

## Lección 6

### Definición de nombre y su uso en los Sistemas Distribuidos

Como se ha podido apreciar en los capítulos anteriores un Sistema Distribuido posee múltiples servidores de diferentes tecnologías en los cuales se encontrara los servicios a los que los clientes desean accesar. Este es una evolución muy importante desde los anteriores días en los sistemas basados en LAN en los cuales era sencillo encontrar el servicio al que se deseaba accesar, esto debido a que todos ellos se encontraban localizados en el servidor de aplicaciones como único servidor existente en este tipo de red.

Es evidente por lo tanto la complejidad en la ejecución de un servicio en un Sistema Distribuido ya que antes de poder invocar a dicho servicio primero debe ser localizado, lo que aparentemente no es tarea fácil no solo porque hay múltiples servidores sino también estos están colocados en múltiples servidores de diferentes tecnologías y construidos en diferentes lenguajes de programación.

Para organizar un modelo de localización de servicios en un entorno distribuido se desarrollaron varias implementaciones, las cuales se resumen de la siguiente manera:

- a) **Modelo Polling:** El cliente al solicitar la localización de un servicio obliga al Middleware a que haga una búsqueda de dicho servicio en cada servidor existente uno por uno, hasta su ubicación. Para ello, cada servidor cuenta con una cabecera en la que están definido la estructura de cada uno de los servicios existentes en dichos servidores.
- b) **Modelo Groupware:** En este modelo el Middleware al recibir la petición de ubicación de servicio por parte del cliente envía una petición de localización del servicio a todos los servidores existentes de manera simultánea, con el objetivo de que el servidor que tiene el servicio a ubicar responda con un mensaje de confirmación y envíe las características de dicho servicio.

En las implementaciones anteriormente señaladas la dificultad en la localización de servicio radica en que el cliente debe solicitar la búsqueda del servicio para que el Middleware conteste con una operación de ubicación que en la mayoría de los casos puede saturar el ancho de banda de la red, sobre todo si son múltiples clientes los que están solicitando servicios de manera simultánea. Debe notarse que en ambas implementaciones el servidor cumple una función pasiva, en la que solo actúa cuando es consultado sobre la existencia de un servicio.

En un tercer modelo de implementación quien inicia las operaciones para la futura identificación de servicios es el propio servidor ya que comienza las operaciones publicando las características de los servicios disponibles en un servidor destinado a guardar descripción de servicios y que por naturaleza será público. Dicho servidor, denominado **servidor de nombramiento**, estará manejado por el Middleware existente.

Posterior a la publicación de los servicios el cliente que quiera localizarlos solo le bastara buscar la descripción del servicio en este servidor de nombramiento para averiguar las características de esta operación a la que desea accesar y que, a continuación este cliente enviara un mensaje contenido solicitando ejecución de servicio en el servidor donde la operación resida, completándose de esta manera la comunicación entre procesos.

Esta implementación podrá verse claramente en el siguiente gráfico.

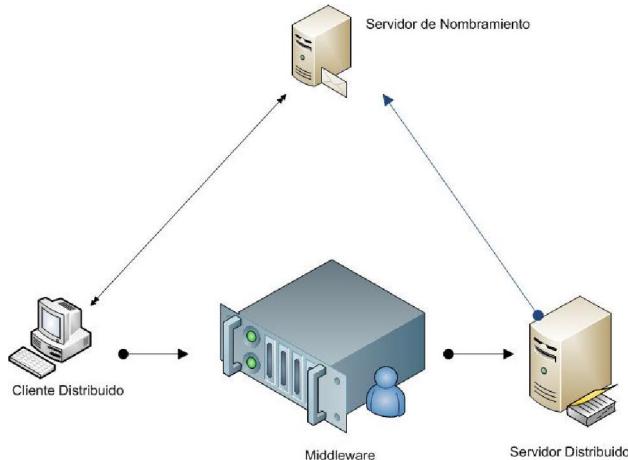


Fig.61 Organización de un Sistema Distribuido

Este tipo de implementación es el que se utiliza actualmente en todos los Sistemas Distribuidos, por su facilidad de implementación y por el control que ejerce el Middleware de las especificaciones de los servicios publicados. Pero al mismo tiempo introduce un problema en este caso adicional: ¿Cuál debe ser el nombre del servicio o recurso, que debe estar registrado en el servidor de nombramiento?, la importancia de la pregunta radica en el hecho de que el nombre asignado debería ser único para todo el Sistema Distribuido ya que de no serlo si un cliente lo usara para identificar a una operación podría obtener referencias de varias operaciones, por lo que no sabría cuál de estas es la correcta.

### 6.1 Determinación del nombre

Definir un nombre apropiado a un recurso va a permitir a usuarios y programas asignar una identificación adecuada a éste con el propósito de poder luego referenciarlo con dicho nombre, esto a su vez facilitara la compartición de objetos. Estos nombres solo se usan en el contexto del servicio que realiza la gestión de los objetos nombrados, sin embargo, pueden ser también usados en el caso de que los clientes quieran comunicarse con objetos compartidos. Los nombres también son usados para nombrar a entidades de un Sistema Distribuido que se encuentran fuera del ámbito de un único servicio por ejemplo computadoras, usuarios, etc.

Esta facilidad se usa normalmente en la tarea de la localización del recurso asociando el nombre con una dirección.

El nombre permite identificar de una manera clara a una entidad a accesar como por ejemplo:

- En comunicación  
jguerra91@gmail.com
- En compartición de recursos  
\\carpeta\\notas.txt  
http://www.universitario.com.pe/index.html

La forma como el nombre del recurso se convierte en una localización para ser usada en la red se puede apreciar en el gráfico a continuación

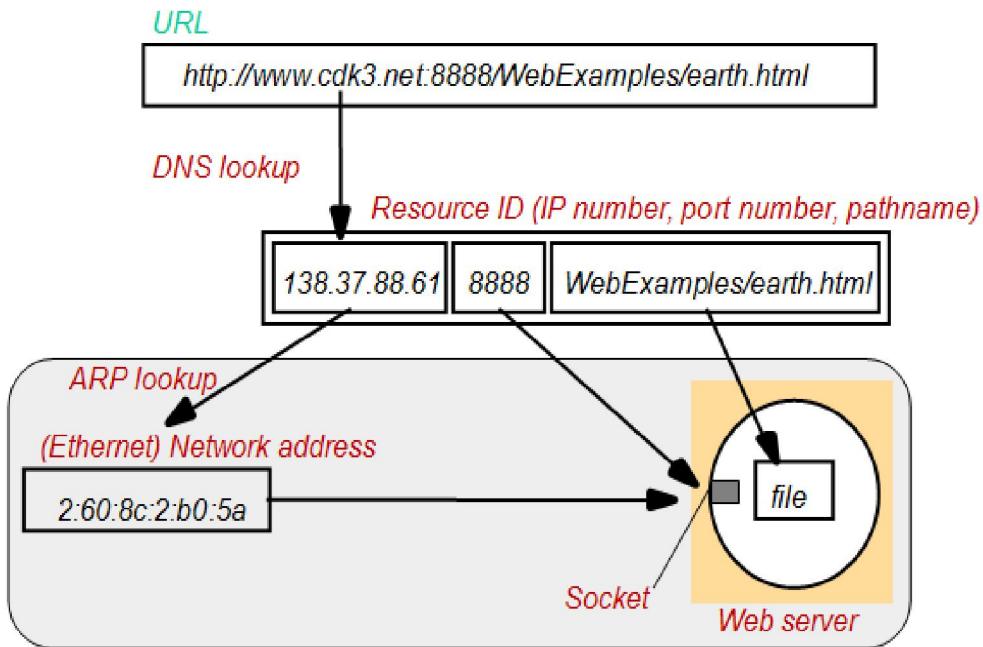


Fig.62 Conversión de una dirección URL en localización vía socket

### 6.1.1 Tipos de nombres

Los tipos de nombres que puede ser utilizado para la identificación de recursos pueden clasificarse en dos categorías:

- Segundo su uso
  - a) **Orientados al usuario**, se utilizan palabras cercanas al habla humana para identificar recursos ejemplo: stream, maquina, Alicia, servidor 1, etc.
  - b) **Orientados al sistema**, generalmente son direcciones que pueden ser cadenas de 32 o 64 bits, otro ejemplo es un código específico como PID y otros.
- Segundo la información
  - a) **Puros**, un identificador conformado por un patrón de bits el cual puede ser discriminado mediante un algoritmo de identificación. En el otro extremo de un nombre puro se sitúa la dirección de un objeto, la cual servirá para acceder a éste, aunque subsiste el problema de que dicho objeto puede cambiar de localización.
  - b) **No Puros**, poseen algún tipo de información que puede ser relevante para su identificación, por ejemplo la ubicación del objeto o recurso.
  - c) **Identificadores**, especial tipo de nombre que tiene las siguientes propiedades:
    - Un ID se refiere a lo más a una entidad
    - Cada entidad es referida por a lo más un ID
    - Un ID siempre referirá a la misma entidad, por lo que no se permite reutilización.

#### 6.1.1.1 Criterios claves sobre nombres

Resaltando la importancia de la definición de un nombre para localizar un recurso debe tomarse en cuenta algunos criterios claves alrededor de este identificador, como son:

- a) Identifica un recurso  
Dentro de esta idea debe definirse que el identificador debe ser único y deberá determinarse si tendrá la capacidad de describir el recurso.
- b) Nos permite localizar el recurso  
El identificador deberá establecer si permite dicha localización de manera directa o indirecta, y si además lo hace por sí mismo o con ayuda de otro tipo de identificador.
- c) El nombre es usado realmente como identificador  
Debe establecerse si el nombre tiene la propiedad biyectiva (nombre localiza un recurso, establece un nombre) y además si es persistente (es el mismo nombre en diferentes contextos).
- d) como es utilizado el nombre  
Debe definirse si el identificador permitirá localizar a un recurso o grupo de recursos, por otro lado, si es posible accesar al recurso a través del nombre o si debe construirse un nombre para encontrar un recurso.
- e) la ubicación de recurso es estático  
Definir si el recurso permanece en una única ubicación o cambia de posición, si cambia de ubicación definir si el nombre también debería cambiar, además determinar el caso en que el recurso es móvil (no tiene una ubicación fija).

#### **6.1.1.2 Determinación del identificador nombre**

Considerando los criterios claves reseñados en el punto anterior y para una mejor implementación del nombre asignado a un recurso se plantean cuatro formas de crear un identificador valido para los recursos a localizar en un Sistema Distribuido:

- identificador único global  
Este tipo de nombre utilizara el identificador Ethernet que poseen las tarjetas de red cuando se comunican a través de una implementación LAN o WAN. Este identificador resuelve de una manera sencilla la ubicación del computador donde reside el recurso, sin embargo, no devuelve ni la descripción ni la ubicación específica del servicio solicitado.
- Identificador único global asignado jerárquicamente  
Ejemplos de este identificador son el número de teléfono o la dirección IP con lo cual se identifica de manera precisa el equipo en el que reside el servicio e inclusive ayuda con su localización, pero no brinda información acerca de la descripción.
- Nombre asignado jerárquicamente  
Definido un contexto este identificador permite la ubicación de un recurso ayuda con su descripción y en algo permite su localización específica dentro del computador. Ejemplos de este tipo de identificador son DNS y URL.
- Nombre único global  
Este tipo de identificador utiliza puertos asignados en base al protocolo TCP/IP, ampliamente utilizado hoy en día por la web. Su deficiencia radica en la dificultad para agregar más identificadores debido a la gran cantidad de puertos ya asignados para otras aplicaciones.
- Espacio de nombres y sistemas de registro  
Está basado en el identificador URN y en el uso de un servicio de registro el que almacena la descripción de los recursos disponibles en el Sistema Distribuido. Esta implementación permite la identificación y la localización precisa del servicio a accesar así como ayuda con su descripción, sin embargo, es el más complicado de implementar de todos.

Dentro de esta clasificación de nombres se utiliza el concepto de **Identificador de Recursos Unificados** (URI), que es un tipo de identificador que permite determinar la localización de un objeto o recurso independientemente de donde se

encuentre y del entorno en el que se use, por esa razón la estructura de un URI será de la siguiente manera:

*Esquema: Parte jerárquica ? solicitud # fragmento*

En donde:

- Esquema, identifica el protocolo que se va a utilizar para el acceso al recurso. normalmente los esquemas actualmente usados son: cyd, data, dav, fax, ftp, file, gopher, http, https, imap, ldap, mailto, mid, news, nfs, nntp, pop, pres, sip, sips, snmp, tel, telnet, urn, wais, xmpp.
- Parte jerárquica, contiene la información del dominio o IP para poder acceder al servidor donde se encuentra el recurso además de la ubicación física dentro de dicho servidor por ejemplo:

//www.trincheranorte.com/principal/inicio.html

Existen caracteres de tipo reservados que se utiliza para delimitar la dirección u organización dentro del servidor en donde se ubica el objeto. Estos caracteres son / : ? # [ ] @ ! \$ & ' ( ) \* + , ; =

Por otro lado, existen algunos caracteres no reservados que pueden ser usados en la construcción de la parte jerárquica, estos son: letras, números y los caracteres - . \_ ~

- Solicitud, nos permite definir variables que pueden ser enviadas como parámetro de entrada al recurso a ubicar. Está separada de la ruta de localización mediante el signo de interrogación y acaba donde aparece el símbolo # (si lo hubiere). Un ejemplo seria:

/ubicarEscuela.jsp?código=2900

- Fragmento, permite identificar una subdirección dentro del objeto al que apunta la ruta original. Está delimitado por el símbolo # y llega hasta donde termina el URI. Por ejemplo:

/ingresantes.jsp#facsistemas

Por otro lado, la implementación de URI se puede realizar de dos formas:

- Localización de recurso uniforme (URL), utilizan direcciones que únicamente apuntan a recursos existentes en la web. Tiene como desventaja de que si el recurso a localizar se mueve o elimina el URL queda roto o podría redireccionarse a un recurso no deseado.
- Nombre uniforme de recurso (URN), es un tipo de URI el cual permite identificar recursos sin importar el lugar donde se encuentre ni el entorno en el que se trabaje. Si se utiliza en la web el administrador enlazará el URN con el URL de dicho recurso, teniendo esto como ventaja de que si el objeto cambia de dirección, se definiría un nuevo URL, manteniéndose constante el URN. Actualmente, la identificación de los recursos utilizados en un Sistema Distribuido utiliza URN como identificador.

