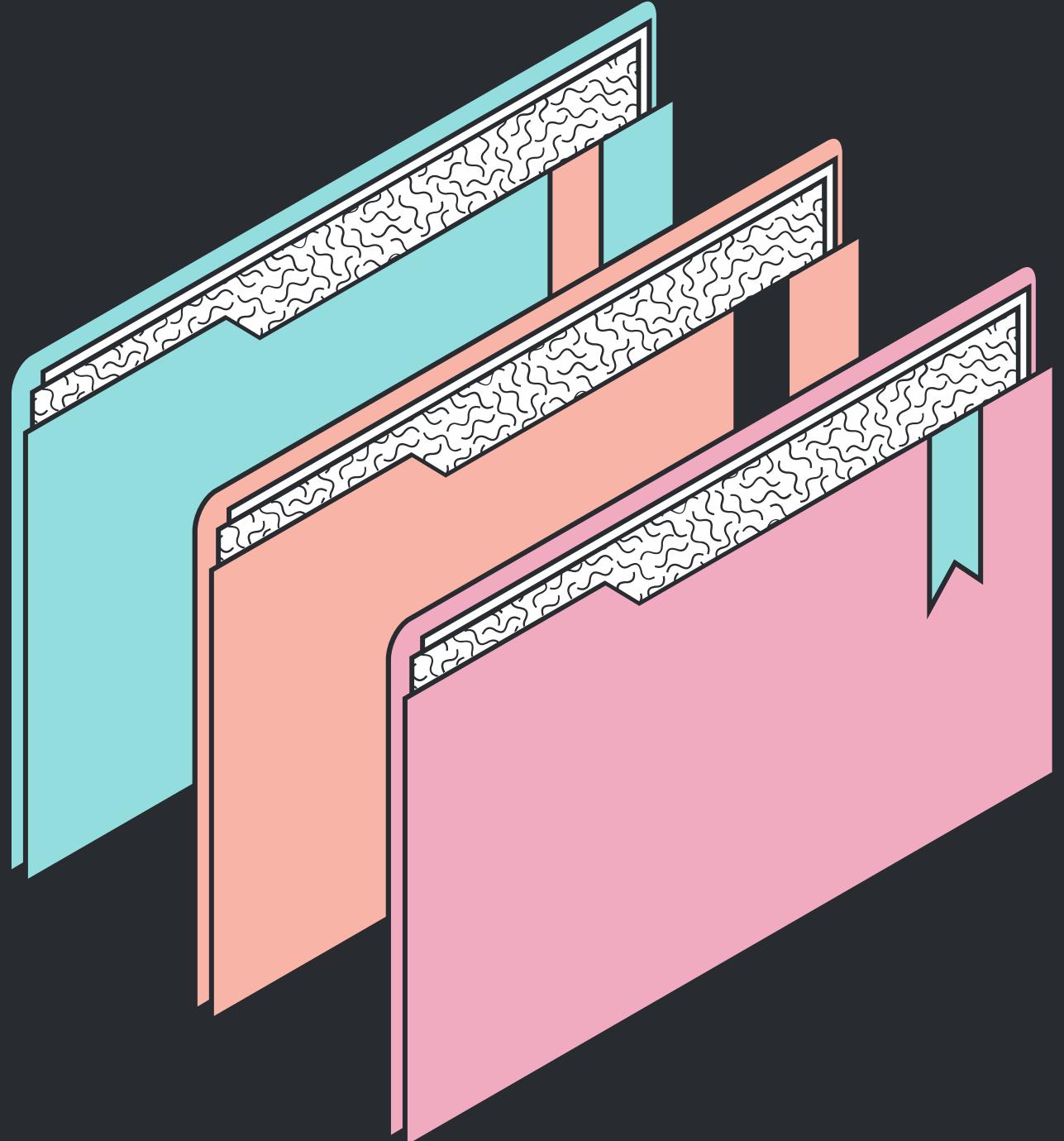


Protocolo DNS

Integrantes del equipo:
Flores Espinosa Edgar Jair
García Márquez Ángel Yahel
García Quiroz Gustavo Iván
Pedroza Villagomez Emir

