# Resumen Tema 6

# Servicios de red implicados en el despliegue de una aplicación web.

## 1.-Servicio de nombres de dominio (DNS)

Debe existir algo que nos traduzca los nombres a IPs o viceversa. Sí, este algo no es otro que el **servidor DNS** o un archivo de texto, denominando **hosts**, como el archivo **/etc/hosts** en sistemas GNU/Linux.

Puede convivir en una misma máquina un servidor DNS y el archivo /etc/hosts, pero primero se intentará la resolución IP/Nombre mediante el archivo /etc/hosts y en caso de no encontrar correspondencia, actuará el servidor DNS.

El fichero **/etc/hosts** permite **alias** de nombres de dominios, esto es, una misma IP puede apuntar a nombres distintos. Cada línea del fichero comenzará con una IP y en la misma línea separados por espacios o tabuladores puedes escribir los nombres de dominios correspondientes. El primer nombre, el más cercano a la IP, es considerado el principal, los demás son alias de éste.

## 1.1.- Ventajas del DNS.

El DNS permite que cualquier cambio efectuado solamente en un servidor se replique en todos los servidores DNS que la configuración permita.

Cualquier cambio es dinámico: configuras solamente un servidor y éste se encarga de replicar el cambio.

Cada servidor DNS se ocupa de su zona, eso no imposibilita el acceso a otras zonas y por lo tanto, a la visibilidad y conectividad de otros dominios que no dependan de ese servidor DNS.

Una zona DNS es aquella parte del DNS para la cual se ha delegado la administración, es decir, cuando configuras un dominio en un servidor DNS éste debe pertenecer a una zona.

Tipos de zonas:

* **Zona de búsqueda directa:** las resoluciones de esta zona devuelven la dirección IP correspondiente al recurso solicitado.
* **Zona de búsqueda inversa:** las resoluciones de esta zona buscan un nombre de equipo en función de su dirección IP.

Los servidores DNS no solamente sirven para la resolución de nombres de Internet, también se pueden utilizar en redes locales.

Cuando un usuario de la red local intenta acceder a un recurso local, podrá utilizar nombre en lugar de direcciones IP. Si el usuario desea acceder fuera de la red local a algún recurso en Internet, el DNS local nunca podrá llevar a cabo dicha resolución y se la traslada al siguiente servidor DNS (en Internet) en su jerarquía de servidores DNS, hasta que la petición sea satisfecha.

* Ping 192.168.200.100
* Ping cliente.local

En ambos casos, deberías obtener la misma respuesta. Esto suele ser muy útil cuando los hosts reciben su IP por DHCP ya que puede ocurrir que desconozcamos la IP que tiene cierto equipo pero sí conocer su nombre en el dominio, que será invariable.

Ventajas:

1.- Desaparece la carga excesiva en la red y en los hosts: ahora la información esta distribuida por toda la red, al tratarse de una base de datos distribuida.

2.- No hay duplicidad de nombres: el problema se elimina debido a la existencia de dominios controlados por un único administrador. Puede haber nombres iguales pero en dominios diferentes.

3.- Consistencia de la información: ahora la información que está distribuida es actualizada automáticamente sin intervención de ningún administrador.

## 1.2.- Nombres de dominio.

Para la redirección deberá existir un servidor DNS que las resuelva o bien, en su defecto o a mayores, deberán existir las entradas correspondientes en el fichero del sistema local **/etc/hosts.** En caso de coexistir, primero se prueba la resolución en el fichero y luego en el servidor.

Si estás pensando en pocos equipos a resolver el nombre de dominio la simplicidad del fichero  te permitiría no tener que montar un servidor, pero si el número de equipos que deben resolver el nombre en IP es elevado, el sistema del fichero es complicado de mantener y deberías pensar en montar un servidor DNS.

La complejidad radica en que en el fichero **/etc/hosts** los cambios son estáticos, así, para actualizar o activar un nuevo cambio debe editarse en todos los ficheros /etc/hosts implicados.

Ejemplo:

* Primero, se debe averiguar que servidor DNS resuelve el domino raíz 'org' a una IP.
* Segundo, una vez obtenida esa IP que gobierna el dominio raíz 'org', se le pregunta por la IP del servidor DNS que gobierna el subdominio 'debian' bajo 'org'.
* Tercero, una vez obtenida la IP del servidor DNS que gobierna el dominio 'debian.org' se le pregunta por la IP del equipo 'www.debian.org'

Bien, pues no existe un número limitado de redirección de consultas, lo que sucede es que las consultas se van escalando hasta encontrar un servidor DNS que las resuelva, y escalando y escalando puede ser que las consultas se resuelvan en los últimos servidores DNS a los cuales se puede preguntar: los servidores raíz.

Todos los servidores DNS son servidores caché, lo que significa que recuerdan las consultas efectuadas. Es más, los equipos clientes, desde donde se hace la consulta a través del navegador también poseen una memoria caché DNS, de tal forma que antes de a preguntar al servidor DNS, se mira en la caché del propio sistema operativo, y si se obtiene la respuesta, el proceso se ha acabado.

***El sistema DNS en realidad es una base de datos distribuida, que permite la administración local de segmentos que juntos componen toda la base de datos local. Los datos de cada segmento están disponibles para toda la red a través de un esquema cliente-servidor jerárquico.***

## 1.2.1.-¿Cómo es un nombre de dominio?

**Primero**: Los puntos separan dominios y subdominios, empezando de derecha a izquierda tendrás dominios de primer nivel y dominios de segundo, tercer, …, n-ésimo nivel, denominados subdominios. Así:

* **org** es el dominio de primer nivel que identifica a organizaciones.
* **debian** es un subdominio, en este caso dominio de segundo nivel bajo org, que identifica al nombre de la organización o al nombre de la empresa, sucursal, etc.
* **www** es un subdominio, en este caso dominio de tercer nivel bajo debian, que identifica al equipo donde está colgada la página web, esto es, identifica el servidor web que aloja la página web. Es el dominio www que el servidor DNS redirecciona a la IP del servidor web.
* **Segundo**: **http://** es el protocolo de hypertexto que permite la correcta visualización de la página web en el navegador. Es lo que el navegador autocomplementa en caso de no estipular uno propio en la barra de direcciones URL con en nombre de dominio.

Los dominios de primer nivel identifican el tipo de página web que solicitas o bien la localización de la misma:

* **net** identifica redes.
* **com** identifica comercio.
* **es** identifica localización España.
* **tk** identifica localización Tokelau.

Esto suele ser lo común, más no es obligatorio.

A nivel gramatical los dominios deben cumplir una serie de requisitos. Por ejemplo:

* Sólo pueden estar compuestos de letras (alfabeto inglés), números y guiones ("-").
* No pueden empezar o terminar por guiones.
* Tienen que tener menos de 63 caracteres sin incluir la extensión, y más de uno o dos dependiendo del dominio de primer nivel.

Hoy día ya es posible registrar dominios con caracteres de otras lenguas no inglesas, como la ñ o la ç. Estos dominios se denominan **multilingües**.

## 1.2.2.-Jerarquía de nombres de dominio.

El espacio de nombres de dominio está organizado de forma jerárquica. El nivel más alto es el dominio raíz, que se representa como un punto ( . ) y el siguiente nivel en la jerarquí se llama dominio de nivel superior (TLD).

Sólo hay un dominio raíz, pero muchos TLDs y cada TLD se llama dominio secundario del dominio raíz.

El dominio raíz es el dominio principal, ya que está un nivel por encima de un TLD y cada TLD, a su vez, pueden tener muchos dominios hijos. Los hijos de los dominios de nivel superior se llaman de segundo nivel, los del segundo nivel se llaman de tercer nivel, los del tercer nivel de cuarto, y así sucesivamente.

Por lo tanto el DNS, organiza los **hostname** en una jerarquía de dominios separados por el carácter punto '.'. Un **dominio** es una colección de nodos relacionados de alguna forma.

* rrhh.departamento.empresa.org
* marketing.departamento.empresa.org
* contabilidad.consultas.empresa.org

Donde:

* La empresa agrupa sus nodos en el **dominio** de primer nivel "org". Éste es un TLD.
* La empresa tiene un **subdominio**, dominio de segundo nivel "empresa" bajo "org". Así "empresa" es un dominio de segundo nivel, hijo del TLD "org".
* A su vez puedes encontrar nuevos **subdominios** dentro, en este caso: "departamento" y "consultas". Es decir, dominios de tercer nivel, hijos a su vez del dominio de segundo nivel "empresa".
* Finalmente, un nodo que tendrá un nombre completo conocido como totalmente cualificado o **FQDN**, que es la concatenación de: TLD, dominio de segundo nivel, dominio de tercer nivel, etc.

El símbolo del dominio raíz es el punto situado más a la derecha del nombre del dominio.

Sólo hay una raíz de dominio, pero hay más de 250 dominios de nivel superior, clasificados en los siguientes tres tipos:

* **TLD de código de país** (ccTLD): dominios asociados con países y territorios. Hay más de 240 ccTLD. Están formados por 2 letras, por ejemplo: **es, uk, en, y jp.**
* Dominios de nivel superior **genéricos** (gTLD): están formados por 3 o más letras. A su vez se subdividen en:
  + **Dominios de internet patrocinados** (sTLD): representan una comunidad de intereses, es decir, detrás existe una organización u organismo público que propone el dominio y establece las reglas para optar a dicho dominio. Por ejemplo: **edu, gov, int, mil, aero, museum.**
  + **Dominios de internet no patrocinados** (uTLD). Sin una organización detrás que establezca las reglas para optar a dicho dominio. La lista de gTLD incluye: **com, net, org, biz, info.**

<https://data.iana.org/TLD/tlds-alpha-by-domain.txt>

## 1.3.-Servidores de raíz.

La organización que gestiona globalmente los servidores raíz por concesión del gobierno estadounidense es la **ICANN**, la cual es una organización sin fines de lucro que opera a nivel internacional, responsable de asignar espacio de direcciones numéricas de protocolo de Internet (IP), identificadores de protocolo y de las funciones de gestión [o administración] del sistema de nombres de dominio de primer nivel genéricos (gTLD) y de códigos de países (ccTLD), así como de la administración del sistema de servidores raíz. Aunque en un principio estos servicios los desempeñaba **IANA** y otras entidades bajo contrato con el gobierno de EE.UU., actualmente son responsabilidad de ICANN.

