UD4. Sistemas operativos

DESARROLLO DE APLICACIONES MULTIPLATAFORMA. 1º DE DAM

Índice:

1. Clasificación del software según su tipo

Definición de software (RAE): “cjto. de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora”.

El software se puede clasificar en una jerarquía, según su cercanía al hardware o al usuario que maneja el equipo informático.

USUARIO	**Usuario**
SOFTWARE DE APLICACIÓN	**Aplicaciones. Ej. Navegador web**
SOFTWARE DE SISTEMA	**Sistema Operativo. Ej. Windows**
HARDWARE	**CPU, Memoria, Periféricos, etc.**

1.1. Software de sistema

Def. SW de sistema o SW de base: “cjto. de SW que se encarga de gestionar los recursos HW del sistema informático, sirviendo de base a la ejecución de los programas de aplicación. Actúa como intermediario entre el HW del equipo (memoria, impresoras, discos, etc.) y los usuarios del mismo (usuarios finales o desarrolladores de software)”.

Intermediación:

- Usuarios: interfaz de usuario, gráfica o “textual”.
- Aplicaciones: “API”, conjunto de funciones y procedimientos.

A nivel del SW del sistema, el elemento fundamental es el SO, pero también se incluyen en este nivel los controladores, las herramientas de diagnóstico y otras utilidades.

Entre los SSOO más utilizados se encuentran MS Windows, GNU/Linux y Apple mac OS.

1.2. Software de aplicación

Formado por los programas que permiten a los usuarios realizar tareas concretas, que pueden ser

- Generales: procesadores de texto, hojas de cálculo, navegadores de internet, etc.
- Específicas: contabilidad, diseño asistido por ordenador, videojuegos, etc.

Es frecuente que junto al SO se instalen algunos programas (calculador, navegador, editor de texto, etc.) que en realidad son **aplicaciones** que los proveedores del SO incorporan como **complemento** para facilitar ciertas tareas comunes.

1.3. Lenguajes de programación

Un lenguaje de programación está formado por un conjunto de símbolos y de reglas sintácticas y semánticas que están diseñadas para crear programas de ordenador.

En la actualidad los lenguajes son de **alto nivel**, cercanos al lenguaje humano. Para ser comprendidos por el ordenador se realiza la **compilación**.

En ocasiones, el código fuente se traduce a un código intermedio, o *bytecode*, que es interpretado por una *máquina virtual* o *framework.* Ejemplos son ":NET Framework" y la "Máquina Virtual Java".

1.3. Lenguajes de programación (II)

Es muy frecuente que, para llevar a cabo su trabajo, los programadores utilicen Entornos de Desarrollo Integrados (IDE). Son aplicaciones especializadas en este tipo de tareas que incluyen:

- Editor para escribir el código fuente.
- Compilador para convertirlo en código objeto.
- Otras herramientas complementarias, como depuradores que permiten detectar y eliminar los errores que se hayan producido al escribir el programa.

Algunos ejemplos de lenguajes de programación son: Visual Basic, Java, C/C++, PHP, Perl, Ruby.

Ejercicios refuerzo / ampliación

1. Clasifica los siguiente programas en software de base y en software de aplicación:
   - Solaris.
   - Gentoo.
   - Google Chrome.
   - Android.
   - Avast.
   - Virtual Box.
2. Busca en Internet 3 entornos IDE, como mínimo, para el desarrollo de programas. Al menos uno para entornos Windows, otro para Linux y otro para MacOS, incluyendo imágenes del mismo.

1. Investiga sobre los lenguajes de programación de bajo nivel. ¿Cuáles son los más importantes? ¿Por qué se llaman así? ¿Qué ventajas y desventajas tienen con respecto a los lenguajes de alto nivel?

2. Clasificación del software según su licencia

UNIDAD 4. SISTEMAS OPERATIVOS. (PARTE 1)

2.1. Tipos de licencias

Def. licencia: “contrato entre el proveedor de un programa informático y el usuario (o empresa) que lo utiliza, para fijar las condiciones a las que se obligan ambas partes durante el tiempo en que el programa esté en uso”.

