

REDES: Clase 28

CONFIGURACIÓN DE DNS

[UNIVERSIDAD TECNOLÓGICA NACIONAL]



[FACULTAD REGIONAL DELTA]

CONFIGURACIÓN DE DNS



ÍNDICE

Composición de un Domain Name	3
Domain Name System	3
¿Por qué se necesita el DNS?	6
Archivos de la base de datos (database files)	9
Archivo /etc/bind/named.conf	10
Registros del servidor de nombres	10
Registro "SOA"	11
Registro "NS"	13
Registro "MX"	14
Registro "A"	14
Registro "Alias"	15
Archivo aula1.zone completo	16
Registro "PTR"	16
Archivo aula1.rev completo	17
Configuración del archivo /etc/resolv.conf	18
Levantando el servicio	19
Consultas al servidor de nombres	19
Consideraciones finales	21
Configuración DNS esclavo	21

CONFIGURACIÓN DE DNS

Composición de un Domain Name

Antes de arrancar, vamos a ver como se compone un dominio de internet, cuyo propósito principal es traducir los nombres de dominio a direcciones IP. Por ejemplo, si tenemos la URL `http://www.carreralinux.com.ar/index.html`, tenemos que el dominio es `www.carreralinux.com.ar`.

Esto se puede descomponer en dos partes:

- una primera parte que es el **dominio**: `carreralinux.com.ar`
- y otra parte que es el **host**: `www`

A su vez, el dominio se divide en:

- **ccTLD** (country code TLD): `.ar`
- **GTLD** (generic Top Level Domain): `.com`
- **domain**: `carreralinux`



Y luego el host, que en este caso es `www`. Por lo que cuando realizamos una búsqueda de una URL, por ejemplo la anterior, estamos diciendo **buscar el host “www” dentro de la organización “carreralinux.com.ar”**.

Domain Name System



El **Sistema de Dominio de Nombres (DNS)** es una **base de datos distribuida**, que **contiene los nombres y números de IP de los equipos que forman parte de una red**. Esta distribución se da cuando se delega un segmento, separado por `“.”` a otro servidor DNS.

Facilita el control local de la totalidad de segmentos de red de datos, permitiendo que cada segmento esté disponible a través de la red mediante esquema de cliente-servidor.



Los **Servidores de Nombres** contienen información sobre un determinado segmento que está disponible para todos los clientes. En la jerga se lo denomina “resolver”.



Los **Resolvers** son muchas veces rutinas de librerías que crean consultas y las envían a través de la red a un Servidor de Nombres que las resuelve.

La totalidad de la base de datos se muestra como un árbol (tree) invertido con la raíz (root) en la punta. El nombre de la raíz es la etiqueta NULL. Se escribe con un solo punto (“.”). Cada nudo del árbol representa tanto una partición de la totalidad de la base de datos, como un Dominio (domain) del Sistema de Nombre de Dominio. En adelante cada dominio puede ser un Subdominio (subdomain).

Cada dominio es marcado por una etiqueta que lo identifica y lo asocia a su dominio parental. El dominio posee un nombre (domain name), que identifica su posición en la base de datos, tal y como una ruta absoluta de un directorio, especifica su lugar en el sistema de archivos de nuestra computadora.

En el DNS el nombre de dominio completo es una secuencia de etiquetas. Empezando por el dominio hasta la raíz (root). Cada etiqueta está separada de la otra por puntos “.”. Por ejemplo red.carreralinux.com.ar.



La ventaja de utilizarlo así es que permite que **cada dominio pueda ser administrado por una organización diferente. Cada organización puede dividir su dominio en varios sub-dominios**, cuya administración puede ser realizada por otras organizaciones.

