

# Sistema de nombres de dominio (DNS)

# Índice

## Índice de contenidos

<b>Objetivo</b>	<b>4</b>
<b>Definición</b>	<b>4</b>
Descripción	4
Funcionamiento	4
<b>Estructura</b>	<b>5</b>
Estructura de los nombres de dominio	5
Estructura de servidores DNS	5
<b>Operación</b>	<b>6</b>
Zonas DNS	6
Tipos de resolución	6
Iterativa	7
Recursiva	7
Caché de registros	7
Resolución inversa	7
<b>Historia</b>	<b>8</b>
Creación	8
Uso actual	8
<b>Seguridad y ataques</b>	<b>8</b>
Envenenamiento de caché	8
Control sobre DNS	9
<b>DNS dinámica</b>	<b>9</b>
<b>Parte práctica</b>	<b>10</b>
Uso de DNS dinámicas	10
DuckDNS	10
Instalación	12
Conclusión	13
Instalación de servidor DNS local propio	14
Dnsmasq	14
Conclusión	16
Cómo evitar anuncios, trackers y bloquear páginas utilizando DNS	16
Archivo de hosts	16
Pi-hole	17
Conclusión	19
DNS teniendo un dominio	20

## Índice de figuras

- [Figura 1. Diagrama de pasos de una petición \(Cloudflare\)](#)
- [Figura 2. Página de inicio de DuckDNS](#)
- [Figura 3. Panel de control de DuckDNS](#)
- [Figura 4. Comprobación del nombre de la DDNS de prueba](#)
- [Figura 5. Comprobación de paquetes necesarios](#)
- [Figura 6. Script de conexión](#)
- [Figura 7. Pasos para la ejecución autónoma del script](#)
- [Figura 8. Crontab de ejecución del script](#)
- [Figura 9. Comprobación del log](#)
- [Figura 10. Comprobación de la IP actualizada desde el panel de control](#)
- [Figura 11. Comprobación de paquetes necesarios](#)
- [Figura 12. Modificación de la configuración de dnsmasq](#)
- [Figura 13. Modificación del tamaño de la caché](#)
- [Figura 14. Servidores DNS escogidos para elevar las peticiones](#)
- [Figura 15. Comprobación del estado del servicio tras configuración](#)
- [Figura 16. Modificación del archivo hosts](#)
- [Figura 17. Comprobación del funcionamiento local del DNS](#)
- [Figura 18. Comprobación del funcionamiento a través de Internet del DNS](#)
- [Figura 19. Pantalla de configuración de Pi-hole](#)
- [Figura 20. Pantalla de inicio de sesión del panel de control de Pi-hole](#)
- [Figura 21. Panel de control de Pi-hole](#)
- [Figura 22. Listado de peticiones](#)
- [Figura 23. Panel de gestión de DNS de Google Domains](#)

## Bibliografía

- [Domain Name System - Wikipedia](#)
- [Sistema de nombres de dominio - Wikipedia, la enciclopedia libre](#)
- [Protocolo de internet - Wikipedia, la enciclopedia libre](#)
- [Envenenamiento de DNS - Wikipedia, la enciclopedia libre](#)
- [What is DNS? | How DNS works | Cloudflare](#)
- [What is DNS? – Introduction to DNS - AWS](#)
- [Dynamic DNS - Wikipedia](#)
- [Top-level domain - Wikipedia](#)
- [ICANN - Wikipedia](#)
- [Root name server - Wikipedia](#)
- [How to Run Your Own DNS Server on Your Local Network](#)
- [DNS dinámico - Wikipedia, la enciclopedia libre](#)
- [Are there really 7 keys to the internet?](#)
- [GitHub - pi-hole/pi-hole: A black hole for Internet advertisements](#)
- [What is reverse DNS? | Cloudflare](#)

# Objetivo

El objetivo de este trabajo es comprender el funcionamiento elemental de DNS y las utilidades que tiene en el mundo real. Además, se realizarán una serie de aplicaciones prácticas donde se aprovecha el protocolo para facilitar el uso cotidiano de varios dispositivos.

## Definición

### Descripción

El sistema de nombres de dominio, en inglés *Domain Name System*, más popularmente conocido por sus siglas *DNS*, es un sistema de nomenclatura jerárquico utilizado por redes que utilicen el protocolo IP que asocia información sobre una máquina a un nombre de dominio. La parte más importante de esa “información” es la IP numérica del equipo que lo identifica en la red. Es un componente esencial para el funcionamiento de Internet como lo conocemos, como se describirá [más tarde](#).

Pese a que puede parecer complejo, su funcionamiento es similar al de una libreta de contactos o a la aplicación de contactos de cualquier dispositivo móvil. De la misma manera que cada persona asigna el nombre de alguien a un número de teléfono, el protocolo DNS asigna un nombre más o menos difícil de recordar, dependiendo del dominio escogido, a una dirección IP.

### Funcionamiento

Suponiendo el ejemplo más sencillo y habitual: cuando un usuario escribe un nombre de dominio en su navegador web, el protocolo DNS realiza una serie de pasos para encontrar la dirección IP correspondiente:

1. Se consulta una “caché local” en el dispositivo del usuario para ver si ya se ha almacenado la dirección IP correspondiente recientemente.
2. De lo contrario, se hace una solicitud al servidor DNS local, que depende de la configuración del router de la red.
  - a. Si no se ha configurado nada manualmente, el servidor DNS suele pertenecer al proveedor de Internet, que gestiona la petición.
  - b. Si se ha cambiado la configuración, la petición se puede realizar a cualquier servidor DNS válido. El *administrador* puede crear su propio servidor o puede escoger entre una gran variedad de servidores disponibles, normalmente de manera gratuita, como el 1.1.1.1 de Cloudflare, el popular 8.8.8.8 de Google, el OpenDNS de Cisco, entre otros.
3. Si este servidor “local” no tiene la información solicitada, se realizan peticiones a servidores de mayor rango hasta encontrar la dirección IP, en caso de que sea válida.

