo puede tener un AMS.
- **Servicio de Transporte de Mensajes (MTS):** El MTS es un servicio que provee la plataforma de agentes para el transporte de mensajes FIPA-ACL entre agentes de una misma plataforma o entre agentes de diferentes plataformas. Los mensajes proveen una cabecera de transporte que contiene un conjunto de parámetros específicos.

1.4.5. Ontologías

Históricamente, una ontología se refiere al estudio filosófico del ser o el existir. En Inteligencia Artificial, una ontología trabaja con categorías que pueden ser sobre cuantificadas y con la forma como estas categorías se relacionan unas con otras [24]. Gruber se refiere a una ontología como: “Una especificación formal explícita de una conceptualización compartida” [25].

Una ontología es un método para representar e interrelacionar objetos o conceptos, en una jerarquía de tipos de conceptos, de acuerdo a diferentes niveles de generalidad. Esta jerarquía de conceptos puede ser útil para indexar y organizar información y para gestionar la resolución de problemas [24].

Las ontologías generalmente se usan para especificar y comunicar el conocimiento del dominio de una manera genérica y son muy útiles para estructurar y definir el significado de los términos [26].

Independientemente del ámbito en que se desarrollen, es necesario aclarar que la base para una ontología es la conceptualización, junto con un vocabulario para referirse a las entidades de un dominio particular. Es decir, las ontologías para representar el conocimiento precisan los siguientes componentes [26]:

- **Conceptos:** Son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- **Relaciones:** Representan la interacción y enlace entre los conceptos, formando la taxonomía del dominio. Las relaciones básicas son: es-una, tiene-una, instancia-de, parte-de, subclase-de, conectada-a.
- **Funciones:** Son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología.
- **Instancias:** Se utilizan para representar objetos determinados de un concepto.
- **Axiomas:** Son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Especifican las definiciones de los términos en la ontología y las restricciones de sus interpretaciones. Los axiomas deben proveerse para definir la semántica o el significado de los términos.

1.5. Sistemas Multi-Agente en Entornos P2P

Existe una gran similitud entre los Sistemas Multi-Agente y los entornos P2P. Ambos se definen como auto-organizables, los cuales operan en entornos potencialmente dinámicos con frecuentes entradas y salidas del sistema, ya sean de agentes o de nodos, respectivamente. Uno de los objetivos principales de ambos sistemas es beneficiarse de los recursos distribuidos compartidos entre las diferentes máquinas que conforman la red [22].

Los sistemas P2P aportan mejor rendimiento, escalabilidad, disponibilidad y fiabilidad, comparado con los sistemas tradicionales centralizados. Por otro lado, los Sistemas Multi-Agente están basados generalmente en sistemas distribuidos tradicionales y por tanto migrar hacia entornos P2P es una buena alternativa para compensar las dificultades que se presentan en los sistemas dinámicos [22].

Ambos sistemas pueden obtener beneficios de esta asociación. Por un lado, los Sistemas Multi-Agente se benefician de los mecanismos P2P de conectividad, tolerancia a fallos y balanceo de carga, para mejorar la calidad de sus servicios. Por otro lado, los sistemas P2P se benefician por la cómoda y sencilla implementación de los nodos mediante agentes, gracias a la alta modularidad ofrecida en los Sistemas Multi-Agente; y de la fácil y útil gestión de los servicios gracias a la característica de directorio de páginas blancas y de páginas amarillas de los Sistemas Multi-Agente [22].

CAPÍTULO II. BÚSQUEDA DE SERVICIOS

En este capítulo se describen las diferentes técnicas y algoritmos de búsqueda de servicios sobre redes superpuestas P2P y Sistemas Multi-Agente.

2.1. Búsqueda y 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 [27].

El descubrimiento sintáctico se basa en técnicas de comparación de interfaces (por ejemplo: WSDL¹¹, IDL¹², Interfaces RMI¹³, 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 [27] [28].

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 [27] [28].

En el presente trabajo de grado, se utilizará el enfoque sintáctico, debido a que este permite realizar búsqueda de servicios más fácilmente, pero sin disminuir la calidad de los resultados. Muchas de las tecnologías de desarrollo, tanto de Sistemas Multi-Agente, como de redes superpuestas P2P, lo utilizan en sus implementaciones.

¹¹ Web Services Description Language, es un formato XML que se utiliza para describir servicios Web.

¹² Interface Description Language, es un lenguaje de especificación de interfaces que se utiliza en software de computación distribuida.

¹³ Remote Method Invocation.

2.2. Búsqueda en Redes Superpuestas P2P No Estructuradas

A continuación, se describe una gran variedad de métodos de búsqueda de servicios y recursos, para redes superpuestas P2P 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 [17].

Los métodos de búsqueda más importantes, en redes superpuestas P2P no estructuradas, se resumen a continuación [13] [17] [29]:

- **Gnutella:** El algoritmo original Gnutella, también conocido como esquema de inundación¹⁴, 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.
- **Modified-BFS¹⁵ (Mod-BFS):** 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. Esto reduce el promedio de producción de mensajes, pero todavía contacta una alta cantidad de nodos.
- **Iterative Deepening:** En este método se inician múltiples búsquedas en anchura (BFS), con límites de profundidad cada vez más largos, hasta que la consulta se realice de manera satisfactoria o hasta que la profundidad máxima D sea alcanzada.
- **Random Walks (RWalks):** En este método 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. Estos mensajes son conocidos como caminantes (walkers). Un caminante termina ya sea con un éxito o un fracaso. Este método logra una reducción considerable de mensajes, comparado con el esquema de inundación original. Por otro lado, las tasas de éxito y el número de hits¹⁶ varían mucho, dependiendo de la topología de red y de las consultas realizadas al azar.

¹⁴ Flooding Scheme: Esquema de Inundación o Búsqueda Ciega

¹⁵ Breadth First Search (Búsqueda Primero en Anchura)

¹⁶ Un hit puede entenderse como un acierto que tiene el método de búsqueda.

- **Super-Peer Approaches - Gnutella2 (G2):** En el método de Súper-Nodos, cuando un súper nodo¹⁷ recibe una consulta de un nodo normal¹⁸, el súper nodo transmite la consulta a sus hojas correspondientes y a sus súper nodos vecinos. Estos súper nodos procesan la consulta a nivel local y la transmiten a sus hojas correspondientes. Ningún otro nodo es visitado en este algoritmo. Los súper nodos vecinos regularmente intercambian tablas de repositorios locales para filtrar el tráfico innecesario. Este esquema se basa en una estructura jerárquica dinámica de la red, para reducir los efectos de la inundación. El número de nodos hoja por súper nodo debe mantenerse alto, incluso después de ser visitado el nodo, para poder reducir el número de mensajes transmitidos y aumentar los objetos descubiertos.
- **Intelligent-BFS (Int-BFS):** También conocido como *Directed-BFS*. 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. Primero un nodo identifica todas las consultas idénticas a la actual, de acuerdo a una métrica de similitud de consultas, luego elige transmitir a un número determinado de nodos vecinos, los cuales hayan retornado anteriormente mayores resultados para esta consulta. Si ocurre un hit, la consulta toma la ruta inversa al nodo solicitante y actualiza los índices locales. Este enfoque se centra más en el descubrimiento de objetos que en la reducción de mensajes. Este algoritmo, comparado con el método Modified-BFS, incrementa el número de hits, pero con un incremento en el número de mensajes producidos, debido al proceso de actualización de índices. Su precisión depende, en gran medida, de que los nodos se especializan en un determinado tipo de objetos o documentos.
- **Adaptive Probabilistic Search (APS):** 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. La búsqueda se basa en el despliegue de k caminantes y de una transmisión probabilística. Cada nodo intermedio transmite la consulta a uno de sus nodos vecinos con la probabilidad dada por su índice local. Los valores del índice se incrementan en las trayectorias de éxito y disminuyen en el caso contrario. La calidad de los índices APS se refina a medida que se realizan más peticiones. Cada ruta de los walkers actualiza sus índices de acuerdo a los resultados de la búsqueda, cada nodo comparte y ajusta esta información a medida que pasa el tiempo.

¹⁷ Un súper nodo también es llamado centro, eje o hub.

¹⁸ Un nodo normal también es llamado hoja o leaf.

- **s-APS:** Es una modificación del método APS. 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. Estas ventajas se ven más claramente cuando muchos nodos diferentes contribuyen con su carga de trabajo. Este método tiene un número k como límite superior en sus hits por búsqueda.
- **Local Indices (LI):** 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. Para minimizar la sobrecarga, la distancia entre dos profundidades consecutivas debe ser: $2r + 1$. La eficacia de este método y los hits son muy altos, debido al sistema de indexación. Por otro lado, la producción de mensajes es semejante al sistema de inundación, incluso aunque el tiempo de procesamiento sea menor debido que mas nodos aceleran el proceso de búsqueda. Este método requiere una inundación con $TTL = r$, cada vez que un nodo se una o deje la red, o siempre que se actualice el repositorio local, por lo tanto, la sobrecarga es mucho más grande en ambientes dinámicos.
- **GIA:** 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. Este método requiere que cada nodo indexe los documentos de sus nodos vecinos. Este sistema también utiliza un algoritmo de adaptación a la topología, que reconfigura la conectividad de la red superpuesta, de tal manera que cada nodo está conectado a un número de nodos, proporcional a su capacidad. Los caminantes parciales se orientan hacia los vecinos altamente conectados, y probabilísticamente, a aquellos con el mayor número de elementos indexados. Finalmente, el sistema proporciona un mecanismo de control de flujo, que permite a los nodos controlar la velocidad a la que pueden aceptar y procesar las peticiones de sus nodos vecinos.
- **Distributed Resource Location Protocol (DRLP):** Los nodos que no tienen ninguna información sobre la localización de un objeto, transmiten la consulta, con una cierta probabilidad, a cada uno de sus nodos vecinos. Si se encuentra un objeto, la consulta toma la ruta inversa hacia el nodo solicitante, almacenando la ubicación del objeto en los nodos de esa ruta. En las solicitudes posteriores, los nodos con la información de la ubicación indexada, se conectan directamente con el nodo específico. Si ese nodo no tiene actualmente el documento, solo se inicia una nueva búsqueda, siguiendo el

proceso anterior. Este algoritmo utiliza inicialmente la inundación para encontrar la ubicación de un objeto, pero en solicitudes posteriores puede requerir de un solo mensaje para encontrar el objeto. Una pequeña producción de mensajes se logra solo con una gran carga de trabajo que permite que el costo inicial sea disminuido en muchas búsquedas. Este método afecta el número de hits de la siguiente manera: Si se hacen muchas búsquedas ciegas en el proceso, entonces se encuentran muchos resultados, pero si se hacen muchas peticiones directas, entonces se descubre una sola réplica del objeto.

- **Gnutella with Shortcuts (GS):** 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. Los nodos que responden satisfactoriamente son indexados por los nodos solicitantes, debido a que estos pueden dar respuestas satisfactorias a más peticiones. Cuando una nueva consulta se realice, los nodos enviarán la petición primero a sus accesos directos, clasificados en un orden descendente de utilidad, generalmente por tasas de éxito. Si fracasan todos los accesos directos, se utiliza de nuevo el sistema de inundación para localizar el objeto. Este enfoque es similar al método DRLP, pero almacena más de un puntero y mantiene estadísticas sobre ellos. Por otro lado, si los nodos hacen consultas que no estén relacionadas, o no se almacena el contenido relevante para lo que se propone, es posible que los accesos directos fallen, lo que significa que el sistema solo usará un sistema de inundación a gran escala. Lo mismo sucede cuando los objetos se eliminan o los nodos salen con frecuencia de la red.
- **FreeNet:** FreeNet es un Sistema P2P 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. Una consulta es básicamente una petición de una clave de un archivo binario, que también contiene un atributo para el contador de saltos (TTL). Una vez se recibe la consulta, el nodo vecino busca el recurso solicitado en su almacenamiento de datos local. Si no se encuentra el recurso solicitado, este busca en la tabla de enrutamiento la clave más cercana e identifica la dirección correspondiente, se decrementa el contador de saltos TTL y la consulta se envía a la dirección correspondiente. Este proceso se repite hasta que el recurso sea encontrado o hasta que el límite de saltos sea alcanzado. La principal ventaja de este método es que las consultas son dirigidas inteligentemente usando tablas de enrutamiento, así los nodos no tienen que procesar las consultas que no son necesarias, reduciendo el costo de ancho de banda, el costo de búsqueda y mejora la escalabilidad.

2.3. Búsqueda de Servicios en Sistemas Multi-Agente

La búsqueda de servicios más utilizada en Sistemas Multi-Agente es el uso del servicio de páginas amarillas, que provee el agente DF. El agente DF mantiene una lista precisa y completa de agentes, que provee la información actualizada acerca de los agentes, la descripción de sus servicios y su directorio de servicios, para todos los agentes autorizados en la plataforma [20].

La búsqueda de servicios, por medio del servicio de páginas amarillas, se hace de manera sintáctica, es decir, se usan palabras clave para realizar la búsqueda de servicios, requiriendo coincidencias exactas a nivel sintáctico entre las descripciones de los servicios y los parámetros empleados [20].

El servicio de páginas amarillas, permite a los agentes descubrir dinámicamente otros agentes, que se encuentran disponibles en la plataforma de agentes. Los agentes pueden publicar las descripciones de uno o más servicios que ellos proveen, para que otros agentes puedan fácilmente descubrir y explotar estos servicios. Cualquier agente puede publicar y buscar servicios. La publicación, eliminación, modificación y búsqueda de servicios, puede ser ejecutada en cualquier momento durante el tiempo de vida del agente [20].

2.4. Características de los Métodos de Búsqueda

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 búsqueda que se desea adecuar.

En [29] se describen las siguientes características que deben tenerse en cuenta para la búsqueda y el descubrimiento de servicios y recursos sobre redes superpuestas P2P:

- **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.

- **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. Esto quiere decir, que la consulta puede obtener resultados que no coincidan con lo que se está buscando. Cuando una coincidencia exacta es difícil de determinar, la precisión de la búsqueda puede ser evaluada por el grado de relevancia de los resultados obtenidos por la consulta.
- **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, como por ejemplo, por palabras clave; además debe proveer más flexibilidad al usuario para especificar una consulta.
- **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. En general, existen mecanismos de replicación y enrutamiento inteligente que permiten incrementar el desempeño de las operaciones de búsqueda.
- **Costo de la Búsqueda:** Para reducir el número de mensajes transmitidos entre los nodos, es necesario, reducir el número de nodos que procesan la consulta. Las consultas deben ser enviadas solo a los nodos que tengan un recurso que coincida con la búsqueda realizada, o a los nodos que puedan estar conectados a otros nodos que tengan el recurso solicitado. 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.
- **Costo de Ancho de Banda:** 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.
- **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.
- **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.
- **Confiabilidad:** La confiabilidad denota el éxito del mecanismo de búsqueda. Es difícil garantizar la confiabilidad de los métodos de búsqueda en sistemas descentralizados, debido a dos razones: Primero, no hay nodos centralizados que gestionen el contenido de los otros nodos, como resultado, los recursos solicitados pueden ser dirigidos de un modo arbitrario y nunca pueden ser alcanzados. Segundo, la disponibilidad del recurso puede encontrarse en peligro, debido a que los nodos entran y salen del sistema constantemente. La combinación de la tolerancia a fallos y la eficiencia del método de búsqueda determinan su confiabilidad.

2.5. Comparación de los Métodos de Búsqueda

A continuación se presentan varios trabajos en donde se muestran las comparaciones de varios métodos de búsqueda en redes superpuestas P2P.

En [30] se propone un algoritmo de búsqueda para redes superpuestas P2P no estructuradas, el cual consiste en mantener un cache de sus vecinos organizado de manera jerárquica, así como también mantener un cache de los hits de las consultas y de los resultados de la replicación de archivos en los otros nodos. Este método se compara con el algoritmo original de Gnutella, mostrando la eficiencia del nuevo método, para extender la región de búsqueda y reducir el tráfico efectivamente.

En [31] se muestra un método de búsqueda basado en el algoritmo original de Gnutella, donde se propone una solución basada en la localización de contenidos, en la cual los nodos se auto-organizan en una estructura basada en los intereses comunes de cada uno de ellos, sobre una red Gnutella existente. Este método mantiene los accesos directos de los nodos que comparten sus mismos intereses; debido a estos accesos directos la cantidad de mensajes por consulta disminuye. Este método se compara con el algoritmo de inundación original de Gnutella, donde se muestra que el nuevo método tiene mayor rendimiento en eficiencia, en carga y en longitud de saltos.

En otros trabajos [32] [33], se describen métodos para realizar búsqueda de contenidos sobre redes superpuestas P2P, estos métodos están basados en DHT y en el algoritmo de inundación original. Los resultados mostrados en estos trabajos analizan la eficiencia del método comparado con el algoritmo de inundación.

En la Tabla 1, se muestra una comparación de siete métodos de búsqueda donde se analizan las características de búsqueda que se describieron anteriormente [29].

Característica	Método de Búsqueda						
	Gnutella	Iterative Deepening	FreeNet	Int-BFS	Local Indices	GIA	APS
Control Local	Si	Si	Si	Si	Si	Si	Si
Precisión de la Búsqueda	5	5	6	6	6	5	6
Flexibilidad de la Búsqueda	1	1	1	1	3	3	1
Desempeño	1	2	3	3	4	3	4
Costo de la Búsqueda	6	5	3	3	3	2	3
Costo de Ancho de Banda	6	5	3	3	3	2	2
Costo de Almacenamiento	1	1	5	3	3	3	3
Tolerancia a Fallos	1	1	3	1	1	3	1
Escalabilidad	2	4	4	4	4	5	4
Confiabilidad	1	1	3	1	1	3	1

Tabla 1. Comparación de las características de siete métodos de búsqueda

Como puede verse en la comparación anterior, los métodos que más se destacan son: *GIA*, por su flexibilidad, tolerancia a fallos y escalabilidad; y *Local Indices*, por su precisión, flexibilidad y desempeño.

En [17] se realiza una comparación de varios métodos de búsqueda en redes superpuestas P2P no estructuradas y se analiza su rendimiento. La comparación de los métodos se centra en el comportamiento de los algoritmos para las siguientes métricas:

- Eficiencia en el descubrimiento de objetos, incluye precisión y número de hits.
- Consumo de ancho de banda.
- Adaptación al cambio de la topología de red.
- Sobrecarga.

Minimizar la producción de mensajes siempre representa una meta de alta prioridad para todos los sistemas distribuidos. Es importante que cualquier método de búsqueda se adapte a las condiciones de cambio, debido a que los usuarios de las redes superpuestas P2P entran y salen del sistema frecuentemente [17].

En la Tabla 2, se muestra el resumen de la comparación de nueve métodos de búsqueda, de acuerdo a las métricas descritas anteriormente [17].

Las topologías utilizadas en la comparación de los nueve métodos son las siguientes:

- **Random:** Topología de aproximadamente 10.000 nodos conectados aleatoriamente.
- **Power-Law:** Topología de aproximadamente 10.000 nodos conectados aleatoriamente. En esta topología los nodos son clasificados de acuerdo a su capacidad de conexión. Los nodos se conectan después de que son clasificados.
- **Gnutella:** La topología de red original de Gnutella. Contiene alrededor de 5.000 nodos conectados aleatoriamente que tienen una distribución Power-Law de dos segmentos.

Como puede verse en los resultados que muestra la tabla, los métodos que más se destacan son: *LI*, por su porcentaje de éxito y número de hits; *DRLP*, por su porcentaje de éxito y número de mensajes enviados por consulta; *GS*, por su porcentaje de éxito y número de hits; y *GIA*, por el número de mensajes enviados por consulta.

Topología	Métrica	Método								
		Mod-BFS	Int-BFS	LI	G2	RWalks	s-APS	DRLP	GS	GIA
Random	Éxito (%)	68,4	69,7	89,9	30,7	29,8	75,2	99,0	89,2	74,4
	Mensajes	118,8	115,4	1511,6	24,9	18,6	24,1	7,1	563,5	18,3
	Hits	2,3	2,4	37,7	0,5	0,4	2,2	1,2	5,0	3,2
Power-Law	Éxito (%)	56,8	62,3	93,3	76,7	22,9	75,7	98,3	88,4	85,7
	Mensajes	73,3	82,0	1473,0	750,3	13,1	15,1	5,0	355,9	19,1
	Hits	1,5	1,8	86,1	17,7	0,3	1,9	1,2	3,0	13,9
Gnutella	Éxito (%)	67,8	76,2	94,7	63,3	33,7	70,1	99,1	83,6	78,8
	Mensajes	145,6	217,4	1325,1	282,1	24,3	33,1	17,1	886,5	20,9
	Hits	2,6	4,4	59,8	5,7	0,5	3,0	2,0	15,3	6,0

Tabla 2. Comparación de las métricas de nueve métodos de búsqueda

En [13] se evalúan tres métodos de búsqueda eficientes en redes superpuestas P2P, utilizando una combinación de análisis y experimentos sobre la red Gnutella. Para evaluar estos métodos, se tienen en cuenta las siguientes métricas de costo y calidad de los resultados:

- **Costo:** Cuando un mensaje de consulta se propaga a través de la red, los nodos gastan recursos de procesamiento para transmitir la consulta y procesarla, y ancho de banda para enviar y recibir los mensajes. El costo principal de las consultas, se debe tanto al ancho de banda como al procesamiento. Como el costo de una consulta no se presenta en un único nodo de la red, tiene sentido hablar del costo total, sobre todos los nodos de la red. Además, no se debería evaluar una única consulta, si no que se debe medir el costo total promedio realizado por un conjunto de consultas.

Por lo tanto los dos indicadores de costo serian:

- Promedio de ancho de banda agregado.
- Promedio de costo de procesamiento agregado.

No es posible medir directamente estos indicadores, sino, que se utiliza un análisis basado en los costos medidos de las acciones individuales para estimar estos valores.

- **Calidad de los Resultados:** A pesar de que se busca reducir el costo de la consulta, se debe asegurar que la técnica no degrada la experiencia del usuario. La calidad de los resultados se mide por los siguientes indicadores:
 - **Número de resultados:** El tamaño del conjunto total de resultados.
 - **Satisfacción:** Algunas consultas pueden recibir cientos o miles de resultados. En lugar de notificar al usuario de cada resultado, se le notifica solo de los primeros z , donde z es un valor especificado por el usuario. La idea es que el usuario pueda encontrar lo que está buscando.

- **Tiempo de satisfacción:** Es el tiempo que ha transcurrido desde que la consulta es enviada por el usuario por primera vez, hasta cuando el usuario recibe el resultado número z.

En general, para que un método de búsqueda sea eficiente, debe observarse un equilibrio entre costo y calidad de los resultados.

En la Tabla 3, se muestra un resumen de los resultados de la evaluación de tres métodos de búsqueda, con los parámetros de costo y calidad, comparándolos con el método BFS, por lo tanto en la tabla los porcentajes de BFS son siempre 100% [13].

Método	Calidad			Costo	
	Tiempo de Satisfacción	Probabilidad de Satisfacción	Número de Resultados	Ancho de Banda Agregado	Procesamiento Agregado
BFS	100%	100%	100%	100%	100%
Iterative Deepening	190%	100%	19%	28%	47%
Intelligent-BFS	140%	86%	37%	38%	28%
Local Indices	100%	100%	100%	39%	51%

Tabla 3. Rendimiento relativo de tres métodos de búsqueda

Como puede verse en los resultados anteriores, el método que más se destaca es: *Local Indices*, por sus parámetros de calidad y costo, que son superiores a todos los demás.

