Fundamentos de los Sistemas Operativos (FSO)

Departamento de Informática de Sistemas y Computadoras (DISCA)
Universitat Politècnica de València

Bloque Temático 1: Introducción

Unidad Temática 1

Concepto de Sistema Operativo

fSO

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DISCA

Objetivo

- Presentar el **concepto de Sistema Operativo**
- Describir las **funciones** que debe llevar a cabo cualquier **Sistema Operativo** actual
- Comprender las características propias de los sistemas que han ido incorporándose durante su evolución para entender los servicios que debe proporcionar

- Bibliografía
  - A. Silberschatz, P. B. Galvin. "Sistemas Operativos". 7ª ed. Capítulos. 1 y 2
  - Wikipedia y otras fuentes de Internet
  - A Brief History of Computing - Operating Systems https://trillian.randomstuff.org.uk/~stephen/history/timeline-OS.html
  - Timeline: 40 years of OS milestones http://www.computerworld.com/article/2531905/operating-systems/timeline--40-years-of-os-milestones.html

- **Concepto de Sistema Operativo**
- Estructura del Sistema Operativo
- Utilización de CPU
- Evolución de los Sistemas Operativos

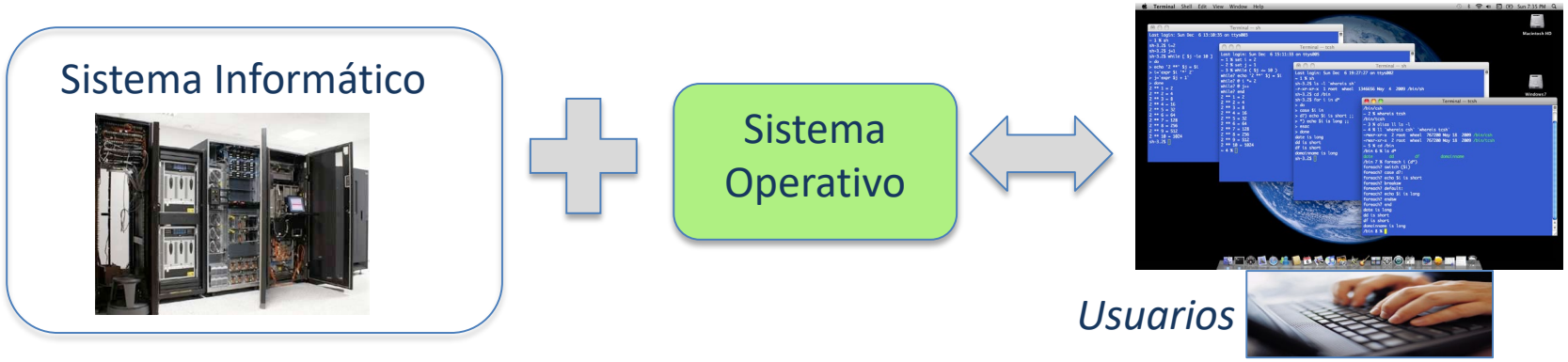
Nomenclatura:	
SO	Sistema operativo
E/S	Entrada /Salida
CPU	Unidad central de procesos
API	Interfaz de programación de aplicaciones (Application Programming Interface)
HAL	Capa de abstracción del hardware (Hardware abstraction layer)
UNIX	Sistema operativo portable, multitarea y multiusuario
Linux	Núcleo libre de SO basado en UNIX

Concepto de Sistema Operativo

- Sistema Informático:
  - puede definirse como el conjunto de **elementos hardware,** organizados mediante una determinada "Arquitectura", que conforman un dispositivo de computación
  - Problemática
    - El **manejo** directo de estos elementos **hardware es complejo** y requiere conocimientos específicos para cada dispositivo
    - **Es necesario establecer criterios** de explotación que **optimicen** el uso de las capacidades del hardware

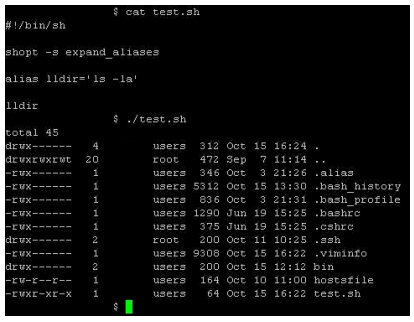
Concepto de Sistema Operativo

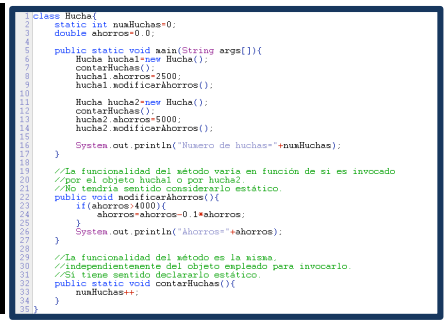
- Definición
  - **Un sistema operativo (SO) es un conjunto de programas (software)** que facilita la explotación de los Sistemas Informáticos ofreciendo al usuario la imagen de que está trabajando con una máquina sencilla (principio de embellecimiento)
- Finalidad
  - Crear un **entorno** cómodo y eficiente **para ejecutar programas**
- Objetivos: accesibilidad, comodidad, eficiencia, seguridad, portabilidad
  - Actuar de **intermediario** entre usuario y hardware
  - Garantizar el **funcionamiento correcto** del computador
  - **Facilitar** la tarea de creación de aplicaciones
  - Administrar **eficientemente** los recursos de la máquina



Concepto de Sistema Operativo

- Un sistema operativo debe **proporcionar servicios** a los diferentes tipos de **usuarios** de la máquina
- Tipos de usuarios
  - Usuario de aplicaciones / órdenes
  - Usuario programador /diseñador de aplicaciones
  - Usuario diseñador /implementador del sistema operativo
  - Administrador del sistema

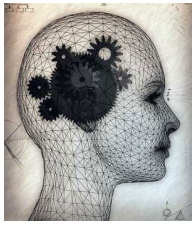




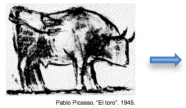
```
class Hucha{
    static int numHuchas=0;
    double ahorros=0.0;

    public static void main(String args[]){
        Hucha hucha1=new Hucha();
        contarHuchas();
        hucha1.ahorros=2500;
        hucha1.modificarAhorros();

        Hucha hucha2=new Hucha();
        contarHuchas();
        hucha2.ahorros=5000;
        hucha2.modificarAhorros();

        System.out.println("Numero de huchas="+numHuchas);
    }

    //La funcionalidad del método varia en función de si es invocado
    //por el objeto hucha1 o por hucha2.
    //No tendria sentido considerarlo estático.
    public void modificarAhorros(){
        if(ahorros>4000){
            ahorros=ahorros-0.1*ahorros;
        }
        System.out.println("Ahorros="+ahorros);
    }

    //La funcionalidad del método es la misma.
    //independientemente del objeto empleado para invocarlo.
    //Si tiene sentido declararlo estático.
    public static void contarHuchas(){
        numHuchas++;
    }
}
```

