



# 1º DAM/DAW    Sistemas Informáticos

---

U1. Sistemas Operativos. Software libre.

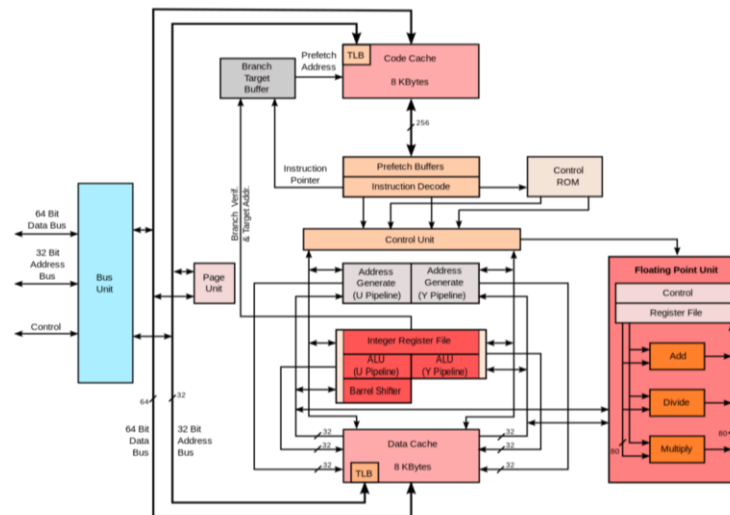
Esquema

## ¿Qué es un Sistema Operativo (S.O.)?

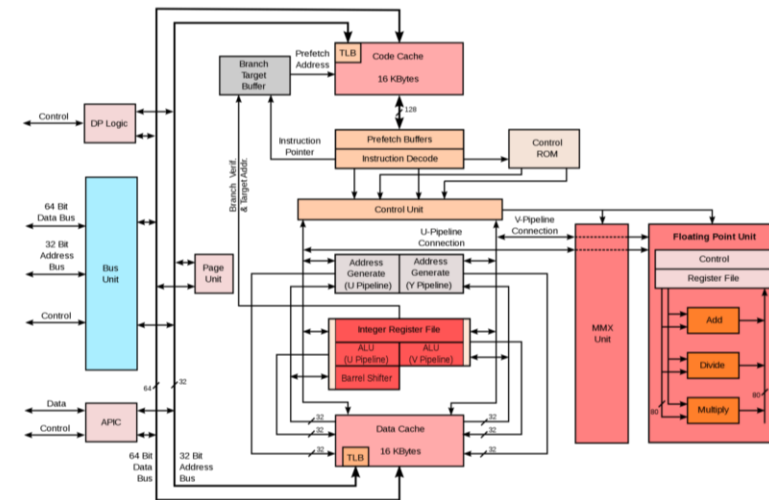
- Ordenador = máquina compleja.
- Permite que un usuario pueda usar el hardware de una forma adecuada y eficiente.



Arquitectura procesadores P5 (Intel Pentium original)

