

[tiendalinux.com](https://www.tiendalinux.com)

Ficheros de configuración BIND

27-34 minutos

El demonio named del servidor de nombres BIND utiliza el fichero `/etc/named.conf` para la configuración. Todos los ficheros de zona se encuentran en `/var/named`.

Advertencia

No modifique manualmente el fichero `/etc/named.conf` o cualquier otro fichero del `/var/named` si está usando la herramienta **BIND Configuration Tool**. Todos los cambios que se realicen manualmente en este fichero o en cualquier otros fichero de éste, se borrarán cuando se use la próxima vez la herramienta **BIND Configuration Tool**.

El fichero `/etc/named.conf` no puede tener errores para poder arrancar el demonio named. Aunque existen algunas opciones erróneas en algunas declaraciones que no son tan importantes como para bloquear el servidor, todos los errores que se encuentren en las declaraciones impedirán que arranque el demonio named.

`/etc/named.conf`

El fichero `/etc/named.conf` es un conjunto de

declaraciones que usa las opciones que se encuentran en los corchetes { }. He aquí cómo está organizado un fichero `/etc/named.conf` [Figura 17-2](#).

```
<statement-1> ["<statement-1-name>"]
[<statement-1-class>] {
    <option-1>;
    <option-2>;
    <option-N>;
};
```

```
<statement-2> ["<statement-2-name>"]
[<statement-2-class>] {
    <option-1>;
    <option-2>;
    <option-N>;
};
```

```
<statement-N> ["<statement-N-name>"]
[<statement-N-class>] {
    <option-1>;
    <option-2>;
    <option-N>;
};
```

Figura 17-2. Organización común de `/etc/named.conf`

La opción "`<statement-name>`" es necesaria solamente con las declaraciones `acl`, `include`, `server`, `view` y `zone`. La opción `<statement-N-class>` aparece sólo con la declaración `zone`.

Se pueden escribir comentarios en el fichero `/etc/named` usando los caracteres de C `/* */` o después de `//` y `#`.

Se pueden usar las siguientes declaraciones en el fichero `/etc/named.conf`:

- `acl <acl-name>` — Configura una lista de control del acceso de las direcciones IP a las que se autorizarán o se denegarán ciertos servicios `named`. En general, las direcciones IP individuales o la nota de IP (como `10.0.1.0/24` sirve para identificar las direcciones IP correctas.

Ya se han definido algunas listas de control de acceso, por lo tanto no tiene que configurar una declaración `acl` para definir las:

- `any` — Corresponde a todas las direcciones IP.
- `localhost` — Corresponde a todas las direcciones IP utilizadas para un sistema local.
- `localnets` — Corresponde a todas las direcciones IP de una red a la que se conecta el sistema local con los dispositivos.
- `none` — No corresponde a ninguna dirección IP.

Cuando se utilizan con otras declaraciones `/etc/named.conf` y sus opciones, las declaraciones `acl` pueden ser muy útiles para usar el servidor de nombres BIND. Observe el ejemplo siguiente en [Figura 17-3](#):

```
acl black-hats {  
    10.0.2.0/24;
```

```
    192.168.0.0/24;
};

acl red-hats {
    10.0.1.0/24;
};

options {
    blackhole { black-hats; };
    allow-query { red-hats; };
    allow-recursion { red-hats; };
}
```

Figura 17-3. Ejemplo de la utilización de la declaración `acl`

`named.conf` contiene dos listas de control de acceso (`black-hats` y `red-hats`).

- `controls` — Configura diversas restricciones de seguridad necesarias para la utilización del comando `rndc` y así poder usar el demonio `named`.

Consulte la [la sección de nombre `/etc/named.conf`](#) para mayor información sobre la declaración `controls`.

- `include "<file-name>"` — Incluye el fichero especificado en el fichero de configuración que se está usando y que permite así situar los datos de configuración sensibles (como `keys`) en un fichero separado con los permiso que impiden a los usuarios sin privilegios leerlos.
- `key "<key-name>"` — Define una clave particular. Estas

claves sirven para autenticar diversas acciones como la actualización de la seguridad o la utilización del comando `rndc`. Se usan dos opciones con `key`:

- `algorithm <algorithm-name>` — El tipo de algoritmo utilizado, como `dsa` o `hmac-md5`.
- `secret "<key-value>"` — La clave encriptada.

