

Administra el hardware y es una suite de aplicaciones (se puede instalar software)

Soporte logico que controla el funcionamiento del equipo fisico.

Empieza a funcionar cuando encendemos el equipo, y cesa cuando lo apagamos.

OS en los dispositivos con microprocesador; administran dispositivos, recursos y procesos.

Entre los recursos que administra tenemos:

- Gestionar RAM y ejecutar aplicaciones, designando recursos necesarios
- Administrar CPU (algoritmo de programacion)
- Direccionar entradas/salidas de datos (con drivers) a través de perifericos
- Administrar informacion
- Dirigir autorizaciones de uso para el usuario
- Administrar archivos

Servidores: Suelen tener OS legados de Unix, como Red Hat, o Windows Server

Se diferencian dado que son *multiusuarios* (Varios usuarios trabajando al mismo tiempo sobre el mismo nucleo), mientras que las domesticas son *monousuarios*

Se pueden clasificar segun licencia en open source y proprietary software.

Open Source permiten modificar, usar y adaptar el OS, x ejemplo Ubuntu y Red Hat.

Proprietary Software no permite modificaciones, como Windows

Los sistemas operativos también se pueden clasificar segun tres aspectos: Usuarios, Gestión de tareas, Gestión de recursos

### **Usuarios**

- Multiusuario : Permite varios usuarios ejecutando programas simultáneamente, por medio de terminales conectadas o por sesiones remotas en una red (Unix, Linux, Solaris)
- Monousuario : Permite ejecutar los programas de un solo usuario a la vez

### **Gestión de Tareas**

- Multitarea : Puede ejecutar varios procesos al mismo tiempo
- Monotarea : Permite un solo proceso a la vez, sin interrupciones (DOS por ejemplo)

### **Gestión de Recursos**

- Centralizado : Permite utilizar los recursos de un solo ordenador. Permiten usar el hardware de donde están físicamente.
- Distribuido : Permite ejecutar los procesos de mas de un ordenador al mismo tiempo. Toma recursos hardware ubicados físicamente en lugares distintos, haciéndolo funcionar como una sola, potenciando sus capacidades (porque no alcanzan los recursos actuales)

### **Estructura Interna**

- Monolítica : Un solo programa, compuesto de rutinas entrelazadas, de forma que se pueden

comunicar. Hechos a medida, rápidos pero no flexibles.

- Jerárquica : A medida que se complejiza, se requiere una organización de OS. Una parte del sistema, contiene subpartes, y organizado en niveles. Se subdivide en capas o anillos, bien definidas y con una interfaz característica.
- Máquina Virtual: Separan dos conceptos que suelen estar unidos, multiprogramación y máquina extendida. El objetivo es integrar distintos OS, pareciendo ser varias máquinas diferentes.
- Cliente-Servidor: Sirve para todas las apps (propósito general). Mantener la visión que tiene un usuario de una computadora personal, pero con la red permitiendo que comparta recursos (espacio disco, impresora) para economizar.

## Arquitectura Cliente-Servidor

Objetivo: procesar información de modo distribuido; dispersos en lugares distintos, acceden a recursos compartidos

Transparencia e independencia hardware y software.

Características:

- Protocolo asimétrico: el servidor se limita a escuchar, esperando que algún cliente inicie una solicitud.
- Acceso transparente, multiplataforma y multiarquitectura
- Fácil escalabilidad: facilidad para añadir nuevos clientes a la infraestructura (escalabilidad horizontal), o bien aumentar la potencia del servidor/servidores, aumentando número o capacidad (escalabilidad vertical)

Componentes:

- **Servidor** - Proceso que ofrece recursos y servicios a los clientes que lo solicitan (Back End). Según el servidor, tendremos una arquitectura diferente. Al tener programas y datos centralizados, se facilita la integridad y mantenimiento.
- **Cliente** - Proceso que solicita los servicios del servidor a través de una petición del usuario (Front End). Un proceso cliente interactúa con el usuario, construido con alguna herramienta que permita implementar interfaces gráficas (GUI)
- **Middleware** - Parte del software del sistema que se encarga del transporte de los mensajes entre cliente y servidor. Facilita la interconexión de sistemas heterogéneos; se ejecuta de ambos lados de la estructura.  
Permite independizar a los clientes de los servidores. Ofrece más control sobre el negocio, dado que permite obtener información desde diferentes orígenes (uniendo tecnologías y arquitecturas distintas)  
Sistemas débilmente acoplados, dado que interactúan mediante mensajes.

