

Servicios de red implicados en el despliegue de una aplicación web



Caso práctico

Programación, como cada lunes, tiene lugar una reunión de seguimiento, la responsable del proyecto es **Ada**, la responsable del Área de Desarrollo, en la que se evalúan el estado actual de los proyectos en desarrollo o a desarrollar. En esta caso, la reunión fue la siguiente:

—He estado viendo las tareas relativas al proyecto en la aplicación de proyectos Redmine y veo que llevamos adelanto sobre lo previsto.

—Sí, **Ada** —dijo **María**—, ya tenemos el proyecto encauzado, solamente quedan por resolver dos tareas: configurar la visibilidad a través de Internet para la aplicación web y la autenticación de usuarios. La verdad es que ha sido todo un acierto el empleo de Redmine para llevar a buen fin el proyecto. Se deja manejar muy bien y permite coordinarse. Ahora la verdad es que tenemos un control exhaustivo sobre el proyecto.

—Sí —reafirmó **Juan**—. La verdad es que poder monitorizar las tareas y ver su ciclo de vida es todo un acierto. A todo esto, entonces, ¿cuándo podemos empezar con las pruebas en el servidor definitivo?

—Pues, pronto —dijo **María**—. Tan pronto como tengamos realizada la primera de las dos tareas, esto es, tan pronto como hayamos configurado la redirección DNS de la aplicación web al servidor destinado al proyecto.

—Bien, —dijo **Juan**—. Entonces iré ya creando las tareas respectivas en el Redmine, asignándolas a los responsables correspondientes, para las pruebas.

—Sí, estaría bien, ya que lo que nos estaba reteniendo en la tarea relativa al DNS era la elección del dominio DNS por parte del cliente, puesto que no lo tenía claro. Una vez resuelto, solamente debemos configurar el servidor DNS para apuntar el nombre de dominio DNS a la IP del servidor destinado al proyecto y verificar que la configuración se replica en el servidor esclavo. Por lo tanto a configurar el servidor DNS **BIND** y listo.

—Por cierto, ¿cuál es el sistema de autenticación elegido por el cliente? —preguntó **Juan**—.



Stockbite.ADA (CC BY-NC)

—Ha elegido autentificación por LDAP —dijo **María**—. Al final se ha decidido por el montaje de un servidor  LDAP frente a la otra opción considerada: una base de datos SQL. Así que lo configuraremos con  **OpenLDAP**.

—Muy bien —dijo **Ada**—, veo que el proyecto va viento en popa, esperemos que continúe.

—Por mi parte —dijo **María**—, lo único que podría ralentizar el proyecto sería el envío de usuarios para darlos de alta en OpenLDAP, por lo demás...

—Bien, si hasta ahora el cliente ha sido efectivo en plazos no hay porque suponer que no siga continuando siéndolo. En fin, manos a la obra. Volvemos a quedar la próxima semana a la misma hora, la sala ya está reservada, y comentamos de nuevo.

—Vale, adiós —dijo **María**—.

—Hasta luego —dijo **Juan**—.

1.- Introducción

¿Te has preguntado alguna vez dónde y de qué manera se almacenan y procesan los datos o ficheros que utilizamos cuando consultamos una página web?

Las Aplicaciones Web son las relacionadas con esta gestión de datos en servidores o en "la nube". Estas aplicaciones reciben este nombre porque se ejecutan desde Internet y no necesitan una instalación previa.

En este tema vamos a ver los distintos servicios de red implicados en el despliegue de una aplicación web, desglosado en:

- ✓ **Servicios de nombre de dominio**, cuyas siglas son **DNS**. Vamos a ver su funcionamiento, tipos de servidores y registros.
- ✓ **Servicio de Directorio**, principalmente en la organización **LDAP** tipo árbol. Además realizaremos la instalación y configuración de OpenLDAP.
- ✓ Monitorización de los sistemas, a través de herramientas específicas de los servicios web y otras más avanzadas, pero muy interesante conocer.



Galopicoita Servicios de Red CC BY-NC

Es fundamental el uso de ciertos servicios para la utilización de las distintas aplicaciones web así como luego ser capaces de ver y realizar seguimiento del proceso que realizan a través de herramientas especializadas comprendiendo el proceso completo. De esta manera, debemos poder detectar posibles errores en la ejecución de dicho servicio si se producen, así como solventarlos.

2.- Servidores de nombres de dominio



Caso práctico

Para poder llevar a buen fin el proyecto, **María** se puso manos a la obra y determinó el siguiente escenario de trabajo para la realización de las dos últimas tareas del proyecto:

- ✓ Sistema Operativo Servidor: Debian GNU/Linux 6.0.
- ✓ Servidor Web: Apache (apache2).
- ✓ Servidor DNS Primario (Maestro): BIND (BIND9).
- ✓ Servidor DNS Secundario (Esclavo): BIND (BIND9).
- ✓ Servidor LDAP: OpenLDAP.
- ✓ Configuración de Red:
 - ⇒ Servidor Web: 192.168.200.250.
 - ⇒ Cliente de pruebas (desde donde se lanza el navegador): 192.168.200.100.
 - ⇒ Servidor DNS Maestro: 192.168.200.250.
 - ⇒ Servidor DNS Esclavo: 192.168.200.249.
 - ⇒ Servidor OpenLDAP: 192.168.200.248.



Stockbyte. María (CC BY-NC)

María, debe garantizar el correcto funcionamiento de la resolución DNS, por lo tanto, como con otros proyectos, prevé la redundancia del servicio mediante dos servidores DNS, uno actuando de primario y otro de secundario, y debido a sus características se ha decantado por el servidor DNS BIND.

Para la autenticación de usuarios, el cliente tras explicarle las alternativas se ha decantado por LDAP, por lo cual **María** configurará el servicio mediante OpenLDAP debido a sus características.



Reflexiona

¿Alguna vez te has parado a pensar qué es lo que pasa desde que escribes una dirección URL en el navegador hasta que puedes ver la página web cargada?

¿Sería posible acordarse de las páginas si tuviésemos que navegar a través de IP y no pudiéramos navegar a través de nombres?

¿Qué es lo que pasa si cambiásemos la redirección DNS a otro servidor? ¿Es automático el cambio? ¿Cuánto tiempo tarda? ¿Qué tiempo se necesita para activar los nuevos cambios?...

Internet funciona mediante el protocolo TCP/IP, efectuando conexiones mediante IP. ¿Qué quiere esto decir? Pues, que cada host en Internet se identifica mediante una IP, así es lo mismo visitar la página <http://www.rediris.es> que <http://130.206.13.20>

Entonces, ¿sería posible visitar cada página web conociendo su IP? Efectivamente, sólo que los seres humanos estamos más acostumbrados, a diferencia de las máquinas, a recordar nombres y no números. ¿Qué te es más fácil recordar el DNI de una persona o su nombre y apellidos? Por ello debe existir algo que nos traduzca los nombres a IP o viceversa. Sí, por supuesto, este algo no es otro que el **servidor DNS** o un archivo de texto, típicamente denominando hosts, como el archivo /etc/hosts en sistemas GNU/Linux.

¿Pueden convivir en una misma máquina un servidor DNS y el archivo /etc/hosts? Pues, sí. Pero hay que tener en cuenta la preferencia. Así, en caso de coexistir, 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.

2.1.- Sistema de nombres de dominio



Reflexiona

¿Cuántos servidores DNS existen?

¿Cuántas redirecciones DNS son posibles?

¿Existe un servidor DNS donde se guarden todos los dominios DNS posibles en Internet?

¿Qué son los servidores DNS Raíz?

¿Es necesario configurar un servidor DNS o se puede hacer la redirección mediante archivos de textos?

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.

Entonces, ¿para qué montar un servidor si simplemente escribiendo en un fichero la relación IP/Nombre el sistema ya funcionaría? Pues, realmente depende, ya que si estás pensando en pocos equipos a resolver el nombre de dominio la simplicidad del fichero /etc/hosts 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. Esto es, supón que posees 20 equipos que quieren resolver una página web, por ejemplo www.debian.org el procedimiento sería aproximadamente el siguiente:

1. Se escribe la página web en cada equipo en la barra de direcciones del navegador.
2. Se traduce el nombre DNS a una IP. ¿Cómo se produce esto? Pues, ahí está el quid de la cuestión: o bien mediante servidores DNS, o bien mediante ficheros estáticos /etc/hosts, con lo cual se debe modificar este fichero en cada cliente. Y esto, como bien puedes pensar, se hace arduo de manejar.

Pero, ¿y si la resolución tiene lugar mediante servidores DNS? ¿Por qué servidores DNS y no servidor DNS? Bien, existe, a modo de resumen, un procedimiento de resolución DNS, más o menos, similar al siguiente (encontrarás el procedimiento exacto un poco más adelante):

1. Primero, se debe averiguar qué servidor DNS resuelve el dominio raíz 'org' a una IP.
2. 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'.
3. 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'

Pero, entonces: ¿cuántos servidores DNS existen a la hora de preguntar? ¿Existen un número limitado de redirecciones de consultas? Y si se vuelve a hacer la misma consulta, hay que repetir el proceso?

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.

Pero, puede ser que no sea necesario escalar las consultas, puesto que todos los servidores DNS son servidores caché, lo que significa que recuerdan las consultas efectuadas. Por lo tanto, si se hace una consulta que ya está guardada en la caché, la respuesta es casi instantánea y ya ha sido resuelta. Es más, los equipos clientes, desde donde se hace la consulta a través del navegador como se indicaba en el ejemplo, también poseen una memoria caché DNS, de tal forma que anteriormente 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.



Debian [Debian](#) (Todos los derechos reservados)

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.



Ejercicio resuelto

Según lo expuesto, y si en tu configuración de red del sistema operativo solamente posees un servidor DNS, entonces: ¿cuál sería el proceso para encontrar la IP de la dirección web: <http://www.debian.org/distrib/netinst>?

[Mostrar retroalimentación](#)



Para saber más

Te proponemos que hagas un viaje por la siguiente página web donde se documenta los servidores raíz DNS.



[Página web donde se documenta los servidores raíz DNS.](#)

2.1.1.- ¿Cómo es un nombre de dominio?

¿Qué es lo que sueles escribir en la barra de direcciones URL del navegador? Normalmente algo similar a: www.debian.org. Entonces, vienen siendo unos caracteres separados por puntos. ¿Qué es lo que significan esos puntos? ¿Qué dividen? Además, en el ejemplo expuesto, al escribir www.debian.org el navegador autocompleta esta petición a http://www.debian.org, pero, ¿por qué?

Todas estas preguntas tienen respuesta, así que vamos a por ellas:



IETF IETF (Todos los derechos reservados)

- ✓ 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 autocompleta 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, por ejemplo:

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

Esto suele ser lo común, pero no es obligatorio, es decir, si una empresa posee un dominio **com** puede dedicarse al sector de redes de comunicaciones y no poseer el dominio **net**, así como puede ser una empresa localizada en España y no poseer el dominio **es**.

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.

Ahora bien, 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**.

La sintaxis de los nombres de dominio se discute en varios RFC (Request for Comments):

[RFC 1035](#)

[RFC 1123](#)

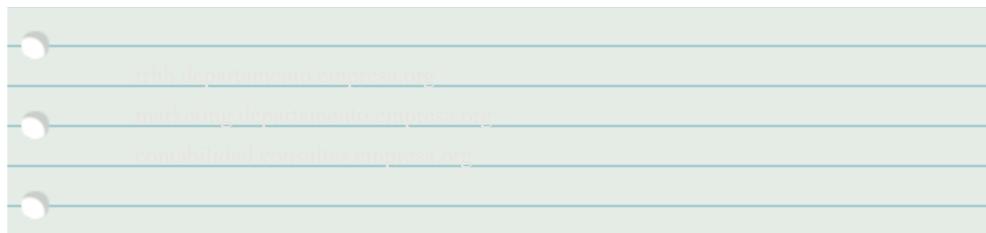
[RFC 2181](#)

2.1.2.- Jerarquía de nombres de dominio

El espacio de nombres de dominio (el universo de todos los nombres de dominio) está organizado de forma jerárquica. El nivel más alto en la jerarquía es el **dominio raíz**, que se representa como un punto (".") y el siguiente nivel en la jerarquía se llama **Dominio de Nivel Superior (TLD)**. Sólo hay un dominio raíz, pero hay muchos **TLD** y cada **TLD** se llama **dominio secundario del dominio raíz**. En este contexto, 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.

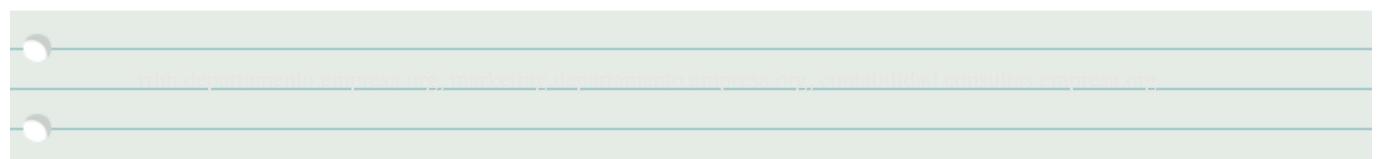
Por lo tanto el **DNS**, organiza los nombres de máquina (hostname) en una jerarquía de dominios separados por el carácter punto '.'.