Consulte [Figura 17-22](#) el ejemplo de la declaración `n key`.

- `logging` — Permite usar varios tipos de logs que se llaman *channels* o *canales*. Usando la opción `channel` en la declaración `logging`, se puede construir un tipo de log personalizado con el nombre del fichero (`file`), con el tamaño(`size`), la versión (`version`) y el nivel de importancia (`severity`). Una vez que se ha definido el canal personalizado, se usa la opción `category` para calificar el canal y comenzar la conexión mientras que se arranca `named`.

Por defecto, `named` envía mensajes de log a los estándares del demonio `syslog`, que les sitúa en `/var/log/messages` por defecto. Esto se debe a que varios canales estándares se encuentran en BIND junto con varios niveles de importancia como el que trata los mensajes de registros informativos (`default_syslog`) y otro que trata específicamente la corrección de errores (`default_debug`). La categoría predeterminada `default`, utiliza los canales de BIND para llevar a cabo la conexión normal sin ninguna configuración especial.

La personalización del proceso de conexión es un proceso

que requiere una explicación muy detallada y no es el objetivo de este capítulo. Para mayor información, consulte el *Manual de referencia del administrador de BIND 9*.

- `options` — asigna valores a muchas opciones entrelazadas, incluidos los comandos que se usan para situar un fichero de funcionamiento de `named`, el nombre de ficheros y otros.

Las siguientes opciones son las más utilizadas:

- `allow-query` — Especifica los hosts que se usarán para establecer las peticiones en el servidor de nombres. Por defecto, todos los hosts están autorizados a presentar peticiones. Se puede usar una lista de control de acceso o una colección de direcciones IP para no autorizar a un determinado número de servidores de nombres.
- `allow-recursion` — Parecida a la opción `allow-query`, salvo que se aplica a las peticiones recursivas. Por defecto, todos los hosts están autorizados a presentar peticiones en el servidor de nombres.
- `directory` — Reemplaza el fichero de funcionamiento de `named` en vez del fichero predeterminado `/var/named`.
- `forward` — Controla cómo se lleva a cabo el forwarding, si la opción `forwarders` contiene direcciones IP válidas que designen dónde enviar las peticiones.

Si se usa la opción `first`, los servidores de nombres especificados en la opción `forwarders` son los primeros en recibir las peticiones y si no pueden solucionarlas el demonio `named` intenta la resolución.

Si se usa la opción `only`, `named` no intentará la resolución si los servidores de nombres han fallado.

- `forwarders` — Especifica una lista de servidores de nombres a los que hay que mandar las peticiones para obtener la resolución.
- `listen-on` — Especifica el dispositivo de red que `named` va a utilizar para recibir las peticiones. Por defecto se usan todos los dispositivos.

Esta opción es útil si dispone de más de un dispositivo de red y desea limitar los sistemas que puedan efectuar las peticiones gateway y servidor de nombres y desea bloquear todas las peticiones excepto las que provienen de su red privada, la opción en su servidor de nombres. Por ejemplo, si tiene una máquina que sirve como gateway y como servidor de nombres la opción `listen-on` se parece a la [Figura 17-4](#).

```
options {  
    listen-on { 10.0.1.1; };  
};
```

Figura 17-4. Ejemplo de la opción `listen-on`

De esta manera, solamente se aceptan las peticiones que provienen del dispositivo de red de la red privada (10.0.1.1).

- `notify` — Determina si `named` envía notificaciones a los servidores esclavos cuando una zona se actualiza. Por defecto, se usa la opción `yes`, pero se puede usar también la opción `no`, para evitar que se manden notificaciones a

los servidores esclavo y así solamente mandar las notificaciones a los servidores de la lista `also-notify`.

- `pid-file` — Permite especificar la localización del fichero del proceso ID creado por `named` cuando arranca.
- `statistics-file` — Permite especificar la localización del fichero de estadística que se ha creado. Por defecto, las estadísticas de `named` se encuentran en `/var/named/named.stats`.

Existen numerosas opciones disponibles, muchas de ellas dependen unas de otras para poder funcionar correctamente. Consulte el *Manual de referencia del administrador BIND 9* para mayor información.

- `server` — Define opciones particulares que afectan a la manera en la que `named` reacciona ante los servidores de nombres distantes y particularmente conciernen a las notificaciones y las transferencias de zona.