El Network Information Center del dominio “com.ar” (comercial) pasa la autoridad sobre el subdominio “carreralinux” al Instituto para que pueda administrar el siguiente subdominio: “red.carreralinux.com.ar”. En Argentina, la entidad encargada del dominio “.” es Nic <http://nic.ar>. Ellos delegan en los proveedores de internet (isp) parte del árbol de resolución. Por lo que la resolución de dominios es una consulta progresiva que se da de derecha a izquierda. Esto es, si tomamos nuevamente “red.carreralinux.com.ar.”

- 01 Primero pregunta a los DNS root (.) quien tiene autoridad sobre la zona *ar*
- 02 Luego pregunta a los DNS que tienen autoridad sobre *ar*, quien tiene autoridad sobre *.com.ar*
- 03 Luego pregunta a los DNS que tienen autoridad sobre *.com.ar*, quien tiene autoridad sobre *carreralinux.com.ar*
- 04 Ahí es donde finalmente tenemos control. Por lo que podremos crear un host, *www.carreralinux.com.ar* o un subdominio *red.carreralinux.com.ar*



Entonces veamos dos cosas:

La primera es que, *ar* es subdominio de “.”, y luego *.com.ar* es subdominio de *ar*.

Y la segunda es que cada servidor DNS delega el segmento en otro servidor DNS. Por ejemplo, NIC Argentina delego el segmento “carreralinux” del dominio “carreralinux.com.ar” a los DNS que especificamos, configuramos y mantenemos en la Academia.

Un dominio puede contener tanto subdominios como hosts. Un host también puede tener uno o más Alias de Dominio de Nombre, es decir un indicador de un nombre de dominio (el alias) para el nombre oficial (canonical domain name).

Por ejemplo:

Si su esposa se llama Ana Paula, en algunos momentos la llamará Ana y en otros Paula. Aunque utilice nombres diferentes se refiere a la misma persona. Las organizaciones de un dominio son libres de elegir nombres dentro de su dominio. No importa cual nombre sea usado, es seguro que no causa conflicto con otro nombre, porque tiene su Nombre de Dominio único adjuntado al final. De este modo pueden existir dos hosts con el nombre equipo1 en nuestra red, por ejemplo, siempre y cuando los dominios paternos sean diferentes.

Es decir que los paquetes que viajen desde equipo1.aula1.red.carreralinux.com.ar siempre van a encontrar su camino a equipo1.aula2.red.carreralinux.com.ar, porque se trata de dominios paternos diferentes: aula1.red.carreralinux.com.ar y aula2.red.carreralinux.com.ar

¿POR QUÉ SE NECESITA EL DNS?



Se necesita el DNS porque para el hombre le resulta mucho más fácil recordar nombres en vez de cifras y aún más, todo cuando se trata de una cantidad de direcciones tan inmensa como las que contiene la red de redes Internet. Las computadoras, por otro lado, trabajan perfectamente con cifras como la dirección IP.

Lo que sucede cuando nosotros nos conectamos a Internet y colocamos una dirección en nuestro navegador como por ejemplo:

`http://www.carreralinux.com`



Dirigimos una petición (request) al Servidor de Dominio de su proveedor, éste intenta resolver el nombre de dominio con la IP correspondiente. Si nuestro proveedor no está autorizado para resolver nombres en esta zona, retransmite la petición (request, query) al servidor de dominio autorizado.

¿Cómo sabe nuestro equipo recién conectado a Internet a qué servidor de nombres debe preguntarle cómo resolver la dirección?

Porque tiene la respuesta en uno de los archivos de configuración:

```
/etc/resolv.conf
```



Cuando nos conectamos a Internet, este archivo es transferido por el Servidor de dhcp.

Pasemos ahora a trabajar con una copia de un archivo resolv.conf de un equipo que no trabaja en una red local:

```
search red.carreralinux.com.ar
nameserver 10.10.3.1
nameserver 8.8.8.8
nameserver 200.49.159.69
```

Cuando un equipo está conectado a una red local debemos agregarle los números IP de los servidores locales al archivo para que funcione:

```
search red.carreralinux.com.ar
nameserver 10.10.3.1
nameserver 8.8.8.8
nameserver 200.49.159.69
```



Cuando una petición (request) se dirige a una zona que está fuera de la zona autorizada, nuestro servidor sabe por dónde buscar gracias a los datos que hemos incorporado.

