

SERVIDORES

101 — Fundamentos

PRESENTADOR

Juan David Hurtado G.

ROL	Gerente Orbiware
CONTACTO	<code>jdhurtado@orbiware.com</code>
HOY	Una introducción a los servidores que sostienen la red — qué hacen, cómo se hablan entre sí, y por qué importan.

EL RECORRIDO

Cuatro frentes, un mismo lenguaje.

Servidores a nivel de red. — DHCP, DNS

Servidores de aplicaciones. — Web, Correo, Acceso remoto (RDP).

Configuraciones de referencia. — nginx.

Preguntas, retos, qué probar.

0

ANTES DE EMPEZAR

Conceptos previos.

Dos definiciones que vamos a usar todo el rato. Si quedan claras desde el principio, el resto cae por su propio peso.

DOS TÉRMINOS DE BASE

Dirección MAC

DIRECCIÓN MAC

Identificador único de 48 bits asignado de fábrica a cada interfaz de red:

70:54:d2:c3:e2:bf

ARP

Protocolo que traduce direcciones IP en direcciones MAC dentro de una red local.

```
red ~ ifconfig
```

```
$ arp-scan --interface=en0 --localnet
192.168.78.100 70:54:d2:c3:e2:bf PEGATRON CORPORATION
192.168.78.233 58:9c:fc:10:3c:7f FreeBSD Foundation
192.168.78.243 30:a1:fa:2c:9b:8f HUAWEI TECHNOLOGIES CO.,LTD
192.168.78.249 10:59:32:3e:1f:d9 Roku, Inc
192.168.78.216 24:4b:03:ed:5f:2a Samsung Electronics Co.,Ltd
```

DOS TÉRMINOS DE BASE

Dirección IP y máscara de red.

DIRECCIÓN IP

Número único en la red que identifica a un equipo. En IPv4: **192.168.0.3**

MÁSCARA

Indica qué parte de la IP es la red:
255.255.255.0

```
red ~ ifconfig

$ ifconfig -a | grep -w "inet"
inet 192.168.0.3 /24      netmask 255.255.255.0
gateway 192.168.0.1 # 24 bits de red, 8 de host
# → 254 equipos en esta LAN
```

1

SERVIDOR A NIVEL DE RED

DHCP.

Dynamic Host Configuration Protocol — el servidor que reparte direcciones IP en la red, sin que nadie tenga que configurarlas a mano.

PASO 1 · DEFINIR TÉRMINOS

¿Qué es un protocolo?

Las reglas que gobiernan una comunicación. Tres componentes:

01 · QUÉ

Semántica

El significado de lo que se comunica.

02 · CÓMO

Sintaxis

La forma en que se expresa.

03 · CUÁNDO

Sincronización

Quién y cuándo transmite en la comunicación.

EN CRISTIANO

// Son las reglas que debo conocer, respetar y cumplir (yo, como computador) para comunicarme correctamente con otros computadores.

DHCP

Dynamic Host Configuration Protocol.

Un protocolo de red que entrega, de forma centralizada y automática, los datos que un equipo necesita para entrar a la red.

UN SOLO LUGAR

Administración centralizada de IPs.

Con un servidor DHCP en la red puedo, desde un solo lugar:

Saber **qué IP tiene** en este momento cada cliente conectado.

Saber **cuánto tiempo más** le queda esa IP antes de renovarla.

Cambiar la IP de uno o varios equipos sin tocarlos físicamente.

Reservar IPs por dirección MAC para equipos que siempre deben tener la misma.

CUATRO DATOS

Lo que entrega en cada solicitud.

01 · IP

192.168.0.42

Dirección única dentro de la red.

02 · MÁSCARA

/24

Qué tan grande es esta red.

03 · GATEWAY

.1

Puerta de enlace hacia afuera.

04 · DNS

.1

A quién preguntarle por nombres.

Resultado: el equipo se conecta a la red **sin intervención del usuario** . Lo mismo en una oficina, un café, un aeropuerto.

DHCP entrega muchos más parámetros (más de 200 opciones definidas). Vale la pena mencionar al menos: servidor NTP (hora), nombre de dominio (search domain), servidor WINS, y para netboot: boot file / TFTP server. Esto conecta con BOOTP/PXE.

2

SERVIDOR A NIVEL DE RED

DNS.

Domain Name System — la base de datos jerárquica que traduce nombres humanos en direcciones que las máquinas saben enrutar.