Entre los aspectos que regula una licencia de SW, están:

- La vinculación del programa a un determinado hardware.
- El número de copias del programa que puede utilizar el usuario.
- Los derechos que el usuario tiene sobre el programa.
- El período durante el que se cederán dichos derechos.
- La responsabilidad que tienen el proveedor sobre los fallos de un producto.
- La posibilidad de que el programa puede ser cedido a otras personas.
- Etc.

2.1. Tipos de licencia (II)

En función de los derechos que el proveedor mantiene sobre su producto, tenemos los siguientes tipos de licencias de software:

- Licencias de **software con código abierto**: pone a disposición de los usuarios el código fuente con el que está construido. Está unido al concepto de *software libre*. Tenemos dos subtipos:
  - Permisivas: los programas pueden modificarse o crear otros nuevos a partir de ellos **sin que el resultado tenga que mantener** las condiciones de la licencia original. Ejemplos: Apache Software License, BSD License, MIT License, PHP License.
  - No permisivas: los programas pueden modificarse o crear otros nuevos a partir de ellos, pero el resultado **deberá publicarse bajo los mismos términos** de la licencia original. Ej. GNU GPL, *Common Public License*, *Open SSL License*, *Eclipse Public License*.

2.1. Tipos de licencia (III)

(continuación):

- Licencias de **software de código cerrado**: **no** se distribuye el código fuente junto con el software. También se conoce como *software privativo*. En las licencias se limitan las posibilidades que tienen los usuarios para utilizar, copiar, modificar, redistribuir o ceder el software. Un ejemplo son las licencias CLUF (Contrato de Licencia para Usuario Final), llamadas *EULA* en inglés.

Variantes:

Código abierto		Código cerrado
Permisivas	No permisivas	
BSD License, MPL, …	GNU GPL, CPL, …	CLUF, Freeware, …

- *Freeware*: tipo de licencia para software que se distribuye gratuitamente y por tiempo indefinido. No suele incluir el código fuente.
- *Shareware*: permite que el software se evalúe, pero puede limitar el tiempo o algunas de sus funcionalidades. Para quitar limitaciones, hay que pagar.
- Dominio público: software publicado sin licencia. Puede utilizarse, modificarse, redistribuirse o licenciarse sin ningún tipo de limitaciones.

2.2. Libertades del software libre

Su definición está asociada a la *Free Software Foundation* (1985).

Para que un programa se considere *software libre* debe garantizar las siguientes libertades:

- Libertad 0: usar el programa, con cualquier propósito (**uso**).
- Libertad 1: estudiar cómo funciona el programa y modificarlo, adaptándolo a las propias necesidades (**estudio**).
- Libertad 2: distribuir copias del programa, con lo cual se puede ayudar a otros usuarios (**distribución**).
- Libertad 3: mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie (**mejora**).

Las libertades 1 y 3 requieren acceso al código fuente.

2.3. Comercialización de productos con licencias propietarias

Las licencias propietarias pueden adquirirse a través de diferentes vías:

- *Retail* (o *Full Packaged Product*): forma habitual de comprar SW en un establecimiento (en caja). Dirigida a usuarios que no necesitan más de 5 licencias y pueden encontrarse 2 variantes:
  - Producto completo: no requiere versión previa para su instalación.
  - Actualización (*upgrade*): parte de una versión previa (con licencia).
- OEM (*Original Equipment Manufacturer*): preinstalado en un nuevo ordenador (licencia OEM). Garantiza una correcta instalación y configuración. La licencia se vincula al HW, normalmente.
- Licencias por volumen: dirigidas a empresas o entidades. Puede incluir derechos como el de transferencia a determinados usuarios. También tiene las opciones de producto completo o actualización. Es frecuente que dispongan derechos de *downgrade* (sistemas más antiguos).