### **6.1.2 Funciones que cumple un servicio de nombramiento**

El servidor de nombramiento dentro de su implementación cumplirá un conjunto de propiedades que determina su funcionamiento las cuales son las siguientes:

- Localización de objetos, deberá tener un mecanismo que permita la ubicación de un objeto al que el cliente desea accesar independientemente de si dicho objeto a cambiado de posición desde la última vez que fue accesado, cumpliéndose de esta forma la transparencia de migración.
- Generación de nombre para grupo de objetos, esta característica permite que un grupo de objetos o recursos que tienen alguna propiedad común pueda ser agrupados a través de un nombre similar o nombre grupal que permita determinar la existencia de dicho grupo, de esta manera se cumple la transparencia de replicación.

- Compartición de recursos, esta propiedad garantiza que diferentes clientes puedan solicitar la ubicación del mismo recurso de diferentes lugares utilizando inclusive diferentes nombres. Puede ser necesario para compartir objetos que han sido creados en diferentes dominios.

Dentro de este punto se debe indicar la importancia del concepto resolución de nombres, que significa la posibilidad de ubicación de un recurso solicitado a través de la utilización efectiva del nombre que posee éste. Para ello, el nombre utilizado estará conformado por un nombre inicial mas una dirección y otros atributos agregados que harán posible que el recurso a localizar sea único.

Bajo esta característica la resolución de nombre se presenta de dos formas:

- a) Convencional:  
A partir del nombre enviado se localiza el recurso solicitado y se obtienen sus atributos.
- b) Por atributos:  
A partir de uno o más atributos enviados se ubica los recursos que tienen estos atributos y a partir de esto se obtiene el resto de la definición de cada recurso localizado.

## 6.2 Implementación de la resolución de nombres

Utilizando los conceptos arriba mencionados de la resolución de nombres existen dos implementaciones definidas que son el servicio de nombres (naming services) y el servicio de directorio (directory services).

### 6.2.1 Servicio de nombres

Implementación fundamental dentro del esquema de localización de objetos debido a que permite localizar y enlazar a un recurso existente en el Sistema Distribuido dado su identificador (nombre) que deberá estar construido de tal manera que deberá ser único.

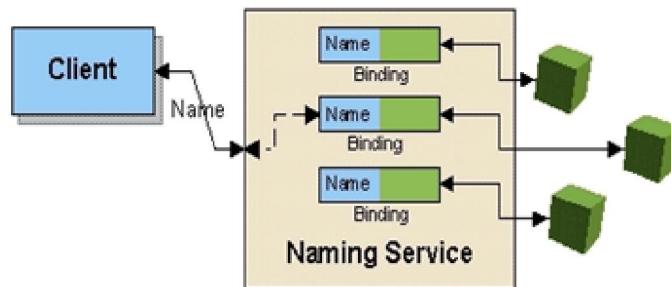


Fig.63 Servicio de nombres

Como se muestra en la figura el servicio de nombres recibe un identificador valido el que apuntará a un objeto cuya referencia ha sido publicada previamente en el naming service lo que permitirá la devolución de las características o atributos del recurso así localizado.

Ejemplos de este tipo de servicio son:

- DNS (Domain Name Services)
- RMI Registry
- CORBA Naming Services

### 6.2.2 Servicio de directorio

Implementación de servicio de nombramiento que tiene la finalidad de localizar un grupo de recursos dado uno o más atributos enviados como criterio de búsqueda.

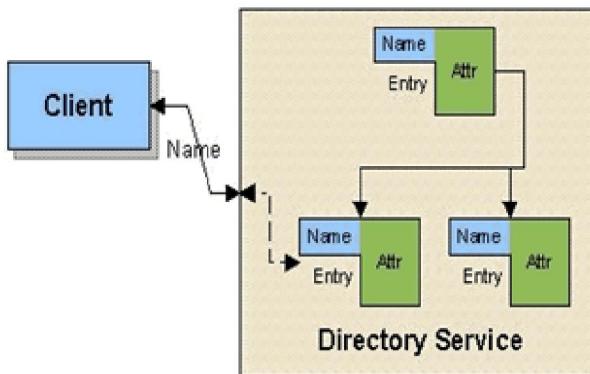


Fig.64 Servicio de Directorio

En la figura se puede observar que el cliente envía como identificador de recurso un atributo o grupo de atributos al servicio de directorio y éste en respuesta a la búsqueda realizada devolverá aquellos objetos o recursos que tienen en común los atributos considerados. Como se puede ver la idea no es localizar a un recurso específico sino a un grupo de ellos y a sus características. En este tipo de servicio se implementan mecanismos de búsquedas y filtros de acuerdo a los atributos enviados.

Ejemplos de servicio de directorio tenemos:

- Protocolo de acceso ligero a directorio (LDAP)
- Protocolo de acceso a configuraciones de aplicaciones (ACAP)

### 6.3 Espacio de nombres

El servicio de nombramiento, como se ha podido observar al inicio de esta lección es parte fundamental de la implementación de un Sistema Distribuido ya que permite a cualquier cliente poder ubicar de manera sencilla el servicio o recurso que desee utilizar. Sin embargo, hay una serie de requerimientos que deberán estar considerados dentro de la implementación práctica de este servicio y que son:

- Escalabilidad, permite un número arbitrario de nombres, asimismo un número arbitrario de organizaciones administrativas (grupos de nombres).
- Largo tiempo de vida, permite colocar variaciones en la organización de la colección de nombres a través del tiempo.
- Alta disponibilidad, debido a que otros sistemas dependen de este servicio
- Aislación de fallas, permite aislar fallas localizadas del servicio entero.
- Tolerancia a la desconfianza, indica la implementación de un sistema abierto grande que no puede tener ningún componente que no sea aceptado por todos los clientes del sistema.

Del punto de vista práctico se necesita un esquema que permita almacenar todos los nombres de los recursos disponibles en un Sistema Distribuido, por tal motivo, se define el concepto de **espacio de nombres**, como el conjunto de todos los nombres validos que pueden ser usados en un cierto contexto, es decir que el servicio de nombramiento puede intentar su búsqueda aunque esté desvinculado, o

sea dicho recurso a localizar no se encuentre en el servidor al que por defecto accesaria el cliente.

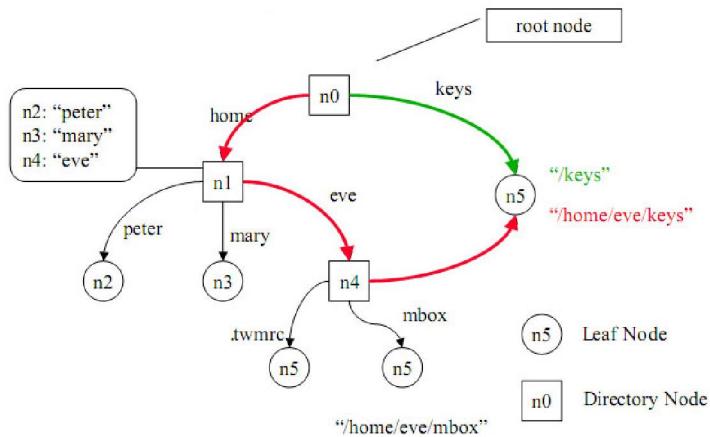


Fig.65 Espacio de nombres

El espacio de nombres requiere de una definición sintáctica, es decir, una forma de organizar los nombres que esta colección almacena, esta estructura interna puede ser:

- Un conjunto plano de identificadores numéricos o simbólicos, en cuyo caso suele definirse un tamaño máximo para el numero de nombres aceptados. Los nombres planos no contienen información para localizar el punto de acceso a su entidad o recurso.
- Una estructura jerárquica representando la posición de los recursos, como los sistemas de directorio en UNIX o los nombres de dominios en Internet. La principal característica de esta organización es que la búsqueda a realizar se hace en relación a un contexto específico por ejemplo en UNIX un archivo se busca al llamarlo desde un directorio concreto y no desde todo el sistema, debido a ello pueden existir iguales nombres pero en contextos distintos. Inclusive distintos contextos podían estar gestionándose por diferentes clientes.
- Estructura jerárquica representando estructura organizacional, por ejemplo la organización de recursos en Internet.

### 6.3.1 Características y detalles del espacio de nombres

Un espacio de nombres debe tener la capacidad de almacenar un número indeterminado de identificadores no importando el tipo de implementación a utilizar, además es conocida la utilización de **alias**, es decir una especie de nombre alternativo más simple que pueda sustituir al nombre original generalmente más complicado.

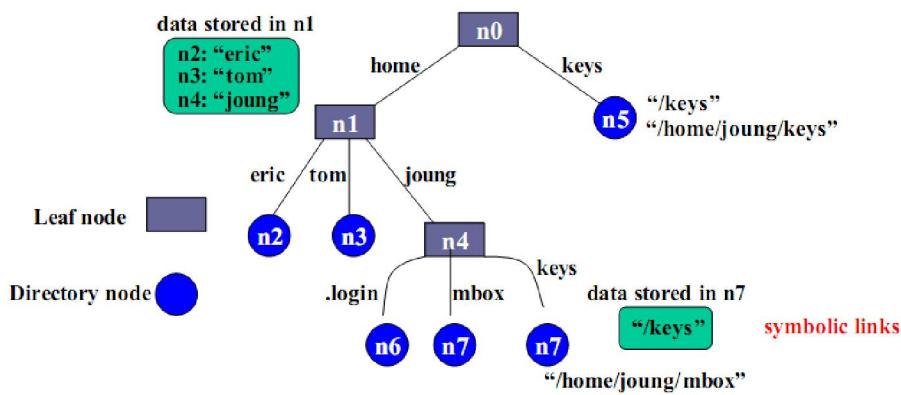


Fig.66 Resolución de nombres en UNIX

En la figura que muestra una resolución de nombres basada en UNIX el alias se le conoce con el nombre de enlace simbólicos (symbolic links) que diferencia a los enlaces reales (hard links).

Existe el **dominio de elección de nombres**, que representa a un espacio de nombres en el cual hay una única autoridad administrativa que puede asignar nombres dentro de él. Un dominio de esta característica puede gestionar los nombres que puede contener pero debe estar de acuerdo con el dominio que se encuentra encima de él. Por otra parte, la responsabilidad de un dominio de elección de nombres suele estar en paralelo con la responsabilidad de administrar y mantener actualizada la porción que le corresponde de la base de datos almacenada en un servidor de nombres autorizado y utilizado por el servicio de nombramiento.

En una implementación estándar los atributos que generalmente están almacenados en un espacio de nombres se registran en el servidor de nombramiento de la siguiente manera:

Tipo	Valor
Usuario	<login name, computadora donde se entregan los mail, nro. TE., etc.>
Servicio	<dirección, versión del servicio>
Computadora	<arquitectura, SO, dirección de red, propietario>
Grupo	<nombre1, nombre2,...>
Alias	<nombre>
Directorio	<nombreComponente1, nombreComponente2,...>

### 6.3.2. Implementación de un espacio de nombres

Para construir una arquitectura que contenga el espacio de nombres de un Sistema Distribuido existen dos alternativas:

- Espacio único, todos los nombres de los recursos identificados se colocara en un único contenedor al que accesarán todos los clientes del entorno distribuido. Este modelo no es el más apropiado por criterios de seguridad y de tratamiento de fallos.
- Combinación de distintos espacios de nombres, en este caso se definen múltiples contenedores de tipo parcial los cuales contendrán los nombres de los recursos descritos por los servidores del entorno distribuido. Para este modelo se debe establecer un mecanismo de resolución de nombres que

permita accesar de manera eficiente a cualquier identificador no importando en que contenedor se encuentre.

### 6.3.3 Resolución de nombres en el espacio de nombres

Es un proceso iterativo en el cual se presenta de manera repetida un nombre sobre los contextos de nombramiento. Primero se presenta sobre un contexto inicial y se repite según se genere otros contextos el objetivo de esto es la traducción de un nombre en sus atributos asociados, es decir, la localización de un recurso específico.

Se afirma que es un proceso iterativo debido a que el servicio de nombres retornara los atributos si la resolución puede ser realizada en el contexto del nombre, si no se puede el servicio envía el requerimiento a otro contexto y la operación nuevamente se ejecuta. La introducción de los alias puede complicar la tarea de resolución de nombres ya que en un contexto dicho alias puede no ser reconocido lo que podría dar lugar a un aborto de resolución después de un número de intentos definido.

- ❖ Definición  
Dada la ruta de un nombre, recuperar una información en el nodo referenciado por éste nombre, de la forma:  
$$N: <\text{atributo}_1, \text{atributo}_2, \dots, \text{atributo}_n>$$
  
Los nodos se resuelven uno tras otro, comenzando por el nodo raíz
- ❖ Mecanismo de cerradura  
Necesidad de seleccionar el nodo inicial desde donde va a iniciarse el proceso de resolución de nombre, en muchos casos este nodo inicial es implícito. En sistemas de archivos el directorio raíz es accesado y conocido independiente del espacio de nombres.

De acuerdo al punto donde comienza la resolución de nombres esta se puede clasificar en resolución absoluta (el contexto inicial corresponde con la raíz de un espacio de nombres parcial), o resolución relativa (el contexto inicial corresponde con un nuevo intermedio del espacio de nombres). Recuérdese que la implementación más utilizada para el espacio de nombres es el modelo jerárquico implementado a través de la estructura de datos llamada árbol, por otro lado, al aceptarse la implementación en base a combinación de espacio de nombres parciales estos serán organizados en dominios donde cada dominio será gestionado por un servidor.

#### 6.3.3.1 Resolución de nombres de tipo planos

Como se indicó párrafos anteriores los nombres planos no contienen información que permitan localizar de manera directa a la entidad o recurso que representan por lo que en el mecanismo de resolución debe establecerse una forma de poder localizar el servicio cuando sea proporcionado dicho identificador. Las implementaciones desarrolladas para resolución de nombres planos son las siguientes:

- Broadcasting y multicasting, utilizado en redes de tipo LAN o WLAN en las cuales hay conexión física entre cliente y servidor y en ella el cliente envía un mensaje con el identificador para todas las máquinas, pasado lo cual la máquina que posee el servicio devolverá respuesta.
- Puntero de avance, se utiliza generalmente para recursos móviles y se basa en la idea de que cuando un recurso se mueve de una posición A hacia la posición B deja en la posición inicial una referencia hacia la nueva localización en B. Cuando un cliente busca el servicio en la posición actual (que sabemos que es la posición inicial) una cadena de punteros se encargará de redireccionar su resolución a la nueva posición donde se encuentra el recurso.

