

# **Un nuevo servicio de búsqueda escalable**

Diseño de una arquitectura de indexación y búsqueda distribuida  
para el Sistema de Distribución de Objetos (ODS)

Tesis de Licenciatura

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

Ana I. Merlino  
L.U. 11/90  
amerlino@dc.uba.ar

Gastón A. Franco  
L.U. 442/90  
gaston@esc.edu.ar

Director: Ing. Claudio E. Righetti  
Co-Director: Lic. Carlos Blanco

Junio 1999

Dedicada a nuestros padres  
Martha, Domingo, Nelly y Ricardo



## Agradecimientos

Quisiéramos agradecer a nuestros directores Claudio Righetti y Carlos Blanco, por su dedicación y paciencia.

A nuestros amigos Diego, Martín, Adrián, Telma, Andrea, Fernando y Jorge, por estar siempre.

A Nicolás Baumgarten y Julián Dunayevich por su colaboración y por darnos la posibilidad de manejar flexiblemente nuestros horarios de trabajo.

A María Eva Lijding por su ayuda y apoyo a lo largo de todo el trabajo.

A Patricia Borejstein por ayudarnos en los comienzos.

A Elena Vinelli y Jorge Mayer por leer y comentar las últimas versiones del trabajo.

De parte de Gastón, a Carolina por todo su apoyo y comprensión. A mis hermanas Laura y Silvina y a la banda de amigos de Junín: Marcelo, Carlos, Andrés, Martín, Gustavo, Matías y Pablo, por estar.

De parte de Ana, a mis hermanos Gustavo y Mariana por todo el apoyo que me brindaron y por acompañarme siempre.

## **Resumen**

En los últimos años se han desarrollado varios servicios que proveen a los usuarios facilidades de localización y búsqueda de recursos en Internet. A pesar de su popularidad dentro de la comunidad de usuarios, estos servicios no han sido diseñados para manejar adecuadamente un entorno caracterizado por un gran volumen de información, una administración descentralizada, y gran cantidad de servidores de información y usuarios distribuidos geográficamente. Estudiamos las fallas de escalabilidad que presentan las soluciones actuales y el comportamiento de varios prototipos basados en arquitecturas distribuidas en el marco global de Internet, más específicamente para la Web.

Este trabajo presenta el diseño de un nuevo servicio de búsqueda escalable que tiene por objetivo permitir a la comunidad de usuarios del Sistema de Distribución de Objetos (ODS) localizar y buscar objetos de interés de entre los producidos. El servicio está basado en una arquitectura de búsqueda e indexación distribuida que contempla la existencia de múltiples servidores de índices y mecanismos de comunicación para la resolución cooperativa de consultas. Los Servidores de Índices son los encargados de indexar la meta-información de los objetos y responder consultas. Están interconectados entre sí por la Red de Agentes de Búsqueda encargada de manejar el contexto global para la resolución distribuida de consultas provenientes de las Interfaces de Usuario. Estas últimas proveen a los usuarios el front-end necesario para acceder y utilizar el servicio de búsqueda.

## **Abstract**

In the last few years a series of services have been developed that provide searching and location of resources in the Internet. In spite of their popularity among users, these services have not been designed to correctly manage an environment characterized by a huge amount of information, distributed administration and a large number of geographically disperse information servers and users. We study the scalability failures that the current solutions show and the performance of different prototypes based on distributed architectures on the global Internet framework, more specifically for the Web.

In this work we present the design of a new scalable searching service which aims to provide searching and location of objects of interest to the Objects Distribution System (ODS) user community. The service is based on a distributed searching and index architecture that takes into account the existence of multiple index servers and communication engines for the cooperative resolution of queries. The Index Servers are in charge of indexing objects' meta-information and answering queries. They are interconnected by the Searching Agents Network which handle the global context for the distributed resolution of queries that come from User Interfaces. The latter provide the users with the necessary front-end to access and use the Searching Service.

# Indice

<b>1. Introducción .....</b>	<b>1</b>
1.1 Organización del trabajo.....	5
<b>2. Conceptos generales .....</b>	<b>6</b>
2.1 Sistemas para el Descubrimiento de Recursos.....	6
2.1.1 Aspectos del problema de Descubrir Recursos.....	7
2.2 Escalabilidad .....	8
2.2.1 Volumen de Información.....	8
2.2.2 Cantidad de Usuarios.....	10
2.2.3 Diversidad de información.....	10
2.3 ODS.....	11
2.3.1 Características.....	11
2.3.2 Comparación con News.....	12
<b>3. Tipos de arquitecturas .....</b>	<b>15</b>
3.1 Componentes y clasificación .....	15
3.2 Arquitecturas centralizadas.....	17
3.2.1 Problemas de escalabilidad.....	20
3.3 Arquitecturas distribuidas.....	23
3.3.1 Mallas de brokers.....	26
3.3.1.1 Composición jerárquica de brokers.....	27
3.3.1.1.1 Algunas reflexiones.....	29
3.3.1.2 Composición no jerárquica de brokers.....	31
3.3.1.2.1 Algunas reflexiones.....	34
3.3.1.3 Colecciones y meta-información de contenido.....	35
3.3.1.3.1 Generación de meta-información.....	35
3.3.1.3.2 Query Routing.....	37
3.3.2 Bibliotecas Digitales.....	37
3.3.2.1 NCSTRL (Networked Computer Science Technical Research Library).....	38
3.3.2.1.1 Arquitectura Dienst.....	38
3.3.2.1.2 Connectivity Regions.....	40
3.4 Conclusiones .....	42
<b>4. Servicio de búsqueda para ODS .....</b>	<b>44</b>
4.1 Introducción .....	44
4.2 Objetivos del diseño.....	45
4.3 Servicios .....	47
4.3.1 Sugerencia de grupos de interés.....	47
4.3.2 Búsqueda dentro de los grupos de interés.....	49
4.4 Resolución distribuida de consultas.....	50
4.4.1 Estrategias de resolución.....	51
4.4.1.1 Alternativa 1: Broadcast a los SA productores.....	51
4.4.1.2 Alternativa 2: Query routing a los mejores SA productores.....	51
4.4.1.3 Alternativa 3: Mantener un índice local para todos los objetos de un grupo.....	52
4.4.1.4 Discusión.....	52
4.4.2 Características particulares de ODS.....	53
4.4.2.1 Disponibilidad de información local en los SA.....	54
4.4.2.2 Múltiples réplicas para los grupos de interés.....	56
4.4.2.2.1 Agentes de Búsqueda.....	59

<b>5. Arquitectura propuesta .....</b>	<b>61</b>
<b>5.1 Introducción .....</b>	<b>61</b>
<b>5.2 Red de Agentes de Búsqueda.....</b>	<b>63</b>
5.2.1 Servicios a proveer a los SA.....	64
5.2.2 Mecanismos de diseminación de información.....	65
5.2.3 Responsables de Grupos (RG).....	67
5.2.3.1 Garantizar existencia de SI para cada grupo.....	67
5.2.3.2 Generación de Meta-Información de Grupo (MiG).....	68
5.2.3.3 Creación de Responsables de Grupo.....	69
5.2.3.4 Cómo contactar a un responsable de grupo.....	70
5.2.4 Tabla de SI de Agente.....	71
5.2.4.1 Vistas de SI por grupo de interés.....	71
5.2.4.2 Políticas de uso.....	73
5.2.4.3 Meta-información de SI.....	74
5.2.4.4 Mantenimiento.....	75
<b>5.3 SA's .....</b>	<b>76</b>
5.3.1 Resolución de consultas para los usuarios.....	76
5.3.2 Resolución de consultas para otros SA.....	77
5.3.3 SA productores.....	78
<b>5.4 Consideraciones finales.....</b>	<b>78</b>
 <b>6. Conclusiones y trabajos futuros .....</b>	 <b>79</b>
6.1 Conclusiones.....	79
6.2 Trabajos futuros.....	83
 <b>Apéndice A - Servicios de Búsqueda Distribuidos para la Web .....</b>	 <b>84</b>
<b>A.1 Meta-Buscadores.....</b>	<b>84</b>
<b>A.2 Mallas Jerárquicas de Brokers.....</b>	<b>84</b>
A.2.1 Content Routing.....	84
A.2.2 Whois++.....	86
A.2.3 Harvest.....	87
<b>A.3 Mallas no jerárquicas de brokers.....</b>	<b>88</b>
A.3.1 AlephWeb.....	88
A.3.2 Sistema distribuido basado en código móvil.....	90
A.3.3 Ingrid.....	91
A.3.4 Mecanismos Probabilísticos para el Descubrimiento de Recursos (MPRDS).....	93
 <b>Apéndice B - News .....</b>	 <b>95</b>
 <b>Apéndice C - Ejemplos de codificación en formato SOIF .....</b>	 <b>99</b>
<b>C.1 Descripción formal de SOIF.....</b>	<b>99</b>
<b>C.2 Consultas.....</b>	<b>100</b>
C.2.1 Sintaxis.....	102
<b>C.3 Resultados.....</b>	<b>103</b>
C.3.1 Sintaxis.....	104
<b>C.4 Meta-información de SI.....</b>	<b>106</b>
C.4.1 Sintaxis.....	107
 <b>Bibliografía .....</b>	 <b>109</b>

# Listado de gráficos

Gráfico 2.1 Dimensiones de escalabilidad.....	8..
Gráfico 2.2 Navegar.....	9...
Gráfico 2.3 Buscar.....	9...
Gráfico 3.1 Componentes de un servicio de búsqueda.....	15
Gráfico 3.2 Clasificación de Arquitecturas.....	16
Gráfico 3.3 Arquitectura de indexación global centralizada.....	18
Gráfico 3.4 Estructura interna de un servicio centralizado.....	19
Gráfico 3.5 Buscador DejaNews.....	20.
Gráfico 3.6 Broadcast Global.....	24.
Gráfico 3.7 Query routing a un subconjunto.....	24
Gráfico 3.8 Etapas del query routing.....	24
Gráfico 3.9 Funcionamiento de un broker.....	25
Gráfico 3.10 Malla Jerárquica de brokers.....	27.
Gráfico 3.11 Funcionamiento interno de un Broker de Content Routing.....	28
Gráfico 3.12 Malla No jerárquica de brokers.....	31
Gráfico 3.13 Estructura interna de un broker.....	32
Gráfico 3.14 Interacción entre Collection Services y otros Servicios de Dienst.....	40
Gráfico 3.15 Connectivity Regions.....	41.
Gráfico 3.16 Collection Service distribuido.....	42
Gráfico 4.1 Arquitectura centralizada en ODS.....	46
Gráfico 4.2 Arquitectura jerárquica para resolución de consultas en ODS.....	50
Gráfico 4.3 Broadcast a SA $P(X)$ .....	51.
Gráfico 4.4 Query routing a los mejores SA $P(X)$ .....	52
Gráfico 4.5 Responsables como consumidores de grupo.....	52
Gráfico 4.6 Uso de información local.....	54
Gráfico 4.7 Resolución de consultas externas de los SAs.....	56
Gráfico 4.8 Grupos de réplicas disponibles.....	57
Gráfico 4.9 Múltiples servidores de índices para un grupo de interés.....	57
Gráfico 4.10 Resolución cooperativa de las consultas.....	58
Gráfico 4.11 Agentes de Búsqueda.....	60.
Gráfico 5.1 Resolución cooperativa de consultas sobre grupos de interés.....	62
Gráfico 5.2 Índice de MiGs.....	64.
Gráfico 5.3 Tabla de SI de Agente.....	64.
Gráfico 5.4 Mecanismos de diseminación entre responsables y agentes.....	67
Gráfico 5.5 Proceso de Generación de MiG.....	68
Gráfico 5.6 Replicación de responsables.....	70
Gráfico 5.7 Diferentes vistas de SI para el mismo grupo de interés.....	72
Gráfico A.1 Meta Buscador.....	84.
Gráfico A.2 Content routing.....	85.
Gráfico A.3 Arquitectura de Discover.....	86
Gráfico A.4 Arquitectura de Harvest.....	87.
Gráfico A.5 Arquitectura de AlephWeb.....	89

Gráfico A.6 Arquitectura Mobile Code Based.....	90
Gráfico A.7 Clusters de RPs.....	91.
Gráfico A.8 Componentes de la topología Ingrid.....	92
Gráfico A.9 Mecanismos probabilísticos para RD.....	93

# Capítulo 1

## Introducción

En los últimos años Internet se ha perfilado como un importante medio que promueve globalmente la comunicación y colaboración social. Para lograrlo, la comunidad Internet se concentró principalmente en la construcción de redes cada vez más veloces y extensas. Sin embargo, la rápida evolución que ha experimentado en la infraestructura de comunicaciones ha dejado en evidencia una infraestructura inmadura para el manejo de información.

Internet alberga una vasta colección de servidores de información que puede ser vista como una gran base de datos heterogénea y distribuida geográficamente que contiene un significativo volumen de información. Crece día a día con la incorporación de nuevos servidores y recursos. Los usuarios acceden el conjunto de recursos disponibles a través de diversos sistemas de información como el Web [Berners-Lee 92], Gopher [Alberti 92], FTP y News [RFC 1036], entre otros.

El gran volumen de información disponible y su amplia distribución dificultan el proceso a través del cual el usuario descubre recursos de interés. Los mecanismos habituales con los que cuenta el usuario (localizar un recurso a partir de su ubicación física) dejan de ser válidos para ser sustituidos por sistemas que lo ayuden a tratar con este espacio de información complejo [Sheldon 95a].

Pero, ¿cuál es el objetivo del proceso de descubrimiento de recursos?. Éste puede ser definido como el de hallar respuesta a alguna necesidad de información, o consulta. Dicha "respuesta" puede tener distintas características desde el momento en que depende de la situación del usuario, de los costos involucrados, de ciertas restricciones de tiempo y de otros factores. En algunos casos, lo que se busca es la mejor respuesta posible<sup>1</sup>. En otros casos, el usuario estará satisfecho con una respuesta menos detallada o más global. Asimismo, los objetivos del proceso de descubrimiento de recursos pueden cambiar a medida que el usuario se ve influenciado por nuevas necesidades o por los recursos ya descubiertos. Finalmente, el proceso puede comenzar sin que el usuario tenga una clara idea de lo que necesita.

John Kunze argumenta que el proceso de descubrimiento de recursos consiste en una serie de movimientos entre dos fases o estados [Kunze 97]. Una fase es la *localización*, durante la cual un usuario formula una serie de criterios de selección (consulta) a partir de los cuales se derivan un

---

<sup>1</sup> Por el momento ignoraremos las diferencias acerca de lo que se considere "mejor".



conjunto de recursos candidatos. La segunda fase es la *examinación*, que involucra el análisis de dichos candidatos. Como resultado de la examinación, el usuario puede dar por terminado el proceso de descubrimiento habiendo encontrado una respuesta satisfactoria, o bien, volver a la fase de localización modificando sus criterios de selección.

En los últimos años se han desarrollado varios servicios conocidos como *Sistemas para el Descubrimiento de Recursos* [Schwartz 93] [Obraczka 93] [Ianella 96] que proveen a la comunidad de usuarios facilidades para la localización y búsqueda de recursos disponibles en Internet. Estos sistemas se clasifican en dos grandes grupos: de navegación y de búsqueda [Bowman 94] [Obraczka 93]. Los primeros organizan el espacio de información de modo tal que los usuarios puedan “navegar” los contenidos en busca de información de su interés. Los segundos, proveen facilidades de búsqueda permitiendo un acceso más directo a la información relevante. Los sistemas de búsqueda se han vuelto muy populares dentro de la comunidad de usuarios, siendo actualmente los más utilizados.

Existen muchos sistemas de búsqueda que podemos diferenciar según su área de cobertura. Los de propósito general tales como Yahoo [YH], Lycos [LY], AltaVista [AV], Excite [EX] permiten realizar búsquedas sobre variados recursos (páginas Web, archivos en repositorios FTP, etc.). Los de propósito específico, se especializan sólo en determinados tipos de recursos, como es el caso de DejaNews [DN] que realiza búsquedas sobre los grupos de noticias de News.

News [RFC 1036] es un sistema de información existente en Internet que permite a los usuarios compartir información. Maneja alrededor de 15.000 grupos de interés y se estima que la producción de información diaria ronda los 900 Mb [DN].

El *Sistema de Distribución de Objetos (ODS)* [Righetti 97] es funcionalmente similar a News. Sin embargo, busca resolver problemas de latencia en el acceso a la información utilizando protocolos dinámicos para la replicación selectiva de objetos desde los nodos productores hacia los nodos consumidores. Permite que los usuarios accedan a copias locales de los objetos producidos en la red, a semejanza de las redes de distribución utilizadas en la vida diaria (cadenas de distribución de productos alimenticios, publicaciones, bibliotecas, etc.).

El gran volumen de información manejado por estos dos sistemas hace que los usuarios deban invertir mucho tiempo para encontrar la información de su interés. En el caso particular de News, ya existe un sistema de búsqueda específico denominado DejaNews [DN] que permite a los usuarios realizar búsquedas dentro de los grupos de noticias.

En este trabajo presentamos un servicio de búsqueda distribuido para ODS, con el objetivo de que los usuarios del sistema puedan buscar información de interés sobre el total de la producida. El servicio de búsqueda abarcará contenidos globales: el hecho de que un usuario consuma o no determinado grupo de interés no restringirá sus posibilidades de búsqueda.

Hemos elegido trabajar con ODS porque fue diseñado para que escale al crecimiento del volumen de información y la cantidad de nodos. El diseño de DejaNews, así como el de la mayor parte de los sistemas de búsqueda de propósito general, se basa en una arquitectura centralizada



caracterizada por utilizar técnicas monolíticas de búsqueda e indexación, donde un único sitio es el encargado de indexar toda la información disponible [W3C 96]. Las arquitecturas centralizadas presentan problemas de escalabilidad al no estar preparadas para manejar adecuadamente un entorno caracterizado por un crecimiento exponencial del volumen de información, una administración descentralizada, y gran cantidad de servidores de información y usuarios distribuidos geográficamente [Bowman 94] [Sheldon 95a] [Weider 96].

Los sistemas de búsqueda centralizados tienen por objetivo reunir localmente la mayor cantidad posible de información de toda la disponible. En su gran mayoría, estos servicios utilizan robots [Koster 95] para la recolección de datos, generando una sobrecarga de trabajo en los sitios proveedores de información ya que deben dedicar gran parte del tiempo a satisfacer los requerimientos de estos programas [Bowman 95] [Sassi 98]. La necesidad de transmitir recursos completos desde los sitios proveedores de información hasta el servidor para la generación del índice provoca excesivo tráfico en la red. La necesidad de mantener la coherencia de los datos almacenados (*up-to-date*) agrava considerablemente este problema.

El gran volumen de información requerirá, para el sitio proveedor del servicio, gran capacidad de procesamiento y cantidad de recursos informáticos (espacio de almacenamiento, ancho de banda de los enlaces, etc.) dado que la curva de crecimiento del índice que mantenga será similar a la del crecimiento del volumen de información producido [Bowman 94]. La administración y el mantenimiento de índices de tal envergadura es un proceso costoso. Las tecnologías actualmente disponibles para la recuperación de información (*information retrieval*) ven disminuida su efectividad a medida que crece el tamaño del índice y aumenta la diversidad de recursos, recuperando muchos documentos irrelevantes [Lagoze 97].

Estos sistemas se vuelven un cuello de botella y un único punto de falla frente al crecimiento del número de usuarios del servicio. Es difícil que un servicio centralizado no colapse cuando la cantidad de consultas a procesar aumenta: sus enlaces se verán saturados, necesitará mayor capacidad de procesamiento, etc.

Resumiendo, los sistemas de búsqueda basados en una arquitectura centralizada constituyen soluciones difícilmente escalables. Esto se debe a la necesidad permanente que tienen de incrementar la cantidad de recursos informáticos necesarios como ser: espacio en disco, capacidad de procesamiento, cantidad de servidores dedicados, etc., para no colapsar y poder contener y manejar adecuadamente las cambiantes condiciones de entorno.

Los prototipos existentes basados en arquitecturas distribuidas de búsqueda e indexación han mostrado tener mejores propiedades de escalabilidad [Sheldon 95a] [Bowman 95] [Rodríguez 96] [W3C 96].

Hemos analizado y estudiado el comportamiento de varios prototipos basados en arquitecturas distribuidas de búsqueda e indexación en el marco de Internet, más específicamente para la Web (ver "Apéndice A"), puesto que para News no existen otras herramientas que utilicen una arquitectura diferente de la centralizada.

Como resultado de este análisis, diseñamos una arquitectura de indexación y búsqueda distribuida para que escale al crecimiento en la cantidad de nodos, la cantidad de usuarios y el volumen de información.

Hemos combinado y adaptado las ideas más relevantes de las arquitecturas estudiadas para explotar las características particulares de ODS, definiendo una arquitectura basada en el uso de múltiples servidores de índices y mecanismos de comunicación para la resolución cooperativa de las consultas.

Los *Agentes de Servicio (SA)*, entidades existentes en ODS, serán los encargados de proveer una interfaz de consulta para que su comunidad de usuarios pueda efectuar requerimientos al servicio de búsqueda. Para ello implementarán el mecanismo de resolución de consultas. Para resolver una consulta, utilizarán la información local y sólo en el caso de no poder satisfacerla requerirán la cooperación de los *Agentes de Búsqueda (AB)*, reenviándoles la consulta para su resolución. La información disponible localmente, que refleja los intereses de los usuarios locales puesto que cubren los grupos que éstos han decidido consumir, evitará en muchos casos una resolución distribuida que es más costosa [Rodríguez 96] a la vez que permitirá a los SA proveer servicios de valor agregado específicos para su comunidad de usuarios.

Los *Agentes de Búsqueda (AB)*, entidades no existentes en ODS, serán los encargados de proveer todo el contexto global a los SA resolviendo las consultas que éstos no puedan satisfacer localmente. Para la resolución distribuida de las consultas utilizarán estrategias de *query routing adaptativo* [Sheldon 95a] [Lagoze 98] para encaminar las consultas sólo a los servidores de índices que mejor puedan satisfacerlas. El comportamiento de cada AB será independiente del resto y estará guiado por las políticas de uso que tenga definidas permitiendo una administración descentralizada del servicio [Rodríguez 96].

## 1.1 Organización del trabajo

En el capítulo 2, “*Conceptos Generales*”, se introducen los conceptos básicos fundamentalmente en lo referido a dimensiones de escalabilidad. Se analiza el funcionamiento del Sistema de Distribución de Objetos (ODS), comparándolo con News.

En el capítulo 3, “*Tipos de Arquitecturas*”, se describe funcionalmente la arquitectura de un servicio de búsqueda. Se presentan y clasifican los tipos de arquitecturas existentes, analizando en profundidad distintas soluciones distribuidas desde el punto de vista de la escalabilidad.

En el capítulo 4, “*Servicio de Búsqueda para ODS*”, se describen los servicios a proveer a los usuarios. Se presentan los objetivos del diseño y se describen las características particulares de ODS analizando como pueden ser utilizadas para la resolución cooperativa de consultas.

En el capítulo 5, “*Arquitectura Propuesta*”, se presenta la arquitectura propuesta para el servicio de búsqueda y se describe su funcionamiento.

En el capítulo 6, “*Conclusiones y Trabajos Futuros*”, se presentan las conclusiones y posibles extensiones al trabajo.

En el apéndice A, “*Servicios de Búsqueda Distribuidos para la Web*”, se enumeran los servicios de búsqueda distribuidos para la Web más relevantes y se da una breve descripción de su arquitectura.

En el apéndice B, “*News*”, se describe en detalle el funcionamiento del sistema News.

En el apéndice C, “*Ejemplos de codificación en formato SOIF*” se presenta un formato posible para la codificación de consultas y resultados utilizando el estándar SOIF<sup>1</sup> (Summary Object Interchange Format).

---

<sup>1</sup> Es una sintaxis legible por una computadora que se utiliza para transmitir resúmenes estructurados de objetos.

## Capítulo 2

# Conceptos generales

Analizaremos las actividades involucradas en el proceso de descubrimiento de recursos en redes de área extendida y caracterizaremos a los Sistema para el Descubrimiento de Recursos (RDS). Introduciremos los conceptos básicos fundamentalmente en lo referido a escalabilidad. Explicaremos el funcionamiento del Sistema de Distribución de Objetos (ODS) comparándolo con News.

### 2.1 Sistemas para el Descubrimiento de Recursos

Al igual que muchos términos utilizados en el área de tecnologías de la información, existen múltiples caracterizaciones del proceso de descubrir recursos.

En [Ianella 96] se define:

***Descubrir Recursos** (Resource Discovery) es el proceso que abarca la localización, el acceso, la recuperación y el manejo de recursos en redes heterogéneas y distribuidas*

Donde:

**Recurso**, es cualquier objeto estático o dinámico que pueda ser identificado. Puede existir por un período corto de tiempo o ser persistente.

**Descubrir**, involucra el proceso de encontrar y recuperar aquellos recursos que sean relevantes a los usuarios.

**Usuarios**, son seres humanos con una necesidad de encontrar información de su interés.

**Proveedores**, son entidades que proveen el acceso y recuperación de recursos a una comunidad de usuarios.

**Espacio de Información**, es aquel constituido por el conjunto de los recursos distribuidos geográficamente en la red.

En grandes redes como en el caso de Internet, el espacio de información se caracteriza por ser muy *amplio, disperso y dinámico*. Estas características están dadas fundamentalmente por la permanente incorporación y eliminación de recursos<sup>1</sup> en los proveedores de información, sumado a la incorporación y eliminación de proveedores de información distribuidos en todo el mundo.

Dichas características de entorno dificultan el proceso a través del cual el usuario descubre recursos de interés. Los mecanismos naturales con los que cuenta un usuario (localizar un recurso a partir de su ubicación física) dejan de ser válidos. El gran volumen de información disponible hace necesaria la construcción de sistemas que ayuden a los usuarios a encontrar los recursos de su interés permitiéndoles tratar con este espacio de información complejo.

Descubrir recursos es un problema global, dado a que la información se encuentra distribuida globalmente. Esto significa que desde un punto de vista técnico, el usuario de un sistema para el descubrimiento de recursos debe poder buscar información en los proveedores disponibles mundialmente. Desde el punto de vista del usuario, la localización de un recurso debe ser transparente, dándole la impresión de que está accediendo una gran y única base de datos con toda la información.

Un *Sistema para el Descubrimiento de Recursos (RDS)* es aquel sistema de información que permite a los usuarios interactuar con un espacio de información complejo a los efectos de poder satisfacer sus necesidades de localizar y recuperar recursos de su interés [Schwartz 92] [Schwartz 93] [Obraczka 93].

### 2.1.1 Aspectos del problema de Descubrir Recursos

Descubrir Recursos en entornos como el de Internet es un proceso difícil principalmente por dos razones: la administración descentralizada y la distribución y dinamismo del espacio de información.

En primer lugar, Internet está compuesta por muchas redes autónomas que se expanden a lo largo de límites geopolíticos, cada una manejada por su propia organización. No existe una única organización que tenga el control administrativo sobre Internet, por el contrario, se caracteriza por tener una administración compartida entre comunidades.

Esta falta de un cuerpo contralor, que por un lado beneficia la libertad de expresión, crea dificultades para los usuarios que tratan de localizar recursos de interés.

En particular, ninguna organización tiene el control de que recursos están disponibles en Internet, no existiendo por ejemplo, un directorio de todos los recursos disponibles. Algunas organizaciones, por el contrario, mantienen directorios personalizados que contienen algunos de

---

<sup>1</sup> La variación que pueda sufrir internamente un recurso a lo largo del tiempo no impacta en el dinamismo del espacio de información. Según [Blaze 93] la probabilidad de que un recurso vuelva a modificarse va disminuyendo a medida que transcurre el tiempo desde su última modificación.

los recursos disponibles en las redes bajo su administración. Cada directorio tiene su propia semántica, organización y alcance. Como resultado, juntar estos pequeños sistemas de información provenientes de diferentes organizaciones en un único sistema para el descubrimiento de recursos no es una tarea trivial y tal vez nunca se logre.

En segundo lugar, el gran dinamismo y la amplia distribución del espacio de información dificultan la tarea de rastrear los recursos existentes. La permanente incorporación y eliminación de los recursos hace que los RDS deban verificar en forma periódica la validez de la información que mantienen, actividad que se dificulta dada la dispersión geográfica de los servidores de información.

## 2.2 Escalabilidad

Un entorno de gran escala como el de Internet impone requerimientos de escalabilidad en los protocolos que se utilicen para la localización y búsqueda de recursos. Para analizar dichos requerimientos, es necesario estudiar el crecimiento de Internet, de manera tal de poder identificar los puntos sobre los que hay que trabajar.

Las propiedades de escalabilidad de un RDS deben estudiarse analizando como impactan, en la arquitectura que lo implementa, las condiciones de entorno que debe manejar. Según como se adecue a estas condiciones, el sistema será o no escalable.

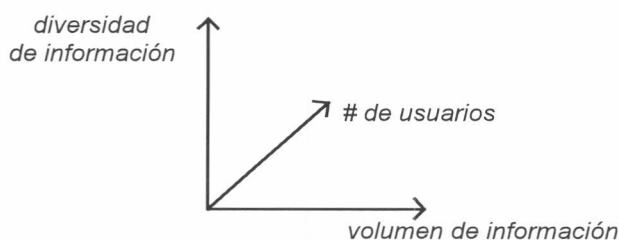


Gráfico 2.1 Dimensiones de escalabilidad

En [Bowman 94] se presenta un marco de trabajo para estudiar el impacto de la escalabilidad en los RDS, caracterizándola en tres dimensiones: volumen de información, cantidad de usuarios y diversidad de información.

### 2.2.1 Volumen de Información

Internet alberga una vasta colección de servidores que contienen recursos. Puede verse como una gran base de datos distribuida geográficamente que contiene un significativo volumen de información que crece día a día con la incorporación de nuevos servidores y recursos.

La *base de información* de un RDS es el conjunto de recursos de información que éste incluye en su alcance. Los RDS de *propósito general* incluyen en su base de información variados recursos

(páginas Web, archivos de repositorios FTP, etc.), mientras que los de *propósito específico* sólo incluyen determinados tipos de recursos, por ejemplo grupos de noticias. El creciente volumen de información hace que los RDS deban manejar bases de información cada vez más grandes.

Para que un RDS sea útil a la comunidad de usuarios, debe permitirle interactuar de una manera amigable con el creciente volumen de información con el objetivo de que no se vean sobrecargados con más información de la que puedan manejar.

Los paradigmas más utilizados para definir la interacción de los usuarios con un RDS, son la *navegación* y la *búsqueda* [Obraczka 93] [Bowman 94]. Analizaremos las ventajas y desventajas de cada uno.

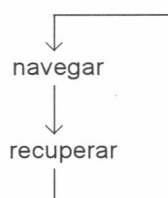


Gráfico 2.2 Navegar

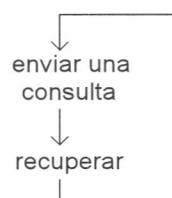


Gráfico 2.3 Buscar

La **navegación** se refiere a la actividad guiada por un ser humano de explorar los contenidos de un espacio de información. Un sistema de navegación provee la funcionalidad del Gráfico 2.2. La efectividad de la navegación depende del grado de organización de la información, actividad que generalmente realiza un ser humano al decidir como se interrelaciona la información, por ejemplo ubicándola en algún tipo de grafo dirigido como una jerarquía de directorios en un servidor FTP o una estructura de hipervínculos en la Web. Sistemas como el Gopher [Alberti 92] y la WWW [Berners-Lee 92] pertenecen a esta categoría.

La **búsqueda** en cambio, es un proceso automatizado donde el usuario expresa sus necesidades formulando una consulta al sistema para que este último encuentre los recursos de red que mejor la satisfagan. Generalmente los resultados se presentan utilizando algún ranking por relevancia. A partir de estos resultados el usuario puede seleccionar un recurso para recuperarlo. Un sistema de búsqueda provee la funcionalidad del Gráfico 2.3. Sistemas como Altavista [AV] y Excite [EX] entre otros, pertenecen a esta categoría.

Algunos de los RDS de propósito general combinan ambos paradigmas proveyendo funcionalidades de búsqueda y navegación. A modo de ejemplo podemos citar el Yahoo [YH] que organiza los recursos en una jerárquica ad-hoc para que los usuarios puedan navegarla y provee facilidades de búsqueda sobre el índice total, como si la jerarquía fuera plana.

La efectividad de la *navegación* depende en gran medida del grado de organización del espacio de información. Mantener un gran volumen de información organizado es una tarea difícil que se agrava aún más si tenemos en cuenta la administración descentralizada. Esto trae como consecuencia que la actividad de navegación resulte más difícil y tediosa para los usuarios. Estos deben invertir mucho tiempo para encontrar lo que buscan y aún así muchas veces no lo logran.



Como consecuencia de los problemas de escalabilidad que presenta el paradigma de navegación, es que los sistemas de búsqueda se han vuelto muy populares dentro de la comunidad de usuarios, siendo actualmente los más utilizados.

## 2.2.2 Cantidad de Usuarios

Cada día nuevos usuarios se incorporan a Internet convirtiéndose en potenciales usuarios de sistemas de información. Los RDS deben tener en cuenta este crecimiento para poder manejar de una manera eficiente el elevado número de requerimientos que los usuarios puedan generar.

Los repositorios de información centralizados se vuelve un cuello de botella cuando el conjunto de los usuarios distribuidos en Internet demandan un acceso eficiente a la información [Bowman 94]. Cualquier RDS debe contemplar un elevado número de requerimientos de usuarios distribuidos geográficamente para no verse saturado (sobrecarga en sus servidores y enlaces).

La replicación [Righetti 97] [Obraczka 94] y el caching [Luotonen 94] son algunas de las técnicas utilizadas para proveer a los usuarios un acceso más eficiente a la información a la vez que minimizan el tráfico en la red.

Otra consecuencia del crecimiento del número de usuarios es la diversificación de intereses que se genera. Esto hace que los RDS de propósito general tengan que incluir en sus alcances recursos muy variados para poder satisfacer las necesidades de todos los usuarios, imponiendo un mayor requerimiento de recursos informáticos para el sitio.

Dada la amplitud y diversidad de intereses, creemos que con el paso del tiempo los RDS de propósito general irán desapareciendo para ser reemplazados por sistemas de propósito específico, de manera tal de dar una mejor solución a una comunidad más pequeña de usuarios.

## 2.2.3 Diversidad de información

El espacio de información en Internet, a parte de su gran volumen y amplia distribución, tiene la característica de utilizar variados tipos de representación para los recursos. Esto se debe a que existen diversos sistemas de información que utilizan su propia forma de representar a los recursos que manejan. Generalmente la extensión asociada al nombre de un archivo identifica el tipo de representación utilizado, por ejemplo los archivos con extensión .ps están en formato PostScript, los .rtf en formato Rich Text Format, los .html en HyperText Markup Language, los .txt en formato ascii (otros formatos se identifican con extensiones .doc, .xls, .tar, .zip, .gz, .wav, .mpeg, .gif, .tif, etc.). A su vez todos estos tipos de recursos pueden ser recuperados por distintos sistemas, como el Web, FTP entre otros.

Aquellos RDS que abarquen múltiples contenidos deberán ser capaces de manejar estas diferencias. A modo de ejemplo, serán necesarias técnicas automáticas para interpretar y clasificar una variedad de recursos diferentes.



## 2.3 ODS

Presentaremos las características del Sistema de Distribución de Objetos(ODS) [Righetti 97], analizando las similitudes y diferencias que presenta con el sistema News [RFC 1036], por ser éste más conocido.

### 2.3.1 Características

El *Sistema de Distribución de Objetos* es un sistema ideado para disminuir el tráfico en redes de área extendida, incrementar la disponibilidad e implementar mecanismos de clasificación de objetos<sup>1</sup> [Righetti 97]. Su principal objetivo es la distribución selectiva de objetos a usuarios distribuidos geográficamente, mediante la utilización de mecanismos de replicación de consistencia débil [Obraczka 94] [Neuman 94].

A grandes rasgos, podemos decir que ODS permite a los usuarios la publicación, clasificación y suscripción a volúmenes de objetos. Los autores envían los objetos que producen a los agentes de publicación. Autoridades de clasificación proveen esquemas de clasificación útiles para clasificar los objetos producidos. Los lectores se suscriben a volúmenes de objetos en un Agente de Servicios (SA) para acceder a copias locales de los objetos producidos globalmente (como en una biblioteca, donde los lectores tienen acceso a los distintos ejemplares disponibles).

Se encuentra formado por dos componentes:

- **Red de Distribución de Objetos (ODN)**
- **Red de Ruteo de Objetos (ORN)**

Se define una *Red de Distribución de Objetos (ODN)* como un conjunto de *Agentes de Servicio (SA)* que cooperan con el fin de replicar objetos a usuarios distribuidos geográficamente, sin que estos deban preocuparse por el origen o la actualidad de los mismos. Los usuarios acceden a los objetos a través del SA que les brinda el servicio y pueden eventualmente registrar allí los objetos que deseen distribuir a través de ODN.

Un SA podría ser un host dentro de un campus universitario, mientras que los usuarios que lo utilicen pueden estar en cualquier sitio vinculados mediante una LAN o WAN.

Cada objeto de ODS se identifica en forma única, pudiendo ser modificado o eliminado sólo por sus dueños. Los atributos asociados a un objeto solo son modificables en el SA donde fuera registrado, denominado *homeSA*.

---

<sup>1</sup> Los objetos de ODS serán considerados recursos disponibles para el servicio de búsqueda.

El espacio de información de ODS está conformado por objetos *clasificados* en el sentido que son validados por autoridades de clasificación. Estas autoridades de clasificación son las encargadas de la definición y generación de *esquemas de clasificación*. Un esquema de clasificación define un conjunto de etiquetas, labels o tópicos utilizados por los autores o “publicadores” para describir los objetos y por los “lectores” para seleccionar la información de su interés.