La opción `transfer-format` determina si se ha enviado el record de los recursos con cada mensaje (`one-answer`) o la grabación de recursos múltiples con cada mensaje (`many-answers`). Aunque la opción `many-answers` es más eficaz, solamente los últimos servidores de nombres BIND la entienden.

- `trusted-keys` — Contiene las claves públicas que usa DNSSEC. Para mayor información sobre la seguridad de BIND, consulte la [la sección de nombre Seguridad](#).
- `view "<view-name>"` — Visualizaciones especiales que responden a un tipo de información particular dependiendo

del host que contacta el servidor de nombres. Esto permite a determinados hosts recibir una respuesta que se refiere a una zona particular mientras que otros hosts reciben información completamente diferente. Alternativamente, ciertos hosts pueden estar autorizados para acceder a determinadas zonas mientras que otros menos autorizados continúan a efectuar peticiones a otras zonas.

Se pueden usar múltiples visualizaciones, las cuales tienen un solo nombre. La opción `match-clients` especifica las direcciones IP que se aplican a una visualización determinada. Se pueden usar todas las declaraciones `option` en una visualización, aunque tienen prioridad las opciones globales ya configuradas para `named`. La mayor parte de las declaraciones `view` contienen múltiples declaraciones `zone` que se aplican a la lista `match-clients`. El orden de aparición en la lista de las declaraciones `view` es importante porque la primera declaración `view` corresponde a la dirección IP de un cliente particular.

Consulte la [la sección de nombre Visualizaciones múltiples](#) para mayor información sobre la declaración `view`.

- `zone "<zone-name>"` — Especifica zonas particulares para las que está autorizado el servidor de nombres. La declaración `zone` se usa sobre todo para especificar el fichero que contiene la configuración de la zona y transmite ciertas opciones de esa zona a `named` que tendrán prioridad sobre todas las otras declaraciones `option` del fichero `/etc/named.conf`.

El nombre de la zona es importante porque representa el valor por defecto asignado a la directiva `$ORIGIN` que se usa en el fichero de zona y que está relacionado con los el no-FQDN. Por ejemplo, si esta declaración `zone` define el espacio de nombre para `domain.com`, hay que usar `domain.com` y no `<zone-name>` para que se sitúe al final de los nombres de los hosts que se usan en ese fichero de zona.

Las opciones más usadas de la declaración `zone` son las siguientes:

- `allow-query` — Especifica los clientes que se autorizan para pedir información sobre una zona. Por defecto todas las peticiones de información son autorizadas.
- `allow-transfer` — Especifica los servidores esclavos que están autorizados para pedir una transferencia de información de la zona. Por defecto, todas las peticiones se autorizan.
- `allow-update` — Especifica los hosts que están autorizados para actualizar dinámicamente la información de la zona. Por defecto, no se autoriza la actualización de la información.



Advertencia

Tenga cuidado cuando autorice a los hosts para actualizar la información de la zona. No habilite esta opción si no tiene confianza en el host que vaya a usar. Es mejor que el administrador actualice manualmente los records de zona y que

recargue el servicio `named`, si es posible.

- `file` — Especifica el nombre del fichero que contiene los datos de configuración de la zona en el fichero de funcionamiento `named` (por defecto `/var/named`).
- `masters` — Se utiliza si la zona se define como `type` esclava. La opción `masters` indica al `named` de un esclavo la/las direcciones en las que se puede pedir información de la zona en la que se tiene autoridad.
- `notify` — Es parecida a la opción `notify` que se usa con la declaración `option`.
- `type` — Define el tipo de zona. Se pueden usar los siguientes tipos:
 - `forward` — Dice al servidor de nombres que lleve a cabo todas las peticiones de información de la zona en cuestión hacia otros servidores de nombres.
 - `hint` — Tipo especial de zona que se usa para orientar hacia los servidores de nombres `root` que sirven para resolver peticiones de una zona que no se conoce. Normalmente, no tendrá que configurar una zona que está situada fuera del `/etc/named.conf`.
- `master` — Designa el servidor de nombres actual que tiene la autoridad en esa zona. Una zona se puede configurar como tipo `master` si tiene ficheros de configuración de la zona en el sistema actual.
- `slave` — Designa el servidor de nombres actual que es servidor esclavo para dicha zona y le dice a `named` que

pida los ficheros de configuración de la zona de las direcciones IP al servidor de nombres master.