Concepto de Sistema Operativo

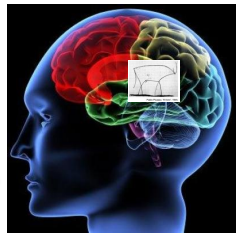
- Perspectivas
  - El **sistema operativo abstrae y gestiona** todos los recursos de la máquina: tanto de sus componentes hardware que forman el sistema informático como de los elementos software



    - **Perspectiva de sistema**: Administrador de recursos y protección
      - Componentes del SO y sus interconexiones
    - **Perspectiva de usuario**: Abstracción de los recursos orientada a facilitar su uso. Máquina extendida
      - Servicios que proporciona
      - Interfaces disponibles para usuarios y programadores



abstracción



Concepto de Sistema Operativo

- Funciones
  - Proporcionar **interfaces amigables** para usuarios y programadores
  - Ofrecer al programador una **abstracción del hardware** facilitando la accesibilidad
  - Ofrecer un conjunto de servicios en la forma de **llamadas al sistema**
  - **Gestionar recursos**: Se encarga de decidir qué programa podrá hacer uso de un dispositivo de hardware y durante cuánto tiempo
    - Gestión de procesos, gestión de memoria, gestión de archivos, gestión de E/S
  - **Protección y Seguridad:** Controlar los accesos a los recursos y evitar los accesos no autorizados a éstos

- Dispositivos con sistema operativo

Server Pool

Computer Server

Personal Computer

Router

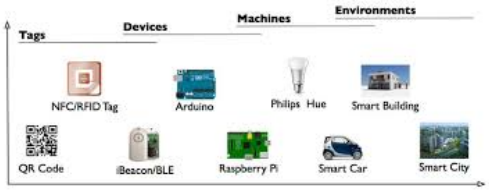
Tablet

Smart phone

Video Console

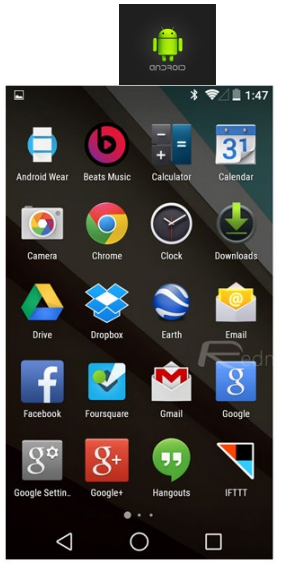
Smart TV

Smart Devices & IoT



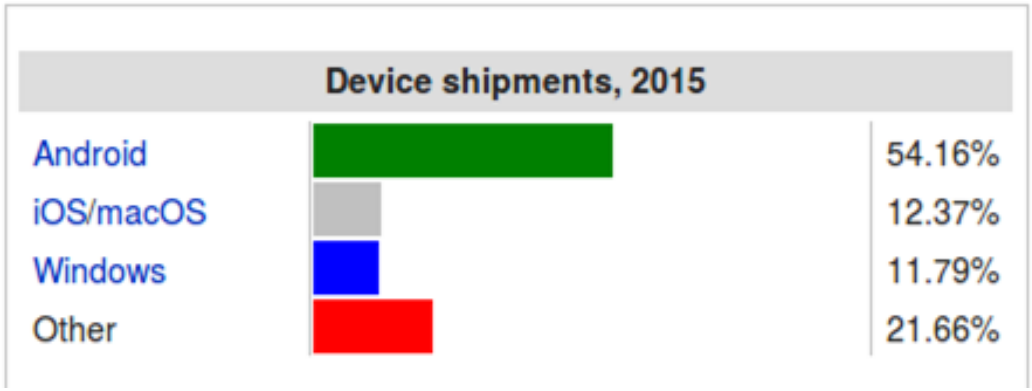
Fuente: https://www.slideshare.net/waltmaclay/trends-in-sensors-wearable-devices-and-iotiot

- Sistemas Operativos actuales

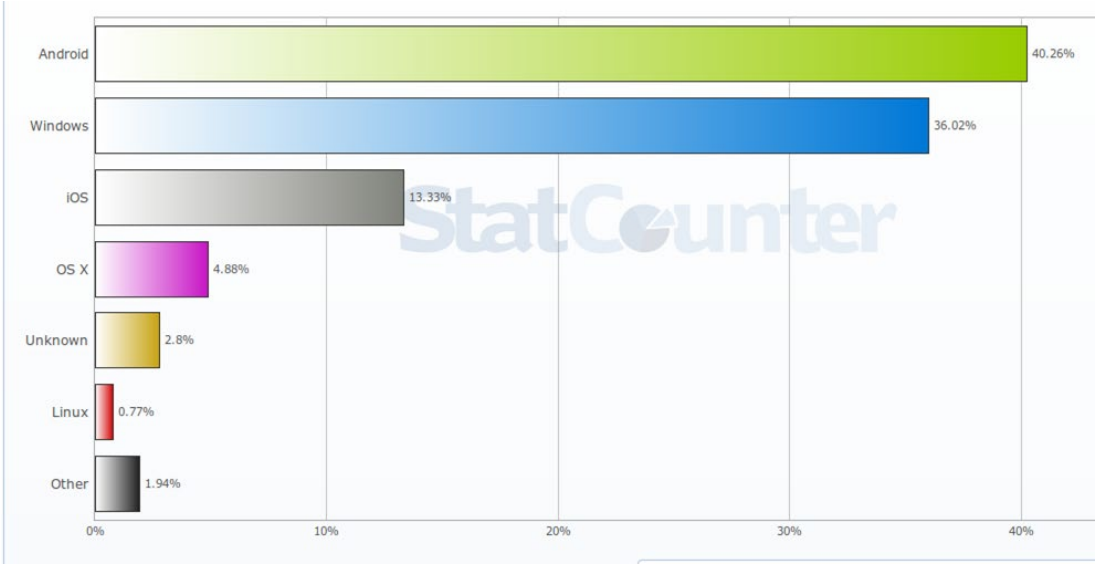


¿Sistemas operativos "más usados"?