Un **dominio** es una colección de nodos relacionados de alguna forma, porque están en la misma red, tal como los nodos de una empresa. Por ejemplo:



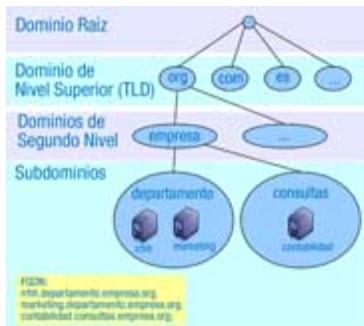
Donde:

- ✓ 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., tal como:



También es posible tener un dominio de cuarto nivel, dominio de quinto nivel, y así sucesivamente.

En la siguiente figura puedes ver una parte del espacio de nombres. La raíz del árbol, que se identifica con un punto sencillo, es lo que se denomina **dominio raíz** y es el origen de todos los dominios. Para indicar que un nombre es FQDN, a veces se termina su escritura en un punto, aunque por lo general se omite. Este punto significa que el último componente del nombre es el dominio raíz. En la imagen de la derecha (*Árbol DNS*) puedes observar un ejemplo de qué elemento forma cada nivel hasta llegar al nombre de dominio.



[rfc](#) Árbol DNS [CC BY-NC-SA](#)

[Resumen textual alternativo](#)

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.



Para saber más

En el siguiente enlace puedes encontrar una lista actualizada de los dominios de primer nivel existentes.



[Lista actualizada de los dominios de primer nivel existentes.](#)

2.2.- Ventajas del DNS



Citas para pensar

“

¿Qué sabe el pez del agua donde nada toda su vida?

Albert Einstein

¿Qué pasaría si dispones de 20 equipos y en todos actualizas una entrada DNS en el ~~file hosts~~, salvo en 3 de ellos? Sí, esos tres quedarían no actualizados. ¿Y si en la próxima actualización el cambio no se replica en otros 3, que pueden ser los mismos o no? ¿Y en la próxima ...?

Bien, parece que el sistema de modificar el archivo /etc/hosts no es muy buena idea, puesto que al ser cambios estáticos, más de un cambio puede quedar en el tintero, obteniendo al final un sistema no homogéneo. Así, parece claro que la solución, para obtener un sistema no heterogéneo es el DNS.

El DNS permite que cualquier cambio efectuado solamente en un servidor se replique en todos los servidores DNS que la configuración permita, de tal forma que el cambio sólo se efectúa en un servidor, obteniendo así facilidad y simplicidad en el cambio. Por lo tanto, cualquier cambio es dinámico: configuras solamente un servidor y éste se encarga de replicar el cambio.

Por otro lado, ¿qué es lo que pasa si un servidor DNS está caído y por lo tanto la conectividad con el mismo no es posible? ¿Quedaría todo el sistema inhabilitado? ¿Te podrías conectar aún a páginas web?

Bien, pues como 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. Es más, es posible que no solamente exista un servidor DNS configurado para controlar esa zona, y por lo tanto tampoco esa zona estuviese no visible.

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. Así, en los archivos de configuración de zona se indicará qué **IP** va con el servicio web **www**, el servicio de correo mail, etc. Los tipos de zonas posibles son dos:

- 1.- **Zona de Búsqueda Directa:** las resoluciones de esta zona devuelven la dirección **IP** correspondiente al recurso solicitado. Realiza las resoluciones que esperan como respuesta la dirección **IP** de un determinado recurso.
- 2.- **Zona de Búsqueda Inversa:** las resoluciones de esta zona buscan un nombre de equipo en función de su dirección **IP**; una búsqueda inversa tiene forma de pregunta, del estilo "¿Cuál es el nombre **DNS** del equipo que utiliza la dirección **IP** 192.168.200.100?"

Los servidores DNS no solamente sirven para la resolución de nombres en Internet, también se pueden utilizar en redes locales. Así, las entradas existentes en los DNS de la red local podrían ser visibles en Internet, o no, solamente sirviendo resolución a los equipos de la red local. De esta forma, cuando un usuario de la red local intenta acceder a un recurso local, podrá utilizar **nombres** 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 (que si estará en Internet) en su jerarquía de servidores DNS, hasta que la petición sea satisfecha.



ojimbo *Ventajas (CC BY-NC-ND)*

Por ejemplo, con un servidor DNS en nuestra red local, que resuelve la IP 192.168.200.100 a cliente.local y viceversa, puedes ejecutar el comando ping indistintamente contra dicha IP o contra el nombre del equipo en el dominio:



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.

Podemos resumir entonces las ventajas de la configuración y empleo de un servidor DNS en las siguientes:

- 1.- **Desaparece la carga excesiva en la red y en los hosts:** ahora la información está 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.

2.3.- Funcionamiento del DNS

La imagen animada que acompaña a este texto presenta gráficamente el funcionamiento del DNS, tomando como ejemplo la página web  www.debian.org y considerando que la información de la petición del dominio a buscar no se encuentra en tu ordenador o en un servidor DNS local existente en tu red o en tu ordenador.



Elaboración propia.Funcionamiento DNS:(CC BY-NC-SA)
[Resumen textual alternativo](#)

1. A través de tu navegador quieres consultar la página web oficial de Debian escribiendo en la barra de direcciones la URL  <http://www.debian.org>.
2. El navegador busca la información de las DNS del dominio debian.org.
3. Internet está ordenada en forma de árbol invertido, si no encuentra la información en tu ordenador, irá a buscarla a los servidores DNS que posees en la configuración de red de tu ordenador, típicamente los proporcionados por tu **Proveedor de Servicios a Internet (ISP)**: DNS Primario (3a) o DNS Secundario (3b). De no estar, seguirá buscándola a niveles superiores y, en último lugar, lo encontrará en el Servidor de Nombres Raíz: DNS Raíz (3n).
4. La información buscada: las IP correspondientes al servidor DNS que gobierna el dominio debian.org, llega a tu ordenador: DNS1→ 206.12.19.7 y DNS2→ 128.31.0.51. Suelen ser dos porque las especificaciones de diseño de DNS recomiendan que, como mínimo, deben existir dos servidores DNS para alojar cada zona, a la que pertenece cada dominio. Tu ordenador ahora intentará conectar con el servidor DNS1 (5a) o ante cualquier problema de conexión con éste lo intentará con el servidor DNS2 (5b). Éstos son los servidores de nombres donde se encuentra información acerca de dónde se puede buscar la página web (servidor de la web), una dirección de correo electrónico (servidor de correo), etc.
5. Tu ordenador recibirá la información acerca de la localización de la página web, o sea, la dirección IP del servidor web donde está alojada la página.
6. Tu ordenador se dirigirá luego al servidor web y buscará la página web en él.
7. Por último, el servidor web devuelve la información pedida y tú recibes la página web, visualizándola en el navegador.

Pero, y si vuelves a consultar la página web oficial de Debian escribiendo en la barra de direcciones la URL <http://www.debian.org>, ¿se repetirá de nuevo todo el proceso? Para contestar esta pregunta hay que establecer dos situaciones:

1. **El host desde el que vuelves a realizar la consulta es el mismo:** si no lo es, antes de repetir todo el proceso se intentaría con lo expuesto en el siguiente punto, pero si es el mismo, al haber hecho la consulta desde este host, la resolución dominio-IP se guarda durante algún tiempo en la memoria caché del mismo, por lo cual no será necesario repetir todo el proceso de nuevo. Si el tiempo en el que la memoria caché guarda la resolución ha expirado se volverá a repetir el proceso de nuevo.
2. **Existe un servidor DNS caché en tu red o en tu host:** por lo tanto, si un segundo cliente, que tiene configurado este servidor DNS, vuelve a realizar la misma petición, como este servidor tiene la respuesta almacenada en su memoria caché, responderá inmediatamente sin tener que cursar la petición a ningún servidor DNS de Internet. Si el tiempo en el que la memoria caché guarda la resolución ha expirado se volverá a repetir el proceso de nuevo.

2.4.- DNS Dinámico

¿Es posible si dispones de una conexión a Internet con IP dinámica ofrecer servicios en Internet?

Parece claro que si dispones de una IP estática de conexión a Internet, previo pago de un plus por disponer siempre de una misma IP para tu conexión a Internet, simplemente deberías enrutar las peticiones de los servicios que ofreces a los hosts que esperan la conexión a esos servicios. Si además, posees nombres de dominios puedes redireccionar esos nombres a las IP de tus hosts a través del servidor DNS.

Pero, volviendo a la pregunta, ¿qué es lo que pasa si quieres hacer lo mismo y no dispones de IP estática? Esto es, cada vez que te conectas a Internet tu IP, aunque a veces sea la misma, no siempre es la misma. Pues sí, sí es posible. ¿Cómo? A priori, si lo piensas un poco, lo único que necesitarías sería:

1. Recoger la IP de tu conexión cada vez que te conectas en Internet.
2. Una vez recogida tu IP difundirla en Internet. Para difundirla, o bien lo haces de forma estática, y cada vez que la recoges te preocupas de hacer los cambios necesarios para difundirla, o bien de forma dinámica configuras un programa para que automáticamente recoja la IP y la difunda.

Está claro, que la mejor opción es difundirla de forma dinámica, para ello puedes aprovecharte de servicios ofrecidos, incluso de forma gratuita, por  **DynDNS**,  **No-IP** y  **FreeDNS**. De hecho, hoy en día, los routers que los ISP suelen montar ya poseen la opción de configuración por DNS dinámica.

Entonces, 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.

De todos modos, aunque existe la posibilidad de ofrecer servicios en Internet desde tu propia casa, debes tener en cuenta que, usualmente, la infraestructura técnica y la electrónica de red que poseas no se pueda comparar con los servidores ofrecidos por empresas de Hosting, así:

- ✓ ¿Posees balanceadores de carga?
- ✓ ¿Redundancia en caso de fallos?
- ✓ ¿Generadores eléctricos que garanticen conexión eléctrica permanente a pesar de caída eléctrica?
- ✓ Y sobre todo, ¿dispones del ancho de banda necesario para permitir múltiples conexiones concurrentes sin perjudicar el servicio ofrecido?



[no-ip](#) No-ip (Todos los derechos reservados)

Si no tienes configurado un servidor DNS con las entradas de dominio necesarias, puedes generar estas entradas modificando el archivo /etc/hosts, añadiéndolas al final del mismo:

- ✓ #IP nombre-dominio
- ✓ 192.168.200.250 empresa1.com www.empresa1.com
- ✓ 192.168.200.250 empresa2.com www.empresa2.com

Cada campo, de cada entrada, puede ir separado por espacios o por tabulados.

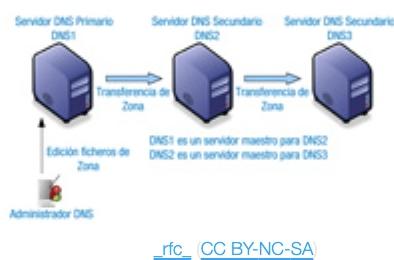
Estas entradas solamente serán efectivas en el equipo en el que se modifique el archivo /etc/hosts. Así, debes modificar el archivo /etc/hosts en cada equipo que quieras que se resuelvan esas entradas.

2.5.- Tipos de servidores DNS

Como puedes comprobar en la imagen de la derecha, existen varios tipos de servidores DNS que describiremos a continuación.

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

- ✓ **Servidores primarios** (primary name servers). 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.
- ✓ **Servidores secundarios** (secondary name servers). También denominados *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*.
- ✓ **Servidores maestros** (master name servers). 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. Así, se evita que los servidores secundarios sobrecarguen al servidor primario con transferencias de zonas. Por ejemplo, en la imagen el servidor DNS3 pide la zona al servidor DNS2 y no al servidor DNS1, con lo cual se evita la sobrecarga del servidor DNS1. Los servidores maestros extraen la información desde el servidor primario de la zona.
- ✓ **Servidores sólo caché** (caching-only servers). Los servidores sólo caché no tienen autoridad sobre ningún dominio: se limitan a contactar con otros servidores para resolver las peticiones de los clientes DNS. Estos servidores 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 IP 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`

donde: **retransfer** → indica que la acción a realizar es una transferencia.

proyecto-empresa.local → es el nombre de la zona que quieras transferir.

2.6.- Servidores raíz

La organización que gestiona globalmente los servidores raíz por concesión del gobierno estadounidense es la ICANN (Corporación para la Asignación de Nombres y Números de Internet), la cual es una organización sin ánimo 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 (Autoridad de Números Asignados de Internet) 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 única 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.

