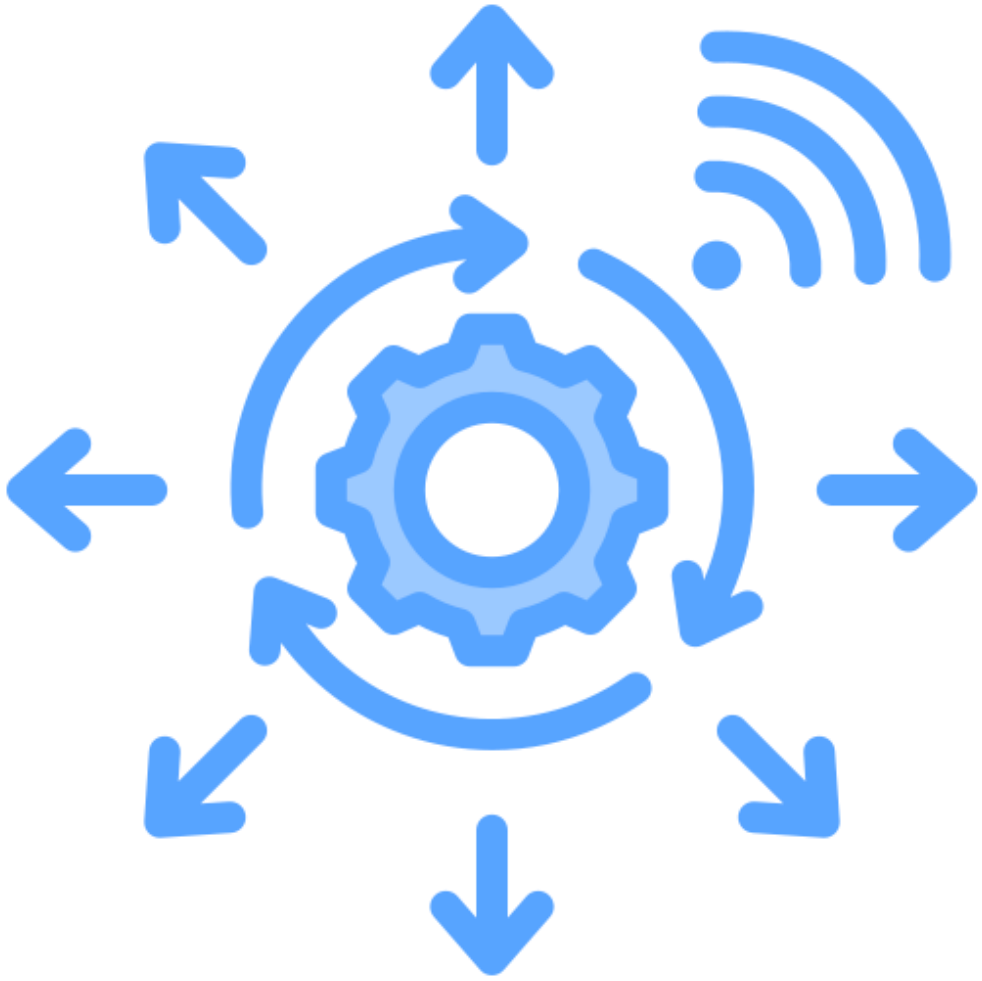
EFA MORATALAZ

*2º CFGS Desarrollo de Aplicaciones Web*

# *DESPLIEGUE DE APLICACIONES WEB*

*DANIEL GONZÁLEZ-CALERO JIMÉNEZ*

# UT1 – SERVICIOS DE RED IMPLICADOS EN EL DESPLIEGUE DE UNA APLICACIÓN WEB



EFA MORATALAZ

*2º CFGS Desarrollo de Aplicaciones Web*

***DESPLIEGUE DE APLICACIONES WEB***

# UT1 – SERVICIOS DE RED IMPLICADOS EN EL DESPLIEGUE DE UNA APLICACIÓN WEB

1. INTRODUCCIÓN
2. SISTEMA DE NOMBRE DE DOMINIO
3. ZONAS DE BÚSQUEDA, TIPOS DE REVIDORES DNS Y REGISTROS
4. FUNCIONAMIENTO DEL DNS Y TIPOS DE CONSULTA
5. INSTALACIÓN Y CONFIGURACIÓN DE UN DNS
6. SERVICIO DE DIRECTORIOS: CARACTERÍSTICAS Y FUNCIONALIDAD
7. ORGANIZACIÓN LDAP
8. ARCHIVOS BÁSICOS DE CONFIGURACIÓN Y USO

# INTRODUCCIÓN

1

En función del tipo de aplicación que tengamos, necesitamos de una serie de servicios en red para poder acceder a la correspondiente información que buscamos:

- **Servidor Físico:** Sistema informático dedicado que proporciona servicios, recursos o datos a otras computadoras a través de la red. Sus características principales son:

  - Se organizan en Racks
  - Están dentro de un centro de procesamiento de datos, donde le proporcionando conectividad, energía eléctrica y refrigeración constante.
  - Disponibilidad Continua: disponibles las 24 horas del día, los 7 días de la semana, con una alta fiabilidad, y con redundancia para evitar colapsos o caídas.
  - Recursos Compartidos: ofrecen recursos compartidos, como almacenamiento, potencia de cálculo, impresoras… de tal forma que varios usuarios o dispositivos accedan y usen estos recursos de manera simultánea.

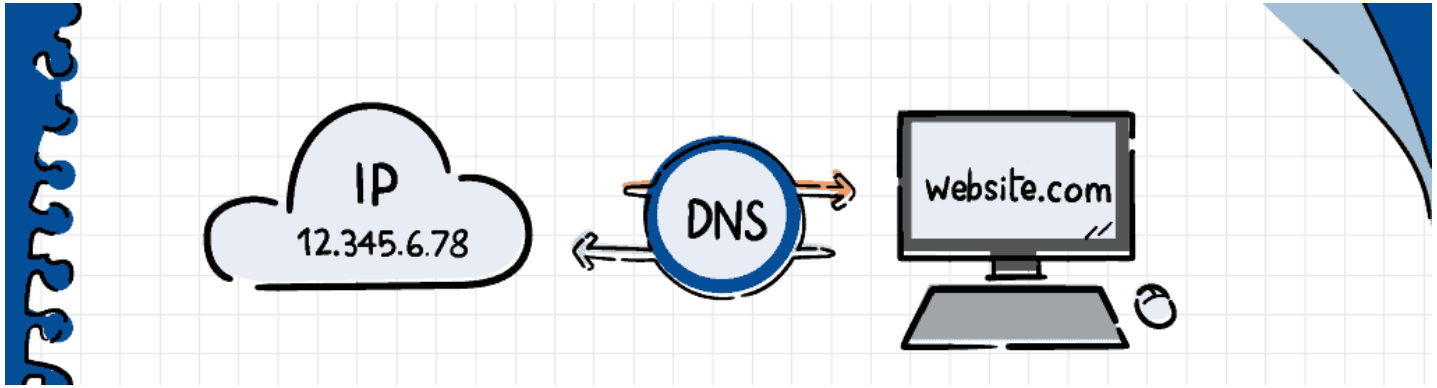
CPD más grande de España

- **Servidor WEB:** es un software que se instala en el servidor, y según el tipo, proporcionan la información o el contenido requerido por el usuario a través de la red. Algunos tipos de servidores web son Apache, HTTP Server, y Nginx.

Pero no solo hay servidores web que gestionan contenido http…, también podemos encontrar servidores de correo electrónico, servidores de bases de datos, de streaming, de juegos en línea, ftp…

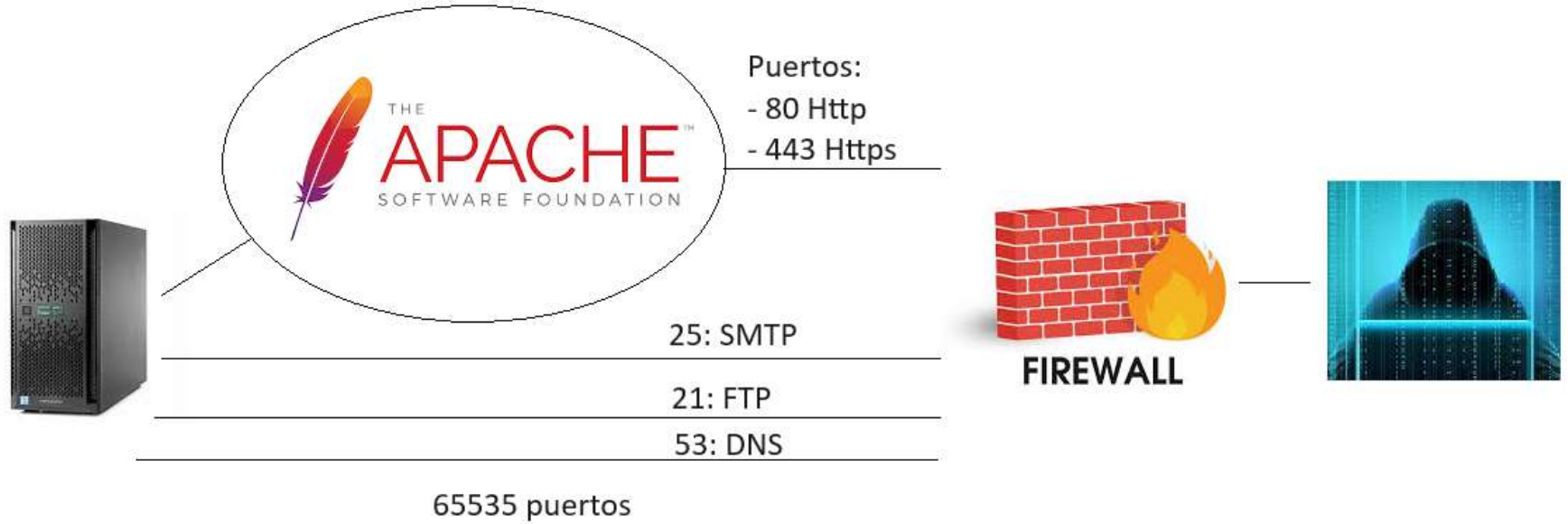

HTTP Server

- **Servidor DNS (Domain Name System :** Es un servidor que traduce los nombres de dominio en direcciones IP comprensibles por las computadoras. Antes de su existencia, los usuarios debían mantener una lista manual de conversiones de nombres a direcciones IP. Esto se volvió cada vez más impracticable a medida que la red crecía, lo que llevó al desarrollo del sistema de nombres de dominio y, en consecuencia, a la creación de servidores DNS




- **Directorio LDAP (Lightweight Directory Access Protocol):** A grandes rasgos, es un software que permite la validación de usuarios de forma centralizada y que permite el acceso a cualquier aplicación que este instalada en el SSOO.

- **Firewall:** Componente de seguridad informática diseñado para proteger una red o un sistema informático al controlar y filtrar el tráfico de datos que entra y sale de la red. Su función principal es actuar como una barrera entre una red privada o un dispositivo e Internet pública, permitiendo o bloqueando el tráfico en función de reglas predefinidas.




Hay muchos más componentes que intervienen, pero en este tema nos centraremos principalmente en los servidores DNS y LDAP.

