

**DEFINICION DE PERFILES UML PARA SERVICIOS TELEMATICOS ORIENTADOS A  
UNA ARQUITECTURA P2P**



**DIEGO RICARDO PAREDES**

**DAVID ALEXANDER RUEDA RAMIREZ**

**Director: Ing. PABLO AUGUSTO MAGE IMBACHI**

**UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES  
DEPARTAMENTO DE TELEMATICA  
POPAYÁN, JUNIO DE 2008**

# Contenido

CAPITULO 1 INTRODUCCION.....	1
JUSTIFICACIÓN .....	1
OBJETIVOS .....	4
OBJETIVO GENERAL .....	4
OBJETIVOS ESPECÍFICOS .....	4
CONTRIBUCIONES .....	4
CAPITULO 2 MARCO TEORICO.....	5
2.1 SERVICIOS TELEMÁTICOS Y ARQUITECTURAS ORIENTADAS A SERVICIOS.....	6
2.2 MODELO P2P: .....	8
2.3 ORÍGENES DE LOS SISTEMAS P2P SOBRE INTERNET .....	8
2.3.1 USENET .....	9
2.3.2 DNS .....	10
2.4 CARACTERÍSTICAS DE LOS SISTEMAS P2P .....	12
2.5 SISTEMAS CLIENTE/SERVIDOR VERSUS SISTEMAS P2P .....	13
2.6 ÁREAS DE APLICACIÓN .....	14
2.6.1 TRABAJO EN GRUPO.....	14
2.6.2 COMPUTACIÓN DISTRIBUIDA.....	14
2.7 PLATAFORMAS P2P.....	17
2.7.1 NET.....	17
2.7.2 JXTA.....	17
2.7.3 SERVICIOS WEB .....	18
2.7.4 COMPUTACIÓN GRID.....	18
2.7.5 JINI [27] .....	18
2.8 PERFILES .....	19
CAPITULO 3 DISEÑO DE PERFILES P2P.....	21
3.1 FUNDAMENTOS DE DISEÑO .....	21
3.1.1 CARACTERÍSTICAS DE DISEÑO .....	21
3.1.2 PRINCIPIOS DE DISEÑO.....	22
3.1.3 PARADIGMAS DE DISEÑO .....	23
3.1.4 PATRÓN DE DISEÑO .....	24
3.1.5 LENGUAJE DE PATRÓN DE DISEÑO.....	25
3.1.6 ESTÁNDAR DE DISEÑO .....	26
3.1.7 MEJOR PRÁCTICA.....	27
3.2 INTRODUCCIÓN A LA COMPUTACIÓN ORIENTADA A SERVICIOS .....	30
3.2.1 ARQUITECTURA ORIENTADA A SERVICIOS .....	30
3.2.2 PARADIGMA ORIENTACIÓN DEL SERVICIO.....	30
3.2.3 COMPOSICIÓN DE SERVICIOS.....	31
3.2.4 INVENTARIO DE SERVICIOS .....	31
3.2.5 MODELOS DE SERVICIOS .....	33
3.2.6 OBJETIVOS Y BENEFICIOS DE LA COMPUTACIÓN ORIENTADA A SERVICIOS .....	34
3.3 PRINCIPIOS DEL SERVICIO - ACOPLAMIENTO DEL SERVICIO.....	34
3.3.1 REUTILIZACIÓN DEL SERVICIO .....	35
3.3.2 GESTIÓN DE ESTADO.....	36
3.3.3 COMPOSICIÓN DEL SERVICIO .....	37
3.4. ESTRUCTURA DEL PERFIL NIVEL – SERVICIO.....	38
3.5 ESTRUCTURA DE LA CAPACIDAD DEL PERFIL .....	39
CAPITULO 4 DEFINICIÓN DE PERFILES P2P.....	40
4.1 PERFIL DE SERVICIOS.....	40
4.2 DEFINICIÓN DE PERFILES UML PARA SERVICIOS EN ENTORNOS P2P – JXTA .....	40
4.2.1 SERVICIOS DE GRUPO DE PEERS .....	40
4.2.2 DEFINICIÓN DEL PERFIL UML PARA EL SERVICIO ESTÁNDAR DE DESCUBRIMIENTO.....	41
4.2.3 DEFINICIÓN DEL PERFIL UML SERVICIO ESTÁNDAR DE AFILIACIÓN O MEMBRECÍA.....	42
4.2.4 DEFINICIÓN DEL PERFIL UML PARA EL SERVICIO ESTÁNDAR DE ACCESO.....	44
4.2.5 DEFINICIÓN DEL PERFIL UML PARA EL SERVICIO ESTÁNDAR DE PIPE .....	45
4.2.6 DEFINICIÓN DEL PERFIL UML PARA EL SERVICIO ESTÁNDAR DE MONITOREO .....	46
4.2.7 ARQUITECTURA APLICACIÓN PROTOTIPO .....	48
4.3 PERFILES DERIVADOS.....	49
4.3.1 METAMODELO P2P SERVICIOS DE PLATAFORMA.....	49

4.3.2 ESPECIFICACIÓN FORMAL PERFILES DERIVADOS.....	50
4.3.4 PERFIL SERVICIOS DE PLATAFORMA.....	54
4.3.5 PERFIL DESCUBRIMIENTO.....	55
4.3.6 DIAGRAMA DE CLASES - PERFIL DE SERVICIOS DE PLATAFORMA.....	56
4.3.7 METAMODELO DE SERVICIOS DE AFILIACIÓN O SERVICIOS DE MEMBRECÍA .....	57
4.3.8 ESTRUCTURA DEL PERFIL - SERVICIO DE AFILIACIÓN.....	58
4.3.9 ESPECIFICACIÓN FORMAL CAPACIDADES PERFIL AFILIACIÓN.....	60
4.3.10 PERFIL SERVICIOS DE AFILIACIÓN Y GESTOR DE GRUPOS .....	61
4.3.11 ESPECIFICACIÓN FORMAL DE ESTEREOTIPOS PERFIL AFILIACIÓN Y GESTOR DE GRUPOS .....	62
4.3.12 DIAGRAMA DE CLASES PERFIL AFILIACIÓN.....	64
4.3.13 PERFIL DE COMUNICACIÓN.....	65
4.3.14 SERVICIO DE ENVIÓ DE MENSAJES ESTÁNDAR .....	66
4.3.15 ESPECIFICACIÓN FORMAL CAPACIDADES PERFIL ENVIÓ DE MENSAJES ESTÁNDAR.....	67
4.3.16 ESPECIFICACIÓN FORMAL SERVICIO DE ENVIÓ DE MENSAJES TIPO ARCHIVOS .....	68
4.3.17 ESPECIFICACIÓN FORMAL CAPACIDADES PERFIL ENVIÓ DE ARCHIVOS .....	69
4.4 ESPECIFICACIÓN FORMAL DE LOS ESTEREOTIPOS DE LOS PERFILES DEFINIDOS .....	72
4.4.1 ESTEREOTIPOS PERFIL ENVIÓ DE MENSAJES ESTÁNDAR .....	72
4.4.2 ESTEREOTIPOS PERFIL ENVIÓ DE MENSAJES TIPO ARCHIVO .....	73
4.4.3 DIAGRAMA DE CLASES PERFIL ENVIÓ DE MENSAJES ESTÁNDAR.....	74
4.4.4 DIAGRAMA DE CLASES PERFIL ENVIÓ DE MENSAJES TIPO ARCHIVO.....	74
CAPITULO 5 ESTRUCTURA GENERAL CASO DE ESTUDIO .....	76
5.1 DETECCIÓN Y MONITOREO DE USUARIOS AGENDA (PEERS) .....	76
5.2 ADMINISTRADOR DE GRUPOS DE TRABAJO .....	77
5.3 SERVICIO DE CHAT .....	79
5.4 ADMINISTRADOR DE AGENDA COMPARTIDA.....	80
5.5 ADMINISTRADOR DE EVENTOS .....	81
5.6 ADMINISTRADOR DE RECURSOS COMPARTIDOS .....	82
5.7 UTILIDAD DE BÚSQUEDAS .....	82
CAPÍTULO 6 CONCLUSIONES Y TRABAJOS FUTUROS.....	83
CAPITULO 7 BIBLIOGRAFÍA.....	85

## LISTA DE FIGURAS

FIGURA 1 REPRESENTACIÓN DE UNA RED SUPERPUESTA.....	1
FIGURA 2 PROTOCOLO JXTA.....	19
FIGURA 3 COMPARACIÓN TRES TIPOS DE DISEÑO.....	21
FIGURA 4 APLICACIÓN PRINCIPIO DE BAJO ACOPLAMIENTO.....	22
FIGURA 5 APLICACIÓN PARADIGMA.....	23
FIGURA 6 APLICACIÓN PATRÓN DE DISEÑO.....	24
FIGURA 7 LENGUAJE DE PATRÓN DE DISEÑO.....	25
FIGURA 8 ESTÁNDAR DE DISEÑO.....	27
FIGURA 9 MEJOR PRÁCTICA.....	28
FIGURA 10 RELACIÓN LAS PARTES DEL MARCO DE TRABAJO DE DISEÑO.....	29
FIGURA 11 RELACIÓN LAS PARTES DEL MARCO DE TRABAJO DE DISEÑO.....	29
FIGURA 12 SÍMBOLO SERVICIO.....	31
FIGURA 13 REPRESENTACIÓN DE COMPOSICIÓN DE SERVICIOS.....	31
FIGURA 14 SÍMBOLO INVENTARIO DE SERVICIOS.....	32
FIGURA 15 CLASIFICACIÓN MODELO PRIMARIO DE SERVICIO.....	34
FIGURA 16 CAPAS DE MODELO DEL SERVICIOS.....	34
FIGURA 17 REDUCCIÓN ACOPLAMIENTO.....	35
FIGURA 18 REUTILIZACIÓN DEL SERVICIO.....	36
FIGURA 19 COMPOSICIÓN DEL SERVICIO.....	37
FIGURA 20 PERFIL UML PARA EL SERVICIO ESTÁNDAR DE DESCUBRIMIENTO DE JXTA.....	42
FIGURA 21 PERFIL UML PARA EL SERVICIO ESTÁNDAR DE MEMBRECÍA DE JXTA.....	44
FIGURA 22 PERFIL UML PARA EL SERVICIO ESTÁNDAR DE ACCESO DE JXTA.....	45
FIGURA 23 PERFIL UML PARA EL SERVICIO ESTÁNDAR DE PIPE DE JXTA.....	46
FIGURA 24 PERFIL UML PARA EL SERVICIO ESTÁNDAR DE PIPE DE JXTA.....	47
FIGURA 25 ARQUITECTURA APLICACIÓN PROTOTIPO.....	48
FIGURA 26 PERFIL SERVICIOS DE PLATAFORMA.....	54
FIGURA 27 DIAGRAMA DE CLASES PERFIL SERVICIOS DE PLATAFORMA.....	56
FIGURA 28 METAMODELO DE SERVICIOS DE AFILIACIÓN O SERVICIOS DE MEMBRECÍA.....	57
FIGURA 29 PERFIL SERVICIOS DE AFILIACIÓN Y GESTOR DE GRUPOS.....	61
FIGURA 30 DIAGRAMA DE CLASES PERFIL AFILIACIÓN.....	64
FIGURA 31 PERFIL SERVICIOS DE COMUNICACIÓN.....	71
FIGURA 32 DIAGRAMA DE CLASES PERFIL ENVÍO DE MENSAJES ESTÁNDAR.....	74
FIGURA 33 DIAGRAMA DE CLASES PERFIL ENVÍO DE MENSAJES TIPO ARCHIVO.....	75
FIGURA 34 MONITOREO USUARIOS AGENDA ONLINE.....	76
FIGURA 35 SELECCIÓN DE CREAR GRUPO.....	77
FIGURA 36 CREACIÓN DE GRUPO.....	77
FIGURA 37 SELECCIÓN DE GRUPO A UNIRSE.....	78
FIGURA 38 USUARIO EN EL NUEVO GRUPO.....	78
FIGURA 39 USUARIOS AGENDA.....	79
FIGURA 40 CHAT USUARIOS AGENDA.....	79
FIGURA 41 AGENDA SEMANAL.....	80
FIGURA 42 AGENDA SEMANAL.....	80
FIGURA 43 ADMINISTRADOR DE EVENTOS.....	81
FIGURA 44 ADMINISTRADOR DE RECURSOS COMPARTIDOS.....	82
FIGURA 45 CONSULTA DE GRUPOS.....	82

## AGRADECIMIENTOS

Agradecer a mi padre Luis Roberto Rueda y madre Doris Yolanda Ramírez Belarcazar que con ayuda y sustento de Dios me dieron el mejor regalo el estudio. Gracias por creer y apoyar mis sueños y ayudarme ha hacerlos realidad. A mis hermanos por su apoyo y compañía a lo largo de la vida y a mi hijo Andrés David Rueda por ser mi principal inspiración y alegría y demás familiares cercanos que estuvieron apoyándome en el transcurso de la carrera.

David Rueda Ramírez.

Quiero agradecer desde el fondo de mi corazón a mi madre Maura Peña que gracias a su enorme amor, esfuerzo y coraje me apoyo siempre incansablemente y con muchos sacrificios para que consiga mis metas, a mi padre que desde el cielo me cuida y me protege, a mi tío Hugo Peña, mi tía Ana Lucia Paredes, por su incondicional apoyo, a mi abuela Natalia Peña por sus enseñanzas y cariño, a mis hermanos John Jairo y Adriana, a mi novia Anyela y a toda mi familia, infinitas gracias.

Ricardo Paredes

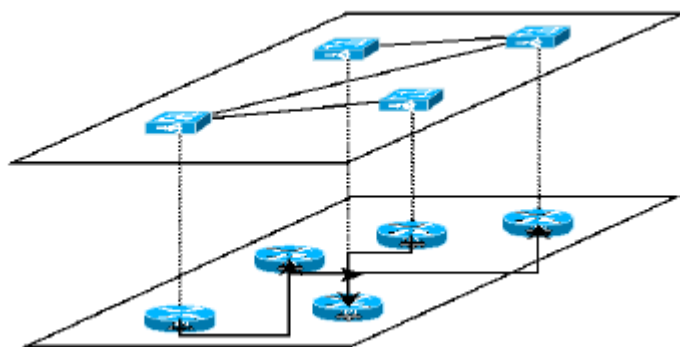
De igual manera nuestro más sincero agradecimiento al Director Pablo Augusto Mage Imbachí por su gran dirección, por ser un gran guía y el principal generador de conocimiento para el desarrollo exitoso del proyecto. Un agradecimiento especial a nuestros docentes: Jorge Jair Moreno Chaustre, Carlos Alberto Ardila Albarracín, Carlos Alberto Cobos Lozada, Miguel Angel Niño Zambrano Pablo Augusto Magé Imbachí, Juan Carlos Vidal, Cesar Alberto Collazos Ordóñez, Luz Marina Sierra Martínez, Roberto Carlos Naranjo Cuervo, Martha Eliana Mendoza Becerra, Erwin Meza Vega, Siler Amador Donado, Ember Ubeimar Martínez Flor y Wilson Libardo Pantoja Yepez; quienes desde el comienzo de la carrera y hasta el final no solo nos formaron como profesionales excelentes sino también como personas a través de su ejemplo, convirtiéndose en nuestros amigos antes que en docentes.

Gracias a todo el personal de la Facultad de Ingeniería Electrónica y Telecomunicaciones, como de Secretaría General, Dirección, Recepción, Post-Grado, Administración, TIC's, Biblioteca, Mantenimiento, Limpieza y Fotocopias, ya que dentro de los espacios que a cada uno le competen nos han asistido eficazmente y amablemente. A nuestros compañeros de carrera: Elvis Herley Perez Hernández, Germain Bolaños Vidal, Diego Fernando Bayona Valverde. Alex Calvache, Jorge Galindes, Yenny Mendez, Edwin Fernando Lara, Gabriel Muñoz, Alex Rosero, por su gran amistad, por el apoyo y animo en todos los momentos buenos y malos a lo largo del desarrollo de la carrera.

Autores: David Alexander Rueda Ramírez, Diego Ricardo Paredes Peña.

## CAPITULO 1 INTRODUCCION

Las redes superpuestas ON's no son un nuevo paradigma en las arquitecturas de comunicaciones, pues podemos observar que muchas de las redes actuales empezaron como redes superpuestas; el ejemplo mas claro es la Internet, la cual se formo como una superposición a la red telefónica. Una red superpuesta (Figura 1) se construye sobre una o varias redes, agregando una capa virtual sobre la red subyacente donde los nodos forman canales de comunicación superpuestos.



**Figura 1 Representación de una Red Superpuesta**

En el espectro de redes que se pueden considerar como superpuestas están los Web Services, la computación en malla, y las redes igual a igual (P2P).

Una ON's P2P posee una arquitectura de red distribuida, donde sus participantes comparten recursos, los cuales son indispensables para proporcionar los servicios y contenidos de la misma, dichos servicios y contenidos son accesibles por otros miembros sin pasar por intermediarios, convirtiéndose así en proveedores y/o consumidores. Siguiendo esta filosofía se han despejado nuevos caminos con el surgimiento de nuevas tecnologías y plataformas para la construcción de nuevos servicios y protocolos, esto sumado a la concepción de diseño de software orientado a servicios SOA [1] la cual se guía en la interoperabilidad y la estandarización de las funcionalidades de las aplicaciones, se tienen nuevos campos por explotar.

### JUSTIFICACIÓN

Los sistemas de información telemáticos tienen comprometidas sus características de disponibilidad, eficiencia y rendimiento en cierta medida con las bondades y limitaciones presentes en la arquitectura en la que están soportados, dicho así, la arquitectura y la plataforma se convierten en un punto crítico a la hora de construir software para la prestación de determinados servicios; basados en esto podemos observar que uno de los modelos de arquitectura que ha tenido más popularidad y acogida es la de los sistemas Cliente/Servidor, siendo la más tradicional y difundida entre las existentes. Sin embargo

se presentan ciertas limitantes y vulnerabilidades en este modelo, entre las cuales podemos hacer mención a los siguientes casos:

- Cuando el número de clientes y solicitudes al servidor aumentan, también se incrementa el consumo de recursos que este necesita para poder responder a dichas solicitudes.
- El estado del sistema depende de forma crítica del estado del Servidor, por lo cual este se convierte en el blanco de los ataques informáticos, es decir, si el Servidor se bloquea o se apaga entonces el sistema colapsa también.
- En la arquitectura Cliente/Servidor se debe hacer una gran inversión en el hardware del Servidor ya que de su desempeño depende el funcionamiento normal de los servicios que presta.
- El Cliente tiene un rol pasivo, que solo se limita a realizar peticiones al Servidor y esperar las respuestas a dichas peticiones, pero es incapaz de proveer servicios a otros Clientes.

En la actualidad la potencialidad del hardware de escritorio ha tenido un gran avance y observamos como en los hogares ya podemos contar con equipos con grandes recursos hardware y conexiones de red aceptables; ahora bien, logramos obtener un amplio espectro de recursos que desafortunadamente no son considerados más allá del funcionamiento local en la arquitectura Cliente/Servidor, dejando al Cliente un papel secundario como consumidor de servicios, es decir, sin ningún otro aporte que el de ejecutar las transacciones de los servicios solicitados.

Considerando esto se puede observar que si se tiene en cuenta el actual potencial, que poseen los Clientes, resulta una considerable cantidad de recursos en almacenamiento, procesamiento y ancho de banda que por cuenta de la arquitectura, se están desaprovechando y que se podrían explotar de manera eficiente en otro modelo que tome dichos recursos para ofrecer más y mejores servicios.

Teniendo en cuenta lo anterior se presenta en el escenario una arquitectura descentralizada como una alternativa para la prestación de servicios y gestión de recursos; ON's P2P posee las capacidades arquitectónicas necesarias no solo para suplir las debilidades de la arquitectura Cliente/Servidor, si no para ofrecer a los desarrolladores un entorno novedoso para el diseño de nuevos servicios que cumplan las características de interoperabilidad, independencia de la plataforma, escalabilidad, tolerancia a fallos, y extensibilidad buscando la calidad de servicio y la estandarización. Ahora bien, en este orden de ideas surgen varios interrogantes: ¿Que características deberán poseer los servicios que se puedan crear y/o adaptar a la filosofía P2P?, ¿Qué conceptos y artefactos de la Ingeniería de Software pueden soportar el modelado y diseño de aplicaciones software para la provisión de servicios P2P?

Hasta el momento el desarrollo de aplicaciones para ON's P2P se han realizado de manera específica, es decir, buscando proveer solución a algún tipo de necesidad

particular como lo es el intercambio de ficheros, distribución de contenidos, mensajería instantánea, tal y como lo hizo en su momento el servicio de Napster[2], el cual popularizó de manera significativa el intercambio de archivos de audio; a partir de Napster empezó el auge de las aplicaciones ON's P2P, pero más allá de un modelo popular para el intercambio de información y un novedoso servicio de comunicaciones, se necesita justificar el porque elegir ON'S P2P como arquitectura base para la construcción de nuevos servicios y justificar los esfuerzos en el área de la ingeniería de software para dichos propósitos, pues bien, se consideró que no bastaba con solo establecer las diferencias sobresalientes entre las diferentes arquitecturas y resaltar las bondades de las redes superpuestas, se necesitaba algo más específico, que revele de forma solida la proyección y las ventajas de estas redes; y gran parte de esa motivación se encontró en el trabajo del Dr. Jorgue Cortell-Albert[3] de la Universidad de Oxford en su estudio sobre el futuro de las redes P2P en entornos corporativos, en el cual se demuestra las bondades del uso de estas redes y la proyección no solo a nivel académico sino a nivel empresarial.

En este orden de ideas se puede destacar que el área de la prestación de servicios orientados a la infraestructura ON's P2P a pesar de no ser un tema nuevo aún se encuentra en proceso de maduración, sobre todo en el área de la ingeniería de software aplicada a este tipo de redes y sus particularidades; siendo este el campo de interés de este proyecto, si bien el espectro de posibles servicios es muy amplio y su cobertura estaría fuera del alcance de este trabajo, se propone hacer un estudio profundo de un rango de servicios sobre los cuales se definirá los artefactos de análisis y diseño propios de la Ingeniería de software que contribuyan a la mejora de la calidad de los sistemas software sobre ON's P2P.



## **OBJETIVOS**

### **OBJETIVO GENERAL**

Definir un conjunto de perfiles que permitan dar soporte al modelado, descripción y construcción de servicios telemáticos para una arquitectura de redes superpuestas Peer to Peer.

### **OBJETIVOS ESPECÍFICOS**

- Definición e identificación de servicios ON'S P2P, plataformas de soporte, un proceso de análisis y modelamiento de software que refleja un espectro determinado de nuevos servicios o adaptación de los existentes.
- Definición y especificación de perfiles que permitan ofrecer soporte para el modelado y descripción de los servicios telemáticos identificados en el objetivo anterior.
- Construcción de un prototipo software que sirva como referencia práctica y de validación para la implementación de un subconjunto de los servicios definidos en el primer objetivo específico tomando como base de análisis y diseño los perfiles propuestos y desarrollados durante el proyecto.

### **CONTRIBUCIONES**

Dentro de las contribuciones más relevantes de este proyecto podemos resaltar:

- El presente trabajo realiza un gran aporte al proceso de desarrollo de aplicaciones punto a punto o P2P, más específicamente a las etapas de análisis, diseño y construcción, al definir perfiles de servicios específicos para esta arquitectura y necesarios para la construcción de cualquier tipo de aplicación.
- Perfiles obtenidos útiles para la construcción de Aplicaciones P2P
- Proceso de ingeniería para el desarrollo de aplicaciones P2P.
- Al hacer un aporte en el proceso de diseño de Aplicaciones P2P estamos iniciando un macro proceso que tendrá por objetivo principal estandarizar y formalizar una arquitectura de servicios para aplicaciones P2P que definitivamente introduzcan al mercado la construcción y desarrollo de tal tipo de Aplicaciones.
- Como caso práctico se implementó como prototipo una agenda colaborativa que representa un caso real especificando todas sus etapas de diseño e implementación y artefactos necesarios para su construcción, confirmando de esta manera la importancia práctica del proyecto y obteniendo una realimentación sobre la base teórica.

## CAPITULO 2 MARCO TEORICO

El concepto de servicio, sin importar el contexto en el que se utilice representa un escenario donde intervienen los siguientes actores:

- La entidad que solicita el servicio
- La entidad que provee el servicio
- El canal o medio por el cual se presta el servicio
- El servicio proporcionado en si mismo.

Los demás aspectos que rodean los elementos anteriores son complementos que buscan cubrir necesidades externas al consumo del servicio.

De lo anterior podemos abstraer que el propósito esencial de un servicio es proveer una solución a una necesidad específica solicitada por el consumidor, lo cual visto así es un concepto muy simple y para nada novedoso, pudiendo clasificarlos en muchos tipos y de diferente naturaleza; como por ejemplo cuando entramos a un cajero electrónico a retirar dinero en efectivo, o cuando buscamos por un sitio de internet en google<sup>1</sup>, cuando tomamos un taxi, cuando hacemos una compra en el supermercado o en internet, etc. La naturaleza del servicio no altera el rol de los que participan en su proceso, de lo cual se generaliza un comportamiento que bajo ciertas condiciones llega a ser estático en lo que al ciclo de vida del servicio se refiere.

Los consumidores juegan un rol muy común en la sociedad actual, la gran mayoría de nosotros requerimos de servicios mínimos para nuestra subsistencia y dependiendo de nuestro ámbito social las necesidades generan nuevos y mas variados requerimientos de todo tipo, por ejemplo, las tarjetas de crédito, son un medio que proveen las entidades bancarias para manejar de forma mas segura, confortable y útil nuestro dinero; sin embargo no todos consumimos los mismos servicios, es razonable que un gerente de ventas de una empresa multinacional requiera funcionalidades financieras mucho más especializadas que un tendero de barrio, es decir, son las necesidades y el ámbito social los que definen que necesitan cierto tipo de consumidores.

Ahora bien, los consumidores no son sólo personas o conjuntos de personas, los consumidores son todas aquellas entidades que dentro de su autonomía no tienen la capacidad de cumplir o satisfacer una necesidad y requieren de las funcionalidades de otras entidades externas para cumplir su propósito, como el cajero del supermercado que para completar un pago con tarjeta bancaria requiere hacer una consulta remota al banco correspondiente para solicitar autorización de pago y así terminar la transacción.

---

1 Google: <http://www.google.com> Servicio de búsqueda en internet

Ya hemos revisado la naturaleza de la provisión de servicios independientemente del contexto, sin embargo este último determina muchas características y comportamientos de los mismos; como se mencionó anteriormente el consumo está muy ligado a las necesidades específicas que rodean al consumidor, y siguiendo la perspectiva de este último encontramos una serie de fenómenos que están impactando en gran medida a la sociedad actual, hablamos de las redes sociales, tan populares hoy en día, las cuales responden simplemente a necesidades y posibilidades que brindan los nuevos o mejores canales de comunicación, principalmente gracias al imparable auge de la internet, de allí que pasemos a unos niveles de interacción social con múltiples posibilidades, todo ello gracias a que la interacción electrónica dejó de ser entre humanos y enormes “main frames” que proporcionaban información y paso a ser entre personas, amigos compañeros de trabajo, estudiantes y profesores, o simplemente entre gente que busca relacionarse; estas posibilidades dispararon la popularidad de estas redes a tal nivel que se convirtieron en focos de las grandes empresas de la informática; pero ¿qué tiene que ver esto con provisión de servicios?, mucho en realidad, ya que la popularidad de las redes sociales en internet debe su masificación a la evolución de las tecnologías de internet y las redes de telecomunicaciones que proveen as funcionalidades necesarias para estos fines.

Continuando con el análisis del contexto de los servicios, ahora enmarcado en las redes sociales hay un aspecto que a pesar de no ser nuevo, es tal vez la principal razón de ser de las redes sociales, hablamos de la posibilidad de compartir información, contenidos o cualquier recurso entre los miembros de la red, entre otras muchas posibilidades y respecto a estas necesidades de las redes sociales surgen múltiples soluciones que buscan cubrirlas y ofrecer al público la mejor opción ya sea con fines comerciales, educativos, o de cualquier tipo; entre las soluciones, una de las más difundidas y a la vez criticada por aspectos que se relacionarán más adelante en este documento, se encuentran las redes entre iguales o más conocidas como redes P2P, las cuales surgen como un modelo de comunicaciones alternativo a los modelos tradicionales para proveer servicios y solucionar necesidades específicas que dada su naturaleza distribuida optimizan mucho más los procesos para abordar ciertas necesidades de las redes sociales.

En éste orden de ideas es necesario definir el concepto de servicio y el contexto en el que se enfoca dentro de las redes P2P y la ingeniería de software, lo cual conforma el escenario de este trabajo de investigación.

## **2.1 SERVICIOS TELEMÁTICOS Y ARQUITECTURAS ORIENTADAS A SERVICIOS**

Un servicio [4] es un conjunto empaquetado de funcionalidades de una red de comunicaciones que maneja el intercambio de información el cual satisface cierta demanda del usuario. Éste servicio es proporcionado por la red al usuario a través de un punto de acceso apropiado. Para utilizar aquellos disponibles en una red, se necesitan dos programas, que se ejecuten en dos computadoras y que se comuniquen entre sí. Estas aplicaciones son las entidades que se comunican entre sí, no las computadoras ni los usuarios.

El auge de los servicios está representado en las arquitecturas orientadas a servicios (SOA), las cuales han evolucionado a través de los últimos años. SOA es un estilo de arquitectura de software que promueve el desacople entre componentes de forma que se puedan reutilizar. Las arquitecturas orientadas a servicios son populares porque permiten reutilizar aplicaciones y prometen interoperabilidad entre aplicaciones y tecnologías heterogéneas [1].

Los servicios son componentes de software que han publicado contratos/interfaces; estos contratos son independientes de la plataforma, lenguaje y sistema operativo. XML y SOAP (Simple Object Access Protocol) están permitidos por que son estándares independientes de la plataforma, los consumidores pueden descubrir servicios dinámicamente y estos son interoperables.

Las soluciones basadas en SOA se han venido popularizando gracias a los servicios Web, sin embargo SOA no está atado a los servicios Web, es decir, se puede construir Aplicaciones SOA basados en otras tecnologías como RMI<sup>2</sup> , Corba<sup>3</sup>, en general SOA, usa estándares abiertos, que garantizan la interoperabilidad. En el área de los sistemas distribuidos también existen varias tecnologías que se acoplan a la filosofía de SOA, las cuales se abordarán mas adelante en este documento.