- Ubicación basada en sitio, es ideal en entidades móviles que se encuentran en redes de gran escala. Se usa el concepto de **localización de sitio** que es un lugar que guarda en todo momento la localización actual de una entidad o recurso, desde el momento que éste ha sido creado, para ello el servidor móvil utiliza una dirección IP fija la que será invocada por el cliente y que redireccionará a un componente de tipo agente que ubicará la posición real del recurso.
- Tabla hash distribuida, en esta implementación se define un tipo de tabla hash que por defecto no contiene información directa de la ubicación del recurso, en vez de ello se organiza un grafo circular en el que los nodos pertenecientes guardan los identificadores que el espacio de nombres posee y que están organizados por proximidad, es decir, dos nodos que identifican recursos contiguos tendrán identificadores cercanos entre sí.
- Aproximación jerárquica, donde la red se dividirá en una colección de dominios a partir de un nodo inicial que es el nodo raíz y hace recorridos similar al de una estructura jerárquica.

#### 6.3.4 Navegación en espacio de nombres

Dado que el servidor de nombramiento se subdivide en varios servidores un servidor de nombre local no siempre podrá responder a una petición sin que reciba la ayuda de otros servidores de nombres, por lo que se define un proceso de búsqueda en múltiples servidores que se conoce con el nombre de navegación. Debe considerarse además de que la navegación concluirá cuando el nombre sea resuelto o se encuentre que el nombre es unBound. Existen cuatro tipos de navegación: navegación iterativa, navegación multicast, navegación no recursiva y navegación recursiva.

##### 6.3.4.1 Navegación iterativa

El cliente enviará la petición a servidores de nombres distintos hasta que uno pueda darle una respuesta determinada, empezando por el servidor de nombre local.

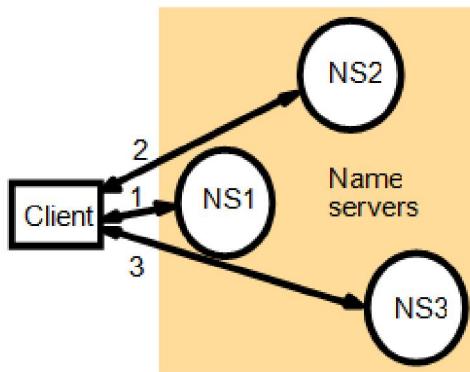


Fig.67 Navegación iterativa

En la figura un cliente contacta de forma iterativa con los servidores de nombres NS1-NS3 para resolver un nombre.

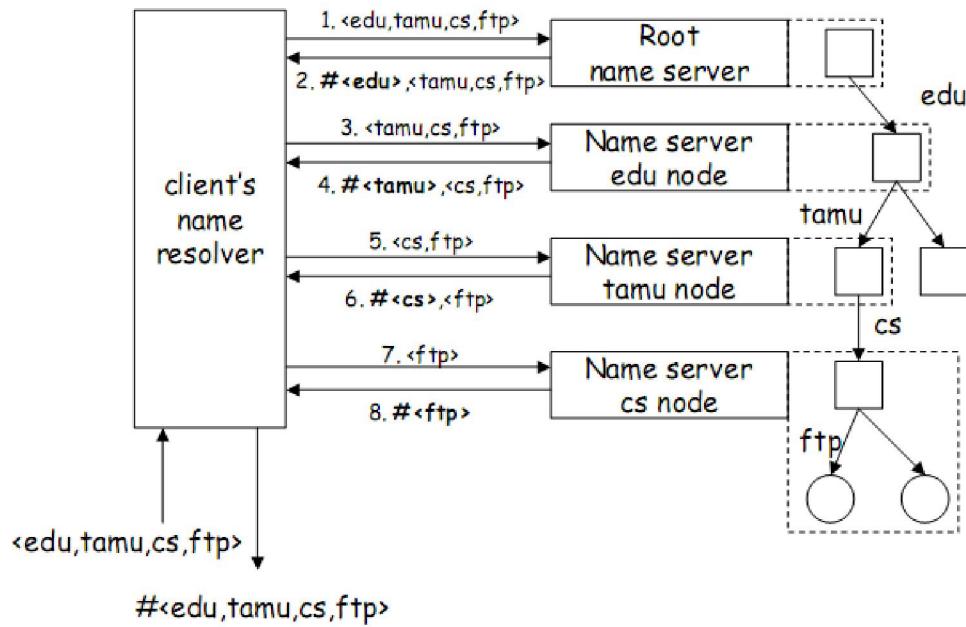


Fig.68 Secuencia de operaciones, navegación iterativa

En el presente grafico se muestra un ejemplo práctico de navegación iterativa el cual después de ocho operaciones de búsqueda localiza el recurso (#<edu,tamu,cs,ftp>). Este tipo de navegación es usado por DNS y NFS.

#### 6.3.4.2 Navegación multicast

Es una variación de la navegación iterativa, la diferencia radica en que el cliente contacta de manera simultánea con todos los servidores de nombres disponibles para la resolución. El servidor que tiene el recurso identificado por el nombre responde enviando los atributos. La dificultad de esta navegación radica en que un nombre podría ser valido a pesar de que ningún servidor conteste, esto podría suceder si el servidor no estuviera disponible u ocupado. Este tipo de navegación se utiliza también en los servicios de descubrimiento.

#### 6.3.4.3 Navegación no recursiva

Es controlada por el servidor de nombres cuando el servidor de nombre local no resuelve el nombre recibido contacta a otros servidores para resolución ya sea por multicast o iterativamente usando contacto directo

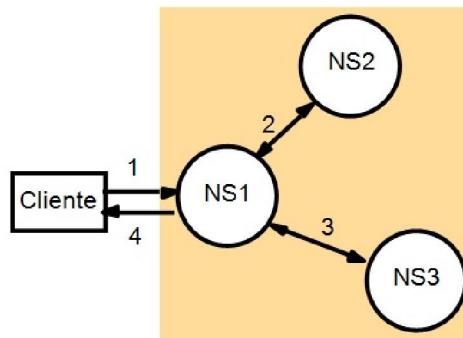


Fig.69 Navegación no recursiva

En la figura un cliente contacta con el servidor de nombres NS1 quien en otro momento se comunica con otros servidores de nombres representando al cliente.

#### 6.3.4.4 Navegación recursiva

Igual que la anterior es controlada por el servidor de nombres. Al recibir una petición de resolución de nombres, si el servidor no puede resolverla contacta a otro servidor superior responsable por un prefijo mucho más extenso del espacio de nombres (nombre más largo), posteriormente se utiliza este mecanismo hasta resolver dicho nombre.

Este tipo de navegación puede usarse si el cliente o servidor de bajo nivel no tienen derecho a accesar directamente a recursos de servidores de alto nivel.

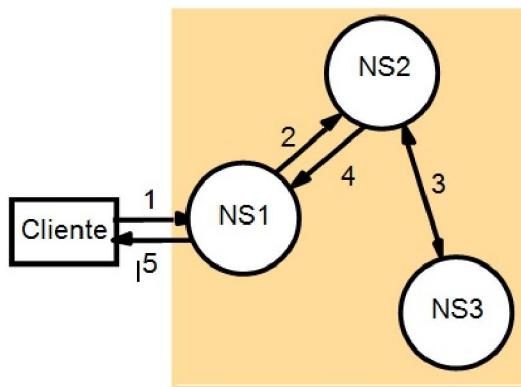


Fig.70 Navegación recursiva

En la figura se muestra a un cliente que contacta con el servidor de nombres NS1 el cual al no resolver el nombre se comunica de manera recursiva con otros servidores de nombres representando al cliente.

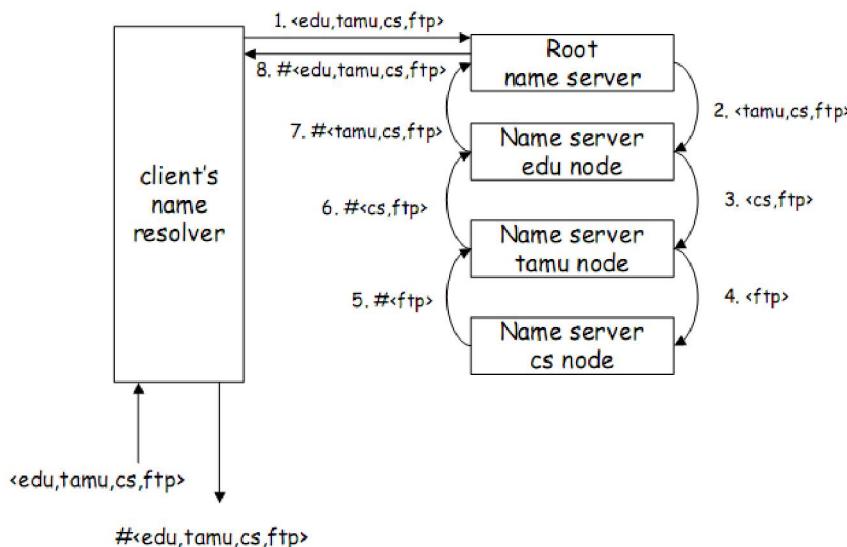


Fig.71 Secuencia de operaciones, navegación recursiva

En el presente gráfico se muestra un ejemplo de navegación recursiva para la ubicación de un recurso (#<edu,tamu,cs,ftp>).

A continuación se adjunta una breve comparación entre la navegación iterativa vs. navegación recursiva

Iterativa	Recursiva
Maneja estado en todo momento	Los servidores de alto nivel necesitan mantener estado para resoluciones
	Uso de cachés efectivo
	Costo reducido de comunicación

Tabla 6 Comparación entre navegación iterativa y recursiva

#### 6.3.4.5 Implementación por capas de espacio de nombres

Haciendo el seguimiento de la implementación del espacio de nombres en una estructura jerárquica (tipo árbol) se definen tres tipos de capas:

- **Capa global**  
Corresponde a los nodos de más alto nivel, dichos nodos se encuentran lógicamente cerca de la raíz. Los nodos de esta capa son estables, es decir raramente se modifica y representan organizaciones o grupos de organizaciones.
- **Capa administracional**  
Son nodos de directorio que son administrados en forma conjunta dentro de una única organización. Dichos nodos representan grupos de entidades o recursos que pertenecen a la misma organización, además, son nodos relativamente estables.
- **Nivel gerencial**  
Son nodos que cambian regularmente y representan recursos específicos como personas, archivos, métodos, máquinas, etc.

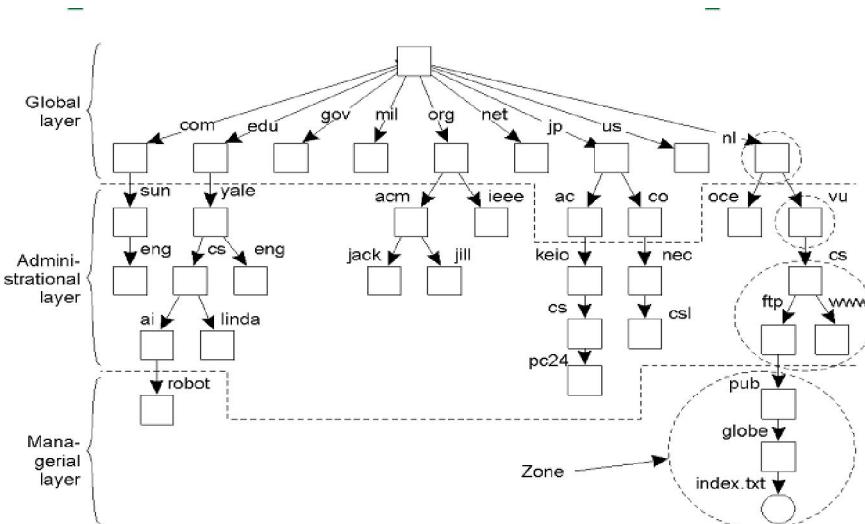


Fig.72 Organización de espacio de nombres por capas

En el gráfico se observa un ejemplo de espacio de nombre DNS en el que se incluye archivos que pueden ser accesados en internet, organizado en las tres capas mencionadas.

## 6.4 Servicio de directorio y descubrimiento

Este tipo de servicio de nombramiento tiene la finalidad de implementar la localización de un recurso o varios no a través de su nombre que generalmente no se conoce, sino mediante de uno o varios de sus atributos. El servicio de directorio almacena enlaces entre nombres y atributos y busca entradas que emparejan resoluciones basadas en atributos.

Este servicio permite devolver atributos de cualquier registro que coincida con los valores de entrada, así también permite que el cliente pueda determinar qué grupo de atributos desea recuperar. Por otra parte, debe recordarse que los atributos no permiten conocer la estructura de las organizaciones, como lo harían los nombres que se encuentran organizados mediante criterios asociados a una cierta organización, sin embargo, su implementación es mucho más compleja.

La base de este servicio es el directorio, que puede definirse como un integrado que representa al espacio de nombres sea global o parcial. El directorio no tiene la intención de ser un sistema de base de datos de propósitos generales, es utilizado mayormente en operaciones de consultas mas que de actualizaciones y permite mantener por un tiempo antiguas y nuevas versiones de la misma información. Una propiedad importante es que los resultados de las consultas a un directorio no dependerá ni de la identidad o localización del cliente, excepto para actualizaciones que requiera verificación o derecho de acceso.

Existen dos implementaciones de servicio de directorio, que son los servicios de descubrimiento y JINI.

### 6.4.1 Servicio de descubrimiento

Es un tipo de servicio de directorio que registra los servicios proporcionados en un entorno de red en la que un servidor previamente ha publicado sus recursos disponibles y que se conecta de manera espontánea por lo que no requiere intervención humana para esta acción. Un ejemplo de servicio de directorio es el siguiente gráfico.