Algunas veces la petición (request) de una dirección tiene que pasar por varios Servidores de Dominio hasta que toma contacto con el destino solicitado. Aunque supiéramos la dirección IP de destino, es imprescindible consultar otros Servidores de Dominio si nuestro equipo no se encuentra en la misma zona.

Es fácil imaginar por qué el Sistema de Dominio de Nombre no puede consistir en una sola base de datos centralizada. Primero, porque tardaría mucho tiempo en encontrar un servidor entre millones de otros y, segundo, habría una cola muy larga de peticiones simultáneas de todo el mundo. Adicionalmente, no tendría sentido dirigirse a un servidor lejano para comunicar con un host de la misma zona.

Hasta ahora hablamos del mapeo de nombres a direcciones. Pero, ¿qué sucede si tenemos la dirección IP y queremos saber el nombre de dominio?

Para solventar este problema fue creado el dominio “in-addr.arpa”.



Este dominio es llamado dominio inverso y la resolución de direcciones IPs a nombres de dominio se denomina mapeo reverso (**reverse mapping o reverse lookup**).



El dominio de nombre inverso se crea escribiendo las cifras de la dirección IP en orden contrario y añadiendo **in-addr.arpa** al final.


Veamos esto en el ejemplo que sigue a continuación:

Si acordamos que el número IP de equipo1 del dominio Carrera Linux es “10.10.1.1” con nombre de dominio “aula1.red.carreralinux.com.ar”.

El dominio “aula1.red.carreralinux.com.ar” tendrá el nombre de dominio inverso: “1.10.10.in-addr.arpa” y el dominio inverso de la computadora equipo1.aula1.red.carreralinux.com.ar será: “1.1.10.10.in-addr.arpa”.

ARCHIVOS DE LA BASE DE DATOS (DATABASE FILES)

Definamos cuáles van a ser los archivos que intervienen en el proceso de resolución de nombres.




named.conf

Este archivo tiene la información de las zonas de mapeo directo e inverso y dónde están ubicadas. En GNU/Debian todos los archivos están almacenados en el directorio `/etc/bind`.



Un archivo proyecta todos los nombres de hosts a direcciones IP, mientras que otros archivos vuelven a proyectar las IPs a nombres de hosts.



La búsqueda de direcciones IP a nombres de hosts se llama **reverse mapping** (mapeo reverso). Cada red tiene su propio archivo para el mapeo reverso.

Vamos a ponernos de acuerdo en cómo se van a llamar los archivos:

- **Mapeo directo:** `aula1.zone`
- **Mapeo reverso:** `aula1.rev`

Archivo /etc/bind/named.conf



La definición de zona es muy importante porque es el nombre del dominio en el que van a trabajar los host.

La variable file contiene el nombre del archivo donde se van a guardar los registros:

```
zone "aula1.red.carreralinux.com.ar" {  
    type master;  
    file "/etc/bind/aula1.zone.";  
};  
zone "10.in-addr.arpa" {  
    type master;  
    file "/etc/bind/aula1.rev";  
};
```

REGISTROS DEL SERVIDOR DE NOMBRES



Los archivos **aula1.zone** y **aula1.rev** contienen los componentes que en DNS llamamos resource records y estos podríamos definirlo como *“que le queremos preguntar”*.