---

2 RMI : Tecnología de invocación remota de metodos (Remote Method Invocation)

3 CORBA: (Common Object Request Broker Arqchitecture) <http://omg.org/corba>

## **2.2 MODELO P2P:**

En los últimos años han surgido una serie de aplicaciones en Internet, operando bajo el modelo de comunicaciones denominado compañero a compañero (P2P). Cada una de estas aplicaciones implementa un servicio que opera de manera distribuida, tomando y brindando recursos, en computadoras de usuario final. Aplicaciones P2P han ganado popularidad, programas de mensajería, tales como ICQ [5] y MSN Messenger [6], y de intercambio de archivos, como Napster [2] y Gnutella [7], son las más notables en el ámbito de las aplicaciones P2P. Este modelo no es nuevo; de forma simple puede verse como la comunicación entre pares o iguales utilizando un sistema de intercambio.

En telefonía se utiliza este modelo, donde se requiere una ó más centrales de conmutación para efectos de mantener una comunicación entre usuarios finales. Actualmente, los usuarios de computadoras poseen equipamiento con buenas capacidades de procesamiento, lo cual ofrece muchas más posibilidades y prestaciones, siendo estas características las que aprovechan y utilizan plenamente las redes P2P.

Los sistemas P2P permiten que computadoras de usuario final se conecten directamente para formar comunidades, cuya finalidad sea el compartir recursos y servicios computacionales [8]. En este modelo, se toma ventaja de recursos existentes en los extremos de la red, tales como tiempo de CPU y espacio de almacenamiento. Las primeras aplicaciones emergentes se orientaban a compartir archivos y a la mensajería.

IBM plantea que el modelo P2P es el resultado natural de las tendencias de descentralización en la ingeniería de software [9], e Internet determinó que aplicaciones clásicas, de tipo monolítico, cambien a esquemas descentralizados.

Para la firma Intel [10] el modelo P2P provee una alternativa a la arquitectura clásica cliente/servidor. Utilizando la infraestructura actual, compuesta de redes, servidores y clientes, P2P ofrece un modelo ortogonal al modelo cliente/servidor; dado que los dos modelos coexisten, se intersecan y se complementan. En un ambiente P2P, cada computador es un compañero que provee servicios de cliente y servidor a la vez, dependiendo de la aplicación, se determinará qué recursos tomará y qué recursos brindaría la red.

## **2.3 ORÍGENES DE LOS SISTEMAS P2P SOBRE INTERNET**

En el año 1969, la Agencia de Proyectos de Investigación Avanzados (ARPA) conectó cuatro computadoras de universidades estadounidenses (UCLA, Stanford Research Institute, UCSD, y la Universidad de Utah). Donde el principal objetivo de este proyecto era desarrollar una red que pudiera resistir a ataques de un país enemigo y como objetivo secundario se propuso el desarrollo de un método que permitiera compartir poder computacional situado en computadoras remotas.

Teniendo en cuenta las opiniones de Arthur Bushkin [11], quien participó en el proyecto bajo la premisa de que ninguna de las personas que formaban parte del grupo inicial sabía que se estaba gestando y eran incapaces de imaginarse que la red ARPAnet iba a ser precursora de Internet. Cuando ARPA puso en línea a las computadoras mencionadas estableció una red P2P, donde cada una ofrecía y tomaba a la vez recursos de las otras.

Las primeras aplicaciones populares, tales como Telnet y FTP, operaban bajo el esquema cliente/servidor. En un principio, aunque las aplicaciones fueran cliente/servidor, el uso de las computadoras donde residían tales clientes y servidores eran simétrico; debido a que casi no existían computadoras dedicadas a tareas de servidor exclusivamente y por otro lado las computadoras personales ó de usuario final no existían. Sistemas propios de Internet, como USENET y DNS, utilizaron la tecnología compañero a compañero [12] en su infraestructura.

### **2.3.1 USENET**

Es un sistema distribuido de conferencias, donde a cada una se las denomina grupos de noticias y a los mensajes artículos [13]. En su infraestructura, USENET implementa un modelo descentralizado de control, donde fundamentalmente permite la copia de archivos entre computadoras sin la existencia de un control central.

Su construcción se basó en un servicio propio del sistema operativo UNIX denominado UUCP (Unix to Unix Copy Protocol). Bajo UUCP una computadora, vía modem, discaba a otra, se conectaba e intercambiaba archivos y finalmente se desconectaba. Este mecanismo permitía en la práctica el intercambio de correo electrónico, revisiones de código, etc.

Estudiantes de la Universidad de Carolina del Norte y de la Universidad Duke implementaron, utilizando UUCP como transporte, un protocolo para el intercambio de artículos de difusión y de respuesta, que abarcaban una serie amplia de tópicos. Más instituciones se sumaron al sistema de conferencias y la red USENET creció hasta llegar a operar con cientos de miles de sitios. Este crecimiento involucró reformas en la infraestructura de la red y se decidió utilizar el juego de protocolo TCP/IP como soporte del protocolo de transporte de noticias USENET (NNTP, Network News Transport Protocol) [14], el cual posibilita que dos computadoras de la red USENET puedan descubrir nuevos grupos de noticias de forma eficiente e intercambiar nuevos mensajes.

Este modelo es un ejemplo de un sistema de gran escala que posee control descentralizado, dado que no existen autoridades que controlan la publicación de noticias. A los efectos de agregar nuevos grupos se realiza un proceso de votación.

En toda organización de mediano porte (por ejemplo una universidad o un proveedor de acceso a Internet), generalmente, existe un servidor de noticias local al cual sus usuarios se conectan para recibir y enviar sus artículos. Sobre el mencionado servidor se implementan dos variantes del protocolo NNTP: una destinada a la comunicación entre pares (otros servidores) y otra destinada a la comunicación con sus clientes. Cada

servidor puede descargar de servidores adyacentes, lo que éste quiera o lo que este disponible.

Existe en cada servidor una lista de seguimiento de grupos de noticias que se utiliza a los efectos de que un servidor pueda actualizar información de los grupos que localmente atiende y no recibir noticias duplicadas (una especie de control de ciclicidad sobre la red). Cada mensaje perteneciente a cualquier grupo de noticias lleva un identificador que lo hace único, por medio del cual los servidores y los clientes pueden administrar sus descargas a los efectos de obtener siempre las noticias nuevas y de no recibir duplicadas.

El servidor de noticias no maneja información alguna acerca del estado de la lectura por parte de sus clientes. Éstos deben implementar mecanismos para el control de la actualización de las noticias.

El modelo propuesto por Usenet, acrónimo de **Users Network** (Red de usuarios) en vigencia desde 1979, ha demostrado ser útil y exitoso, Permitiendo la comunicación eficaz entre grupos de personas que forman comunidades temáticas a lo largo del mundo.

### **2.3.2 DNS**

El sistema de registro de nombres de dominio (DNS) combina una red P2P, conformado por el conjunto de servidores de nombres, con un modelo de información jerárquico [12]. Esta infraestructura permitió que DNS escale de miles de equipos a millones actualmente desde su creación en 1983. La razón por la cual surgió DNS fue a partir de buscar un mecanismo más ágil que el intercambio del archivo “/etc/hosts”, que contenía la asociación de nombres de equipos con direcciones de red. DNS fue desarrollado como una forma de distribuir datos capaces de ser compartidos a lo largo de la red Internet.

El espacio de nombres definido en DNS es jerárquico, y los equipos servidores de nombres operan bajo una estructura simétrica, dado que actúan como clientes y servidores a la vez.

Son servidores desde el punto de vista que brindan información sobre su dominio ó eventualmente sobre otros dominios (en el caso de que esté almacenada en su memoria cache); y clientes porque envían consultas a otros pares, en nombre de sus clientes. La escalabilidad del sistema DNS se basa en una serie de elementos de diseño, como son:

a- Que los servidores DNS pueden operar como servidores y clientes, propagando consultas de terceros.

b- Que los servidores pueden almacenar y consultar su memoria cache a los efectos de satisfacer consultas sobre otros dominios de forma local.

c- Que un servidor puede consultar a otro, pero existe un camino de consulta en base a una jerarquía preestablecida o cadena de autoridad (aquí las cargas son balanceadas por medio de la jerarquía). Analizando otros protocolos, tales como SMTP (Single Mail Transport Protocol)

También se puede observar que el principio de simetría, clásico en entornos P2P, está presente entre el conjunto de servidores que conforman la red de correo electrónico de Internet.

Tradicionalmente el modelo P2P es un tipo de red en la cual cada estación de trabajo tiene capacidades y responsabilidades equivalentes. La diferencia con la arquitectura cliente/servidor está dada por que en este caso algunas computadoras están exclusivamente dedicadas a brindar servicio a otras.

La definición anterior está basada en la concepción tradicional de las redes P2P, donde computadoras de usuario final, generalmente en el ámbito de una red local, formaban un grupo de trabajo y cada usuario podía permitir que otros compañeros accedan a sus archivos o impresoras locales.

El profesor Clay Shirky<sup>2</sup> [15] caracteriza a la tecnología P2P como “ una clase de aplicaciones que toman ventaja de los recursos –almacenamiento, ciclos de CPU, contenidos, presencia humana- disponibles en Internet”, y define una prueba, que consta de dos preguntas, a los efectos de validar si una aplicación determinada puede ser definida como compañero a compañero

1. ¿Es normal que los usuarios tengan conectividad variable y se les asigne direcciones de red temporales?
2. ¿Se da a los nodos situados en los extremos de la red un importante grado de autonomía?

Donde si ambas preguntas son contestadas afirmativamente la aplicación es compañero a compañero.

La organización FIPA -Foundation for Intelligent Physical Agents- [8] ve los sistemas P2P como un medio para “compartir recursos y servicios computacionales por medio de intercambio directo”. Las aplicaciones populares incluyen intercambio de archivos y procesamiento distribuido. Debido a que la computación P2P involucra la participación de los nodos de una red sin administración central, se deben considerar las siguientes cuestiones:

Los nodos en una red P2P pueden ingresar y salir constantemente de forma arbitraria.

La ubicación de los servicios y recursos es dinámica.

- Cada nodo regula su grado de participación en la red
- La red es un ambiente dinámico y heterogéneo.

García-Molina [16] define un sistema P2P como “nodos de computación distribuida con iguales roles y capacidades de intercambio de información y servicios directamente entre ellos”. La gran ventaja que aporta P2P es que los recursos de muchos usuarios y



computadoras pueden unirse para producir un gran cúmulo de información y un significativo poder de cómputo.

Para la firma Intel [10] la computación P2P es una forma de compartir recursos computacionales y servicios mediante intercambio directo. Un nodo puede actuar como cliente y servidor en el contexto de una aplicación dada. La interacción entre nodos (requerimientos y respuestas), su ámbito y como son procesados los mensajes son cuestiones específicas de la aplicación de usuario final. Por otro lado, la firma indicó que uno de los más grandes beneficios de P2P es el concepto de comunidad, dado que hace posible que los usuarios se organicen ellos mismos en grupos ad hoc, y puedan eficientemente y de forma segura llenar requerimientos, compartir recursos, colaborar y comunicarse.

Bajo la óptica de los autores de este documento, un sistema compañero a compañero es un modelo de uso coordinado de recursos de ciclos de CPU, espacio de almacenamiento, archivos, etc. Sobre la base de la comunicación directa entre nodos que los requieren y los poseen, bajo una topología de red donde todos los participantes tienen igual acceso a todos sus pares y eventualmente pueden existir nodos con alguna capacidad especial.

Los sistemas P2P utilizan la descentralización, como una forma de incrementar el rendimiento, escalabilidad y la disponibilidad de los recursos de los usuarios que conforman tal sistema.

## **2.4 CARACTERÍSTICAS DE LOS SISTEMAS P2P**

La primera característica que define a un nodo en un sistema P2P, es que cumple tanto el rol de cliente como de servidor (en la terminología se lo denomina *servent*, palabra que deriva de la conjunción de los términos *server-client*). Este modelo basado en interacciones entre pares, puede reemplazar al paradigma cliente/servidor, donde cada nodo es capaz de generar y contestar consultas de otros nodos. Desde el punto de vista de la comunicación, las interacciones son simétricas.

- Los nodos participantes son autónomos. Cada uno regula su grado de participación en la red, a través de definir que recursos ofrece y en que cantidad.
- Ninguno de los nodos tiene una vista global del sistema.
- El comportamiento global emerge de las interacciones individuales directas.
- Todos los datos y servicios deben ser accesibles por cualquier nodo.
- Implementan un sistema de nombres alternativo al sistema DNS, que satisfaga sus propias necesidades.
- Los nodos en una red P2P pueden ingresar y salir constantemente de forma Arbitraria. Como son nodos de usuario final no se encuentran permanentemente Conectados a la red, es decir, su conectividad es variable.

- La ubicación de los recursos y disponibilidad de servicios es dinámica, ya que depende del estado del sistema en un momento del tiempo.
- La red es un ambiente dinámico y heterogéneo, dado que la conforman nodos con conectividad variable y de diversas plataformas de hardware y software. Toda computadora en Internet tiene la posibilidad de que sobre éste corra un nodo P2P.
- La empresa Microsoft en el marco de su producto .NET brindó una serie de características propias de cada nodo en un sistema compañero a compañero.
- Capacidad de descubrir compañeros. El nodo debe poder encontrar a sus pares a los efectos de saber quiénes son y donde están. Esta información se puede obtener consultando a un servidor central que registra nodos ó a través de algoritmos de descubrimiento.
- Realizar consultas a compañeros.
- Compartir recursos con otros compañeros.

## **2.5 SISTEMAS CLIENTE/SERVIDOR VERSUS SISTEMAS P2P**

Las ventajas y desventajas de las redes basadas en un esquema cliente-servidor frente a una red P2P son los derivados de la centralización de recursos:

- Un servidor dedicado ofrece una mayor capacidad de trabajo que una máquina que opera además como estación de trabajo.
- El poseer información centralizada ofrece más seguridad contra accesos no autorizados.
- Es más práctico actualizar programas y realizar copias de respaldo en un modelo de red centralizada.
- No se corre el riesgo de que por mal funcionamiento de una estación de trabajo corra peligro todo el sistema.
- Los sistemas centralizados son responsables de la mayoría de la carga de procesos, lo cual contrasta con los sistemas P2P los cuales al ser sistemas distribuidos reparten dicha carga entre los nodos de la red.

La desventaja fundamental es que si ocurriese algún problema en el servidor todas las estaciones de trabajo podrían verse afectadas.

## 2.6 ÁREAS DE APLICACIÓN

### 2.6.1 TRABAJO EN GRUPO

La colaboración incrementa la productividad disminuyendo el tiempo de múltiples revisiones por parte de los participantes de un proyecto. Además, permite trabajar equipos distribuidos geográficamente. Disminuye el tráfico en la red por la disminución de e-mail y disminuye la necesidad de almacenamiento en servidores centrales, utilizando almacenamientos locales con posibilidad de replicación automática.

Groove [17] es una aplicación reciente, ejemplo de ambiente colaborativo, donde los usuarios pueden crear "espacios compartidos". Los objetos y el contenido dentro de dichos espacios aparecen en todas las máquinas del equipo, y cada usuario puede contribuir con los archivos o los programas de su elección.

Replicación de contenidos: Una red de aplicación soportada sobre una infraestructura compañero a compañero puede ayudar a distribuir contenidos a través de distintas áreas geográficas [18]. Esencialmente pueden moverse los datos más cerca del punto donde realmente son consumidos, implementando mecanismos de replicación. Las redes de distribución de contenidos (CDN, Contents Distribution Networks), que tienen por objetivo lograr que los usuarios finales accedan más rápidamente a contenidos web, ya han comenzado a incorporar el modelo P2P.

### 2.6.2 COMPUTACIÓN DISTRIBUIDA

Necesidades de procesamiento en gran escala pueden ser provistas utilizando una red de computadoras donde sus componentes puedan aportar capacidad ociosa de CPU y espacio en disco. Este servicio permite a Universidades, institutos de investigación y empresas obtener un gran poder de cómputo, a partir de múltiples computadoras operando en paralelo (procesamiento distribuido sobre una red débilmente acoplada) para aplicar a tareas específicas. Los resultados se pueden obtener en tiempos y costos menores.

Actualmente proyectos como [19], FightAIDS@Home<sup>4</sup>, Folding@Home<sup>5</sup> Folding y Distributed.net [20] utilizan computación distribuida para resolver problemas de la más variada gama.

No todos los proyectos utilizan una red compañero a compañero pura. Intel, en su servicio de computación distribuida colaborativa, y SETI@home implementan servidores centralizados, sin comunicación entre pares. Sin embargo, estos clientes son participantes activos del sistema y el proceso de cómputo se basa en poseer redundancia masiva. En algunos casos, utilizan una red P2P como soporte de comunicaciones. Intercambio de archivos: Permiten que usuarios de computadoras personales intercambien directamente

---

<sup>4</sup> <http://fightaidsathome.scripps.edu/>

<sup>5</sup> <http://folding.stanford.edu/>

entre ellos archivos de cualquier tipo. Estas aplicaciones han obtenido una gran popularidad gracias a Napster, que en su momento acumuló 18 millones de usuarios en 18 meses de existencia. Se considera que no es un sistema P2P puro, debido a que utiliza un servidor central donde los usuarios consultan para saber dónde están los recursos. Actualmente estos sistemas focalizan sus contenidos en la distribución de archivos musicales, videos y fotografías.

**Mensajería instantánea:** La mensajería instantánea permite la comunicación en tiempo real entre personas conectadas a la red al mismo tiempo. El intercambio se realiza al escribir mensajes o al enviar archivos. Programas de mensajería como AOL Instant Messenger [21] e ICQ son algunas de las opciones comerciales que existen en esta área.

**Búsquedas:** Debido a la expansión de la web, los motores de búsqueda tradicionales experimentan problemas de cobertura y de actualización de contenidos. La implementación de servicios de búsquedas distribuidas basadas en redes P2P puede mejorar esta situación, permitiendo a sus usuarios la obtención de resultados de mejor calidad [18]. El proyecto InfraSearch [22] fue un punto de partida en este tipo de aplicaciones. Si bien puede ser complicado realizar búsquedas de gran cobertura del espacio web, los sistemas P2P ofrecen otra visión de cómo abordar el problema y plantean soluciones para búsquedas extendidas sobre temas específicos.

**Comercio:** Algunas compañías están experimentando con modelos de conexión directa entre usuarios a los fines de implementar sistemas de comercio electrónico. La idea es reemplazar un mercado electrónico centralizado por intercambios directos entre compradores y vendedores. La empresa Netrana está desarrollando herramientas y estrategias de comercio electrónico para situaciones que escapen al ámbito de los mercados electrónicos convencionales. Su objetivo es desarrollar una plataforma de comunicaciones que permita a los participantes llevar una negociación directa, sin la intervención de un intermediario. Un área propicia para las redes P2P será el comercio electrónico, debido al potencial de la tecnología de enlazar directamente vendedores y compradores. La empresa Biz To Peer Technologies está diseñando estrategias para proporcionar a sus usuarios herramientas de informática distribuida con el fin de establecer transacciones entre particulares de manera directa.

Las aplicaciones P2P que se ejecutan sobre Internet residen en computadoras personales, equipos que se los conoce como nodos de los "bordes de Internet" (Internet edges); por otro lado tales nodos no están registrados en el sistema de nombres de dominio debido a su conectividad variable y por consiguiente su asignación dinámica de dirección de red.

Aplicaciones P2P, tales como Napster, Popular Power, Freenet e ICQ son populares debido a que utilizan recursos, hasta el momento no utilizados, disponibles en los bordes de Internet. Sus protocolos contemplan las características de comportamiento enunciadas en el párrafo anterior.

Hasta el año 1994, la red Internet en su conjunto poseía un modelo de conectividad. Se asumía que los equipos siempre estaban operativos, sus interfaces de red poseían direcciones fijas. El sistema DNS fue pensado sobre la base de estas características. A partir del establecimiento del sistema de información denominado World Wide Web ó simplemente web, un modelo alternativo al planteado comenzó a entrar en vigencia. Computadoras personales, con capacidades gráficas importantes, comenzaron a navegar a demanda por el sistema web. Fue necesario implementar centros de acceso a Internet (ISP), donde los usuarios utilizando el sistema telefónico obtenían una dirección de red temporal. He aquí que este modelo se caracterizó por que los usuarios ingresaban y salían frecuentemente, en forma no predictiva, a la red; sin estar registrada su condición en los servidores DNS.

En el año 1996 se comenzó a utilizar el sistema de mensajería ICQ, fue la primera vez que las computadoras personales situadas en los bordes de Internet comenzaron a poder ser visibles por otros pares, dado que ICQ estableció un sistema de localización de usuarios donde se mapea la dirección temporal de red con el nombre de usuario. Esta técnica, luego utilizada por sistemas tales como Napster y Groove, permitió superar las carencias del sistema DNS sobre tales nodos. Una característica de los sistemas P2P es que permiten crear direcciones casi ilimitadamente para máquinas y otros recursos (volúmenes, personas). Aplicaciones como Freenet y MojoNation crean direcciones o nombres para máquinas y elementos que no lo son.

Dale Dougherty [23] utiliza el término PIE para describir los elementos centrales de las aplicaciones P2P. Donde PIE significa Presencia, Identidad y recursos de borde (Edgeresources).

## **2.7 PLATAFORMAS P2P**

Varias compañías pugnan por imponer sus proyectos de infraestructura para soportar aplicaciones P2P. Microsoft propone su medioambiente de trabajo denominado .NET y Sun Microsystems a JXTA [24]. Otra compañía denominada Groove Networks [17] presentó una plataforma descentralizada de groupware y ha incorporado servicios P2P, tales como mensajería instantánea e intercambio de archivos.

A los efectos de consolidar el cómputo distribuido se necesitará desarrollar APIs y compiladores robustos desarrollados específicamente para tal ambiente. Compañías como Sun a través de su producto JXTA y Microsoft con Hailstorm Technologies están intentando brindar ambientes de desarrollo unificado para construir tales aplicaciones.

### **2.7.1 NET**

A fines del año 2001, Microsoft elaboró una estrategia de tipo participativa para popularizar el desarrollo y utilización de servicios compañero a compañero a través de servidores web basados en XML donde todo equipo participante de la red .NET colabora según sus recursos disponibles en torno al potencial de la infraestructura que la soporta.

.NET se concibió como una plataforma de servicios web basados en XML, que les permiten a las aplicaciones de usuario final comunicarse y compartir datos sobre Internet.

### **2.7.2 JXTA**

En el año 2001 Sun Microsystems lanzó el proyecto JXTA [24], e indicó que es una plataforma, que aumenta la habilidad de los usuarios para compartir la información de sus computadoras personales y otros dispositivos. El objetivo de JXTA es permitir a los usuarios, por medio de nuevas aplicaciones, disponer de información clave a través de numerosos puntos de acceso, como dispositivos portables, notebooks y teléfonos móviles.

Según Bill Joy, director científico de Sun, "Esta es la tecnología que dará acceso a una Red más amplia y profunda", "La meta es suministrar un método común para cualquier nodo de la Red para que acceda a cualquier dato, contenido u otro nodo". Según Sun, las ventajas que presenta JXTA son las siguientes:

- ° Independencia de plataforma. JXTA es soportada por distintos sistemas operativos, tales como Linux, Solaris, Microsoft Windows y por distintos lenguajes de programación (C, Java).
- ° Ubicuidad. Es posible implementar JXTA sobre una amplia gama de dispositivos como computadoras de mano, notebooks, computadoras personales, dispositivos de red, sistemas de almacenamiento, aparatos varios, etc.
- ° Interoperabilidad. Dado que JXTA ha sido diseñada para localizar y comunicar fácilmente cada nodo participante en la red.

Se detalla JXTA con detalle más adelante en este documento.

### **2.7.3 SERVICIOS WEB**

Definición del World Wide Web Consortium (W3C)<sup>6</sup>: Una aplicación software identificada por un URI, cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Un servicio Web soporta interacciones directas con otros agentes software utilizando mensajes XML intercambiados mediante protocolos basados en Internet.

Los servicios Web garantizan la interoperabilidad y su funcionalidad gracias a la especificación WSDL [25] el cual es un lenguaje basado en XML para describir servicios Web, localizarlos y como accederlos. Esta tecnología se convierte también en un modelo ejemplar para uno de los propósitos de este trabajo en aras de llevar a los servicios ON's P2P a un alto nivel de interoperabilidad y funcionalidad.

### **2.7.4 COMPUTACIÓN GRID**

Grid [26] es un tipo de sistema paralelo y distribuido que permite dinámicamente (“at runtime”) compartir, seleccionar y agregar recursos anónimos (y de diferentes dueños) que se encuentran geográficamente distribuidos, dependiendo de su disponibilidad, capacidad, rendimiento, costo y los requerimientos de calidad del servicio, con el fin de resolver problemas computaciones de gran escala en campos como las ciencias, ingeniería y comercio.

### **2.7.5 JINI [27]**

(Java Intelligent Network Infrastructure), Propuesto por Sun<sup>7</sup> en 1999 es una plataforma para el desarrollo en Computación Distribuida, adaptada a la realidad dinámica de las redes, busca unificar criterios en cuanto al desarrollo de redes P2P.

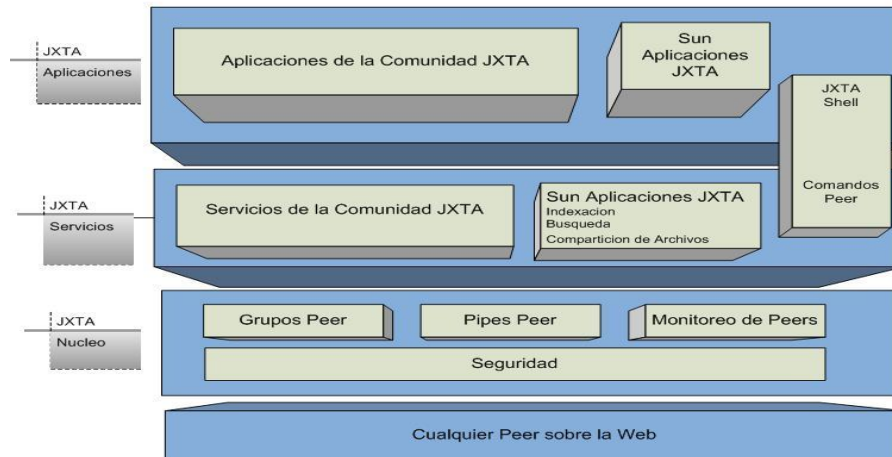
No profundizamos mucho en las anteriores tecnologías ya que parte de nuestro estudio será precisamente comparar las mismas para seleccionar la que mejor se adecúe a nuestros objetivos; sin embargo, proyectando un poco y estimando alrededor de estas tecnologías vamos a hablar un poco más de la plataforma JXTA, la cual ha sido creada por Sun Microsystems Inc. específicamente para ofrecer una plataforma de desarrollo de aplicaciones ON's P2P, y por lo cual desde ya se convierte en la tecnología candidata para el desarrollo de nuestro trabajo.

Basado en tecnología ya probada y estándares como HTTP, TCP/IP y XML. La tecnología JXTA no depende de un lenguaje de programación particular, plataforma de red o plataforma de sistema, y puede trabajar con cualquier combinación de éstas.

---

6 <http://www.w3c.es/>

7 SUN microsystems: <http://www.sun.com>



**Figura 2 Protocolo JXTA**

JXTA define 6 protocolos [28]:

**Peer Discovery Protocol:** utilizado para descubrir y publicar recursos.

**Peer Information Protocol:** para obtener información del estado de otros peers.

**Peer Binding Protocol:** utilizado para establecer canales o tuberías de comunicación.

**Peer Resolver Protocol:** utilizado para enviar consultas genéricas.

**End Point Routing Protocol:** para hacer enrutamiento y búsqueda de otros peers.

**Rendezvous Protocol:** utilizado para propagar mensajes dentro de un grupo de peers.

## 2.8 PERFILES

Los perfiles [29] constituyen el mecanismo que proporciona el propio UML para extender su sintaxis y su semántica para expresar los conceptos específicos de un determinado dominio de aplicación. Los perfiles se definieron originalmente en la versión 1 de UML aunque era difícil saber si se estaban aplicando correctamente, debido a que su definición era un tanto ambigua. La nueva versión UML 2.0 [30] mejora substancialmente su definición, especificándose de forma más clara las relaciones permitidas entre los elementos del modelo a especificar y el uso de las metaclasses de un metamodelo dentro de un perfil.

Un perfil se define en un paquete UML, estereotipado «profile», que extiende a un metamodelo o a otro Perfil. Tres son los mecanismos que se utilizan para definirlos: estereotipos (stereotypes), restricciones (constraints), y valores etiquetados (tagged values).

También hay perfiles definidos por otras organizaciones y empresas fabricantes de lenguajes de programación y herramientas que, aun no siendo estándares oficiales, están



disponibles de forma pública y son comúnmente utilizadas (convirtiéndose por tanto en estándares “de facto”). Un ejemplo es el “UML/EJB Mapping Specification”, que ha sido definido por JCP (Java Community Process). También se dispone de perfiles UML para determinados lenguajes de programación, como pueden ser Java o C#. Cada uno de estos ellos define una forma concreta de usar UML en un entorno particular. Así por ejemplo, el perfil UML para CORBA define una forma de usar UML para modelar interfaces y artefactos de CORBA, mientras que el de UML para Java define una forma concreta de modelar código Java usando UML.

Existen varios motivos por los cuales se requiera definir perfiles, entre los más importantes se encuentran: [29]

- Proporcionar una terminología y definiciones propias de un dominio de aplicación o de una plataforma de implementación concreta, para precisar sus propios conceptos.
- Describir una sintaxis conceptual propia o adaptar las existentes al ambiente o contexto requerido.
- Especificar una notación para símbolos ya existentes, más acorde con el dominio de la aplicación objetivo que represente claramente un concepto o entidad en dicho dominio.
- Agregar la semántica que no aparece determinada de forma precisa en el metamodelo para adaptar su significado a nuestras necesidades.

Como se puede observar, los perfiles UML son la herramienta que nos ofrece UML en su especificación 2, para poder definir, especificar y particularizar nuestro dominio de aplicación, que en este caso son los servicios ON's P2P, buscando con ello llevar esta tecnología a un nivel más alto y lograr su posicionamiento como una muy buena alternativa a la hora de desarrollar soluciones informáticas orientadas a los servicios.

## CAPITULO 3 DISEÑO DE PERFILES P2P COMPUTACIÓN ORIENTADA A SERVICIOS Y SOA

### 3.1 FUNDAMENTOS DE DISEÑO

Bajo una arquitectura P2P los servicios juegan un papel importante ya que estos representan funcionalidades específicas, éstos pueden estar compuestos a la vez por servicios más simples.

Para modelar servicios P2P y acoplarlos dentro de una Arquitectura Orientada a Servicios es necesario familiarizarse con un marco de trabajo de comunicación que está comprendido por una serie de términos con definiciones muy explícitas. Es por eso que debemos unificar tal lenguaje y se presentan las siguientes definiciones.