Los objetos se componen de dos partes: *meta-información* y *cuerpo*.

La *meta-información* de los objetos está definida por su URC (*Uniform Resource Characteristic*) [Baleani 98]. Un URC describe las características de un objeto utilizando un conjunto de pares atributo/valor como: versión, labels de clasificación, autor, tema, entre otros. El objetivo de un URC es describir el objeto al cual está asociado. El *cuerpo* es el objeto propiamente dicho.

Un conjunto de atributos define un *grupo de interés*. Los SA unidos por un interés común definen un grupo. Los SA poseen distintos roles dentro de cada grupo según sean consumidores y/o productores del mismo. El rol de un SA está determinado por la suscripción de sus usuarios a los grupos de interés. Dentro de cada grupo se mantiene una cadena de distribución que asegura que todos los SA consumidores del grupo recibirán los objetos que se produzcan en éste.

La *Red de Ruteo de Objetos (ORN)* es la componente encargada de construir dinámicamente las cadenas de distribución entre los SA que conforman ODN, haciendo que el sistema escale cuando el número de productores y consumidores crezca considerablemente. La componen un conjunto de *Agentes de Ruteo (RA)* a los cuales los SA notifican sus roles de productor y/o consumidor para uno o más grupos de interés. Estos roles son utilizados por los RA para armar las cadenas de distribución.

Los protocolos de ruteo utilizados por los RA para la replicación de objetos tienen la importante característica de ser *adaptativos*. Tienen en cuenta el estado de la red subyacente, pudiendo modificar la composición de las cadenas de distribución en forma dinámica de manera tal que todas las réplicas converjan a un estado consistente aunque existan particiones de red.

De esta manera la distribución de un objeto comienza en el SA donde éste se encuentre registrado (homeSA, SA productor). En función de la cadena de distribución, dicho SA lo ofrece a sus consumidores y estos, a su vez, hacen lo mismo con los suyos.

### 2.3.2 Comparación con News

ODS es similar al sistema News [RFC 1036] (ver “Apéndice B”) en cuanto al tipo de servicios que provee a su comunidad de usuarios. Ambos conforman una red de nodos que cooperan con el fin de distribuir a la comunidad de usuarios objetos producidos globalmente (newsgroups en News, objetos no específicos en ODS). Estos nodos se encuentran ampliamente distribuidos en Internet con el objetivo de cubrir la demanda de muchos usuarios distribuidos geográficamente, permitiéndoles acceder a copias locales de objetos producidos globalmente.

La información se organiza en un conjunto preestablecido de *grupos de interés*. Cualquier objeto que se produce se hace para un grupo de interés determinado. Para recibir los objetos que se producen a nivel global, cada nodo consume uno o más grupos de interés.

Los usuarios acceden a copias locales de los objetos producidos globalmente conectándose al nodo que les provee el servicio y seleccionando el grupo de interés deseado. Para muchos de los grupos, los usuarios pueden publicar objetos y registrarlos en el nodo local para que la red los distribuya a todos los nodos consumidores de esos grupos.

En News, la mayoría de los grupos de interés son públicos, permitiéndose a cualquier usuario publicar objetos para el grupo. No existen autoridades de clasificación que determinen que se publica. La validez de lo que se produce queda supeditada al criterio de los usuarios.

A diferencia de News, ODS fue concebido entre otros, para distribuir objetos clasificados. Solo los usuarios autorizados pueden publicar objetos y en caso de hacerlo, éstos deben estar previamente clasificados siguiendo los esquemas de clasificación que se hayan definido. Esto podría compararse en News con el uso de newsgroups moderados, en los cuales existe un moderador encargado de autorizar la información que se publica.

En ODS los objetos son *persistentes* y se mantienen replicados en todos los SA interesados como en una biblioteca, pudiendo ser modificados y/o eliminados solo en los SA donde fueron registrados (*homeSA*). En cambio en News, los administradores de cada nodo del sistema son quienes deciden el tiempo de vida de los objetos mantenidos. En general el tiempo de vida se limita a unos pocos días o semanas. Una vez que dicho tiempo de vida expira, los objetos son eliminados del nodo.

El hecho de que ODS maneje objetos persistentes (como en una biblioteca) tiene la importante ventaja de que los SA consumidores de un grupo son realmente réplicas completas de los contenidos del grupo, cosa que no sucede en News. En éste último distintos nodos consumidores de un mismo grupo de interés pueden mantener contenidos diferentes.

Una última diferencia, que consideramos la más importante, tiene que ver con las propiedades de escalabilidad de ambos sistemas. Tanto News como ODS construyen cadenas de distribución entre los nodos que conforman la red utilizando como unidad el grupo de interés. La diferencia fundamental entre ambos es la manera en la cual construyen las cadenas de distribución.

News utiliza cadenas de distribución estáticas donde la relación entre dos nodos es una relación proveedor-consumidor. El nodo consumidor acuerda con el nodo proveedor cuales de los newsgroups disponibles va a consumir. Este acuerdo lo realizan los administradores de los nodos y es configurado manualmente en los servidores. Los usuarios solo pueden acceder a los grupos que provee el nodo al cual se conectan. La relación proveedor-consumidor entre los nodos es siempre la misma (salvo que el administrador modifique la configuración del sitio). Esto trae como consecuencia que ante particiones de red algunos sitios no reciban información pudiendo quedar desactualizados por amplios períodos de tiempo.

En cambio en ODS, las cadenas de distribución son dinámicas. La composición de las cadenas depende en todo momento de los intereses de los usuarios y de la información disponible, haciendo que dos nodos cooperen solo si tienen un grupo de interés en común. La relación proveedor-consumidor la definen los Agentes de Ruteo en base a los roles de los Agentes de Servicio (productor y/o consumidor) teniendo en cuenta el estado de la red subyacente. Esto garantiza que ante particiones de red las cadenas se reconfiguren dinámicamente, haciendo que las réplicas converjan a un estado consistente.

A medida que crece el número de réplicas y su grado de distribución en Internet, reconfigurar manualmente los sitios es prácticamente imposible. En News son los administradores de los sitios los que deben reconfigurar la topología de distribución lógica entre los servidores a partir de la cual se actualizan las distintas réplicas. Estas modificaciones deben darse con frecuencia para adecuarse a los cambios dinámicos que se dan en la topología física de Internet. Esta actividad se dificulta a medida que la escala y complejidad de Internet aumenta, ya que las particiones de red se dan con mayor frecuencia.

ODS es más escalable que News. Esto se debe a que en ODS, los cambios en la topología física de Internet se reflejan automáticamente en la topología de distribución lógica entre los nodos. Esto se da gracias a la utilización de protocolos de ruteo de nivel de aplicación que tienen en cuenta el estado de la red subyacente para confeccionar las cadenas de distribución [Obraczka 94].

## Capítulo 3

# Tipos de arquitecturas

Describiremos funcionalmente la arquitectura de un servicio de búsqueda. Clasificaremos los servicios de búsqueda según su arquitectura. Caracterizaremos las arquitecturas centralizadas y distribuidas con el objetivo de determinar sus propiedades de escalabilidad, es decir como se comportan en un entorno caracterizado por un gran volumen de información, una administración descentralizada, y gran cantidad de servidores de información y usuarios distribuidos geográficamente.

### 3.1 Componentes y clasificación

La estructura interna de cualquier servicio de búsqueda está conformada por un número de componentes o subsistemas funcionalmente diferentes, invisibles al usuario final. Estos componentes son tres: subsistema de recolección, subsistema de indexación y subsistema de búsqueda y recuperación [Neuman 93] [Koch 96].

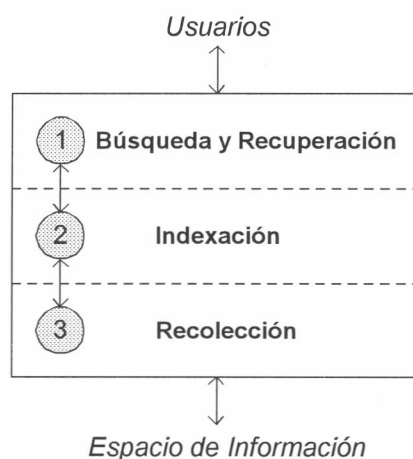


Gráfico 3.1 Componentes de un servicio de búsqueda

#### *Búsqueda y Recuperación*

Es el encargado de resolver consultas para los usuarios del servicio. Provee la interfaz de consulta, a través de la cual los usuarios especifican sus requerimientos al sistema. Procesa

las consultas, recuperando los recursos que mejor la satisfagan para devolver los resultados, generalmente utilizando algún ranking por relevancia para ordenarlos.

### *Indexación*

Es el encargado de construir un índice de manera tal de organizar la información disponible para su fácil búsqueda y recuperación.

### *Recolección*

Es el encargado de determinar qué recursos de red serán alcanzables desde el servicio de búsqueda. Se concentra en la recolección de datos.

Como se observa en el Gráfico 3.1, el subsistema de nivel superior es el más próximo a los usuarios finales del servicio, mientras que el de nivel inferior interactúa con el espacio de información subyacente.

La indexación es importante para cualquier servicio de búsqueda. Las consultas se resuelven utilizando una o más bases de datos con índices asociados a los efectos de organizar la información y así optimizar las búsquedas. En una implementación dada el índice físico puede no existir, pero dado que la indexación constituye un importante modelo de abstracción, de ahora en adelante hablaremos de índices como el modelo de datos de un servicio de búsqueda. Existen diversas formas de construir un índice, así como existen diferentes tipos de índices y dado un índice, pueden existir diferentes formas de utilizarlo para resolver consultas.

Los servicios de búsqueda se clasifican en centralizados y distribuidos dependiendo de la arquitectura de indexación y búsqueda que utilicen [W3C 96] [Weider 96] [Denenberg 96].

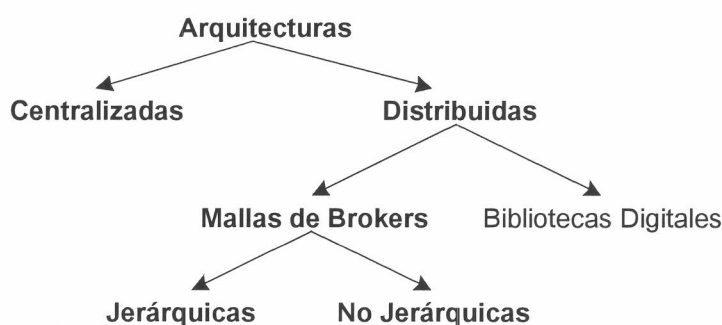


Gráfico 3.2 Clasificación de Arquitecturas

Un servicio de búsqueda es **centralizado** si utiliza una arquitectura caracterizada por usar técnicas monolíticas de búsqueda e indexación, donde un único sitio es el encargado de recuperar e indexar toda la información disponible.

Por el contrario, un servicio de búsqueda es **distribuido** si utiliza una arquitectura caracterizada por distribuir el índice en varios servidores<sup>1</sup> e implementar mecanismos de acceso a los índices para la resolución de las consultas.

La *escalabilidad* de la arquitectura de un servicio de búsqueda tiene que ver fundamentalmente con el manejo de recursos. Una **arquitectura escalable** es aquella que puede crecer sin requerir más recursos de los disponibles en cualquier parte del sistema [Sheldon 95a]. A modo de ejemplo, cuando aumenta el volumen de información disponible y la cantidad de usuarios, la arquitectura debe ser capaz de adaptarse con un aumento mínimo de costo, degradación de performance y complejidad administrativa. El manejo de recursos abarca: la capacidad de procesamiento, el uso del ancho de banda de los enlaces, la capacidad de almacenamiento, la complejidad administrativa, etc.

## 3.2 Arquitecturas centralizadas

La arquitectura de búsqueda e indexación más fácil de implementar, no por ello la mejor, consiste en centralizar en un único punto toda la información. De esta manera una consulta puede resolverse fácilmente accediendo a un único repositorio, no requiriéndose de ningún procesamiento distribuido.

Las arquitecturas centralizadas se caracterizan por implementar una estrategia de indexación global, siendo su principal objetivo la construcción de un gran índice para la mayor cantidad posible de recursos, que denominaremos de ahora en adelante *índice global*.

En la actualidad de Internet encontramos numerosos servicios de búsqueda, denominados comunmente *buscadores*, que proveen a la comunidad de usuarios facilidades de búsqueda y localización de recursos de interés. Aunque tengan entre sí algunas diferencias de implementación, todos proveen una solución centralizada.

Entre los ejemplos que podemos mencionar se encuentran los servicios de búsqueda de propósito general basados en robots: Excite [EX], Altavista [AV], Lycos [LY] que recolectan información sobre variados recursos (páginas Web, grupos de noticias, listas de discusión, etc.) y proveen una interfaz de consulta para su recuperación.

La misma estrategia es utilizada por los sistemas de propósito específico VERONICA [Hahn 96], Archie [Entage 92] y DejaNews [DN]. VERONICA es un servicio que mantiene un índice de los títulos de los menús Gopher [Alberti 92]; Archie barre los servidores FTP dispersos en Internet e indexa los nombres de los archivos encontrados; DejaNews [DN] es un servicio específico para News [RFC 1036] que indexa los documentos que circulan por los grupos de noticias.

---

<sup>1</sup> No estamos considerando una red de workstations u otras arquitecturas paralelas, donde el índice está distribuido en un conjunto de sistemas altamente acoplados sobre una red de alta velocidad como una LAN.



En [Koch 96] se expone una exhaustiva revisión de los servicios de búsqueda basados en robots, así como de las herramientas de software disponibles para la recuperación, recolección e indexación.

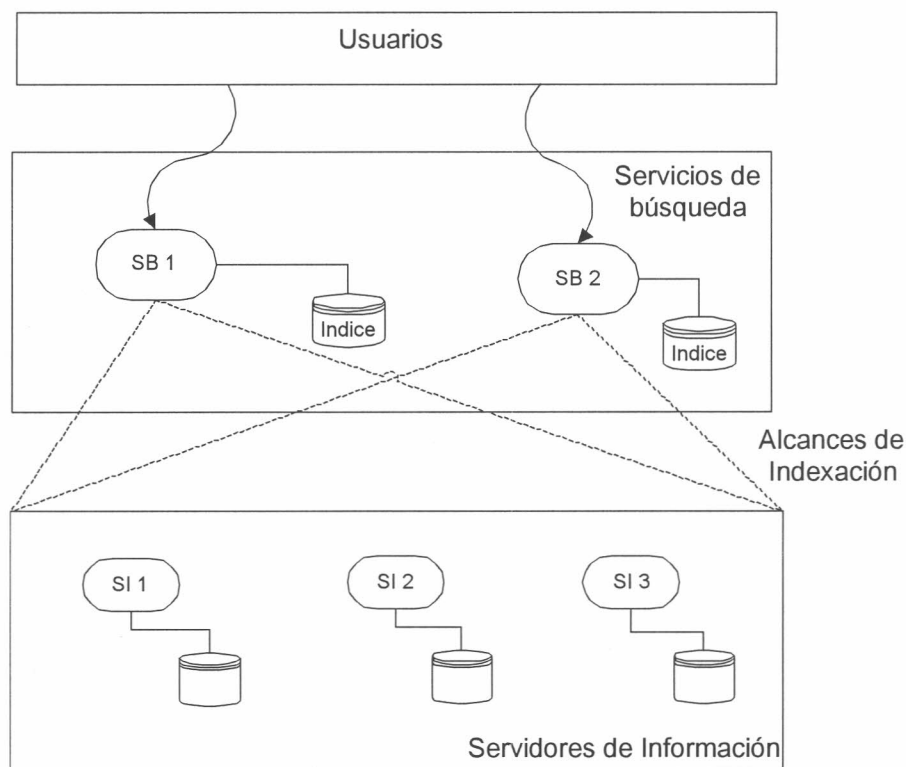


Gráfico 3.3 Arquitectura de indexación global centralizada

En el Gráfico 3.3 se observa el enfoque de una arquitectura de indexación y búsqueda centralizada. El nivel de los servidores de información representa el espacio de recursos a ser indexado. El nivel de los servicios de búsqueda representa a los servicios centralizados que tienen como alcance de indexación la totalidad del espacio de información<sup>1</sup>. El nivel de los usuarios representa a la comunidad de usuarios que acceden y utilizan los servicios de búsqueda.

El espacio de recursos a abarcar se determina mediante la aplicación de estrategias de recolección. Estas definen cuales servidores serán alcanzados y que tipo de recursos serán recolectados. Las estrategias aplicadas varían de un servicio a otro. Generalmente combinan una especificación ad-hoc hecha por los administradores del servicio con una registración manual de sitios hecha por los proveedores interesados en que su información sea accesible. La mayoría de los servicios son de propósito general dado que indexan variados recursos: páginas Web, archivos de repositorios FTP, etc.

Los *robots* o *spiders* [Koster 95] son programas encargados de aplicar las estrategias de recolección que se hayan definido. Estos programas son los más utilizados puesto que permiten

<sup>1</sup> Intentan abarcar la totalidad del espacio de información, prácticamente imposible.



automatizar el proceso de recolección al no requerir ningún tipo de intervención humana, a excepción de cierto conocimiento inicial.

La mayoría de los servicios centralizados implementan la recolección de datos mediante robots motivo por el cual también se los denomina *índices basados en robots* [Koch 96].

Una vez que los recursos completos han sido recolectados, el subsistema de indexación se encarga de procesarlos antes de su incorporación al índice global. La indexación es el proceso encargado de *caracterizar* cada recurso mediante la extracción de una descripción que lo identifique, usualmente conocida como *meta-información del recurso* [Marshall 98] o *surrogate* [Lagoze 97].

La diferencia entre un esquema de indexación y otro es la meta-información que producen para los recursos. Los esquemas utilizados por los servicios de búsqueda varían y es uno de los componentes que diferencia las posibilidades de búsqueda que ofrece cada uno. Algunos servicios como Archie indexan solo nombres de archivos, mientras que sistemas como Altavista y Excite generan índices de texto completo.

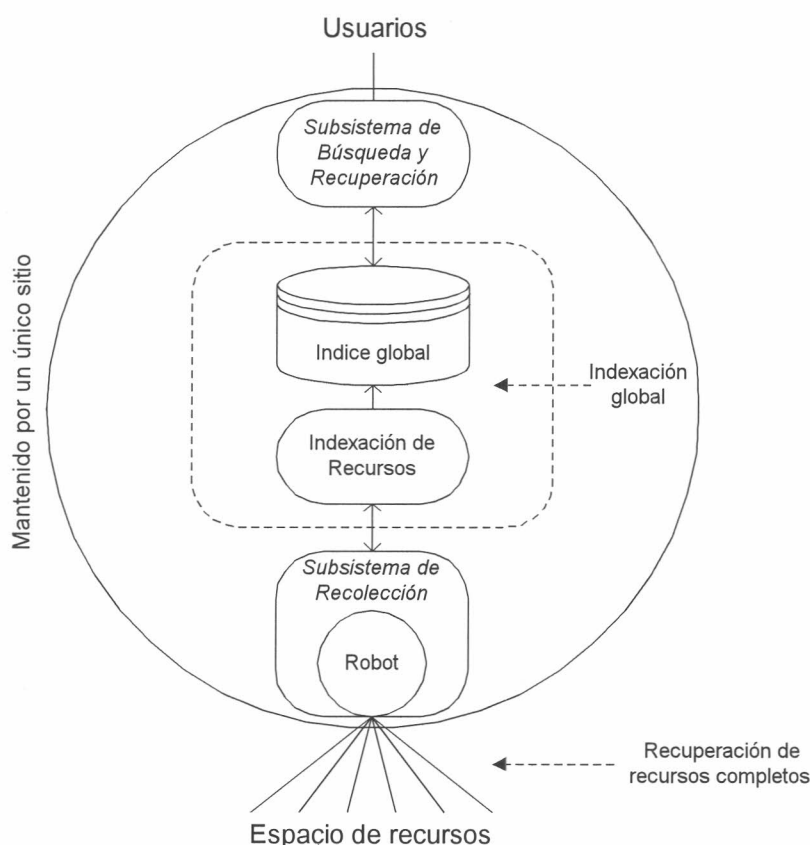


Gráfico 3.4 Estructura interna de un servicio centralizado

Las técnicas de búsqueda y recuperación de recursos (*information retrieval*) se utilizan para recuperar del índice global los recursos que satisfagan una necesidad de información expresada al sistema mediante una consulta.

El Gráfico 3.4 muestra la interacción de los componentes de un servicio de búsqueda centralizado.

El buscador DejaNews [DN] es un servicio de búsqueda específico para el sistema News. El espacio de información que maneja se restringe únicamente a grupos de noticias. Difiere del resto en cuanto a como implementa la recolección de la información que mantiene en su índice. No utiliza robots, sinó que se comporta como un servidor de news más dentro de la red global de servidores consumiendo la mayor cantidad posible de grupos de noticias de entre los disponibles. De esta manera se utilizan los mecanismos de replicación propios de News para la recolección de información. El Gráfico 3.5 muestra como el servicio DejaNews hace uso de la red de News para la recolección de información.

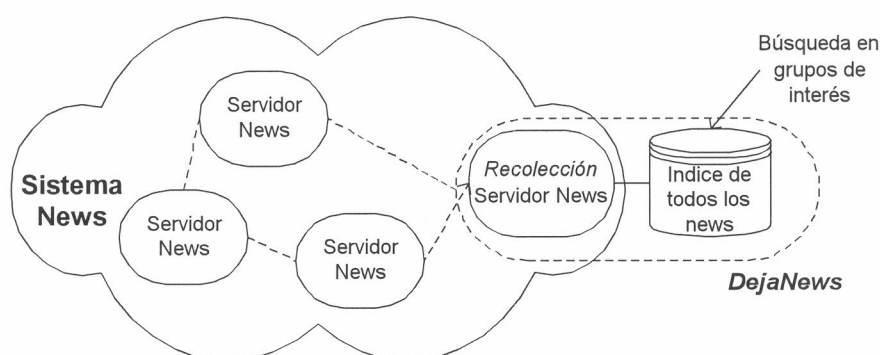


Gráfico 3.5 Buscador DejaNews

Aquí encontramos una diferencia importante entre DejaNews y los buscadores de propósito general. DejaNews simplifica el proceso de adquisición del conocimiento de los recursos de red existentes que es uno de los procesos más complejos involucrados en la construcción de servicios de búsqueda.

### 3.2.1 Problemas de escalabilidad

Analizaremos como un entorno de gran escala, como es Internet, impacta en el comportamiento de los servicios descritos con el objetivo de determinar las propiedades de escalabilidad de las arquitecturas centralizadas como solución para la búsqueda de información.

El primer punto a analizar, es el impacto que provoca el espacio de información a tratar. Espacio caracterizado principalmente por el crecimiento exponencial del volumen de información accesible, por su amplia distribución y por el gran dinamismo generado por la permanente incorporación y eliminación de recursos.

El objetivo de todo servicio centralizado es reunir la mayor cantidad de información disponible, es por esto que la recolección de datos es uno de los procesos más críticos.

La necesidad de transmitir grandes volúmenes de información desde los sitios proveedores de información hasta el servidor generará excesivo tráfico en la red independientemente de como se implemente el proceso de recolección de datos. Este tráfico, si se quiere mantener la coherencia

de los datos almacenados en el índice (*up-to-date*), reducirá considerablemente el ancho de banda disponible para otras actividades.

En su gran mayoría, los servicios centralizados existentes en Internet utilizan robots para el proceso de recolección, a pesar de ser considerados ineficientes y dañinos [Koster 95] [Sassi 98]. El motivo principal por el que siguen utilizándose es su fácil implementación y puesta en marcha.

Las implementaciones más sencillas no requieren de ningún tipo de coordinación entre las partes involucradas (proveedores de información y servicio de búsqueda). La utilización de otros métodos para la recolección requerirían, por ejemplo, la definición de estándares para el intercambio de información de indexación, más difíciles de implementar dada la administración descentralizada que caracteriza el entorno de estudio [W3C 96].

Los robots utilizan en su gran mayoría, protocolos estándares de transferencia como HTTP y FTP para extraer recursos completos desde los proveedores de información. Estos métodos generan un exceso de trabajo en los sitios proveedores de información ya que éstos deben dedicar gran parte del tiempo a satisfacer los requerimientos de estos programas [Bowman 95] [Sassi 98].

Otra importante consecuencia del gran volumen de información es la capacidad de procesamiento y la cantidad de recursos informáticos que necesita el sitio proveedor del servicio para mantener el índice global.

El tamaño del índice depende de la técnica de indexación que aplique cada sistema. Estas suelen focalizarse en un espectro que va desde el espacio de almacenamiento requerido hasta el grado de representatividad deseado para los datos.

Los sistemas que hacen más hincapié en minimizar el espacio de almacenamiento necesario representan a los objetos con la mínima información posible, por ejemplo solo nombres de archivos en el sistema Archie, permitiéndoles almacenar mayor cantidad de objetos pero satisfacer consultas muy limitadas. Los sistemas que utilizan técnicas de indexación de texto completo, como Altavista y Excite, pueden satisfacer consultas más complejas pero presentan problemas de escalabilidad. Los índices de texto completo representan a cada objeto por todas sus palabras. Generalmente son índices invertidos que almacenan punteros a cada ocurrencia de cada palabra. El principal problema que presentan es el gran tamaño del índice final, el tiempo requerido para su construcción y/o actualización y el espacio de almacenamiento necesario.

Más allá de la técnica de indexación que se utilice, la curva de crecimiento del índice global será similar a la del crecimiento del volumen de información producido [Bowman 94].

La administración y el mantenimiento de índices de tal envergadura es un proceso costoso, potenciado por el hecho de que un alto porcentaje de los datos mantenidos son raramente consultados por los usuarios. Esto se debe a que mucha de la información producida (en la Web,

en News, etc.) es ruido<sup>1</sup>, por lo que solo un pequeño porcentaje es relevante para un elevado número de los usuarios.

Los índices monolíticos, al ser grandes y heterogéneos dificultan el proceso de recuperación de información (*information retrieval*). Las tecnologías de *information retrieval* disponibles ven disminuida su efectividad a medida que crece el tamaño del índice y aumenta la diversidad de recursos, recuperando muchos documentos irrelevantes [Lagoze 97].

El segundo punto a analizar, es el impacto provocado por el creciente número de usuarios del servicio.

Los servicios centralizados se vuelven un cuello de botella y un único punto de falla frente al crecimiento del número de usuarios del servicio. Es difícil que un sistema centralizado no colapse cuando la cantidad de consultas a procesar aumente, ya que siempre deben resolverse accediendo al índice global.

Para minimizar estos problemas, muchos servicios (por ej. Archie) replican sus índices con el objetivo de distribuir el procesamiento de las consultas. Pero la replicación masiva de los índices es un proceso administrativamente costoso. La necesidad de mantener consistente las réplicas requiere que cada sitio disponga de los recursos informáticos necesarios. El hecho de replicar porciones del índice que tal vez nunca sean consultadas genera un costo adicional innecesario [Bowman 94].

En el caso particular de Internet, los problemas se agravan aún más. Existen múltiples servicios de búsqueda centralizados mantenidos por entidades diferentes. Cada uno trabaja en forma aislada, sin ningún tipo de cooperación entre sí. Por consiguiente, se duplican los esfuerzos en la tarea de rastrear los recursos disponibles y aumenta la cantidad total de recursos informáticos utilizados.

Los problemas detectados en los sistemas descritos, nos permite concluir que en entornos de las características de Internet, los servicios de búsqueda basados en una arquitectura centralizada constituyen soluciones difícilmente escalables [Bowman 94] [Sheldon 95a] [Weider 96]. Esto se debe a la necesidad permanente que tienen de incrementar la cantidad de recursos informáticos necesarios como ser: espacio en disco, capacidad de procesamiento, cantidad de servidores dedicados, etc., para no colapsar y poder contener y manejar adecuadamente las cambiantes condiciones de entorno.

Su falencia más importante radica en el modelo que guía su diseño: manejar un entorno ampliamente distribuido, en constante crecimiento y cambio mediante la centralización de información.

---

<sup>1</sup> Información no relevante para un usuario o comunidad de usuarios.

### 3.3 Arquitecturas distribuidas

En la actualidad, la mayoría de los servicios de búsqueda en Internet se basan en una arquitectura centralizada. Este enfoque, tal como analizáramos, presenta problemas de escalabilidad tanto para los servidores de información como para los mismos servicios de búsqueda. La actual línea de investigación se concentra en el diseño y estudio de arquitecturas distribuidas de búsqueda e indexación para la construcción de servicios de búsqueda [W3C 96] [Weider 96].

Una arquitectura de búsqueda e indexación distribuida es aquella en la cual el índice en sí mismo está distribuido en varios servidores y se implementan mecanismos de comunicación para la resolución de las consultas. No estamos considerando una red de workstations u otras arquitecturas paralelas, donde el índice está distribuido en un conjunto de sistemas altamente acoplados sobre una red de alta velocidad como una LAN.

En un índice distribuido la información de indexación ya no se encuentra más en un único servidor como en los servicios centralizados, sino que está distribuida en varios sitios. Una búsqueda en un índice de estas características requiere que la misma sea *lógicamente* ejecutada en cada uno de los servidores y que los resultados se combinen de una manera adecuada para presentárselos al usuario [Viles 95].

La pregunta que surge es: ¿cómo pueden resolverse las consultas de una manera eficiente y escalable, teniendo en cuenta que los índices pueden ser numerosos y estar ampliamente distribuidos en Internet?. Serán necesarios protocolos para el intercambio de información de indexación o mejor aún protocolos para la búsqueda distribuida.

La estrategia más intuitiva que se podría utilizar es reenviar (*forward*) la consulta a todos los servidores existentes, combinar los resultados y devolverlos al usuario como se muestra en el Gráfico 3.6. Puede automatizarse de una manera sencilla a través de un mecanismo de red como el broadcast, garantizando encontrar la información que se busca (ya que se estarían contactando *todos* los servidores). Sin embargo el broadcast requerirá muchos recursos de red y tendrá problemas de performance a medida que la cantidad de usuarios y de servidores crezca, no constituyendo una solución escalable [Neuman 94]. El impacto provocado por muchos usuarios haciendo *broadcast* de sus consultas sobrecargaría rápidamente los servidores y enlaces, haciendo que el sistema colapse.

Los servicios conocidos con el nombre de meta-buscadores (*meta-search*) utilizan este tipo de estrategia de resolución evitando a los usuarios tener que emitir la misma consulta a más de un servicio de búsqueda. El WebCrawler [WC] es un ejemplo de meta-buscador. También se los conoce como servicios de búsqueda *simultáneos* [Koch 96] puesto que reenvían la consulta simultáneamente a diferentes servicios de búsqueda, por ejemplo a los centralizados, luego de lo cual procesan las respuestas eliminando duplicados para devolver los resultados al usuario.

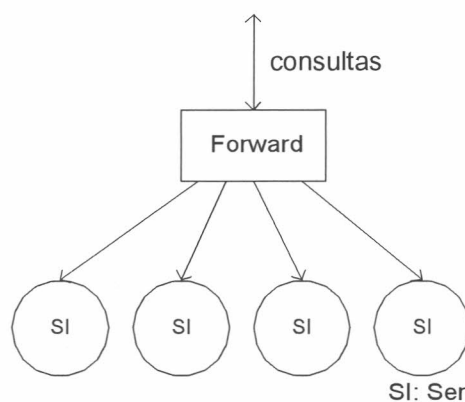


Gráfico 3.6 Broadcast Global

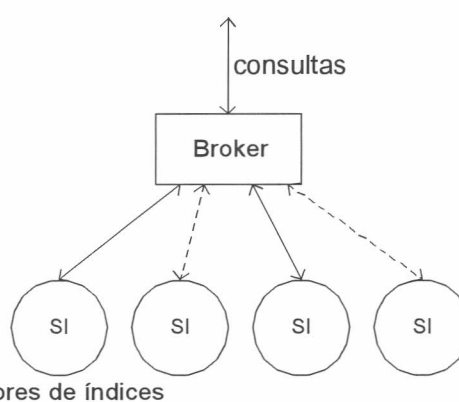


Gráfico 3.7 Query routing a un subconjunto

Una estrategia de resolución más eficiente que el broadcast global sería un servicio a través del cual una consulta se resolviera reenviándola *solo* al subconjunto de servidores que se consideren más relevantes para satisfacerla, como se observa en el Gráfico 3.7. De esta manera se estaría disminuyendo el conjunto de servidores a contactar, haciendo que la estrategia de resolución sea más escalable.

Se denomina *Query Routing* al proceso de seleccionar, del conjunto de servidores disponibles, el subconjunto que mejor pueda satisfacer/responder una consulta, enviarles la consulta y combinar los resultados [Sheldon 95a] [RFC 1913] [Xu 98]. El principal objetivo de una estrategia de *query routing* es **minimizar** el número de servidores a los cuales reenviar una consulta.

El proceso de selección de servidores requerirá algún conocimiento que describa los contenidos mantenidos por cada posible servidor (*meta-información de contenido*). Del estudio/evaluación de dichas descripciones con respecto a una consulta, surgirán los servidores más valiosos a los cuales habrá que derivarla para su resolución.



Gráfico 3.8 Etapas del query routing

En el Gráfico 3.8 se observan las tres etapas del query routing: selección de los servidores, evaluación de la consulta y combinación de resultados. En la etapa de selección se eligen los mejores servidores para evaluar la consulta basándose en la meta-información de contenido de éstos. Luego la consulta es enviada a dichos servidores para su evaluación. Por último, los resultados obtenidos desde los distintos servidores se combinan obteniéndose la respuesta final a enviar al usuario.

La estructura de meta-información de contenido dependerá de los servicios a proveer y de la estrategia de query routing que utilice el sistema. El proceso utilizado por un servidor para generar la meta-información de contenido de su colección de recursos se denomina *proceso de sumarización*, al tener como objetivo resumir los contenidos en una descripción de alto nivel.

El término *Broker*<sup>1</sup> [Gravano 94] es comunmente utilizado para denotar al agente encargado de proveer facilidades de búsqueda sobre un conjunto de servidores distribuidos utilizando mecanismos de *query routing*. Básicamente, la función de un *broker* consiste en almacenar de una manera conveniente la *meta-información de contenido* de cada potencial servidor y de implementar el algoritmo de *query routing* para la resolución de las consultas.

El servicio provisto por un Broker es similar al que provee un meta-buscador en el sentido que ambos reenvían la consulta a otros servidores para su resolución. La diferencia fundamental entre ambos es que el primero utiliza una estrategia basada en query routing para decidir a que servidores reenviar la consulta, mientras que el segundo reenvía siempre a todos los que tenga configurados.

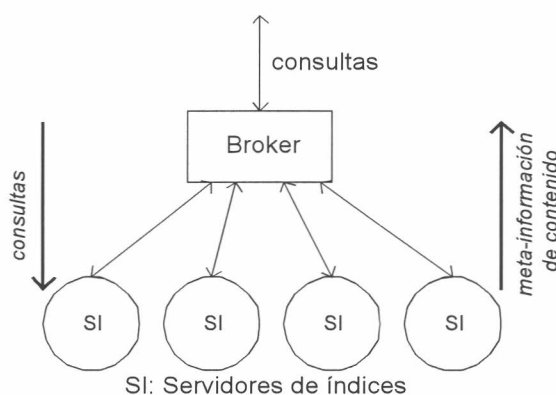


Gráfico 3.9 Funcionamiento de un broker

Sin embargo, centralizar la meta-información de contenido en un único broker como muestra el Gráfico 3.9 no escala. El crecimiento del número de servidores y de consultas a procesar tendrá como consecuencia una disminución de la performance del broker, a pesar de que solo se esté almacenando una pequeña descripción por servidor.

Con el objetivo de resolver los problemas de eficiencia, performance y escalabilidad, la comunidad Internet se ha abocado al diseño de arquitecturas distribuidas más sofisticadas en las cuales varios brokers se distribuyen el conjunto de servidores y cooperan entre sí con el objetivo de encaminar las consultas a los servidores finales. Estas mallas de brokers se organizan basándose en una relación jerárquica o bien en una no jerárquica.

El diseño de arquitecturas distribuidas para la búsqueda de información es un problema que también tiene considerable atención en los grupos de investigación de Bibliotecas Digitales. Se

<sup>1</sup> El término Broker fue introducido en [Bowman 95] pero con un significado diferente al que utilizaremos a lo largo del trabajo.



han estudiado algunos problemas puntuales como sumarización de contenidos para query routing [Gravano 94], protocolos de diseminación de meta-información para la construcción de brokers [Gravano 97]. También se han diseñado y testeado arquitecturas para la búsqueda en servidores de índices distribuidos globalmente en Internet y protocolos para la diseminación de la meta-información necesaria para soportar la búsqueda distribuida [Lagoze 98].

### 3.3.1 Mallas de brokers

Las arquitecturas que utilizan mallas de brokers se basan en un modelo en el cual muchos servidores independientes indexan localmente sus recursos y un conjunto de servidores especiales (*brokers*) se encargan de encaminar las consultas a los servidores de índices.

La mayoría de los prototipos existentes para la búsqueda en la Web, componen los brokers en una estructura jerárquica [Sheldon 95a] [Gravano 95] [RFC 1913] a través de la cual se sumanizan sumalizaciones. El broker de la raíz contiene toda la información necesaria para encaminar las consultas a los brokers de nivel inferior y así para todos hasta llegar a los servidores de índices. Estas arquitecturas jerárquicas son las más utilizadas porque tienen buenas propiedades de escalabilidad en cuanto al manejo de recursos y son más fáciles de implementar.