Otros asuntos que preocupan a los usuarios de Internet, como reglamentación para transacciones financieras, control del contenido de Internet, correo electrónico de publicidad no solicitada (SPAM) y protección de datos, están fuera del alcance de la misión de coordinación técnica de ICANN.



ICANN. ICANN (Todos los derechos reservados)



Para saber más

En el siguiente enlace puedes encontrar más información sobre ICANN.

 [ICANN.](#)

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. Esta iniciativa permitirá que las direcciones de los dominios puedan terminar con el nombre de compañía, ciudad, etc., en vez de .com; o .org.

"ICANN ha abierto el sistema de direcciones de Internet a las ilimitadas posibilidades de la imaginación humana. Nadie puede saber a dónde nos llevará esta histórica decisión", dijo Rod Beckstrom, presidente y jefe ejecutivo de la organización.

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 (los servidores que tienen una de las 13 direcciones IP pueden encontrarse en docenas de ubicaciones físicas distintas). 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 encontrarse la propia agenda de direcciones de ese registro.

En realidad, los servidores raíz no se consultan con mucha frecuencia (considerando el tamaño de Internet) porque una vez que los ordenadores de la red conocen la dirección de un dominio de nivel principal concreto pueden conservarla, y sólo comprueban de forma ocasional que esa dirección no haya cambiado. Sin embargo, los servidores raíz siguen siendo una parte vital para el buen funcionamiento de Internet.

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 ejército de EE.UU. Aproximadamente la mitad depende de organizaciones públicas estadounidenses.



Para saber más

En el siguiente enlace encontrarás la lista actualizada de los servidores raíz DNS.

 [Los servidores raíz DNS.](#)

En el siguiente enlace accederás al contenido de la zona de los servidores raíz DNS. Esta información es publicada por los servidores raíz DNS.

 [Información publicada por los servidores raíz DNS.](#)

2.7.- Tipos de registros 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, según se describe en la tabla siguiente:

Formato de los registros de recursos DNS

Campo	Descripción
Propietario	Indica el nombre de dominio <u>DNS</u> que posee un registro de recursos. Este nombre es el mismo que el del nodo del árbol de la consola donde se encuentra un registro de recursos.
Tiempo de vida (TTL)	Para la mayor parte de los registros de recursos, este campo es opcional. Indica el espacio de tiempo utilizado por otros servidores <u>DNS</u> para determinar cuánto tarda la información en caché en caducar un registro y descartarlo. Por ejemplo, la mayor parte de los registros de recursos que crea el servicio del servidor <u>DNS</u> heredan el <u>TTL</u> mínimo (predeterminado) de 1 hora desde el registro de recurso de inicio de autoridad (<u>SOA</u>) que evita que otros servidores <u>DNS</u> almacenen en caché durante demasiado tiempo. En un registro de recursos individual, puede especificar un <u>TTL</u> específico para el registro que suplante el <u>TTL</u> mínimo (predeterminado) heredado del registro de recursos de inicio de autoridad. También se puede utilizar el valor cero (0) para el <u>TTL</u> en los registros de recursos que contengan datos volátiles que no estén en la memoria caché para su uso posterior una vez se complete la consulta <u>DNS</u> en curso.
Clase	Contiene texto nemotécnico estándar que indica la clase del registro de recursos. Por ejemplo, el valor " IN " indica que el registro de recursos pertenece a la clase Internet. Este campo es <i>obligatorio</i> .
Tipo	Contiene texto nemotécnico estándar que indica el tipo de registro de recursos. Por ejemplo, el texto nemotécnico " A " indica que el registro de recursos almacena información de direcciones de host. Este campo es <i>obligatorio</i> .
Datos específicos del registro	Un campo de longitud variable y <i>obligatorio</i> con información que describe el recurso. El formato de esta información varía según el tipo y clase del registro de recursos.

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

Nota: en los siguientes ejemplos de registros de recurso, el campo TTL se omite en caso de ser opcional. El campo TTL se ha incluido en la sintaxis de cada registro para indicar dónde puede agregarse.

Tipos de registros DNS

Registro	Descripción, sintaxis y ejemplo
A	<p>Descripción: Address (Dirección). Este registro se usa para traducir nombres de host a dirección IP versión 4.</p> <p>Sintaxis: propietario clase ttl A IP_version4</p> <p>Ejemplo: host1.ejemplo.com IN A 127.0.0.1.</p>
AAAA	<p>Descripción: Address (Dirección). Este registro se usa para traducir nombres de host a dirección IP versión 6.</p> <p>Sintaxis: propietario clase ttl AAAA IP_version6</p> <p>Ejemplo: host1.ipv6.ejemplo.com. IN AAAA 1234:0:1:2:3:4:567:89ab.</p>
CNAME	<p>Descripción: Canonical Name (Nombre Canónico). Se usa para crear nombres de host alias. Hay que tener en cuenta que el nombre de host al que el alias referencia debe haber sido registrado previamente como registro tipo "A". Comúnmente usado cuando un servidor con una sola dirección IP ejecuta varios servicios, como: <u>ftp</u>, <u>web</u>... y cada servicio tiene su propia entrada <u>DNS</u>. Un ejemplo es cuando un solo servidor web aloja distintos dominios en una misma IP ( virtualhost).</p> <p>Sintaxis: propietario ttl clase CNAME nombreC</p> <p>Ejemplo: nombrealias.ejemplo.com CNAME nombreverdadero.ejemplo.com. Como se ha comentado anteriormente nombrealias.ejemplo.com debe estar definido como registro tipo A.</p>
NS	<p>Descripción: Name Server (Servidor de Nombres). Indica qué servidores de nombre tienen autoridad sobre un dominio concreto. Cada dominio se puede asociar a una cantidad de servidores de nombres.</p> <p>Sintaxis: propietario ttl IN NS nombreServidorNombreDominio.</p> <p>Ejemplo: ejemplo.com. IN NS nombreservidor1.ejemplo.com.</p>
MX	<p>Descripción: Mail eXchange (Registro de Intercambio de Correo). Asocia un nombre de dominio con una lista de servidores de intercambio de correo para ese dominio.</p> <p>Sintaxis: propietario ttl clase MX preferencia hostIntercambiadorDeCorreo.</p> <p>Ejemplo: ejemplo.com. MX 10 servidorcorreo1.ejemplo.com. El número, en este caso 10, indica la preferencia, y tiene sentido en caso de existir varios servidores de correo. A menor número mayor preferencia.</p>
PTR	<p>Descripción: Pointer (Indicador). Traduce direcciones IP en nombres de dominio. También conocido como 'registro inverso', ya que funciona a la inversa del registro "A".</p> <p>Sintaxis: propietario ttl clase PTR nombreDominio</p> <p>Ejemplo: 1.0.0.10.in-addr.arpa. PTR host.ejemplo.com.</p>

	<p>Descripción: Start Of Authority (Autoridad de la zona). Proporciona información sobre el servidor principal de la zona.</p> <p>Sintaxis: propietario clase SOA servidorNombre personaResponsable numeroSerie intervaloActualización intervaloReintentos caducidadTiempoDeVidaMínimo .</p> <p>Ejemplo:</p>
SOA	<p>El propietario (servidor <u>DNS</u> principal) se especifica como "@" porque el nombre de dominio es el origen de todos los datos de la zona (ejemplo.com). Se trata de una convención estándar para registros de recursos y se utiliza más a menudo en los registros <u>SOA</u>. El número de versión de esta base de datos. Debes incrementar este número cada vez que actualizas la base de datos.</p>
TXT	<p>Descripción: TeXT (Información textual). Permite a los dominios identificarse de modo que no se envíen correos no deseados.</p> <p>Sintaxis: propietario ttl clase TXT cadenaDeTexto.</p> <p>Ejemplo: ejemplo.com. TXT "Ejemplo de información de nombre de dominio"</p>
SPF	<p>Descripción: Sender Policy Framework. Es un registro de tipo <u>TXT</u> que va creado en el <u>DNS</u>, en la cual se ponen las informaciones del propio servidor de correo con la sintaxis para evitar el envío de correos suplantando identidades. Por lo tanto, ayuda a combinar en este registro se especifica qué hosts están autorizados a enviar correo desde el dominio. El servidor que recibe, consulta el <u>DNS</u> para comparar la <u>IP</u> desde la cual le llega, con lo que el registro.</p> <p>Sintaxis: propietario ttl clase IN SPF cadenaDeTexto.</p> <p>Ejemplo: ejemplo.com IN SPF "v=spf1 a:mail.ejemplo.com -all".</p>

Para saber más

En el siguiente enlace encontrarás más información sobre el registro SPF.

[Más información sobre el registro SPF.](#)

The SPF Project. [SPF \(CC BY-NC-SA\)](#)

2.8.- Funcionamiento del cliente DNS



Citas para pensar

“

Los sabios son los que buscan la sabiduría; los necios piensan ya haberla encontrado.

Napoleón Bonaparte

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.

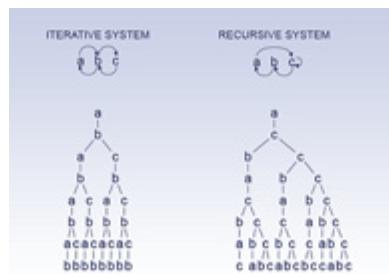
Por ejemplo, el nombre especificado puede ser el nombre completo de un equipo, como rrhh.departamento.empres.org., y el tipo de consulta especificado para buscar un registro de recursos de dirección (**A**) por ese nombre. Considera una consulta DNS como una pregunta de un cliente a un servidor en dos partes, como: "¿Tiene algún registro de recursos de dirección (A) de un equipo llamado 'rrhh.departamento.empres.org.'?". Cuando el cliente recibe una respuesta del servidor, lee e interpreta el registro de recursos "A" respondido, y aprende la dirección IP del equipo al que preguntó por el nombre.

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.



2.8.1.- Consultas recursivas



Citas para pensar

“

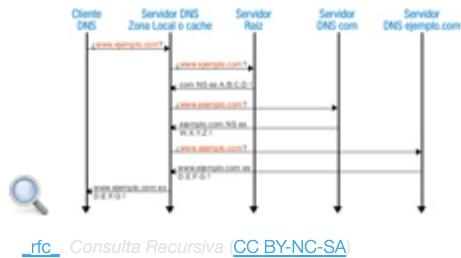
Daría todo lo que sé por la mitad de lo que ignoro.

René Descartes

Tu ordenador (cliente DNS) formula una **consulta** a tu servidor DNS preferido (el que tienes configurado como primero en tu configuración de red, generalmente el proveedor de Internet). 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. En la mayor parte de los casos, el servidor DNS se configura, de forma predeterminada, para admitir el proceso de recursividad como se muestra en el gráfico siguiente.



[Resumen textual alternativo](#)

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.

Por ejemplo, piense en la posibilidad de usar el proceso de recursividad para localizar el nombre "www.ejemplo.com" cuando el cliente consulte un único servidor DNS. El proceso ocurre cuando un servidor y un cliente DNS se inician y no tienen información almacenada en la caché local disponible para ayudar a resolver la consulta de un nombre. El servidor supone que el nombre consultado por el cliente es para un nombre de dominio del que el servidor no tiene conocimiento local, según sus zonas configuradas.

Primero, el servidor preferido analiza el nombre completo y determina que necesita la ubicación del servidor con autoridad para el dominio de nivel superior "com". A continuación, utiliza una consulta iterativa al servidor DNS "com" para obtener una referencia al servidor "ejemplo.com". Finalmente, se entra en contacto con el servidor "ejemplo.com". Ya que este servidor contiene el nombre consultado como parte de sus zonas configuradas, responde con autoridad al servidor original que inició la recursividad. Cuando el servidor original recibe la respuesta que indica que se obtuvo una respuesta con autoridad a la consulta solicitada, reenvía esta respuesta al cliente solicitante y se completa el proceso de consulta recursiva.

2.8.2.- Consultas iterativas



Citas para pensar

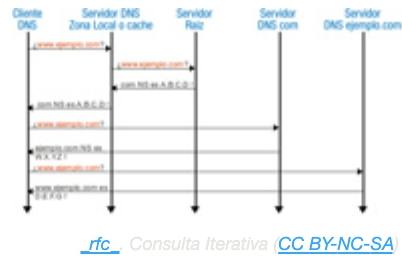
“

Yo no procuro conocer las preguntas; procuro conocer las respuestas.

Confucio.

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



[rfc](#) Consulta Iterativa [CC BY-NC-SA](#)

[Resumen textual alternativo](#)

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. Por ejemplo, si un servidor DNS de una intranet recibe una consulta de un cliente local para "www.ejemplo.com", es posible que devuelva una respuesta de su caché de nombres. Si el nombre consultado no está almacenado actualmente en la caché de nombres del servidor, puede que, para responder, el servidor proporcione una referencia, es decir, una lista de registros de recursos de dirección (A) y de servidor de nombres (NS) para otros servidores DNS que estén más cerca del nombre consultado por el cliente.