#### 3.1.1 CARACTERÍSTICAS DE DISEÑO

Es un atributo o cualidad que una solución de negocio automatizada puede poseer. Pueden ser numerosas y únicas características, que pueden ser cuantificables a través de varias medidas [31]. Como por ejemplo podemos mencionar que una solución lógica puede tener características como: ser modularizada, altamente acoplada, tener una base de Datos compartida, ser distribuida, etc. Estas características son planeadas en el diseño y aplicadas en el desarrollo. Y entonces podemos decir que las características de diseño son el resultado de la aplicación de un principio de diseño. Las características que estamos interesados en este momento son atributos específicos o cualidad del cuerpo de la solución lógica que documentamos en la especificación de diseño y planeamos realizar en desarrollo [31].

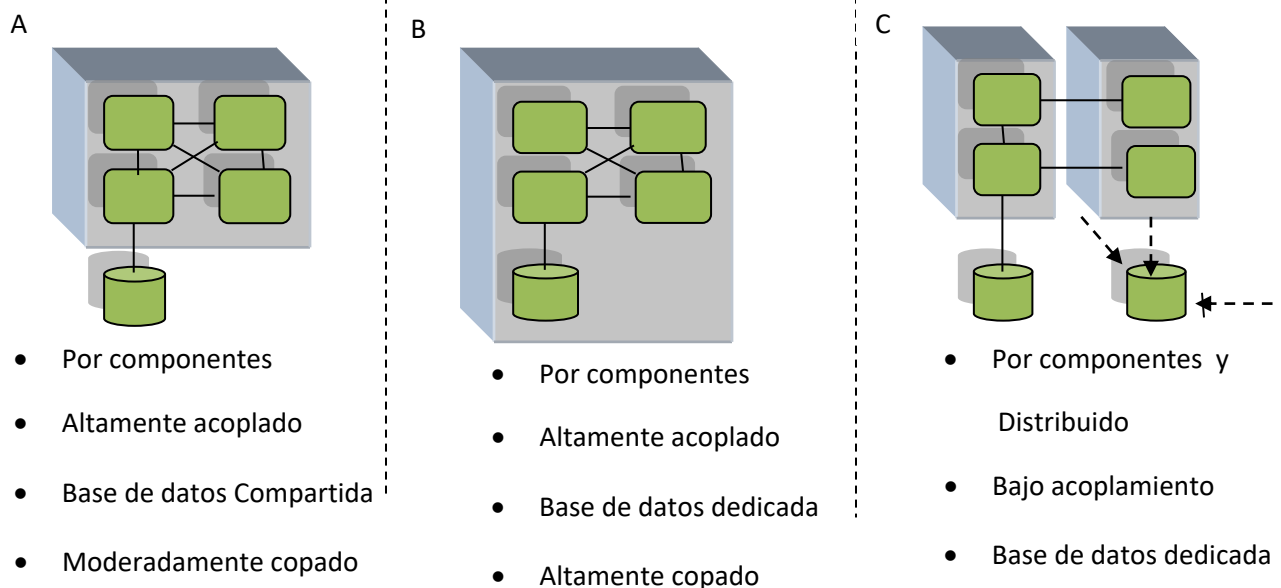


Figura 3 Comparación tres tipos de Diseño

En la figura 3 podemos ver tres tipos de diseño de una aplicación (A, B, C), cada uno de los cuales tiene sus propias características de diseño. Note que los pequeños rectángulos representan pequeñas unidades de solución lógica, las flechas solidas representan la reutilización de los objetos compartidos y las flechas interlineadas representan los estados de transferencia de datos.

### 3.1.2 PRINCIPIOS DE DISEÑO

Los principios de diseño son directrices o pautas recomendadas que hacen que una aplicación adquiera unas características de diseño requeridas o planeadas inicialmente, estos principios proveen reglas y directrices que ayudan a determinar exactamente como una solución lógica debería ser descompuesta y formada en unidades distribuibles [31]. Un estudio más detallado revela que estas unidades de características deberían ser clasificadas como cualidades del servicio capaces de cumplir con la visión y con los servicios asociados con SOA y computación orientada a servicios [31]. Cuando los principios vienen a construir soluciones estos son directrices o pautas que forman la solución lógica en un cierto modo y con ciertos objetivos en mente, objetivos que casi siempre están relacionados con el establecimiento de una o más características de diseño (características que son resultado de la aplicación de estos principios). En el caso de la figura 4 el acoplamiento entre la solución lógica A y B ha sido perdido (Como indica por una reducción en sus puntos de conexión). El principio de Bajo acoplamiento ha sido implementado en la aplicación, principio que genera que la interacción entre los componentes no sea tan dependiente como en la figura anterior.

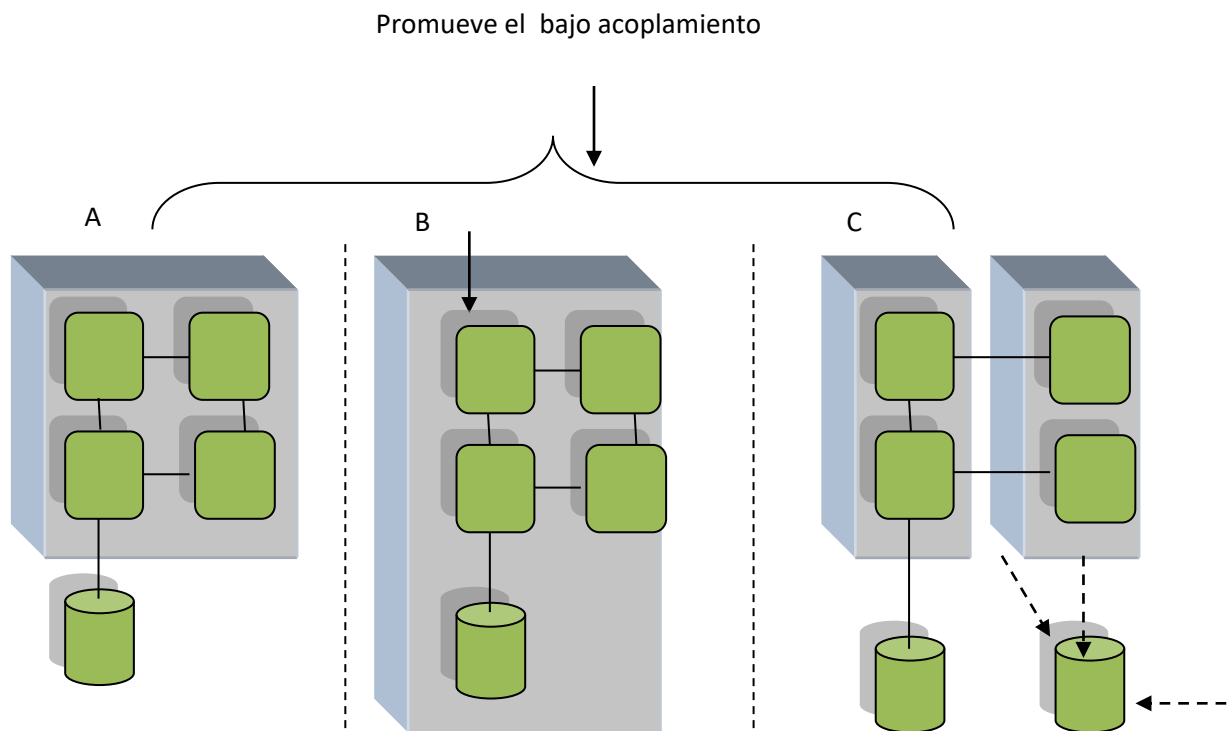
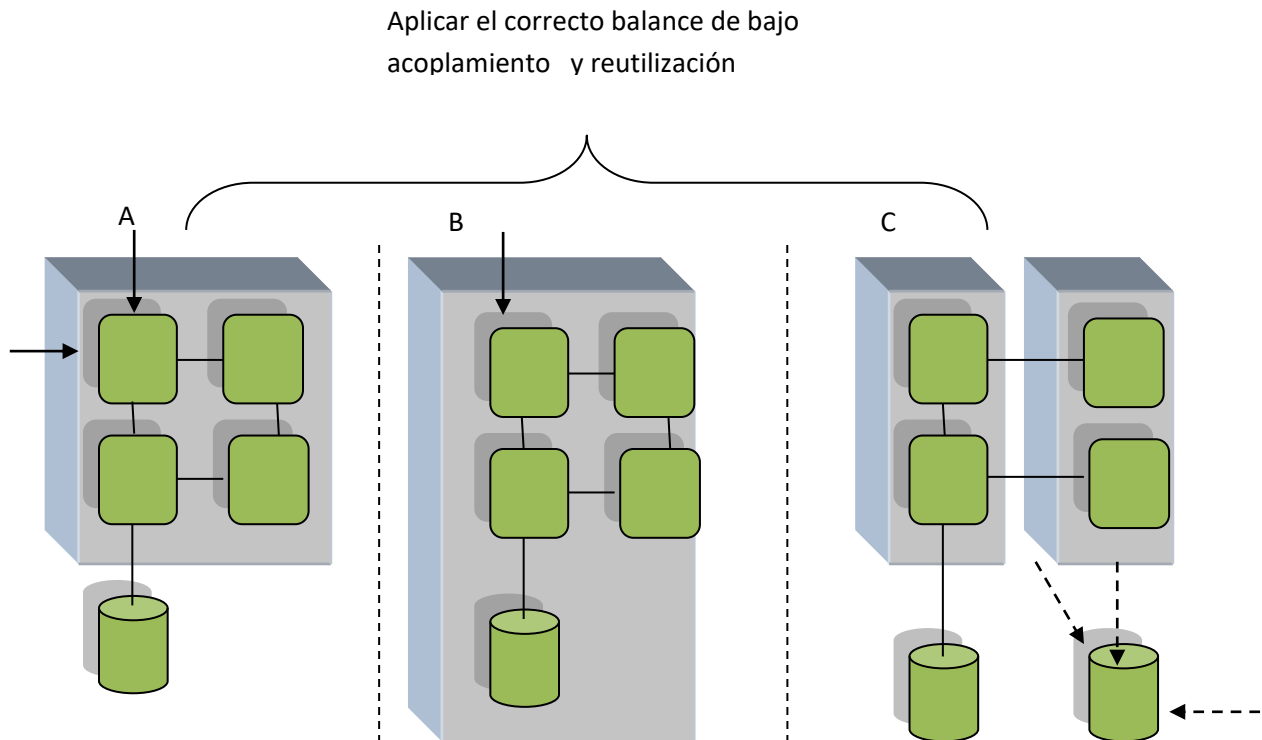


Figura 4 Aplicación Principio de Bajo Acoplamiento

### 3.1.3 PARADIGMAS DE DISEÑO

Estos están formado por una serie de principios o reglas complementarias de diseño y coordinadas al momento de ejecución por ejemplo: La programación orientada a objetos esta formado por una serie de principios de diseño que al ser aplicados logran como resultado que la solución lógica contenga ciertas características de diseño como modularización, que son comunes para ese tipo de paradigma y que cumplen ciertos objetivos [31].

El paradigma de diseño es una aproximación a una posible solución u orientación hacia el diseño de una solución lógica [31]. Es decir un paradigma de diseño intenta que una posible solución contenga los principios de diseño requeridos y planeados.



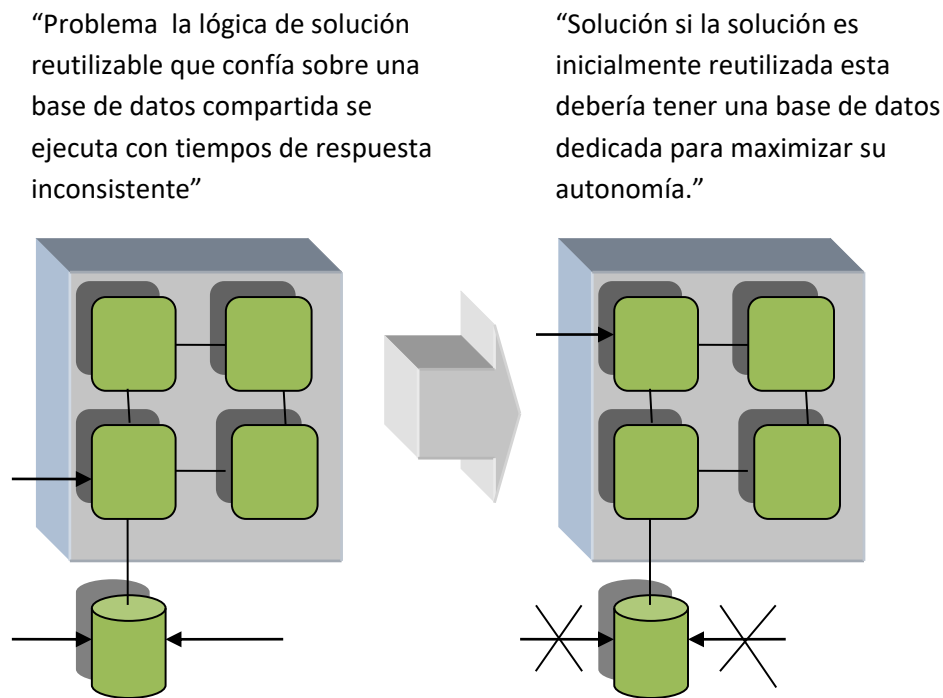
**Figura 5 Aplicación Paradigma**

Ya que un paradigma de diseño representa una colección de principios de diseño. Además incrementa el grado de características similares de diseño comunes a través de los diferentes cuerpos de la solución lógica. Por ejemplo en la figura 5 el monto de la reutilización en A y B se ha incrementado.

### 3.1.4 PATRÓN DE DISEÑO

Hasta el momento hemos dicho que la orientación a servicios es un paradigma comprometido por una serie de principios, cada uno de los cuales son una regla generalizada o directriz para la realización de ciertas características de diseño. [31]

Los patrones de diseño son soluciones a problemas planteados, describen un problema común y proveen una solución adecuada. Son plantillas de formato genérico que pueden ser aplicadas repetidamente. El conocimiento del Modelo de diseño no solo lo arma de entendimiento de los problemas potenciales de diseño a que pueden estar sujetos. Este provee respuestas para como estos problemas sean repartidos de la mejor forma [31].



**Figura 6 Aplicación Patrón de Diseño**

En la figura 6 un patrón sugiere reducir el acceso externo hacia una base de datos para incrementar la autonomía de la aplicación [31].

Los patrones de diseño nacieron de la experiencia. Pioneros en cualquier campo después de un ciclo de prueba y error lograron sus objetivos con éxito y aprendieron de sus errores. Cuando un problema y su correspondiente solución fueron identificados como comunes, la base de un patrón fue formada [31]. El patrón de diseño puede ser además combinado y componer patrones para resolver grandes problemas y una serie de patrones pueden formar la base de un lenguaje de patrones [31].

### 3.1.5 LENGUAJE DE PATRÓN DE DISEÑO

Cuando un modelo es implantado aparecen problemas, entonces es necesario utilizar otro modelo complementario para solucionar esos problemas se dice entonces que aparece un lenguaje de patrón de diseño, lenguaje que contiene patrones con soluciones comunes [31]. Un lenguaje del modelo esta formado por una cadena de patrones de diseño relacionados que establecen secuencias configurables en las cuales los modelos o patrones pueden ser aplicados, un lenguaje provee un significado práctico de los aspectos de fundamentales de la comunicación en un diseño dado ya que este ayuda con una documentación detallada de cada paso en un proceso de diseño que forma las características de diseño de la solución lógica [31].

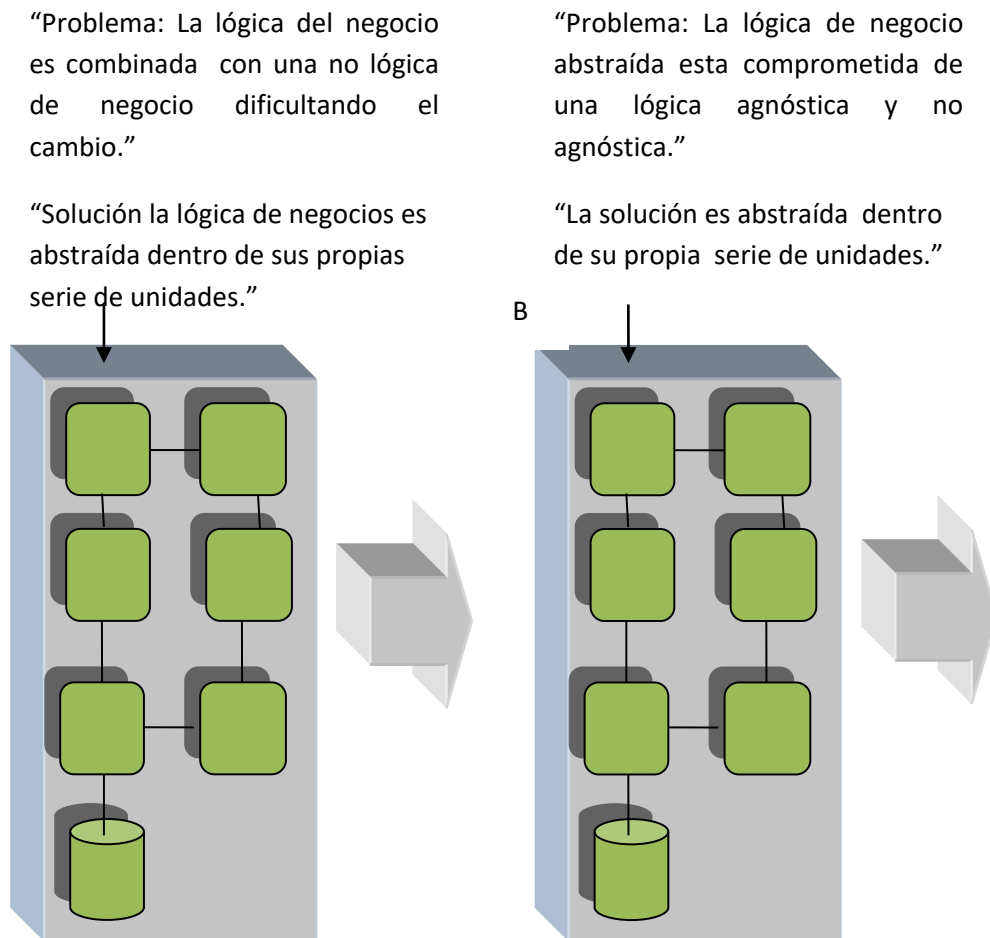


Figura 7 Lenguaje de Patrón de Diseño

Una secuencia de patrones de diseño relacionados formaliza el punto primario de un paradigma de diseño. En la figura 7 la lógica en aplicación del diseño B esta descompuesta como el resultado de un patrón y entonces lleva mas allá la descomposición como un resultado de otro. Patrones fundamentales subsecuentes continúan formando la lógica [31].

### 3.1.6 ESTÁNDAR DE DISEÑO

Los estándares de diseño (casi siempre obligatorios) son convenciones de diseño personalizadas para conscientemente personalizar la solución con sus características de diseño para el soporte de los objetivos de la organización y optimización para los ambientes empresariales específicos. A través del uso de estos estándares la organización puede conscientemente desarrollar soluciones moldeadas a sus ambientes, recursos, objetivos y necesidades y prioridades [31].

Para una organización el éxito de la aplicación de un paradigma requerirá más que a la adherencia de los principios asociados y un conocimiento en soporte de modelos de diseño. Cada organización tiene unos únicos objetivos estratégicos y un único ambiente empresarial, estos forman una serie de requerimientos distintos y restricciones que necesitan ser organizadas dentro de la solución de diseño. [31]

Como con los principios de diseño la aplicación de un estándar de diseño resultan características de diseño. Como con un modelo de diseño los estándares de diseño adoptan y refinan estas características para eludir problemas potenciales y para fortalecer el diseño de la solución global. ¿Usted puede tener un estándar de diseño sin principios de diseño? Si, actualmente es común tener muchos estándares de diseño de los cuales solo algunos pueden ser relacionados con principios de diseño para mirar a través de la aplicación del paradigma de diseño global. Diferentes estándares de diseño pueden ser creados simplemente para soportar otros objetivos o compensar por restricciones impuestas por el ambiente específico, cultural o por factores relacionados a la tecnología. Aunque algunos estándares pueden no tener una asociación directa con principios de diseño aceptados siempre debería haber un esfuerzo de mantener todos los estándares en una relativa alineación [31].

¿Pueden existir principios de diseño si estándares de diseño? Esto depende básicamente de ¿cómo una organización se compromete a dirigir el paradigma de diseño? Si esta mira que el potencial solo existe en una parte de los principios, puede darse que algunos principios de diseño no sean soportados por el actual estándar de diseño. Esencialmente con los principios de diseño a través de la estandarización queremos construir consistencia dentro de las características de diseño específicas, consistencia en la calidad de las características y en como frecuentemente ellas son implementadas [31].

También se derivan de los principios de diseño, con la diferencia de que ellos son propuestos para cumplir objetivos de diseños tenidos en mente y pensados para un ambiente empresarial en un proceso más estricto y documentado que explica el uso y aplicación de paradigma o patrón. En la figura 8 un estándar diseño requiere que C el original diseño sea alterado para eliminar el acceso hacia una BD compartida y externa

La seguridad especificada y requerimientos de privacidad de datos de estado no pueden ser compartidos en una base de datos separada.

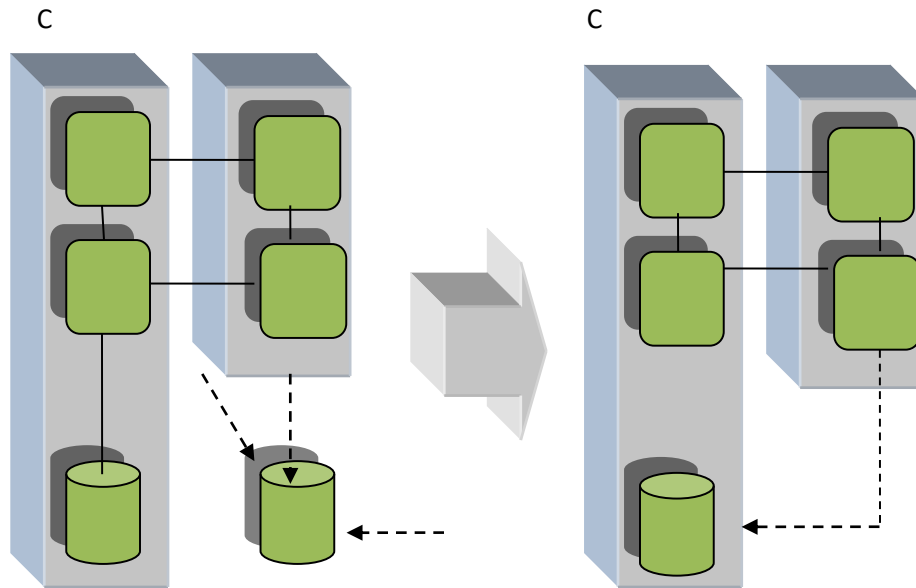


Figura 8 Estándar de Diseño

### 3.1.7 MEJOR PRÁCTICA

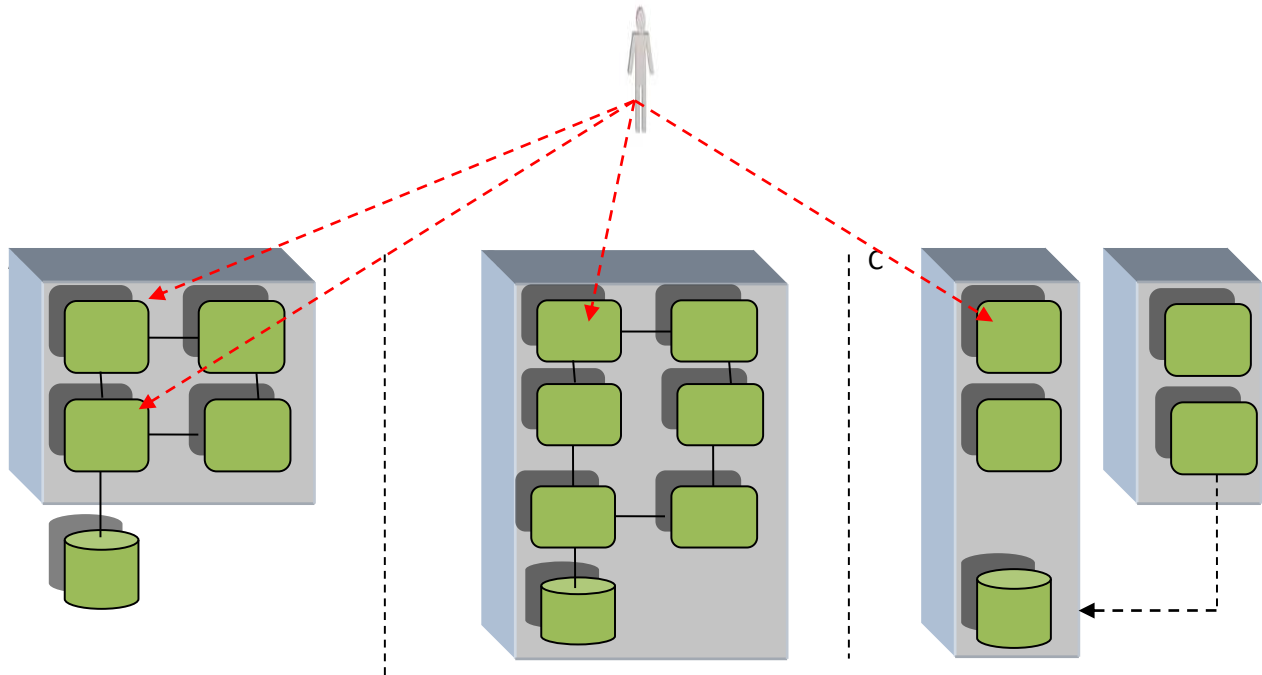
Una mejor práctica es conocida como una técnica de solución o prevención de ciertos problemas. Son recomendaciones de pautas y reglas a seguir que al ponerse a prueba obtenemos soluciones deseadas, que ya han sido obtenidos en desarrollos anteriores, son experiencias documentadas y aprobadas por la industria para tenerse en cuenta en desarrollo de una solución de negocio [31].

¿Cual es la diferencia entre una mejor práctica y un principio de diseño? Encontramos que una clara distinción que el principio de diseño esta limitado al solo hacer parte del proceso de diseño en el proceso de desarrollo. Una mejor práctica puede corresponder o ser referido hacia o por cualquier otro artefacto del proyecto entregado para solucionar problemas organizacionales [31].

Un principio de diseño puede verse como una mejor práctica asociada a la solución de diseño. Estas prácticas proveen una guía general, la figura 9 sugiere que el

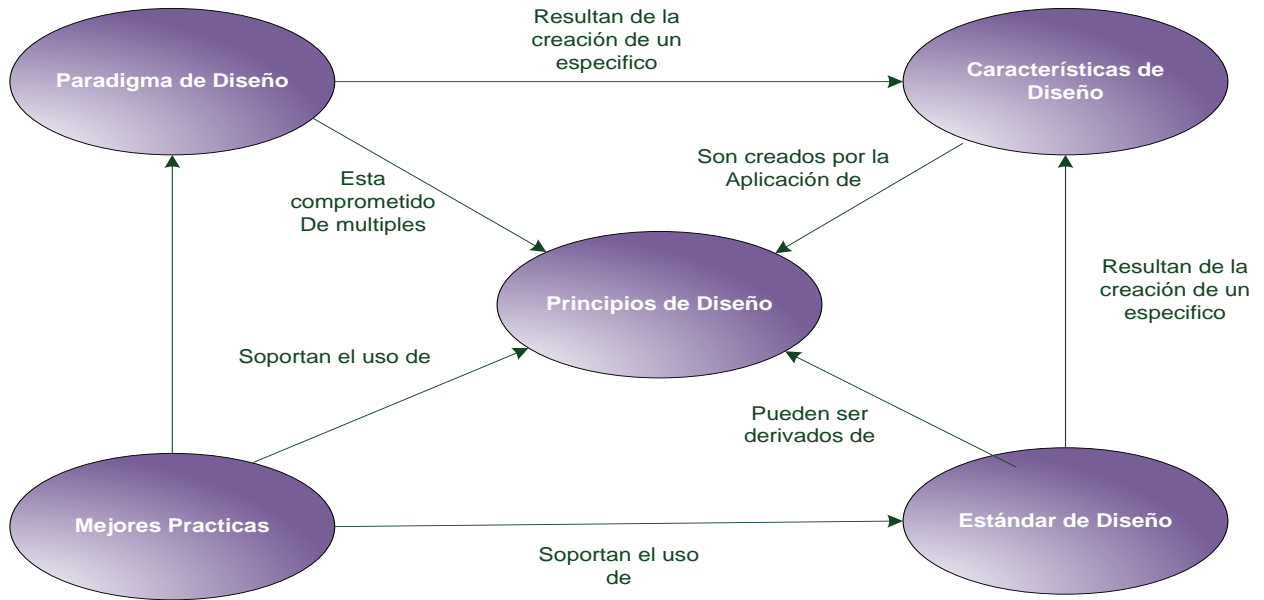


mantenimiento en marcha de unidades de solución lógica reutilizables de toda la aplicación está bajo un solo encargado. [31]



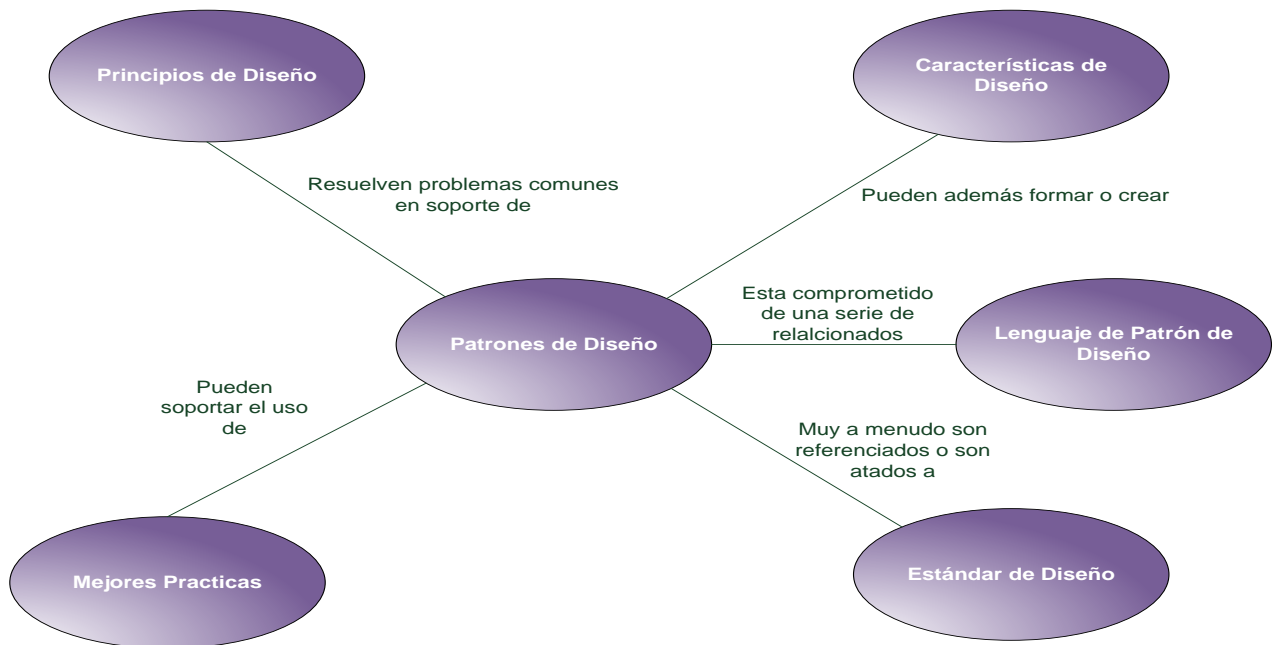
**Figura 9 Mejor Práctica**

Cada una de las piezas anteriormente descritas puede actuar como una pieza de entrada hacia un proceso de diseño global. Cuando diseñamos soluciones orientadas a servicios es casi inevitable que usemos algunas o todas estas piezas juntas. Es por eso importante comprender la relación existente entre ellas para que podamos ganar experiencia para saber como y donde fueron mejor utilizadas.