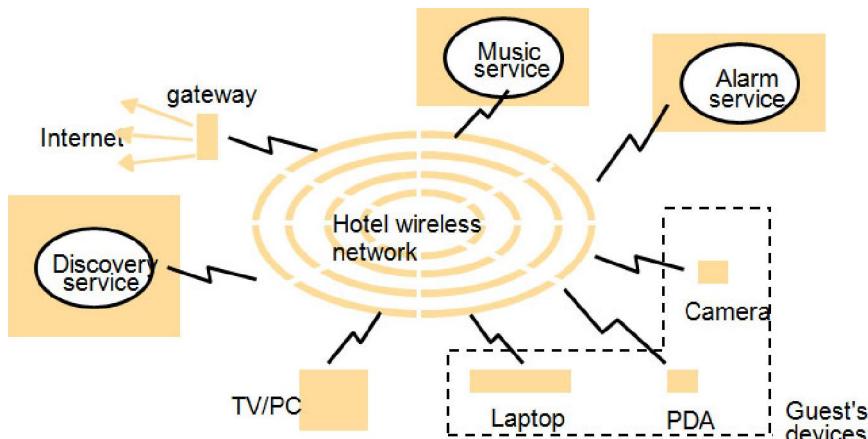


Fig.73 Servicio de descubrimiento

En la figura mostrada se observa una implementación de una red inalámbrica para un hotel en la que varios de los servicios se encuentran registrados en el servicio de descubrimiento, el cual se comporta como una base de datos de servicio con opciones de localización basadas en descripción de servicios como tipo, localización y otros criterios, por ejemplo encontrar un servicio de impresión en el citado hotel.

#### 6.4.1.1 Universal Description Discovery and Integration (UDDI)

El estándar UDDI es un tipo de servicio de descubrimiento que es utilizado por los servicios web para que puedan describirse a sí mismos y a las operaciones que proporcionan. Esta descripción puede registrarse y publicarse en un registro UDDI. Una vez registrados los servicios web pueden ser buscados, consultado o descubiertos por otro sistema usando mensajería apropiada (SOAP). Una vez descubierto los servicios web con quien puedan asociarse, se utiliza este mecanismo para integrar los recursos localizados con el Middleware y a través de él proveer a los clientes que lo requieren.



Fig.74 UDDI

UDDI permite localizar un web service y su descripción mediante búsquedas por negocio o por tipo de servicio.

La organización de los registros que representan a los servicios web en UDDI tiene los siguientes estándares:

- Páginas blancas, contiene dirección e información de contacto del proveedor del servicio.
- Páginas amarillas, permite buscar servicios por categorías y clasificaciones.
- Páginas verdes, contiene suficiente información técnica sobre los servicios ofrecidos para permitir su invocación.

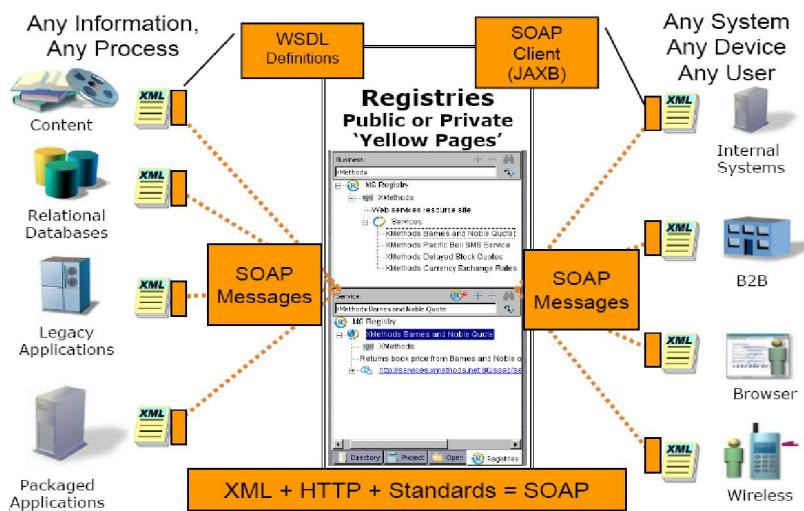


Fig.75 Uso de UDDI en comunicación entre clientes y servidores

En el gráfico mostrado se observa la forma como se utiliza UDDI en un entorno de tipo web service en la que los servidores de la derecha han registrado sus servicios y que posteriormente los clientes de la izquierda los ubicaran, pidiéndole al UDDI las características de las operaciones referenciadas.

#### 6.4.1.2 JINI

Jini es una API desarrollada por Sun Microsystems. El objetivo es convertir la red en un sistema flexible y fácil de administrar en el cual se puedan encontrar rápidamente los recursos disponibles tanto por clientes humanos como computacionales. Un sistema Jini consiste en un sistema distribuido basado en la idea de grupos federativo de usuarios y de recursos requeridos por otros usuarios. Los recursos pueden ser implementados tanto por dispositivos hardware y software. Las partes de un sistema Jini son:

- Un conjunto de componentes que proporcionan una infraestructura de servicios federativos en un Sistema Distribuido.
  - Un modelo de programación que soporta y estimula la producción fiable de servicios distribuidos.
  - Los servicios que pueden ser parte de un sistema federativo Jini y los cuales ofrecen funcionalidad a cualquiera de los miembros de la federación.

Jini supone que la infraestructura de red sobre la que se monta tiene el ancho de banda y es lo suficientemente fiable para funcionar, por lo que no aporta mecanismo para mejorar estos dos puntos. También se asume que los dispositivos Jini tienen capacidad de procesamiento y memoria suficientes.

Para saber los servicios disponibles se utiliza el servicio de búsqueda (lookup service). Este mapea las interfaces que indican la funcionalidad de un servicio con el conjunto de objetos que implementan dicho servicio. El servicio de búsqueda se organiza de forma jerárquica. Cuando se quiere añadir un servicio a la tabla se utiliza el protocolo discovery y el protocolo join. El primero se encarga de buscar el lookup service y el segundo de añadir el servicio. Cuando se quiere utilizar el servicio se busca en la tabla de servicios (lookup service) si existe. En caso de encontrarlo el cliente se descarga el código de control de ese servicio, que puede ir desde una interfaz hasta la implementación completa del servicio.

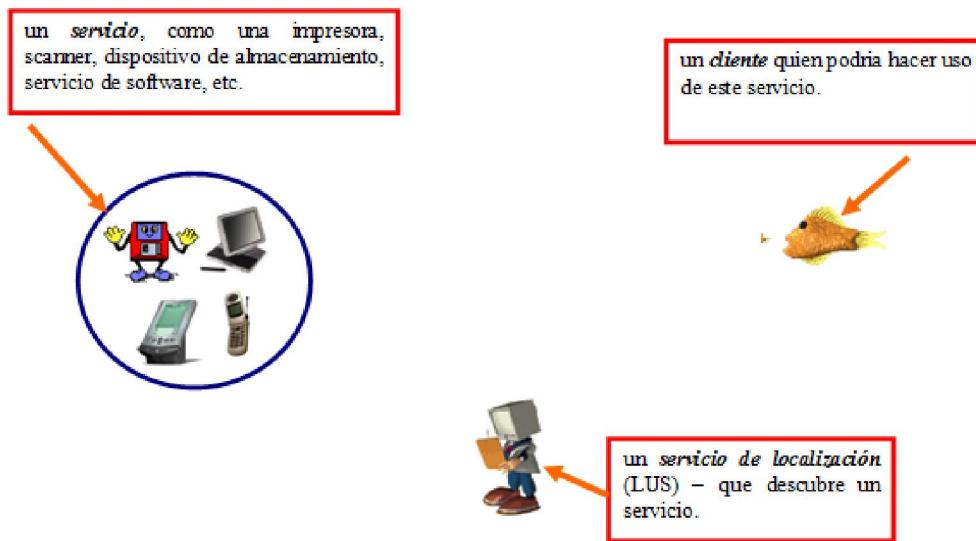


Fig.76 Representación del funcionamiento de JINI

JINI define una infraestructura de ejecución que provee mecanismos que permiten añadir, eliminar, localizar y accesar servicios. Participan en ella tres roles importantes: el servicio, el cliente y el servicio de localización, los cuales son mostrados en el gráfico anterior.

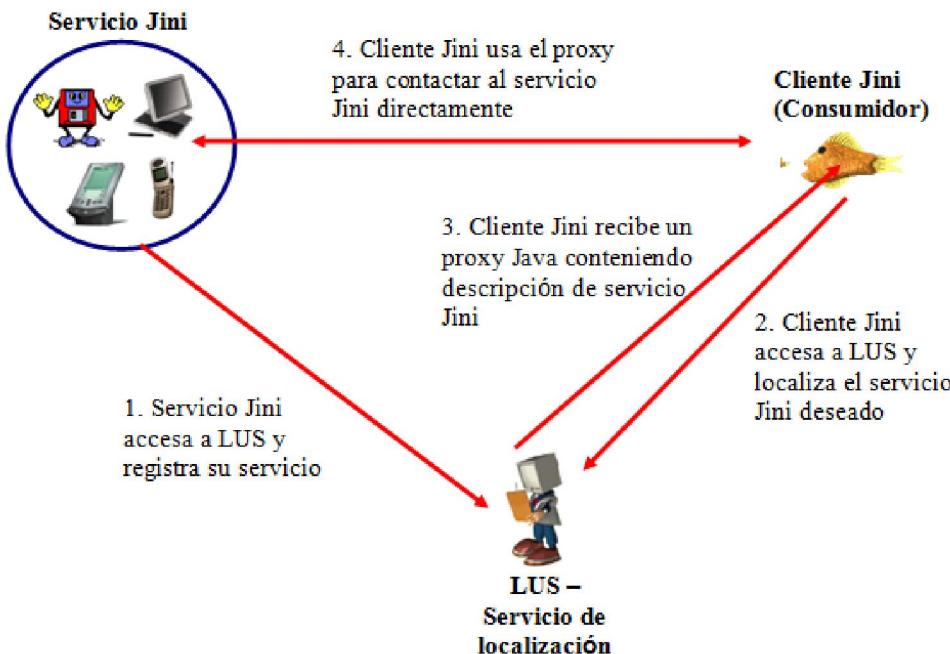


Fig.77 Interacción entre los componentes JINI

Como se aprecia en este gráfico el procedimiento de uso de los servicios JINI es similar al de cualquier servicio de directorio, utilizando como mecanismo de transporte de mensaje un Proxy basado en Java, conteniendo la información del servicio localizado.

## 6.5 Implementaciones existentes de servicio de nombramiento

Dados todos los conceptos basados en servicio de nombramiento como servidor de nombres y el servidor de directorio, se presentan a continuación las más importantes implementaciones desarrolladas por parte de los fabricantes acerca de localización de servicios. Estas implementaciones son:

- X.500
- DNS
- LDAP
- Active Directory

### 6.5.1 X.500

La implementación basada en OSI X.500 fue definida el año 1988 y extendida en 1993, 1997 y 2001, es considerada la implementación estándar de los servicios de nombramiento, inclusive implementaciones posteriores de otros fabricantes utilizan como base esta implementación.

Tiene implementación tanto para servicio de directorio como para servicio de nombre y la información acerca de los servicios registrados está organizada de

manera jerárquica. Además, utiliza el concepto de espacio de nombres parciales, maneja operaciones de replicación y uso de caché, así como permite el uso de alias para acceso rápido a recursos.

Existen varios usos para este servicio como por ejemplo operaciones análogas a consulta de directorios telefónicos, acceder a datos personales de usuarios. La solicitudes de servicios pueden venir de usuarios específicos o de procesos, de tal manera que X.500 puede ser usado para mostrar información sobre sí mismo, así como otros usuarios pueden buscar esos mismos datos en el servicio de directorio.

Se puede considerar a X.500 como un servicio para acceso de información de entidades del mundo real, así como también para el acceso a información sobre servicios o dispositivos de hardware y software. Si bien es cierto, esta implementado a nivel de la capa de aplicación OSI, su construcción no depende de las otras capas OSI por lo que se entiende que es de propósito general. X.500 se organiza en tres servicios que son: modelo de directorio, modelo de información y modelo de seguridad.

La descripción de servicios se organiza en una estructura de árbol. Cada nodo de dicho árbol tiene un nombre, pero además, de él puede almacenarse una gran cantidad de atributos por cada nodo, lo que permite que el acceso no se realice solamente por nombre sino que se puede usar cualquier combinación de atributos.

El árbol de nombres de X.500 se denomina árbol de información de directorio (DIT); así como, la estructura completa con la descripción de todos los servicios y los atributos asociados a todos los nodos disponibles se denomina base de información de directorio (DIB). La información que el DIB almacena no depende si el sistema utiliza arquitectura distribuida o centralizada, además, no depende de la jerarquía.

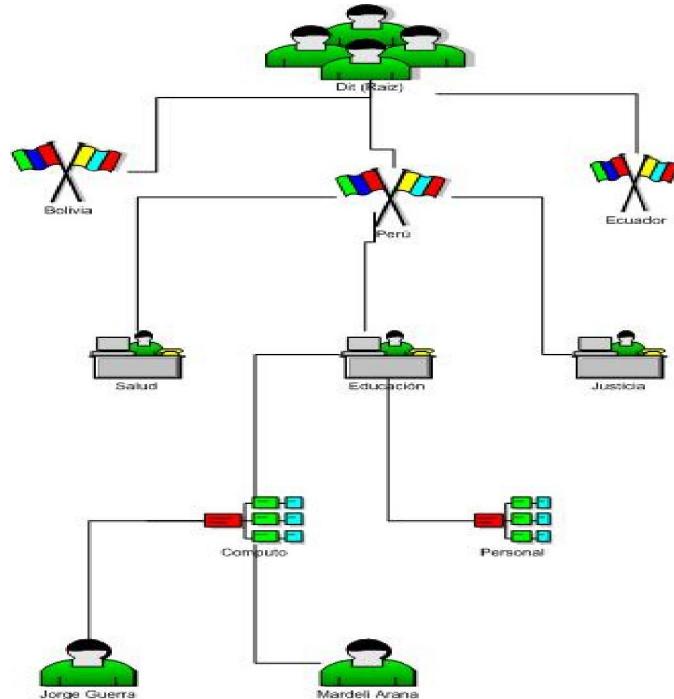


Fig.78 Representación de DIT

El gráfico muestra un ejemplo de estructura X.500 donde se establece la localización de un recurso dado un DIT, un país (Perú), una organización (Educación), una unidad organizacional (Cómputo) y finalmente el recurso (Jorge Guerra).

Profundizando en el modelo, cada una de las entradas o recursos en el árbol mencionado, estará organizada en atributos, en el cual cada atributo tendrá un grupo de valores, como se muestra a continuación.