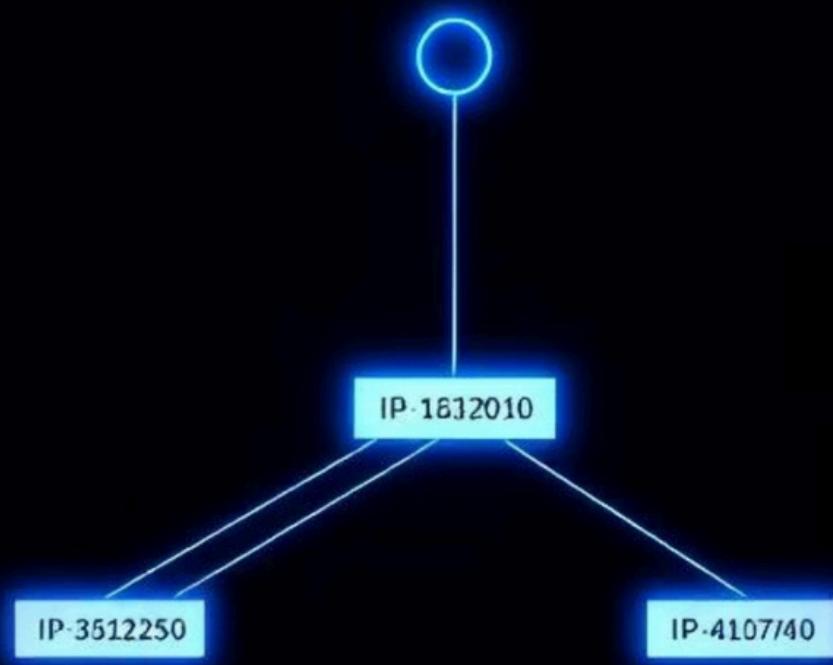


Contenidos



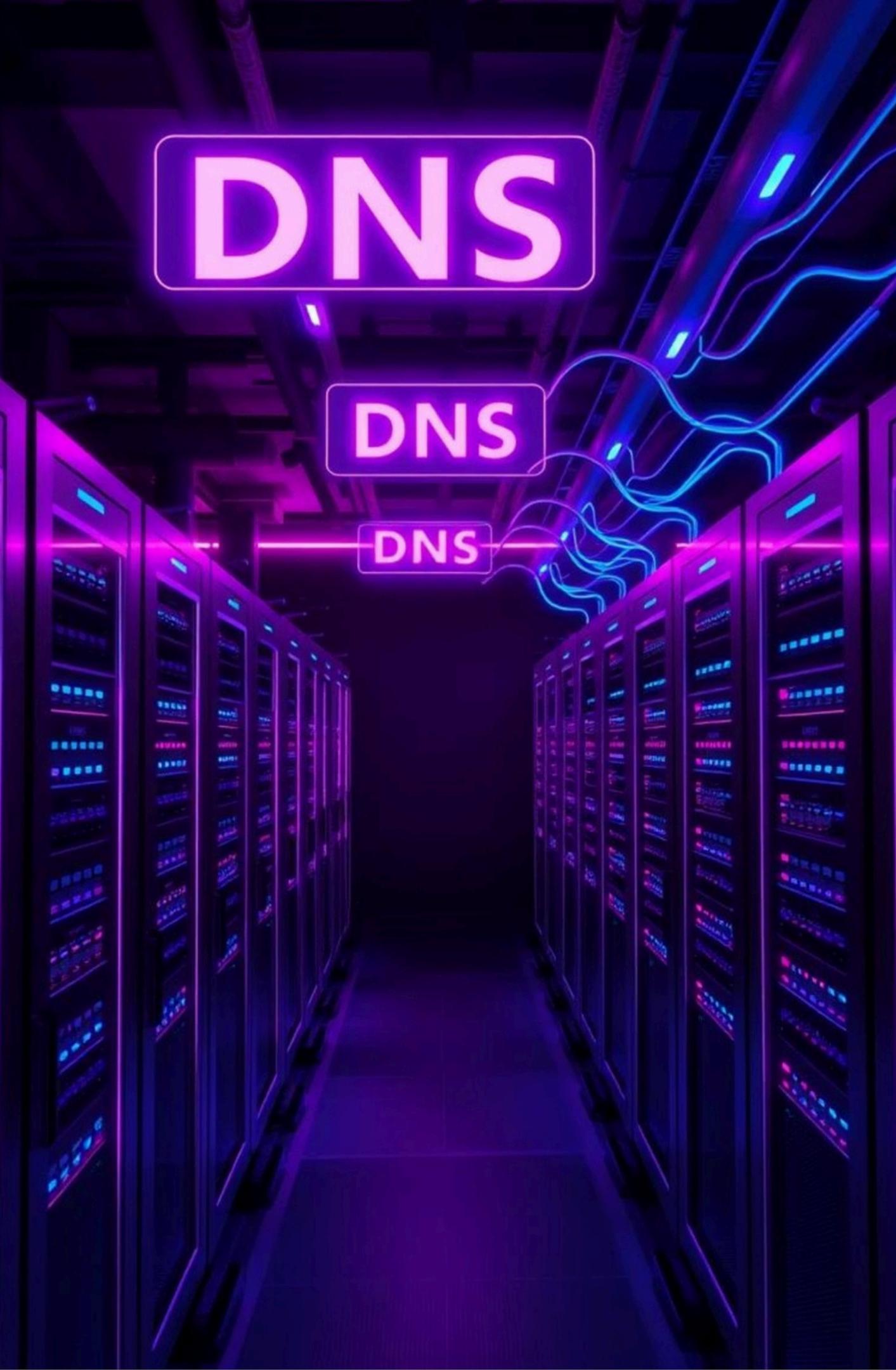
- Protocolo y características
- Versiones del protocolo
- RFC
- Protocolo de transporte del protocolo
- Puertos del protocolo
- Funcionamiento con diagrama.
- Implementación del cliente servidor.
- Tipo de petición de los cliente y servidor
- Simulación de cliente y servidor en java.