Las peticiones son generadas por un cliente (navegador, app móvil, o app desktop).

Las peticiones definen qué tipo de información se va a pedir y a dónde.

Como cliente y servidor no hablan el mismo lenguaje, aparece middleware. Es una capa de software

que se encarga de traducir los pedidos de forma que el servidor los comprenda.

Dentro del middleware, los DBMS, sistema de administración de bases de datos, que permiten crear, recuperar, actualizar y administrar datos. También API, interfaz de programación de aplicaciones, brindan un conjunto de subrutinas, funciones y procedimientos para ser utilizados por otro software. Una vez que la petición llega al servidor, este la procesa y prepara una respuesta (*response*). La *response* pasa por el mismo proceso que la petición (middleware y luego cliente).

Cuando el ciclo de request-response es masivo, Servidor lo mas protegido posible (condiciones optimas de temperatura, seguridad de acceso y energía ininterrumpida). Con este fin existen los Centros de Datos, o Datacenters; inmuebles construidos para albergar sistemas de información y comunicación, protegidos. Además, nos permitirían tener servicio en varios servidores, arquitectura tolerable a fallas.

El Sistema Operativo será el motor que permita aprovechar todos los recursos de este despliegue. OS conjunto de programas capaces de administrar recursos físicos del servidor, y los protocolos de ejecución de otro software. Es el director de orquesta. Es necesario utilizar OS de red.

## Características fundamentales SO para Servidores

- *Soporte de red* - Indispensable para poder brindar conectividad
- *Amplia compatibilidad hardware* - Permiten exprimir al máximo las características técnicas del hardware donde se ejecuta. Clave las actualizaciones y soporte importante de controladores (características avanzadas de gestión de disco rígido, etc.)
- *Seguridad* - Teniendo en cuenta el rol, clave la seguridad. No solo SO actualizado, sino que aplicación de políticas estrictas de acceso.
- *Tolerancia a fallos* - SO que en arquitectura permita tolerancia a fallos (generación de granja de servidores por ejemplo)

## SO específicos para Servers

Difieren de los SO domésticos por su:

- *Manejo del hardware* - Por ejemplo hasta 24TB de RAM, mientras que W10 hasta 6
- *Características soportadas* - Brinda ciertas funcionalidades nativos que otro tipo de SO no
- *Soporte* - Los SO de servidores cuentan con soportes específicos para dicho uso.

25% Unix vs. 75% Windows, teniendo dentro de Unix a GNU Linux, FreeBSD, macOSServer, y dentro de Windows a toda la familia de Windows Servers (2003 a 2019)

## Windows

Lanzado 1985, al como interfaz gráfica sobre MS-DOS. 88.14% del mercado, Windows Defender uno de los mejores antivirus, ideal gaming.

### Pros

- Integración: Casi todas las apps compatibles con W
- Soporte: Mucho, oficial y en foros/webs

- Facilidad de uso: Fácil, mayoría está familiarizado

#### Contras

- Seguridad: Al ser el más utilizado, más malware
- Costo: SO Costoso
- Tamaño: Algunas versiones ocupan mucho espacio en el disco

### **Mac**

1984, comenzó como interfaz gráfica. Segundo más utilizado mundialmente, basado en Unix.

#### Pros

- Estabilidad: Dada por la minuciosa optimización entre hardware y software, casi nunca fallos
- Seguridad: Menos vulnerabilidad ante malware
- Diseño: Estético y fácil de utilizar

#### Contras

- Costo: Equipo costoso
- Compatibilidad: A pesar de la tasa haber aumentado, aún muchas apps no compatibles
- No apto Gaming: No corre la mayoría de los últimos juegos

### **Linux**

GNU/Linux son SO basados en Unix cuyo núcleo, kernel, se lo conoce como Linux. Emplean múltiples componentes y herramientas del proyecto GNU: ambiente escritorio gráfico, editor de imágenes, biblioteca para lenguajes de programación, compiladores, etc.