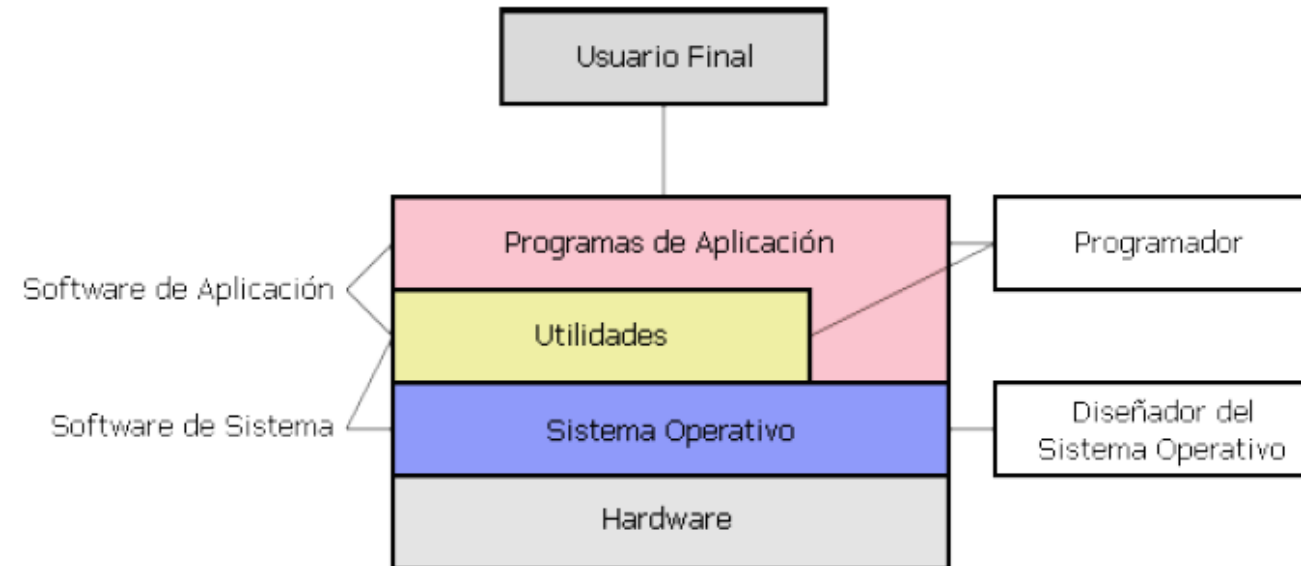


Modificaciones de la arquitectura P5 con MMX



## ¿Qué es un Sistema Operativo (S.O.)?

- Capa software de un dispositivo instalada directamente sobre el hardware.
- Controla los distintos elementos de un ordenador de una manera simplificada, adecuada, y eficiente.



## ¿Qué es un Sistema Operativo (S.O.)?

Puntos de vista:

- **Función que desempeña:** controla el **funcionamiento** del hardware y la **ejecución** de programas.
- **Visión desde el usuario:** ofrece un **mecanismo intuitivo** y guiado para interactuar con el hardware.
- **Gestor de recursos:** administra recursos hardware para obtener el mayor **rendimiento** posible.

Es el conjunto de **programas** que hace más **sencillo el uso del hardware** por parte del usuario, **administrando de una manera más segura** este hardware y **optimizando su rendimiento**.

## ¿Qué es un Sistema Operativo (S.O.)?

Ejemplos de S.O.:



## Componentes.

En un sistema operativo podemos encontrar los siguientes **componentes**:

- Núcleo o kernel
- Administrador de memoria
- Sistema de entrada/salida
- Administrador de ficheros
- Sistema de protección
- Interfaz de usuario

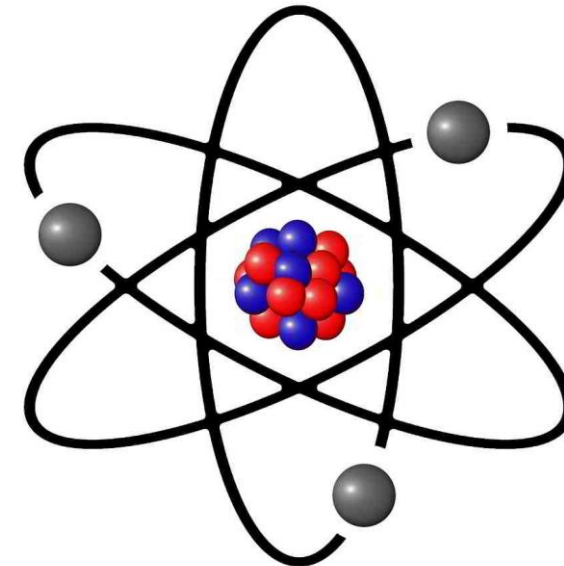


## Componentes. Núcleo o Kernel

- El **núcleo o kernel** es el componente **central** de un sistema operativo.

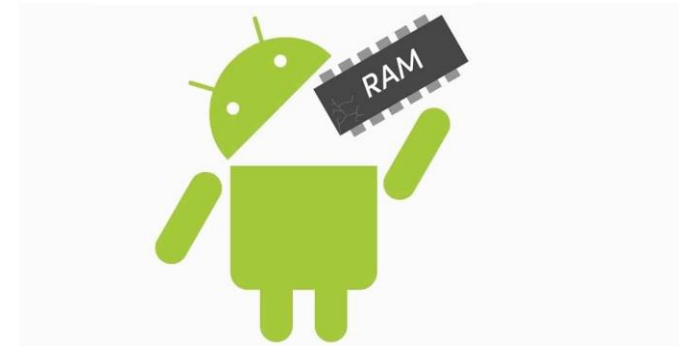
Funciones que realiza:

- Gestión de **interrupciones**.
- Asignación de **tareas** al procesador.
- **Comunicación** entre los distintos programas.



## Componentes. Administrador de memoria

- **Asigna porciones de la memoria principal (RAM)** a los distintos programas que la necesiten. Para ello:
  - **Ubica, reemplaza, carga y descarga** los procesos en memoria.
  - **Protege la memoria** de los accesos indeseados.
  - Permite **compartir zonas** de memoria.





