

## ARQUITECTURA

### ¿En que capa se encuentra el protocolo dns?

Se encuentra en la capa de **aplicacion** del modelo TCP/IP.

### ¿Diferencias entre pagina web y aplicacion web?

Una pagina web en un sitio estatico que tiene como principal objetivo informar a los usuarios, una aplicación web es una pagina dinamica que tiene una logica de negocio y una capa de datos, su objetivo es que los usuarios tengan a sus disposicion herramientas para realizar tareas.

### ¿Que es una plataforma web?

Una plataforma web es más que una página web. Una plataforma incluye elementos adicionales a la página web tales como aplicaciones, carritos de compras, formularios, convertidores, instancias de aprobación y alguna otra solución específica para la necesidad del cliente y el mercado. Estos elementos pueden ser públicos o privados, tales como sistemas de comunicación interna o inventarios.

## LAMP

Sus siglas significan LINUX APACHE MYSQL PHP.

Un ejemplo del modelo completo estaría compuesto por un servidor Apache y un Tomcat que se conecte a una base de datos. Un ejemplo del simplificado sería un servidor LAMP.

## La estructura de una Arquitectura Web actual sigue el siguiente modelo de tres capas:

1. **Una capa cliente:** es generalmente el navegador Web ejecutándose en el ordenador del usuario final.
2. **Un servidor Web (capa de negocio):** La capa cliente puede acceder a diferente lógica y procedimientos que existen en la capa de negocio. Aquí la lógica puede ser mucho más compleja que en la capa anterior. Los componentes de esta capa pueden ser desde simples archivos HTML hasta Servlets de Java.
3. **Una capa de datos:** Se compone de un sistema de almacenamiento acceso a datos que se utilizan para confeccionar la página Web. Generalmente es un gestor de bases de datos relacionales (SGDB) pero pueden ser ficheros de texto plano, ficheros XML, etc.

La **capa de negocio** puede estar a su vez **dividida** en dos partes si el sistema es suficientemente grande o complejo. Puede dividirse en una capa de **presentación** y una capa de **lógica de negocio**.

- **La capa de presentación** se encarga de componer las páginas integrando la parte dinámica en la estática. Además también procesa las páginas que envía el cliente (por ejemplo datos en formularios). Algunas soluciones para esta subcapa son los ASP de Microsoft o los JSP de Java. Esta parte la realiza generalmente un servidor web.
- **La capa de lógica de negocio** lleva a cabo operaciones más complejas. Se corresponde con la implantación de un servidor de aplicaciones. Realiza muchos tipos de operaciones entre los que destacan:
  - Realizar todas las operaciones y validaciones. o Gestionar el flujo de trabajo (workflow) incluyendo el control y gestión de las sesiones y los datos que se necesitan.
  - Gestionar todas las operaciones de accesos a datos desde la capa de presentación.

## Fases de un proyecto de aplicación web.

Se pueden considerar cuatro fases en el proyecto:

- 1. Concepto:** Durante esta fase se debe obtener una idea clara y concreta de qué quiere el cliente. Además hay que obtener una idea general de cómo se llevará a cabo y de si es viable o no.
- 2. Diseño:** Esta fase se centra en responder a cómo haremos la aplicación. Hay que concretar las tecnologías (tanto software como hardware) que usaremos y cómo se van a comunicar entre ellas. También hay que determinar los distintos módulos que usaremos y sus interfaces. Es muy importante realizar un plan de proyecto realista en el que se dividan las tareas y responsabilidades y se calculen los tiempos para cada elemento así como su secuencia y dependencias. También hay que obtener una especificación funcional en la que se detallen tanto el funcionamiento como el flujo de la aplicación.
- 3. Desarrollo:** En esta fase se debe desarrollar el proyecto en sí. Es muy importante llevar a cabo pruebas tanto unitarias como de integración así como gestionar una documentación del desarrollo y un control de versiones.
- 4. Pruebas e implantación:** Cuando el proyecto está totalmente terminado es necesario probarlo intensivamente antes de ponerlo en producción.