Protocolo DNS: Fundamentos y Funcionamiento



El protocolo DNS (Domain Name System) es fundamental para la comunicación en internet, permitiendo traducir nombres de dominio legibles por humanos, como `www.google.com`, en direcciones IP numéricas necesarias para conectar dispositivos. Esta traducción facilita la navegación, el envío de correos y otras funciones de red.

DNS funciona con una arquitectura jerárquica y distribuida que utiliza archivos de zona y registros específicos para manejar información clave, manteniendo una estructura descentralizada que otorga robustez y escalabilidad.



Propósito y Características Principales del DNS

Función Principal

Traducción de nombres de dominio en direcciones IP y viceversa, facilitando la localización de recursos de red.

Características

- Sistema distribuido y jerárquico
- Compatible con IPv4 e IPv6
- Soporta funciones avanzadas como resolución inversa y autenticación (SPF)
- Incluye caché local para mejorar la eficiencia

Registros DNS

Utiliza diferentes registros como A, AAAA, MX, CNAME, PTR y TXT para diversos propósitos dentro del sistema.

Versiones y Extensiones del Protocolo DNS

Extensiones Clave

- DNSSEC: Seguridad mediante criptografía para evitar suplantación
- SPF: Autorización para servidores de correo vía registros TXT
- EDNS0: Amplía capacidades del protocolo, soportando paquetes UDP más grandes

Protocolos de Privacidad

- DNS sobre HTTPS (DoH): Cifra consultas en tráfico HTTPS
- DNS sobre TLS (DoT): Cifra consultas mediante seguridad TLS



RFCs Fundamentales para la Implementación de DNS

1 RFC 7858 y RFC 8484

Definen DNS sobre TLS y DNS sobre HTTPS para cifrado seguro en las consultas DNS.

2 RFC 4033 a RFC 4035

Especifican DNSSEC, garantizando la autenticidad e integridad de los datos DNS.

3 RFC 6840 y RFC 6761

Regulan la autorización de certificados y reservan nombres de dominio para evitar conflictos.

Protocolos de Transporte y Puertos Utilizados por DNS

Transporte

DNS utiliza UDP para consultas rápidas, no orientado a conexión, y TCP para transferencias de zona y consultas que requieren más fiabilidad.

Puertos

- Puerto 53 para UDP y TCP estándar
- Puerto 853 para DNS sobre TLS
- Puerto 443 para DNS sobre HTTPS

Implementación Cliente-Servidor en DNS

Cliente DNS
Envía consultas para resolver nombres de dominio hacia un servidor DNS configurado.



