

**EJERCICIO DE MODELADO Y
DISEÑO DE APLICACIONES DISTRIBUIDAS**

Reto 2(Segunda parte)

Presentado por:

JORGE ANDRÉS GAVIRIA

DANIEL MÚNERA SÁNCHEZ

JAVIER QUINTERO GONZÁLEZ

Presentado a:

EDWIN MONTOYA MÚNERA

Profesor

UNIVERSIDAD EAFIT

DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS

TÓPICOS ESPECIALES EN TELEMÁTICA

MEDELLÍN

2012

DISEÑO DE APLICACIONES DISTRIBUIDAS

1. IDENTIFICACIÓN DEL PROBLEMA

“Se tiene un conjunto de servidores de archivos en Internet, los cuales se coordinan entre ellos para ofrecer un servicio uniforme de búsqueda y transferencia de archivos. Cada servidor tiene su propio conjunto de archivos almacenados, sin embargo si un usuario (cliente) se conecta a cualquier servidor debe ofrecerle el servicio de búsqueda de archivos sobre todo el conjunto de servidores. Una vez encuentra un archivo para el cliente, el servidor retornará la ubicación (url del archivo) del archivo solicitado.

Un cliente se puede conectar a cualquier servidor y de manera transparente buscar y localizar el archivo (o patrón de búsqueda) que el usuario quiere.

Nota: NO se puede emplear un esquema de índice centralizado.”

2. SERVICIOS O FUNCIONALIDADES DE LA APLICACIÓN

Requisitos Funcionales

- Ubicar y retornar información sobre la ubicación de un archivo.
- Permitir la comunicación entre el servidor que contiene un archivo con el cliente que desea recuperarlo.

3. REQUISITOS NO FUNCIONALES DEBIDO AL SISTEMA DISTRIBUIDO

Heterogeneidad:

Para garantizar la heterogeneidad es necesario idear, una forma de compartir los archivos de todos los servidores, de forma que independiente del sistema operativo del usuario pueda usar el aplicativo.

Extensibilidad:

El sistema debe ofrecer un mecanismo de comunicación tan generico que posibilite en el futuro exponer nuevos servicios en este tipo de sistema sin que se tenga que cambiar el protocolo de comunicación entre servidores.

Seguridad:

Se debe garantizar que unicamente se retorne valores que usuarios certificados puedan acceder de modo que no se ponga en riesgo la integridad de los servidores.

Escalabilidad:

Utilizando un sistema eficiente de comunicación entre los servidores, se puede garantizar un mayor numero de usuarios, ya sea clientes o servidores, debido a que esta es una funcion esencial se debe prestar mas cuidado en este ambito.

Tolerancia a Fallos:

En caso de que el sistema presente un error en la ubicación de un archivo el servicio debe tener la capacidad de continuar en funcionamiento, de igual manera si alguno de los que se encuentran en el conjunto de servidores se convierte en uno inaccesible la operación de la aplicación debe continuar.

Transparencia:

Para el cliente quien realiza la búsqueda debe ser transparente la comunicación que deba realizar el servidor que le recibió la petición con los demás peers, puesto que a él solo le importa que le retornen la URL donde pueda consultar el archivo deseado.

Concurrencia

El sistema debe soportar el acceso de múltiples usuarios al sistema los cuales desean consultar la ubicación de ciertos archivos.

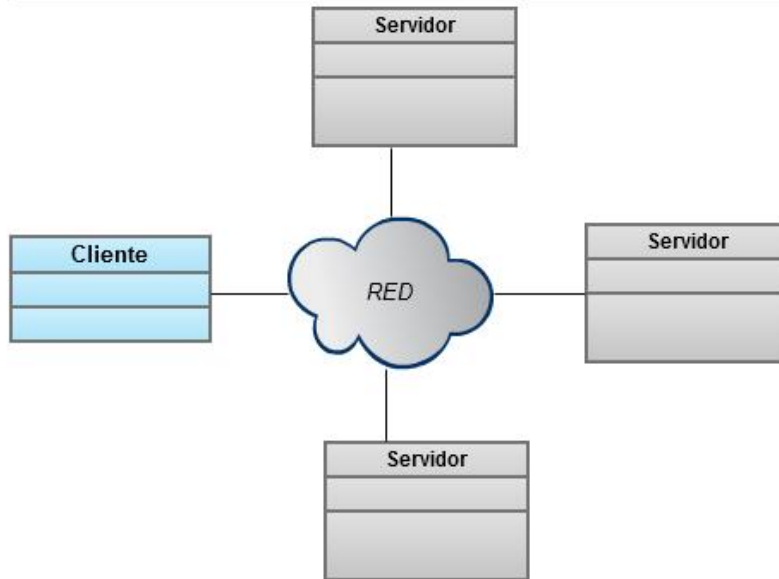
4. ENTIDADES EN EL SISTEMA

En este sistema de búsqueda y recuperación de información podemos identificar 2 entidades: cliente y servidor (El cual tiene ciertos tipos de entidades internamente).