Una fase común a todos los proyectos informáticos y que no se incluye aquí es el **mantenimiento**.

Este concepto incluye dos partes.

El mantenimiento del servicio y corrección de errores y las mejoras.

La primera consiste en asegurarse de que todo sigue funcionando y solucionar los posibles errores y “caídas” del servicio.

El **orden** correcto para el desarrollo es empezar de abajo a arriba. Es decir, primero la **capa de datos**, luego la de **negocio** y por último la **presentación** al cliente.

## DNS

### ¿Que es el DNS y para que sirve?

El Domain Name System (DNS) es una base de datos distribuida y jerarquica que almacena informacion asociada a nombres de dominio en redes coo internet, como puede ser un servidor web.

**El DNS se utiliza para distintos propósitos. Los más comunes son:**

- **Resolución de nombres (Directa):** Dado el nombre completo de un host obtener su direccion ip.
- **Resolucion inversa de direcciones:** Dada una direccion ip, obtener un nombre asociado a las misma.

### **Tipos de servidores DNS**

- **Primarios o maestros:** Guardan los datos de una parte del espacio de nombres (zona) en sus ficheros. Las modificaciones, altas, o bajas de registros se realizan en estos servidores.
- **Secundarios o esclavos:** Obtienen los datos de zona de los servidores primarios a través de una **transferencia de zona**.
- **Locales o caché:** Funcionan con el mismo software, pero no contienen información de ninguna zona del espacio de nombres. Cuando los clientes les realizan una consulta, estos a su vez consultan a otros servidores DNS, almacenando la respuesta en caché para agilizar la repetición de estas peticiones en el futuro.

## Tipos de búsqueda.

### Resolución iterativa:

El servidor DNS local devuelve la mejor respuesta que puede ofrecer al cliente en función del contenido de su cache, pero si el servidor no dispone de la información solicitada indica la IP del siguiente servidor de nombres autorizado a preguntar, comenzando siempre por un servidor Raíz.

El proceso es el siguiente: cuando una aplicación (cliente) necesita resolver un FQDN envía un requerimiento al servidor de nombres configurado en el sistema (normalmente, el provisto por el ISP). A partir de entonces se desencadena el proceso de resolución del nombre:

1. El sistema tiene configurado el servidor de nombres 200.49.156.3 (perteneciente al proveedor ). Por lo tanto envía a éste el requerimiento de resolver "blog.smaldone.com.es".
2. El servidor de 200.49.156.3 envía la consulta root server 198.41.0.4.
3. 198.41.0.4 le informa que el servidor con autoridad sobre "es" es athea.es, cuya dirección IP es 200.16.98.2. (En realidad, informa la lista de todos los servidores con tal autoridad, pero para simplificar el ejemplo tomaremos solamente uno.)
4. 200.49.156.3 envía nuevamente el requerimiento a athea.es (el cual, recordemos, también tiene autoridad sobre "com.es").
5. athea.es responde que la autoridad sobre smaldone.com.es la tiene ns1.mydomain.com cuya dirección IP es 64.94.117.213.
6. 200.49.156.3 envía ahora la consulta a ns1.mydomain.com.
7. ns1.mydomain.com informa que la dirección IP de "blog.smaldone.com.es" es 208.97.175.41.
8. Finalmente, 200.49.156.3 devuelve este resultado a la aplicación que originó la consulta.

## Resolución recursiva.

Se realiza una petición de resolución de nombres al servidor local, y si el servidor no dispone de la información solicitada consulta al servidor raíz que menos tarde en ofrecer una respuesta, el servidor raíz consultará al intermedio, y así sucesivamente hasta llegar hasta el autorizado.

Una vez el autorizado responde al nivel anterior (con acierto o error), se van devolviendo los mensajes hasta llegar de nuevo al servidor local, y de este al cliente.