- `zone-statistics` — Dice a `named` que conserve las estadísticas que conciernen a esa zona escribiéndolas bien en la localización por defecto de `(/var/named/named.stats)`, o en la localización designada por la opción `statistics-file` en la declaración `server`, si existe.

Ejemplos de declaraciones de zona

La mayor parte de los cambios del fichero `/etc/named.conf` de un servidor de nombres maestro o esclavo se refieren a añadir, modificar o suprimir declaraciones de `zone`. Aunque estas declaraciones pueden contener muchas opciones, la mayor parte de los servidores de nombres usan pocas. Las declaraciones de `zone` siguientes son ejemplos básicos que se pueden usar en una relación de servidores de nombre maestro/esclavo.

Una declaración de `zone` en un servidor de nombres primario del dominio `domain.com` se parece a la [Figura 17-5](#).

```
zone "domain.com" IN {  
    type master;  
    file "domain.com.zone";  
    allow-update { none; };  
};
```

Figura 17-5. Ejemplo de una declaración de `zone`

maestra simple

Esta declaración de zone nombra la zona `domain.com`, define la opción `type` como `maître`, dice a `named` que lea el fichero `/var/named/domain.com.zone` para configurar la zona e impide que se actualice otro host.

La declaración de zone para `domain.com` se parece a [Figura 17-6](#).

```
zone "domain.com" {  
    type slave;  
    file "domain.com.zone";  
    masters { 192.168.0.1; };  
};
```

Figura 17-6. Ejemplo de declaración de zone esclava simple

Esta declaración de zone dice a `named` en el servidor esclavo que busque el servidor maestro `192.168.0.1` para encontrar la información de configuración para la zona `domain.com`. La información que el servidor esclavo recibe se registran en el fichero `/var/named/domain.com.zone`.

Ficheros de zona

Los *ficheros de zona* que contienen información sobre el espacio de nombre particular se almacenan en el fichero de funcionamiento de `named`, que por defecto es `/var/named`. Cada fichero de zona se nombra según los datos de las opciones de `file` en la declaración `zone`,

generalmente de tal manera que se refiere al dominio en cuestión e identifica el fichero que contiene los datos de zona como `example.com.zone`.

Cada fichero de zona puede contener directivas y registros de recursos. Las *directivas* dicen al servidor de nombres que efectue una determinada acción o que aplique una configuración especial a la zona. Los *registros de recursos* definen los parámetros de la zona asignando una identidad a sistemas particulares en el interior del espacio de nombre de la zona. Las directivas no son obligatorias pero los registros de recursos son necesarios para ofrecer un servicio de nombres a dicha zona. Todas las directivas y registros de recursos tienen que estar en la línea correspondiente.

Se pueden escribir comentarios en los ficheros de zona después de los puntos y comas (;).

Directivas de los ficheros de zona

Las directivas se identifican por el caracter \$ que se sitúa delante del nombre de la directiva y generalmente en la parte de arriba del fichero de zona.

A continuación le mostramos las directivas más usadas:

- **\$INCLUDE** — Dice a `named` que incluya otro fichero de zona en el fichero de zona donde se usa la directiva. Así se pueden almacenar configuraciones de zona suplementarias que dependen del fichero de zona principal.
- **\$ORIGIN** — Determina el nombre del registro no

cualificado, como por ejemplo los que especifican solamente el host.

Por ejemplo, un fichero de zona puede contener la línea siguiente:

Todos los nombres que se usan en los registros de recursos y que no acaban por un punto (.) se añaden al nombre de dominio. Es decir, cuando el servidor lee el registro de zona, la primera línea de arriba se interpretará como segunda línea:

```
ftp                IN      CNAME    server1
ftp.domain.com.   IN      CNAME
server1.domain.com.
```

Nota

La utilización de la directiva \$ORIGIN no es necesaria si nombramos la zona en /etc/named.conf con el valor de la opción \$ORIGIN. El nombre de la zona se usa por defecto como valor de la directiva \$ORIGIN.