Según las Referencias de DNS, los resource records tienen el orden siguiente:

- **Registro SOA:** indica la autoridad para estos datos de dominio, si quisiéramos saber quien tiene autoridad sobre este dominio
- **Registro NS:** indica el servidor de nombre para este dominio, si quisiéramos saber quien resuelve los nombres de este dominio. Con esto se inicia todo, por lo que primero averiguamos el resource NS y luego el resource A que veremos más adelante.
- **Registro MX:** indica el servidor de correos para ese dominio, si quisiéramos saber quien maneja los mails de este dominio

Los siguientes registros indican datos de hosts en este dominio:

- **A:** mapeo de nombres a direcciones, si quisiéramos saber la dirección IP de un host. Esta es la más típica.
- **PTR:** mapeo de direcciones a nombres (mapeo reverso), si quisiéramos saber el host asociado a una IP.
- **CNAME:** canonical name (nombre oficial para alias). Esta es una respuesta y es que hay una alias como veremos más adelante.
- **TXT:** información de texto.
- **RP:** persona responsable.

REGISTRO “SOA”



El primer resource record de cada archivo de base de datos es el SOA record (start of authority). **El SOA record indica que este Servidor de Nombres es la mejor fuente de información para los hosts dentro de este dominio.**

Nuestro servidor de nombres (ns1) es autorizado para el dominio “aula1.red.carrera-linux.com.ar” debido al record SOA:

```
@ IN SOA ns1.red.carreralinux.com.ar. admin.red.carreralinux.com.ar. (
1      ; Serial for updates
10800  ; Refresh after 3 hours
3600   ; Retry after 1 hours
604800 ; Expire after 1 week
86400) ; Minimum TTL of 1 week
```

El signo “@” de la primera columna se reemplazará en las consultas por el nombre del dominio en este caso “aula1.red.carreralinux.com.ar”, es por eso que debe estar en la primera columna. La variable IN se usa para Internet. Existen otras clases, pero ninguna es de uso común. El primer nombre después de SOA, ns1.red.carreralinux.com.ar, es el nombre del Servidor de Nombres para estos datos. El segundo nombre, admin.red.carreralinux.com.ar, es la dirección de correo de la persona encargada de estos datos.

El registro SOA es el más complejo guarda datos cada cuanto hay que transferir una zona hasta cuanto tiene que almacenarse una consulta a este servidor.

Número de serie: es el número de la revisión de este archivo de zona. Tenemos que incrementar este número cada vez que se cambie el archivo de zona. Es importante incrementar este valor cada vez que se realice un cambio, para que los cambios se distribuyan a los servidores DNS secundarios.

Tiempo de actualización: el tiempo en segundos que un servidor DNS secundario espera antes de consultar el registro SOA del servidor DNS principal para comprobar los cambios. Cuando caduca el tiempo de actualización, el servidor DNS secundario solicita una copia del registro SOA actual del primario. El servidor DNS principal cumple con esta solicitud. El servidor DNS secundario compara el número de serie del registro SOA actual del servidor DNS principal y el número de serie en su propio registro SOA. Si son diferentes, el servidor DNS secundario solicitará una transferencia de zona desde el servidor DNS principal. El valor predeterminado es 3.600.

Tiempo de reintento: el tiempo en segundos que un servidor secundario espera antes de volver a intentar una transferencia de zona que ha fallado. Normalmente, el tiempo de reintento es menor que el tiempo de actualización. El valor predeterminado es 600.

Tiempo de expiración: el tiempo en segundos que un servidor secundario seguirá intentando completar una transferencia de zona. Si este tiempo transcurre antes de una transferencia de zona exitosa, el servidor secundario caducará su archivo de zona. Esto significa que el secundario dejaría de responder las consultas, ya que considera sus datos demasiado antiguos como para ser confiables. El valor predeterminado es 86.400.

TTL mínimo: el valor mínimo de tiempo de vida se aplica a todos los registros de recursos en el archivo de zona. Este valor se proporciona en respuestas a consultas para informar a otros servidores de cuánto tiempo deben mantener los datos en caché. El valor predeterminado es 3.600.



¡Es muy importante poner un punto al final de los nombres! Si no es añadido automáticamente el dominio “aula1.red.carreralinux.com.ar”, lo que tendría poco sentido.