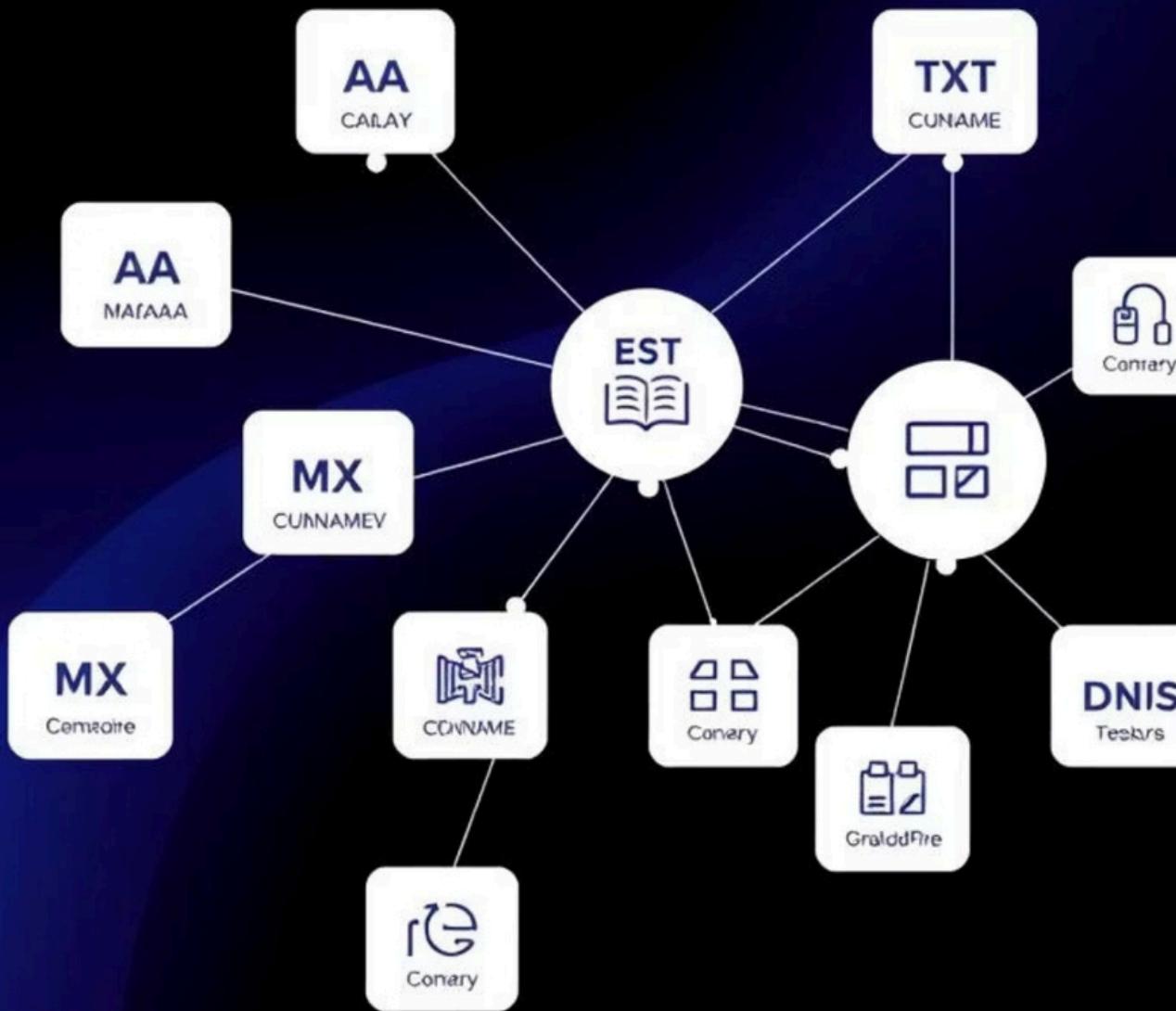
Servidor Recursivo

Gestiona la resolución consultando servidores autoritativos cuando es necesario.

Servidor Autoritativo

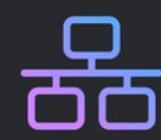
Proporciona respuestas definitivas basadas en registros de zona almacenados.

Tipos de Peticiones en DNS y Su Aplicación



Consulta Recursiva

El servidor DNS se encarga de resolver completamente la consulta para el cliente.



Consulta Iterativa

El servidor devuelve la mejor respuesta que pueda o referencias para continuar la búsqueda.

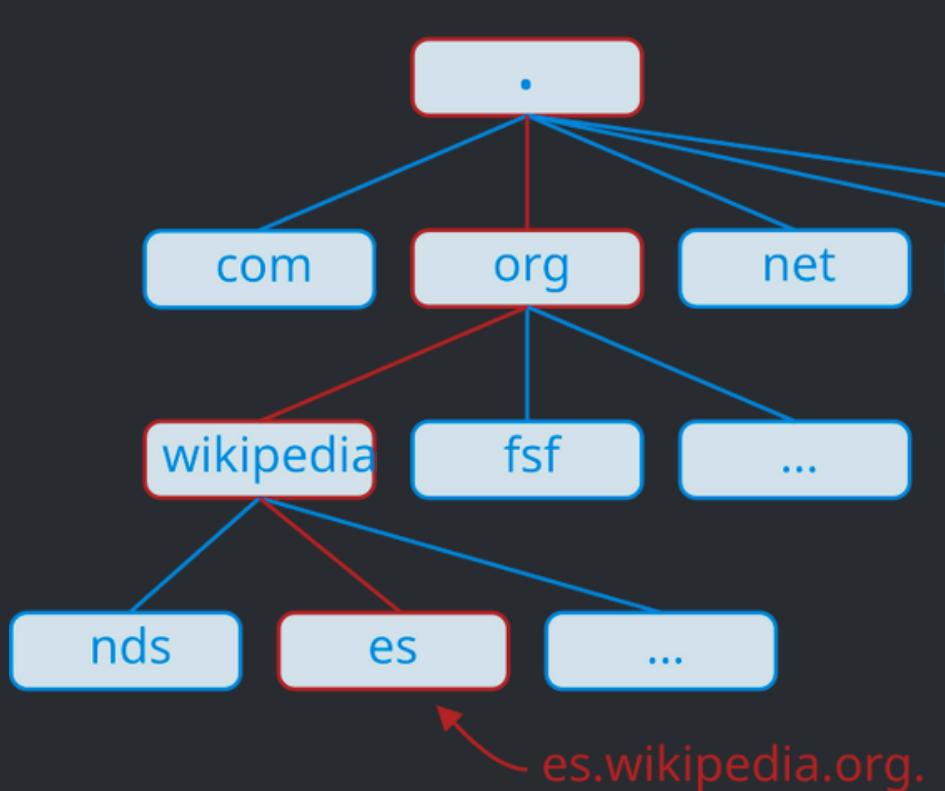


Consulta Inversa

Se busca resolver una dirección IP hacia un nombre de dominio.

Jerarquía DNS

Un nombre de dominio usualmente consiste en dos o más partes “etiquetas”, separadas por puntos cuando se las escribe en forma de texto.



- A la etiqueta ubicada más a la derecha se le llama dominio de nivel superior (Top Level Domain). Como “com” en www.ejemplo.com u “org” en es.wikipedia.org
- Cada etiqueta a la izquierda especifica una subdivisión o “subdominio”.
- La parte más a la izquierda del dominio suele expresar el nombre de la máquina (hostname) Por ejemplo, el dominio es.wikipedia.org tendría el nombre de la máquina «es».

Cuando se realiza una resolución DNS, se empieza por el dominio TLD y se sigue siguiendo una estructura jerárquica.