- \$TTL — Ajusta el valor *Time to Live (TTL)* predeterminado para la zona. Es el nombre, en segundos, que se da a los servidores de nombres para determinar cuánto tiempo los registros de recursos de la zona serán válidos. Un registro de recursos puede contener su propio valor TTL, que tendrá prioridad sobre la directiva presente.

Cuando se decide aumentar este valor la esta directiva dice a los servidores de nombres que metan en caché esta

información de zona durante más tiempo. Esto reduce el número de peticiones de la zona pero aumenta el tiempo necesario para aumentar la cantidad de registros de recursos.

Registros de recursos de ficheros de zona

Los registros de recursos de ficheros de zona contienen columnas de datos, separadas por un espacio, que definen estos registros. Todos los registros de recursos de ficheros de zona están asociados a un tipo particular que designa el motivo del registro. A continuación le mostramos los tipos de registros más frecuentes:

- A — Registro de dirección que especifica una dirección IP que se debe asignar a un nombre.

Figura 17-7. Ejemplo de configuración del registro A

Si se omite el valor `<host>`, el registro A designa una dirección IP predeterminada para la parte superior del nombre. Este sistema es la base de todas las peticiones que no sean FQDN.

Consideremos los siguiente ejemplos del registro A:

	IN	A	10.0.1.3
server1	IN	A	10.0.1.5

Figura 17-8. Ejemplos de registros de A

Las peticiones para el dominio `domain.com` se dirigen al valor `10.0.1.3`, mientras que las del dominio `server1.domain.com` se dirigen al valor `10.0.1.5`.

- CNAME — Registro del nombre canónico que dice al servidor de nombres que todos los nombres son conocidos.

```
<alias-name>      IN      CNAME
<real-name>
```

Figura 17-9. Ejemplo de configuración del registro CNAME

En [Figura 17-9](#), todas las peticiones enviadas a *<alias-name>* se dirigen al host *<real-name>*. Los registros CNAME son los más utilizados para orientar servicios que usan un procedimiento común para dar nombres a los hosts.

Examinemos el ejemplo de [Figura 17-10](#), donde el registro A fija un nombre de un host particular a una dirección IP y un registro CNAME que orienta los nombres de los hosts más utilizados *www*.

```
server1      IN      A      10.0.1.5
www          IN      CNAME   server1
```

Figura 17-10. Ejemplo de registro CNAME

- MX — Registro Mail eXchange, que dice dónde se tiene que dirigir el correo enviado a un nombre de espacio particular controlado para esa zona.

```
      IN      MX      <preference-value>
<email-server-name>
```

Figura 17-11. Ejemplo de registro MX configuration

En [Figura 17-11](#), la opción *<preference-value>* le permite listar numéricamente los servidores de correo que seleccione para recibir mensajes para ese espacio de nombres, dando preferencia a ciertos sistemas de correo. El registro de recursos MX tiene el valor más bajo de la opción *<preference-value>*, pero puede ajustar varios servidores de correo con el mismo valor para distribuir el tráfico de correo entre ellos.

La opción *<email-server-name>* puede ser un nombre de host o u FQDN, siempre y cuando se oriente hacia el sistema adecuado.

```

            IN      MX      10
mail.domain.com.
            IN      MX      20
mail2.domain.com.
```

Figura 17-12. Ejemplo de registro MX

En este ejemplo, el primer servidor de correo `mail.domain.com` se prefiere al servidor `mail2.domain.com` para recibir los correos destinados al dominio `domain.com`.

- NS — egistro de servidor de nombres (NameServer) que indica a los servidores los servidores de nombres que tienen la autoridad de una determinada zona.

Figura 17-13. Ejemplo de configuración del registro NS

La opción *<nameserver-name>* tiene que ser un FQDN.

En la [Figura 17-14](#), aparecen dos servidores de nombres

que tienen la autoridad de un dominio. El hecho de que dos servidores sean esclavos o maestros no es importante. Los dos tienen la autoridad en esa zona.

```
IN      NS      dns1.domain.com.
IN      NS      dns2.domain.com.
```

Figura 17-14. Ejemplo de registro NS

- PTR — Registro PoinTeR creado para orientar hacia otra parte del espacio de nombres.

Los registros PTR sirven, sobre todo, para la resolución inversa de nombres ya que reorientan las direcciones IP hacia un nombre determinado. Consulte la [la sección de nombre *Ficheros de resolución de nombres inversa*](#) para ver más ejemplos de esta opción.