**ICANN** es responsable de la coordinación de la administración de los elementos técnicos del DNS para garantizar una resolución unívoca de los nombres, de manera que los usuarios de Internet puedan encontrar todas las direcciones válidas. Para ello, se encarga de supervisar la distribución de los identificadores técnicos únicos usados en las operaciones de Internet, y delegar los nombres de dominios de primer nivel, como: com, info, etc.

***Las empresas, ciudades u organizaciones podrán registrar sus propios dominios genéricos, tras la decisión adoptada el 20 de Junio de 2011 por la ICANN en Singapur.***

Los servidores raíz son entidades distintas. Hay 13 servidores raíz o, más precisamente, 13 direcciones IP en Internet en las que pueden encontrarse a los servidores raíz.

Todos estos servidores almacenan una copia del mismo archivo que actúa como índice principal de las agendas de direcciones de Internet. Enumeran una dirección para cada dominio de nivel principal (.com, .es, etc.) en la que puede encontrase la propia agenda de direcciones de ese registro.

Las entidades encargadas de operar los servidores raíz son bastante autónomas pero, al mismo tiempo, colaboran entre sí y con ICANN para asegurar que el sistema permanece actualizado con los avances y cambios de Internet.

Los **trece** servidores raíz DNS se denominan por las primeras trece letras del alfabeto latino, de la **A** hasta la **M** (A.ROOT-SERVERS.NET., B.ROOT-SERVERS.NET., …, M.ROOT-SERVERS.NET.), y están en manos de 9 organismos y corporaciones diferentes e independientes, principalmente universidades, empresas privadas y organismos relacionados con el ejercito de EE.UU. Aproximadamente la mitad depende de organizaciones públicas estadounidenses.

<http://www.internic.net/zones/named.root>

<https://root-servers.org/>

## 1.4.- Funcionamiento del cliente DNS.

Cuando utilizas en un programa un nombre DNS, éste debe ser resuelto a una IP. Entonces, un cliente DNS busca el nombre que se utiliza en el programa, consultando los servidores DNS para resolver el nombre. Cada mensaje de consulta que envía el **cliente** contiene tres grupos de información, que especifican una pregunta que tiene que responder el servidor:

* Un nombre de dominio DNS especificado, indicado como un nombre de dominio completo (FQDN).
* Un tipo de consulta especificado, que puede establecer un registro de recursos por tipo o un tipo especializado de operación de consulta.
* Una clase especificada para el nombre de dominio DNS.

Las consultas DNS se resuelven de diferentes formas:

* A veces, un cliente responde a una consulta localmente mediante la información almacenada en la caché obtenida de una consulta anterior.
* El servidor DNS puede utilizar su propia caché de información de registros de recursos para responder a una consulta.
* Un servidor DNS también puede consultar, o ponerse en contacto con otros servidores DNS, en nombre del cliente solicitante para resolver el nombre por completo y, a continuación, enviar una respuesta al cliente. Este proceso se llama **recursividad**.
* Además, el mismo cliente puede intentar ponerse en contacto con servidores DNS adicionales para resolver un nombre. Cuando un cliente lo hace, utiliza consultas adicionales e independientes en función de respuestas de referencia de los servidores. Este proceso se llama **iteración**.

En general, el proceso de consulta DNS se realiza en dos partes:

* La consulta de un nombre comienza en un equipo cliente y se pasa al solucionador (resolver), el servicio Cliente DNS, para proceder a su resolución.
* Cuando la consulta no se puede resolver localmente, se puede consultar a los servidores DNS según sea necesario para resolver el nombre.

## 1.4.1.- Consultas recursivas.

El cliente DNS formula una consulta a tu servidor DNS preferido. Cuando el servidor DNS recibe una consulta, primero comprueba si puede responder la consulta en las zonas configuradas localmente en el servidor, esto es, en las zonas que posee autoridad. Así, pueden ocurrir dos situaciones:

1. Si el nombre consultado existe, esto es, coincide con un registro de recursos correspondiente en la información de zona local, el servidor responde con autoridad y usa esta información para resolver el nombre consultado.

2. Si el nombre consultado no existe, esto es, no existe ninguna información de zona para el nombre consultado, a continuación el servidor comprueba si puede resolver el nombre mediante la información almacenada en la caché local de consultas anteriores.

De nuevo, se dan dos situaciones:

a. Si el servidor preferido puede responder al cliente solicitante con una respuesta coincidente de su caché, finaliza la consulta y responde con esta información.

b. Si el servidor preferido no puede responder al cliente solicitante con una respuesta coincidente de su caché, el proceso de consulta puede continuar y se usa la **recursividad** para resolver completamente el nombre. Esto implica la asistencia de otros servidores DNS para ayudar a resolver el nombre. De forma predeterminada, el servicio cliente DNS solicita al servidor que utilice un proceso de recursividad para resolver completamente los nombres en nombre del cliente antes de devolver una respuesta. El servidor DNS se configura, de forma predeterminada, para admitir el proceso de recursividad como se muestra en el gráfico siguiente.

Para que el servidor DNS realice la recursividad correctamente, primero necesita información de contacto útil acerca de los otros servidores DNS del espacio de nombres de dominio DNS. Esta información se proporciona en forma de **sugerencias de raíz**, una lista de los registros de recursos preliminares que puede utilizar el servicio DNS para localizar otros servidores DNS que tienen autoridad para la raíz del árbol del espacio de nombres de dominio DNS. Los servidores raíz tienen autoridad para el dominio raíz y los dominios de nivel superior en el árbol del espacio de nombres de dominio DNS.

Un servidor DNS puede completar el uso de la recursividad utilizando las sugerencias de raíz para encontrar los servidores raíz. En teoría, este proceso permite a un servidor DNS localizar los servidores que tienen autoridad para cualquier otro nombre de dominio DNS que se utiliza en cualquier nivel del árbol del espacio de nombres.

Aunque el proceso de consulta recursiva puede usar muchos recursos cuando se realiza como se describe anteriormente, tiene algunas ventajas en el rendimiento para el servidor DNS. Por ejemplo, durante el proceso de recursividad, el servidor DNS que realiza la búsqueda recursiva obtiene información acerca del espacio de nombres de dominio DNS.

Esta información se almacena en la caché del servidor y se puede utilizar de nuevo para ayudar a acelerar la obtención de respuestas a consultas subsiguientes que la utilizan o concuerdan con ella. Con el tiempo, esta información almacenada en caché puede crecer hasta ocupar una parte significativa de los recursos de memoria del servidor, aunque se limpia siempre que el servicio DNS se activa y desactiva.

## 1.4.2.- Consultas iterativas.

La iteración es el tipo de resolución de nombres que se utiliza entre clientes y servidores DNS cuando se dan las condiciones siguientes:

* El cliente solicita el uso de recursividad, pero ésta se encuentra deshabilitada en el servidor DNS.
* El cliente no solicita el uso de recursividad cuando consulta el servidor DNS.

Una solicitud iterativa de un cliente informa al servidor DNS de que el cliente espera la mejor respuesta que el servidor DNS pueda proporcionar inmediatamente, sin entrar en contacto con otros servidores DNS.

Cuando se utiliza la iteración, un servidor DNS responde al cliente en función de su propio conocimiento específico acerca del espacio de nombres, sin tener en cuenta los datos de los nombres que se están consultando.

Cuando se proporciona una referencia, el cliente DNS asume la responsabilidad de continuar efectuando consultas iterativas a otros servidores DNS configurados para resolver el nombre.

Cuando se utiliza la iteración, un servidor DNS puede ayudar en la resolución de la consulta de un nombre además de devolver su mejor respuesta propia al cliente. En la mayor parte de las consultas iterativas, un cliente utiliza su lista de servidores DNS configurada localmente para entrar en contacto con otros servidores de nombres a través del espacio de nombres DNS si su servidor DNS principal no puede resolver la consulta.

## 1.4.3.-Consultas inversas.

En la mayoría de la consultas DNS los clientes normalmente realizan una búsqueda directa. Este tipo de consulta espera recibir una dirección IP como respuesta a la consulta. Pero, DNS también proporciona un proceso de búsqueda inversa, es decir, buscar un nombre de host a través de una dirección IP. Así, una búsqueda inversa busca la respuesta a una pregunta tipo como la siguiente: ¿Cuál es el nombre DNS del host que utiliza la dirección IP 192.168.200.100?.

DNS no se diseñó originalmente para aceptar este tipo de consulta. Un problema de compatibilidad con el proceso de consulta inversa es la diferencia en la forma en que el espacio de nombres DNS organiza e indexa los nombres, y cómo se asignan las direcciones IP. Si el único método para responder a la pregunta anterior fuera buscar en todos los dominios del espacio de nombres DNS, una consulta inversa llevaría demasiado tiempo y requeriría un procesamiento demasiado largo como para ser útil.