2.6. Selección del Método de Búsqueda

Analizando las características de búsqueda y los resultados de las comparaciones, mostradas anteriormente, entre los diferentes métodos de búsqueda, y teniendo en cuenta que es mejor obtener una buena calidad de los resultados y un número de hits alto, sacrificando el costo de ancho de banda y procesamiento, se han elegido los siguientes métodos de búsqueda para ser utilizados en la adecuación de la Técnica de Búsqueda:

- **Local Indices:** Este método se escoge por el alto número de hits que se obtienen y por el alto porcentaje de éxito de la búsqueda. La deficiencia de este método es el alto número de mensajes producidos y su tolerancia a fallos.
- **Gnutella with Shortcuts (GS):** Al igual que LI, se escoge este método porque tiene un número de hits bastante alto, en relación con los demás, y su porcentaje de éxito es bastante alto también. La deficiencia de este método es el alto número de mensajes producidos.
- **Super-Peer Approaches - Gnutella2 (G2):** Se escoge este método porque a pesar que el número de hits obtenidos es muy bajo, el número de mensajes se reduce considerablemente. Por otro lado, este método se utiliza en el

descubrimiento de recursos por muchas tecnologías P2P, entre las cuáles se incluye JXTA [34].

Se eligen estos tres métodos porque cada uno de ellos, teóricamente hablando, puede eliminar las deficiencias de los otros dos. Sin embargo, cabe mencionar que de una u otra forma, los métodos pueden ser modificados para ajustarse a los requerimientos de la técnica propuesta, pero siempre conservando su idea central.

2.7. Priorización de las Características de los Métodos de Búsqueda

De acuerdo a la comparación y selección de los métodos de búsqueda y las características de cada uno de ellos, y teniendo en cuenta que es mejor obtener una buena calidad de los resultados y un número de hits alto, sacrificando el costo de ancho de banda y procesamiento, se han priorizado las características de búsqueda, las cuales se muestran a continuación:

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

La descripción de cada una de estas características, se muestra en la Sección 2.4.

CAPÍTULO III. TÉCNICA DE BÚSQUEDA PARA LA PRESTACIÓN DE SERVICIOS

3.1. Introducción

Después de investigar los diferentes métodos y técnicas de búsqueda usadas en las redes superpuestas P2P no estructuradas y definir sus características más importantes, se pasa a describir la adecuación de la Técnica de Búsqueda.

La mayor parte de la solución del problema se centra en las características que pueden brindar los Sistemas Multi-Agente para la publicación, localización y búsqueda de los servicios. Como parte de la solución también se usan las características de transporte y comunicación de las redes superpuestas P2P.

Se debe tener en cuenta que para modelar y acoplar los servicios al entorno de trabajo, hay que establecer un canal de comunicación entre los dos sistemas, el Sistema Multi-Agente y el Sistema P2P.

Cada componente de la solución tiene como principal característica, la independencia de su funcionamiento, por lo tanto, es deseable mantener esta independencia en la adecuación de la técnica de búsqueda.

3.2. Arquitectura del Sistema

Después de describir los beneficios ofrecidos de los Sistemas Multi-Agente y los sistemas P2P, se puede pensar en conectar dichos sistemas, por medio de un mecanismo que aproveche en gran medida sus recursos. Lo importante de este mecanismo consiste en poder ofrecer una comunicación transparente al usuario, para que no note la integración entre los dos sistemas, es aquí donde se propone crear un modulo llamado *Comunicador* que tiene como objetivo principal brindar la comunicación entre el *Sistema Multi-Agente* y el *Sistema P2P*.

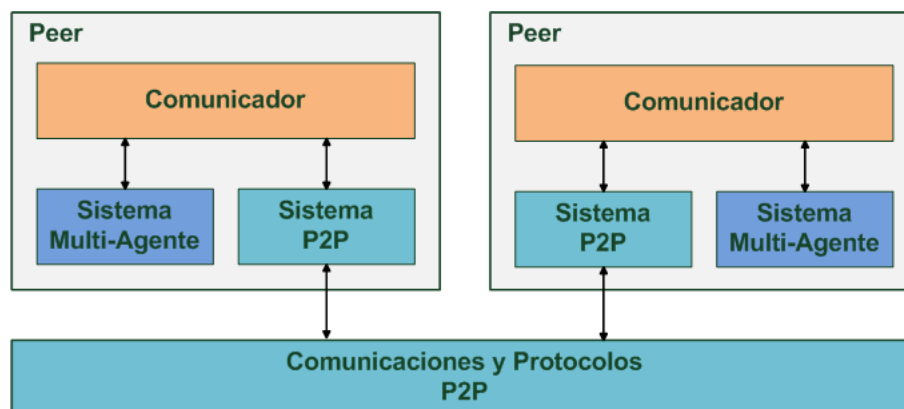


Figura 2. Arquitectura del Sistema

En la Figura 2, se muestra la arquitectura general del sistema, la cual contiene los tres elementos principales del sistema: El *Sistema Multi-Agente*, El *Sistema P2P* y El *Comunicador* que están presentes en cada nodo de la red superpuesta P2P.

3.2.1. Sistema Multi-Agente

El *Sistema Multi-Agente* cumple con las especificaciones FIPA e incluye todos los elementos obligatorios para administrar la Plataforma de Agentes (AP), los cuales son el Sistema de Gestión de Agentes (AMS), el Facilitador de Directorio (DF) y el Servicio de Transporte de Mensajes (MTS). Toda la comunicación de agentes se realiza a través del paso de mensajes, donde FIPA-ACL es el lenguaje para representar los mensajes [20]. En la Figura 3, se muestran los componentes del Sistema Multi-Agente.

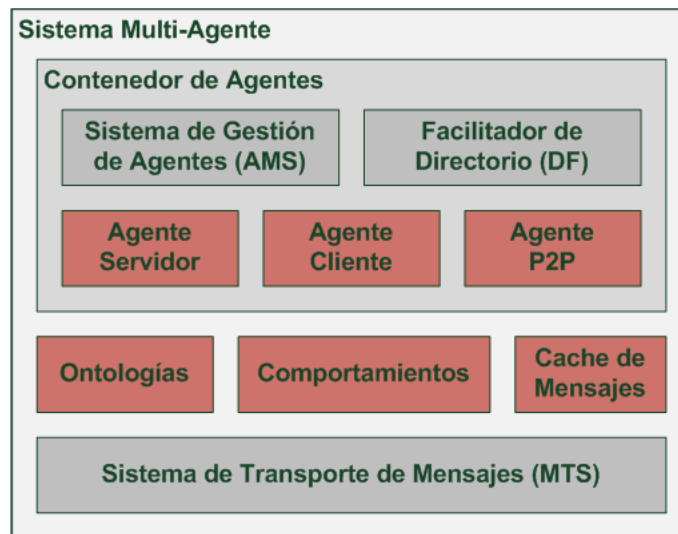


Figura 3. Componentes del Sistema Multi-Agente

El *Sistema Multi-Agente* maneja un solo contenedor de agentes, el contenedor principal. Debido a que cada nodo de la red va a tener su propia plataforma de agentes, solo se necesita un contenedor, el cual será el encargado de gestionar todos los agentes del sistema.

En el contenedor principal están presentes los respectivos agentes del sistema, los cuales se describen a continuación:

- **Agente-Servidor:** El agente que se encarga de la provisión del servicio. Este agente debe registrar el servicio en el agente DF y contiene todos los elementos necesarios para proveer el servicio. Este servicio será el que se publique posteriormente en la red superpuesta P2P.

- **Agente-Cliente:** El agente que consume el servicio publicado por el *Agente-Servidor*. Este agente debe registrarse en el DF, debido a que prestará servicios locales en la plataforma de agentes.
- **Agente-P2P:** Se encarga de recibir los mensajes de los agentes *Servidor* y *Cliente* y transmitirlos al módulo de comunicaciones (*Comunicador*), para que sean enviados posteriormente a los demás nodos a través de la red superpuesta P2P. Este agente debe registrarse en el DF, debido a que prestará servicios locales en la plataforma de agentes. Este agente tiene la particularidad de permitir la comunicación no solo entre agentes, sino también hacia y desde objetos.

Los agentes descritos anteriormente son los agentes básicos que debe tener el sistema. Adicionalmente, pueden crearse los agentes que se consideren convenientes, como por ejemplo, un agente que sea el encargado de gestionar el paso de mensajes hacia y desde la interfaz gráfica.

3.2.2. Sistema P2P

El *Sistema P2P* provee un conjunto común de protocolos para el desarrollo de aplicaciones P2P. En la Figura 4, se muestran los componentes del Sistema P2P.

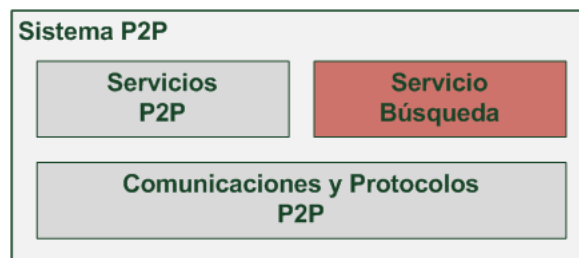


Figura 4. Componentes del Sistema P2P

Es el encargado de crear, publicar y buscar los respectivos advertisements¹⁹, que representan el servicio en la red superpuesta P2P. También se encarga de las comunicaciones y del envío y recepción de mensajes entre cada nodo de la red superpuesta P2P, que se realizan por medio de los diferentes protocolos P2P.

3.2.3. Comunicador

El *Comunicador*, como su nombre lo indica, es el encargado de las comunicaciones entre las dos plataformas y realiza la tarea de conversión entre los tipos de mensajes, los tipos de objetos y los parámetros de los respectivos métodos que cada una de las plataformas maneja. En la Figura 5, se muestran los componentes del Comunicador.

¹⁹ Son documentos XML, que son utilizados para describir recursos de la red superpuesta P2P.



Figura 5. Componentes del Comunicador

En la siguiente sección se describirán más específicamente los componentes de la técnica.

3.3. Descripción de la Técnica de Búsqueda

Para describir la técnica de búsqueda, es necesario describir los comportamientos de cada uno de los agentes que están presentes dentro del sistema, como se realiza la comunicación entre agentes y las ontologías utilizadas, las interacciones que se presentan entre los agentes, cómo se realiza la conversión de mensajes en el *Comunicador* y por último describir como se envían los mensajes a través de la red superpuesta P2P.

3.3.1. Comportamientos de los Agentes

3.3.1.1. Comportamientos del Agente Servidor

- **Iniciar Servidor:** Está compuesto de varios sub-comportamientos: Buscar Agente-P2P, Crear Grupo P2P, Guardar Grupo P2P y Publicar Servicio. Estos comportamientos deben ejecutarse de manera secuencial y ordenada, para garantizar que el servidor se inicie de una manera correcta.
- **Buscar Agente-P2P:** Este comportamiento es el encargado de buscar en el registro del DF el *Agente-P2P*. Si no se encuentra registrado el *Agente-P2P* en el DF, este comportamiento permanecerá activo hasta que el *Agente-P2P* se registre y pueda ser encontrado. El comportamiento debe permanecer activo buscando el *Agente-P2P*, debido a que sin este agente serian imposibles las comunicaciones a través de la red superpuesta P2P, lo que implicaría que el servicio no podría ser publicado y por lo tanto no podría prestarse el servicio en la red.
- **Crear Grupo P2P:** Este comportamiento crea un grupo P2P 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 P2P:** El proceso de creación de un grupo en una red superpuesta P2P lleva consigo la asignación de un ID único para cada grupo. Por lo tanto, cada nodo de la red debe guardar este ID, para un uso futuro de

los servicios dentro del grupo al cual se conecta. Este comportamiento guarda la identificación del grupo.

- **Publicar Servicio:** Este comportamiento registra el servicio en el agente DF, dentro del contenedor de agentes. Posteriormente, envía un mensaje FIPA al *Agente-P2P*, para publicar el servicio en la red superpuesta P2P.
- **Recibir Mensajes:** Este comportamiento es el encargado de recibir los mensajes FIPA que llegan al agente y procesarlos. Los mensajes se seleccionan dependiendo del tipo de mensaje FIPA o de la ontología utilizada y del tipo de predicado o acción a ejecutar. El comportamiento siempre debe estar activo esperando la recepción de mensajes.
- **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 el envío de mensajes posteriormente.

3.3.1.2. Comportamientos del Agente Cliente

- **Iniciar Cliente:** Está compuesto de varios sub-comportamientos: Buscar Agente-P2P, Crear Grupo P2P, Guardar Grupo P2P, Publicar Servicio e Iniciar Conexión al Servidor. Estos comportamientos deben ejecutarse de manera secuencial y ordenada, para garantizar que el cliente se inicie de una manera correcta.
- **Publicar Servicio:** Este comportamiento registra el servicio en el agente DF, dentro del contenedor de agentes. Posteriormente, envía un mensaje FIPA al *Agente-P2P* para crear un ID único, para el nodo en la red superpuesta P2P.
- **Iniciar Conexión al Servidor:** Este comportamiento inicia la conexión del nodo cliente, hacia un nodo que este prestando el servicio. Está compuesto de varios sub-comportamientos: Buscar Nodo, Guardar Nodo y Conectar al Nodo. Estos comportamientos deben ejecutarse de manera secuencial y ordenada, para garantizar que el cliente se conecte al servidor de una manera correcta.
- **Buscar Nodo Servidor:** Es el comportamiento encargado de buscar el nodo que presta el servicio deseado. El servicio es buscado por su nombre, y se crea un mensaje FIPA, para ser enviado al *Agente-P2P*, el cual es el encargado de redirigir la búsqueda a la red superpuesta P2P.
- **Guardar Nodo Servidor:** Después de buscar y encontrar el servicio deseado, este comportamiento se encarga de guardar el acceso directo del nodo que presta el servicio, para su posterior uso. Gracias a este acceso directo, la

próxima vez que el cliente requiera el uso del servicio no necesitará buscar de nuevo un nodo servidor, solo usará el acceso directo para hacer uso del servicio y así evitar una nueva búsqueda en la red superpuesta P2P.

- **Conectar al Nodo Servidor:** Este comportamiento envía el ID del nodo cliente, y una lista de sus recursos disponibles hacia el nodo servidor, si los tiene, para que sea almacenado en el servidor a manera de índices locales, para así, facilitar la localización de recursos y el envío de mensajes.

Los comportamientos Buscar Agente-P2P, Crear Grupo P2P, Guardar Grupo P2P y Recibir Mensajes, realizan la misma función que los descritos en el *Agente-Servidor*.

3.3.1.3. Comportamientos del Agente P2P

- **Buscar Agentes:** Este comportamiento es el encargado de buscar, en el registro del DF, el *Agente-Cliente* o el *Agente-Servidor*, según el tipo de nodo, ya sea un nodo cliente o un nodo servidor. Si no se encuentra registrado alguno de estos agentes en el DF, este comportamiento permanecerá activo hasta que los agentes se registren y puedan ser encontrados. El comportamiento debe permanecer activo, buscando a los agentes, debido a que sin estos serían imposible realizar la creación, búsqueda y recuperación del servicio, y las comunicaciones entre el *Sistema Multi-Agente* y el *Sistema P2P*.
- **Recibir Mensajes:** Este comportamiento es el encargado de recibir los mensajes FIPA, que llegan al agente y procesarlos. Este comportamiento es el encargado de recibir los mensajes FIPA que llegan al agente y procesarlos. Los mensajes se seleccionan dependiendo del tipo de mensaje FIPA o de la ontología utilizada y del tipo de predicado o acción a ejecutar. El comportamiento siempre debe estar activo esperando la recepción de mensajes. El comportamiento siempre debe estar activo, esperando la recepción de mensajes. Cuando el agente recibe los mensajes FIPA, este los redirige a los objetos presentes en el *Comunicador*, para que sean procesados, convertidos y enviados luego hacia el sistema y la red superpuesta P2P.
- **Recibir Mensajes de Objetos:** Al contrario del comportamiento “Recibir Mensajes”, este recibe los mensajes desde los objetos presentes en el *Comunicador* y los reenvía hacia los agentes, encapsulando cada mensaje en un mensaje FIPA, que es el estándar de comunicación entre agentes. Dependiendo del tipo de mensaje, de la ontología usada y del predicado o acción a ejecutar, el *Agente-P2P* sabe a qué agente debe ser redirigido cada mensaje.

3.3.2. Comunicación entre Agentes y Ontologías

Para que los agentes se comuniquen de una mejor manera, es necesario que cada uno de los mensajes enviados maneje un lenguaje común, que pueda ser usado para construir conversaciones más complejas; para este propósito es necesario utilizar las siguientes características [20]:

- Los protocolos de interacción FIPA.
- El lenguaje de comunicación FIPA-ACL.
- El lenguaje de contenido FIPA-SL, para expresar los mensajes, debido a que este ha sido promovido como un estándar.
- El uso de ontologías, para estructurar los mensajes y mejorar la comunicación.

Las ontologías son un método para representar e interrelacionar objetos o conceptos específicos, y son de gran uso para estructurar los mensajes FIPA-ACL. Por lo tanto, se deben crear ontologías para describir mejor los mensajes que van a ser utilizados en la comunicación de los agentes, dentro de la técnica de búsqueda.

Mediante el uso de ontologías se incorpora contenido semántico, y no sólo datos, para representar todos los conceptos relacionados con un tema o dominio específico; así como también para representar las relaciones, funciones y axiomas entre estos conceptos mediante los diferentes predicados.

A continuación, se presentan las ontologías utilizadas en la técnica de búsqueda.

3.3.2.1. Ontología Servicio

Esta ontología representa todos los conceptos y predicados, que son usados para la publicación, búsqueda y prestación del servicio. La Figura 6, muestra los elementos de la ontología servicio.

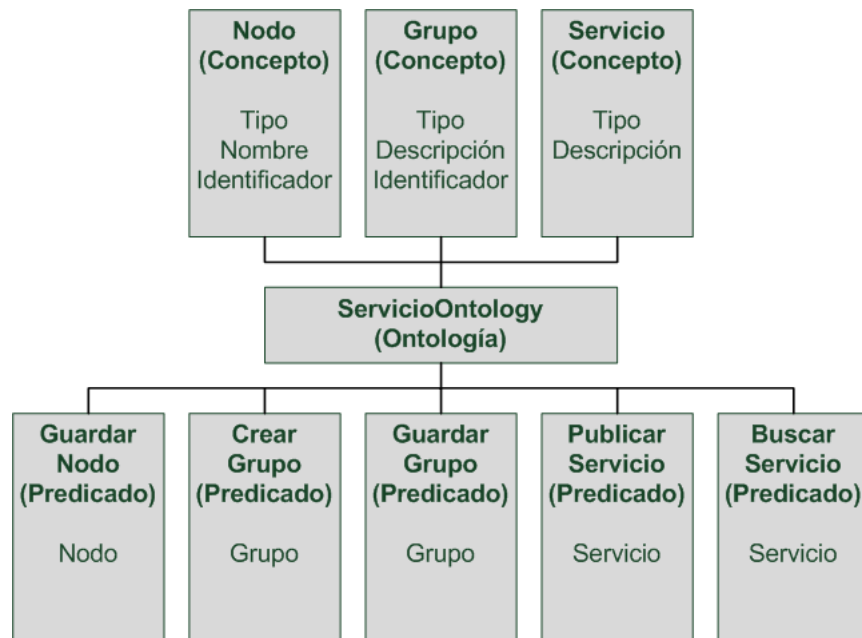


Figura 6. Ontología Servicio

❖ **Conceptos:**

- **Nodo:** Representa un nodo dentro del sistema.
- **Grupo:** Representa un grupo dentro del sistema, específicamente un grupo en la red superpuesta P2P.
- **Servicio:** Representa el tipo y la descripción del servicio dentro del sistema.

❖ **Predicados:**

- **Guardar Nodo:** Representa la situación de guardar un nodo que presta el servicio, es decir un nodo servidor.
- **Crear Grupo:** Representa la situación de crear un grupo en la red superpuesta P2P.
- **Guardar Grupo:** Representa la situación de guardar un grupo P2P, que ha sido creado previamente.
- **Publicar Servicio:** Representa la situación de publicar el servicio que se desea prestar.
- **Buscar Servicio:** Representa la situación de buscar el servicio que se desea utilizar.

3.3.2.2. Ontología Conexión

Esta ontología representa todos los conceptos y predicados, que son usados para la búsqueda y conexión de nodos en la red superpuesta P2P. La Figura 7, muestra los elementos de la ontología conexión.

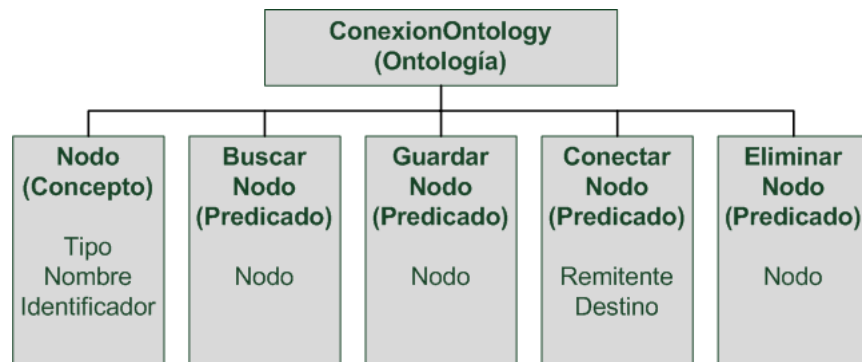


Figura 7. Ontología Conexión

❖ **Conceptos:**

- **Nodo:** Representa un nodo dentro del sistema.

❖ **Predicados:**

- **Buscar Nodo:** Representa la situación de buscar un nodo dentro del sistema.
- **Guardar Nodo:** Representa la situación de guardar un nodo, que ha sido encontrado previamente.
- **Conectar Nodo:** Representa la situación de conectarse a un nodo, que ha sido encontrado y guardado previamente. Los campos Remitente y Destino, son conceptos de tipo Nodo.
- **Eliminar Nodo:** Representa la situación de eliminar un nodo, que por algún motivo no ha podido ser alcanzado.

3.3.2.3. Ontología Mensaje

Esta ontología representa todos los conceptos y predicados, que son usados para el envío y recepción de mensajes. La Figura 8, muestra los elementos de la ontología mensaje.

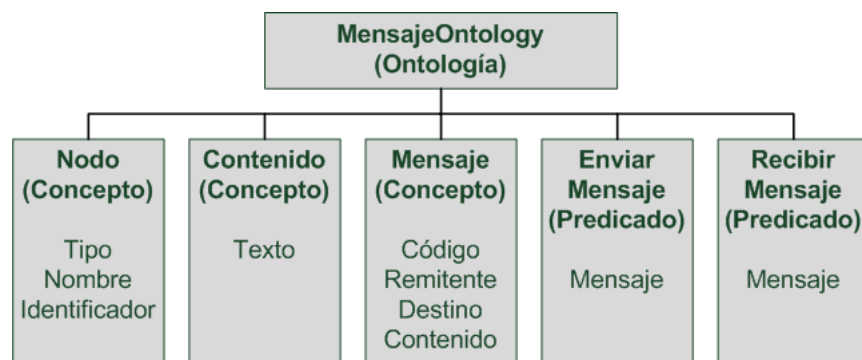


Figura 8. Ontología Mensaje

❖ **Conceptos:**

- **Nodo:** Representa un nodo dentro del sistema.
- **Contenido:** Representa el contenido de un mensaje. Este concepto solo contiene un campo de texto, pero pueden agregarse más campos según se necesite.
- **Mensaje:** Representa el mensaje que será utilizado en la técnica. El campo Código es un identificador único para cada mensaje, que puede ser una cadena de texto. Los campos Remitente y Destino, son conceptos de tipo Nodo.