2.4. Ventajas y desventajas del SW libre

Entre las ventajas, resaltamos:

- Es más económico, o de ningún coste. Permite a las empresas de pocos recursos seguir creciendo → en España se gastan más de 1000 millones de euros en licencias de Microsoft cada año.
- El soporte puede ser local, evitando la dependencia de una compañía extranjera.
- Los programas se pueden instalar tantas veces y en tantos equipos como sea necesario. Además, pueden compartirse libremente.
- El uso de formatos abierto (ej. odf) facilita la interoperabilidad entre sistemas (independencia de HW, aplicación o SO).
- La posibilidad de acceder al código fuente permite la creación de nuevos productos, evitando la necesidad de comenzar desde cero.
- Este modelo ayuda a la reducción de las brechas tecnológicas entre países avanzados y países en vías de desarrollo.

2.4. Ventajas y desventajas del SW libre (II)

Ventajas destacables (continuación):

- Un fallo de seguridad en un producto de software libre es resuelto por la comunidad en un plazo muy breve. Pero las empresas disponen de menos medios humanos dedicados a estas tareas y la solución puede tardar meses.
- El modelo de negocio del software libre suele basarse en el servicio, no en las licencias → mayor atención al cliente y creación de puestos de servicio.
- No existe la "obsolescencia programada", es decir, interés comercial porque el producto deje de funcionar, en beneficio de uno nuevo. Ej. XP.
- El acceso al código fuente ofrece la absoluta seguridad de que no incluye ningún tipo de código malintencionado, o sencillamente, no autorizado.
- El uso de repositorios permite el acceso a miles de aplicaciones y otras herramientas, revisadas y garantizadas de forma gratuita. El método de instalación es sencillo y uniforme. La alternativa en el mundo del SW propietario es recurrir a las descargas ilegales → virus, troyanas, etc.

2.4. Ventajas y desventajas del SW libre (III)

Entre los inconvenientes, podemos incluir estos:

- Es común que el acabado estético sea inferior al del SW propietario.
- Existen muchas aplicaciones propietarias que son un estándar de facto en algunos ámbitos y, no conocerlas puede suponer una desventaja. Ej: Microsoft Office, Photoshop, etc.
- Los programas de juegos, por cuestiones comerciales, suelen estar disponibles sólo para Microsoft Windows.
- Suele haber un mayor desconocimiento de su funcionamiento → mayor esfuerzo formativo del personal de mantenimiento.
- Existen dispositivos HW que sólo disponen de drivers compatibles con Microsoft Windows, por lo que la utilización de GNU/Linux puede ser inviable.
- Existe un mercado laboral inferior en la implantación de sistemas libres, aunque más especializado.

Ejercicios refuerzo

3. Realiza una tabla como la del ejemplo para buscar las licencias del siguiente software: Windows Server 2016, Gentoo, FreeBSD, Mozilla Firefox, Apache Tomcat, Nitro PDF, NetBeans, Avast Antivirus.

Software	Licencia	Libertades y restricciones
Windows Server 2012	??	??
…	…	…

Escoge una de las siguientes organizaciones/empresas (ponte de acuerdo con tus compañeros para no repetir) y describe brevemente su historia, su producto o productos estrella y la licencia que utiliza con él: The Document Foundation, Microsoft, Adobe, Apache, GNU, Oracle. Puedes escoger otra que no esté en la lista tras consultarlo con el profesor.

3. Sistemas Operativos

UNIDAD 4. SISTEMAS OPERATIVOS. (PARTE 1)

3.1. Tipología de los SSOO

La misión de todo SO es doble:

- Administrar los recursos HW: memoria, procesador, unidades E/S (drivers).
- Actuar como interfaz “sencilla” de usuario (de texto o gráfica –GUI-).