- Depende de la plataforma
  - *Gartner* [1] (2015)
    - *Smartphones, tablets, laptops and PCs together*



  - *Statcounter* [2] (2017)



Fuente:
[1] Wikipedia <https://en.m.wikipedia.org/wiki/Usage_share_of_operating_systems>
[2] Statcounter. Operating System Market Share Worldwide. Junio 2017. <http://gs.statcounter.com/#desktop-os-ww-monthly-201508-201508-bar>

Caracterización de los SO “más usados”

- Android
  - *Android Runtime* + *kernel GNU/Linux*
  - *Apache License 2.0* + GNU GPL v2
  - “Android Terminal Emulator", …
- IOS/macOS (Mac OS X, OS X)
  - *Darwin* + *kernel XNU*
  - *Closed source* (*with open source components*) ← *NeXTSTEP, BSD, FreeBSD, Mach, …*
  - “Terminal” (*bash*)
- Windows
  - *Universal App Platform* + *kernel Windows NT*
  - *Closed / shared source*
  - “PowerShell” = “Windows PowerShell” (privativo) + “PowerShell Core” (abierto → GitHub)

Escenarios de trabajo actualmente

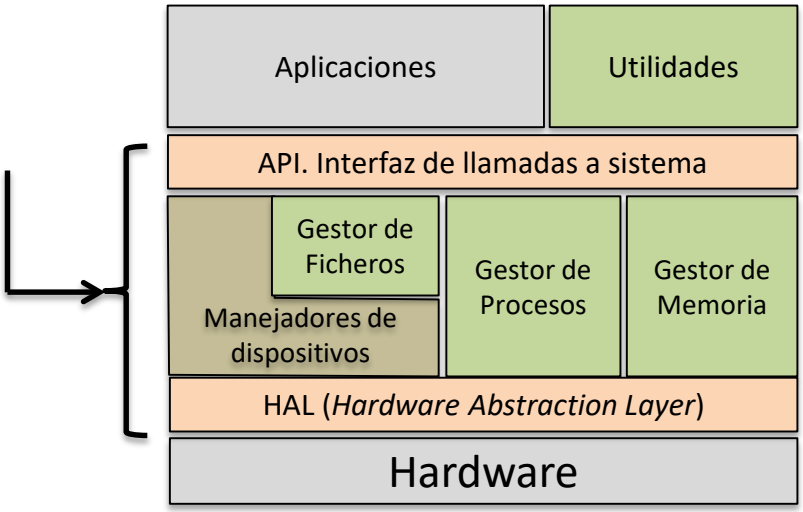
Category	Linux	Unix and Unix-like	Windows	In-house	Other
Desktop, laptop (excluding Android and Chrome OS)	2.18% (Ubuntu, etc.)	6.43% (macOS)	91.39% (10, 8.1, 7, Vista)		
Smartphone, tablet	68.31% (Android)	23.35% (iOS)	1.25% (Windows 10 Mobile, Windows Phone 8.1 and older)		9.86%
Server (web)	66.6% (Ubuntu 35.8%, Debian 31.9%, CentOS 20.6%, Red Hat 3.3%, Gentoo 2.7%, Fedora 0.9%)	1% (BSD)	33% (Windows Server 2016, W2K12, W2K8)		
Supercomputer	99.79% (Custom)	0.21%			
Mainframe	28% (SLES, RHEL)	72% (z/OS) UNIX System Services			
Gaming console, Handheld game console (7th & 8th generation only)		34.1% (PS4, PS3, Vita, PSP) ?? PS4 Obis OS (FreeBSD) ??% N Switch (FreeBSD)	16.36% (Xbox One, Xbox 360)	49.54% (Wii U, Wii, 3DS, DS)	0%
Embedded (automotive, avionics, health, medical equipment, consumer electronics, intelligent homes, telecommunications)	29.44% (Android plus other non-Android Linux)	4.29% (QNX)	11.65% (WCE 7)	13.5%	41.1%

Fuente: datos propios y Wikipedia <https://en.m.wikipedia.org/wiki/Usage_share_of_operating_systems>

- Concepto de Sistema Operativo
- **Estructura del Sistema Operativo**
- Utilización de CPU
- Evolución de los Sistemas Operativos

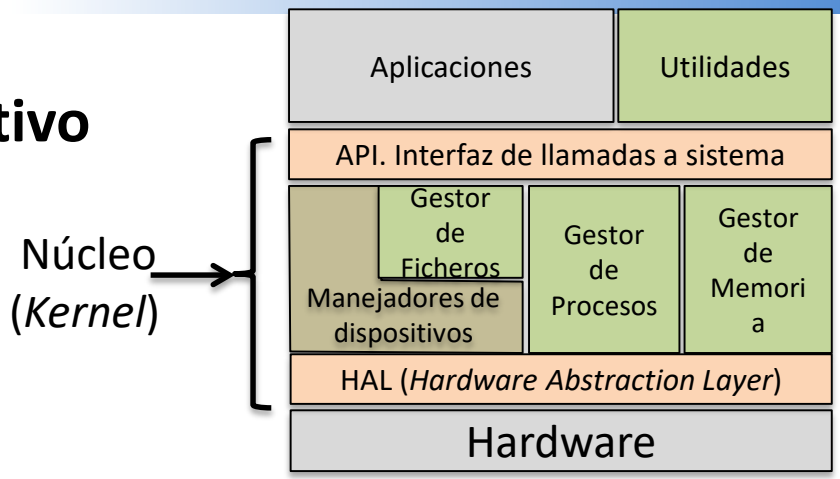
Nomenclatura:	
SO	Sistema operativo
E/S	Entrada /Salida
CPU	Unidad central de procesos
API	Interfaz de programación de aplicaciones (Application Programming Interface)
HAL	Capa de abstracción del hardware (Hardware abstraction layer)
UNIX	Sistema operativo portable, multitarea y multiusuario
Linux	Núcleo libre de SO basado en UNIX

- Estructura de un SO
  - **Componentes de un SO**
    - Gestor de archivos
    - Gestor de memoria
    - Gestor de procesos
    - Manejadores de dispositivos
  - **Utilidades de sistema**
    - Extienden el SO
    - Proporcionan herramientas para realizar actividades fundamentales que no están incluidas en el núcleo del SO
      - Intérprete de órdenes, Editor, Compilador, Interfaz gráfico, Monitorización, Mantenimiento, Administración…



Estructura del Sistema Operativo

- **Núcleo (*Kernel*) del sistema operativo**
  - Programa individual que siempre está cargado en memoria principal y que está permanentemente listo para ofrecer sus servicios