# Estructura

## Estructura de los nombres de dominio

Los nombres de dominio están divididos según una estructura de árbol, donde cada nodo del árbol tiene una *etiqueta*, de manera que, para la URL `ssh.mier.info`, `ssh` sería un nodo hijo del nodo padre `mier.info`, que a su vez depende del dominio de nivel superior `info`.

- La parte más a la derecha, el dominio de nivel superior (más conocido por sus siglas en inglés *TLD*) depende de la gestión de organizaciones como la *ICANN*<sup>1</sup>.
  - Los *Top Level Domains* pueden ser de varios tipos y tener varias restricciones. Pueden representar a empresas, países, reservados para la infraestructura, genéricos, genéricos restringidos...
  - Existen muchos dominios de nivel superior y se añaden nuevos frecuentemente. Algunos son muy comunes, como `.com` o `.net`, pero otros son muy raros e infrecuentes, como `.voyage` o `.exchange`.
- Los nodos hijos de los dominios de nivel superior son los nodos *raíz* de cada dominio, que otorgan el nombre reconocible y diferenciable de cada uno, como por ejemplo `mier` o `uniovi`.
- Cada etiqueta a la izquierda del nodo raíz supone un “subdominio”. Tradicionalmente, el subdominio más a la izquierda es el nombre de la máquina a la que se le quiere dar nombre, como `di119` en `di119.edv.uniovi.es`. Cada subnivel divide al nivel superior en varias partes, aunque hoy en día no se utilice con el propósito de dividir y se utilice más frecuentemente para acceder a servicios o funcionalidades diferentes dentro de cada página web.

Todos los nombres de dominio deben seguir una determinada estandarización y reglas de internacionalización convencionales, como el uso de los puntos como delimitadores, el límite de subdominios, la longitud y caracteres permitidos...

## Estructura de servidores DNS

Como ya se ha comentado anteriormente, una *resolución* DNS pasa por varios servidores hasta ser completada, es decir, hasta que se encuentre un servidor que consiga traducir el nombre de dominio. Existen cuatro tipos de servidores DNS:

- El servidor DNS local del equipo (recursor), que guarda en caché las respuestas que haya obtenido hasta que se cumpla su *TTL* (tiempo de vida) para evitar hacer la misma consulta varias veces. Es el responsable de hacer peticiones adicionales para satisfacer la resolución.

---

<sup>1</sup> La ICANN es una organización internacional sin fines de lucro responsable de dividir el espacio de los protocolos IP, gestión de sistema de dominio y otras administraciones referentes al funcionamiento general de Internet.

- El “root server”, que es el primer paso que utiliza el servidor local para traducir la dirección IP. Normalmente, sirve de referencia a otras localizaciones más específicas.
- El servidor TLD, que almacena la información sobre servidores que contienen las [zonas DNS](#).
- El servidor autoritativo es el servidor final que se encarga de traducir la petición. Si el servidor tiene la información requerida, se la devuelve al DNS recursor que hace la petición inicial.

