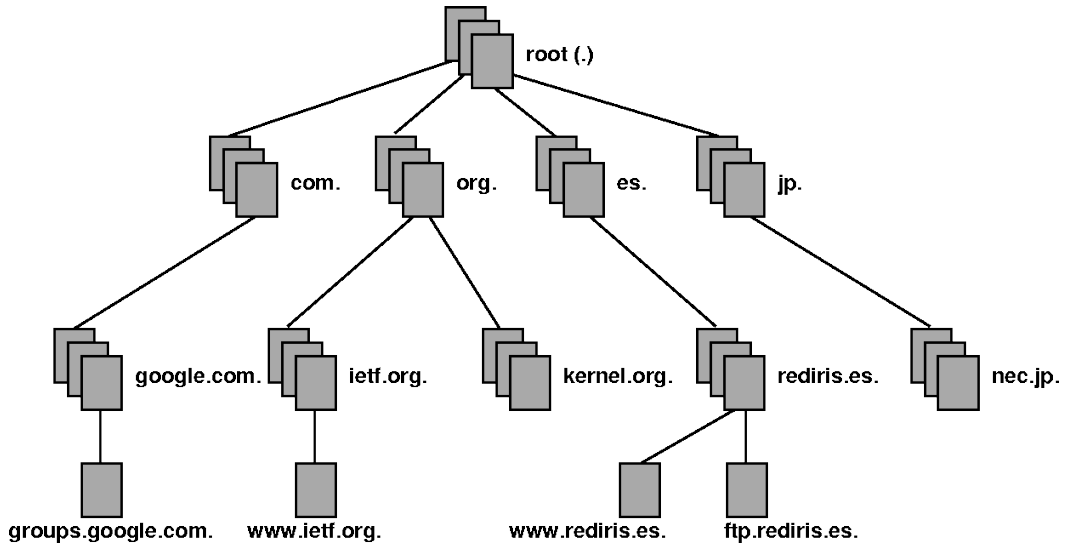
DNS + DHCP + PROXY + Lighttpd

¿ Qué es DNS (Domain Name System) ?

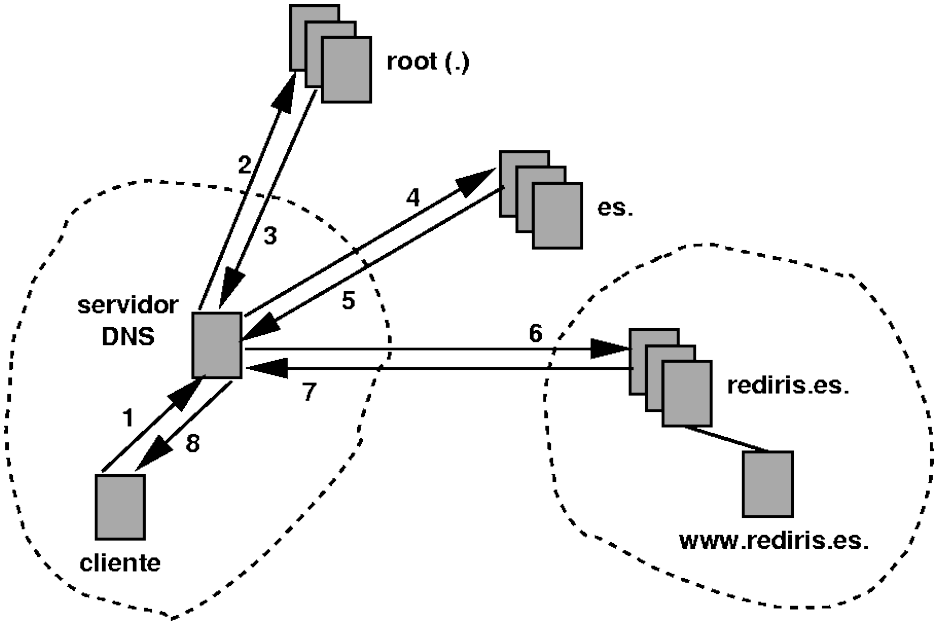
Es una base de datos distribuida, con información que se usa para traducir los nombres de dominio, fáciles de recordar y usar por las personas, en números de protocolo de Internet (IP) que es la forma en la que las máquinas pueden encontrarse en Internet.