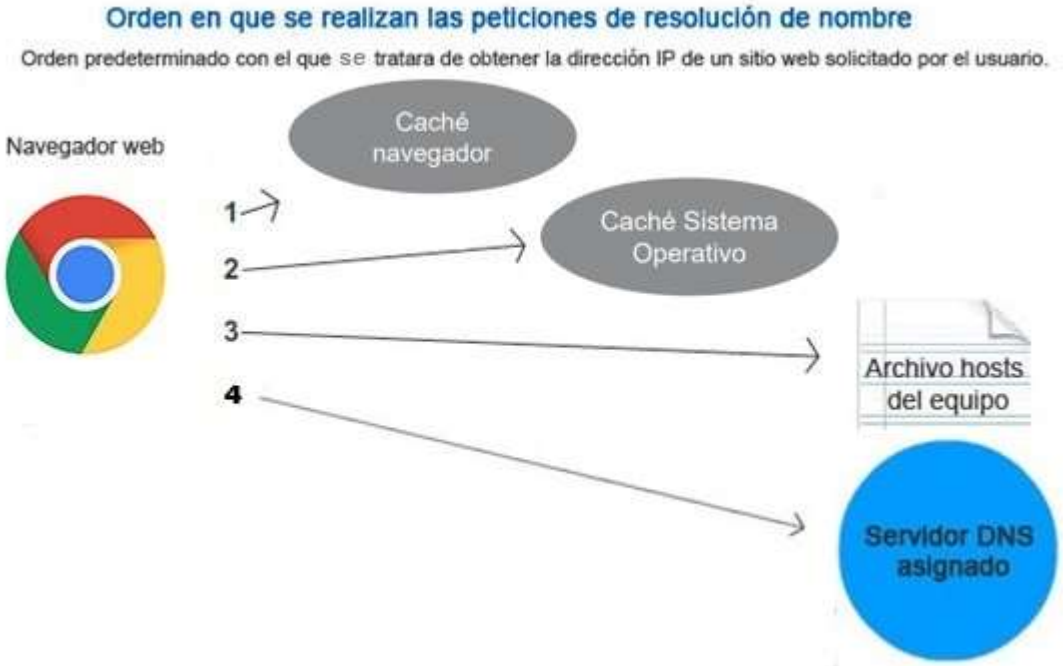
# SISTEMA DE NOMBRE DE DOMINIO

2

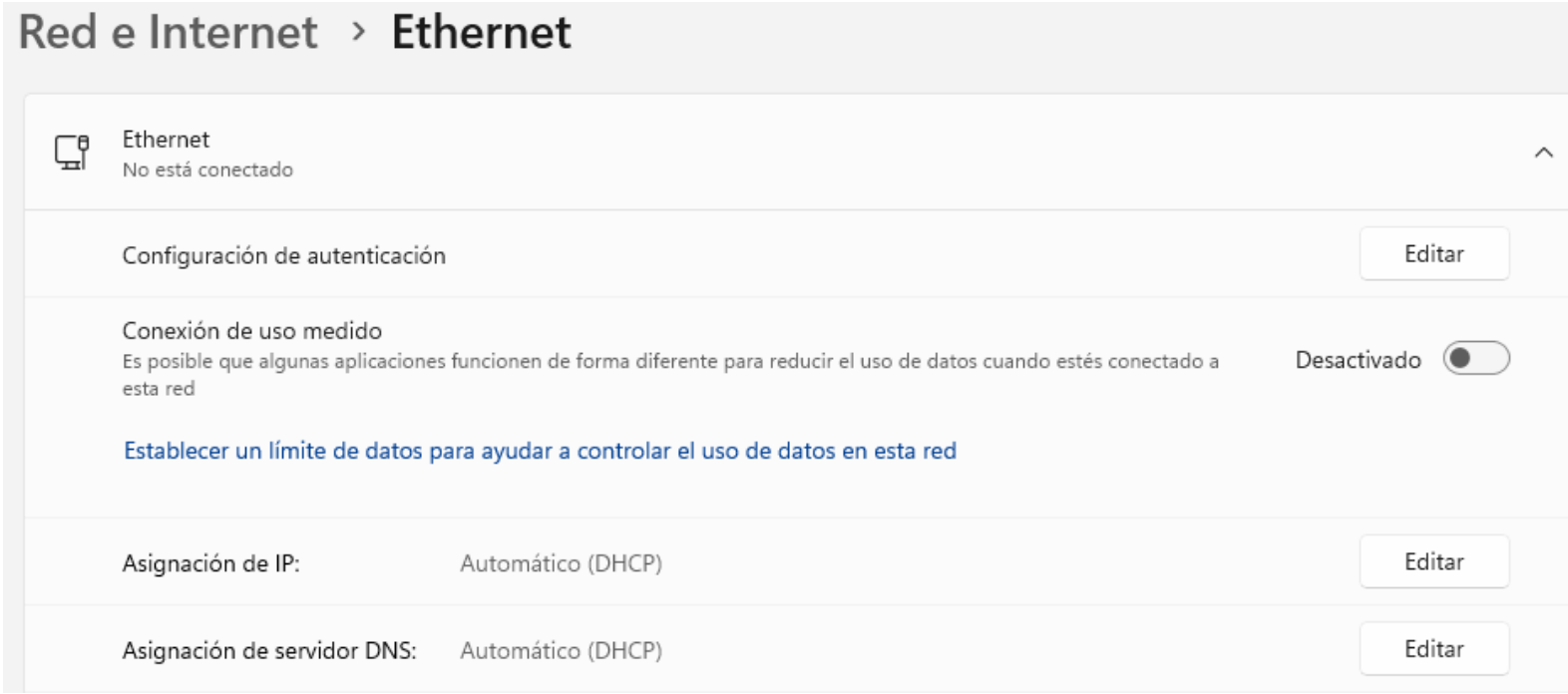
Es la forma en que los nombres de dominio se encuentran en internet. Estos se traducen en IP evitando así tener que recordarlas. El proceso y secuencia desde que un usuario solicita una dirección web y ésta es convertida en IP, es el siguiente:

1. El sistema busca la resolución de ese DNS de forma local en el archivo HOSTS: En **Linux** puedes encontrarlo en la ruta /etc/host y en **Windows** en la ruta C:\Windows\System32\drivers\etc.

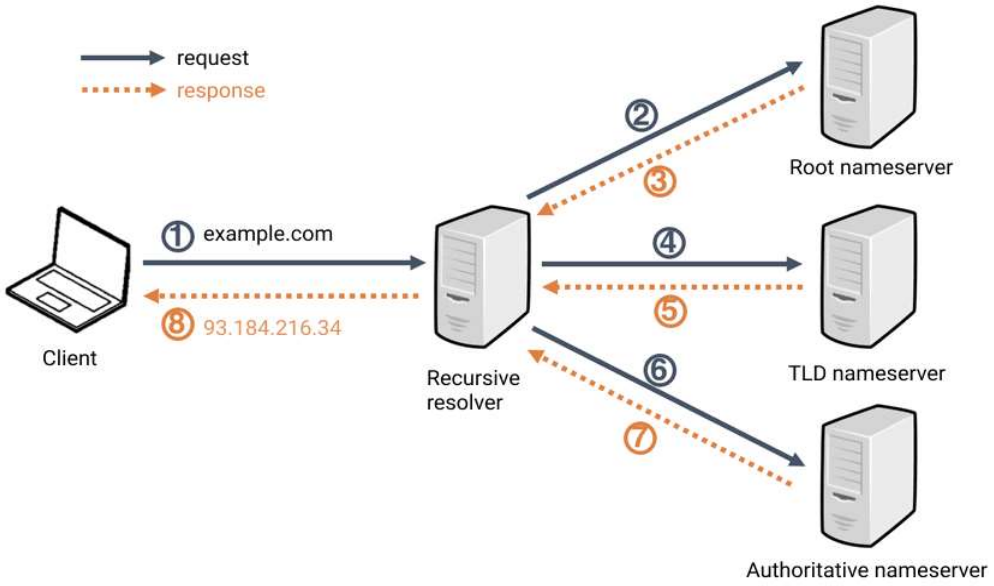
2. Si no hay ninguna entrada en “hosts”, buscará en la caché del navegador, y si no obtiene esa información, continuará buscando en la configuración de la tarjeta de red.



3. Si no encuentra el dominio solicitado por el usuario en el archivo hosts, pasará a buscarla en la configuración de la tarjeta de red del equipo. Si el DNS está asignado manualmente, recurrirá a la dirección del servidor DNS configurada. Si por el contrario la asignación del servidor DNS es automática, el sistema buscará el dominio solicitando a nuestro router esa información.



4. El **servidor DNS de tu proveedor** intenta averiguar cuál es la dirección IP del nombre de dominio mediante **una comparación con su base de datos**. Para recabar información al respecto, los servidores de dominio también utilizan resolvers. Muchas veces no ponemos los DNS correspondientes al nuestro ISP (Proveedor de Servicios de Internet), y ponemos los DNS de Google(8.8.8.8) o de Cloudflare (1.1.1.1)



5. Si aun así no se logra ningún resultado, el servidor DNS del proveedor se pone en contacto con un servidor raíz de nombres de dominio (root name server) y le reclama información adicional sobre el dominio de nivel superior (TLD) de la página web solicitada (la última parte de un nombre de dominio es la que representa al TLD; por ejemplo, .com o .es).

En el archivo de la zona raíz del DNS root server aparece información sobre el servidor de nombres de dominio de nivel superior (servidor de nombres TLD) encargado de proporcionar más información para un TLD determinado. La información que sea acorde con la petición realizada se transmite al servidor de nombres del proveedor de Internet. En el caso del nombre de dominio “www.ejemplo.es”, el root server remitiría al servidor de nombres de dominio de nivel superior de Red.es, que es la entidad responsable del registro de los nombres de dominio .es.

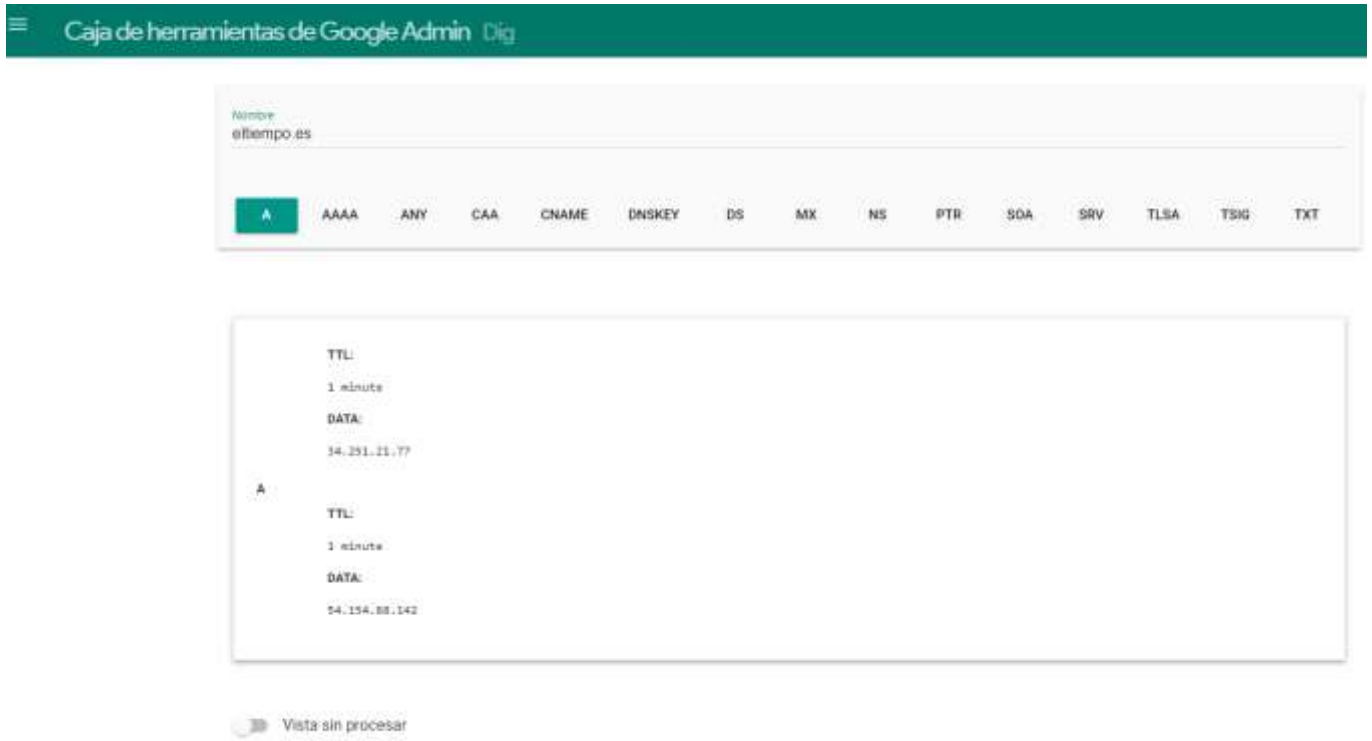
6. Tras ello, el servidor de nombres del proveedor de Internet emite una petición al servidor de nombres de dominio de nivel superior, pero en este caso tampoco obtiene una respuesta definitiva, sino que se reenvía: los servidores de nombres de dominio de nivel superior actúan meramente de transmisores. Estos informan a los servidores de nombres solicitantes acerca de los servidores DNS autoritativos en los que se deposita el nombre de dominio buscado.