❖ **Predicados:**

- **Enviar Mensaje:** Representa la situación de enviar un mensaje a través del sistema.
- **Recibir Mensaje:** Representa la situación de recibir un mensaje que fue enviado previamente.

3.3.2.4. Ontología Búsqueda

Esta ontología representa todos los conceptos y predicados, que son usados para la búsqueda y recuperación del servicio, específicamente para el envío del directorio de recuperación, lo que la diferencia de la Ontología Mensaje. La Figura 9, muestra los elementos de la ontología búsqueda.



Figura 9. Ontología Búsqueda

❖ **Conceptos:**

- **Nodo:** Representa un nodo dentro del sistema.
- **Contenido:** Representa el contenido de un mensaje. Este concepto solo contiene un campo de texto, pero pueden agregarse más campos según se necesite.
- **Mensaje:** Representa el mensaje que será utilizado en la técnica. El campo Código es un identificador único para cada mensaje, que puede ser una

cadena de texto. Los campos Remitente y Destino, son conceptos de tipo Nodo.

- **Directorio:** Representa el directorio de recursos o recuperación de mensajes, que es enviado a un nodo servidor para recuperar el estado de los mensajes de un nodo cliente. El campo Servidor es un concepto de tipo Nodo, mientras que el campo Mensaje es una lista de conceptos de tipo Mensaje.

❖ **Predicados:**

- **Enviar Directorio:** Representa la situación de enviar el directorio de recursos o recuperación a un nodo servidor, para que este sea procesado. El campo Destino es un concepto de tipo Nodo.
- **Recibir Directorio:** Representa la situación de recibir el directorio de recursos o recuperación que fue enviado previamente, para que este sea procesado. El campo Destino es un concepto de tipo Nodo.

3.3.3. Envío de Mensajes

El envío de mensajes entre agentes se realiza por medio de las especificaciones FIPA y los mensajes FIPA-ACL. Cada mensaje debe utilizar una de las ontologías que se describieron anteriormente, debe utilizar los conceptos y predicados descritos en cada una de ellas. Gracias a las ontologías se agrega contenido semántico a los mensajes, lo que permite a los agentes una comunicación más clara y un mejor entendimiento de los mensajes.

Una de las desventajas de los sistemas P2P, es que no se han implementado estándares para la comunicación y el envío de mensajes, por lo tanto cada una de las implementaciones posee su propia serie de protocolos P2P para este propósito [5]. Debido a este problema, la comunicación entre los nodos de la red superpuesta P2P y el envío de mensajes dependerá en gran medida, de la implementación usada y de los protocolos que utilice esta implementación.

El problema de la comunicación entre nodos de la red superpuesta P2P, se soluciona en gran parte con el uso del *Sistema Multi-Agente*; el cual nos permite manejar un estándar para la especificación de servicios y las comunicaciones. Esta solución puede verse en las siguientes secciones.

3.3.4. Interacción entre los Agentes y el Mundo Exterior

La interacción entre agentes es una parte fundamental del *Sistema Multi-Agente*, esta interacción hace uso del intercambio de mensajes FIPA-ACL entre los agentes y el uso de las ontologías que se describieron anteriormente. Pero, como el *Sistema Multi-Agente* debe poder comunicarse con el mundo exterior, es

conveniente crear una jerarquía de comunicación entre los agentes, para que solo uno de ellos gestione las comunicaciones hacia el mundo exterior. En el caso del sistema propuesto, el mundo exterior será el *Comunicador* y todos los objetos que estén presentes dentro de él.

En la Figura 10, se muestra la interacción entre los agentes del sistema y el *Comunicador*.

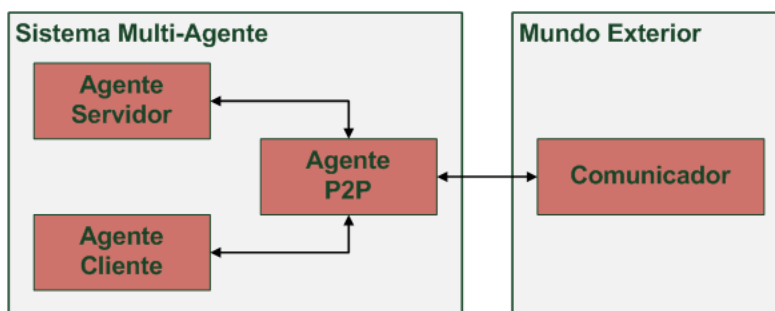


Figura 10. Interacción entre los Agentes y el Comunicador

Dentro del *Sistema Multi-Agente*, los agentes se comunican entre ellos mediante mensajes FIPA-ACL. Por otro lado, el único agente con la capacidad para comunicarse con el mundo exterior es el *Agente-P2P*. Esto significa que si un agente desea enviar un mensaje hacia el exterior, el mensaje deberá ser dirigido por medio del *Agente-P2P*, lo que conlleva a tener un punto central de comunicaciones hacia el exterior, que facilita el mantenimiento y la administración de las comunicaciones hacia y desde el *Sistema Multi-Agente*.

El *Agente-P2P* recibe los mensajes FIPA-ACL de los otros agentes, y extrae el contenido de cada uno de ellos, seleccionando el predicado para ejecutar el método correcto en el *Comunicador*, y seleccionando los conceptos para utilizarlos como parámetros de estos métodos. Por otro lado, el *Comunicador* convierte los parámetros recibidos desde el *Sistema P2P* y crea los conceptos necesarios para enviárselos al *Agente-P2P*, con los cuales, el agente crea los predicados correctos y se los envía a los agentes respectivos. En la siguiente sección se describe este proceso con más detalle.

3.3.5. Conversión de Mensajes

El *Comunicador* cumple con la función de realizar la conversión entre los tipos de mensajes, que se envían entre los dos sistemas, para que cada uno cumpla con las características específicas de cada plataforma. En la Figura 5, se muestran los componentes del *Comunicador*.

El *Comunicador* contiene dos módulos básicos para realizar la conversión entre mensajes: El *Transmisor* y el *Receptor*.

3.3.5.1. Transmisor

Es el encargado de realizar la conversión de mensajes FIPA-ACL, que salen desde el *Sistema Multi-Agente*, al tipo de mensajes utilizados en el *Sistema P2P*.

Como se mostró en la sección anterior, el *Agente-P2P* le envía al *Comunicador* los conceptos, para que este los procese y los envíe hacia el *Sistema P2P* correctamente.

Para mostrar este proceso, se muestra el siguiente ejemplo: El *Agente-Servidor* desea publicar el servicio, por lo tanto usa la ontología servicio, específicamente el concepto "Servicio" y el predicado "Publicar Servicio", y los envía al *Agente-P2P* por medio de un mensaje FIPA-ACL. El *Agente-P2P* recibe el mensaje, mediante la extracción del predicado "Publicar Servicio", el agente invoca al método *publicarServicioP2P()* del *Comunicador*, pasándole como parámetro el concepto "Servicio". El *Transmisor* procesa el concepto, extrayendo sus campos: el tipo y la descripción del servicio, enviándolos al *Sistema P2P* para que se publique el servicio con los parámetros establecidos.

3.3.5.2. Receptor

Es el encargado de realizar la conversión del tipo de mensajes que llegan desde el *Sistema P2P*, a mensajes FIPA-ACL, que son los utilizados en el *Sistema Multi-Agente*.

Como se mostró en la sección anterior, el *Sistema P2P* le envía al *Comunicador* los parámetros, para que este los procese, cree los conceptos adecuados y los envíe hacia el *Agente-P2P*.

Para mostrar este proceso, se muestra el siguiente ejemplo: Un nodo servidor recibe una petición de conexión de un nodo cliente, por lo tanto el nodo servidor debe guardar la referencia del nodo cliente. El *Sistema P2P* llama al método *guardarNodo()* del *Comunicador* y le pasa tres parámetros: El tipo, el nombre y el identificador del nodo. El *Receptor* usa estos tres parámetros para crear el concepto "Nodo" y el predicado "Guardar Nodo" de la ontología Conexión y los envía al *Agente-P2P*, gracias a la característica de intercambio de objetos que tiene este agente. El *Agente-P2P* recibe el predicado como un objeto y lo usa para crear un mensaje FIPA-ACL, que posteriormente envía al *Agente-Servidor* para que guarde la referencia al nodo cliente.

3.4. Prestación de Servicios

Después de describir cada uno de los componentes de la Técnica de Búsqueda, se muestra cómo se publica el servicio, cómo se realiza la búsqueda del servicio y cómo se realiza la recuperación del servicio.

Antes de pasar a describir cada proceso por separado, se mostrarán los diagramas de flujo que describen cada uno de ellos.

3.4.1. Publicación del Servicio

La creación del servicio, es el proceso mediante el cual, un agente publica el servicio en el sistema.

En la Figura 11, se muestra el diagrama de flujo del proceso de publicación del servicio por parte de un nodo servidor.

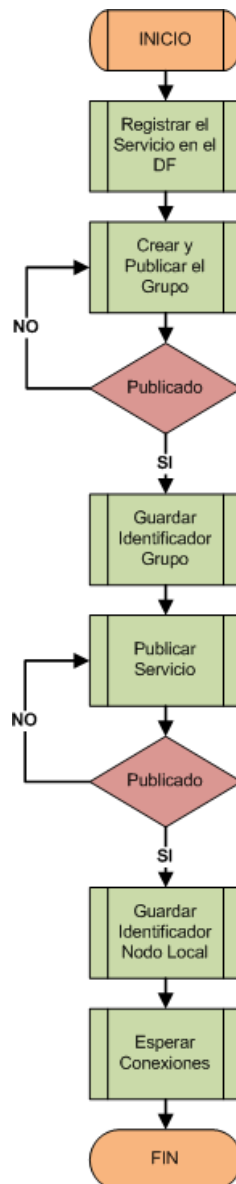


Figura 11. Diagrama de Flujo - Publicación del Servicio

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 12, muestra la secuencia de pasos necesarios para registrar el servicio en el Agente DF. Todos los agentes que presten un servicio local dentro del *Sistema Multi-Agente* deben registrarse en el DF. El agente debe crear el concepto "Servicio" mediante los campos tipo y descripción, y posteriormente registrar el servicio en el DF.

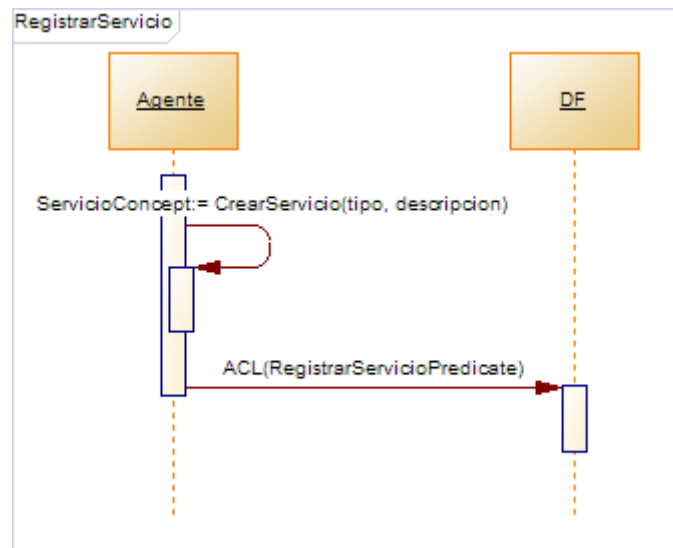


Figura 12. Registrar un Servicio

Después de registrarse en el DF, cada agente 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. Es necesario guardar el identificador del grupo que se crea, por este motivo, luego de una publicación exitosa del grupo en la red superpuesta P2P, el agente debe guardar el respectivo identificador para ser utilizado posteriormente.

El siguiente paso es la publicación del servicio en la red superpuesta P2P, haciendo uso del *Sistema P2P*. Para este propósito se utiliza el grupo que se creó y guardó anteriormente. La Figura 14, muestra la secuencia de pasos necesarios para realizar la publicación del servicio.

Si el agente está prestando un servicio, es decir es un *Agente-Servidor*, el servicio debe publicarse en la red superpuesta P2P, mediante el método *PublicarServicio(identificador)*; si por el contrario es un agente que desea utilizar un servicio, es decir un *Agente-Cliente*, no se llama al método *PublicarServicio(identificador)*, para que no se publique el servicio en la red superpuesta P2P, pero se deben seguir todos los demás pasos para poder

obtener el identificador del nodo que el agente está representando en la red superpuesta P2P. Es necesario que cada agente guarde el identificador del nodo que representa en la red superpuesta P2P, esto se hace mediante la creación de un concepto "Nodo" en cada agente.

Después de la publicación del servicio en la red superpuesta P2P, el *Agente-Servidor* ya está en la capacidad de recibir conexiones desde los clientes y prestar el servicio.

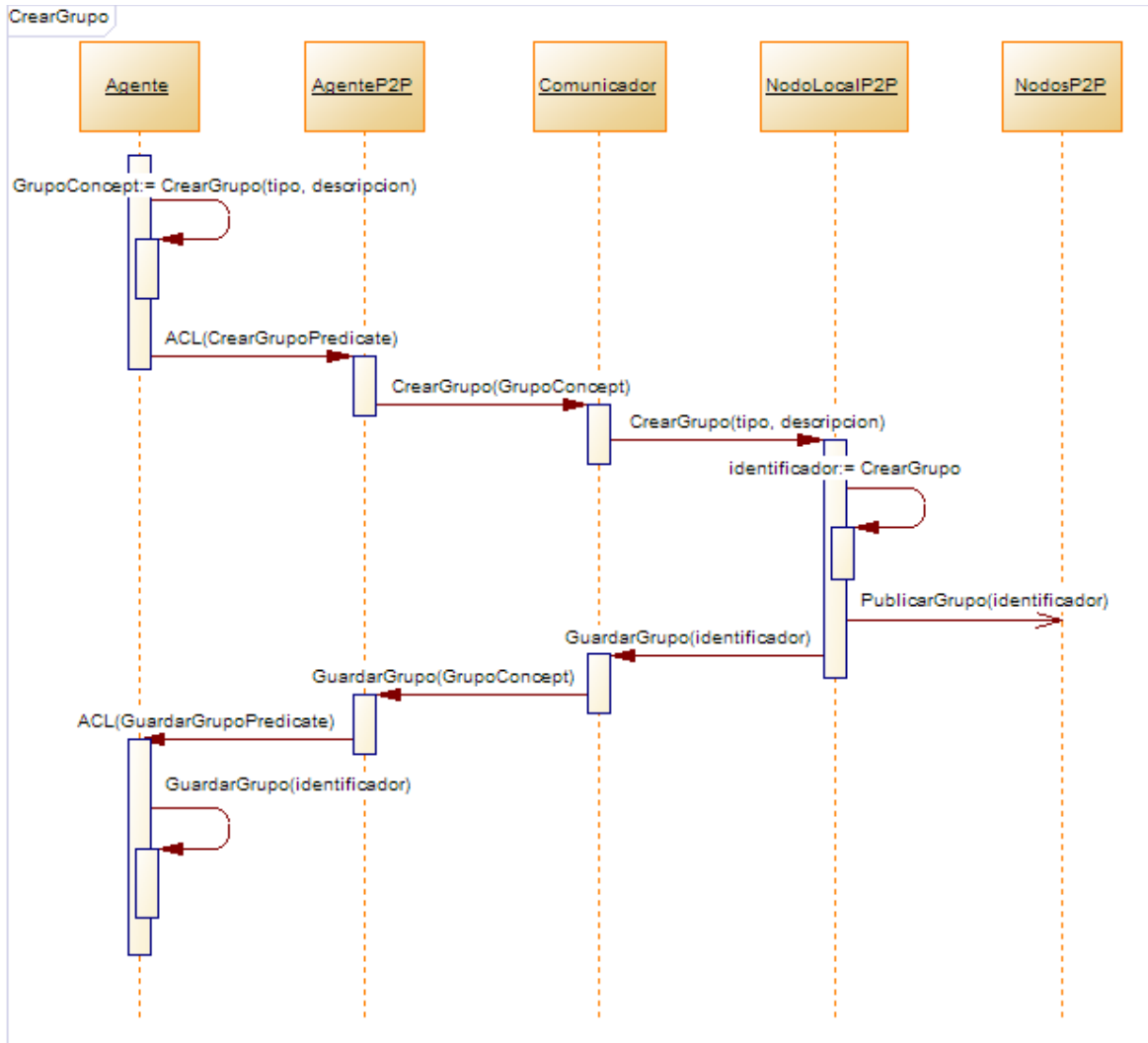


Figura 13. Crear un Grupo para el Servicio

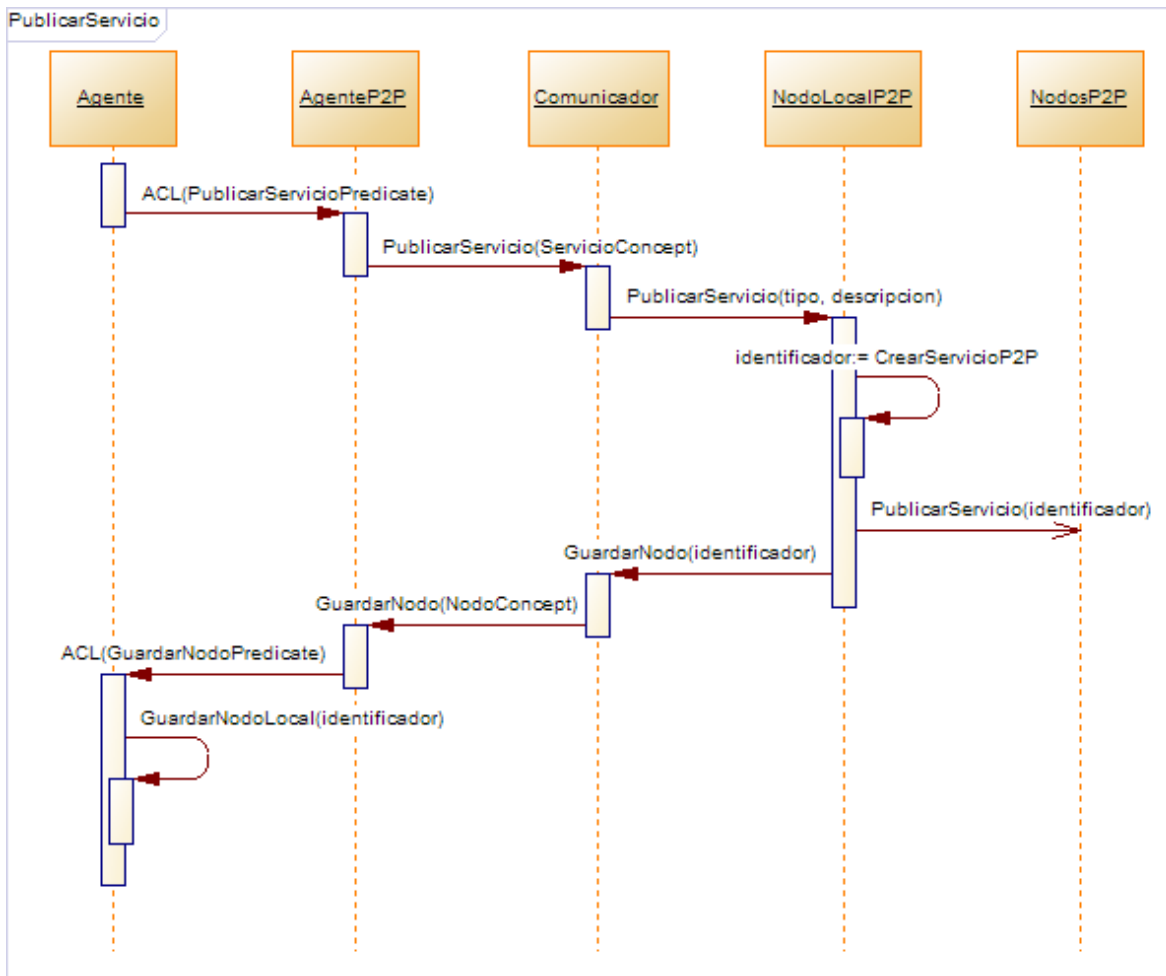


Figura 14. Publicar un Servicio

3.4.2. Búsqueda del Servicio

La búsqueda del servicio, es el proceso mediante el cual, un agente busca el servicio requerido en el sistema.

La recuperación del servicio, es el proceso mediante el cual, un agente que ha perdido la conexión a un servidor, vuelve a conectarse, sin perder los datos del proceso que estaba realizando.

En la Figura 15, se muestra el diagrama de flujo del proceso de búsqueda y recuperación del servicio por parte de un nodo cliente.

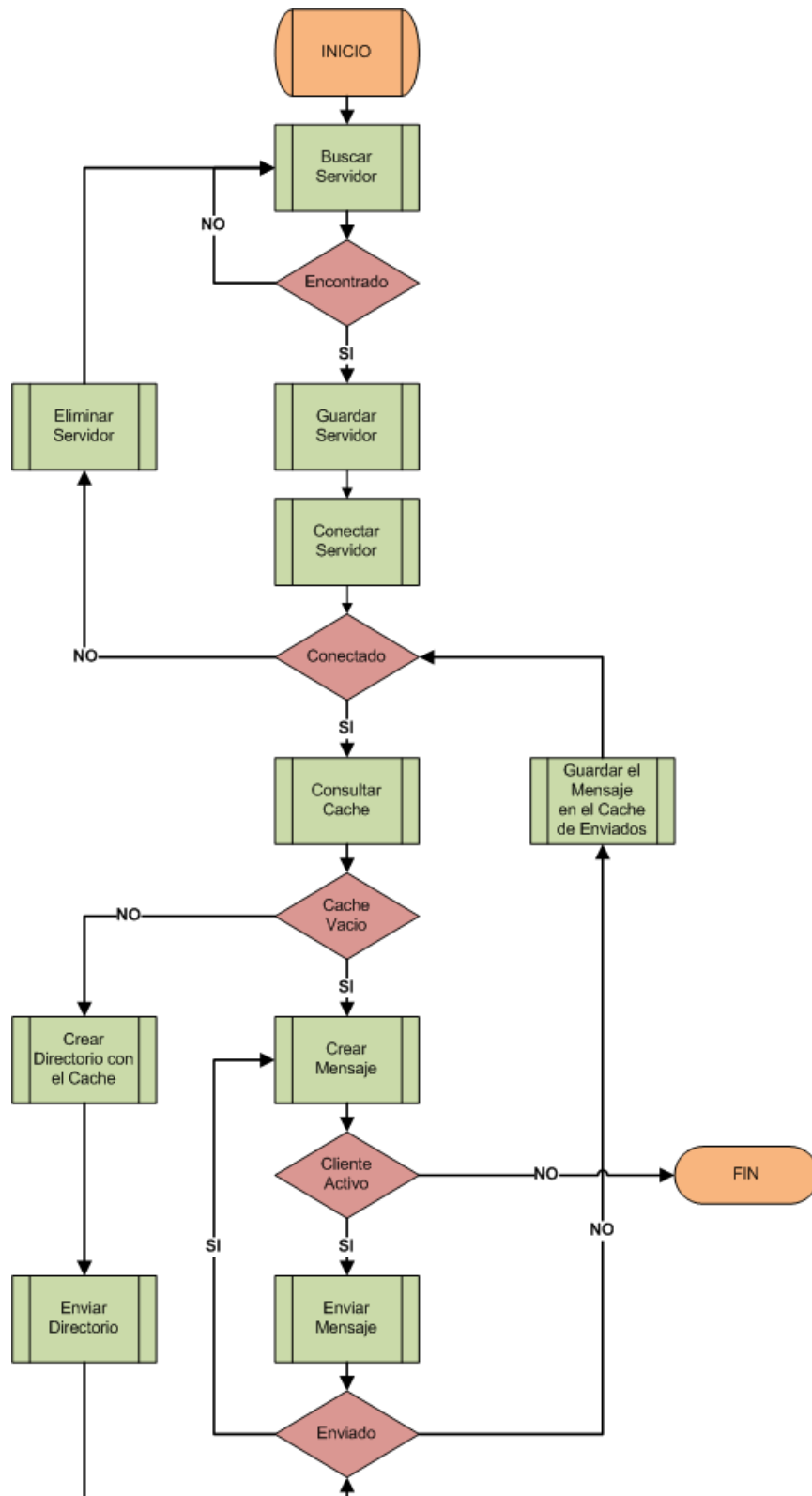


Figura 15. Diagrama de Flujo - Búsqueda y Recuperación del Servicio

Cuando un servicio se publica en la red superpuesta P2P, un *Agente-Cliente* puede buscarlo y utilizarlo si lo necesita. La Figura 16, muestra la secuencia de pasos para buscar un nodo servidor. El *Agente-Cliente* busca repetitivamente un nodo servidor, hasta que pueda encontrarlo. Si se encuentra un nodo servidor, el *Sistema P2P* enviará un mensaje con el identificador del nodo encontrado, para que una referencia a este nodo sea guardada en el *Agente-Cliente* y se use como Acceso Directo para futuras comunicaciones.