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. Por ejemplo, en el caso más complicado, el cliente DNS puede expandir su búsqueda a los servidores de dominio raíz en Internet en un esfuerzo por localizar los servidores DNS que tienen autoridad para el dominio "com". Una vez en contacto con los servidores raíz de Internet, puede recibir más respuestas iterativas de estos servidores DNS que señalan a los servidores DNS de Internet reales para el dominio "ejemplo.com". Cuando se proporcionan registros de estos servidores DNS al cliente, éste puede enviar otra consulta iterativa a los servidores DNS externos del dominio ejemplo en Internet, que pueden responder con una respuesta definitiva y con autoridad.

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.

2.8.3.- Consultas inversas



Citas para pensar

“

*El modo de dar una vez en el clavo es dar cien veces
en la herradura.*

Miguel de Unamuno.

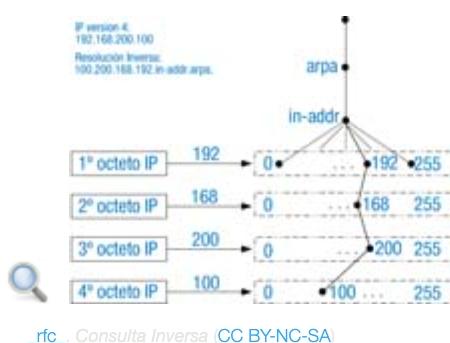
En la mayoría de las consultas DNS los clientes normalmente realizan una búsqueda directa. Este tipo de consulta espera recibir una dirección IP como respuesta a la consulta. Pero, DNS también proporciona un proceso de búsqueda inversa, es decir, buscar un nombre de host a través de una dirección IP. 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. Por ejemplo, para la IP 192.168.200.100 su resolución inversa sería:



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.



[Resumen textual alternativo](#)

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.



Para saber más

En el siguiente enlace puedes encontrar más información disponible acerca de IPv6 y DNS, con ejemplos acerca de cómo crear y usar nombres de dominio ip6.arpa, en el documento RFC 3596 Extensiones DNS compatibles con IP versión 6.

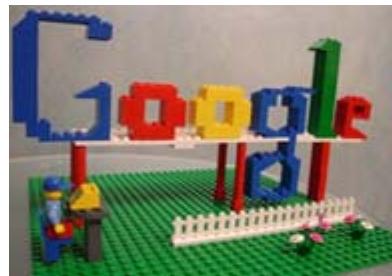
 [Documento RFC 3596 Extensiones DNS compatibles con IP versión 6 existentes.](#)

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.

2.9.- Cómo funcionan los DNS preferidos y alternativos

El servidor DNS preferido es aquel con el que el cliente prueba en primer lugar. También es el servidor en el que el cliente DNS actualiza sus registros de recursos. 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 servidores DNS preferidos  alternativos especificados se consultan en el orden que aparezcan en la lista.



[keso](#) [Google](#) ([CC BY-NC-ND](#))

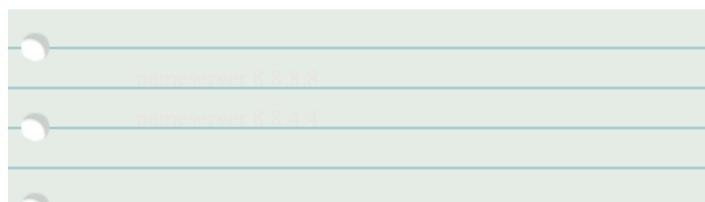
Sin un servidor DNS preferido, el cliente DNS no puede consultar un servidor DNS. Sin un DNS alternativo, las consultas no se resolverán si el servidor DNS preferido falla.

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

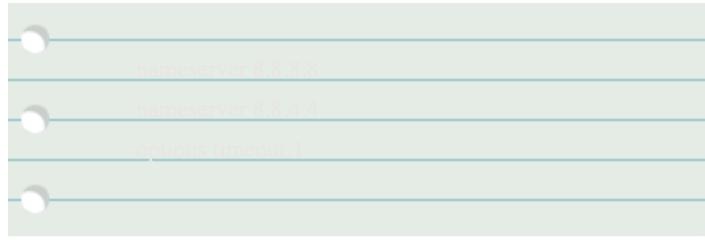
1. El servidor DNS preferido responde primero a una consulta DNS o a una actualización DNS.
2. Si el servidor DNS preferido no responde a una consulta DNS o a una actualización DNS, la consulta o actualización se redirige al servidor DNS alternativo.
3. Si el servidor DNS alternativo no responde y el cliente DNS está configurado con las direcciones IP adicionales de servidores DNS, el cliente DNS envía la consulta o actualización al siguiente servidor DNS de la lista.
4. Si alguno de los servidores DNS (un servidor preferido, un servidor alternativo o cualquier otro de la lista) no responde, dicho servidor se quita temporalmente de la lista.
5. Si ninguno de los servidores DNS responden, la consulta o actualización del cliente DNS no se realiza.

En los equipos tipo GNU/Linux puedes configurar estos servidores en el archivo /etc/resolv.conf e incluso puedes realizar balanceo de carga entre ellos, así como la modificación del tiempo de espera efectuado desde que un servidor falla hasta que se prueba con otro.

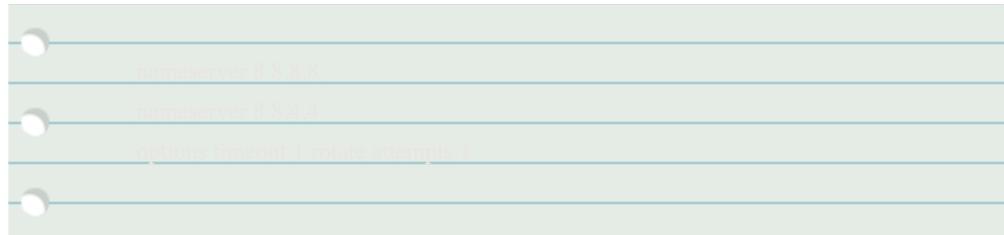
La configuración sería algo así:



En este caso si 8.8.8.8 falla la resolución se realizará a través de 8.8.4.4. El problema es que por defecto el valor de tiempo de espera (timeout) asignado es 5 segundos, por lo que tardará un tiempo en detectar que tiene que utilizar el segundo DNS y todo irá muy lento. Para solucionarlo, tienes que usar la directiva "options" y modificar el timeout. Así, puedes poner 1 segundo como se demuestra en el siguiente ejemplo:



Otra opción también interesante es "rotate", que permite distribuir la carga entre todos los servidores listados y evitar que todas las peticiones vayan siempre al primero:



Configurando estas opciones aseguramos que en caso del fallo del servidor DNS preferido el rendimiento de la máquina no se degrade.

Ten en cuenta que es posible y más que probable que el fichero `resolv.conf` sea modificado cuando configuras la red mediante un gestor de conexión de redes, como: NetworkManager, wicd ... Por lo tanto, revisa este fichero.

Es conveniente que le des una visita al manual de `resolv.conf`: `man resolv.conf`.



Para saber más

Es recomendable que visites el siguiente enlace sobre los servidores DNS públicos de Google: 8.8.8.8 y **8.8.4.4**.

 [Servidores DNS públicos de Google: 8.8.8.8 y 8.8.4.4.](#)

 [OpenDNS](#)

2.10.- Comandos (I)

A la hora de saber si tienes conectividad con alguna máquina en Internet, o en red local, se suele utilizar el **comando ping**, el cual indica, según su respuesta, si posees conectividad con la máquina en cuestión. El comando ping lo puedes utilizar para consultar direcciones IP o nombres de dominios.

Por lo tanto, el comando **ping** debe ser capaz de consultar información sobre el sistema de nombres de dominio; es un resolutor, un programa cliente capaz de consultar información sobre el sistema de nombres de dominio.

Normalmente, un resolutor trabaja discretamente en segundo plano y los usuarios no conocen su presencia, es decir, que toda consulta de un cliente DNS a su servidor suele realizarla el programa que invocamos (ping, ftp, telnet, mail, navegador web, etc.). Por ejemplo, si solicitas una conexión ftp a **ftp.rediris.es**, la aplicación ftp que emplees llama a un programa resolutor local que busca la dirección IP de ese ordenador **130.206.1.5** sin que tengas conciencia de ello, esto es, para ti el proceso es transparente. Además de este trabajo en segundo plano, el usuario puede conectarse directamente al programa resolutor enviando consultas y resolviendo respuestas. Comandos resolutores típicos en sistemas operativos GNU/Linux son: nslookup, host y dig.

El comando **nslookup**, en algunas distribuciones GNU/Linux ya no está soportado pues está obsoleto (deprecated). Por lo tanto, hoy en día, se suelen utilizar el comando host para consulta de direcciones IP y el comando **dig** para consulta de servidores DNS activos.

¿Cómo funcionan todos estos comandos? Veamos:

Ejemplos de resolución directa: Resolución de nombre a IP.

```
alumno@servidor:~$ nslookup ftp.rediris.es
Server: 192.168.0.5#53
Address: 192.168.0.5#53

Non-authoritative answer:
Name: ftp.rediris.es
Aliases: www.rediris.es
Address: 130.206.1.5

alumno@servidor:~$ host ftp.rediris.es
ftp.rediris.es has address 130.206.1.5
alumno@servidor:~$ dig ftp.rediris.es

;; qtype: A, id: 4096, flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;QUESTION SECTION:
ftp.rediris.es.          IN      A

ANSWER SECTION:
ftp.rediris.es.          5364   IN      A        130.206.1.5
;; Query time: 82 msec
;; SERVER: 192.168.0.5#53(192.168.0.5)
;; WHEN: Fri Mar 3 09:51:39 2001
;; MSG SIZE rcvd: 68

alumno@servidor:~$
```

[rfc](#) Resolución Directa (CC BY-NC-SA)

2.10.1.- Comandos (II)

Ejemplos de resolución inversa: Resolución de IP a nombre.

1. Comando nslookup:

Para consultar el nombre de la IP 130.206.1.5, basta con ejecutar:

```
nslookup 130.206.1.5
Server: 192.168.1.1
Address: 192.168.1.100

Non-authoritative answer:
130.206.1.5 in-addr.arpa domain name pointer zeppo.rediris.es

Authoritative answers can be found from:
```

Donde puedes ver que la IP 130.206.1.5 corresponde con el nombre de dominio zeppo.rediris.es

2. Comando host:

Para consultar el nombre de la IP 130.206.1.5, basta con ejecutar:

```
host 130.206.1.5
130.206.1.5 domain name pointer zeppo.rediris.es
```

Donde puedes ver que la IP 130.206.1.5 corresponde con el nombre de dominio zeppo.rediris.es

3. Comando dig:

Para consultar el nombre de la IP 130.206.1.5, basta con ejecutar:

```
dig > 130.206.1.5  
admino@servidor:~$ dig > 130.206.1.5  
  
; <--> 130.206.1.5  
;; global options: +cmd  
;; Got answer:  
;; ->HEADER: opcode: QUERY, status: NOERROR, id: 33994  
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0  
  
;; ANSWER SECTION:  
130.206.1.5. IN PTR zeppo.rediris.es.  
  
;; Query time: 73 msec  
;; SERVER: 127.0.0.1#53  
;; WHEN: Sun Jul 10 11:20:20 2016  
;; MSG SIZE rcvd: 35
```

Donde puedes ver que la IP 130.206.1.5 corresponde con el nombre de dominio zeppo.rediris.es, e incluso el registro de recursos empleado: PTR.



Reflexiona

Es conveniente que le hagas una visita al manual de los comandos nslookup, host y dig.



Debes conocer

En la siguiente página web se ofrecen herramientas DNS, herramientas de red, herramientas de correo electrónico, información de DNS y recopilación de información IP que te pueden ser muy útiles para gestionar, monitorizar y analizar tu servidor DNS.

 [DNSstuff](#)

2.11.- Instalación del servidor DNS BIND

Para una instalación del servidor DNS BIND en Debian 6 (Squeeze) realiza el siguiente procedimiento como usuario root, teniendo en cuenta que el servidor está identificado como sigue:

- ✓ Hostname: debian-servidor-fp.
- ✓ IP: 192.168.200.250.



Xesc DNS BIND [CC BY-NC-SA]

• 2011 • 第五輯 • 10月號 • 10

udp 0.0.0.0:16389 250.53.0.0.0.71442 named

edu 0.127 0.0153 0.000 *14



Para saber más

Para mantenerte informado y actualizado es recomendable que visites la página oficial del servidor [BIND](#).