286 distribuciones actualmente, software libre, usado por 1.34% del mercado, el 97% de las computadoras más rápidas del mundo utilizan Linux

#### Pros

- Costo: Cualquiera puede usar, modificar y redistribuir
- Seguridad: Pocos o inexistentes virus
- Estabilidad: Tanto que es ampliamente utilizado en servidores

#### Contras

- Complejidad: Difícil de entender y manejar, incluso para instalar programas
- Compatibilidad: Muchos programas no funcionan o necesitan parches/conversiones para funcionar
- Portabilidad: No es prioritaria para muchos desarrolladores

## **Servicios ofrecidos por los SO**

Servicio: Puerta de entrada ofrecido por SO para ingresar y consumirlo. Útil para ejecutar programas, manipular archivos y asignar recursos.

Algunos se ofrecen de manera nativa, y otros instalando un software específico para que trabajen de manera simultánea. por ejemplo:

- Servicio de publicación web
- Servicio de base de datos
- Servicio de correo electrónico
- Servicio de archivos
- Servicio de red
- Servicio de dominio

## Servicio de publicación web

Software se encarga de despachar contenido del sitio al usuario.

El proceso comienza desde el navegador, al ingresar una dirección en el comienza el ruedo: el sistema realiza búsqueda en servidores DNS para encontrar el servidor donde se aloja el sitio en cuestión. Cuando se ubica el servidor, el navegador solicita el contenido del sitio, el webserver procesa el pedido, y envía el contenido al navegador, resultando en la visualización.

Lo mas utilizados:

- *Apache* - Multiplataforma, open source, mas popular (por defecto en muchas dist. Linux)
- *Internet Information Services (IIS)* - Windows, propietario, en WServer desde NT3.5
- *Nginx* - Multiplataforma, open source, orientado alto rendimiento, uso exponencial desde 2004
- *LiteSpeed* - Linux, propietario, buena integración a tecnologías como Cpanel, de allí su uso en servidores cloud

## Servicio de base de datos

Servidor de base de datos, database server o RDBMS (Relational DataBase Management Systems) en caso de BBDD relacionales, aunque en actualidad se esta extendiendo el uso de BBDD no relacionales (No-SQL).

Tipo de software de servidor que permiten la organización de la información mediante tablas, indices y registros

El **servicio**, provee información a otras aplicaciones web o equipos/host (tal como se especifica en el modelo cliente servidor)

Teniendo usuario, contraseña y nombre del host, es posible realizar diferentes tareas (selección, actualización y borrado de datos) en las distintas bases de datos, dependiendo de privilegios.

Servicios de bases de datos:

- MySQL: Multiplataforma, relacional, líder en desarrollo web y aplicaciones (lenguajes como PHP, Python o Ruby)
- PostgreSQL: Multiplataforma, relacional, orientado a objetos. Se caracteriza por robustez y velocidad
- Microsoft SQL SERVER: Multiplataforma, relacional, popular entre usuarios de WServer, debido a compatibilidad nativa con lenguaje ASP/ASP.NET

compatibilidad nativa con lenguaje JSON / BSON (JSON)

- MongoDB: Multiplataforma, no relacional. No guarda datos en tablas, sino estructuras BSON (Similar JSON) dinámicas, haciendo el acceso rápido y sencillo

## Servicio de Correo Electrónico

El servidor de correo encargado de enviar y recibir mensajes de correo electrónico entre hosts, usuarios y servidores.

Una de las aplicaciones mas populares del protocolo TCP/IP, permite que nos comuniquemos en cuestión de segundos.

El servidor cuenta con dos componentes fundamentales:

- MTA (Mail Transport Agent) - Software encargado de transferir el email de un host (equipo informático) a otro.

Mas populares: Qmail, Exim, Postfix, Microsoft Exchange Server, Courier, Cyrus

- MDA (Mail Delivery Agent) - Recibe el correo de un MTA, y lo lleva al inbox de la casilla, que previamente se comunico con el servidor POP o IMAP.

Mas populares: Dovecot, Procpmail, Maildrop

## Servicio de Archivos

File server es un servidor central de un red que pone a disposición del cliente conectado un conjunto de archivos. Proporciona un lugar de almacenamiento centralizado.