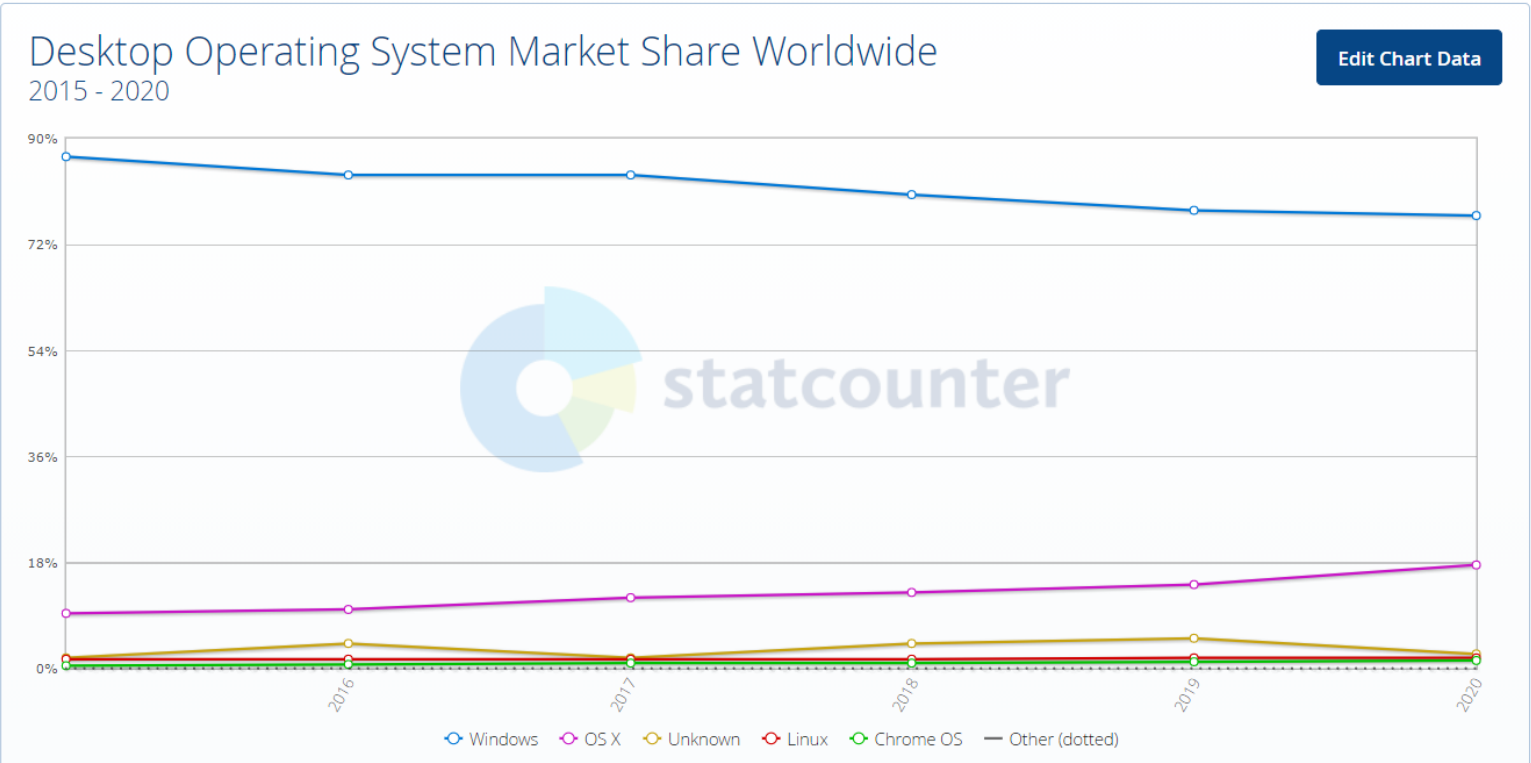
Hay varias maneras de clasificar a los SSOO:

- Monousuario o multiusuario: usuarios que pueden usar el sistema al mismo tiempo.
- Monopuesto o multipuesto: personas que interactúan “a la vez” a través de terminales.
- Monotarea o multitarea: tareas de una en una, o en paralelo o forma concurrente.
- Monoprocesador o multiprocesador: una o varias CPUs.
- Monoprogramado o multiprogramado: uno varios programas en memoria al mismo tiempo.