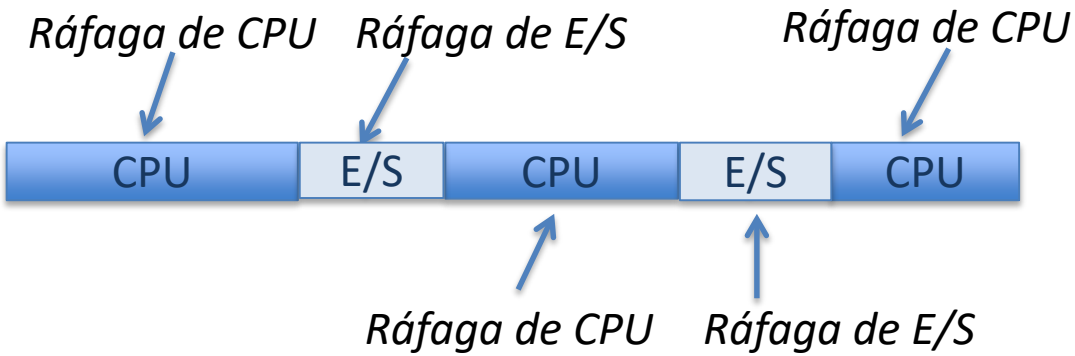
- Arquitectura del Núcleo
  - **Micronúcleo**: proporciona sólo las abstracciones básicas del hardware y los servicios mínimos.
    - Las políticas de uso de recursos se implementan como "servidores" que corren en espacio de usuario. Se ha discutido mucho sobre sus problemas de eficiencia.
    - Ejemplos: Mach, QNX
  - **Monolítico**: Todos los componentes del núcleo están en el mismo espacio de direccionamiento.
    - Un único programa contiene todas las funcionalidades (se ha de recompilar cada vez)
    - Ejemplo: Linux
  - **Híbrido**: Es un micronúcleo modificado que incluye componentes no esenciales cuya velocidad de ejecución es crítica
    - Ejemplos: Windows NT, XNU (MacOSX)
  - ….

- Concepto de Sistema Operativo
- Estructura del Sistema Operativo
- **Utilización de CPU**
- Evolución de los Sistemas Operativos

Nomenclatura:	
SO	Sistema operativo
E/S	Entrada /Salida
CPU	Unidad central de procesos
API	Interfaz de programación de aplicaciones (Application Programming Interface)
HAL	Capa de abstracción del hardware (Hardware abstraction layer)
UNIX	Sistema operativo portable, multitarea y multiusuario
Linux	Núcleo libre de SO basado en UNIX

Utilización de CPU

- **Carga de un sistema**
  - La carga de un Sistema Informático consiste en el **conjunto de programas que ha de ejecutar**
    - Los programas informáticos se modelan como **secuencias de operaciones**
    - Las operaciones pueden realizarse en la **CPU** o en un dispositivo de **Entrada/Salida**
  - De una forma esquematizada los programas se representan como una alternancia de ráfagas de CPU y ráfagas de E/S
    - Ráfaga de CPU→ intervalo de tiempo que se necesita para realizar el conjunto de operaciones consecutivas de CPU de un programa
    - Ráfaga de E/S→ intervalo de tiempo que se necesita para realizar el conjunto de operaciones consecutivas de E/S de un programa



- **Programas limitados por CPU o E/S**

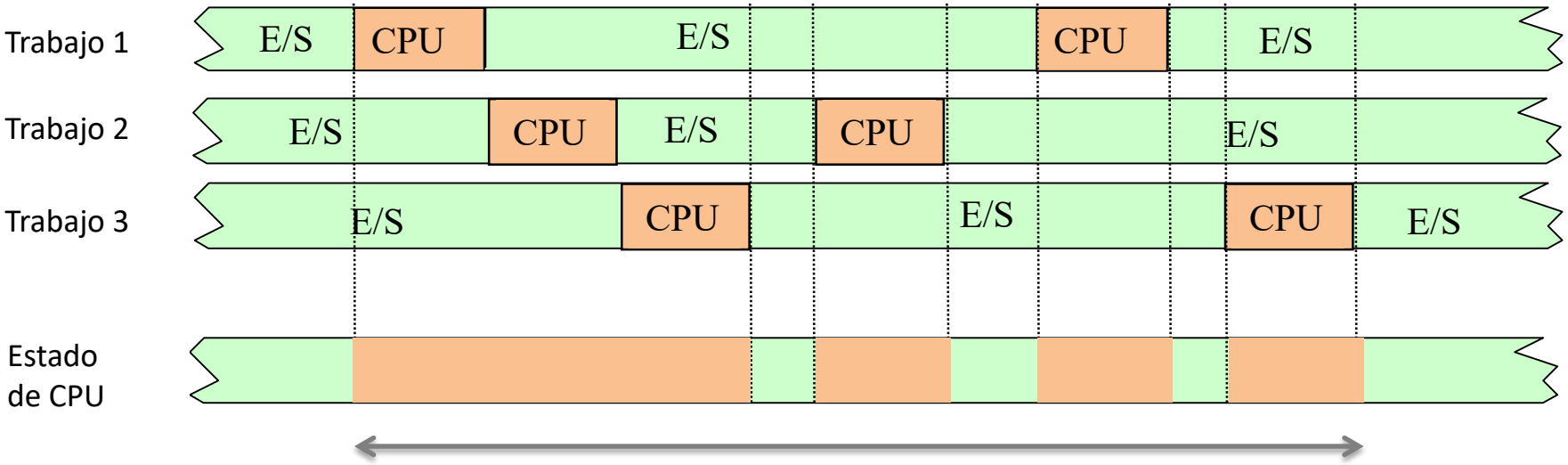
Utilización de CPU

- **Concepto de Utilización de CPU**
  - Históricamente las CPU han tenido un elevado coste
  - Los sistemas operativos deben trabajar para aumentar el uso de la CPU y evitar que se encuentre desocupada
  - **Utilización de la CPU**: Porción de tiempo que realmente se encuentra la CPU ocupada respecto al tiempo total que la tarea o tareas se encuentra/n en el sistema hasta ser finalizada/s

$$\text{Utilización\_CPU} = \frac{\text{Tiempo\_CPU\_ocupada}}{\text{Tiempo\_total}}$$

Utilización de CPU

- Concepto de Multiprogramación
  - **Alternancia entre procesos** en el uso de la CPU
    - Cuando un proceso se bloquea, esperando llevar a cabo una operación de E/S, en la CPU se ejecutan instrucciones de otro proceso
    - Se realiza un **"cambio de contexto"** cuando se invoca a una operación de E/S
  - Aumenta la utilización de CPU
  - Aumenta el rendimiento del sistema: finalizan más trabajos en menos tiempo