Entonces, para resolver este problema, en el estándar DNS se definió y se reservó un dominio especial para las **IP versión 4,** el dominio **in-addr.arpa**, en el espacio de nombres DNS de Internet con el fin de proporcionar una forma práctica y confiable para realizar las consultas inversas. Al crear el espacio de nombres inverso, los subdominios del dominio in-addr.arpa se crean con el orden inverso de los números en la notación decimal con puntos de las direcciones IP. ***100.200.168.192.in-addr.arpa.***

Este orden inverso de los dominios para el valor de cada octeto es necesario porque, a diferencia de los nombres DNS, cuando se leen las direcciones IP de izquierda a derecha se interpretan al contrario. Cuando se lee una dirección IP de izquierda a derecha, se ve desde su información más general (una dirección IP de red) en la primera parte de la dirección a la información más específica (una dirección IP de host) que contienen los últimos octetos. Por esta razón, se debe invertir el orden de los octetos de las direcciones IP cuando se crea el árbol del dominio in-addr.arpa..

Finalmente, el árbol del dominio in-addr.arpa, tal como se crea en DNS, requiere que se defina un tipo de registro de recursos adicional: el registro de recursos de puntero (PTR). Este registro de recursos se utiliza para crear una asignación en la zona de búsqueda inversa que, normalmente, corresponde a un registro de recurso de dirección (A) de host con nombre para el nombre del equipo DNS de un host en su zona de búsqueda directa.

**El dominio in-addr.arpa se usa en todas las redes TCP/IP que se basan en el direccionamiento del Protocolo de Internet versión 4 (IPv4). Para el Protocolo de Internet versión 6 (IPv6) se usa un nombre de dominio especial diferente, el dominio ip6.arpa.**

Ten en cuenta que, si el servidor DNS no puede responder el nombre de la consulta inversa, se puede utilizar la resolución DNS normal (ya sea la recursividad o la iteración) para localizar un servidor DNS con autoridad para la zona de búsqueda inversa y que contenga el nombre consultado. En este sentido, el proceso de resolución de nombres utilizado en una búsqueda inversa es idéntico al de una búsqueda directa.

## 1.5.- Tipos de servidores DNS.

Dependiendo de la configuración y funcionamiento de los servidores, éstos pueden desempeñar distintos papales:

* **Servidores primarios:** estos servidores almacenan la información de su zona en una base de datos local. Son los responsables de mantener la información actualizada y cualquier cambio debe ser notificado a este servidor.
* **Servidor maestro:** los servidores maestros son los que transfieren las zonas a los servidores secundarios. Cuando un servidor secundario arranca busca un servidor maestro y realiza la transferencia de zona. Un servidor maestro para una zona puede ser a la vez un servidor primario o secundario de esa zona. Se evita sobrecarga en los servidores primarios con transferencias de zonas.
* **Servidores secundarios:** o esclavos, aunque a su vez pueden ser maestros de otros servidores secundarios. Son aquellos que obtienen los datos de su zona desde otro servidor que tenga autoridad para esa zona. El proceso de copia de la información se denomina transferencia de zona.

Sevidores sólo cache: no tienen autoridad sobre ningún dominio, se limitan a contactar con otros servidores para resolver las peticiones de los clientes DNS. Mantienen una memoria caché con las últimas preguntas contestadas. Cada vez que un cliente DNS le formula una pregunta, primero consulta en su memoria caché. Si encuentra la dirección IP solicitada, se la devuelve al cliente; si no, consulta a otros servidores, apunta la respuesta en su memoria caché y le comunica la respuesta al cliente. Disponer de un servidor caché DNS en nuestra red local aumenta la velocidad de la conexión a Internet pues cuando navegamos por diferentes lugares, continuamente se están realizando peticiones DNS. Si nuestro caché DNS almacena la gran mayoría de peticiones que se realizan desde la red local, las respuestas de los clientes se satisfarán prácticamente de forma instantánea proporcionando al usuario una sensación de velocidad en la conexión. Muchos routers ADSL ofrecen ya este servicio de caché, tan solo hay que activarlo y configurar una o dos IPs de servidores DNS en Internet. En los equipos de nuestra red local podríamos poner como DNS primario la IP de nuestro router y como DNS secundario una IP de un DNS de Internet.

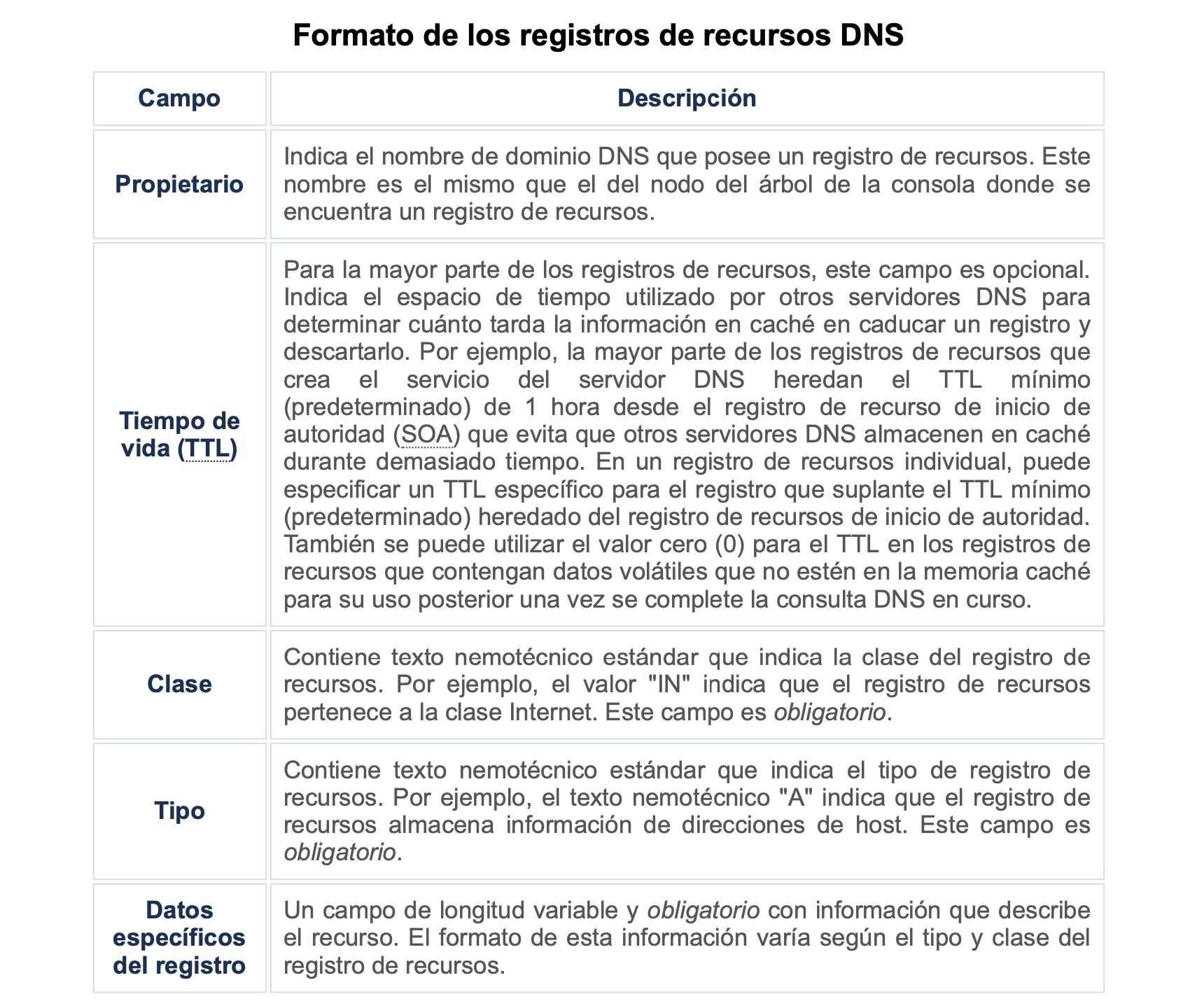
***Los servidores secundarios son importantes por varios motivos. En primer lugar, por seguridad: debido a que la información se mantiene de forma redundante en varios servidores a la vez. Si un servidor tiene problemas, la información se podrá recuperar desde otro. Y en segundo lugar, por velocidad: porque evita la sobrecarga del servidor principal distribuyendo el trabajo entre distintos servidores situados estratégicamente (por zonas geográficas, por ejemplo). Todos los servidores DNS guardan en la caché las consultas que resolvieron.Una transferencia de zona puede darse en cualquiera de los casos siguientes:***

* ***Cuando vence el intervalo de actualización de una zona.***
* ***Cuando un servidor maestro notifica los cambios de la zona a un servidor secundario.***
* ***Cuando se inicia el servicio Servidor DNS en un servidor secundario de la zona.***
* ***Cuando se utiliza el comando rndc en un servidor secundario de la zona para iniciar manualmente una transferencia desde su servidor maestro, por ejemplo:***
  + ***rndc retransfer proyecto-empresa.local***