Registros de recursos

El Sistema de nombres de dominio especifica una base de datos de elementos de información para recursos de red. Los tipos de elementos de información se clasifican y organizan con una lista de tipos de registros de DNS: los registros de recursos (RR). Cada registro tiene un tipo (nombre y número), un tiempo de expiración (tiempo de vida), una clase, y datos específicos del tipo.

Campo	Descripción
NAME	Es el nombre de dominio completo del nodo en el árbol.
TYPE	Un número que indica qué tipo de información contiene este registro. El 1 (A) es para la Dirección IPv4.
CLASS	Especifica la "familia" o tipo de red del registro. La clase 1 (IN) es : Internet.
TTL	El tiempo en segundos que el registro es almacenado en caché.
RDLENGTH	Longitud del campo RDATA
RDATA	Los datos reales del registro (ejemplo: 142.250.191.14).

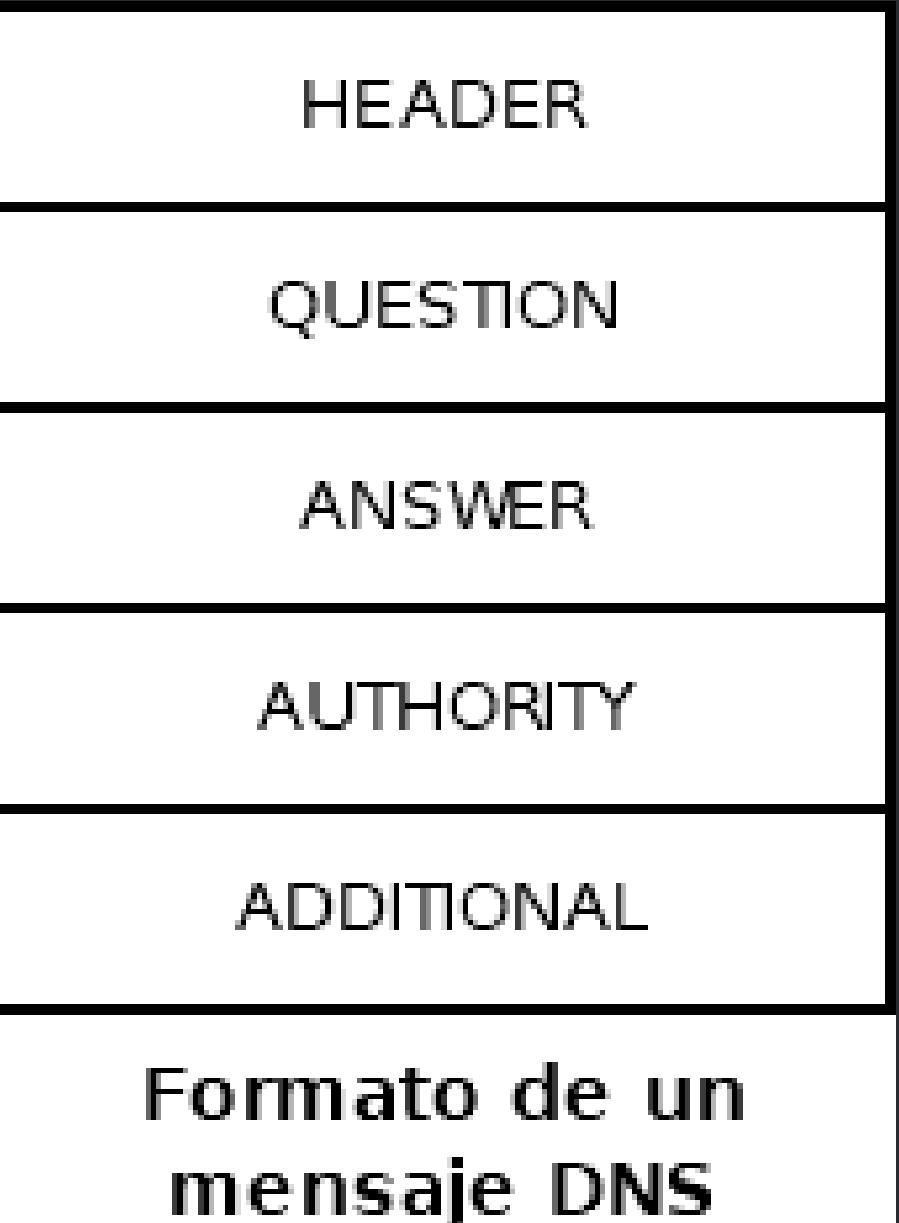


Estructura del Mensaje DNS

El protocolo DNS establece las normas y reglas en base a las cuales dialogan los clientes y servidores DNS. El formato de un mensaje DNS es el siguiente

Todos los mensajes tienen el mismo formato general:

- **HEADER:** Siempre presente en el mensaje. Especifica cuales son las secciones presentes, el tipo de mensaje y otros campos.
- **QUESTION:** Campos que definen una consulta a un servidor DNS.
- **ANSWER:** Registros de recursos que responden a la consulta.
- **AUTHORITY:** Registros de recursos que apuntan a los servidores DNS autoritativos.
- **ADDITIONAL:** Registros de recursos que contienen información adicional.





