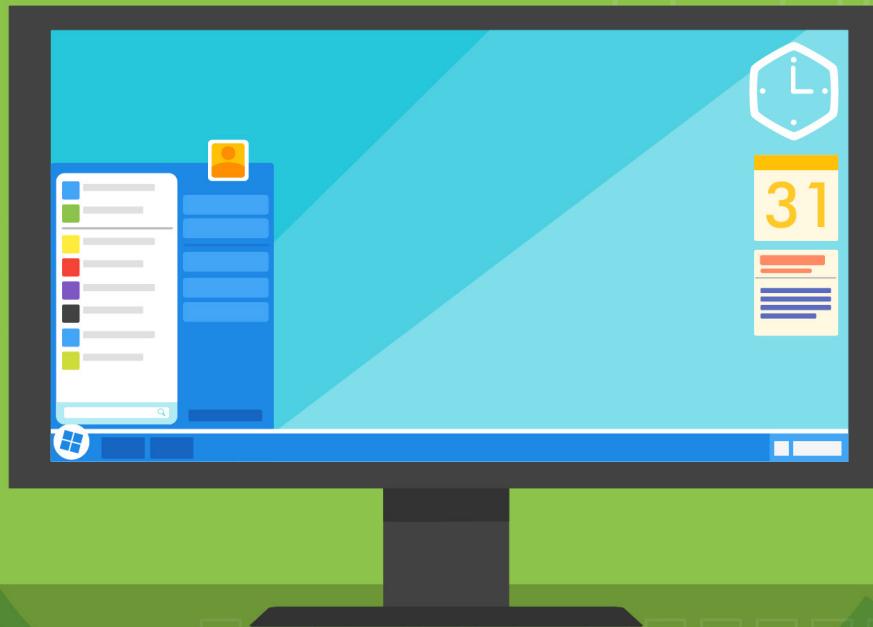


Tecnología en Desarrollo de Software

Unidad 1.

Introducción a los

sistemas operativos



Unidad 1. Introducción a los sistemas operativos

¿Sabías que un sistema operativo es el componente lógico más importante del software de un sistema computacional?

Está conformado por un conjunto de programas que realizan diversas tareas, como:

- Administración eficaz de recursos (empleo eficiente del hardware)
- Administración de memoria
- Recuperación de errores
- Control de operaciones de entrada y salida, etc.

El concepto de sistema operativo (SO) en un dispositivo electrónico ha evolucionado de forma notable en las últimas décadas, transformándose en un elemento vital al cual se dedican grandes investigaciones e inversiones en la industria, las universidades y la comunidad en general.

Su objetivo no solo ha sido extender, adaptar y mejorar la interfaz hacia el usuario, para acercar este dispositivo a personas inexpertas o sin conocimientos tecnológicos (es el caso paradigmático, por ejemplo, de la evolución de los sistemas operativos para dispositivos móviles), sino también en gestionar las cada vez más complejas y potentes infraestructuras hardware que permiten que un dispositivo móvil actual tenga igual potencia de procesamiento que un supercomputador de la década de los 80.



```
7 require 'rspec/rails'  
8  
9 require 'capybara/rspec'  
10 require 'capybara/rails'  
11  
12 Capybara.javascript_driver = :webkit  
13 Category.delete_all; Category.create  
14 Shoulda::Matchers.configure do |config|  
15   config.integrate do |t|  
16     t.with.test_framework :rspec  
17     t.with.library :rails  
18   end  
19 end  
20  
21 # Add additional requires below this line if you need them  
22  
23 # Requires supporting ruby files with custom matchers  
24 # and helpers with additional requirements for this spec  
25 # run as spec files by default. This means you don't  
26 # need explicit files for integration tests.  
27 # run twice. It is recommended that you do this unless you  
28 # end with _spec.rb. You can mitigate this issue by using  
29 # option on the command line.  
30  
31 # No results found for 'mongoid'  
32  
33 mongoid  
34  
35 buffer
```

Tema 1

Definición y concepto

Comencemos definiendo qué es un sistema operativo.

Podemos definir un sistema operativo como el software esencial de todos los programas disponibles en un computador o estación de trabajo, y que utiliza microprocesadores.