Es importante notar que la búsqueda del nodo servidor en la red superpuesta P2P se hace por medio de los protocolos de la implementación específica de la red superpuesta P2P 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 P2P.

El *Agente-Cliente* guarda la identificación del nodo servidor como un acceso directo, debido a que este puede dar una respuesta satisfactoria a las peticiones que se realicen posteriormente. Este método también permite enviar las nuevas consultas a los accesos directos de primero, lo que evita una nueva búsqueda de un nodo servidor en la red superpuesta P2P.

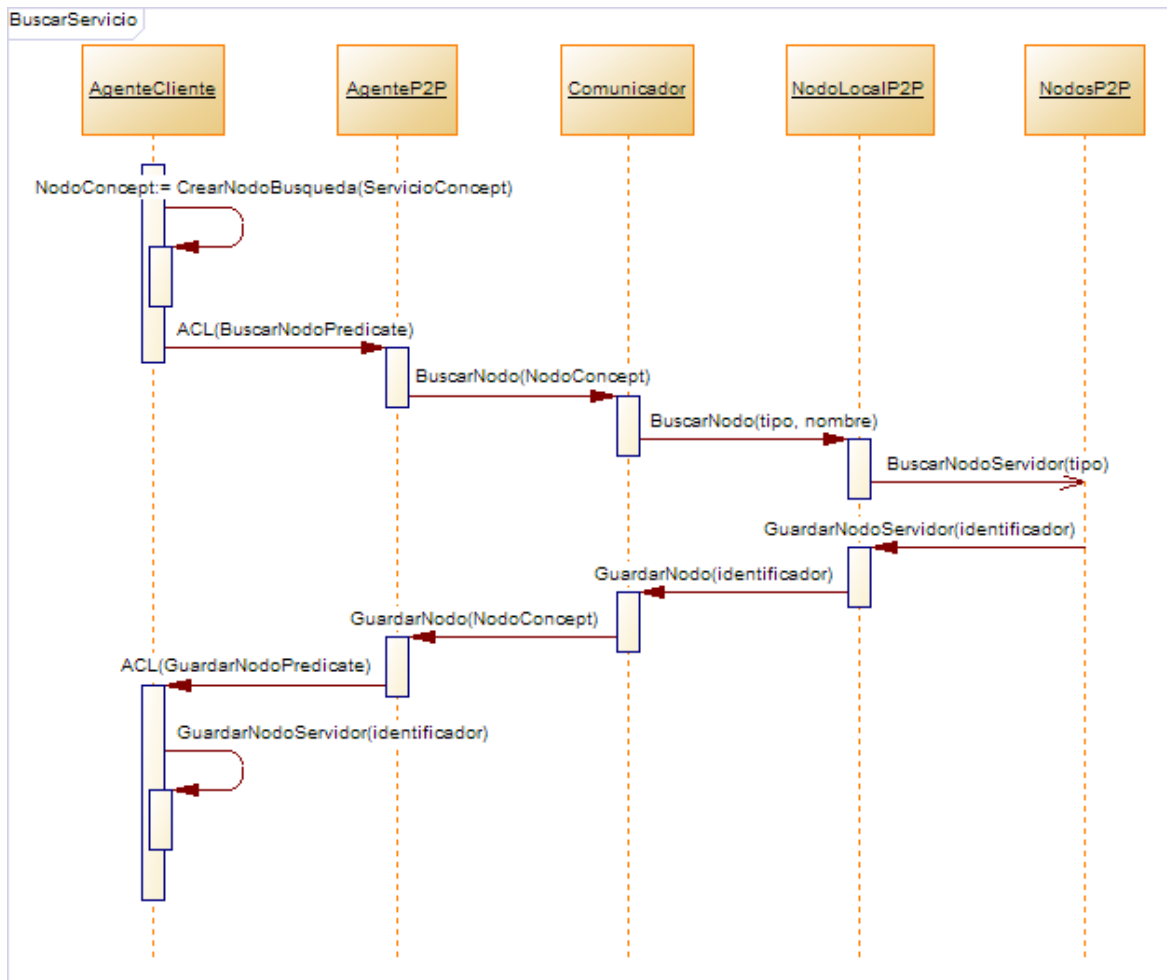


Figura 16. Buscar un Servicio

Cuando un nodo cliente encuentra un servicio, el siguiente paso es enviar su identificador y una lista de sus recursos al nodo servidor. El método de índices locales, permite guardar e indexar en el nodo servidor los objetos almacenados de cada uno de los nodos clientes que estén conectados a él, para que este pueda responder a las peticiones en nombre de todos ellos. Para realizar este proceso, el Agente-Cliente debe conectarse a un *Agente-Servidor*. En la Figura 17, se muestra la secuencia de pasos para realizar la conexión del *Agente-Cliente* y la secuencia de pasos que se realiza en caso de que la conexión al nodo servidor falle. El envío del directorio de recursos o recuperación se muestra en la siguiente sección.

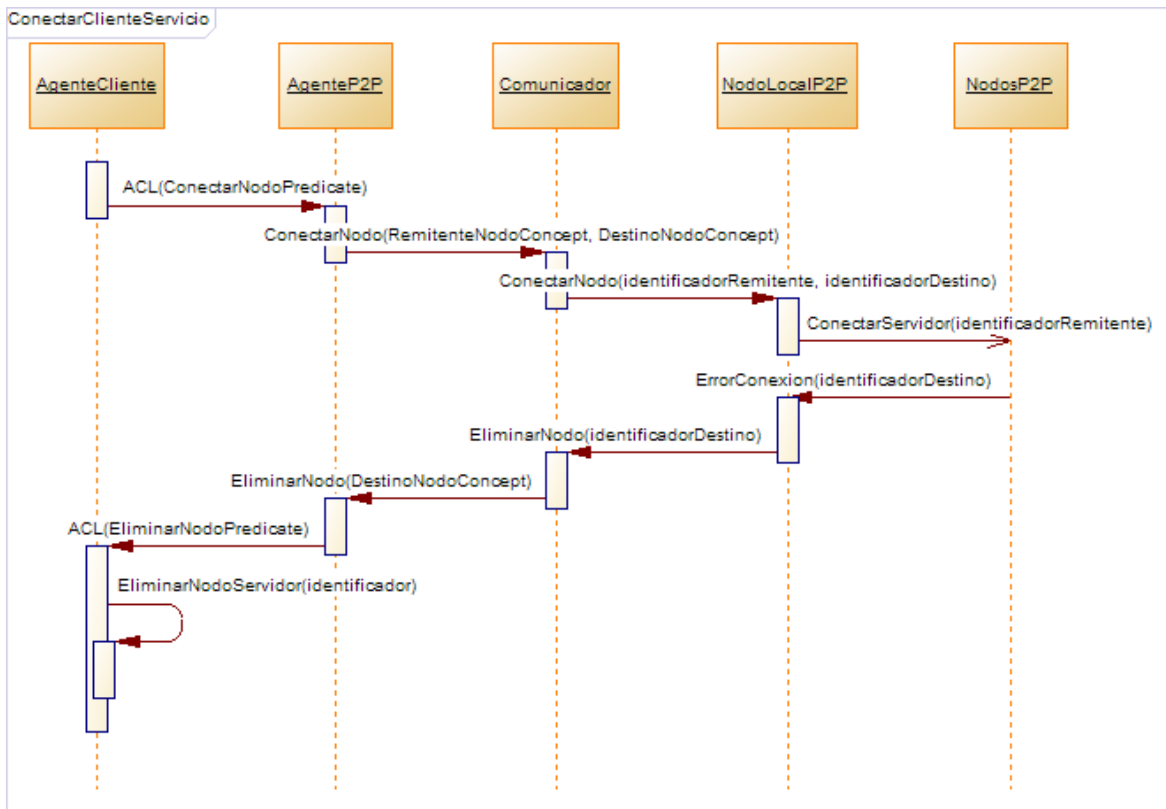


Figura 17. Conectar el Cliente al Servidor (a)

Si existe un error en la conexión hacia un nodo servidor, el identificador que está guardado como un acceso directo debe eliminarse y se debe volver a realizar el proceso de búsqueda del servicio mostrado en la Figura 16.

Si la conexión de un nodo cliente hacia un nodo servidor es exitosa, el *Agente-Servidor* debe guardar el identificador del nodo cliente, en su lista de índices locales, para su posterior uso. La Figura 18, muestra la secuencia de pasos necesaria para este propósito.

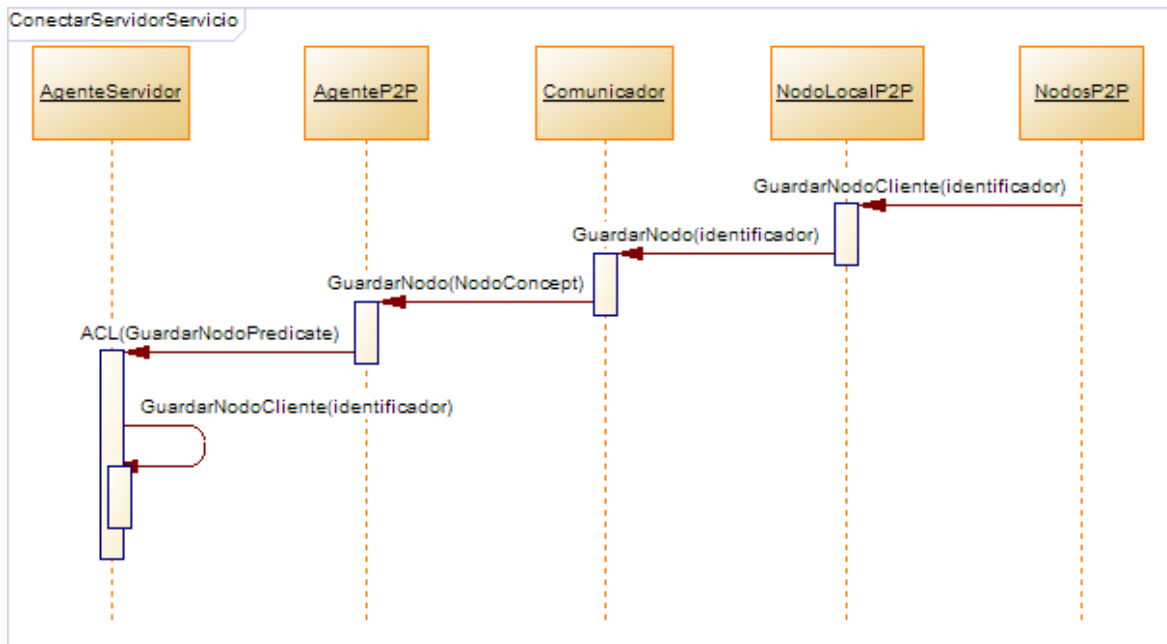


Figura 18. Conectar el Cliente al Servidor (b)

3.4.3. Recuperación del Servicio

Una de las características de las redes superpuestas P2P es que los nodos dentro de la red pueden conectarse y desconectarse de la red de una manera arbitraria. 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, perdiendo todo el proceso que el nodo servidor está realizando en ese momento.

Para solucionar este problema, el *Agente-Cliente* posee un mecanismo de recuperación, implementado mediante un cache local. El cache local almacena el proceso que el nodo cliente está recibiendo actualmente del nodo servidor. En caso de una caída del nodo servidor, el proceso que el servidor estaba realizando queda almacenado en el cache local del nodo cliente, cuando el nodo cliente se conecta a un nuevo nodo que tenga el servicio buscado, le envía el cache local que tenía almacenado por medio del directorio de recuperación, para que el nuevo nodo servidor continúe con el proceso y así el cliente puede recuperar el servicio que estaba utilizando. De esta manera, el servicio que está utilizando el nodo cliente no se ve interrumpido por una caída de un nodo servidor, mejorando la tolerancia a fallos del método índices locales.

En la Figura 15, se muestra el diagrama de flujo del proceso de búsqueda y recuperación del servicio por parte de un nodo cliente.

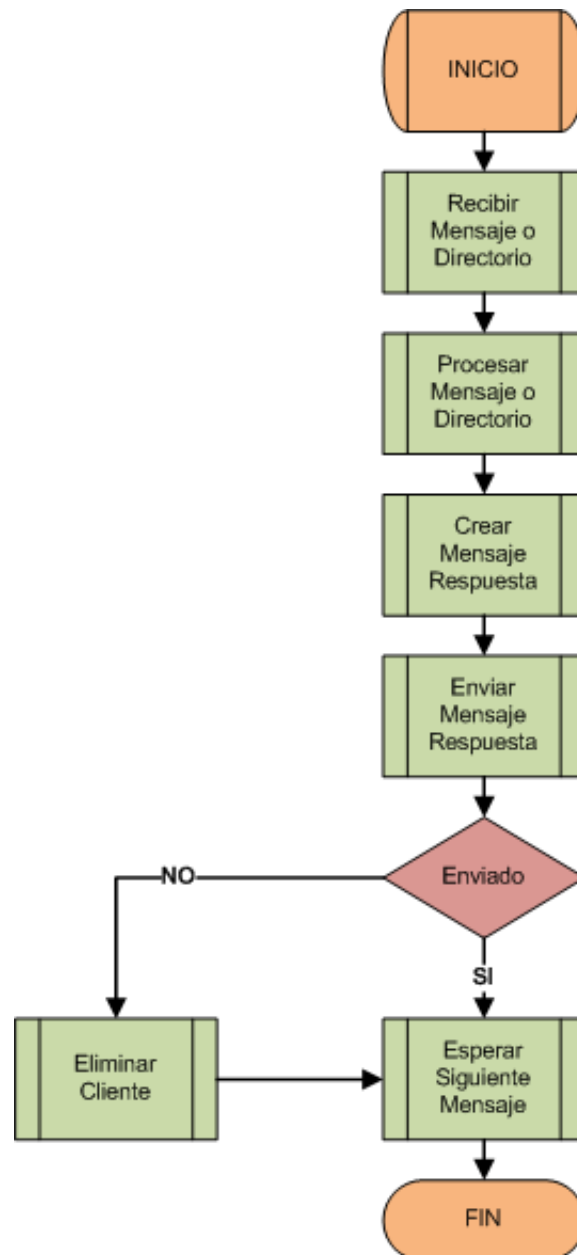


Figura 19. Diagrama de Flujo - Prestación del Servicio

En la Figura 19, se muestra el diagrama de flujo del proceso de prestación del servicio por parte de un nodo servidor. En este diagrama se muestra la recepción y el procesamiento de los mensajes que hace el nodo servidor.



Figura 20. Diagrama de Flujo - Utilización del Servicio

En la Figura 20, se muestra el diagrama de flujo del proceso de utilización del servicio, por parte de un nodo cliente. En este diagrama se muestra la recepción y el procesamiento de los mensajes que hace el nodo cliente, almacenando en el cache los mensajes procesados por el servidor, para llevar un registro del proceso actual que está realizando el servidor.

El proceso de recuperación del servicio, debe ser ejecutado cuando un mensaje no puede llegar a su destino. La Figura 21, muestra la secuencia de pasos para enviar un mensaje.

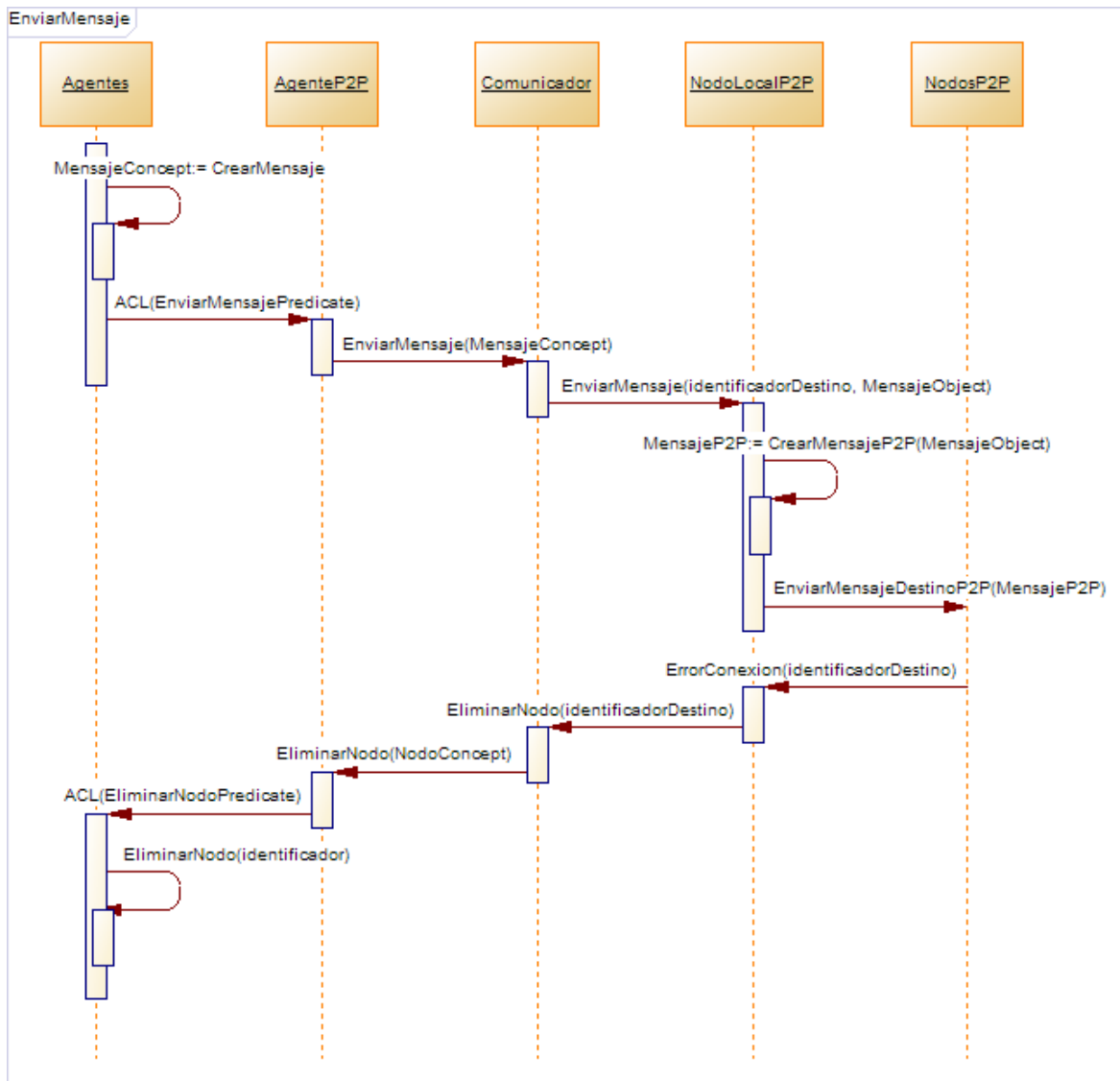


Figura 21. Enviar Mensajes

Si un mensaje no puede ser enviado a su destino, el agente que está enviando el mensaje debe eliminar el identificador, que representa al nodo destino. Si un mensaje se envía correctamente a través de la red superpuesta P2P, el nodo destino debe recibirlo y procesarlo de acuerdo a su funcionalidad. Si el agente que recibe el mensaje es un *Agente-Servidor*, este debe crear un mensaje de respuesta hacia el *Agente-Cliente*, con el resultado del proceso solicitado. La Figura 22, muestra la secuencia de pasos necesaria para recibir y procesar un mensaje.

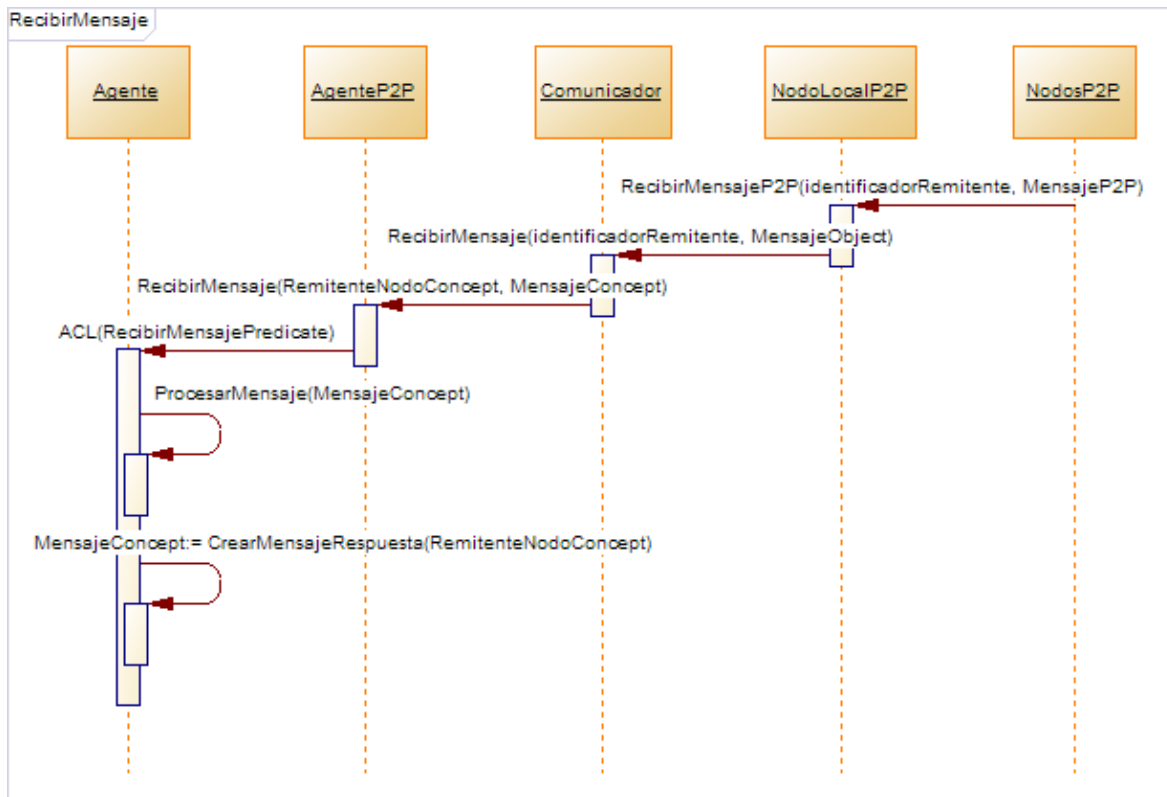


Figura 22. Recibir Mensajes

Cuando el *Agente-Cliente* no puede encontrar el servidor actual, para enviarle mensajes, además de eliminar el identificador, el *Agente-Cliente* crea un directorio de recuperación donde almacena todos los mensajes que han sido creados, pero que no han sido enviados a un servidor para su procesamiento. Este directorio será enviado al próximo servidor, que se encuentre en el proceso de búsqueda, para que todos los mensajes que se encuentran dentro del directorio sean procesados por el nuevo servidor.