Estructura del Mensaje DNS

```
vboxuser@u22:~$ dig a wikipedia.org

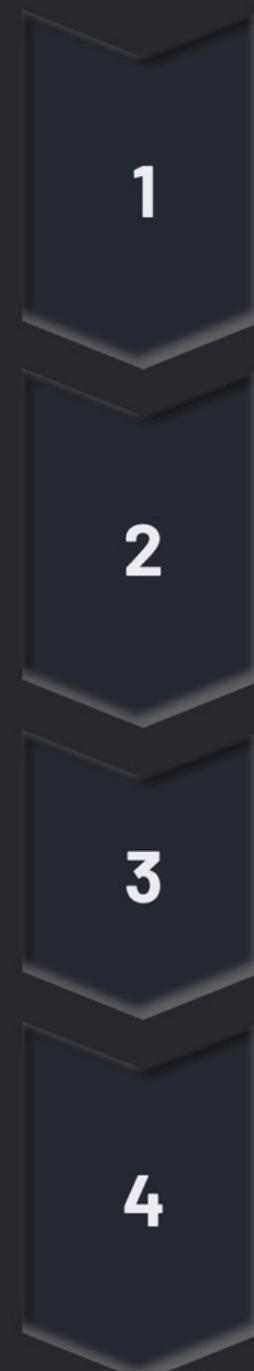
; <<>> DiG 9.18.30-0ubuntu0.22.04.2-Ubuntu <<>> a wikipedia.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 5584
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;wikipedia.org.           IN      A

;; ANSWER SECTION:
wikipedia.org.        162     IN      A      208.80.153.224

;; Query time: 10 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Thu May 29 07:16:11 CST 2025
;; MSG SIZE  rcvd: 58
```

Funcionamiento del Protocolo DNS: Diagrama y Procesos



Cliente realiza consulta

El cliente envía una petición DNS para resolver un nombre de dominio.

Servidor DNS recursivo

El servidor consulta a otros servidores si no tiene la información en caché.

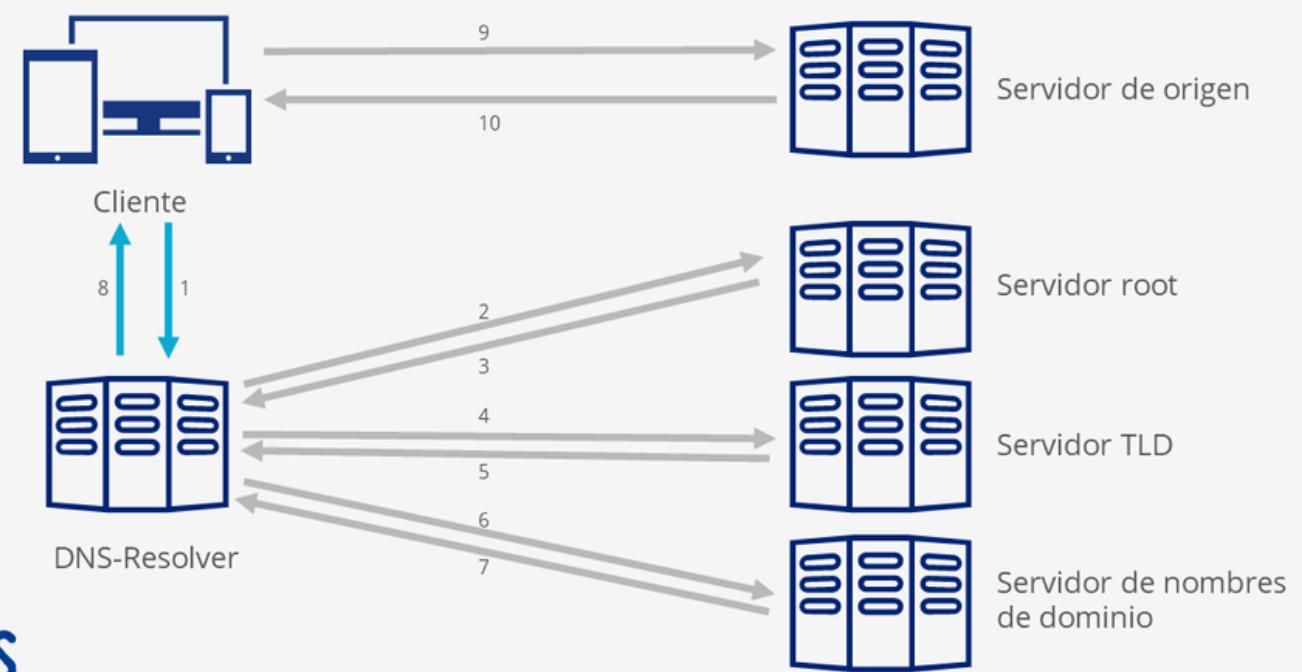
Servidor autoritativo

Responde con la dirección IP correspondiente al nombre solicitado.