Sistema Operativo

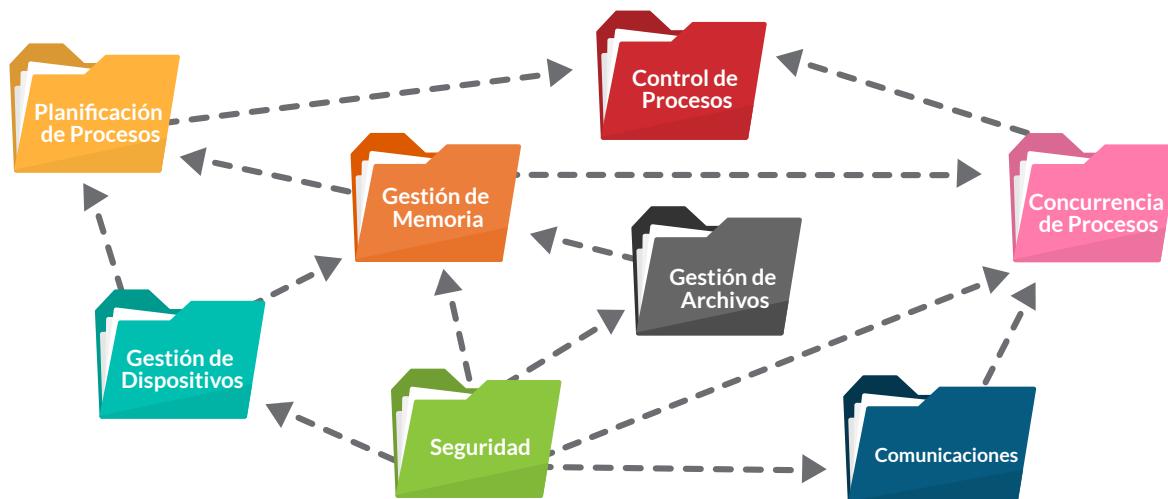
- Agrupa y controla un conglomerado de programas, gestiona los recursos del hardware suministrando una interfaz entre el usuario y los programas instalados.
- Es el programa principal que ofrece multiplicidad de usos y administración de recursos, proporciona una serie de abstracciones (oculta la complejidad interna) para que los usuarios se enfoquen en utilizar cada elemento del sistema y aísla procesos, de forma que todos ellos puedan ejecutarse en paralelo sin interferencias. (Wolf, Ruiz, Bergero, & Meza Vega, 2014)

De acuerdo con lo anterior, podemos decir que un sistema operativo es tanto el director de orquesta como la plataforma sobre la cual se despliegan todos los demás programas o software en el equipo informático; proporcionando acceso y control de todo el hardware.

El sistema operativo es, entonces, el intermediario entre el usuario y el hardware, partiendo desde los niveles más básicos; toda interacción entre humano y máquina pasa por el sistema operativo.

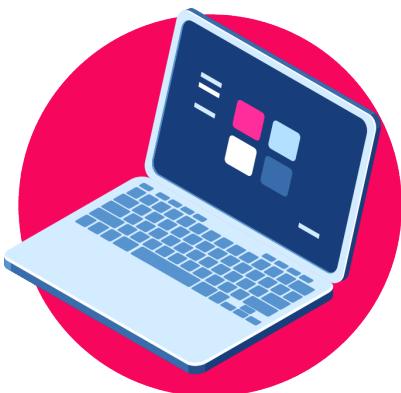
Nuestro sistema informático es un conjunto de capas o niveles superpuestos, donde el sistema operativo es el eje central. El manejo del disco duro o cualquier dispositivo de almacenamiento, la gestión de la comunicación en red, el control del procesador, la lectura de teclas presionadas, el despliegue o apertura de un programa, la gestión de usuarios, entre otras funciones primarias, son responsabilidad del sistema operativo.

Desde el nivel más bajo (el hardware) hasta el nivel más alto (software de aplicaciones), el sistema operativo se encuentra monitoreando todos los recursos y trabajando constantemente desde el momento en que el equipo se enciende.



La anterior imagen nos muestra las principales funciones del sistema operativo. Mantiene operantes todos los programas de un sistema. (La Red Martínez, 2002).

Ten presente que el componente fundamental del sistema operativo es su núcleo o Kernel, el cual gestiona todos los procesos por medio de servicios de llamadas al sistema (gestión de hardware) y comunicación del hardware y software (González Moreno, 2008)



Desktop (equipos de escritorio) / Laptop (equipos portátiles)

Sistemas operativos con aplicaciones de oficina, hogar o propósito general, donde predomina el entorno gráfico. Tradicionalmente son monousuarios y multitareas:

- Windows
- MacOS
- Chrome OS
- Linux Ubuntu. En general, Linux y sus derivaciones.



