DNS LINUX BIND

Descargar PDF

INTRODUCCIÓN

Los sistemas Linux incorporan en sus repositorios los paquetes `dnsmasq` y `bind9` para implementar servidores DNS, siendo `bind9` el servidor más utilizado por su pontencia y versatilidad.

INSTALACIÓN

Podemos instalar BIND en prácticamente cualquier distribución Linux, nosotros nos centraremos en las distribuciones Linux Alpine y Linux Debian.

Instalación

En un sistema Linux Alpine

```
root@pc~# apk update && apk add bind
root@pc~# apk update && apk add bind-tools
```

En un sistema Linux Debian

```
root@pc~# apt update && apt install bind9 -y
root@pc~# apt update && apt install dnsutils -y
```

Para comprobar la versión instalada ejecutar el siguiente comando.

```
#Ejemplo salida named-v
 named -v
 BIND 9.16.50
```

Los archivos ejecutables instalados son:

Comando	Ruta más común	Descripción
`named`	`/usr/sbin/`	servidor DNS
`named-checkconf`	`/usr/sbin/`	verifica fichero de configuracion
`named-checkzone`	`/usr/sbin/`	verifica fichero de zona
`named-compilezone`	`/usr/sbin/`	verifica y convierte en binario un archivo de zona
`rndc`	`/usr/sbin/`	control del servicio named

Las utilidades instaladas son:

Comando	Ruta más común	Descripción
`dig`	`/usr/bin/`	consultas y diagnostico detallado de DNS
`nslookup`	`/usr/bin/`	consultas básicas de DNS

Comando	Ruta más común	Descripción
`host`	`/usr/bin/`	consultas rápidas y simples de DNS

Los servicios instalados son:

Sistemas	Servicio
System V	`/etc/init.d/named`
SystemD	`/lib/systemd/system/named.service`

Linux Debian utiliza System V o SystemD dependiendo de la versión del sistema. Los sistemas Linux Alpine utilizan OpenRC una versión ampliada de System V.

Para forzar que el servicio se ejecute solo usando la pila IPv4 debemos modificar los parametros de llamada al servicio y añadir la opción `-4`.

named IPv4

En un servidor DNS sobre un sistema Linux con SystemD se quiere configurar BIND para utilizar solamente la pila de protocolos IPv4.

```
#parar el servicio
systemctl stop named

#editar fichero de opciones del servicio y modificar OPTIONS
nano /etc/default/named
....
OPTIONS="-u bind -4"
....

#iniciar el servicio
systemctl start named
```

Las acciones más comunes para gestioner el servicio DNS son:

Acción	System V	SystemD	OpenRC
Habilitar	`update-rc.d named defaults`	`systemctl enable named`	`rc-update add named default`
Deshabilitar	`update-rc.d -f named remove`	`systemctl disable named`	`rc-update del named`
Iniciar	`service named start`	`systemctl start named`	`rc-service named start`
Parar	`service named stop`	`systemctl stop named`	`rc-service named stop`
Reiniciar	`service named restart`	`systemctl restart named`	`rc-service named restart`
Estado	`service named status`	`systemctl status named`	`rc-service named status`

ARCHIVOS DE CONFIGURACIÓN

Los archivos de configuración se almacenan en la carpeta `/etc/bind/`. A continuación se describen los principales archivos de configuración que permiten una configuración modular y flexible:

NAMED.CONF

El archivo `named.conf` es el archivo de configuración general e incorpora una serie de clausulas `include` que especifican archivos que configuran distintos aspectos del servidor.

named.conf

```
//use this file for global options
include "/etc/bind/named.conf.options";
//use this file for local zones
include "/etc/bind/named.conf.local";
//default zones do not modify
include "/etc/bind/named.conf.default-zones";
```

> **Nota:** la clausula include se puede utilizar en cualquier archivo de configuración. Por ejemplo, podriamos poner dentro del archivo named.conf.options una clasula include al archivo named.conf.acl utilizado para definir reglas de acceso.
> **Nota:** el archivo named.conf.default-zones incluye las zonas por defecto en BIND y no se debe modificar.

named.conf.default.zones

```
//root-servers
zone "."{
    type hint;
    file "/usr/share/dns/root.hints";
};
//RFC 1912
//localhost
zone "localhost."{
    type master;
    file "/etc/bind/db.local";
};
//localhost inverse
zone "127.in-addr.arpa."{
    type master;
    file "/etc/bind/db.127";
};
//net inverse
zone "0.in-addr.arpa."{
    type master;
    file "/etc/bind/db.0";
};
//broadcast inverse
zone "255.in-addr.arpa."{
    type master;
    file "/etc/bind/db.255";
};
```

NAMED.CONF.OPTIONS

El archivo `named.conf.options` es el archivo en el que se **definen las opciones globales** de configuración del servidor DNS, como pueden ser los reenviadores (forwarders), listas de control de acceso (ACL), modo de resolución (recursivo o iterativo), la seguridad avanzada (DNSSEC) o la actualización dinámica (DDNS). Las opciones globales se definen dentro de la clausula **`options`**.Evidentemente, existen números opciones globales que modifican el comportamiento de BIND y se pueden encontrar en la documentación oficial de BIND.

named.conf.local

```
options {
    // Directorio donde BIND buscará archivos de zona
    directory "/var/cache/bind";

    // Manejo de DNSSEC (Validación Automática)
    dnssec-validation auto;

    // Si no se usa el reenvío, esto evita errores RFC1035
    auth-nxdomain no;

    // Escucha en todas las interfaces IPv6
    listen-on-v6 { any; };

    // Permite peticiones de cualquier red
    allow-query { any; };
};
```

NAMED.CONF.LOCAL

El archivo `named.conf.local` es el archivo en el que se **definen las zonas** que gestionará el servidor, aparte de las zonas por defecto que se han creado en el momento de la instalación. Las zonas se definen mediante la clausula `zone`.

named.conf.local

zone define nombres de zona

```
zone "ser.smr."{
    type master;
    file "/etc/bind/db.ser.smr";
};
zone "192.168.0.in-addr.arpa."{
    type master;
    file "/etc/bind/db.192.168.0.in-addr.arpa";
};
```

DB.ZONA

Por cada zona de la que queremos tener resolución se define, mediante de la clausula `file`, un archivo de resolución de zona (`db`). En este archivo se **definen los registros de recursos** que configuran la zona.

named.conf.local

file define registro de recursos

```
zone "ser.smr."{
    type master;
    file "/etc/bind/db.ser.smr";
};
zone "192.168.1.in-addr.arpa."{
    type master;
    file "/etc/bind/db.192.168.1.in-addr.arpa";
};
```

Podemos comprobar la sintaxis de un archivo de configuración mediante el comando `named-checkconf` y la sintaxís de un archivo de recursos de una zona mediante el comando `named-checkzone`.