Respuesta al cliente

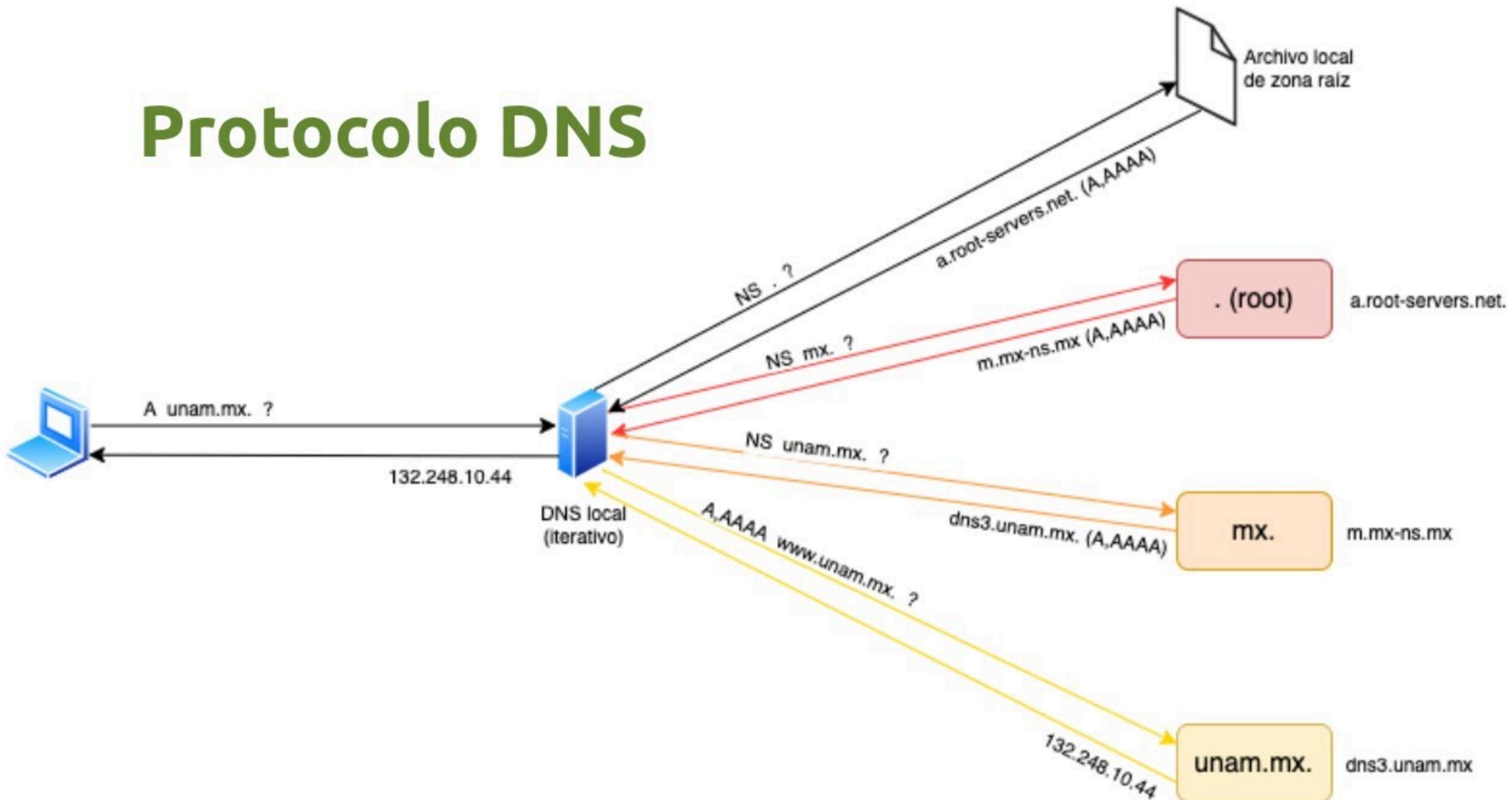
La dirección IP es enviada, permitiendo la conexión al recurso solicitado.

Cómo resuelve un servidor DNS una solicitud DNS



IONOS

Protocolo DNS



Implementación en Java



Servidor DNS

```
import java.net.*;
import java.io.*;
import java.util.HashMap;
import java.util.Map;

/**
 * Servidor DNS básico simulado.
 * Escucha peticiones de clientes, recibe un nombre de dominio y responde con una IP predefinida.
 */
public class DNSServer {
    Run | Debug
    public static void main(String[] args) throws IOException {
        int port = 5353; // Puerto donde el servidor escuchará (53 es el estándar DNS, 5353 es alternativo para pruebas)
        ServerSocket serverSocket = new ServerSocket(port); // Crea el socket del servidor
        System.out.println("DNS Server iniciado en el puerto " + port);

        // Tabla DNS simulada: asocia nombres de dominio con direcciones IP
        Map<String, String> dnsTable = new HashMap<>();
        dnsTable.put(key:"www.ejemplo.com", value:"192.168.1.10");
        dnsTable.put(key:"www.google.com", value:"8.8.8.8");
        dnsTable.put(key:"www.github.com", value:"140.82.112.3");
```

