

Modelo de arquitectura OSI.

Open Systems Interconncetion. Es el modelo de referencia para los protocolos de red.

Se estructura por capas, cada una basada en un nivel inferior.

Niveles:

- Nivel 1: físico: medio de transmisión, señal eléctrica.
- Nivel 2: enlace: dirección MAC, acceso al medio y detección de errores.
- Nivel 3: red: enrutamiento origen-destino, firewalls e ip.
- Nivel 4: transporte: envío de datos, segmentos, protocolos tcp/udp.
- Nivel 5: sesión: mantener la comunicación abierta y reanudarla.
- Nivel 6: presentación: hacer reconocible la información, cifrado de datos.
- Nivel 7: aplicación: acceso a servicios.

Modelo cliente / servidor.

La mayoría de los servicios actuales. Elementos que intervienen:

- Proceso cliente: inicia la comunicación y solicita el servicio.
- Proceso servidor: espera la petición del cliente.
- Aplicación de red: programa que usa protocolos para interactuar con otras.
- Protocolo: conjunto de normas que detallan la forma en la que se comunican las aplicaciones para ofrecer un servicio.

Servicio	Aplicación servidor	Aplicación cliente	Protocolos
Web	<ul style="list-style-type: none"><li>• Apache</li><li>• IIS</li></ul>	<ul style="list-style-type: none"><li>• Firefox</li><li>• Internet Explorer</li><li>• Opera</li><li>• Safari</li><li>• Chrome</li><li>•</li></ul>	<ul style="list-style-type: none"><li>• HTTP</li><li>• HTTPS</li></ul>
Correo electrónico	<ul style="list-style-type: none"><li>• Exchange</li><li>• Postfix</li><li>• Sendmail</li></ul>	<ul style="list-style-type: none"><li>• Evolution</li><li>• Outlook</li><li>• Thunderbird</li></ul>	<ul style="list-style-type: none"><li>• POP</li><li>• IMAP</li><li>• SMTP</li></ul>

Modelo TCP/IP.

Simplificación optimizada del modelo OSI. Compuesto de 4 capas: acceso a red, red, transporte y aplicación.

La información intercambiada en cada capa está contenida en unidades de datos.

Cada capa encapsula las unidades de datos superiores.

- Acceso a red: identificación mediante la dirección MAC. Formato 6 octetos hexadecimales. Trabaja con frames como unidad de datos. Usa hubs y switches para la transmisión. Los switches recuerdan la dirección MAC y los hubs son menos eficientes.

- Red: transmite paquetes a través del router. Enruta las redes y fragmenta los envíos grandes. Las IPv4 son números binarios de 32bits en 4 octetos, los dos primeros es la id de la red, y los finales la del host.  
Los Gateway son dispositivos que nos permite interconectar redes. Puede ser un pc, router Gateway especializado o un servidor proxy. El más común es el router.

CLASE IP	VALOR DE CAMPO	IDENTIFICADOR DE RED	IDENTIFICADOR DE ESTACIÓN
A	0	7 bits	24 bits
B	10	14 bits	16 bits
C	110	21 bits	8 bits
D	1110	28 bits	x
E	11110	27 bits	x

- Transporte: gestiona a la vez varios orígenes y destinos en una misma comunicación. Se usan como añadidos a las direcciones IP. Conocido: 0-1023 para una aplicación estándar. Registrado: 1024-49151 no estándar. Dinámicos 49152-65535 inicio de conexión.  
UDP: sin establecimiento de conexión previo ni control de flujo.  
TCP: establecimiento de conexión, control de flujo y errores.
- Aplicación: protocolos que se usan para aplicaciones de más alto nivel.
  - DHCP: asignación dinámica de direcciones.
  - FTP: transferencia de ficheros.
  - HTTP, HTTPS: páginas web.
  - DNS: resolución de nombres.
  - SMTP, POP3, IMAP: correos.

Servidor.