Comprobación de archivos

```
#comprobar archivo de configuracion opciones globales
named-checkconf /etc/bind/named.conf.options

#comprobar archivo de configuracion de zonas
named-checkconf /etc/bind/named.conf.local

#comprobar fichero de recursos de zona ser.smr
named-chekzone ser.smr /etc/bind/db.ser.smr
```

ZONAS

RESOLUCIÓN

En cuanto a la **resolución del nombre** tenemos 2 tipos de zonas:

DIRECTAS

Son aquellas zonas cuyos registros resuelven la relación Nombre - IP. Es decir, son aquellas zonas a las que se les pregunta el nombre y devuelven la IP correspondiente. Se definen usando el siguiente esquema:

```
//zona directa
zone <dominio.> {
    type {master,slave,stub} ;
    file "/etc/bind/db.<dominio>"
    [opciones]
};
```

Definición zona directa

```
//zona directa para el nombre ser.smr
zone "ser.smr."{
    type master;
    file "/etc/bind/db.ser.smr";
};
```

INVERSAS

Son aquellas zonas cuyos registros resuelven la relación IP - Nombre. Es decir, son aquellas zonas a las que se les pregunta por una IP y devuelven el nombre correspodiente. **Como nombre de zona se utiliza la inversa de la IP de red y el sufijo in-addr.arpa**. Se definen usando el siguiente esquema:

```
//zona inversa
zone <net.in-addr.arpa.> {
    type {master,slave,stub} ;
    file "/etc/bind/db.<net.in-addr.arpa>"
    [opciones]
};
```

Definición zona inversa

```
//zona inversa para la red 192.168.1.0/24
    zone "1.168.192.in-addr.arpa."{
    type master;
    file "/etc/bind/db.1.168.192.in-addr.arpa";
};
```

TIPO

En cuanto al **tipo de zona** (opción `type`) tenemos 4 tipos de zonas:

MASTER

Una zona master define una **zona principal** o primaria sobre la que el servidor DNS tiene **autoridad**. Se definen usando el siguiente esquema:

```
//zona directa
zone <dominio.> {
    type master;
    file "/etc/bind/db.<dominio>";
    [opciones]
};

//zona inversa
zone <net.in-addr.arpa.> {
    type master;
    file "/etc/bind/db.<net.in-addr.arpa>";
    [opciones]
};
```

Zonas Master

Un servidor DNS define la zona ser.smr para la red 192.168.1.0/24. La definición de la zona master directa e inversa sería:

```
zone "ser.smr." {
    type master;
    file "/etc/bind/db.ser.smr";
};
zone "1.168.192.in-addr.arpa." {
    type master;
```

```
    file "/etc/bind/db.1.168.192.in-addr.arpa";
};
```

SLAVE

Una zona slave define una **zona esclava** o secundaria de una zona master sobre la que el servidor DNS tiene **autoridad**. Las zonas esclavas son una **copia completa de una zona master** y por tanto son autoritativas. El proceso por el cual la zona esclava realiza la copia o sincronizaión de la zona master se denomina **transferencia de zona**. Se ha de asegurar que los archivos de recursos de las zonas slave tengan los permisos necesarios para poder ser escritos tras las transferencias de zonas. El directorio `/var/lib/bind/` tiene los permisos adecuados y suele ser el utilizado para guardar estas zonas slaves. El esquema de definición de las zonas slaves es el siguiente:

```
//zona directa
zone <dominio.> {
    type slave;
    file "/var/lib/bind/db.<dominio>";
    masters { <IP-master> ; <IP-master> };
    [opciones]
};
//zona inversa
zone <net.in-addr.arpa.> {
    type slave;
    file "/var/lib/bind/db.<net.in-addr.arpa>";
    masters { <IP-master> ; <IP-master> };
    [opciones]
};
```

Zonas Slave

Un servidor DNS define la zona esclava ser.smr que esta definida en el servidor 192.168.1.254. La definición de la zona slave directa e inversa sería:

```
zone "ser.smr." {
    type slave;
    file "/var/lib/bind/db.ser.smr";
    masters { 192.168.1.254; };
};
zone "1.168.192.in-addr.arpa." {
    type slave;
    file "/var/lib/bind/db.1.168.192.in-addr.arpa";
    masters { 192.168.1.254; };
};
```

Los archivos de registros de las zonas transferidas se guardan en formato RAW para agilizar la carga por parte del servidor DNS, podemos convertirlos a texto usando la utilidad `named-compilezone`.

RAW a TXT

Convertir a TXT el archivo de registros de recursos en formato RAW para la zona esclava ser.smr

```
#named-compilezone -f raw -F text -o <fichero.txt>  <zona>  <fichero.raw>
 named-compilezone -f raw -F text -o db.ser.smr.txt ser.smr /var/lib/bind/db.ser.smr.saved
```

STUB

Una zona stub define una **zona esclava** o secundaria de una zona master sobre la que el servidor DNS tiene **autoridad**. Las zonas stub son una **copia parcial de una zona master**. Estas zonas solo copian los registros SOA y NS del archivo de resolución de la zona master y se utilizan fundamentalmente para facilitar la resolución recursiva de nombres en otra zona sin transferir toda la base de datos de esa zona. Permite que el servidor sepa qué servidores son autoritativos para dicha zona. El esquema de definición es el siguiente.

```
//zona directa
zone <dominio.> {
    type stub;
    file "/var/lib/bind/db.<dominio>";
    masters { <IP-master> ; <IP-master> };
    [opciones]
};
//zona inversa
zone <net.in-addr.arpa.> {
    type stub;
    file "/var/lib/bind/db.<net.in-addr.arpa>";
    masters { <IP-master> ; <IP-master> };
    [opciones]
};
```

FORWARD

Una zona foward define una **zona de reenvío** de otra zona definida en un servidor DNS. Estas zonas no responden autoritativamente ya que no definen la zona sino que reenvían las peticiones a otros servidores que si las definen. No se hace forwarding para zonas inversa. El esquema de definición es el siguiente.

```
//zona directa
zone <dominio.> {
    type forward;
    forwarders { <IP-fowarder> ; <IP-forwarder> };
    foward only;
    [opciones]
};
```

Zona Foward

Un servidor DNS define la zona forward ser.smr que esta definida en el servidor 192.168.1.254. La definición de la zona de forwarding sería:

```
zone "ser.smr." {
    type forward;
    forwarders { 192.168.1.254; };
    forward only;
};
```

REGISTROS DE RECURSOS

Los archivos de resolución de nombres (`db.<zona>`) consisten en una colección de elementos llamados **RR** o **registros de recurso** (*resource records*). Los registros de recurso **son las unidades de consulta**, es decir, son los registros que un servidor DNS devuelve cuando se le realiza una consulta.