Los paréntesis permiten que el SOA record se extienda sobre varias líneas. La mayoría de los renglones dentro de los paréntesis sirven para informar a los Servidores de Nombres secundarios.

REGISTRO “NS”

El próximo renglón que añadimos a cada archivo de base de datos es el NS record (name server: Servidor de Nombres). Los NS records de nuestro dominio son:

```
@ IN NS ns1.aula1.red.carreralinux.com.ar.  
@ IN NS ns2.aula1.red.carreralinux.com.ar.
```

Estos records indican que existen dos Servidores de Nombres para el dominio “aula1.red.carreralinux.com.ar”. Los Servidores de Nombres se encuentran en los hosts “ns1” y “ns2”.

REGISTRO "MX"

El MX record indica a la red cuales son los servidores de correo que se van a usar para el envío. Estos servidores normalmente son dos y tienen un número de prioridad. El que tiene el número de prioridad más baja es el encargado de procesar el correo de la red:

```
@ IN MX 10 relay1.aula1.red.carreralinux.com.ar.  
@ IN MX 20 relay2.aula1.red.carreralinux.com.ar.
```

En nuestra red tenemos dos servidores de correo llamados "relay1" prioridad más baja y "relay2" con prioridad más alta.

REGISTRO "A"

El paso siguiente es el mapeo de direcciones a nombres. Añadimos los siguientes resource records al archivo:

```
ns1      IN  A    10.10.1.1  
ns2      IN  A    10.10.1.2  
relay1   IN  A    10.10.1.3  
relay2   IN  A    10.10.1.4  
equipo1  IN  A    10.10.3.5  
equipo2  IN  A    10.10.3.6
```



La "A" indica dirección (address) y cada resource record mapea un nombre a una dirección.

REGISTRO "ALIAS"

Para los primeros dos alias creamos "CNAME" (canonical name, fully qualified host name = nombre de host de primera orden) resource record:

equipo1	IN	A	10.10.3.5
equipo2	IN	A	10.10.3.6
www.equipo1	IN	CNAME	equipo1
www.equipo2	IN	CNAME	equipo2



Cuando un Servidor de Nombre busca un nombre y encuentra un CNAME-record correspondiente, reemplaza el nombre con el nombre de host de primera orden y sigue buscando el nuevo nombre.

En este caso hacemos que el host equipo1 también pueda llamarse www.equipo1, esto nos va a servir cuando tengamos que colocar en cada equipo un webserver y poder ubicarlo con un nombre que empiece con www.

ARCHIVO AULA1.ZONE COMPLETO

```
@ IN SOA ns1.red.carreralinux.com.ar. admin.red.carreralinux.com.ar. (
    1          ; Serial for updates
    10800      ; Refresh after 3 hours
    3600       ; Retry after 1 hours
    604800     ; Expire after 1 week
    86400)     ; Minimum TTL of 1 week

@      IN  NS   ns1.aula1.red.carreralinux.com.ar.
@              IN  NS   ns2.aula1.red.carreralinux.com.ar.
@      IN  MX   10    relay1.aula1.red.carreralinux.com.ar.
@      IN  MX   20    relay2.aula1.red.carreralinux.com.ar.
ns1      IN  A    10.10.3.1
ns2      IN  A    10.10.3.2
relay1 IN  A    10.10.3.3
relay2 IN  A    10.10.3.4
equipo1  IN  A    10.10.3.5
equipo2  IN  A    10.10.3.6
```

REGISTRO "PTR"

Ahora creamos los mapeos de dirección a nombre. El archivo aula1.rev mapea direcciones a nombres de hosts para la red 10.10.1.0.

El resource record para este mapeo es el PTR (pointer = puntero, indicador) record. Existe un record para cada host de esta red.



Recordemos que en DNS estamos buscando nombres para direcciones.