3.2. Panorámica actual de los SSOO

Ordenadores de escritorio:

- Los más utilizados son Microsoft Windows, GNU/Linux y OSX.



Ejercicios refuerzo

4. Clasifica según la “tipología de los SSOO” vista, los principales SSOO actuales, a saber, Windows 10, Linux Ubuntu, mac OS y Android. Puedes utilizar una tabla como la siguiente:

Tipología\ S. O.	Windows 10	Linux Ubuntu	mac OS	Android
Nº de usuarios				
Nº de procesos				
Nº de puestos				
Nº de procesadores				
Nº de procesos en memoria				

Ejercicios refuerzo / ampliación

5. Investiga en la web Statcounter Globalstat cuáles son los sistemas operativos más utilizados en la actualidad. Incluye una imagen de las estadísticas desde Enero de 2014 hasta la actualidad. ¿Qué ha ocurrido con los sistemas operativos de Windows? ¿Y con los de escritorio en general?

5. Crea una tabla como la del ejercicio 4 de refuerzo pero para Ubuntu Server, MacOS Server y Windows Server.

3.2. Panorámica actual de los SSOO (II)

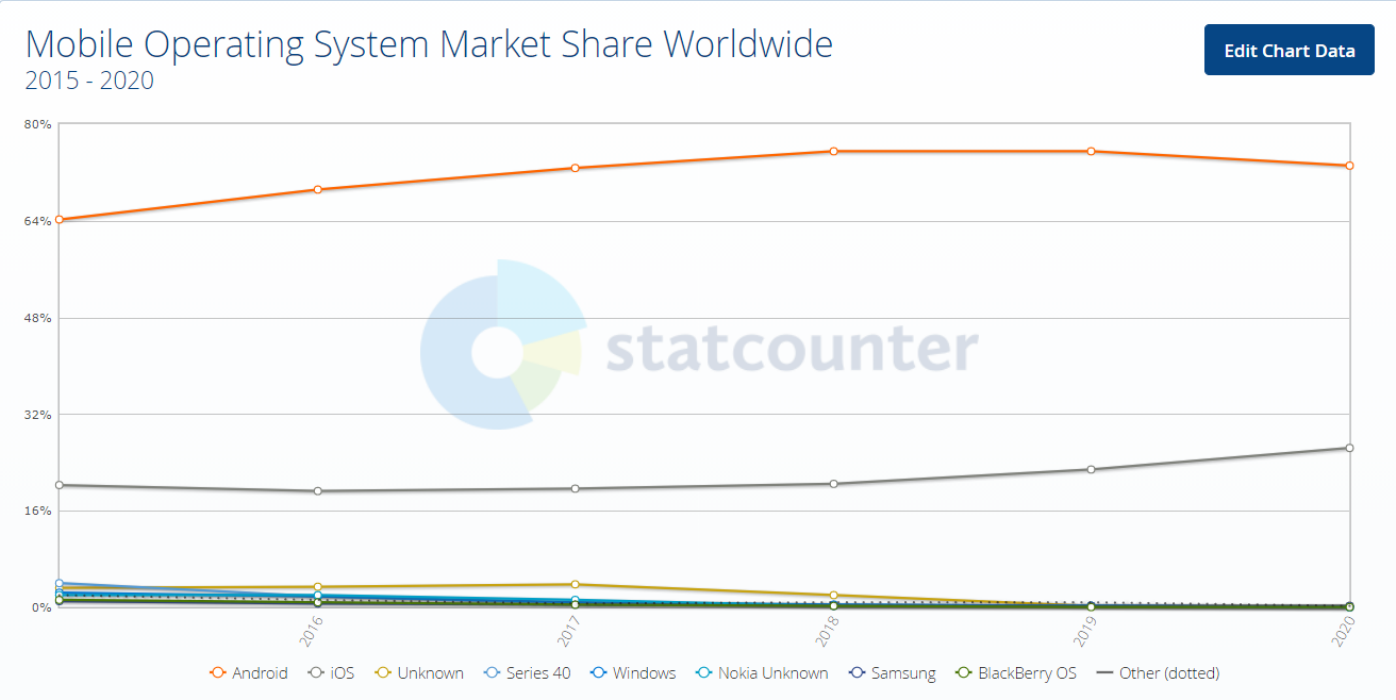
Ordenadores servidores:

- Microsoft Windows Server: principalmente las versiones 2012 y 2016.
- GNU/Linux Server: distribuciones RedHat, Ubuntu Server, CentOS, SuSe…
- UNIX (IBM AIX, HP-AU).
- Solaris/OpenSolaris.
- Apple OS X Server.

3.2. Panorámica actual de los SSOO (III)

Ordenadores móviles (*smartphones*, *tablets, smartwatches, …*):

- Los más utilizados son Android, iOS, Windows Mobile, BlackBerry OS, etc …



3.2. Panorámica actual de los SSOO (IV)

Otros sistemas:

- Chrome OS: basado en GNU/Linux, desarrollado por Google y orientado a Internet.
- webOS: basado en GNU/Linux, desarrollado por LG para incluirlo en sus televisores.
- Tizen: basado en GNU/Linux, apoyado por Linux Mobile Foundation (LiMo), Linux Foundation y Samsung, para ser instalado en teléfonos inteligentes, tabletas, teléfonos inteligentes, etc.

- Sistemas empotrados: Android, iOS y Windows disponen de versiones específicas para embeberlos en dispositivos como smartwatches, automóviles, y otros dispositivos, tratando de orientarse hacia el “Internet de las cosas”.

Ejercicios refuerzo / ampliación

6. Investiga sobre los SSOO Android e iOS:

- Breve historia (vinculación con SSOO anteriores).
- Tipo de licencia.
- Clasificación/Tipología.
- Sistemas a los que está dirigido.
- Forma de nombrar las versiones y versión más actual.
- Imagen corporativa.
- Fuentes (URLs).

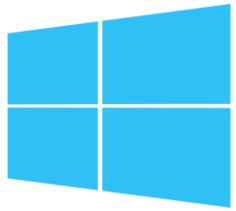
6. Investiga sobre los SSOO kaiOS y Tizen

- Breve historia (vinculación con SSOO anteriores).
- Tipo de licencia.
- Clasificación/Tipología.
- Sistemas a los que está dirigido.
- Forma de nombrar las versiones y versión más actual.
- Imagen corporativa.
- Fuentes (URLs).

3.3. SSOO libres y propietarios (escritorio)

Tenemos, principalmente, 3 “familias” de SSOO :

- Microsoft Windows: principal exponente de los sistemas con licencia de software privativo.
- Linux Ubuntu: el de mayor repercusión con licencia de código abierto.
- Mac OS: innovador, orientado a un HW concreto, y con licencia privativa.



3.3.1 Microsoft Windows

Microsoft dispone de una gama completa de SSOO que cubren dispositivos móviles, sistemas empotrados, ordenadores de sobremesa y servidores.

Sin embargo, el primer Windows (1985) era sólo una GUI de MS-DOS. Trataba de equipararse al Mac OS de Apple (1984).

A partir de de Windows XP, el núcleo de MS-DOS fue sustituido por núcleo NT de Windows 2000. (gama profesional)

El siguiente gran cambio vino con Windows 8: remodelación de la GUI (eliminación del botón Inicio y pantalla de Inicio ideada para dispositivos móviles). Más adelante, por presión de los usuarios, Windows 8.1 recuperó el botón de Inicio.