Los registros de recursos tienen 5 campos, algunos de ellos opcionales. Los campos que no se especifican toman un valor por defecto.

ESQUEMA RR

Nombre	TTL	Clase	Tipo	Datos

- **Nombre**
  Nombre DNS del recurso. Puede ser un nombre fqdn (Fully Qualified Domain Name), o un nombre corto o no fqdn.

Ejemplo fqdn y no fqdn

```
ser.smr.    ; es un nombre fqdn ya que acaba en . ,es decir, en la raíz.
ns          ; es un nombre no fqdn
www.ser     ; es un nombre no fqdn
```

- **TTL (time to live)**
  Tiempo en segundos que un registro de recurso **permanece** en la **memoria caché** de un cliente. Puede especificarse en semanas (`w`) días (`d`), horas (`h`), minutos (`m`) o segundos (`s`), si no se especifica unidad de tiempo se suponen segundos. Si el valor es 0, el registro no se guarda en la memoria caché.

TTL

```
1h30m       ; TTL de 1 hora y 30 minutos
86400       ; TTL de 1 dia  por defecto en segundos
3600s       ; TTL de 1 hora expresado en segundos
```

- **Clase**
  Espacio de nombres a los que pertenece el registro. Aunque existen varios espacios hoy solo se usa el de Internet, el resto están en desuso. Su valor es el siguiente:
  - **IN** Internet
  - CH Chaos (diagnóstico)
  - HS Hesiod (MIT)
- **Tipo**
  Identifica el tipo de registro de recurso. Algunos valores habituales son: SOA, NS, A, AAAA, PTR, MX, CNAME, etc.
- **Datos**
  Datos del registro. Dependen del tipo de registro.

DIRECTIVAS

Para mejorar la legibilidad de los archivos de resolución de zonas se suelen definir las siguientes **directivas**:

- **$ORIGIN**
  Define el **nombre fqdn de la zona** que se configura en el archivo. La directiva $ORIGIN es **usada** por BIND **para completar los nombres de registro no fqdn**. El caracter **@** se utiliza para expandir el valor de esta

directiva en el campo nombre de los RR. Además si en un RR no se especifica nombre (espacio en blanco) se sustituye por el nombre del registro anterior. Si en un fichero `db` no se define la directiva entonces su valor por defecto es el nombre de la zona, sin embargo, por legibilidad se suele recomendar definirla.

> **Ejemplo $ORIGIN**
>
> En el archivo `db.ser.smr` que define la zona `ser.smr` se define la siguiente directiva $ORIGIN
>
> ```
> $ORIGIN ser.smr.   ; acaba en . -> fqdn
> ...
> ;nombre ttl  clase  tipo  dato
> @       1h   IN     NS    ns.ser.smr.    ; @=$ORIGIN=ser.smr.
> ns      1h   IN     A     192.168.1.254  ; ns.ser.smr.
>         1h   IN     A     192.168.1.253  ; espacio=ns.ser.smr.
> ...
> ```

- **$TTL (time to live)**
  Define el **tiempo** en segundos que **por defecto** que un registro DNS permanece en la **caché**. Este valor se aplica a aquellos registros de recursos que no definan su propio TTL. Si no se define esta directiva su valor por defecto es 1 hora. Para definir el tiempo se puede usar el valor en segundos o bien los símbolos w (week), h (hour), m (minute), s (second).

> **Ejemplo $TTL**
>
> En el archivo `db.ser.smr` que define la zona `ser.smr` se define la siguiente directiva $TTL
>
> ```
> $TTL 1h30m
> ...
> ;nombre ttl  clase  tipo  dato
> @       2h   IN     NS    ns.ser.smr.    ; TTL 2h
> ns           IN     A     192.168.1.254 ; TTL=$TTL=1h30m
> ...
> ```

TIPOS DE REGISTROS

SOA

Define las características globales de la zona como por ejemplo el servidor que la define o la sincronización con zonas esclavas. En cada archivo `db` solamente puede existir un registro SOA y es el primer registro que se debe definir en la zona. Se define con el siguiente esquema

```
<nombre> <ttl> IN SOA <server> <email> (
        <serial>
        <refresh>
        <retry>
        <expire>
        <nxdomain>
)
```

campo	descripción
`<nombre>`	Nombre fqdn de la zona. Si se ha definido directiva $ORIGIN su valor suele ser @.
`<ttl>`	TTL del registro. Si se omite su valor será $TTL.
`IN`	Clase del registro Internet
`SOA`	Tipo de Registro Start Of Authority
`<server>`	Nombre fqdn del servidor que define la zona
`<email>`	Dirección de correo del administrador de la zona. Como las direcciones de correo tienen la forma usuario@servidor y la @ es un simbolo reservado se utiliza el caracter "." en lugar de arroba. Por ejemplo, si el email es admin@google.com se escribe admin.google.com
`<serial>`	Se utiliza en la sincronización con las zonas esclavas. Cuando una zona esclava solicita transferencia de zona comprueba el serial, si este es mayor que el de la zona esclava entonces se produce la transferencia de zona. Como número serial se suele utilizar la fecha en formato americano seguido de un valor secuencial de dos digitos. Por ejemplo, si se ha modificado la zona el 15 de Diciembre de 2025 por segunda vez en el día su serial sería 2025121502.
`<refresh>`	Se utiliza en la sincronización con las zonas esclavas. Tiempo de espera de las zonas esclavas para volver a sincronizar la zona.
`<retry>`	Se utiliza en la sincronización con las zonas esclavas. Tiempo de espera de las zonas esclavas después de un fallo en la sincronización de la zona.
`<expire>`	Se utiliza en la sincronización con las zonas esclavas. Tiempo máximo que una zona esclava puede seguir usando la zona sin sincronizarse.
`<nxdomain>`	TTL que un registro nxdomain (dominio no existe) de esta zona debe ser almacenado en la cache

Nota: Los tiempos se expresan en segundos aunque se pueden usar las unidades w,h,m,s de forma similar a los TTL.