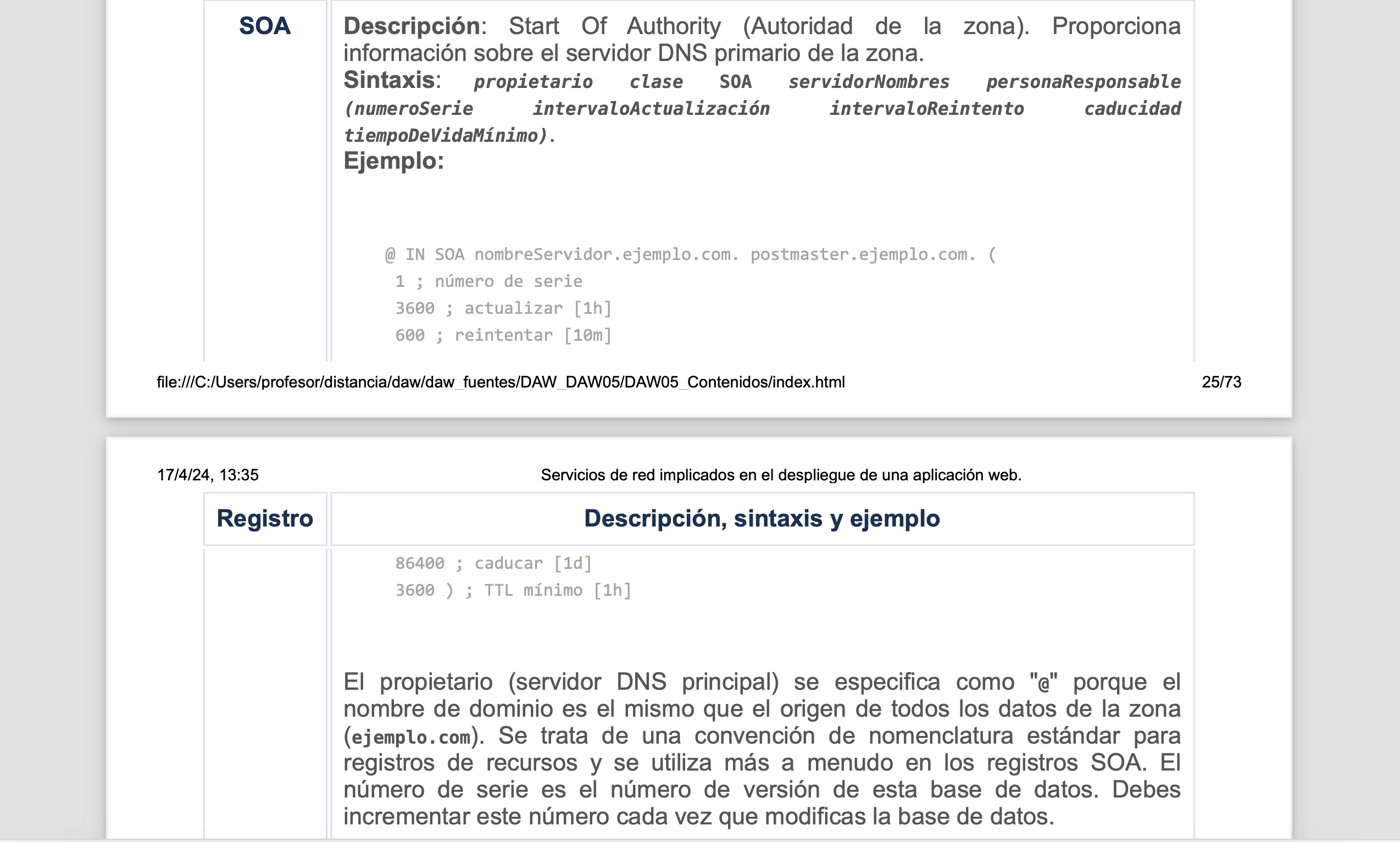
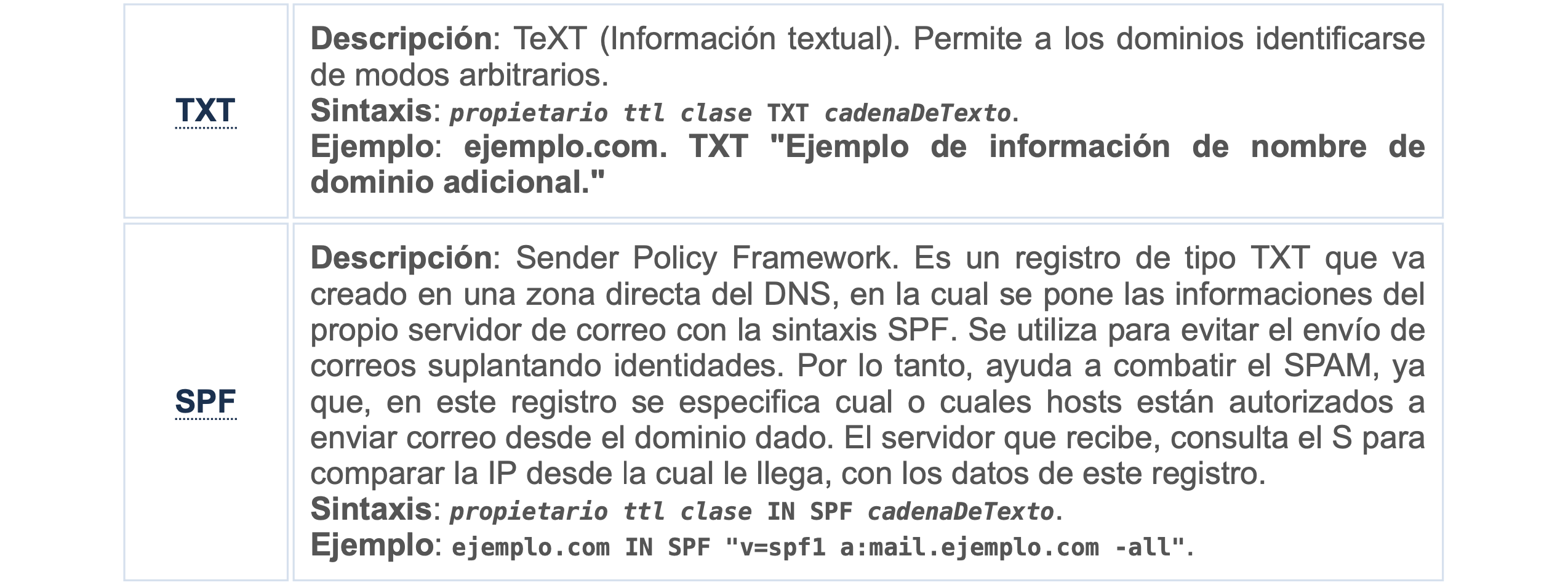
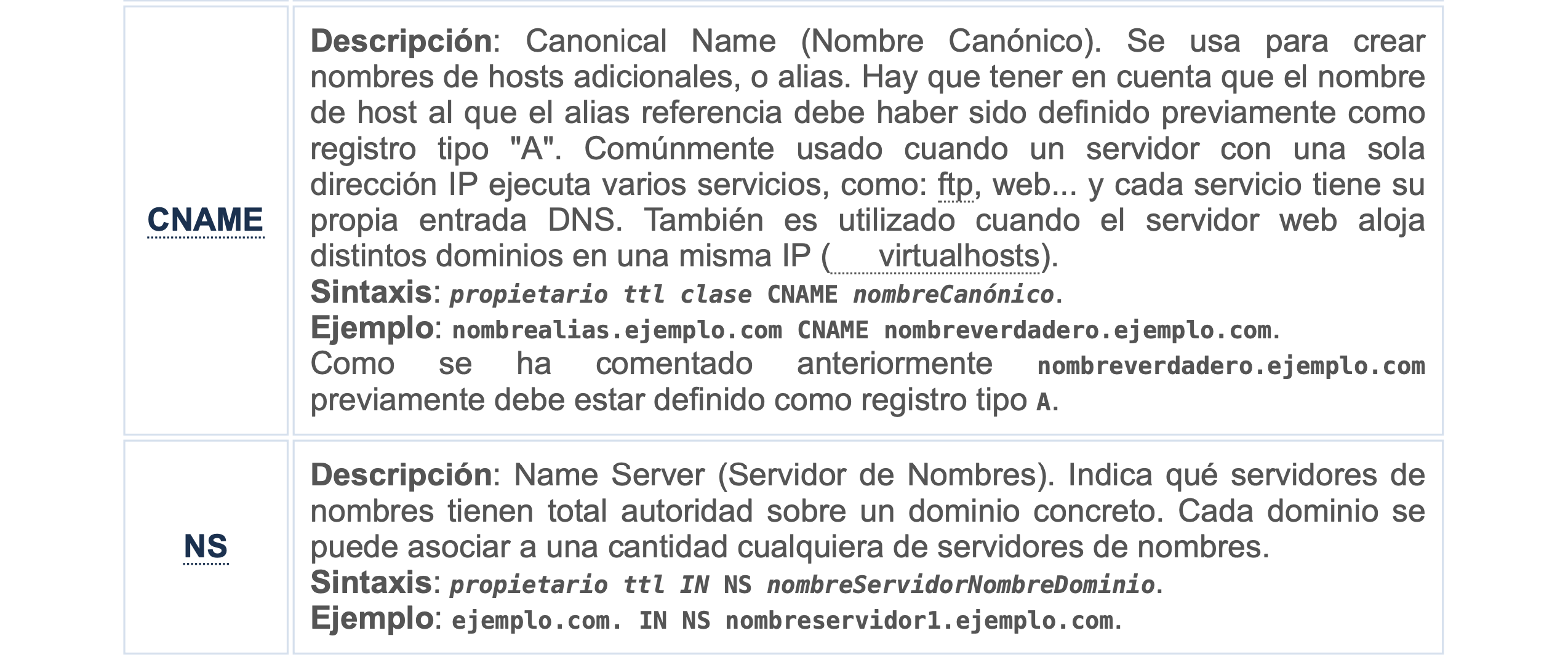
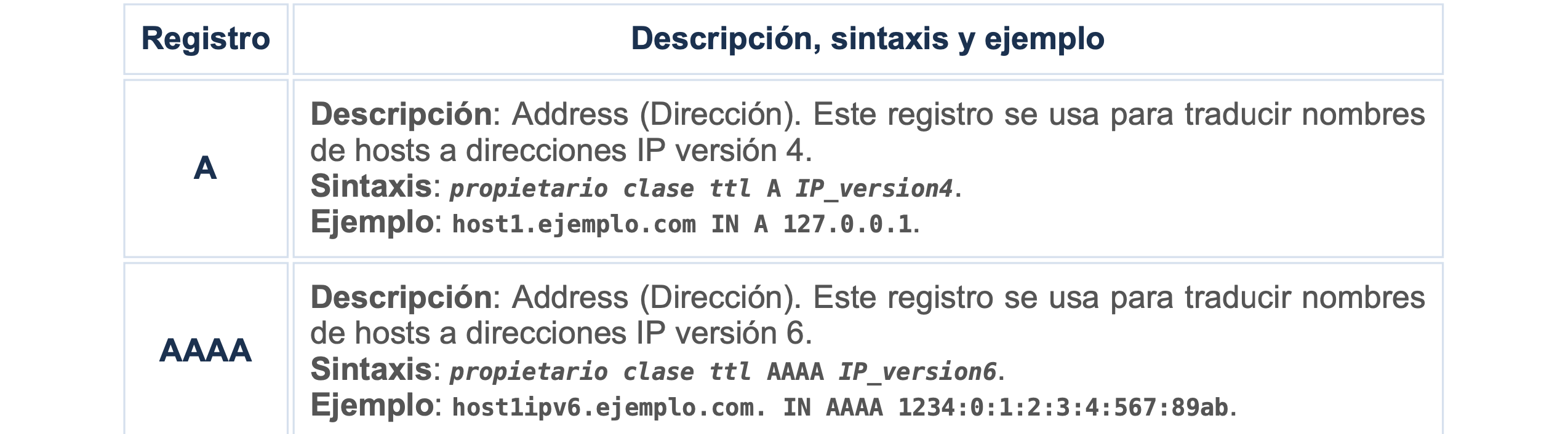
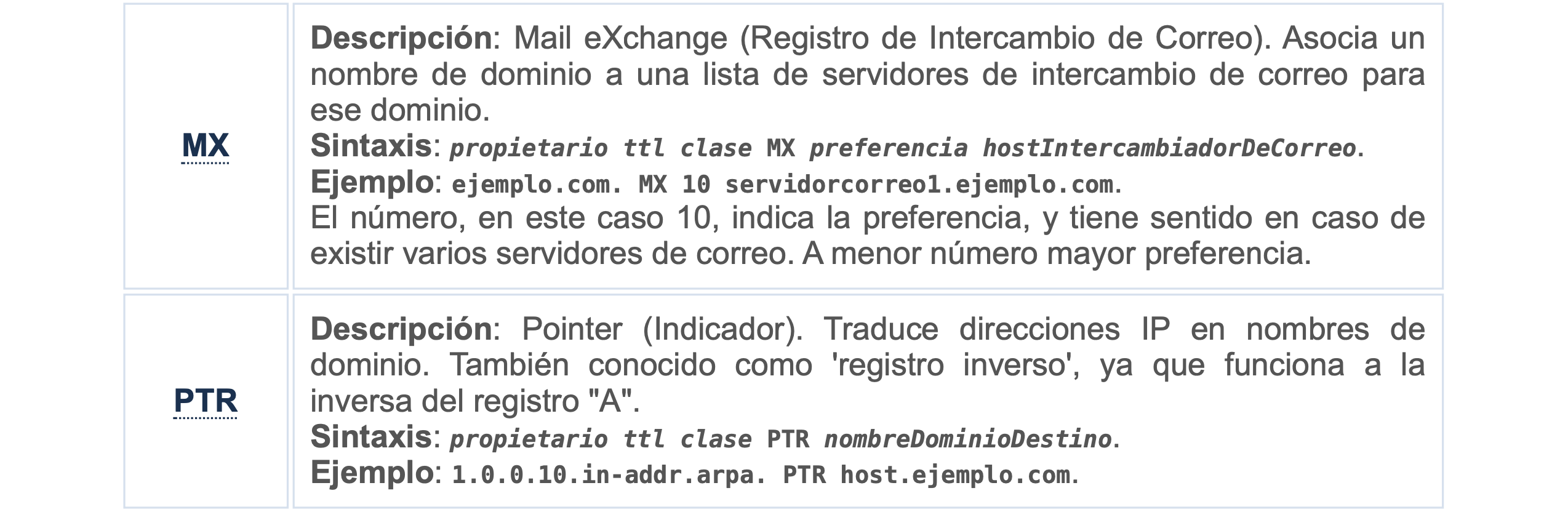
## 1.6.- Tipos de registro DNS.