La dirección está escrita en orden inverso y es adjuntada a in-addr.arpa:

```
1 IN PTR ns1.aula1.red.carreralinux.com.ar.
2     IN PTR ns2.aula1.red.carreralinux.com.ar.
3     IN PTR relay1.aula1.red.carreralinux.com.ar.
4     IN PTR relay2.aula1.red.carreralinux.com.ar.
5 IN PTR equipo1.aula1.red.carreralinux.com.ar.
6 IN PTR equipo2.aula1.red.carreralinux.com.ar.
```

ARCHIVO AULA1.REV COMPLETO

```
@ IN SOA ns1.red.carreralinux.com.ar. admin.red.carreralinux.com.ar. (
    1          ; Serial for updates
    10800      ; Refresh after 3 hours
    3600       ; Retry after 1 hours
    604800     ; Expire after 1 week
    86400)     ; Minimum TTL of 1 week

@     IN  NS   ns1.aula1.red.carreralinux.com.ar.
@     IN  NS   ns2.aula1.red.carreralinux.com.ar.
1     IN  PTR   ns1.
2     IN  PTR   ns2.
3     IN  PTR   relay1.
4     IN  PTR   relay2.
5     IN  PTR   equipo1
6     IN  PTR   equipo2.
```



CONFIGURACIÓN DEL ARCHIVO /ETC/RESOLV.CONF

Ahora estamos en condiciones de configurar nuestro archivo resolv.conf para que podamos resolver nombres de dominio.

En el archivo configuramos la variable search apuntando a nuestro dominio, esto lo hacemos para poder usar los nombres de host para acceder a un servicio, olvidándonos del dominio al cual pertenece.

El registro nameserver lo apuntamos al servidor de nombres ns1 de nuestra red y al segundo servidor de nombres, para poder resolver nombres de dominio fuera de nuestra red usamos el nameserver de google 8.8.8.8.

```
search aula1.red.carreralinux.com.ar
nameserver 10.10.3.1
nameserver 10.10.3.2
nameserver 8.8.8.8
```



Estamos poniendo en primer lugar el número IP de nuestro servidor de nombres, y después los de nuestro proveedor de Internet.

LEVANTANDO EL SERVICIO

Para levantar el servicio debemos ejecutar el siguiente comando:

```
# /etc/init.d/bind9 start
```

o bien usar

```
# systemctl start bind9
```

CONSULTAS AL SERVIDOR DE NOMBRES

Ahora queremos saber si nuestro servidor de nombres está funcionando correctamente, para eso vamos a usar el comando host.

Este comando es muy claro a la hora de devolvernos la información. Supongamos que queremos averiguar la IP de alguien. Por ejemplo que IP le corresponde a `www.arnet.com.ar`:

```
~$ host www.arnet.com.ar
www.arnet.com.ar A 200.45.190.232
```

Ahora probemos la misma consulta con un nombre en nuestra red:

```
~$ host www.equipo1.red.carreralinux.com.ar
www.equipo1.red.carreralinux.com.ar A 10.10.3.1
```

Esta es la primera función que cumple el DNS, traducir el nombre en una IP. Ahora queremos saber ¿qué DNS lo está apuntando? Es decir: ¿Quién lo maneja?

```
~$ host -t ns arnet.com.ar
arnet.com.ar NS oktubre.arnet.com.ar
arnet.com.ar NS gulp.arnet.com.ar
```

Este comando nos devuelve el nombre de los dos servidores de nombres de arnet. Nos otorga el nombre que se le ocurre al administrador, generalmente se les coloca ns1 y ns2. Ese no es el caso de arnet como podemos ver. Ahora vamos a ejecutar la misma consulta usando nuestro servidor:

```
~$ host -t ns red.carreralinux.com.ar
red.carreralinux.com.ar NS ns1.red.carreralinux.com.ar
red.carreralinux.com.ar NS ns2.red.carreralinux.com.ar
```

También podríamos preguntarnos quién es el encargado de gestionar el correo en arnet. Veamos cómo podríamos averiguarlo:

```
~$ host -t mx arnet.com.ar
arnet.com.ar MX 5      lobosuelto.arnet.com.ar
arnet.com.ar MX 15     corderoatado.arnet.com.ar
```



Esto nos devuelve los dos servidores de correo. Observemos que tienen dos servidores: uno funciona como master y otro como esclavo.