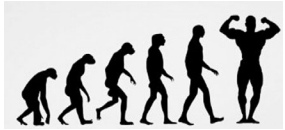


- Concepto de Sistema Operativo
- Estructura del Sistema Operativo
- Utilización de CPU
- **Evolución de los Sistemas Operativos**

Nomenclatura:	
SO	Sistema operativo
E/S	Entrada /Salida
CPU	Unidad central de procesos
API	Interfaz de programación de aplicaciones (Application Programming Interface)
HAL	Capa de abstracción del hardware (Hardware abstraction layer)
UNIX	Sistema operativo portable, multitarea y multiusuario
Linux	Núcleo libre de SO basado en UNIX

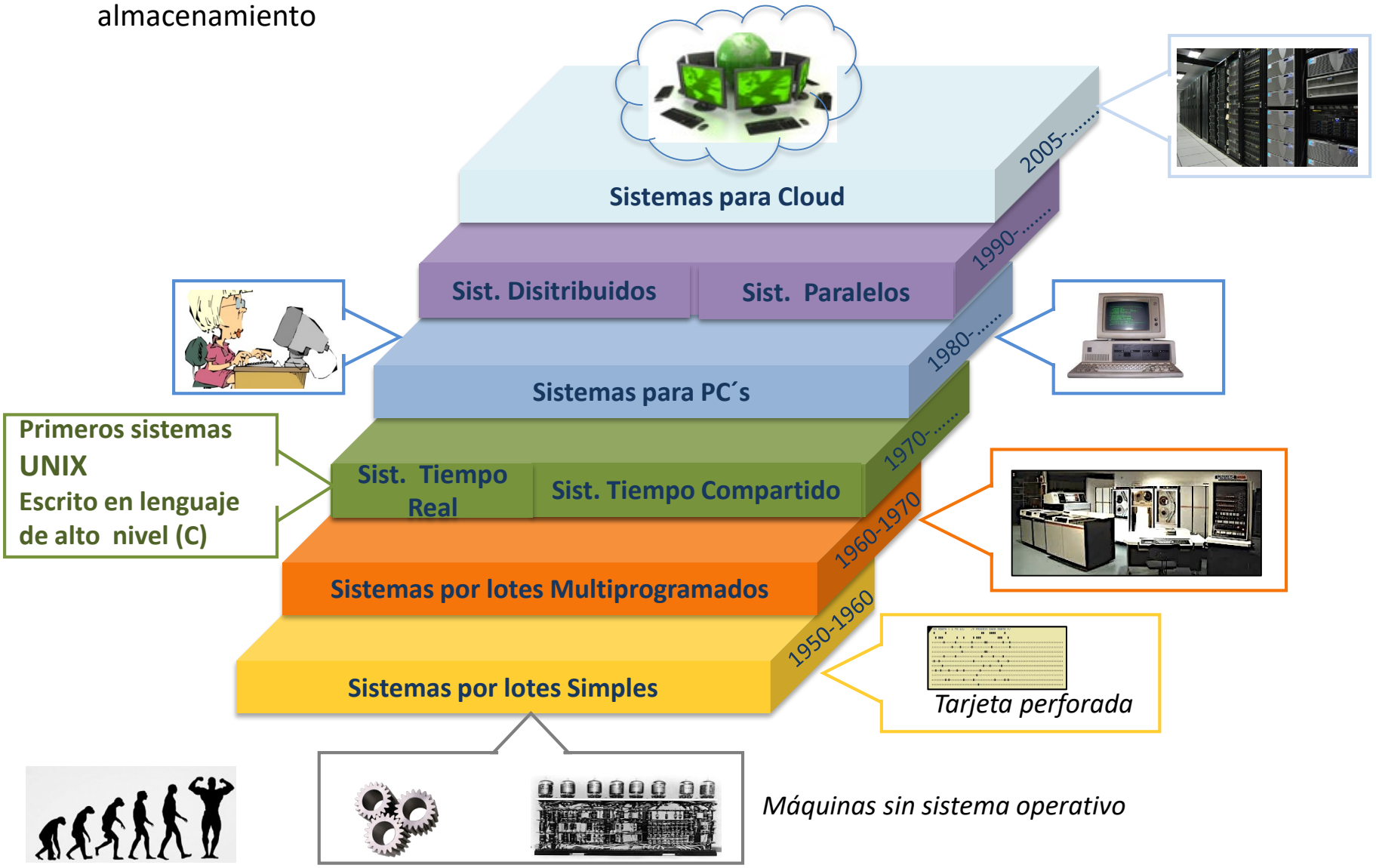
- Capacidades de los sistemas operativos

Monusario: El sistema operativo sólo puede trabajar con un único usuario a la vez	Soporte de Usuarios	**Multiusario**: El sistema operativo puede trabajar con varios usuario a la vez
Sin interacción directa usuario-máquina: Operadores profesionales trabajan directamente con la máquina	Soporte de Interacción directa con usuario	**Interacción directa usuario-máquina:** Los usuarios escriben órdenes y esperan respuestas
Modo carácter: La interfaz con los usuarios se realiza mediante la escritura de comandos	Soporte Interfaz de Usuario (UI)	**Modo gráfico**: Capacidad para soportar una interfaz gráfica con los usuarios (*GUI*)
Monotarea: El sistema operativo procesa tareas de forma secuencial. Hasta que una tarea no termina no procesa otra	Soporte de tareas	**Multitarea**: El sistema operativo permite que varias tareas sean procesadas de forma solapada en el tiempo
Monoprocesador: El sistema operativo sólo puede trabajar con un único procesador a la vez	Soporte de Procesadores	**Multiprocesador**: El sistema operativo puede trabajar con varios procesadores a la vez



Evolución de los Sistemas Operativos

- La evolución de los sistemas operativos ha estado y está determinada por las **innovaciones tecnológicas** en la arquitectura de los computadores, en las comunicaciones y en los medios de almacenamiento