- SOA — Registro "Start Of Authority", que proclama información importante sobre la autoridad de determinados servidores sobre determinados espacios de nombres.

Está situado detrás de las directivas y es el primer registro en un fichero de zona.

```
@      IN      SOA      <primary-name-
server>      <hostmaster-email> (
                                <serial-number>
                                <time-to-refresh>
                                <time-to-retry>
                                <time-to-expire>
                                <minimum-TTL> )
```

Figura 17-15. Ejemplo de configuración del registro SOA

El símbolo @ sitúa la directiva \$ORIGIN (o el nombre de zona, si la directiva \$ORIGIN no está instalada) mientras que el espacio de nombre esta7aacute; definido por el registro presente de recursos SOA. El servidor de nombres primario que tiene la autoridad para este dominio la usa la opción *<primary-name-server>*, y la dirección de correo electrónico de la persona con la que hay que contactar se reemplaza con *<hostmaster-email>*.

La opción *<serial-number>* aumenta cada vez que cambia el fichero de zona con el fin de que named sepa que debe recargar esta zona. *<time-to-refresh>* dice a todos los servidores esclavos cuánto tiempo tienen que esperar antes de pedir información al servidor de nombres maestro si se han realizado cambios en la zona. El valor *<serial-number>* lo utiliza el servidor esclavo para determinar si está usando datos caducados o si tiene que actualizarlos.

La opción *<time-to-retry>* informa al servidor de nombres esclavo sobre el intervalo de tiempo que teiene que esperar antes de emitir una petición de actualización de datos en caso de que el servidor de nombres maestro no le responda. Si el servidor maestro no ha respondido a una petición de actualización de datos antes que se acabe el intervalo de tiempo *<time-to-expire>*, repone el servidor esclavo presentándose como servidor con la autoridad en este espacio de nombres.

La opción *<minimum-TTL>* pide a los otros servidores de nombres que situen en caché la información de esa zona

durante al menos ese periodo.

En BIND, el tiempo se mide en segundos. Sin embargo, puede usar las abreviaciones de las otras unidades de tiempo como minutos (M), horas (H), días (D), y semanas (W). La [Tabla 17-1](#) le muestra la cantidad de tiempo en segundos y el periodo equivalente en otro formato.

Tabla 17-1. Equivalencia en segundos de las otras unidades de tiempo

Segundos	Otras unidades de tiempo.
60	1M
1800	30M
3600	1H
10800	3H
21600	6H
43200	12H
86400	1D
259200	3D
604800	1W

El ejemplo siguiente le muestra a qué se parece el registro de recursos básico SOA.

```
@      IN      SOA      dns1.domain.com.  
hostmaster.domain.com. (  
                          2001062501 ; serial  
                          21600      ;
```

```

refresh after 6 hours
                                3600          ; retry
after 1 hour
                                604800        ;
expires after 1 week
                                86400 )      ;
minimum TTL of 1 day

```

Figura 17-16. Ejemplos de registros SOA

Ejemplos de ficheros de zona

Si observamos las directivas y los registros de recursos por separado, quizás tengamos algunas dificultades en comprenderlos. Sin embargo, si los estudiamos en su conjunto en un fichero no resulta tan difícil.

He aquí un fichero de zona clásico en la [Figura 17-17](#)

```

$ORIGIN domain.com
$TTL 86400
@      IN      SOA      dns1.domain.com.
hostmaster.domain.com. (
                                2001062501 ; serial
                                21600       ; refresh
after 6 hours
                                3600       ; retry
after 1 hour
                                604800    ; expires
after 1 week
                                86400 )    ; minimum
TTL of 1 day

```

```

                IN      NS      dns1.domain.com.
                IN      NS      dns2.domain.com.

                IN      MX      10      mail.domain.com.
                IN      MX      20
mail2.domain.com.

                        IN      A      10.0.1.5

server1          IN      A      10.0.1.5
server2          IN      A      10.0.1.7
dns1             IN      A      10.0.1.2
dns2             IN      A      10.0.1.3

ftp             IN      CNAME    server1
mail            IN      CNAME    server1
mail2           IN      CNAME    server2
www             IN      CNAME    server2

```

Figura 17-17. Ejemplo de fichero de zona básico

En este ejemplo se usan las directivas estándar y los valores SOA. Se ve que los servidores de nombres que tendrán la autoridad son `dns1.domain.com` y `dns2.domain.com`, que tienen como registros A que les conecta respectivamente a `10.0.1.2` y `10.0.1.3`.