## Componentes. Sistema de Entrada/Salida

- Coordina el **intercambio de información** entre el procesador y los periféricos.



## Componentes. Administrador de ficheros

- Gestiona la **estructura y seguridad de los datos**.
- Asegurar un **uso efectivo** de los medios de almacenamiento masivo.
- **Supervisa los privilegios de acceso** a la información.



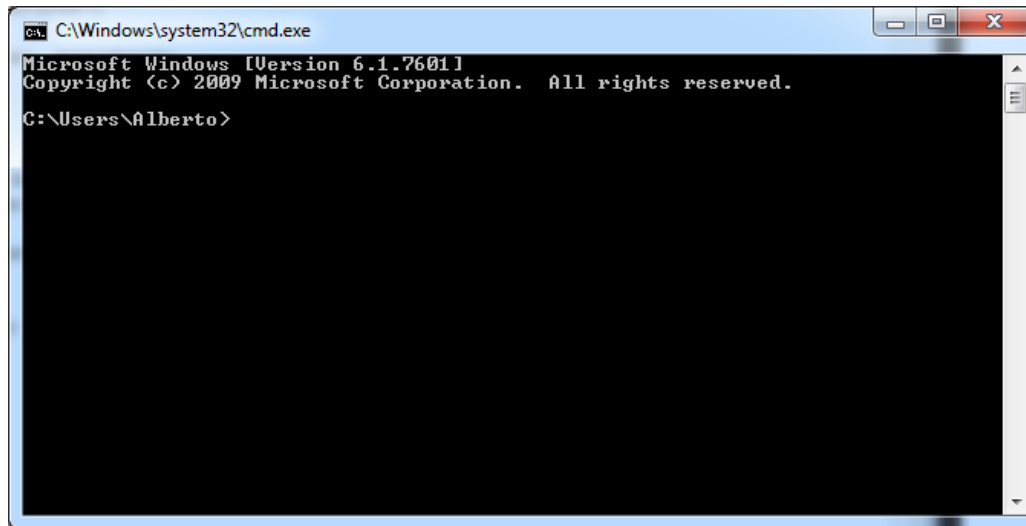
## Componentes. Sistema de protección

- Controla el **acceso a los programas y recursos del sistema**. Se encarga de:
  - Distinguir el **uso no autorizado**.
  - Especificar los **controles de seguridad**.
  - **Forzar** el uso de mecanismos de protección.



## Componentes. Interfaz de usuario

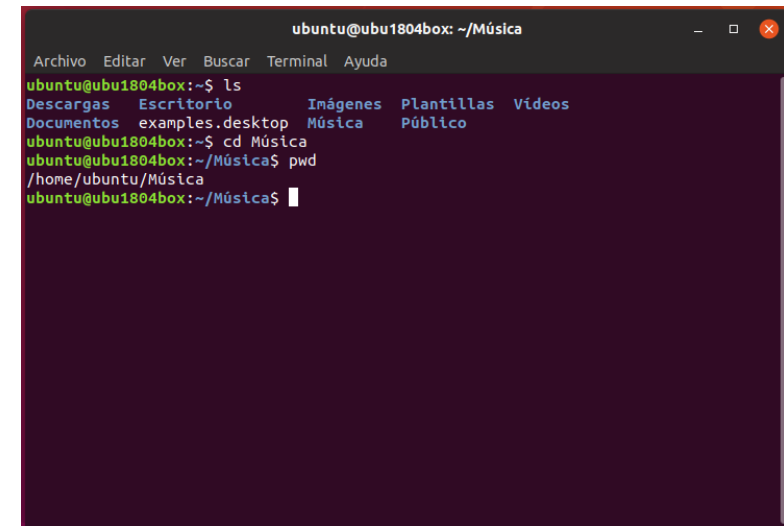
- Permite la **interacción entre el usuario y el dispositivo**.
- **Tipos** de interfaces de usuario:
  - Interfaz de **línea de comandos**: entorno textual. El usuario puede introducir comandos.