7. En este paso, el servidor de nombres del proveedor se dirige al servidor de nombres autoritativo responsable del nombre de dominio y obtiene finalmente la dirección IP deseada.

8. En último lugar, el servidor de nombres del proveedor transmite la dirección IP al servidor DNS de tu router, el cual la entrega a tu resolver local, desde donde se transfiere a tu browser, de tal forma que puede solicitar, cargar y mostrar la página web.

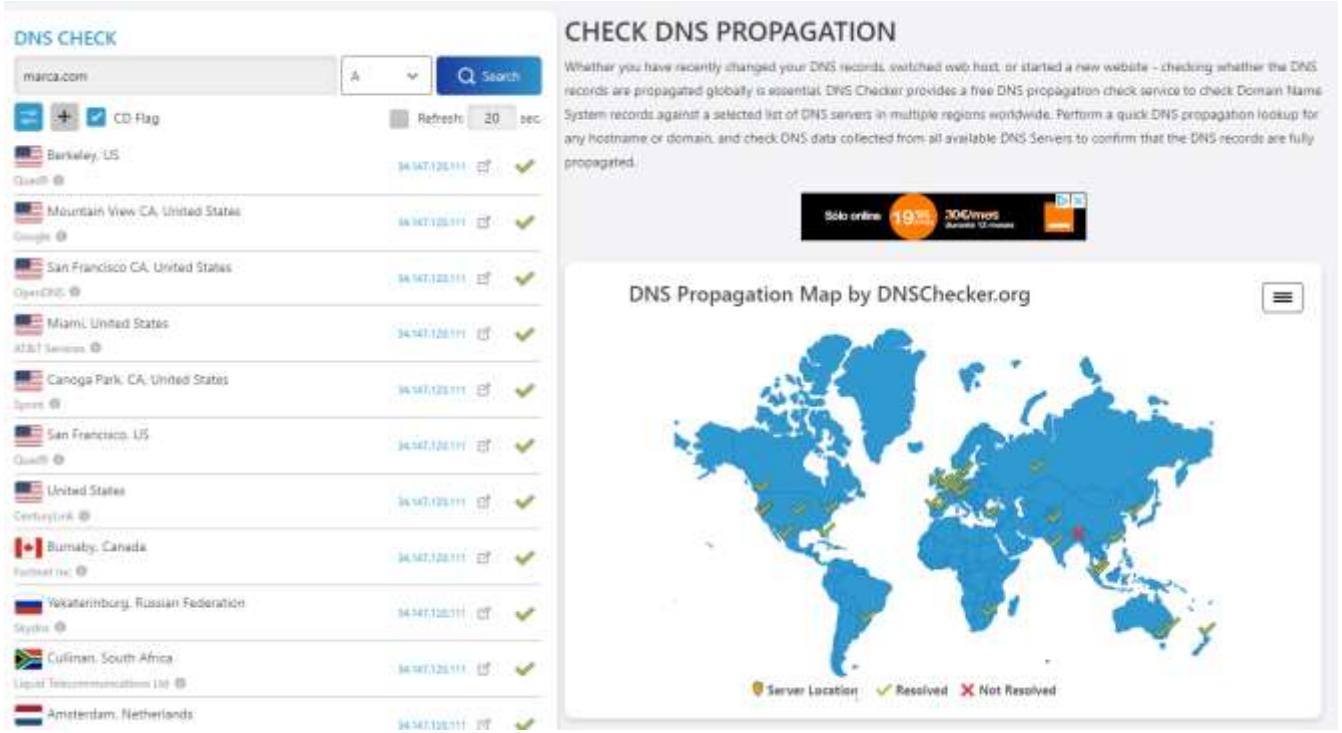
Es importante mencionar que todo este proceso de resolución DNS puede llevar un tiempo variable, y la velocidad de carga de una página web puede verse afectada si la resolución DNS demora demasiado. Además, la información en caché juega un papel crucial en la velocidad de navegación, ya que evita la necesidad de realizar consultas DNS completas en cada solicitud de un sitio web.

***Ejercicio 1:*** Acceder a la web https://toolbox.googleapps.com/apps/dig/#A/, y conseguir la dirección ip de una web que elijáis. (Si no devuelve un valor IP correctamente, es por que los servidores de esa página web no están configurados para que puedan acceder por IP, tendréis que probar con otro dominio.)



***Ejercicio 2:*** Acceder a “cmd” en Windows y con el comando nslookup, realizar búsquedas tanto de IP como de dominios, y compararlos con los resultados obtenidos anteriormente. ¿Qué conclusiones podemos sacar?

***Ejercicio 3:*** Acceder a la web https://dnschecker.org/, elegir 5 dominios distintos y buscar dónde se encuentran los servidores DNS que contienen la información en sus registros para poder resolver la consulta y transformar esa dirección en una IP. Compararlos entre ellos y sacar conclusiones.



***Ejercicio 4:*** Buscar el archivo público que contiene absolutamente toda la información sobre el Sistema de Nombres de Dominio: RFC 1034

1. ESTADO DE ESTE MEMORÁNDUM

Este RFC es una introducción al Sistema de Nombres de Dominio (DNS), y no tiene en cuenta muchos detalles que pueden encontrarse en el RFC-1035 "Nombres de Dominio - Implementación y Especificaciones". El RFC-1035 asume que el lector está familiarizado con los conceptos descritos en este memorándum (RFC-1034).

El protocolo oficial está compuesto por un subconjunto de funciones DNS y tipos de datos DNS. También incluye consultas estándar, sus respuestas y la mayoría de los formatos de datos de las clases de Internet, (direcciones de host).

De todas formas, la intención del sistema de dominios es escalable. Los investigadores contínuamente proponen, implementan y experimentan con nuevos tipos de datos, tipos de consultas, clases, funciones, etc. Normalmente los componentes del protocolo oficial no suelen cambiar y trabajan en servicio de producción. Los componentes experimentales son extensiones más allá del protocolo oficial. Las características experimentales u obsoletas están claramente indicadas en estos RFC's, y tal información ha de ser utilizada con precaución.

El lector debe tener especial cuidado de no depender de los valores que aparecen en los ejemplos, debido a que su propósito es fundamentalmente pedagógico. La distribución de este memorándum es ilimitada.

2. INTRODUCCIÓN

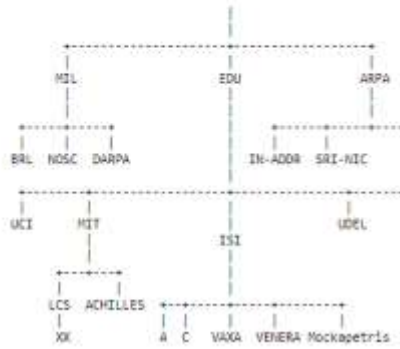
Este RFC es una introducción de los estilos de los nombres de dominio, su uso para correo de Internet, soporte de direcciones de host, y los protocolos y servidores utilizados para implementar instalaciones de nombres de dominio.

2.1 La historia de los nombres de dominio

El ímpetu por el desarrollo del sistema de dominios ha crecido en Internet:

3.4. Ejemplo de espacio de nombres

La siguiente figura muestra una parte del espacio de nombres de dominio actual, y es utilizado en varios ejemplos de este RFC. Nótese que el árbol es un subconjunto más pequeño del actual espacio de nombres.




En este ejemplo, el dominio raíz tiene tres subdominios inmediatos: MIL, EDU, y ARPA. El dominio LCS.MIT.EDU tiene un subdominio inmediato llamado XX.LCS.MIT.EDU. Todas las hojas también son dominios.

3.5. Sintaxis de nombres preferida

Las especificaciones DNS intentan ser lo más generales posibles en cuanto a las reglas para construir nombres de dominio.

La idea es que pueda expresarse cualquier objeto existente como un nombre de dominio con los mínimos cambios. Sin embargo, cuando asignamos

Mockapetris [Página 11]

RFC 1034 Conceptos e instalación de dominios Noviembre 1987

un nombre de dominio a un objeto, el usuario prudente seleccionará un nombre que cumpla las reglas del sistema de dominios y las del objeto, ya estén publicadas o implícitas por los programas existentes.

Por ejemplo, cuando nombramos a un dominio mail, el usuario puede cumplir las reglas de este memorándum y las del RFC-822. Cuando creamos un nuevo nombre de host, se seguirán las antiguas reglas para HOSTS.TXT. Esto evita problemas con el software antiguo a la hora de utilizar nombres de dominio.

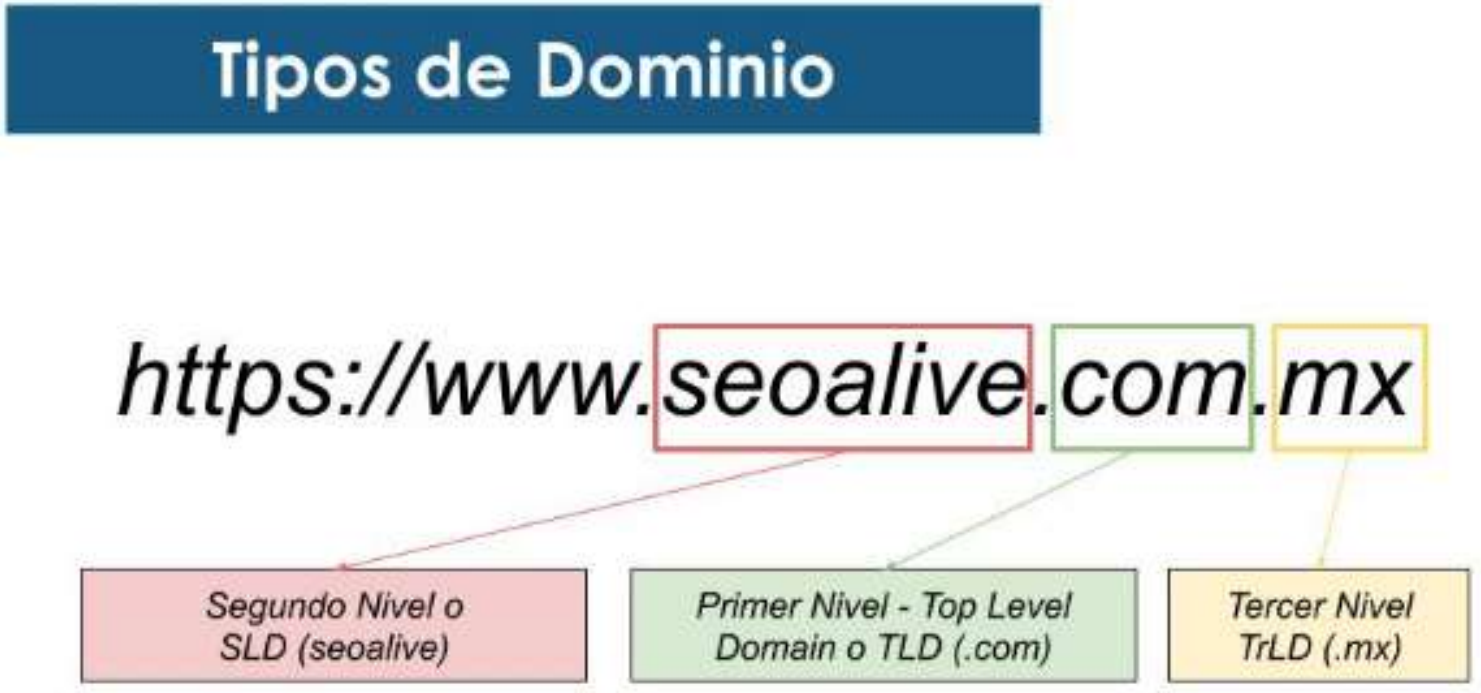
***Ejercicio 5:*** Dividir la clase en 3 grupos y elegir cada uno un dominio distinto. Utilizando el comando “tracert” en cmd, rastrear la ruta de la solicitud DNS desde el ordenador hasta el servidor web del nombre de dominio elegido.