Móviles

Sistemas operativos orientados a dispositivos móviles, como tablets o smartphones:

- Android
- iOS



Servidores

Sistemas operativos que ofrecen programas multiusuario, aplicaciones en redes y herramientas centralizadas o empresariales:

- Windows Server
- Mac OS X Server
- CentOS
- Solaris

Ten en cuenta que, aunque un sistema operativo puede trabajar sobre diferentes características de hardware, siempre se tienen requisitos específicos para cada uno de ellos y se puede trabajar bajo dos entornos fundamentales (Raya Cabrera & Raya González, 2014):



Modo Comando

En este modo, el usuario interactúa con una interfaz carente de ventanas; se tiene disponible una línea de comandos donde solo se escriben órdenes para obtener un resultado cada vez que se presiona la tecla Enter. Cada comando puede tener parámetros específicos para su ejecución, los cuales se pueden consultar al acompañar el comando con /?.

Este entorno se denomina CLI (Command Line Interface).



En el sistema Windows, este entorno se conoce como Símbolo del sistema.

Los comandos principales son:

- **help**

Invoca la ayuda del sistema. Muestra una lista de todos los comandos con su respectiva descripción.

- **cd**

Permite cambiar de directorio. Si se adiciona el nombre de un directorio, facilita ingresar a él para visualizar o manipular su contenido. Por ejemplo, con *cd Windows*, se ingresa a la carpeta Windows. Si se acompaña de un doble punto, retrocede un directorio, ejemplo: *cd..*

- **chkdsk**

Permite comprobar, recuperar, reparar y eliminar los errores y sectores del disco duro.

- **dir**

Muestra una lista con el contenido del directorio actual.

- **tasklist:**

Muestra lista con los procesos actuales del sistema.

- **ping**

Permite ejecutar una prueba de comunicación en red con un recurso determinado por medio de su dirección IP o nombre de dominio.

- **taskkill**

Se encarga de finalizar un proceso determinado.

• **shutdown**

Se encarga de apagar o reiniciar el sistema.

• **Ipconfig**

Muestra información de la tarjeta o interfaz de red y permite realizar acciones sobre ella.

- Existe otro modo de comando en Windows, denominado *PowerShell*, que permite automatizar tareas.



Para el sistema operativo Linux, este entorno se conoce como Terminal.

A diferencia de Windows, Linux es sensible a las mayúsculas, es decir, puede generarse un error de sintaxis en el cambio de minúsculas a mayúsculas en el nombre de la instrucción o sus parámetros.

En este caso, los comandos principales son:

• **ls**

Muestra una lista con los archivos y carpetas del directorio actual.

• **cd**

Permite cambiar de directorio.

• **cp**

Permite realizar una copia de un archivo o carpeta.

• **mv**

Permite mover un archivo o carpeta.

• **rm**

Se encarga de borrar un archivo o carpeta.

• **chmod**

Permite cambiar los permisos de un archivo o directorio. Se define en tres niveles: dueño, grupos, y otros (esto se refiere a los usuarios del sistema o grupo de ellos).

Para cada grupo, por medio de un valor numérico, se asignan permisos de lectura (4), escritura (2) y ejecución (1), estos valores son acumulables. Para el ejemplo **chmod 754 holaMundo.java** se tiene la siguiente interpretación de permisos: para el dueño, propietario o creador del archivo, se tienen permisos de lectura, escritura y ejecución: $4 + 2 + 1$. Para el grupo, solo se tienen permisos de lectura y ejecución: $4 + 1$. Para otros usuarios, solo se permite la lectura del archivo: 4.

• **chown**

Permite cambiar el dueño o propietario del archivo o directorio.

• **grep**

Permite buscar un texto o cadena de caracteres en el archivo especificado, mostrando las líneas que contienen la búsqueda determinada.

• **apt**

Permite activar el sistema de gestión de paquetes, puede instalar, actualizar o desinstalar programas en sistemas basados en *Ubuntu*. Su sintaxis básica para buscar, instalar o actualizar un programa es:

- apt-get search
- apt-get install
- apt-get update

• **passwd**

Permite modificar la contraseña del usuario actual.

• **pwd**

Muestra la ruta del directorio actual.

• **whoami**

Muestra el nombre de usuario actual.