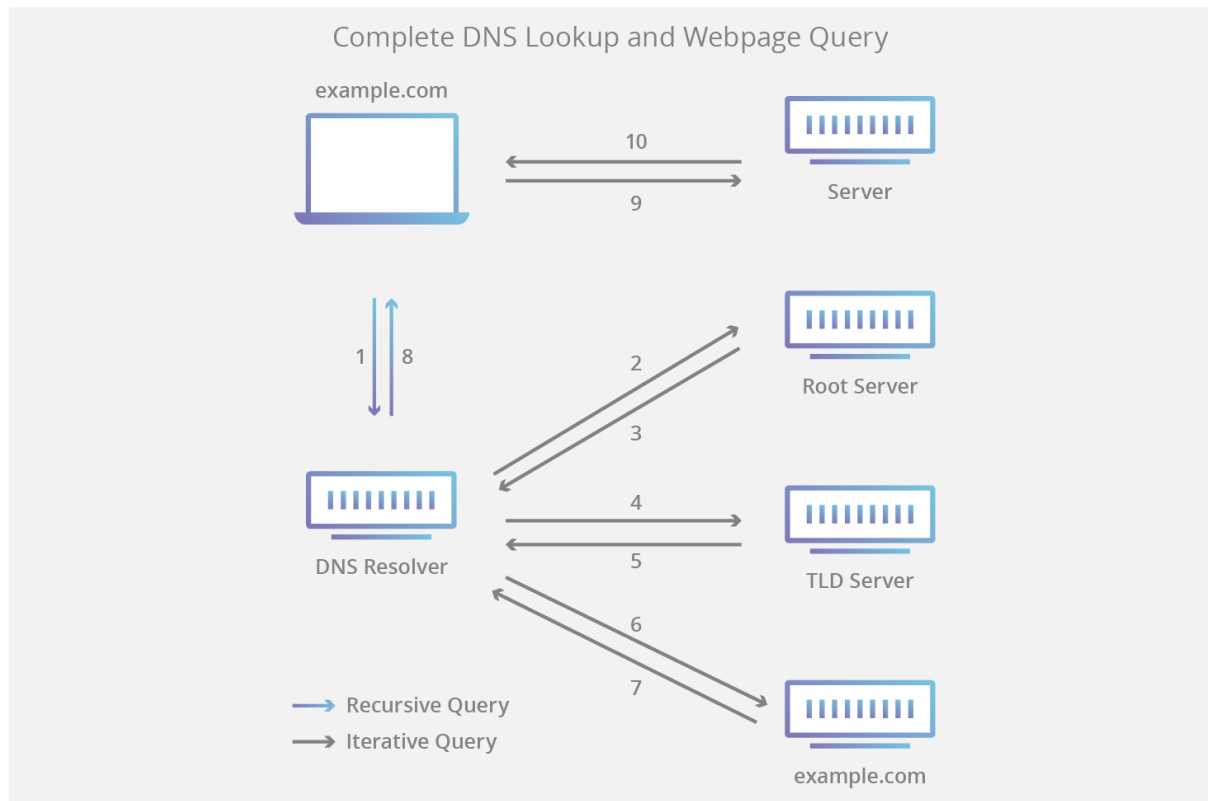


Figura 1. Diagrama de pasos de una petición (Cloudflare)

## Operación

### Zonas DNS

Las zonas DNS son los nodos que contienen la información de cada dominio, es decir, toda la información sobre ese dominio. Normalmente, un servidor DNS contiene una zona, como cuando se [establece](#) la información DNS sobre un dominio, aunque un servidor puede contener la información de más de una zona.

### Tipos de resolución

Las consultas realizadas al servidor DNS local son siempre recursivas y las peticiones que realiza dicho servidor son recursivas pero, ¿cuál es la diferencia entre ambas?

## Iterativa

En una consulta iterativa, el servidor DNS no otorga una respuesta completa, sino que el primer servidor al que se le consulta devuelve la dirección del siguiente servidor que se debe consultar para completar la petición.

## Recursiva

En una consulta recursiva, el cliente solicita a un servidor que obtenga la respuesta completa, de manera que el servidor se encarga de realizar la petición a otros servidores y devolver la resolución.

## Caché de registros

A la hora de resolver dominios, es normal que se almacene información reciente para reducir la carga generada. Los resultados obtenidos siempre van asociados a un *TTL* o *Time To Live*, que supone la cantidad de tiempo tras la que expira la información. Este tiempo de vida lo establece el administrador de los servidores autoritativos, y pueden variar de unos pocos segundos a semanas.

Como resultado de esto, los cambios en los registros de DNS no se propagan<sup>2</sup> inmediatamente y normalmente se tarda horas en que todas las cachés expiren y la información se refresque.

## Resolución inversa

La resolución inversa es la estrategia de resolución de nombres cuando lo que se conoce es la IP numérica de host, en lugar de su nombre de dominio. Para esto, la DNS almacena las direcciones de puntero (PTR), sin las cuales no podría realizar este servicio. Normalmente, se almacenan las direcciones IP al revés con el dominio especial `.in-addr.arpa`, como por ejemplo `88.0.168.192.in-addr.arpa`.

Estas direcciones suelen ser utilizadas por servidores de correo electrónico para comprobar la validez de los correos y también por servicios administrativos para mostrar nombres de dominio en lugar de IPs numéricas, como lo hace [Pi-hole](#).

---

<sup>2</sup> Propagar (en inglés *propagate*) es un término muy común a la hora de modificar registros de nombres.

# Historia

## Creación

Los orígenes del protocolo DNS datan de la época de ARPANET<sup>3</sup>, cuando se mantenían archivos “hosts”<sup>4</sup> de texto con las direcciones de los servidores introducidos de manera manual. Con el crecimiento del número de nodos conectados a la red, el sistema se volvió lento e ineficiente, motivo por el que se comenzaron a desarrollar diferentes alternativas. La escogida, obviamente, fue el protocolo DNS, escrito por Paul Mockapetris. Las especificaciones originales fueron publicadas en noviembre de 1983 y actualizadas por última vez en enero de 1986.

## Uso actual

El protocolo DNS es una parte esencial de la infraestructura de Internet y se utiliza millones de veces por hora. Debido a que es una parte tan esencial para su funcionamiento, cualquier problema o vulnerabilidad supone un gran riesgo para la estabilidad y seguridad de todo Internet.

## Seguridad y ataques

Puesto que se desarrolló en una época donde el uso de Internet no estaba para nada extendido y el acceso al mismo estaba limitado, la seguridad no es una prioridad en los estándares que definen al protocolo. Sin embargo, el paso del tiempo y la masificación y globalización de Internet han generado la necesidad de mejorar la seguridad y robustez del protocolo DNS.

## Envenenamiento de caché

El problema de seguridad más importante que supone el protocolo es el *envenenamiento de caché*<sup>5</sup>, que consiste en la distribución por parte de un actor maligno de registros que no se originan de servidores autoritativos y que están maliciosamente diseñados para engañar tanto al resto de la estructura de servidores como al propio usuario, de manera que el envenenamiento se propaga a otras partes del ciclo y se puede mantenerse hasta que el paquete alcance el *TTL*.

---

<sup>3</sup> ARPANET fue la primera red de computadores a distancia como medio de comunicación entre instituciones y universidades en los Estados Unidos a principios de los 80.

<sup>4</sup> Los archivos *hosts* mantienen su utilidad original a día de hoy y están presentes en todos los dispositivos. En Windows, está en C:\WINDOWS\system32\drivers\etc\hosts, en Linux en /etc/hosts.

<sup>5</sup> El envenenamiento de caché fue descubierto por [Steve Bellovin](#) en 1990.



Para intentar evitar el envenenamiento de cachés, existen varias alternativas propuestas por organizaciones internacionales y utilizadas comúnmente, siendo la más importante *DNSSEC*. Todas ellas consisten en el cifrado de respuestas, de manera que se pueda verificar la fuente de los registros.