## Delegación

El objetivo principal del diseño del sistema de nombres de dominio en forma jerárquica fue su **administración descentralizada**. Este objetivo se consigue a través de la **delegación**. Una organización que administra un dominio puede dividirla en **subdominios**. Cada subdominio puede ser delegado a diferentes organizaciones, lo cual implica que esa organización será responsable de mantener los datos de ese subdominio.

## Tipos de registro en un servidor de nombres

- **A (Address)**: Este registro se utiliza para traducir nombres de hosts del dominio en cuestión a direcciones IP.
- **CNAME (Canonical Name)**: El nombre canónico es un alias para un host determinado. (No define una dirección IP, sino un nuevo nombre.)
- **NS (Name Server)**: Especifica el servidor (o servidores) de nombres para un dominio.
- **MX (Mail Exchange)**: Define el servidor encargado de recibir el correo electrónico para el dominio.
- **PTR (Pointer)**: Especifica un “registro inverso”, a la inversa del registro A, permitiendo la traducción de direcciones IP a nombres.
- **TXT (Text)**: Permite asociar información adicional a un dominio. Esto se utiliza para otros fines, como el almacenamiento de claves de cifrado, “DomainKeys” o “Sender Policy Framework”.

## ¿Qué es un servidor de nombres autoritario?

Con este nombre se conoce al servidor que contiene los registros de recursos para una zona (SOA y NS).

Cada zona puede tener uno o más servidores de nombres de dominio autoritarios. Uno de ellos debe ser primario. Si tiene varios, el resto pueden ser secundarios o caché. En el caso de que actúe como servidor primario, los registros de recursos para la zona se encuentran en los archivos de la zona, almacenados en el propio servidor DNS. Si se trata de un servidor secundario, los registros de recursos de la zona se cargan desde otro servidor de nombres (primario) mediante la transferencia de zona. Si, en cambio, el servidor de nombres autoritario es un servidor caché, se utiliza el método de búsquedas recursivas. De este modo, almacenará en la caché los resultados de las búsquedas que se hayan realizado.

## Transferencias de zona

Una **transferencia de zona** es el proceso por el que se copia el contenido de un archivo de zona DNS de un servidor DNS principal a un servidor DNS secundario.

## Dominios y Zonas

**Cada servidor** de nombres posee información sobre una parte contigua del espacio de nombres. Dicha parte del espacio se denomina **zona**, y se dice que el servidor de nombres tiene **autoridad** sobre ella.

Un servidor de nombres puede tener autoridad sobre múltiples zonas, y obtiene la información que describe la zona (los registros de recursos) o bien de un fichero local o bien de otro servidor de nombres

(zona)

Una zona aloja todo o una parte de un dominio y sus subdominios. En la definición de zonas debemos seleccionar el tipo de zona que se utilizará, así como donde almacenamos los datos de la misma.

La **zona DNS** es la información asociada a un nombre de dominio.

## Delegación vs Subdominios

La división en subdominios y la delegación de dichos subdominios son cosas distintas. Un dominio que tenga capacidad de autogestión (autoridad), siempre puede decidir subdividirse en diferentes subdominios.

Posteriormente, se puede decidir delegar la autoridad de algunos sus subdominios en otras organizaciones.

## Delegación

El objetivo principal del diseño del sistema de nombres de dominio en forma jerárquica fue su administración descentralizada. Este objetivo se consigue a través de la delegación. Una organización que administra un dominio puede dividirla en subdominios. Cada subdominio puede ser delegado a diferentes organizaciones, lo cual implica que esa organización será responsable de mantener los datos de ese subdominio.

## Espacio de nombres de dominio

- DNS está organizado en **nombres de dominio**, donde cada nombre de dominio es una trayectoria en un árbol invertido llamado **espacio de nombres de dominio**. El árbol tiene una única raíz superior (root) y una profundidad máxima de 127 niveles.
- Se denomina **dominio** a cualquier **subárbol** del espacio de nombres de dominio. De esta forma, cada dominio puede contener, a su vez, otros dominios. Generalmente, los **hosts** están representados por las hojas del árbol.