Los servidores de correo configurados por los registros MX se orientan hacia los servidores `server1` y `server2` gracias a los registros CNAME. Debido a que los nombres

de los servidores `server1` y `server2` no acaban con un punto (`.`), el dominio `$ORIGIN` se situa a continuación, extendiendo los dominios `server1.domain.com` y `server2.domain.com`. Gracias a los registros de recursos `A` se pueden determinar las direcciones IP.

Los servicios `ftp` y de `web`, disponibles en `ftp.domain.com` y `www.domain.com`, están orientados a máquinas que ofrecen los servicios apropiados para dichos nombres usando el registro `CNAME`.

Ficheros de resolución de nombres inversa

Una resolución de nombres inversa sirve para traducir una dirección IP en un espacio de nombres particular en un FQDN. Esto se parece mucho a un fichero de zona estándar, salvo que los registros de recursos `PTR` sirven para relacionar las direcciones IP con los nombres de un sistema determinado.

La escritura de un registro `PTR` se hace de la misma manera que en la [Figura 17-18](#).

```
<last-IP-digit>      IN      PTR      <FQDN-  
of-system>
```

Figura 17-18. Ejemplo de configuración del registro `PTR`

`<last-IP-digit>` hace referencia al último nombre en una dirección IP que tiene que orientar el FQDN de un determinado sistema.

En la [Figura 17-19](#), las direcciones IP de `10.0.1.20` a `10.0.1.25` se orientan a los FQDNs correspondientes.


```

$ORIGIN 1.0.10.in-addr.arpa
$TTL 86400
@      IN      SOA      dns1.domain.com.
hostmaster.domain.com. (
                                2001062501 ; serial
                                21600       ; refresh
after 6 hours
                                3600        ; retry
after 1 hour
                                604800     ; expire
after 1 week
                                86400 )    ; minimum
TTL of 1 day

      IN      NS      dns1.domain.com.
      IN      NS      dns2.domain.com.

20    IN      PTR     alice.domain.com.
21    IN      PTR     betty.domain.com.
22    IN      PTR     charlie.domain.com.
23    IN      PTR     doug.domain.com.
24    IN      PTR     ernest.domain.com.
25    IN      PTR     fanny.domain.com.

```

Figura 17-19. Ejemplo de un fichero básico de resolución inversa de zona

Este fichero de zona se tiene que poner en funcionamiento con una declaración zone en el fichero /etc/named.conf, parecido a la [Figura 17-20](#).

```
zone "1.0.10.in-addr.arpa" IN {  
    type master;  
    file "domain.com.rr.zone";  
    allow-update { none; };  
};
```

Figura 17-20. Ejemplo de declaración zone de resolución inversa

Existe algunas pequeñas diferencias ente este ejemplo y una declaración zone estándar, salvo en la manera de nombrar el host. Observe que una zona de resolución de nombres inversa necesita que los tres primeros bloques de la dirección IP sean invertidos y que ".in-addr.arpa" esté a su vez incluido. Esto permite asociar correctamente a la zona el bloque único de nombres que se usa en el fichero de zona de resolución de nombres inversa.

Keywords: linux colombia tienda debian redhat red hat informática virtual compra comprar venta hosting cd libre gpl software hardware freeware slackware SuSE caldera mandrake stampede linux libro libros cd-rom cdrom nuevo ultimo último descarga administrador dns precio staroffice business card manual manuales release versión version documentacion documentación docs documentación info información distribución distribucion distribuciones oficial openlinux open aplicaciones kilyx comercial productos servicios catalogo barato bajo costo noticias recursos comentarios comercio electronico electrónico desarrollo tiendas virtuales linea online on-line soporte bogota nfs ftp http interchange minivend seguridad freebsd winlinux

instalación instalacion configuracion configuraci3n corel
carro carrito descripcion descripci3n mercado mercados
gratis gnu iso image unix windows download 2000 2001 5.2
8.1 7.1 7.0 6.1 6.2 6.3 7.2 7.3 8.0 8.1 8.2 9.0 9.1 9.2 1.0