Otros prototipos componen los brokers en una estructura no jerárquica [Rodríguez 96] [Francis 95] [Schwartz 90] [Gonçalves 97]. Ningún broker dispone de toda la información necesaria al momento de resolver una consulta. La escalabilidad de este tipo de arquitecturas depende en gran parte de los mecanismos que se utilicen para diseminar la meta-información de contenido en la malla de brokers, es decir de como cada broker adquiere el conocimiento de lo que existe.



### 3.3.1.1 Composición jerárquica de brokers

La dinámica de una arquitectura jerárquica de brokers se observa en el Gráfico 3.10.

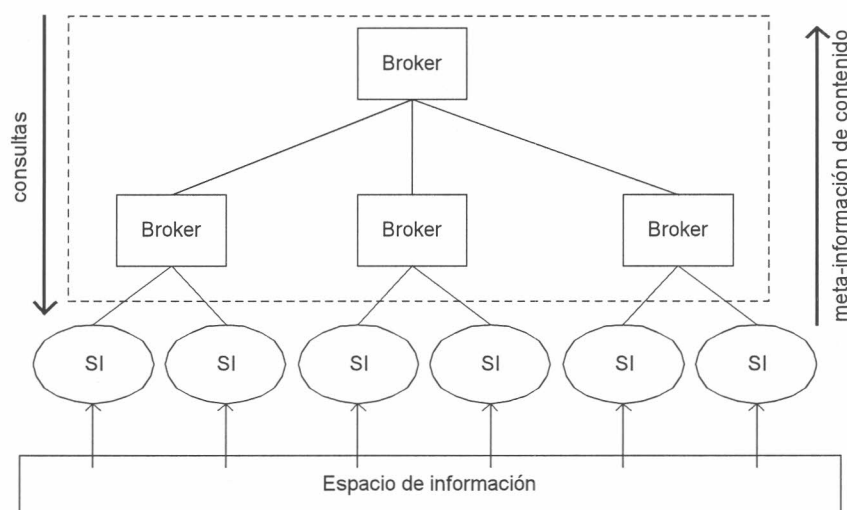


Gráfico 3.10 Malla Jerárquica de brokers

Whois++ [RFC 1913], Content Routing [Sheldon 95a] y HGloss [Gravano 95] son sistemas de búsqueda basados en una arquitectura jerárquica de brokers. El primero fue especialmente diseñado para servicios de directorios, mientras que el resto para la búsqueda en la Web y bases de datos textuales respectivamente.

Estos sistemas utilizan fuertemente el concepto de *colección de documentos*, a la que definen como el conjunto de recursos almacenados en un servidor. La característica más relevante que incorporan es que una colección se representa por una descripción de alto nivel (*meta-información de contenido*). La ventaja de utilizar descripciones es que el resultado de realizar una operación sobre la meta-información de contenido se aproxima al resultado de realizar la misma operación sobre el espacio de información que ésta describe [Sheldon 95a].

Los índices se organizan en una jerarquía utilizando descripciones de colecciones, donde los servidores de índices de nivel inferior son los que almacenan y administran su colección local de documentos. Los contenidos de estas colecciones se sumarizan en meta-información de contenido para ser almacenadas por los brokers del primer nivel. A su vez, estos brokers sumarizan estas descripciones enviándolas a los broker de nivel superior. De esta manera, cada broker almacena las descripciones suministradas por los brokers del nivel inmediatamente inferior las que son a su vez objeto de sumariación y utilizadas para el query routing.

Entonces, hacia arriba en la jerarquía se transmite meta-información con el objetivo de resumir, en cada broker, el espacio de información alcanzable desde éste. Cualquier servidor de índices que desee participar en una arquitectura de estas características deberá ser capaz de generar la meta-información de contenido requerida por los brokers que compongan la arquitectura.

A través de la jerarquía de brokers las consultas de los usuarios se *encaminan* hacia los servidores de índices finales, que son los que tiene los objetos indexados y que producirán el listado final de objetos que satisfagan la consulta (*result set*).

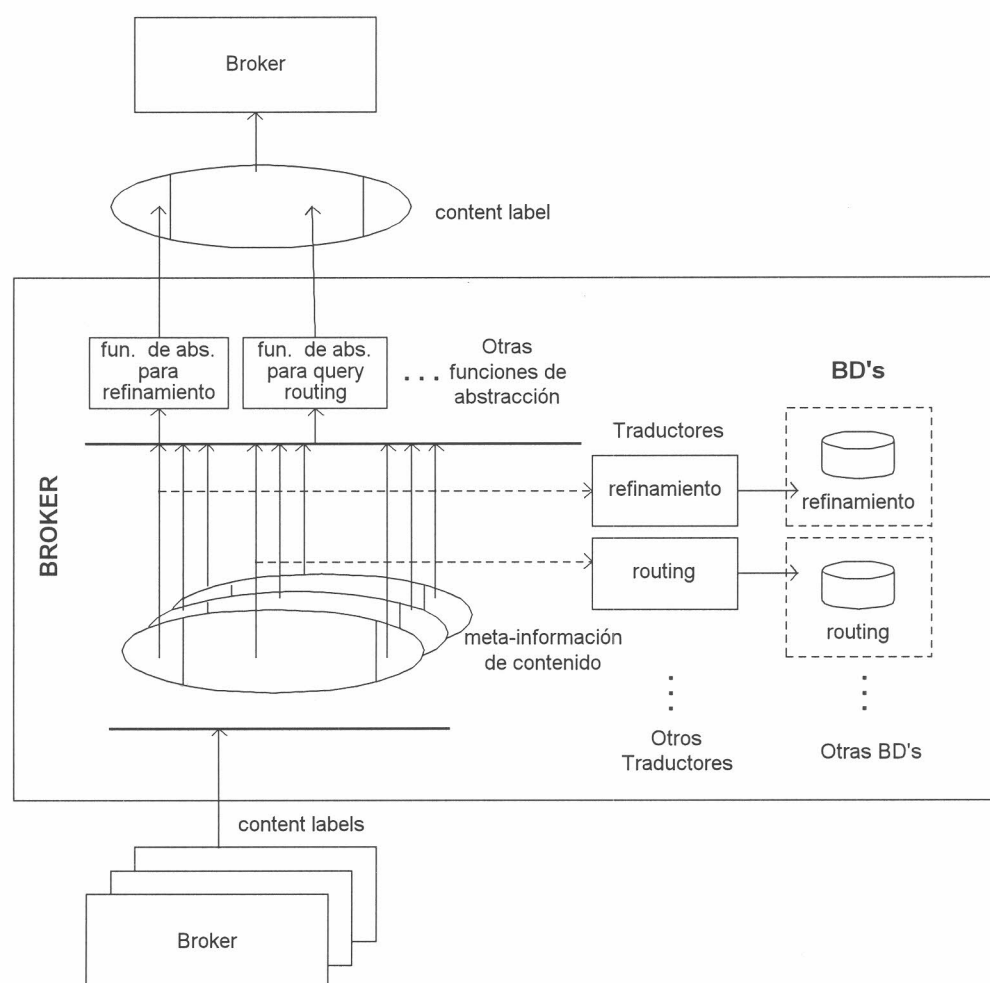


Gráfico 3.11 Funcionamiento interno de un Broker de Content Routing

Los brokers que componen la malla jerárquica tienen entre sí una relación preestablecida, generalmente determinada por los intereses particulares de la organización que la administre. Una vez establecida la jerarquía, ésta se mantiene independientemente de los contenidos existentes y es utilizada para la resolución de las consultas de los usuarios.

El mecanismo de diseminación de la meta-información de contenido es siempre el mismo, un broker envía la meta-información al broker de nivel inmediatamente superior y así para todos.

En este tipo de arquitecturas, la estructura de la meta-información de contenido reúne el conjunto de datos necesarios para el funcionamiento general y el tipo de servicios a proveer. Lo que diferencia a estos sistemas entre sí es el tipo y estructura de meta-información que utilizan, Whois++ utiliza *centroides*, Content routing utiliza *content label* y HGloss utiliza *gloss summary*. A modo de ejemplo, en el sistema Content Routing los *content labels* tienen información útil para

refinamiento<sup>1</sup> y ruteo de consultas puesto que son estos los servicios a proveer por los brokers que la componen. En el Gráfico 3.11 se observa el funcionamiento interno de un broker de Content Routing. Este utiliza *funciones de abstracción* para generar un content label a partir de los content labels recibidos de otros brokers.

En Whois++ los encargados de implementar el query routing son los clientes [RFC 1914] utilizando la información provista por los brokers, que en este sistema se llaman *Index Servers*. La estructura de meta-información de contenido que utiliza se basa en templates de atributos llamados *centroides*. Los clientes utilizan un Index Server por defecto para comenzar las búsquedas. Para resolver una consulta, el cliente la envía al Index Server que tenga configurado por defecto, el cual, tomando como base los centroides que tenga indexados, obtiene los que mejor satisfagan la consulta devolviéndolos al cliente como *Query Referrals*. Los *query referrals* serán utilizados por el cliente para avanzar en la resolución de la consulta, contactando a otros Index Servers.

El mismo esquema sigue HGloss, donde los brokers no realizan la búsqueda en los servidores de índices remotos, sino que solo sugieren los más relevantes para que el cliente se encargue de continuar la búsqueda.

En Content Routing los brokers se denominan *content routers*, y tienen como función implementar servicios de refinamiento de consulta y query routing sobre la colección de documentos que mantengan. A diferencia de los dos anteriores, los content routers saben encaminar las consultas a los servidores más relevantes hasta obtener los documentos finales. Discover [Sheldon 95b] y HyPursuit [Weiss 96] son prototipos de servicios basados en la arquitectura de Content Routing.

### 3.3.1.1.1 Algunas reflexiones

- El proceso de sumarización es difícil cuando los servidores no están especializados en ningún tópico. Las colecciones se orientan a los objetos que pertenecen a un servidor, aunque estos sean de múltiples temas. En estos casos generar meta-información de contenido que describa los contenidos de recursos tan diversos puede ser muy difícil o bien requerir estructuras de gran tamaño con el objetivo de que sean más descriptivas. Este problema se aprecia más en los brokers intermedios, los que deben producir una sumarización adecuada para la combinación de las colecciones que mantengan.
- Las estructuras de meta-información deben mantener un tamaño razonable para escalar. Esto hace que a medida que aumente la profundidad de la jerarquía se produzca pérdida de información, con la consiguiente pérdida de funcionalidad del sistema en su habilidad de encaminar eficientemente la consulta [Sheldon 95a].

---

<sup>1</sup> El proceso de refinamiento de consultas tiene por objetivo generar una lista de términos relacionados con la consulta que puedan ser utilizados para la confección de consultas más específicas que permitan reducir el conjunto de recursos que la satisfagan [Sheldon 95a].

- Como los servidores no están especializados, la información relevante para un cierto tópico se encontrará distribuida en la jerarquía. Esto significa que muchos brokers deberán ser consultados para producir la lista final de los recursos disponibles en respuesta a una consulta determinada. Si las colecciones fueran especializadas, serían más fácil el proceso de sumarización y más eficiente el encaminamiento de las consultas a lo largo de la jerarquía al disminuir la cantidad de nodos a contactar para resolver una consulta. Actualmente, para minimizar estos problemas se utilizan técnicas de cache de consultas en varios niveles de la jerarquía [Sheldon 95a]. El sistema WHERE [Rio 97] incorpora a los términos de los centroides de Whois++ un peso que denota su importancia y que es utilizado para el query routing. Con ese agregado se logró reducir el número de nodos a contactar obteniéndose un algoritmo de resolución más eficiente que el utilizado por Whois++.
- Las estructuras jerárquicas tienen problemas cerca de la raíz, es decir en los brokers de nivel superior. Estos contienen toda la información de indexación provista por los brokers de nivel inferior y son los más utilizados para resolver las consultas. Con el objetivo de mejorar los tiempos de respuesta y lograr que el sistema sea más robusto, algunos sistemas incorporan mecanismos de replicación en varios niveles de la jerarquía, como es el caso de Harvest [Bowman 95].
- Las estructuras de información jerárquicas tienden a ser resistentes frente a cambios en los contenidos. En general son poco flexibles, pudiendo volverse inconsistentes a medida que se incorporan nuevos recursos, con la necesidad tal vez de tener que reestructurar la jerarquía para mantener la eficiencia del sistema [Sheldon 95a].
- En las jerarquías los brokers tienen el mismo comportamiento y su funcionamiento está regido por una misma autoridad administrativa. No es posible que los administradores apliquen políticas propias que guíen el comportamiento. La resolución de las consultas siempre se realiza de la misma manera. La formación de una jerarquía requiere un nivel de coordinación y organización que puede dificultar la autonomía de los sitios que la componen.
- Las arquitectura jerárquicas tienen buenas propiedades de escalabilidad en cuanto al manejo de recursos. Cada broker puede mantener el control a nivel estructura de meta-información. Esto permite que el requerimiento de recursos informáticos no crezca demasiado en ningún punto del sistema [Sheldon 95a].
- La falla de un broker aislaría un subárbol completo de la jerarquía, dejando inaccesible el espacio de información alcanzable desde él.

### 3.3.1.2 Composición no jerárquica de brokers

En una malla no jerárquica de brokers, no existe una relación explícita o predefinida entre los brokers que la componen como en el caso de una malla jerárquica.

El objetivo de este tipo de arquitecturas es que los *brokers* involucrados organicen el espacio de información dinámicamente acorde a los recursos existentes y las consultas que realicen los usuarios. Este no es el caso de una jerarquía, donde un grupo administrativo es el que se encarga de especificar como se organiza el espacio de información. La dinámica de una arquitectura no jerárquica se observa en el Gráfico 3.12.

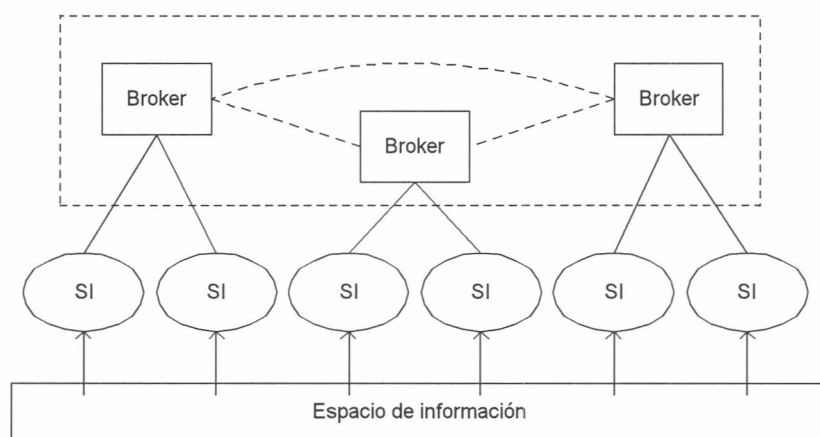


Gráfico 3.12 Malla No jerárquica de brokers

La información que intercambian los brokers no se disemina en forma estática como en una jerarquía. Por el contrario, la diseminación de la información y la distribución de las consultas se dan de manera tal que la información fluya hacia donde sea necesaria [Schwartz 90].

La idea es que cada broker *aprenda/descubra* la existencia de otros brokers a partir de la interacción requerida para resolver las consultas que realicen sus usuarios. El objetivo de cada broker es mantener localmente *vínculos* a otros brokers que contengan información relacionada con los intereses de los usuarios que atienda. Estos vínculos evitarán al broker tener que salir a *descubrir* que hay en la red al momento de la resolución de una consulta.

En la mayoría de las arquitecturas, los brokers almacenan la información referida a otros brokers de la red en una *base de datos local* como muestra el Gráfico 3.13. Estas bases son *dinámicas*, en el sentido que albergan vínculos a otros brokers que pueden variar en función de los contenidos existentes y de las necesidades puntuales de la comunidad de usuarios. Esta base de datos generalmente se la denomina *base local de conocimiento adquirido* y constituye la *visión local* que tiene cada broker de la red global de brokers. Se utiliza para distribuir las consultas (query routing), al contener la información acerca de otros brokers del sistema.

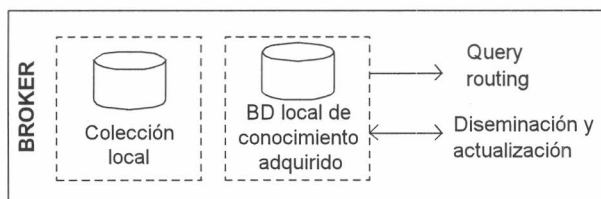


Gráfico 3.13 Estructura interna de un broker

Ningún broker de la red tiene, en cualquier momento, información sobre *todos* los brokers restantes de la red. En general, dos brokers estarán vinculados si uno contiene información relevante al otro.

Este tipo de relación entre los brokers permite un mayor dinamismo que una jerarquía. Los brokers tienden a relacionarse en función de intereses comunes. Estas relaciones varían acorde a los contenidos existentes y la necesidades de la comunidad de usuarios que atiende cada broker [Schwartz 90].

Cada broker opera con información parcial al momento de resolver una consulta. El objetivo de cada broker es que su *base local de conocimiento adquirido* reúna la suficiente cantidad de información sobre otros brokers para que, al momento de resolver una consulta, se disponga localmente de la información necesaria para el *query routing*.

En este tipo de arquitecturas los mecanismos de comunicación que se utilizan para diseminar la información entre los brokers son muy importantes. En definitiva, son los que harán que el sistema sea utilizable y que la arquitectura que lo implemente sea escalable.

Los mecanismos de diseminación deben garantizar a los brokers reunir localmente la información suficiente para que a la hora de resolver las consultas, el costo se reduzca al mínimo (evitando tener que salir a buscarla al momento de resolver una consulta).

En primera instancia, un mecanismo de diseminación posible sería que todos los brokers propaguen al resto la información que disponen a intervalos regulares para que cada uno actualice su base local reflejando los cambios que pudieran producirse. Este mecanismo asegurará a cada broker tener todos los elementos al momento de resolver las consultas. El valor del intervalo de tiempo debe garantizar que la información local de cada broker se mantenga coherente y a la vez no sobrecargue la red con excesivo tráfico.

Sin embargo, la cantidad de mensajes por nodo crecerá linealmente con el tamaño de la red. Un intervalo de tiempo corto generará un elevado costo de comunicación a medida que la cantidad de brokers crezca. La red se sobrecargaría con excesivo tráfico generándose un aumento (*overhead*) de trabajo en los brokers para procesar toda la información.

Por otro lado, si un broker no tuviera ningún tipo de información sobre otros brokers al momento de resolver una consulta, debería contactar al resto para saber a quién puede derivársela, estilo broadcast a todos. Esto es costoso en términos de tiempos de respuesta y cantidad de mensajes en la red si la malla de brokers tiene muchos nodos.



Pero, si cada broker tuviera información de todos los brokers restantes, tendría siempre localmente la información a la hora de resolver consultas, pero es probable que los costos de tales mecanismos de diseminación sean altos en cantidad de mensajes en la red.

AlephWeb [Rodríguez 96], Mecanismos Probabilísticos para el Descubrimiento de Recursos (MPRS) [Schwartz 90] e Ingrid [Francis 95] son sistemas de búsqueda basados en una arquitectura no jerárquica de brokers.

En Ingrid, cada nodo de la red tiene dos componentes principales: IRSE (Information Retrieval Search Engine) y FISE (Forward Information Search Engine). El primero indexa la colección local de recursos, como en cualquier servicio de búsqueda. El segundo soporta la búsqueda distribuida al contener la información acerca de otros servidores de la topología Ingrid. En este sistema, el vínculo entre los nodos de la red se da a nivel recurso y no a nivel colección. El servicio que provee es fundamentalmente de navegación. Dos servidores están relacionados si mantienen recursos similares.

AlephWeb y MPRS presentan una arquitectura similar entre sí, pero su principal diferencia radica en los mecanismos de diseminación y búsqueda que utilizan en la malla de brokers.

AlephWeb utiliza una estructura de federación entre los brokers definida por el *Aleph Computational Model* [Rodríguez 95]. La federación es una estructura organizacional en la cual distintas entidades independientes cooperan para compartir recursos sin perder su autonomía, manteniendo sus propias políticas administrativas [Marqués 95]. En términos de autonomía, una estructura de federación es un caso extremo de organización que permite escalar. En el otro extremo están las jerarquías, un tipo de organización que generalmente requiere que todos los sistemas dentro de la estructura adhieran a los mismos mecanismos de comunicación, seguridad, etc.

Cada broker de AlephWeb tiene un gestor de federación cuya función es vincularlo con la estructura federada de brokers, resolviendo las consultas externas e implementando los mecanismos de aprendizaje necesarios para mantener la coherencia de su base local de conocimiento con respecto a los cambios que pudieran producirse.

El autor, conjuntamente con investigadores del laboratorio de Bibliotecas Digitales de la Universidad de Stanford adaptaron el sistema HGloss para que utilice el Aleph Computational Model, generando el prototipo GLOSS-Aleph [Rodríguez 98]. Los resultados obtenidos a partir de las pruebas efectuadas fueron buenos. Se demostró que el porcentaje de hits (éxitos) en las tablas locales de los gestores de federación fue del orden del 30%. Este porcentaje es similar al logrado con el uso de Proxies de Web. Este resultado demostró que los mecanismos de aprendizaje utilizados por los gestores de federación lograban reunir localmente información altamente reutilizable.

MPRS presenta protocolos probabilísticos para la diseminación y búsqueda de información en la malla de brokers y protocolos de cache para el manejo de la base de datos local de conocimiento



adquirido. El objetivo de este sistema es lograr que los brokers se agrupen en lo que llama *subgrafos especializados*.

Los protocolos propuestos en MPRS contemplan dos fases: *diseminación* y *búsqueda*. En la fase de *diseminación* los nodos intercambian la información de su base local a intervalos regulares. La *diseminación* se realiza siempre a un conjunto  $N$  (fijo) de vecinos de manera tal que durante la fase de *búsqueda* la información pueda ser encontrada utilizando pruebas *random simples*. Inicialmente, las *búsquedas* se resolverán contactando a un conjunto  $N$  de brokers en forma *random*. Luego de un tiempo de interacción y del uso de los protocolos para el mantenimiento de *cache*, cada nodo reunirá vínculos con brokers que mantengan información relacionada al interés de sus usuarios, conformándose los *subgrafos especializados*. Lo importante de estos protocolos, es que cada broker siempre contacta un número fijo de nodos que es independiente del tamaño de la *mall*a de brokers. Esta característica le da *escalabilidad*.

### 3.3.1.2.1 Algunas reflexiones

- Es posible la aplicación de políticas locales para guiar el comportamiento de cada broker. La independencia de una estructura predefinida como una jerarquía permite que cada broker tenga mayor libertad para implementar la estrategia de *query routing* más apropiada para resolver las consultas. De esta manera es posible que cada broker mantenga una visión personalizada del espacio de información que se ajuste mejor a las necesidades de la comunidad de usuarios que atiende.
- Al igual que las jerárquicas, utilizan el concepto de colección orientado a los documentos mantenidos por un servidor. Aquí también puede dificultarse el proceso de *sumarización* al no estar especializados los servidores.
- Son más tolerantes a fallas. La falla de un broker dejaría solo inaccesible su colección local y no a muchas colecciones como si en una jerarquía llegara a fallar un broker intermedio.
- Se adaptan mejor a entornos caracterizados por una administración descentralizada.
- Los mecanismos de *diseminación* de meta-información que se utilicen en la *mall*a de brokers afectan fuertemente la *escalabilidad* del sistema.
- El uso de información local es muy importante a la hora de optimizar los recursos de red y minimizar la necesidad de tener procesamiento distribuido que es más costoso.

### 3.3.1.3 Colecciones y meta-información de contenido

Con el objetivo de lograr una arquitectura escalable en lo que respecta al manejo de recursos informáticos, las mallas de brokers manejan el concepto de *colección de recursos*. La abstracción de colección permite utilizar unidades de información más pequeñas.

Una colección está lógicamente definida como un conjunto de criterios para la selección de recursos de un espacio de información, por ejemplo la Web. La naturaleza de dichos criterios pueden variar en complejidad [Lagoze 98], por ejemplo:

- El conjunto de recursos que se encuentren almacenados en un repositorio.
- Otros criterios son aquellos que permiten un crecimiento dinámico de la colección a partir de recursos que estén distribuidos en múltiples repositorios. Un simple ejemplo de esto sería definir a una colección como el conjunto de recursos en los que el valor de un campo de su meta-información se corresponda con cierto valor de un diccionario controlado (por ejemplo, que el elemento *subject* de Dublin Core [Weibel 97] contenga el valor “computación científica”)

En las mallas de brokers, tanto las estructuras jerárquicas como las no jerárquicas, definen una *colección* como el conjunto de recursos almacenados en un servidor. Cada colección tiene asociada una descripción de alto nivel que hemos denominado *meta-información de contenido*, la que reúne la información necesaria para que el sistema pueda identificar y diferenciar una colección de otra, y por ende un servidor de otro.

La ventaja de utilizar descripciones de alto nivel es que el tamaño de la descripción es más pequeño que el de toda la colección de recursos, haciéndola más práctica de manejar (en términos de requerimientos de ancho de banda para su transmisión y espacio de almacenamiento).

#### 3.3.1.3.1 Generación de meta-información

En las mallas de brokers, la estrategia generalmente utilizada para la generación de meta-información de contenido es la de *surrogate document* para la colección

Meta-información de contenido = surrogate document para la colección

Un *surrogate document* es un conjunto de atributos, tales como título, autor, etc., con valores obtenidos de los recursos contenidos en la colección. Tienen la importante ventaja de que pueden ser indexados como cualquier documento estándar bajo los términos que aparecen en el.

La estructura de meta-información de contenido tienen múltiples usos:

- Para describir a los usuarios los contenidos de una colección de recursos;
- Para implementar query routing con el objetivo de encaminar las consultas a las colecciones que mejor puedan satisfacerlas;

Para que la meta-información de contenido sea útil al sistema, debe manejar por lo menos dos tipos de atributos:

- Un conjunto de atributos determinados por el administrador del servidor que pueden o no formar parte de los documentos de la colección que representa.
- Atributos que se deriven en forma automática de los contenidos de la colección, posiblemente utilizando herramientas estadísticas. Por ejemplo términos descriptivos con alta frecuencia pueden ser buenos candidatos. Estos atributos se los denomina atributos naturales porque se obtienen directamente de los datos.

Los administradores de los repositorios de información son quienes deben decidir los términos a incluir en la meta-información teniendo en cuenta que ésta será utilizada por el sistema para determinar la relevancia de sus colecciones en la resolución de las consultas.

Una estrategia para la generación de la meta-información de contenido deben tener en cuenta tres objetivos:

- contener términos que caractericen la colección como un todo (típicamente términos muy frecuentes);
- tener en cuenta términos con gran poder discriminatorio, permitiendo distinguir un servidor de otro, es decir una colección de otra;
- garantizar que cambios mínimos en los contenidos de un recurso no necesariamente requieran un cambio en la meta-información de contenido, de manera tal que esta última cambie con menor frecuencia.

En las mallas de brokers, es posible dividir estas actividades entre un proveedor de información y un broker del sistema. El proveedor de información debe proveer meta-información que contenga términos más frecuentes, mientras que un broker del sistema puede decidir guardar solo los términos que considere con mayor poder discriminatorio a los efectos de implementar el query routing.

Los términos pueden elegirse basándose en un significado estadístico o bien en otro análisis de su importancia, por ejemplo términos que aparecen en los títulos o resúmenes pueden tener precedencia sobre otros.

Existen mecanismos prácticos para la extracción de meta-información de contenido de un conjunto de recursos. Pueden utilizarse varias herramientas para seleccionar los términos a incluir en la meta-información, por ejemplo en Whois++ sería computar un *centroid document* [RFC 1913].

### 3.3.1.3.2 Query Routing

Para que un broker pueda encaminar y resolver una búsqueda a partir de una consulta, debe ser capaz de identificar el conjunto de colecciones más relevantes, enviar la consulta a los servidores que las alberguen, obtener los resultados y combinarlos para presentar la respuesta al usuario [Sheldon 95a].

Con la estructura de meta-información definida anteriormente, una colección satisface una consulta si la meta-información de contenido que la representa la satisface.

Esta aproximación del espacio de recursos será más o menos exacta dependiendo de la meta-información utilizada. Por ejemplo, si la meta-información contiene todos los términos, entonces el broker tendrá un índice completo y la aproximación será perfecta. Sin embargo esto no escala ya que el tamaño de la meta-información será muy grande. En estos casos, el sistema debe disminuir la calidad de la aproximación para escalar generando descripciones más pequeñas.

## 3.3.2 Bibliotecas Digitales

Cuando hablamos de arquitecturas de búsqueda e indexación distribuida, no podemos dejar de mencionar alguno de los proyectos de investigación que se están dando en el área de Bibliotecas Digitales. Tienen por objetivo la búsqueda de soluciones para varios de los aspectos del problema de la construcción de Bibliotecas Digitales Distribuidas. Entre éstos se destacan los de las Universidades de Stanford [Stanford-DL] y Cornell [Cornell-DL].

Hoy en día los usuarios de distintas redes de computadoras pueden comunicarse de una manera efectiva gracias a la utilización de los mismos protocolos de red. Las Bibliotecas Digitales tienen por objetivo lograr esa misma interoperabilidad pero en un plano de más alto nivel que los protocolos orientados a la comunicación entre redes (inter-net). Dicha interoperabilidad pretende lograr que los usuarios puedan interactuar con distintos sistemas de información de una manera uniforme.

Una Biblioteca Digital se define como colecciones de servicios autónomos y distribuidos que interoperan entre sí con el objetivo de brindar a los usuarios un acceso coherente y uniforme a un amplio y organizado repositorio de información [Levy 95].

El grupo de investigación de la Universidad de Stanford ha estudiado problemas puntuales como sumariaización de contenidos para query routing [Gravano 94] y diseño de protocolos de diseminación de meta-información para la construcción de brokers [Gravano 97].

En cambio, el grupo de investigación de la Universidad de Cornell se concentra en la investigación, desarrollo y testeo de arquitecturas de bibliotecas digitales globalmente distribuidas y federadas. En contraste a los sistemas centralizados o standalone, estos sistemas federados están compuestos por servicios semi-autónomos, distribuidos globalmente en Internet, que interoperan entre sí por medio de un protocolo abierto<sup>1</sup>. El principal trabajo de este grupo se da en el contexto del sistema NCSTRL<sup>2</sup> (Networked Computer Science Technical Research Library) [NCSTRL].

### 3.3.2.1 NCSTRL (Networked Computer Science Technical Research Library)

NCSTRL es una biblioteca digital de reportes de investigación sobre computación científica basada en la arquitectura Dienst [Lagoze 95a] que utiliza protocolos abiertos para bibliotecas digitales distribuidas. Actualmente, la colección de NCSTRL brinda acceso a papers de más de 100 institutos de investigación con más de 60 servidores distribuidos en Estados Unidos, Europa y Asia. Este sistema es utilizado fundamentalmente para experimentar con tecnologías de bibliotecas digitales distribuidas a escala global.

#### 3.3.2.1.1 Arquitectura Dienst

La arquitectura Dienst<sup>3</sup> tiene las siguientes características fundamentales: un modelo lógico de documento, servicios de bibliotecas digitales distribuidos y un protocolo abierto para la interacción entre los servicios.

**Modelo Lógico de Documento:** En el corazón de la arquitectura de Dienst está la noción de documento, una abstracción lógica que incorpora los siguientes conceptos:

- Cada documento utiliza un nombre globalmente único (URN [Baleani 98]).
- Un documento está constituido por un conjunto de componentes. Los dos componentes actualmente utilizados en NCSTRL son una descripción bibliográfica (*meta-información*) y el cuerpo (*body*) del documento.
- Cada componente puede encontrarse en uno o más formatos. Por ejemplo, el cuerpo del documento puede estar disponible en PostScript, HTML, etc.
- Cada componente puede tener subcomponentes. Por ejemplo, si el cuerpo está disponible en PostScript, podría estar dividido en páginas.

<sup>1</sup> Aquí, este concepto tiene el mismo significado que la utilización del protocolo IP en el área de interconexión de redes.

<sup>2</sup> Se pronuncia "ancestral".

<sup>3</sup> Se puede encontrar una descripción más detallada de Dienst en el Manual de Referencia [Lagoze 95b].

**Servicios de Bibliotecas Digitales:** La funcionalidad de la arquitectura Dienst está lógicamente dividida en un conjunto de servicios distintivos. Existen tres servicios centrales y un servicio exclusivo para el manejo de colecciones (*Collection Service*):

- *Repository Service (RS)*: almacena y provee acceso a documentos identificados utilizando el servicio de nombre global (URN) y estructurados según el modelo de documento.
- *Index Service (IS)*: almacena información de indexación acerca de los documentos de la colección y responde consultas sobre esta información.
- *User Interface Service (UI)*: provee un front-end para la interacción de los usuarios con los otros servicios.

**Protocolo Abierto:** Los servicios de Dienst interoperan entre sí utilizando un protocolo bien definido. Está estructurado sobre la noción de servicios descrita anteriormente. Cada requerimiento del protocolo está encapsulado como una solicitud de servicio.

El *Collection Service (CS)* es un servicio exclusivo para el manejo de colecciones. En NCSTRL la colección está lógica y administrativamente dividida en autoridades de publicación. Cada autoridad de publicación tiene el control de la administración de las sub-colecciones mantenidas en sus repositorios. Los campos de meta-información (por ej.: título, autor, resumen) de cada documento de estos repositorios es indexada por alguno de los servidores de índices. El objetivo principal del CS es permitir la federación de los servidores de índices en una única colección uniforme.

Este servicio brinda acceso a la siguiente información:

- *La lista de las organizaciones que forman parte de la colección.* En NCSTRL la granularidad de una organización se corresponde con departamentos de computación científica o institutos de investigación miembros de NCSTRL.
- *La ubicación en la red.* El servicio provee la dirección y el port de los servidores de índices que almacenan información de indexación para cada organización. Por ejemplo, la información de indexación para el departamento de computación científica de la Cornell University está en los servidores foo.ncstrl.org port 80 y bar.ncstrl.org port 8083.
- *Meta-información sobre cada uno de los servidores de índices.* En el prototipo actual esta información indica el carácter de cada servidor de índices para la organización, primario o secundario. Esta meta-información es utilizada para la toma de decisiones de query routing. Sería conveniente expandirla para incluir otros datos útiles como: última actualización, información de performance, etc..

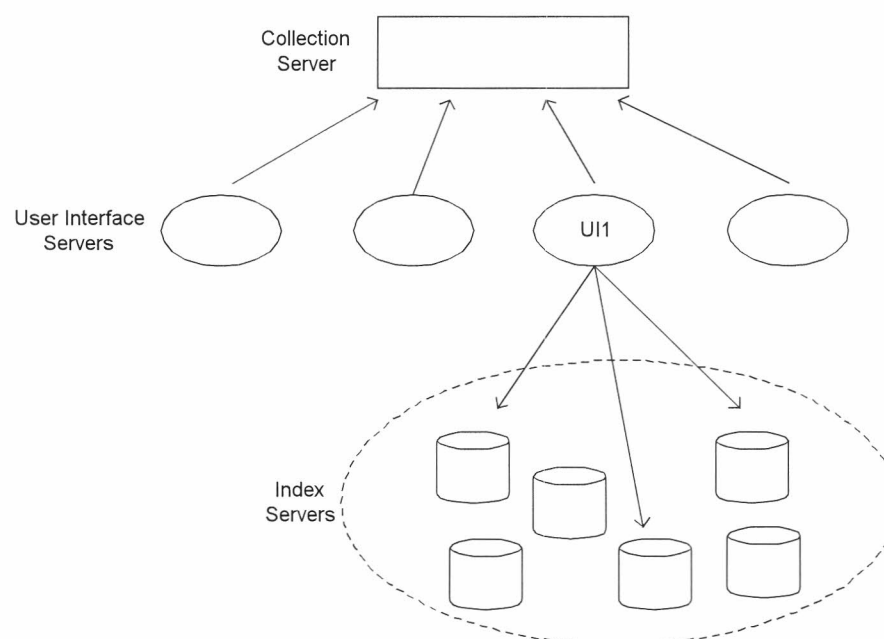


Gráfico 3.14 Interacción entre Collection Services y otros Servicios de Dienst

El Gráfico 3.14 ilustra la interacción entre el CS y los otros servicios de Dienst. Cada *UI* obtiene del CS la información sobre la colección global. El *UI* (UI1 en el gráfico) utiliza esta información para determinar los servidores de índices que procesarán las consultas que deba resolver.

Cuando los usuarios consultan la colección de NCSTRL, eligen que organizaciones desean incluir en su búsqueda. La *UI* respectiva puede determinar hacia donde encaminar (*query routing*) la consulta basándose en la información de ubicación y contenidos provista por el CS. Con esta información la *UI* encamina la consulta a los servidores de índices correspondientes, obteniendo los resultados y combinándolos para generar la respuesta al usuario.

Dado que pueden existir servidores de índices replicados para una misma organización, cada *UI* implementa un algoritmo de *query routing* adaptativo basándose en la historia de las interacciones con los servidores de índices para decidir cual es la mejor réplica a utilizar en la resolución de una consulta.

### 3.3.2.1.2 Connectivity Regions

Los problemas de conectividad entre los diferentes sitios comenzó a afectar la performance del sistema a medida que la colección NCSTRL se fué expandiendo. Este problema llevó a los desarrolladores a modificar el funcionamiento del CS para incorporar el concepto de *connectivity regions*, abstracción que permite modelar los patrones de conectividad entre sitios.