SOA

Ejemplo registro SOA para el dominio ser.smr.

```
$ORIGIN ser.smr.
$TTL 2d

@ IN SOA ns.ser.smr. admin.ser.smr. (
    2025121502     ; serial
    1d             ; refresh
    3h             ; retry
    1w             ; expire
    1h             ; nxdomain
)
```

NS

Define los servidores autorizados para la zona. Es un registro obligatorio. Puede existir más de un registro NS, por ejemplo, cuando una zona esta definida en otros servidores que definen la zona como esclava. Se define con

el siguiente esquema:

```
<nombre> <ttl> IN NS <server>
```

campo	descripción
`<nombre>`	Nombre fqdn de la zona. Si se ha definido directiva $ORIGIN su valor suele ser @.
`<ttl>`	TTL del registro. Si se omite su valor será $TTL.
`IN`	Clase del registro Internet
`NS`	Tipo de Registro Name Server
`<server>`	Nombre fqdn del servidor que define la zona

NS

Ejemplo registro NS para el dominio ser.smr.

```
$ORIGIN ser.smr.
$TTL 2d
....
@ IN NS ns.ser.smr.   ;server master
  IN NS nss.ser.smr.  ;server slave
....
```

A

Define la IPv4 asociada a un nombre. Se utiliza en la resolución directa de un nombre. Se define con el siguiente esquema:

```
<nombre> <ttl> IN A <IPv4>
```

campo	descripción
`<nombre>`	Nombre
`<ttl>`	TTL del registro. Si se omite su valor será $TTL.
`IN`	Clase del registro Internet
`A`	Tipo de Registro Address IPv4
`<IPv4>`	Dirección IPv4 asociada al nombre

A

Ejemplo registros A para el dominio ser.smr.

```
$ORIGIN ser.smr.
$TTL 2d
```

```
....
ns  IN A 192.168.1.254
nss IN A 172.10.10.254
www IN A 192.168.1.250
....
```

AAAA

Define la IPv6 asociada a un nombre. Se utiliza en la resolución directa de un nombre. Se define con el siguiente esquema:

```
<nombre> <ttl> IN AAAA <IPv6>
```

campo	descripción
`<nombre>`	Nombre
`<ttl>`	TTL del registro. Si se omite su valor será $TTL.
`IN`	Clase del registro Internet
`AAAA`	Tipo de Registro Address IPv6
`<IPv6>`	Dirección IPv6 asociada al nombre

AAAA

Ejemplo registros A para el dominio ser.smr.

```
$ORIGIN ser.smr.
$TTL 2d
....
ns  IN AAAA 2001:db8:a0b:12f0::10
nss IN AAAA 2001:aaa:bbb:ccc0::25
www IN AAAA 2001:db8:a0b:12f0::15
....
```

PTR

Define el nombre asociado a una IP. Se utiliza en la resolución inversa. Se define con el siguiente esquema:

```
<IP> <ttl> IN PTR <nombre>
```

campo	descripción
`<IP>`	Dirección IP
`<ttl>`	TTL del registro. Si se omite su valor será $TTL.
`IN`	Clase del registro Internet
`PTR`	Tipo de Registro Address IPv6

campo	descripción
`<nombre>`	Nombre asociado a la IP. Debe ser siempre fqdn

PTR

Ejemplo registros PTR para el dominio ser.smr.

```
$ORIGIN 1.168.192.in-addr.arpa.
$TTL 2d
....
254 IN PTR ns.ser.smr.
250 IN PTR www.ser.smr.
....
```

CNAME

Define el una alias para un nombre canónico, es decir, aquel que tiene asociado un registro A o AAA. Se utiliza en la resolución directa. Se define con el siguiente esquema:

```
<alias> <ttl> IN CNAME <nombre>
```

campo	descripción
`<alias>`	Nombre que se utilizará como alias
`<ttl>`	TTL del registro. Si se omite su valor será $TTL.
`IN`	Clase del registro Internet
`CNAME`	Tipo de Registro Canonical Name
`<nombre>`	Nombre asociado a la IP. Debe ser siempre fqdn

CNAME

Ejemplo registros CNAME para el dominio ser.smr.

```
$ORIGIN ser.smr.
$TTL 2d
....
ns  IN A 192.168.1.254
nss IN A 172.10.10.254
www IN A 192.168.1.250
; alias
web IN CNAME www.ser.smr.
....
```

MX

Define el nombre del servidor de correo que se encarga de gestionar los correos de la zona. Se define con el siguiente esquema:

```
<nombre> <ttl> IN MX <prioridad> <servidor>
```

campo	descripción
`<nombre>`	Nombre de la zona. Si se ha definido $ORIGIN normalmente será @.
`<ttl>`	TTL del registro. Si se omite su valor será $TTL.
`IN`	Clase del registro Internet
`MX`	Tipo de Registro Mail Exchanger
`<prioridad>`	Número que determina la prioridad de los servidores disponibles. Los servidores con menor prioridad se utilizan primero.
`<servidor>`	Nombre del servidor que administra el correo. Debe ser siempre fqdn

MX

Ejemplo registros MX para el dominio ser.smr.

```
$ORIGIN ser.smr.
$TTL 2d
....
@   IN MX 10 mail.ser.smr.
    IN MX 20 aspmx.l.google.com.

;registro A del mail de la zona
mail  IN A 192.168.1.201
; observar que aspmx.l.google.com. es externo
; a la zona y por tanto no hay que definir
; registro A
....
```

SRV

El registro SRV (Service Record) se utiliza para especificar los servicios disponibles en un dominio, y cómo los clientes deben encontrarlos. El propósito principal de este registro es ayudar a localizar servicios específicos (como servidores de ftp, www, ldap, etc.) dentro de un dominio.

```
<_servicio>.<_protocolo>.<nombre>. IN SRV <prioridad> <peso> <puerto> <servidor>.
```

campo	descripción
`<_servicio>`	Nombre del servicio que se quiere definir. Por ejemplo, ftp, ldap, etc.
`<_protocolo>`	Protocolo de transporte para el servicio. Por ejemplo, tcp o udp
`<nombre>`	Nombre del dominio
`IN`	Clase del registro Internet
`SRV`	Tipo de registro Servicio

campo	descripción
`<prioridad>`	Número que determina la prioridad de los servidores disponibles. Los servidores con menor prioridad se utilizan primero.
`<peso>`	Número que determina el peso de los servidores de la misma prioridad. Los servidores de igual prioridad con mayor peso se utilizan primero.
`<puerto>`	Número de puerto para acceder al servicio.
`<servidor>`	Nombre del servidor que administra el servicio. Debe ser siempre fqdn