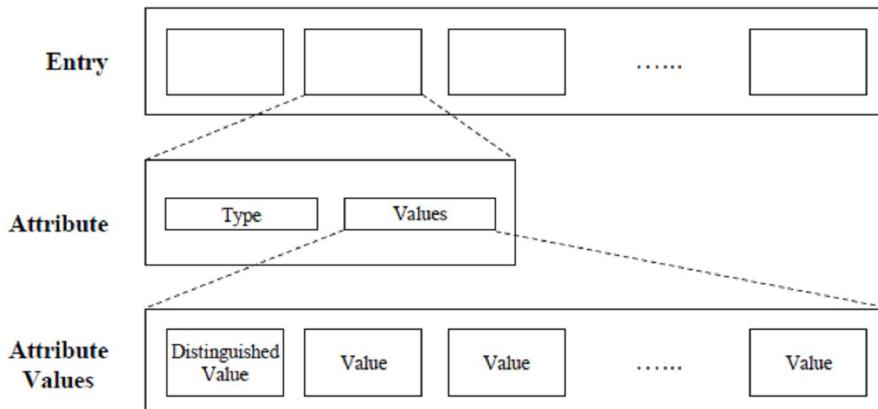


Fig.79 Organización de entidades y atributos

Estableciendo en forma gráfica la relación entre el DIT con las entradas y sus atributos obtenemos el siguiente gráfico:

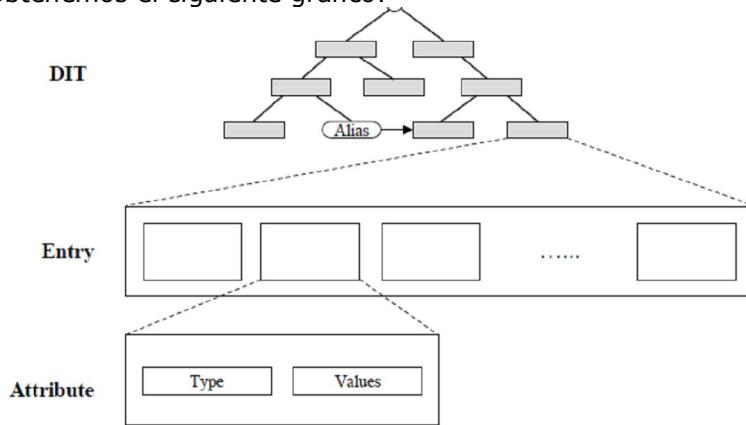


Fig.80 Relación entre DIT entidad y atributo

Para la descripción formal de la localización del recurso o servicio dentro del DIB la resolución de nombre sería de la siguiente manera (usando como referencia la figura 78):

nombre c=pe,o=educacion,ou=computo,cn=jorge guerra,s=lima,sa="av.caminos del inca 123,l=surco"

Donde se define los siguientes atributos

- País (c)

- Nombre de la organización (o)
- Nombre de la unidad organizacional (ou)
- Nombre común (cn)
- Nombre de localidad (l)
- Dirección de la calle (sa)
- Nombre del estado o provincia (s)

En el ejemplo del nombre utilizado, cuando permite la localización de un objeto específico y de los atributos que posee, dicho nombre se le llamará nombre distintivo (**Distinguished Name: DN**). En otro caso, cuando el nombre omite algun atributo clave y no devuelve un único objeto, se le denomina nombre relativo (**Relative Distinguished Name: RDN**).

Una observación importante: El RDN se utiliza generalmente cuando se desea localizar un recurso dentro de un dominio específico y no dentro del espacio de nombres completo.

La localización de un recurso usando DN dentro del DIB nos permite obtener una entrada que devolverá un grupo de atributos especificado para dicho objeto. Como ejemplo de lo afirmado para el nombre distinguido señalado líneas arriba, obtendremos la siguiente entrada de DIB:

*inf*  
Jorge Guerra, Cómputo, Educación, Perú.

<i>Nombre comun</i>	<i>uid</i>
Jorge L.Guerra	9753509097
Jorge Guerra	<i>mail</i>
J.Guerra	jguerra91@gmail.com
<i>apellido</i>	jguerra42@hotmail.com
Guerra	<i>Numero Oficina</i>
<i>Numero telefono</i>	Z42
+44 986 33 4604	<i>Cargo</i>
	Docente

#### 6.5.1.1 Protocolos de X.500

Para la implementación del servicio X.500 se desarrollaron diferentes protocolos destinados a realizar las operaciones, tanto de servidor de nombres como servidor de directorio, dichos protocolos son los siguientes:

- DUA, son las siglas de Agente Usuario de Directorio, es un proceso que representa a un usuario que solicita la localización de un servicio a X.500.
- DSA, son las siglas de Agente Servidor de Directorio, representa al proceso de servidor que interactúa con el cliente (DUA) para la localización del servicio solicitado.
- DAP, son las siglas de Protocolo de Acceso a Directorio, es el medio estándar a través del cual interactúan el cliente (DUA) y el servidor (DSA) para la localización de un servicio.
- DSP, son las siglas de Protocolo de Sistema de Directorio, controla la interacción entre dos o mas servidores (DSA) y entre un servidor y un cliente solicitante (DUA). Esto se hace para que, un usuario final pueda accesar a la

información del directorio, sin necesidad de conocer la localización exacta del servicio que solicita.

- DISP, son las siglas de Protocolo de Replicación de Información de Directorio, este protocolo de carácter administrativo realiza tareas de replicación del contenido del directorio, con el fin de optimizar el uso de la red y sus recursos. Dicha replicación se puede hacer de manera programada o a pedido y controla todas las actualizaciones del servicio de directorio original, para replicarlas en la copia en el momento adecuado.

X.500 puede establecer localización de información, tanto por un mecanismo de encadenamiento o por multitransmisión (multicasting), como se observa en el siguiente gráfico.

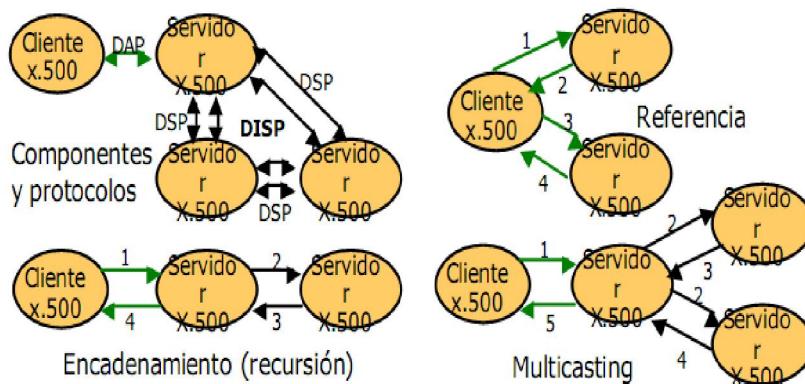


Fig.81 Protocolos usados en X.500

En resumen un DIB estará repartido entre diferentes DSAs y una forma de expresar la relación entre el DUA y el DSA, para localizar servicios, se muestra a continuación.

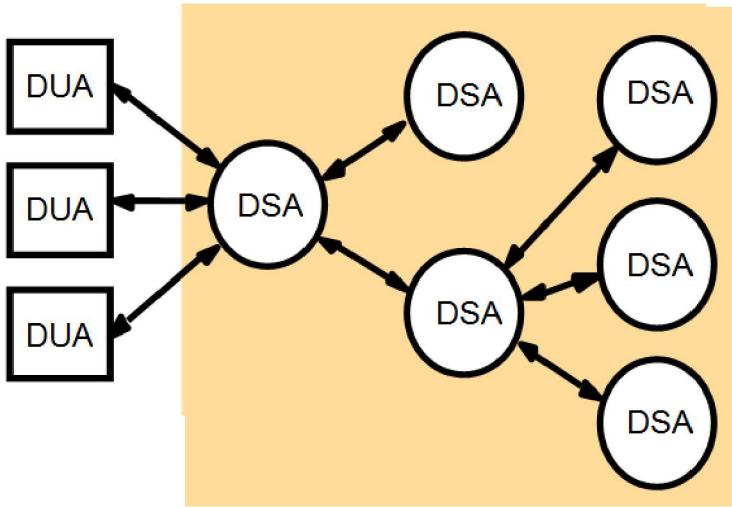


Fig.82 Ejemplo de interacción DUA y DSA

Para una mejor comprensión de la relación entre los protocolos X.500, el modelo funcional permite tres tipos de interacción: referencias, encadenamiento y multitransmisión.

- Referencias

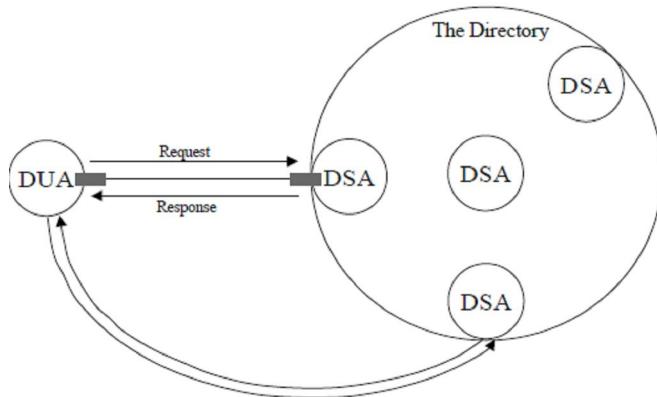


Fig.83 Interacción por referencia

- Encadenamiento

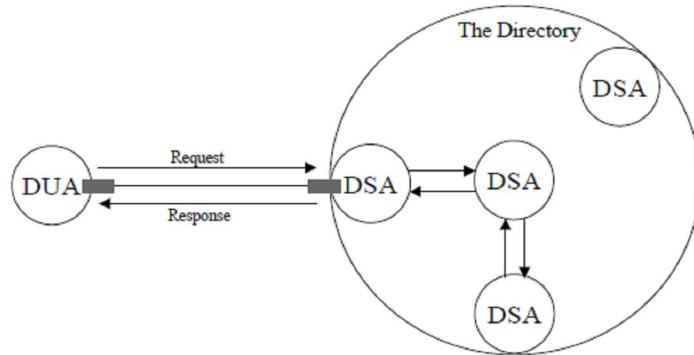


Fig.84 Interacción por encadenamiento

- Multitransmisión

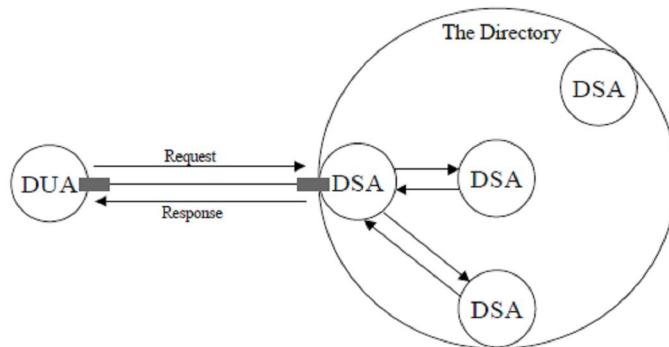


Fig.85 Interacción por multitransmisión (Multicast)

### 6.5.1.2 Sistema de nombres de dominio (DNS)

Es una implementación de servicio de nombre. Su repositorio principal se utiliza a lo largo de Internet y los recursos que nombra DNS son en primer lugar computadoras, utilizando para ello direcciones IP, sin embargo, este modelo podría nombrar cualquier objeto. Cualquier nombre podrá ser resuelto por cualquier cliente a lo largo de Internet mediante la partición jerárquica que el servicio de nombre posee, gracias a la replicación de los datos de nombres y al mecanismo de caché.

DNS pertenece a la capa de aplicación y por ello la resolución de un nombre involucra servidores de nombres, routers y host que se enlazan de manera automática.

Dentro de DNS los servidores de nombres pertenecen a cada compañía u organización, de tal manera que, una consulta de un host del DNS irá primero a este servidor, a su vez, estos servidores se contactarán con los denominados **servidores de nombre raíz**, dicho acceso se realizará en caso no se conozca el mapeo del nombre. Los servidores de nombre raíz realizarán el mapeo de forma global y retornarán la localización del recurso al servidor de nombre local. Actualmente en el mundo existe aproximadamente una docena de servidores de nombres raíz.

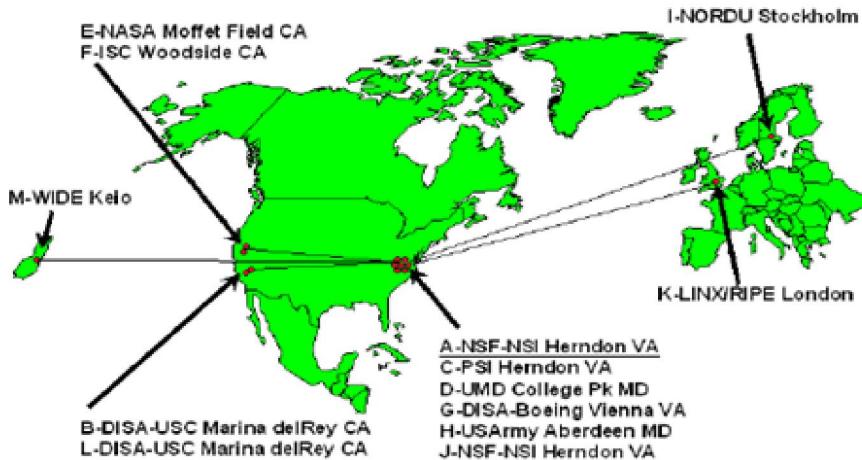


Fig.86 Servidores de nombre raíz

El uso de DNS en Internet es crítico hoy en día, de modo que, si DNS fallara en algún momento, los sitios web no serían localizados ni los correos electrónicos enviados.

#### 6.5.1.2.1 Estructura de un nombre DNS

Para entender cómo se implementa el nombre distinguido en DNS debemos entender las características de implementación de este servicio.

Como ya se dijo el DNS, posee un repositorio de tipo distribuido el que es mantenido localmente y es disponible en forma global, utiliza replicación y operaciones de caché para dar robustez y performance.

Un sub árbol en el sistema DNS es llamado dominio y un nombre de ruta hacia la raíz es llamado nombre de dominio.