Otro gran cambio fue el soporte para arquitecturas de procesadores ARM.

3.3.1 Microsoft Windows (II)

Versiones de Microsoft Windows		
Año	Logotipo	Versión
1985		Windows 1.0
1995		Windows 95
2001		Windows XP
2009		Windows 7
2012		Windows 8
2015		Windows 10

3.3.2. GNU/Linux

La historia comienza en 1960 con **Multics**, un proyecto de SO llevado a cabo por los Laboratorios Bell de AT&T.

Ken Thompson y Dennis Ritchie continuaron el proyecto, rebautizándolo como **Unics**, y finalmente como Unix (**1969**).

En **1972** UNIX fue rescrito en el **lenguaje C** (inventado por D. Ritchie).

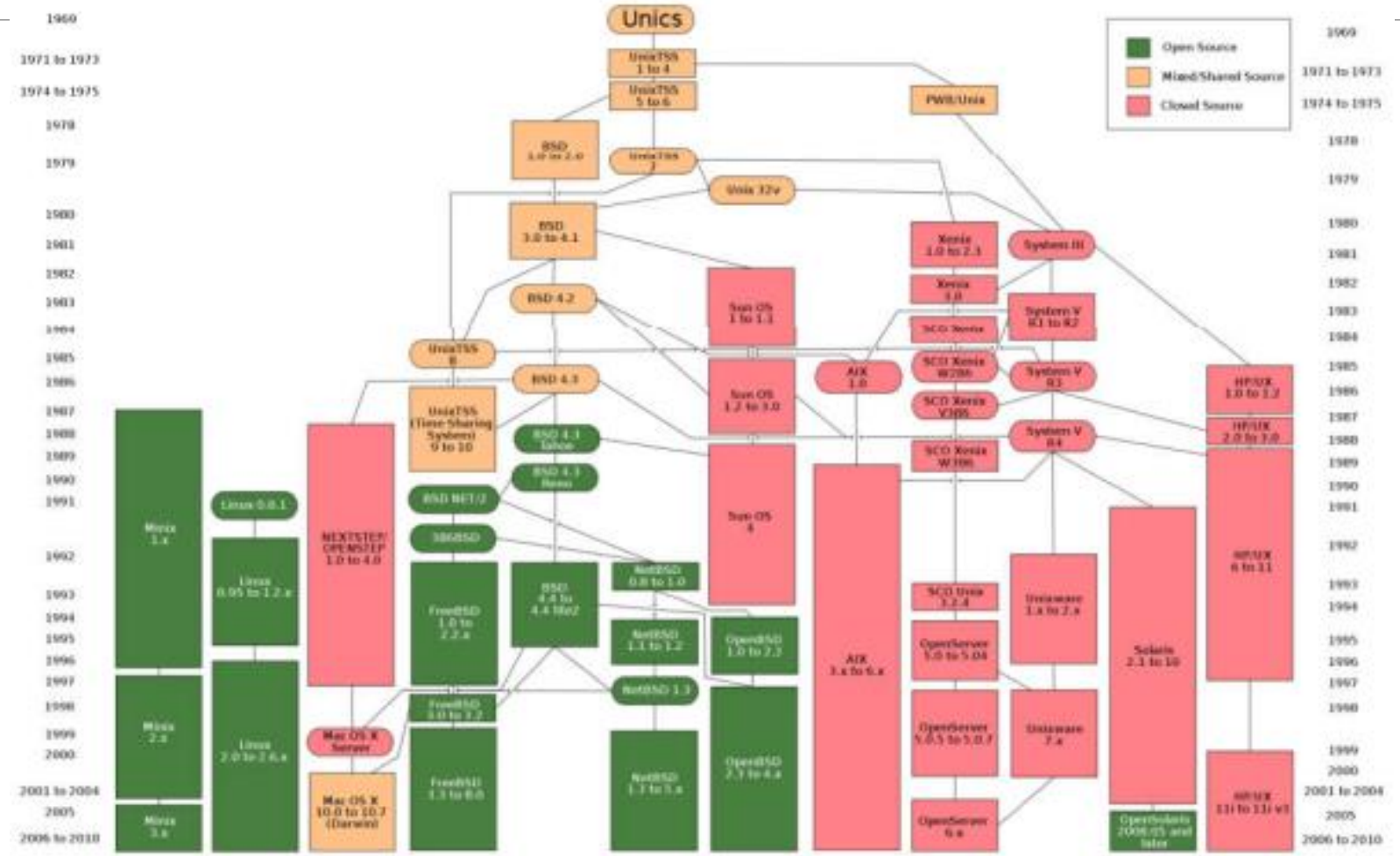
AT&T puso UNIX© a disposición de las universidades, lo que provocó que aparecieran variaciones: BSD, Solaris, AIX, System V, etc.