A screenshot of a Windows Command Prompt window. The title bar shows 'C:\Windows\system32\cmd.exe'. The window content displays the Microsoft Windows version 6.1.7601 copyright notice and the current directory 'C:\Users\Alberto>'.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Alberto>
```

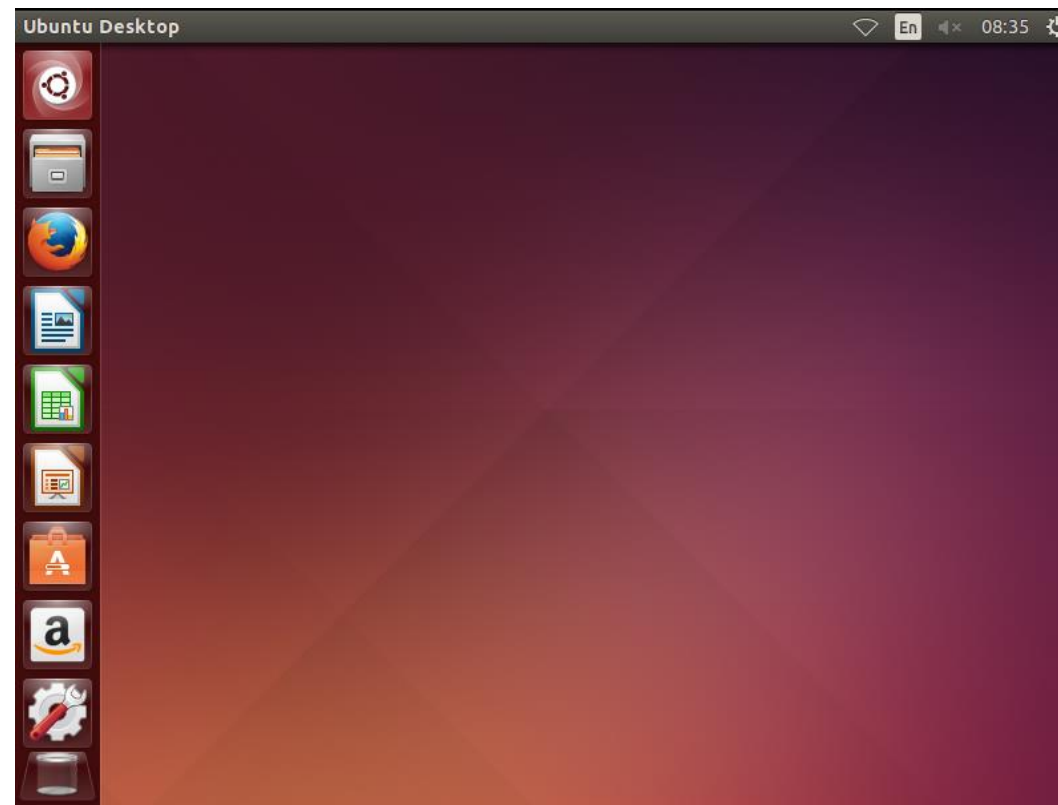


A screenshot of an Ubuntu terminal window. The title bar shows 'ubuntu@ubu1804box: ~/Música'. The window content shows the terminal output of 'ls' and 'cd Música' commands, followed by the 'pwd' command showing the current directory as '/home/ubuntu/Música'.