DNS

Domain Name System.

Una base de datos **distribuida** y **jerárquica** que asocia nombres de dominio con distintos tipos de información — sobre todo, direcciones IP.

Distribuida — ningún servidor tiene toda la información. Se reparte por niveles.

Jerárquica — los nombres se leen de derecha a izquierda. Cada punto es un escalón.

Asociativa — a un mismo nombre puede colgarle más de un tipo de registro (A, MX, TXT...).

OJO CON LA DISTINCIÓN

Nombre de dominio vs. nombre de equipo.

DOMINIO

google.com.

Un espacio dentro del árbol DNS. No identifica una máquina — identifica un **conjunto** de máquinas y registros.

vs

EQUIPO

www.google.com

El equipo **www** dentro del dominio google.com . Es lo que el navegador realmente busca.

TRES OPERACIONES

Lo que un servidor DNS sabe responder.

01 · DIRECTA

Resolución de nombres.

Dado un nombre completo, devolver su IP. Es la consulta más común — la que hace tu navegador.

02 · INVERSA

Resolución inversa.

Dada una IP, saber qué nombres responden con ella. Útil para auditoría y registros.

03 · MX

Resolución de correo.

Dado un dominio, saber por qué servidor se entrega su correo electrónico.

Muchos más tipos de registros importantes en la práctica: A (IPv4), AAAA (IPv6), CNAME (alias), NS (delegación), SOA (autoridad de zona), TXT (SPF, DKIM, verificación), SRV (servicios), CAA (autoridades de certificación).

EJEMPLO 01 · DIRECTA

Nombre → IP.

Dado el nombre completo de un host, devolver su dirección IP.

<code>www.google.com</code>	→	<code>142.251.155.119</code>
<code>admin.google.com</code>	→	<code>172.217.28.110</code>
<code>www.hacklaboriente.org</code>	→	<code>104.21.7.107</code> <code>172.67.130.31</code>

EJEMPLO 02 · INVERSA

IP → nombres.

Dada una dirección IP, conocer el nombre canónico asociado a ella (el PTR registrado por el dueño de la IP). Útil para auditoría, logs y antispam.

200.3.145.32 → u145-32.upb.edu.co.

```
red ~ dig
$ dig -x 200.3.145.32 +short
u145-32.upb.edu.co.
```

EJEMPLO 03 · MX

Dominio → servidor de correo.

Dado un dominio, obtener el servidor por el cual debe entregarse el correo electrónico de sus direcciones.

```
red ~ dig MX
```

```
$ dig MX hacklaboriente.org +short
```

```
27 route3.mx.cloudflare.net.
```

```
61 route2.mx.cloudflare.net.
```

```
81 route1.mx.cloudflare.net.
```

CASO DE USO

¿Qué pasa entonces cuando consulto en mi navegador web por www.hacklaboriente.org?

RECORRIDO DE UNA CONSULTA

Cuatro saltos, de la raíz al equipo final.

CLIENTE → Pregunta al resolver recursivo configurado (vía DHCP). Si está en caché, responde inmediatamente. Si no, inicia el recorrido raíz → TLD → dominio.

01 · RAÍZ

Consulta a los root-servers.

Pregunto: «¿quién sabe del TLD [.org](#)?»

>>

02 · TLD

Root devuelve el DNS con autoridad sobre [.org](#)

«Pregúntale a este servidor.»

>>

03 · DOMINIO

El TLD dirige al DNS con autoridad sobre [hacklaboriente.org](#)

«Tu respuesta vive ahí.»

>>

04 · RESULTADO

El NS del dominio entrega la IP final.

[www.hacklaboriente.org](#) → [104.21.7.107](#)

CASO LÍMITE

¿Y si en vez de www.hacklaboriente.org escribo sólo hacklaboriente.org?

El servidor DNS busca el registro **por defecto** del dominio — un apex record . Si no existe, la consulta falla. Por eso muchos sitios redirigen @ → www.

3

SERVIDOR DE APLICACIÓN

Servidor web.

El programa que habla HTTP y entrega hipertexto — la cosa que tu navegador interroga miles de veces por minuto sin que lo notes.

SERVIDOR WEB

Un programa que implementa HTTP.

Diseñado para transferir **hipertexto** : documentos con enlaces, formularios, recursos embebidos.

HTTP

Cómo se piden y se entregan las páginas.

HTML

La estructura de cada documento entregado.