El esquema que se muestra en la figura siguiente expone los conceptos mencionados de manera gráfica. En dicho esquema los tres niveles superiores representan servidores DNS mientras que el nivel inferior representa máquinas cualquiera de Internet. Las líneas que unen unos servidores con otros y con las máquinas de Internet muestran la relación existente entre las citadas máquinas, que siempre es desde un nivel a uno situado por debajo.



En el tope de la jerarquía está la raíz, representada por un punto. El servidor de nombres raíz sólo tendrá información acerca de cómo llegar a los servidores de nivel inmeditamente inferior, que son conocidos como dominios de primer nivel, o **TLD** (Top Level Domain). Existen más de doscientos **TLD** definidos actualmente, entre dominios nacionales (uno por cada nación, definidos por su identificador de dos letras según el estándar ISO-3166) y dominios genéricos (com, net, org, gov, mil, edu, y algunos más). Cada uno de estos dominios está gestionado por una empresa u organismo diferente, y configurado en uno o más servidores DNS repartidos por Internet.

El otro detalle interesante que comentar es que cada una de estas equivalencias entre nombre y dirección **IP**, IP y nombre, nombre y servidor de correo, etc. se identifican por un tipo de consulta DNS, lo que se conoce como el tipo de registro de recurso, o **RR** (Resource Record). Por ejemplo, la equivalencia entre nombre de máquina y sus direcciones **IP** (que pueden ser varias) se almacena en registros de tipo **A** (Address). La equivalencia inversa de dirección IP a nombre de máquina en registros de tipo **PTR** (PoinTeR), el registro que indica el servidor de correo entrante para un nombre de dominio concreto es de tipo **MX** (Mail eXchanger), los servidores de DNS para cada dominio se representan mediante registros **NS** (Name Server), y así un largo etcétera que puede consultar en el sitio oficial del **IANA** (Internet Assigned Numbers Authority).



¿ Que es un servidor de nombre solo cacheo ?

Un servidor de "sólo cache" encontrará las respuestas a las consultas de nombres y las recordará para la próxima vez que las necesites. Esto acorta el tiempo de espera significativamente para la próxima vez, especialmente si dispones de una conexión lenta.

Una primera nota para la configuración de DNS, muy útil para usuarios de módem, cable-módem, ADSL y similares.

Paquetes a instalar

```
# apt-get install bind9 bind9-host bind9utils bind9-doc
```

- **bind9** = Servidor de DNS.
- **bind9-host** = Utilidad host.
- **bind9utils** = Utilidades para chequear.
- **bind9-doc** = Documentación.

Lo primero que necesitamos es un fichero llamado **/etc/named.conf** (Debian: **/etc/bind/named.conf**). Este fichero se lee cuando arranca **bind9**. Por ahora simplemente contendrá:

```
# more /etc/bind/named.conf
```

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

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

more /etc/bind/named.conf.options

```
options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

         forwarders {
             IP_ISP_1;
             IP_ISP_2;
         };

        auth-nxdomain no;    # conform to RFC1035
        // listen-on-v6 { any; };
};
```

more /etc/bind/named.conf.default-zones

```
// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};
```

Reiniciamos el DNS.

/etc/init.d/bind9 restart

Modificamos el archivo **resolv.conf** que es donde le decimos cuales son nuestro **dns**.

vi /etc/resolv.conf

```
nameserver 127.0.0.1
```

Armando nuestro servidor de DNS

Creamos ahora nuestras zonas para nuestro servidor.

vi /etc/bind/named.conf.default-zones

```
// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};

zone "midominio.com.ar" {
        type master;
        file "/etc/bind/midominio.com.ar";
};

zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192.168.1.1";
};
```

vi /etc/bind/midominio.com.ar

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.midominio.com.ar. root.midominio.com.ar. (
                              1             ; Serial
                         604800       ; Refresh
                          86400        ; Retry
```

```
                        2419200         ; Expire
                        604800 )        ; Negative Cache TTL

        @       IN      NS      ns1.midominio.com.ar.

                IN      MX      10      mail.midominio.com.ar.