```
ubuntu@ubu1804box: ~/Música
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
ubuntu@ubu1804box:~$ ls
Descargas  Escritorio  Imágenes  Plantillas  Videos
Documentos  examples.desktop  Música  Público
ubuntu@ubu1804box:~$ cd Música
ubuntu@ubu1804box:~/Música$ pwd
/home/ubuntu/Música
ubuntu@ubu1804box:~/Música$
```

## Componentes. Interfaz de usuario

- **Interfaz gráfica de usuario (GUI):** interfaz visual. Uso de elementos gráficos.

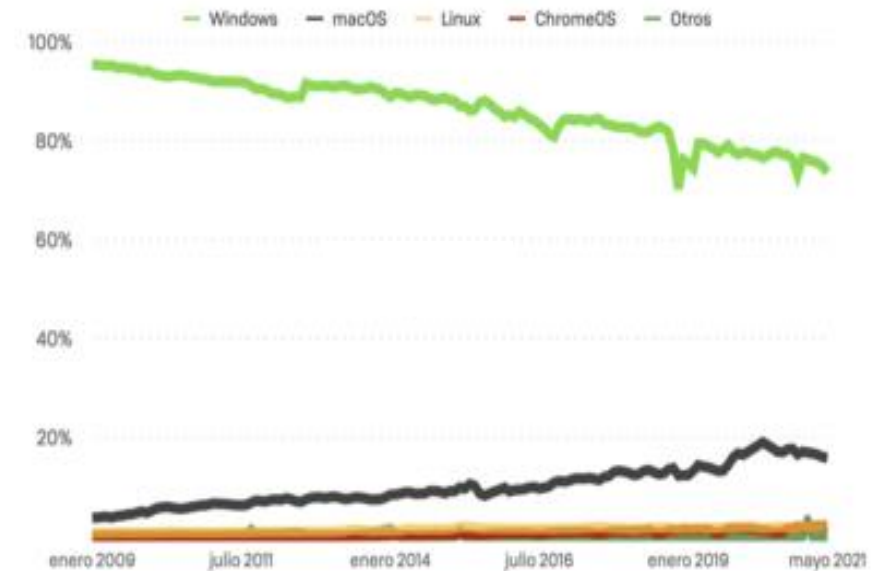


## Clasificación

- Podemos **clasificar** los distintos sistemas operativos según múltiples criterios:
  - **Monotarea o multitarea:** según el número de **programas** que pueden ejecutar **simultáneamente**.
  - **Monousuario o multiusuario:** según el número de **usuarios** que puedan trabajar **simultáneamente**.
  - **Centralizados o distribuidos:** según la **distribución de las tareas** del sistema en uno o varios equipos.

## Historia de Linux.

- Windows el más usado en PCs.
- En los últimos años mayor interés por el uso de sistemas operativos libres.
- Cada vez son más potentes, estables y competitivos frente a los sistemas propietarios.
- Existe un extenso conjunto de comunidades, distribuciones.



Evolución de la cuota de mercado de sistemas operativos de escritorio desde 2009 hasta 2021. Datos: Stat Counter Global Stats.

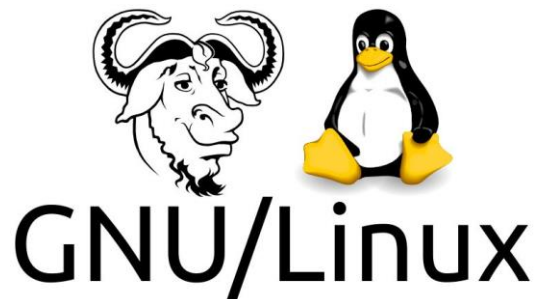
## Historia de Linux.

- La historia del **software libre** arranca en el **1983**, con Richard Stallman y el **proyecto GNU** (es un acrónimo recursivo: “GNU is NotUnix”).
- El **Objetivo** principal fue **crear un S.O. libre**.
- El sistema GNU está **basado en** el sistema **UNIX**.
- Existen utilidades GNU que mejoraron el sistema UNIX.
- **Surge un problema**, en GNU se estaban desarrollando partes desde cero y, otras se estaban adaptando desde UNIX, pero **quedaba pendiente la parte del kernel o núcleo del S.O.**

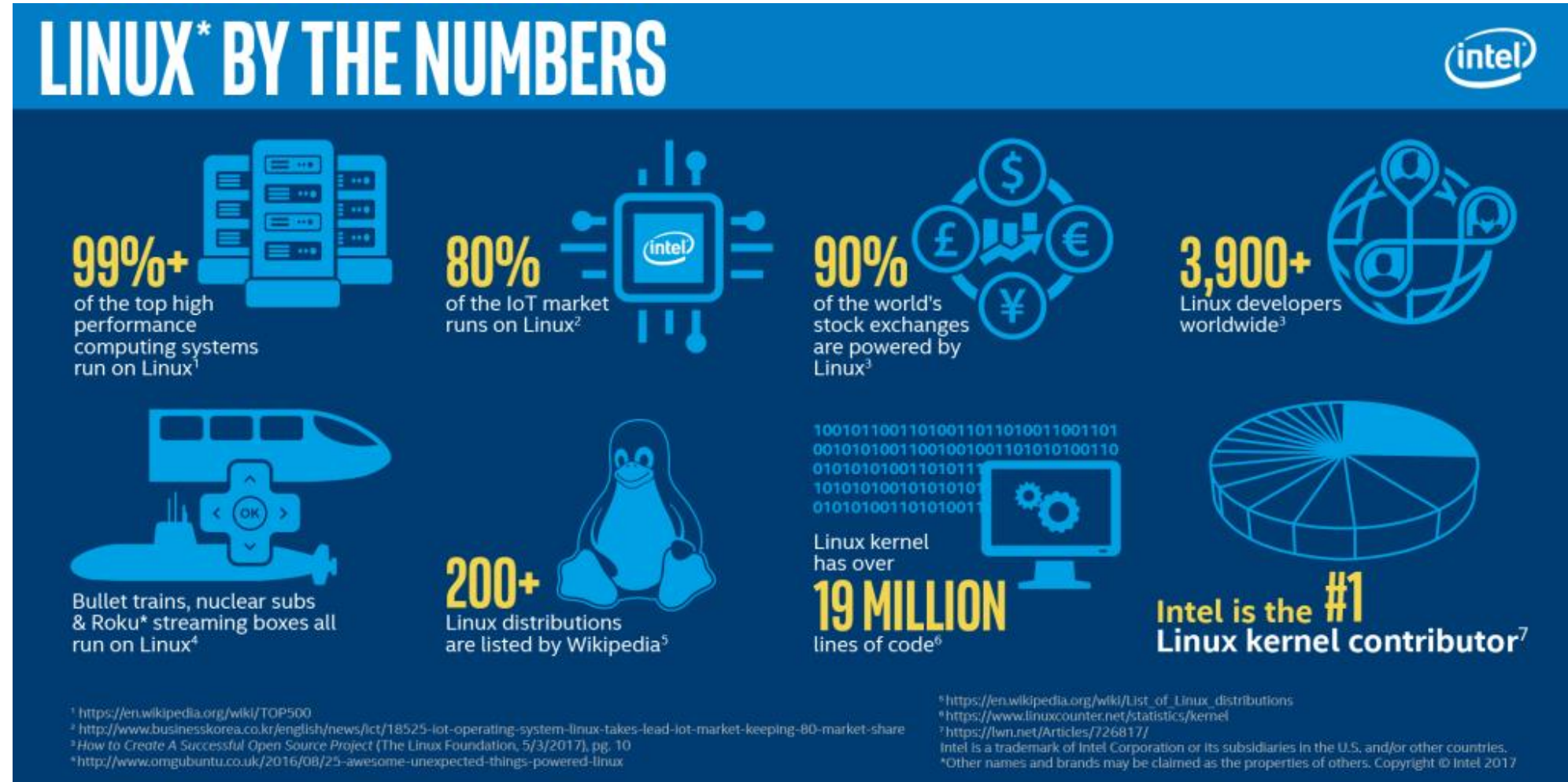