Una base de datos DNS se compone de uno o varios archivos de zonas utilizados por el servidor DNS. Cada zona mantiene un conjunto de registros de recursos estructurados.

Todos los registros de recursos **(RR)** tienen un formato definido que utiliza los mismos campos de nivel superior.



En la siguiente tabla se muestran los registros DNS más utilizados:



## 1.7.- DNS Dinámico.

El DNS dinámico es un sistema que permite la actualización en tiempo real de la información sobre nombres de dominio situados en un servidor de nombres, siendo usado, mayoritariamente, para asignar un nombre de dominio de Internet a un ordenador con dirección IP variable (dinámica).

El DNS dinámico, así, puede ofrecer servicios en Internet en **hosts** que posean conexión con dirección IP dinámica, la típica configuración que los **ISP** ofrecen para conectarse a Internet.

Modificar el archivo **/etc/hosts** para añadir nuevas entradas de dominio.

## 1.8.- Cómo funcionan los DNS preferidos y alternativos.

El servidor DNS preferido es aquel con el que el cliente prueba en primer lugar. Si el servidor DNS preferido falla, el cliente prueba con el servidor DNS alternativo.

Opcionalmente, puedes especificar una lista completa de servidores DNS alternativos.

Los pasos siguientes indican el proceso para entrar en contacto con servidores DNS preferidos y alternativos:

1.- El servidor DNS preferido responde primero a una consulta DNS.

2.- Si el servidor DNS preferido no responde a una consulta DNS, la consulta se redirige al servidor DNS alternativo.

3.- Si el servidor DNS alternativo no responde y el cliente DNS está configurado con las direcciones IP adicionales de servidores DNS, el cliente DNS envía la consulta al siguiente servidor DNS de la lista.

4.- Si ninguno de los servidores DNS responden, la consulta del cliente DNS falla.

## 1.9.- Comandos

Comando **ping**, el cual indica, según su respuesta, si posees conectividad con la máquina en cuestión. Para consultar direcciones IP o nombres de dominios.

Comandos resolutores típicos en sistemas operativos GNU/Linux son: **nslookup**, **host** y **dig**.

En Debian / Ubuntu:

* sudo apt install dnsutils
* **sslookup** esta obsoleto
* **host** para consulta de direcciones IP y **dig** para consulta de servidores DNS activos.
* dig [@8.8.8.8](http://@8.8.8.8) [www.dawdistancia.net](http://www.dawdistancia.net) +answer +noall

## 1.10-Instalación del servidor DNS BIND

1. sudo apt update
2. sudo apt install bind9 bind9utils (crea el usuario bind y ejecuta el demonio)
3. sudo systemctl status bind9
4. sudo netstat –natp | grep named
5. sudo systemctl stop systemd-resolved
6. sudo systemctl disable system-resolved
7. sudo rm /etc/resolv.conf
8. sudo nano /etc/resolv.conf
   1. nameserver 127.0.0.1
   2. El servidor DNS activo solamente es el local, configurado como caché.
9. Systemctl reload bind9

## 1.10.1- Archivos de configuración del servidor DNS.

La ruta de instalación de bind9 es **/etc/bind.**

Fichero de configuración principal **/etc/bind/named.conf.**

include "/etc/bind/named.conf.options";

include "/etc/bind/named.conf.local";

include "/etc/bind/named.conf.default-zones";

**/etc/bind/named.conf.options:** hace referencia al archivo de configuración que posee opciones genéricas.

**/etc/bind/named.conf.local:** hace referencia al archivo de configuración para opciones particulares.

**/etc/bind/named.conf.default-zones:** hace referencia al archivo de configuración de zonas.

Dentro de cada uno de estos archivos encontrarás partes de código agrupadas entre llaves que finalizan con el carácter punto y coma (;), conocidos como **declaraciones**, las cuales indicarán secciones de ejecución. Cualquier código en un archivo de configuración que comience con los caracteres doble barra (//), almohadilla (#) o aparezca encerrado entre barra asterisco (/\*) y asterisco barra (\*/) son considerados comentarios.

Se pueden crear nuevos ficheros de configuración tan solo tienen que ser llamadas mediante la directiva **include**.

Comandos de verificación:

* **named-checkconf** –p fichero (comprueba la sintaxis, p imprime por pantalla, fichero de configuración)
* **named-checkzone** zonename fichero (comprueba la sintaxis y la integridad de un archivo de zona, semántica, nombre de dominio de la zona, fichero de zona)

## 1.10.2.- Arranque y parada del servidor

Arrancar el servidor DNS:

* sudo systemctl start bind9

Parar el servidor DNS:

* sudo systemctl stop bind9

Comprobar el estado activo/inactivo del servicio:

* sudo systemctl status bind9

Otros comandos:

* reload
* restart
* force-reload

1.10.3.-Configuración como cache DNS.

**Todos los servidores DNS son servidores caché**, pero no por ello deben ser maestro o esclavo. Así, existe la posibilidad que un servidor DNS funcione solamente como servidor caché, sin que sea maestro o esclavo.

En el archivo **/etc/bind/named.conf.options**, donde se indica: el directorio de caché y los servidores DNS a reenviar las peticiones que no se pueden resolver de forma local mediante la caché: los servidores **forwarders**, para que luego estas consultas se vayan guardando en la caché.