En la Figura 23, se muestra la secuencia de pasos para enviar el directorio a un servidor. Como se puede ver, si existe un error en la conexión, se elimina el identificador del nodo servidor y se realiza nuevamente el proceso de búsqueda del servicio.

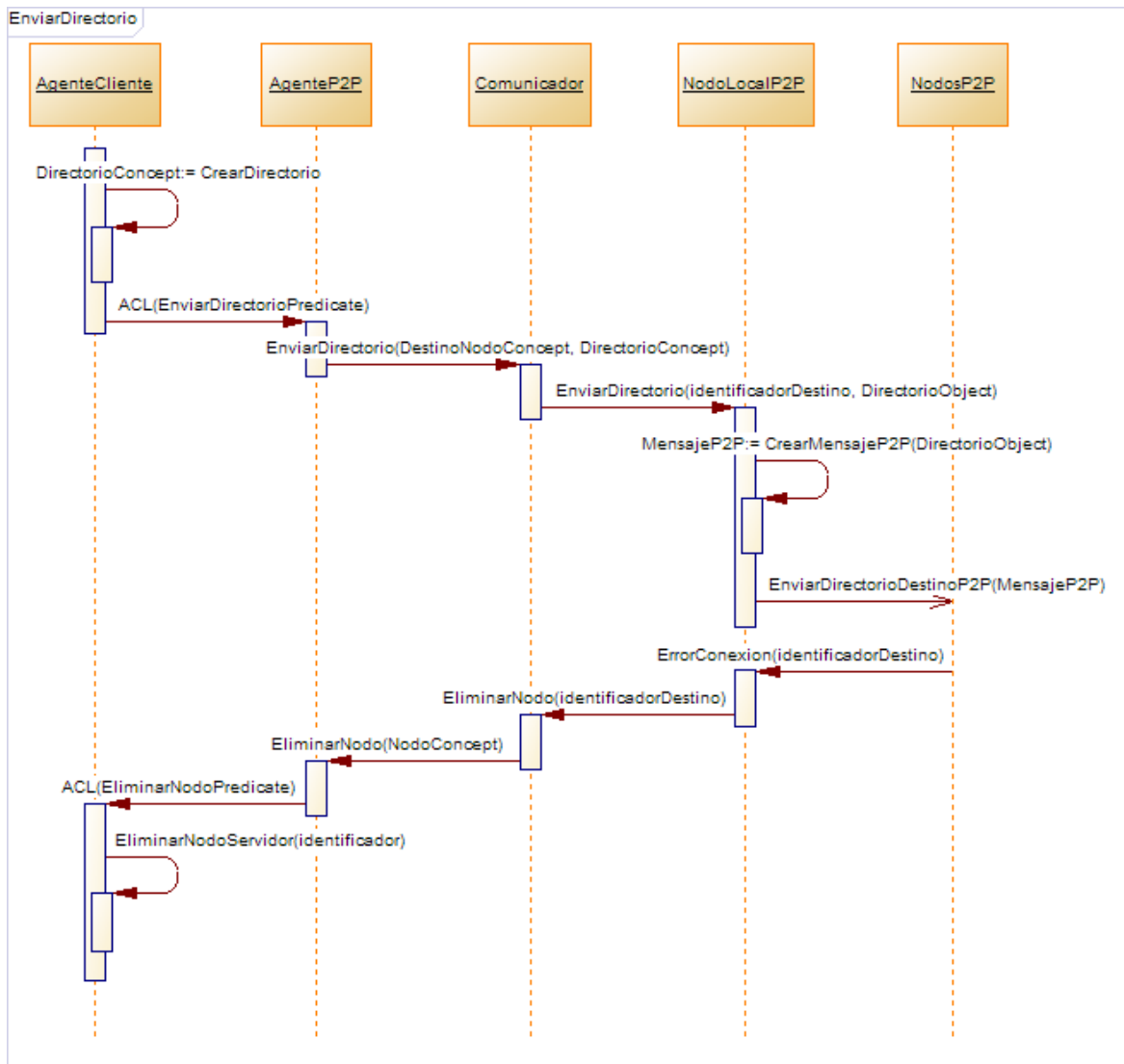


Figura 23. Enviar Directorio

En la Figura 24, se muestra la secuencia de pasos para recibir un directorio, procesarlo y crear un mensaje de respuesta, que contenga el resultado del proceso realizado por el servidor.

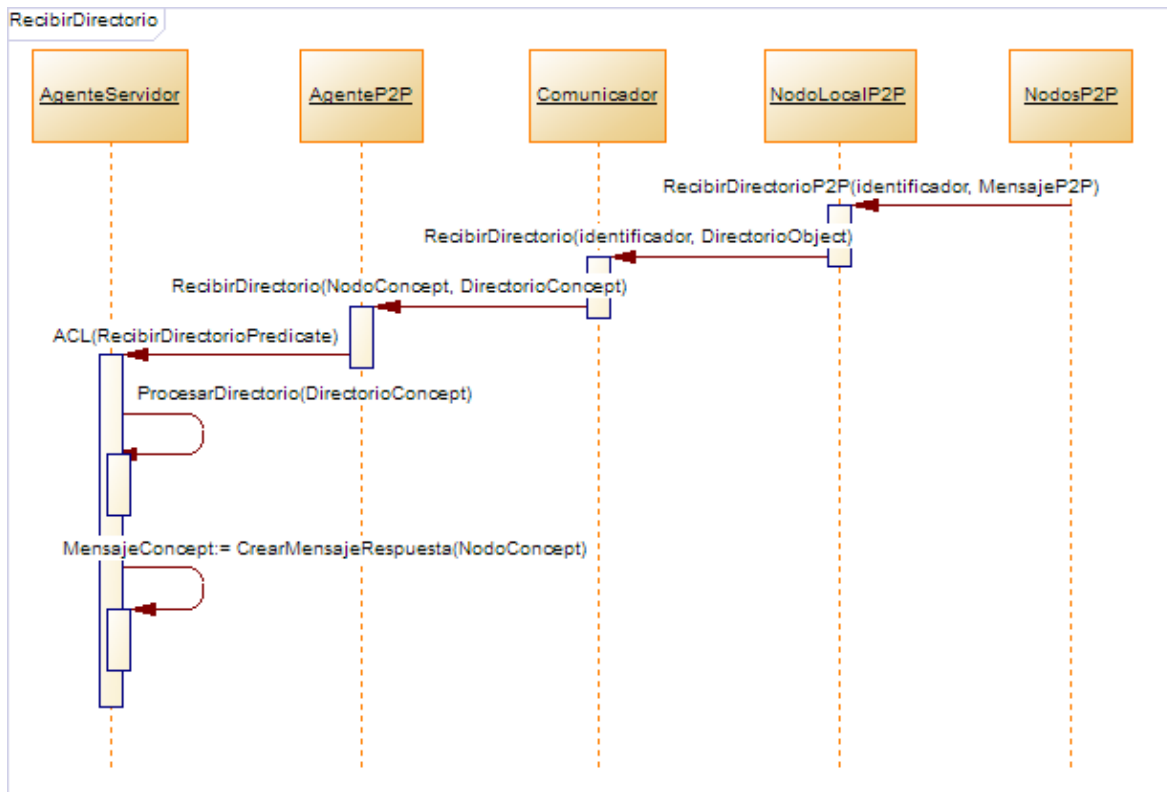


Figura 24. Recibir Directorio

La principal característica de este mecanismo, es proveer al *Agente-Cliente* la capacidad de guardar los mensajes localmente, para que la caída del servicio no se vea interrumpida y pueda ser recuperada fácilmente.

CAPÍTULO IV. TECNOLOGÍAS PARA EL DESARROLLO DE REDES SUPERPUESTAS P2P Y SISTEMAS MULTI-AGENTE

En este capítulo se describen las tecnologías escogidas para la implementación de la red superpuesta P2P y el Sistema Multi-Agente en el prototipo software.

4.1. JXTA

4.1.1. ¿Qué es JXTA?

JXTA es un conjunto de protocolos P2P abiertos y generalizados, que permiten a cualquier dispositivo de red²⁰ comunicarse y colaborar mutuamente como peers o iguales. Los protocolos JXTA son independientes del lenguaje de programación, y existen múltiples implementaciones para diferentes entornos. El uso común de los protocolos JXTA significa que ellos son totalmente interoperables [34].

JXTA provee un conjunto común de protocolos abiertos, que cuentan con implementaciones de referencia, de código abierto para el desarrollo de aplicaciones P2P. Los protocolos JXTA estandarizan la forma en que sus nodos: Se descubren, se auto-organizan en grupos de nodos, publican y descubren recursos en la red, se comunican entre sí y se supervisan [34].

4.1.2. ¿Por qué JXTA?

Un principio de diseño primario de JXTA es proveer una plataforma que contenga las funciones básicas de las redes superpuestas P2P. JXTA supera los defectos potenciales de muchos sistemas P2P existentes [34]:

- **Interoperabilidad:** La tecnología JXTA, está diseñada para que un nodo pueda proveer servicios P2P, 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 del lenguaje de programación, protocolos de transporte de red y plataformas de despliegue.
- **Ubicuidad²¹:** La tecnología JXTA, está diseñada para ser accesible por cualquier dispositivo digital y en cualquier plataforma de despliegue.

²⁰ Entre los dispositivos de red podemos encontrar: Sensores, Teléfonos Celulares, PDAs, Computadores Portátiles y de Escritorio, Estaciones de Trabajo, Servidores y Supercomputadores.

²¹ Ubicuidad significa que puede estar presente en cualquier parte.

4.1.3. Arquitectura JXTA

La arquitectura software de JXTA está dividida en tres capas, como se muestra en la Figura 25 [15] [34].

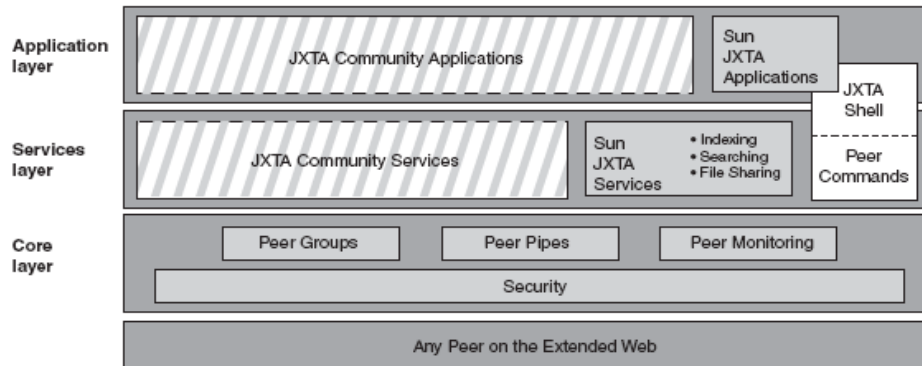


Figura 25. Arquitectura JXTA

Uno de los puntos importantes de la arquitectura JXTA, es que la línea entre las capas de la arquitectura no es rígida. JXTA está diseñada para ser modular, permitiendo a los desarrolladores seleccionar y escoger una colección de servicios y aplicaciones de acuerdo a sus necesidades [34].

4.1.4. 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 [34]. 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).

4.1.5. Servicios JXTA

Todos los protocolos estándar de JXTA son implementados como servicios, llamados servicios del núcleo o *core services* [16]. Los servicios JXTA son: Discovery Service, Peer Info Service, Resolver Service, Pipe Service, Endpoint Service, Rendezvous Service.

La plataforma JXTA y todos sus componentes, se describen más extensamente en el Anexo D.

4.2. JADE

4.2.1. ¿Qué es JADE?

JADE²² es un framework para desarrollar aplicaciones distribuidas basadas en agentes, en conformidad con las especificaciones FIPA, para la interoperabilidad inteligente de Sistemas Multi-Agente.

El objetivo de JADE es simplificar el desarrollo, mientras se asegura el cumplimiento de los estándares, a través de un amplio conjunto de servicios del sistema y de los agentes. JADE se ocupa de todos aquellos aspectos que no son propios de los agentes internos a la plataforma y que son independientes de las aplicaciones, tales como el transporte de mensajes, codificación y análisis, o el ciclo de vida de los agentes.

4.2.2. ¿Por qué JADE?

El objetivo de JADE es simplificar el desarrollo de Sistemas Multi-Agente, que cumplan estándares, a través de un amplio conjunto de servicios del sistema y de los agentes en el cumplimiento de las especificaciones FIPA: Servicio de nombres y servicio de páginas amarillas, transporte de mensajes y servicio de análisis, y una biblioteca de protocolos de interacción FIPA lista para ser utilizada [35].

La plataforma de agentes JADE, cumple con las especificaciones FIPA e incluye todos los elementos obligatorios para administrar la plataforma, los cuales son el AMS, el DF y el MTS. Toda la comunicación de agentes se realiza a través del paso de mensajes, donde FIPA-ACL es el lenguaje para representar los mensajes [35].

El modelo completo de comunicaciones FIPA ha sido implementado y sus componentes han sido claramente diferenciados y totalmente integrados: protocolos de interacción, envolturas, ACL, lenguajes de contenido, esquemas de codificación, ontologías y finalmente, protocolos de transporte [35].

4.2.3. Arquitectura JADE

La Figura 26, muestra los elementos de la arquitectura principal de la plataforma JADE [20]. Una plataforma JADE está compuesta de contenedores de agentes que pueden estar distribuidos sobre la red. Los agentes viven en contenedores, que proveen el tiempo de ejecución de JADE y todos los servicios necesarios para almacenar y ejecutar agentes. Hay un contenedor especial, llamado Contenedor Principal, el cual representa el punto de arranque de una plataforma.

²² Java Agent Development Framework

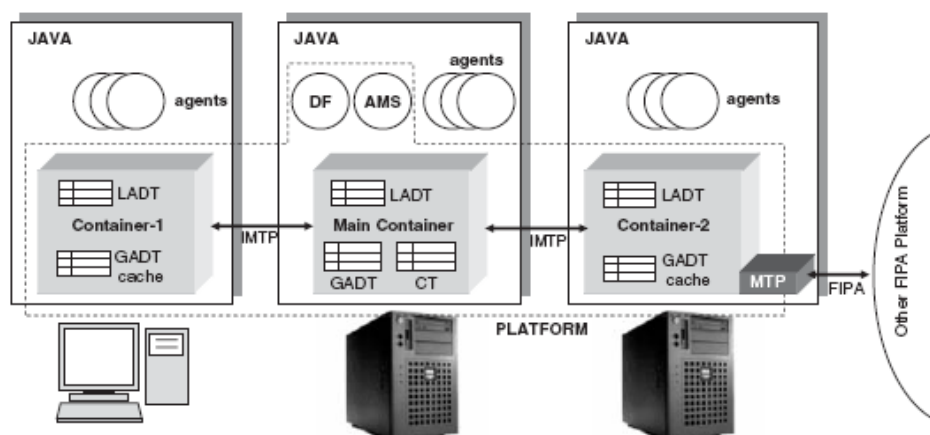


Figura 26. Arquitectura JADE

4.2.4. Comportamientos en JADE

Un comportamiento, representa una tarea que un agente puede ejecutar en un momento dado. Para que un agente ejecute la tarea asignada, el comportamiento debe ser agregado al agente. Un comportamiento puede ser agregado cuando se inicia el agente, dentro de otros comportamientos ya iniciados o por respuesta a un evento ejecutado por un usuario del sistema [20].

Cada comportamiento en JADE debe implementar dos métodos. El método *action()* define las operaciones a ser realizadas cuando el comportamiento se está ejecutando. El método *done()* retorna un valor booleano para indicar cuando un comportamiento ha sido completado y debe ser removido del conjunto (pool) de comportamientos que un agente está ejecutando. Cada comportamiento tiene una variable miembro llamada *myAgent* que apunta al agente que ejecuta el comportamiento. Esta variable provee una vía fácil para acceder a los recursos del agente dentro del comportamiento [20].

4.2.5. Servicios en JADE

JADE, provee un servicio de páginas amarillas, que permite a los agentes descubrir dinámicamente otros agentes que se encuentran disponibles en la plataforma. El servicio de páginas amarillas, permite a los agentes publicar las descripciones de uno o más servicios que ellos proveen, para que otros agentes puedan fácilmente descubrir y explotar estos servicios [20].

El servicio de páginas amarillas en JADE, en concordancia con las especificaciones FIPA, es suministrado por un agente especializado llamado Facilitador de Directorio o DF. Cada plataforma JADE tiene un agente DF por defecto. Como el DF es un agente, es posible que otros agentes interactúen con él mediante el intercambio de mensajes FIPA-ACL, usando un lenguaje de contenido

y una ontología propia. Para simplificar estas interacciones JADE provee la clase *jade.domain.DFService* con la cual es posible publicar y buscar servicios a través de una amplia variedad de métodos [20].

4.2.5.1. Publicando Servicios

Un agente que desee publicar uno o más servicios, debe proveer al DF una descripción que incluye: Su propio AID, una lista de los servicios que provee y opcionalmente una lista de lenguajes y ontologías que otros agentes deben usar para interactuar con él. Cada descripción del servicio publicado debe incluir el tipo de servicio, el nombre del servicio, los lenguajes y ontologías requeridos para usar el servicio y una colección de propiedades específicas del servicio, expresadas en pares de la forma clave-valor [20].

4.2.5.2. Buscando Servicios

Un agente que desee buscar servicios debe proveer al DF una plantilla (template) de la descripción del servicio que desea buscar y llamar al método estático *search()* de la clase *DFService*. El resultado de la búsqueda es una lista de todas las descripciones que coinciden con la plantilla que se pasó al agente DF. De acuerdo a las especificaciones FIPA, una descripción del servicio coincide con la plantilla si todos los campos especificados en la plantilla están presentes en la descripción del servicio y todos sus valores coinciden [20].

El agente DF de JADE también provee un mecanismo de suscripción, que permite a los agentes ser notificados tan pronto como otros agentes publiquen o despubliquen sus servicios [20].

La plataforma JADE y todos sus componentes, se describen más extensamente en el Anexo D.

CAPÍTULO V. DESARROLLO DEL 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 4.

5.1. Especificación de Requerimientos

Después de haber definido la Técnica de Búsqueda, 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 Búsqueda propuesta sobre el servicio de envío y recepción de mensajes de eco²³. Por lo tanto, la mayoría de la funcionalidad del prototipo software, se ejecutará de manera automática, sin ninguna 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 superpuestas P2P y Sistemas Multi-Agente, la primera, JXTA, para el desarrollo de Redes Superpuestas P2P; y la segunda, JADE, para el desarrollo de Sistemas Multi-Agente.

5.1.2. Funciones del Producto

El prototipo software permitirá el envío y recepción de mensajes de eco. La propagación del mensaje se realizará mediante una conexión uno-a-uno (unicast), no por medio de la típica conexión uno-a-muchos (multicast). El envío de mensajes de eco se realizará cada 6 segundos, para tener un total de 10 mensajes por minuto. El contenido del mensaje de eco será la hora actual del sistema en el que se está ejecutando el cliente.

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.

²³ Un servicio de mensajes de eco consiste en el envío y propagación de un mensaje hacia todos los nodos que estén conectados actualmente al nodo que presta el servicio en la red superpuesta P2P.

5.1.3. Características del Usuario

La mayoría de la funcionalidad del prototipo software se ejecutará de manera automática, sin ninguna 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, como servidor o como ambos. Sin embargo, el usuario podrá ver el envío y recepción de mensajes de eco, tanto del cliente como del servidor (si está activada la respectiva funcionalidad), y los mensajes de depuración del sistema.

5.1.4. Priorizar los Requisitos

- **Primera Iteración:** Desarrollar la funcionalidad del Sistema Multi-Agente del prototipo software, de la técnica de búsqueda y el servicio de eco, mediante la plataforma de agentes JADE.
- **Segunda Iteración:** Desarrollar la funcionalidad del Sistema P2P y de las comunicaciones en la red superpuesta P2P del prototipo software, de la técnica de búsqueda y el servicio de eco, mediante la plataforma JXTA. Desarrollar la interfaz de comunicaciones entre las dos plataformas (JADE y JXTA) y el prototipo software.

5.1.5. Requerimientos Funcionales

Los requerimientos funcionales del prototipo software se describen a continuación:

- **Publicar el Servicio de Eco (R-1):** El sistema debe crear y publicar el servicio de eco.
- **Buscar el Servicio de Eco (R-2):** El sistema debe buscar el servicio de eco.
- **Recuperar el Servicio de Eco (R-3):** El sistema debe recuperar el servicio de eco. El sistema hace uso del cache para recuperar el estado del envío de mensajes.
- **Enviar el Mensaje de Eco (R-4):** El sistema debe enviar un mensaje de eco. Esta funcionalidad pertenece a un nodo cliente.
- **Transmitir el Mensaje de Eco (R-5):** El sistema debe transmitir un mensaje de eco. Esta funcionalidad pertenece a un nodo servidor.

- **Recibir el Mensaje de Eco (R-6):** El sistema debe recibir un mensaje de eco, que fue transmitido por un nodo servidor. Esta funcionalidad pertenece a un nodo cliente.
- **Iniciar el Servidor de Eco (R-7):** El sistema debe permitir al usuario iniciar el servidor de eco.
- **Detener el Servidor de Eco (R-8):** El sistema debe permitir al usuario detener el servidor de eco.
- **Iniciar el Cliente de Eco (R-9):** El sistema debe permitir al usuario iniciar el cliente de eco.
- **Detener el Cliente de Eco (R-10):** El sistema debe permitir al usuario detener el cliente de eco.

