



Servicios de Red

Configuración y Administración en Linux

↔ DHCP: Dynamic Host Configuration Protocol

ⓘ ¿Qué es DHCP?

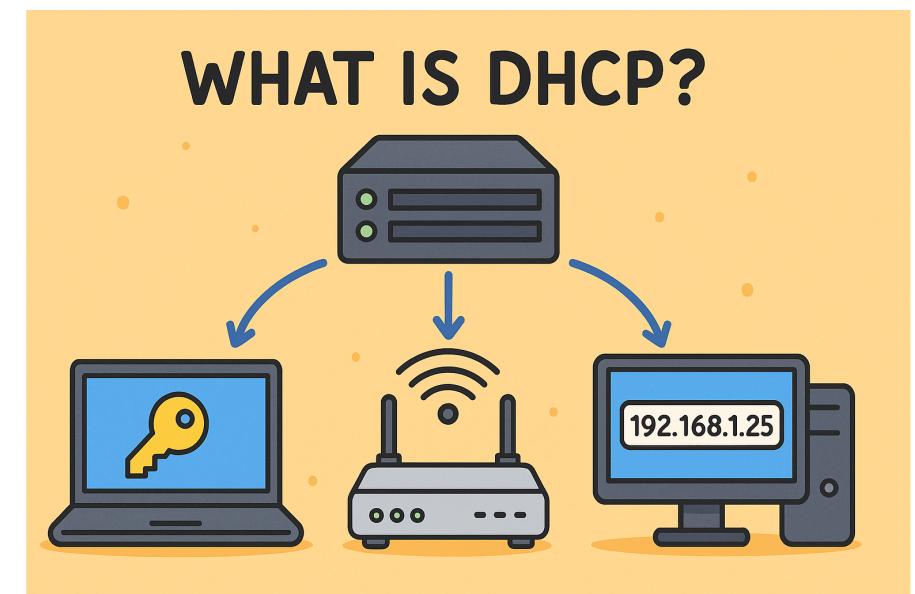
Protocolo que **asigna automáticamente** direcciones IP y otros parámetros de configuración de red a dispositivos.

Propósito

Configuración **automática y centralizada** de dispositivos en la red.

↔ Funcionamiento

Arquitectura **cliente-servidor** para gestionar y distribuir direcciones IP.



↔ Flujos DHCP: DORA

1

🔍 Discover

Cliente difusión buscando servidor DHCP

2

💡 Offer

Servidor ofrece dirección IP disponible

3

💰 Request

Cliente solicita la dirección ofrecida

4

✓ Acknowledge

Servidor confirma asignación de dirección

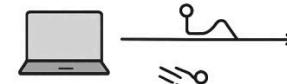
🔗 DHCP (Dynamic Host Configuration Protocol)

DHCP automáticamente asigna direcciones IP y configuraciones de red (máscara de subred, gateway, DNS) a los dispositivos.

- Usa UDP: Puerto 67 (servidor), Puerto 68 (cliente)
- Propósito: Asegura que cada dispositivo obtiene una dirección IP única sin configuración manual

⚙️ DHCP DORA Process

1. Discover



Client broadcasts request:
"Any DHCP server available?"

2. Offer



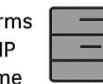
Server responds with an available IP + configuration (subnet mask, gateway, DNS, lease time).

3. Request



Client requests to use the offered IP.

4. Acknowledge



Server confirms and assigns IP with lease time
→ Client is now ready to communicate.

✓ In short: Discover → Offer → Request → Acknowledge

⚠ APIPA: Automatic Private IP Addressing

ℹ ¿Qué es APIPA?

Asignación **automática de direcciones IP** privadas cuando no hay servidor DHCP disponible.

▀ Rango de Direcciones

169.254.0.0/16 (169.254.0.1 - 169.254.255.254)

❗ Activación

Se activa cuando **falla el servidor DHCP** y no hay dirección configurada estáticamente.