Esta criptografía es multinivel, lo que significa que los mensajes están firmados por varias claves. La ICANN es la organización que se encarga, explicado de manera sencilla, de firmar las firmas del resto de organizaciones y empresas que administran los registros de dominio. Esta firma de firmas supone un punto débil muy importante para todo Internet, por lo que el proceso de generación es muy transparente, largo y seguro que se graba y publica en directo en diversas plataformas y cuenta con múltiples testigos y barreras de seguridad que aseguran y verifican todo el proceso.

## Control sobre DNS

Tener el control sobre servidores autoritarios o sobre la cadena del protocolo DNS supone tener el control sobre Internet. Por supuesto, esto no significa que las páginas en sí tengan vulnerabilidades o que existan problemas de conexión entre una página y un cliente, simplemente nadie sería capaz de traducir los nombres de dominio de las páginas a direcciones IPs, que son lo necesario para establecer una conexión a ellas.

Rusia, por ejemplo, ha realizado diversas [pruebas de desconexión](#) de servidores raíz globales con éxito, gracias a la cooperación de teleoperadoras y grandes compañías rusas como Yandex o Runet. Esto permitiría a Rusia establecer su [propio Internet](#), separado del resto del mundo o incluso [conectado a otros países](#) más afines como China.

## DNS dinámica

*Dynamic DNS* (DDNS) es un método de actualización automática de los registros de direcciones IP, normalmente ante cambios generados por la variación de IPs debido al funcionamiento del proveedor de Internet. Este sistema permite la conexión con una máquina sin necesidad de conocer su dirección IP en ese momento.

Esto facilita, por ejemplo, la conexión o el alojamiento con un servidor de nuestra casa, sin necesidad de una infraestructura cara. Normalmente, la actualización del registro se lleva a cabo en la fase de arranque o reinicio de la máquina en cuestión.

## Parte práctica<sup>67</sup>

### Uso de DNS dinámicas

Existen servicios en Internet que ofrecen dominios con DNS dinámicas, algunos de ellos gratuitos. En esta sección se explica cómo instalar y utilizar uno de ellos.

#### DuckDNS

Para la práctica, se va a utilizar *DuckDNS*, un servicio gratuito de alojamiento de DNS dinámicas en la nube de Amazon, con un sistema de inicio de sesión muy sencillo y con tutoriales y guías excelentes para la instalación en múltiples aparatos, en este caso una Raspberry Pi.



Figura 2. Página de inicio de DuckDNS

Para comenzar, iniciamos sesión utilizando cualquiera de las plataformas ofrecidas.

<sup>6</sup> Todas las prácticas se realizan sobre máquinas Linux reales a través de SSH (*Raspberry Pi 3B+*, *Dell Poweredge T110 II*).

<sup>7</sup> Todas las capturas de pantalla son originales.

The screenshot shows the Duck DNS control panel. At the top, there's a navigation bar with links: spec, about, why, install, faqs, logout, and a 'logged in with' status. The main header features the Duck DNS logo (a yellow duck) and the text 'Duck DNS'. Below this, account details are displayed: 'account' (redacted), 'type' (free), 'token' (redacted), 'token' (1 year ago), 'generated' (11 May 2021, 18:21:08), and 'created date'. The 'domains' section shows a list of domains with columns: domain, current ip, ipv6, and changed. A domain is listed with a redacted current ip, a redacted ipv6 address, and a 'changed' date of '6 months ago'. Below the domain list, there are buttons for 'update ip', 'update ipv6', and 'delete domain'. At the bottom, a note states: 'This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply.'

*Figura 3. Panel de control de DuckDNS*

Tras iniciar sesión, podemos crear un nombre de dominio y asignarlo a una dirección IP de manera instantánea, teniendo en cuenta claro que, al tratarse de un servicio gratuito, será un subdominio de la misma página web. Por defecto, al generar un subdominio se utilizará la IP desde la que se está conectando a la página web.

Si nuestro objetivo es simplemente tener un dominio fácil de recordar, ya no hay nada más que hacer: tras asignar la IP, podemos hacer uso del nombre de subdominio escogido y probarlo:

```

mier@apd ~ → ping admsisU0283319.duckdns.org
PING admsisU0283319.duckdns.org 56(84) bytes of data.
64 bytes from : icmp_seq=1 ttl=64 time=0.578 ms
64 bytes from : icmp_seq=2 ttl=64 time=0.341 ms
64 bytes from : icmp_seq=3 ttl=64 time=0.347 ms
64 bytes from : icmp_seq=4 ttl=64 time=0.407 ms
^C
--- admsisU0283319.duckdns.org ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3089ms
rtt min/avg/max/mdev = 0.341/0.418/0.578/0.095 ms
mier@apd ~ → |

```

*Figura 4. Comprobación del nombre de la DDNS de prueba*

Como se puede comprobar, el dominio funciona correctamente y se hace ping a la dirección deseada (<1 ms porque es la misma dirección IP).

## Instalación

El punto fuerte de estos servicios está en establecer una conexión con el proveedor de manera que la IP se actualice cada cierto tiempo. Para ello, dentro de la pestaña *install* se encuentran una serie de guías de instalación para una gran cantidad de sistemas operativos y dispositivos especiales. En este caso, puesto que vamos a instalarlo en una máquina Linux, se hará uso de *cron*, que es un “demonio” de automatización muy sencillo.