5.1.6. Requerimientos No Funcionales

- **Fiabilidad:** El prototipo software debe ser tolerante a fallos. 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 superpuestas P2P 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 superpuesta P2P.
- **Seguridad:** El prototipo software no tendrá en cuenta aspectos de seguridad, tales como nombre de usuarios o contraseñas, o sistemas de autenticación. Si se llegará a solicitar algún nombre de usuario o contraseñas, solo se hará con fines informativos.
- **Consistencia:** La consistencia es un punto importante en las aplicaciones P2P, 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 que tenga instalada una Máquina Virtual Java. Se recomienda la versión 1.6.0 del JDK o del JRE de Sun Microsystem.

5.2. Análisis y Diseño

Después de definir los requerimientos del prototipo software, se realizaron los modelos de caso de uso en formato de alto nivel y extendido. A continuación, se presenta la definición de los actores y los diagramas de casos de uso. Los modelos y sus descripciones se muestran más detalladamente en el ANEXO C: Documento de Casos de Uso.

5.2.1. Actores del Sistema

A continuación se presenta la definición de los actores que interactuarán con el prototipo software. La mayoría de la funcionalidad del prototipo software se ejecutará de manera automática, sin ninguna interacción con el usuario final del sistema (Ver Anexo: Documento de Especificación de Requerimientos Software - SRS). Por lo tanto, el sistema como tal se describirá como un actor.

- **Actor Sistema (ACT-1):** El sistema será el encargado de publicar, buscar y recuperar el servicio de eco, y procesar el cache del cliente para la recuperación del servicio de eco. También será el encargado de enviar, transmitir y recibir los mensajes de eco, y de lanzar la funcionalidad inicial del servidor y el cliente del servicio de eco.
- **Actor Usuario (ACT-2):** El usuario podrá iniciar y detener el servidor de eco. El usuario también podrá iniciar y detener el cliente de eco.

5.2.2. Casos de Uso en Formato de Alto Nivel

Esta sección contiene los diferentes diagramas de casos de uso, que representan las funcionalidades del sistema relacionadas con los actores del mismo, los cuales se definieron anteriormente. La Figura 27, muestra el diagrama de casos de uso de una manera general, organizado por paquetes de acuerdo a la funcionalidad.

Los paquetes están representados de la siguiente manera:

- **Paquete Servicio:** Agrupa los casos de uso relacionados con el servicio de eco. Incluye los casos de uso Publicar Servicio, Buscar Servicio y Recuperar Servicio.
- **Paquete Mensajes:** Agrupa los casos de uso relacionados con los mensajes de eco. Incluye los casos de uso Enviar Mensaje, Transmitir Mensaje y Recibir Mensaje.
- **Paquete Opciones:** Agrupa los casos de uso relacionados con las opciones que puede realizar el actor Usuario. Incluye los casos de uso Iniciar Servidor, Detener Servidor, Iniciar Cliente y Detener Cliente. Según la generalización de actores vista en la Figura 27, estos casos de uso también puede ser realizados por el actor Sistema.

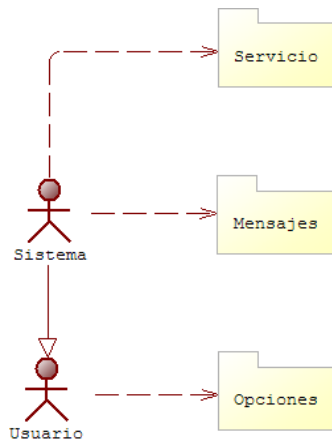


Figura 27. Diagrama de Casos de Uso – Paquetes

Los diagramas de caso de uso en formato nivel y su descripción se encuentran en el ANEXO C: Documento de Casos de Uso.

5.2.3. Casos de Uso en Formato Extendido

Esta sección contiene los diferentes diagramas de casos de uso en formato extendido, que representan las funcionalidades del sistema relacionadas con los actores del mismo, los cuales se definieron anteriormente. Los paquetes representados aquí, son los mismos paquetes que se representaron en el diagrama de casos de uso de alto nivel de la Figura 27.

La Figura 28, muestra el diagrama de casos de uso extendido del paquete Servicio.

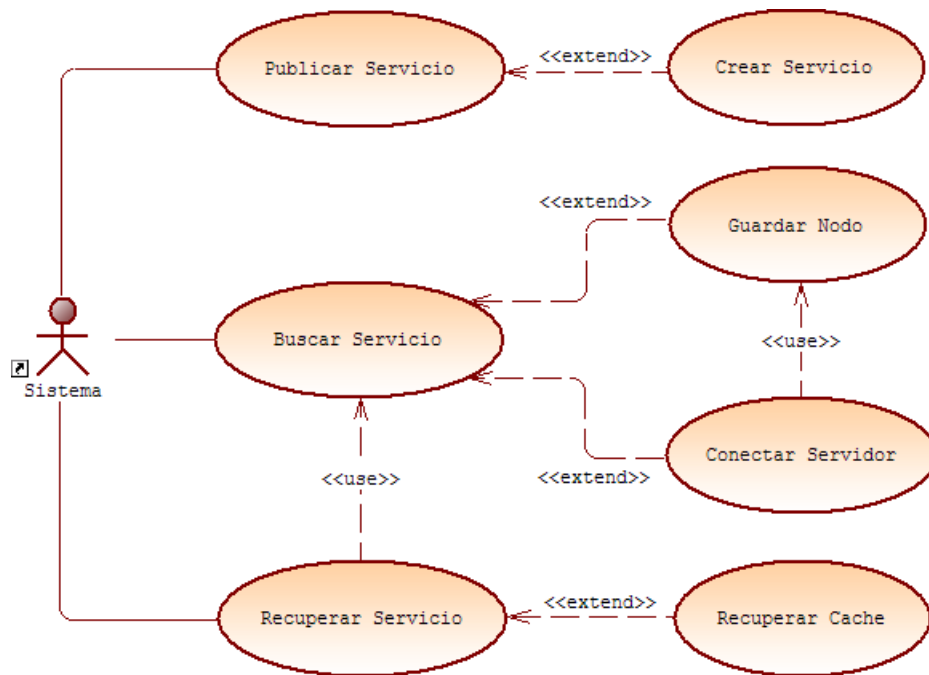


Figura 28. Diagrama de Casos de Uso Extendido - Paquete Servicio

La Figura 29, muestra el diagrama de casos de uso extendido del paquete Mensajes.

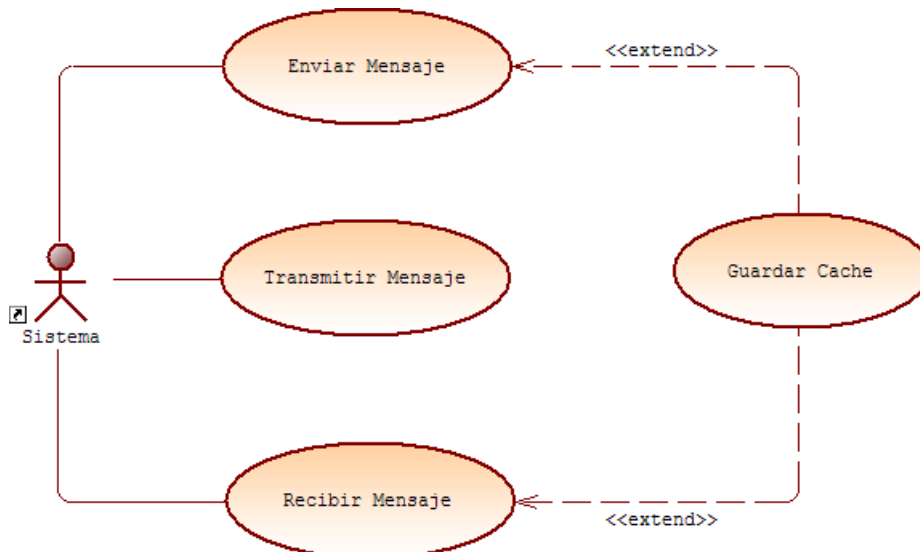


Figura 29. Diagrama de Casos de Uso Extendido - Paquete Mensajes

La Figura 30, muestra el diagrama de casos de uso extendido del paquete Opciones.

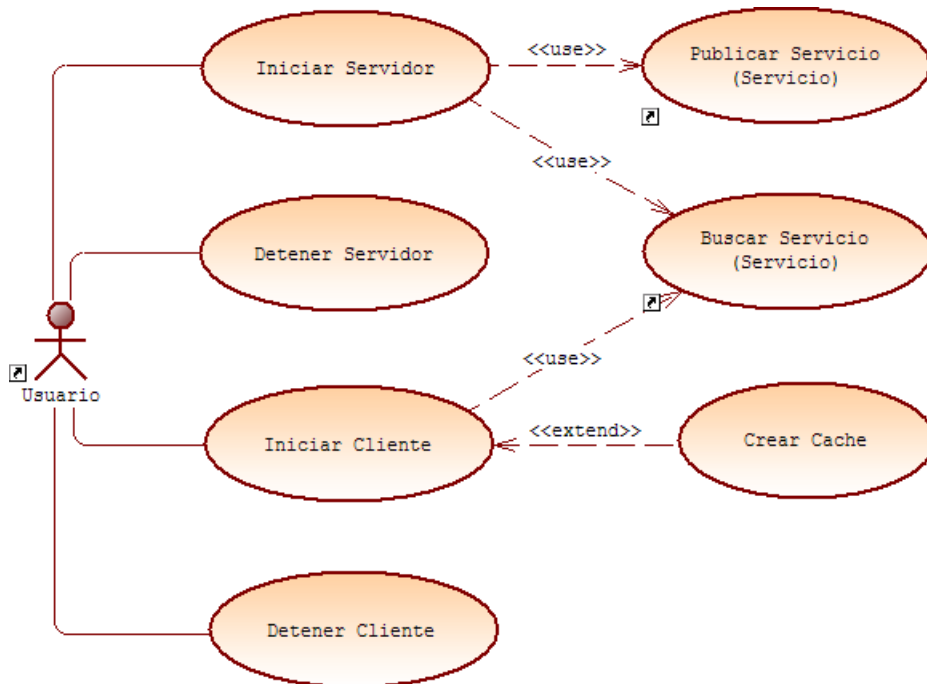


Figura 30. Diagrama de Casos de Uso Extendido - Paquete Opciones

5.2.4. Arquitectura del Prototipo Software

A continuación se describe la arquitectura lógica del sistema, esta arquitectura es el resultado de la integración de las dos plataformas de desarrollo: JADE y JXTA. Estas plataformas se describen con detalle en la Sección 4.

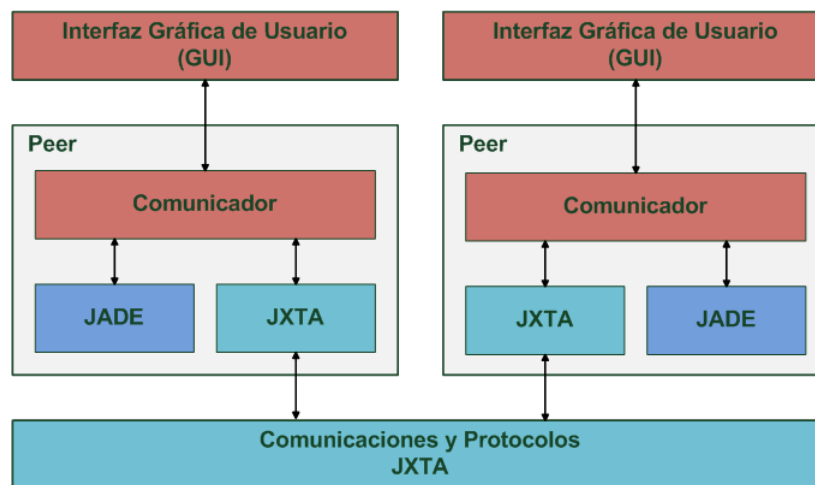


Figura 31. Arquitectura General del Prototipo

En la Figura 31, se muestra la arquitectura general del prototipo software, la cual contiene cuatro elementos principales: La plataforma JADE, La plataforma JXTA, El Comunicador y La Interfaz Gráfica de Usuario (GUI).

5.2.4.1. Módulo JADE

Contiene las funcionalidades del Sistema Multi-Agente, a través de la plataforma JADE. La plataforma JADE cumple con las especificaciones FIPA e incluye todos los elementos obligatorios para administrar la plataforma, los cuales son el AMS y el DF. Toda la comunicación de agentes se realiza a través del paso de mensajes, donde FIPA-ACL es el lenguaje para representar los mensajes [35].

El módulo JADE maneja un solo contenedor de agentes, el contenedor principal (Main Container), en el que están presentes los respectivos agentes del sistema: El Agente-Eco-Servidor que se encarga de la provisión del servicio debe registrar el servicio en el agente DF y contiene todos los elementos necesarios para proveer el servicio; El Agente-Eco-Cliente que hace uso del servicio; El Agente-JXTA, que se encarga de recibir los mensajes de los Agente de Eco y transmitirlos al módulo de comunicaciones (El Comunicador) para que sean enviados posteriormente a través de la red superpuesta P2P, por intermedio de los protocolos JXTA; y el Agente-GUI que se encarga de actualizar y gestionar los mensajes desde y hacia la interfaz gráfica.

En la Figura 32, se muestran los componentes del Módulo JADE.

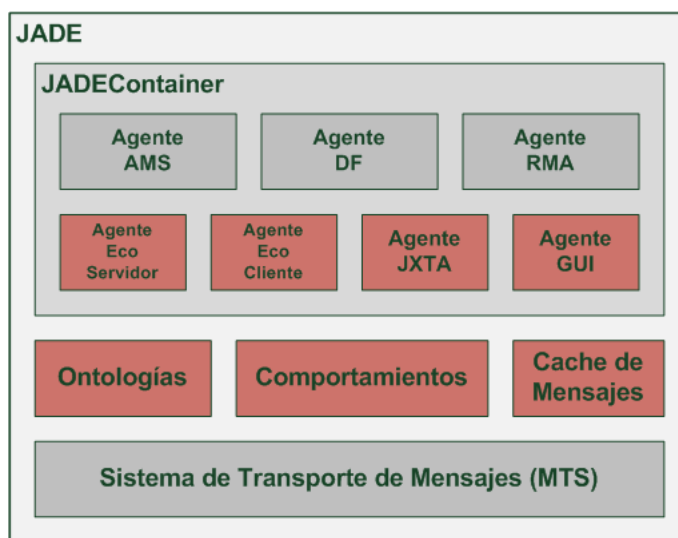


Figura 32. Módulo JADE

5.2.4.2. Módulo JXTA

Contiene las funcionalidades del Sistema P2P, así como también las comunicaciones y protocolos usados en la red superpuesta P2P, a través de la plataforma JXTA. La plataforma JXTA provee un conjunto común de protocolos para el desarrollo de aplicaciones P2P [34].

El módulo JXTA, es el encargado de crear, publicar y buscar los respectivos advertisements que representan el servicio en la red superpuesta P2P. También es el encargado de las comunicaciones y del envío y recepción de mensajes, entre cada nodo de la red superpuesta P2P, que se realizan por medio de los diferentes protocolos de JXTA.

En la Figura 33, se muestran los componentes del Módulo JXTA.

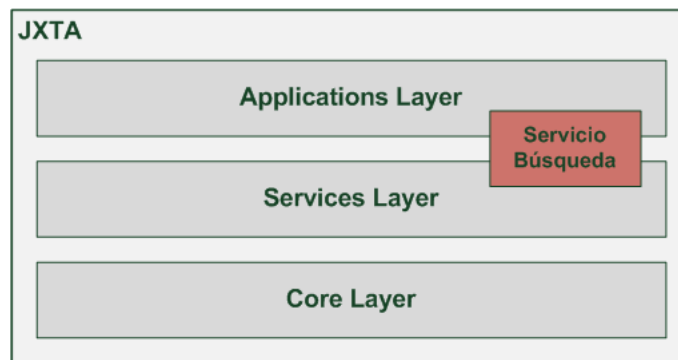


Figura 33. Módulo JXTA

5.2.4.3. Módulo Comunicador

Es el encargado de las comunicaciones entre las dos plataformas, y de las comunicaciones entre las plataformas y la interfaz gráfica. Realiza la tarea de conversión entre los lenguajes, tipos de objetos y parámetros de los respectivos métodos que cada una de las plataformas maneja. En la Figura 34, se muestran los componentes del Módulo Comunicador.

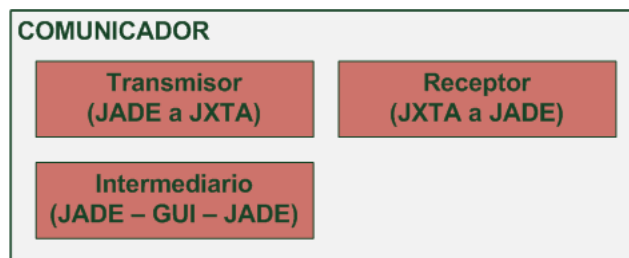


Figura 34. Módulo Comunicador

5.2.4.4. Interfaz Gráfica de Usuario

Representa la interfaz gráfica, por medio de la cual el usuario interactúa con el sistema.

5.2.5. Diagrama de Clases y Diagramas de Secuencia

En el diagrama de clases de la Figura 35, se muestran las clases necesarias para el desarrollo del prototipo software. Los agentes EcoCliente y EcoServidor no tienen visibilidad del JADEContainer, por lo tanto, para comunicarse con él, deben hacerlo por intermedio del AgenteJXTA, para enviar mensajes hacia la red superpuesta P2P; o por medio del AgenteGUI, para enviar mensajes hacia la interfaz de usuario. La comunicación entre agentes debe realizarse por medio de mensajes FIPA-ACL.

La Clase JXTAEco provee la funcionalidad necesaria para permitir la prestación y uso del servicio en la red superpuesta P2P. Esta funcionalidad incluye: La creación y publicación de los diferentes advertisements, así como también el envío y recepción de mensajes.

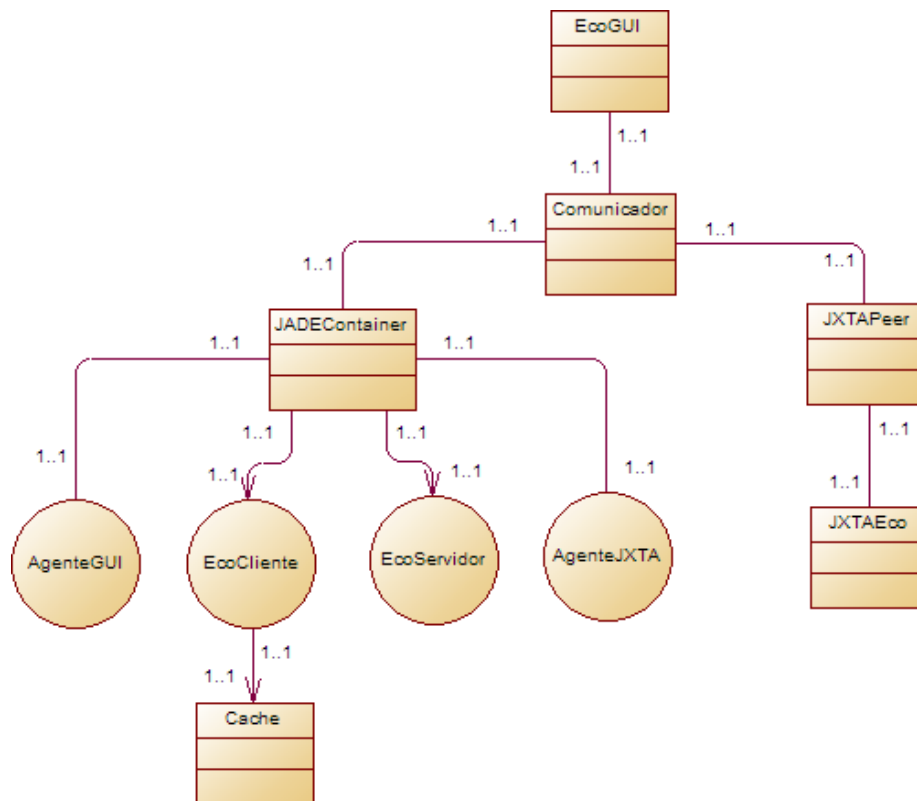


Figura 35. Diagrama de Clases del Prototipo

Para el desarrollo del prototipo software, se utilizarán los diagramas de secuencia que se muestran en la Sección 3.4. Prestación de Servicios, teniendo en cuenta que la funcionalidad del NodoLocalP2P es realizada por la clase JXTAEco; y que el JADEContainer y el JXTAPeer funcionan como mediadores entre los diferentes agentes y objetos que están ligados a ellos.

5.3. Implementación

La implementación del prototipo software se realizará mediante dos iteraciones, teniendo en cuenta la metodología de desarrollo Agile Unified Process (AUP) [36]. Las iteraciones se realizarán de la siguiente manera:

- **Primera Iteración:** Desarrollar la funcionalidad del Sistema Multi-Agente del prototipo software, de la técnica de búsqueda y el servicio de eco, mediante la plataforma de agentes JADE.
- **Segunda Iteración:** Desarrollar la funcionalidad del Sistema P2P y de las comunicaciones en la Red superpuesta P2P del prototipo software, de la técnica de búsqueda y el servicio de eco, mediante la plataforma JXTA. Desarrollar la interfaz de comunicaciones, El Comunicador, entre las dos plataformas, JADE y JXTA, y el prototipo software.

Para la implementación del prototipo software, se uso el lenguaje de programación java (versión 1.6.18), la plataforma de agentes JADE (versión 3.7), la plataforma JXTA (versión 2.5) y el entorno de desarrollo integrado NetBeans IDE (versión 6.8).

5.4. Pruebas y Verificación de la Técnica de Búsqueda

Después de definir la técnica de búsqueda para la prestación de servicios e implementar el prototipo software, se definió un plan de pruebas para la verificación de la técnica propuesta, usando el prototipo software construido.

5.4.1. Definición de los Escenarios de Prueba

Para la realización del plan de pruebas, se definieron una serie de escenarios posibles, para probar las características de búsqueda de servicios en la técnica de búsqueda propuesta.

Como se describió en el Capítulo I, las redes superpuestas P2P 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 P2P, puede comportarse como un

cliente o un servidor. Estas características permiten definir los escenarios que serán usados para verificar la técnica de búsqueda propuesta.

Los escenarios definidos, para la verificación de la técnica de búsqueda propuesta, se muestran en la Tabla 4.

Número de Escenario	Nodos Cliente	Nodos Servidor
1	Uno o Muchos	Ninguno
2	Ninguno	Uno o Muchos
3	Uno	Uno
4	Uno	Muchos
5	Muchos	Uno
6	Muchos	Muchos

Tabla 4. Escenarios de Prueba

La Tabla 5, muestra el formato utilizado para registrar los resultados de los escenarios de prueba.

FORMATO DE PRUEBAS				
Número de Escenario:				
Nodo Cliente	Número de Mensajes	Tiempo de Respuesta	Nodo Servidor	Estado del Servidor
Descripción de Resultados:				

Tabla 5. Formato de Registro para los Resultados de las Pruebas

Los campos del formato de registro se describen a continuación:

- **Número de Escenario:** Es el número del escenario, el cual se encuentra especificado en la Tabla 4.
- **Nodo Cliente:** Es el nombre del nodo cliente que se está usando en el escenario.
- **Número de Mensajes:** Es el número de mensajes de búsqueda, enviados por un nodo cliente, antes de encontrar un nodo servidor. Para medir el número de mensajes enviados por cada nodo cliente, se creó una funcionalidad en el prototipo software, que muestra un mensaje de información, cada vez que un nodo cliente envía la consulta de búsqueda y esta no produce un resultado satisfactorio.
- **Tiempo de Respuesta:** Para medir el tiempo de conexión, se inicia un *timer* al momento que el cliente envía el mensaje de conexión al servidor y

se detiene cuando el cliente recibe la respuesta enviada por el servidor. Para medir el tiempo de respuesta de mensajes de eco, se utiliza la hora actual del sistema, dada en milisegundos. Se guarda la hora del sistema cuando se envía un mensaje hacia un nodo servidor y cuando llega la respuesta del mismo mensaje. La diferencia entre estas dos horas del sistema, es el tiempo de respuesta, dado en milisegundos.