El directorio de caché, **/var/cache/bind,** está configurado y habilitado por defecto tras la instalación y los servidores DNS a reenviar las peticiones que no se pueden resolver de forma local mediante la caché, los servidores forwarders, aparecen en una sección del mismo nombre y que **por defecto** está comentada, esto es, **deshabilitada**.

Para activar la caché debes realizar el siguiente procedimiento:

Verifica que el contenido del fichero **/etc/bind/named.conf.options**, tras la instalación.

El borrado de la caché DNS la puedes realizar en el cliente DNS y en el propio servidor DNS:

* sudo apt install nscd
* sudo /etc/init.d/nscd restart (borrado del cliente NameServerCacheDaemon)
* sudo /usr/sbin/rndc flush (borrado del servidor DNS BIND RemoteNameDaemonControl)

## 1.10.4.-Configuración como DNS maestro.

Puedes configurar un servidor DNS BIND como maestro modificando el archivo **/etc/bind/named.conf.local.**

1. Configuras el fichero **/etc/bind/named.conf.local** para indicar: qué zonas son servidas por el servidor, qué zonas son servidas como master y el fichero donde se guarda el contenido de la zona.

Normalmente los ficheros de zona están situados en la ruta **/var/lib/bind.**

|  |
| --- |
| //zonas creadas tipo master  zone "dawdistancia.net" {  type master;  file "/var/lib/bind/master/db.dawdistancia.net.hosts";  }; |

2. Configuras el fichero **/var/lib/bind/master/db.dawdistancia.net.hosts** para agregar los **registros RR** a la zona. Crea el directorio master si no está creado previamente.

|  |
| --- |
| ;  ; BIND Database file for dawdistancia.net zone  ;  $TTL 2d  $ORIGIN dawdistancia.net.  @ IN SOA ns.dawdistancia.net. hostmaster.dawdistancia.net. (  2011091610 ; serial number  12h ; refresh  15m ; retry  1w ; expire  1h ) ; default TTL  ;  IN NS ns1.dawdistancia.net.  IN NS ns2.dawdistancia.net.  IN MX 10 mail.dawdistancia.net.  IN TXT ( "v=spf1 mx ~all" )  ;  ns1 IN A 192.168.56.56  ns2 IN A 192.168.56.46  mail IN A 192.168.56.47  www IN A 192.168.56.56  ftp IN A 192.168.56.56 |

3. Recargas el servidor con el comando: **systemctl reload bind9** ó **service bind9 reload** ó **/etc/init.d/bind9 reload**.

4. Realizas la siguiente consulta: **nslookup www.dawdistancia.net** obteniendo una salida similar a la siguiente.

## 1.10.5.- Configuración como DNS esclavo.

Puedes configurar un servidor DNS BIND como esclavo modificando el archivo **/etc/bind/named.conf.local.**

1. Configuras el fichero **/etc/bind/named.conf.local** del servidor esclavo para indicar: qué zonas son servidas por el servidor, qué zonas son servidas como slave, la IP del servidor master -de donde se transferirá la zona cuando se reciba una notificación de cambio, o se supere el TTL de la zona- y el fichero donde se guarda el contenido de la zona.

|  |
| --- |
| //zonas creadas tipo esclavo  zone "dawdistancia.net" {  type slave;  masters {  192.168.56.56;  };  file "/var/lib/bind/slave/db.dawdistancia.net.hosts";  }; |

Es necesario crear el directorio slave en la ruta especificada en el parámetro file. Además es necesario cambiar el propietario y los permisos del directorio para que se puedan crear el fichero de configuración transferido desde el DNS primario. El propietario correcto será bind:

* sudo chown -R bind:bind /var/lib/bind/slave/
* sudo chmod -R 770 /var/lib/bind/slave/

2. En el servidor maestro configuras la sección correspondiente al servidor master en el fichero **/etc/bind/named.conf.local:**

* Para indicar qué servidores tienen permitido la transferencia de los ficheros de zona, mediante la directiva allow-transfer: allow-transfer {192.168.56.46;};
* En este listado deberán estar incluidos todos los servidores **slave** que tengan configurado a éste como servidor **master**, y adicionalmente alguna IP que debiera tenerlo permitido por alguna razón.

3. Mediante la directiva **notify-yes** se consigue enviar automáticamente una notificación de cambio de zona del maestro, cuando ésta se produce, a los servidores DNS especificados en la zona mediante el registro de recurso NS. Adicionalmente, se puede enviar una notificación de cambio de zona a servidores esclavos que no aparecen en la misma, mediante la directiva **also-notify.**

**Mediante la directiva also-notify se mantienen los servidores DNS sincronizados.**

**Además, en caso de recibir múltiples conexiones concurrentes, siendo, por tanto, el número de peticiones muy elevado, la carga se distribuye entre los servidores.**

|  |
| --- |
| //zonas creadas tipo master  zone "dawdistancia.net" {  type master;  file "/var/lib/bind/master/db.dawdistancia.net.hosts";  allow-transfer {  192.168.56.46;  };  notify yes;  also-notify {  192.168.56.46;  };  }; |

4. Para comprobar si funciona la transferencia de zona después de una modificación del fichero de configuración de zona puedes realizar las siguientes operaciones:

* Modifica algún dato del fichero de configuración de zona e incrementa el número de serie en el registro SOA.
* Recarga el fichero de configuración de zona modificado en bind9 con el siguiente comando:
* sudo rndc reload dawdistancia.net.

5. Comprueba que en el servidor esclavo se ha transferido el fichero del servidor DNS primario comprobando el estado de la zona con el comando:

* rndc zonestatus dawdistancia.net.

# 2.- Servicio de directorio

Los directorios anteriormente comentados presentan una serie de problemas, en contra de los directorios electrónicos, a saber:

- **Son estáticos:** la información no se actualiza

Por contra, los directorios electrónicos pueden ser consultados/actualizados en tiempo real, por lo que su fiabilidad es mucho mayor.

- **Son inflexibles:** en el contenido y en su organización.

Por contra, los directorios electrónicos pueden modificar cualquier contenido y éste se verá reflejado al instante.

Los directorios electrónicos permiten que la búsqueda de información sea localizada de distintas maneras, gracias a cómo está organizada.

- **Son inseguros:** dificultad para controlar el acceso a la información.

Los directorios electrónicos sí permiten controlar el acceso a la información: solamente aquel que disponga de las claves de acceso obtendrá la información.

- **Difícilmente configurable**

Por contra, los directorios electrónicos pueden establecer la información que recibe una persona en función de sus necesidades.

## 2.1.-¿Para que usar un servicio de directorio?

Por lo visto anteriormente los directorios electrónicos permiten, de forma eficiente:

**1.- Encontrar información:**

Los directorios electrónicos a diferencia de los clásicos permiten acceder a la información contenida en los mismos de múltiples formas.

**2.- Gestionar información:**

En los directorios electrónicos pueden existir varios usuarios que en tiempo real estén realizando modificaciones, como agregar/editar/eliminar distintos usuarios con sus correspondientes campos. Además, esta información ya estaría visible para todas aquellas aplicaciones que accedan a la misma. Centralizar así los datos en un directorio evita tener que sincronizar varios directorios, con el consiguiente riesgo que esto provoca.

**3.- Control de seguridad:**

Los servicios de directorios no simplemente permiten delimitar el acceso a los usuarios, sino que también proporcionan una solución al problema de gestión de certificados digitales. Así, permiten:

3.1.- Su creación: Incorporar a los certificados los datos contenidos en el directorio.

3.2.- Su distribución: Tener accesibles mediante un protocolo estándar los certificados.

3.3.- Su destrucción: Revocar los certificados de forma sencilla simplemente borrando el certificado del directorio.

3.4.- Su ubicación: Los usuarios pueden acceder a través del directorio a los certificados de los restantes usuarios, de forma muy sencilla y fácil de integrar con las aplicaciones.

## 2.2.-Directorio vs DNS

Tanto un servicio de directorio como un servicio DNS proporcionan acceso a una base de datos jerárquica, pero difieren en:

1.- Los servidores de directorio no están particularizados a una acción concreta sino orientados de forma más general, mientras que el servicio DNS está dedicado a la traducción de nombres de dominios a direcciones IP.

2.- La información almacenada en el servicio de directorio no es fija, mientras que en el servicio DNS tiene una estructura fija.

3.- El servicio de directorio permite actualizaciones, mientras que el servicio DNS no las permite.

4.- Los servicios de directorio suelen utilizar protocolos orientados a conexión (**TCP**), mientras que el servicio DNS opera con protocolos no orientados a conexión (**UDP**).

Normalmente los encontrarás unidos de la mano en aplicaciones web con distintas funcionalidades, como: servidores de correo, gestión de proyectos e incidencias, servidores RADIUS, etc. Así, suele ser necesario acceder a las URL de las aplicaciones web mediante nombres de dominio DNS y una vez en ellas autenticarse por medio de LDAP.

## 2.3.- Organización del directorio LDAP.

El servicio de directorio puede estar centralizado o distribuido:

* **Centralizado**: En este caso un único servidor ofrece todo el servicio de directorio respondiendo a todas las consultas de los clientes.
* **Distribuido**: Si el directorio está distribuido, varios servidores proporcionan el servicio de directorio. Cuando está distribuido, los datos pueden estar fraccionados y/o replicados:
  + Cuando está fraccionada, cada servidor de directorio almacena un subconjunto único y no solapado de la información, es decir, una entrada es almacenada en un solo servidor.
* Cuando la información está replicada, una entrada puede estar almacenada en varios servidores.

Generalmente cuando el servicio de directorio es distribuido, parte de la información está fraccionada y parte está replicada.