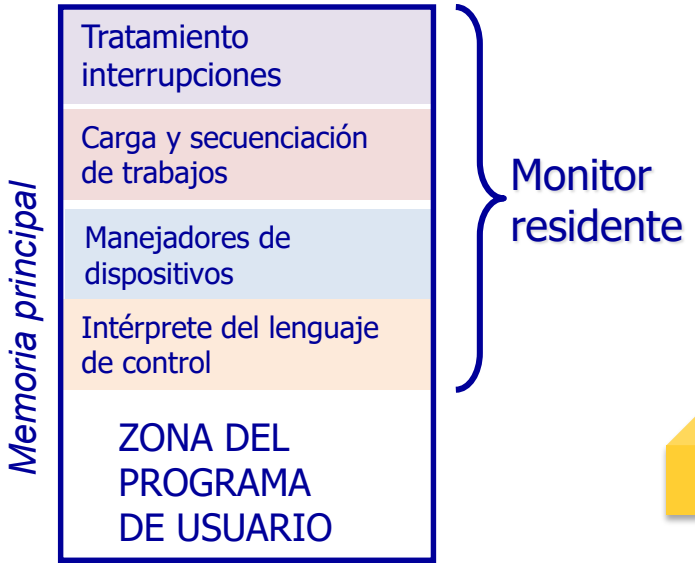
- Primeros Sistemas: Sistemas por Lotes

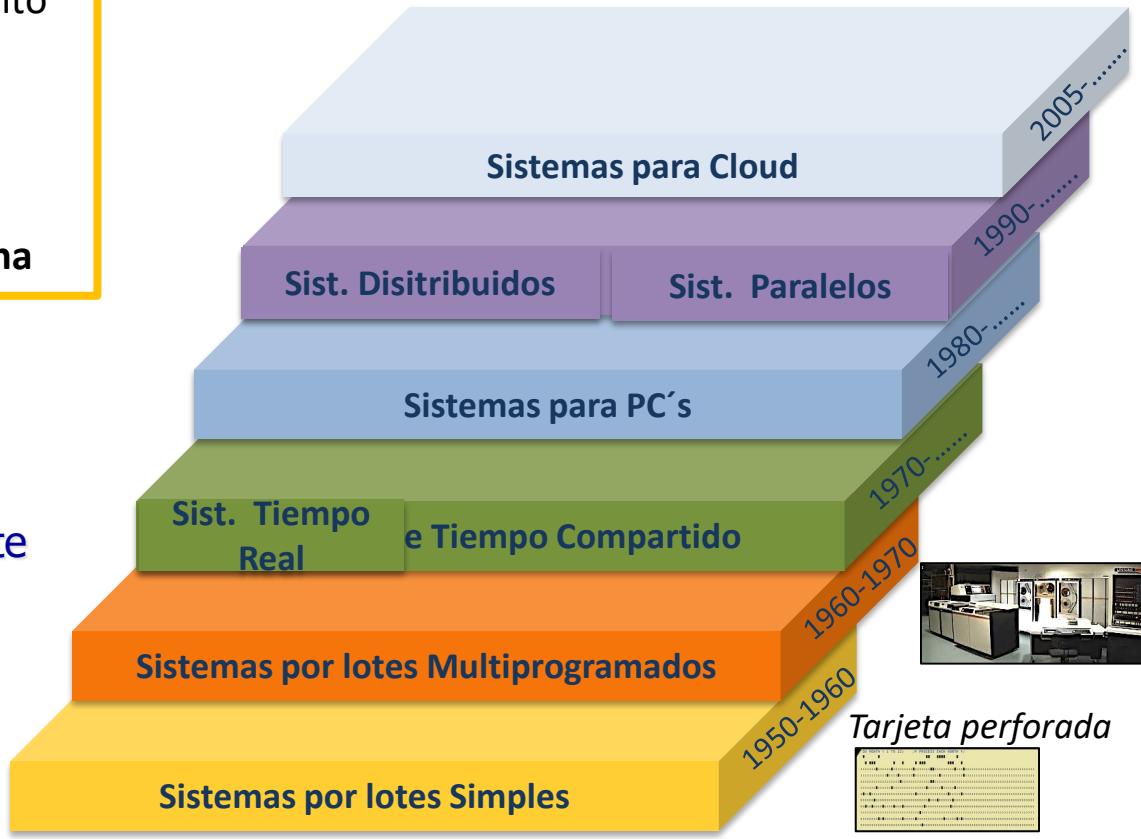
Sistemas por Lotes Simples

- Se **procesan secuencialmente** los trabajos: No se hace uso de la CPU mientras se hace una operación de E/S
- Baja utilización de la CPU
- Monitor residente. Control automático de: finalización de tareas, tratamiento de errores, carga y ejecución de la siguiente tarea
- Procesamiento por lotes
- Acceso a dispositivos E/S
- **No hay interacción usuario-máquina**

Sistemas por Lotes Multiprogramados

- Planificación de trabajos/CPU.
- Multiprogramación
- Gestión y protección de memoria (particiones fijas)
- Gestión de disco
- **No hay interacción usuario máquina**