SRV ldap

Ejemplo registros SRV para el dominio ser.smr. y administración del servicio LDAP

```
$ORIGIN ser.smr.
$TTL 2d
....
_ldap._tcp.ser.smr.  IN SRV 0 10 389 ldap.ser.smr.
_ldap._tcp.ser.smr.  IN SRV 0 5  389 ldap.secure.smr.

;registro A del ldap de la zona
ldap  IN A 192.168.1.230
; observar que ldap.secure.smr. es externo
; a la zona y por tanto no hay que definir
; registro A
....
```

OPCIONES

Las opciones permiten modificar le comportamiento del servidor. Algunas opciones sólo se pueden definir a nivel global (`options`) otras se pueden definir también a nivel de zona (`zone`).

ACL

Las ACL (Access Control List) permiten definir mediante un nombre descriptivo un **grupo de direcciones IP o redes** para posteriormente utilizarlas dentro de alguna opción de configuración. Las ACL se evaluan desde la primera condición hasta que se cumpla alguna de sus condiciones. Exiten varias **ACL** ya creadas **por defecto**:

ACL	descripción
`none`	Ninguna dirección IP
`any`	Cualquier dirección IP
`localhost`	Cualquier dirección IP configurada en el host
`localnets`	Cualquier dirección IP de la red o redes del host

Se pueden definir a nivel global y a nivel de zona siempre fuera de la clausula option y/o zone. Es habitual definirlas en su propio fichero de configuración, por ejemplo, `/etc/bind/named.conf.acl` para luego realizar un include en algún archivo de configuración o de zona. Su esquema es el siguiente:

```
    acl "<nombre>" {
        <IP>;            // IP  aceptada
        <IP>/<mascara>;  // Red aceptada
        "acl";           // ACL aceptada
        !<IP>;           // IP  rechazada
        !<IP>/<mascara>; // Red rechazada
        !"acl";          // ACL rechazada
};
```

ACL

Un servidor DNS quiere crear un ACL para permitir peticiones de cualquier IP menos de la red 172.20.20.0/24

```
// archivo /etc/bind/named.conf.options
acl "query-net" {
    !172.20.20.0/24;
    any;
};
options {
    allow-query { "query-net"; };
};
```

DIRECTORY

Permite definir el directorio en el que BIND buscará los archivos `db.zona` cuando no se utilicen path absolutos en la clausula `file` de la definición de la zona. Se define a nivel global. Su esquema es el siguiente:

```
    directory "directorio";
```

directory

El servidor DNS grabará el archivo db.ser.smr en el directorio /var/cache/bind

```
// archivo /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";
};
// archivo /etc/bind/named.conf.local
zone "ser.smr." {
    type master;
    file "db.ser.smr";
};
```

FORWARDERS

Permite definir una lista de servidores DNS a los que reenviar las peticiones de resolución de nombres cuando no se pueden resolver en las zonas definidas en el propio servidor. En este caso, el servidor se comportará como un servidor DNS recursivo caché y responderá de manera no autoritativa para dichas peticiones. Se define a nivel global. Su esquema es el siguiente:

```
    fowarders { <IP-forward>; <IP-forward>; };
```

DNS forwarder

Un servidor DNS con IP 192.168.1.254 define la zona ser.smr. , el resto de peticiones las reenvía al servidor 1.1.1.1.

```
// archivo /etc/bind/named.conf.options
options {
    fowarders { 1.1.1.1; };
};
```

LISTEN-ON

Permite definir las interfaces IPv4, IPv6 y puertos por los que se servirán las consultas DNS. Se puede usar `any` o `none` para indicar todas o ningua interfaz. Se definen a nivel global. Su esquema es el siguiente:

```
listen-on  [<port>]   { <IPv4>; <IPv4>; };
listen-on-v6 [<port>] { <IPv6>; <IPv6>; };
```

listen-on

Un servidor DNS con dos interfaces de red de direcciones 192.168.1.254 y 172.18.1.1.1 quiere escuchar peticiones DNS solo para la interfaz localhost y la interfaz de IP 192.168.1.254

```
// archivo /etc/bind/named.conf.options
options {
    listen-on {127.0.0.1; 192.168.1.254; };
    listen-on-v6 {none; };
};
```

RECURSION

Configura al servidor para realizar resoluciones de nombres recursivas. Por defecto, es recursivo. Se define a nivel global. Su esquema es el siguiente:

```
recursion <yes|no>;
```

ALLOW-RECURSION

Especifica que equipos podrán realizar consultas recursivas al servidor. Se define a nivel global. Por defecto, por motivos de seguridad, su valor es `{ localnets; localhost; }` por lo que solamente se permite recursión al propio equipo y a su red a no ser que se defina explícitamente. Su esquema es el siguiente:

```
allow-recursion { <IP>; <IP>/<mascara>; acl; };
```

allow-recursion

Un servidor DNS quiere ser recursivo solo para las redes 172.16.16.0/24 y 192.168.1.0/24 además de para su propia red y él mismo.

```
// archivo /etc/bind/named.conf.options
acl "recursive-net" {
    172.16.16.0/24;
    192.168.1.0/24;
    localhost;
    localnets;
};
options {
    recursion yes;
    allow-recursion { "recursive-net"; };
};
```

ALLOW-QUERY-CACHE

Especifica que equipos podrán realizar consultas a la cache. Se define a nivel global. Por defecto, por motivos de seguridad, su valor es `{ localnets; localhost; }` Si se define allow-recursion entonces hereda el valor definido en dicha opción, por ese motivo, suele ser habitual definir solo allow-recursion y no esta clausula. De esta forma se evitan los ataques de envenamiento de cache o analicen el historial de tráfico DNS del servidor. Su esquema es el siguiente:

```
allow-query-cache { <IP>; <IP>/<mascara>; acl; };
```

CACHE

Existen varias clausulas para controlar el comportamiento de la cache, entre ellas: `max-cache-size`, `max-cache-ttl` y `max-ncache-ttl`.