**Figura 10** Relación las partes del marco de Trabajo de Diseño

Términos de diseño fundamentales establecen una taxonomía básica usada posteriormente. El diagrama mostrado en la figura 10 indica la relación entre las partes del marco de trabajo de diseño.



**Figura 11** Relación las partes del marco de Trabajo de Diseño

Los patrones de diseño proveen una inteligencia adicional que pueden enriquecer un entorno de trabajo con una colección de soluciones probadas a problemas comunes, ver figura 11.

## **3.2 INTRODUCCIÓN A LA COMPUTACIÓN ORIENTADA A SERVICIOS**

La computación orientada a servicios representa una nueva plataforma de computación distribuida, como tal, abarca muchas cosas incluyendo su propio paradigma de diseño, principios de diseño, catálogos de modelos de diseño, modelos de lenguajes, un modelo de arquitectura distinto y conceptos relacionados y tecnologías [31]. La computación orientada a servicios construye sobre las plataformas pasadas y adiciona nuevas capas de diseño, consideraciones de dirección y una vasta serie de tecnologías preferidas [31]. Para comprender mejor el aspecto fundamental de una plataforma típica de computación orientada a servicios necesitamos describir cada parte primaria de esta, las cuales son referidas como elementos.

### **3.2.1 ARQUITECTURA ORIENTADA A SERVICIOS**

SOA establece un modelo arquitectural que anima a mejorar la eficiencia, agilidad y productividad para posicionar servicios. Sobre una base fundamental la plataforma de la computación orientada a servicios se envuelve alrededor del paradigma de diseño orientado a servicios y esta se relaciona con la arquitectura orientada a servicios [31]. Como modelo de arquitectura, una implementación SOA puede consistir en una combinación de tecnologías, productos, APIs, extensiones de soporte de infraestructura y varias otras partes. La cara actual de un despliegue de arquitectura orientada a servicios es única dentro de cada empresa sin embargo esta es tipificada por el ingreso de nuevas tecnologías y plataformas que soportan la creación, ejecución, evolución de soluciones orientadas a servicio.

### **3.2.2 PARADIGMA ORIENTACIÓN DEL SERVICIO**

La orientación del servicio es un paradigma de diseño comprometido por una serie de principios de diseño. El servicio existe como un programa de software físicamente independiente con características de diseño distinguidas que soportan el logro de objetivos estratégicos asociados con computación orientada a servicios. Cada servicio es asignado a su propio contexto funcional y es comprometido por una serie de capacidades relacionadas a ese contexto [31]. Estas actividades propias para ser invocadas por programas de consumo externos son comúnmente expresadas por vía de contratos de servicio publicados.

Un servicio es representado por una esfera amarilla o se puede utilizar el círculo alternativamente como lo indica la figura 12.

Símbolo que es usado para representar un servicio.

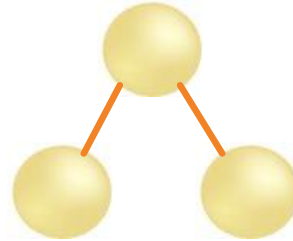


**Figura 12 Símbolo servicio**

### **3.2.3 COMPOSICIÓN DE SERVICIOS**

Es una agregación coordinada de servicios, una composición de servicios es comparable a una aplicación tradicional en que el alcance funcional es usualmente asociado con la automatización de un proceso de negocio, ver figura 13.

El símbolo comprende de tres esferas conectadas que representan una composición del servicio



**Figura 13 Representación de Composición de Servicios**

### **3.2.4 INVENTARIO DE SERVICIOS**

Un inventario de servicios es una colección independiente estandarizada y dirigida de servicios complementarios. Un ambiente de prestación de servicios de Información puede incluir un inventario de servicios que representan y conforman la arquitectura que los contiene.

Alternativamente, pueden existir en nuestro inventario, servicios autónomos y ajenos, pero necesarios dentro del contexto de la arquitectura.

Los inventarios de servicios son típicamente creados a través de un proceso de entrega de arriba hacia abajo que resulta de la definición del inventario de servicios. La aplicación subsecuente de los principios de diseño orientados al servicio y estándares de diseño personalizados durante un inventario de servicio es de importancia máxima para establecer un grado alto de interoperabilidad nativa entre el servicio. Este soporta la repetición, creación ágil de composiciones de servicio efectivo [31], ver figura 14.

El símbolo inventario de Servicios

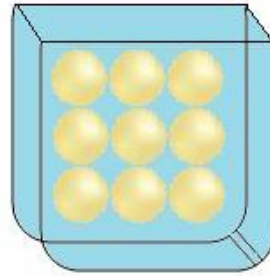


Figura 14 Símbolo Inventario de Servicios

### Elementos computacionales orientados al servicio

- **Arquitectura Orientada al Servicio:** representa una forma distinta de tecnología, diseño de arquitectura en soporte a la solución lógica orientada a servicios la cual esta comprometida de servicios y formada por composiciones de servicios, diseñados de acuerdo con la orientación a servicio.
- **Orientación a Servicios:** es un paradigma de diseño comprometido de principios de diseño orientados al servicio. Cuando se aplica a las unidades de solución lógica, estos principios crean servicios con características de diseño distintas que soportan los objetivos globales y la visión de la computación orientada a servicios.
- **Computación Orientada a Servicios:** Representa una nueva generación de plataformas de computación que comprende el paradigma orientado a servicio y la arquitectura orientada a servicios con el último objetivo de creación y ensamble de uno o mas inventarios de servicios.

Para apreciar completamente como estos elementos son usados necesitamos explorar como ellos se traducen en el mundo real. Para hacer esto, necesitamos distinguir claramente el rol y la posición de cada elemento dentro de una perspectiva de implementación física, como sigue:

- Solución lógica orientada al servicio esta implementada como servicios y diseñada por una composición de servicio de acuerdo con los principios de diseño orientado al servicio.
- Una composición consta de servicios que han sido ensamblados para proveer las funcionalidades requeridas para automatizar una tarea o proceso de un negocio específico.
- Ya que la orientación de servicios perfila muchos servicios como recursos empresariales, un servicio puede ser invocado por múltiples programas consumidores cada uno de los cuales puede implicar el mismo servicio pero en una composición de servicios diferente.
- Una colección de servicios estandarizados pueden formar la base de un inventario de servicios que pueden ser administrados independientemente dentro de su propio ambiente de desarrollo físico.

### 3.2.5 MODELOS DE SERVICIOS

Cuando construimos varios tipos de servicios. Se hace evidente que ellos pueden ser categorizados dependiendo de:

- El tipo de lógica que encapsulan
- El potencial generalización de reutilización que esta lógica tiene.
- Como esta lógica se relaciona con los dominios existentes dentro de la empresa.

Como un resultado, se presenta tres clasificaciones comunes que representan el modelado del servicio primario estas son: (ver figura 15)

- Entidad del Servicio
- Tarea del Servicio
- Utilidad del Servicio.

El uso de estos modelos de servicios resulta en la creación de capas de abstracción de lógica de servicios como se indica en la figura 16.

#### 3.2.5.1 Entidad de Servicios

El modelo de entidad de servicio representa un servicio céntrico del negocio que esta basado en una característica funcional y sobre el contexto de uno o más entidades del negocio relacionadas. La Entidad del Servicio es también conocida como entidad céntrica del servicio de negocio o servicios de entidad del negocio [31].

#### 3.2.5.2 Tarea del Servicio

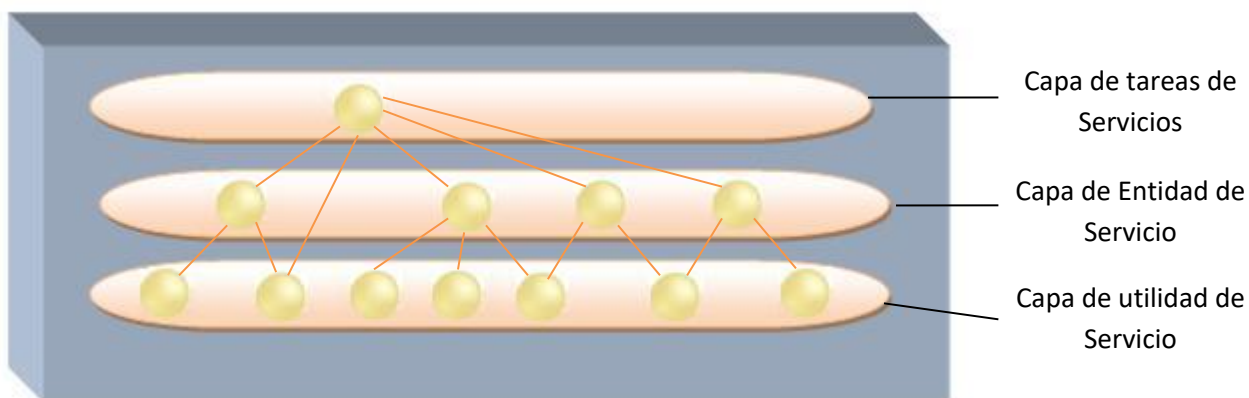
Un servicio del negocio con una cualidad funcional directamente asociada con una tarea específica o proceso, es basado sobre un modelo de tarea del servicio. Este tipo de servicios tienden a disminuir el potencial de reutilización [31].

#### 3.2.5.3 Utilidad del Servicio

Cada modelo de servicio anteriormente descrito, tiene un enfoque muy claro sobre la representación de la lógica del negocio. Sin embargo dentro del dominio de la automatización, este no siempre necesita asociar la lógica con el modelo o con el proceso de negocio. Por consiguiente puede ser altamente beneficioso deliberadamente establecer un contexto funcional que es el centro de la lógica del negocio. El modelo de utilidad del servicio logra esto buscando proveer funcionalidades y utilidades reutilizables. Estas son también conocidas como *aplicación de servicios*, infraestructura de servicios o tecnología de servicios [31].



**Figura 15 Clasificación Modelo Primario de Servicio**



**Figura 16 Capas de Modelo del Servicios**

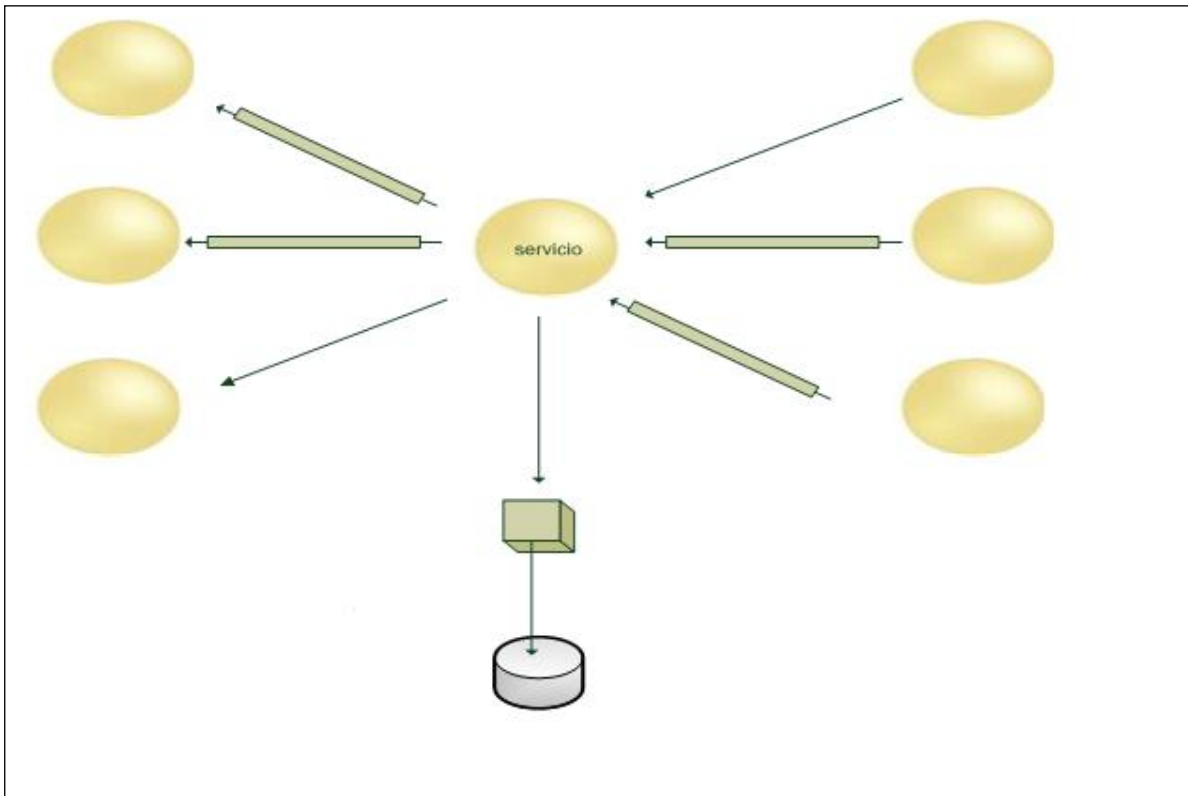
### 3.2.6 OBJETIVOS Y BENEFICIOS DE LA COMPUTACIÓN ORIENTADA A SERVICIOS

El paradigma de Orientación a servicios consiste en 8 principios de diseño distintos cada uno de los cuales sostiene características de diseño

### 3. 3 PRINCIPIOS DEL SERVICIO - ACOPLAMIENTO DEL SERVICIO

Cualquier módulo de un sistema que puede separarse tiene un potencial de acoplamiento hacia otra cosa con el fin de impartir este valor. La esencia de este término implica la existencia de dos cosas y que tienen una relación. Una medida de acoplamiento entre dos cosas es equivalente al nivel de dependencia que existe entre estos. Por ejemplo: La *interoperabilidad* puede estar asociada con una medida de acoplamiento que puede estar representada en la relación entre un programa software con otro. Lo anterior deja ver otra propiedad del acoplamiento que es la *direccionalidad* que se refiere a la dependencia bidireccional en que cada aplicación requiere la existencia de la otra o quizás incluso la disponibilidad de otras para funcionar propiamente.

La figura 16 enfatiza la reducción del acoplamiento entre las partes de una solución orientada al servicio especialmente cuando comparamos hacia cual aplicación tiene un diseño tradicional. En otro caso la unidireccionalidad es el caso en el que un programa depende de otro pero al contrario no. Una relación entre una BD y una aplicación es el ejemplo común de una aplicación *Unidireccional*.



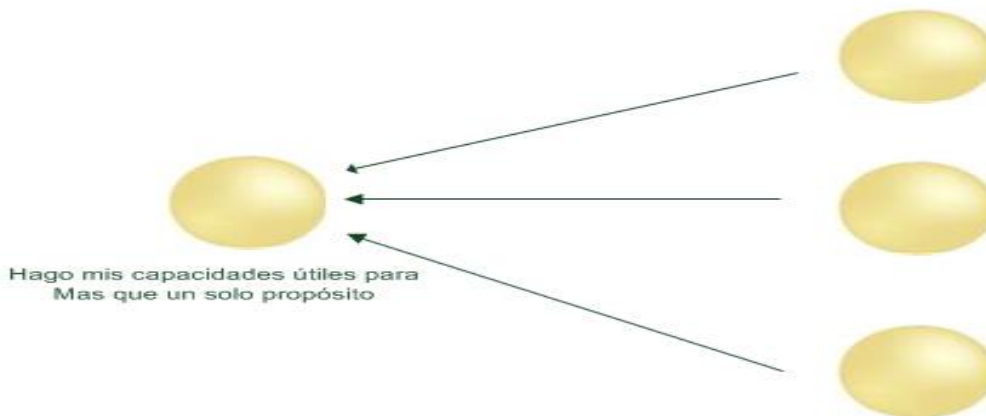
**Figura 17 Reducción Acoplamiento**

Un nivel de acoplamiento es comparable hacia un nivel de dependencia. El Acoplamiento software simplemente representa una conexión o relación entre dos programas o componentes. Muchos modelos arquitecturales pasados establecieron relaciones altamente acoplados a través de sus partes.

### **3.3.1 REUTILIZACIÓN DEL SERVICIO**

Puede proponer usar las capacidades del servicio y hacerlas útiles para más de un solo propósito. Simplemente hace a un programa útil para más que un sólo propósito. Históricamente lograr la reutilización no ha sido tan fácil como se ve en la figura 18.





**Figura 18 Reutilización del Servicio**

La reutilización es un término utilizado para indicar el potencial de un programa software para que sea reutilizable. Sin embargo el pasado intenta siempre lograr amplia reutilización que tenga resultados mixtos.

### **3.3.2 GESTIÓN DE ESTADO**

El estado se refiere a la condición general de algo. En la automatización del negocio, se entiende que un programa del software puede tener transición a través de diversos estados generalmente debido a su implicación en una actividad en tiempo de ejecución [31].

Cada estado se puede representar y describir por los datos que gestiona, por las funcionalidades que requiere, por el propósito y las transformaciones que hagan sobre estos datos. Consecuentemente, todas las variaciones de la información del estado tienden a ser temporales. Por lo tanto, la gestión del estado debe considerar la administración de datos temporales y los procesos que se apliquen a estos.

Los siguientes tipos de condiciones de estados pueden existir:

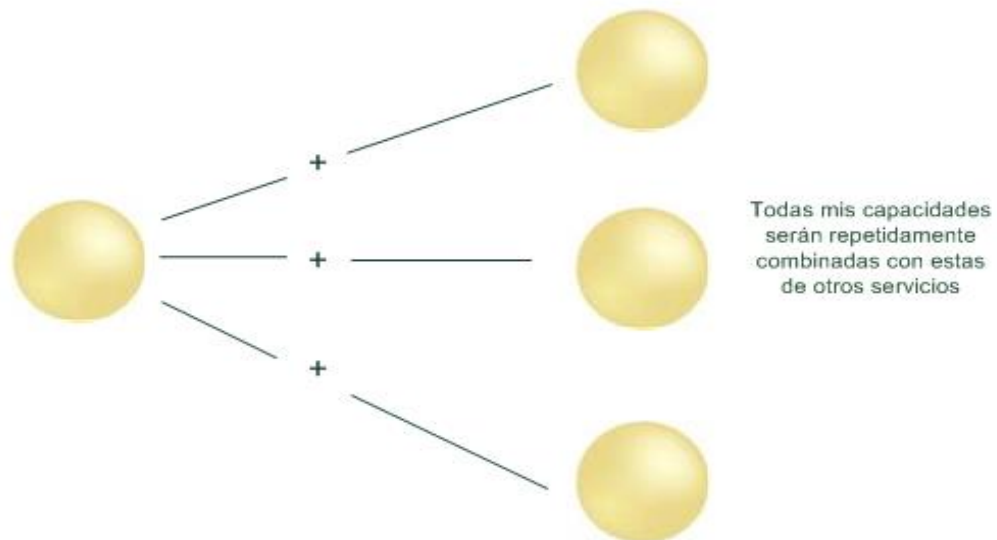
- Estados pasivos y activos
- Condiciones con estado lleno y sin estados.
- Contexto, sesión y estados de datos del negocio.
- Contexto de datos y contexto de reglas.

El estado de datos es información principalmente relacionada con una actividad actual y representa la gestión del estado y el procesamiento de esta información. Arquitecturas pasadas han cambiado de puesto la responsabilidad de la gestión del estado a través de capas cliente servidor[31].

Las extensiones arquitectónicas del aplazamiento del estado pueden ser empleadas para temporalmente aliviar programas de carga de gerencia de estado en orden para incrementar la escalabilidad global.

### 3.3.3 COMPOSICIÓN DEL SERVICIO

La composición del software no es nada nuevo. Muchos programas han sido compuestos de archivos y componentes que juntos forman un sistema funcional en tiempo de ejecución [31]. Entendiendo como se constituye la composición de software y como esta ha sido aplicada en el pasado se puede apreciar mejor el nivel de sofisticación que intenta aplicar la orientación a servicios, ensamblando capacidades de diferentes fuentes para resolver un gran problema; ese es el fundamento de la computación distribuida. Este principio introduce nuevas consideraciones de diseño que aseguran que el servicio sea capaz de participar en múltiples composiciones para resolver múltiples problemas.



**Figura 19 Composición del Servicio**

El concepto de composición es una parte fundamental del diseño de software donde la descomposición de la solución lógica aumenta la reutilización para que esta pueda ser recompuesta dentro de nuevas configuraciones y resolver una gran variedad de problemas. Varias formas de composición han existido en el pasado, al extenderse del ensamblaje de simple programación de simple librerías hacia más aproximaciones formales definida por la orientación a servicios[31].

### 3.4. ESTRUCTURA DEL PERFIL NIVEL – SERVICIO

Los siguientes puntos sirven como guía para formalizar la descripción de un servicio, este formato no es una estructura oficial del perfil de servicio, pero es útil para un entendimiento de cómo los servicios se especifican a lo largo del ciclo de vida del proyecto [31]:

- Nombre del Servicio
- Descripción Propósito (Corta): Una explicación concisa, una oración de descripción del contexto del servicio y del propósito.
- Descripción Propósito (detallada): Una explicación completa del contexto del servicio y su limitación funcional con los detalles necesarios.
- Modelo del Servicio: Entidad, utilidad, tareas, orquestación de tareas, o una variación personalizada.
- Requerimientos de calidad del servicio: Estos campos capturan varios requerimientos anticipados de calidad del servicio, características, o limitaciones que afectan el servicio como un todo. Los ejemplos incluyen seguridad, desempeño, disponibilidad y requerimientos de transacción (Cada uno de los cuales puede fácilmente justificar este propio campo en el perfil).
- Capacidades: El perfil debería documentar las capacidades que existen y están en desarrollo tales como las que son solo planeadas y tentativamente definidas. El código del color es también útil para hacer estas distinciones como es el uso de la capacidad y su estado.
- Palabras clave: Este campo puede contener uno o más palabras clave idealmente tomadas de un nivel taxonómico de inventario de servicio oficial o vocabulario. Las palabras clave del perfil del servicio debería corresponder a las palabras clave usadas por el registro del servicio.
- Versión: El número de la versión del servicio actual. Dependiendo sobre el sistema de control de versión en uso, el número de versión puede solo ser aplicado para capacidades del servicio.
- Estatus: El desarrollo del estatus del servicio es expresado en el campo usando términos estándar que identifican un estado del ciclo de vida del proyecto, tal como análisis, contrato de diseño, desarrollo, o producción. Si el servicio no está en producción este puede ser útil para incluir una fecha de desarrollo estimado.
- Guardián: Detalles sobre cómo alargar el guardián del servicio oficial o propietario, tal como otros que contribuyen para su documentación.

### 3.5 ESTRUCTURA DE LA CAPACIDAD DEL PERFIL.

Ya que los servicios actúan como un contenedor para una colección de capacidades adicionales sub-perfiles, los siguientes puntos son una guía para formalizar la descripción de una capacidad de un servicio como sigue [31]:

- Nombre de la Capacidad.
- Propósito: Una explicación concisa de su capacidad global y el contexto funcional (igual a la descripción corta del servicio).
- Descripción lógica: Es detallar paso a paso la lógica llevada a cabo por la capacidad. Esta puede ser complementada con algoritmos, diagramas de flujo de trabajo o incluso definiciones del proceso del negocio entero dependiendo de la etapa de desarrollo en que se encuentre.
- Entradas – Salidas: Estos dos campos proveen definiciones de una capacidad capas de entrar y sacar valores y restricciones asociadas. Estas pueden ser útiles para describir estas en un ingles sencillo durante la fase de modela miento del servicio.
- Rol Composición: La ejecución de la capacidad lógica puede poner un servicio dentro de varios roles temporales en tiempo de ejecución dependiendo sobre su posición dentro de configuraciones de composiciones del servicio.
- Capacidades Miembro – Composición: Una lista de servicios que componen la lógica de la capacidad. Esta provee una referencia cruzada hacia otras capacidades del servicio sobre cual capacidad actual ha formado dependencias. Idealmente identifica la capacidad miembro de la composición que serán mapeadas hacia las porciones de la lógica del proceso de negocio.
- Requerimiento Aseguramiento de la Calidad: Como con el campo correspondiente en la estructura del perfil del nivel del servicio, este campo es dedicado para recopilar detalles de la calidad del servicio. Sin embargo la información documentada corresponde especialmente a las capacidades del servicio lo cual significa que estas necesitan ser derivadas o correlacionadas con la calidad del nivel del servicio o detalles del servicio en algún caso.

## CAPITULO 4 DEFINICIÓN DE PERFILES P2P

### 4.1 PERFIL DE SERVICIOS

A continuación encontramos que la computación P2P demanda servicios que al agruparse en perfiles se convierte en un recurso software útil y mas completo para crear componentes reutilizables y adaptables para cualquier otra aplicación.

### 4.2 DEFINICIÓN DE PERFILES UML PARA SERVICIOS EN ENTORNOS P2P – JXTA

#### 4.2.1 SERVICIOS DE GRUPO DE PEERS

Un grupo de peers provee una serie de servicios colectivamente conocidos como servicios de grupo peers. JXTA define una serie de núcleos de servicios de grupos de peers [24]. Estos servicios de grupo de peers conforman la sociedad de servicios del grupo de peers, como cada peer que se une al grupo necesita implementar servicios de grupo. Adicionalmente estos servicios pueden ser desarrollados para entregar servicios específicos [24]. En este orden dos peers deben interactuar vía un servicio, ambos deben ser miembros del mismo grupo peer.

La parte principal de los servicios del grupo en el cual cada peer o nodo debe implementar en orden para participar en la red de JXTA son:

- **Servicio de Punto Final:** Es usado para enviar y recibir mensajes entre peers. Este utiliza el protocolo de enrutamiento de punto final a través de mínimas ejecuciones, provee solo una pequeña porción del protocolo completo [24].
- **Resolver Servicio:** Es usado para enviar peticiones de consulta genérica hacia otros peers. Estos pueden definir e intercambiar consultas para encontrar cualquier información que pueda ser necesitada.

En adición para el núcleo del servicio de grupo de peers se requiere obligatoriamente que cada uno adicione un estándar que sea comúnmente definido por cada grupo. No todos los estándares de servicios deben ser implementados por cada grupo esto depende en la definición del grupo de peers en donde se especifica si estos servicios pueden ser requeridos [24].

Los estándares para servicios de grupos peers más comunes son:

- **Servicio de Descubrimiento:** El Servicio de descubrimiento es usado por miembros de grupos peers para buscar recursos como, grupo de peers, pipes, servicios y nodos.
- **Servicio de Afiliación:** El servicio de afiliación es usado por miembros de grupos para establecer identidades firmemente y confiar dentro de un grupo. Las identidades permiten a la aplicación y servicios determinar quien está enviando la petición de una operación y decide si esta operación debería ser permitida. Las

aplicaciones pueden desempeñar su propio control de acceso o pueden usar el servicio de acceso. La definición de grupo especifica las políticas por las cuales un nodo puede establecer una identidad.

- **Servicio de acceso:** Es usado para validar peticiones hechas por un peer hacia otro. El receptor de la petición chequea las credenciales del emisor que envía la petición y la información acerca de la petición será hecha para determinar si el acceso es permitido.
- **Servicio Pipe:** Este servicio es usado para crear y administrar conexiones pipe o canales de comunicación entre ellos miembros del grupo de peers.
- **Servicio de Monitoreo:** Es usado para permitir que un nodo monitoree otros miembros del mismo grupo.

#### 4.2.2 DEFINICIÓN DEL PERFIL UML PARA EL SERVICIO ESTÁNDAR DE DESCUBRIMIENTO

Si se quiere implementar una aplicación P2P se deberá iniciar con funcionalidades como las anteriores, que no forman parte de la lógica de la aplicación por lo tanto estas deberían encapsularse en una entidad aparte e independiente.

La funcionalidad de descubrimiento en JXTA es la capacidad de identificar cualquier recurso publicado por un peer al momento que este inicie su conexión al grupo inicial por defecto. Los recursos son representados por medio de anuncios. Un recurso puede ser un peer, pipe, servicio, un grupo de peers o cualquier otro que tenga un anuncio [24].

Esta funcionalidad es implementada por medio del servicio pipes el cual es de bajo nivel y realiza funcionalidades de envío y recepción de mensajes; un peer lo usa para enviar mensajes, ellos pueden ser asíncronos, unidireccionales e inseguros. Hay 6 protocolos estándar de comunicación entre los cuales están PDP (Protocolo de Descubrimiento de Peers) el cual trabaja a través del modelo consulta /respuesta es decir un peer para descubrir cualquier recurso en la red envía un mensajes de consulta este mensaje es respondido por los peers que estén en línea.

El servicio de descubrimiento escucha y detecta permanentemente anuncios de peers conectados por medios de servicios de más bajo nivel como el servicio de pipes o canal de comunicación anteriormente explicado.

Para la definición del perfil se ha optado por establecer varios estereotipos: <<Publicador>> representa un objeto que publica anuncios, <<escuchador>> representa un objeto que permanentemente está escuchando anuncios y <<Pipe canal>> representa el canal de comunicación mediante el cual el publicador y el escuchador podrán detectar los anuncios en la red, ver figura 20.