• **SU**

Permite cambiar de usuario y administrar el sistema. El usuario administrador de *Linux* se conoce como *root*. Si se adiciona el nombre de un usuario, se accede a la terminal con el usuario especificado.

• **sudo**

Permite ejecutar un comando. Frecuentemente, se combina con el comando *su*, permitiendo ejecutar una instrucción con perfil de administrador: *sudo su*

• **ifconfig**

Permite listar y configurar las tarjetas o interfaces de red del sistema.

• **top**

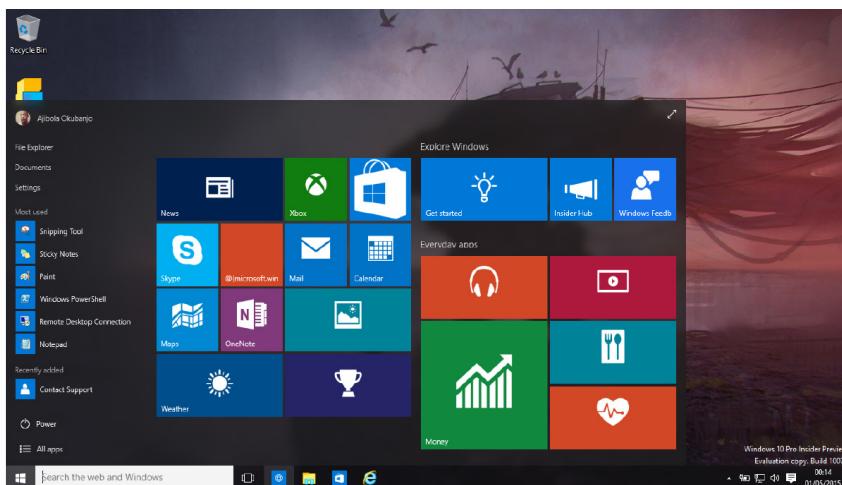
Muestra una lista que se actualiza periódicamente con los procesos del sistema.



Modo Gráfico

Este modo utiliza una interfaz que, por medio de ventanas, interacción con el puntero del mouse, objetos gráficos como botones, cuadros de texto, menús, íconos e imágenes, permite un trabajo más simple y amigable en el sistema operativo, pero implica un mayor consumo de memoria. Cuenta con una representación gráfica para todos los componentes del hardware y el software. Tiene animaciones y reproducciones de sonido para las acciones del usuario, las cuales buscan mejorar su experiencia.

Este entorno se denomina **GUI (Graphical User Interface)**.



La imagen nos muestra el entorno tradicional de los sistemas operativos tipo *Windows*, donde se despliega el conocido menú *Inicio* que da acceso a los aplicativos del sistema.



En esta imagen podemos observar el escritorio de *MAC OS*, sistema operativo de *Apple*, característico por su barra de íconos de programas en la barra inferior o *dock*.



Esta imagen nos presenta el entorno del sistema operativo *Ubuntu*, con su barra propia de herramientas o *lanzador* ubicada en la columna izquierda.

Adicionalmente, todos los sistemas operativos interactúan con otros programas especializados para lograr su tarea de administración, estos son:



1. Sistema de archivos

Hace referencia a los métodos y estructuras de datos utilizados por el sistema operativo para enlazar los archivos en un disco o partición; es decir, cómo se distribuyen y acceden los archivos en una unidad de disco. Cada sistema operativo usa su propio sistema de archivos.

Típicamente los datos son almacenados en bloques de un mismo tamaño, denominados sectores o clúster. El registro que relaciona la información y su respectivo sector de almacenamiento es gestionado por el sistema de archivos.

- El sistema operativo Windows utiliza **NTFS (New Technology File System)**, recomendado para tamaños entre 10 GB y 2 TB.
- El sistema operativo MAC OS utiliza **HFS Plus (Hierarchical File System +)** con bloques divididos en sectores de 512 Bytes.
- El sistema operativo Linux utiliza **EXT (extended file system)**; actualmente se tiene el cuarto sistema de archivos extendidos o **ext4**.



2. Controlador de dispositivo

Conocido como *Driver* o *controlador*, es un programa especializado o software utilitario que permite al sistema operativo acceder y gestionar óptimamente un periférico o hardware determinado. Sirve de enlace entre el núcleo del sistema y un hardware específico; sin él, el sistema operativo no puede identificar y comunicarse con una pieza del hardware. Son diseñados para una versión de sistema operativo específica. Existen controladores genéricos, incorporados en el sistema operativo y creados por el fabricante.