Lagoze en [Lagoze 98] define a una *connectivity region* como un grupo de nodos de la red que tienen buena conectividad entre sí con respecto a los nodos que están fuera de la región. Este nuevo concepto redefine la estrategia de búsqueda distribuida descrita anteriormente de la siguiente manera: ante la ausencia de fallas (en la red o servidores) el *query routing* (realizado por una *UI*) debe restringirse a aquellos servidores de índices pertenecientes a su misma región. En el



caso de fallas, puede seleccionarse un servidor alternativo dentro de la misma región o de otra región con la que exista buena conectividad.

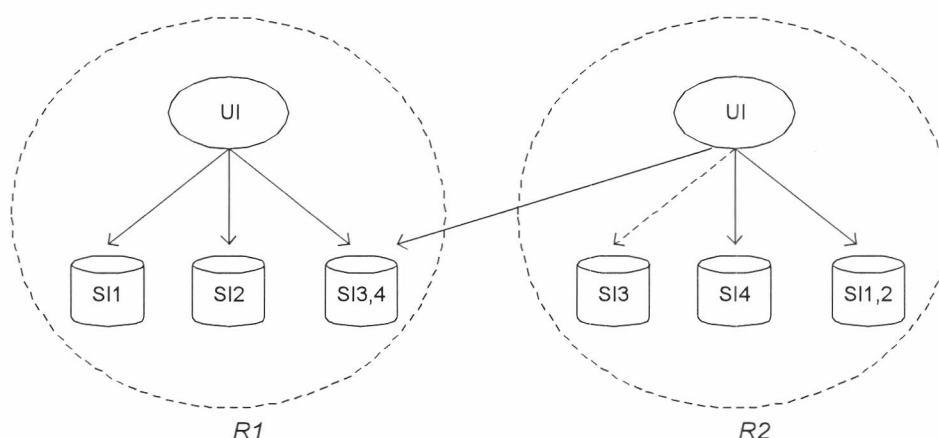


Gráfico 3.15 Connectivity Regions

El Gráfico 3.15 ilustra un ejemplo de *connectivity regions*. En el gráfico se muestran dos regiones R1 y R2. Cada región contiene un UI que encamina consultas y combina resultados y tres servidores de índices que responden consultas. Como puede observarse, la falla de un servidor de índices en una región (SI3 en R2) requiere que la consulta se encamine a un servidor de índices fuera de la región (SI3,4 en R1).

En la implementación original del CS de Dienst, todos los UI utilizan el mismo conjunto de servidores de índices para procesar sus consultas, es decir tienen la misma vista sobre la colección global al obtener la información del mismo CS.

Para que el concepto de *connectivity region* tenga sentido, las réplicas disponibles de servidores de índices deben distribuirse en las distintas regiones de manera tal que un UI pueda encaminar las consultas a los servidores de índices pertenecientes a su misma región.

Esto motivó la redefinición del CS con el objetivo de que UI's de diferentes regiones basen sus decisiones de query routing en diferentes *vistas* de la colección global. Por ejemplo, la UI de la región R1 debería considerar a SI1 como el servidor de índices primario para la organización 1. Por el contrario, la UI de la región R2 debería considerar a SI1,2 como el servidor de índices primario para la organización 1. De esta manera, la vista de la colección es distinta en los UI de R1 que en los de R2.

Esto permite que una única colección como la de NCSTRL, pueda tener múltiples vistas correspondientes a las *connectivity regions* que hayan sido definidas para la colección.

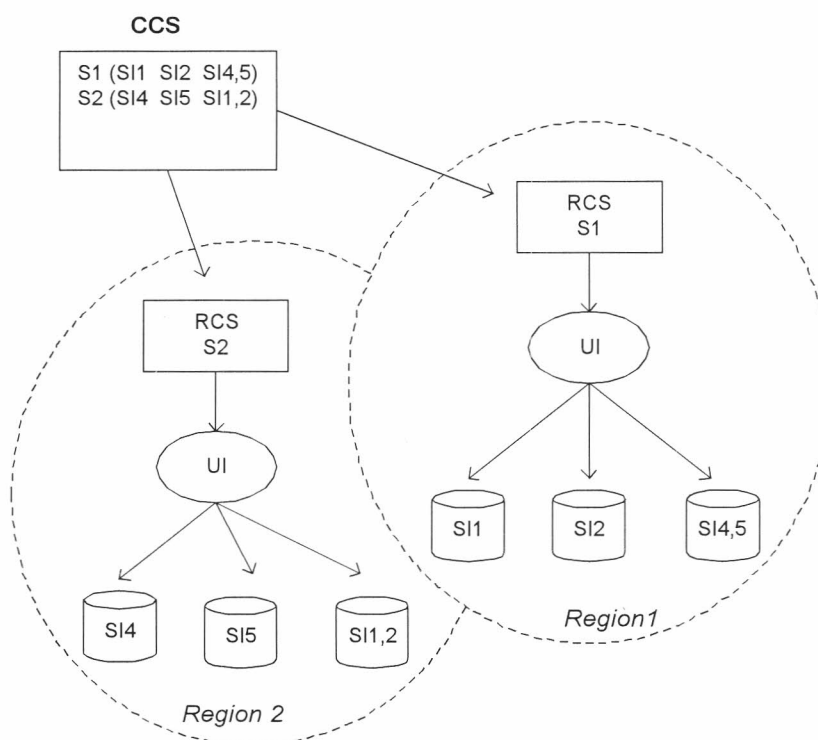


Gráfico 3.16 Collection Service distribuido

Con el objetivo de poder manejar múltiples vistas de la misma colección, se distribuyó el CS dividiéndolo en dos tipos de servidores: *Central Collection Server (CCS)* y *Regional Collection Servers (RCS)*. Como muestra el Gráfico 3.16 los UI's de la región 1 utilizarán la vista de colección proporcionada por el S1 perteneciente a su misma región, mientras que los UI's de la región 2 utilizarán la vista de la colección proporcionada por S2.

Los resultados obtenidos de la redefinición de NCTRL utilizando el concepto de *connectivity regions* fueron buenos, a pesar de que las regiones fueron configuradas estáticamente. Se logró un sistema más confiable, desde la perspectiva de las UI, que es en definitiva la percepción de los usuarios. Por otro lado la arquitectura resultó ser dinámica y fácil de manejar y administrar. Las modificaciones en las tablas del CCS son rápidamente propagadas a los RCS's y de éstos últimos a las UI's de su región [Lagoze 98].

### 3.4 Conclusiones

Si bien consideramos que los servicios centralizados tienen las siguientes ventajas:

- El índice se encuentra en un único sitio lo que permite obtener mejores tiempos de respuesta que en los distribuidos.
- Los problemas de congestión y sobrecarga del servidor pueden evitarse parcialmente replicando el índice.

- La organización que mantiene el índice tiene el control absoluto sobre el servidor pudiendo aplicar las políticas que desee.
- La centralización provee un único punto para la recolección de datos.

El análisis realizado nos permite concluir que las arquitecturas distribuidas poseen mejores propiedades de escalabilidad:

- **Reducción del costo local de procesamiento.** La ventaja más significativa es que el procesamiento y el almacenamiento se distribuye en más de un servidor. De esta manera un único servidor no necesita mantener un índice enorme ni resolver búsquedas masivas. Contar con varios índices permite atacar de una manera más eficiente los problemas de sobrecarga de trabajo, pudiéndose asignar más recursos a los servidores más utilizados [Weider 96].
- **Construcción eficiente de los índices.** En general, los servidores indexarán documentos relativamente locales, pudiendo caracterizar mejor a los recursos. Así el proceso de construcción de cada índice requerirá menor ancho de banda. Cada servidor requerirá acceder a un menor volumen de recursos para generar su índice [Bowman 95].
- **Paralelismo.** Múltiples servidores permiten lograr paralelismo en la construcción de los índices y en la resolución de las consultas.
- **Tolerancia a cambios.** Los índices distribuidos son más tolerante a los cambios en los contenidos. Dado que entre los brokers se reportan solo sumalizaciones (*meta-información de contenido*), los cambios en los contenidos no siempre requerirán una propagación [Sheldon 95a].
- **Autonomía.** Un servicio distribuido permitirá a cada servidor tener el control sobre lo que se indexa y exporta a la red.
- **Distintas estrategias de indexación.** Puesto que cada servidor solo reporta sumalizaciones y acepta requerimientos en un formato estandarizado, es posible que se utilicen diferentes mecanismos para la construcción de índices [Bowman 95].
- **Tolerancia a fallas:** Los servicios distribuidos son potencialmente más tolerantes a fallas. Esta característica se da fundamentalmente en las mallas de brokers no jerárquicas.

## Capítulo 4

# Servicio de búsqueda para ODS

Presentamos un servicio de búsqueda para que la comunidad de usuarios del Sistema de Distribución de Objetos (ODS) pueda localizar y buscar objetos de interés de entre los producidos. Se presentan los objetivos del diseño, la funcionalidad del servicio y se discuten y analizan distintas alternativas para el diseño de la arquitectura. Tomaremos las ideas y características más relevantes de los servicios distribuidos estudiados y las adaptaremos para aprovechar al máximo las características particulares de ODS.

### 4.1 Introducción

Funcionalmente, se proveerán servicios de: *sugerencia de grupos de interés y búsqueda dentro de los grupos de interés*. Dichos servicios son similares a los provistos por el buscador DejaNews [DN], pero a diferencia de éste último que utiliza una arquitectura centralizada, nuestra propuesta es el diseño de una arquitectura distribuida de búsqueda e indexación.

Con el uso del servicio de búsqueda, los usuarios de ODS podrán acceder de una manera más eficiente a los contenidos existentes. Podrán determinar cuales son los grupos de interés a los que deberían suscribirse y descubrir qué objetos específicos de un conjunto dado de grupos de interés satisfacen ciertos criterios de búsqueda. De esta forma los usuarios podrán hacer un mejor uso del sistema de distribución.

Se proveerá búsqueda focalizada en categorías de objetos como en características semánticas y no solo búsqueda por texto libre, dado que a diferencia de la Web y News, el espacio de información de ODS está conformado por objetos *clasificados*.

Los objetos de ODS tienen dos componentes: meta-información (URC) y cuerpo. La existencia de meta-información nos facilitará el proceso de indexación de los objetos. Al ser los propios generadores de información los encargados de confeccionar los URC [Baleani 98], o sea de *catalogar* los objetos que producen, resultan ser de mayor valor para el proceso de indexación.

Por tal motivo los URCs de los objetos serán utilizados para la indexación, recuperación y visualización de los resultados al usuario. Por el contrario, los servicios actuales como AltaVista, Excite o DejaNews tratan a los recursos y a las consultas como colecciones de atributos no estructuradas. Esto significa que los URC's no juegan ningún rol durante el proceso de búsqueda, en el cual la precisión de los resultados se mejora con el uso de heurísticas que intentan interpretar

qué atributos tienen mayor relevancia semántica para la consulta. Durante la visualización de resultados, estos servicios construyen un URC informal compuesto generalmente por el URL, título de la página Web y algún resumen textual extraído en base a alguna heurística predefinida [Lagoze 97].

Los objetos de ODS son *persistentes*, por lo que se mantienen replicados en todos los Agentes de Servicio (SA) interesados (como en una biblioteca). Los SA consumidores de un mismo grupo serán considerados como réplicas completas de los contenidos del grupo y utilizados para la resolución distribuida de las consultas. A su vez, cada SA podrá proveer a su comunidad de usuarios servicios de valor agregado.

## 4.2 Objetivos del diseño

La arquitectura deberá cumplir los siguientes objetivos:

**Ser escalable.** Soportar, con un mínimo aumento de costo administrativo y degradación de performance, la adición de servidores de índices y de usuarios así como el gran aumento del volumen de información y cantidad de grupos de interés.

**Ser adaptativa.** Adaptarse fácilmente a condiciones de entorno cambiantes, por ejemplo alta/baja de grupos de interés.

**Utilizar características particulares de ODS.** Aprovechar al máximo las características ya disponibles, evitando malgastar recursos de red y duplicar esfuerzos.

**Preservar la autonomía de los SA.** No deben imponerse fuertes restricciones a los SA con el objetivo de facilitar su participación. En ODS no existe una única entidad administrativa para todos los SA, por el contrario la administración está descentralizada. Los SA no manejan la misma información sino solo la que es de interés para sus usuarios.

Analizaremos a continuación los aspectos particulares de ODS que nos llevarán a definir las características de la arquitectura a utilizar.

En función de la definición de *Sistemas para el Descubrimiento de Recursos* vista en el “Capítulo 2”, las entidades de ODS involucradas en el servicio de búsqueda son:

**Usuarios:** cualquier potencial usuario que se conecte a un SA para utilizar los servicios de ODS. Están distribuidos geográficamente.

**Recursos:** cualquier objeto que pueda ser producido en ODS.

**Proveedores de información:** son los SAs, al ser las entidades encargadas de proveer a los usuarios el acceso y la producción de los recursos del sistema. Los SA de ODN (Object Distribution Network) están ampliamente distribuidos en Internet.

**Espacio de información:** es aquel conformado por el conjunto de recursos existentes para todos los grupos de interés disponibles en el sistema. Es amplio y sus contenidos están distribuidos en los SA productores y consumidores.

Las características de entorno de ODS a tener en cuenta en el diseño de la arquitectura son:

**Volumen de información.** Consideramos que el volumen de información que circulará en ODS puede llegar a ser considerable tomando como base el tráfico existente hoy en News que según [DN] ronda los 900 MB de información por día para un total de 15.000 grupos de interés;

**Espacio de información distribuido.** La información en ODS no se produce en un único sitio. Cualquier SA del sistema puede ser un potencial productor de objetos para uno o más grupos de interés;

**Amplia distribución de usuarios.** Existirá un gran número de usuarios distribuidos geográficamente accediendo al sistema a través de alguno de los SA de ODN.

Una arquitectura centralizada podría aplicarse fácilmente en ODS como muestra del Gráfico 4.1. El servicio podría estar implementado por un SA dedicado que recolecte los objetos producidos en el sistema, indexe su meta-información (URC del recurso) en un índice local y provea una interfaz de consulta para toda la comunidad de usuarios.

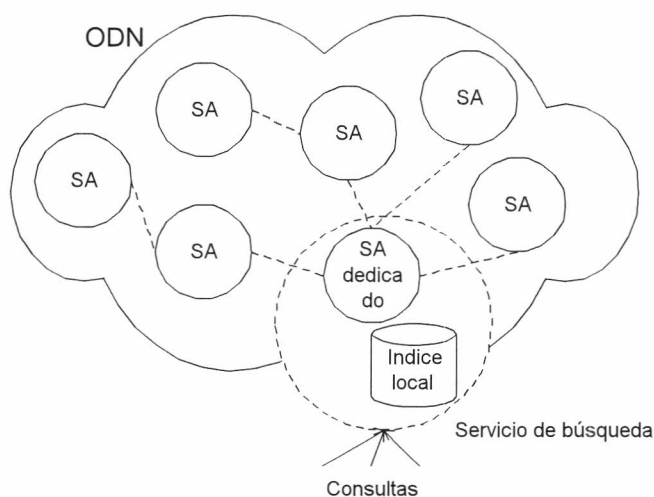


Gráfico 4.1 Arquitectura centralizada en ODS

Para la recolección de objetos, el SA proveedor del servicio debería comportarse, dentro de ODS, como consumidor de meta-información de objetos para todos los grupos de interés existentes. De esta manera serían los mecanismos de replicación propios de ODS los que implementen el proceso de recolección. Una vez recibida la meta-información de los objetos (los URCs), el SA las incorporaría a su índice local. Los usuarios de ODS contactarían al SA proveedor del servicio para satisfacer sus requerimientos de búsqueda de información. El índice sería utilizado para la resolución de las consultas de los usuarios.

Independientemente de la fácil implementación de este modelo en el contexto de ODS, esta solución adolece de los mismos problemas detectados en las soluciones centralizadas existentes en Internet, tanto para la Web como para News. El motivo es que en ODS están presentes las mismas características de entorno que en la Web y News convierten a estas soluciones en inapropiadas.

Con el objetivo de que el servicio resultante sea *escalable*, discutiremos alternativas para el diseño de una arquitectura de indexación y búsqueda distribuida en la cual se contemple la existencia de múltiples servidores de índices y mecanismos de comunicación para la resolución cooperativa de las consultas. Tomaremos las ideas y características más relevantes de varios prototipos distribuidos existentes para la Web (ver “Apéndice A”) y Bibliotecas Digitales (ver “3.3.2 Bibliotecas Digitales”) y las adaptaremos para aprovechar al máximo las características particulares de ODS.

## 4.3 Servicios

El servicio de búsqueda proveerá la siguiente funcionalidad:

- 1) **Sugerencia de grupos de interés.** Orientar a los usuarios sobre cuales son los grupos de interés a los que deberían suscribirse en función de los intereses expresados en la consulta, evitando suscripciones a grupos que no cubran sus expectativas. Esta facilidad permitirá mejorar el uso del sistema de distribución, al no malgastar recursos de red para replicar objetos que no sean útiles a los usuarios finales. Permitirá a los usuarios tener un acceso más directo a los grupos de interés más relevantes, evitando que deban *navegar* por los contenidos de distintos grupos para determinar si contienen información que les es relevante.
- 2) **Búsqueda dentro de los grupos de interés.** Permitirá a los usuarios focalizar sus consultas en ciertos grupos de interés, es decir en ciertas categorías de objetos. Resolver una consulta para un grupo de interés dado es la selección de los objetos producidos para el grupo que satisfagan los criterios de búsqueda.

El servicio de búsqueda abarcará contenidos globales: el hecho de que un usuario consuma o no determinado grupo de interés no restringirá sus posibilidades de búsqueda. A continuación, centraremos nuestro análisis en alternativas para implementar los servicios descriptos.

### 4.3.1 Sugerencia de grupos de interés

Este servicio tiene por objetivo sugerir a los usuarios los grupos de interés a los cuales deberían suscribirse en función de los intereses que expresen a través de una consulta.



Para estar en condiciones de sugerir los *mejores*<sup>1</sup> grupos para una consulta dada, es necesario tener algún conocimiento acerca de los contenidos que circulen en cada grupo de interés. Este conocimiento debe ser tal que, a partir de una consulta específica, se pueda discriminar un grupo de otro y quedarse solo con aquellos que mejor puedan satisfacerla (filtrado de grupos).

Conceptualmente, el objetivo de sugerir los *mejores* grupos es similar al proceso de *query routing* [Sheldon 95a] utilizado por las arquitecturas distribuidas para la selección de los mejores servidores. Mientras que una estrategia de *query routing* se utiliza para seleccionar los *mejores* servidores a los cuales reenviar una consulta, el filtrado de grupos de interés se utilizará para seleccionar los *mejores* grupos a los que el usuario deberá suscribirse.

En las arquitecturas que utilizan mallas de brokers (ver “Apéndice A”), tanto las jerárquicas como las no jerárquicas, una colección está definida por el conjunto de recursos almacenados en un servidor. Estas utilizan meta-información de contenido de colecciones para la implementación de estrategias de *query routing*. La indexación de la meta-información de contenido les permite la selección de las mejores colecciones para una consulta dada, tratando a éstas como a cualquier objeto.

En nuestro caso, la definición de colección es diferente, entendiendo la misma como el conjunto de objetos producidos para un grupo de interés dado, que pueden encontrarse en varios SA. En consecuencia, tendremos una colección por cada grupo de interés existente. Cada una de estas colecciones estará descripta por meta-información de contenido, que en nuestro contexto llamaremos *Meta-información de Grupo (MiG)*. Veremos a la MiG de los grupos como conjuntos de atributos, de manera tal que a través de una consulta puedan localizarse los grupos deseados. La MiG nos permitirá ver a una colección como a cualquier otro objeto.

A diferencia de las arquitecturas de mallas de brokers, donde el proceso de sumarización de la meta-información de contenido es función de cada nodo del sistema, nuestro caso presenta diferencias por manejar un concepto de colección diferente.

En ODS, cualquier SA puede ser proveedor de información para uno o más grupos de interés. Esto significa que los objetos producidos para un grupo de interés, por ejemplo X, estarán distribuidos en muchos SA, exactamente en cada SA productor de X (abreviado como P(X)) lo que dificulta el proceso de sumarización de la MiG(X).

Cada SA consumidor de X (abreviado como C(X)) mantiene una copia de los objetos producidos para el grupo X. De esta manera, cada SA C(X) concentra la totalidad de los objetos producidos a nivel global para X. Podríamos utilizar esta característica para facilitar el proceso de sumarización de MiG(X), considerando a cualquier SA C(X) como potencial generador de la MiG(X) necesaria para el servicio de búsqueda.

---

<sup>1</sup> Los mejores grupos para una consulta serán aquellos que satisfagan los criterios de selección expresados en la consulta.

Como ODS utiliza un mecanismo de replicación de consistencia débil [Obraczka 94], podría darse el caso de que un SA  $C(X)$  genere la  $MiG(X)$  usando un pequeño porcentaje del total de objetos existentes para  $X$ . Varias situaciones pueden hacer que un SA  $C(X)$  no contenga todos los objetos producidos para  $X$ : sus enlaces se caigan con frecuencia; el nodo no funcione correctamente; etc.

Dado que la  $MiG$  es fundamental como herramienta de filtrado de grupos, si utilizáramos los SA  $C(X)$  como generadores de  $MiG(X)$ , correríamos el riesgo de que la  $MiG(X)$  no refleje la totalidad de los contenidos del grupo  $X$  con la consiguiente pérdida de funcionalidad del servicio de búsqueda. Además, no necesariamente siempre existirá un SA  $C(X)$  para cada  $X$ , ya que esto depende de las suscripciones de los usuarios.

El sistema utilizará *Meta-información de Grupo* ( $MiG$ ) para cada grupo de interés existente en ODS. El proceso de generación de la  $MiG$  **no** involucrará a los SA consumidores, utilizando para su generación exclusivamente a los SA productores. La  $MiG$  podrá ser indexada como cualquier otro objeto. Indexando la  $MiG$  de cada grupo, sugerir los mejores grupos de interés para una consulta dada (filtrado de grupos) significará recuperar las  $MiGs$  que mejor puedan satisfacerla obteniendo así los grupos a sugerir.

#### 4.3.2 Búsqueda dentro de los grupos de interés

Este servicio tiene por objetivo que los usuarios puedan focalizar sus consultas sobre ciertos grupos de interés. Los usuarios obtendrán un listado de los objetos pertenecientes a los grupos involucrados en la consulta que satisfagan los criterios de búsqueda. Al tener la posibilidad de *descubrir* que objetos de un grupo satisfacen los criterios de búsqueda, los usuarios podrán determinar si los objetos producidos para el grupo son de su interés. En el caso que así sea, deberán suscribirse en el SA que les brinda el servicio para disponer de una copia local, caso contrario deberán recuperarlos directamente desde los SA productores.

La idea es que este servicio se utilice conjuntamente con el de sugerencia de grupos de interés. A modo de ejemplo, la interacción de un usuario con el servicio de búsqueda sería:

- 1) especificar los criterios de búsqueda para que el sistema sugiera potenciales grupos de interés,
- 2) focalizar la misma consulta en los grupos sugeridos,
- 3) analizar los resultados,
- 4) decidir a que grupo suscribirse o eventualmente recuperar un objeto específico directamente desde el SA productor.

Aunque esta interacción es la más adecuada, podría darse el caso de que los usuarios no realicen un filtrado de grupos antes de remitir una consulta. Esta situación haría que la consulta deba resolverse sobre la totalidad de los grupos disponibles. Este tipo de consultas *deben* evitarse

teniendo en cuenta que la cantidad de grupos de interés puede ser grande<sup>1</sup>. Además, seguramente no todos los grupos sean relevantes para la consulta.

Para manejar estas situaciones adecuadamente, cuando el sistema considere que los grupos involucrados en la consulta son demasiados, realizará un filtrado de grupos antes de procesarla logrando así que el proceso de resolución de las consultas sea escalable. Utilizaremos la Meta-información de Grupo a la hora de resolver una consulta, recortando el conjunto de grupos sobre el cual aplicarla.

## 4.4 Resolución distribuida de consultas

Analizaremos distintas alternativas de resolución distribuida de consultas para grupos de interés.

A diferencia de las arquitecturas distribuidas de mallas de brokers, las búsquedas en ODS están orientadas a ciertas categorías de objetos, o lo que es lo mismo: a ciertos grupos de interés. La resolución de una consulta debe darse en el contexto de un grupo de interés, esto significa que la respuesta solo debe contener referencias a objetos que pertenezcan a los grupos involucrados en la consulta. En consecuencia, la agrupación de la información debe darse a nivel grupo de interés.

En el caso de las mallas de brokers jerárquicas (ver “Apéndice A”), la composición jerárquica de brokers se establece en función a los criterios definidos por la organización que la administra. Estos criterios definen como se organiza el espacio de información para la posterior resolución de consultas.

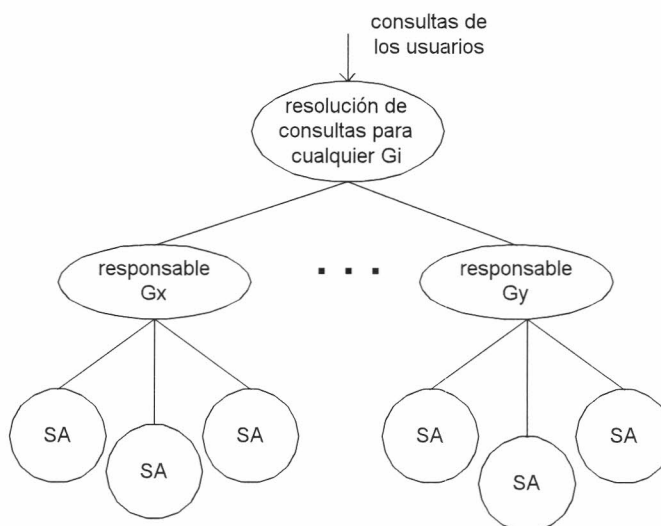


Gráfico 4.2 Arquitectura jerárquica para resolución de consultas en ODS

Si quisiéramos utilizar una jerarquía para la resolución de consultas en el contexto de ODS, alcanzaría con una estructura de tres niveles como muestra el Gráfico 4.2. Los SA de ODN

<sup>1</sup> Recordar que en News son aproximadamente 15.000, según datos extraídos de DejaNews [DN].

conformarían el nivel inferior. En el segundo nivel tendríamos un nodo responsable para cada grupo de interés existente encargado de resolver las consultas para el grupo que administre. Por último, el primer nivel tendría un único nodo encargado de resolver las consultas de los usuarios, derivándolas a los nodos responsables correspondientes para su resolución y combinando las respuestas para devolver el resultado al usuario.

#### 4.4.1 Estrategias de resolución

En toda arquitectura de búsqueda distribuida el mecanismo de resolución de consultas es uno de los aspectos más importantes. Generalmente, de éste dependen las propiedades de escalabilidad que pueda tener la arquitectura que lo implemente.

Dado que en ODS pueden existir varios SA productores distribuidos en la red para un mismo grupo de interés, se pueden plantear distintas estrategias para que el nodo responsable del grupo resuelva las consultas.

##### 4.4.1.1 Alternativa 1: Broadcast a los SA productores

Una alternativa es que el responsable del grupo X conozca los SA  $P(X)$  existentes y resuelva las consultas mediante un broadcast global a todos como muestra el Gráfico 4.3. Este mecanismo de resolución de la consulta puede resultar muy costoso, en términos de cantidad de mensajes en la red y tiempo requerido para el procesamiento de las respuestas cuando la cantidad de SA  $P(X)$  sea considerable.

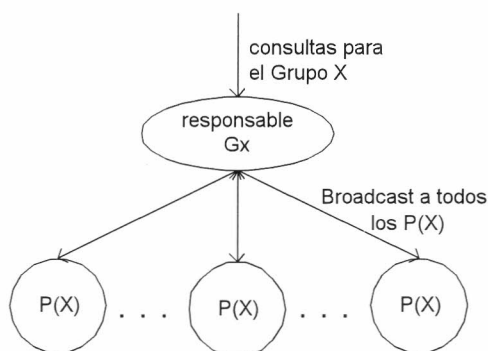
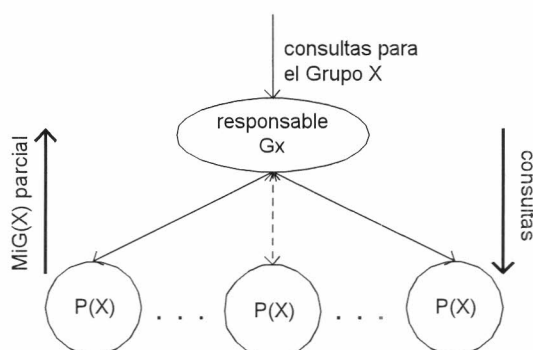


Gráfico 4.3 Broadcast a SA  $P(X)$

##### 4.4.1.2 Alternativa 2: Query routing a los mejores SA productores

Otra alternativa es que cada SA  $P(X)$  genere, para el responsable del grupo X, una descripción de alto nivel que describa los objetos producidos para X (una especie de  $MiG(X)$  parcial, restringida solo a lo producido para X en un SA). De esta manera, el responsable del grupo tendría la función de un *broker* utilizando la  $MiG(X)$  parcial provista por cada SA  $P(X)$  para implementar el query routing. Así las consultas se reenviarían solo a los SA  $P(X)$  más prometedores y luego se combinarían las respuestas para generar un resultado como muestra el Gráfico 4.4.

Gráfico 4.4 Query routing a los mejores SA  $P(X)$ .

#### 4.4.1.3 Alternativa 3: Mantener un índice local para todos los objetos de un grupo

Como última alternativa podemos pensar que el responsable del grupo recolecte los objetos producidos para el grupo o mejor solo la meta-información de los objetos (URC's), por ejemplo suscribiéndose como consumidor de meta-información dentro de ODN. Para resolver las consultas, alcanzaría con que el responsable construya un índice y provea facilidades de búsqueda como muestra el Gráfico 4.5.

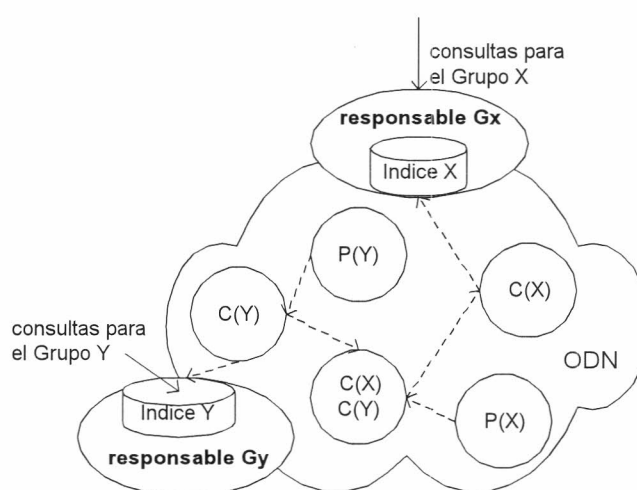


Gráfico 4.5 Responsables como consumidores de grupo

Esta situación haría que cada responsable de grupo se comporte como un servicio centralizado pero **sólo** para el grupo que represente. Como hemos visto, una de las ventajas de los servicios centralizados es la eficiencia del proceso de resolución de una consulta ya que basta con acceder al índice local. Esto no significa que estemos proponiendo una arquitectura centralizada. Es importante dejar en claro que la centralización de información sería solo para el grupo administrado por el responsable.

#### 4.4.1.4 Discusión

Tanto en la alternativa 1 como en la 2, los SA  $P(X)$  deberían indexar los documentos producidos para el grupo X y proveer facilidades de búsqueda. Toda consulta para el grupo X implicaría un

procesamiento distribuido de la misma. La ventaja de la alternativa 2 es que siempre se contactaría sólo a un subconjunto del total de SA  $P(X)$ , los que se consideren *mejores*.

Con una solución como la planteada en la alternativa 3, podría darse el caso que las consultas para un grupo dado se resuelvan accediendo a un índice que no contemple *todos* los documentos existentes para el grupo. Esto es posible por los mecanismos de replicación utilizados por ODS. Con las alternativas 1 y 2 esta situación solo se daría si un SA  $P(X)$  no se encontrara disponible al momento de la consulta.

A pesar de que la utilización de una estrategia como la explicada en la alternativa 1 garantizaría a los usuarios del servicio una respuesta exhaustiva, ésta sería a costa de sacrificar la performance del sistema (mucho tiempo para procesar las consultas, mal uso de recursos de red). Generalmente los usuarios no están dispuestos a esperar un tiempo excesivamente largo para obtener una respuesta. La mayoría de las veces alcanza con una respuesta parcial que involucre solo algunos objetos representativos. Entendemos, en función de nuestra experiencia en el uso de servicios de búsqueda, que para un usuario es más útil obtener respuestas rápidas aunque no sean exhaustivas, ya que podrán ser utilizadas como punto de partida. Por otro lado, en el caso puntual de ODS, utilizar respuestas parciales no afectaría la provisión de los servicios descriptos. Este tipo de respuestas alcanzará para orientar a los usuarios sobre los contenidos de los grupos.

Resolver consultas mediante broadcast a los SA  $P(X)$  no es una estrategia escalable [Neuman 94], motivo por el cual **no** la utilizaremos en la arquitectura. Más adelante serán evaluadas las otras dos alternativas.

#### 4.4.2 Características particulares de ODS

En ninguna de las estrategias descriptas “4.4.1 Estrategias de resolución” se utilizan características propias de ODS. Sin embargo, ODS contiene un conjunto de características específicas cuya utilización puede resultar beneficiosa para el servicio.

Las características de ODS que consideramos más relevantes son:

**Existencia de múltiples réplicas para un mismo grupo de interés.** Todos los SA consumidores de un grupo de interés mantienen una copia local de los objetos producidos para el grupo (recordar que los objetos de ODS son persistentes). En consecuencia existirán múltiples réplicas distribuidas para cada grupo. Si además, cada SA consumidor indexara la meta-información de los objetos consumidos, existirán múltiples *Servidores de Índices (SI)* para cada grupo de interés distribuidos en Internet.

**Disponibilidad de información local en los SA.** Los usuarios utilizan el sistema de distribución accediendo a un SA local que les provee servicios de suscripción y acceso a grupos. La información disponible localmente en cada SA puede utilizarse para proveer servicios de valor agregado a la comunidad de usuarios, por ejemplo búsqueda restringida a los grupos consumidos.

**Objetos clasificados.** El hecho de que los objetos tengan asociado un URC permitirá que los SA puedan indexar los grupos de interés sin necesidad de tener que generar explícitamente una descripción para cada objeto. Esto garantizará que todos los SA utilicen los URC generados por los propios productores de información.

Analizaremos las ventajas que pueden obtenerse de la utilización de dichas características.

#### 4.4.2.1 Disponibilidad de información local en los SA

A través del SA local, los usuarios de ODS se suscriben a grupos de interés para recibir los objetos que se produzcan globalmente para el grupo, evitando tener que transferirlos desde los sitios remotos. Cada SA del sistema mantiene localmente una copia de todo lo producido para cada grupo consumido por su comunidad de usuarios. Esta información disponible localmente en cada SA, que refleja los intereses de los usuarios que lo utilizan, puede resultar muy útil a la hora de resolver consultas sobre grupos de interés.

¿Qué ventajas podrían obtener los usuarios de un SA si éste construyera un índice para los grupos consumidos y proveyera una interfaz de consulta?. En primer lugar cada SA podría brindar servicios de valor agregado específicos para sus usuarios, como búsqueda restringida a los grupos consumidos por éstos. En segundo lugar, como cada SA es compartido por muchos usuarios, las consultas podrían resolverse utilizando todos los datos disponibles, derivando a otros servidores solo las consultas que no puedan resolverse localmente como muestra el Gráfico 4.6.

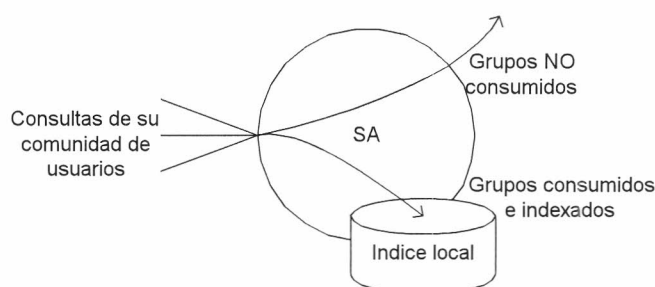


Gráfico 4.6 Uso de información local

Si tomamos como base que cada SA atiende a usuarios con intereses similares, utilizar información local para la resolución de consultas presenta ventajas comparables a las obtenidas con el uso de servidores Proxies [Luotonen 94]. Estos, ante el pedido de una página Web, primero la buscan localmente en su cache y solo en caso de no disponer de la información, van a buscarla al servidor remoto y la guardan en la cache para futuros accesos. De la misma manera, al indexar los grupos consumidos, un SA estaría en condiciones de resolver muchas consultas en forma local, y distribuir solo un subconjunto del total de consultas lográndose una mejor performance en tiempos de respuesta.

Sin embargo, la construcción de un índice requiere disponibilidad de recursos informáticos como espacio en disco, capacidad de procesamiento, etc. Dependiendo del número de grupos



consumidos y del volumen de información que circule en éstos, la demanda de recursos para el SA puede ser importante.

A pesar de que los costos del hardware y de enlaces de alta velocidad están disminuyendo a nivel mundial, muchos países disponen de poco presupuesto para invertir en tecnología. Esta situación se da fundamentalmente en los países del tercer mundo, siendo este el caso de Argentina.

Con el objetivo de preservar al máximo la autonomía de los SA no debemos imponer fuertes restricciones que impidan su participación. Por este motivo cada SA tendrá la libertad de decidir construir o no un índice local para los grupos consumidos, en función de los recursos informáticos disponibles y la similitud de intereses de su comunidad de usuarios.