Primero de todo, nos aseguramos de tener *curl* y *cron* instalados en el sistema:

```

pi@raspberrypi:~$ sudo apt install curl cron
Reading package lists... Done
Building dependency tree
Reading state information... Done
cron is already the newest version (3.0pl1-134+deb10u1).
curl is already the newest version (7.64.0-4+deb10u2).
0 upgraded, 0 newly installed, 0 to remove and 15 not upgraded.

```

*Figura 5. Comprobación de paquetes necesarios*

A continuación, se crea un pequeño script de shell que se encargará de hacer las peticiones al servidor. Dichas peticiones deberán incluir el token personal y el nombre de dominio que se quiera actualizar.

```

pi@raspberrypi:~/duckdns$ cat duck.sh
echo url="https://www.duckdns.org/update?domains=admsisU0283319&token=_____&ip=" | curl -k
-o ~/duckdns/duck.log -k -

```

*Figura 6. Script de conexión*

Por último, se deben otorgar permisos de ejecución e incluir el script en el listado de *cron* con el comando `crontab -e`.

```

pi@raspberrypi:~/duckdns$ chmod u+x duck.sh
pi@raspberrypi:~/duckdns$ crontab -e
crontab: installing new crontab

```

*Figura 7. Pasos para la ejecución autónoma del script*

```

GNU nano 3.2 /tmp/crontab.PqQzGj/crontab
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
0 */8 * * tue,fri,sun ~/duckdns/duck.sh >/dev/null 2>&1

```

*Figura 8. Crontab de ejecución del script*

Puesto que no es para nada necesario ejecutar el script muy a menudo, en este caso se ejecutará cada ocho horas los martes, viernes y domingos.

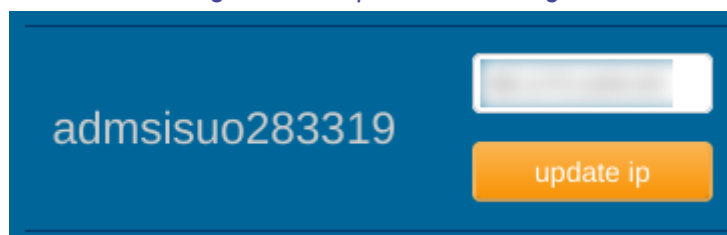
Una vez ejecutado, se observa el contenido del archivo log para comprobar que todo funciona y que la IP asignada es la correcta:

```

pi@raspberrypi:~/duckdns$ cat duck.log
OKpi@raspberrypi:~/duckdns$ |

```

*Figura 9. Comprobación del log*



*Figura 10. Comprobación de la IP actualizada desde el panel de control*

## Conclusión

Además de *DuckDNS*, existen otras páginas web tanto gratuitas como de pago con este objetivo, algunas de ellas con mayor personalización a la hora de escoger el nombre de dominio y con diferentes características.

Personalmente, llevo utilizando este tipo de servicios durante años ya que es muy cómodo a la hora de realizar tareas rutinarias como conexiones SSH, alojamiento de partidas de videojuegos, páginas web de monitorización, etc. Pese a que no ofrece el nivel de personalización que se tiene al tener y gestionar la DNS de un dominio propio, es un sistema fácil de implementar y más aún de utilizar.

De manera alternativa, algunos routers tienen integrada esta funcionalidad, de manera que las llamadas de actualización se realizan de manera automática.

## Instalación de servidor DNS local propio

Utilizar un servidor personal de DNS supone una posible mejora de rendimiento y de protección frente a caídas de DNS generales, además de permitir la configuración y personalización de las respuestas de DNS no solo de la red de casa, sino de tus dispositivos sobre la marcha al utilizarlo de servidor DNS normal.

### Dnsmasq

En Linux, es muy sencillo establecer tu propio servidor gracias a *Dnsmasq*, que viene incluido en la mayoría de distribuciones.

Primero de todo, debemos comprobar que esté instalado, como de costumbre:

```
server@poweredge ~> sudo dnf install dnsmasq
[sudo] password for server:
Failed to set locale, defaulting to C.UTF-8
Last metadata expiration check: 2:25:28 ago on Sat Dec 31 08:26:24 2022.
Package dnsmasq-2.86-10.fc36.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

*Figura 11. Comprobación de paquetes necesarios*

Tras esto, se modifica el archivo de configuración por defecto para que funcione en direcciones locales:

```
# Never forward plain names (without a dot or domain part)
domain-needed
# Never forward addresses in the non-routed address spaces.
bogus-priv
```

*Figura 12. Modificación de la configuración de dnsmasq*

Al descomentar las dos opciones de arriba:

- Se reservan los nombres de dominio sin TLD para la red local y el resto de dominios para resolución normal.
- Las direcciones locales no se elevarán a otros servidores DNS.

Ahora, se debe escoger el servidor DNS al que se eleven las peticiones no resueltas y se cambia el tamaño de la caché para almacenar más *queries*:

```
# Set the cachesize here.
cache-size=1000|
```

*Figura 13. Modificación del tamaño de la caché*

```
server=1.1.1.1
server=1.0.0.1|
```

*Figura 14. Servidores DNS escogidos para elevar las peticiones*

Por último, se reinicia el servicio y se comprueba el estado del mismo:

```
server@poweredge ~> sudo systemctl restart dnsmasq
[sudo] password for server:
server@poweredge ~> sudo systemctl status dnsmasq
● dnsmasq.service - DNS caching server.
   Loaded: loaded (/usr/lib/systemd/system/dnsmasq.service; disabled; vendor preset: disabled)
   Active: active (running) since Sat 2022-12-31 11:07:40 CET; 5s ago
```

Figura 15. Comprobación del estado del servicio tras configuración

Para añadir resoluciones locales (como en las prácticas de laboratorio de la asignatura) o bloquear direcciones, se puede hacer uso del archivo *hosts* del sistema, ya que *dnsmasq* lo lee al inicio del servicio:

```
GNU nano 6.0 /etc/hosts
# Loopback entries; do not change.
# For historical reasons, localhost precedes localhost.localdomain:
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
# See hosts(5) for proper format and other examples:
# 192.168.1.10 foo.mydomain.org foo
# 192.168.1.13 bar.mydomain.org bar
192.168.0.2 pi
192.168.0.10 apd
```

Figura 16. Modificación del archivo hosts

```
server@poweredge ~> dig apd @localhost

; <<>> DiG 9.16.33-RH <<>> apd @localhost
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 28684
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;apd. IN A

;; ANSWER SECTION:
apd. 0 IN A 192.168.0.10

;; Query time: 0 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sat Dec 31 11:14:52 CET 2022
;; MSG SIZE rcvd: 48
```

Figura 17. Comprobación del funcionamiento local del DNS

```

server@poweredge ~> dig mier.info @localhost

; <<>> DiG 9.16.33-RH <<>> mier.info @localhost
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 10678
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
;; QUESTION SECTION:
;mier.info.                IN      A

;; ANSWER SECTION:
mier.info.                3600    IN      A      199.36.158.100

;; Query time: 114 msec
;; SERVER: ::1#53(::1)
;; WHEN: Sat Dec 31 11:15:37 CET 2022
;; MSG SIZE rcvd: 54

```

Figura 18. Comprobación del funcionamiento a través de Internet del DNS

## Conclusión

*dnsmasq* ofrece mucho más de lo que se ve en este sencillo tutorial, con una gran cantidad de opciones que permiten sacarle el máximo partido a tener un servidor DNS propio. Además, la configuración de nombres para la red local es extremadamente sencilla y funcional. Es recomendable hacer uso de IPs estáticas dentro de la red, tanto para el servidor DNS como para otros dispositivos a los que se quieran asignar nombres.

## Cómo evitar anuncios, trackers y bloquear páginas utilizando DNS

Aparte de los populares *ad-blockers*, se puede utilizar el protocolo DNS para bloquear peticiones a ciertos servidores traduciendo la IP numérica por 0.0.0.0. Esto supone una mejora en la privacidad, seguridad y experiencia de uso de Internet, que dependiendo de la implementación y configuración escogida puede ser individual a cada dispositivo o para la red entera. Normalmente, se utilizan listas de dominios de *adware*, *malware*, relativas al juego, pornográficos o referentes a redes sociales intrusivas.

Para conseguir esto, existen dos métodos relativamente conocidos dentro del mundo de la informática: el archivo *hosts* y *Pi-hole*, siendo el primero el más sencillo y el segundo el más poderoso.