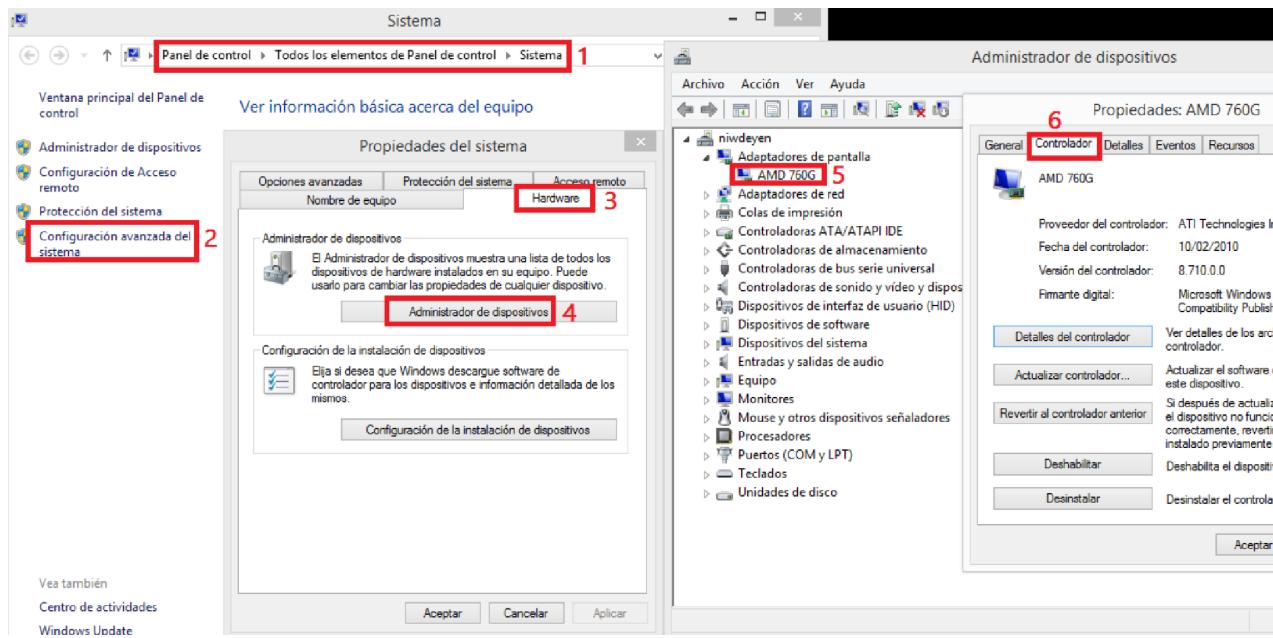
Estos últimos son proporcionados junto con el hardware (como es el caso de un lector de código de barras o un adaptador de red), bien sea por medio físico (CD de instalación de adaptador de red Inalámbrica USB ENCORE ENUWI G2), o en la página web oficial.



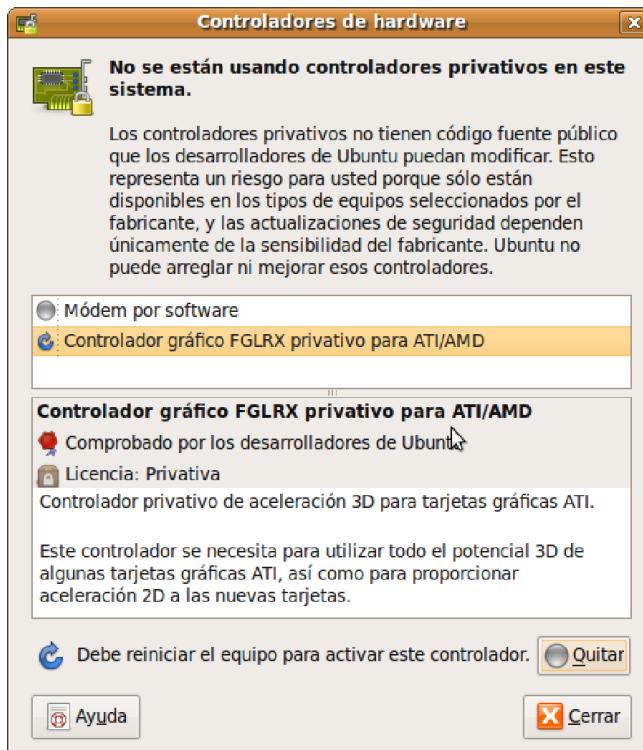
En el sistema operativo Windows, los controladores se gestionan de la siguiente forma (Imagen):

1. Abrir *Panel de control*, seleccionar opción: *Sistema*.
2. Clic en la opción: *Configuración avanzada del sistema*.
3. Abrir la pestaña: *Hardware*.
4. Clic en el botón: *Administrador de dispositivos*.
5. Seleccionar el hardware.
6. Clic en la pestaña: *Controlador*.