 [Página oficial del servidor BIND.](#)

2.11.1.- Archivos de configuración del servidor DNS

Tras la instalación del servidor **DNS BIND** (bind9) existe la ruta **/etc/bind**, la cual contiene sus ficheros de configuración. Una estructura tipo de /etc/bind que puedes encontrar al instalar **bind** sería similar a la que se muestra en la siguiente imagen:

```
root@debian-servidor-fp:~# tree /etc/bind
/etc/bind
├── bind.keys
├── db.0
├── db.127
├── db.255
├── db.empty
├── db.local
├── db.root
├── named.conf
├── named.conf.default-zones
├── named.conf.local
└── named.conf.options
└── rndc.key
└── zones.rfc1918

0 directories, 13 files
root@debian-servidor-fp:~#
```

[rfc](#) Directorio etc/bin. (CC BY-NC-SA)

El servidor **DNS BIND (bind9)** posee por defecto en su instalación el fichero  [etc/bind/named.conf](#) (conf - 463 B)



```
options {
    directory "/var/named";
};

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

Donde,

`/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 y por lo tanto no se ejecuta.

Puedes modificar los ficheros de configuración a tu antojo. Así, puedes crear incluso nuevos ficheros de configuración que sean llamados desde otros mediante la directiva `include`.



Para saber más

Enlace a los elementos que puedes emplear en los ficheros de configuración, de la versión BIND 9.11 empleada en esta unidad

 [Documentación oficial en formato HTML sobre BIND.](#)

Puedes realizar una verificación de los ficheros de configuración y de zona por posibles fallos mediante los comandos "named-checkconf" y "named-checkzone" respectivamente. Estos comandos suelen ejecutarse con la siguiente sintaxis:

named-checkconf [-p] {filename}

donde:

named-checkconf → comprueba la sintaxis pero no la semántica de un fichero de configuración named. El fichero se analiza y comprueba por errores de sintaxis, junto con todos los archivos incluidos en él. Si no se especifica ningún fichero, por defecto se comprueba /etc/named.conf. -p → imprime la salida de named.conf y los ficheros incluidos en forma canónica si no fueron detectados errores.

**filename → El nombre del archivo de configuración que desea comprobar.
Si no se especifica, por defecto es /etc/named.conf.**

named-checkzone {zonename} {filename}

donde:

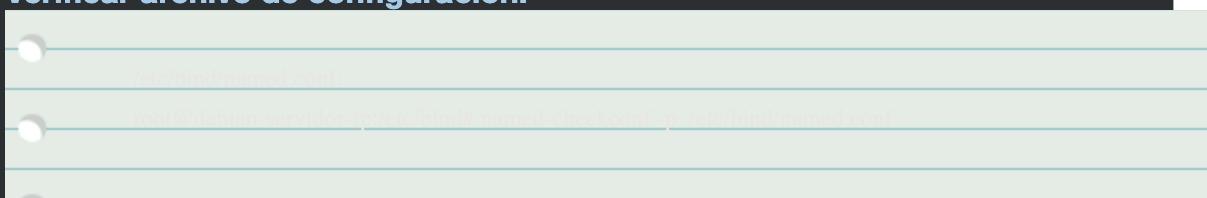
named-checkzone → comprueba la sintaxis y la integridad de un archivo de zona. Realiza las mismas comprobaciones que named hace al cargar una zona. Esto hace que sea útil para comprobar los archivos de zona antes de configurarlos en un servidor de nombres.

zonename → El nombre de dominio de la zona que se comprueba.

filename → **El nombre del archivo de zona.**

Ejemplos de ejecución:

1. Verificar archivo de configuración.



2. Verificar el dominio de zona ejemplo.com en el archivo de zona.



2.11.2.- Arranque y parada del servidor DNS

En un sistema operativo Debian 6.0 (Squeeze) puedes comprobar el estado del servicio bind mediante el comando `service` o mediante el comando `/etc/init.d/bind`:

✓ Comando `service`:

1.- Comprobar las opciones del comando:

```
root@laptop:~# service --help
  Usage: service {start|stop|status|restart|force-reload} [service ...]
```

Donde:

`start` → opción que permite arrancar el servicio.

`stop` → opción que permite apagar el servicio.

`reload` → opción que permite recargar la configuración del servicio sin tener que reiniciarlo.

`restart` → opción que permite reiniciar el servicio.

`force-reload` → opción que permite forzar la recarga de configuración del servicio.

`status` → opción que permite comprobar si el servicio está activo o inactivo.

2.- Arrancar el servidor DNS:

```
root@laptop:~# service bind start
Starting domain name server: bind[1] ...done.
```

3.- Parar el servidor DNS:

```
root@laptop:~# service bind stop
Stopping domain name server: bind[1] ...done.
```


✓ Comando /etc/init.d/bind9:

1.- Comprobar las opciones del comando:

```
root@rhel7: ~]# man bind9
[...]
  -c          Configuration file for the daemon
  -t          Test mode (don't accept connections from clients)
[...]
```

2.- Arrancar el servidor DNS:

```
root@rhel7: ~]# ./etc/init.d/bind9 start
Starting domain name service: bind9
[...]
```

3.- Parar el servidor DNS:

```
root@rhel7: ~]# ./etc/init.d/bind9 stop
Stopping domain name service: bind9
[...]
```

4.- Comprobar el estado activo/inactivo del servicio:

```
root@rhel7: ~]# ./etc/init.d/bind9 status
bind9 not running
```

Como puedes comprobar la salida del comando determina que el servicio está inactivo, entonces lo arrancamos:

```
root@rhel7: ~]# ./etc/init.d/bind9 start
Starting domain name service: bind9
[...]
```

Se vuelve a lanzar el comando para comprobar de nuevo el estado:

```
root@rhel7: ~]# ./etc/init.d/bind9 status
bind9 is running
```

Ahora puedes comprobar que la salida del comando determina que el servicio está activo.

2.11.3.- Configuración como caché DNS



Citas para pensar

“

Buena memoria es la escritura, pues para siempre dura.

Refranero español

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 GNU/Linux Debian 6.0 (Squeeze) la configuración de un servidor DNS BIND (bind9) como caché viene establecida 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:

1. Verifica que el contenido del fichero /etc/bind/named.conf.options. Tras la instalación, es el siguiente:

```
options {
    directory "/var/cache/bind";
    // If your BIND providers one or more IP addresses for static
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses for
    // the public name resolution service.
    //forwarders {
    //    1.1.1.1;
    //};

    // You can define multiple sections in this file.
    // For example, you can have multiple sections for
    // different domains such as
    //   options {
    //       ...
    //   };
    //   options {
    //       ...
    //   };
    //   options {
    //       ...
    //   };
}
```

2. Modificas el fichero /etc/resolv.conf para que solamente tenga activa la siguiente línea:

```
nameserver 127.0.0.1
```

De tal forma que ahora el servidor DNS activo solamente es el local, que tienes configurado como caché.

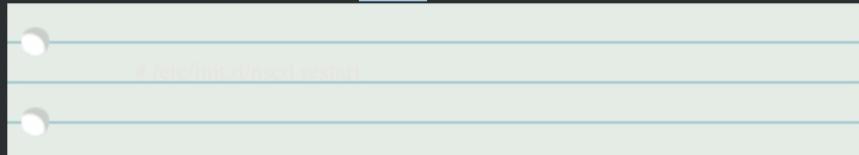
3. Una vez efectuados los cambios recargas el servidor con el comando: service bind9 reload o también con /etc/init.d/bind9 reload.

Puedes realizar una comprobación del funcionamiento de la caché, si una vez realizado lo expuesto, sigues el procedimiento que encontrarás en el fichero siguiente:

 [Comprobar caché](#) (pdf - 57.53 KB)

El borrado de la caché DNS lo puedes realizar en el cliente DNS y en el propio servidor DNS. Así, para un sistema operativo GNU/Linux podrías realizar los siguientes comandos según el caso:

1. Borrado de la caché del cliente DNS:



2. Borrado de la caché del servidor DNS BIND:



2.11.4.- Configuración como DNS maestro



Citas para pensar

“ *El maestro sabe lo que hace.*

Proverbio español.

En GNU/Linux Debian 6.0 (Squeeze) puedes configurar un servidor DNS BIND como maestro modificando el archivo `/etc/bind/named.conf.local` realizando el siguiente procedimiento:

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. Por ejemplo:



[jasonxcombs.. Master Slave \(CC BY-NC-SA\)](#)

• Zonas creadas tipo master
• tipo "ejemplo.com."
• tipo master
• /var/lib/bind/master/db.ejemplo.com.hosts

En este ejemplo, el servidor sirve el dominio "ejemplo.com" como master, y la zona se guarda en el fichero `/var/lib/bind/master/db.ejemplo.com.hosts`.

Habrá una entrada de este tipo por cada zona servida.

Normalmente los ficheros de zona están situados en la ruta /var/lib/bind. Para una mejor comprensión, y para facilidad de uso en posteriores momentos, estaría bien que crearas los directorios master y slave dentro de esa ruta. Así, los ficheros con zonas maestras se pueden encontrar en /var/lib/bind/master/db.*.hosts y los ficheros con zonas esclavas se pueden encontrar en /var/lib/bind/slave/db.*.hosts.



Ejercicio resuelto

¿Cuál sería mediante dig el comando que deberías ejecutar para obtener la IP del servidor de correo correspondiente al dominio ejemplo.com, si el servidor de correo posee el nombre mail?

[Mostrar retroalimentación](#)

2.11.5.- Configuración como DNS esclavo



Citas para pensar

“

Hace mal quien lo secundario hace principal.

Proverbio español.

En GNU/Linux Debian 6.0 (Squeeze) puedes configurar un servidor DNS BIND como esclavo modificando el archivo `/etc/bind/named.conf.local` realizando el siguiente procedimiento:

5. 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. Por ejemplo:



```
seismic007.
BY-NC
// Zona recibida (proceso local)
zone "example.com" {
    type slave;
    masters { 192.168.200.250; };
    file "/var/lib/bind/slave/db.example.com.hosts";
};
```

En este ejemplo, el servidor sirve el dominio "example.com" como slave, y la zona se guarda en el fichero `/var/lib/bind/slave/db.example.com.hosts`.

Habrá una entrada de este tipo por cada zona servida.

Normalmente los ficheros de zona están situados en la ruta / var/lib/bind. Entonces, para una mejor comprensión y facilidad de uso en posteriores momentos, estaría bien que crearas los directorios master y slave dentro de esa ruta. Así, los ficheros con zonas maestras se pueden encontrar en /var/lib/bind/master/db.*.hosts y los ficheros con zonas esclavas se pueden encontrar en /var/lib/bind/slave/db.*.hosts.

6. En el servidor maestro configuras la sección correspondiente al servidor master en el [fichero bind/named.conf.local](#):

1. Para indicar qué servidores tienen permitido la transferencia de los ficheros de zona, mediante la directiva allow-transfer. Por ejemplo:

```
allow-transfer {192.168.200.100;192.168.210.100;10.10.42.41;10.10.42.42;};
```

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.

2. 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:

```
also-notify {192.168.200.100;10.10.42.41;};
```

Por ejemplo, una zona tipo master con las directivas anteriores podría ser la siguiente:

```
zone "ejemplo.com" {
    type master;
    file "/var/lib/bind/master/db.ejemplo.com.hosts";
    allow-transfer {192.168.200.100;10.10.42.41;};
    also-notify {192.168.200.100;10.10.42.41;};}
```

Mediante la directiva `also-notify` se mantienen los servidores DNS sincronizados. Así, el servidor DNS esclavo podrá satisfacer las peticiones DNS al igual que lo haría el maestro. Esto implica que se garantiza la disponibilidad del servicio DNS puesto que aunque el servidor maestro deje de funcionar, el servidor esclavo podrá seguir ofreciendo el servicio. Además, en caso de recibir múltiples conexiones concurrentes, siendo por tanto el número de peticiones muy elevado, la carga se distribuye entre los servidores.



Para saber más

Puedes visitar el siguiente enlace donde encontrarás la documentación ofrecida sobre el servidor DNS BIND en su página oficial.



[Documentación ofrecida sobre el servidor DNS BIND en su página oficial.](#)

3.- Servicio de directorio



Caso práctico

A BK Programación, una empresa, con la que ya han trabajado anteriormente en proyectos asignados a **Juan**, le han encargado un proyecto con las siguientes especificaciones para el departamento de atención al cliente:

1. Controlar el acceso de usuarios a los equipos de la empresa, de tal forma que, independientemente del ordenador con el que trabajen en la empresa, mediante autenticación de usuario y contraseña, puedan tener acceso al mismo.
 2. Controlar el acceso de usuarios a la herramienta de gestión de incidencias y proyectos.



Stockbyte. Juan (CC BY-NC)

Para ello BK Programación ha determinado realizar una autenticación por LDAP mediante OpenLDAP, puesto que aunque la configuración y el tiempo empleado va a ser más costoso que empleando otras alternativas, determina que la empresa necesita una centralización de esa base de datos de usuarios para que la aplicación de gestión de incidencias y proyectos, y los equipos ofrecidos por la empresa a sus trabajadores, puedan beber de la misma fuente: la base de datos de OpenLDAP.



[herzogbr](#). Guía Telefónica (CC BY-NC)

3.1.- ¿Para qué usar un servicio de directorio?



Citas para pensar

“

Siempre deseé que mi computadora fuera tan fácil de usar como mi teléfono. Mi deseo se ha hecho realidad: ya no sé usar mi teléfono.

Bjarne Stroustrup

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. Así, comparando con la guía telefónica tradicional, un directorio electrónico permite realizar búsquedas, no solamente por orden alfabético, sino también por apellido, dirección, teléfono... ¿Cómo realizarías una búsqueda por teléfono en una guía telefónica tradicional?



Lori Greig Dirección CC BY-ND

Es más, podrías sumar campos de búsqueda, como por ejemplo: dirección y apellido.

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. ¿Qué pasaría si la sincronización no tuvo lugar y una aplicación accede a los datos? Pues sí, obtendría los datos no actualizados, o error en los mismos.

Un caso muy común es el de los servidores Web con autenticación: si solamente dispones de un servidor web la solución es sencilla, puesto que solamente se necesitaría actualizar una base de datos de usuarios, pero ¿y si dispones de más de un servidor web que debe acceder a la misma base de datos? Entonces, la cosa se complica, puesto que debes sincronizar a los distintos servidores. Es más, y si esa base de datos la quisiéramos aprovechar para ofrecer otro servicio distinto del de los servidores web? Pues, todo el trabajo no sería aprovechable, y por lo tanto sería mejor desde un principio adaptar este sistema a los servicios de directorios.

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:

1. **Su creación:** incorporar a los certificados los datos contenidos en el directorio.
2. **Su distribución:** tener accesibles mediante un protocolo estándar los certificados.
3. **Su destrucción:** revocar los certificados de forma sencilla simplemente borrando el certificado del directorio.
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.

Por todo ello las aplicaciones prácticas que poseen los servicios de directorio son muy diversas y ventajosas, como por ejemplo:

- ✓ Autenticación de usuarios: en aplicaciones web, correo electrónico,  RADIUS..., sistemas de control de entradas a edificios, bases de datos comunes en organizaciones,
- ✓ En sistemas operativos: gestión de cuentas de acceso, servidores de certificados, libretas de direcciones compartidas...

3.2.- 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, ¿o puedes actualizar a tu antojo los servidores raíz DNS?
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).

Pero, no por ello, poseen el impedimento de trabajar juntos, es más, usualmente 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.



Debes conocer

Antes de intentar configurar una aplicación web con autenticación LDAP deberías probar la instalación, configuración y autenticación por medio de una base de datos SQL.

En el siguiente enlace encontrarás información detallada sobre el procedimiento de instalación, configuración y autenticación por medio de una base de datos SQL de la aplicación web OpenCart.

 [Instalación, configuración y autenticación por medio de una base de datos SQL de la aplicación web OpenCart.](#) (pdf - 0.12 MB)

3.3.- 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:** Varios servidores proporcionan el servicio de directorio. Cuando está distribuido, los datos pueden estar fraccionados y/o replicados:
 - Cuando la información 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.



geralt Licencia Pixabay

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

En 1988, la CCITT (Comité Consultivo Internacional de Telégrafos y Teléfonos), ahora ITU-T (Unión Internacional de la Comunicación) 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 (Interconexión de Sistemas Abierta).

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** (Árbol de Directorios de la Información). 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** (Nombre "distinguible" o Completo) y está compuesto por una secuencia de partes más pequeñas llamadas **RDN** (Nombre Completo Relativo), 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.

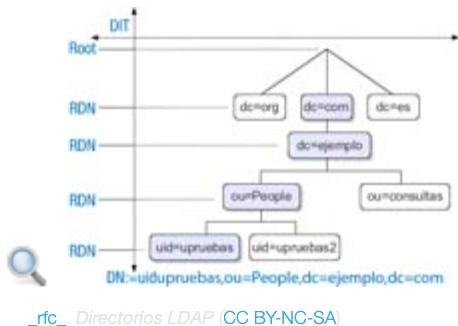
Un **esquema (schema)** define qué clases de objetos se pueden almacenar en el directorio, qué atributos deben contener, qué atributos son opcionales y el formato de los atributos.

Por lo general, existen dos tipos 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) u o (organization).

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



La figura representa un DIT ficticio con entradas en cuatro niveles. Cada entrada se corresponde con una casilla en la figura. En este caso, el nombre válido completo DN del empleado ficticio upruebas es:



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 2252** y **RFC 2256**). 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.

3.4.- 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 la base 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, por ejemplo: la configuración, información sobre el control de acceso, etc.

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

Otra utilidad que puede resultar interesante es la de 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. Aquí es donde el servicio de directorio puede jugar un papel importante, ya que proporciona un **acceso uniforme a la información** contenida en él.

Esta última puede ser una de las mayores utilidades de los directorios, ya que permiten separar la operación de localización de la información del servidor que la contiene.



 Servicios Web (CC BY)

3.5.- El formato de intercambio de datos LDIF

El formato LDIF (Formato de Intercambio de Datos), es el estándar para representar entradas del directorio en formato texto ASCII que posee la siguiente sintaxis:



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 más 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.

A continuación puedes observar un ejemplo de una entrada para describir una cuenta de usuario en un servidor:

3.6.- Instalación de OpenLDAP

El proceso de instalación de OpenLDAP en un sistema basado en Debian es sencillo, aunque la configuración, no tanto.

Para una instalación de OpenLDAP en Debian 6 (Squeeze) realiza el siguiente procedimiento como usuario root, teniendo en cuenta que el servidor está identificado como sigue:

- ✓ Hostname: debian-servidor-fp
- ✓ IP: 192.168.200.250

1. Actualiza los repositorios del sistema operativo.



[openldap](#) OpenLDAP (Todos los derechos reservados)

	Actualización completa de los repositorios.

NOTA: es necesario para el buen funcionamiento del comando que tengas configurada correctamente la conexión a Internet.

2. Actualiza el sistema operativo.

	Actualización completa de los repositorios.

3. Instala los paquetes necesarios para el funcionamiento de OpenLDAP. La instalación te pedirá una contraseña, como puedes ver a continuación.

	Introduzca la contraseña de su usuario root: [REDACTED]
	Introduzca la contraseña de su usuario root: [REDACTED]
	Introduzca la contraseña de su usuario root: [REDACTED]

4. Verifica que el servidor OpenLDAP está activo, por defecto, en el puerto TCP 389.

	Introduzca la contraseña de su usuario root: [REDACTED]
	Introduzca la contraseña de su usuario root: [REDACTED]
	Introduzca la contraseña de su usuario root: [REDACTED]



Para saber más

Es recomendable que visites el siguiente enlace a una URL que contiene cómo instalar y configurar un servidor OpenLDAP en un GNU/Linux basado en Debian.

 [Instalación y configuración del servidor OpenLDAP](#)

3.6.1.- Configuración de OpenLDAP

Una de las principales novedades de la versión 2.4 de OpenLDAP es que se incluye toda la configuración del servidor slapd en un directorio de base cn=config, (`/etc/ldap/slapd.d/cn=config`), en lugar del habitual fichero `/etc/ldap/slapd.conf`. 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`.



[openldap](#) OpenLDAP (Todos los derechos reservados)



Para saber más

En el siguiente enlace puedes encontrar información sobre las especificaciones del esquema (schema) en OpenLDAP.



[Especificaciones del esquema \(schema\) en OpenLDAP.](#)

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`.

La tabla siguiente ofrece un resumen de las clases de objetos utilizadas en el ejemplo de `core.schema` e `inetorgperson.schema` junto con los atributos obligatorios y los valores adecuados de atributo.

Clases de objetos y atributos de uso extendido

Clase de objeto	Significado	Entrada de ejemplo	Atributo obligatorio
dcObject	domainComponent (partes del nombre del dominio).	ejemplo.	dc
organizationalUnit	organizationalUnit (unidad organizativa).	People.	ou
inetOrgPerson	inetOrgPerson (datos sobre personal para Internet/intranet).	Usuario Pruebas Pruebas.	cn ; sn

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.

La configuración de OpenLDAP puede resultar ardua, así que ármate de paciencia y procede como se te indica a continuación.

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

root@luis-OptiPlex-5070:~# dpkg-reconfigure slapd
• Desea editar la configuración del servicio OpenLDAP [N]:
• Introduzca su nombre de dominio DNS, por ejemplo empresas.lnog:
• Nombre de la base de datos de directorios:
• Clave de cifrado para la base de datos:
• Nombre de la base de datos de usuarios:
• Clave de cifrado para la base de datos de usuarios:
• Desea que se haga la base de datos cuando se inicie el servicio slapd [S]:
• Desea iniciar la base de datos en el puerto 389 [S]:
• Desea permitir el protocolo LDAPv2 [S]:

Resumen textual alternativo

Puedes revisar en contenido tipo del fichero /etc/ldap/slapd.d/cn=config/olcDatabase={1}hdb.ldif el siguiente enlace: [!\[\]\(41df40dc33fcd308308830cf1c068f50_img.jpg\) hdb.ldif](#) (ldif - 1.03 KB)

3.6.2.- Arranque y parada del servidor LDAP

En un sistema operativo Debian 6.0 (Squeeze) puedes comprobar el estado del servicio OpenLDAP mediante el comando service o mediante el comando /etc/init.d/slapd:



[openldap](#) OpenLDAP (Todos los derechos reservados)

✓ Comando service:

1.- Comprobar las opciones del comando:

- `start` arranca el servicio.
- `stop` para el servicio.
- `reload` recarga la configuración sin reiniciar.
- `restart` reinicia el servicio.
- `force-reload` fuerza la recarga de configuración.
- `status` comprueba si el servicio está activo.

Donde:

`start` → opción que permite arrancar el servicio.

`stop` → opción que permite apagar el servicio.

`reload` → opción que permite recargar la configuración del servicio sin tener que reiniciarlo.

`restart` → opción que permite reiniciar el servicio.

`force-reload` → opción que permite forzar la recarga de configuración del servicio.

`status` → opción que permite comprobar si el servicio está activo o inactivo.

2.- Arrancar el servidor OpenLDAP:

- `sudo /etc/init.d/slapd start`
- `Starting OpenLDAP: slapd`

3.- Parar el servidor OpenLDAP:

- `sudo /etc/init.d/slapd stop`
- `Stopping OpenLDAP: slapd`

✓ Comando /etc/init.d/slapd:

1.- Comprobar las opciones del comando:

```
root@rhel8: ~ # man slapd
root@rhel8: ~ # man /etc/init.d/slapd
```

2.- Arrancar el servidor OpenLDAP:

```
root@rhel8: ~ # /etc/init.d/slapd start
root@rhel8: ~ # ps aux | grep slapd
root      111  0.0  0.0  1000  100 R 0.0  0:00.00 slapd[111]
```

3.- Parar el servidor OpenLDAP:

```
root@rhel8: ~ # /etc/init.d/slapd stop
root@rhel8: ~ # ps aux | grep slapd
root      111  0.0  0.0  1000  100 S 0.0  0:00.00 slapd[111]
```

4.- Comprobar el estado activo/inactivo del servicio:

```
root@rhel8: ~ # systemctl status slapd
root@rhel8: ~ # ps aux | grep slapd
root      111  0.0  0.0  1000  100 S 0.0  0:00.00 slapd[111]
```

Como puedes comprobar la salida del comando determina que el servicio está inactivo, entonces lo arrancamos:

```
root@rhel8: ~ # systemctl start slapd
root@rhel8: ~ # ps aux | grep slapd
root      111  0.0  0.0  1000  100 R 0.0  0:00.00 slapd[111]
```

Se vuelve a lanzar el comando para comprobar de nuevo el estado:

```
root@rhel8: ~ # systemctl status slapd
root@rhel8: ~ # ps aux | grep slapd
root      111  0.0  0.0  1000  100 R 0.0  0:00.00 slapd[111]
```

Ahora puedes comprobar que la salida del comando determina que el servicio está activo.

El comando slapttest permite verificar la configuración del servidor OpenLDAP.

3.6.3.- Administrando un servidor LDAP

OpenLDAP ofrece una serie de comandos para la administración de datos en el directorio **LDAP**, contenidos en el paquete `ldap-utils`. Los cuatro comandos más importantes para añadir, modificar, buscar y eliminar son explicados a continuación.



[openldap](#). OpenLDAP (Todos los derechos reservados)

[Resumen textual alternativo](#)

1. Añadir entradas: comando ldapadd.

- a. Crea la estructura básica del dominio LDAP mediante la ejecución de un fichero

◆ [estructura basica.ldif](#) (ldif - 0.39 MB)

root@debian-servidor-ldap:~# ldapadd -x -D cn=admin,dc=proyecto,dc=local -w admin -f estructura basica.ldif
adding new entry "cn=users,dc=proyecto,dc=local"
adding new entry "cn=groups,dc=proyecto,dc=local"

- b. Añade un usuario a LDAP de nombre 'upruebas' y contraseña '123456' mediante el archivo ◆ [usuario.ldif](#) (ldif - 0.31 MB)

root@debian-servidor-ldap:~# ldapadd -x -D cn=admin,dc=proyecto,dc=local -w admin -f usuario.ldif
adding new entry "cn=upruebas,dc=users,dc=proyecto,dc=local"

2. Modificar entradas: comando ldapmodify.

- a. Modificar la contraseña del usuario anterior 'upruebas' mediante la ejecución del archivo ◆ [cambiar usuario.ldif](#) (ldif - 0.14 MB)

root@debian-servidor-ldap:~# ldapmodify -x -D cn=admin,dc=proyecto,dc=local -w admin -f cambiar usuario.ldif
modifying entry "cn=upruebas,dc=users,dc=proyecto,dc=local"

3. Buscar entradas: comando ldapsearch.

- a. Buscar todos los usuarios cuyo nombre contenga los caracteres 'pru':

root@debian-servidor-ldap:~# ldapsearch -x -D cn=admin,dc=proyecto,dc=local -w admin "(cn=pru*)"

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

root@debian-servidor-ldap:~# ldapsearch -x -D cn=admin,dc=proyecto,dc=local -w admin "(&(cn=pru*)(mail=daw05*))"

4. Eliminar entradas: comando ldapdelete.

- a. Eliminar el usuario 'upruebas':

root@debian-servidor-ldap:~# ldapdelete -x -D cn=admin,dc=proyecto,dc=local -w admin "cn=upruebas,dc=users,dc=proyecto,dc=local"

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 192.168.200.250 -x -b dc= proyecto-empresa,dc=local  
"(objectclass=*)"
```

Conectaría con el servidor LDAP en la IP 192.168.200.250 para buscar el DIT del dominio proyecto-empresa.local.