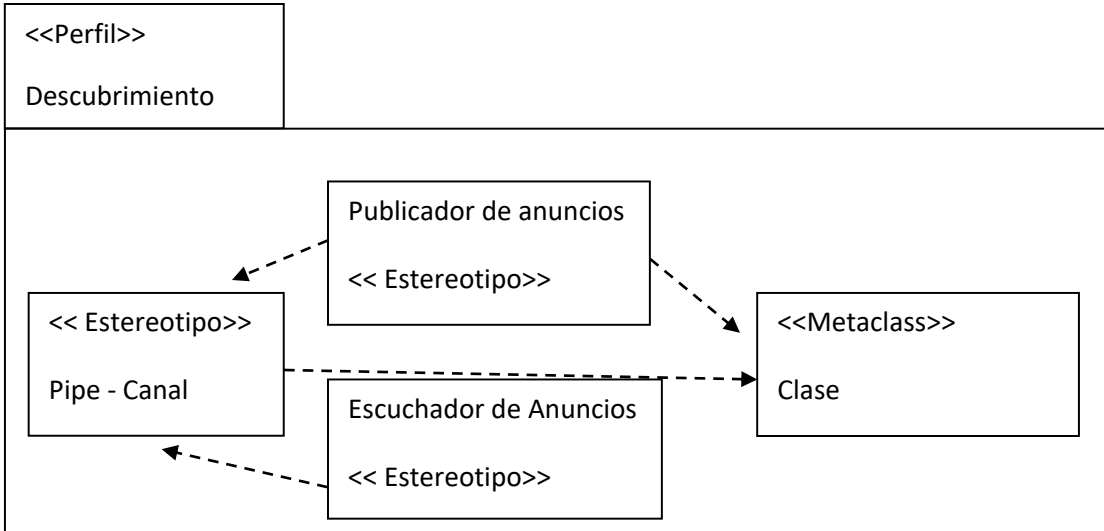


Figura 20 Perfil UML para el servicio estándar de Descubrimiento de JXTA

La figura anterior representa las relaciones entre los estereotipos que conforman el perfil de descubrimiento.

#### 4.2.3 DEFINICIÓN DEL PERFIL UML SERVICIO ESTÁNDAR DE AFILIACIÓN O MEMBRESÍA

El servicio de afiliación es el responsable de la administración de identidades dentro de JXTA. Estas identidades son normalmente pasadas o comunicadas como parte de protocolos de mensajes de JXTA. Permiten a las aplicaciones y servicios determinar quien es el que envía la petición u operación y decide si esta operación debería ser permitida. Las aplicaciones pueden desempeñar su propio control de acceso o pueden usar el servicio de acceso de JXTA. Este servicio de acceso toma decisiones de consideraciones de acceso con respecto a identidades y operaciones a permitir. Todos los servicios de miembros o afiliación administran identidades usando una estructura en común la credencial JXTA. Una credencial es una ficha la cual es pasada entre los peers para comunicar la identidad. Son normalmente inteligibles esto significa que solo pueden ser interpretadas por el servicio de membresía o afiliación el cual creo estas. Las credenciales típicamente incluyen mecanismos de validación para asegurar que ellos no puedan ser fácilmente olvidados.

Las credenciales son creadas por servicios como un resultado de un proceso de autenticación.

El servicio de membresía puede generar dos formas de credenciales. Las locales son generadas vía autenticación y pueden ser usadas por servicios y aplicaciones para mensajes sean enviados por el peer local. Las remotas son generadas de mensajes recibidos de un peer remoto y es usado por aplicaciones y servicios para determinar la identidad del peer remoto.

El alcance del servicio de afiliación es solo un grupo peer. De hecho cada instancia de servicios de afiliación es asociada con un grupo peer. La implementación de JXTA JSE provee un servicio de membresía con cada grupo peer. El cual es usado para un grupo peer particular es parte de una definición de grupos peer. Un servicio de membresía es iniciado cuando el grupo peer inicia. Desde que el servicio de membresía es un servicio requerido por el grupo de JXTA JSE todas las aplicaciones y servicios pueden depender sobre el hecho de que este servicio siempre estará disponible, aunque no siempre se puede conocer qué servicio de membresía un peer puede proveer. Aplicaciones y servicios normalmente interactúan dentro del servicio de membresía o de problemas de credenciales por el servicio de membresía.

Cada peer al momento de ingresar en la red puede crear un grupo y publicar su existencia con la ayuda de un anuncio que lo describe específicamente. Otro peer que desee pertenecer al primero tendrá que detectar los anuncios de grupo disponibles luego tendrá que enviar una petición de querer ingresar; esta petición es atendida y resuelta satisfactoriamente si el grupo aun existe. Al momento que este peer ya es miembro podrá detectar los pertenecientes a este. Como podemos ver este servicio funciona de manera similar al de descubrimiento, es decir se necesita un objeto publicador de anuncios, en este caso contendrá la intención de afiliación del peer y por otro lado estará el escuchador el cual podrá aceptar a cualquier futuro miembro interesado. Entonces podemos definir varios estereotipos que cumplen una funcionalidad importante en esta actividad.

#### **4.2.3.1 OPERACIÓN BÁSICA DEL SERVICIO DE MEMBRESÍA**

La operación fundamental del servicio de membresía en una aplicación o servicio puede usar el método `getDefaultCredential()`. Este método retorna la credencial por defecto. La credencial permanecerá válida hasta que el servicio de membresía es terminado o la credencial expira o hasta que sea inválida. La aplicación puede chequear la validez de la credencial usando el método `isValid()`. La aplicación o servicio puede también proveer un método `CredentialListener` para que la credencial que genere un callback cuando se vuelva inválida o expire. Esta aproximación es mejor que estar chequeando continuamente si es válida o no. Las aplicaciones puede pedir una serie completa de credenciales locales activas por medio de `getDefaultCredential()`. Este retornará todas las credenciales activas. Las Aplicaciones también pueden registrar escuchadores para recibir notificaciones siempre que una nueva credencial sea generada o la credencial por defecto cambie.

La mayoría de las aplicaciones o servicios incluyen funcionalidades estándares que usaran la credencial local por defecto en su protocolo de mensaje para identificar el peer.



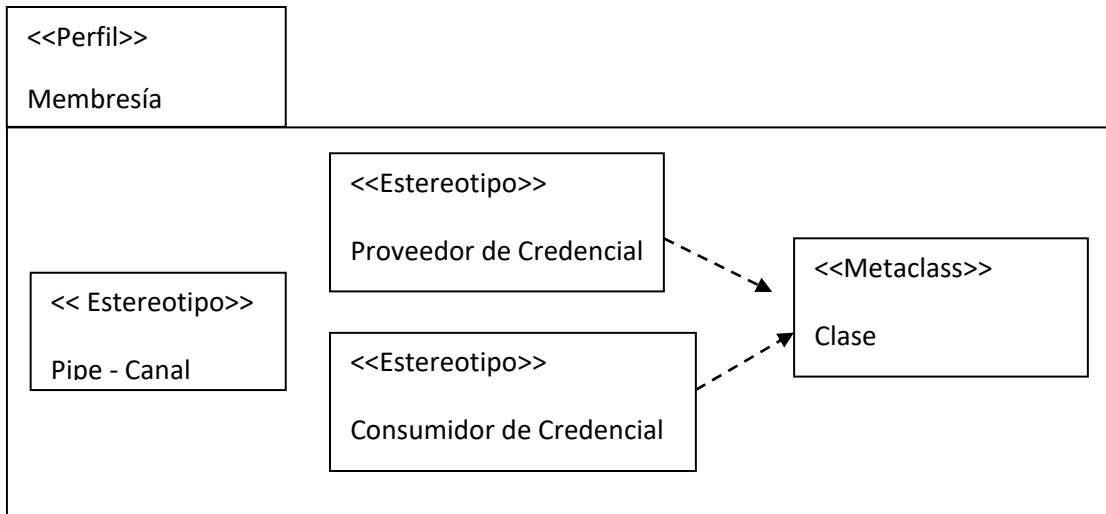


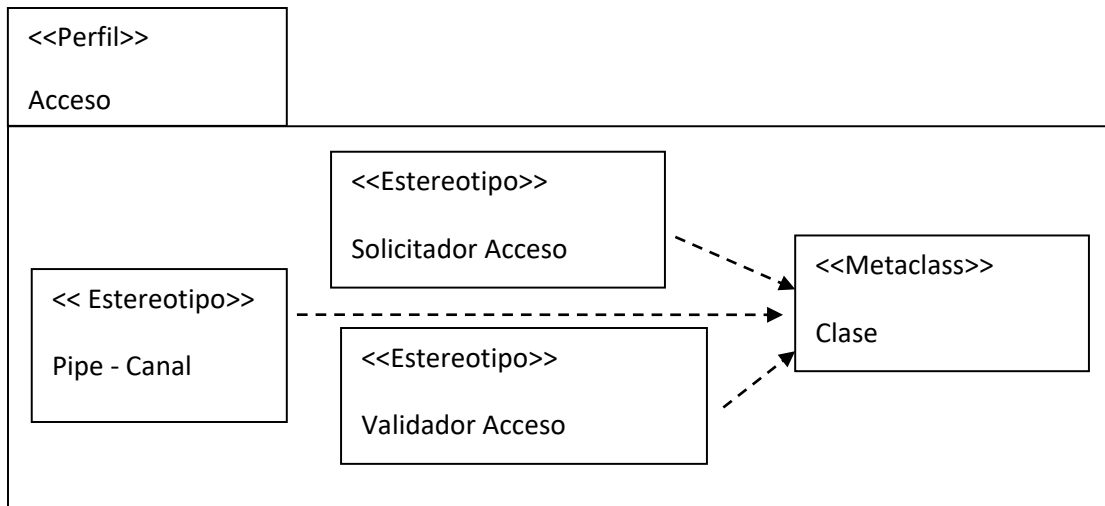
Figura 21 Perfil UML para el servicio estándar de Membresía de JXTA

La figura 21 representa la relación entre los estereotipos que conforman el perfil de membresía.

#### 4.2.4 DEFINICIÓN DEL PERFIL UML PARA EL SERVICIO ESTÁNDAR DE ACCESO

La mayoría de servicios de membresía no proveen una credencial por defecto en el inicio. De hecho un peer local tiene que establecer una credencial. La primera identidad autenticada también se volverá la credencial por defecto. Unos servicios de membresía pueden proveer uno o más métodos de autenticación. Cada método generalmente provee un mecanismo apropiado para un estilo particular de autenticación. Estos pueden incluir pantallas gráficas de usuarios y otros métodos. Las aplicaciones son libres de escoger el método de autenticación mas apropiado para sus necesidades. Cada servicio de membresía provee por lo menos un método de autenticación y puede proveer una interfaz interactiva de autenticación.

El proceso de autenticación de los servicios de membresía tiene tres pasos distintos: configuración, aplicación y validación. El paso de configuración ocurre antes de que el servicio de membresía es invocado, en él se escoge un método de autenticación y se determinan los parámetros para el método. Si el servicio de membresía determina que la autenticación puede comenzar entonces una autenticación exitosa es retornada. El paso de la aplicación comprende concluir con la autenticación hasta que esta retorne verdadero.



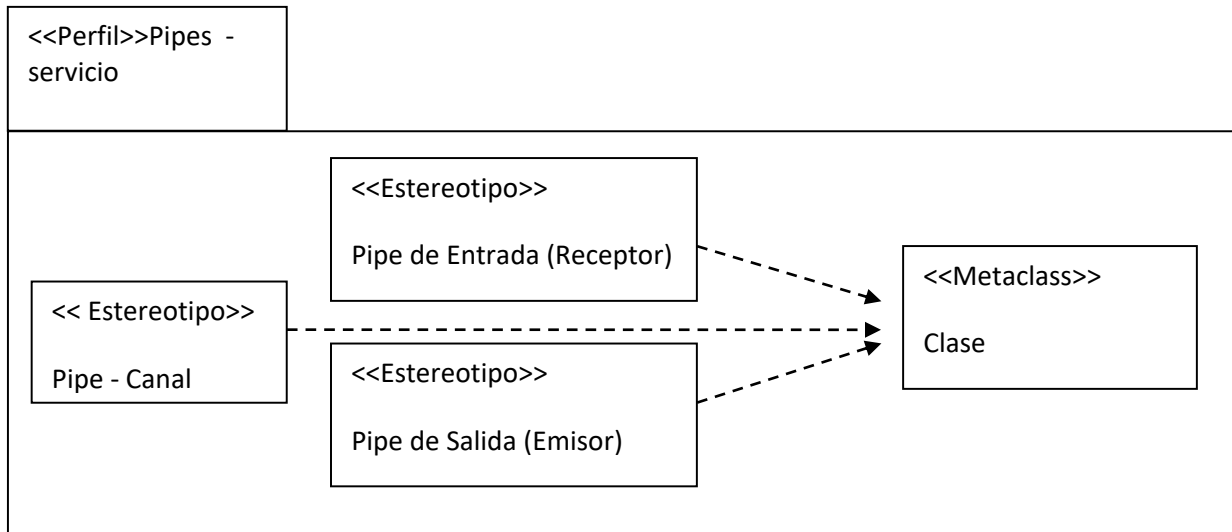
**Figura 22 Perfil UML para el servicio estándar de Acceso de JXTA**

La figura 22 representa las relaciones entre los estereotipos que conforman el perfil de Acceso

#### **4.2.5 DEFINICIÓN DEL PERFIL UML PARA EL SERVICIO ESTÁNDAR DE PIPE**

La clase de pipeService define una serie de interfaces para crear y acceder a servicios pipes dentro de un grupo peer. Estos servicios prestan mecanismos claves para intercambiar mensajes entre dos aplicaciones JXTA o servicios. Estos proveen un canal unidireccional simple y asíncrono de comunicación entre dos peers. Los mensajes de JXTA son intercambiados entre pipes de entrada y salida. Una aplicación que quiere abrir un canal de comunicación de recepción con otro peer crea un servicio de entrada y se une hacia un anuncio específico de pipe. La aplicación entonces publica el anuncio de manera que otras aplicaciones o servicios puedan obtener el anuncio y creen el correspondiente servicio de salida para enviar mensajes al pipe de entrada.

Los anuncios servicios pipes son identidades únicas en JXTA. El id del pipe único es usado para crear la asociación entre los pipes de entrada y salida. Estos son canales de comunicación no localizados que no obligan a los peers específicos. Esta es la única característica de los pipes JXTA. La manera de resolver la localización de un peer hacia un peer físicamente es hecho de una manera descentralizada en JXTA es por medio de una unión o el protocolo PBP de JXTA. Este protocolo usa un mecanismo adaptativo, dinámico de búsqueda que intenta siempre encontrar los peers donde una instancia de los servicios pipes este en ejecución.

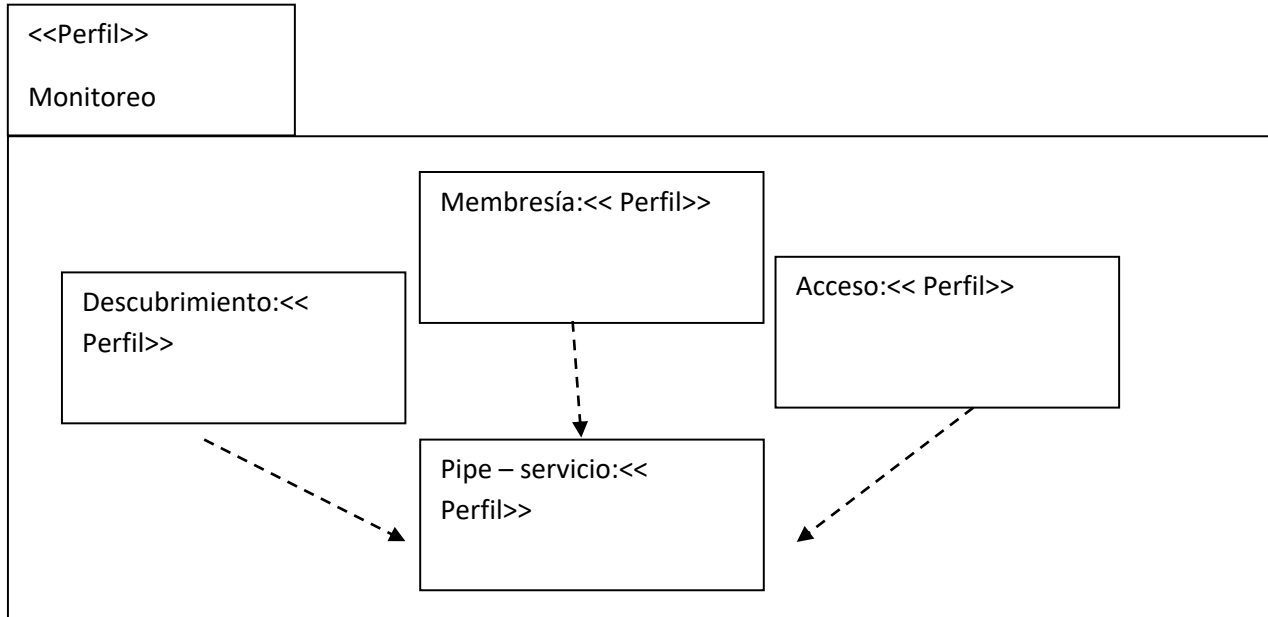


**Figura 23 Perfil UML para el servicio de pipes de JXTA**

En la figura 23 representa los estereotipos para el servicio de pipe en JXTA, como hemos dicho este servicio inicialmente crea un canal de entrada el cual es el receptor o hace el rol de receptor de mensajes de cualquier conducto de salida en la red o grupo que este afiliado el peer. También encontramos el estereotipo de Emisor que es el pipe de salida no olvidemos que ambos pipes son conocidos por medio de anuncios en los cuales se comunica toda información única de cada uno de ellos.

#### **4.2.6 DEFINICIÓN DEL PERFIL UML PARA EL SERVICIO ESTÁNDAR DE MONITOREO**

El servicio de monitoreo está diseñado para que un peer pueda estar informado permanentemente del estado de un peer en la red. Este servicio puede estar enfocado en un solo nodo o varios, este está conformado por varios servicios y perfiles anteriormente definidos por ejemplo: Los servicios de descubrimiento, membresía, acceso y pipes, como podemos observar, conforman en conjunto el monitoreo, podemos decir entonces que el servicio esta compuesto por varios servicios de JXTA.

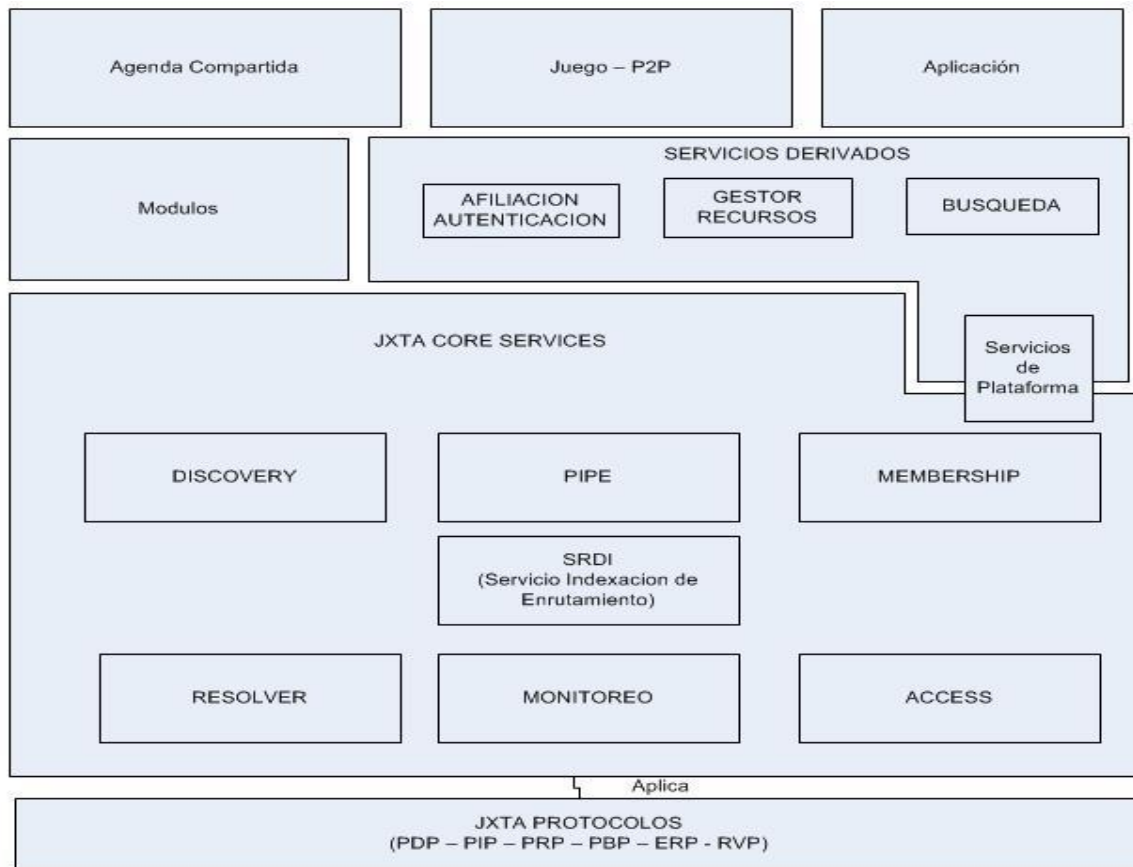


**Figura 24 Perfil UML para el servicio estándar de Pipe de JXTA**

La figura 24 representa la relación entre los estereotipos que conforman el servicio estándar de Pipe.

En resumen hemos definido cuatro perfiles: Descubrimiento, Membresía, Acceso y monitoreo, los cuales se construyen sobre el servicio de pipes. El perfil de descubrimiento necesita crear un canal de comunicación para enviar anuncios que indiquen los nuevos nodos que están conectados al grupo por defecto y a su vez tendrán que escuchar por ese canal los anuncios que indican los nuevos recursos disponibles. El servicio de membresía debe establecer credenciales a cada peer que inicia la plataforma JXTA, estas credenciales identificarán a sus dueños para cualquier acción futura, para hacer esto el servicio de membresía debe contar con el envío de anuncios que comuniquen todas los datos de sus credenciales. El servicio de acceso también necesita validar sus identificaciones por medio del paso de mensajes y publicación de anuncios.

## 4.2.7 ARQUITECTURA APLICACIÓN PROTOTIPO



**Figura 25 Arquitectura Aplicación Prototipo**

La figura 25 representa la arquitectura de JXTA con la diferencia del módulo de servicios derivados el cual es el aporte de este trabajo, estos servicios son de un nivel más alto que los servicios base de JXTA. Los servicios derivados están compuestos por los servicios base e intentan facilitar el diseño y desarrollo de una aplicación P2P. Los servicios de Plataforma agrupan funcionalidades tales como inicialización y descubrimiento, están estrechamente relacionados con los servicios base de JXTA.

Los servicios de afiliación permiten gestionar la creación y unión de grupos P2P, los servicios de gestión de recursos permite publicar cualquier tipo de archivo a la red y ser distribuido sin problemas, los servicios de búsqueda consultan recursos en la red. En conjunto estos servicios representan los nuevos elementos o artefactos a un modelo P2P que está en sus primeras etapas de maduración.

### **4.3 PERFILES DERIVADOS**

Anteriormente hemos definido varios perfiles básicos en el funcionamiento de una aplicación P2P, estos perfiles pueden formar la base para la construcción de perfiles derivados que al parecer no hacen parte de una lógica del negocio particular si no por el contrario son vistos como funcionalidades necesarias pero independientes de cualquier lógica de aplicación P2P. Estas funcionalidades pueden ser por ejemplo: El proceso de registro de usuario o inicio de sesión, el cual a su vez debería configurar e iniciar la plataforma base en la cual va a correr el nodo, debería ser una funcionalidad ya modelada y fácil de reutilizar en cualquier desarrollo. El proceso de descubrimiento de nodos en la red, también debería ser una funcionalidad de fácil acoplamiento, así como funcionalidades relacionadas con la gestión de grupos y distribución de archivos, etc.

Es por eso que hemos definido tres perfiles que pueden ser derivados de los servicios básicos de una aplicación P2P, los cuales son: Perfil de servicios de Plataforma, Perfil del servicio de afiliación o gestión de grupos, y el Perfil de gestión de recurso el cual cumplirá funciones relacionadas con la distribución de recursos y demás.

#### **4.3.1 METAMODELO P2P SERVICIOS DE PLATAFORMA**

En una arquitectura P2P se hace necesario las siguientes clases que van a cumplir funciones relacionadas con la configuración e inicio de la plataforma en la que el nodo se va a conectar y va a detectar a los demás nodos en la Red. Ellas realizan procedimientos de iniciación de plataforma e implementan los protocolos necesarios para la detección de los nodos.

### 4.3.2 ESPECIFICACIÓN FORMAL PERFILES DERIVADOS.

#### 4.3.2.1 ESTRUCTURA DEL PERFIL - SERVICIOS DE PLATAFORMA

Estructura del Perfil - Servicios de Plataforma
Nombre del Servicio: Servicios de Plataforma.
Descripción Propósito: El servicio de plataforma gestionará varias funcionalidades que son necesarias para la implementación inicial de una aplicación P2P. Este servicio proveerá una interfaz amigable para que el usuario pueda registrarse e iniciar la aplicación de una manera segura. Este servicio deberá por debajo configurar todo lo relacionado con la plataforma P2P utilizada además deberá iniciar dicha plataforma. Este servicio tendrá capacidad de detectar los demás nodos en la red que estén conectados y que cumplan con una política configurable definida específicamente para cada aplicación. Para no permitir el ingreso de nodos desconocidos a la Red.
Modelo del Servicio: Este servicio es clasificado como una utilidad esencial para cualquier aplicación P2P. El cual podrá ser extendido para prestar todas sus funcionalidades anteriormente descritas.
Requerimientos Calidad del Servicio: Este servicios deberá estar disponible en cualquier momento ya que si el ningún peer podrá ingresar a la red. A demás deberá trabajar con sub – servicios que constantemente deberán estar funcionando. La seguridad también es una característica importante ya que este servicio inicia un nodo a la red identificándolo, esta no puede ser falsa ni suplantada. El desempeño también debe ser un punto crítico en su funcionamiento ya que este servicio formara la base de cualquier aplicación P2P.
Capacidades: <ul style="list-style-type: none"><li>• Inicio-configuración Plataforma: Proceso relacionado con la configuración de una plataforma P2P. En donde se puede fijar que tipo de nodo va ser en la Red, se registraran datos de identificación como el nombre, clave de acceso, IP, nombre PC, etc., se inicia la plataforma es decir se realizan todas las operaciones de configuración e inicialización necesarias.</li><li>• Descubrimiento: Capacidad de detectar e identificar cada peer que se conecta a la red en todo momento.</li></ul>
Palabras clave: <ul style="list-style-type: none"><li>• Modelos de utilidad del Servicio</li><li>• Disponibilidad del Servicio</li><li>• Agenda Conjunta</li><li>• Credenciales de Administración</li><li>• Protocolos de Comunicación de JXTA.</li></ul>
Versión: Versión inicial para todas las capacidades. 1.0

Estatus: El servicio se encuentra en etapa de diseño. Se estima dos semanas de desarrollo e implementación.

#### 4.3.3 Estructura Formal Capacidad: Inicio - Configuración Plataforma.

Nombre de la Capacidad: Inicio- Configuración Plataforma.

Propósito Descripción: Proceso relacionado con la configuración de una plataforma P2P. En donde se puede fijar que tipo de nodo va ser en la Red, se registraran datos de identificación como el nombre, clave de acceso, IP, nombre PC, etc., se inicia la plataforma en donde el nodo podrá detectar a otros.

Lógica Descripción: Esta capacidad será muy necesaria en el momento en que cualquier nodo o peer que quiera conectarse a la red P2P. Ya que esta se encarga de configurar diferentes variables importantes en una red de igual a igual. Al modelarla podemos ver un estereotipo de administrador de red el cual será el encargado de: Proveer una manera simplificada la configuración de plataforma. El proceso general que este servicio llevara a cabo será el siguiente:

1. Creamos nodo configurándolo con los parámetros de entrada necesarios.
2. Guardamos los archivos de configuración para que este proceso no se realice cada vez que se inicie la aplicación.
3. Iniciamos la Plataforma.

Entradas – Salidas:

Para el proceso de inicio y configuración de plataforma contara con entradas como:

Parámetros de Entrada

- Tipo de nodo
- IP – Nombre PC

Rol Composición: Administrador de Red

Capacidades Miembro:

Requerimiento Aseguramiento de la Calidad: La capacidad de descubrimiento deberá tener varias características de calidad importantes ya que esta formara la base de aplicaciones mas grandes entre las cuales son : fiabilidad, disponibilidad, reutilización, etc.

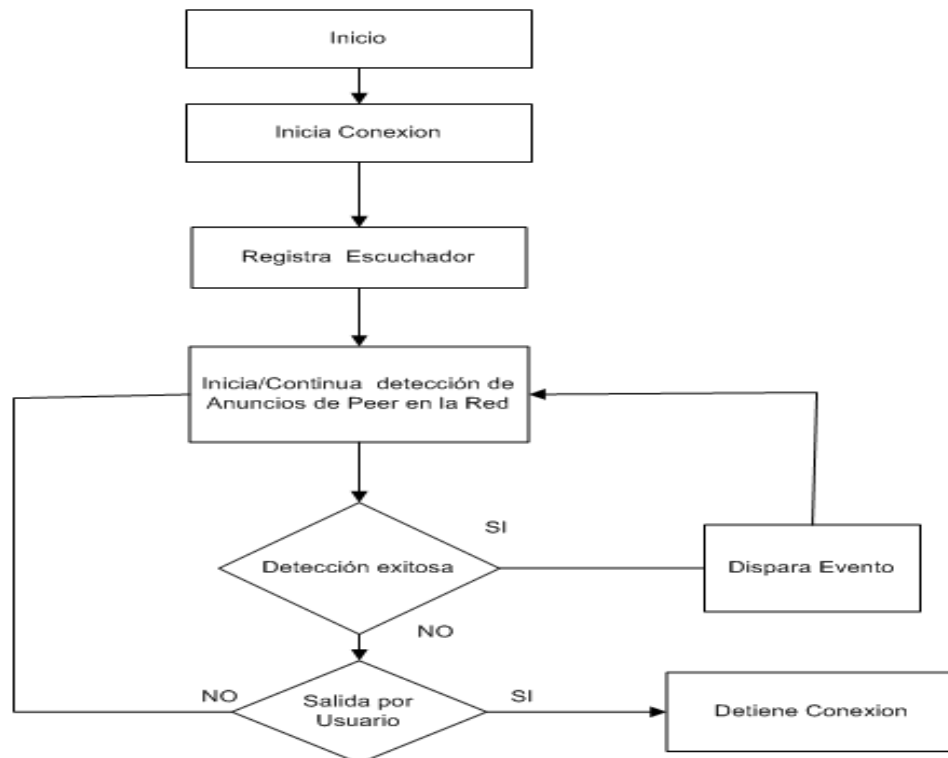
Nombre de la Capacidad: Descubrimiento.