Recursos

Imágenes, scripts, estilos — todo por el mismo canal.

```
curl ~ http  
  
$ curl -I https://hacklaboriente.org  
HTTP/2 200  
date: Sat, 23 May 2026 16:12:50 GMT  
content-type: text/html; charset=utf-8  
server: cloudflare
```

DOS CAMINOS CLÁSICOS

El protocolo es uno. Las implementaciones, varias.

MICROSOFT

IIS

Internet Information Services. Viene de fábrica con Windows Server. Integrado con Active Directory y el resto del stack de Microsoft.

OPEN SOURCE

Apache

Servidor histórico del mundo Unix/Linux — pero también disponible en Windows. Modular, configurable, sobrevivió tres décadas por algo.

HOY EN DÍA

Nginx, Caddy...

El ecosistema creció. Nginx domina alto-tráfico, Caddy se configura solo. El mismo HTTP, distintos pesos y filosofías.

UN SERVIDOR, MUCHOS SITIOS

Hosting virtual.

En un mismo servidor web pueden vivir varios sitios, cada uno con su propio nombre de dominio. El truco está en el header **Host:** de HTTP.

```
●●● nginx ~ vhosts
```

```
# mismo servidor, misma IP, distintos sitios  
server { server_name hacklaboriente.org; root /var/www/hacklab; }  
server { server_name elcarmen.hacklaboriente.com; root /var/www/elcarmen; }  
# el servidor lee `Host:` y decide a quién entregarle.
```

4

SERVIDOR DE APLICACIÓN

Servidor de correo.

El servicio asíncrono más antiguo de Internet. Sigue funcionando en buena parte porque a nadie le ha dado por reemplazarlo.

CORREO ELECTRÓNICO

Mensajería que no necesita que estés en línea.

Permite enviar mensajes a otros usuarios **sin importar en qué red estén**.

Tampoco importa si el destinatario está conectado, ni si tiene el equipo encendido.

Los mensajes viajan por una red privada o por Internet, y quedan **almacenados** hasta que alguien los pida.

TRES PROTOCOLOS

Para mandar, para bajar, para sincronizar.

PUERTO 25 / 587

SMTP

Simple Mail Transfer Protocol.

Para que dos servidores de correo **intercambien** mensajes. También para que tu cliente envíe lo que escribes.

PUERTO 110 / 995

POP3

Post Office Protocol.

Sirve para que tu cliente **descargue y borre** los mensajes guardados en el servidor y te los entregue.

PUERTO 143 / 993

IMAP

Internet Message Access Protocol.

Misma idea que POP, pero **sin descargar** : los mensajes viven en el servidor y se sincronizan.

CICLO CLIENTE · SERVIDOR POP

Cuatro pasos para recoger el correo.

01 · CLIENTE

Envía petición POP.

Tu cliente abre una conexión al servidor POP del dominio.



02 · SERVIDOR

Recibe petición y verifica ID.

El servidor autentica usuario y contraseña antes de soltar nada.



03 · SERVIDOR

Transmite los mensajes.

El servidor envía cabeceras y cuerpos al cliente.



04 · CLIENTE

Recibe y guarda local.

Tu cliente los almacena — y, según la configuración, los borra del servidor.

5

SERVIDOR DE APLICACIÓN

Terminal server.

Usar un computador a través de la red — como si estuvieras sentado frente a él. La forma más vieja, y todavía vigente, de trabajo remoto.

RDP

Remote Desktop Protocol.