- Comentar los resultados, intentando identificar los distintos saltos intermedios y los tiempos de respuesta, así como cualquier información interesante que obtengáis.
- Analizar los patrones comunes obtenidos entre los 3 grupos.

## *Ejercicio 6*

- *Obtener la IP de un dominio al que se pueda acceder también con su propia IP y no devuelva fallo de acceso. (ej. "el tiempo.es")*
- *Crear copia de seguridad del archivo "hosts" del equipo a utilizar.*
- *Modificar el archivo hosts, de tal forma que cuando pongamos en nuestro navegador lo siguiente www.alumno.com, nos redirija automáticamente al dominio que hemos elegido anteriormente.*

- **Nombre de dominio:** Se define como la dirección de una empresa, organización, asociación o grupos de personas en internet. Permite que su información, sus productos o servicios sean accesibles a todo el mundo a través de la red. El objetivo de esto es poder recordar dichos sitios de forma ágil y sencilla. Están divididos en los siguientes niveles:



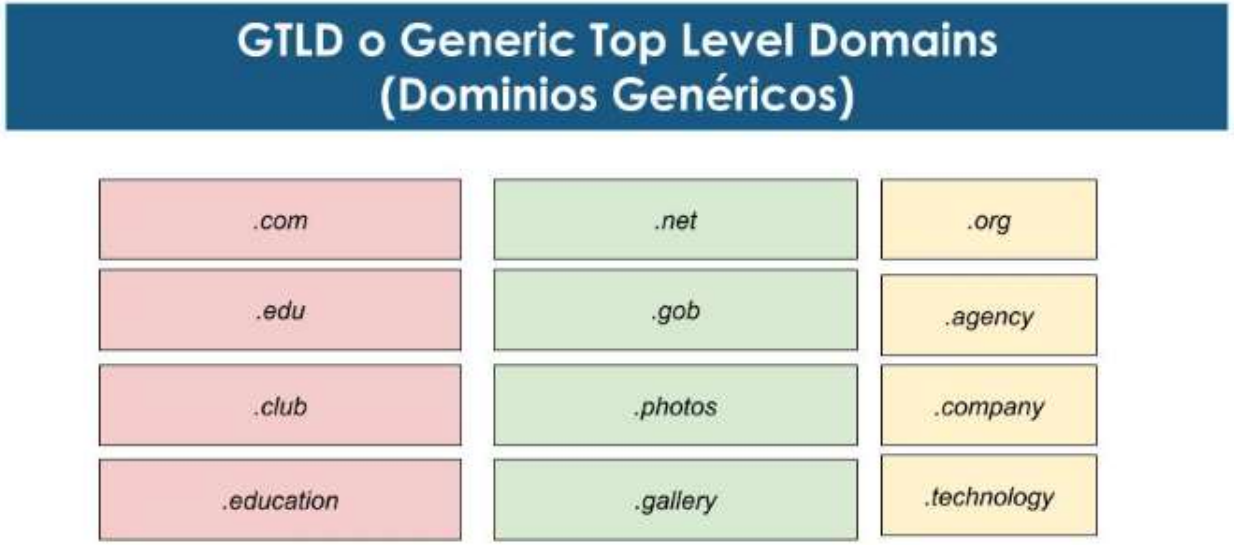
- **De primer nivel:** TLD (Top-Level Domain), son dominios con mayor jerarquía según el Sistema de Nombres de Dominio (DNS). Están Regulados por la Corporación de Internet para la Asignación de Nombres y Números (ICANN). Los dominios no están sujetos a ninguna comprobación. Identifican recursos en Internet como sitios web, servidores de correo electrónico... Cada TLD tiene sus propias reglas de registro y restricciones, y algunos como .gov y .edu, están restringidos y solo pueden ser registrados por entidades específicas que cumplan con ciertos requisitos.

**.com:** Más común, asociado principalmente con web comerciales o empresas.
**.org:** En su origen, usado para organizaciones sin ánimo de lucro.
**.gov**: Webs gubernamentales de los Estados Unidos.
**.edu:** Instituciones educativas, universidades, institutos, colegios….
**.net**: ISP, alojamiento web, bases de datos o herramientas colaborativas…


Los **TLD** tienen dos sub-divisiones:

Dominios Genéricos (GTLD o Generic Top Level Domains).
Dominio de nivel superior geográfico (ccTLD o Country Code Top Level).

**Dominios Genéricos (GTLD):** identifican recursos en línea de manera más general. Además, la ICANN ha abierto nuevas extensiones más específicas y temáticas, como .app, .blog, .guru, .music, .travel…. Ampliando así las opciones de personalización en la elección de nombres de dominio.




**Dominio de nivel superior geográfico (ccTLD):** Identifican sitios web que tienen una conexión o presencia geográfica en el país o región correspondiente.


- **De segundo nivel (*Second Level Domain):*** se encuentra directamente a la izquierda del dominio de primer nivel, siendo usualmente el nombre de la marca, organización o persona en la que se enfoca el sitio. Por ejemplo, en la dirección URL www.seoalive.com, «seoalive» sería el SLD, actuando como identificador en lo que a usuario se refiere.

- **De tercer nivel (*Third Level Domain)***: Mezclan un dominio genérico o gTLD con un dominio específico de país o ccTLD. Por ejemplo, la terminación del dominio ***.com.mx*** nos indicaría que se trata de un dominio genérico pero específico para usuarios de México. Otros ejemplos sería org.es, .edu.es, .gob.es.

- **Ejemplo de niveles de dominio:**

***https://www.subdominio.dominio.org***

| Parte | Descripción |
|---|---|
| https:// | Protocolo de hipertexto que permite visualizar cualquier página web en un navegador de forma correcta y |
| subdominio | Subdominio o parte opcional de una dirección web que precede al dominio principal. |
| dominio | Es el SLD, cuyo dominio padre en nuestro caso es org, que normalmente identifica de forma coherente al nombre de nuestra organización, empresa… |
| org | Es el dominio de primer nivel que identifica a organizaciones y el padre de todos los subdominios |
| ¿www? | Buscar el significado de las www y comentarlo en clase. |

# ZONAS DE BÚSQUEDA, TIPOS DE DNS Y REGISTROS

3

**Registros DNS:** Un registro DNS (Domain Name System) es una entrada en una base de datos de un servidor DNS que almacena información asociada con un nombre de dominio específico. Estos registros permiten la traducción de nombres de dominio legibles por humanos en direcciones IP numéricas que las computadoras y los servidores utilizan para identificarse y comunicarse en Internet. Los registros DNS se utilizan para diversas finalidades, y cada uno tiene un propósito específico, como veremos a continuación.

# Cuadro de registros DNS (I)

| Tipo de registro | **Descripción** | Sintaxis |
|---|---|---|
| A (Address) | Traduce nombres de dominio en direcciones IP | New.com A xx.xx.xx.xx |
| PTR (Pointer) | Traduce direcciones IP en nombres de dominio | 3.0.0.20in-addr.arpa PTR host.new.om |
| MX (Mail Exchanger) | Asocia un nombre de dominio a un servidor de correo. El número 10 indica preferencia a menor número, mayor preferencia | Nex.com MX 10 correo.new.com |
| CNAME | Es un alias que se le asigna a un host que tiene una dirección IP | Alias.new.com CNAME nombre.new.com |
| NS | Define los servidores principales de un dominio, al menos debe haber uno. | New.com IN NS servidor1.new.com |
| SOA | Es el primer registro de la zona; solo puede haber uno configurado. Especifica el servidor DNS primario del dominio. Pieza clave del archivo de zona. | Es un tipo de registro que se especifica información del DNS. Los campos se definen más adelante. |

# Cuadro de registros DNS (II)

| Tipo de registro | **Descripción** | Sintaxis |
|---|---|---|
| TXT | Ofrece información adicional a un dominio. También se usa como almacenamiento en claves de cifrado. | New.com TXT “Información adicional” |
| SPF | Es un registro de tipo texto que se crea en la zona directa del DNS. Se usa principalmente para evitar la suplantación de identidad | New.com IN SPF “v=spf1a:Exchange.new.com –all” |

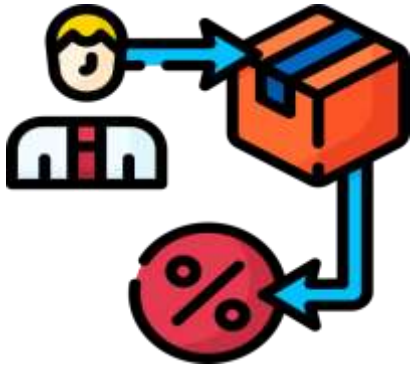
**Zonas de búsqueda:** es una porción de un espacio de nombres de dominio que está administrada por un servidor DNS específico. En otras palabras, es un conjunto de registros DNS relacionados que pertenecen a un dominio o subdominio específico y que un servidor DNS tiene la responsabilidad de administrar y responder cuando se realizan consultas de resolución de nombres de dominio para esa zona en particular.

Cada zona de búsqueda DNS contiene registros DNS que permiten la resolución de nombres de dominio en direcciones IP o en otros recursos de red. Estos registros se detallan en las siguientes diapositivas.

Las zonas de búsqueda DNS se utilizan para organizar y administrar de manera lógica y jerárquica los nombres de dominio y sus correspondientes registros DNS. Cada dominio (como "ejemplo.com") puede tener su propia zona de búsqueda, y se pueden crear subdominios (como "blog.ejemplo.com") con sus propias zonas de búsqueda dentro de la zona principal.

Existen 2 tipos de zona de búsqueda:

Directa



Indirecta

- **Zona de Búsqueda Directa (Forward Lookup Zone):** En una zona de búsqueda directa, los servidores DNS resuelven nombres de dominio (como www.ejemplo.com) en direcciones IP correspondientes (por ejemplo, 192.168.1.1). Esto es lo que sucede comúnmente cuando ingresas una URL en tu navegador web, y el servidor DNS traduce el nombre del sitio web en la dirección IP necesaria para establecer la conexión.

***Ejemplo***: Si tienes un servidor web con la dirección IP 192.168.1.100 y le asignas el nombre de dominio "www.misitio.com", la zona de búsqueda directa contendrá una entrada que relaciona "www.misitio.com" con la dirección IP 192.168.1.100.

- **Zonas de búsqueda indirecta o inversa:** Los registros que se definen en esta zona permiten obtener un nombre de un dominio a partir de una dirección IP. Es útil sobre todo en la solución de problemas de red y en la identificación de hosts.

  **Ejemplo:** Si deseas conocer el nombre de dominio asociado con la dirección IP 192.168.1.100, la zona de búsqueda inversa contendrá una entrada que relaciona 192.168.1.100 con "www.misitio.com".

# Las zonas de búsqueda además, pueden ser de 2 tipos:

Master

Slave

- **Master (**Master DNS Zone**):** Es un tipo de configuración de zona de búsqueda que crea sus propios registros sin copiarlos de otros servidores DNS. Se caracteriza por:

---

**- Autoridad Total:** Tiene la capacidad de crear, modificar y eliminar registros en la zona de búsqueda sin necesidad de consultar a otros servidores DNS.

- **Actualización y Mantenimiento:** Responsable de mantener y actualizar los registros en la zona de búsqueda, lo que supone agregar nuevos registros, actualizar los existentes y eliminar los obsoletos.

- **Registro SOA:** En una zona de búsqueda maestra, se configura un registro SOA (Start of Authority) que define la autoridad de la zona y proporciona información sobre la frecuencia de actualización de la zona y otros detalles importantes.

- **Transferencia de Zona (Zone Transfer):** El MDZ configura la transferencia de zona para permitir que otros servidores DNS secundarios (o esclavos) obtengan una copia actualizada de la zona de búsqueda. Los servidores secundarios copian los registros de la zona de búsqueda maestra para proporcionar redundancia y distribución de carga.