En 1983, Richard Stallman anunció el proyecto GNU, que tenía como objetivo la creación de un SO libre. Pero el proyecto para el núcleo se estancó.

En 1991, **Linus Torvalds**, estudiante finlandés, termina el núcleo de un SO que se inspiraba en el SO *Minix* del profesor A. S. Tanembaum.

En **1992**, Linus publica el núcleo con licencia GPL de GNU, y así fue adaptado por el proyecto y apareció el SO **GNU/Linux**.

3.3.2.1. La familia UNIX



3.3.2.2. Las “distros” de GNU/Linux

GNU/Linux es de **código abierto**, lo que permite que multitud de programadores distribuidos por todo el planeta colaboren, no sólo en el SO, sino también en aplicaciones.

Han surgido diversas distribuciones (núcleo+ aplicaciones): ***Debian***, ***Red Hat*** y ***Slackware*** son las distribuciones “matrices”

Una de las más representativas es **Ubuntu**, basada en Debian y patrocinada por la empresa Canonical (Mark Shuttleworth).

Ubuntu se ofrece en diferentes versiones, para ámbitos domésticos y profesionales. Se actualiza cada 6 meses, y cada versión tienen un nombre de un animal y un adjetivo (siguen el orden alfabético).

Cada 2 años, aparece una **versión LTS** que recibe el soporte de Canonical durante 5 años (el resto, sólo 9 meses)

3.3.2.2. Las “distros” de GNU/Linux (II)

3.3.2.3. Las versiones LTS de Ubuntu

Versión	Nombre en clave	Lanzamiento
6.06	Drapper Drake	01/06/2006
8.04	Hardy Heron	24/04/2008
10.04	Lucid Lynx	29/04/2010
12.04	Precise Pangolin	26/04/2012
14.04	Trusty Talhr	17/04/2014
16.04	Xenial Xenux	21/04/2016
18.04	Bionic Beaver	26/04/2018
20.04	Focal Fossa	20/06/2020
22.04	Jammy JellyFish	22/04/2022

3.3.3. mac OS

Anteriormente llamado OS X, es un SO basado en UNIX (**BSD**, 1996), desarrollado, comercializado y vendido por **Apple** Inc.

A diferencia de los SSOO anteriores, mac OS está implementado pare ser ejecutado en un tipo de **HW concreto**. Por un lado esto asegura la venta de la máquina, y por otro, garantiza su eficiencia.

Está basado en el **núcleo Darwin** e incluye la GUI *Aqua* y el *Finder*.

Continuando el legado de su antecesor, nombra a sus versiones con **lugares de California** (Yosemite, El Capitán, Sierra, …y la actual Monterey.).

macOS

Ejercicios refuerzo

7. Realiza un pequeño trabajo de investigación sobre las versiones de los principales SSOO (Windows, GNU/Linux y macOS) para sistemas **servidores**:

- Breve historia sobre sus comienzos (fechas).
- Clasificación/tipología.
- Versiones destacables y última versión.
- Requisitos mínimos de instalación de las últimas versiones.