La clausula **`max-cache-size`** especifica el tamaño máximo de la memoria cache en RAM.

```
max-cache-size ( default | unlimited | <sizeval> | <percentage> );
```

Valor	Descripción
`default`	Depende del valor de `recursion`. Si `recursion yes` es el 90% del tamaño de la memoria del equipo. Si `recursion no` entonces es 2 MB.
`unlimited`	Hasta acabar con los recursos de memoria del servidor
`<sizeval>`	Valor en `K`iloBytes, `M`egaBytes, `G`igaBytes.
`<percentage>`	Valor en % sobre el total de la memoria

La clausula **`max-cache-ttl`** especifica el tiempo máximo que un registro permanece en la cache. Se pueden especificar con w,d,h,m,s como $TTL. El valor por defecto es 1w y no se puede poner a 0.

```
max-cache-ttl <duration>;
```

La clausula **`max-ncache-ttl`** especifica el tiempo máximo que un registro no resuelto (NXDOMAIN) permanece en la cache. Se pueden especificar con w,d,h,m,s como $TTL. El valor por defecto es 3h y su valor máximo es 7d;

```
max-ncache-ttl <duration>;
```

Configuración cache

Un servidor DNS recursivo quiere restringir el tamaño de su memoria cache a 512 MB, que el tiempo máximo de los registros resueltos sea de 3 días y de los no resueltos de 1 hora.

```
// archivo /etc/bind/named.conf.options
options {
    max-cache-size 512M;
    max-cache-ttl 3d;
    max-ncache-ttl 1h;
};
```

ALLOW-QUERY

Especifica que equipos podrán realizar consultas al servidor. Se define a nivel global o de zona. Por defecto, su valor es `{ any; }` por lo se permite la consulta a cualquier equipo. Su esquema es el siguiente:

```
allow-query { <IP>; <IP>/<mascara>; acl; };
```

allow-recursion

Un servidor DNS quiere permitir consultas solo a las redes 172.16.16.0/24 y 192.168.1.0/24 además de su propia red y él mismo.

```
// archivo /etc/bind/named.conf.options
acl "query-net" {
    172.16.16.0/24;
    192.168.1.0/24;
    localhost;
    localnets;
};
options {
    allow-query { "query-net"; };
};
```

DELEGACIÓN DE DOMINIOS

La delegación de dominios es el mecanismo de DNS que permite que la autoridad sobre un dominio o subdominio pase a otros servidores DNS sin perder la jerarquía sobre dicho dominio. La delagación de dominios es la piedra de toque en la que se basa todo el sistema de distribución de la base de datos de DNS.

La delegación de domino se realiza fundamentalmente mediante la definición de los registros NS que permiten alcanzar dichos dominios. Veamoslo con un ejemplo.

Delegación de dominios

Un servidor DNS de nombre ns.ser.smr e IP 192.168.1.254 define el dominio ser.smr y delega los subdominio ga y gb en los servidores ns.ga.ser.smr y ns.gb.ser.smr que tienen IP 192.168.2.254 y 192.168.3.254 respectivamente.

```
$ORIGIN ser.smr.
$TTL  1d
@ IN SOA ns.ser.smr. admin.ser.smr. (
    2025121502     ; serial
    1d             ; refresh
    3h             ; retry
    1w             ; expire
    1h             ; nxdomain
)

@      IN NS ns.ser.smr.
ga     IN NS ns.ga.ser.smr. ;delegacion
gb     IN NS ns.gb.ser.smr. ;delegacion

ns     IN A 192.168.1.254
ns.ga  IN A 192.168.2.254   ;delegacion
ns.gb  IN A 192.168.3.254   ;delegacion

www    IN A 192.168.1.250
ftp    IN A 192.168.1.249
```

TRANSFERENCIAS DE ZONAS

Las transferencias de zonas es el mecanismo que utilizan las zonas master y slave para transferirse los registros de recursos. Básicamente el mecanismo de transferencias de zona es el siguiente:

Cuando un administrador modifica una zona master aumenta el `serial` definido en el registro SOA de la zona. El servidor que contiene la zona slave intenta sincronizarse periódicamente, según el tiempo definido en `refresh`, con la zona master. Si el `serial` de la zona master es superior al de la zona slave se producira la transferencia de zona y por tanto la sincronización de la zona.

Existen dos tipos de transferencia de zonas:

- **AXFR** (Asynchronous Full Transfer)
  Se produce cuando la zona slave realiza una petición de sincronización con la zona master. La zona master envía todos los registros que definen la zona. Este tipo de transferencia se produce siempre la primera vez que se sincronizan las zonas master y slave y tambien cuando no se soporta el otro tipo de transferencia. El problema es que consume mucho ancho de banda y es algo lento ya que envía todos los registros de la zona.
- **IXFR** (Incremental Zone Transfer)
  Se produce cuando la zona slave realiza una petición de sincronización con la zona master. La zona slave manda un número de serie indicando la ultima sincronización realizada, entonces la zona master envía sólo los registros que definen la zona y que se modificaron posteriormente a dicho número de serie. Las transferencias consumen menos ancho de banda y son más rápidas.

ALLOW-TRANSFER

Es evidente que la seguridad de la base de datos de una zona se puede ver comprometida si algún "hacker" define un servidor DNS con una zona esclava apuntando a la dirección del servidor DNS master. Para evitar que cualquier servidor pueda apuntar a nuestras zonas se utiliza la clausula `allow-transfer`.

Pasos:

- 1.Servidor master. En la configuración global "bloqueamos" todas las transferencias.

```
//archivo named.conf.options
options {
...
  allow-transfer { none; };
...
};
```

- 2.Servidor master. En la configuración de la zona master "permitimos" las transferencias para los servidores secundarios que consideremos seguros.

```
//archivo named.conf.local
...
//zona master
zone "<dominio.>" {
...
  allow-transfer { IP-slave; IP-slave; };
...
};
```

allow-transfer

Un servidor DNS define la zona master ser.smr y quiere permitir solo las transferencias de zona para los servidores esclavos con dirección 192.168.1.20 y 172.16.10.10:

```
//archivo named.conf.options
options {
    ...
    allow-transfer { none; };
    ...
};

//archivo named.conf.local
...
zone "ser.smr." {
    type master;
    file "/etc/bind/db.ser.smr";
    allow-transfer { 192.168.1.20; 172.16.10.10; };
};
...
```

CLAVES TSIG

Aparentemente la protección mediante IP podría ser suficiente. Sin embargo, dista mucho de la realidad, cualquier "hacker" podría averigurar IPs en la red simplemente analizando el tráfico de las peticiones DNS el cual no es cifrado.