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.



Para saber más

Una forma más sencilla de interactuar con el servidor OpenLDAP sería la posibilidad de gestionar el servidor mediante alguna interface gráfica, éstas existen tanto de pago como libres. A continuación se recogen varios enlaces que ofrecen información sobre estas interfaces gráficas (exploradores de directorios LDAP):

[Exploradores de directorios LDAP.](#)

[Explorador de directorio LDAP.](#)

3.6.4.- Configuración de los clientes. Instalación de librerías de autentificación

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

```
cat /etc/nsswitch.conf
passwd: files ldap
group: files ldap
shadow: files ldap
```

3. Reinicia el servicio nscd para que se activen los cambios efectuados en el paso anterior, esto es, para que el sistema operativo recoja los usuarios en primer lugar de los ficheros locales de usuarios y grupos y a continuación del servidor LDAP.

```
reboot
```

4. 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.

```
pam-auth-update --force --all
PAM service: auth
PAM service: account
PAM service: password
PAM service: session
```

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

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

```
getent passwd
root:x:0:0::/root:/bin/bash
nico:x:10001:10001::/home/nico:/bin/bash
upruebas2:x:10002:10001::/home/upruebas2:/bin/bash
```

2. Iniciar sesión en un consola de texto en el equipo cliente con un usuario del **LDAP**. En esta caso, con el usuario upruebas o el usuario upruebas2.

3.6.5.- Probar la autenticación con pamtest

Ahora que la autenticación de usuarios por LDAP está activada en el sistema operativo, es recomendable que efectúes algunas pruebas con la nueva configuración para comprobar si todo funciona correctamente.

El comando pamtest puede ayudarte a realizar estas pruebas. La instalación del mismo se efectúa realizando el siguiente comando:

```
root@debian-servidor:~# apt-get install libpam-test
```



[openldap](#) OpenLDAP (Todos los derechos reservados)

El comando pamtest 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. Veamos unos ejemplos:

1. Intentar autenticar al usuario upruebas2 en el servicio passwd mediante una clave correcta:

```
root@debian-servidor:~# pamtest passwd upruebas2
Trying to authenticate upruebas2 to service passwd
Password:
Authentication succeeded
```

2. Intentar autenticar al usuario upruebas2 en el servicio passwd mediante una clave incorrecta:

```
root@debian-servidor:~# pamtest passwd upruebas2
Trying to authenticate upruebas2 to service passwd
Password:
Authentication failed
```

3. Intentar autenticar al usuario upruebas2 en el servicio ssh mediante una clave correcta:

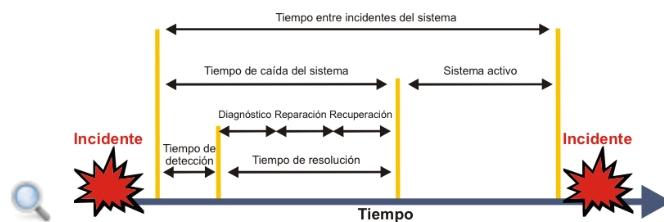
```
root@debian-servidor:~# pamtest ssh upruebas2
Trying to authenticate upruebas2 to service ssh
Password:
Authentication succeeded
```


4.- Servicios de red implicados en la monitorización de sistemas

En la actualidad, es totalmente necesario el uso de herramientas de monitorización ante el gran crecimiento de las tecnologías, para conseguir así un buen funcionamiento de los diferentes servicios y equipos.

Las herramientas de monitorización son las encargadas de realizar continuamente una supervisión de los diferentes recursos y servicios garantizando una disponibilidad y ante errores o fallos del sistema poder tener constancia para que las personas responsables sean capaces de solventarlo.

A continuación, podemos observar la evolución de un incidente del sistema:



Elaboración propia. Evolución de un incidente del sistema [CCO](#)

Las principales ventajas u objetivos de la monitorización son:

- ✓ Minimizar el tiempo del servicio detenido.
- ✓ Mayor disponibilidad del servicio.
- ✓ Supervisión y análisis de las distintas fases en un sistema.



Para saber más

Existen numerosas herramientas de monitorización de sistemas y ranking de ellas. A continuación te dejo dos enlaces dónde puedes ampliar esta información y conocerlas.

[Monitoreo de Redes](#)

[Las 10 mejores herramientas para monitorizar aplicaciones y servidores.](#)

4.1.- Herramientas de monitorización específicas para servicios web

Existen numerosas herramientas gratuitas y de pago para monitorizar nuestros servicios web. Medir el rendimiento de nuestro servidor web de Apache es muy sencillo con herramientas como  [ApacheBench](#) que es gratuita.

ApacheBench (ab) es un programa utilizado para monitorizar el rendimiento de nuestro servidor Web [HTTP](#) o [HTTPS](#) a través de líneas de comandos, mostrándote las peticiones que es capaz de servir, rechazar e incluso no ser capaz de procesar por segundo nuestro servidor Web.

A continuación vamos a:

Instalarlo, lanzar un test de rendimiento y mostrar los resultados, también de una manera gráfica.



[Bioxid](#). Monitorización Servidor Web [CC BY-NC](#)

Todas estas funciones a través de nuestro terminal.



Recomendación

Para ampliar la información respecto a la monitorización de tu servidor Web con Apache Bench puedes ver el siguiente vídeo:

[Guía ApacheBench \(ab\)](#)

[Resumen textual alternativo. \(CC BY-NC\)](#)

4.1.1.- Instalación y utilización de ApacheBench



Citas Para Pensar

“

Solo podrás tomar decisiones acertadas si sabes cómo analizar e interpretar los datos.

Avinash Kaushik

En primer lugar vamos a proceder a la instalación de nuestra herramienta de ApacheBench. Ésta la podemos encontrar dentro del paquete [apache2-utils](#) .



Una vez instalado podemos lanzar ApacheBench y hacer un test de rendimiento a una página elegida a la que se conectará nuestro servidor Web. Para los distintos test vamos a utilizar la página de la junta de Andalucía (<https://educacionadistancia.juntadeandalucia.es/cursos/login/index.php/>).

Por lo tanto, vamos a proceder a lanzar el test, para ello debemos tener en cuenta los distintos parámetros utilizados en la orden:

-n 1000: le estamos indicando que realice un total de 1000 peticiones.

-c 50: le estamos definiendo los bloques.

-f All: se utiliza cuando es una url https.



En la siguiente imagen podemos observar la respuesta una vez lanzamos desde nuestro terminal el test:

```

[elmer@elmer:~/inet]$ ab -n 1000 -c 10 http://educacionadistantia.juntadeandalucia.es/cursos/login/index.php
This is ApacheBench, Version 2.3 <devRevision: 1807734 $sot:>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
<DPL>
Licensed to The Apache Software Foundation, http://www.apache.org/<br>
</p>
<p><br></p>

```

Server software: Apache/2.4.25
server Hostname: educacionadistantia.juntadeandalucia.es
HTTP/1.1 Protocol: TCP/1.1
Document Path: /cursos/login/index.php
Document Length: 5981 bytes
Concurrency Level: 10
Time per request: 222.659 [ms] (mean)
Time per request: 44.433 [ms] (mean, across all concurrent requests)
Transfer rate: 806.93 [Kbytes/sec] received

Completed 1000 requests
Completed 200 requests
Completed 400 requests
Completed 600 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Time per request: 222.659 [ms] (mean)
Time per request: 44.433 seconds
Complete requests: 1000
Total transferred: 5642120 bytes
HTML transferred: 5642120 bytes
Requests per second: 22.30 [#/sec] (mean)
Time per request: 222.659 [ms] (mean)

Elaboración propia. Resultado Test de Rendimiento ([GNU/GPL](#))
[Descripción Larga](#)

```

Time per request: 222.659 [ms] (mean)
Time per request: 44.433 [ms] (mean, across all concurrent requests)
Transfer rate: 806.93 [Kbytes/sec] received

Connection Times (ms)
              min  mean  median  max
Connect:  100  363.9  721  31879
Processing: 106  434 215.1  364  1595
Waiting:   55  193 180.8  128  1297
Total:    289  199 3646.4  646  52555

Percentage of the requests served within a certain time (ms)
  50%   813
  60%   813
  75%  1034
  80%  1138
  85%  1244
  90%  1351
  95%  7885
  98%  8725
  99%  9348
 100% 32335 (longest request)

```

Elaboración propia. Resultado Test de Rendimiento 2 ([GNU/GPL](#))
[Descripción Larga](#)

También contamos con la opción de poder exportar los resultados a un fichero [HTML](#) (Lenguaje de Marcas de Hipertexto). Para ello debemos lanzar el mismo comando que para el test, pero vamos a añadir dónde queremos que se almacene la salida así como el parámetro -w de escritura. Sería de la siguiente manera:



Una vez lancemos dicho comando, se generará un fichero "resultados.html" con el contenido y lo tendrás en el directorio que lo hayas lanzado. Para ver las carpetas y ficheros lanzaremos el comando dir.



El contenido del fichero será el siguiente, bien puedes verlo con el comando gedit o mediante un navegador como Firefox (imagen ampliada):

```

<html><head><title>ApacheBench</title></head><body>
<pre>
This is ApacheBench, Version 2.3 <devRevision: 1807734 $sot:>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
<DPL>
Licensed to The Apache Software Foundation, http://www.apache.org/<br>
</p>
<p><br></p>

```

ApacheBench 2.3	
Server Software:	Apache/2.4.25
Server Hostname:	educacionadistantia.juntadeandalucia.es
Document Path:	/cursos/login/index.php
Document Length:	5981 bytes
Concurrency Level:	10
Time per request:	222.659 [ms] (mean)
Time per request:	44.433 [ms] (mean, across all concurrent requests)
Transfer rate:	806.93 [Kbytes/sec] received
Completed 1000 requests	
Completed 200 requests	
Completed 400 requests	
Completed 600 requests	
Completed 800 requests	
Completed 900 requests	
Completed 1000 requests	
Time per request: 222.659 [ms] (mean)	
Time per request: 44.433 seconds	
Complete requests: 1000	
Total transferred: 5642120 bytes	
HTML transferred: 5642120 bytes	
Requests per second: 22.30 [#/sec] (mean)	
Time per request: 222.659 [ms] (mean)	

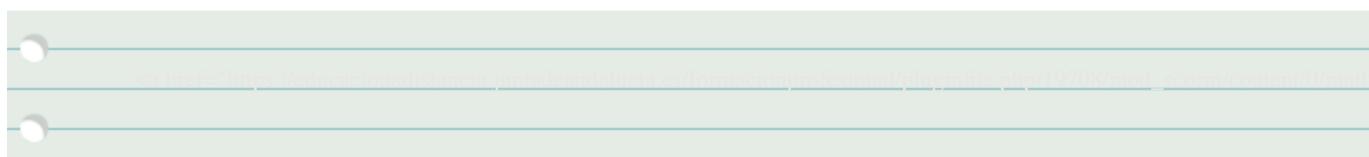
Elaboración propia. Contenido HTML ([GNU/GPL](#))
[Descripción Larga](#)

4.1.2.- Visualización Gráfica con GNuplot

Para que sea más visual, podemos contar con la opción de poder ver nuestro resultado del test de una forma gráfica. Para ello debemos instalar una herramienta denominada  **GNuplot**



Una vez instalada la herramienta, vamos a mostrar gráficamente el test realizado con ApacheBench. Para ello en primer lugar, lanzaremos nuevamente el test y lo guardaremos, lanzando el siguiente comando, en un fichero con extensión .tsv (*Tab Separated Values* - Valores separados por comas).