## Historia de Linux.

- Se intentó en primer lugar con una versión llamada **Trix**, y después con un sistema llamado **Hurd**, pero finalmente el proyecto quedó **aparcado**.
- Interviene la figura de **Linus Torvalds**, por el año 1991, creando un kernel de sistema operativo llamado Linux, completamente compatible con UNIX y bajo la licencia GPL (General Public License of GNU).
- Finalmente, se dispuso de un sistema operativo completo llamado **GNU/Linux**.



## Historia de Linux.



## Licencias.

- La licencia es el **contrato entre el autor o propietario** de un software **y el usuario**.
- El copyright es una forma de protección proporcionada por las leyes vigentes.



## Licencias. Clasificación

Atendiendo a la **disponibilidad de uso**:

- **Software privativo:** software que, para utilizarlo, copiarlo, modificarlo o distribuirlo, se requiere el permiso de su autor.
- **Software libre:** responde a una serie de principios de libertad que defendió y propició Stallman.



## Licencias. Clasificación

Atendiendo a **cómo se realice la distribución**:

- Software de **pago**.
- Software **gratuito**:
  - **Freeware**: se puede redistribuir, pero no se puede modificar el código fuente.
  - **Shareware**: igual al anterior, pero además, transcurrido un período de tiempo suele ser necesario pagar una licencia para continuar utilizándolo.
  - **Adware**: se puede redistribuir, pero incluye publicidad durante la instalación y/o durante su uso.
  - **Software de dominio público**: es un software que no requiere licencia. Los derechos de explotación son para toda la humanidad y éste pertenece a todos por igual.

## Licencias. Clasificación

Atendiendo a los **términos** de la licencia:

- **En propiedad:** quien la compra se convierte en titular indefinido de la misma.
- **En alquiler:** se autoriza el uso de lo contratado por un plazo establecido, habitualmente a nivel temporal.



## Licencias. Resumen

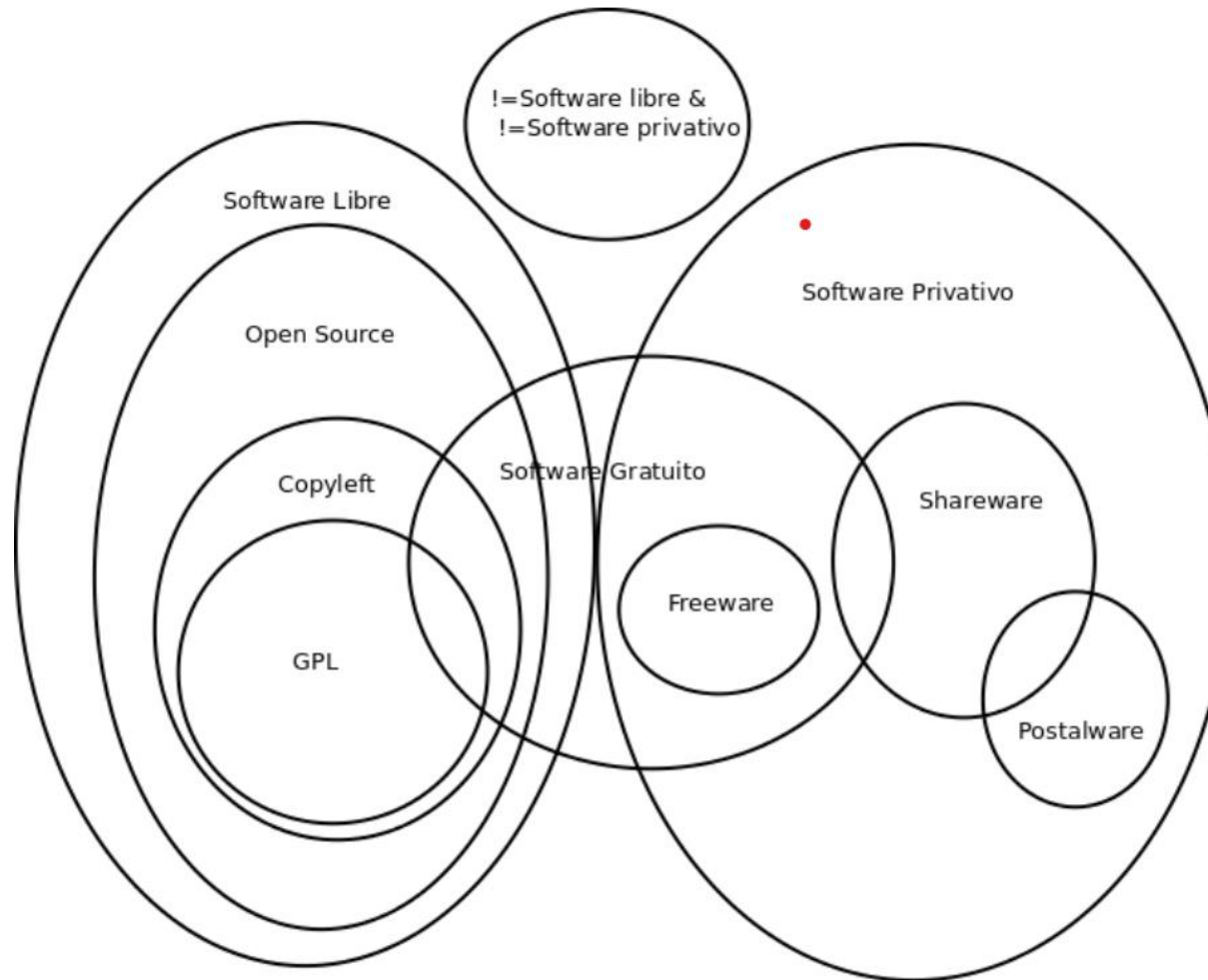
El hecho de que **un software sea libre no significa que tenga que ser gratuito, ni viceversa.**