        ;
        ; Servidor dns
        ;
        ns1.midominio.com.ar.           IN      A       192.168.1.1

        ;
        ; Servicios
        ;
        wiki.midominio.com.ar.          IN      A       192.168.1.1
        pandora.midominio.com.ar.       IN      A       192.168.1.1
        proxy.midominio.com.ar.         IN      A       192.168.1.1
        svn.midominio.com.ar.           IN      A       192.168.1.1
```

vi /etc/bind/db.192.168.1.1

```
        ;
        ; BIND data file for local loopback interface
        ;
        $TTL    604800
        @       IN      SOA     ns1.midominio.com.ar. root.midominio.com.ar. (
                                1               ; Serial
                                604800          ; Refresh
                                86400           ; Retry
                                2419200         ; Expire
                                604800 )        ; Negative Cache TTL

        @       IN      NS      ns1.midominio.com.ar.

                IN      MX      10      mail.midominio.com.ar.

        ;
        ; Servidor dns
        ;
        1               IN      PTR     ns1.midominio.com.ar.
```

vi /etc/resolv.conf

```
        search midominio.com.ar
        nameserver 192.168.1.1
```

Chequeamos que todo funcione :

```
# named-checkconf /etc/bind/named.conf
# named-checkzone midominio.com.ar /etc/bind/db.midominio.com.ar
```

Probando :

```
# nslookup ns1
# nslookup ns1.midominio.com.ar
# nslookup 192.168.1.1
```

¿ Qué es un servidor dhcp ?

El protocolo de configuración dinámica de Host o DHCP es un protocolo que permite a los administradores de red automatizar y gestionar de manera centralizada la asignación de direcciones del protocolo Internet (IP) en una red de una organización o de un proveedor de servicios de Internet (ISP). Usando el conjunto de protocolos de Internet (TCP/IP), cada ordenador que puede conectarse a Internet necesita una dirección IP exclusiva. Cuando una organización configura los ordenadores de diferentes usuarios para que éstos se conecten a Internet, debe asignar una dirección IP a cada ordenador.

Sin DHCP, la dirección IP debe entrarse manualmente en cada ordenador, y si los ordenadores cambian de sitio a otro lugar de la red, hay que introducir una nueva dirección IP. DHCP permite al administrador de la red supervisar y distribuir las direcciones IP de forma centralizada enviando automáticamente una nueva dirección IP cada vez que un ordenador se conecta en un lugar diferente de la red o cuando llama al ISP.

DHCP usa el concepto de "alquiler" o "préstamo" de dirección IP, cuyo significado es que una dirección IP determinada será válida para un ordenador durante un cierto período de tiempo. La duración del préstamo puede variar dependiendo de cuánto tiempo suele conectarse a Internet el usuario de una ubicación determinada. Es especialmente útil en educación y en otros entornos en los que los usuarios cambian con frecuencia. Utilizando "préstamos" muy cortos, DHCP puede reconfigurar dinámicamente las redes en las cuales hay más ordenadores que direcciones IP disponibles.

Paquetes a instalar

apt-get install dhcpd

vi /etc/dhcpd.conf

```
subnet 192.168.1.0 netmask 255.255.255.0 {
        option domain-name "midominio.com.ar";
        option domain-name-servers 192.168.1.1;
        option subnet-mask 255.255.255.0;
        default-lease-time 3600;
        max-lease-time 7200;
        option routers 192.168.1.1;
        option broadcast-address 192.168.1.255;
        option netbios-name-servers 192.168.1.1;
        range  192.168.1.101 192.168.1.200;
}

        # Podemos asignar direcciones IP estáticas basadas en la MAC del cliente, como se muestra en el siguiente ejemplo.
        # Nota: Pueden conocerse las MAC de las máquinas conectadas a la red ejecutando el comando arp desde la terminal.

        host workstation {
                hardware ethernet 00:00:17:93:8D:05;
                fixed-address 192.168.1.100;
        }
```

/etc/init.d/dhcpd restart

Para los usuarios en GNU/Linux que quieran conectarse :

```
# dhclient ethX
```

O con el archivo de configuración **/etc/network/interfaces**

```
# vi /etc/network/interfaces
```

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
#NetworkManager#iface eth0 inet dhcp
```

¿ Qué es un servidor proxy ?

Un servidor **proxy** es un equipo intermediario situado entre el sistema del usuario e Internet. Puede utilizarse para registrar el uso de Internet y también para bloquear el acceso a una sede Web. El servidor de seguridad del servidor **proxy** bloquea algunas sedes o páginas Web por diversas razones.