## Archivo de *hosts*

Como se ha visto en la parte anterior, el archivo *hosts* contiene reglas de traducción casi instantáneas que evitan tener que hacer llamadas a servidores DNS. Sin embargo, se puede utilizar esto para enviar peticiones a ciertos servidores al vacío. Al utilizar una de las listas de dominios ya mencionados, como las populares [listas de StevenBlack](#), se puede bloquear una gran cantidad de anuncios y otros de manera rápida y segura.

## Pi-hole

*Pi-hole* utiliza una versión modificada de *dnsmasq* para bloquear peticiones DNS actuando como un servidor DNS propio. Lo bueno de este servicio es que obtiene listas de internet que se actualizan, además de contar con una interfaz sencilla de operar y que ofrece una gran cantidad de datos sobre la red, las peticiones filtradas, una lista de dispositivos según las peticiones y otras funcionalidades excelentes. Al igual que un servidor DNS cualquiera, está pensado para reemplazar el servidor DNS del proveedor de Internet para que actúe sobre dispositivos que normalmente no cuentan con la posibilidad de instalar otros métodos de bloquear anuncios (teléfonos móviles sin root, televisiones inteligentes, dispositivos inteligentes...)

Para instalar *pi-hole*, solo hay que ejecutar el siguiente comando: `curl -sSL https://install.pi-hole.net | sudo bash`. Al ejecutarlo, aparecerá un instalador gráfico muy sencillo de utilizar, donde se puede configurar las listas iniciales y decidir la instalación de la interfaz de administración (recomendado):

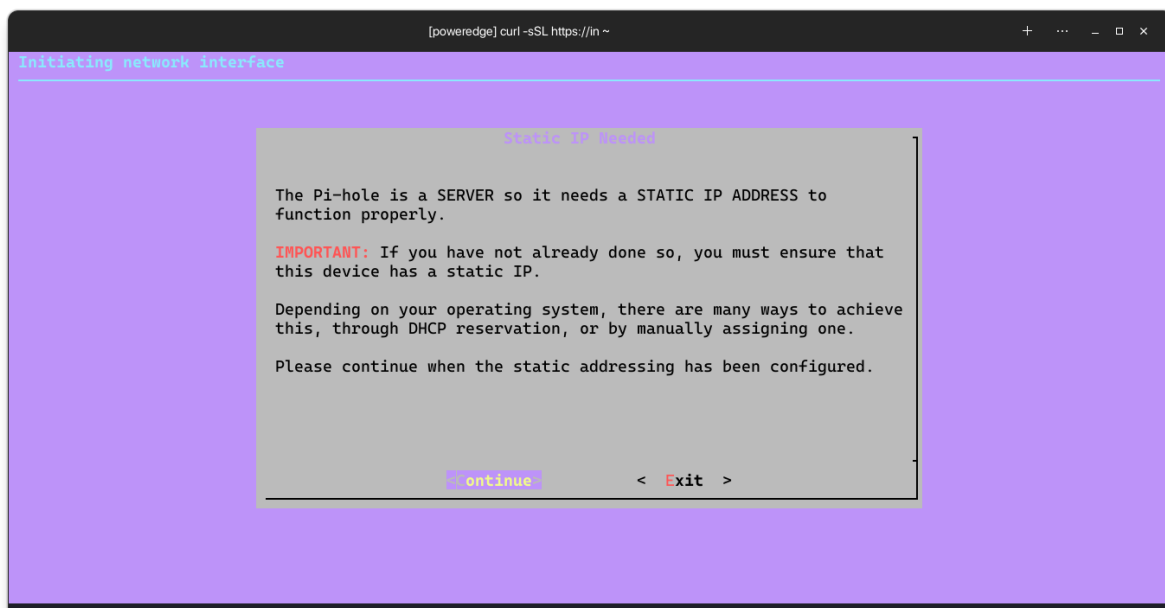
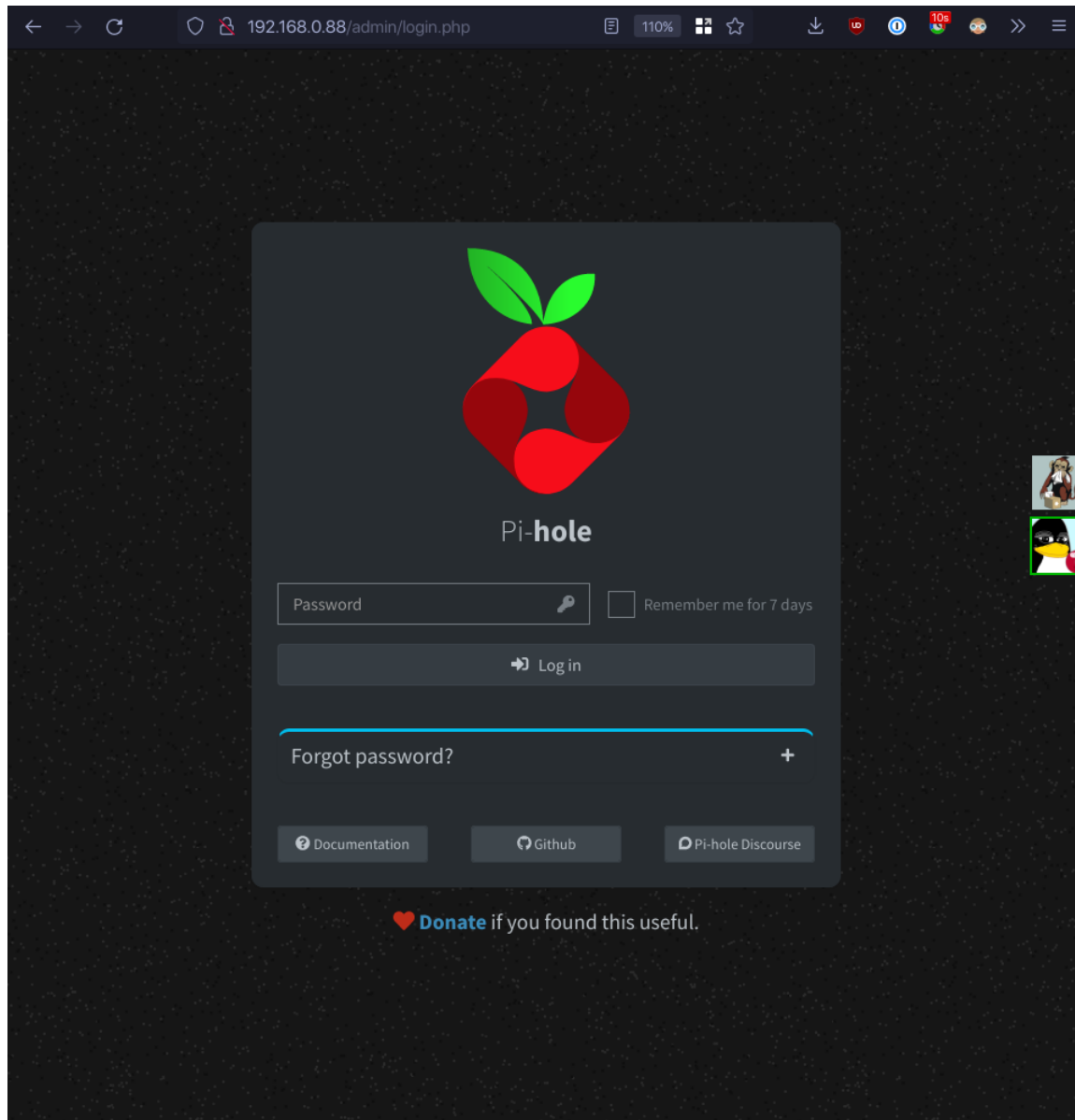


Figura 19. Pantalla de configuración de Pi-hole

Tras reiniciar, el servicio ya está listo para ser utilizado. En general, para llevar a cabo la administración de servicio se hará uso de la interfaz web instalada:



*Figura 20. Pantalla de inicio de sesión del panel de control de Pi-hole*

Para establecer una contraseña, se utiliza el comando `pihole -a -p`. Para actualizar el servicio, tan solo hay que escribir `pihole -g`.

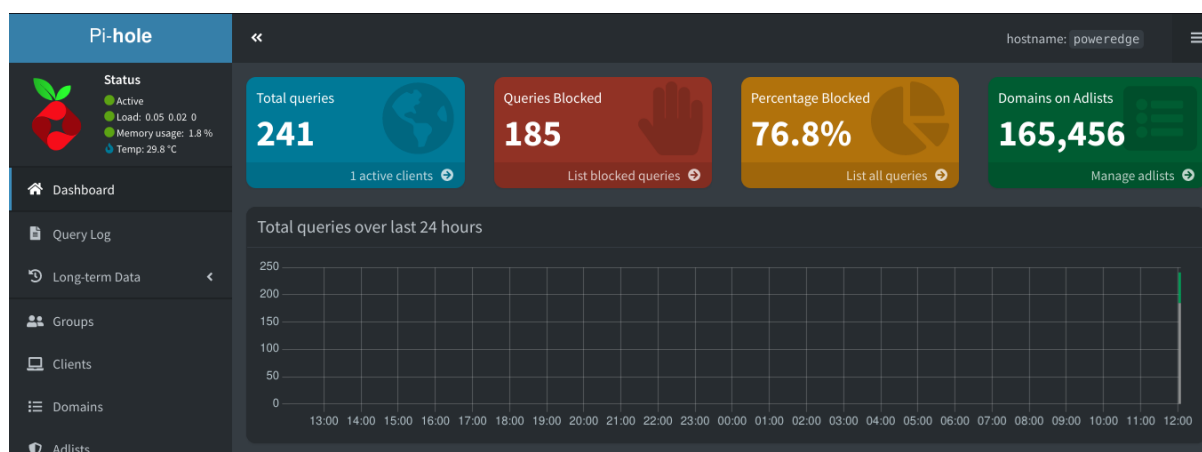


Figura 21. Panel de control de Pi-hole

Como se puede comprobar, el servicio funciona satisfactoriamente. Se filtran las peticiones a los servidores presentes en las listas y el resto de peticiones las responde el servidor DNS indicado, en este caso `one.one.one.one` de Cloudflare.

Show 10 entries

Previous

1

2

3

4

5

...

10

Next

Time	Type	Domain	Client	Status	Reply	Action
2022-12-31 12:08:28	A	addons-pa.clients6.google.com	apd	OK (answered by one.one.one.one#53)	IP (18.8ms)	Blacklist
2022-12-31 12:08:28	AAAA	addons-pa.clients6.google.com	apd	OK (answered by one.one.one.one#53)	IP (18.9ms)	Blacklist
2022-12-31 12:08:13	AAAA	api.accounts.firefox.com	apd	OK (answered by one.one.one.one#53)	NODATA (18.4ms)	Blacklist
2022-12-31 12:08:13	A	api.accounts.firefox.com	apd	OK (answered by one.one.one.one#53)	IP (18.5ms)	Blacklist