Propósito Descripción: Capacidad de detectar e identificar cada peer que se conecta a la red en todo momento para una aplicación P2P desarrollada en JXTA. Esta capacidad puede verse como un perfil independiente, y reutilizable fácilmente. Un perfil base o nucleó de una plataforma P2P.

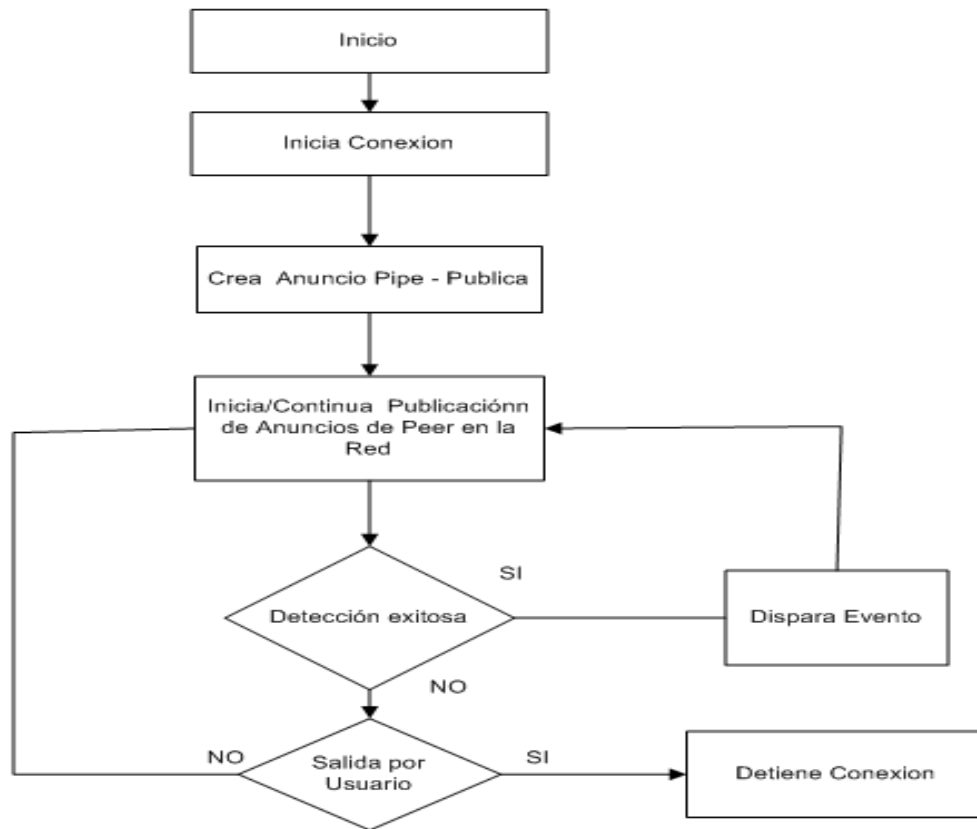


Lógica Descripción: Para una aplicación P2P esta capacidad aparecerá en cualquier nodo que desee descubrir uno o varios nodos en la red. Esta capacidad está dividida en dos estereotipos principales anteriormente descritos un Publicador de Anuncios, y un Escuchador de Anuncios. Ambos actúan continuamente publicando y descubriendo anuncios en la red para así poder identificarse. Como podemos observar en el diagrama de flujos de esta capacidad inicia con el proceso general llamado

1. Registra Escuchador: registra un escuchador para el evento que será llamado siempre que un anuncio de respuesta sea recibido de un peer remoto por el servicio de descubrimiento.
2. Inicia detección de Anuncios: Iniciamos la detección de peers este casi siempre es un ciclo infinito que provee una exhaustiva búsqueda. Una instancia de esta provee un "Mejor esfuerzo" de búsqueda la cual proveerá una selección de anuncios que cumplen el criterio de búsqueda. La selección de anuncios retornados pueden ser aleatorios o predecibles dependiendo de la configuración actual de la red y no un comportamiento particular. El alcance de los peers retornados podría limitarse al especificarse un atributo y sus valores.
3. Dispara Evento: Siempre que se detecte un anuncio de respuesta este evento se dispara el cual puede ser configurado para realizar cualquier acción necesaria con la información del peer remoto. El ciclo continua hasta que el usuario del peer desee desconcertarse o cambiar de grupo.



### Diagrama de Flujos Publicador de Anuncios



4. Crea anuncio de Pipe: Se crea un anuncio de pipe el cual tendrá todas las credenciales de identificación de este peer que serán utilizadas para una futura comunicación entre peers.
5. Inicia Publicación de Anuncios: Iniciamos la publicación de peers este casi siempre es un ciclo infinito cuando un anuncio es publicado, este es almacenado, e indexado en el cache local de los peers. Los índices de anuncios son también compartidos con otros peers inmediatamente, pero pueden ser actualizados como parte periódica de un proceso. El servicio de descubrimiento de JXTA publica e actualiza índices cada 30 segundos.
6. Dispara Evento: Siempre que se detecte un anuncio de respuesta este evento se dispara el cual puede ser configurado para realizar cualquier acción necesaria con la información del peer remoto. El ciclo continua hasta que el usuario del peer desee desconcertarse o cambiar de grupo allá una excepción externa que lo obligue a salir de la red al peer.

Rol Composición: Como ya habíamos explicado anteriormente en este servicio se detecta dos roles importantes los cuales son: Un Escuchador de Anuncios y un Publicador de Anuncios.

Capacidades Miembro: Podemos decir que todas las capacidades de este servicio serán parte importante de cualquier aplicación P2P no queremos que formen parte de la lógica del

negocio si no hagan parte de funcionalidades necesarias pero no centrales a su lógica.

Requerimiento Aseguramiento de la Calidad: La capacidad de descubrimiento deberá tener varias características de calidad importantes ya que esta formara la base de aplicaciones mas grandes entre las cuales son : fiabilidad, disponibilidad, reutilización, etc.

#### 4.3.4 PERFIL SERVICIOS DE PLATAFORMA

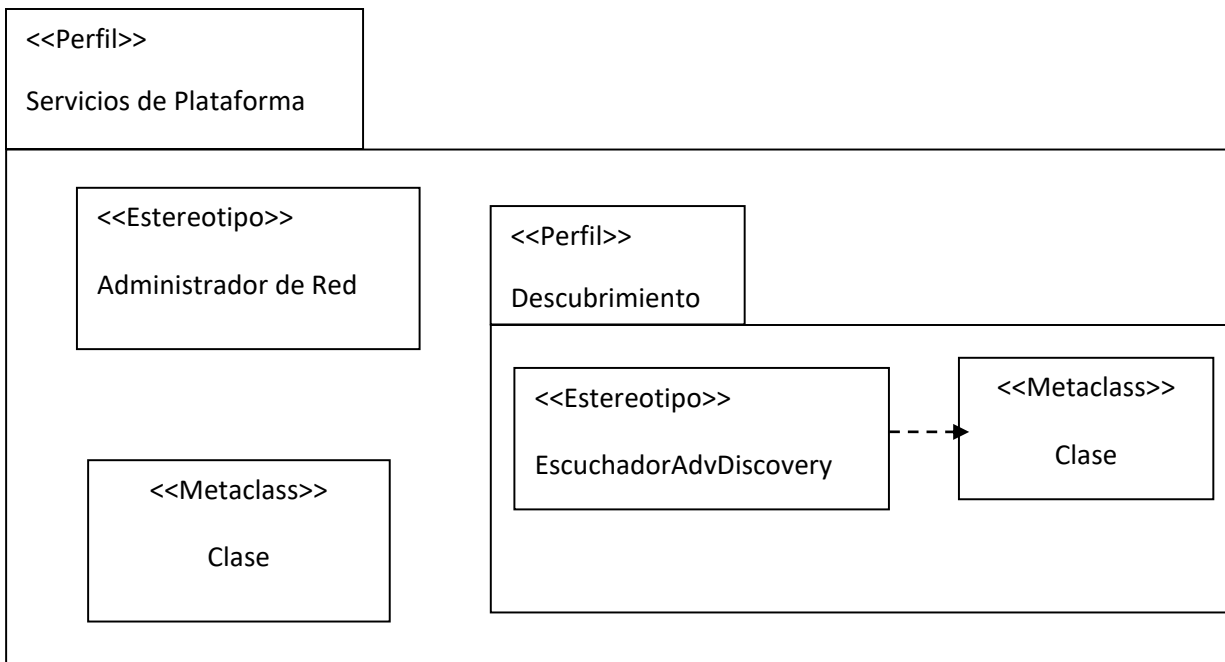


Figura 26 Perfil Servicios de Plataforma

La figura 26 representa la composición del perfil de servicios de plataforma el cual está formado por el perfil de descubrimiento y sus estereotipos

#### Especificación Formal de Estereotipos Perfil Servicios de Plataforma

Nombre	Administrador de Red
Clase Base	Class
Descripción	Clase que va administrar todo lo relacionado con la configuración e iniciación de la plataforma, va ofrecer métodos como inicio y detención de plataforma.
Restricciones	
Valores Etiquetados	Ninguno

#### 4.3.5 PERFIL DESCUBRIMIENTO

El perfil del servicio de Descubrimiento puede verse como una clase que realiza dos funciones o procedimientos necesarios en un entorno P2P:

- Obtiene los anuncios publicados por los demás nodos o peers en la Red.
- Publica sus propios anuncios a los demás nodos conocidos.

Nombre	EscuchadorAdvDiscovery
Clase Base	Class
Descripción	Clase que representa a la lógica que se encarga de escuchar o de obtener los anuncios publicados por uno o varios peers remotos. Y de publicar sus propios recursos los cuales pueden ser identificaciones, grupos IDs, archivos, etc.
Restricciones	
Valores Etiquetados	Ninguno
Icono	

#### 4.3.6 DIAGRAMA DE CLASES - PERFIL DE SERVICIOS DE PLATAFORMA

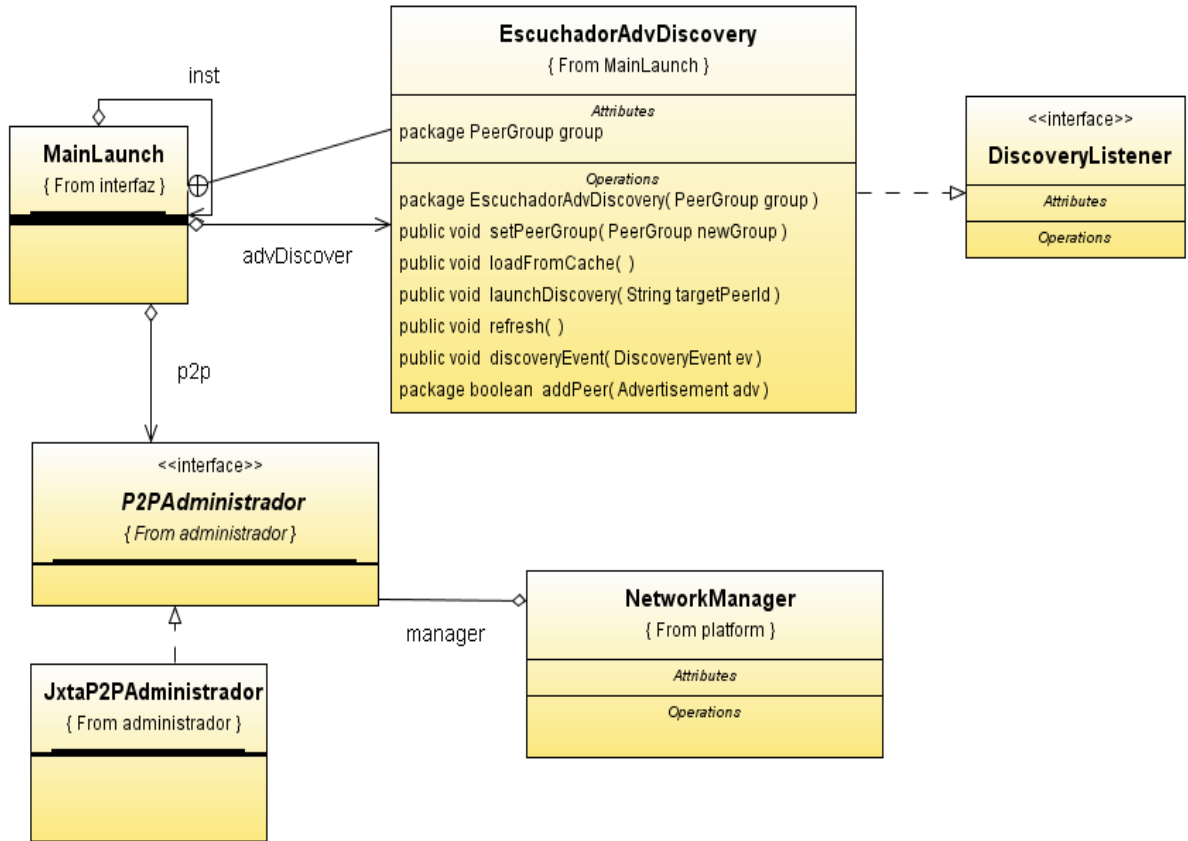
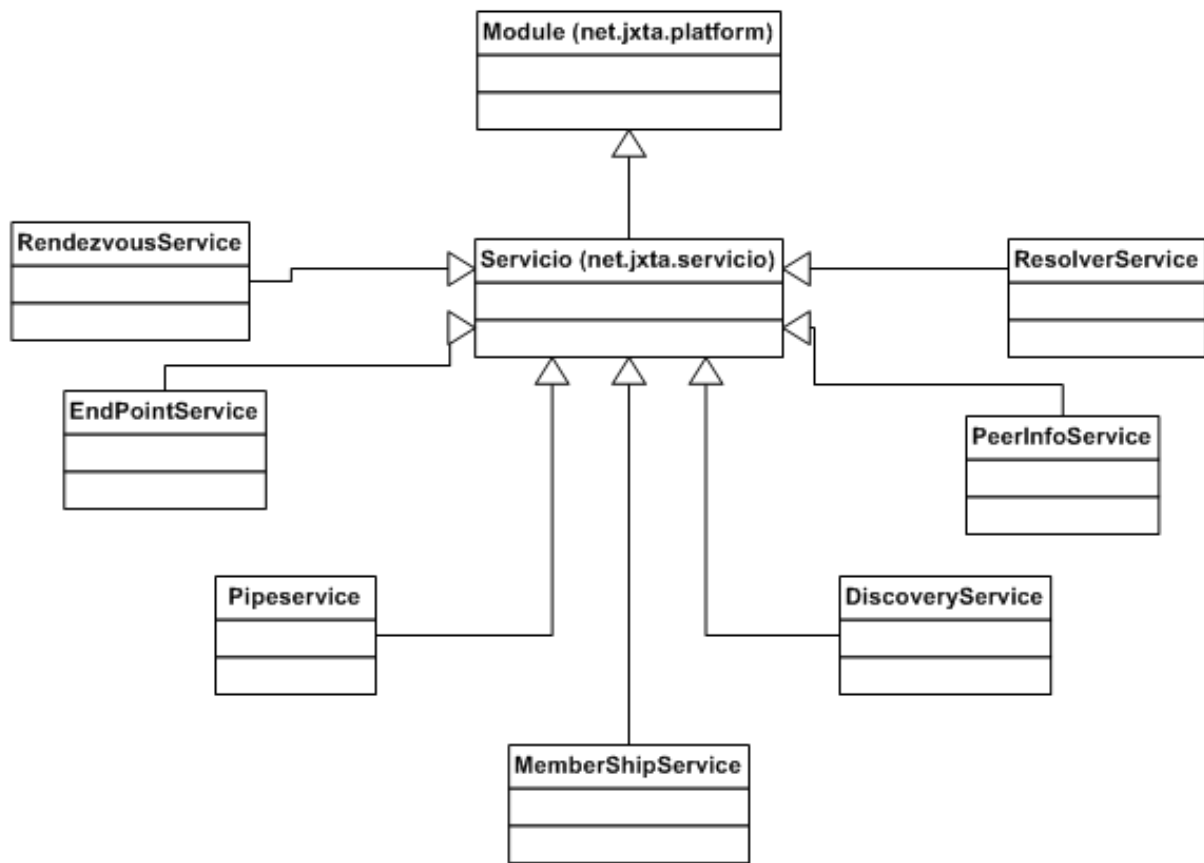


Figura 27 Diagrama de Clases Perfil Servicios de Plataforma

La figura 27 representa las clases que implementan el servicio de Plataforma, está compuesto por la clase **NetworkManager** que es la encargada de gestionar los servicios de plataforma tales como inicialización, configuraciones de peers y es administrada por medio de la interfaz **P2PAdministrador**, el servicio de descubrimiento que hace parte de este perfil es implementado con la interfaz **DiscoveryListener** la cual es utilizada por la clase **EscuchadorAdvDiscovery**.

#### 4.3.7 METAMODELO DE SERVICIOS DE AFILIACIÓN O SERVICIOS DE MEMBRESÍA



**Figura 28 Metamodelo de Servicios de Afiliación o Servicios de Membresía**

La figura 28 representa las clases que implementan a bajo nivel el servicio de membresía en una aplicación P2P, prestan funcionalidades de creación, detección y unión de grupos, podemos observar entonces que este servicio utiliza a su vez servicios como descubrimiento el cual es el encargado de detectar los grupos. RendezvousService es el encargado de propagar recursos a varios peers. PeerInfoService el cual presta información específica de un peer. Pipeservice presta los canales de comunicación entre peers y ResolverService es el encargado de detectar el emisor de un mensaje.

#### 4.3.8 ESTRUCTURA DEL PERFIL - SERVICIO DE AFILIACIÓN

Estructura del Perfil - Servicio de Afiliación
Nombre del Servicio: Perfil Afiliación.
Descripción Propósito: El servicio de afiliación gestionará varias funcionalidades que son necesarias para la implementación inicial de una aplicación P2P. Este servicio proveerá una interfaz amigable para administrar la creación, eliminación de grupos, deberá proveer credenciales de identificación a cada peer que le permitirán unirse a cualquier grupo, además será el encargado de descubrir todos los peers que estén en la red o podrá preguntar por alguno en especial. Se encargara de publicar sus propios recursos necesarios para poder comunicarse con los demás peers. Como estamos en un contexto P2P cada peer deberá poseer un servicio como este. Este servicio podrá ser extendido para hacer parte funcional en cualquier tipo de aplicación P2P en nuestro caso será útil para la construcción de una agenda para un grupo de trabajo que necesite comunicarse todos los días, tener una agenda de tareas pendientes y que tiene una jerarquía de trabajo y que necesita entregar avances cada semana. En resumen podemos decir que este perfil está compuesto por varios perfiles base definidos anteriormente los cuales son: Descubrimiento, Membresía, Acceso, Inicio y Configuración de plataforma.
Modelo del Servicio: Este servicio es clasificado como una utilidad esencial para cualquier aplicación P2P. El cual podrá ser extendido para prestar todas sus funcionalidades anteriormente descritas.
Requerimientos Calidad del Servicio: Este servicios deberá estar disponible en cualquier momento ya que sin el ningún peer podrá ingresar a la red. A demás deberá trabajar con sub – servicios que constantemente deberán estar funcionando. La seguridad también es una característica importante ya que este servicio otorgara credenciales de identificación para cualquier peer y validara el acceso a cualquier peer por eso es necesario que sea muy fiable y seguro. El desempeño también debe ser un punto crítico en su funcionamiento ya que este servicio formara la base de cualquier aplicación P2P. Su comunicación será realizada por protocolos base de JXTA anteriormente descritos y por consiguiente seguirá con sus mismos requisitos.

Capacidades:

- Inicio-configuración Plataforma: Proceso relacionado con la configuración de una plataforma P2P. En donde se puede fijar que tipo de nodo va ser en la Red, se registraran datos de identificación como el nombre, clave de acceso, IP, nombre PC, etc., se inicia la plataforma es decir se realizan todas las operaciones de configuración e inicialización necesarias.
- Descubrimiento: Capacidad de detectar e identificar cada peer que se conecta a la red en todo momento.
- Gestión de Grupos: Capacidad de crear, eliminar, unión a grupos de peers.
- Validador de Acceso: Capacidad de validar positivamente o negativamente el acceso a recursos, métodos o funciones a un peer dependiendo de su identificación y permisos.

Palabras clave:

- Modelos de utilidad del Servicio
- Disponibilidad del Servicio
- Agenda Conjunta
- Credenciales de Administración
- Protocolos de Comunicación de JXTA.

Versión: Versión inicial para todas las capacidades. 1.0

Estatus: El servicio se encuentra en etapa de diseño. Se estima dos semanas de desarrollo e implementación.



#### 4.3.9 ESPECIFICACIÓN FORMAL CAPACIDADES PERFIL AFILIACIÓN

Nombre de la Capacidad: Gestión de Grupos o Membresía
Propósito Descripción: Capacidad de creación, unión y descubrimiento de grupos en un entorno P2P.  Esta capacidad esta predefinida como un servicio base en la creación de aplicaciones en un Entorno P2P. La cual puede verse como el perfil base de membresía definido anteriormente.
Lógica Descripción: La Membresía es la funcionalidad principal del perfil de afiliación o gestión de grupos. Este proceso inicia cuando un nodo ingresa a la red y necesita una identificación única la cual va ser útil para que el sea identificado. Posteriormente si este nodo desea pertenecer a un grupo deberá ofrecer métodos o funciones que le ayuden a encontrar el grupo deseado y operaciones para unirse y dejarlo cuando quiera. Para este último paso hay varios procedimientos importantes como el de autenticación y acceso estos procedimientos están definidos en varios su perfiles descritos anteriormente.  Otra función importante de este principios es la detección de grupos en la Red, este monitoreo deberá ser constantemente actualizado para que le nodo observe que grupos existen en la red en ese momento. A demás este perfil deberá se relacionara con el perfil de Descubrimiento ya que este deberá saber en grupo actualmente estamos y por lo configurar el rango de alcance.
Entradas – Salidas: El nodo en un grupo actual y su correspondiente salida seria el nodo en un nuevo grupo con sus nodos en línea.
Rol Composición: Un gestor de Grupos.
Capacidades Miembro: Detección de grupos, gestión de grupos.
Requerimiento Aseguramiento de la Calidad: La capacidad de gestionar grupos deberá tener características de calidad importantes ya que esta formara la base de aplicaciones más grandes entre las cuales son: fiabilidad, disponibilidad, reutilización, etc.

Las capacidades de validación de acceso y administrador de credenciales hacen parte de los perfiles base que son: el servicio de membresía, el servicio de acceso. Estos perfiles serán necesarios para la construcción de un Perfil más general que estará compuesto por varias funcionalidades en nuestro caso estos perfiles base formaran el perfil de Afiliación el cual es el encargado de crear grupos, unir nodos a grupos, descubrir los grupos en la red, descubrir los nodos dependiendo a grupo que se pertenece actualmente.

Un gestor de grupos puede verse como un modulo independiente, reutilizable y que será fácil de usar a la hora de construir una aplicación P2P. Este modulo es una aplicación de Perfiles que cumplen funcionalidades de más bajo nivel pero muy esenciales.

#### 4.3.10 PERFIL SERVICIOS DE AFILIACIÓN Y GESTOR DE GRUPOS

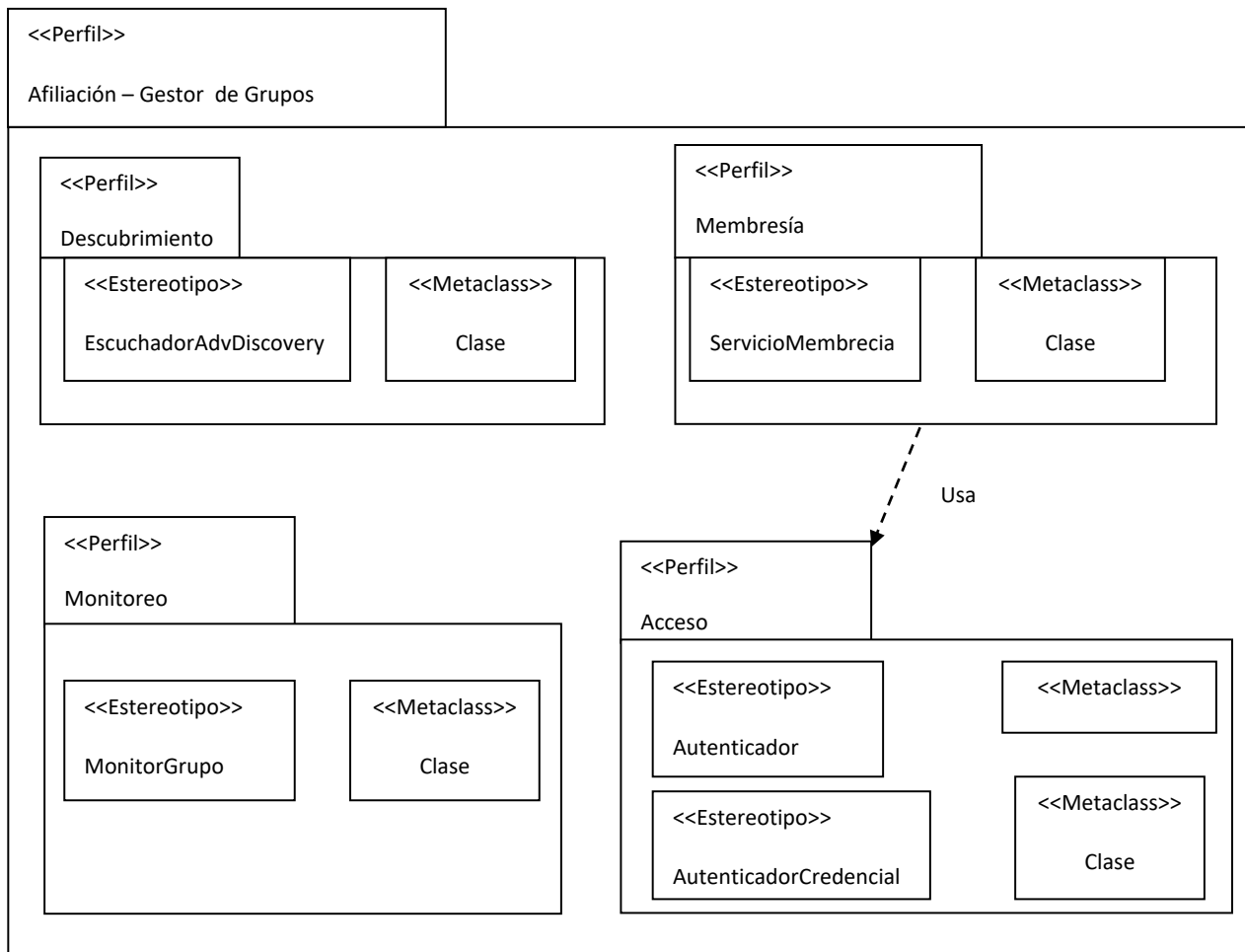


Figura 29 Perfil Servicios de Afiliación y Gestor de Grupos

El perfil de Afiliación está compuesto por cuatro sub perfiles que trabajan en conjunto para prestar varios servicios útiles para cualquier tipo de Aplicación P2P. Estos perfiles se extenderán a clases que implementarán sus funcionalidades correspondientes que son especificadas a continuación, ver figura 29.

#### 4.3.11 ESPECIFICACIÓN FORMAL DE ESTEREOTIPOS PERFIL AFILIACIÓN Y GESTOR DE GRUPOS

Nombre	ServicioMembresia
Clase Base	Class
Descripción	Clase que permite establecer una identidad dentro de un grupo de nodos. Estas identidades son usadas por los servicios y aplicaciones para validar si el nodo tiene los suficientes permisos para un recurso dado.
Restricciones	
Valores Etiquetados	Ninguno
Icono	

Nombre	Autenticador
Clase Base	Class
Descripción	Un autenticador es un objeto único generado para cada método de autenticación. La única operación común para completar este objeto es la validación si está listo para unirse, la cual provee información acerca de si o no usted a completado correctamente el autenticador.
Restricciones	
Valores Etiquetados	Ninguno
Icono	

Nombre	AutenticadorCredencial
Clase Base	Class
Descripción	Clase que es usada por el servicio de membresía como base para la aplicación de grupos de peers y miembros. Esta provee dos piezas o importantes de información: el método de autenticación inicialmente requerido, información de identidad la cual será dada al método de autenticación.
Restricciones	
Valores Etiquetados	Ninguno
Icono	

Nombre	MonitorGrupo
Clase Base	Class
Descripción	Clase responsable de la propagación de mensajes dentro de un grupo de nodos. Y detección de nodos en el grupo actual.
Restricciones	
Valores Etiquetados	Ninguno
Icono	

### 4.3.12 DIAGRAMA DE CLASES PERFIL AFILIACIÓN

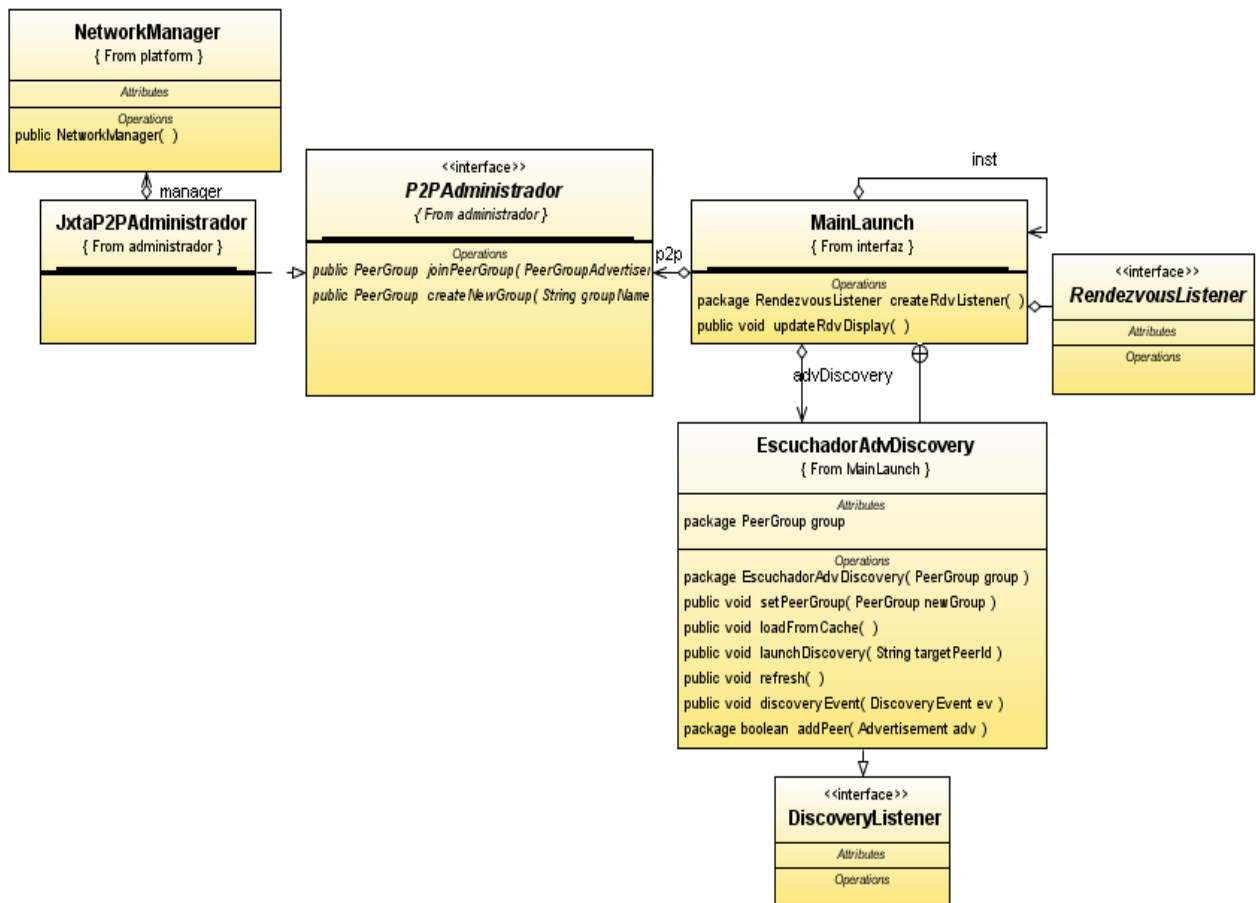


Figura 30 Diagrama de Clases Perfil Afiliación

La figura 30 representa las clases que implementan el servicio de Afiliación utiliza la interfaz DiscoveryListener la cual es la encargada de detectar los grupos en la red, esta es utilizada por la clase EscuchadorAdvDiscovery la cual presta funcionalidades de cambio, detección de grupos, la clase JxtaP2PAdministrador es utilizada por la interfaz P2P para obtener funcionalidades de Administración de la plataforma P2P.