A modo de ejemplo, un SA con escasos recursos podría decidir indexar solo aquellos grupos consumidos para los cuales tenga un elevado promedio de consultas y para el resto de los grupos siempre derivar las consultas a otros servidores. De la misma forma, si un SA atiende a usuarios con intereses muy diferentes, es probable que la mayoría de las consultas deba derivarlas, no teniendo sentido gastar recursos locales para mantener un índice que no lo beneficia.

Para poder utilizar estas características, nuestra propuesta consiste en que los SA provean el punto de acceso al servicio de búsqueda. Esto permitirá mejorar la performance del servicio, ya que cada SA podrá resolver localmente muchas de las consultas derivando solo unas pocas.

Además, cada SA podrá proveer servicios adicionales específicos para su comunidad de usuarios. Algunos ejemplos son: búsqueda restringida solo a los grupos consumidos; interfaces de consulta personalizadas que utilicen un cierto lenguaje, pantallas de ayuda, etc..

Las mínimas funcionalidades que hemos previsto para cada SA son:

*a) Obligatorias*

Constituye el conjunto de tareas que debe realizar un SA para que el servicio de búsqueda pueda funcionar, estas son:

*a.1)* Proveer una interfaz de consulta a la comunidad de usuarios que atiende.

*a.2)* Implementar el mecanismo de resolución de consultas.

*b) Opcionales:*

Constituye el conjunto de tareas optativas para los SA. Su implementación mejorará la performance y calidad del servicio de búsqueda, aprovechando información disponible localmente para la resolución de consultas. Por otro lado, permitirá a los SA's proveer servicios de valor agregado a su comunidad de usuarios.

b.1) Indexar la meta-información de los objetos de algunos o todos los grupos consumidos por el SA.

Con estas funcionalidades será responsabilidad de los SA resolver, para su comunidad de usuarios, las consultas sobre cualquier grupo sea o no consumido por el SA. El mecanismo de resolución de consultas deberá ser tal que priorice el uso de información disponible localmente. Dicho mecanismo se explica con mayor nivel de detalle en el “Capítulo 5”.

#### 4.4.2.2 Múltiples réplicas para los grupos de interés

Habiendo definido que los SA serán los encargados de proveer el punto de acceso al servicio de búsqueda y de implementar el mecanismo de resolución de consultas, queda por analizar como éstos resolverán las consultas que involucren grupos no indexados localmente.

Los SA podrán decidir en cualquier momento indexar algunos o todos los grupos consumidos para reducir los costos que implica una resolución externa de las consultas. Aquellos SA que indexen grupos serán considerados por el sistema como servidores de índices para determinados grupos a la hora de resolver en forma cooperativa consultas provenientes de otros SA.

Cuando un SA deba resolver una consulta sobre grupos no disponibles en su índice podría utilizar la arquitectura de resolución descrita en el Gráfico 4.2. Así, el SA derivaría todas las consultas externas al nodo raíz de la jerarquía para su resolución. Otra forma alternativa sería que los SA conozcan a los responsables de grupo y les reenvíen las consultas externas directamente a éstos según el Gráfico 4.7.

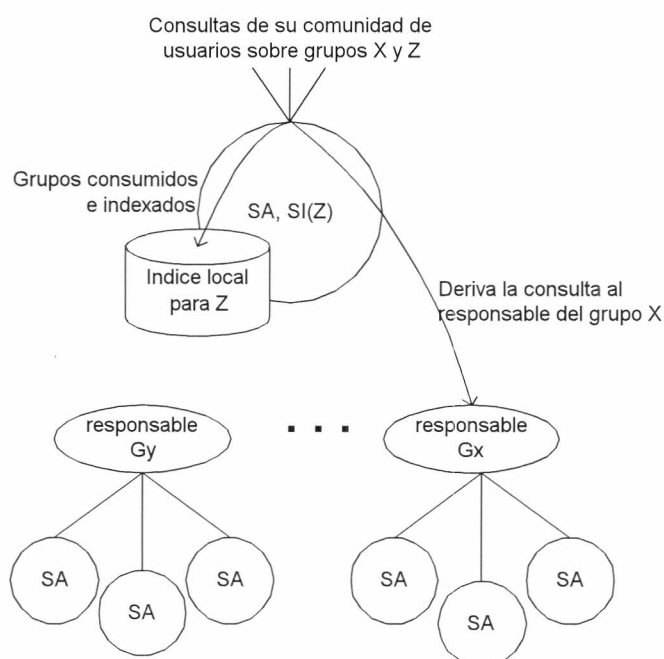


Gráfico 4.7 Resolución de consultas externas de los SAs

Las alternativas analizadas para que un responsable de grupo resuelva las consultas eran:

- 1) Query routing a los mejores SA  $P(X)$ .
- 2) Mantener un índice local para todos los objetos del grupo.

El hecho de que en ODS puedan existir múltiples SA consumidores de un mismo grupo de interés hace que nos replanteemos la segunda alternativa de la siguiente manera:

Sean  $X$  e  $Y$  dos grupos de interés. Sean  $C(X)$  y  $C(Y)$  SAs consumidores de  $X$  e  $Y$  respectivamente. Cada  $C(X)$  es una réplica disponible para  $X$ , y pueden existir varios distribuidos geográficamente en Internet. Si agrupáramos lógicamente a todos los  $C(X)$ , podemos ver que en ODS existirán  $N$  réplicas para  $X$ , siendo  $N$  la cantidad de  $C(X)$  disponibles como muestra el Gráfico 4.8. El mismo razonamiento puede hacerse para  $C(Y)$ .

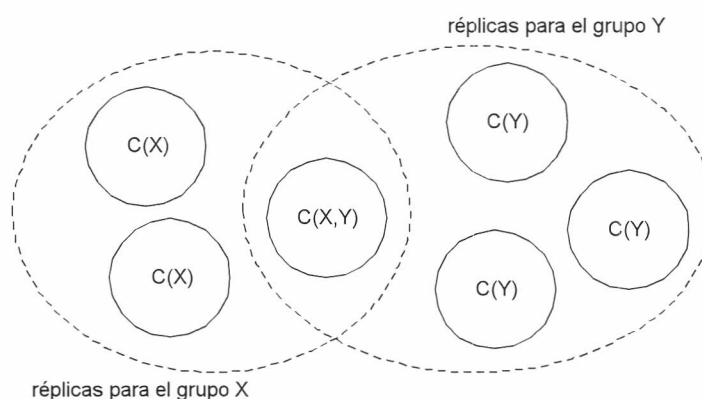


Gráfico 4.8 Grupos de réplicas disponibles

Suponiendo además que todos los  $C(X)$  y  $C(Y)$  decidieran indexar los grupos consumidos, podemos ver a estos conjuntos como *Servidores de Índices (SI)* para  $X$  e  $Y$  como muestra el Gráfico 4.9.

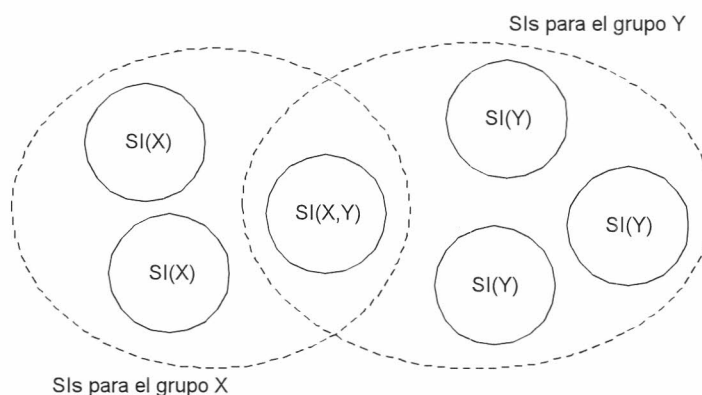


Gráfico 4.9 Múltiples servidores de índices para un grupo de interés

De esta manera, en lugar de que el responsable de grupo se comporte como consumidor y genere un índice para resolver las consultas, podría utilizar alguno de los servidores de índices disponibles para el grupo y derivarle la consulta para su resolución. Esta nueva alternativa evitará

al responsable tener que consumir el grupo y generar un nuevo índice cuando en el sistema exista por lo menos un servidor de índices para el grupo, haciendo un mejor uso de los recursos disponibles y evitando duplicar esfuerzos. Además, los SA que construyan índices funcionarán como servidores de índices del sistema a los efectos de cooperar en la resolución de las consultas de los SA como se muestra en la Gráfico 4.10.

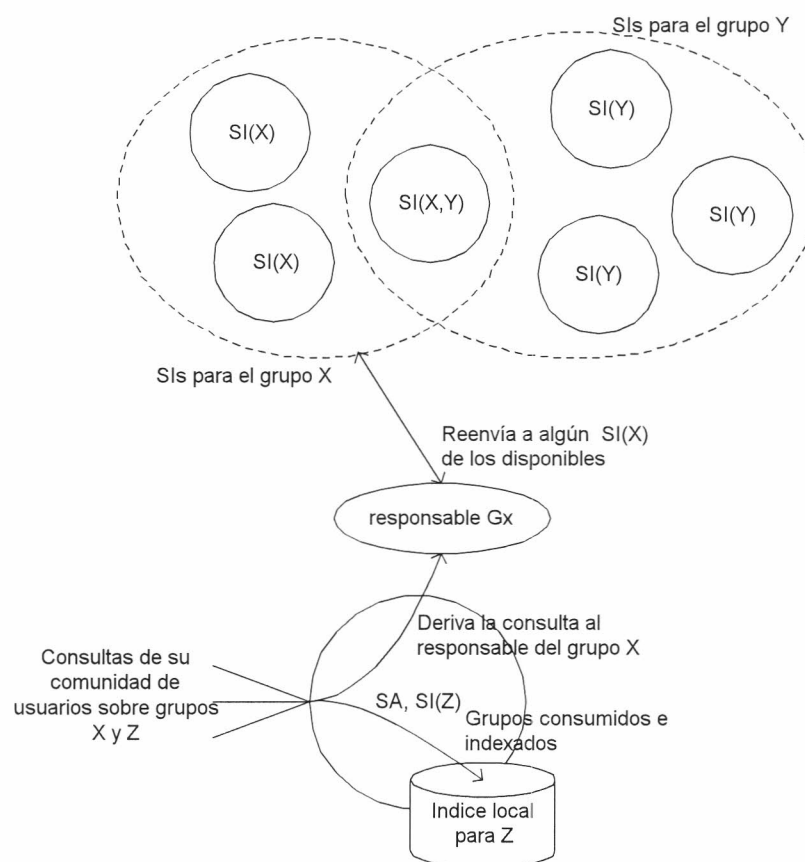


Gráfico 4.10 Resolución cooperativa de las consultas

En el caso que para algún grupo no existiera ningún servidor de índices disponible, el responsable del grupo deberá garantizar la existencia de por lo menos uno. Esta situación puede darse ya que un SA consumirá o no determinado grupo de interés dependiendo de la suscripción de sus usuarios, a la vez que podrá optar por generar un índice. Para garantizar la existencia de un SI por grupo, una posibilidad es que algún SA consumidor del grupo genere un índice.

En lo que respecta a cómo el responsable de un grupo *elige* el servidor de índice al cual derivar una consulta (del total disponible para el grupo), manejaremos alguna estrategia de query routing adaptativo. [Lagoze 98] define el *query routing adaptativo* a la capacidad de que las decisiones de ruteo a las réplicas se basen en información específica sobre los SI (refresh, performance, etc.), que en nuestro caso estarán mantenidas y administradas por los responsables de grupo. Es *adaptativo* en el sentido que la selección del SI puede ser diferente para cada consulta. Analizaremos en la especificación de la arquitectura distintas alternativas de query routing adaptativo basándonos en los resultados obtenidos en el uso de la arquitectura Dienst [Lagoze 95b] en el contexto del proyecto NCSTRL [NCSTRL].

Esta nueva alternativa será más eficiente que utilizar directamente a los SA  $P(X)$  para resolver las consultas del grupo  $X$  (recordar la alternativa 1). Si se utilizara directamente a los SA  $P(X)$ , las consultas siempre deberían resolverse involucrando a más de un SA. El responsable del grupo  $X$ , a partir de una consulta, seleccionaría los mejores SA  $P(X)$  (query routing), les enviaría la consulta para luego procesar las respuestas y devolver el resultado.

Al utilizar los servidores de índices disponibles para  $X$ , el responsable de grupo puede resolver las consultas contactando solo a un SI( $X$ ) para obtener los resultados. En este caso, alcanza con que el responsable de grupo tenga la capacidad de identificar a los SI( $X$ ) disponibles e implemente algún algoritmo de query routing adaptativo para la selección de alguno de ellos.

En función de estas ventajas, nuestra propuesta utilizará los SI disponibles para cada grupo para la resolución de las consultas externas de los SA.

#### **4.4.2.2.1 Agentes de Búsqueda**

Hasta el momento, la resolución de consultas externas de los SA se planteó como función de los responsables de grupo. Estos tienen las funciones de conocer a los SI disponibles y de implementar la selección de algún SI para responder las consultas provenientes de los SA.

Tener un único responsable por grupo para la resolución de consultas, tiene el problema de que las decisiones de query routing adaptativo que éste pueda tomar serán siempre las mismas para cualquier SA. El responsable siempre utilizará el mismo conjunto de SI y la misma política de selección para procesar las consultas de cualquier SA. Además, la única posibilidad para un SA de resolver una consulta sobre  $N$  grupos es contactando a cada uno de los responsables correspondientes.

La posibilidad de utilizar diferentes conjuntos de SI (de entre los disponibles) para un mismo grupo de interés, nos permitiría implementar el concepto de *connectivity regions* definido en [Lagoze 98]. Por otro lado, distintos SA podrían querer utilizar distintas políticas para guiar las decisiones de query routing adaptativo, por ejemplo utilizar los SI más actualizados, los que tengan mejor conectividad, etc. Para este último caso se podría utilizar la idea de los gestores de federación definidos en AlephWeb [Rodríguez 96].

Si agrupáramos a los SA en distintas comunidades coordinadas por una nueva entidad responsable de la resolución de *sus* consultas, podríamos hacer que las decisiones de query routing adaptativo se realicen en función de políticas locales establecidas en base a diferentes conjuntos de SI.

Cada una de estas nuevas entidades utilizaría su propia estrategia para la selección de los SI que resolverán las consultas de los SA que administren. Así, las decisiones de query routing adaptativo serán iguales sólo para los SA que utilicen la misma entidad para resolver las consultas externas.

En función de estas ventajas hemos decidido agrupar a los distintos SA en comunidades coordinadas por una nueva entidad que denominaremos *Agente de Búsqueda (AB)*. Los AB serán utilizados para la resolución distribuida de consultas brindando servicios de búsqueda distribuida a un conjunto de SA, administrando y manteniendo información global sobre el servicio. En el Gráfico 4.11 se observan dos conjuntos de SA, cada uno de los cuales utiliza un AB diferente encargado de resolver sus consultas externas. En el caso de tener que resolver consultas sobre el grupo X, cada AB podrá basar sus decisiones de query routing adaptativo sobre diferentes conjuntos de SI(X) de todos los disponibles.

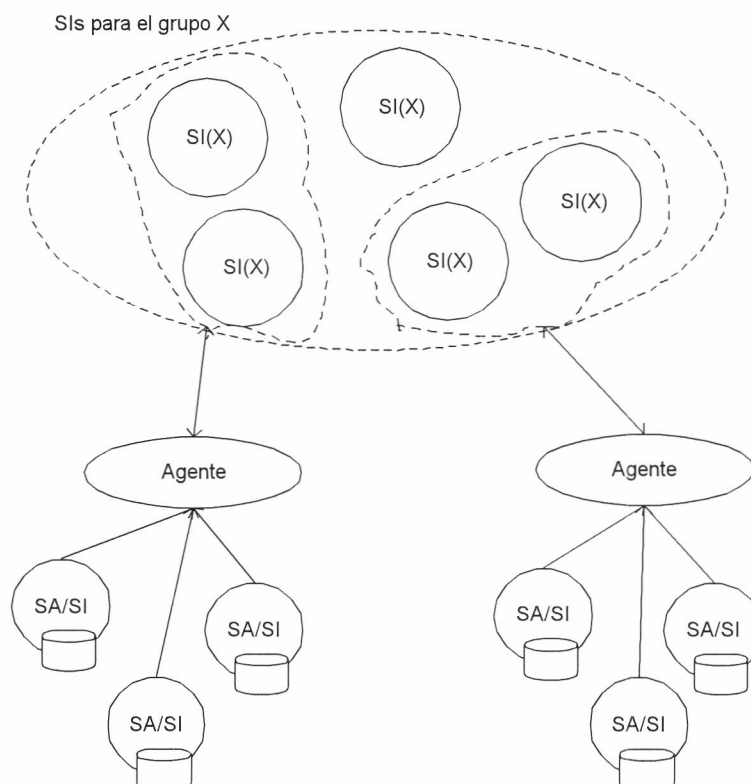


Gráfico 4.11 Agentes de Búsqueda

Esta idea es conceptualmente similar a la utilizada en [Lagoze 98] para la definición de los Collection Regional Servers (CRS) vista en el capítulo anterior. Los CRS fueron definidos para poder manejar el concepto de connectivity regions, de manera tal que cada uno utilice un conjunto diferente de SI (los que tienen mejor conectividad) para que los UI de su misma región resuelvan las consultas. A diferencia de los UI de Dienst, proponemos que las decisiones de query routing a los SI de cada grupo sea una función específica de cada AB y no de los SA. Esta característica permitirá a los SA independizarse del entorno global y delegar dicha función a su AB. Por otro lado, cada AB podrá utilizar diferentes conjuntos de SI para cada grupo de interés, seleccionados en función de alguna política local.

En lo que respecta a los responsables de grupo, seguirán existiendo pero su función se restringirá exclusivamente a tareas de coordinación global: mantener el conocimiento de los SI disponibles para cada grupo y la generación de la MiG. En cuanto a la resolución distribuida de consultas, será una actividad exclusiva de los Agentes de Búsqueda.

## Capítulo 5

# Arquitectura propuesta

Definiremos los componentes y el funcionamiento de la arquitectura de búsqueda e indexación distribuida propuesta para el servicio de búsqueda para ODS.

### 5.1 Introducción

Como resultado del análisis realizado en el “Capítulo 4”, la arquitectura cuenta con tres componentes que en conjunto proveen la funcionalidad deseada:

**Servidores de Índices.** Son los encargados de indexar la meta-información de los objetos producidos para uno o más grupos de interés y responder consultas. Dicho servicio será provisto por aquellos SA de ODN que indexen todos o algunos de los grupos consumidos localmente.

**Interfaces de Usuario.** Proveen a los usuarios el front-end necesario para acceder y utilizar el servicio de búsqueda. Cada SA de ODN, mantenga o no servicio de índices, será el encargado de brindar a su comunidad de usuarios el punto de acceso al servicio de búsqueda, implementando el mecanismo de resolución de consultas.

**Red de Agentes de Búsqueda.** Es la encargada de proveer todo el contexto global a los SA de ODN, proveyendo servicios de búsqueda distribuida. Está compuesta por Agentes de Búsqueda (AB) y Responsables de Grupo (RG). Los AB tienen las funciones de proveer a los SA todo el entorno global, implementar el query routing adaptativo para soportar la búsqueda distribuida y recolectar la meta-información sobre los servidores de índices. Los RG tienen la función de administrar y mantener datos globales sobre los grupos existentes: meta-información de los servidores de índices disponibles en el sistema y Meta-información de Grupo (MiG) para cada grupo de interés.

Los SA de ODN serán los encargados de proveer el punto de acceso al servicio de búsqueda brindando a su comunidad de usuarios servicios de: *sugerencia de grupos de interés y búsqueda dentro de los grupos de interés*. Implementarán el mecanismo de resolución de consultas de manera tal de utilizar al máximo la información disponible localmente. En aquellos casos en los que el SA no disponga de información local para la resolución de una consulta, requerirá los servicios de resolución distribuida a la Red de Agentes de Búsqueda.



La Red de Agentes de Búsqueda está conformada por Agentes de Búsqueda (AB) y Responsables de Grupo (RG). Los AB serán los responsables de conocer la globalidad del servicio de búsqueda para proveer todo el entorno global al conjunto de SA que administren, evitando que cada SA deba mantener información global. Los RG serán los encargados de mantener el conocimiento de los servidores de índices disponibles para cada grupo y la generación de la MiG, actividades que requieren de una coordinación global.

Cada vez que un SA tenga que resolver una consulta para un conjunto de grupos no disponibles localmente, contactará a su AB para resolverla. El Gráfico 5.1 muestra un ejemplo de resolución cooperativa de consultas sobre grupos de interés utilizando la arquitectura descrita. Para cada SA la resolución de las consultas tendrá dos componentes: *una local y una externa*. La resolución de una consulta tendrá una componente local cuando contemple grupos consumidos e indexados por el SA, y una componente externa cuando contemple grupos no disponibles localmente. Para la resolución local, el SA utilizará el índice local, mientras que para la resolución externa solicitará la colaboración de su AB para resolverla (1). El AB seleccionará, utilizando políticas propias, el conjunto de SI a utilizar para resolverla, les enviará la consulta (2), procesará los resultados y devolverá al SA una respuesta (3).

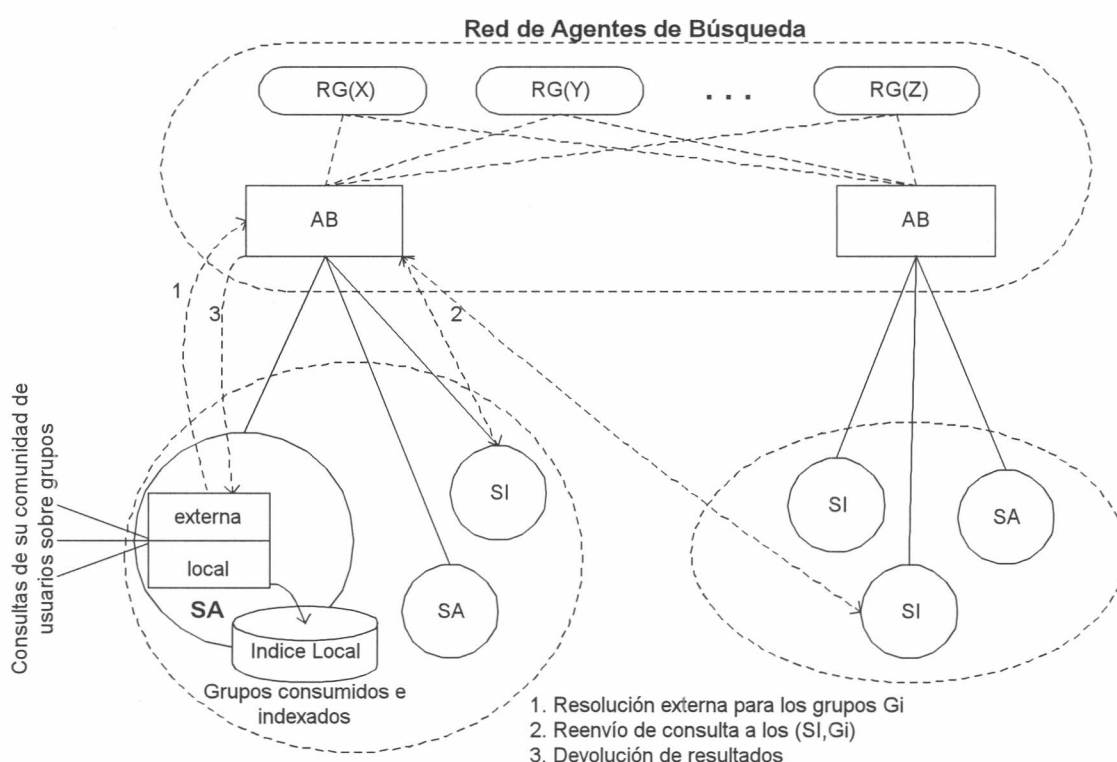


Gráfico 5.1 Resolución cooperativa de consultas sobre grupos de interés

## 5.2 Red de Agentes de Búsqueda

La *Red de Agentes de Búsqueda* es la encargada de recolectar y administrar la información necesaria para proveer servicios de búsqueda global a los SA de ODN. Está conformada por *Agentes de Búsqueda (AB)* y *Responsables de Grupo (RG)*.

Cada uno de los AB representa a un conjunto de SA. Cada SA es cliente de un AB a través del cual hará uso del servicio de búsqueda. Los AB tienen la función de proveer a cada SA bajo su administración capacidades de búsqueda global sobre grupos de interés. Proveen a los SA todo el entorno global, y serán utilizados por éstos como la puerta de acceso al mundo exterior. Así, cuando un SA tenga que resolver una consulta externa para la cual necesite cooperar con otros SIs, requerirá la colaboración de su AB.

De esta manera, los AB permitirán a los distintos SA federarse para acceder a servicios de búsqueda global. Cada AB permitirá la federación del conjunto de SA que administre. Conceptualmente, la idea es similar a la utilizada en AlephWeb [Rodríguez 96], donde distintos servicios de búsqueda se federan a través de un gestor de federación encargado de proveer todos los servicios.

Los AB deben realizar todas las tareas necesarias para proveer los servicios a los SA que administren. Deben ser capaces de: elegir los mejores servidores de índices para resolver una consulta; mantener y actualizar la información acerca de los posibles servidores de índices a utilizar; ocultar la complejidad del entorno global al conjunto de SA que administran con el objetivo de que cada SA vea el resto de las entidades como si fueran un todo.

La Red de Agentes de Búsqueda es la componente más importante del servicio de búsqueda, puesto que los AB deben mantener la dinámica del servicio y proveer todo el contexto global a los SA. Cada AB tiene las siguientes funciones específicas:

- **Elegir qué SI van a cooperar para resolver las consultas externas de los SA.** Las consultas provenientes de los SA siempre serán para un conjunto de grupos de interés. El AB debe seleccionar los *mejores* SI, reenviarles la consulta, combinar los resultados y devolver una respuesta. En aquellos casos en los que un SA desee realizar una consulta sobre todos los grupos, será función del AB filtrar el conjunto de grupos antes de procesar la consulta con el objetivo de asegurar escalabilidad al mecanismo de resolución, evitando contactar a muchos SI.
- **Tratar con los cambios de entorno que puedan producirse.** Debe actualizar y mantener información local acerca del entorno global de manera tal de reflejar nuevas condiciones que pudieran surgir. Por ejemplo, incorporación y/o eliminación de servidores de índices, etc.

- **Sugerir los mejores grupos de interés para una consulta.** Debe adquirir y mantener de alguna manera conveniente la MiG para los grupos. Utilizará la MiG para seleccionar los grupos que mejor satisfagan las consultas recibidas de sus SA.

Se utilizarán mecanismos de diseminación con el objetivo de que los distintos AB adquieran la información necesaria para proveer los servicios descriptos. Los Responsables de Grupo (RG) serán los encargados de mantener y administrar datos globales sobre los grupos existentes y serán utilizados para la diseminación de la información a los AB (ver “5.2.2 Mecanismos de diseminación de información”).

### 5.2.1 Servicios a proveer a los SA

Los AB serán consultados por los SA para acceder a servicios de búsqueda global. De esta manera cada AB provee a su comunidad de SA los siguientes servicios:

- 1) **Filtrado de grupos de interés.** Devolver el conjunto de grupos de interés que mejor satisfagan los criterios especificados en la consulta efectuada por el SA.
- 2) **Resolución cooperativa de consultas.** Resolver, para los SA, consultas que involucren un conjunto de grupos de interés, eventualmente determinando cuales son éstos.

Para satisfacer consultas del tipo 1), cada AB utilizará las MiGs de los grupos. A partir de la consulta, el AB seleccionará las MiGs que mejor puedan satisfacerla, identificando así los grupos a sugerir en la respuesta al SA. Con este objetivo indexará la MiGs de todos los grupos de interés creando un índice local, que denominaremos *Índice de MiGs*. (ver “5.2.3.2 Generación de Meta-Información de Grupo (MiG)”). Para encontrar los mejores grupos para una consulta, bastará con seleccionar del índice de MiGs aquellas que satisfagan los criterios de búsqueda como muestra el Gráfico 5.2.

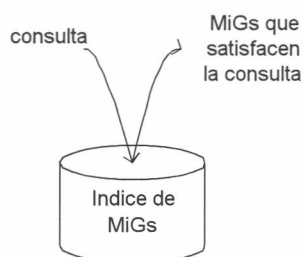


Gráfico 5.2 Índice de MiGs

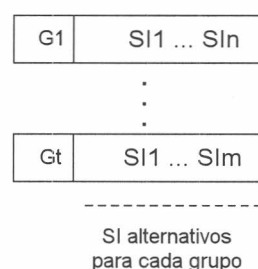


Gráfico 5.3 Tabla de SI de Agente

Para satisfacer consultas del tipo 2), cada AB utilizará una tabla que denominaremos *Tabla de SI de Agente* en la cual mantendrá la información acerca de los servidores de índices alternativos para los grupos como se observa en el Gráfico 5.3. La Tabla de SI de Agente constituye para el AB su visión local del entorno global. Será utilizada para la resolución de las consultas sobre

grupos de interés provenientes de sus SA. Por este motivo, cualquier cambio de entorno que pudiera producirse deberá quedar reflejado en esta tabla para que las decisiones del AB se basen en datos coherentes. Utilizará un conjunto de políticas locales que definirán como usar la información mantenida en dicha tabla para la resolución de consultas.

### 5.2.2 Mecanismos de diseminación de información

Para el servicio de búsqueda cada AB tiene la función de administrar a un conjunto de SA. Los SA de ODN se agruparán en comunidades coordinadas por un AB.

Cada AB debe reunir la información necesaria para la provisión de los diferentes servicios a sus SA. Con el objetivo de que cada AB pueda conocer la globalidad del servicio, necesita intercambiar información con el resto de los AB. El intercambio de información debe garantizar a cada AB:

- Adquirir la MiG de los grupos, necesaria para proveer servicios de filtrado de grupos de interés;
- Adquirir para cada grupo, los SI más convenientes, de entre los disponibles en el sistema, para proveer servicios de resolución cooperativa de consultas.

El tipo y estructura de la información a intercambiar entre los agentes debe ser tal que les permita actualizar y mantener el *Indice de MiGs* y la *Tabla de SI de Agentes*.

El dinamismo de las condiciones de entorno manejadas por cada AB: altas, bajas y/o modificaciones de los SI disponibles así como modificaciones en la MiG de los grupos, es importante que se vea reflejado en las estructuras de datos manejadas por cada AB. Es por ello que los mecanismos de diseminación deben garantizar la dinámica del servicio de búsqueda, de manera tal que los AB basen sus decisiones en datos coherentes y actualizados.

Antes de definir los mecanismos de diseminación necesarios, analizaremos cuales son las características particulares de la información a intercambiar con el objetivo de determinar las variables a tener en cuenta.

La *MiG* es utilizada por los AB para proveer servicios de filtrado de grupos de interés, siendo fundamental para garantizar la escalabilidad del proceso de resolución de consultas. Con este objetivo, la MiG debe reflejar fielmente los contenidos que circulan en cada grupo y estar disponible para todos los AB del sistema. Como resultado del análisis hecho en “4.3.1 Sugerencia de grupos de interés” la MiG se generará a partir de datos provistos por los SA productores de ODN.

La *Tabla de SI de Agente*, en cambio, es utilizada por los AB para la resolución cooperativa de consultas provenientes de sus SA. Esta tabla mantiene para cada grupo, un conjunto de SI disponibles en el sistema para ser utilizados en la resolución de consultas sobre el grupo. Los

cambios que puedan producirse en las condiciones de entorno globales, como la creación y/o eliminación de SI en cualquier punto del sistema deben quedar reflejados en las Tablas de SI de Agente de todos los AB. Sin embargo, la frecuencia de propagación de los cambios no es tan crítica como para la MiG. Esto se debe a que pueden existir SI alternativos para cada grupo que garantizarían la provisión del servicio gracias al uso de estrategias de query routing adaptativo en cada AB. Además, como cada AB administra un conjunto de SA, puede darse el caso de que para algunos grupos existan SI locales al AB, cuyo conocimiento depende exclusivamente de la relación SA-AB y no de la relación AB-AB.

Al ser los AB los encargados de manejar la globalidad del servicio, además de adquirir la información necesaria para la provisión de los servicios, deben encargarse de mantener y actualizar la información acerca de los SA que administren de manera tal de poder resumirla y propagarla al resto de los AB.

Tal como viéramos en “4.4.2.2.1 Agentes de Búsqueda”, existen dos funciones que requieren de una coordinación global a nivel de grupo: la generación de la MiG y el mantenimiento de todos los SI disponibles en el sistema para cada grupo. Estas actividades serán llevadas a cabo por los Responsables de Grupo (RG).

La necesidad de tener una coordinación global para mantener los SI disponibles para cada grupo se debe al problema que se presenta cuando para algún grupo no existe en el sistema ningún SI disponible. Esta situación hará que el sistema de búsqueda no pueda resolver a nivel global ninguna consulta que involucre al grupo en cuestión. Para evitarlo, es necesario garantizar en el sistema por lo menos un SI para cada grupo. Si no existieran responsables de grupo, ¿cómo podrían enterarse los AB que no existen SI para algún grupo si todos manejan información parcial? Y entonces ¿cuál de ellos debería tomar la decisión de elegir algún SA bajo su administración para que indexe el grupo faltante ... y si lo hicieran todos?. En respuesta a estos interrogantes, la solución más adecuada es manejar esta decisión a nivel responsable de grupo. De esta manera centralizamos en un único nodo la toma de decisión acerca de la elección de algún SA para que se convierta en SI para el grupo faltante.

Dada la necesidad de tener roles de coordinación de grupos, utilizaremos a los Responsables de Grupo (RG) para diseminar la información a los AB del sistema. Contactando a los responsables de grupo cada AB podrá obtener la MiG así como los SI disponibles en el sistema para cada grupo de interés.

Tal como se muestra en el Gráfico 5.4, utilizaremos una jerarquía de dos niveles entre los ABs y los RGs para la diseminación de información. Con esta organización, cada AB resumirá la información referente a sus SI para reportarla a los responsables correspondientes. De la misma forma, para que cada AB pueda actualizar sus tablas locales acerca de datos globales de otros AB, consultará las novedades a los responsables a través de requerimientos de servicios.

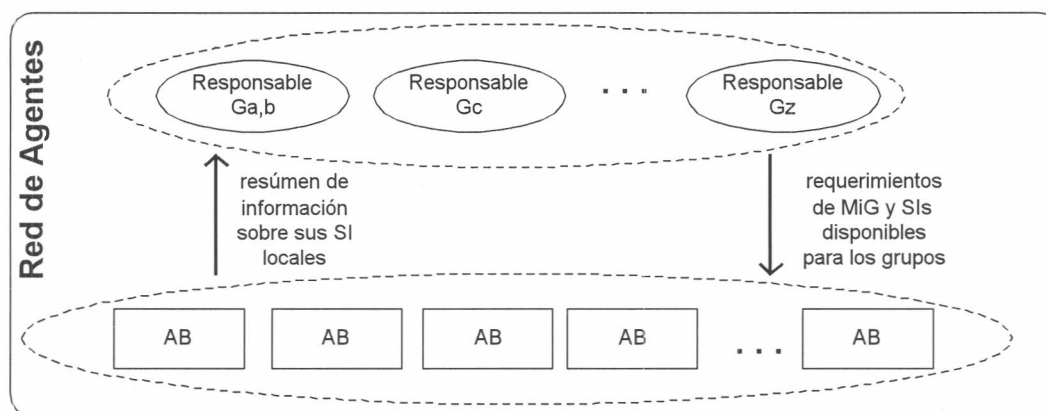


Gráfico 5.4 Mecanismos de diseminación entre responsables y agentes

### 5.2.3 Responsables de Grupos (RG)

Los Responsables de Grupo (RG) mantendrán información global a nivel grupo de interés. Esta información será provista por los AB y podrá ser requerida por éstos a través de requerimientos de servicios. Cada responsable de grupo mantendrá la siguiente información para el/los grupo/s que administre:

- SI disponibles en el sistema para el grupo. Mantendrá meta-información (que definiremos más adelante) sobre los SI disponibles en el sistema. Cuando no haya ningún SI para el grupo deberá llevar a cabo las acciones necesarias para garantizar la existencia de por lo menos uno.
- Generación y mantenimiento de la MiG del grupo.

#### 5.2.3.1 Garantizar existencia de SI para cada grupo

Con el objetivo de que un RG pueda determinar si en el sistema existen SI disponibles para el grupo que administra, debe saber de alguna manera qué SI hay disponibles en el sistema.

Cada AB registrará sus SI locales ante los responsables de grupo correspondientes. Deberá hacerlo en todos los responsables de grupo para los cuales tenga por lo menos un SI para el grupo.

Con esta información, los responsables de grupo estarán en condiciones de saber si existen o no SIs para cada grupo. En el caso de no existir ningún SI disponible, el responsable garantizará la existencia de por lo menos uno, seleccionando a algún AB para que éste realice las acciones necesarias y genere el SI faltante. La selección del AB podría ser en forma random o utilizando alguna estrategia más inteligente que tenga en cuenta por ejemplo, sobrecarga de trabajo de los AB, etc.

Cuando el AB seleccionado tenga disponible el SI solicitado para el grupo faltante, éste lo registrará ante el coordinador correspondiente, situación que se propagará a los AB del sistema gracias a los mecanismos de diseminación de información utilizados.

### 5.2.3.2 Generación de Meta-Información de Grupo (MiG)

La generación de la MiG es un proceso que involucra a los responsables de grupo y a los AB que administren SA productores para algún grupo de interés.

Cada AB del sistema que tenga algún SA productor deberá generar y mantener la *Meta-información de grupo parcial* (MiGp). Generarán una MiGp para cada grupo del cual tengan algún SA productor bajo su administración. Una vez generadas las MiGps deberán enviarlas a los responsables de grupo correspondientes.