Nos damos cuenta de esto porque cada uno tiene una prioridad. El que tiene el número de prioridad más chico es el servidor master. Ejecutemos la misma consulta pero a nuestro servidor de nombres:

```
~$ host -t mx arnet.com.ar
red.carreralinux.com.ar MX 10 relay1.red.carreralinux.com.ar
red.carreralinux.com.ar MX 20 relay2.red.carreralinux.com.ar
```

Y si preguntamos por el número de IP...

```
~$ host 200.81.15.141
Name: mail.weimaraners.com.ar
Address: 200.81.15.141
```

Cuando consultamos por el número IP el servidor nos devuelve el nombre del equipo. Ahora vamos a ejecutar la misma consulta usando nuestro servidor de nombres:

```
~$ host 10.10.3.1  
Name: ns1.red.carreralinux.com.ar  
Address: 10.10.3.1
```

CONSIDERACIONES FINALES

Es muy importante para el administrador de redes que el servicio de DNS funcione correctamente. Este servicio junto con el de DHCP son los más importantes para que nuestra red funcione bien.



Es uno de los servicios que más cuesta configurar pero vale la pena el esfuerzo.

Configuración del DNS esclavo

Si deseamos configurar nuestro servidor DNS para que actúe como esclavo de un servidor DNS maestro, la configuración es mucho más sencilla que en el caso anterior ya que únicamente será necesario indicar en el DNS esclavo quién es el servidor DNS maestro, y en el DNS maestro, la IP del DNS esclavo.

Como vimos el nombre del DNS master es ns1.aula1.red.carreralinux.com.ar y su número IP 10.10.3.1 y que el nombre del DNS esclavo es ns2.aula1.red.carreralinux.com.ar

En el archivo `/etc/bind/named.conf` del servidor DNS esclavo debemos indicar que se trata de un servidor esclavo y también debemos indicar quién es el maestro:

```
zone "aula1.red.carreralinux.com.ar" {
    type slave;
    file "/etc/bind/aula1.zone";
    masters { 10.10.3.1; };
};

zone "3.10.10.in-addr.arpa" {
    type slave;
    file "/etc/bind/aula1.rev";
    masters { 10.10.3.1; };
};
```

En el archivo `/etc/bind/named.conf` del servidor DNS maestro podemos utilizar `also-notify` para mantener los DNS sincronizados. Con `also-notify` pasamos los cambios de zonas en el maestro al esclavo:

Archivo `/etc/bind/named.conf` del server master

```
zone "aula1.red.carreralinux.com.ar" {
    type master;
    file "/etc/bind/aula1.zone";
    also-notify {10.10.3.2;}
};

zone "3.10.10.in-addr.arpa" {
    type master;
    file "/etc/bind/aula1.rev";
    also-notify {10.10.3.2;}
};
```

De esta forma dispondremos en la red de un servidor DNS esclavo que podrá satisfacer las peticiones DNS al igual que lo haría el maestro.



Es interesante si el número de peticiones es muy elevado y se requiere distribuir la carga entre los dos servidores, o si deseamos disponer de servicio DNS de alta disponibilidad de forma que, aunque el servidor maestro deje de funcionar, el servidor esclavo podrá seguir ofreciendo el servicio.



Cada vez que hagamos un cambio en los archivos `/etc/bind/aula1.zone` y `/etc/bind/aula2.rev` del maestro, **debemos acordarnos de actualizar el parámetro serial** (incrementar en una unidad) para que los DNS dependientes del maestro sepan que ha cambiado y actualicen su información para mantenerse perfectamente sincronizados.