#### 4.3.13 PERFIL DE COMUNICACIÓN

Estructura del Perfil - Perfil de Comunicación
Nombre del Servicio: Perfil de Comunicación
Propósito Descripción: Enviar cualquier tipo de mensajes a cualquier nodo que se detecte en la red. Los mensajes enviados pueden ser archivos o simplemente una cadena o cualquier otro recurso.
Lógica Descripción: Este servicio está compuesto por dos perfiles base que se encargan del envío y recepción de mensajes . Estos perfiles se diferencian por el tipo de mensajes que pueden enviar. El perfil de mensajes solo envía mensajes como su nombre lo dice y el perfil de envío de Archivos el cual esta descrito a continuación.
Modelo del Servicio: Este servicio es clasificado como una utilidad esencial para cualquier aplicación P2P.
Requerimientos Calidad del Servicio: Este servicios deberá estar disponible en cualquier momento ya que sin el ningún peer podrá ingresar a la red. A demás deberá trabajar con sub – servicios que constantemente deberán estar funcionando. La seguridad también es una característica importante ya que este servicio otorgara credenciales de identificación para cualquier peer y validara el acceso a cualquier peer por eso es necesario que sea muy fiable y seguro. El desempeño también debe ser un punto crítico en su funcionamiento ya que este servicio formara la base de cualquier aplicación P2P. Su comunicación será realizada por protocolos base de JXTA anteriormente descritos y por consiguiente seguirá con sus mismos requisitos.
Capacidades: <ul style="list-style-type: none"><li>• Envío y recepción de Mensajes Estándares y con Archivos</li><li>• Intercambio de Archivos</li><li>• Detección de Nuevos Recursos en la Red</li><li>• Envío se hace a todos los miembros de un grupo.</li><li>• Partición de grandes archivos en bloques.</li></ul>
Palabras clave: <ul style="list-style-type: none"><li>• Modelos de utilidad del Servicio</li><li>• Disponibilidad del Servicio</li><li>• Agenda Conjunta</li><li>• Credenciales de Administración</li><li>• Protocolos de Comunicación de JXTA.</li></ul>

Este servicio está compuesto por dos Perfiles que gestionas el envío de mensajes, uno es el Perfil de Envío de Mensajes Estándares y el otro es el Perfil de Envío de Mensajes de tipo Archivo.

#### 4.3.14 SERVICIO DE ENVIÓ DE MENSAJES ESTÁNDAR

Estructura del Perfil - Servicio de Envío de Mensajes Estándar
Nombre del Servicio: Servicio de Envío de Mensajes Estándar
Descripción Propósito: Este servicio es el encargado del envío de mensajes estándares entre dos nodos en línea. Este servicio establece un canal de comunicación para el envío y recepción de mensajes.
Modelo del Servicio: Este servicio es clasificado como una utilidad esencial para cualquier aplicación P2P. El cual podrá ser extendido para prestar todas sus funcionalidades anteriormente descritas.
Requerimientos Calidad del Servicio: Este servicios deberá estar disponible en cualquier momento ya que sin el ningún peer podrá ingresar a la red. A demás deberá trabajar con sub – servicios que constantemente deberán estar funcionando. La seguridad también es una característica importante ya que este servicio otorgara credenciales de identificación para cualquier peer y validara el acceso a cualquier peer por eso es necesario que sea muy fiable y seguro. El desempeño también debe ser un punto crítico en su funcionamiento ya que este servicio formara la base de cualquier aplicación P2P. Su comunicación será realizada por protocolos base de JXTA anteriormente descritos y por consiguiente seguirá con sus mismos requisitos.
<p>Capacidades:</p> <ul style="list-style-type: none"> <li>• Envío y Recepción de Mensajes.</li> <li>• Detección de Peticiones de Chat.</li> <li>• Envío y detección de Respuestas de Peticiones.</li> </ul>
<p>Palabras clave:</p> <ul style="list-style-type: none"> <li>• Modelos de utilidad del Servicio</li> <li>• Disponibilidad del Servicio</li> <li>• Agenda Conjunta</li> <li>• Credenciales de Administración</li> <li>• Protocolos de Comunicación de JXTA.</li> </ul>
Versión: Versión estable. 2.0
Estatus: El servicio se encuentra en listo para ser reutilizado para cualquier Aplicación P2P.

#### 4.3.15 ESPECIFICACIÓN FORMAL CAPACIDADES PERFIL ENVIÓ DE MENSAJES ESTÁNDAR

Nombre de la Capacidad: Proceso General de Envió de Archivos Estándar.
Propósito Descripción: Este servicio y sus capacidades son usados para enviar a cualquier usuario de la aplicación P2P una petición de chat y por consiguiente un servicio de chat.
Lógica Descripción: El servicio inicia cuando un usuario envía una petición de chat a otro usuario destino. Este usuario detecta la petición de chat y la responde si la respuesta es positiva se registra la información importante de los dos nodos en cada nodo para construir un canal de comunicación entre ellos. Cuando el canal se ha quedado listo cualquier nodo puede enviar un mensaje. Básicamente el proceso general es:
<pre> sequenceDiagram     participant N1 as Nodo 1     participant N2 as Nodo 2     N1-&gt;&gt;N2: 1 - Envió de Petición de Chat     N2-&gt;&gt;N2: 1.1     N2-&gt;&gt;N1: 2 - Envió de Respuesta Petición de Chat     N1-&gt;&gt;N1: 2.1     N1-&gt;&gt;N2: 3 - Envió de mensaje     </pre>
Paso 1.1: Si la respuesta es positiva. Este nodo registra la Información del nodo que solicito la petición de Chat e inicia la creación del canal de comunicación y por ultimo envía un mensaje de respuesta con la información de identificación.
Paso 2.1: Si la respuesta es positiva. Este nodo registra la Información del nodo que respondió la petición de Chat y termina con la creación del canal de comunicación con la información del nodo destino y por ultimo envía un mensaje de respuesta con la información de identificación.
Entradas – Salidas:
Rol Composición: Un Emisor y Receptor.
Capacidades Miembro: envío, recepción de mensajes.
Requerimiento Aseguramiento de la Calidad:



#### 4.3.16 ESPECIFICACIÓN FORMAL SERVICIO DE ENVIÓ DE MENSAJES TIPO ARCHIVOS

Estructura del Perfil - Servicio de Envió de Mensajes tipo Archivos
Nombre del Servicio: Envió de Mensajes tipo Archivos
Descripción Propósito: Este servicio es el encargado de compartir recurso a través de los diferentes nodos en la red. Esta funcionalidad es una característica esencial en una red P2P ya que le permite a los diferentes nodos intercambiar, diferentes tipos de archivos los cuales pueden formar un papel importante en el negocio de una Aplicación. Debe ofrecer todos los mecanismos que habiliten el fácil funcionamiento y acoplamiento de esta funcionalidad a cualquier tipo de Aplicación que la necesite.
Modelo del Servicio: Este servicio es clasificado como una utilidad esencial para cualquier aplicación P2P. El cual podrá ser extendido para prestar todas sus funcionalidades anteriormente descritas.
Requerimientos Calidad del Servicio: Este servicios deberá estar disponible en cualquier momento ya que sin el ningún peer podrá ingresar a la red. A demás deberá trabajar con sub – servicios que constantemente deberán estar funcionando. La seguridad también es una característica importante ya que este servicio otorgara credenciales de identificación para cualquier peer y validara el acceso a cualquier peer por eso es necesario que sea muy fiable y seguro. El desempeño también debe ser un punto crítico en su funcionamiento ya que este servicio formara la base de cualquier aplicación P2P. Su comunicación será realizada por protocolos base de JXTA anteriormente descritos y por consiguiente seguirá con sus mismos requisitos.
Capacidades: <ul style="list-style-type: none"><li>• Intercambio de Archivos</li><li>• Detección de Nuevos Recursos en la Red</li><li>• El Envió se hace a todos los miembros de un grupo.</li><li>• Partición de grandes archivos en bloques.</li></ul>
Palabras clave: <ul style="list-style-type: none"><li>• Modelos de utilidad del Servicio</li><li>• Disponibilidad del Servicio</li><li>• Agenda Conjunta</li><li>• Credenciales de Administración</li><li>• Protocolos de Comunicación de JXTA.</li></ul>
Versión: Versión estable. 2.0

Estatus: El servicio se encuentra en listo para ser reutilizado para cualquier Aplicación P2P.

#### 4.3.17 ESPECIFICACIÓN FORMAL CAPACIDADES PERFIL ENVIÓ DE ARCHIVOS

Nombre de la Capacidad: Proceso General de Gestión de Recursos.

Propósito Descripción: Este servicio y sus capacidades son usados para enviar a todos los miembros de un grupo cualquier tipo de archivos.

Lógica Descripción: Este servicio y sus capacidades son usados para enviar a todos los miembros de un grupo cualquier tipo de archivos. Los grandes archivos son partidos en bloques, los bloques son puestos en los mensajes y enviados a través de la Red. Los peers que reciben los archivos re ensamblan el archivo de los bloques. Los mensajes con los bloques pueden llegar desordenados y algunos de ellos pueden ser borrados. Los nodos deben negociar entre ellos mismos para obtener cualquier bloque que se haya perdido de la transmisión original. Un nodo puede enviar y recibir múltiples archivos de una vez, de múltiples fuentes.

Este servicio usa canales de propagación de archivos hacia un grupo de nodos. Un mensaje se envía a través de un canal de salida de propagación que será recibido por todos los nodos de ese grupo que tengan creado un canal de entrada de propagación igual. Este servicio crea canales de entrada y salida de propagación usando anuncios conteniendo IDs de canales conocidos por el uso de estos IDs, este servicio puede inmediatamente comunicar con otro nodo igual en el grupo sin tener que usar el servicio de descubrimiento para encontrar el anuncio del pipe. Una cadena especificada por la aplicación es incluida como parte del IDs permitiendo así diferenciar entre aplicaciones P2P. Para enviar un archivo a todos los miembros de un grupo este servicio primero carga el archivo de datos dentro de un array de bytes en memoria. Entonces crea un mensaje de archivos conteniendo el primer bloque de datos y envía este a través del canal de propagación de salida. El siguiente reto es enviar los mensajes con los bloques restantes del archivo de datos en una proporción que pueda ser absorbida por el sistema. Si solo vamos dentro de un ciclo y enviamos los mensajes tan rápido como sea posible, la mayoría de estos serán borrados. Este servicio usa dos métodos para controlar la proporción de mensajes de salida el primero es un método llamado ACK, cuando recibimos un mensaje con archivo ACK reconociendo que el nodo ha recibido nuestro último mensaje, enviamos el siguiente. Como una copia de seguridad el segundo está en una proporción de salida cronometrada: si aquí no se recibe un mensaje ACK durante el periodo cronometrado, el servicio enviara el siguiente bloque. En la práctica esto significa que la proporción de salida es controlada por el archivo ACK, cuando dos de los nodos están sobre una red rápida tal como una LAN. Pero cuando todos los nodos son separados del remitente por una red lenta tal como cruzar un propagador HTTP, la proporción de salida será controlada por el tiempo de retraso. Este servicio le permite a una aplicación registrar un escuchador para el evento de este servicio, después de que cada bloque de un archivo es enviado, este evento será enviado a los

escuchadores. Una aplicación puede chequear estos eventos para mirar cuando un archivo de transmisión de salida está completo. Cuando este servicio recibe mensaje de Archivo este chequea para mirar si un archivo de entrada ha sido ya creado para manejar estos archivos de entrada. El mensaje de archivos con elementos de archivos clave es usado para mirar el pendenciero en nuestra lista. Si ninguno es encontrado se crea uno. El primer mensaje de archivo no es necesariamente el primer bloque del archivo de datos. Por cada nuevo mensaje de archivo que recibimos, respondemos con un mensaje ACK a través del canal de propagación, cualquier mensaje de archivo duplicado es ignorado. Estos mensajes de ACK son usados para ayudar al emisor del mensaje controlar la proporción de mensajes de salida. Cuando todos los bloques son recibidos el archivo es almacenado en el disco.

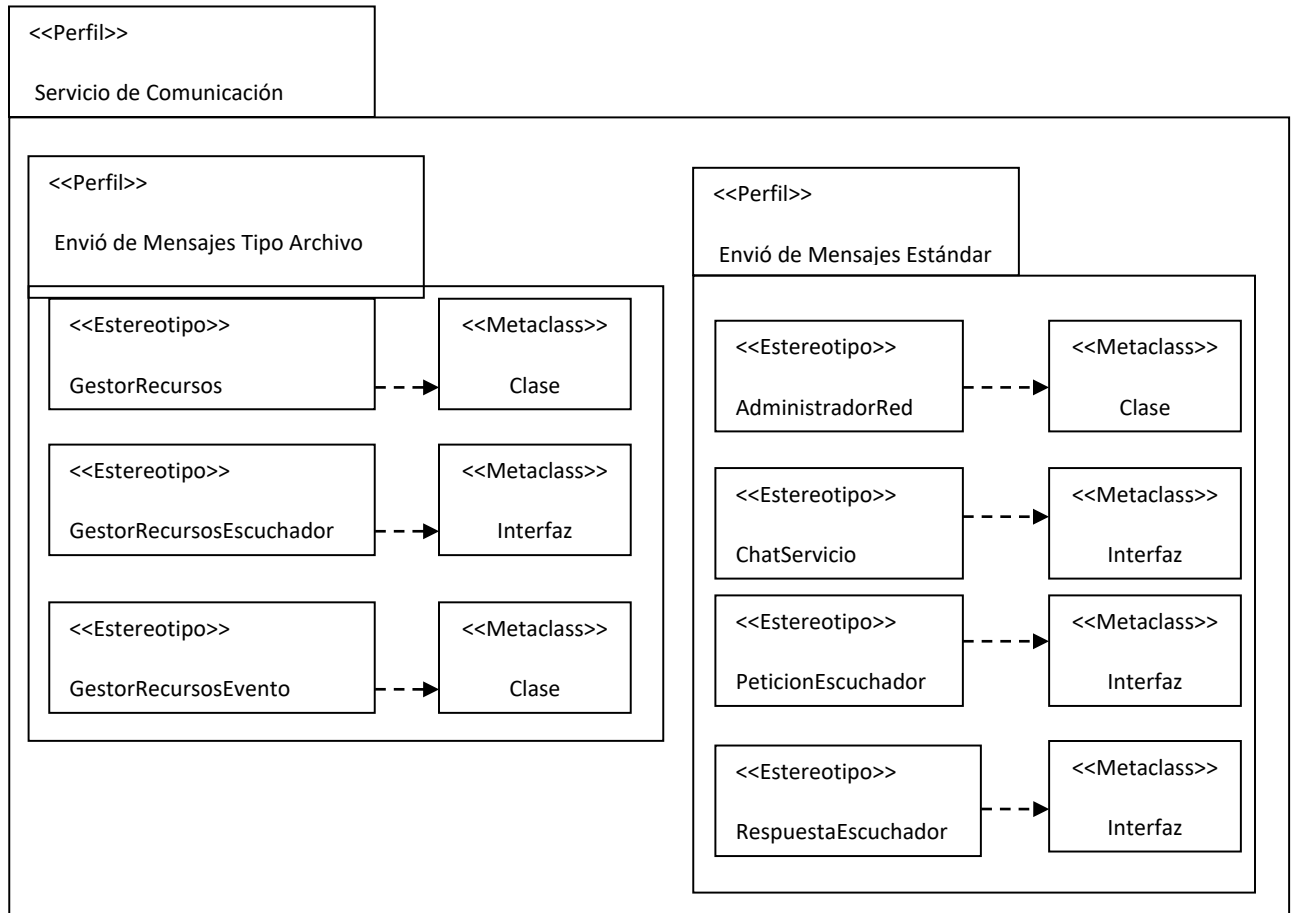
Entradas – Salidas: El nodo en un grupo actual y su correspondiente salida seria el nodo en un nuevo grupo con sus nodos en línea.

Rol Composición: Un Gestor de Recursos.

Capacidades Miembro: envío, recepción de mensajes de Archivos, Partición de bloques.

Requerimiento Aseguramiento de la Calidad: Este servicio tiene algunas desventajas en la fiabilidad ya que algunos bloques de datos pueden perderse en el proceso de envío.

## Perfil Servicios de Comunicación



**Figura 31 Perfil Servicios de Comunicación**

La figura 31 representa como el Perfil de Comunicación está conformado por los perfiles de Envío de Mensajes tipo Archivo y el perfil de envío de mensajes estándar, además cada perfil esta conformado por sus estereotipos encontrados.

## 4.4 ESPECIFICACIÓN FORMAL DE LOS ESTEREOTIPOS DE LOS PERFILES DEFINIDOS

### 4.4.1 ESTEREOTIPOS PERFIL ENVÍO DE MENSAJES ESTÁNDAR

Nombre	AdministradorRed
Clase Base	Class
Descripción	Clase que administra los métodos de principales de este servicio tales como envío de Petición, detección de Petición, envío de Mensajes y recepción de Mensajes.
Restricciones	
Valores Etiquetados	Ninguno
Icono	

Nombre	ChatServicio
Clase Base	Class
Descripción	Interfaz que sirve para instanciar el servicio de Mensajería para cualquier nodo en la Red.
Restricciones	
Valores Etiquetados	Ninguno
Icono	

Nombre	PeticionEscuchador
Clase Base	Class
Descripción	Escuchador que detecta todas las peticiones de Chat.
Restricciones	
Valores Etiquetados	Ninguno
cono	

Nombre	RespuestaEscuchador
Clase Base	Class
Descripción	Escuchador que detecta todas las Respuestas de Chat.
Restricciones	
Valores Etiquetados	Ninguno
Icono	

#### 4.4.2 ESTEREOTIPOS PERFIL ENVÍO DE MENSAJES TIPO ARCHIVO

Nombre	GestorRecursos
Clase Base	Class
Descripción	Clase que envía archivo de datos hacia todos los nodos en un grupo de (estos están escuchando por este). Recibe archivo de datos enviados por otros usuarios del mismo gestor de recursos.
Restricciones	
Valores Etiquetados	Ninguno
Icono	

Nombre	GestorRecursosEscuchador
Clase Base	Class
Descripción	Interfaz para escuchar eventos de archivos del Gestor de Recursos.
Restricciones	
Valores Etiquetados	Ninguno
Icono	

Nombre	GestorRecursosEvento
Clase Base	Class
Descripción	Evento para enviar hacia los escuchadores del Gestor de recursos, para notificar a estos del archivo nuevo en progreso.
Restricciones	
Valores Etiquetados	Ninguno
Icono	

#### 4.4.3 DIAGRAMA DE CLASES PERFIL ENVÍO DE MENSAJES ESTÁNDAR

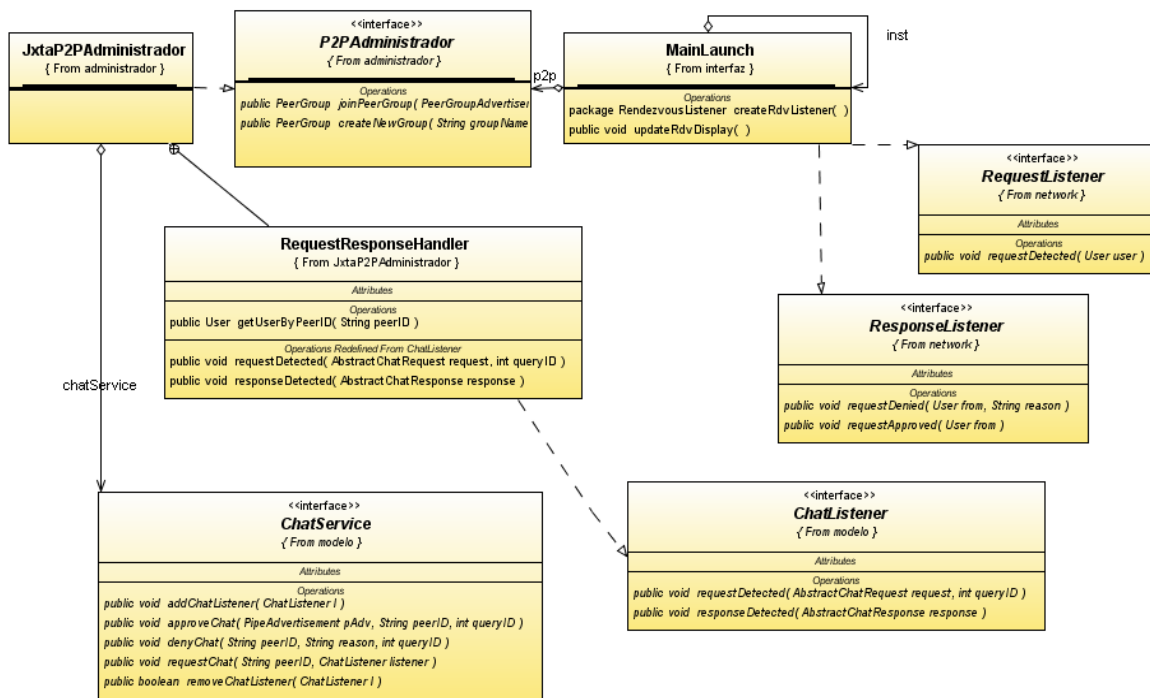


Figura 32 Diagrama de Clases Perfil Envío de Mensajes Estándar

La figura 32 representa las clases que implementan el perfil envío de mensajes estándar, este es implementado por las interfaces ChatListener el cual es el encargado de mandar y responder peticiones de chat, la interfaz Chatservice la cual presta funcionalidades de aprobar, denegar peticiones de chats, estas interfaces son utilizadas por la clase RequestResponseHandler, las interfaces RequestListener y ResponseListener son las encargadas de detectar los mensajes enviados entre dos peer al momento de un chat.

#### 4.4.4 DIAGRAMA DE CLASES PERFIL ENVÍO DE MENSAJES TIPO ARCHIVO

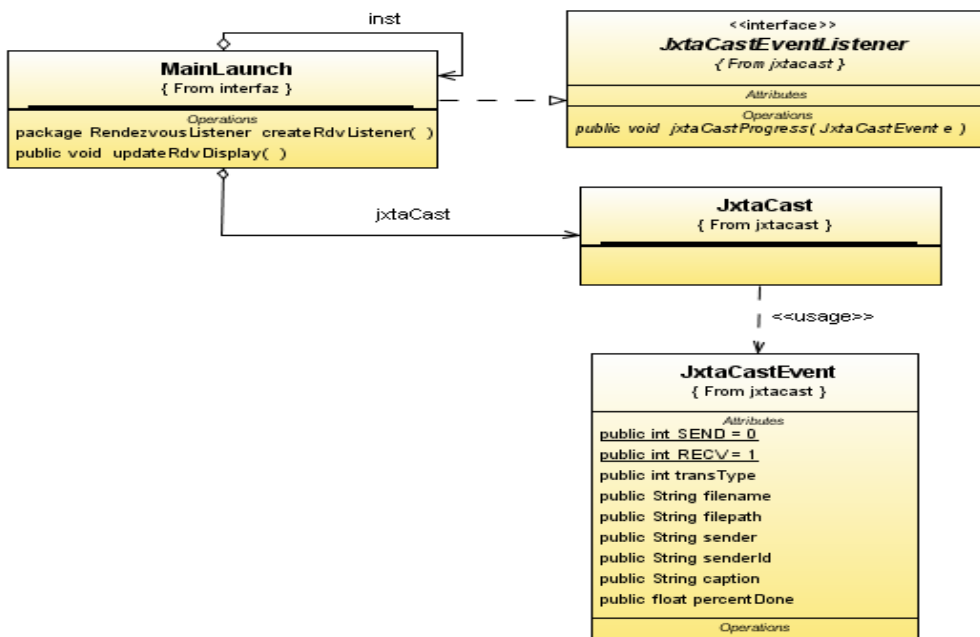


Figura 33 Diagrama de Clases Perfil Envío de Mensajes Tipo Archivo

La figura 33 representa las clases que implementan el perfil de envío de mensajes tipo archivo, esta funcionalidad desarrollada por la interfaz `JxtaCastEventListener` la cual presta la funcionalidad de detectar y procesar el envío y recepción de un archivo de cualquier tamaño en la red, es utilizada por medio de la clase `JxtaCast`.

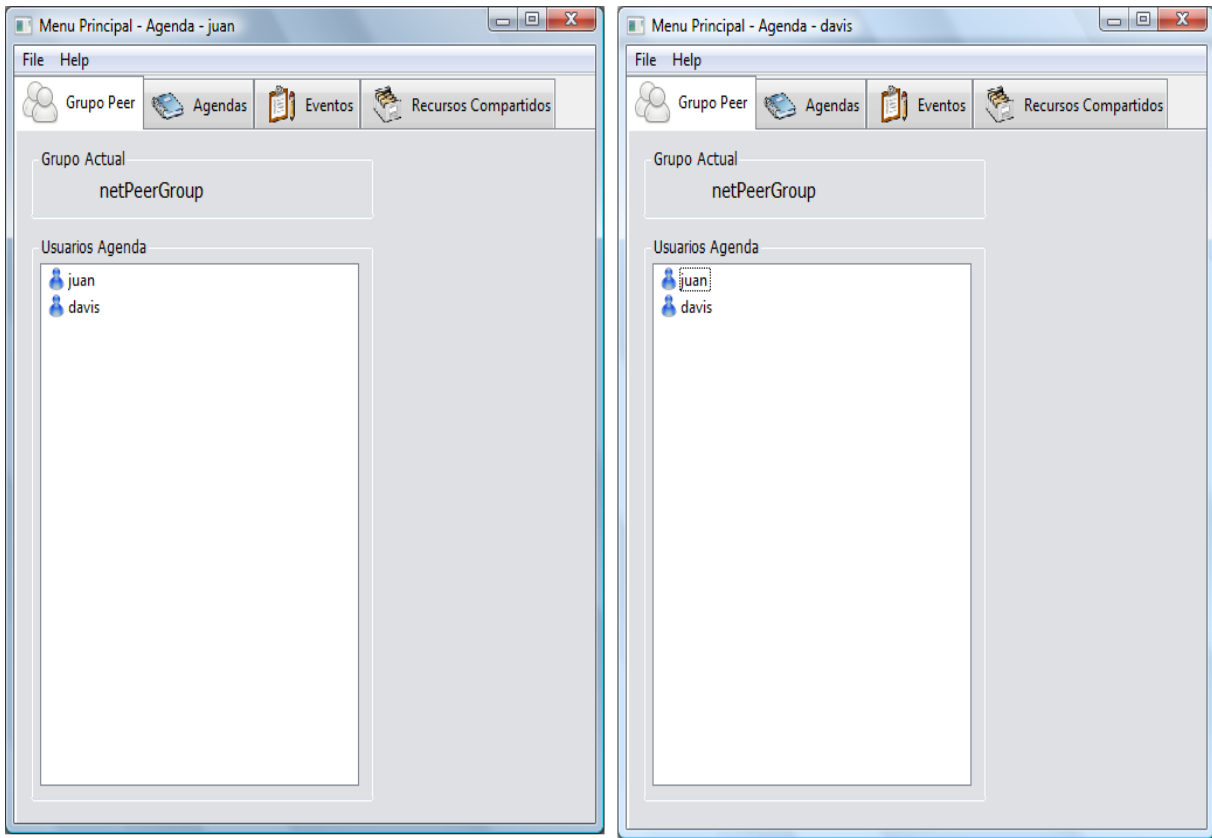


## CAPITULO 5 ESTRUCTURA GENERAL CASO DE ESTUDIO

El caso de estudio desarrollado es un prototipo de Agenda Colaborativa la cual tiene las siguientes funcionalidades y requerimientos:

### 5.1 DETECCIÓN Y MONITOREO DE USUARIOS AGENDA (PEERS)

Detectar usuarios que están conectados al aplicativo, en este caso cada usuario al registrarse por defecto se registra en el grupo NetPeerGroup.



**Figura 34 Monitoreo Usuarios Agenda Online**

Como podemos observar en la figura cada Agenda tiene su lista de usuarios conectados en el momento, esta lista se actualiza automáticamente detectando así cuando un usuario se desconecta de la aplicación, ver figura 34.

## 5.2 ADMINISTRADOR DE GRUPOS DE TRABAJO

Un administrador de grupos tiene las funcionalidades de creación, cambio y búsqueda de grupos.