Cada responsable de grupo recibirá la MiGp de todos los AB del sistema que tengan productores para ese grupo quedando en condiciones de generar la MiG definitiva para el grupo.

De la misma manera, los AB que deban generar MiGp recibirán, de cada SA productor que administren, la *Meta-información de grupo de SA* (MiGSA) a partir de las cuales generarán la MiGp.

El proceso de generación de MiG se observa en el Gráfico 5.5.

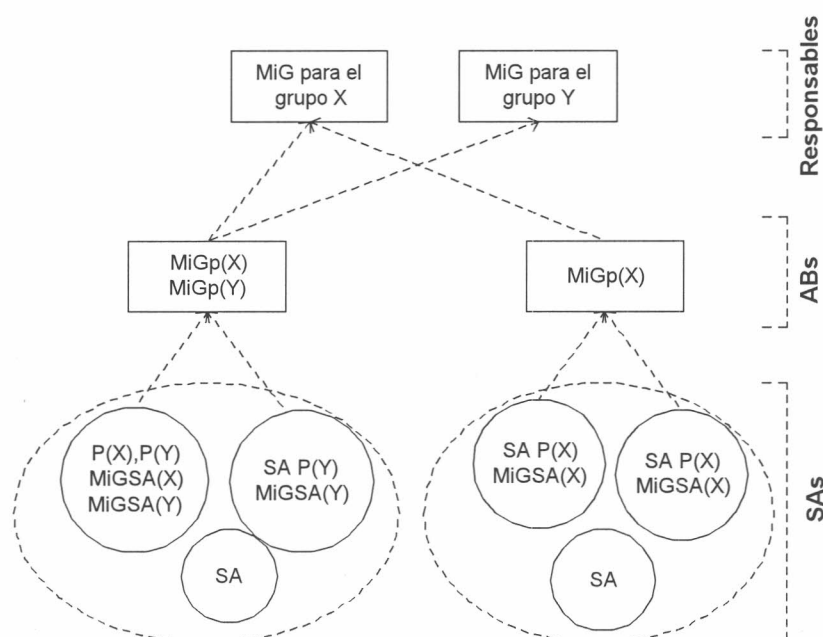


Gráfico 5.5 Proceso de Generación de MiG

El proceso de sumariación que se utilice para la generación de MiG, MiGp y MiGSA debe ser tal que la frecuencia con la que éstas se modifiquen sea mucho menor a la frecuencia con la que se modifican los contenidos de los objetos. Esto hará que la propagación de las MiG no deba darse cada vez que un objeto sufra cambios poco significativos.

Dado que cada AB almacenará la MiG en su *Índice de MiGs* para poder hacer el filtrado de grupos, el proceso de sumariación que realicen los SA, AB y los responsables debe generar como



resultado un *surrogate document* para el grupo. Algunos de los estándares que se pueden utilizar son los *centroides* de Whois++ [RFC 1913], *SOIF*<sup>1</sup> de Harvest [Hardy 96] (ver “Apéndice C”), *content label* de Content Routing [Sheldon 95a] entre otros.

A los efectos de poder determinar el grado de actualización de un SI, la MiG de cada grupo deberá contener un atributo extra que indique la cantidad total de objetos producidos para el grupo que represente. Para generar dicho valor bastará con que los SA productores especifiquen en la MiGSA la cantidad de objetos producidos localmente, los AB calculen en la MiGp la cantidad como la suma de las cantidades de todas las MiGSA recibidas para el grupo, y por último los responsables calculen en la MiG la cantidad como la suma de las cantidades de todas las MiGp recibidas para el grupo.

### 5.2.3.3 Creación de Responsables de Grupo

El rol de responsable de grupos podrá ser cumplido por uno o más nodos. Independientemente de la cantidad de nodos responsables, entre todos los existentes deberán cubrir la totalidad de grupos de interés existentes en ODS.

El caso más sencillo es tener un único nodo responsable para todos los grupos de interés, mientras que la situación extrema sería tener un nodo responsable para cada grupo de interés existente en el sistema.

La creación/eliminación de un nuevo nodo responsable o bien el agregado o quitado de responsabilidad a un nodo ya existente son actividades que dependen exclusivamente de la creación o eliminación de grupos de interés en ODS.

Consideramos que las actividades de administración deben ser realizadas por las autoridades de clasificación de ODS. Actualmente, estas autoridades son las encargadas de la creación o eliminación de los grupos de interés en función a los esquemas de clasificación que se vayan generando.

Existen un conjunto de actividades que se desencadenan con la creación de un nuevo grupo de interés: notificación a todos los SA de ODN; definición del esquema de clasificación a utilizar para catalogar los objetos, etc. A este conjunto deberá agregarse una nueva actividad: garantizar la existencia de un nodo responsable de la administración del nuevo grupo para el servicio de búsqueda. El rol de coordinación para el nuevo grupo podrá ser llevado a cabo por algún nodo responsable ya existente o bien requerir la creación de uno nuevo.

Los responsables de grupo son nodos críticos para el servicio de búsqueda al ser los encargados de proveer a los AB la MiG y los SI disponibles a utilizar para la resolución de consultas.

---

<sup>1</sup> Summary Object Interchange Format.

Para la robustez del sistema es aconsejable que puedan existir varios nodos responsables para un mismo grupo, manejándose un esquema similar al uso de primarios y secundarios en el DNS [Albitz 98], según muestra el Gráfico 5.6.

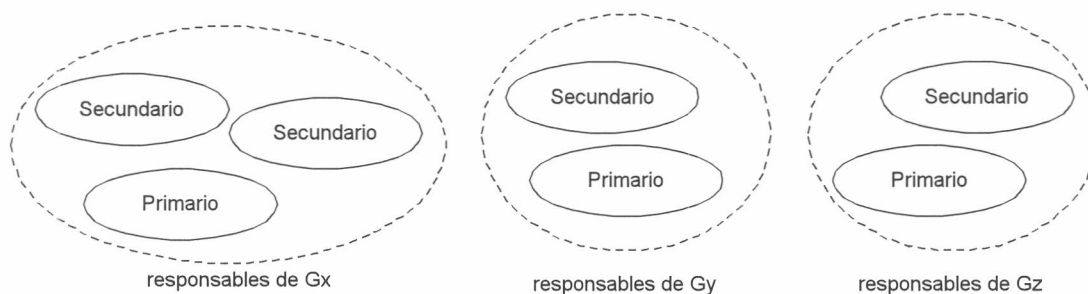


Gráfico 5.6 Replicación de responsables

Para aquellos grupos para los que exista más de un responsable, los AB del sistema deberán registrar los datos solo en los RG *primarios*, pero podrán obtener la MiG y los SI disponibles tanto de los RG primarios como desde los secundarios.

Utilizando un esquema como el descripto, cada vez que en ODS se dé de alta un nuevo grupo de interés deberá registrarse un *RG primario* para el grupo, pudiendo ser un RG ya existente para otro grupo o bien crearse uno nuevo. En cualquier momento podrán darse de alta uno o más *RGs secundarios* que replicarán la información de MiG y SI disponibles desde el RG primario.

#### 5.2.3.4 Cómo contactar a un responsable de grupo

Dado que pueden existir varios nodos responsables de grupo, los AB deben utilizar algún mecanismo que les permita saber cuales son los responsables de cada grupo y como contactarlos.

Utilizaremos un mecanismo funcionalmente similar a la delegación de dominios en el DNS. En el DNS, cada nombre de dominio (por ejemplo: uba.ar) tiene asociado un conjunto de servidores de nombres (*name servers*) encargados de responder las consultas específicas para ese dominio. Estos servidores de nombres pueden ser contactados utilizando una dirección ip y generalmente escuchan en un port bien conocido (53). De la misma manera, las autoridades de clasificación deberán mantener en un port bien conocido un servidor en el cual se defina, para cada grupo, el conjunto de responsables disponibles conjuntamente con su rol de primario o secundario.

Los AB contactarán a este servidor para obtener el listado de responsables disponibles para cada grupo.

A modo de ejemplo, supongamos que existieran los grupos X, Y, Z y que RG1 fuera responsable de X e Y, que RG2 fuera responsable de Y y Z y que RG3 fuera responsable de X; entonces los registros a generarse en el servidor serían:

X	IN	R	P	RG1
	IN	R	S	RG3
Y	IN	R	P	RG1
	IN	R	S	RG2
Z	IN	R	P	RG2

*R indica responsable de grupo, P y S indican si es primario o secundario respectivamente.*

Al momento que un AB necesite contactar al responsable del grupo Y, consultará al servidor (al igual que una aplicación cualquiera consulta un servidor de nombres) y obtendrá como respuesta RG1 primario y RG2 secundario, que son los responsables del grupo Y.

#### 5.2.4 Tabla de SI de Agente

La *Tabla de SI de Agente* es utilizada por los AB para satisfacer los pedidos de resolución de consultas sobre grupos provenientes de los SA que administran. La información mantenida en esta tabla y la forma de utilizarla definen como los SA bajo la administración de un AB utilizarán el servicio de búsqueda.

Cada AB administra su propia *Tabla de SI de Agente* en la cual mantiene, para cada grupo de interés, un conjunto de SI alternativos (de entre los disponibles en el sistema) que puede utilizar para la resolución cooperativa de las consultas. La composición de la tabla se observa en el Gráfico 5.3.

##### 5.2.4.1 Vistas de SI por grupo de interés

Para cada grupo de interés existirá un conjunto de SI que definen para el AB su vista local del entorno global. De esta manera, las decisiones de query routing adaptativo que realicen distintos AB podrán basarse en diferentes conjuntos de SI para el mismo grupo de interés según muestra el Gráfico 5.7.

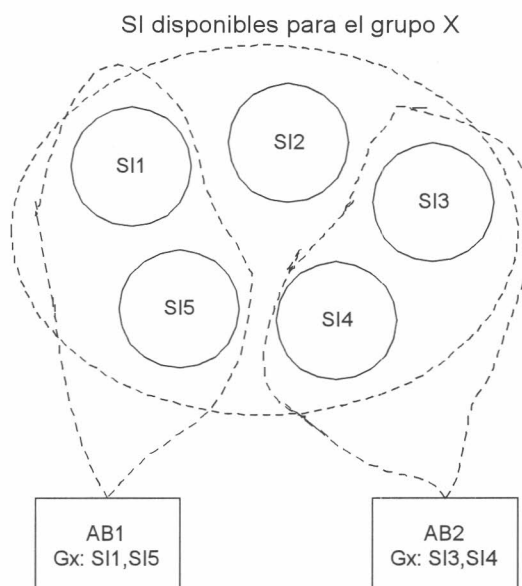


Gráfico 5.7 Diferentes vistas de SI para el mismo grupo de interés

Como ya definimos, cada responsable de grupo es el encargado de mantener, para los grupos que administre, el conjunto de todos los SI disponibles en el sistema. Los AB solicitarán a los distintos responsables la información de SI para actualizar su Tabla de SI de Agente.

Utilizar diferentes conjuntos de SI disponibles para un mismo grupo de interés nos permitirá implementar el concepto de *connectivity regions* definido en [Lagoze 98]. Con este objetivo cada AB será el encargado de administrar una región y utilizará para cada grupo, el conjunto de SI que se consideren con mejor conectividad con respecto al AB. Denominaremos a dicho conjunto la *vista de SI* para un AB sobre un grupo.

Los responsables de grupo serán los encargados de definir las vistas de SI para proveer a cada AB del sistema. La configuración de las distintas regiones (que servidores de índices son miembros de una región para cada grupo) puede ser estática o dinámica.

En el prototipo de NCSTRL [NCSTRL] se realizaron pruebas utilizando una configuración estática de las regiones, presentándose algunos problemas. Las réplicas de servidores de índices de la región de Hungría nunca lograban actualizarse por problemas de conectividad. Este inconveniente hacía que el Collection Regional Server de Hungría utilizara servidores de índices poco actualizados para resolver las consultas de su región con la consiguiente mala prestación del servicio a los UI de su misma región.

Este resultado nos permite concluir que los SI con mejor conectividad con un AB no necesariamente pueden ser los mejores para resolver las consultas, ya que podrían no tener buena frecuencia de actualización. En función de la gran variación en los patrones de conectividad, sería aconsejable que los SI que se asignen a las regiones puedan variar en función a los cambios que puedan suscitarse en la red subyacente, es decir que la configuración de las regiones sea dinámica.

Con respecto al prototipo de bibliotecas digitales, ODS presenta una diferencia sustancial y es que utiliza protocolos de ruteo adaptativos para la replicación de objetos entre los SA de ODN. Estos

protocolos de ruteo tienen en cuenta el estado de la red subyacente, pudiendo modificar la composición de las cadenas de distribución en forma dinámica de manera tal que todas las réplicas converjan a un estado consistente aunque existan particiones de red.

Esta importante diferencia minimiza la posibilidad de que se presenten problemas como los encontrados en NCSTRL, donde algunas réplicas puedan quedar desactualizadas durante mucho tiempo. Podríamos de esta manera utilizar una configuración estática de regiones.

Sin embargo, para que los responsables de grupo puedan configurar estáticamente las regiones, los SI disponibles para los grupo deberían ser siempre los mismos, cosa que no ocurre en nuestro caso. La disponibilidad de SI varía en función a la suscripción de los usuarios y en la decisión de los SA de indexar o no la meta-información de los objetos. Esto hace que la disponibilidad de SI para los grupos varíe con el tiempo, haciendo que las regiones deban reconfigurarse. Por este motivo los responsables de grupo configurarán las regiones dinámicamente, para lo cual recibirán de los AB información específica sobre los SI, que definiremos más adelante.

Una vez definido el concepto de vistas de SI disponibles para cada grupo que tendrán los AB, queda por definir como éstos los utilizarán para la resolución cooperativa de las consultas.

#### 5.2.4.2 Políticas de uso

Cada AB utilizará la Tabla de SI de Agente cada vez que tenga que resolver una consulta sobre determinados grupos. Para cada grupo involucrado en la consulta, el AB elegirá el *mejor* SI de entre los disponibles en la tabla para ese grupo. Una vez elegidos los SI de cada grupo, les enviará la consulta, recibirá los resultados y confeccionará una respuesta para el SA. El método de selección del mejor SI para cada grupo dependerá de la estrategia de query routing adaptativo que implemente el AB.

El query routing que realicen los AB se basará tanto en contenidos (que SI indexan los grupos definidos en la consulta) como en prioridades (como se elegirá el mejor SI de la vista disponible).

Los AB utilizarán las vistas provistas por los representantes de cada grupo para saber que SI indexan los grupos consultados.

En lo que respecta a como los AB *eligen* el SI (del total de réplicas disponibles) al cual encaminar una consulta, utilizarán políticas propias que denominaremos *políticas de uso*. Una forma posible para la selección del mejor SI para un grupo, es ordenar la vista de SI en función de distintos criterios según la conveniencia y los objetivos del AB. Las *políticas de uso* definen esos criterios.

Las políticas de uso deben aplicarse siempre que se modifique la vista de SI para algún grupo y luego de procesar una consulta, de manera tal que la selección de los SI sea dinámica y adaptativa (por eso hablamos de query routing adaptativo).

El significado de *mejor SI* puede variar para los distintos AB y contemplar una combinación de alternativas. A modo de ejemplo, un AB podría utilizar como política de uso una que considere el mejor SI al más actualizado o bien al que tenga mejores tiempos de respuesta, o una combinación de ambas. Otro AB podría elegir el mejor SI en forma random.

Para que los AB puedan implementar distintas estrategias de query routing adaptativo, la información a intercambiar sobre los SI deberá contemplar la provisión de atributos como: refresh del SI, costo de utilizar ese SI, etc., que definiremos en “5.2.4.3 Meta-información de SI”.

Entre las políticas de uso posibles de utilizar podemos mencionar:

- *Histórico de fallas.* Permitiría alterar las decisiones de ruteo en respuesta a fallas de los SI. Alcanzaría con que el AB registre un histórico del éxito o falla para cada SI al cual encamine una consulta. Esto evitaría al AB la selección de un SI para el cual tenga registradas repetidas fallas por un período de tiempo.
- *Utilizar SI locales.* Ante determinadas situaciones, un AB podría preferir utilizar en primera instancia siempre SI locales y solo en caso de fallas intentar encaminar las consultas a algún SI perteneciente a la vista disponible.
- *Utilizar los SI más actualizados.* Recordar que los SI disponibles en el sistema no son más que SA consumidores que mantienen un índice local. Como ODS utiliza un mecanismo de replicación de consistencia débil [Obraczka 94], puede darse el caso que para el mismo grupo, algunas réplicas estén más actualizadas que otras.

#### 5.2.4.3 Meta-información de SI

La *Meta-información de SI* define el conjunto de atributos que caracterizan a un SI. Estos atributos serán utilizados por los responsables de grupo para la asignación de SI a regiones y por los AB para la aplicación de sus políticas de uso.

Algunos atributos definirán características globales al SI mientras que otros son específicos a un grupo de interés dado que un SI puede indexar. Entre los atributos posibles para caracterizar a un SI podemos mencionar:

##### Globales:

- Dirección y port al cual se deben encaminar las consultas.
- Estado de conectividad con el SI. Esta información será utilizada por los responsables para la confección de las vistas de los distintos AB, es decir de las distintas regiones.

- Capacidades de consulta. La idea es proveer información acerca de las capacidades de consultas provistas por un SI, como ser campos soportados en las consultas (por ejemplo: título, autor) y de los operadores soportados (por ejemplo: '>', '>=').

#### **Específicos por grupo de interés:**

- Cantidad de objetos indexados. Este atributo permitirá determinar cuan actualizado se encuentra el SI. Recordar que en la MiG de cada grupo se especifica el total de objetos producidos, de manera tal que comparando ambos valores un AB estará en condiciones de determinar el grado de actualización de un SI.
- Costos de utilización del SI. Podría darse el caso que algunos SI restrinjan la resolución de consultas externas.
- Carga de trabajo del SI.

El grupo de trabajo del proyecto de Bibliotecas Digitales de la Universidad de Stanford [Stanford-DL] ha especificado una serie de estándares para la representación y transmisión de información como la definida anteriormente. Estos estándares fueron definidos en el contexto del proyecto STARTS (Stanford Protocol Proposal for Internet Retrieval and Search) [Gravano 97] y en el "Apéndice C" pueden encontrarse ejemplos de codificación de meta-información de SI.

#### **5.2.4.4 Mantenimiento**

Puesto a que la cantidad de grupos de interés puede ser grande, es probable que algunos AB no dispongan de espacio de almacenamiento suficiente para mantener vistas de SI para todos los grupos existentes en el sistema. Además, dependiendo de los intereses de los usuarios, puede darse el caso de que las consultas se apliquen siempre para un conjunto reducido de grupos de interés.

En consecuencia, se debe prever que los AB puedan trabajar con información parcial, es decir solo para algunos grupos, utilizando la tabla como si fuera una cache. Para mantener la tabla los AB utilizarán políticas de mantenimiento de cache, de manera tal de garantizar mantener la información más frecuentemente utilizada para resolver las consultas de sus SAs.

Al manejar información parcial, puede darse el caso de que un AB no disponga localmente de la información necesaria al momento de resolver una consulta, por ejemplo la vista de SIs para un grupo determinado. En estos casos el AB podrá solicitar la vista de SI al responsable de grupo correspondiente y almacenarla en la cache para ser utilizada en consultas futuras.

Para estos casos, la política de mantenimiento de cache más útil es almacenar las vistas de SI para los grupos más frecuentemente consultados (Least Frequently Used - LFU). Si fuera necesario eliminar alguna entrada de la tabla para almacenar la vista de SI de un nuevo grupo, la política debería eliminar la vista de SI correspondiente al grupo menos consultado.



Dado que pueden producirse cambios en las vistas de los grupos, los AB deberán utilizar algún mecanismo que les permita mantener la consistencia de la información de la tabla. Una posibilidad es que a intervalos regulares obtengan la vista de SI desde los responsables de grupo. Esta actividad podrían realizarse en momentos de menor actividad con el objetivo de no perjudicar los tiempos de respuestas que puedan obtener los SA bajo su administración.

## 5.3 SA's

Cada SA, además de sus tareas habituales como entidad de ODN, ahora tendrá la función adicional de proveer a la comunidad de usuarios que atienda el punto de acceso al servicio de búsqueda. Los usuarios contactarán a los SA locales para formular sus consultas, ya sea de sugerencia de grupos o bien focalizadas en ciertos grupos de interés.

Todo SA proveerá estos servicios independientemente de que genere o no un índice local para los grupos que consuma. Aquellos SA que indexen los objetos de uno o más grupos serán considerados por el servicio de búsqueda como SI disponibles para la resolución cooperativa de consultas. Para ser identificado como tal (por los AB) registrará en su AB la meta-información de SI especificando para que grupos provee servicio de índices.

Aquellos SA que además sean productores de objetos para algún grupo tendrán la tarea adicional de confeccionar la MiGSA y suministrarla a su AB.

### 5.3.1 Resolución de consultas para los usuarios

Todo SA debe proveer una interfaz de consulta e implementar el mecanismo de resolución. La comunidad de usuarios de cualquier SA podrá formular los siguientes requerimientos al servicio de búsqueda:

1. Requerir el listado de grupos más relevantes para su consulta.
2. Focalizar sus consultas sobre ciertos grupos de interés.

Optativamente, los SA podrán proveer servicios de valor agregado específicos para su comunidad de usuarios, por ejemplo búsqueda en los grupos de interés consumidos localmente en el SA.

En respuesta a las consultas del tipo 1, el SA siempre la derivará a su AB el cual le devolverá el listado de grupos que satisfagan la consulta del usuario.

Para responder consultas del tipo 2, los SA deben implementar el mecanismo de resolución de consultas. El mecanismo de resolución tendrá dos componentes: una local y otra externa. La resolución de la consulta tendrá una componente local cuando contemple grupos consumidos e indexados por el SA, y una componente externa cuando contemple grupos no disponibles

localmente. Para la resolución local se utilizará el índice mantenido por el SA, mientras que para la externa se solicitará la colaboración del AB para resolverla.

Veamos el mecanismo de resolución con un ejemplo. Supongamos que  $N$  sea el conjunto de grupos sobre el cual el usuario desea realizar la consulta. Sea  $I$  el índice local mantenido por el SA construido para los grupos  $G(Csa)$ , donde  $Csa$  son los grupos consumidos por el SA y  $G(Csa)$  es el subconjunto de  $Csa$  que se indexa. En el caso de que el SA no indexe información entonces  $I = \text{vacío}$ .

- 1) Si el conjunto de grupos  $N - G(Csa)$  es vacío, esto significa que la resolución es local y puede ser satisfecha localmente utilizando  $I$ .
- 2) Si el conjunto de grupos  $N - G(Csa)$  no es vacío, significa que la consulta involucra a un conjunto de grupos no disponibles en el índice local (ya sea porque no se indexan o no son consumidos por el SA). En este caso la resolución siempre tendrá una componente externa pudiendo o no tener una local.
  - Si  $N \cap G(Csa) = \emptyset$  la resolución es externa para todos los grupos.
  - Si  $N \cap G(Csa) \neq \emptyset$ , entonces para los grupos  $N \cap G(Csa)$  la resolución será local y para el resto la resolución será externa.

En el caso de ser necesaria una resolución externa, el SA enviará a su AB la consulta y el conjunto de grupos sobre el que haya que aplicarla para obtener las respuestas y mostrarlas al usuario.

### 5.3.2 Resolución de consultas para otros SA

Los únicos SA que podrán recibir consultas de los AB del sistema serán aquellos que sean servidores de índices registrados ante algún AB.

Cualquier solicitud de resolución de consultas que provenga de un AB será restringida a un conjunto específico de grupos indexados localmente, por lo que siempre serán resueltas en forma local. Los SI responderán a estas solicitudes devolviendo el conjunto de objetos del grupo que satisfagan la consulta.

Dado que los SI pueden ser consultados por distintos AB, es necesario utilizar un formato estándar para la representación y transmisión de las consultas así como de los resultados. En el proyecto STARTS [Gravano 97] se define un formato estándar sencillo y útil para la codificación de las consultas y resultados. El formato que definen consiste en que tanto las consultas como los resultados se codifiquen utilizando los pares atributo/valor de un objeto SOIF [Hardy 96] (ver "Apéndice C") y se transmitan utilizando el protocolo HTTP. De esta manera los objetos SOIF correspondientes a las consultas se enviarían en una página HTML mediante el método POST utilizando un tag "soif" cuyo valor es el objeto SOIF que representa la consulta. Los resultados de

las consultas también serían transmitidos con el protocolo HTTP. En el “Apéndice C” se describen algunos ejemplos de codificación de consultas y resultados siguiendo el formato definido en STARTS.

### 5.3.3 SA productores

Con el objetivo de producir la MiG, todo SA productor de objetos para algún grupo deberá confeccionar la MiGSA con el objetivo de resumir lo producido localmente para el grupo. Enviarán las MiGSAs a su AB cada vez que sea modificada.

## 5.4 Consideraciones finales

Creemos importante remarcar la diferencia entre los mecanismos de diseminación y los de resolución de consultas explicados. La organización jerárquica entre responsables y ABs es utilizada exclusivamente para diseminar información entre las entidades que componen la Red de Agentes. Para la resolución de consultas **no** se utiliza la jerarquía, por el contrario cada AB resolverá las consultas de sus SA contactando directamente a los SI necesarios, en función de la información que tenga disponible en sus tablas locales.

De esta manera, **no** utilizamos la misma arquitectura de diseminación para la resolución de consultas, como sí lo hacen los prototipos de Content Routing [Sheldon 95a] y Whois++ [RFC 1913]. Estos organizan los nodos en una malla jerárquica a través de la cual se disemina la información pero también se resuelven las consultas. En general las mallas jerárquicas se utilizan para organizar el espacio de información para la posterior resolución de consultas.

En la arquitectura propuesta, la organización de contenidos se da a través de los responsables de grupo siendo independiente de la estrategia de resolución de consultas.

En cuanto a los protocolos de diseminación, la mayoría de las mallas de brokers no jerárquicas utilizan protocolos costosos en términos de mensajes en la red y exceso de trabajo en cada uno de los nodos para procesar la información [Schwartz 90] [Francis 95], por este motivo hemos elegido una estructura jerárquica para la diseminación.

Nuestra arquitectura combina las ventajas de las mallas jerárquicas y de las no jerárquicas. Utilizamos una organización jerárquica entre responsables y AB para la diseminación de información, mientras que la resolución distribuida de consultas se realiza en función de la información local y de las necesidades puntuales de los usuarios de cada nodo, pudiendo cada AB aplicar sus propias políticas de uso para definir su comportamiento.

## Capítulo 6

# Conclusiones y trabajos futuros

### 6.1 Conclusiones

Nuestra principal motivación fue la dificultad que tienen los servicios de búsqueda para adaptarse a entornos de redes como Internet, caracterizados por un gran volumen de información, una administración descentralizada, y una gran cantidad de servidores de información y usuarios distribuidos geográficamente.

Por este motivo hemos analizado y estudiado las dimensiones de escalabilidad a tener en cuenta en la construcción de servicios de búsqueda para entornos de gran escala: volumen de información, diversidad de recursos y elevado número de usuarios [Bowman 94].

Luego, describimos y discutimos un número de servicios de búsqueda existentes en el marco de la Web, News (ver “Apéndice B”) y Bibliotecas Digitales que hemos clasificado de acuerdo a sus arquitecturas en: *centralizadas* y *distribuidas*. Estudiamos los problemas de escalabilidad de las arquitecturas centralizadas y analizamos varios prototipos basados en arquitecturas distribuidas (ver “Apéndice A”) determinando sus características más relevantes desde el punto de vista de las dimensiones de escalabilidad mencionadas anteriormente.

De los trabajos analizados, rescatamos principalmente algunas ideas propuestas por AlephWeb [Rodríguez 96] y por [Lagoze 98] para el proyecto NCSTRL [NCSTRL].

Finalmente, presentamos un nuevo servicio de búsqueda para que la comunidad de usuarios del Sistema de Distribución de Objetos (ODS) pueda localizar y buscar objetos de interés de entre los producidos. El servicio está basado en una arquitectura de indexación y búsqueda distribuida que contempla la existencia de múltiples servidores de índices y mecanismos de comunicación para la resolución cooperativa de consultas, logrando así un servicio escalable.

El servicio de búsqueda provee a los usuarios la siguiente funcionalidad:

- **Sugerencia de grupos de interés.** Los usuarios tendrán un acceso más directo a los grupos de interés más relevantes, evitando tener que navegar por los contenidos de distintos grupos para determinar si contienen información de su interés. A partir de una necesidad de información expresada mediante una consulta, el sistema sugerirá los grupos que mejor la

satisfagan, lográndose mejorar el uso del sistema de distribución al no malgastar recursos de red para replicar objetos que no serán útiles a los usuarios.

- **Búsqueda dentro de los grupos de interés.** Los usuarios podrán restringir sus consultas a ciertas categorías de objetos. El sistema buscará en los contenidos de los grupos deseados, devolviendo como resultado el listado de objetos que satisfagan los criterios especificados en la consulta. Las búsquedas abarcarán contenidos globales, no siendo restrictivo el hecho de que un usuario consuma o no determinado grupo de interés.

Los usuarios podrán buscar los objetos que traten la temática de su interés a través de una interfaz de consulta accesible localmente. Motivados por el hecho de que los objetos están clasificados en varios grupos de interés, el servicio provee formas alternativas para encontrar la información útil a un usuario, proveyendo búsqueda focalizada en categorías de objetos y en características semánticas.

Los principales objetivos del diseño han sido:

**Ser escalable.** Soportar, con un mínimo aumento de costo administrativo y degradación de performance, la adición de servidores de índices y de usuarios así como el gran aumento del volumen de información y cantidad de grupos de interés.

**Ser adaptativa.** Adaptarse fácilmente a las condiciones de entorno cambiantes.

**Utilizar características particulares de ODS.** Aprovechar al máximo las características ya disponibles, evitando malgastar recursos de red así como duplicar esfuerzos.

**Preservar la autonomía de los SA.** No imponer fuertes restricciones a los Agentes de Servicio (SA)<sup>1</sup> con el objetivo de facilitar su participación. En ODS no existe una única entidad administrativa para todos los SA, por el contrario, la administración está descentralizada. Los SA no manejan la misma información sino solo la que es de interés para sus usuarios.

La arquitectura cuenta con tres componentes principales que en conjunto proveen la funcionalidad deseada. Estos son:

**Servidores de Índices.** Son los encargados de indexar la meta-información de los objetos producidos para uno o más grupos de interés y responder consultas. Dicho servicio será provisto por aquellos SA de ODN<sup>2</sup> (Object Distribution Network) que indexen todos o algunos de los grupos consumidos localmente.

---

<sup>1</sup> Los SA son entidades pertenecientes al sistema ODS. Los usuarios se suscriben a volúmenes de objetos en un SA para acceder a copias locales de los objetos producidos globalmente y eventualmente para registrar allí los que deseen distribuir en la red.

<sup>2</sup> Es la red encargada de la replicación de objetos desde los SA productores a los SA consumidores.

**Interfaces de Usuario.** Proveen a los usuarios el front-end necesario para acceder y utilizar el servicio de búsqueda. Cada SA de ODN, mantenga o no servicio de índices, será el encargado de brindar a su comunidad de usuarios el punto de acceso al servicio de búsqueda, implementando el mecanismo de resolución de consultas.

**Red de Agentes de Búsqueda.** Es la encargada de proveer todo el contexto global a los SA de ODN, proveyendo servicios de búsqueda distribuida. Está compuesta por Agentes de Búsqueda (AB) y Responsables de Grupo (RG). Los AB tienen las funciones de proveer a los SA todo el entorno global, implementar el query routing adaptativo para soportar la búsqueda distribuida y recolectar la meta-información sobre los servidores de índices. Los RG tienen la función de administrar y mantener datos globales sobre los grupos existentes: meta-información de los servidores de índices disponibles en el sistema y Meta-información de Grupo (MiG) para cada grupo de interés.

A través de los Agentes de Búsqueda hemos federado los distintos servidores de índices en un único servicio de búsqueda global. De esta manera, logramos una administración descentralizada del servicio, importante para la escalabilidad. Los AB aplican políticas locales (políticas de uso) para definir cómo utilizar el servicio independientemente de como lo hace el resto. Esta característica se basa en la idea de AlephWeb [Rodríguez 96] de componer los brokers de su arquitectura mediante una *estructura de federación*.

La disponibilidad de múltiples servidores de índices para un mismo grupo de interés se debe al hecho de que distintos SA consumidores de un mismo grupo indexen la meta-información de los objetos. Tal disponibilidad de índices replicados para un mismo grupo de interés nos ha permitido distribuirlos en regiones de manera tal que las consultas se resuelvan accediendo a las réplicas más cercanas. Esta idea rescata los conceptos de *connectivity regions* y *query routing adaptativo* utilizados en NCSTRL por [Lagoze 98].

Las características relevantes de la arquitectura propuesta son:

- **Provee localmente el punto de acceso al servicio.** Los usuarios acceden a los servicios de búsqueda conectándose al SA local que les provee el punto de entrada al sistema ODS. Esto permite que los SA puedan proveer servicios de valor agregado específicos para su comunidad de usuarios: interfaces personalizadas, pantallas de ayuda, etc. Permite explotar información disponible localmente en cada SA para la resolución de consultas, evitando en muchos casos la resolución distribuida que es más costosa. Permite distribuir el procesamiento de consultas, evitando problemas de sobrecarga y cuellos de botella.
- **Manejo del entorno global a nivel Agentes de Búsqueda.** Al ocultar la complejidad del entorno global a los SA, hemos logrado que éstos mantengan su autonomía e independencia. A su vez, el comportamiento de cada AB es independiente del resto y está guiado por las políticas de uso que tenga definidas permitiendo una administración descentralizada del servicio.



- **Utiliza al máximo las múltiples réplicas disponibles para los grupos.** Las réplicas existentes para los distintos grupos de interés se distribuyen en conjuntos de manera tal que cada Agente de Búsqueda utilice los que estén mejor conectados con éste para resolver las consultas. Reutilizando los recursos ya disponibles en ODS, evitamos duplicar esfuerzos y malgastar recursos de red.
- **Procesamiento distribuido de las consultas.** Cada AB utiliza estrategias de query routing adaptativo para resolver las consultas externas de los SA. Gracias a la cooperación de los distintos componentes para la resolución de las consultas, hemos logrado distribuir la carga de trabajo en cada sitio. Permite manejar adecuadamente las fallas temporales en los servidores de índices así como problemas de conectividad en la red al utilizar más de un servidor de índices para el mismo grupo de interés, dando mayor robustez al servicio.
- **Utiliza meta-información.** Se utilizan pequeñas unidades de información durante la etapa de diseminación entre las entidades de la arquitectura, logrando escalabilidad en cuanto al manejo de recursos.
- **Utiliza URC (Uniform Resource Characteristic).** Se utiliza la meta-información de los objetos para la indexación, recuperación y visualización de resultados. Permite la búsqueda focalizada en ciertas categorías de objetos así como en características semánticas. Esto hace que los índices contengan información de calidad, al ser los propios generadores de información los encargados de confeccionar los URC de los objetos.
- **Interoperabilidad entre los distintos componentes.** El formato utilizado para la representación y transmisión de las consultas, los resultados y la meta-información de SI entre los Agentes de Búsqueda y los Servidores de Índices se basa en el estándar definido en STARTS [Gravano 97]. La utilización de SOIF (ver "Apéndice C") y del protocolo HTTP provee un mecanismo útil para unificar las distintas interfaces.

Consideramos que el diseño presentado logra los objetivos propuestos porque se basa en muchas de las ideas utilizadas por los prototipos distribuidos existentes para la Web y Bibliotecas Digitales que hemos adaptado para explotar las características particulares de ODS. La construcción de un modelo de referencia<sup>1</sup> nos permitiría evaluar el comportamiento de la arquitectura propuesta, sin embargo esto queda fuera de los alcances de este trabajo.

---

<sup>1</sup> Prototipo en el que solo se implementan funcionalidades mínimas con el objetivo de probar y evaluar el comportamiento y así determinar las mejoras a realizar al diseño, de manera tal de retroalimentarlo con pruebas experimentales.



## 6.2 Trabajos futuros

La construcción de un modelo de referencia para la arquitectura propuesta es una de la actividades principales a ser desarrollada. Un prototipo permitiría realizar pruebas experimentales para estudiar y analizar el comportamiento de la arquitectura propuesta, sacar estadísticas y eventualmente, determinar que partes deberían ser cambiadas u optimizadas.

Como paso previo a la implementación, sería importante determinar como el diseño del servicio de búsqueda afectará el núcleo de ODS. Será necesario identificar qué elementos de la arquitectura propuesta pueden resolverse utilizando lo ya existente o con mínimas modificaciones y cuales requieren de un cambio sustancial. A modo de ejemplo, para lograr las Interfaces de Usuarios solo será necesario modificar levemente a los SA ya existentes mientras que para lograr los Agentes de Búsqueda será necesario incorporar un nuevo componente al núcleo de ODS.