Se puede hacer negocios con el software libre:

- Distribución de pago.
- Formación.
- Modificaciones particulares.
- Soporte técnico.
- ...



## Licencias. Resumen





## Filosofía del software libre.

La idea de software libre responde a una serie de **principios de libertad** otorgada a los usuarios.

Para ello, se otorgan cuatro libertades básicas a los usuarios:

- **Libertad 0:** es la libertad de **ejecutar** el programa como se desee.
- **Libertad 1:** es la libertad de **estudiar** el código fuente y **cambiarlo** para adaptarlo a las necesidades particulares.
- **Libertad 2:** es la libertad de **ayudar** a otros. Es la libertad para hacer copias y distribuirlos como se desee.
- **Libertad 3:** es la libertad de **contribuir** a la comunidad. Es la libertad para hacer copias de las versiones modificadas y distribuirlos en beneficio de todos.

## Filosofía del software libre.

- La idea del **software libre** es distribuir un **código**, para que cualquier persona lo pueda ver y modificar, con la condición de que **los cambios realizados continúen siendo código libre**.
- Para llevar a cabo esta condición en el proyecto cooperativo, Stallman creó el concepto de **copyleft** en contraposición al copyright.



## Filosofía del software libre.

- Es habitual entremezclar el concepto de **software libre** con el de **código abierto**.
- Aparentemente, son conceptos similares, pero hay un **importante matiz que los diferencia**.



**V/S**



## Sistemas operativos de código abierto.

- El **código abierto** implica tener **acceso al código fuente**, aunque es posible que no se **permita su modificación y distribución**.
- Digamos que el software de código abierto no tiene porqué cumplir las libertades 1, 2 y 3 del código libre.



## Ejemplos de distribuciones Linux.

Como ejemplos de distribuciones Linux, podríamos destacar las siguientes:

- **Debian:** software libre con licencia GPL, desarrollado por un conjunto de programadores.