En 1988, la CCITT (ahora ITU-T) creó el estándar X.500 sobre servicios de directorio, el cual organiza las entradas en el directorio de manera jerárquica, capaz de almacenar gran cantidad de datos, con grandes capacidades de búsqueda y fácilmente escalable. X.500 especifica que la comunicación entre el cliente y el servidor de directorio debe emplear el protocolo DAP, pero DAP es un protocolo a nivel de aplicación, por lo que, tanto al cliente como el servidor debían implementar completamente la torre de protocolos OSI.

LDAP surge como una alternativa a DAP. Las claves del éxito de LDAP en comparación con DAP de X.500 son:

* El modelo funcional de LDAP es más simple y ha eliminado opciones raramente utilizadas en X.500, siendo más fácil de comprender e implementar.
* LDAP representa la información mediante cadenas de caracteres en lugar de complicadas estructuras ASN.1.

El directorio LDAP tiene una estructura en forma de árbol denominado **DIT**. Cada entrada del directorio describe un **objeto**: persona, impresora, etc. La ruta completa a una entrada la identifica de modo inequívoco y se conoce como DN y está compuesto por una secuencia de partes más pequeñas llamadas **RDN**, de forma similar a como el nombre de un fichero consiste en un camino de directorios en muchos sistemas operativos.

Una **clase de objeto (objectClass)** es una descripción general de un tipo de objeto. Todos los objetos de LDAP deben tener el atributo objectClass. La definición de **objectClass** especifica qué atributos requiere un objeto LDAP, así como las clases de objetos que pueden existir. Los valores de este atributo los pueden modificar los clientes, pero el atributo objectClass en sí no puede eliminarse.

Un **esquema (schema)** define: qué clases de objetos se pueden almacenar en el directorio, qué atributos deben contener, qué atributos son opcionales y el formato de los atributos.

Por lo general, existen **dos tipo**s de objetos:

* **Contenedor**: Este tipo de objeto puede contener a su vez otros objetos. Algunos ejemplos de estos elementos son: **Root** (elemento raíz del árbol de directorios que no existe en realidad), **c** (country), **ou** (OrganizationalUnit) y **dc** (domainComponent).

La figura análoga al contenedor es el directorio (carpeta) de un sistema de archivos.

* **Hoja**: Este tipo de objeto se encuentra al final de una rama y carece de objetos subordinados. Algunos ejemplos son: **Person/InetOrgPerson** o **groupofNames**.

En la cúspide de la jerarquía del directorio se encuentra el elemento raíz Root. A este elemento le puede seguir en un nivel inferior c (country), dc (domainComponent) ó o (organization).

La siguiente imagen ilustra las relaciones jerárquicas dentro de un árbol de directorios LDAP:

dn: uid=upruebas,ou=People,dc=ejemplo,dc=com

La definición global de qué tipo de objetos han de guardarse en el **DIT** se realiza mediante un esquema. El tipo de objeto se determina mediante la clase de objeto. La clase de objeto especifica qué atributos deben o pueden ser asignados a un objeto determinado. Por lo tanto, un esquema debe contener definiciones de todas las clases de objetos y atributos que van a utilizarse en el escenario de aplicación. Existen algunos esquemas de uso extendido (**véase RFC 4512 y 4519**). No obstante, si el entorno en el que va a utilizarse el servidor LDAP lo requiere, también pueden crearse nuevos esquemas en función del usuario o pueden combinarse varios esquemas entre sí.

## 2.4.- Integración del servicio de directorio con otros servicios.

De lo expuesto anteriormente puede deducirse que el servicio de directorio es importante en sí mismo, pero es fundamental para aglutinar información que puede ser fuente de objeto para desplegar nuevos servicios basados en la cooperación entre las distintas aplicaciones y el servicio de directorio.

Así, el servicio de directorio puede actuar como servidor de autenticación, proporcionando el servicio de contraseña única. Además puede contener información necesaria para que los distintos servidores puedan decidir si un usuario puede acceder a determinada información.

Puedes utilizar el servicio de directorio como repositorio en el cual almacenar la información que varios servidores deben compartir.

Además, el directorio proporciona un protocolo estándar para gestionar toda la información contenida en él evitando la necesidad de desarrollar dicho protocolo.

Emplear el servicio de directorio para indexar la documentación almacenada en el servidor Web, con la precisión que otras herramientas no pueden generar.

Debido a XML, los documentos contarán con metainformación, es decir, información sobre la información que contienen, lo cual hará más fácil y eficaz la labor de indexación de los contenidos del servidor Web.

## 2.5.- El formato de intercambio de datos LDIF.

El formato LDIF es el estándar para representar entradas del directorio en formato texto **(UTF-8 Y BASE64)**, que posee la siguiente sintaxis:

dn: <nombre distinguido>

<nombre\_atributo>: <valor>

<nombre\_atributo>: <valor>

<nombre\_atributo>: <valor>

Entonces, una entrada del directorio en formato de intercambio de datos LDIF consiste en **dos partes**:

* El DN que debe figurar en la primera línea de la entrada y que se compone de la cadena dn: seguida del **nombre distinguido** (DN) de la entrada.
* La segunda parte son los atributos de la entrada. Cada atributo se compone de un nombre de atributo, seguido del carácter dos puntos ':' y el valor del atributo. Si hay atributos multivaluados deben ponerse seguidos. No existe ningún orden preestablecido para la colocación de los atributos, pero es conveniente listar primero el atributo objectclass, para mejorar la legibilidad de la entrada.

En un archivo **LDIF** puede haber mas de una entrada definida, cada entrada se separa de las demás por una línea en blanco. A su vez, cada entrada puede tener una cantidad arbitraria de pares **<nombre\_atributo>: <valor>.**

Este formato es útil tanto para realizar copias de seguridad de los datos de un servidor LDAP, como para importar pequeños cambios que se necesiten realizar manualmente en los datos, siempre manteniendo la independencia de la implementación LDAP y de la plataforma donde esté instalada.

Un ejemplo de una entrada para describir una cuenta de usuario en un servidor:

|  |
| --- |
| # Usuario  dn: uid=upruebas,ou=usuarios,dc=proyecto-empresa,dc=local  objectClass: inetOrgPerson  objectClass: posixAccount  uid: upruebas  cn: usuario  sn: pruebas  loginShell: /bin/bash  uidNumber: 10001  gidNumber: 678  homeDirectory: /home/upruebas  gecos: Pruebas DAW05  userPassword: 123456  mail: upruebas.daw05@proyecto-empresa.local |

## 2.6.- Instalación de OpenLDAP

* Actualiza los repositorios del sistema operativo.
  + sudo apt update
* Instalación de los paquetes necesarios para el funcionamiento de OpenLDAP
  + sudo apt install slapd ldap-utils
    - admin
    - admin
* Verifica que el servidor OpenLDAP está activo puerto **TCP 389.**
  + sudo netstat –natp | grep 389

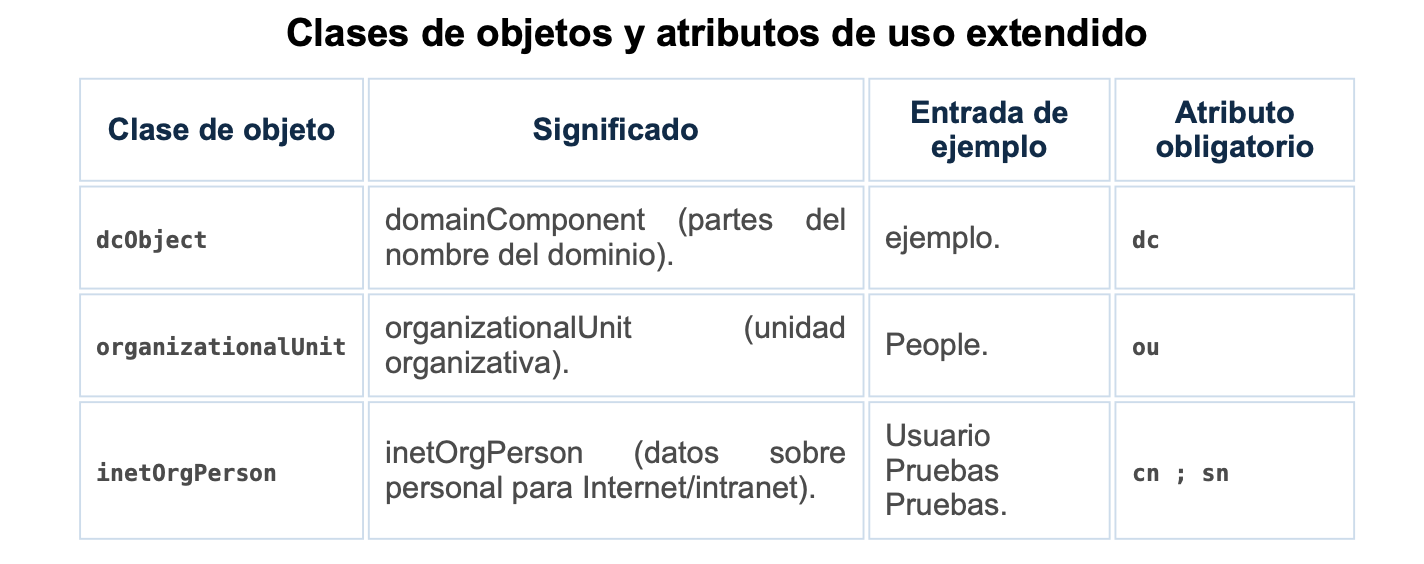
## 2.6.1.-Configuración de OpenLDAP

Configuración del servidor slapd en un directorio de base cn=config, **(/etc/ldap/slapd.d/cn=config)**.

Esto tiene la ventaja de que las modificaciones de configuración se pueden hacer sin tener que reiniciar el servicio.