2022-12-31 12:08:13	A	sync-1-us-west1-g.sync.services.mozilla.com	apd	OK (answered by one.one.one.one#53)	IP (24.2ms)	Blacklist
2022-12-31 12:08:13	AAAA	profile.accounts.firefox.com	apd	OK (answered by one.one.one.one#53)	NODATA (19.2ms)	Blacklist
2022-12-31 12:08:13	A	profile.accounts.firefox.com	apd	OK (answered by one.one.one.one#53)	IP (26.5ms)	Blacklist
2022-12-31 12:07:38	A	ssl.gstatic.com	apd	OK (answered by one.one.one.one#53)	IP (19.7ms)	Blacklist

Figura 22. Listado de peticiones

## Conclusión


Pi-hole es una de las herramientas de administración de la red de casa más poderosas, sencillas de utilizar y útiles que existen. Como siempre, el tema tiene mucha más profundidad de lo que se trata aquí. Como nota adicional, se puede utilizar el servicio como servidor DHCP en caso de que no se pueda modificar el servidor DNS del router.

## DNS teniendo un dominio

Dependiendo del proveedor escogido, la compra de un dominio otorga la posibilidad de configurar todo lo referente a la DNS de manera manual, de manera que no solo se puede, obviamente, enlazar tu dominio con la IP del servidor que aloje tu página web, sino que puedes utilizar tu dominio y [subdividirlo](#) en la estructura ya mencionada.


En mi caso, adquirí mi dominio utilizando *Google Domains*, que me ofrece una personalización del sistema DNS del dominio muy completa.

[Servidores de nombres predeterminados \(Activa\)](#) [Servidores de nombres personalizados](#)

 Esta es la configuración de DNS activa. Los cambios se publicarán de inmediato, pero puede que tarden en aplicarse.

Registros de recursos [Exportar registros DNS](#)

Los registros de recursos indican los servicios que usa tu dominio, incluidos los servicios web y de correo electrónico. [Más información sobre los registros de recursos](#)

Registros personalizados  
mier.info/A, epicalendar.mier.info/CNAME y 3 más 

[Gestionar registros personalizados](#)

Nombre del host	Tipo	TTL	Datos
mier.info	A	1 hora	199.36.158.100
epicalendar.mier.info	CNAME	1 hora	cubed-taiga-zgvij7sar2igsysluo9durhm.herokudns.com.
mc.mier.info	CNAME	1 hora	
ssh.mier.info	A	1 hora	
www.mier.info	A	1 hora	199.36.158.100


Google Workspace  
mier.info/MX, mier.info/SPF y 2 más 

Figura 23. Panel de gestión de DNS de Google Domains

Utilizo este servicio DNS no solo para la propia página web, sino también para conectarme con distintos dispositivos, en combinación del sistema de DNS dinámica de *DuckDNS* establecido anteriormente y para acceder a otras páginas web usando el mismo dominio a través de otra plataforma de hosting.

Además, Google ofrece un servicio de DNS dinámica al estilo de los ya vistos en este informe, aunque sea un poco más complejo de instalar y utilizar.