- **Nodo Servidor:** Es el nombre del nodo servidor que se está usando en el escenario.
- **Estado del Servidor:** Es el estado en el que se encuentra un nodo servidor. Puede tener dos estados: *Ocupado*, si al menos un nodo cliente está conectado a él; o *Libre*, si no hay ningún nodo cliente conectado a él.
- **Descripción de Resultados:** Es la descripción de los resultados que se obtuvieron en el desarrollo del escenario de prueba.

5.4.2. Descripción del Marco Experimental

Para la verificación de la técnica de búsqueda, se realizó el marco experimental sobre una red LAN de 10/100, conectada por medio de una topología física de red tipo BUS, en donde se desactivo el firewall, para evitar problemas de comunicación entre los diferentes nodos de la red superpuesta P2P.

La sala de informática, sobre la que está montada la topología de red física y en donde se realizaron las pruebas, está compuesta por 16 computadores, cada uno con diferentes características hardware y software. La sala estaba dedicada solo 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 E: Especificación de los Equipos Utilizados en las Pruebas. En la Figura 36, se muestra la sala de informática utilizada para las pruebas.



Figura 36. Sala de Informática utilizada para las Pruebas

5.4.3. Desarrollo de los Escenarios de Prueba

Para cada uno de los escenarios de prueba, todos los servidores estaban inicialmente libres, y los nodos cliente no se habían conectado a ningún servidor con anterioridad, por lo tanto, no tienen mensajes de recuperación para ser procesados al inicio. 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:

5.4.3.1. Escenario de Prueba No. 1



FORMATO DE PRUEBAS				
Número de Escenario: 1				
Nodo Cliente	Número de Mensajes	Tiempo de Respuesta	Nodo Servidor	Estado del Servidor
Cliente-01	-	-	Ninguno	-
Cliente-02	-	-	Ninguno	-
Descripción de Resultados:				
En este escenario, los clientes envían mensajes de consulta para buscar un nodo servidor, como no existen servidores activos, los mensajes de búsqueda son enviados constantemente, mientras el cliente permanezca activo o hasta que se encuentre un servidor. En este caso, el cliente no guarda mensajes de recuperación, porque no ha empezado a consumir el servicio.				

Tabla 6. Escenario de Prueba No. 1

5.4.3.2. Escenario de Prueba No. 2



FORMATO DE PRUEBAS				
Número de Escenario: 2				
Nodo Cliente	Número de Mensajes	Tiempo de Respuesta	Nodo Servidor	Estado del Servidor
-	-	-	Servidor-01	Libre
-	-	-	Servidor-02	Libre

Descripción de Resultados:
En este escenario, los servidores permanecen activos, esperando conexiones de uno o muchos nodos cliente. Cada servidor tiene publicado su respectivo servicio en la red superpuesta P2P.

Tabla 7. Escenario de Prueba No. 2

5.4.3.3. Escenario de Prueba No. 3

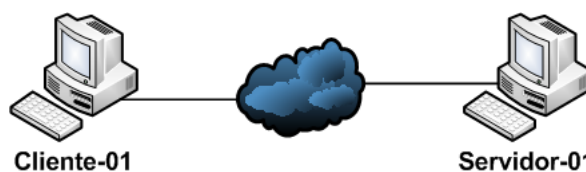


Figura 39. Representación del Escenario de Prueba No. 3

FORMATO DE PRUEBAS				
Número de Escenario: 3				
Nodo Cliente	Número de Mensajes	Tiempo de Respuesta	Nodo Servidor	Estado del Servidor
Cliente-01	24	0,720 segundos	Servidor-01	Ocupado

Descripción de Resultados:
En este caso, el nodo cliente envió 24 mensajes de búsqueda, antes de encontrar el servidor. El tiempo de respuesta de la conexión fue de 0,720 segundos. El tiempo de respuesta para los mensajes de eco tiene un promedio de 0,674 segundos.

Tabla 8. Escenario de Prueba No. 3

5.4.3.4. Escenario de Prueba No. 4

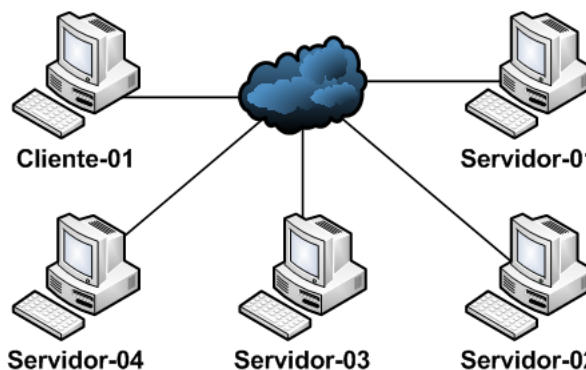


Figura 40. Representación del Escenario de Prueba No. 4

FORMATO DE PRUEBAS				
Número de Escenario: 4				
Nodo Cliente	Número de Mensajes	Tiempo de Respuesta	Nodo Servidor	Estado del Servidor
Cliente-01	16	0,864 segundos	Servidor-01	Ocupado
-	-	-	Servidor-02	Libre
-	-	-	Servidor-03	Libre
-	-	-	Servidor-04	Libre

Descripción de Resultados:
Este caso es muy parecido al del Escenario 3. En este caso, el nodo cliente envió 16 mensajes de búsqueda, antes de encontrar el servidor. El tiempo de respuesta de la conexión fue de 0,864 segundos. El tiempo de respuesta para los mensajes de eco tiene un promedio de 0,679 segundos. De acuerdo a la topología de red tipo BUS utilizada y la organización que tenían los nodos en la red, el nodo cliente encontró al nodo servidor más cercano a él.

Tabla 9. Escenario de Prueba No. 4

5.4.3.5. Escenario de Prueba No. 5

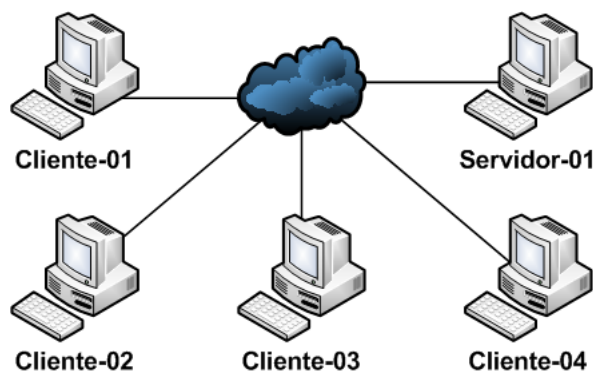


Figura 41. Representación del Escenario de Prueba No. 5

FORMATO DE PRUEBAS				
Número de Escenario: 5				
Nodo Cliente	Número de Mensajes	Tiempo de Respuesta	Nodo Servidor	Estado del Servidor
Cliente-01	12	0,726 segundos	Servidor-01	Ocupado
Cliente-02	9	0,752 segundos	Servidor-01	Ocupado
Cliente-03	2	0,835 segundos	Servidor-01	Ocupado
Cliente-04	13	0,737 segundos	Servidor-01	Ocupado

Descripción de Resultados:
Los clientes comienzan el envío de mensajes de búsqueda, conectándose al único servidor presente en la red. En este escenario se puede ver que en tiempo de respuesta del servidor, no tiene nada que ver con el número de mensajes de consulta que envía el cliente, como puede verse, el Cliente-03 solo envió dos mensajes de consulta, pero el tiempo de respuesta fue más alto que los demás. El promedio del tiempo de respuesta fue de 0,7625 segundos. El tiempo de

respuesta para los mensajes de eco tiene un promedio de 0,636 segundos.

Tabla 10. Escenario de Prueba No. 5

5.4.3.6. Escenario de Prueba No. 6

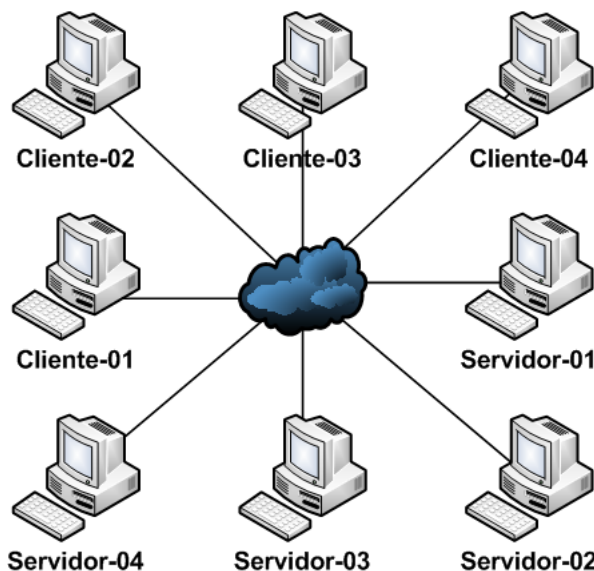


Figura 42. Representación del Escenario de Prueba No. 6

FORMATO DE PRUEBAS				
Número de Escenario: 6				
Nodo Cliente	Número de Mensajes	Tiempo de Respuesta	Nodo Servidor	Estado del Servidor
Cliente-01	6	0,648 segundos	Servidor-01	Ocupado
Cliente-02	16	0,793 segundos	Servidor-01	Ocupado
Cliente-03	12	0,661 segundos	Servidor-01	Ocupado
Cliente-04	9	0,550 segundos	Servidor-02	Ocupado
Cliente-05	4	0,769 segundos	Servidor-02	Ocupado
Cliente-06	24	0,897 segundos	Servidor-03	Ocupado
Cliente-07	11	0,671 segundos	Servidor-03	Ocupado
Cliente-08	17	0,759 segundos	Servidor-03	Ocupado
-	-	-	Servidor-04	Libre
-	-	-	Servidor-05	Libre
-	-	-	Servidor-06	Libre
-	-	-	Servidor-07	Libre
-	-	-	Servidor-08	Libre

Descripción de Resultados:
Los clientes se activan consecutivamente para el envío de mensajes de búsqueda. Se evidencia la recepción de mensajes en los servidores más cercanos a los clientes. Como puede verse, los clientes no se conectan todos al servidor más cercano, sino que se reparte la carga de conexiones aleatoriamente. Nuevamente se encuentra, que el número de mensajes de consulta, no es proporcional al

tiempo de respuesta.

El promedio del tiempo de respuesta fue de 0,7185 segundos. El tiempo de respuesta para los mensajes de eco tiene un promedio de 0,698 segundos.

En este escenario se prueba el proceso de recuperación del servicio, empezando a conectar y desconectar de la red los nodos cliente y los nodos servidor aleatoriamente, para probar el manejo del cache de mensajes y los métodos de índices locales y accesos directos. En la Figura 43 y en la Figura 44 se muestra el momento en que se realiza el proceso de activación y desactivación de los nodos en la sala de informática.

Cuando un cliente se desactiva, el acceso directo del último servidor utilizado queda almacenado, por lo tanto cuando el cliente se activa nuevamente, este envía el mensaje de conexión directamente al servidor representado por el acceso directo, si el servidor se encuentra conectado, el cliente empieza el proceso de consumo del servicio; si por el contrario, el servidor ya no está conectado a la red, el cliente elimina el acceso directo y empieza un nuevo proceso de búsqueda, encontrando satisfactoriamente un nuevo nodo servidor. Además cuando un cliente se desactiva, el servidor que lo tenía almacenado en sus índices locales, lo elimina al no poder enviarle un mensaje de respuesta.

Cuando un servidor al que están conectados uno o muchos clientes se desactiva, el servicio se cae, y los nodos cliente que estaban conectados a él empiezan a guardar sus mensajes de eco en el cache de mensajes. Cuando encuentran un nuevo servidor, crean el directorio de recuperación y se lo envían al nuevo servidor.

Nota: En la Figura 42, se muestra la representación del escenario de prueba, con 4 clientes y 4 servidores. La representación, para más nodos, no se realiza gráficamente por motivos de espacio cuando se dibujan los nodos.

Tabla 11. Escenario de Prueba No. 6



Figura 43. Activando y Desactivando Nodos Servidor



Figura 44. Activando y Desactivando Nodos Cliente

5.4.4. Características Probadas

Las características de los métodos de búsqueda que se analizaron en las pruebas, se describen en la Sección 2.4. Características de los Métodos de Búsqueda, y fueron las siguientes:

5.4.4.1. Desempeño

Por medio de los métodos de Índices Locales y Accesos Directos aplicados en la técnica, se incrementa el desempeño de la búsqueda, debido a que no se necesita realizar el proceso típico del esquema de inundación cada vez que se quiera enviar un mensaje; en cambio, se utiliza la información proporcionada por estos métodos para realizar el proceso.

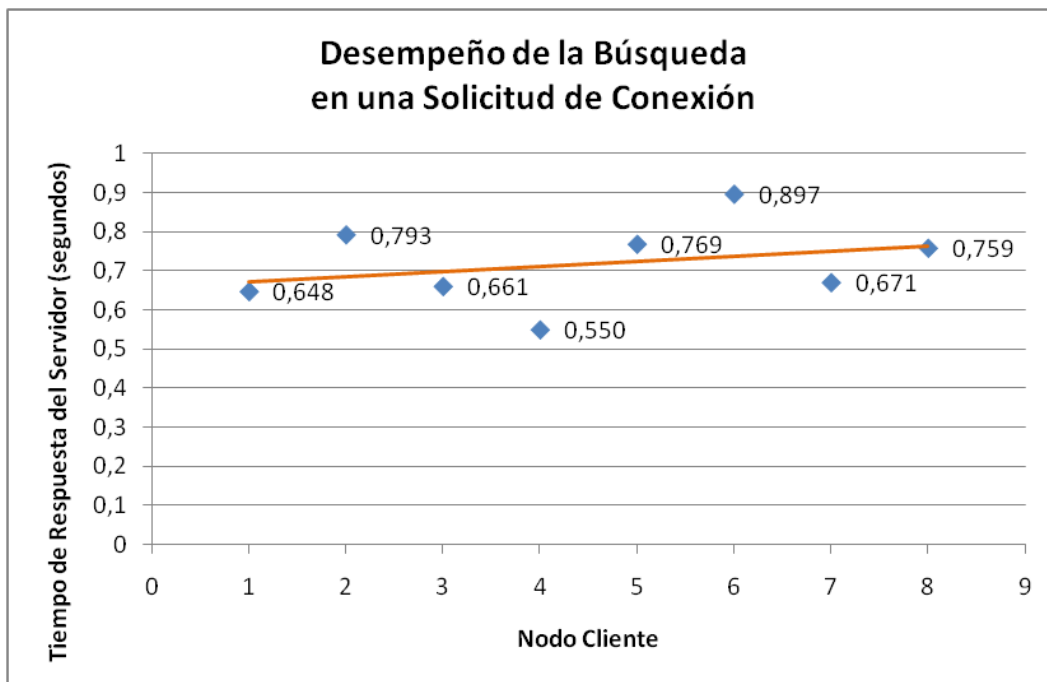


Figura 45. Gráfica de Desempeño de la Búsqueda

En la Figura 45, se muestra el desempeño de la búsqueda cuando un nodo cliente realiza una solicitud de conexión a un nodo servidor. En la gráfica puede verse el tiempo de respuesta del nodo servidor, para cada uno de los ocho nodos cliente. El promedio del tiempo de conexión y respuesta del servidor es de 0,7185 segundos.

5.4.4.2. Tolerancia a Fallos

El principal aporte de la técnica, implementado mediante el cache de mensajes y el directorio de recuperación. Los resultados de las pruebas muestran que los

clientes no se ven afectados si se cae un servidor y después se conectan a otro, por el contrario el envío de mensajes continua normalmente y ningún mensaje de eco se pierde en el proceso de recuperación.

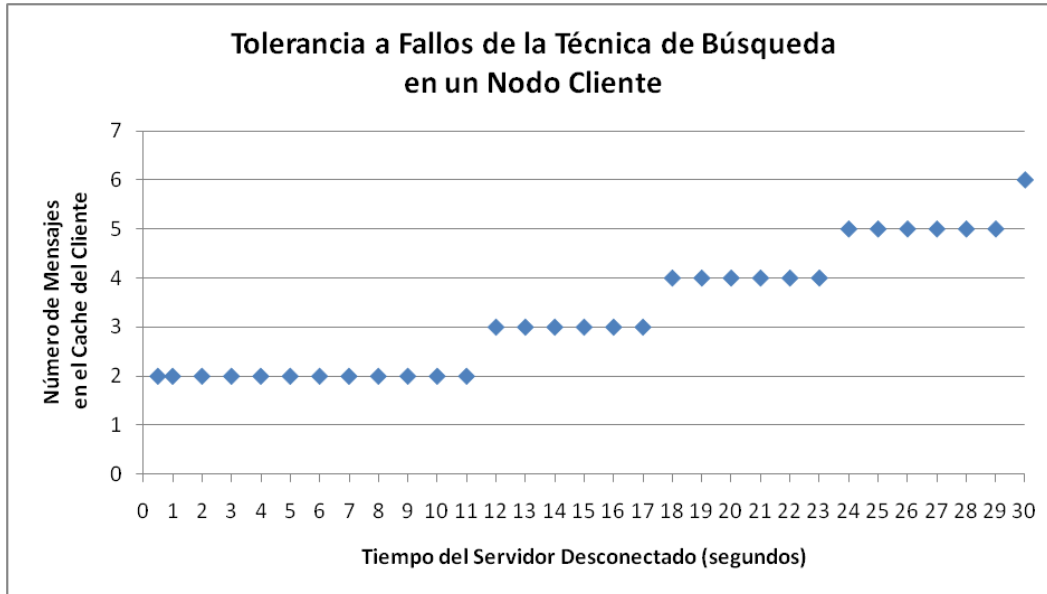


Figura 46. Gráfica de Tolerancia a Fallos (a)

En la Figura 46, se muestra el número de mensajes que almacena un nodo cliente en su cache de mensajes, dependiendo del tiempo que el nodo servidor este desconectado. Como el envío de mensajes de eco se realiza cada 6 segundos, el número de mensajes en el cache del cliente se incrementa en uno, cada 6 segundos. En la gráfica se muestran los mensajes que son almacenados en el cache hasta un tiempo de 30 segundos.

En la Figura 47, se muestra el tiempo de procesamiento de cada mensaje, cuando el servidor se desconecta durante 30 segundos. Como puede verse en la grafica, a medida que se procesan los mensajes que estaban guardados en el cache, el tiempo de respuesta de procesamiento disminuye, hasta estabilizarse cuando se envía el sexto mensaje. El intervalo entre cada uno de los primeros seis mensajes procesados, es aproximadamente de 6 segundos, que es el tiempo definido entre cada intervalo para el envío de mensajes de eco.

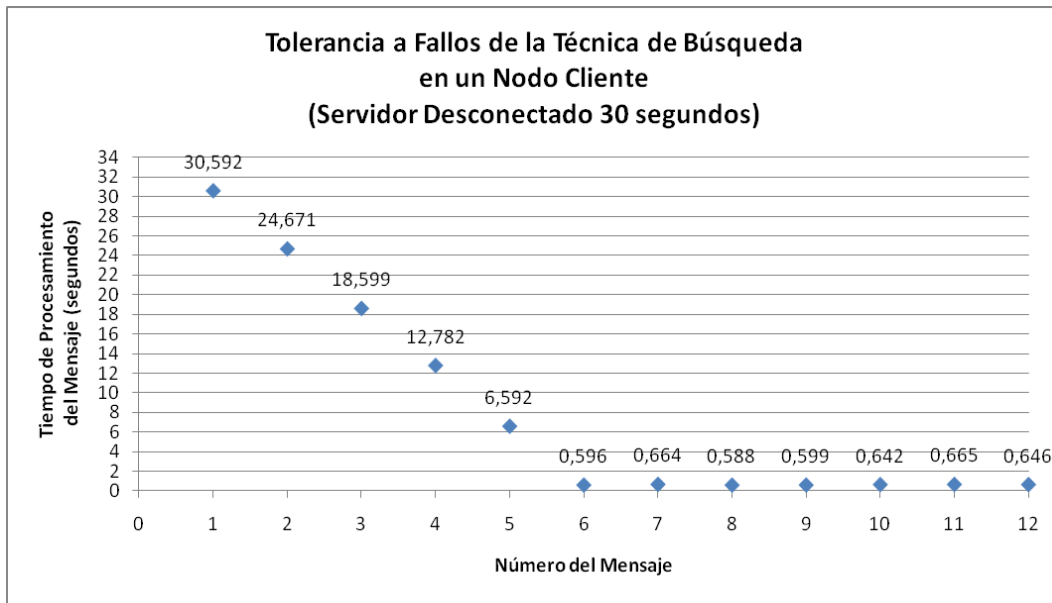


Figura 47. Gráfica de Tolerancia a Fallos (b)

5.4.4.3. Costo de la Búsqueda

Mediante el método de accesos directos, un nodo cliente, solo envía la consulta al nodo servidor al cual está conectado.

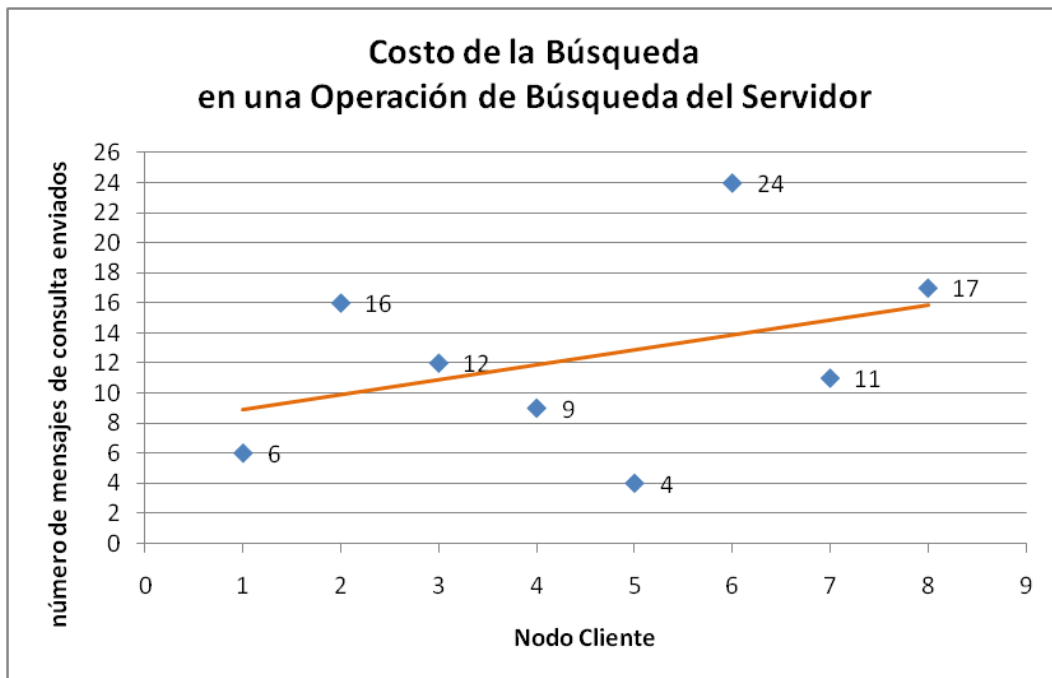


Figura 48. Gráfica de Costo de la Búsqueda

En la Figura 48, se muestra el costo de la búsqueda, representado por el número de mensajes de consulta que envía un nodo cliente antes de encontrar satisfactoriamente un nodo servidor. El promedio del costo de la búsqueda fue de 12,375 mensajes de consulta, antes de encontrar un nodo servidor.