Otras actividades que creemos interesantes son:

- Comparar distintas políticas de uso en los Agentes de Búsqueda para la implementación de estrategias de query routing adaptativo y comparar su performance. Analizar distintos tipos de meta-información sobre los SI. El estudio de estrategias de query routing adaptativo es una área de investigación poco explotada.
- Sería interesante estudiar el grado de reutilización de la información disponible localmente en los SA para la resolución de consultas.
- Evaluar herramientas para la generación automática de MiG. La utilización de herramientas automáticas es muy importante para minimizar el trabajo requerido en los SA para mantener la MiG actualizada (fundamentalmente si los cambios en los objetos se dan con frecuencia). Podría evaluarse el uso de la herramienta “Essence” utilizada por los gatherers de Harvest [Hardy 96] para la sumarización de recursos.
- Analizar la escalabilidad de la arquitectura propuesta. Sería importante realizar o simular pruebas en entornos de gran escala con gran cantidad de usuarios, volumen de información y gran cantidad de servidores de índices.
- Comparar la performance del modelo propuesto con la obtenida por un servicio centralizado como DejaNews. Comparando también la escalabilidad de ambos.
- Evaluar y analizar la factibilidad de adaptar la arquitectura propuesta para el sistema News. Al no poder considerar a los servidores de news consumidores como una réplica para los grupos consumidos creemos que dificultaría la aplicación de la arquitectura propuesta.

## Apéndice A

# Servicios de Búsqueda Distribuidos para la Web

## A.1 Meta-Buscadores

Los meta-buscadores son sistemas que resuelven consultas reenviándolas (*forward*) a otros servicios de búsqueda como ser a AltaVista [AV], Excite [EX], etc. A partir de los resultados obtenidos, se encarga de combinarlos para generar la respuesta al usuario. El WebCrawler [WC] es un ejemplo de meta-buscador.

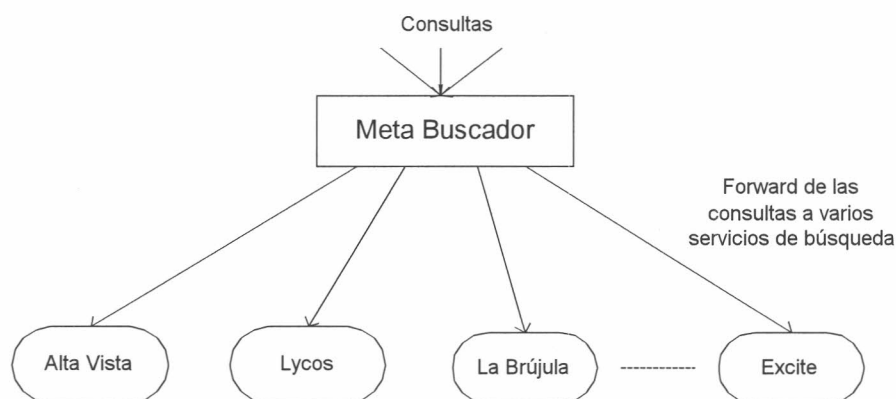


Gráfico A.1 Meta Buscador

También se los conoce como servicios de búsqueda *simultáneos* puesto que reenvían la consulta simultáneamente a diferentes servicios de búsqueda. Se caracterizan por no disponer de un índice local.

## A.2 Mallas Jerárquicas de Brokers

### A.2.1 Content Routing

Es una arquitectura basada en una jerarquía de *content routers* que en conjunto proveen servicios de búsqueda y navegación para los usuarios [Sheldon 95a]. Los nodos hojas son servidores de

información que mantienen sus colecciones de recursos, mientras que los nodos intermedios son los content routers. La arquitectura se observa en el Gráfico A.2.

Utiliza el concepto de *colección de documentos* como unidad de información en cada nodo de la jerarquía. La unidad de información que se transmite hacia arriba en la jerarquía son descripciones compactas de las colecciones llamadas *content labels*. Agrupar un conjunto de documentos en colecciones permite que el espacio de información pueda ser explorado usando distintos niveles de granularidad. Los content routers tratan a los content labels como documentos y son utilizados para la construcción de nuevas colecciones. La estructura de los content labels contiene información útil para proveer servicios de refinamiento y query routing.

El usuario interactúa con un conjunto manejable de colecciones, las que pueden ser expandidas para exponer en mayor detalle sus contenidos a medida que se avanza en la jerarquía.

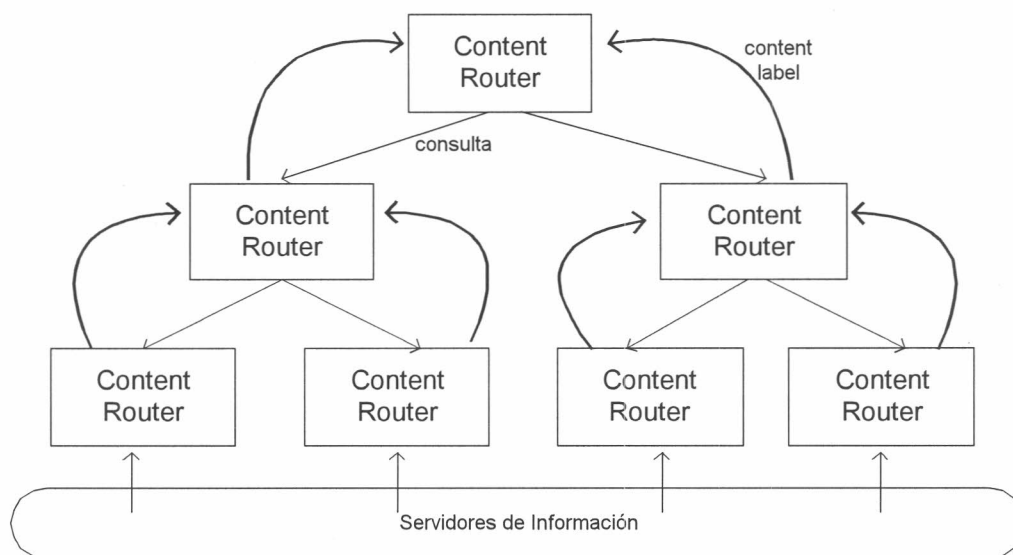


Gráfico A.2 Content routing

Discover [Sheldon 95b] y HyPursuit [Weiss 96] son dos prototipos implementados sobre la arquitectura de Content Routing.

Discover diseña un content router basado en el protocolo HTTP. El mismo provee acceso basado en consultas para un conjunto de 500 servidores WAIS. Este prototipo fue diseñado con el objetivo de probar que el refinamiento y ruteo de consultas es una solución posible para el problema de la búsqueda y el descubrimiento de recursos relevantes para los usuario. El Gráfico A.3 muestra la arquitectura de Discover.

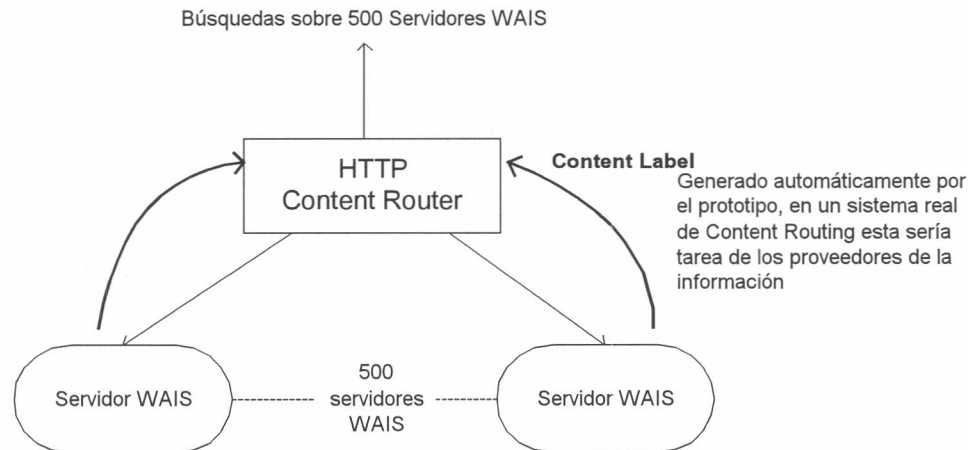


Gráfico A.3 Arquitectura de Discover

HyPursuit es un prototipo que estudia el uso de técnicas de clustering de hipertextos para la construcción de jerarquías de colecciones. Construye un content router basado en el protocolo HTTP para proveer acceso a más de 100 servidores Web. Analiza los contenidos y links en documentos HTML para construir las jerarquías de clusters.

## A.2.2 Whois++

Es una arquitectura basada en una jerarquía de Servidores de Índices [RFC 1913]. Los nodos hojas son Servidores Whois++ (servidores de directorios) mientras que los nodos intermedios son Servidores de Índices Whois++. La unidad de información que se transmite hacia arriba en la jerarquía son descripciones compactas de los contenidos almacenados por un Servidor Whois++ o un Servidor de Índice. Estas descripciones se denominan *centroides*. Los centroides son templates de pares atributos/valor.

Esta arquitectura fue diseñada para la búsqueda de información en el sistema de directorios Whois. Sin embargo, es una arquitectura similar a Content Routing, teniendo como principal diferencia la estructura de meta-información de contenido que utiliza y el algoritmo de query routing que utiliza.

La estructura de los centroides fue pensada fundamentalmente para el tipo de información de directorios. Los servicios de directorios almacenan la información en templates compuestos por pares atributo/valor.

Los encargados de implementar el query routing son los clientes [RFC 1914]. Los clientes generalmente utilizan un Index Server por defecto para comenzar las búsquedas. Para resolver una consulta, el cliente la envía al Index Server que tenga configurado por defecto, el cual, tomando como base los centroides que tenga indexados, obtiene los que mejor satisfagan la consulta devolviéndolos al cliente como *Query Referrals*. Los *query referrals* son utilizados por el cliente para avanzar en la resolución de la consulta, contactando a otros Index Servers.

### A.2.3 Harvest

Harvest [Bowman 95] se basa una arquitectura distribuida para la recolección de información que permite la construcción de Servidores de Índices de tópico específico llamados *Brokers*. Utiliza el par de componentes *Gatherer/Broker*.

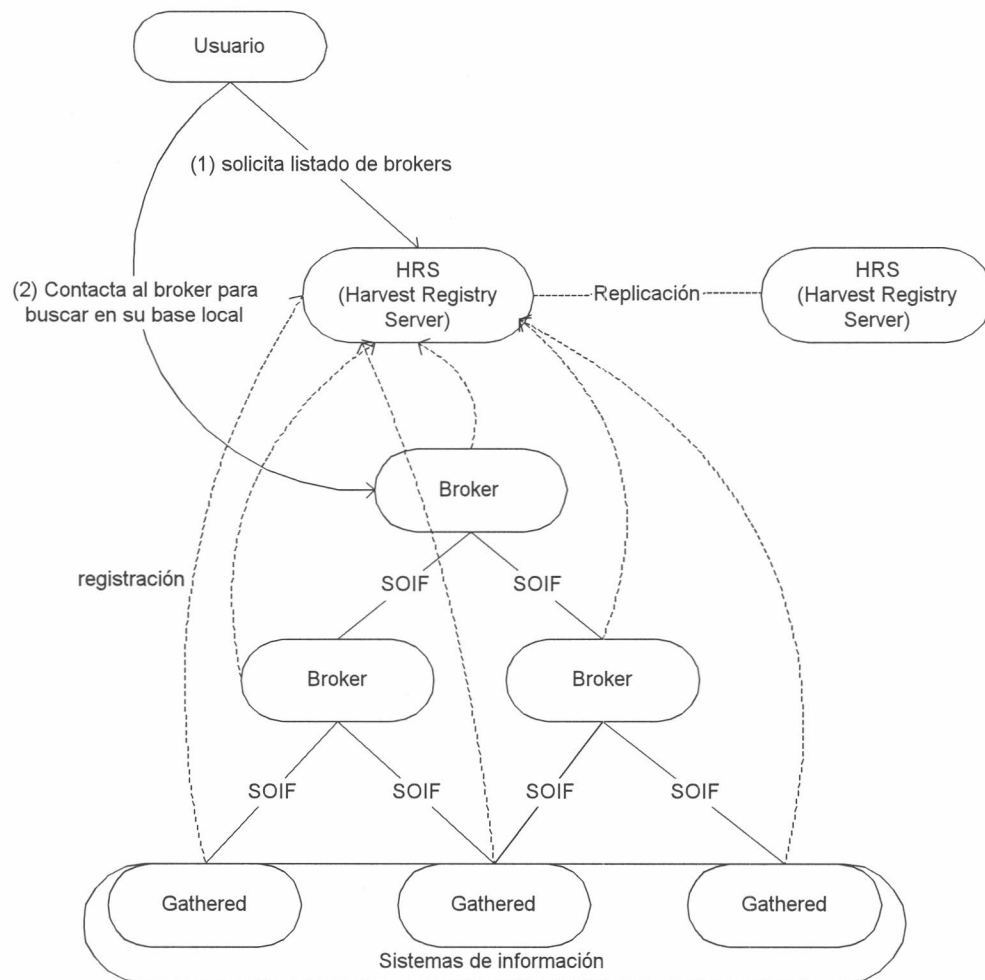


Gráfico A.4 Arquitectura de Harvest

Los *Gatherers* son programas que corren en el servidor de información y son los encargados de generar la meta-información de los recursos mantenidos en la colección local, haciéndolos disponibles para los *Brokers*. La meta-información de los recursos se codifica en formato SOIF [Hardy 96].

Los *Brokers* recolectan los objetos SOIF de uno o más *Gatherers* y/o *Brokers*, construyen un índice y proveen una interfaz de consulta. De esta manera no se transmiten recursos completos entre los nodos, optimizando el uso de los enlaces. A diferencia de Content Routing y Whois++, la unidad de transmisión no es una descripción de alto nivel para la colección de recursos, sino que cada *Broker* mantiene un índice que es la suma de los índices de nivel inferior. De esta manera cada *Broker* resuelve las consultas localmente, no teniendo procesamiento distribuido. La dinámica de la arquitectura se observa en el Gráfico A.4.

La arquitectura no define el mecanismo de composición de los brokers para la resolución cooperativa de consultas. Por el contrario, dispone de un Broker especial llamado *HRS (Harvest Registry Server)* cuya función es mantener un registro de todos los Broker y Gatherers existentes. Cada vez que un Gatherer o Broker se crea, debe registrarse en el HRS para que los usuarios puedan utilizarlo.

Para la resolución de consultas, un usuario interactúa con el HRS seleccionando algún Broker y luego debe contactarlo para formularle la consulta. El sistema no provee ningún mecanismo de query routing. Esto significa que la arquitectura no resuelve las consultas en forma cooperativa involucrando a más de un Broker. Por el contrario son los mismos usuarios los encargados de seleccionar los Brokers y enviarles las consultas.

## A.3 Mallas no jerárquicas de brokers

### A.3.1 AlephWeb

AlephWeb [Rodríguez 95] es un modelo basado en contextos o dominios, donde cada dominio representa un contexto organizativo. Las entidades de la arquitectura proveen todos los elementos necesarios para que cada dominio sea accesible al resto de la comunidad AlephWeb.

Cada dominio de la arquitectura de AlephWeb [Rodríguez 96] tiene tres componentes: *Servidor AlephWeb (SAW)*, *Gestor de Federación (GF)* y las *Bases de Datos (DB)*. Estos componentes se agrupan según su funcionalidad en “objetos orientados a la búsqueda” y “objetos orientados a la federación”. Los Servidores AlephWeb y las Bases de Datos pertenecen al primer grupo y son los encargados de mantener las referencias para cada proveedor de información (servidores Web) existente en su entorno, así como de crear/mantener un índice para permitir posteriores búsquedas (equivalentes a un servicio de búsqueda centralizado actual, pero restringido a su entorno).

En el segundo grupo (objetos orientados a la federación) están los Gestores de Federación que tienen la función de conectar los diferentes Servidores AlephWeb para la resolución cooperativa de consultas. Para mantener la estructura de federación activa realizan las siguientes tareas:

- Conectar diferentes servidores AlephWeb para resolver una consulta.
- Encaminar las consultas al conjunto de servidores más relevantes (query routing), descartando los que podrían producir información irrelevante.
- Actualizar la información utilizada (descripción) para guiar las consultas de acuerdo a los cambios de entorno que se vayan produciendo.
- Adaptar la información de “query routing” cuando la cardinalidad de la federación se modifica, bien sea por el alta o baja de alguno de los servidores participantes.

La dinámica de la arquitectura se observa en el Gráfico A.5.

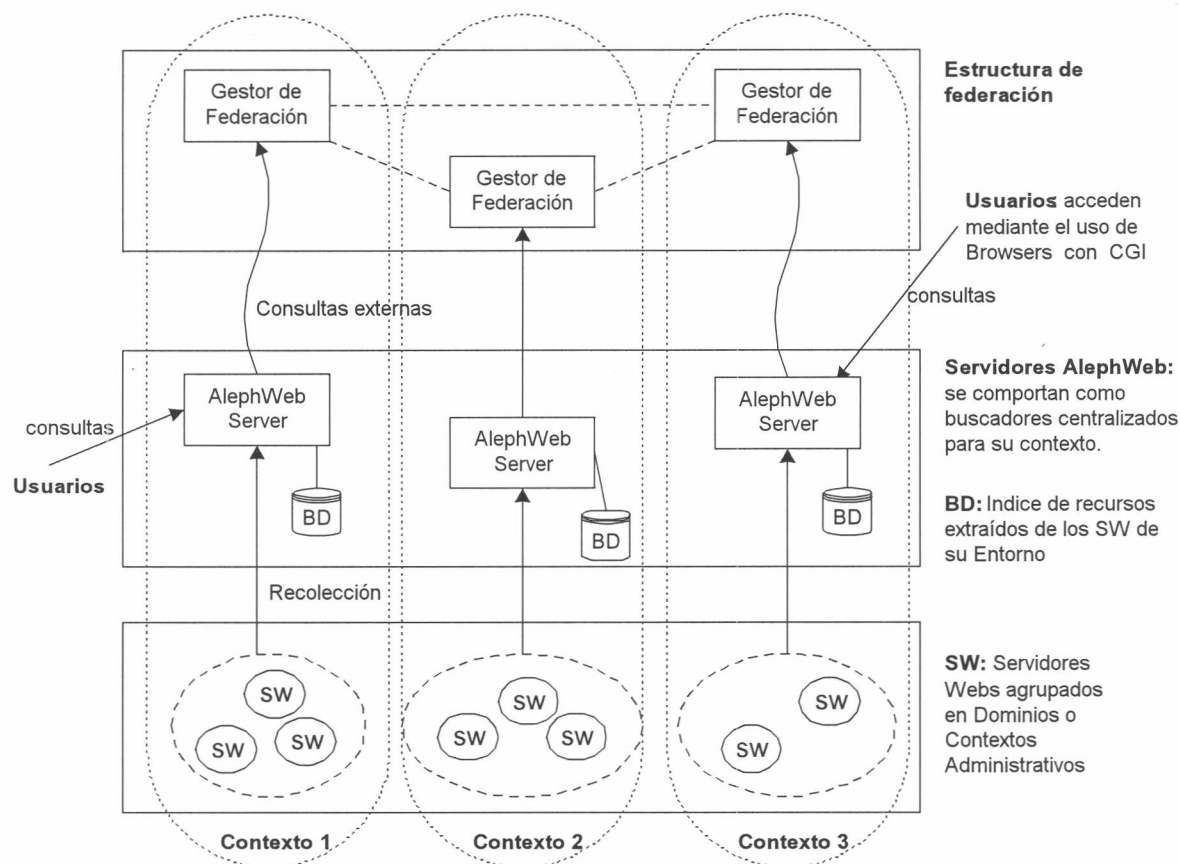


Gráfico A.5 Arquitectura de AlephWeb

En AlephWeb el servicio de búsqueda se distribuye utilizando la estructura de federación. En un entorno federado varias entidades deciden cooperar sin perder su independencia, incrementándose la libertad de acción de los miembros y por lo tanto no restringiendo su natural dinamismo. Cada entidad miembro está representada por su propio Servidor AlephWeb. Toda la comunidad de servidores cooperan entre sí proveyendo servicios de búsqueda globales.

La cooperación se lleva a cabo utilizando una estrategia de *query routing dinámico* [Sheldon 95a] que se basa en una descripción de lo que tiene cada servidor AlephWeb. Estas descripciones son almacenadas en las tablas dinámicas de *partners* de cada Gestor de Federación y son utilizadas para resolver las consultas externas de los usuarios. Cada búsqueda lanzada por el usuario se resuelve reenviándola *solo* a los servidores AlephWeb que tienen mayor probabilidad de responder correctamente.

AlephWeb propone una estructura de federación dinámica, a diferencia de Harvest [Bowman 95] a la que considera estática. Los agentes no están organizados en una jerarquía como es el caso de Discover [Sheldon 95b], por lo que se adapta más fácilmente a los cambios de entorno.



### A.3.2 Sistema distribuido basado en código móvil

Es una arquitectura compuesta por Servidores Web (SW), Servidores de Índices (SI) y Brokers [Gonçalves 97].

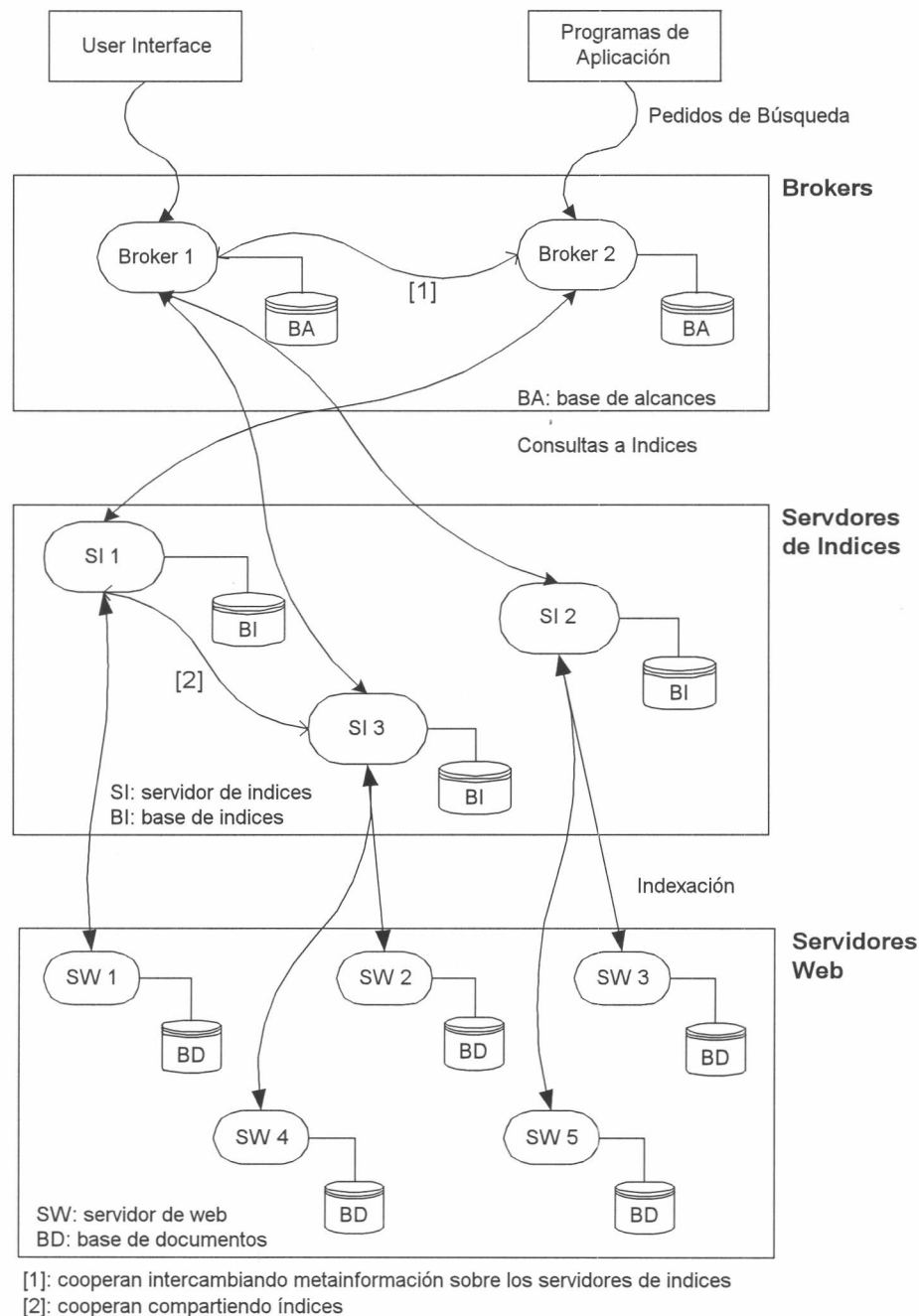


Gráfico A.6 Arquitectura Mobile Code Based

Cada SI tiene definido un *alcance* que le especifica la amplitud del espacio de recursos que abarcará. Su función es recolectar información de un conjunto de SW (contemplados en su alcance) y almacenarlos en su índice local. En los casos donde ya existiera en el sistema algún otro SI que cubra una parte del alcance que tiene definido, cooperará con éste reutilizando su

índice. Esta característica evita la duplicación de esfuerzos en la recolección de la información. Los SI registran sus alcances en un Broker al momento de la registración.

Cada Broker almacena el alcance de cada SI registrado así como el conocimiento de los otros Brokers existentes en la red. Cuando un Broker es contactado para resolver una consulta, busca una combinación de SI locales que pueda satisfacerla (analizando sus alcances con respecto al de la consulta). Si así no pudiera ser satisfecha contacta a otros Brokers para saber que otros SI existen y repetir el proceso.

Define un álgebra de vistas para generar los alcances y analizarlos. La dinámica de la arquitectura se observa en el Gráfico A.6.

### A.3.3 Ingrid

Es una arquitectura compuesta por *Publicadores*, *Servidores Ingrid* y *Navegadores* [Francis 95].

Los Publicadores tienen la función de incorporar *resource profiles*<sup>1</sup> (RPs) de los documentos en un Servidor Ingrid. El Servidor Ingrid tiene la función de instalar los RPs recibidos en la topología Ingrid y responder consultas. Los Navegadores proveen la interfaz con el usuario, su función es navegar la topología Ingrid para recuperar los RPs relevantes a los usuarios.

Los Servidores Ingrid están organizados en una red que sigue una estructura no jerárquica. Cada vez que un Servidor Ingrid recibe un RP de un Publicador, lo almacena en su base local y luego inserta el RP en la red vinculándolo con Servidores Ingrid que mantengan RPs similares. De esta forma un Servidor Ingrid quedará vinculado a otros Servidores solo si éstos últimos mantienen RPs similares a los almacenados por el primero.

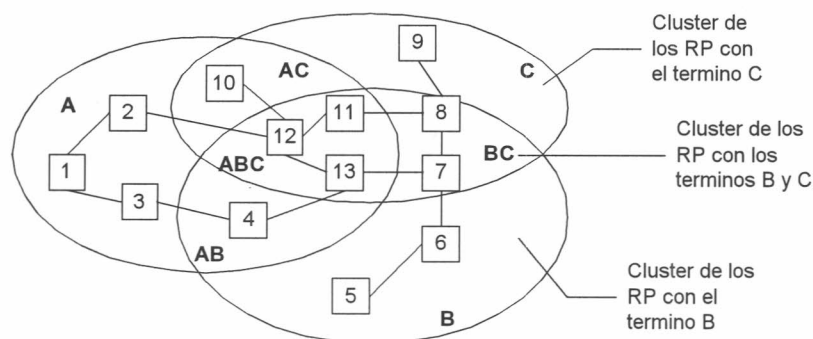


Gráfico A.7 Clusters de RPs

A nivel RP, el resultado final es una malla de RPs conectados que forman clusters, como muestra el Gráfico A.7.

<sup>1</sup> Especie de resumen para documento. Es una estructura de meta-información de recurso.

Existe un conjunto de servidores especiales llamados *Global Term Servers (GTS)* encargados de mantener, para cada categoría, un punto de entrada en la topología Ingrid.

Los Navegadores son los encargados de recorrer la topología Ingrid. Al contactar un Servidor Ingrid por una consulta dada, este último busca en su índice local RPs relacionados a la consulta. A partir de los RPs recuperados comienza a recorrer la topología Ingrid en busca de nuevos RPs almacenados en otros Servidores de la red Ingrid. En el caso de no disponer de RPs locales, contacta a un GTS para obtener puntos de entrada en la topología y así comenzar a recorrerla. En el Gráfico A.8 se observa la dinámica de los distintos componentes de la topología.

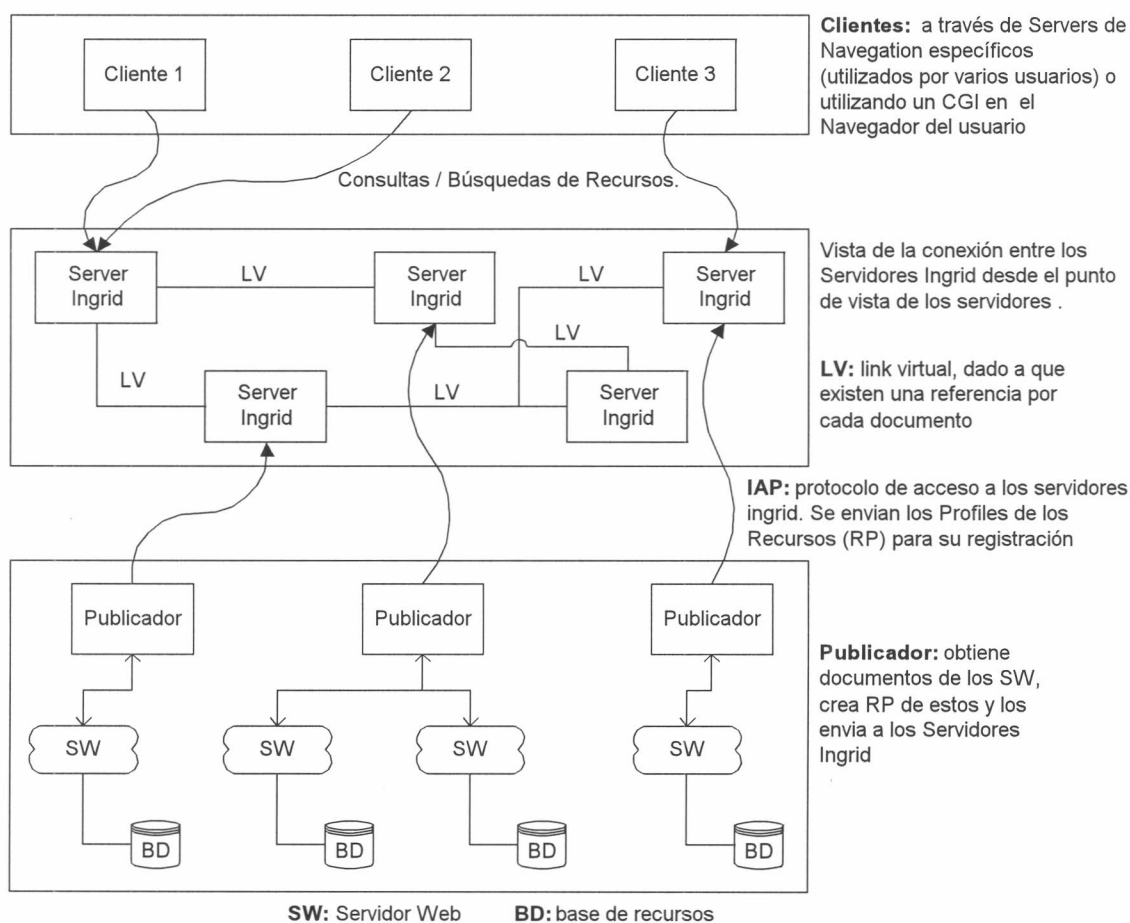


Gráfico A.8 Componentes de la topología Ingrid

Esta arquitectura utiliza los documentos como unidad de información para organizar la red de Servidores Ingrid. De esta manera, para resolver una consulta, es necesario recorrer vínculos entre documentos. Este mecanismo no reduce la complejidad del espacio de información, por lo cual el recorrido de los clusters puede ser una tarea costosa. Los autores detectaron estos problemas y proponen el uso de técnicas de cache en distintos puntos de la topología para minimizar la necesidad de recorrer distintos servidores durante la resolución de las consultas.

### A.3.4 Mecanismos Probabilísticos para el Descubrimiento de Recursos (MPRDS)

Es una arquitectura compuesta por *Agentes*, *Brokers* y *Repositorios de Información* [Schwartz 90].

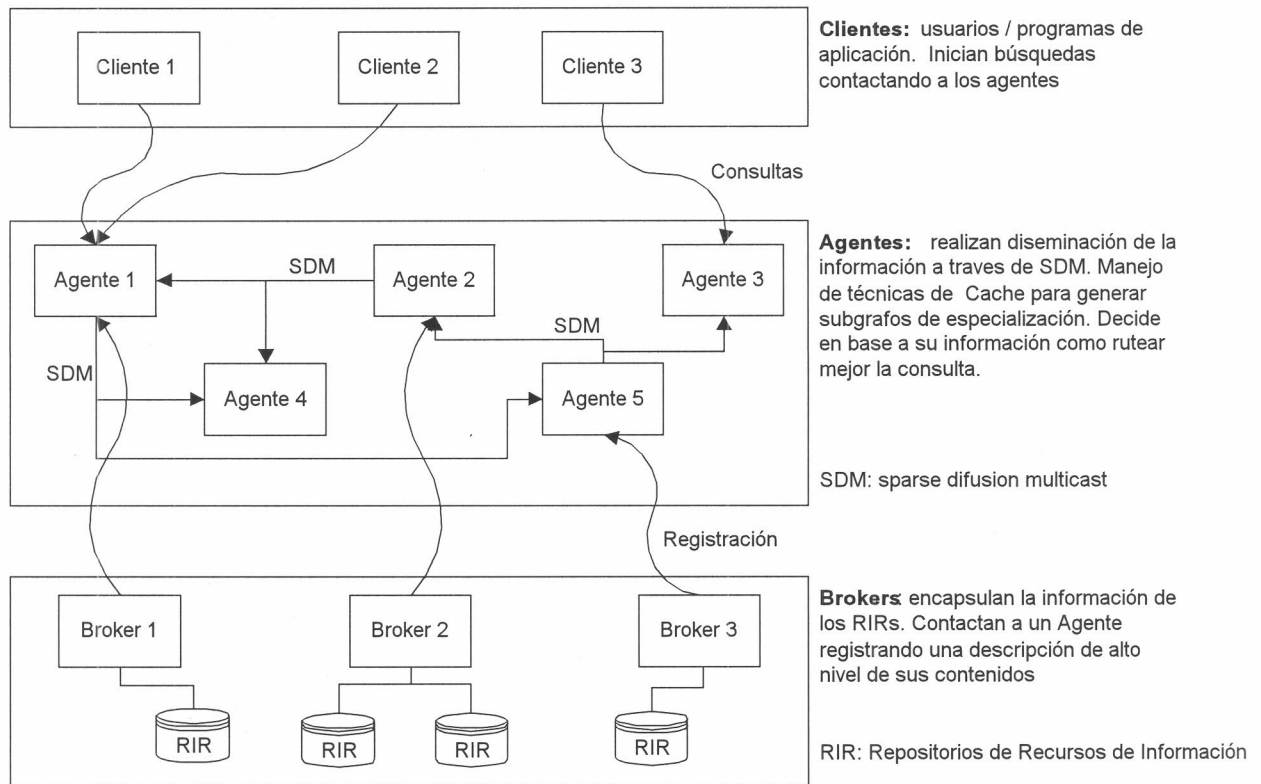


Gráfico A.9 Mecanismos probabilísticos para RD

Los Brokers brindan acceso a un conjunto de Repositorios de información (RIRs) elevando a algún Agente una descripción de alto nivel que sumalice los contenidos de la colección que alberga.

Los Agentes están organizados en una red que sigue una estructura no jerárquica. El objetivo de esta red es tender a formar dinámicamente *subgrafos de especialización* a partir de las consultas de los usuarios y de la información provista por otros Agentes.

Cada Agente mantiene en una cache la información de las descripciones recibidas por los Brokers suscriptos a éste. Además, mantiene el conocimiento adquirido en la interacción con otros Agentes también en una estructura de cache.

La interacción entre los Agentes se da utilizando un protocolo probabilístico llamado *Sparse Diffusion Multicast (SDM)*. Este protocolo permite contactar a un subconjunto del conjunto multicast, cuya cardinalidad está dada por un número fijo que indica el grado de densidad de la red. La interacción entre los Agentes se da en dos fases: búsqueda y diseminación.

Un Agente puede activar una primitiva de diseminación cuando recibe la registración de un Broker con el objetivo de propagar a otros Agentes la nueva información; o en intervalos regulares de tiempo enviando toda la información que contiene en su cache a otros Agentes y recibiendo de éstos sus caches.

Cuando un Agente recibe una consulta de un usuario, si no tiene localmente la información suficiente como para resolverla involucra a otros Agentes en la búsqueda. La dinámica de la arquitectura se observa en el Gráfico A.9.

## Apéndice B

# News

Usenet News [RFC 1036] es un foro electrónico público de Internet para la discusión de una gran variedad de temas de interés. Es utilizado por millones de personas localizadas alrededor de todo el mundo.

Las personas utilizan Usenet News por diferentes razones. Mientras que algunas lo hacen con fines académicos, otras lo hacen con propósitos recreativos y otras simplemente para seguir sus hobbies. La comunidad que integra Usenet News es inmensa y diversa. No todas las personas que integran Usenet News están vinculadas a universidades o instituciones académicas. Se pueden observar también muchas personas que trabajan en el sector privado y también del área militar.

Las discusiones que se dan en Usenet News están divididas por tópicos generales (categorías) que son llamados *newsgroups* (grupos de noticias o simplemente news). Estos son a su vez agrupados en su propia categoría, que se denomina jerarquía. Existen algunas jerarquías principales y varias alternativas (por ejemplo: *comp:* para lo referente a computacion, *rec:* orientado a las artes, hobbies y actividades recreativas, etc.). Muchas de las jerarquías se agrupan a su vez en nuevas jerarquías más específicas<sup>1</sup> (por ejemplo: *rec.art:* donde se discute arte en general, *rec.art.tv:* referente a televisión, etc.).