Un paso más en la seguridad de las transferencias entre zonas master y slaves es el uso de una clave simétrica para permitir las trasnferencias. Las claves **TSIG** (Transaction Signature) son claves simétricas compartidas por los servidores DNS, normalmente uno master y otro slave. Se usan para firmar criptográficamente los mensajes DNS y permite verificar:

- Autenticidad. El mensaje proviene de un servidor autorizado.
- Integridad. El mensaje no ha sido modificado.
- Fiablidad. La comunicación es fiable.

Pasos:

- 1.Servidor master. Crear clave.

```
tsig-keygen -a hmac-sha256 clave > /etc/bind/tsig-clave.key
chmod 640 /etc/bind/tsig-clave.key
chown root:bin /etc/bind/tsig-clave.key
```

  tsig-keygen habrá creado con el algoritmo hmac-sha256 una key de nombre clave que define una contraseña o secret. El archivo .key tiene la siguiente forma.

```
//archivo tsig-clave.key
key "clave" {
  algorithm hmac-sha256;
  secret "mXJ7p7b9zG+YxLQ8Q0dKxZQpXxK9WZP3R9uGZJ0sTn4=";
};
```

- 2.Servidor master. Incluir en el archivo de zonas la clave y asignarla a la zona que se desea proteger.

```
//archivo named.conf.local
...
include "/etc/bind/tsig-clave.key"
...
zone "<dominio.> {
  type master;
  file /etc/bind/db.dominio";
  allow-transfer { key clave; };
};
```

- 3.Servidores slave. Copiar el archivo .key a los servidores DNS slave que se desean puedan hacer transferencias de zonas.

```
cp tsig-clave.key /etc/bind/tsig-clave.key
```

- 4.Servidores slave. Incluir en el archivo de zonas la clave y asignarla a las zonas en las que se quiere realizar transferencias.

```
//archivo named.conf.local
...
include "/etc/bind/tsig-clave.key"
...
zone "<dominio.> {
  type slave;
  file /var/lib/bind/db.dominio";
  masters { <IP-master> key clave; };
};
```

TSIG

Un servidor DNS con IP 192.168.1.254 define la zona master ser.smr y quiere permitir las transferencias de zona solo para los servidores esclavos que tengan la clave TSIG.

```
#1.Servidor master. Generar TSIG
tsig-keygen -a hmac-sha256 clave > /etc/bind/tsig-clave.key
chmod 640 /etc/bind/tsig-clave.key
chown root:bin /etc/bind/tsig-clave.key

#2.Servidor master. Incluir TSIG en la zona
//archivo named.conf.local
...
include "/etc/bind/tsig-clave.key"
```

```
...
zone "ser.smr." {
    type master;
    file /etc/bind/db.ser.smr";
    allow-transfer { key clave; };
};

#3.Servidores slave. Copiar tsig-clave.key
cp tsig-clave.key /etc/bind/tsig-clave.key

#4.Servidores slave. Incluir TSIG enla zona
//archivo named.conf.local
...
include "/etc/bind/tsig-clave.key"
...
zone "ser.smr." {
    type slave;
    file /var/lib/bind/db.ser.smr";
    masters { 192.168.1.254 key clave; };
};
```

NOTIFY

Los servidores DNS BIND permiten no solo que el servidor slave inicie la sincronización con el servidor master, sino que sea el propio servidor master quien envíe notificaciones al servidor slave cuando haya realizado modificaciones en la zona para de este modo iniciar el proceso de sincronización. Se realiza mediante las clausulas `notify` , `also-notify` y `allow-notify`

En la **zona master** usamos.

`notify <yes | no | explicit>`

valor	descripción
`yes`	Se envia notificación a todos los servidores esclavos definidos en los registros NS de la zona. Es el valor por defecto.
`no`	No se envia ninguna notificación
`explicit`	Se envia notificación sólo a los servidores esclavos definidos en la clausula `also-nofify`

`also-notify { <IP-slave>; <IP-slave>; | <IP-slave> key <clave>;}`

Lista de servidores slaves a los que se les pasa notificación, también se puede incluir una clave TSIG.

En la **zona slave**, usamos.

`allow-notify { <IP-master>; <IP-master>; | key <clave>; }`

Por defecto, si no se indica allow-notify es igual a la lista de masters definidos en la zona slave.

Por tanto, en BIND, por defecto se realizan notificaciones de las zonas master (`notify yes`) a las zonas slave definidas en los registros NS de la zona. Estas zonas slave aceptan notificaciones de los servidores master definidos (`allow-notify = { masters; }`).

BALANCEO DE CARGA

En BIND podemos, por ejemplo, definir varios registros que apuntan a la misma dirección (`A`) o varios registros que definen un mismo dominio (`NS`). Por ejemplo:

```
$ORIGIN ser.smr.
$TTL 2d

@ IN SOA ns.ser.smr. admin.ser.smr. (
        2025121502     ; serial
        1d             ; refresh
        3h             ; retry
        1w             ; expire
        1h             ; nxdomain
    )

@   IN NS   ns.ser.smr.
    IN NS   ns2.ser.smr.
    IN NS   ns3.ser.smr.

ns  IN A    192.168.1.254
ns2 IN A    192.168.1.253
ns3 IN A    192.168.1.252
www IN A    192.168.1.100
    IN A    192.168.1.101
    IN A    192.168.1.102
```

Por ejemplo, una consulta del registro www provocaría la contestación de un conjunto de registros de recursos (**`RRSet`**), en concreto 3 registros. Es muy importante el orden en que estos registros aparecen en la sección de respuesta del mensaje DNS ya que el cliente que ha realizado la consulta intentará la conexión en el orden propuesto. Variando el orden del la respuesta BIND puede **"balancear la carga"** sobre el servicio www, es decir, puede hacer que el orden varíe mejorando de esa manera la carga de trabajo de cada servidor wwww.