El **cliente** es el interesado en recuperar la información que se encuentra almacenada en los servidores, puede conectarse a cualquiera de ellos para recuperar la información que desea, y no tiene que conocer de antemano quién tiene dichos archivos.

El **servidor** es quien contiene almacenada la información y responde a la petición de los clientes. Para dar respuesta a las peticiones realiza una búsqueda y si no se encuentra en la información que él tiene almacenada, debe encargarse de retornar la dirección URL del archivo solicitado. El cliente sólo realiza una consulta y es el servidor consultado el que debe dar respuesta al cliente de la ubicación de los recursos.

Entidades del Sistema

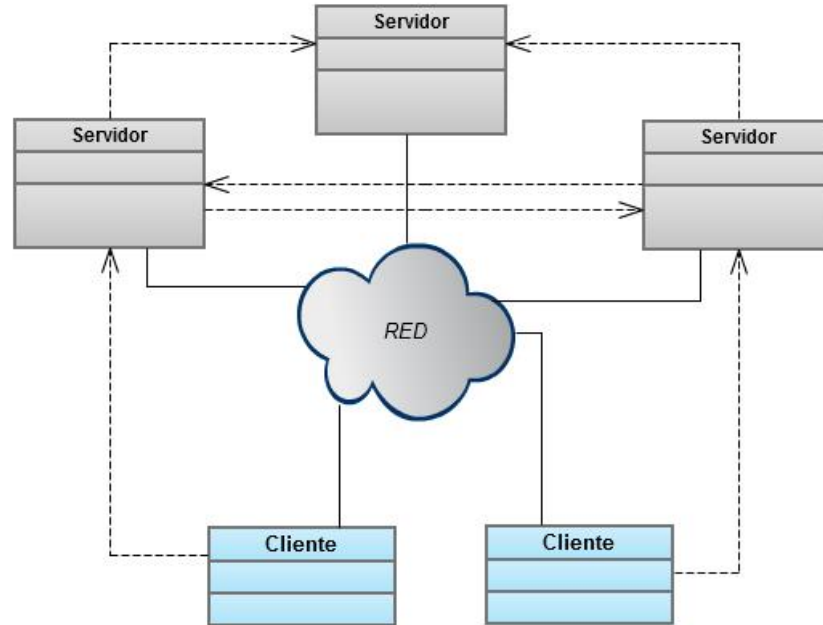


5. MODELO DE INTERACCIÓN ENTRE LAS ENTIDADES

El diseño de esta aplicación tiene una característica particular, dadas las condiciones dadas en las cuales no se puede tener un esquema de índice centralizado, lo que sería un modelo cliente servidor puro, nos encontramos en que debemos utilizar dos tipos de arquitectura. En primer lugar las interacciones entre el usuario y los contenedores de la información (servidores) responden a una arquitectura Cliente/Servidor, en la cual el cliente se encarga de hacer la solicitud de un recurso y el servidor se encarga de proveer respuesta a dicha petición retornando la dirección del archivo solicitado. Ahora bien, ya que el servidor que sea consultado debe dar una respuesta al cliente y no se cuenta con un esquema de índice centralizado, a nivel de los servidores debemos utilizar una arquitectura Peer to Peer en la que cualquier peer puede ofrecer un servicio a otro, en este caso el de indicar si se tiene o no el archivo por el cual se está preguntando.

El modelo de este sistema es cliente servidor, en el cual un servidor ofrece un servicio de búsqueda de archivos a los clientes, todo el procesamiento se lleva a cabo en el servidor y los clientes son livianos. Para el cliente las interacciones que deben darse entre los servidores para poder ubicar el recurso seleccionado son transparentes.