Cada mensaje ingresado a un newsgroup se denomina artículo. Un artículo se compone de dos partes: encabezado y cuerpo y su formato debe respetar el estándar adoptado para los mensajes de correo electrónico de Internet, definidos en el RFC 822.

El encabezado esta formado por muchos campos. Cada uno de estos campos identifica una parte de la información correspondiente al artículo. Existen campos que son obligatorios mientras otros son optativos. A modo de ejemplo, algunos de estos son:

- *Newsgroup:* nombre/s de el/los newsgrop/s en donde se va a intentar ingresar el artículo.
- *Subject:* una línea describiendo el tema del artículo.
- *Date:* la fecha en la que el artículo fue ingresado.

---

<sup>1</sup> En 1995 existían alrededor de 2000 newsgroup y hoy en día son más de 15000.

- *From*: el nombre y la dirección electrónica de la persona que ingresó el artículo.
- *Distribución*: es utilizado para indicar el alcance del artículo.(local al sitio, al país, al mundo, etc.).

El Cuerpo de un artículo es lo que el usuario escribe. Este puede ser desde una corta y simple pregunta hasta un gran documento. No importa su longitud, siempre se lo considera un artículo.

La acción de ingresar un artículo a un newsgroup se denomina *posting*. Un artículo puede ser ingresado en más de un newsgroup a la vez, acción que se denomina *cross-posting* (ingreso cruzado). Esta mal considerado realizar un cross-posting sin buenas razones.

Usenet News puede ser modelado como un grafo dirigido. Cada nodo en el grafo se corresponde a un host y cada arco a una camino de transmisión de un host a otro. Cada arco es rotulado con un patrón que identifica la clases de newsgroups que son pasados por ese camino de transmisión. Cuando un host A envía una clase de newsgroups al host B y el host B envía los mismos newsgroups al host A, el arco que los une es bidireccional.

Entre dos nodos existe una relación proveedor-consumidor, donde el host consumidor acuerda con el host proveedor cuales de los newsgroups disponibles va a consumir. Este acuerdo lo realizan los administradores de los host y es configurado manualmente en los servidores.

Un nodo puede tener múltiples proveedores y múltiples consumidores para el mismo o diferentes newsgroups. Tomando una vista del grafo por cada newsgroup se obtiene un subgrafo conexo. Este subgrafo define la cadena de distribución que sigue un artículo al ser ingresado en un sitio para un newsgroup determinado.

La mayoría de los newsgroups son públicos. Nadie decide cuando el artículo puede o no ser incluido en el newsgroup especificado.

Sin embargo, existen algunos grupos a los cuales no se puede postear un artículo sin previa autorización. Estos newsgroups se denominan *moderados*, y la persona que autoriza el pasaje de los artículos se llama *moderador*. Cuando alguien envía un artículo a un newsgroup moderado, el artículo es enviado automáticamente a través de un e-mail al moderador de ese newsgroup. Es el moderador el que realmente realiza el posting del artículo al newsgroup.

Cuando un artículo es posteoado en un host a una lista de newsgroup, este host acepta localmente el artículo. Posteriormente lo reenvía a todos sus vecinos que estén interesados en al menos uno de los newsgroups a los que esta dirigido (el nodo A decide que el nodo B esta interesado en un newsgroup si matchea con el patrón que rotula el arco que va desde el nodo A al B). Cada host que recibe un artículo entrante lo examina para estar seguro que realmente lo desea, si lo acepta entonces lo almacena localmente luego de lo cual lo vuelve a reenviar a todos sus vecinos interesados. El proceso continua hasta que todos los hosts de la red interesados reciben una copia del artículo.



El espacio de almacenamiento necesario en un servidor de news para almacenar todos los artículos es muy grande. Por esta razón, un artículo no permanece en el servidor por tiempo ilimitado. Los servidores news utilizan un sistema de expiración de artículos. Cuando un artículo llega a un servidor, este puede permanecer por un período de tiempo antes de que se lo elimine. Existen dos tiempos de expiración. Uno, es el que establece el autor en el encabezado del artículo y que depende de la información que esta publicando (por ejemplo el anuncio de un workshop). El otro, es establecido por el administrador de cada servidor y fijado por jerarquía, dependiendo del uso y los recursos que disponga el servidor. Es por ello que puede darse el caso que, para un mismo newsgroup, existan artículos disponibles en algunos servidores mientras que en otros ya hayan sido eliminados.

Es importante notar que como la mayoría de los servicios de Internet, no existe una organización central que sea dueña, controle o maneje los Usenet News. De hecho, los participantes de Usenet News frecuentemente la denominan el mundo, simplemente por la anarquía que la gobierna que se deriva de su naturaleza distribuida. Debido a que Usenet News funciona transfiriendo artículos entre computadoras, el único control que cualquier sitio individualmente tiene es sobre los artículos que ingresan y salen del sitio.

Esto no quiere decir que no existan reglas. Con el pasar de los años, los participantes han acordado un conjunto amplio de normas. Estas implican un gran costo administrativo para las personas que manejan los sitios. Entre estas normas tenemos, el método utilizado para crear un nuevo newsgroup y las condiciones (denominadas *netiquette*) que deben ser utilizadas cuando se ingresen artículos al mismo.

Algunas de estas netiquette establecen, por ejemplo:

- Leer un tiempo el newsgroup, antes de ingresar algún artículo en él.
- Antes de ingresar una consulta, verificar si la misma no esta contestada en las FAQ del newsgroup.
- Cuando se contesta a un artículo tratar de ingresar material nuevo. No simplemente un mensaje que incluya al mensaje original con el agregado de un texto que diga “estoy de acuerdo”
- Cuando se realiza un pregunta, para la cual se esperan muchas respuestas, el autor del artículo debería ofrecerse a recibir las respuestas en forma personal e ingresar posteriormente otro artículo conteniendo un resumen de las mismas.
- No corregir los errores de ortografía que cometen otros en los artículos. Tratar de verificar bien la ortografía antes de ingresar un artículo en los newsgroups.
- No escribir los artículos en mayúsculas. Por convención, toda palabra en mayúscula se utiliza para dar mayor énfasis.

Por último, el sistema Usenet News puede ser comparado con el sistema de Mailing Lists (Listas de Discusión) de Internet. Un newsgroup se correspondería con una lista de discusión y un artículo postado en un newsgroup se correspondería con un mensaje de correo electrónico dirigido a la lista. Al igual que en News, si bien la mayoría de las listas son públicas, existen listas moderadas en las cuales los mensajes deben ser aprobados por un moderador. La diferencia entre ambos sistemas es que en News se almacena solo una copia de cada artículo postado a un newsgroup (en cada servidor que consume el newsgroup), mientras que en el sistema de listas cada servidor de correo almacenará una copia del mensaje por cada dirección de correo correspondiente a un usuario local que esté suscrito a la lista.

## Apéndice C

# Ejemplos de codificación en formato SOIF

Con el objetivo de lograr la interoperabilidad entre los componentes de la arquitectura propuesta (Agentes de Búsqueda, Agentes de Servicio, Servidores de Índices) describimos a continuación un formato que puede ser utilizado para la representación y transmisión de las consultas, de los resultados y de la meta-información de SI. Dicho formato se basa en el estándar definido en el proyecto STARTS [Gravano 97] el cual provee un mecanismo para unificar las interfaces de consultas a múltiples servidores de índices.

En STARTS se utilizan los pares atributo/valor de un objeto SOIF para la codificación de consultas y de resultados que son transmitidos utilizando el protocolo HTTP. De esta manera los objetos SOIF correspondientes a las consultas se podrían enviar en una página HTML mediante el método POST utilizando un tag “soif” y cuyo valor sea el objeto SOIF que representa la consulta. Los resultados de las consultas también serían transmitidos con el protocolo HTTP encapsulados en objetos SOIF. En lo que respecta a la codificación y transmisión de la meta-información de SI también podrían utilizarse objetos SOIF.

## C.1 Descripción formal de SOIF

SOIF (Summary Object Interchange Format) es una sintaxis legible por una computadora que se utiliza para transmitir resúmenes estructurados de objetos. Actualmente, se la utiliza principalmente en el contexto de la Web.

Fue introducido en 1995 como parte del proyecto Harvest [Bowman 95]. Se deriva de una modificación de los templates IAFA (Internet Anonymous FTP Archives) [Beckett 95] del IETF Working Group.

Cada objeto SOIF tiene tres componentes principales: un *tipo de template*, un *URL* y cero o más *pares Atributo/Valor*. El tipo de template se utiliza para identificar el conjunto de Atributos contenidos en un determinado objeto SOIF. El URL se utiliza como identificador del objeto SOIF. Por último, los pares Atributo/Valor contienen meta-información acerca del recurso referenciado por el URL. Los pares se componen de un identificador de Atributo, la longitud del Valor, un delimitador y el Valor.

La sintaxis de SOIF (tal como se describe en [Hardy 96]) está definida por la siguiente gramática:

SOIF	-> OBJECT SOIF   OBJECT
OBJECT	-> @ TEMPLATE-TYPE { ATTRIBUTE-LIST }
ATTRIBUTE-LIST	-> ATTRIBUTE ATTRIBUTE-LIST   ATTRIBUTE
ATTRIBUTE	-> IDENTIFIER {VALUE-SIZE} DELIMITER VALUE
TEMPLATE-TYPE	-> Alpha-Numeric-String
IDENTIFIER	-> Alpha-Numeric-String
VALUE	-> Arbitrary-Data
VALUE-SIZE	-> Number
DELIMITER	-> ";<tab>"

## C.2 Consultas

Describiremos las características básicas del lenguaje de consultas que un Servidor de Índices debe soportar. Este lenguaje se basa en un subconjunto simple de consultas type-101 del estándar Z39.50-1995<sup>1</sup>.

Toda consulta tiene dos componentes:

- *Una expresión de filtro;*
- *Una expresión de ranking.*

Una *expresión de filtro* es booleana por naturaleza y define que tipo de objetos serán incluidos en la respuesta. La *expresión de ranking*, en cambio, permite asociar a cada objeto un puntaje para ordenarlos adecuadamente.

### **Ejemplo:**

*Si una consulta tiene la siguiente expresión de filtro:*

```
((author "García Molina") and (title "databases"))
```

*y como expresión de ranking:*

```
list((body-of-text "distributed") (body-of-text "databases"))
```

*La consulta devolverá como resultado objetos que tengan "García Molina" como uno de sus autores y la palabra "databases" en su título. Los objetos que cumplan la expresión de filtro serán ordenados acorde a como satisfagan las palabras "distributed" y "databases".*

En principio, una consulta no necesariamente debe contener una expresión de filtro, en cuyo caso se asume que todos los objetos satisfacen la consulta y serán ordenados según la expresión de

<sup>1</sup> <http://lcweb.loc.gov/z3950/agency/1995doce.html>

ranking. De la misma forma, una consulta no necesariamente debe contener una expresión de ranking, en cuyo caso el resultado será el conjunto de objetos que satisfacen la expresión de filtro.

Ambos tipos de expresiones pueden contener múltiples *términos*. Las expresiones de filtro y ranking combinan dichos términos con operadores como "and" y "or". La expresión de ranking combina los términos utilizando el operador "list" los que deben tener asociado un *peso* (*weight*) que indica la importancia relativa del término en la consulta.

Un *término* se define como una lista no ordenada de atributos. A su vez, un atributo puede ser un *campo* o un *modificador*. Por ejemplo, el término (date-last-modified > "1997-08-01") tiene un campo date-last-modified y un modificador ">". Matchean este término los objetos cuya fecha de modificación es posterior al 1 de agosto de 1997.

STARTS define un conjunto de atributos. Algunos campos son obligatorios (**Req**) en el sentido que cualquier Servidor de Índices debe reconocerlos como parte de una consulta, mientras que otros son opcionales. Aquellos marcados con (**Nuevo**) son los que hemos incorporado en la arquitectura propuesta.

**Campo:** Es la porción del texto asociada a un término (por ejemplo, author o title). Es obligatorio especificar un campo para cada término.

- Title (**Req**)
- Author
- Body-of-text
- Document-text
- Date/time-last-modified (**Req**)
- Any (**Req**)
- Linkage: URN del objeto.
- Cross-reference-linkage: Lista de URNs asociados al objeto.
- Language: Por ejemplo: el término (language "en-US") matchea aquellos objetos cuyo valor en el campo language es "en-US es".
- Grupo-de-interes: (**Nuevo**) (**Req**) Indica que los objetos del conjunto resultado solo deben incluir los pertenecientes al grupo de interés que se especifique.

**Modificadores:** Especifica el valor que representa el término (por ejemplo: tratar el término como right-truncation, left-truncation):

- <, <=, =, >=, >, != Si fuera aplicable, por ejemplo a campos como "Date/time-last-modified", por defecto vale "=".
- Right-truncation: Por defecto no trunca a derecha, toma el valor tal cual es.
- Left-truncation: Por defecto no trunca a izquierda, toma el valor tal cual es.
- Case-sensitive: Por defecto no es sensitivo.

## C.2.1 Sintaxis

Una consulta es un objeto SOIF del template "SQuery".

*Ejemplo:*

El número entre llaves luego de cada atributo (por ejemplo "103" luego del atributo FilterExpression) indica el número de bytes para dicho atributo, para facilitar el parsing.

```
@SQuery{
Version{10}: STARTS 1.0
FilterExpression{103}: ((autor "García Molina") and (title "database") and
((grupo-de-interes "1") or (grupo-de-interés "2")))
RankingExpression{61}: list((body-of-text "distributed") (body-of-text
"databases"))
DropStopWords{1}: T
DefaultAttributeSet{7}: basic-1
DefaultLanguage{5}: en-US
AnswerFields{11}: title autor
MinDocumentScore{3}: 0.5
MaxNumberDocument{2}: 10
}
```

A continuación especificamos cada atributo del template "SQuery" con su correspondiente valor:

Atributo	Valor
Version	Alpha-Numeric-String
FilterExpression	FILTER
RankingExpression	RANKING
DropStopWords	Boolean
DefaultAttributeSet	ATTRIBUTE-SET
AnswerFields	SORT-FIELD-LIST
MinDocumentScore	Number
MaxNumberDocuments	Number

La gramática para los valores de los atributos es la siguiente:

FILTER	-> TERM   (TERM PROX-OP TERM)   (FILTER BOOLEAN-OP FILTER)
RANKING	-> TERM   (TERM PROX-OP TERM)   list( RANKING-LIST)   List( W-RANKING-LIST)   (RANKING BOOLEAN-OP RANKING)   (W-RANKING BOOLEAN-OP W-RANKING)
TERM	-> L-STRING   ( MODIFIER-LIST L-STRING)   ( FIELD MODIFIER-LIST L-STRING)
L-STRING	-> "String"   [ LANGUAGE "String" ]
RANKING-LIST	-> RANKING   RANKING RANKING-LIST
W-RANKING-LIST	-> W-RANKING   W-RANKING W-RANKING-LIST
W-RANKING	-> ( RANKING Number )
BOOLEAN-OP	-> and   or   and-not
PROX-OP	-> prox[ Number, Boolean]
ATTRIBUTE-SET	-> basic-1   ...
FIELD	-> BASIC1-FIELD   [ basic-1 BASIC1-FIELD ]   ...
BASIC1-FIELD	-> title   autor   body-of-text   document-text   Date-last-modified   any   linkage   linkage-type   Cross-reference-linkage   Language   grupo-de-interes
FIELD-LIST	-> FIELD   FIELD FIELD-LIST
MODIFIER	-> BASIC1-MODIFIER   { basic-1 BASIC1-MODIFIER }   ...
BASIC1-MODIFIER	-> RELATION   left-truncation   right-truncation   Case-sensitive
RELATION	-> <   >=   =   >=   >   !=
MODIFIER-LIST	->   MODIFIER MODIFIER-LIST
LANGUAGE	-> LanguageCode   LanguageCode-CountryCode   ...
SORT-FIELD-LIST	-> SORT-FIELD   SORT-FIELD SORT-FIELD-LIST
SORT-FIELD	-> {FIELD a}   {FIELD d}   {SCORE d}

## C.3 Resultados

En respuesta a una consulta, un Servidor de Índices devuelve los objetos que la satisfacen. Dado que un SI puede modificar una consulta antes de procesarla (tal vez porque no soporta algunos campos especificados), también retornará la consulta tal cual fué procesada.



### **Ejemplo:**

*Supongamos que un SI no soporta la expresión de ranking como parte de una consulta. Considerando una consulta con la expresión de filtro*

```
((autor "García Molina") and (title "databases"))
```

*y la expresión de ranking*

```
list ((body-of-text "distributed") (body-of-text "databases"))
```

*dado que el SI ignorará la expresión de ranking por no soportarla, la consulta que realmente procesará será:*

```
((autor "García Molina") and (title "databases"))
```

Cuando los Agentes de Servicio reciban los resultados, éstos podrán combinarlos de diferentes formas: agrupados por grupo de interés, agrupados por importancia de los términos de la consulta, etc.. Para ello, los Servidores de Índices deberán devolver, para cada objeto del conjunto resultado, los siguientes datos:

- El/los grupo/s de interés a los que pertenece el objeto (estos deben ser un subconjunto del especificado en la consulta).
- Estadísticas para cada término en la expresión de ranking.
- Term-frequency: el número de veces que el término aparece en el objeto.
- Term-weight: el peso (*weight*) que tiene el término en el objeto.
- Document-frequency: el número de objetos que tiene el SI para el grupo que contienen el término.
- Document-size: el tamaño del documento en bytes.

### **C.3.1 Sintaxis**

Los resultados de una consulta comienzan con un objeto SOIF de tipo "SQResults" seguido de una serie de objetos SOIF de tipo "SQRDocument". Cada objeto "SQRDocument" se corresponde con un objeto del conjunto resultado.

### Ejemplo:

Tomando como ejemplo la consulta definida en C.2.1, el resultado sería el siguiente:

```
@SQResults{
Version{10}: START 1.0
Source{8}: Source-1
ActualFilterExpression{103}: ((author "García Molina") and (title "databases")
and ((grupo-de-interes "1") or (grupo-de-interés "2")))
ActualRankingExpression{56}: ((body-of-text "databases") (body-of-text
"distributed"))
NumDocsSOIFs{1}: 4
}

@SQRDocument{
Version{10}: START 1.0
Grupos-de-interes{3}: 1,2
Linkage{14}: URN del objeto
Title{44}: Generalizing GLOSS to Vector-Space Databases
Author{34}: Luis Gravano, Hector García-Molina
TermStats{79}: (body-of-text "distributed") 10 0.31 190
                (body-of-text "databases") 15 0.51 232
DocSize{3}: 248
}
. . .
@SQRDocument{
. . .
}
```

A continuación especificamos los atributos del template "SQResults" con su correspondiente valor:

Atributo	Valor
Version	Alpha-Numeric-String
Source <sup>1</sup>	SOURCE-ID-LIST
ActualFilterExpresion	FILTER
ActualRankingExpression	RANKING
NumDocsSOIFs <sup>2</sup>	Number

A continuación especificamos cada atributo del template "SQRDocument" con su correspondiente valor:

Atributo	Valor
Version	Alpha-Numeric-String
Grupos-de-interes	GRUPO-ID-LIST
FIELD	Alpha-Numeric-String
TermStats	TERM-STATS-LIST
DocSize	Number (en kb)

<sup>1</sup> Servidor de Indices de donde proviene la respuesta

<sup>2</sup> Número de objetos SOIF de tipo "SQRDocument" contenidos en la respuesta

Donde la gramática para los valores de los atributos es la siguiente:

```
GRUPO-ID-LIST      -> GRUPO-ID | GRUPO-ID, GRUPO-ID-LIST
GRUPO-ID           -> Number | String1
TERM-STATS-LIST    -> TERM Number Number Number |
                    TERM Number Number Number TERM-STATS-LIST
```

## C.4 Meta-información de SI

La arquitectura utilizará meta-información de SI para representar a cada SI<sup>2</sup>. Dicha estructura contendrá atributos útiles para definir las características de los SI que serán utilizadas por los Agentes de Búsqueda y los Responsables de grupo.

Para la definición de los pares atributo/valor a incluir en la meta-información de SI tomamos como base el tipo *Source Metadata* de STARTS. La estructura Source Metadata la hemos modificado para incluir los atributos particulares necesarios en la arquitectura propuesta que indicaremos con **(Nuevo)**. Algunos atributos definirán características globales al SI mientras que otros corresponderán a características específicas para cada grupo de interés indexado por el SI. Los atributos obligatorios estarán indicados con **(Req)**.

### Globales:

- **FieldSupported: (Req)** Qué campos adicionales, además de los requeridos, soporta un Servidor de Indices en las consultas que se le especifiquen.
- **QueryPartsSupported: (Req)** Cuándo el Servidor de Indices solo soporta expresiones de ranking ("R"), solo expresiones de filtro ("F") o ambas ("RF").
- **Linkage: (Req)** Dirección y port al cual se deben encaminar las consultas.
- **Contact:** Información acerca del administrador del Servidor de Indices.
- **Connectivity: (Nuevo) (Req)** Información sobre el estado de los enlaces del SI, útil para la definición de vistas de SI.
- **NumGroupSOIFs: (Nuevo) (Req)** Indica cuantos grupos indexa el SI.

<sup>1</sup> Este valor depende de como se identifique un grupo de interés en ODS.

<sup>2</sup> Ver "5.2.4.3 Meta-información de SI"

### **Específicos por grupo de interés:**

- **Grupo: (Req)** Indica para que grupo de interés se proveen los atributos.
- **CantDoc: (Req)** Cantidad de objetos indexados por el SI para el grupo. Se utiliza para determinar el grado de actualización del SI con respecto a la totalidad de objetos producidos para el grupo.
- **DateChanged: (Req)** Fecha de última modificación de los atributos específicos del grupo.
- **DateExpires: (Req)** Fecha de expiración de la meta-información del grupo, en cierta forma indica hasta que momento la información puede considerarse como válida.
- **Usage:** Indica el promedio de consultas respondidas (para el grupo) para otros Agentes de Servicio. Esta información es útil para determinar si el SI es muy utilizado para el grupo.

### **C.4.1 Sintaxis**

La meta-información de SI comienza con un objeto SOIF de tipo "SMetaAttributes" seguido de una serie de objetos SOIF de tipo "SMetaGroup". Existirá un objeto "SMetaGroup" por cada grupo de interés indexado por el SI y contendrá la información específica del grupo.

#### *Ejemplo:*

```
@SMetaAttributes{
Version{10}: START 1.0
FieldSupported{16}: [basic-1 author]
QueryPartsSupported{2}: RF
Linkage:{36}: http://www.uba.ar:8000/cgi-bin/query
Connectivity{10}: (formato a definir)
NumGroupSOIFs{1}: 2
}

@SMetaGroup{
Version{10}: START 1.0
Grupo{1}: 1
CantDoc{4}: 1000
Date-Changed{10}: 1999-03-20
Date-Expires{10}: 1999-04-10
}
@SMetaGroup{
Version{10}: START 1.0
Grupo{1}: 2
CantDoc{3}: 100
Date-Changed{10}: 1999-03-21
Date-Expires{10}: 1999-03-30
}
```

A continuación especificamos cada atributo del template "SMetaAttributes" con su correspondiente valor:

Atributo	Valor
Version	Alpha-Numeric-String
FieldSupported	L-FIELD-LIST
QueryPartsSupported	Alpha-String
Linkage	Alpha-Numeric-String
Connectivity	LINK-STATE-LIST
NumGroupSOIFs <sup>1</sup>	Number

A continuación especificamos cada atributo del template "SMetaGroup" con su correspondiente valor:

Atributo	Valor
Version	Alpha-Numeric-String
Grupo	GRUPO-ID
CantDoc	Number
DateChanged	DATE
DateExpires	DATE
Usage	Number

Donde la gramática para los valores de los atributos es la siguiente:

L-FIELD-LIST           -> L-FIELD | L-FIELD L-FIELD-LIST

L-FIELD               -> FIELD | (FIELD LANGUAGE-LIST)

DATE                   -> Number "-" Number "-" Number

LINK-STATE-LIST       -> LINK-STATE<sup>2</sup> | LINK-STATE LINK-STATE-LIST

<sup>1</sup> Número de objetos SOIF de tipo "SMetaGroup".

<sup>2</sup> La información de estado de cada enlace de un SI dependerá del algoritmo que utilicen los Responsables de Grupo para definir las regiones de conectividad.

# Bibliografía

- [AV] *Altavista Internet Service*.  
URL: <http://www.altavista.com/>
- [Alberti 92] Alberti, B.; Anklesaria, F.; Linkner, P.; McCahill, M.; Torrey, D.  
*The Internet Gopher protocol: A distributed document search and retrieval protocol*.  
University of Minesota Microcomputer and Workstation Networks Center. 1991. Revisado en 1992.
- [Albitz 98] Albitz, P.; Liu, C.  
*DNS and BIND, 3<sup>rd</sup> Edition*.  
ISBN 1-56592-512-2, O'Reilly & Associates Inc.. Setiembre 1998.
- [Baleani 98] Baleani, M.; Cervini, C.; Rodríguez, L.  
*Localizadores Universales de Recursos en Internet*.  
Tesis de Licenciatura, Departamento de Computación, Facultad de Ciencias Exactas y Naturales,  
Universidad de Buenos Aires. Diciembre 1998.
- [Beckett 95] Beckett, D.  
*IAFA Templates in Use as Internet Metadata*.  
4th Int'l WWW Conference, Diciembre 1995.
- [Berners-Lee 92] Berners-Lee, T; Cailliau, R.; Groff, J.; Pollermann, B.  
*World-Wide Web: The information universe*.  
Electronic Networking, 2(1):52-58, 1992.
- [Blaze 93] Blaze, M.  
*Caching in Large Scale Distributed File Systems*.  
Ph.D, Department of Computer Science, Princeton University. Enero 1993.
- [Bowman 94] Bowman, M.; Danzing, Peter B.; Manber, U.; Schwartz, Michael F.  
*Scalable Internet Resource Discovery: Research Problems and Approaches*. Communications of the  
ACM. Vol. 37, Nro. 8 pp. 98-107. Agosto 1994.
- [Bowman 95] Bowman, M.; Danzig, P.; Hardy, R.; Manber, U.; Schwartz, M., Wessels, D.  
*Harvest: A scalable, customizable discovery and access system*.  
Technical Report CU-CS-732-94, Department of Computer Science. University of Colorado-  
Boulder. Agosto 1994. Revisado en marzo 1995.
- [Cornell-DL] *HomePage del grupo de investigación sobre bibliotecas digitales de la Cornell University*.  
URL: <http://www.cs.cornell.edu/NCSTRL/CDLRG/cdlrg.htm>
- [Denenberg 96] Denenberg, R.  
*Structuring and Indexing the Internet*.  
Library of Congress. Diciembre 1996.
- [DN] *Deja News Inc.*  
URL: <http://www.dejanews.com/>

- [Emtage 92] Emtage, A.; Deutsch, P.  
*Archie: An electronic directory service for the Internet.*  
Proceedings of the Winter USENIX Conference, pp 93-110. Enero 1992.
- [EX] Excite Search Engine.  
URL: <http://www.excite.com>
- [Francis 95] Francis, P.; Takashi, K.; Shin Ya, S.; Susumu, S.  
*Ingrid: A Self-Configuring Information Navigation Infrastructure*  
NTT Software Laboratories. Four International Conference for WWW. 1995.  
URL: <http://www.w3.org/Conferences/WWW4/Papers/300>
- [Gonçalves 97] Gonçalves, P.; Meira, S.; Salgado, A.  
*A Distributed Mobile Code-Based Architecture for Information Indexing Searching, and Retrieval on the World Wide Web.*  
In Proceedings of the Annual Conference of the Internet Society INET'97. Kuala Lumpur, Malaysia. 24-27 junio, 1997.  
URL: [http://www.isoc.org/inet97/proceedings/A7/A7\\_2.htm](http://www.isoc.org/inet97/proceedings/A7/A7_2.htm)
- [Gravano 94] Gravano, L.; García-Molina, H.; Tomasic, A.  
*The Effectiveness of GLOSS for the Text Database Discovery Problem.*  
In Proceedings of the 1994 ACM SIGMOD Conference on the Management of Data. ACM Press. 1994.
- [Gravano 95] Gravano L., García-Molina, H.  
*Generalizing GLOSS to vector-space databases and broker hierarchies.*  
Technical Report STAN-CS-TN-95-12, Stanford University Department of Computer Science, 1995. To appear in VLDB 95.
- [Gravano 97] Gravano, L.; Chang, K.; García-Molina, H.; Lagoze, C.; Paepcke, A.  
*STARTS: Stanford Protocol Proposal for Internet Metadata Searching.*  
In Proceedings of ACM SIGMOD Conference, pp. 207-218, Tucson, Mayo 1997.  
URL: <http://www-db.stanford.edu/~gravano/starts.html>.
- [Hahn 96] Hahn, H.  
*The Internet Complete Reference, 2<sup>nd</sup> Edition.*  
ISBN 0-07-882138-X, Osborne McGraw-Hill. Enero 1996.
- [Hardy 96] Hardy, R.; Schwartz, M.; Wessels, D.  
*Harvest User's Manual.*  
Technical Report CU-CS-743-94, Department of Computer Science. University of Colorado-Boulder. Enero 1996.  
URL: <http://www.tardis.ed.ac.uk/harvest/docs/old-manual/>
- [Ianella 96] Ianella, R.  
*Internet Resource Discovery Issues.*  
Research Data Network CRC, University of Queensland, Australia. 1996.
- [Koch 96] Koch, T.; Ardö A.; Brümmer A.; Lundberg, S.  
*The building and maintenance of robot based internet search services: A review of current indexing and data collector methods.*  
Prepared to meet the requirements of Work Package 3 of EU Telematic for Research, DESIRE. Setiembre 1996.



- [Koster 95] Koster, M.  
*Robots in the Web: threat or treat?*.  
ConneXions, Vol. 9 Nro. 4, Abril 1995.
- [Kunze 97] Kunze, J.  
*A Citation Model for Resource Discovery and Retrieval*.  
To appear in D-Lib Magazine. 1997.
- [Lagoze 95a] Lagoze, C.; Davis, J.  
*Dienst: An Architecture for Distributed Document Libraries*.  
Communication of the ACM, Vol. 38 Nro. 4, Abril 1995.
- [Lagoze 95b] Lagoze, C.; Shaw, E.; et. al.  
*Dienst: Implementation Reference Manual*.  
Cornell computer Science Technical Report TR95-1514. 1995.
- [Lagoze 97] Lagoze, C.  
*From Static to Dynamic Surrogates*.  
D-Lib Magazine, Junio 1997.
- [Lagoze 98] Lagoze, C.; Fielding, D; Payette, S.  
*Making Global Digital Libraries Work: Collection Services, Connectivity Regions, and Collection Views*.  
In Proceedings of the 3<sup>rd</sup> ACM Conference on Digital Libraries, Pittsburgh, PA, 24-27 Junio, 1998.
- [Levy 95] Levy, D.; Marshall, C.  
*Going Digital: A look at Assumptions Underlying Digital Libraries*.  
Communication of the ACM Vol. 38 Nro. 4. Abril 1995.
- [Luotonen 94] Luotonen, A.; Altis, K.  
*World Wide Web Proxies*.  
First International Conference on the World Wide Web. Geneva. Mayo 1994. Y en Computer Networks and ISDN Systems 27(2), Elsevier Science BV 1994.
- [LY] Lycos HomePage  
URL: <http://www.lycos.com>
- [Marqués 95] Marqués, J. M.; Navarro, L.; Sarmiento, M.  
*From Small to Large Scale*.  
Comic Deliverable D.1.1. ISBN 0-901800-55-4. Universitat Politecnica de Catalunya (UPC).  
Revisado en Mayo 1995.
- [Marshall 98] Marshall, C.  
*Making Metadata: a study of metadata creation for mixed phisical-digital collections*.  
In Proceedings of the 3<sup>rd</sup> ACM Conference on Digital Libraries, Pittsburgh, PA, 24-27 Junio, 1998.
- [NCSTRL] *Networked Computer Science Technical Research Library*.  
URL: <http://www.ncstrl.org>
- [Neuman 93] Neuman, B. C.; Augart, S.  
*Prospero: A Base for Building Information Infraestructure*.  
In Proceedings of the Annual Conference of the Internet Society INET'93. 1993.

- [Neuman 94] Neuman, B. C.  
*Scale in Distributed Systems.*  
In Readings in Distributed Computing Systems. IEEE Computer Society Press, 1994.
- [Obraczka 93] Obraczka, K.; Danzing, P.; Li, S.  
*Internet resource discovery services.*  
IEEE Computer, 26(9), pp 8-22. 1993.
- [Obraczka 94] Obraczka, K.  
*Massively replicating services in wide area internetworks.*  
Ph.D. Thesis. University of Southern California. Diciembre 1994.
- [RFC 1036] Adams, R.; Horton, M.  
*Standard for Interchange of USENET Messages.*  
Request for Comments 1036. Marina del Rey, CA, Information Sciences Institute. 1987.
- [RFC 1913] Weider, C.; Fullton, J.; Spero, S.  
*Architecture of the Whois++ Index Service.*  
Request for Comments 1913. 1996.
- [RFC 1914] Weider, C.; Faltstrom, P.; Schoultz, R.  
*How to Interact with a Whois++ Mesh.*  
Request for Comments 1914. 1996.
- [Righetti 97] Righetti, C.; Navarro, L.; Lijding, M.  
*Object Distribution Networks for world-wide document circulation.*  
Third International Workshop on Groupware CRIWG'97. Madrid, España. 1-3 de Octubre 1997.
- [Rio 97] Rio, M.  
*A Distributed Weighted Centroid-based Indexing System.*  
In Proceedings of the 8<sup>th</sup> Joint European Networking Conference JENC8. Edinburgh International Conference Centre. Edinburgh, Scotland. 12-15 de Mayo 1997.
- [Rodríguez 95] Rodríguez, G.; Navarro, L.  
*The Aleph Computational Model.*  
Universitat Politècnica de Catalunya (UPC). Barcelona, España. 1995.
- [Rodríguez 96] Rodríguez, G.; Navarro, L.  
*AlephWeb: un "search engine" basado en la estructura de federación.*  
Universitat Politècnica de Catalunya (UPC). Barcelona, España. Mayo 1996.
- [Rodríguez 98] Rodríguez, G.  
*The Gloss-Aleph prototype.*  
Informe enviado por el autor en comunicaciones varias mantenidas por e-mail. Mayo 1998.
- [Sassi 98] Sassi, M.; Fabian, A.  
*EROS: Un servicio para los robots de descubrimiento de recursos en la World Wide Web.*  
Tesis de Licenciatura, Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires. 1998.

- [Schwartz 90] Schwartz, M. F.  
*A Scalable Non-Hierarchical Resource Discovery Mechanism Based on Probabilistic Protocol.*  
Technical Report (CU-OS-474-90). Departamento de Computación Científica, Universidad de Colorado, Boulder. Junio 1990.
- [Schwartz 92] Schwartz, M. F.; Emtage, A.  
*A Comparison of Internet Resource Discovery Approaches.*  
Computing Systems 5(4), 1992.
- [Schwartz 93] Schwartz, M. F.  
*Internet resource discovery at University of Colorado..*  
IEEE Computer, 27(9), 1993.
- [Sheldon 95a] Sheldon, M.  
*ContentRouting: A Scalable Architecture for Network-Based Information Discovery.*  
Ph.D, Department of Computer Science, MIT. Diciembre 1995.
- [Sheldon 95b] Sheldon, M.; Duda, A.; Weiss, R.; Gifford, D.  
*Discover: A Resource Discovery System based on Content Routing.*  
Computer Networks and ISDN Systems, Vol. 27 Nro. 6, pp. 953-972. Abril 1995.
- [Stanford-DL] *HomePage del grupo de investigación sobre bibliotecas digitales de la Stanford University.*  
URL: <http://www-diglib.stanford.edu/>
- [Viles 95] Viles, C.; French, J.  
*Dissemination of Collection Wide Information in a Distributed Information Retrieval System.*  
In Proc. Of SIGIR-95. Seattle, Washington. Julio 1995.
- [W3C 96] *W3C's Distributed Indexing/Searching Workshop.* Mayo 1996.  
URL: <http://www.w3.org/Search/9605-Indexing-Workshop/>
- [WC] *WebCrawler HomePage.*  
URL: <http://webcrawler.com>
- [Weibel 97] Weibel, S.; Ianella, R.  
*The 4<sup>th</sup> Dublin Core Metadata Workshop Report.*  
DC-4 National Library of Australia, Canberra. 3-5 Marzo 1997.
- [Weider 96] Weider, C.  
*The Future of Search on the Internet.*  
In Proceedings of the Annual Conference of the Internet Society INET'96. Montreal, Canada. 24-28 junio, 1996.  
URL: [http://www.isoc.org/isoc/whatis/conferences/inet/96/proceedings/a2/a2\\_1.htm](http://www.isoc.org/isoc/whatis/conferences/inet/96/proceedings/a2/a2_1.htm)
- [Weiss 96] Weiss, R.; Vélez, B.; Sheldon, M.; Namprempre, C.; Szilagyi, P.; Duda, A.; Gifford, D.  
*HyPursuit: A Hierarchical Network Search Engine that Exploit Content-Link Hypertext Clustering.*  
In Proceedings of Hypertext'96, Washington, DC, Marzo 1996.
- [Xu 98] Xu, J.; Cao, Y.; Lim, E.; Ng, W.  
*Database Selection Techniques for Routing Bibliographic Queries.*  
In Proceedings of the 3<sup>rd</sup> ACM Conference on Digital Libraries, Pittsburgh, PA, 24-27 Junio, 1998.
- [YH] *Yahoo! Corporation.*  
URL: <http://www.yahoo.com/>