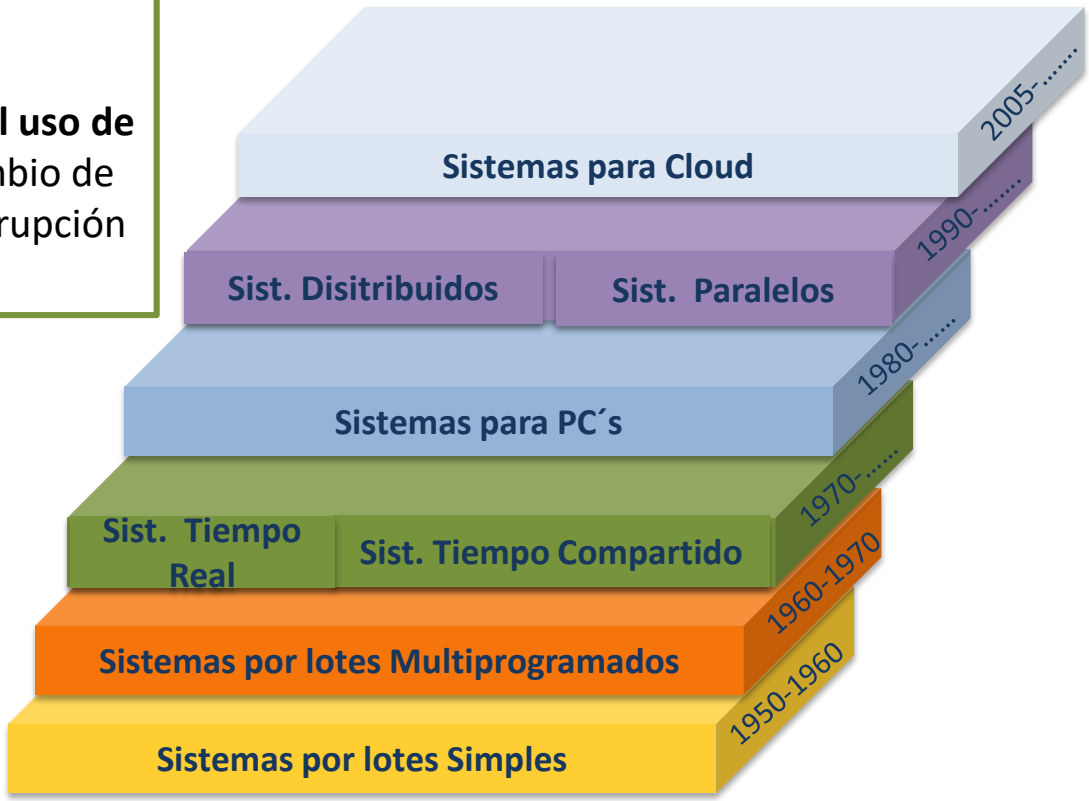
Tarjeta perforada

Evolución de los Sistemas Operativos

- Sistemas modernos:

Sistema de Tiempo Compartido
- **Interacción usuario máquina y multiprogramación**
- Sincronización y comunicación de trabajos
- Sistema de archivos. Gestión de ficheros
- Protección
- Memoria virtual
- Planificador de proceso: El SO **limita el uso de CPU** de un proceso asociando un "cambio de contexto" a la ocurrencia de una interrupción de reloj

Sistemas para PC´s
- Destinados al uso individual
- Interfaz amigables basada en ventanas y ratón
- Capacidad multimedia
- Capacidad Plug-and-Play
- Acceso a red



Sistema de Tiempo real
- Para ejecución de tareas que han de completarse en un plazo prefijado

Ley de Corbató: El número de líneas de código que un programador puede escribir en un período de tiempo dado, es el mismo con independencia del lenguaje de programación utilizado.
Cuanto más sofisticado sea el lenguaje un mismo número de líneas de código implementa algoritmos más complejos.

Evolución de los Sistemas Operativos

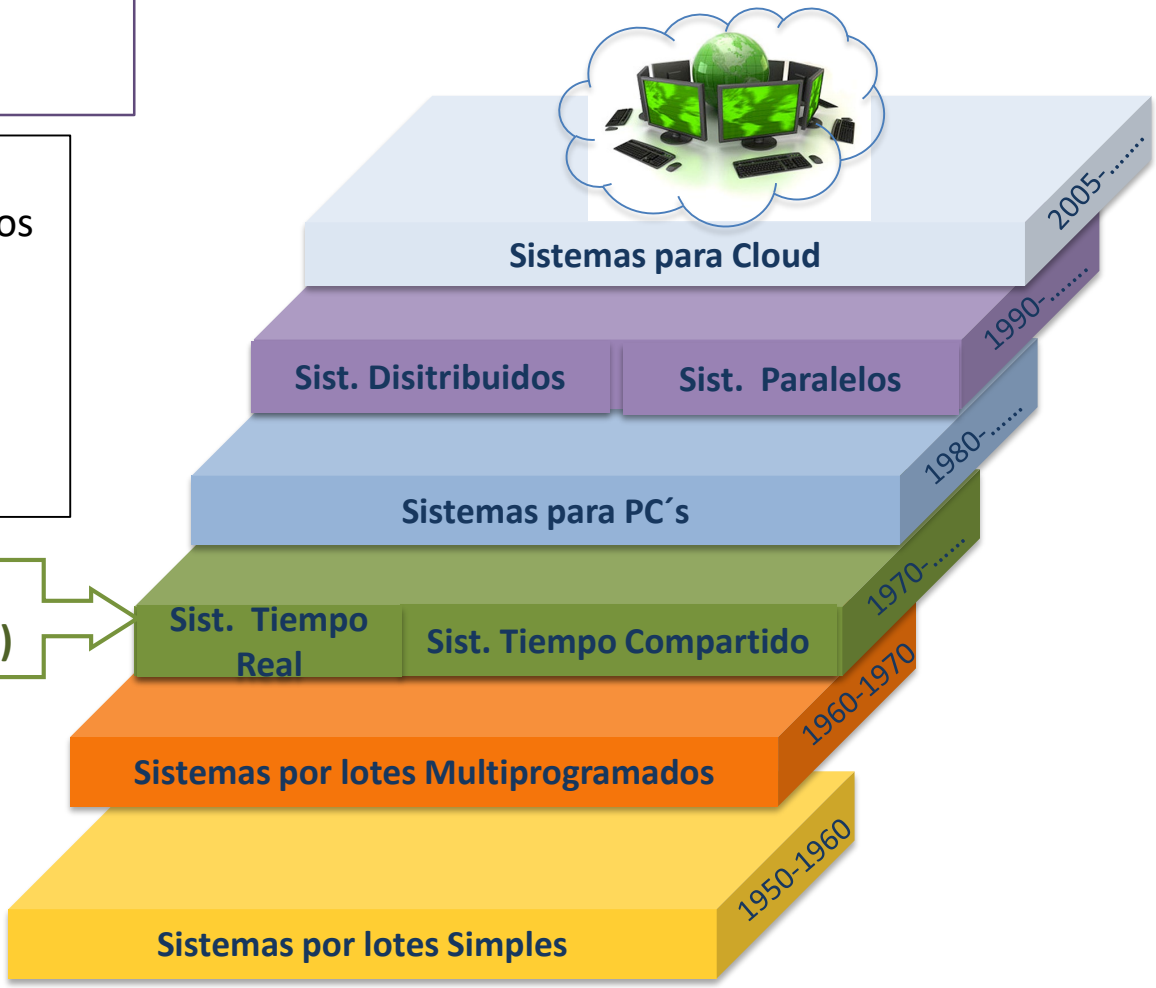
- Sistemas modernos:

Sistemas Paralelos
- Multiprocesador (Multicore):
  - Varios procesadores acoplados compartiendo bus y memoria
- Tolerancia a fallos

Sistemas Distribuidos
- El cómputo se reparte entre varios procesadores conectados mediante una red
- Comunicación entre nodos
- Compartición de recursos
- Carga compartida

Primeros sistemas UNIX
Escrito en lenguaje de alto nivel (C)

Sistema Cloud
- Almacenamiento y computación como servicio



Evolución de los Sistemas Operativos

- Historia de UNIX y C

1968
- Crisis del software

1969
- Primeros sistemas UNIX
  - SO Escrito en lenguaje de alto nivel (C)

1988
  - Primera versión del estándar POSIX. IEEE 1003. Normalización del interfaz de llamadas a sistema y otros componentes de UNIX. Interoperabilidad a nivel de código fuente.