Modelo y Arquitectura



6. SERVICIOS Y FUNCIONALIDADES DE LA ENTIDADES

El Servidor debe ofrecer la funcionalidad **de retornar la ubicación (url) de un archivo que ha solicitado el cliente**. Para poder dar respuesta a dicha petición el servidor debe realizar una búsqueda sobre todo el conjunto de servidores, ya que si él no tiene la información debe consultar con los otros servidores para poder responder al cliente. Para ello debe encargarse de realizar consultas a los demás servidores sobre la información solicitada, así mismo este servidor puede consultar otras consultas hechas por otros servidores (los cuales han recibido la solicitud de algún cliente)

También destacamos otras funcionalidades de control que ayudan a soportar el sistema, estos son los **servicios de descubrimiento y de enrutamiento**, que son llevadas a cabo por los superpeers, estos permiten el descubrimiento de nuevos nodos en la red y el reporte de que servicios está prestando cada peer, que para nuestro caso sería el servicio de búsqueda de archivos y retorno de información relevante para el peer que hizo la consulta.

7. CARACTERÍSTICAS DE LAS INTERACCIONES RESPECTO A TIPOS DE INVOCACIÓN

Asimétrico entre cliente y servidor : Ya que el cliente hace los request al servidor, pero el servidor nunca hará un request al servidor.

Sincrónico, Simétrico y con estado entre servidores

Sincrónico:

Gracias a que el sistema requiere concurrencia se debe garantizar que un mismo usuario no realice múltiples peticiones al mismo tiempo o el sistema puede sobrecargarse por una gran cantidad de flujo por inundación o consumo de recursos

Simétrico:

Por las características solicitadas el conjunto de servidores tiene que comunicarse entre sí implicando la utilización de comunicación recíproca, por lo tanto implicando el uso de interacción simétrica.

Estado:

Se debe manejar el estado entre los servidores ya que es de suma importancia saber quiénes están disponibles en el momento para atender y ofrecer su servicio y archivos al sistema.

8. MIDDLEWARE SELECCIONADO

Debido a las características del sistema es necesaria la implementación de una arquitectura p2p pura entre servidores, de manera que permita un manejo eficiente del conjunto de servidores y de las interacciones con respecto de los clientes, por lo tanto se decide utilizar JXTA, gracias a que permite la característica de heterogeneidad, transparencia y soporte de fallos que son las características más importantes del sistema.

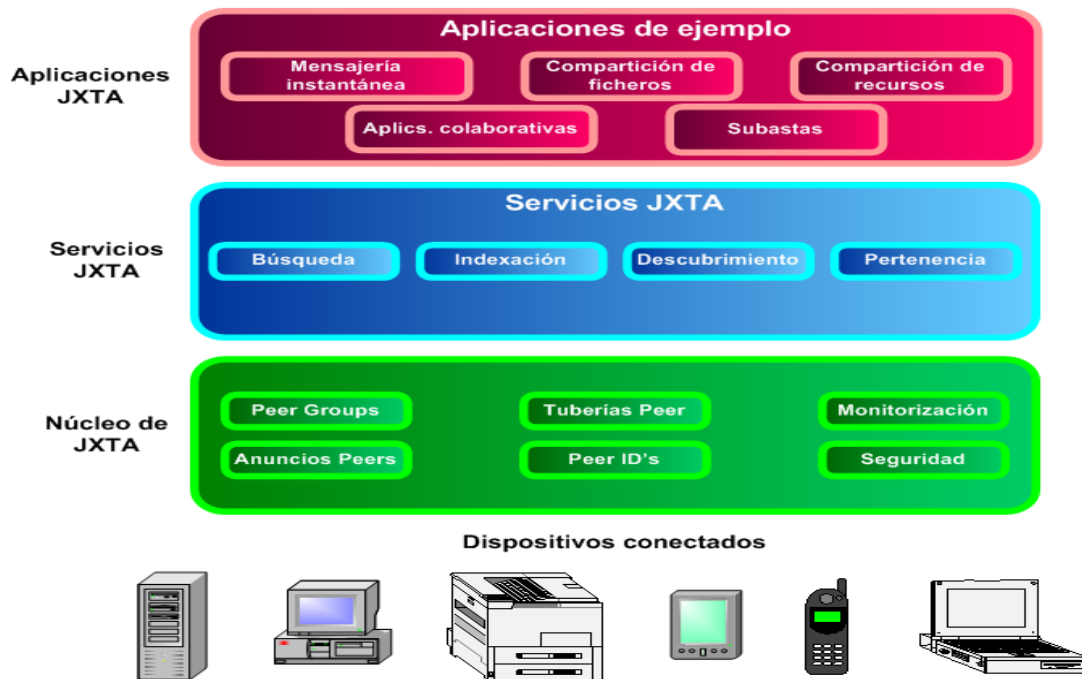
JXTA debido a su funcionamiento ofrece las siguientes ventajas y desventajas:

Ventajas: Escalabilidad, agregación de información, disponibilidad y tolerancia a fallos

Desventaja: Redundancia y las peticiones del servicio no son determinísticas

Presentamos una imagen que ilustra la estructura de JXTA

Arquitectura software del Proyecto JXTA



9. DISEÑO ESPECÍFICO

JXTA permite el uso de servicios p2p en los cuales se encuentran indexación y ubicación de archivos en redes p2p, permitiendo una mejora en terminos de eficiencia para estos dos puntos criticos.