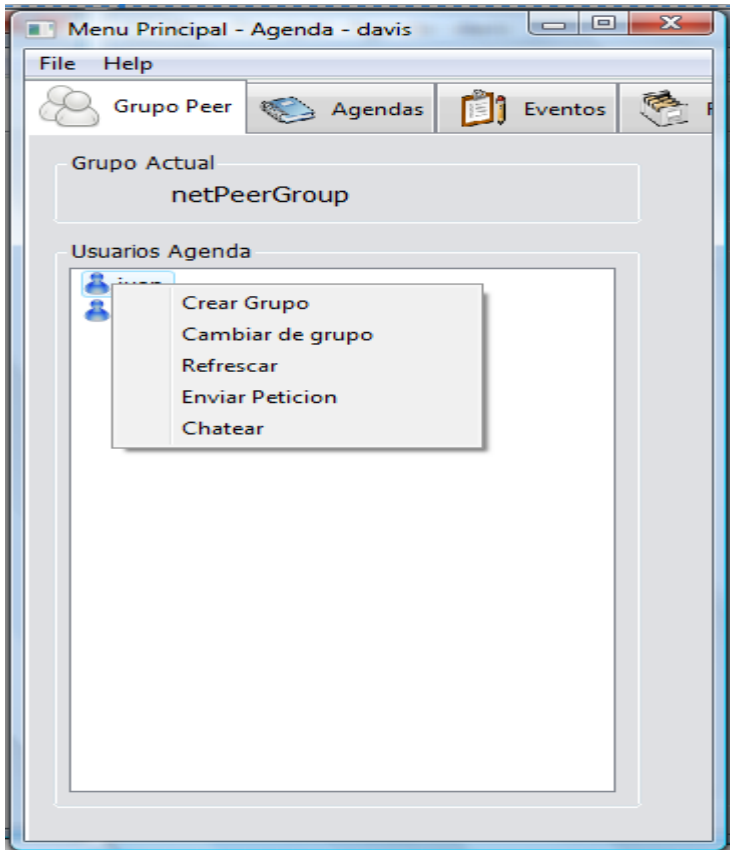


Figura 35 Selección de Crear Grupo

En la figura 35 podemos observar las funcionalidades del creación y cambio de grupo

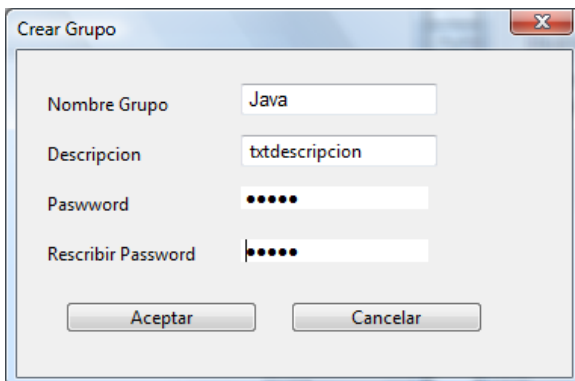
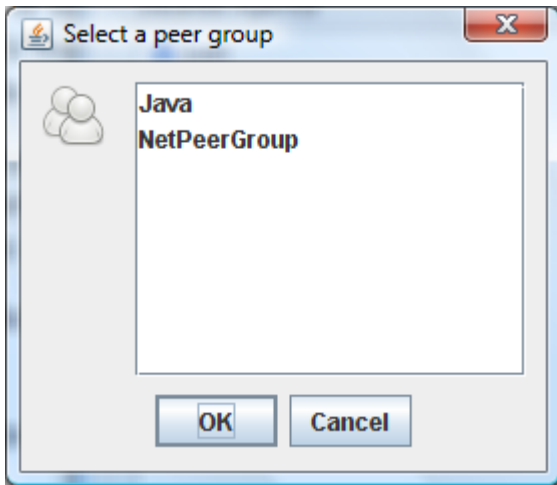


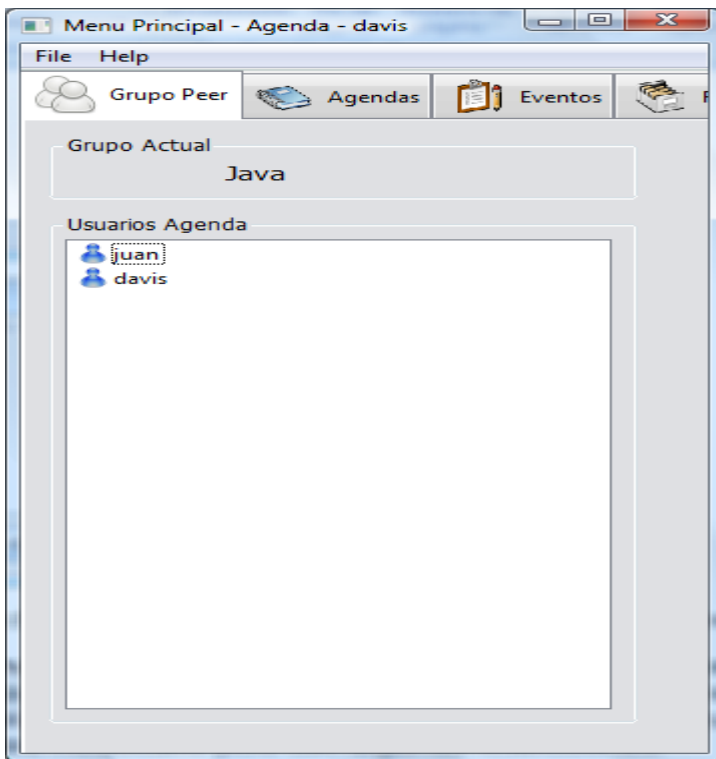
Figura 36 Creación de Grupo

En la figura 36 observamos la creación del grupo java



**Figura 37 Selección de Grupo a unirse**

En la figura 37 observamos los diferentes grupos disponibles en la red, podemos ver el grupo Java anteriormente creado.



**Figura 38 Usuario en el nuevo Grupo**

En la figura 38 el usuario Davis ya se ha unido al grupo Java anteriormente creado.

### 5.3 SERVICIO DE CHAT

Cada usuario puede chatear con otro usuario que este conectado enviando una petición de chat cuando esta es aceptada este usuario puede iniciar la conversación.

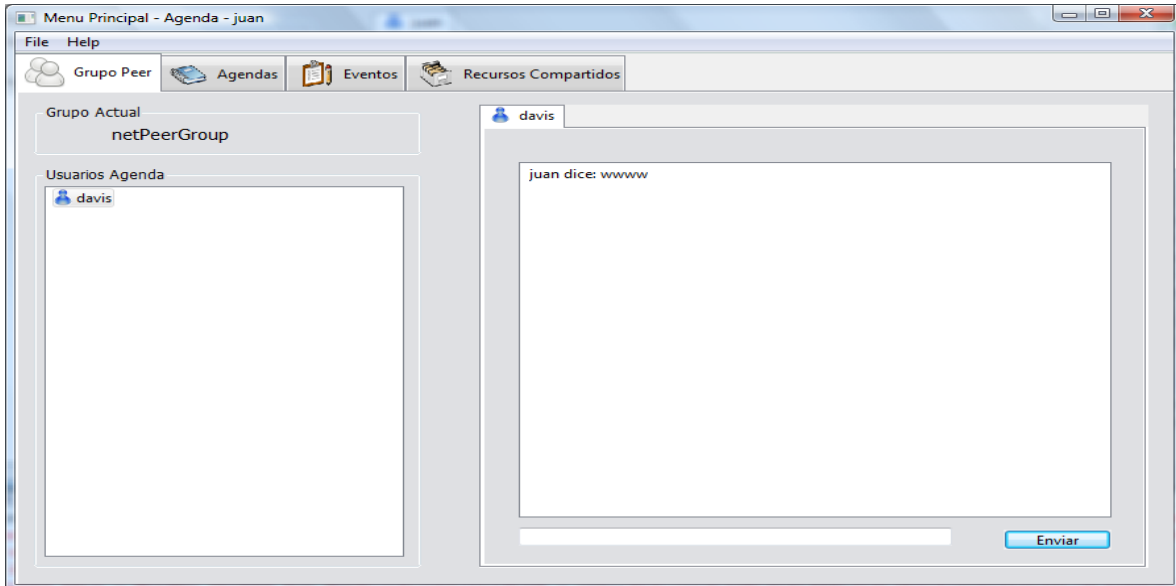


Figura 39 Usuarios Agenda

En la figura 39 observamos como el usuario Juan ha detectado al usuario Davis

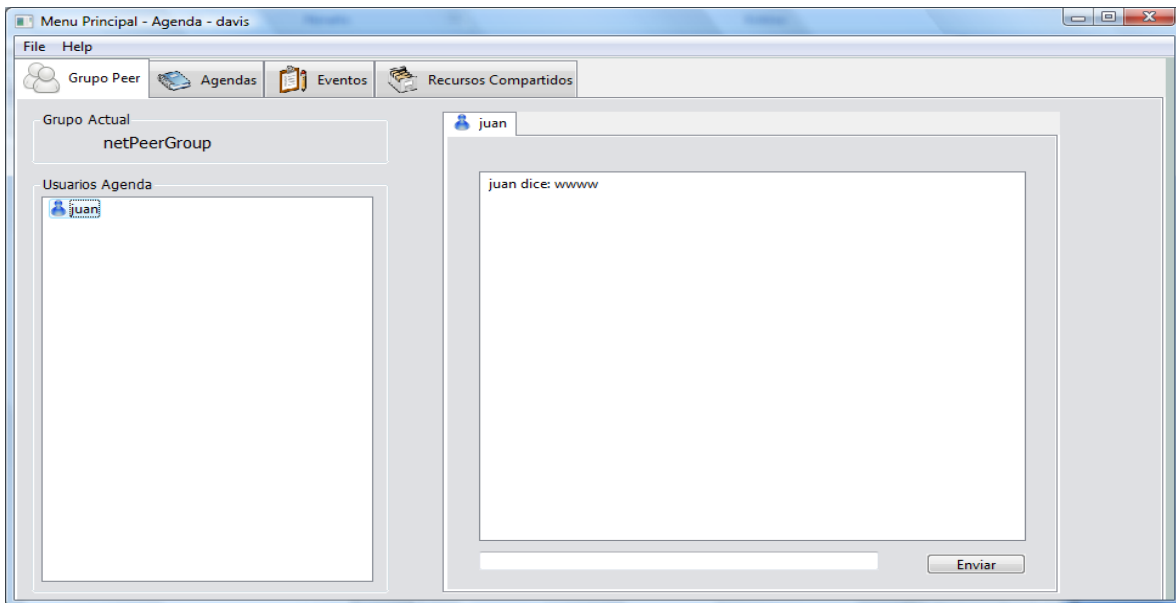


Figura 40 Chat Usuarios Agenda.

En la figura 40 observamos como el usuario Davis ha detectado al usuario Juan.

## 5.4 ADMINISTRADOR DE AGENDA COMPARTIDA

En esta agenda compartida se pueden registrar y eliminar las actividades que el equipo deberá cumplir o desarrollar en una fecha límite. Cada actividad es registrada por el usuario con el rol de director del grupo y cada participante podrá modificarlo pero con algunas restricciones.

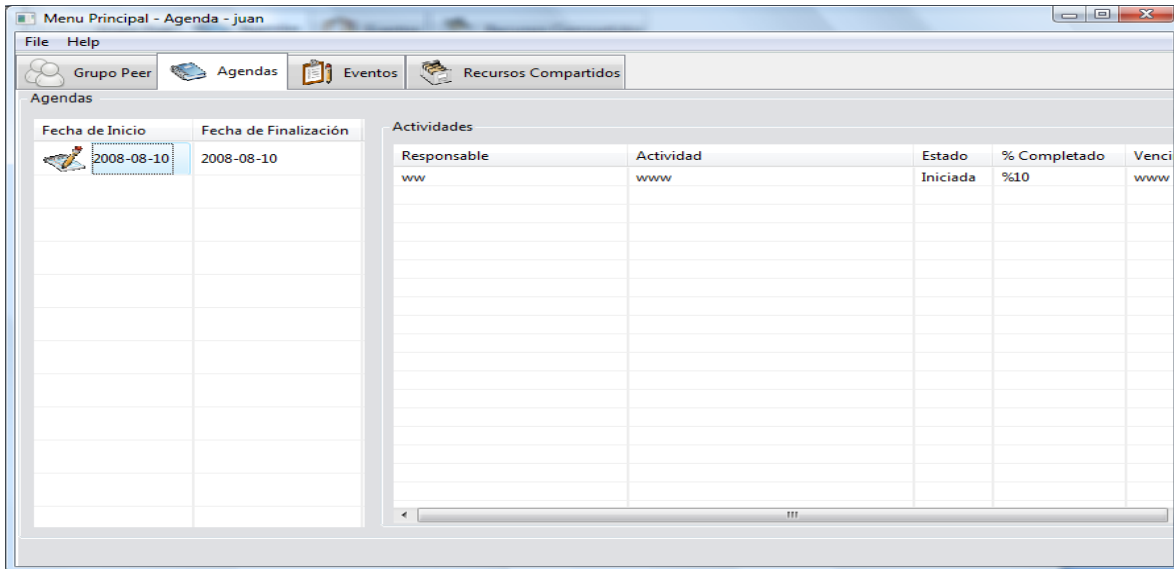


Figura 41 Agenda Semanal

En la figura 41 podemos observar una Agenda creada por el usuario Juan que va ser compartida

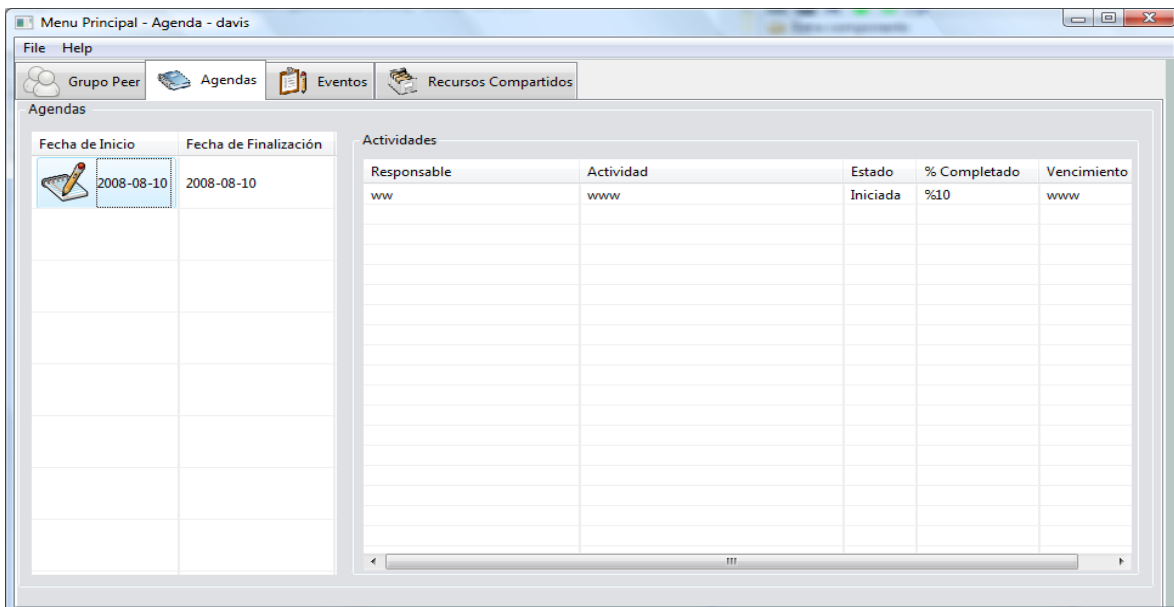
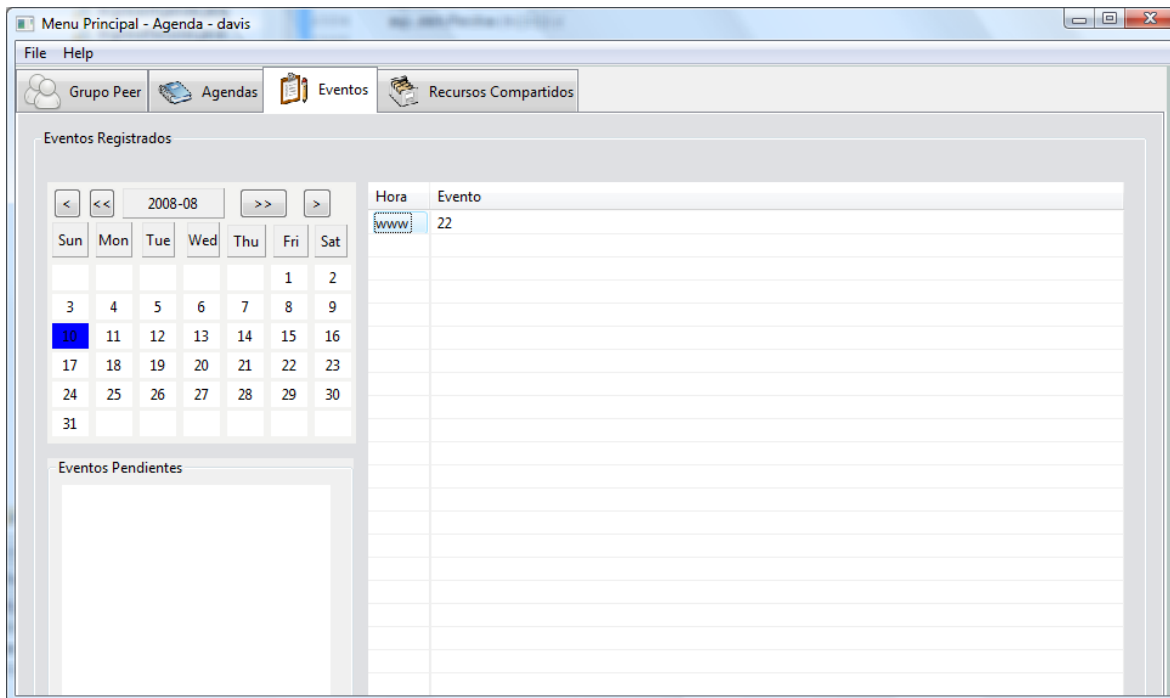


Figura 42 Agenda Semanal.

En la figura 42 podemos observar que la Agenda creada por el usuario Juan ya ha sido compartida.

## 5.5 ADMINISTRADOR DE EVENTOS

Este administrador cuenta con funcionalidades de ingreso, eliminación y consulta de eventos o sucesos para el grupo. La aplicación alerta al usuario de los eventos próximos a suceder.



**Figura 43** Administrador de Eventos.

En la figura 43 podemos observar la creación de eventos para el grupo

## 5.6 ADMINISTRADOR DE RECURSOS COMPARTIDOS.

Cada usuario puede compartir cualquier clase de recurso por medio de la ayuda de este administrador el cual lleva un registro que recursos están disponibles que usuario lo subió.

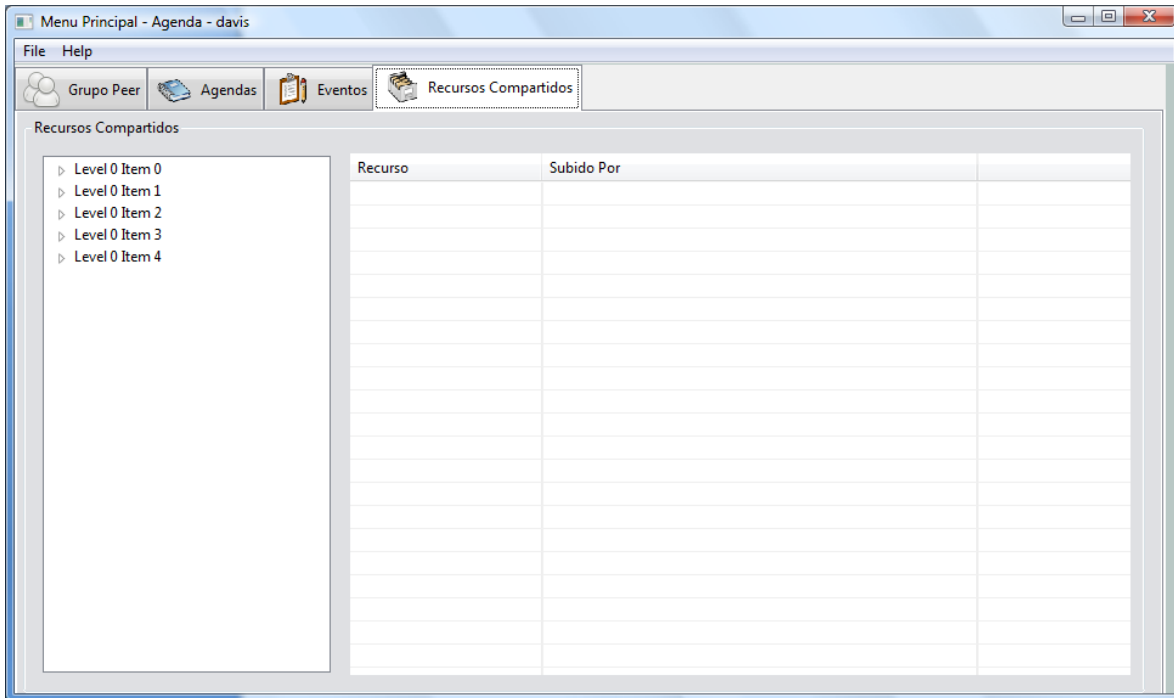


Figura 44 Administrador de Recursos compartidos.

La figura 44 representa la creación de recurso por el usuario Davis que luego serán compartidos.

## 5.7 UTILIDAD DE BÚSQUEDAS

La aplicación cuenta con la funcionalidad de búsqueda de grupos, usuarios, demás recursos.

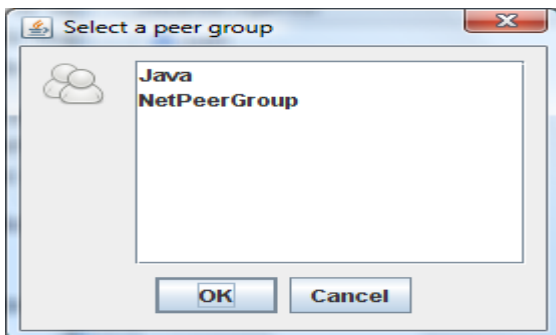


Figura 45 Consulta de Grupos

La figura 45 representa la funcionalidad de búsqueda de grupos

## CAPÍTULO 6 CONCLUSIONES Y TRABAJOS FUTUROS

Finalizado el análisis, diseño e implementación de perfiles UML para el desarrollo de aplicaciones P2P, hemos encontrado varios artefactos y entregables que reflejan en forma global las conclusiones obtenidas al desarrollar este trabajo de Grado. En cada capítulo se obtienen varios recursos software que mirados en perspectiva muestran un proceso de Ingeniería de software que será muy útil para el análisis, diseño e implementación de aplicaciones P2P con funcionalidades autónomas y reutilizables.

En el capítulo 3, diseño de Perfiles P2P, se definió un vocabulario que conforma un marco de trabajo de comunicación el cual es útil para evitar un doble significado y un mal entendido entre los términos utilizados en el presente trabajo de Grado, este vocabulario en general describe todos los elementos útiles y tenidos en cuenta en una Arquitectura Orientada a Servicios. Arquitectura en la cual el desarrollo de aplicaciones P2P es muy útil, eficiente e innovador.

En el capítulo 4, definición de Perfiles P2P, se detectan los servicios base y comunes en aplicaciones P2P los cuales son perfilados y clasificados de acuerdo a sus funcionalidades, cada perfil es detallado muy rigurosamente en un formato estándar de definición de Perfiles UML. Cada perfil completo será mas adelante un componente autónomo y reutilizable que podrá formar parte de una aplicación P2P más grande. Es decir podrá conformar un elemento más en un API base e inicial que facilite el desarrollo P2P.

En el capítulo 5 y en el anexo principalmente buscamos validar el modelado de perfiles P2P anteriormente obtenido desarrollando un Prototipo de Aplicación P2P basada en los perfiles definidos inicialmente. En el anexo se describen los diferentes diagramas de clases mapeadas a partir de los Perfil P2P encontrados en el capítulo 4.

Dentro de las contribuciones más relevantes de este proyecto podemos resaltar:

- Después del desarrollo de este trabajo se ha realizado un importante acercamiento e interrelación entre las redes P2P como plataforma de prestación de servicios y los conceptos, principios y guías de diseño de servicios, enmarcados bajo la teoría y especificaciones que provee UML a través de la definición de perfiles, lo que permitió proporcionar un enfoque desde la ingeniería de software al desarrollo de aplicaciones para plataformas P2P.
- Un vocabulario especializado en una Arquitectura Orientada a Servicio (SOA), este representa una nueva terminología que facilita la comprensión de nuevas ideas arquitecturales de diseño y construcción de una aplicación P2P.
- Proceso de Desarrollo de aplicaciones P2P más específicamente en las diferentes etapas de análisis, diseño y construcción, al definir tales procesos de diseño.
- Perfiles definidos útiles para el modelado y construcción de Aplicaciones P2P.



- Caso práctico que implementa un prototipo en este caso una Agenda Colaborativa con el fin de implementar todas las etapas de diseño e implementación y así obtener artefactos necesarios para su construcción de aplicaciones P2P, confirmando de esta manera la importancia práctica del proyecto y obteniendo una realimentación sobre la base teórica.
- En conclusión, este trabajo establece un punto de partida muy importante, desde la perspectiva de la ingeniería de software, para definir bases conceptuales de diseño que puedan enriquecer diferentes plataformas de más alto nivel, en lo que a integración de componentes, reutilización de software, elementos y artefactos propios del modelado de sistemas se refiere.

### **Trabajo Futuro**

Al hacer un aporte en el proceso de análisis y diseño de Aplicaciones P2P estamos iniciando un macro proceso que tendrá por objetivo principal estandarizar y formalizar una arquitectura de servicios P2P, a partir de un modelamiento fácil de servicios P2P, este macro proceso tendrá como base los perfiles definidos que se transformarán en el API inicial para la construcción de cualquier tipo de aplicación P2P.

El trabajo siguiente será establecer una plataforma completa de desarrollo reutilizable, modularizada y documentada que implemente servicios P2P base. Esta sería un API completo y documentado que al ser comprendido hará el desarrollo de una aplicación inicial P2P muy fácil.

Además será necesaria la implementación de un Modelo de Gestión de Servicios que administre de manera correcta la utilización de servicios comunes en aplicaciones P2P. Tal administrador de servicios deberá tener en cuenta características como eficiencia, tiempo de uso, configuración, monitoreo de servicios P2P, etc.

## CAPITULO 7 BIBLIOGRAFÍA

- [1] Farrell T, Service-oriented architect re: beyond web service, <http://onjava.com/pub/a/onjava/2005/01/26/soa-intro.htm>, 2005.
- [2] Napster. Inc, Napster, <http://www.napster.com>, 2005.
- [3] Dr. Jorge Cortell-Albert. El futuro de redes p2p en entornos corporativos, 2004.
- [4] Rudiger Shollmeier, A definition of peer to peer networking, Universidad técnica de Munich, 2005.
- [5] ICQ Inc. Página principal del software ICQ, <http://www.icq.com>.
- [6] Microsoft. Proyecto de software MSN Messenger, <http://messenger.msn.com/>.
- [7] Bordignon, F. y Tolosa, G., Gnutella: distributed system for information storage and searching. model description, JIT- Journal of Internet Technology. Taipei (Taiwan) Vol. 2, No. 5, <http://www.tyr.unlu.edu.ar/TYR-publica/paper-final-gnutella-english-v2.pdf>, 2001.
- [8] FIPA, Fipa peer-to-peer positioning paper, Technical Report F-OUT-00076, Foundation for Intelligent Pyhsical Agents, 2000.
- [9] Sunsted T, The practice of peer-to-peer computing: introduction and history, <http://www-106.ibm.com/developerworks/java/library/j-p2p/>.IBM Developer Networks
- [10] Barkay D, An introduction to peer-to-peer computing, 2000.
- [11] Computer-Based National Information Systems, Technology and public policy issues, 1981.
- [12] Minar, N. y Hedlund, M., A network of peers. peer-to-peer models through the history of the internet, A. Oram, editor, Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology, 2001.
- [13] Palmer J, Libro: electronic mail, ed. artech house publishers, isbn: 0-89006-802-x, 1995.
- [14] Kantor, B. y P. Lapsley, P., Network news transfer protocol, RFC 977.1986,
- [15] Shirky C, Listening to napster, peer to peer. Harnessing the power of disruptive technologies, O'Reilly & Associates. 2001.
- [16] Yang, B. y García-Molina, H, Comparing hybrid peer-to-peer systems, Proceedings of the 27th VLDB Conference, Roma, Italia, 2001.
- [17] Proyecto Groove, Groove, <http://www.groove.net>.
- [18] Tolosa, G., Bordignon, F. y Rodriguez, C., Una arquitectura cooperativa destinada a la distribución de un servicio de transmisión de radio por internet, I Congreso

Internacional de Tecnologías y Contenidos Multimediales en Ambientes Web. La Habana, Cuba, 2002.

[19] University of California, Seti@home, 2008..<http://setiathome.ssl.berkeley.edu/>.

[20] Distributed.net, <http://distributed.net/>, 1997.

[21] American Online Inc., Página principal del software aim, aol instant messenger, <http://www.aim.com>.

[22] Koman R, Infrasearch, [http://www.openp2p.com/pub/a/p2p/2001/03/19/kan\\_clary.html?page=2](http://www.openp2p.com/pub/a/p2p/2001/03/19/kan_clary.html?page=2).

[23] Dougherty D, All the pieces of pie, p2p networking overview, O'Reilly, 2001.

[24] JXTA.org, Jxta, <https://jxta.dev.java.net/>, 2008.

[25] Erik Christensen, Microsoft Francisco Curbera, IBM Research Greg Meredith, Microsoft Sanjiva Weerawarana, IBM Research, Web services description language (wsdl) 1.1, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>, 2001.

[26] Baker, M., Buyya, M. y Laforenza, D., The grid: international efforts in global computing, [www.cs.mu.oz.au/~raj/papers/TheGrid.pdf](http://www.cs.mu.oz.au/~raj/papers/TheGrid.pdf), 2000.

[27] SUN Microsystems, Jini, java intelligent network infrastructure, [http://www.jini.org/wiki/Main\\_Page](http://www.jini.org/wiki/Main_Page).

[28] Brendon J. Wilson, Jxta book, 2002.

[29] Lidia Fuentes AV, Una introducción a los perfiles uml, Depto. de Lenguajes y Ciencias de la Computación, Universidad de Málaga

[30] Carol Sliwa, Sidebar: waiting for uml 2.0, <http://www.computerworld.com/developmenttopics/development/story/0,10801,91325,00.html>.Computerworld

[31] Erl T, Soa principios de diseño de servicio, Boston USA, Julio 2007.