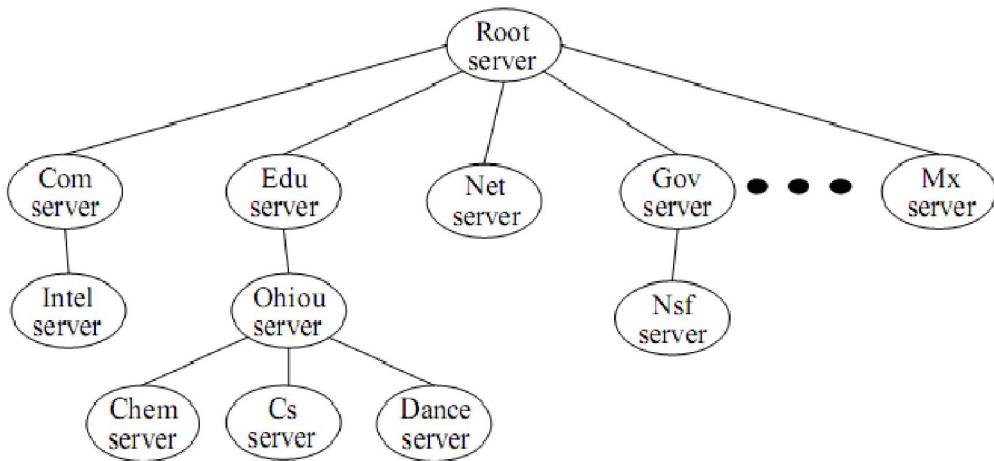


Fig.87 Organización de DNS

Toda zona tiene por lo menos dos servidores de nombres y cada servidor mantiene una tabla de correspondencia nombre de dominio-IP generando una colección de registros del tipo <nombre,valor, tipo, clase, duracion de vida>. Dentro del contenido de las tablas se define:

- Tipo, indica como el campo valor debe interpretarse, valores:
  - Tipo=A, corresponde a una dirección IP
  - Tipo=NS, corresponde a un nombre de dominio de un servidor de nombres
  - Tipo=CNAME, corresponde al nombre canónico de un host, es decir, el nombre de referencia en caso de alias.
  - Tipo=SOA, marca el inicio de datos de una zona
  - Tipo=PTR, es un puntero que tiene el nombre de un dominio
  - Tipo=HINFO, información de un host
  - Tipo=TXT, representa a un texto arbitrario
- Clase, valores:
  - IN, se refiere a Internet
  - DNS, cuando se utiliza como directorio para otro servicio de nombres
- Duración de vida, indica el tiempo de validez de un registro, conocido también como TTL.

Normalmente el espacio de nombres está dividido en base a dos criterios:

- Organizativa, representa el tipo de organización que contiene los recursos: com, edu, gob, mil, net, org, int.

Nombre de dominio	Asignado a:
com	Organización comercial
edu	Institución educativa
gob	Organización de gobierno
mil	Grupo militar
net.	Centro de soporte de red principal
org	Organización general
int	Organización internacional

Tabla 7 sufijos de tipo organizativos

- Geográfica, indica el lugar geográfico donde se encuentra el recurso: es, pe, uca, fr, zn.

Una observación importante es que los servidores DNS no reconocen nombres relativos.

#### 6.5.1.2.2 Resolución de nombres en DNS

El repositorio de nombramiento de DNS está dividido en zonas y cada zona contiene lo siguiente:

- Atributo de datos para nombres de dominio, excepto nombres que pertenecen a subdominios.
- Nombres y direcciones de por los menos dos servidores, del tipo servidores de autoridad (también llamados servidores de nombres raíz) para esta zona.
- Nombres y direcciones de autoridad para subdominios
- Parámetros para replicación y caché para esta zona

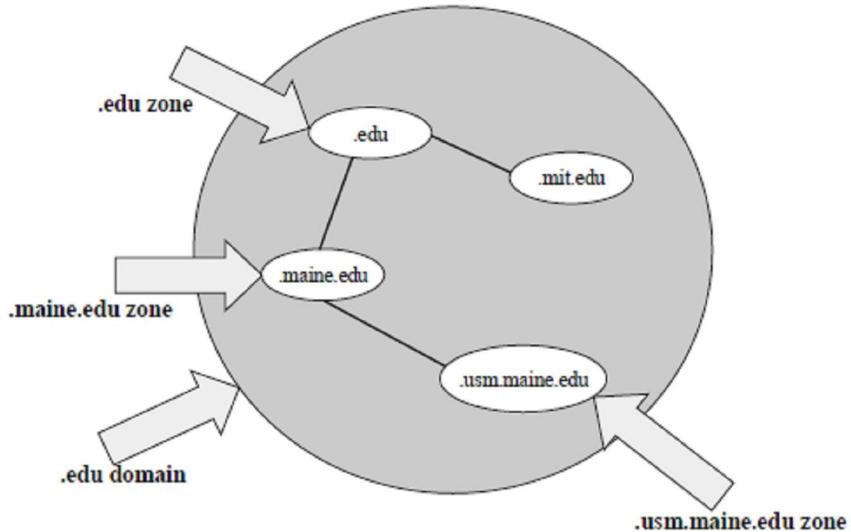


Fig.88 Resolución de nombres en DNS

En el grafico se muestra la relación entre dominios y zonas. Se observa el dominio.edu que agrupa cuatro zonas, .edu,.mit.edu, .maine.edu, .usm.maine.edu. Cada zona a su vez representa a un servidor autorizado que maneja los servicios declarados dentro de ellas.

Para el proceso de resolución de nombres se consideran las siguientes características:

- a) cada host conoce por lo menos dos servidores DNS en dominio local
- b) cada servidor DNS conoce el nombre de un servidor en dominio padre y al menos uno o más servidor raíz
- c) los servidores raíz conocen servidores de nombre raíz para dominios de alto nivel
- d) la resolución de nombres DNS está basado en mensajes UDP y TCP con tiempo de espera y máximo de número de intentos. El envío de mensajes utilizará el puerto 53, se utilizará UDP para las consultas de rutina (resolución de nombres), y mensajes TCP en las operaciones de transferencias de zona a zona.

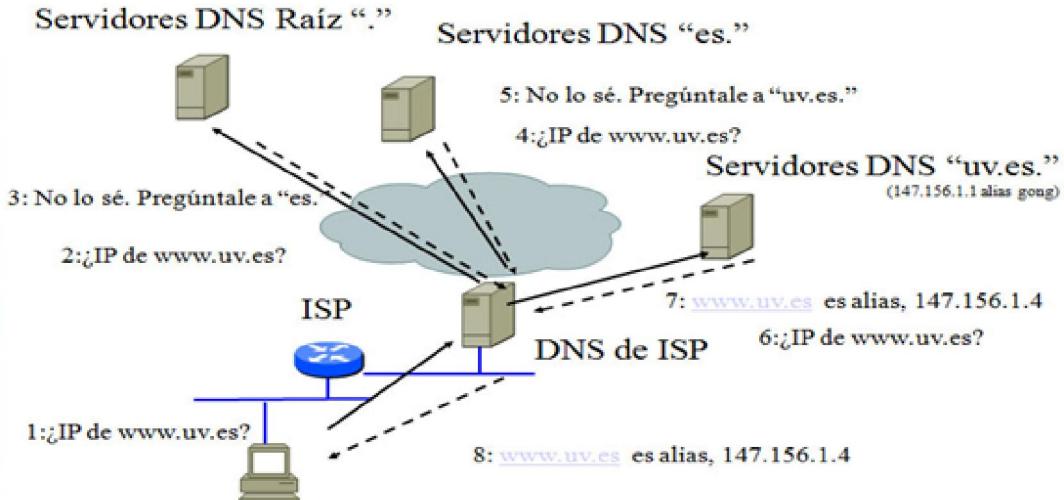


Fig.89 Funcionamiento del DNS

En el grafico se muestra la secuencia de operaciones dentro del DNS destinadas a resolver el nombre www.uv.es, teniendo como respuesta la dirección IP 147.156.1.4

Dentro del concepto de resolución de nombre en DNS existen las siguientes modalidades de resolución:

- resolución directa, en la que dado un nombre DNS se va a localizar su dirección IP asociada.

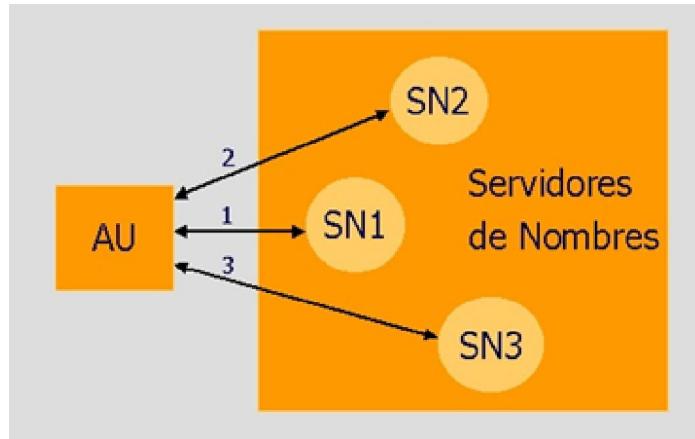


Fig.90 Resolución directa en DNS

- Resolución inversa, en la que conocida la dirección IP se desea localizar el nombre DNS asociado.
- Resolución iterativa gestionada por el cliente, en la que el cliente teniendo el nombre lo envía alternativamente a cada servidor de nombres hasta alcanzar la resolución.
- Resolución iterativa gestionada por el servidor, en la que el cliente envía el nombre al servidor de nombre local, el cual sino resuelve el nombre contacta

con otros servidores de nombres de manera consecutiva, hasta alcanzar la resolución.

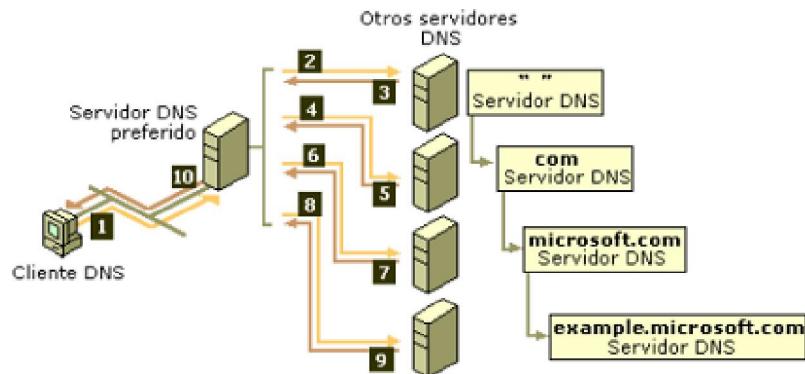


Fig.91 Resolución iterativa gestionada por el servidor

- Resolución recursiva gestionada por el servidor, en la que el cliente envía el nombre al servidor de nombres local, el cual si no tiene el nombre a buscar deriva la solicitud a otro servidor el que a su vez si no conociera dicho nombre asume la responsabilidad de localizar este nombre, contactándose con otros servidores posteriormente.

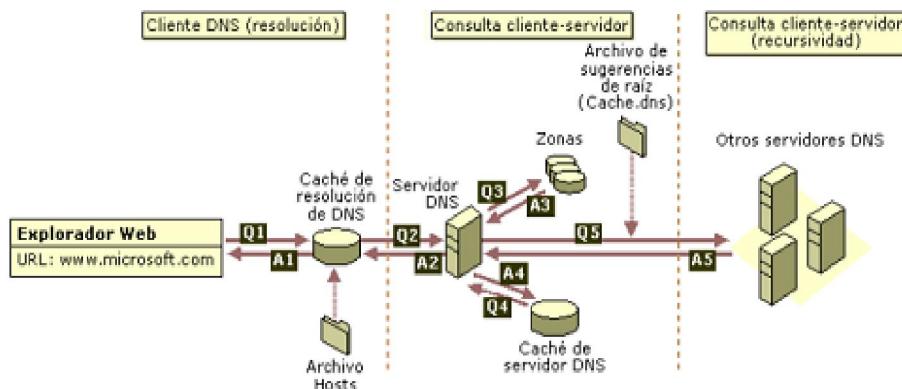


Fig.92 Resolución recursiva gestionada por el servidor

### 6.5.1.2.3 Ejemplo de resolución de nombres en DNS

Para ilustrar el proceso de resolución de nombres en DNS presentaremos el siguiente caso:

Supongamos que el host db.utoronto.ca desea encontrar la dirección IP deseada la dirección de IP de malliag.math.uwaterloo.ca, las operaciones que realizaría para la resolución sería

- en primer lugar, se conecta a su servidor de nombre local DNS.utoronto.ca
- el servidor de nombre local contacta al servidor de nombre raíz si es necesario
- el servidor de nombre raíz contacta al servidor de autoridad DNS.uwaterloo.ca, si es necesario.

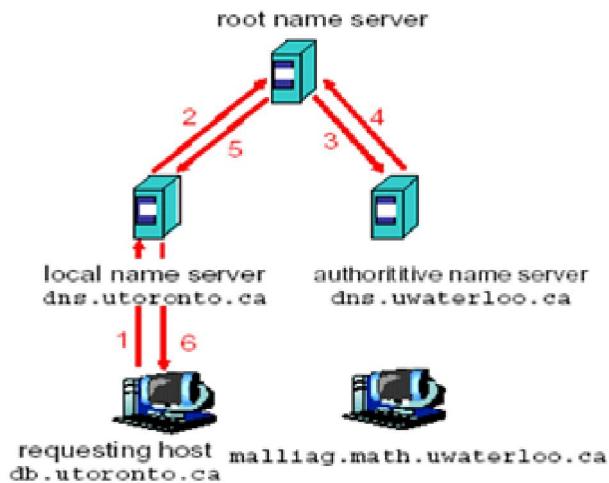


Fig.93 Ejemplo de resolución de nombres

Consideremos la utilización del servidor de nombre raíz, su uso puede ser útil en la medida de que si no conoce el servidor de nombre de autoridad (es decir el servidor que tiene el recurso), podría conocer al servidor de nombres intermedio, a través del cual podría averiguar la ubicación del servidor de nombres de autoridad y por lo tanto completar la resolución.

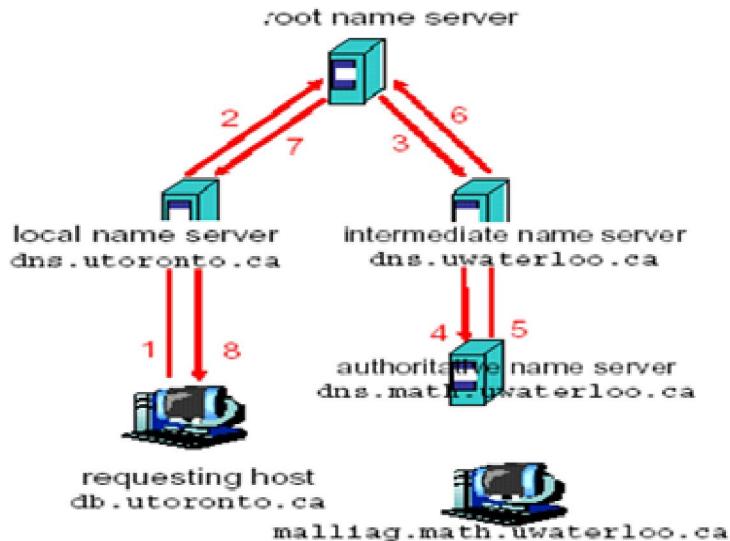


Fig.94 Uso de servidor intermediario para resolución de nombres

teniendo en consideración el tipo de navegación iterada gestionada por el servidor del ejemplo que estamos estudiando el siguiente grafico muestra su utilización.