Servidor DNS

```
// Bucle infinito: el servidor siempre está esperando conexiones de clientes
while (true) {
    // Espera y acepta la conexión de un cliente
    Socket clientSocket = serverSocket.accept();

    // Prepara para leer y escribir datos con el cliente
    BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
    PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), autoFlush:true);

    // Recibe el nombre de dominio enviado por el cliente
    String domain = in.readLine();
    System.out.println("Consulta recibida: " + domain);

    // Busca la IP correspondiente al dominio en la tabla, o responde "Dominio no encontrado"
    String ip = dnsTable.getOrDefault(domain, defaultValue:"Dominio no encontrado");
    out.println(ip); // Envía la IP (o mensaje de error) al cliente

    clientSocket.close(); // Cierra la conexión con el cliente
}
```

Cliente DNS

```
import java.net.*;
import java.io.*;

/**
 * Cliente DNS básico.
 * Solicita al usuario un nombre de dominio, lo envía al servidor y muestra la IP recibida.
 */
public class DNSclient {
    Run | Debug
    public static void main(String[] args) throws IOException {
        String server = "localhost"; // Dirección del servidor (puede ser "localhost" para pruebas locales)
        int port = 5353; // Puerto del servidor (debe coincidir con el del servidor DNS)

        // Sigue el código de la imagen
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Ingrese el dominio a consultar: ");
        String domain = reader.readLine();
```

Cliente DNS

```
// Conecta con el servidor DNS usando sockets
Socket socket = new Socket(server, port);

// Prepara los flujos para enviar y recibir datos del servidor
PrintWriter out = new PrintWriter(socket.getOutputStream(), autoFlush:true);
BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

out.println(domain); // Envía el dominio al servidor

// Espera y recibe la respuesta del servidor (dirección IP o mensaje de error)
String response = in.readLine();
System.out.println("Respuesta del servidor DNS: " + response);

socket.close(); // Cierra la conexión
}
```

Resultados

Servidor

```
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Presentacion\Code> & 'C:\Program Files\Amazon Corretto\jdk17.0.14_7\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ivan\AppData\Roaming\Code\User\workspaceStorage\c4eec6c60ee63523882569782f1bffa\redhat.java\jdt_ws\Code_da88c00\bin' 'DNSServer'  
DNS Server iniciado en el puerto 5353  
Consulta recibida: www.github.com  
Consulta recibida: www.google.com  
Consulta recibida: www.ejemplo.com  
[]
```

Cliente

```
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Presentacion\Code> & 'C:\Program Files\Amazon Corretto\jdk17.0.14_7\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ivan\AppData\Roaming\Code\User\workspaceStorage\c4eec6c60ee63523882569782f1bffa\redhat.java\jdt_ws\Code_da88c00\bin' 'DNSClient'  
Ingrese el dominio a consultar: www.github.com  
Respuesta del servidor DNS: 140.82.112.3  
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Presentacion\Code> & 'C:\Program Files\Amazon Corretto\jdk17.0.14_7\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ivan\AppData\Roaming\Code\User\workspaceStorage\c4eec6c60ee63523882569782f1bffa\redhat.java\jdt_ws\Code_da88c00\bin' 'DNSClient'  
Ingrese el dominio a consultar: www.google.com  
Respuesta del servidor DNS: 8.8.8.8  
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Presentacion\Code> & 'C:\Program Files\Amazon Corretto\jdk17.0.14_7\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\ivan\AppData\Roaming\Code\User\workspaceStorage\c4eec6c60ee63523882569782f1bffa\redhat.java\jdt_ws\Code_da88c00\bin' 'DNSClient'  
Ingrese el dominio a consultar: www.ejemplo.com  
Respuesta del servidor DNS: 192.168.1.10  
PS C:\Users\ivan\Documents\GitHub\Aplicaciones-para-Comunicaciones-en-Red\Presentacion\Code> []
```

Referencias

- [1] P. Mockapetris, “Nombres de dominio – conceptos y facilidades,” RFC 1034, Grupo de Trabajo de Ingeniería de Internet, Nov. 1987. [En línea]. Disponible en: <https://www.rfc-editor.org/rfc/rfc1034>. [Consultado: 16 Jun. 2025].
- [2] P. Mockapetris, “Nombres de dominio – implementación y especificación,” RFC 1035, Grupo de Trabajo de Ingeniería de Internet, Nov. 1987. [En línea]. Disponible en: <https://www.rfc-editor.org/rfc/rfc1035>. [Consultado: 16 Jun. 2025].
- [3] A. S. Tanenbaum y D. J. Wetherall, *Redes de Computadoras*, 5^a ed. México: Pearson Educación, 2012.
- [4] R. Braden, “Requisitos para hosts de Internet – Aplicación y soporte,” RFC 1123, Grupo de Trabajo de Ingeniería de Internet, Oct. 1989. [En línea]. Disponible en: <https://www.rfc-editor.org/rfc/rfc1123>. [Consultado: 16 Jun. 2025].
- [5] P. Hoffman y K. Fujiwara, “Terminología DNS,” RFC 9499, Grupo de Trabajo de Ingeniería de Internet, Mar. 2024. [En línea]. Disponible en: <https://www.rfc-editor.org/rfc/rfc9499>. [Consultado: 16 Jun. 2025].

¡Gracias!