- **Fedora:** software libre con licencia GPL desarrollado por la comunidad Fedora, patrocinada por Red Hat.
- **Red Hat:** distribución de la empresa Red Hat. Código abierto (FOSS, Free OpenSource Software). Contribuye en proyectos de software libre.
- **Suse:** distribución de la empresa Novell. Hay una licencia libre (OpenSuse) y una privativa (Professional).
- **Ubuntu:** distribución basada en Debian y enfocada al usuario medio. Combina software libre y abierto.
- **Mint:** distribución basada en Debian. Combina software libre y abierto.
- **Mandriva:** distribución de la empresa del mismo nombre.



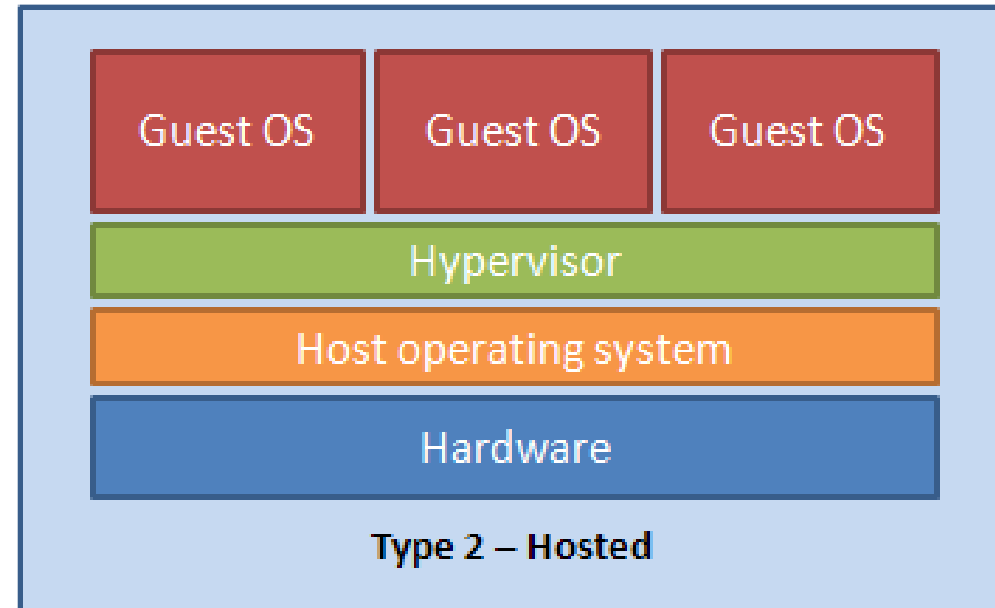
## Virtualización.

- **Virtualización:** consiste en abstraer los recursos de una máquina física, para poder ejecutar en ella una o varias máquinas virtuales.
- **Máquina física (host, anfitrión) VS Máquina virtual (guest, huésped)**
- **Comparten** los mismos **recursos** (CPU, memoria, disco, etc.).
- Se repartirán el hardware de una manera **transparente al usuario**.
- **Encapsulación:** las máquinas virtuales se ejecutan con independencia del hardware físico que haya por debajo.



## Hypervisor.

- **Hypervisor:** capa software que permite la virtualización en las máquinas virtuales de sistema.
- Ejemplos: **VirtualBox o VmWare.**



## Instalación de Virtualbox.

- Virtualbox es un software hipervisor para virtualización.
- Actualmente pertenece a la empresa Oracle.
- Entre los sistemas operativos que podemos instalar en VirtualBox tenemos sistemas GNU/Linux, Mac OS X, MS Windows o Solaris.
- Web de Virtualbox: <https://www.virtualbox.org/>



VirtualBox



## Instalación de Ubuntu.

**Descarga última versión Ubuntu Desktop:**

<https://ubuntu.com/download/desktop>

**Distintas versiones de Ubuntu:**

<https://releases.ubuntu.com/>

<https://wiki.ubuntu.com/Releases>



**Instalación mediante Virtualbox como MV:**

<https://ubuntu.com/tutorials/how-to-run-ubuntu-desktop-on-a-virtual-machine-using-virtualbox#2-create-a-new-virtual-machine>