Ahora debemos crear un fichero gnuplot de configuración, el cual podemos llamar por ejemplo "gnuplot1.p". Se recomienda realizarlo en el mismo directorio que resultados .tsv .



El contenido del fichero debe ser:

```
Abre □ Guardar □ gnuplot1.p
set terminal png size 600,400
set output "monitorizacion_web.png"
set title "1000 peticiones, 50 peticiones concurrentes"
set size ratio 0.6
set grid y
set xlabel "# de peticiones"
set ylabel "Tiempo de respuesta (ms)"
plot "resultados.tsv" using 9 smooth sbezier with lines title
"educacionadistancia.juntadeandalucia.es"
exit
```

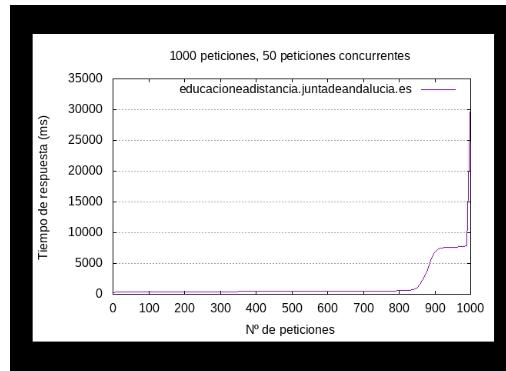


Elaboración propia. Contenido del fichero de configuración. [GNU/GPL](#)
[Descripción Larga](#)

A continuación, ya sólo nos quedaría lanzar gnuplot y así se generará el fichero gráfico.



La gráfica sería la siguiente:



Elaboración propia. [Gráfica del test \(GNU/GPL\)](#)
[Descripción Larga](#)



Para saber más

Gnuplot es una herramienta muy útil y multiplataforma. Te recomiendo que entres en el siguiente enlace para ampliar la información respecto la herramienta:

 [Página Oficial de Gnuplot](#)

4.2.- Herramientas de monitorización avanzadas (nagios y centreon)

En esta unidad vamos a trabajar con herramientas de monitorización más avanzadas como son **Nagios** o **Centreon**. Estas herramientas monitorizan el sistema de una forma automática, y en el caso de que existan fallos nos avisan para solucionar el problema.

Nagios nos permite monitorizar los diferentes servicios de una red y dispositivos, llegando hasta el mínimo detalle de éstos, avisando de la existencia de algún fallo, si se da el caso. Se instala en una máquina GNU/Linux y obtiene la información de los clientes conectados en red, trabajando así en una arquitectura cliente/servidor. Puede trabajar con cualquier dispositivo con protocolo SNMP.



Para saber más

Para ampliar más sobre Nagios y profundizar recomiendo los siguientes enlaces:

[Tutorial de Nagios](#)

[Monitorización de Nagios en Linux](#)

[Monitorización de Nagios en Windows](#)

4.2.1.- Instalación de Nagios

En primer lugar, antes de iniciar la instalación de Nagios, debemos realizar la instalación de una serie de paquetes necesarios para ello. La distribución utilizada es en **Fedora**.

- `yum install httpd`
- `yum install perl`
- `yum install libxml2-devel`
- `yum install libcurl-devel`

Procedemos a la instalación automática:

- `yum install nagios`
- `yum install nagios-nagiosini`
- `cd /etc/nagios`
- `cp config.nagios config.nagios.bak`
- `cp config.nagiosini config.nagiosini.bak`

Ahora vamos a iniciar el sistema de Nagios, pero antes debemos revisar los archivos de configuración:

- `vi /etc/nagios/nagios.cfg`
- `vi /etc/nagios/nagiosini.cfg`
- `/etc/init.d/nagios start`
- `service nagios start`
- `ps aux | grep nagios`
- `netstat -an | grep :80`

Una vez instalado, nos queda configurar la interfaz web. Para ello se crea el siguiente fichero de configuración en apache / etc/http/conf.d/nagios. Es necesario que lo tengamos para que se pueda acceder desde cualquier IP, por lo tanto, debe quedar de la siguiente manera:

SuerteAlas /etc/nagios/cgi-bin/nagios/nagios.cgi?cmd=

Options Page[0]

Options Help[0]

Options Config[0]

Options Plugins[0]

Options PluginsDir[0]

AuthName "nagios"

Require valid-user

AllowOverride All[0]

Options None[0]

DirectoryIndex index.php[0]

Options Indexes[0]

AuthType Basic[0]

AuthName "nagios"

Require valid-user

Es necesario crear un usuario y darlo de alta en el fichero admin. Puesto que la web de nagios tiene un acceso limitado a los usuarios autenticados que se encuentran dentro en el fichero /etc/nagios/passwd<code>.

/etc/nagios/

llamamos el siguiente comando:

chmod 700 /etc/nagios/nagios.cfg.html[0]

/etc/nagios/

chmod 700 /etc/nagios/nagios.cfg.html[0]

chmod 700 /etc/nagios/nagios.cfg.html[0]

/etc/nagios/

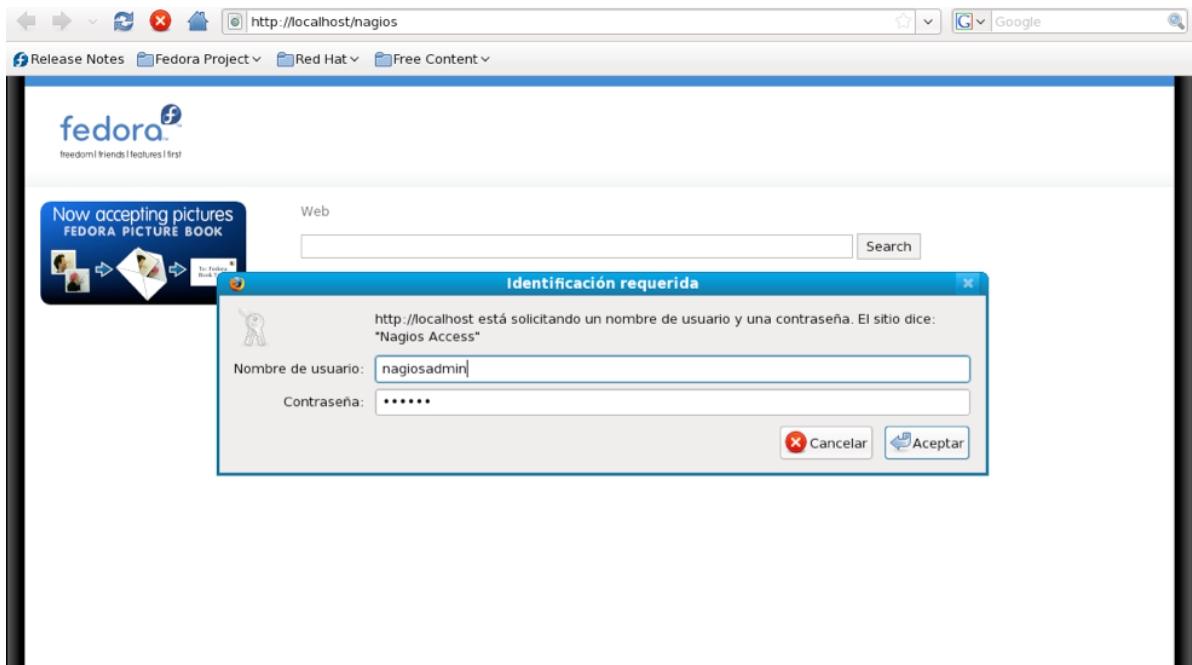
chmod 700 /etc/nagios/nagios.cfg.html[0]

chmod 700 /etc/nagios/nagios.cfg.html[0]

chmod 700 /etc/nagios/nagios.cfg.html[0]

Reiniciamos nuestro servidor y ya podremos acceder a Nagios a través de un navegador web mediante la dirección  <http://localhost/nagios> e introducimos usuario y contraseña.

Autentificación:



Elaboración propia. Acceso a la interfaz web de Nagios ([GNU/GPL](#))

Dentro de la interfaz web de Nagios, cabe destacar las partes más importantes como son:

- ✓ **Map:** que te permite ver el diagrama de estado de los distintos equipos monitorizados.
- ✓ **Tactical overview:** muestra un resumen de todos los equipos y servicios que se están monitorizando.
- ✓ **Host y Services:** listado de todos los hosts y servicios que está monitorizando.
- ✓ **Problems:** indica los distintos equipos o servicios que están dando error.
- ✓ **Reports:** herramienta donde generar tus informes o formularios.
- ✓ **System:** información general sobre el sistema.

4.2.2.- Herramienta Web Centreon

Existen distintos proyectos libres con el objetivo de mejorar la interfaz web y facilitar la configuración de Nagios directamente desde la web. Para ello, el más utilizado es **Centreon**. Por lo tanto, podemos definir **Centreon** como una herramienta web utilizada por Nagios para la simplificación de su configuración así como una mayor funcionalidad.



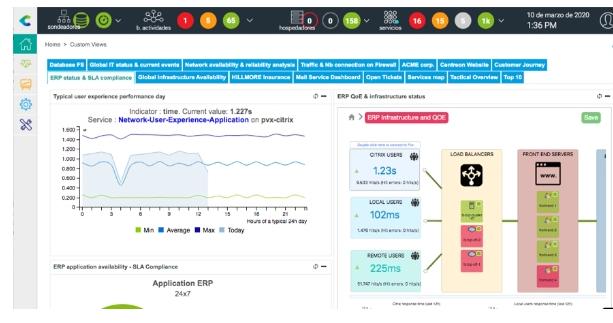
[Elaboración propia](#). Web de Centreon (Todos los derechos reservados) [Descripción larga](#)

La interfaz de Centreon es intuitiva y resulta fácil de usar. Si vas navegando por las distintas pestañas que te proporciona podrás ver:

- ✓ **Home:** página de inicio, donde puedes ver el estado de los equipos y servicios que se están monitorizando.
- ✓ **Monitoring:** podemos observar un listado detallado de todos los servicios asociados a cada equipo. Mostrando su estado, el tiempo que lleva el servicio monitorizado, el último chequeo y la información de salida
- ✓ **Reporting:** tendrás distintas opciones de informes y gráficas para ver los resultados.
- ✓ **Configuration:** lugar donde podemos configurar todos los equipos y servicios que monitoriza Nagios.
- ✓ **Administration:** podremos configurar todas las opciones que presenta Centreon.

A screenshot of the Centreon Monitoring interface. At the top, there's a header with the Centreon logo, a poller status summary (Poller States), a host count (0 Hosts), a service count (0 Services), and navigation links for Home, Monitoring, Reporting, Configuration, and Administration. Below the header, there are links for Custom Views and Poller Statistics. The main content area shows a 'Home' page with a single line of text.

[Elaboración propia](#). Opciones de Centreon (Todos los derechos reservados)



Elaboración propia. Visualización Centreon (Todos los derechos reservados)

[Descripción larga](#)



Recomendación

Te recomiendo el siguiente enlace sobre Centreon que te ayudará con la herramienta:

[Primeros pasos en Centreon](#)

Condiciones y términos de uso de los materiales

Materiales desarrollados inicialmente por el Ministerio de Educación, Cultura y Deporte y actualizados por el profesorado de la Junta de Andalucía bajo licencia Creative Commons BY-NC-SA.



Antes de cualquier uso leer detenidamente el siguiente [Aviso legal](#)

Histórico de actualizaciones

Versión: 02.00.00	Fecha de actualización: 25/03/20	Autoría: María Yolanda Jiménez Capel
<p>Ubicación: 1</p> <p>Mejora (tipo 2): Repasar los comandos y actualizar las versiones.</p> <p>Ubicación: No especificada.</p> <p>Mejora (tipo 2): En DAW05 se habla de servicios de red implicados en el despliegue de aplicaciones web. Se centra exclusivamente del servidor DNS y ni si quiera menciona la posibilidad de obtener con el servidor DNS balanceo de carga.</p> <p>Ubicación: No especificada.</p> <p>Mejora (tipo 3): No se utilizan herramientas para evaluar el rendimiento de los servicios web (RA3).</p> <p>Ubicación: UNIDAD 04</p> <p>Mejora (Mapa conceptual): Actualización del mapa con los nuevo epígrafes, unión y esquematización de toda la unidad de una manera mas visual.</p> <p>Ubicación: UNIDAD 04</p> <p>Mejora (Orientaciones del alumnado): Se ha tenido que actualizar los contenidos para el alumnado en los puntos ORIENTACIONES, DATOS GENERALES DE LA UNIDAD DE TRABAJO ASÍ COMO SUS OBJETIVOS debido a que se han añadido nuevos apartados y subapartados.</p>		
<p>Versión: 01.00.00</p> <p>Fecha de actualización: 30/10/13</p>		
Versión inicial de los materiales.		