Servidores:

Debido a que los servidores se pueden encontrar en redes diferentes es necesaria la definicion de **peers rendevouz y enrutadores**, estos se pueden asignar de manera dinamica conociendo las capacidades de los servidores y debe haber por lo menos un servidor rendevouz por isla en el sistema distribuido. Igualmente se podria considerar que cada isla tuviera un servidor enrutador, ya que este permite encontrar la ruta mas corta entre dos servidores de la red.

Debido a estas características, los servidores que no cumplan con alguna de las 2 propiedades, rendevouz o enrutamiento, se les considera simples y solo se utilizaran para la consulta de la informacion y distribucion de consultas en la red.

Los peers utilizaran los protocolos de comunicación ofrecidos por JXTA para comunicar sus peticiones y para transmitir la informacion de control necesaria

para estar siempre sincronizados y preparados para recibir peticiones de los clientes. Estos protocolos basicos son:

“Peer Resolver Protocol

Peer Information Protocol

Rendezvous Protocol

Peer Membership Protocol

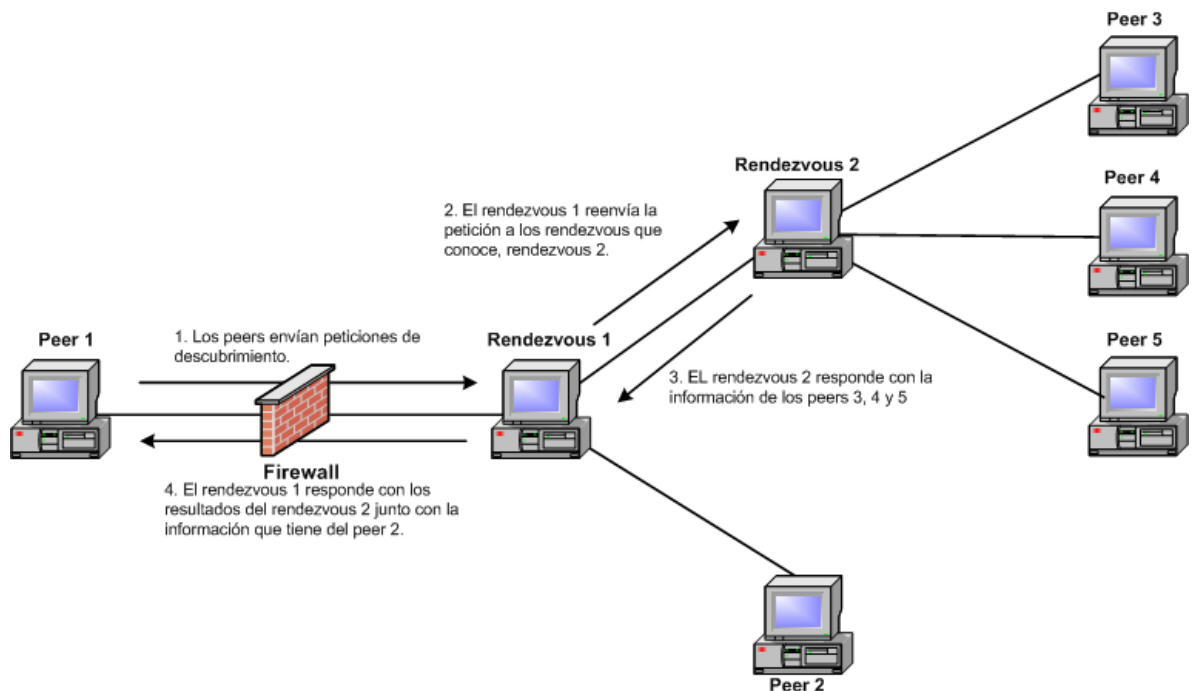
Pipe Binding Protocol

Endpoint Routing Protocol”