Dirección APIPA

169.254.x.x

Sin servidor DHCP

💻 Cliente DHCP en Linux

⚙️ Implementación

Cliente **dhclient** para gestión de direcciones IP

- NetworkManager (GNOME/KDE)
- systemd-networkd (servidor)
- interfaces (Debian/Ubuntu)

☒ Verificación

- **dhclient -v** - Verbose mode
- **/var/lib/dhcp/dhclient.leases** - Historial
- **ip a | grep inet** - Dirección actual

☒ Terminal

```
$ dhclient -v  
$ cat /var/lib/dhcp/dhclient.leases  
$ ip a | grep inet
```

Output:

```
inet 192.168.1.100/24 brd 192.168.1.255 scope  
dynamic eth0  
Valid: 345s left
```

Resolución de Nombres: NSSWITCH

Archivo /etc/nsswitch.conf

Define **fuentes de información** para resolver nombres y direcciones

Orden de Búsqueda

- 1 files
- 2 dns
- 3 myhostname

/etc/hosts

Servidores DNS

Nombre local

Configuración

```
# /etc/nsswitch.conf

hosts:
    files dns myhostname

networks:
    files dns

services:
    files
```

≡, Fichero /etc/hosts

📄 Utilidad

- Archivo **local de resolución** de nombres
- No requiere **conexión DNS**
- Útil para **desarrollo y pruebas**



📝 Formato y Modificación

- **IP hostname alias**
- Se edita con **permisos de root**
- `sudo nano /etc/hosts`

📅 Configuración

```
# /etc/hosts  
127.0.0.1 localhost  
127.0.1.1 mi-pc  
192.168.1.10 servidor web
```

Formato:

`IP hostname [alias]`

Sistema DNS: Domain Name System

Funcionamiento

Traducción de **nombres a direcciones IP** y viceversa

Resolver

Cliente DNS que realiza **consultas recursivas** a otros servidores

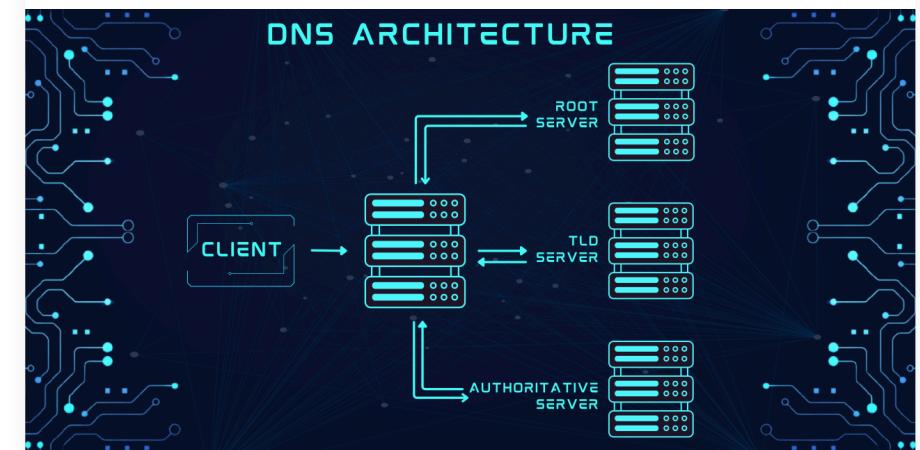
FQDN

Fully Qualified Domain Name

hostname + dominio completo

ejemplo: servidor.miempresa.com

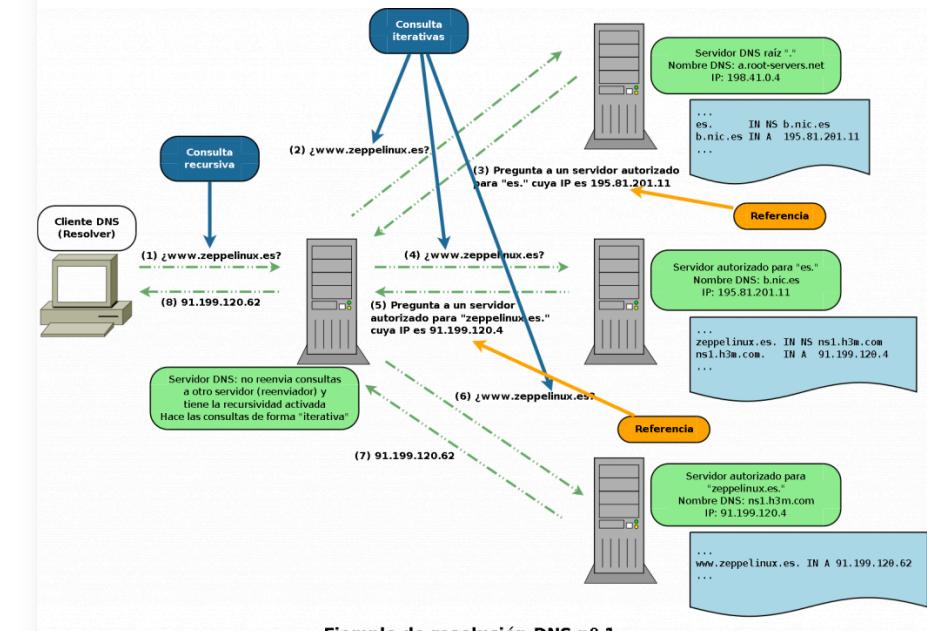
Estructura Jerárquica



Proceso de Resolución DNS

↔ Flujo de Resolución Iterativa

- 1 Cliente envía **consulta recursiva** al servidor DNS
- 2 Servidor DNS consulta **iterativa** al **servidor raíz (.)**
- 3 Servidor raíz referencia al **servidor del TLD** (ej. **.es**)
- 4 Servidor DNS consulta **iterativa** al servidor del TLD
- 5 Servidor del TLD referencia al **servidor del dominio** (ej. **zeppelinlinux.es**)
- 6 Servidor DNS consulta **iterativa** al servidor del dominio
- 7 Servidor del dominio responde con la **dirección IP**
- 8 Servidor DNS responde al **cliente** con la información



Ejemplo de resolución DNS nº 1

Configuración del Cliente Resolver

⚙ Archivo /etc/resolv.conf

- **nameserver** - IP del servidor DNS
- **search** - Dominios de búsqueda
- **domain** - Dominio local

Ⓜ Multicast DNS (mDNS)

- **Sin servidor DNS** centralizado en redes locales
- Consultas multicast: **224.0.0.251** (IPv4) / **ff02::fb** (IPv6)
- Utiliza dominio **.local**
- **systemd-resolved + avahi-daemon**
- Descubrimiento automático: **Bonjour/Avahi**

▣ Configuración

```
# /etc/resolv.conf  
  
nameserver 8.8.8.8  
nameserver 8.8.4.4  
  
search miempresa.com  
domain miempresa.com
```

systemd-resolved:

```
/etc/systemd/resolved.conf  
/run/systemd/resolve/resolv.conf
```

Multicast DNS:

Resolución .local en red local

↔ FTP y SSH: Protocolos de Transferencia

FTP - File Transfer Protocol

A Activo

Servidor inicia datos

↔ Puertos: 20 (datos) / 21 (control)

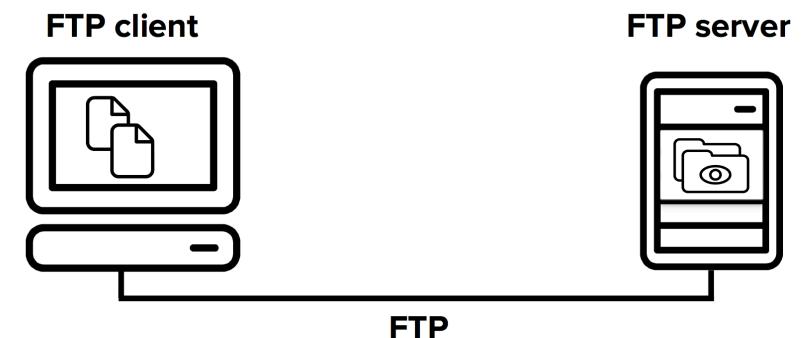
P Pasivo

Cliente inicia datos

SSH - Secure Shell

- Encriptación segura
- scp - Copia segura

- SFTP - Subsistema seguro
- SSHFS - FS remoto



Preguntas de Evaluación

1. ¿Cuál es el orden del flujo DHCP?

2. ¿Qué rango de direcciones usa APIPA?

3. ¿Qué comando verifica el servidor DHCP en Linux?

4. ¿Qué archivo define el orden de resolución de nombres?

5. ¿Cuál es el formato del fichero /etc/hosts?

6. ¿Qué significa FQDN en DNS?

7. ¿Cuál es la función de /etc/resolv.conf?

8. ¿Qué diferencia hay entre FTP activo y pasivo?