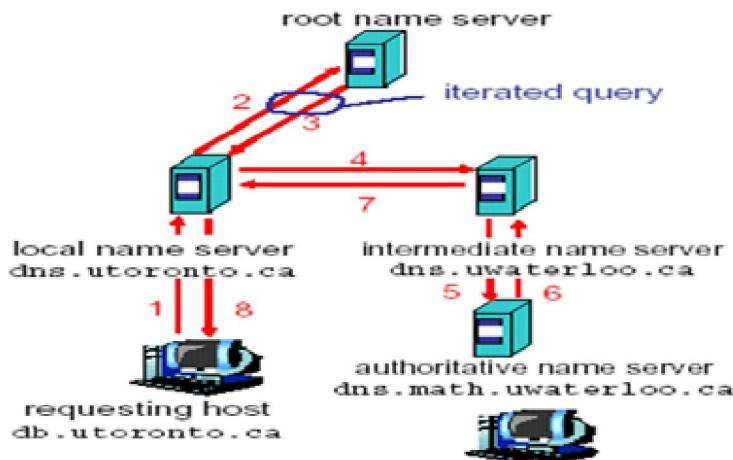


Fig.95 Resolución de nombres por iteración

#### 6.5.1.2.4 Consideraciones a tener en cuenta durante resolución de nombres en DNS

Un cliente DNS es llamado resolvedor, teniendo la responsabilidad de solicitar la resolución de nombres.

Dentro de la ejecución de su responsabilidad el cliente normalmente invoca un programa de aplicación y especifica el nombre de una maquina y posteriormente la aplicación debería buscar la dirección IP de dicha maquina, por lo que contactara con el resolvedor de nombres de la maquina local, a continuación se creara un mensaje y lo enviara al servidor de nombres local. Cuando el servidor del nombre recibe la consulta tiene que realizar las siguientes acciones:

- ¿Es el servidor una autoridad para el nombre?, si lo es ubicara la dirección para el nombre en su repositorio y transferirá la respuesta a la consulta y se la devolverá al cliente.
- ¿Está el nombre en la caché del servidor?, si lo está, del reporte de la caché será mapeado al cliente y se etiquetará como no autoritativo y da el nombre del servidor que proveerá el enlace.

Otra consideración importante es que el cliente debe recibir la respuesta rápidamente de lo contrario será considerada fuera de fecha u obsoleta.

Debe considerarse también de que los servidores mantienen una caché como los nombres usados recientemente y el servidor que proveyó resolución de cada nombre.

Finalmente, la resolución de nombres utilizada por DNS es de tipo bottom-up por lo que se inicia en el servidor de nombre local y si no se encuentra sube al padre de éste.

#### 6.5.1.3 Protocolo de acceso ligero a directorio (LDAP)

La implementación LDAP aparece como una forma de simplificar el modelo X.500 anteriormente reseñado, manteniendo sin embargo, su nivel de seguridad y servicios, como diferencias fundamentales con respecto a X.500 podemos indicar que esta implementado utilizando el modelo TCP/IP de cuatro capas en vez del modelo X.500 basado en OSI de siete capas, por otra parte, su uso es para

búsqueda de recursos y esto se va a realizar mediante conexión a un solo servidor por vez.

Otra característica importante, es la referida al control de acceso que se hace por cada nombre distinguido y por ruta. Existen dos implementaciones de LDAP la v2 y v3.

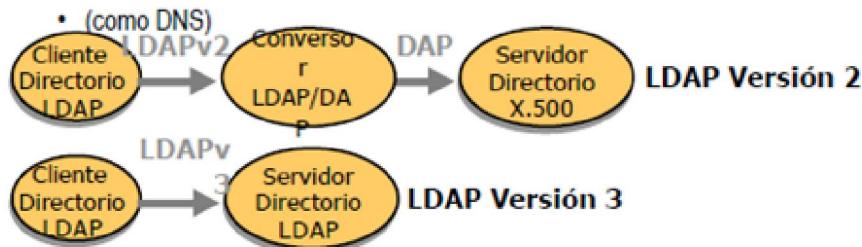


Fig.96 LDAP v2 y v3

En el grafico se observa el comportamiento de LDAP v2. en ella se aprecia de que los servicios o recursos están declarados inicialmente en un servidor de directorio X.500 por lo que el servidor LDAP se convierte simplemente en un conversor o adaptador que va a permitir que un cliente LDAP pueda solicitar resolución de nombres. Otra vista de la misma operación se puede ver en el siguiente grafico.

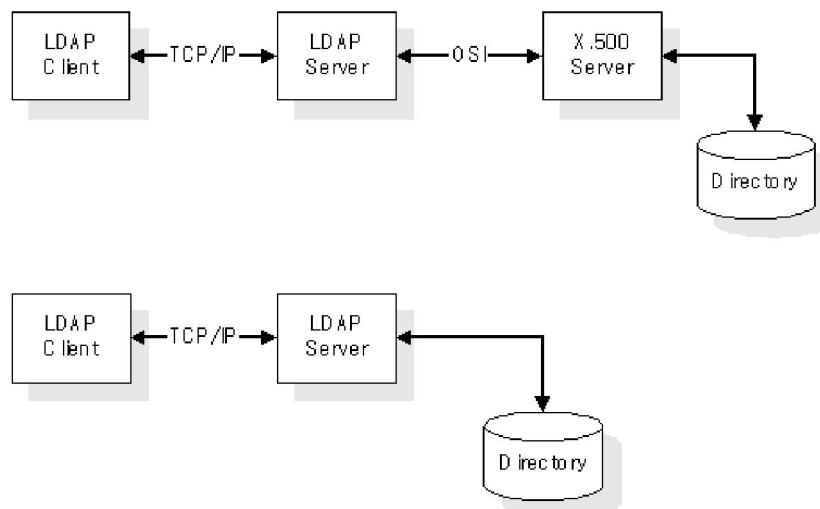


Fig.97 Relación entre X.500 y LDAP

Posteriormente, el LDAP v3 ya no necesita a X.500 como repositorio de información de servicios. Por otro lado, otra ventaja de v3 es la flexibilidad para conocer las clases de objetos y atributos, a diferencia de v2 en que el esquema de resolución era fijo, además, v3 usa Unicode y soporta SSL. A continuación de manera resumen mostramos una tabla donde comparamos las características de las diferentes versiones de LDAP con X.500.

	X.500	LDAPv1	LDAPv2	LDAPv3
Client to Server	OSI	TCP/IP	TCP/IP	TCP/IP
Server to Server	OSI	X.500 OSI	None	TCP/IP (replication only)*
Security	Password Strong	Password	Password	Password Strong*
Not found action	Servers chain requests	Servers chain requests	Dumb referrals	Referrals
Schema	Fixed	Fixed	Fixed	Dynamic

Tabla 8 Versiones de LDAP

En relación con otras aplicaciones podemos considerar algunos detalles:

- En relación con la web se puede utilizar llamadas LDAP usando formato URL
- Con relación a DNS se utiliza el concepto domain component, que permite usar nombre distinguido y nombre distinguido relativo.
- En lo que se refiere a la localización de servidores LDAP se utiliza el concepto de registros SRV que permiten anunciar localización de servicios LDAP.
- Con relación a correo se utiliza vCard y el protocolo mime.

En el grafico siguiente se observa una interacción típica entre cliente LDAP con servidor LDAP en la que se establece la resolución de nombres.

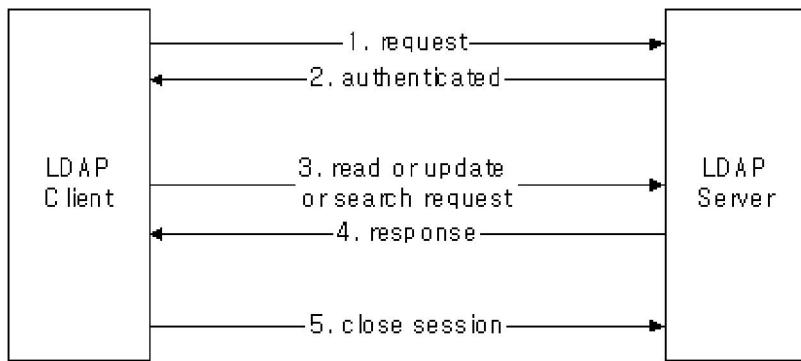


Fig.98 Secuencia de operaciones en LDAP

Dentro de la implementación LDAP se han definido cuatro modelos que son: el modelo de información, modelo de nombres, modelo funcional y modelo de seguridad.

- **Modelo de información**

Indica la forma como se encuentra organizada la información, en este caso el directorio va a almacenar y organizar estructuras de datos a través de objetos, los que pueden representar cualquier recurso definido, estos están descritos por esquemas, los cuales guardaran información sobre la clase de objetos que están permitidos, los atributos que contienen, los atributos opcionales, la sintaxis de cada atributo y si tiene herencia con algún otro objeto.

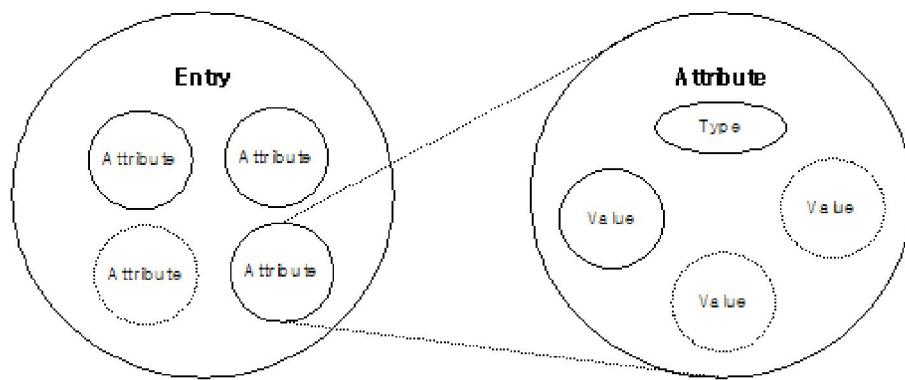


Fig.99 Atributo y entidades en LDAP

- **Modelo de nombres**

Describe la organización de la información y el mecanismo de resolución, en este caso al ser un descendiente de X.500 utiliza tanto el concepto de nombre distinguido y de DIT para la representación de un espacio de nombres parcial.

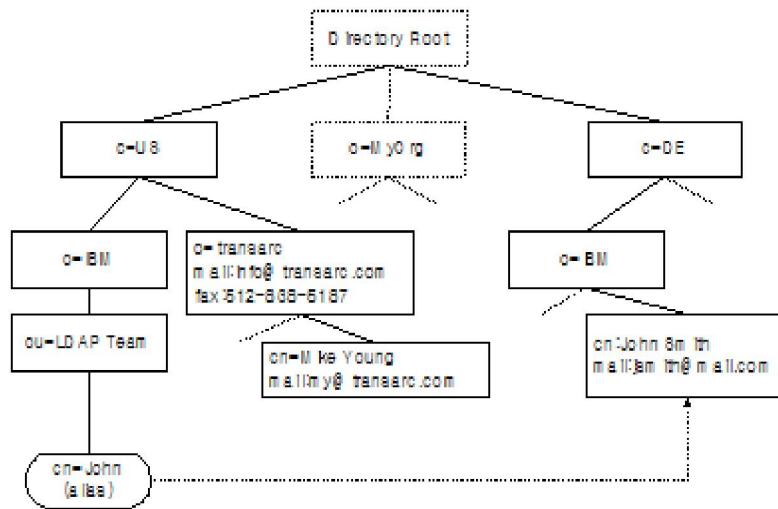


Fig.100 Modelo de nombres

Por otro lado, la navegación mediante el DIT es de tipo referencial (iterativa) y gestionado por el cliente.

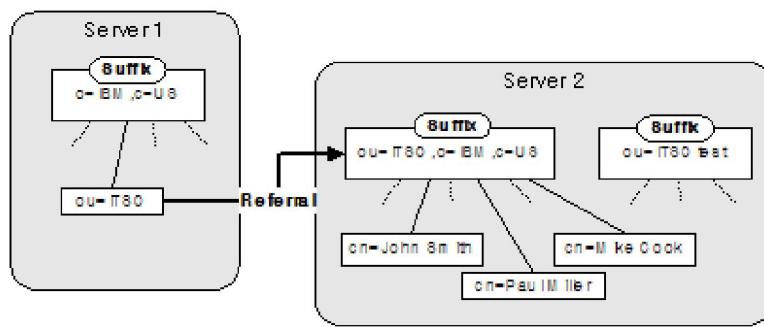


Fig.101 Recorrido iterativo

- **Modelo funcional**

Es el que describe las operaciones que pueden ser ejecutadas sobre la información del recurso localizado. Una de las operaciones claves es la operación de consulta o localización de recursos, la operación de actualización que permite modificar recursos y atributos, así como agregar nuevos o eliminarlos, la operación de autenticación, que permite la validación de la información.

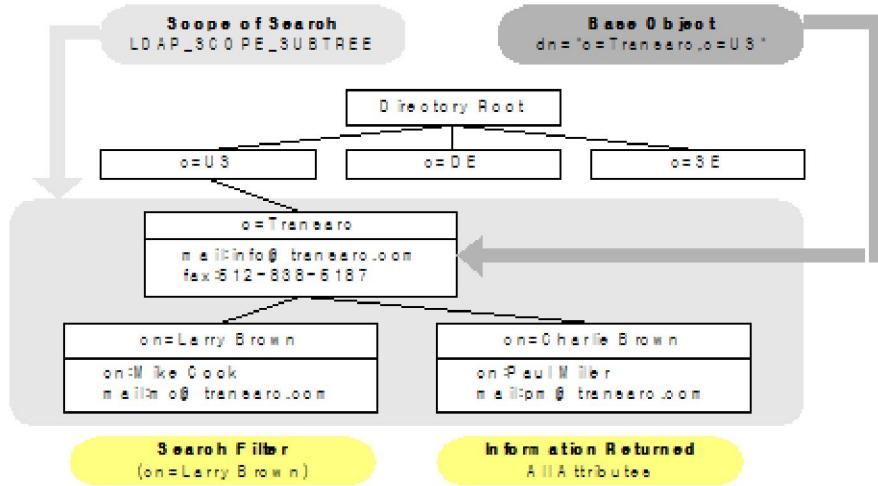


Fig.102 Modelo funcional

Se debe indicar también que para realizar la operación de resolución de nombres en LDAP v2 se utiliza una conexión de tipo TCP usando el puerto 389, la conexión será de tipo asíncrona permitiendo múltiples peticiones en curso, pero el servidor contesta cada una de ellas cuando pueda. En el caso del LDAP v3 generalmente utiliza una conexión de tipo UDP por ser ésta más rápida.