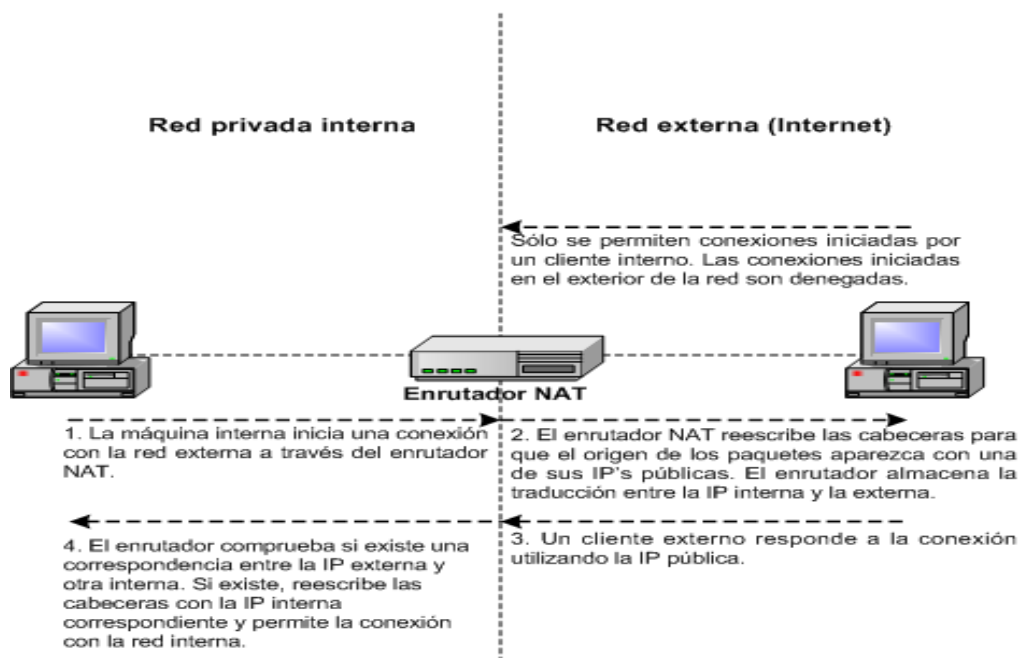
Nota: Tomados de Wikipedia.

Cada peer ofrecera a los demas un **servicio de busqueda de archivos** por tal motivo la red prestara **servicio por peer** y no por grupo de peers. Estos servicios ofrecidos y los peers conectados a una respectiva LAN seran descubiertos utilizando **anuncios**. Esto significa que cada peer de una LAN se anunciara y anunciara sus recursos, este mensaje llegara al **superPeer rendezvous**, quien sera el encargado de reportar a los demas miembros de la red fuera de la LAN que existe un nuevo peer con cierto recursos expuestos. Esto se llama **metodo de descubrimiento indirecto**.

Agregamos una imagen de una presentacion consultada para ilustrar el proceso de descubrimiento utilizando el metodo indirecto:



Para combatir el problema que presentan los firewalls para la comunicación entre peers que están ubicados en redes privadas diferentes utilizamos **Network Address Translation**, porque presenta las siguientes ventajas: Elimina la necesidad de una dirección IP pública por cada peer dentro de la red privada y permite proteger la red al existir un único punto de entrada. Utilizaremos de nuevo la ilustración de la presentación que utilizamos como referencia para ilustrar el proceso:



Teniendo ya claro como es la comunicación entre peers, podemos concluir que la comunicación entre peers para la búsqueda y entrega de la url donde se encuentra el archivo ya esta solucionada.

El paso final seria una comunicación tipo cliente servidor entre el cliente interesado en el archivo y el servidor propietario de el.

Clientes:

Clientes son las entidades que van a hacer la consulta de un archivo en la red, nosotros los tomamos como entidades que no van a ofrecer sus archivos a la red, sino que simplemente van a lanzar consultas sobre a los servidores. Por lo explicado anteriormente, se piensa que la comunicación e los usuarios y los servidores, sera una comunicación cliente servidor, donde la complejidad del sistema estara sobre la red de servidores.

Bibliografia

Medios magneticos:

1. Guia de programador para JXTA: <http://es.scribd.com/doc/13093191/JXTA-Developer-Guide-25-Spanish>

2. Presentacion resumen sobre JXTA conceptos basicos y comparacion de redes P2P con cliente servidor:

<http://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&cts=1331443854818&sqi=2&ved=0CFYQFjAF&url=http%3A%2F%2Fditec.um.es%2Fssdd%2Ftrabajos%2FJXTA.ppt&ei=Xx9cT6X8MMG1twerwemEDA&usg=AFQjCNHv6EbHv0w8fdv2lzwy3capvYtuCw>

3. <http://es.wikipedia.org/wiki/JXTA>