5.4.4.4. Precisión de la Búsqueda

La eficiencia de la técnica, respecto a esta característica, consiste en poder traer una respuesta adecuada al mensaje de búsqueda enviado. Los resultados de las pruebas nos muestran que los clientes siempre encuentran el servicio, si hay, por lo menos, un servidor conectado.

5.4.4.5. Confiabilidad

Depende bastante de la tolerancia a fallos y la precisión de la búsqueda. Se demuestra con el éxito de la técnica, al momento de realizar el proceso de recuperación, sin ninguna pérdida de mensajes de eco. También se demuestra con la eficiencia de la técnica, debido a que siempre trae una respuesta adecuada al mensaje de búsqueda enviado.

5.4.4.6. Control Local

Los nodos tienen control total sobre sus recursos. Cada agente representa y maneja la información de un nodo en la red superpuesta P2P, además el agente tiene la capacidad de guardar en cache la información del proceso del servicio y del servidor actual al que está conectado, poder hacer la recuperación de mensajes cuando se pierde la conexión con un servidor.

El control local de la técnica de búsqueda, puede demostrarse con las siguientes afirmaciones:

- Los nodos tienen completo control sobre sus recursos.
- Los nodos guardan información sobre los nodos a los cuales están conectados.
- Los nodos cliente guardan su propio cache de los mensajes procesados hasta el momento.
- Los nodos cliente pueden recuperarse de una caída del servicio, gracias a su cache de mensajes.

5.4.4.7. Escalabilidad

En el desarrollo del marco experimental de estas pruebas, se llegó a establecer un tope real de 16 nodos conectados simultáneamente (8 Clientes y 8 Servidores). Para probar una escalabilidad más grande, se lanzaron 100 nodos en un mismo

equipo, cada uno tenía la funcionalidad tanto de cliente como de servidor. Poco a poco se empezaron a reducir el número de servidores, hasta que quedara solo un servidor y todos los clientes conectados a este. En este experimento se noto una disminución en el tiempo de respuesta, debido a que todos los nodos se ejecutan en el mismo equipo, pero el resultado fue el esperado. En la Figura 49, se muestran 4 nodos realizando el proceso de envío de mensajes de eco.

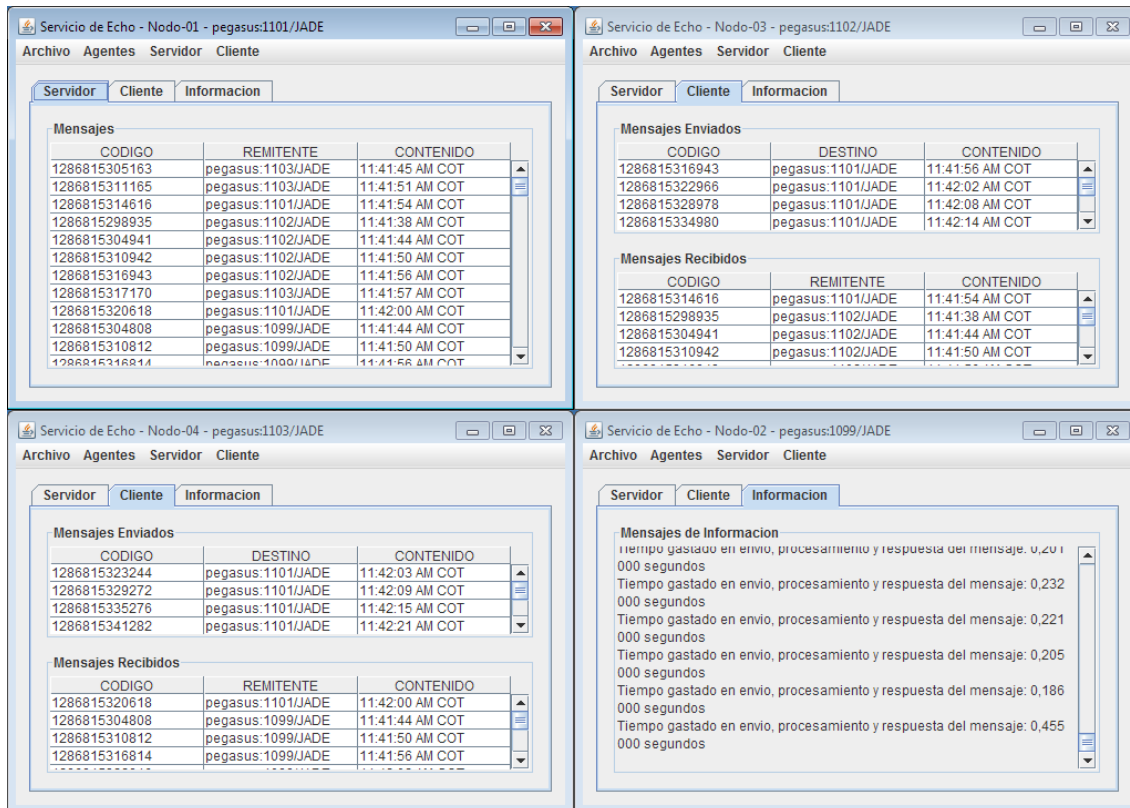


Figura 49. Escenario de Escalabilidad

Como puede verse en el nodo inferior derecho, el tiempo de envío de mensajes de eco se reduce por estar en un mismo equipo. En la imagen el tiempo de respuesta varía entre 0,186 segundos y 0,455 segundos.

En la Figura 50, se muestra el tiempo de respuesta del servidor en una solicitud de conexión, para un total de 32 nodos cliente. El promedio del tiempo de respuesta del servidor está entre 0,16 y 0,18 segundos. Como puede notarse en la gráfica el desempeño de la búsqueda no disminuye cuando se conectan muchos más nodos al sistema, por lo tanto puede concluirse que el sistema es escalable y tiene un buen desempeño.

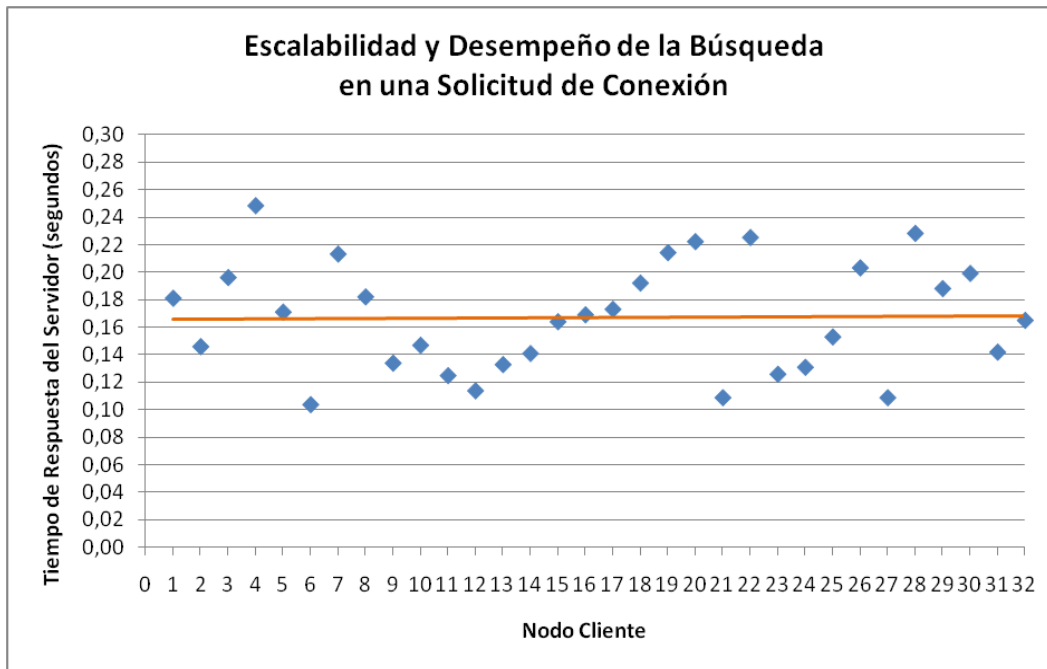


Figura 50. Gráfica de Escalabilidad y Desempeño de la Búsqueda

Las siguientes características, no se analizan, debido a que, como se menciona en la Sección 2.6. Selección del Método de Búsqueda, es mejor obtener una buena calidad de los resultados y un número de hits alto, sacrificando el costo de ancho de banda y procesamiento:

- **Flexibilidad de la Búsqueda:** La técnica de búsqueda no utiliza ningún mecanismo para proveer flexibilidad. La búsqueda se realiza por el nombre y el tipo del servicio.
- **Costo de Ancho de Banda:** Se sacrifica esta característica para obtener una buena calidad de resultados y un número de hits alto.
- **Costo de Almacenamiento:** Se sacrifica esta característica para obtener una buena calidad de resultados y un número de hits alto.

CAPÍTULO VI. CONCLUSIONES

- En este trabajo de grado, se mencionó que las redes superpuestas P2P, tienen las capacidades para prestar una gran cantidad de servicios. Pero desafortunadamente, se encuentran varios problemas en la forma en que estas prestan los servicios, debido a que no existe un estándar establecido para la definición de estos y cada una de las implementaciones, de este tipo de redes, crean su propia serie de protocolos P2P, lo que impide la comunicación entre ellas. Por lo tanto, se necesitan mecanismos para la interconexión y la interoperabilidad entre las redes superpuestas P2P.
- 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 definieron y describieron las características de los agentes y los Sistemas Multi-Agente, y las ventajas que presentan este tipo de sistemas. La comunicación en los Sistemas Multi-Agente, es un componente principal de estos, debido a que, los agentes deben tener la capacidad de cooperar, colaborar, negociar y realizar otras tareas que lo ameriten, para poder cumplir sus objetivos. El lenguaje que utilizan los agentes para su comunicación es ACL, el cual permite una separación entre la comunicación y el lenguaje del contenido, permitiendo usar ontologías para especificar y comunicar el conocimiento del dominio del problema de una manera genérica y darle contenido semántico los mensajes ACL.
- FIPA, desarrolla una colección de estándares relacionados con la tecnología de agentes software. Por lo tanto, se estudió el conjunto de especificaciones, que define FIPA, para la comunicación y gestión de agentes. Gracias al estudio realizado, se encontró que, FIPA define un conjunto de estándares para la gestión de agentes, el cual establece un modelo de referencia lógica para la creación, registro, localización, comunicación, migración y operación de agentes. Además, FIPA-ACL es el lenguaje de comunicación de agentes, más estudiado y utilizado actualmente.
- Las investigaciones realizadas sobre redes superpuestas P2P, y sobre agentes y Sistemas Multi-Agente, evidenciaron que existe una gran relación entre estos dos sistemas, lo que permite que los beneficios de esta relación, se usen para mejorar la calidad de los servicios prestados.
- Existe una gran variedad de métodos de búsqueda de servicios, en las redes superpuestas P2P 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 P2P. Por otro lado, en los Sistemas Multi-Agente, existe el servicio de páginas amarillas, el cual permite que los agentes puedan crear, publicar y buscar uno o más servicios. El servicio de páginas amarillas, también permite a los agentes descubrir dinámicamente otros agentes. Gracias al análisis de los métodos de búsqueda y sus características, se eligieron tres métodos de búsqueda, para ser utilizados en la Técnica de Búsqueda propuesta, estos tres métodos fueron: *Índices Locales*, *Accesos Directos* y *Súper-Nodos*.

- La adecuación de la técnica de búsqueda para la prestación de servicios, se centró en las características que pueden brindar los Sistemas Multi-Agente para la publicación, localización y búsqueda de los servicios. Además, las características de transporte y comunicación, de las redes superpuestas P2P, se utilizaron para la adecuación de la técnica.
- La técnica de búsqueda tiene tres componentes: El Sistema Multi-Agente, El Sistema P2P y El Comunicador, los cuales tienen como principal característica, la independencia de su funcionamiento. El Sistema Multi-Agente, cumple con las especificaciones FIPA, e incluye todos los elementos obligatorios para administrar la plataforma de agentes y realizar las comunicaciones entre los agentes. El Sistema P2P, es el encargado de representar el servicio, en la red superpuesta P2P, y de las comunicaciones entre cada nodo de la red. El Comunicador, es el encargado de las comunicaciones entre las dos plataformas y realiza la tarea de conversión entre los tipos de mensajes, los tipos de objetos y los parámetros de los respectivos métodos que cada una de las plataformas maneja.
- La prestación de servicios, en la técnica de búsqueda, se divide en tres procesos principales: Creación del Servicio, Búsqueda del Servicio y Recuperación del Servicio. La creación del servicio, es el proceso mediante el cual, un agente crea y publica el servicio en el sistema. La búsqueda del servicio, es el proceso mediante el cual, un agente busca el servicio requerido en el sistema. La recuperación del servicio, es el proceso mediante el cual, un agente que ha perdido la conexión a un servidor, vuelve a conectarse, sin perder los datos del proceso que está realizando.
- El mayor aporte, de la técnica de búsqueda, se ve reflejado en la recuperación del servicio y los procesos que se llevan a cabo en este. En el proceso de recuperación del servicio, se hace uso de los tres métodos de búsqueda seleccionados. Primero, se usa el método de súper-nodos, para encontrar el servidor; Segundo, se usa el método de accesos directos, para guardar el ID del servidor, como un acceso directo en el cliente; Tercero, se usa el método de índices locales, para enviar el ID del cliente al servidor, y todos los

mensajes que no han sido procesados, así el servidor guarda una lista de todos sus clientes y sus recursos, los cuales son representados por cada mensaje en el directorio de recuperación.

- Se estudiaron las tecnologías de desarrollo, JXTA y JADE, para realizar la implementación de la red superpuesta P2P y el Sistema Multi-Agente en el prototipo software. JXTA, es una plataforma, diseñada para computación P2P, la cual contiene un conjunto de protocolos P2P, que permiten a cualquier dispositivo de red, comunicarse y colaborar mutuamente como peers. JADE, es un framework, para desarrollar aplicaciones distribuidas basadas en agentes, en conformidad con las especificaciones FIPA, para la interoperabilidad inteligente de Sistemas Multi-Agente, simplificando su implementación.
- Se desarrolló un prototipo software, con el fin de verificar la técnica de búsqueda para la prestación de servicios sobre redes superpuestas P2P no estructuradas. Para realizar dicho proceso, se empleó la metodología de desarrollo Agile Unified Process (AUP), utilizando técnicas de modelado ágil. Se demostró, que la metodología AUP, fue apta para el proceso de desarrollo, de acuerdo al alcance del proyecto.
- Los resultados obtenidos, en la aplicación de las pruebas, para la verificación de la técnica de búsqueda, demostraron que la técnica realiza sus procesos de una forma correcta y se comporta de la manera esperada. Por lo tanto, puede afirmarse que la técnica de búsqueda, funciona correctamente y muestra resultados satisfactorios en la creación, búsqueda y recuperación de servicios en redes superpuestas P2P, mediante el uso de Sistemas Multi-Agente.

CAPÍTULO VII. RECOMENDACIONES Y TRABAJO FUTURO

- Se propone la implementación de un servicio más completo y avanzado, para probar como se comportaría la técnica de búsqueda. Sería interesante probar la técnica de búsqueda, por ejemplo, con servicios como: Voz IP o Televisión IP Interactiva. Estos servicios, envían flujos de datos constantemente, y es indispensable, que la comunicación entre los nodos sea muy fluida y no se pierda información relevante. Además, si se desea y se cree conveniente, se pueden usar otras tecnologías de desarrollo para el prototipo software, en vez de usar JXTA, JADE y los estándares FIPA que fueron usados en el desarrollo del prototipo software.
- Se propone verificar la técnica de búsqueda propuesta, con un conjunto más amplio de servicios, lo que permitirá probar más ampliamente la búsqueda y localización de un servicio específico, así como también las características de búsqueda que se priorizaron.
- Se propone realizar una comparación, de la técnica de búsqueda para la prestación de servicios sobre redes superpuestas P2P no estructuradas, con otras técnicas o métodos de búsqueda existentes, para verificar su desempeño en relación con las demás.
- 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 búsqueda, 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 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.
- Extender la técnica de búsqueda 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] E. Fuentes, A. Segovia, and E. Jeltsch. (2006) Phase 0 Network. *Revista Ingeniería Informática, Edición 12, Abril 2006*. Disponible: <http://www.inf.udec.cl/~revista/ediciones/edicion12/articulo%2012-1.pdf>
- [2] R. Schollmeier, "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications", presented at the Proceedings of the First International Conference on Peer-to-Peer Computing, 2001.
- [3] V. Olmedo, V. Villagrà, E. Pastor, and J. Berrocal, "El Papel de las Redes Superpuestas en la Evolución de la Arquitectura de Internet", presented at the XVI Jornadas Telecom I+D 2006, Madrid, 2006.
- [4] M. Sánchez, "A Hierarchical Framework for Peer-to-Peer Systems: Design and Optimizations", Tesis Doctoral, Universitat Pompeu Fabra, 2008.
- [5] V. Raisanen, *Service Modeling: Principles and Applications*: John Wiley & Sons, Ltd., 2006.
- [6] F. Bordignon and G. Tolosa. (2003) Redes Compañero a Compañero (P2P): Conceptos y Tendencias de Aplicación. *Novática: Revista de la Asociación de Técnicos de Informática, Nº.166, 2003 (Ejemplar dedicado a: Planes de contingencia TIC y continuidad de negocio)*. 57-60. Disponible: <http://dialnet.unirioja.es/servlet/articulo?codigo=2063988>
- [7] P2PRG. (2008, *Internet Research Task Force. Peer-to-Peer Research Group*. Disponible: <http://www.irtf.org/charter?gtype=rg&group=p2prg>
- [8] Osiatis. (2006, *ITIL - Gestión de Servicios TI. Fundamentos de la Gestión TI. Provisión del Servicio*. Disponible: http://itil.osiatis.es/Curso_ITIL/Gestion_Servicios_TI/fundamentos_de_la_gestion_TI/que_es_ITIL/provision_del_servicio.php
- [9] K. J. Turner, E. H. Magill, and D. J. Marples, *Service Provision: Technologies For Next Generation Communications*: John Wiley & Sons, Ltd., 2004.
- [10] R. Mondéjar, P. García, and C. Pairet, "Bunshin: DHT para aplicaciones distribuidas", presented at the XIII Jornadas de Concurrencia y Sistemas Distribuidos (JCSD). I Congreso Español de Informática (CEDI 2005), Granada, España, 2005.
- [11] J. Lu, "Full-text federated search in peer-to-peer networks", *SIGIR Forum*, vol. 41, pp. 121-121, 2007.

- [12] RAE. (2010, *Real Academia Española. Diccionario de la Lengua Española*. Disponible: <http://www.rae.es/rae.html>)
- [13] B. Yang and H. Garcia-Molina, "Improving Search in Peer-to-Peer Networks", presented at the Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02), 2002.
- [14] Palowireless. (2010, *Bluetooth Resource Center. Service Discovery Protocol (SDP)*. Disponible: <http://www.palowireless.com/infotooth/tutorial/sdp.asp>)
- [15] J. D. Gradecki, *Mastering Jxta: Building Java Peer-to-Peer Applications*: John Wiley & Sons, Inc., 2002.
- [16] B. J. Wilson, *JXTA Book*: New Riders Publishing, 2002.
- [17] D. Tsoumakos and N. Roussopoulos, "Analysis and comparison of P2P search methods", presented at the Proceedings of the First International Conference on Scalable Information Systems, Hong Kong, 2006.
- [18] Gnutella. (2010, *Gnutella*. Disponible: <http://www.gnutella.org/>)
- [19] D. Brookshier, D. Govoni, N. Krishnan, and J. C. Soto, *JXTA: Java P2P Programming*: Sams Publishing, 2002.
- [20] F. L. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*: John Wiley & Sons, Ltd., 2007.
- [21] M. Wooldridge, *An Introduction to MultiAgent Systems*: John Wiley & Sons, Ltd., 2002.
- [22] R. Mondéjar, J. Pujol, P. García, and C. Pairot, "Sistemas multi-agente en entornos P2P", Department of Computer Science and Mathematics, Universitat Rovira i Virgili, Tarragona, Spain, 2006.
- [23] FIPA. (2010, *The Foundation for Intelligent Physical Agents*. Disponible: <http://www.fipa.org/>)
- [24] M. Afsharchi and B. H. Far, "Automated ontology evolution in a multi-agent system", presented at the Proceedings of the 1st international conference on Scalable information systems, Hong Kong, 2006.
- [25] T. Gruber, "The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases", presented at the Principles of Knowledge

- Representation and Reasoning: Proceedings of the Second International Conference, Cambridge, MA, 1991.
- [26] G. Barchini, M. Alvarez, S. Herrera, and M. Trejo. (2007) El Rol de las Ontologías en los SI. *Revista Ingeniería Informática, Edición 14, Mayo 2007*. Disponible: <http://www.inf.udec.cl/~revista/ediciones/edicion14/barchini.pdf>
- [27] L. J. Suarez and L. A. Rojas, "Descubrimiento de Servicios en Ambientes Ubicuos", Departamento de Telemática, Universidad del Cauca, Popayán, 2010.
- [28] J. C. Corrales, "Bematch: A platform for matchmaking service behavior models", presented at the Proceedings of EDBT, Nantes, France, 2008.
- [29] G. Suryanarayana and R. N. Taylor, "A Survey of Trust Management and Resource Discovery Technologies in Peer-to-Peer Applications", Institute for Software Research, University of California, Irvine, 2004.
- [30] F. Yuan, J. Liu, and C. Yin, "A study of scalable search algorithm on unstructured P2P system", presented at the Proceedings of the 2nd international conference on Scalable information systems, Suzhou, China, 2007.
- [31] K. Sripanidkulchai, B. Maggs, and H. Zhang, "Efficient content location using interest-based locality in peer-to-peer systems", in *IEEE Infocom*, 2003.
- [32] H. Chen, H. Jin, J. Wang, L. Chen, Y. Liu, and L. M. Ni, "Efficient multi-keyword search over p2p web", presented at the Proceeding of the 17th international conference on World Wide Web, Beijing, China, 2008.
- [33] H. Chen, H. Jin, X. Luo, Y. Liu, and L. M. Ni, "BloomCast: Efficient Full-Text Retrieval over Unstructured P2Ps with Guaranteed Recall", presented at the Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid - Volume 00, 2009.
- [34] JXTA. (2010, *JXTA Community Project*. Disponible: <http://www.jxta.org/>
- [35] JADE. (2010, *Java Agent Development Framework*. Disponible: <http://jade.tilab.com/>
- [36] AUP. (2006, *The Agile Unified Process v1.1*. Disponible: <http://www.ambysoft.com/unifiedprocess/agileUP.html>