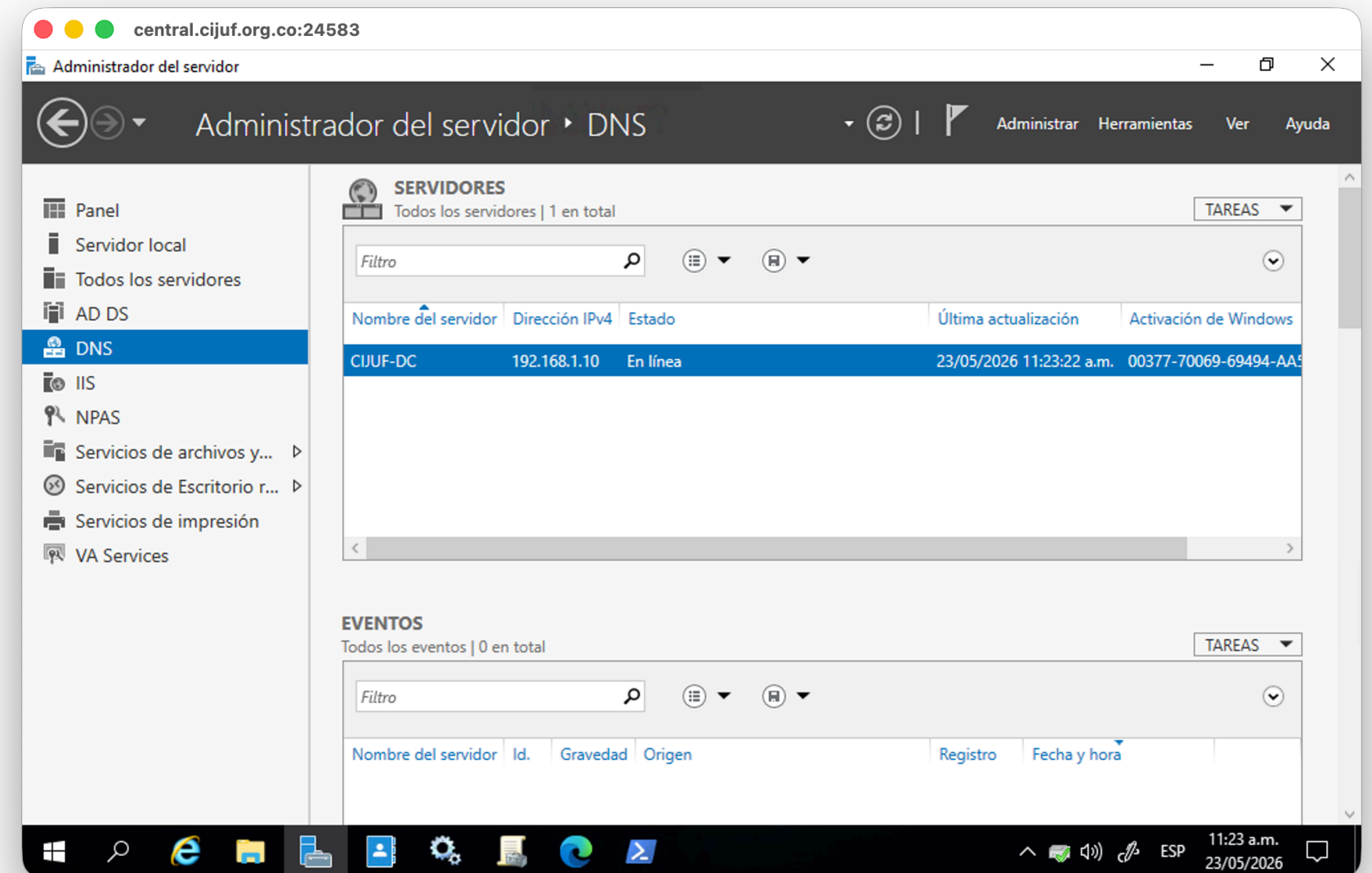
Permite a un usuario acceder a las aplicaciones y datos almacenados en otro equipo, vía red. Lo que pasa en la pantalla del servidor llega a tu pantalla — lo que tipeas, va al servidor.

Microsoft, desde Windows 2000.

Cliente de Terminal Services en todos los Windows.

VNC, NoMachine, Apache Guacamole.

El mismo problema resuelto desde otros frentes.



PARA SEGUIR

Recursos y bibliografía.

ESTÁNDARES

datatracker.ietf.org/doc/html/rfc2131

RFC 2131 · DHCP

DOCS

httpd.apache.org · nginx.org/en/docs

Apache + Nginx, documentación oficial

CORREO

datatracker.ietf.org/doc/html/rfc5321

RFC 5321 · SMTP

ESTÁNDARES

datatracker.ietf.org/doc/html/rfc1034

RFC 1034 · DNS — concepts and facilities

PRÁCTICA

dnschecker.org · mxtoolbox.com

Inspeccionar DNS y registros MX en vivo

COMUNIDAD

jdhurtado@orbiware.com

Escríbeme — Hacklab Oriente



HACKLAB is **True** Networking

PREGUNTAS · COMENTARIOS · IDEAS

JUAN DAVID HURTADO G. · jdhurtado@orbiware.com

HACKLAB ORIENTE · © 2026 ORBIWARE LLC

"HAC" < "LAB" → True