- **Modelo de seguridad**

Utilizando las características inherentes de X.500 el modelo de seguridad de LDAP permite autenticación, integridad de datos, un nivel de confidencialidad bastante alto y mecanismo de autorización definidos.

Para lograr esto, se establece un sistema de autenticación y de seguridad basado en dos elementos importantes como son kerberos y SSL. Dentro de estas características se puede considerar un eficiente mecanismo de replicación de datos que permitirá la disponibilidad del servidor en cualquier momento.

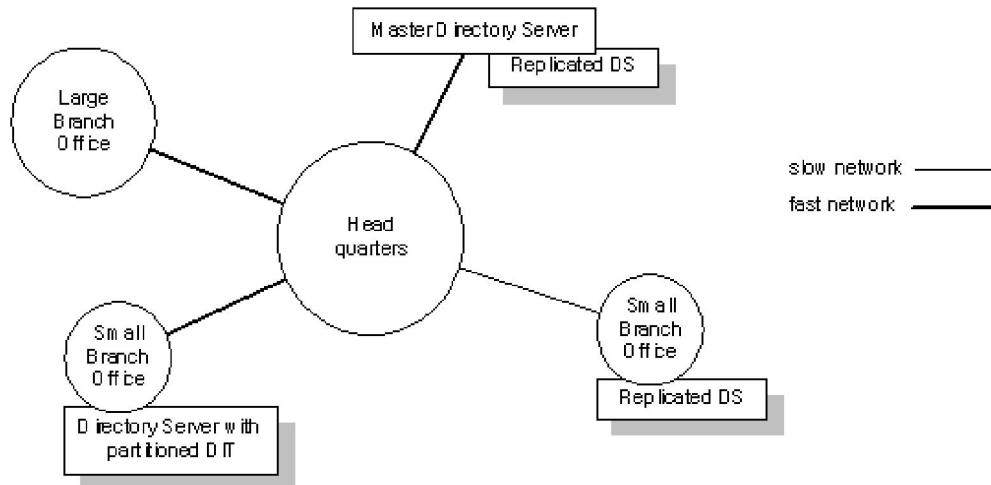


Fig.103 Replicación de datos en LDAP

Finalmente, podemos consignar como ventajas importantes del uso de LDAP el que es un estándar abierto y por lo tanto, adaptable de acuerdo al tipo de organización que lo desee utilizar, está disponible virtualmente en todos los sistemas operativos y existen múltiples aplicaciones diseñadas para trabajar con LDAP. Pero también, tiene desventajas como que es mas lento que el modelo X.500 original, el control de acceso y replicación estará definido por la implementación del fabricante y no por un estándar establecido, además los clientes tienden a solicitar recursos cada vez más complejos lo que se traducirá en una disminución de la performance de este modelo.

Implementaciones prácticas de LDAP son OpenLDAP, Netscape Directory Server, Secure Way Directory de IBM y Active Directory Service de Microsoft.

#### 6.5.1.4 Directorio Activo (Active Directory)

Active Directory es un servicio de directorio que provee los fundamentos para resolución de nombres en redes distribuidas construidas sobre Windows 2000 y versiones superiores, debido a que su API provee acceso a los datos almacenados en dicho directorio.

Para lograr esto, se utiliza el Agente de Sistema de Directorio (DSA) el que se conectara con otro similar para realizar búsqueda utilizando una interface de llamada a procedure remoto propietario. La estructura de datos es similar a X.500, debido a que Active Directory desciende de LDAP. Además, utiliza DNS SRC para localizar servicios y DNS dinámico que permitirá actualizar la información de tipo no manual.

El directorio, elemento base de esta implementación estará dividido en dominios el cual se encuentra a su vez controlado por el domain controller. El domain controller es el responsable de la replicación de todos aquellos cambios de información realizados dentro del dominio y maneja las operaciones que se puedan realizar dentro de los objetos pertenecientes al dominio.

Microsoft utiliza las mismas características que se utilizan en X.500 para definir los atributos que pueden almacenarse en un servicio de directorio para un recurso identificado y que van a residir en el propio directorio del servidor

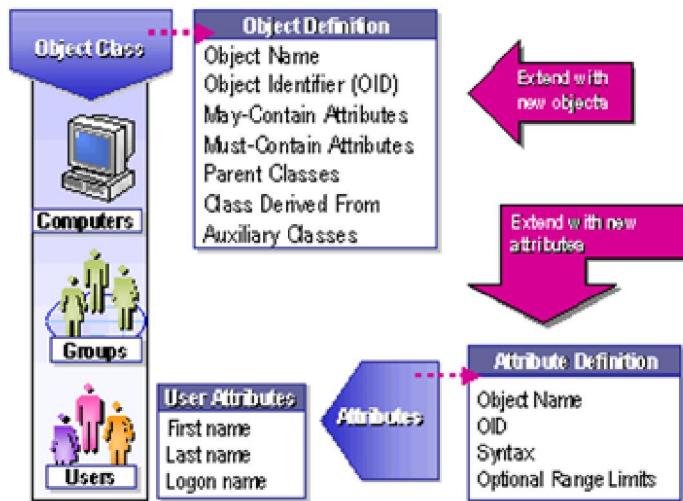


Fig.104 Servicio de directorio

se utiliza el concepto de árbol para describir una jerarquía de objetos y contenidos (grupo de objetos), además Active Directory soporta nombre distinguido y nombre distinguido relativo. Una particularidad es que existen los dominios como un área de seguridad dentro de la red, así como la presencia de árbol de dominios en la que se agrupan dominios dentro de un mismo esquema.

Para que los servicios de directorio puedan funcionar correctamente existe el concepto de catálogo global que es un repositorio central de información de los objetos que se encuentran organizados en un árbol (análogo al DIT) o al bosque (análogo al DIB).

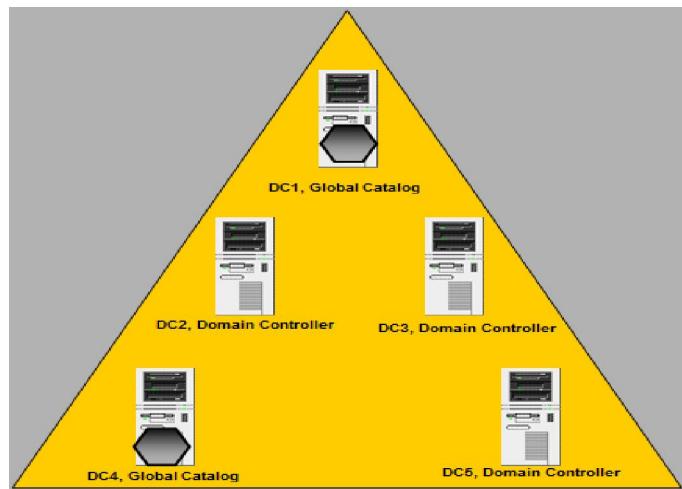


Fig.105 Árbol

Debe existir por lo menos un catalogo global por dominio, pudiendo haber controladores adicionales en los controladores dominio, dependiendo del número de usuarios y el número de consultas, pero para ello se requerirá memoria adicional.

Además existe el concepto de servidor de catalogo global que es un controlador de dominio que almacena una copia del catalogo global.

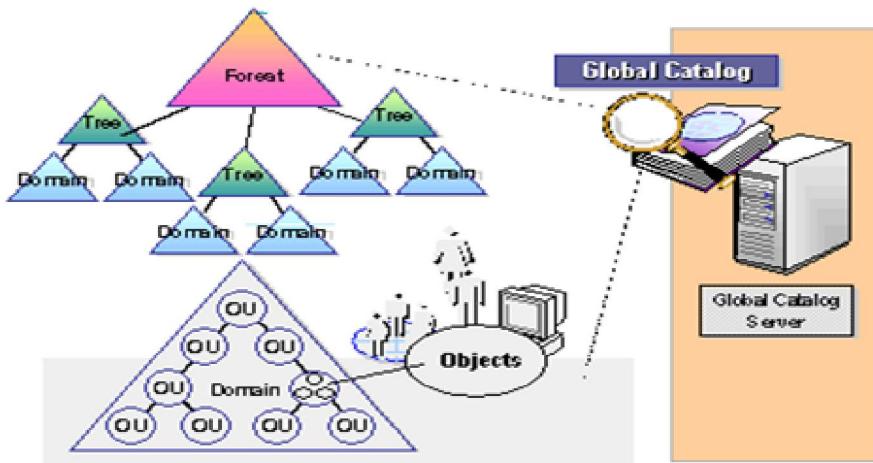


Fig.106 servidor de catalogo global

#### 6.5.1.4.1 Resolución de nombres en Active Directory

En el procedimiento de resolución de nombres en Active directory las operaciones que se llevan a cabo para localización de un servicio o recurso, se realiza lo siguiente:

- El cliente al enviar el nombre a localizar, se contacta con el controlador de dominio.
- El controlador de dominio busca la dirección IP en la maquina cliente
- Si el cliente no se encuentra en la red local, el controlador de dominio mirara en el catalogo global para buscar otro controlador de dominio que este cercano al cliente y vuelve a repetirse el procedimiento hasta resolver el nombre.
- Se notificara al cliente si no se ha podido conectar al controlador de dominio mas cercano.

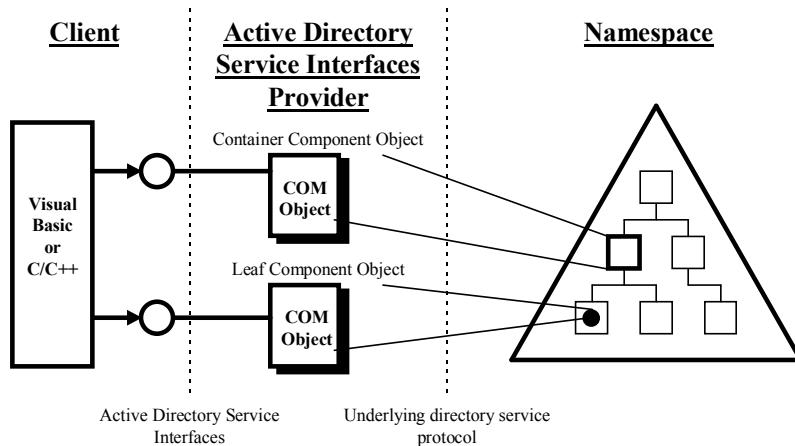


Fig.107 espacio de nombres

El contenedor que se utiliza para organizar los objetos en un dominio se denomina OU y pueden haber varios OUs dentro de un dominio.

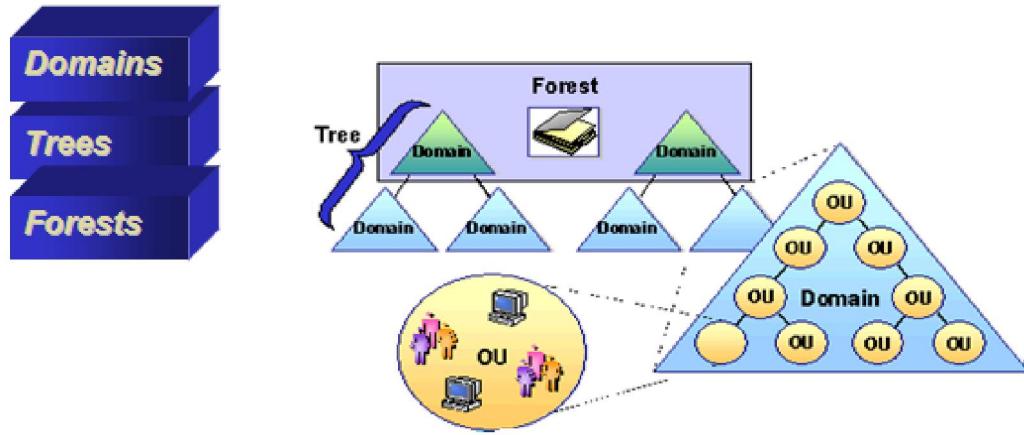


Fig.108 Dominios

Como en un DIT, los OUs están jerarquizados y servirán como parte de la ruta a seguir para localización de un recurso.

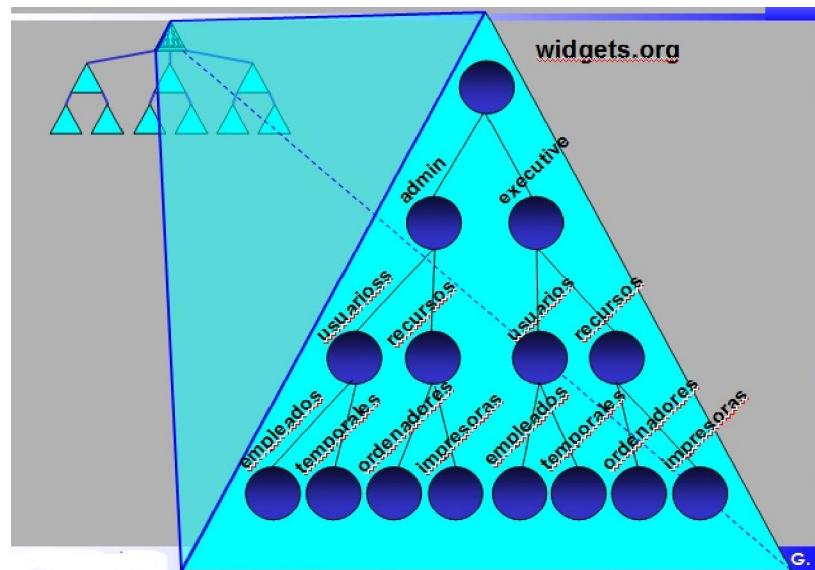


Fig.109 ejemplo de árbol en Active Directory