Paquetes a instalar

```
# apt-get install squid squidguard
```

Configuramos nuestro proxy

```
# vi /etc/squid/squid.conf
```

```
…
acl all src all
…
#
# RED
#
acl red src 192.168.1.0/24

#
# Redirecciono la salida al SquidGuard
#
redirect_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf
…
#
# Habilito RED
#
http_access allow red

http_access deny all
```

Una vez terminado esto creamos el sistema de ficheros y directorios que usara **Squid** :

```
# squid -z
```

Configuración del squidGuard

El paquete **squidGuard** nos sirve para poder bloquear ciertas páginas.

```
# vi /etc/squid/squidGuard.conf
```

```
#
# CONFIG FILE FOR SQUIDGUARD
#

dbhome /var/lib/squidguard/db
logdir /var/log/squid

#
# TIME RULES:
# abbrev for weekdays:
# s = sun, m = mon, t =tue, w = wed, h = thu, f = fri, a = sat

time workhours {
        weekly mtwhf 08:00 - 16:30
        date *-*-01  08:00 - 16:30
}

#
# REWRITE RULES:
#

#rew dmz {
#       s@://admin/@://admin.foo.bar.de/@i
#       s@://foo.bar.de/@://www.foo.bar.de/@i
#}

#
# SOURCE ADDRESSES:
#

#src admin {
#       ip              1.2.3.4 1.2.3.5
#       user            root foo bar
#       within          workhours
#}

#
# Direcciones ip restringidas
#
src restringidas {
      ip      192.168.1.100
}

#
# Direcciones ip usuarios_permitidos
#
src usuarios_permitidos {
      ip      192.168.1.101
}

#
```

```
# adult - Sites containing adult material such as swearing but not porn
#
dest adult {
  domainlist     adult/domains
  urllist        adult/urls
  log            adult.log
  redirect       http://proxy.midominio.com.ar/squidGuard-simple.cgi?clientaddr=%a&clientname=
%n&clientuser=%i&clientgroup=%s&targetgroup=%t&url=%u
}

acl {
#       usuarios_permitidos {
#               pass permitidos none
#       }

        restringidas {
                pass !adult permitidos all
}
        default {
                pass none
                redirect http://proxy.midominio.com.ar/squidGuard-simple.cgi?clientaddr=%a&clientname=
%n&clientuser=%i&clientgroup=%s&targetgroup=%t&url=%u
        }
}
```

Bajamos las listas negras de : http://urlblacklist.com/?sec=download y lo descomprimimos en :

```
# tar xvf bigblacklist.tar.gz -C /var/lib/squidguard/db
```

Creamos la base de datos para cada una de las listas negras :

```
# squidGuard -c /etc/squid/squidguard.conf -C all
```

Si queremos alguna en particular :

```
# cd /var/lib/squidguard/db/adult
# squidGuard -c /etc/squid/squidguard.conf -C domain


# chown -R proxy:proxy /var/lib/squidguard/db
```

Instalamos un servidor de web pequeño

```
# apt-get install lighttpd lighttpd-doc
# vi /etc/lighttpd/lighttpd.conf
```

```
server.modules = (
        "mod_access",
        "mod_alias",
        "mod_compress",
        "mod_redirect",
#        "mod_rewrite",
)

server.document-root        = "/var/www"
server.upload-dirs          = ( "/var/cache/lighttpd/uploads" )
server.errorlog             = "/var/log/lighttpd/error.log"
server.pid-file             = "/var/run/lighttpd.pid"
```

```
server.username             = "www-data"
server.groupname            = "www-data"

index-file.names            = ( "index.php", "index.html",
                              "index.htm", "default.htm",
                             " index.lighttpd.html" )

url.access-deny             = ( "~", ".inc" )

static-file.exclude-extensions = ( ".php",".fcgi" )

cgi.assign = ( ".pl"  => "/usr/bin/perl",
               ".cgi" => "/usr/bin/perl" )


###include_shell "/usr/share/lighttpd/use-ipv6.pl"

dir-listing.encoding        = "utf-8"
server.dir-listing          = "enable"

compress.cache-dir          = "/var/cache/lighttpd/compress/"
compress.filetype           = ( "application/x-javascript", "text/css", "text/html", "text/plain" )

include_shell "/usr/share/lighttpd/create-mime.assign.pl"
include_shell "/usr/share/lighttpd/include-conf-enabled.pl"
```

Configuración de los clientes

Tenemos que configurar en cada pc cada navegador poniendo el **proxy** configurado.

Proxy : 192.168.1.1
Puerto : 3128