- **Administración Centralizada:** Cualquier cambio realizado en el servidor maestro se propagará a los servidores secundarios mediante transferencias de zona.

- **Slave** (Slave DNS Zone)**:** Es un tipo de configuración en un servidor DNS que obtiene copias actualizadas de los registros DNS de una zona de búsqueda maestra (Master Zone) desde otro servidor DNS que actúa como el servidor maestro (Master Server) para esa zona específica. En esencia, el servidor esclavo replica la información de la zona de búsqueda maestra para proporcionar redundancia y distribución de carga, ofreciendo respaldo en caso de fallo y reducción de la carga de los DNS principales. Sus características principales son:

  **- Copia de Datos:** El servidor DNS esclavo no tiene autoridad primaria sobre la zona de búsqueda. Su función principal es obtener una copia actualizada de la información de la zona de búsqueda desde el servidor DNS maestro.
  **- Transferencia de Zona:** La replicación de datos entre el servidor maestro y el servidor esclavo se realiza mediante un proceso llamado "transferencia de zona“. El servidor esclavo solicita periódicamente los cambios a la zona de búsqueda maestra y actualiza su copia local de la zona según sea necesario.
  - **Mejora la redundancia y Disponibilidad:** Si el Servidor maestro falla, el esclavo tiene una copia local de los registros y puede seguir respondiendo a las consultas DNS.
  - **Distribución de Carga:** Al tener múltiples servidores esclavos, la carga de las consultas de DNS se distribuye entre ellos, lo que mejora el rendimiento y la capacidad de respuesta.
  - **Configuración:** Se establece el Maestro desde donde se obtendrán los datos de la zona, y se configura para escuchar las actualizaciones de dicho servidor.

- **Tipos de servidores DNS:** Al igual que los tipos de zona existen 3 tipos de servidores DNS.

Servidor primario

Servidor secundario

Servidor caché

- **Servidor primario:** Son servidores que guardan la información relacionada con las zonas de las que son autorizados. Sus archivos son de lectura y escritura. El administrador es el encargado de añadir, modificar o eliminar nombres de dominio.

- **Servidor secundario:** Es un servidor que no tienen los propios archivos de zona, sino que están transferidos al 2º o 3ºer nivel jerárquico. Estos actúan cual el servidor principal no puede resolver la petición. Los datos de DNS se guardan de forma temporal en caché para peticiones futuras.

- **Servidor Caché:** Un servidor DNS en caché tiene como función principal almacenar temporalmente las respuestas a consultas de resolución de nombres de dominio que realiza un usuario o una red, de modo que si se realiza la misma consulta nuevamente en el futuro, el servidor pueda responder de manera más rápida consultando su memoria en lugar de buscar la información en otros servidores DNS en la red. No tiene autoridad sobre ninguna zona. Sus características son:

  - **Almacenamiento Temporal:** El servidor DNS en caché almacena temporalmente las respuestas a consultas de DNS. Esto reduce la carga en los servidores DNS autorizados y acelera las respuestas a las consultas recurrentes.
  - **Tiempo de Vida (TTL):** Cada respuesta DNS almacenada en la caché tiene un valor de Tiempo de Vida (TTL) asociado que determina cuánto tiempo puede mantenerse en la caché antes de ser descartada. Cuando el TTL expira, la respuesta se elimina de la caché y el servidor deberá buscar una respuesta actualizada si se realiza la misma consulta.
  - **Resolución Eficiente:** El servidor DNS en caché es el primer punto de contacto para las consultas de resolución de nombres de dominio. Si tiene la respuesta en su caché, la proporciona de inmediato. Si no, busca la respuesta consultando a otros servidores DNS en Internet.
  - **Mejora de la Velocidad:** Al tener respuestas en caché, se reduce significativamente el tiempo de resolución de nombres de dominio. Esto es especialmente importante para acelerar la navegación web y las comunicaciones en línea.

- **Ventajas de los DNS:**

**- Facilita la Navegación Web:** Los servidores DNS traducen nombres de dominio amigables para las personas en direcciones IP, lo que facilita la navegación en Internet al permitir a los usuarios acceder a sitios web utilizando nombres en lugar de direcciones IP numéricas.
**- Reducción de la Carga de Memoria:** Almacenan las respuestas de DNS en caché temporalmente, lo que reduce la carga en los servidores raíz y autorizados, acelerando las consultas posteriores para los mismos nombres de dominio.
**- Redundancia y Disponibilidad:** La estructura jerárquica de DNS permite una redundancia y distribución de carga efectivas. Si un servidor DNS falla, otros pueden asumir sus funciones.
**- Escalabilidad:** DNS es escalable y puede manejar una gran cantidad de nombres de dominio y consultas, lo que lo hace adecuado para el crecimiento constante de Internet.
**- Flexibilidad:** Los administradores de redes pueden configurar y personalizar servidores DNS para satisfacer las necesidades específicas de su red.
**- Soporte Multilingüe:** DNS admite nombres de dominio en diferentes idiomas y caracteres no latinos, lo que facilita la internacionalización de Internet.

- **Desventajas de los DNS :**

**- Tiempo de Inactividad:** Si un servidor DNS principal experimenta un tiempo de inactividad o es atacado, puede causar problemas de acceso a sitios web y servicios en línea.
**- Riesgo de Cache Poisoning:** Los servidores DNS en caché pueden estar en riesgo de envenenamiento de caché si un atacante proporciona respuestas falsas que se almacenan en la caché del servidor.
**- Desafíos de Seguridad:** Los servidores DNS deben protegerse contra amenazas de seguridad, y una configuración incorrecta puede dejarlos vulnerables a multitud de ciberataques, y principalmente a ataques de suplantación (spoofing), donde los atacantes falsifican respuestas de DNS para redirigir a los usuarios a sitios web maliciosos.
**- Limitaciones de Privacidad:** Las consultas DNS pueden filtrar información sobre la actividad en línea de un usuario, lo que plantea preocupaciones de privacidad.
**- Latencia:** Si un servidor DNS está geográficamente lejos del usuario, puede aumentar la latencia en las consultas de DNS y ralentizar la resolución de nombres de dominio.

# • Curiosidades de los DNS:

- **El Primer Nombre de Dominio:** El primer nombre de dominio registrado en la historia fue "symbolics.com". Fue registrado el 15 de marzo de 1985 por la empresa Symbolics, Inc., que fabricaba computadoras.
- **13 Servidores Raíz:** A nivel global, hay 13 servidores raíz de DNS designados con letras de la A a la M. Estos servidores son la autoridad para las zonas de nivel superior y son cruciales para la resolución de nombres de dominio en Internet.
- **El Protocolo UDP:** La mayoría de las consultas DNS utilizan el Protocolo de Datagramas de Usuario (UDP) en lugar del Protocolo de Control de Transmisión (TCP). UDP es más eficiente para consultas de resolución rápida y requiere menos recursos.
- **DNS Censura y Bloqueo:** Algunos países y organizaciones bloquean o censuran ciertos nombres de dominio mediante manipulación de DNS para restringir el acceso a sitios web específicos. Esto es una preocupación importante en términos de libertad de Internet.
- **DNS en el Espacio:** La NASA ha utilizado el DNS para asignar nombres de dominio a sistemas y dispositivos en el espacio, como satélites y estaciones espaciales
- **DNS Gigante:** Uno de los servidores DNS más grandes del mundo es el servidor de raíz L (letra "L"). Se encuentra en Japón y gestiona una gran cantidad de consultas de DNS debido a la población y el uso de Internet en ese país.
- **Dominios primer nivel existentes:** http://data.iana.org/TLD/tlds-alpha-by-domain.txt

- # Curiosidades de los DNS:

## ¿Dónde están ubicados los 13 servidores raíz?

10 de ellos están situadas en Estados Unidos y 3 en el resto del mundo, he aquí la muestra del potencial de los EEUU en cuanto a IT.

**Servidor A:** Network Solutions, Herndon, Virginia, USA.
**Servidor B:** Instituto de Ciencias de la Información de la Universidad del Sur de California, USA.
**Servidor C:** PSINet, Virginia, USA.
**Servidor D:** Universidad de Maryland, USA.
**Servidor E**: NASA, en Mountain View, California, USA.
**Servidor F:** Internet Software Consortium, Palo Alto, California, USA.
**Servidor G:** Agencia de Sistemas de Información de Defensa, California, USA.
**Servidor H:** Laboratorio de Investigación del Ejercito, Maryland, USA.
**Servidor I:** NORDUnet, Estocolmo, Suecia. Servidor J: (TBD), Virginia, USA.
**Servidor K:** RIPE-NCC, Londres, Inglaterra. Servidor L: (TBD), California, USA.
**Servidor M:** Wide Project, Universidad de Tokyo, Japón.

| Letra | Dirección IPv4 | Nombre Antiguo | Operador |
|---|---|---|---|
| a.root-servers.net. | 198.41.0.4 | ns.internic.net | Verisign |
| b.root-servers.net. | 192.228.79.201 | ns1.isi.edu | USC-ISI |
| c.root-servers.net. | 192.33.4.12 | c.psi.net | Cogent Communications |
| d.root-servers.net. | 199.7.91.13 | terp.umd.edu | Universidad de Maryland |
| e.root-servers.net. | 192.203.230.10 | ns.nasa.gov | NASA |
| f.root-servers.net. | 192.5.5.241 | ns.isc.org | Internet Systems Consortium |
| g.root-servers.net. | 192.112.36.4 | ns.nic.ddn.mil | Defense Information Systems Agency |
| h.root-servers.net. | 128.63.2.53 | aos.arl.army.mil | U.S. Army Research Lab |
| i.root-servers.net. | 192.36.148.17 | nic.nordu.net | Netnod |
| j.root-servers.net. | 192.58.128.30 | | Verisign |
| k.root-servers.net. | 193.0.14.129 | | RIPE NCC |
| l.root-servers.net. | 199.7[illegible]83.42 | | ICANN |
| m.root-servers.net. | 202.12.27.33 | | Proyecto WIDE |

**Práctica 7:** Mediante los siguientes comandos de Windows / Linux, busca los 13 servidores DNS raiz:

```
nslookup -type=ns .
```

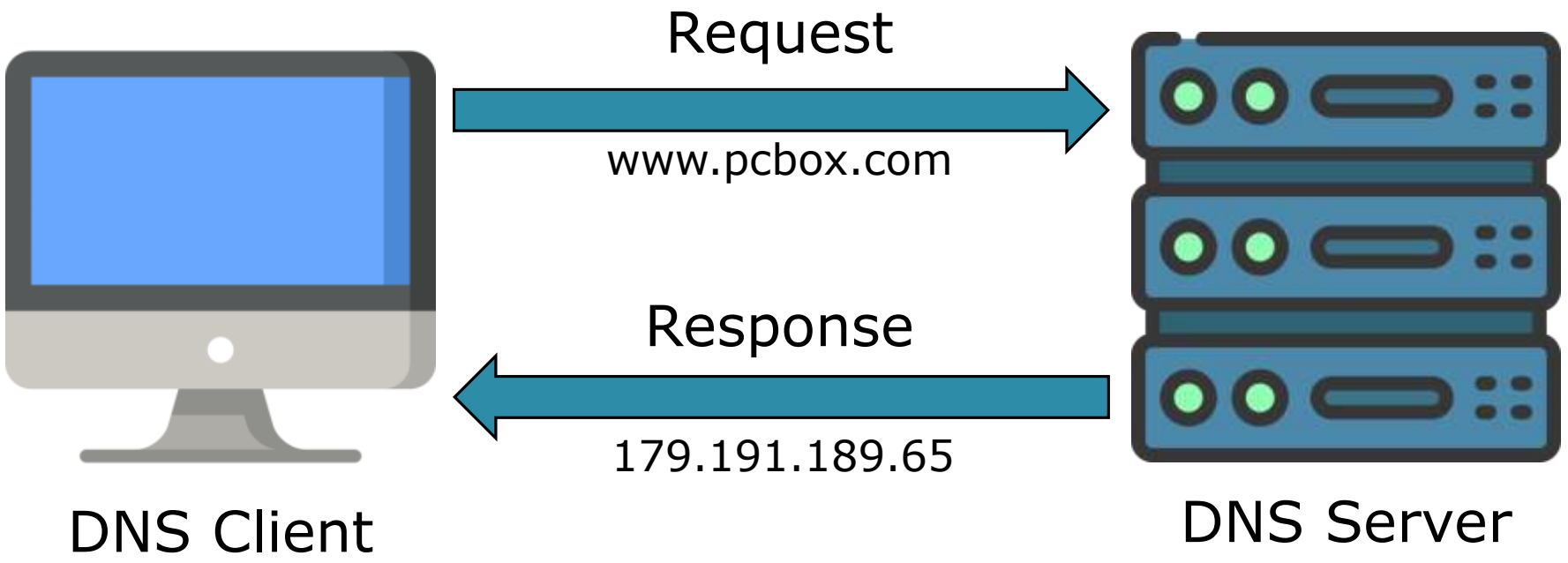
```
dig +short @a.root-servers.net. .
```

Los servidores raíz se nombran desde la letra "A" hasta la "M". Estos comandos le pedirán a tu servidor DNS local que realice una búsqueda de servidores de nombres (NS) para el dominio ".", que es la representación de la raíz del sistema DNS. La respuesta incluirá una lista de los servidores raíz principales, que son representados por las letras "A" a "M".