Proporciona servicios, datos o recursos a otros equipos. Acepta peticiones y responde a estas. Antes la infraestructura era física, actualmente se virtualizan.

- Servidores web: procesa peticiones http y devuelve el contenido. Apache.
- Servidores de aplicaciones: facilita la ejecución de aplicaciones dinámicas y maneja las interacciones cliente servidor. Puede proporcionar servicios a las aplicaciones que ejecutan como autenticación, autorización, sesión, etc. Suelen interactuar con otros servidores. Tomcat.
- Servidores on-premise: se encuentran en las instalaciones de una organización. Esta organización es responsable de la compra, instalación y mantenimiento del hardware y software.
- Servidores cloud: infraestructura proporcionada por proveedores de servicios en la nube. Los recursos se alquilan y escalan en función de las necesidades.

Los servidores on-premise son mas caros, necesita personal especializado para el mantenimiento, la escalabilidad es limitada, el acceso es local, tienes mayor seguridad, pero la iniciación es mas lenta ya que necesita instalación.

Los servidores cloud son mas baratos, el mantenimiento lo hace el proveedor en la nube, la escalabilidad es alta, se puede acceder desde cualquier sitio, es menos segura, pero la iniciación es más rápida.

Escalabilidad.

Capacidad de un sistema para adaptarse al aumento de demanda. Existen dos tipos:

- Vertical: aumentar los recursos para mejorar el rendimiento.  
Ventajas: fácil de implementar, las aplicaciones no necesitan estar preparadas.  
Desventajas: limite físico, baja tolerancia a los fallos.
- Horizontal: aumentar el número de servidores.  
Ventajas: mayor tolerancia a fallos y es fácil adaptarse.  
Inconvenientes: las aplicaciones tienen que estar preparadas, puede ser difícil de gestionar.

Claves:

- Balanceo de la carga: distribuye las peticiones entre los servidores. Mejora la disponibilidad, aumenta la escalabilidad y optimiza el rendimiento.
- Replicación: replica sistemas para mejorar la disponibilidad de la información. La información debe estar sincronizada entre las réplicas, esto puede ser en tiempo real (síncrona) o con retraso (asíncrona). Existen dos tipos:
  - replicación maestra-esclavo: solo el maestro acepta las escrituras y las réplicas manejan la lectura.
  - Replicación maestro-maestro: todos pueden recibir lectura y escritura y luego se sincronizan.
- Cache: almacenamiento de rápido acceso para guardar información frecuente.
- Microservicios: una aplicación se divide en múltiples servicios mas pequeños. Los microservicios pueden escalar en función de la demanda.
- Contenedores y orquestaciones: los contenedores o dockers sirven para empaquetar aplicaciones. La orquestación o kubernetes para escalar los contenedores. Facilita el despliegue y la gestión en entornos distribuidos.

Característica	Balanceo de Carga	Replicación
Objetivo principal.	Distribuir la carga de trabajo entre varios servidores.	Mantener múltiples copias de datos en diferentes nodos.
Tolerancia a fallos.	Redirige el tráfico a otras instancias si una falla.	Garantiza datos disponibles, aunque un nodo falle.
Enfoque en la disponibilidad.	Mejora disponibilidad de las peticiones de los clientes.	Mejora la disponibilidad de los datos y evita la pérdida.
Datos sincronizados.	No necesariamente implica que los datos estén sincronizados.	Asegura que todas las réplicas contengan los mismos datos.
Escalabilidad.	Se enfoca en escalar el número de servidores para manejar más tráfico.	Se enfoca en la disponibilidad y consistencia de datos en varios nodos.
Tipo de fallos que cubre.	Fallos en la sobrecarga de consultas o fallos temporales de un nodo.	Fallos permanentes o catastróficos en el almacenamiento de datos.
Relación entre nodos.	Los nodos pueden no tener datos idénticos.	Los nodos tienen copias idénticas de la base de datos.