El balanceo de carga se puede configurar en la clausula **`rrset-order`** de las opciones globales de BIND.

```
rrset-order {
    [class <class_name>] [type <type_name>] [name "<domain_name>"] order <ordering>;
    [class <class_name>] [type <type_name>] [name "<domain_name>"] order <ordering>;
    [class <class_name>] [type <type_name>] [name "<domain_name>"] order <ordering>;
    }
```

Valor	Descripción
`class`	Clase del registro. Si no se define es ANY
`type`	Tipo del registro. Si no se define es ANY
`nombre`	Nombre del registro. Si no se define es *
`order`	`fixed` según estan escritos en la zona `random` aleatorio `cyclic` round-robin `none` indeterminado según la base de datos

Si no se configura `rrset-order` , el valor por defecto es `random` para todos los RRSet que se devuelvan. Si se configura `rrset-order` se aplican las reglas secuencialmente la primera que coincida se aplica.

rrset-order

En un servidor DNS se quiere que los registros de direcciones correspondientes al nombre www.ser.smr. se devuelvan en orden round-robin y el resto de registros en orden random.

```
//archivo /etc/bind/named.conf.options
options {
    rrset-order {
        type A name www.ser.smr. order cyclic;
        order random;
    };
};
```

Para comprobar el orden en que nos devuelve la respuestas podemos usar la utilidad `dig`.

dig y order

Para el servidor del ejemplo anterior se podrían realizar las siguientes consultas sobre www para comprobar la respuesta round-robin.

```
dig @127.0.0.1 www.ser.smr. +short
  www 192.168.1.100
  www 192.168.1.101
  www 192.168.1.102
dig @127.0.0.1 www.ser.smr. +short
  www 192.168.1.101
  www 192.168.1.102
  www 192.168.1.100
dig @127.0.0.1 www.ser.smr. +short
  www 192.168.1.102
  www 192.168.1.100
  www 192.168.1.101
```

DNSSEC

BIND permite el uso del protocolo DNSSEC para autenticar y mantener la integridad de los mensajes DNS.

La opción global **`dnssec-validation`** activa o desactiva el uso de DNSSEC y la verificación de firmas digitales en las consultas DNS. El objetivo es garantizar que los datos recibidos de la consulta no hayan sido manipulados por ningun atacante.

```
//archivo named.conf.options
options {
    ...
    dnssec-validation <yes | no | auto>;
    ...
};
```

valor	descripción
`auto`	El servidor utiliza DNSSEC con las **trust anchors** (claves de confianza) integradas en BIND para validar la zona raíz. La primera vez que conecta con los servidores raíz lo hace con la clave

valor	descripción
	almacenada en `/etc/bind/bind.keys`, posteriormente la clave es actualizada mediante el módulo managed-keys en el archivo `/var/cache/bind/managed-keys.bind`
`yes`	El servidor utiliza DNSSEC pero requiere que el administrador configure manualmente las **trust anchors** (claves de confianza) para validar la zona raíz
`no`	El servidor desactiva DNSSEC y realiza consultas no seguras

Nota: En una red configurada sólo con IPv4 es necesario arrancar el servicio en modo IPv4 para que funcione dnssec.

Nota: Actualizar las claves de confianza integradas.
systemctl start named ; servidor en funcionamiento
rndc managed-keys refresh ; actualiza claves
rndc managed-keys status ; comprueba claves
rndc flush ; limpia cache

Nota: Forzar la carga de las claves de confianza.
systemctl stop named
rm -f /var/cache/bind/managed-keys.bind*
systemctl start named

Para comprobar el funcionamiento de DNSSEC se pueden realizar consultas mediante el comando `dig` o el comando `delv` diseñado por BIND para comprobar exclusivamente el funcionamiento de DNSSEC.

Consulta DNSSEC

```
dig @127.0.0.1 google.com +dnssec
delv @127.0.0.1 google.com
```

DDNS

DDNS (Dynamic DNS) permite realizar actualizaciones de los registros de recursos del servidor. BIND permite controlar DDNS mediante la clausula `allow-update` que se puede definir tanto a nivel global (`options`) como a nivel de zona (`zone`). Por defecto, su valor es `{ none; }` por lo no se permite la actualizacion a ningún equipo.

```
allow-update { <IP>; <IP>/<mascara>; acl; | key "clave"; };
```

Como se observa en la definición se puede usar una lista de IPs o bien una clave TSIG, método mucho más seguro.

allow-update

Un servidor DNS quiere permitir actualizaciones de registros solo desde el equipo 192.168.1.250 y locahost.

```
// archivo /etc/bind/named.conf.options
acl "update-net" {
    192.168.1.250;
    localhost;
};
options {
    allow-update { "update-net"; };
};
```

allow-update

Un servidor DNS quiere permitir actualizaciones de registros solo a los equipos que tengan la clave TSIG para la zona ser.smr.

```
// archivo /etc/bind/named.conf.local
include "/etc/bind/ddns.key"
zone "ser.smr." {
    type master;
    // el directorio debe tener permisos de escritura
    file "/etc/bind/db.ser.smr";
    allow-update { key "ddns"; }
}
```

FICHEROS DE LOG

El servidor DNS BIND envía mensajes de log para indicar multiples situaciones, desde simples informaciones a notificaciones de error.
Hay 3 maneras de configurar y acceder a estos mensajes.

RSYSLOGD

El daemon `rsyslogd` gestiona los mensajes de log mediante el protocolo SYSLOG y por defecto sus mensajes acaban en el archivo `/etc/log/syslog` donde se mezclan mensajes de diferentes servicios y aplicaciones. Para filtar los mensajes del servicio DNS podremos ejecutar.

```
cat /var/log/syslog | grep named | more

fgrep named /var/log/syslog
```

JOURNALD

El daemon `journald` se gestionan los mensajes de log enviandolos a un archivo binario que se puede consultar mediante la utilidad `journalctl`. Para filtar los mensajes del servicio DNS podremos ejecutar.

```
journalctl -u named.service -xe
```

CLAUSULA LOGGING

BIND9 permite mediante la clausula `logging` personalizar la salida de sus mensajes de log. Esta clausula se puede especificar por ejemplo en el archivo `/etc/bind/named.conf.options`. El sistema de logging se basa en los conceptos de canal y categoria.

- Canal
  Define donde se enviarán los mensajes de log. Puede ser un fichero, rsyslogd, la salida estandard, null, etc...
- Categoria
  Se refiere a categorias de mensajes de log. Por ejemplo: default, network, security, database,...La que más interesa es default que representa a todas las categorias.

El formato general de `logging` es:

```
logging {
    category <categoria> { <canal>; };
    channel  <canal> {
        <proceso_salida> <salida> [<opciones>];
        [opciones print];
    }
}
```

Logging

Se hará log de todos los mensajes (default) sobre un fichero de nombre /var/lib/bind/bind.log del que se guardarán hasta 3 versiones con un tamaño máximo de 100K. Además del mensaje se imprimirá en el archivo la hora, la gravedad del mensaje y su categoria.

```
logging {
    category default { "canal-log"; };
    channel "canal-log" {
        file "/var/lib/bind/bind.log" versions 3 size 100k;
        print-time yes;
        print-severity yes;
        print-category yes;
    };
};
```