# FUNCIONAMIENTO DEL DNS Y TIPOS DE CONSULTA

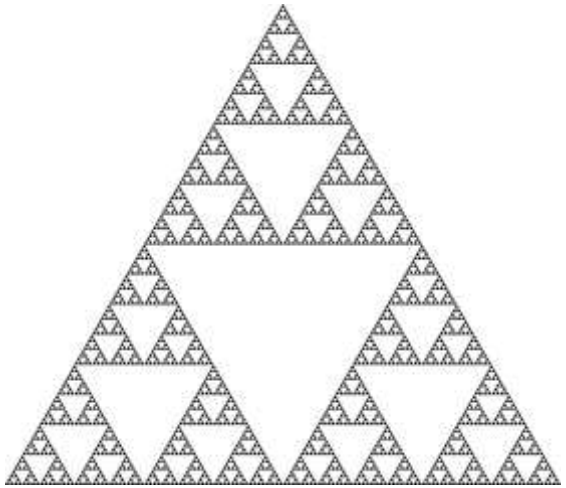
4

- **Funcionamiento del DNS:** Ya hemos estudiado anteriormente el flujo de resolución básico desde que se realiza una petición DNS hasta que se obtiene una respuesta. Ahora vamos a profundizar en los tipos de consulta que se pueden realizar a estos servidores, y en sus características principales.

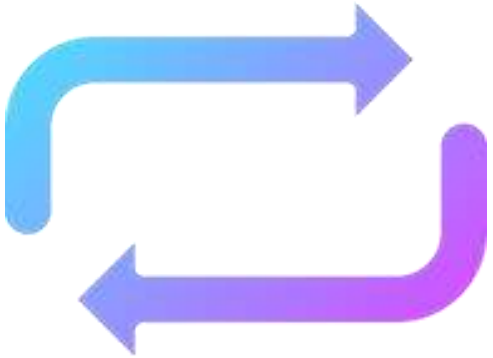


- **Tipos de consulta:** Un servidor DNS puede realizar los siguientes tipos de consulta, para resolver una request.

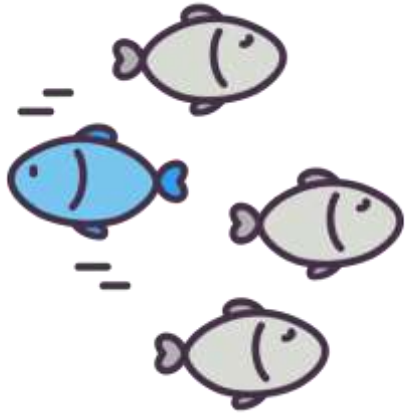
---



Recursiva



Iterativa

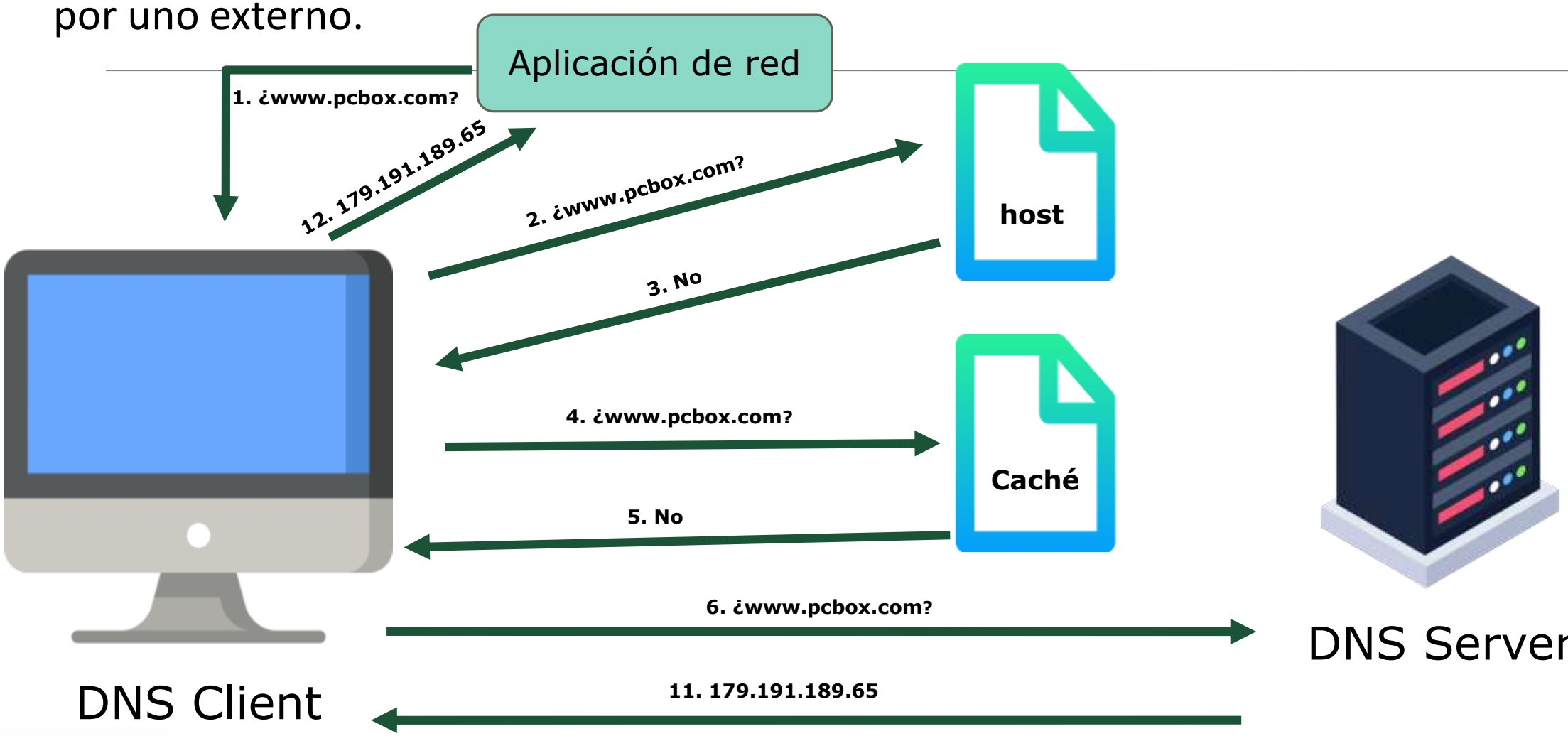


Inversa

- **Consulta recursiva:** El servidor DNS consultado realiza todas las acciones necesarias para obtener de manera completa y eficiente la respuesta completa a esa petición, y devolverla al cliente que hizo la solicitud, incluso interactuando con otros servidores DNS si fuera necesario.

  - El cliente envía una solicitud de resolución DNS
  - El servidor DNS recursivo local verifica su caché interna para ver si tiene la respuesta a esa consulta almacenada. Si la respuesta está en la caché y aún es válida, la devuelve al cliente de inmediato.
  - Si la respuesta no está en la caché o ha caducado, el servidor DNS recursivo local contacta a otros servidores DNS de manera secuencial a través de la jerarquía de servidores DNS de su confianza hasta encontrar la respuesta. Puede consultar a los servidores raíz para obtener información sobre los servidores TLD, luego a los servidores TLD para obtener información sobre el dominio específico y, finalmente, a los servidores autoritarios del dominio para obtener la dirección IP correspondiente.
  - Una vez que el servidor DNS recursivo ha obtenido la respuesta completa, la almacena en su caché para futuras consultas y la devuelve al cliente que hizo la solicitud original.

- **Funcionamiento de consulta recursiva:** se indica si es dada por el mismo servidor o por uno externo.



- **Ventajas e inconvenientes de la consulta recursiva:** Por un lado, el hecho de que los registros DNS se guarden en caché durante un cierto tiempo de vida "TTL", permite que la resolución de una petición DNS sea mucho más rápida de forma recursiva. Por otra parte, al permitir este tipo de consultas recursivas se crea una vulnerabilidad de seguridad debido a posibles ciberataques que conllevan el envenenamiento de los registros DNS.

- Además de la resolución del dominio, un servidor DNS que resuelva de forma recursiva sus peticiones puede enviar la siguiente información:

  - Un error **NXDOMAIN**, que significa que el dominio no existe
  - Un error temporal, por problemas de conectividad
  - La respuesta con la dirección del registro **A** y acompañada del **CNAME**. En esta respuesta siempre se indica si es dada por el mismo servidor o por uno externo.

- **Consulta iterativa:** En una consulta de DNS iterativa, cada consulta de DNS responde directamente al cliente con una dirección para que pregunte a otro servidor DNS, y el cliente sigue así consultando a los servidores DNS hasta que uno de ellos responda con la dirección IP correcta para el dominio dado.

Dicho de otro modo, el cliente lleva a cabo una forma de delegación en una consulta de DNS recursiva. Le dice al solucionador de DNS: "Oye, necesito la dirección IP de este dominio, por favor, búscala y no vuelvas a llamarme hasta que la tengas". Entretanto, en una consulta iterativa, el cliente le dice al solucionador de DNS: "Oye, necesito la dirección IP de este dominio. Por favor, dime la dirección del siguiente servidor DNS en el proceso de búsqueda para que pueda buscarla yo mismo".

Este caso se dá cuando el cliente solicita el uso de la recursividad, pero ésta se encuentra deshabilitada en el
servidor DNS, o cuando el cliente directamente no solicita el uso de la recursividad al consultar el servidor DNS.

- Además de la resolución del dominio, un servidor DNS que resuelva de forma iterativa sus peticiones puede enviar la siguiente información:

  - Un error **NXDOMAIN**, que significa que el dominio no existe
  - Un error temporal, por problemas de conectividad
  - La respuesta con la dirección del registro **A** y acompañada del **CNAME**. En esta respuesta siempre se indica si es dada por el mismo servidor o por uno externo.
  - Lista de servidores para preguntar por la petición del cliente para avanzar con la búsqueda. Esta respuesta es habitual en servidores raíz o TLD (solo aceptan peticiones iterativas)

**Consulta inversa:** En la mayoría de la consultas DNS los clientes normalmente realizan una búsqueda directa. Este tipo de consulta espera recibir una dirección IP como respuesta a la consulta. Pero, DNS también proporciona un proceso de búsqueda inversa, es decir, buscar un nombre de host a través de una dirección IP. Así, una búsqueda inversa busca la respuesta a una pregunta tipo como la siguiente: **¿Cuál es el nombre DNS del host que utiliza la dirección IP 192.168.200.100?.**

No confundir con el tipo de búsqueda DNS inversa, lo cual se refiere específicamente a la acción de buscar el nombre de dominio asociado a una dirección IP, mientras que el término "tipo de búsqueda DNS inversa" podría referirse a buscar registros PTR o realizar consultas relacionadas con DNS inverso en una base de datos de DNS.