Dennis Ritchie y Ken Thompson en computadores PDP-11 durante el primer desarrollo de Unix

1975
- Incorporación del direccionamiento virtual de memoria en el procesador PDP-11.
  - Computadores DIGITAL (DEC) VAX 11/780 con Sistema Operativo VAX11/VMS (VMS: *Virtual Memory System*)

Evolución de los Sistemas Operativos

- Primeros computadores personales

1964 – ¿*Programma 101*

→ calculadora programable?

1972 – ¿*Xerox Alto* → *Alto Executive*

Workstation?

1975 – ¿*Altair 8800 Kit*?

CP/M, Altair BASIC, ..

1977 – *Apple II* → intérprete BASIC

¿DISER Lilith (*Workstation*)

→ Oberon?

1980 – 86-DOS /x86DOS /QDOS

1981 – IBM PC → PC-DOS / MS-DOS

1982 – Amstrad CPC → CP/M

ZX Spectrum → Sinclair BASIC

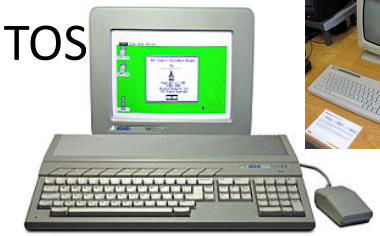
Commodore 64 → GEOS

1984 – Apple Macintosh → MacOS

Commodore Amiga 1000 → AmigaOS

Amstrad PCW → CP/M Plus

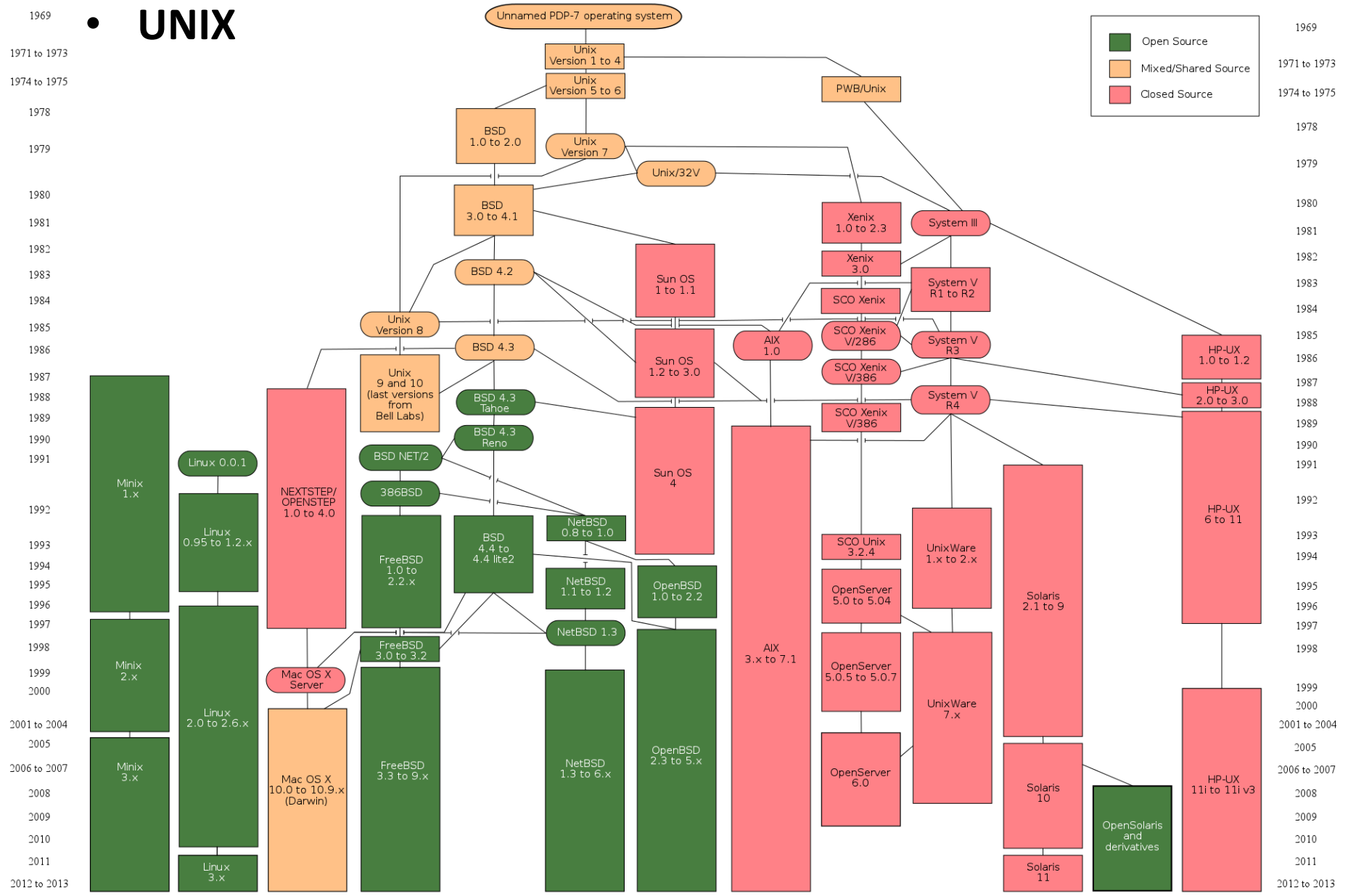
1985 – Atari ST → Atari TOS



Fuente: imágenes de <http://www.wikipedia.com >

Escenarios de trabajo actuales

- **UNIX**



- **Ejercicio UT01.1_** Intente averiguar tanto la versión de núcleo o Kernel de sistema operativo que hay instalada en su máquina Linux, como la distribución. Para ello utilice los comandos del shell `uname` y `lsb_release`.

  a) Obtenga la versión del Kernel del sistema, ejecutando:

  ```
  $ uname -rs
  ```

  b) Obtenga el nombre del sistema operativo, ejecutando:

  ```
  $ uname -o
  ```

  c) Obtenga la arquitectura del procesador (i386, i686, x86_64), ejecutando:

  ```
  $ uname -m
  ```

  d) Obtenga la distribución de sistema operativo, ejecutando:

  ```
  $ lsb_release -i
  ```

¡Aviso!: el `$` que encabeza cada orden representa el prompt de la máquina UNIX

¡Aviso!: puede utilizar las ordenes `man uname` y `man lsb_release` para obtener información de ayuda