En esta ventana es posible comprobar si todos los componentes de hardware en el sistema, cuentan con su controlador; además, es posible actualizar, desinstalar o deshabilitar uno de ellos. Adicionalmente, es posible obtener una lista de todos los controladores en el equipo con el comando **driverquery**.



En el sistema operativo Linux, en su distribución Ubuntu, es necesario abrir la aplicación *Software y Actualizaciones* y, en su opción de *Controladores*, se encuentran los controladores respectivos. En algunos casos, los controladores, en Linux, requieren procesos diferentes de administración. La lista de detalles del hardware y sus controladores se pueden obtener con el comando **lspci**.



Ten en cuenta que los sistemas operativos, dada la arquitectura del procesador, pueden ser de 32 o 64 bits, en los cuales se puede aprovechar hasta 4 gigabytes en RAM, o superior, respectivamente. Al momento de instalar un sistema operativo, se debe analizar el hardware disponible para seleccionar cuál es el apropiado bajo los siguientes criterios:

- Tamaño de memoria RAM disponible: si tiene 4 gigabytes o menos, el sistema operativo debe ser de 32 bits. Para más de 4 gigabytes, se recomienda un sistema operativo de 64 bits; en este caso también es posible instalar un sistema de 32 bits, pero no aprovechará más de 4 gigabytes.
- Características del procesador: en las características técnicas del procesador, proporcionadas por el fabricante, es posible identificar si el procesador tiene la arquitectura de 64 bits. Por ejemplo, en el caso de un procesador Intel Core i9 7940X se confirma el soporte de 64 bits. En este caso, y si la memoria es superior a 4 gigabytes, es preferible que el sistema operativo a instalar sea de 64 Bits.

Para finalizar, los sistemas operativos, y en general todo el software, poseen un modelo de licenciamiento o protección de la propiedad intelectual; es decir, la autorización que el autor o autores conceden a otros para utilizar su creación. Esto hace referencia a los permisos de uso, modificación y distribución del software a manera de contrato. Estos permiten tener un soporte en el software, bien sea para recibir actualizaciones o apoyo en fallas.

Se clasifican de la siguiente forma:

1. Licencia de software de código abierto

Permite acceder al código fuente y crear modificaciones en él, con diversas restricciones en la distribución, como puede ser:

- **GNU General Public License (GPL):** permite la libertad de usar, copiar y modificar el software, favoreciendo el estudio y ampliación del producto original. Ampliamente usado por la comunidad del software libre. Licencia estrechamente relacionada con el *Copyleft*. Usualmente no se asocia un pago por el uso de software bajo este modelo de licenciamiento.
- **Mozilla Public License:** parte desde el GNU GPL, pero posibilita la reutilización no libre del software, es decir, uso privativo que requiere pago por su uso y puede restringir su distribución.

2. Licencia de software propietario

Tradicionalmente, no se permite que el software sea modificado, no se autoriza el acceso al código fuente y se busca prevenir la piratería de software, distribución no autorizada (Licencia vinculada al *Copyright*). Estas se pueden incorporar a la compra de un hardware o por volumen.

En Colombia, las entidades encargadas de la propiedad intelectual del software son:

- Dirección Nacional de Derechos de Autor
- DIAN

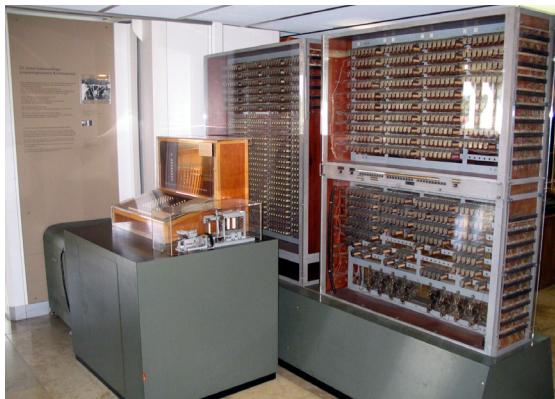


2 Tema

Breve historia de los sistemas operativos

La historia de los sistemas operativos está estrechamente ligada a la evolución del hardware, de las innovaciones microelectrónicas y las generaciones de los computadores (Wolf, Ruiz, Bergero, & Meza Vega, 2014).