- **Consulta Recursiva**  **Consulta iterativa.**

| Aspecto | Consulta Recursiva | Consulta Iterativa |
|---|---|---|
| **Propósito** | Obtener la respuesta completa y resuelta. | Recopilar información paso a paso y dirigir al cliente a lo largo de la jerarquía de DNS. |
| **Flujo de Consulta** | El servidor DNS consultado realiza todas las consultas y devuelve la respuesta completa al cliente. | El servidor DNS consultado proporciona una referencia a otro servidor DNS y el cliente debe realizar consultas adicionales secuenciales. |
| **Respuesta** | Respuesta completa y resuelta, generalmente una dirección IP. | Respuesta parcial con referencias a otros servidores DNS. |
| **Uso Común** | Para obtener la dirección IP de un nombre de dominio. | Utilizado cuando un servidor DNS local no realiza consultas recursivas o para obtener información específica de los servidores autoritarios. |
| **Ejemplo** | Resolviendo "www.ejemplo.com" a una dirección IP. | Consultando a los servidores raíz y TLD para obtener información de un dominio. |

# - Práctica 1 - INSTALACIÓN Y CONFIGURACIÓN DE UN SERVIDOR DNS

5

# SERVICIO DE DIRECTORIO: CARACTERISTICAS Y FUNCIONALIDAD

6

**Caso práctico**

*A BK Programación, una empresa, con la que ya han trabajado anteriormente en proyectos asignados a **Juan**, les ha encargado un proyecto con las siguientes especificaciones para el departamento de atención al cliente:*

1. *Controlar el acceso de usuarios a los equipos de la empresa, de tal forma que, independientemente del ordenador con el que trabajen en la empresa, mediante autenticación de usuario y contraseña, puedan tener acceso al mismo.*
2. *Controlar el acceso de usuarios a la herramienta de gestión de incidencias y proyectos.*

*Para ello BK Programación ha determinado realizar una autenticación por LDAP mediante OpenLDAP, puesto que aunque la configuración y el tiempo empleado va a ser más costoso que empleando otras alternativas, determina que la empresa necesita una centralización de esa base de datos de usuarios para que la aplicación de gestión de incidencias y proyectos, y los equipos ofrecidos por la empresa a sus trabajadores, puedan beber de la misma fuente: la base de datos de OpenLDAP.*

- **Directorios:** Permiten encontrar información de forma más eficaz que otros métodos más convencionales como por ejemplo: una guía telefónica impresa en papel o una revista con la programación televisiva.

Estos directorios tradicionales tienen los siguientes problemas:

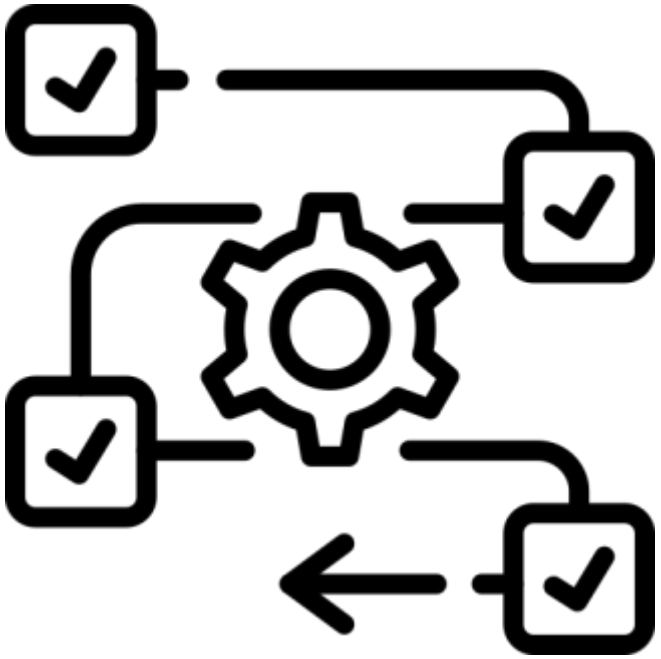
**- Son estáticos e inflexibles:** Una guía telefónica impresa en 2013 no contiene ni modificaciones posteriores, ni por ejemplo, datos de 2022, sin embargo un directorio electrónico puede ser consultado y actualizado con nuevos datos en tiempo real, y con un tipo de búsqueda más ágil basado en el tipo de organización implementada.

**- Son inseguros:** ¿Cómo impides que un usuario busque un número de teléfono en esa guía telefónica? En un directorio electrónico, sólo el que disponga las claves de acceso obtendrá la información.

**- Son difícilmente configurables:** ¿Cómo buscar en una guía de tv los programas de deporte que se emitan para Europa vía satélite en una franja horaria determinada?. Por contra, los directorios electrónicos pueden establecer la información que recibe una persona en función de sus necesidades.

- **Funciones básicas de los directorios.**


Buscar información



Gestionar información

Control de seguridad

- **¿Para qué usar un servicio de directorio?**

En resumen, un directorio es un sistema que organiza información sobre los usuarios, sobre los equipos de una red corporativa y los recursos. Además, permite administrar los recursos sobre una red.

**- Encontrar información:** Permiten realizar búsquedas por orden alfabético, apellido, dirección, teléfono, código postal…

**- Gestionar información:** Uno o varios usuarios a la vez pueden agregar, editar, eliminar datos en tiempo real, de forma centralizada en un solo directorio, evitando que tener que actualizar varios directorios independientes con el riesgo de que si no se hiciera correctamente la sincronización, se obtendrías datos obsoletos o erróneos. Ejemplo: Varios servidores web con autenticación, que deben de acceder a la misma base de datos.

**- Control de seguridad:** Los directorios tienen la función de delimitar el acceso a los usuarios a los datos o los recursos disponibles.

- **¿Para qué usar un servicio de directorio?**

Algunos ejemplos de aplicaciones de los servicios de directorio son:

- Autenticación de usuarios: En aplicaciones web, correo electrónico, RADIUS (*protocolo de autenticación y autorización para aplicaciones de acceso a la red o movilidad IP*)...,
- Sistemas de control de entradas a edificios.
- Bases de datos comunes en organizaciones.
- Sistemas operativos: gestión de cuentas de acceso, servidores de certificados, libretas de direcciones compartidas...

- **Directorio vs DNS: Ambos proporcionan acceso a una base de datos jerárquica, pero se diferencian en:**

| | Servidores de Directorio | Servicio DNS |
|---|---|---|
| **Propósito** | General | Traducción de nombres de dominio a direcciones IP |
| **Estructura de la Información** | No fija | Estructura fija |
| **Actualizaciones** | Permitidas | No permitidas para servidores raíz DNS |
| **Protocolo de Comunicación** | TCP (orientado a conexión) | UDP (no orientado a conexión) |

- **Directorio vs DNS**

Pero, a pesar de sus diferencias no poseen el impedimento de trabajar juntos, es más, usualmente los encontrarás complementándose en aplicaciones web con distintas funcionalidades, como: servidores de correo, gestión de proyectos e incidencias, servidores RADIUS, etc. Así, suele ser necesario acceder a las URL de las aplicaciones web mediante nombres de dominio DNS y una vez en ellas autenticarse por medio de LDAP.

# ORGANIZACIÓN LDAP

7

- **En primer lugar, ¿LDAP qué es?:**

LDAP (Lightweight Directory Access Protocol) es un protocolo de aplicación utilizado para acceder y administrar servicios de directorio en una red. Los servicios de directorio son sistemas que almacenan información sobre recursos de red, como usuarios, grupos, dispositivos y otros objetos, en una estructura jerárquica. Estos servicios se utilizan comúnmente para gestionar la autenticación, la autorización, la configuración de recursos y la búsqueda de información en una red.

LDAP se utiliza principalmente para realizar operaciones de búsqueda, lectura, escritura y modificación en un servicio de directorio. Algunos ejemplos de servicios de directorio populares que utilizan LDAP son:

- **Active Directory**: Un servicio de directorio de Microsoft utilizado en entornos de red basados en Windows.
- **OpenLDAP**: Una implementación de código abierto de LDAP que se utiliza ampliamente en sistemas Unix y Linux.
- **Novell eDirectory**: Un servicio de directorio de Novell que se utiliza en redes empresariales.
- **Oracle Internet Directory**: Un servicio de directorio de Oracle utilizado en entornos de bases de datos Oracle.

- **De acuerdo, pero para que lo entendamos todos:**

Imagina que estamos en una biblioteca gigante (directorio), y necesitamos encontrar un libro (cualquier dato del directorio, nombres de usuario, contraseñas…).


- Para poder encontrar fácilmente el libro que buscamos, en la Biblioteca se ha implementado un sistema de indexación y búsqueda avanzado (LDAP)
- Tú, como usuario de la biblioteca (directorio), utilizas el catálogo de búsqueda (LDAP) para encontrar el libro (información) que necesitas.
- El catálogo de búsqueda (LDAP) te permite buscar libros por autor, título, género, fecha de publicación, o cualquier otro criterio que desees.
- Una vez que encuentras el libro que buscas en la biblioteca utilizando el catálogo de búsqueda, puedes obtener la información que necesitas rápidamente. De manera similar, LDAP te permite acceder y recuperar información rápidamente de una red.

Entonces, LDAP es como un catálogo de búsqueda avanzado que te ayuda a encontrar y acceder a la información almacenada en una red, de la misma manera en que un catálogo de búsqueda te ayuda a encontrar libros en una gran biblioteca.

- La organización de estos tipos de directorios puede ser de dos tipos.

Centralizada

Distribuida

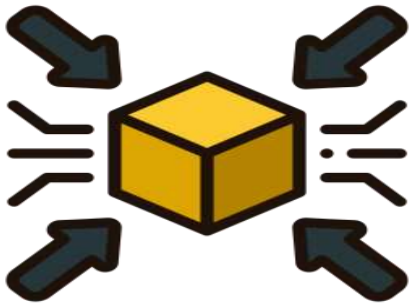
- **Centralizada:** En este caso las consultas se canalizan hacia un mismo servidor y este responde a todas ellas, no siendo por tanto necesario sincronizarlo con otras máquinas, pero tiene el inconveniente de que, en caso de fallo, se dejan a las aplicaciones sin validación.

- **Distribuida:** Permite que la información este dividida en varios servidores y cada servidor puede contestar en función de que si la respuesta la tienen ellos o no. Los datos pueden estar:

  ✓ **Fraccionada :** Ocurre cuando los servidores no tienen toda la información sino un subconjunto de ella. Cada servidor de directorio almacena un subconjunto único y no solapado de la información, es decir, una entrada es almacenada en un solo servidor.

  ✓ **Replicada:** Ocurre cuando toda la información está en todos los servidores

- La mejor opción para configurar y dar servicio de directorios es la combinación de ambas estrategias. Una parte debería estar fraccionada y otra replicada, así salvamos los inconvenientes de cada una de ellas.



Centralizada


Distribuida

- En 1988, se creó el estándar X.500, sobre servicios de directorio, que organizaba las entradas de manera jerárquica, con gran capacidad de búsqueda y de almacenamiento. Usaba el protocolo a **nivel de aplicación** DAP, para la comunicación entre cliente-servidor, que tenían que implementar toda la torre de protocolos OSI.

- LDAP surge como alternativa a DAP, teniendo gran éxito debido a:

  ✓ LDAP usa **TCP/IP,** por lo que es más fácil de implementar **cliente-servidor.**
  ✓ LDAP presenta la información mediante cadena de caracteres en lugar de complicadas estructuras ASN.1.
  ✓ El modelo funcional de LDAP es más simple y se ha eliminado las opciones raras usadas en X.500, siendo más fácil de comprender.

- LDAP define un protocolo para el contenido de los mensajes entre un cliente y el servidor propiamente dicho.

- LDAP es implementado mediante Open LDAP de código abierto.

# ARCHIVOS BÁSICOS DE CONFIGURACIÓN Y USO

- El directorio LDAP tiene una estructura en forma de árbol denominado **DIT**.

---

- Cada entrada del directorio describe un **objeto**: persona, impresora, etc.