CDN: es una red de servidores distribuidos geográficamente especializados en entregar contenido web rápidamente. Almacena recursos estáticos de los sitios web. Reducen la latencia, distribuyen la carga y aumentan la tolerancia a fallos.

## Arquitectura de servidores.

Se refiere a cuantos servidores se utilizan, la función de cada uno y como interactúan.

- Servidor único: un único servidor aloja la aplicación web y la bbdd. Adecuado para aplicaciones pequeñas, bajo coste, simplicidad administrativa con limitaciones en escalabilidad y tolerancia a fallos.
- Servidor web+ servidor bbdd: mejora el rendimiento y permite distribuir los recursos, mayor seguridad y mejora de la escalabilidad.
- Balanceo de carga solo en web: un solo servidor de bbdd, pero varios para la aplicación web. Mejora el tiempo de respuesta y la tolerancia a fallos en la web, pero si falla la bd falla todo el sistema. Mejora la escalabilidad y tolerancia a fallos de la página web, pero no de la base.
- Balanceo de carga en web y bbdd: múltiples servidores tanto para la web como para la bd. Mejora el tiempo de respuesta y la tolerancia a fallos, mejora la escalabilidad y si hay fallo en un servidor la información está replicada.
- N-niveles: división física de los componentes del sistema. Puede ser:
  - Estricto: la solicitud no puede saltarse ningún nivel. Mayor latencia y sobrecarga.
  - Relajado: la solicitud puede saltarse algunos niveles si es necesario. Mas acoplamiento difíciles de cambiar.
  - Puede ser híbrido.

Los niveles pueden llamar a otros directamente o usar patrones. Se pueden hospedar varias capas en el mismo nivel. La separación física de los niveles mejora la escalabilidad y la resistencia, pero aumenta la latencia.

- Microservidores: son pequeños e independientes. Están acopladas de forma imprecisa, no existe una gran dependencia entre ellos. Cada servicio es un código independiente. Los servicios son los responsables de conservar sus datos o estados. Se comunican entre si mediante API bien definidas, los detalles de implementación interna esta oculta para otros servicios. Admite múltiples lenguajes de programación.

Ventajas: agilidad en la modificación y la actualización, equipos de desarrollos más pequeños, mantenimiento fácil del código, pueden incorporarse nuevas tecnologías. Pueden escalar de manera independiente. Se puede modificar el esquema de los datos sin afectar a otros servicios.

Inconvenientes: las partes del sistema son simples, pero en conjunto son mas complejas. Dificultad ante la realización de pruebas, falta de homogeneidad ya que existen distintos lenguajes, los datos hay que diseñarlos teniendo en cuenta la independencia y debe existir alguien que conozca bien la arquitectura.

## Apache.

El mas conocido es el servidor web de Apache, pero tienen otros proyectos como:

- Tomcat: servidor de aplicaciones.
- Hadoop: big data.
- Maven: compilación de aplicaciones.

## Servidor web Apache.

Servidor http: el código fuente esta disponible para descargar y modificar. Se publico en 1995. Es uno de los mas utilizados junto a nginx. Es un servidor modular permitiéndole ampliar las capacidades del servidor.

Un modulo es un software que se integra en apache en tiempo de ejecución. Aportan autenticación y encriptación, soporta varios lenguajes de programación y permiten el registro y diagnóstico. Los módulos pueden desarrollarse con C.

Se instala con el comando “sudo apt install apache2”. El fichero “/etc/apache2/apache2.conf” es el fichero principal de configuración, si no se especifica una directiva se usará su valor por defecto. # para comentar.

Fichero “/etc/apache2/ports.conf” define las ip y puertos que el servidor escucha.

“/etc/apache2/mods-available” configura los módulos disponibles, el .load indican como cargar los módulos y el .conf la configuración de los módulos.

“/etc/apache2/mods-enabled” los módulos activos, son los que se cargan.