Dentro del directorio **/etc/ldap/slapd.d/cn=config,** en una instalación limpia, puedes observar el objeto cn=schema, donde se encuentran los cuatro esquemas instalados por defecto: **core**, **cosine**, **nis** e **inetorgperson**.

Puedes encontrar más esquemas dentro de **/etc/ldap/schema**. Para añadir un esquema nuevo al directorio hay que subir un fichero ldif con el nuevo esquema al dn: cn=schema,cn=config.



El árbol completo LDAP se genera a partir de archivos esquema, en **/etc/ldap/schema**, que definen el árbol de clases y atributos permitidos para la organización.

1. Configura el servidor OpenLDAP mediante el comando **dpkg-reconfigure slapd.** Los valores utilizados los puedes ver a continuación del comando:

|  |
| --- |
| $ sudo dpkg-reconfigure slapd  ¿Desea omitir la configuración del servidor OpenLDAP? No  Introduzca su nombre de dominio DNS: dawdistancia.net  Nombre de la organización: dawdistancia.net  Contraseña del administrador: admin  Verificación de la contraseña: admin  ¿Desea que se borre la base de datos cuando se purgue el paquete slapd? No  ¿Desea mover la base de datos antigua? Sí |

## 2.6.2.-Arranque y parada del servidord LDAP.

* Comprobar las opciones del comando:
  + sudo systemctl <start/stop/reload/restart/force-reload/status> slapd
* Comando: **slaptest –f file** permite verificar la configuración del servidor OpenLDAP.
* Herramienta gráfica: jxexplorer.

## 2.6.3.- Administrando un servidor LDAP.

Antes de presentar ejemplos de las utilidades LDAP vamos a listar el significado de las posibles opciones que pueden acompañar a los comandos:

-x - Se usa un tipo de autenticación simple en vez de SASL

-h - Indica el host donde reside el servicio de directorio

-p - Indica el número de puerto

-D - Indica el "Nombre Distinguido" o DN del usuario con el que se realiza la operación

-w - Indica la clave de dicho usuario

-b - La base DN a partir de la que se hace la búsqueda

-s - El ámbito de la búsqueda

-f - El fichero .ldif utilizado para ejecutar el comando

1. **Añadir entradas:** comando **ldapadd**.

1.a. **Crea** la estructura básica del dominio LDAP mediante la ejecución de un fichero estructura\_basica.ldif.

|  |
| --- |
| # Usuarios  dn: ou=usuarios,dc=dawdistancia,dc=net  objectClass: organizationalUnit  ou: usuarios  # Grupos  dn: ou=grupos,dc=dawdistancia,dc=net  objectClass: organizationalUnit  ou: grupos |

ldapadd -x -D cn=admin,dc=dawdistancia,dc=net -w admin -f estructura\_basica.ldif

1.b. **Añade** un grupo a LDAP con el nombre pruebas mediante el archivo grupo.ldif

|  |
| --- |
| # Grupo  dn: cn=pruebas,ou=grupos,dc=dawdistancia,dc=net  objectClass: top  objectClass: posixGroup  gidNumber: 678  cn: pruebas |

$ ldapadd -x -D cn=admin,dc=dawdistancia,dc=net -w admin -f grupo.ldif

1.c. **Añade** dos usuario a a LDAP, el primero de nombre upruebas y contraseña 123456 y el segundo de nombre u2pruebas y contraseña abcdef mediante el archivo **usuario.ldif**

|  |
| --- |
| # Usuario  dn: uid=upruebas,ou=usuarios,dc=dawdistancia,dc=net  objectClass: inetOrgPerson  objectClass: posixAccount  uid: upruebas  cn: usuario  sn: pruebas  loginShell: /bin/bash  uidNumber: 10001  gidNumber: 678  homeDirectory: /home/upruebas  gecos: Pruebas DAW05  userPassword: 123456  mail: upruebas@dawdistancia.net  dn: uid=u2pruebas,ou=usuarios,dc=dawdistancia,dc=net  objectClass: inetOrgPerson  objectClass: posixAccount  uid: u2pruebas  cn: usuario2  sn: pruebas  loginShell: /bin/bash  uidNumber: 10002  gidNumber: 678  homeDirectory: /home/u2pruebas  gecos: Pruebas DAW05  userPassword: abcdef  mail: u2pruebas@dawdistancia.net |

$ ldapadd -x -D cn=admin,dc=dawdistancia,dc=net -w admin -f usuario.ldif

2. **Modificar entradas:** comando **ldapmodify**.

2.a. Modificar la contraseña del usuario anterior 'pruebas' mediante la ejecución del archivo **cambiar\_usuario.ldif**

|  |
| --- |
| # Cambiar contraseña Usuario  dn: uid=upruebas,ou=usuarios,dc=dawdistancia,dc=net  changetype: modify  replace: userPassword  userPassword: 654321 |

$ ldapmodify -x -D cn=admin,dc=dawdistancia,dc=net -w admin -f cambiar\_usuario.ldif

3. **Buscar entradas:** comando **ldapsearch**.

3.a. Buscar todas las entradas cuyo atributo cn (common name) contenga los caracteres 'pru':

$ ldapsearch -x -b dc=dawdistancia,dc=net "(cn=\*pru\*)"

3.b. Buscar todos los usuarios cuyo nombre contenga los caracteres 'pru' y cuyo correo electrónico contengan los caracteres 'daw05':

$ ldapsearch -x -b dc=dawdistancia,dc=net "(&(uid=\*pru\*)(gidNumber=678))"

4. **Eliminar entradas:** comando **ldapdelete**.

a. Eliminar el usuario upruebas:

$ ldapdelete -x -D cn=admin,dc=dawdistancia,dc=net -w admin uid=u2pruebas,ou=usuarios,dc=dawdistancia,dc=net

Los comandos anteriores poseen la opción **-h** con la cual se puede indicar el host (nombre de dominio o IP) que identifica al servidor LDAP. Por ejemplo: ldapsearch

-h 192.168.56.56 -x -b dc=dawdistancia,dc=net "(objectclass=\*)"

conectaría con el servidor LDAP en la IP 192.168.56.56 para buscar el **DIT** del dominio **dawdistancia.net.**.

Existe un paquete de nombre **ldapscripts** que contiene una serie de scripts para administrar de forma sencilla los usuarios y grupos almacenados en el servidor LDAP. Puedes encontrar plantillas de ejemplo, formato LDIF, situadas en **/usr/share/doc/ldapscripts/examples/** cuando se instala el paquete **ldapscripts**.

## 2.6.4.- Configuración de los clientes. Instalación de librerías de autentificación.

Utilidad más importante la de servidor es la de autentificación.

Esta configuración debe ser replicada en todos los clientes LDAP pertenecientes al dominio, incluido el propio servidor LDAP, si se quiere que los clientes accedan al mismo.

Para ello realiza el siguiente procedimiento:

1.- Instala y configura los paquetes libnss-ldap, libpam-ldap y nscd:

sudo apt-get install libnss-ldap libpam-ldap

|  |
| --- |
| URI del servidor de LDAP: ldap://192.168.56.56  El nombre distintivo (DN) de la base de búsquedas: dc=dawdistancia,dc=net  Versión de LDAP a utilizar: 3  Make local root Database admin: yes  Does the LDAP database require login; No  Cuenta LDAP para root: cn=admin,dc=dawdistancia,dc=net  Contraseña para la cuenta LDAP de root: admin |

Toda esta configuración se ha guardado en el fichero **/etc/libnss-ldap.conf**

2.- Modifica en el archivo **/etc/nsswitch.conf**:

|  |
| --- |
| passwd: files ldap  group: files ldap  shadow: files ldap |

3.- Revisa mediante el comando **pam-auth-update** que los servicios: Unix authentication y LDAP Authentication, que el sistema operativo usa para autenticar los usuarios, están activados. Para establece la opción usa la barra espaciadora o "space bar".

sudo pam-auth-update

4.- Por último, prueba que la configuración del cliente es correcta:

4.1.- Mediante el comando **getent passwd**, que proporciona todos los usuarios del sistema operativo, en este caso los de Unix authentication y LDAP Authentication.

sudo getent passwd | grep upruebas

4.2.- Iniciar sesión en un consola de texto en el equipo cliente con un usuario del LDAP. En esta caso, con el usuario **upruebas**.

## 2.6.5- Probar la autenticación con pamtest.

Instalación del comando **pamtester**:

* sudo apt install pamtester

Acepta dos parámetros: el primero es el nombre del servicio al cual se va a conectar para realizar la autenticación y el segundo es el nombre del usuario que se va a autenticar sobre dicho servicio.

* Intenta autenticar al usuario upruebas en el servicio passwd mediante una clave correcta:
  + sudo pamtester passwd upruebas authenticate
* Intenta autentificar al usuario upruebas en el servicio passwd mediante una clave incorrecta:
  + sudo pamtester passwd upruebas authenticate
* Intentar autentificar al usuario upruebas2 en el servicio ssh mediante una clave correcta:
  + sudo pamtester ssh upruebas authenticate
* Intentar autentificar al usuario upruebas2 en el servicio ssh mediante una clave incorrecta:
  + sudo pamtester ssh upruebas authenticate
* Intenta autentificar al usuario upruebas2 en el servicio ftp mediante una clave correcta:
  + sudo pamtester ftp upruebas authenticate
* Intentar autentificar al usuario upruebas2 en el servicio ftp mediante una clave incorrecta:
  + sudo pamtest ftp upruebas authenticate

Una vez se ha llegado a este punto, el sistema ya está preparado para autentificar a los usuarios a través de LDAP.

<https://alinuxaday.wordpress.com/2015/12/27/como-instalar-y-configurar-openldap-en-gnulinux/>

Fin.