- La ruta completa a una entrada la identifica de modo inequívoco y se conoce como **DN** y está compuesto por una secuencia de partes más pequeñas llamadas **RDN**, de forma similar a como el nombre de un fichero consiste en un camino de directorios en muchos sistemas operativos.

- Una clase de objeto (**objectClass**) es una descripción general de un tipo de objeto. Todos los objetos de LDAP deben tener el atributo objectClass. La definición de objectClass especifica qué atributos requiere un objeto LDAP, así como las clases de objetos que pueden existir. Los valores de este atributo los pueden modificar los clientes, pero el atributo objectClass en sí no puede eliminarse.

- Un **esquema (schema)** define: qué clases de objetos se pueden almacenar en el directorio, qué atributos deben contener, qué atributos son opcionales y el formato de los atributos.

---

Por lo general, existen dos tipos de objetos:

- **Contenedor**: Este tipo de objeto puede contener a su vez otros objetos. Algunos ejemplos de estos elementos son:
  - **-Root** (elemento raíz del árbol de directorios que no existe en realidad).
  - **-c** (country).
  - **-ou** (OrganizationalUnit).
  - **-dc** (domainComponent).

La figura análoga al contenedor es el directorio (carpeta) de un sistema de archivos.


- **Hoja**: Este tipo de objeto se encuentra al final de una rama y carece de objetos subordinados.

Algunos ejemplos son: Person/InetOrgPerson o groupofNames.

- El formato **LDIF** es el estándar para representar entradas del directorio en formato texto ASCII.

- La sintaxis que posee LDIF es la siguiente:

*dn: <nombreDistinguido>*
*<nombreAtributo>: <valor>*
*<nombreAtributo>: <valor>*
*<nombreAtributo>: <valor>*

| Atributo | Significado |
|---|---|
| cn | "common name", nombre |
| sn | "surname", apellido |
| uid | "userid", nombre de usuario |
| mail | e-mail |
| ou | "organizational unit", unidad organizativa |
| telephoneNumber | Número de teléfono |

- Como se puede observar en la diapositiva anterior el formato LDIF se pude dividir en dos partes:

  - DN que debe configurar en la primera línea de entrada y que se componen de la cadena dm: seguida del nombre distinguido (DN) de la entrada.
  - La segunda parte son los atributos de la entrada, como se puede observar en el ejemplo anterior.

- Es recomendable  colocar como primer atributo *objectclass*, para mejorar la legibilidad de este.

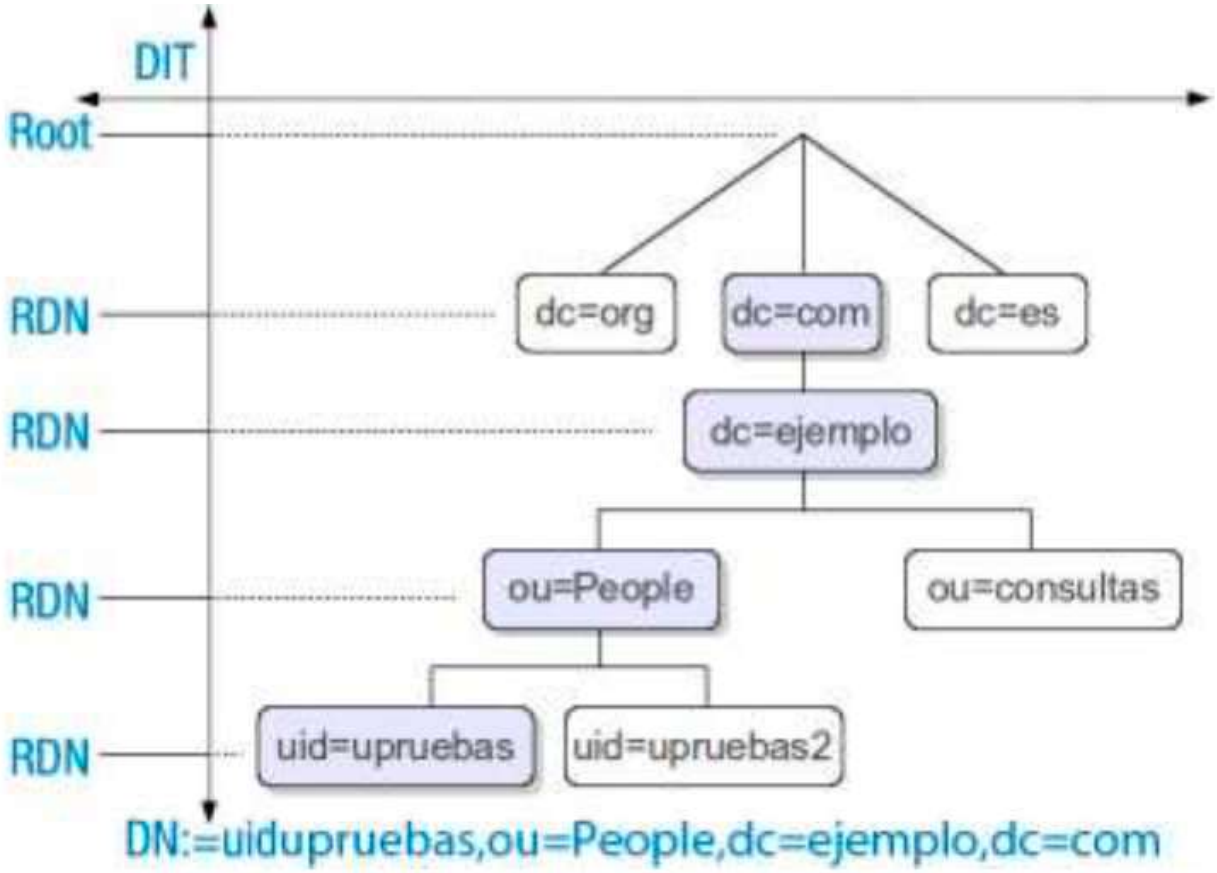
- Cada entrada se separa de la anterior con una línea en blanco, y a su vez en cada entrada puede existir cualquier cantidad de atributos (<nombre_atributo>: <valor>.)

En la siguiente imagen se ilustra las relaciones jerárquicas dentro de un árbol de directorios LDAP denomiado **DIT.**

A este elemento le puede seguir en un nivel inferior **c** (country), **dc** (domainComponent) ó **o** (organization).

La figura representa un DIT ficticio con entradas en cuatro niveles. Cada entrada se corresponde con una casilla en la figura. En este caso, el nombre válido completo DN del empleado ficticio **upruebas** es:

***dn: uid=upruebas,ou=People,dc=ejemplo,dc=com***



- **Ejemplo 1:** Crear una unidad organizativa llamada “empleados” dentro del dominio prueba.com

*dn:ou=empleados,dc=prueba,dc=com*
objectClass:organizationalUnit
ou:empleados

- **Ejemplo 2:** Crear un empleado de la organización con sus atributos dentro de la OU anterior.

*dn:uid=Fernando,ou=empleados,dc=prueba,dc=com*
objectClass:inetOrgPerson
uid:Fernando
ou:empleados
cn:Fernando
sn:Sanchez

Resumiendo, para aclararnos un poco afiancemos los conceptos:

- **Esquema**: Es como un conjunto de reglas que nos dice qué tipo de cosas podemos guardar en un directorio (como empleados), y cómo debe verse la información de esas cosas (nombre, correo electrónico, etc.).

- **Clase de objeto**: Es como una categoría que define qué información debe tener una cosa. Por ejemplo, si hablamos de "empleados", la clase de objeto nos dirá que un empleado debe tener un nombre, una dirección de correo electrónico, etc.

- **Atributos**: Son las partes específicas de una cosa. Si hablamos de un empleado, el nombre y el correo electrónico son ejemplos de atributos.

- **Ejemplo 3:** Supongamos que estamos creando un directorio LDAP para una pequeña empresa llamada "EjemploCorp" que tiene varios departamentos y empleados. Aquí tienes una estructura de organización LDAP para esta empresa:

**dc=ejemplocorp,dc=com**: Este es el punto de inicio de la organización, a menudo se llama la raíz del directorio. Aquí, "dc" significa "domain component" (componente de dominio) y "ejemplocorp" es el nombre de la empresa. "com" es el dominio de nivel superior.

**ou=Departments,dc=ejemplocorp,dc=com**: Esta es una "unidad organizativa" (Organizational Unit - OU) que contiene sub-unidades organizativas para cada departamento de la empresa. Puedes tener OUs para "Ventas", "Marketing", "TI", "Recursos Humanos", etc.

**ou=Ventas,ou=Departments,dc=ejemplocorp,dc=com**: Una OU específica para el departamento de Ventas.

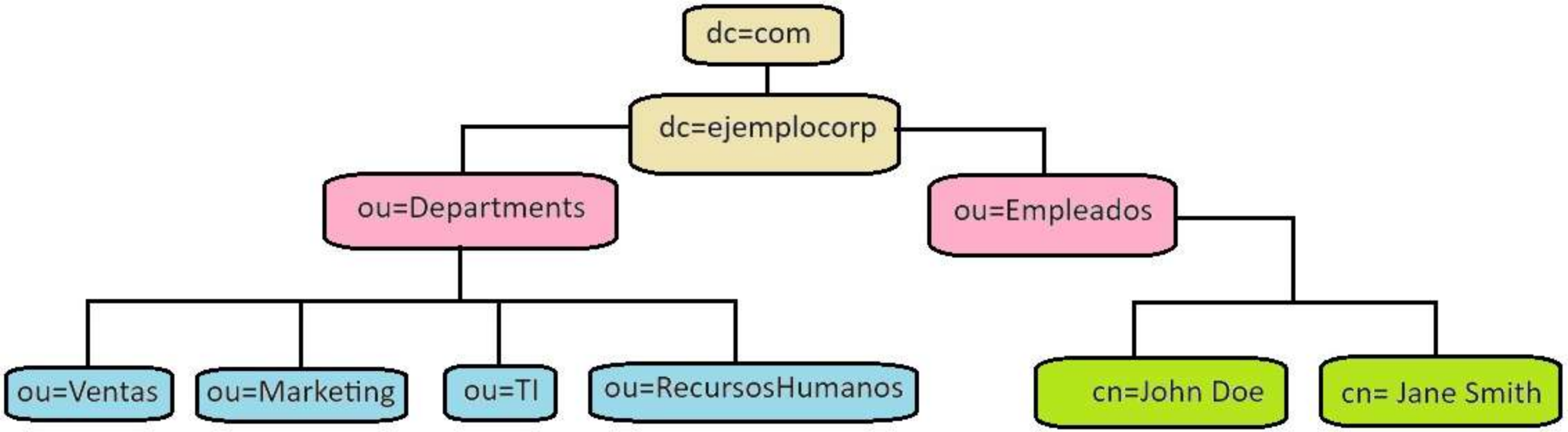
**ou=Marketing,ou=Departments,dc=ejemplocorp,dc=com**: Una OU específica para el departamento de Marketing.

**ou=Empleados,dc=ejemplocorp,dc=com**: Una OU para almacenar información sobre los empleados de la empresa.

**cn=John Doe,ou=Empleados,dc=ejemplocorp,dc=com**: Un objeto de entrada para un empleado llamado John Doe, que se encuentra en la OU de Empleados. Cada empleado puede tener atributos como "nombre", "apellido", "correo electrónico", etc.

**cn=Jane Smith,ou=Empleados,dc=ejemplocorp,dc=com**: Otro objeto de entrada para un empleado llamado Jane Smith.

- Este sería el árbol de directorios DIT del ejemplo anterior:




**Otro ejemplo más avanzado:** https://devopsideas.com/planning-of-ldap-dit-structure-and-config-of-overlays-access-ppolicy/

- **Integración del servicio de directorio con otros servicios**.

- Un servicio de directorio puede actuar como servidor de autenticación, además de contener información decidir un usuario puede acceder a determinada información.

- Se usa como repositorio para almacenar información que varios servidores deben compartir, como la configuración sobre el control de acceso.

- Otra utilidad es la de indexar la documentación que puede contener un servidor web, con la precisión que otras herramientas no tienen.