Estudiemos un poco más esta historia:



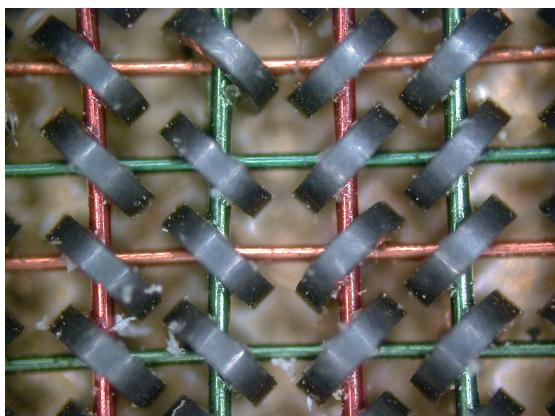
1950

En la primera generación de computadores, la interacción se da por manejo de interruptores que llevan las órdenes directamente a la memoria. En la década de 1950, la automatización se propicia con las tarjetas perforadas que permitían realizar diferentes tareas de cálculo, introducción al manejo de procesos por lotes. La evolución de este proceso se da gracias a sistemas en lotes con dispositivos de carga, más conocidos como *spool*.



1954

En abril de 1954, surge el primer sistema operativo desarrollado para la máquina IBM 704, controlando la ejecución secuencial de programas.



1960

Para la década de 1960, el poder de procesamiento de los computadores aumenta gracias al circuito integrado, favoreciendo la programación multitarea o sistemas multiprogramados, que maximizan el tiempo de uso efectivo del procesador, ejecutando múltiples procesos simultáneamente.



1970

A inicios de 1970, la empresa Xerox introduce el mouse y la metáfora del escritorio para gestionar el trabajo en el sistema operativo con su modelo Xerox Alto. Si bien no se trató de un producto comercializado, a partir de este momento, la interfaz gráfica predomina en los sistemas operativos. En esta misma época, se extienden las características de multitarea y multiusuario.

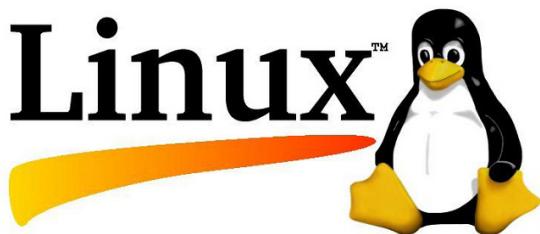
En esta época se destacan los sistemas operativos MS-DOS (Micro Soft Disk Operating System), para IBM PC, equipos con procesador Intel y UNIX.



1980

A inicios de la década de 1980, los grandes colosos del software actual incursionan fuertemente en el mercado. Apple, con su Macintosh en 1984, propone con gran éxito una interfaz gráfica multipanel en el sistema operativo Mac OS. Por otra parte, Microsoft, en 1985, lanza Windows 1.0. Surge, además, la comunidad de software libre.

1990



A principio de los años 90, la masiva adopción de computadores personales para hogares y pequeñas empresas, popularizó los sistemas informáticos para el público sin conocimientos especializados o entrenamiento en manejo de tecnología. El rápido crecimiento del mercado abre la oportunidad de explotación comercial y son las interfaces gráficas, caracterizadas por su fácil uso, las que propician la mayor demanda de sistemas operativos. En esta era

se da una convergencia en el trabajo empresarial y casero, se desarrollan plenamente las computadoras portátiles y predomina la arquitectura de IBM y Apple Macintosh. En este periodo, aparece Linux en 1991. Los sistemas informáticos abarcan múltiples esferas del sector productivo, desde base de datos hasta aplicaciones de escritorio especializadas.



Siglo XXI

Para inicios del siglo XXI, Apple lanza al mercado su sistema operativo Mac OS X, basado en UNIX, se propician derivaciones del kernel de Linux y Microsoft es líder en el mercado. El Internet y los dispositivos móviles dan origen a grandes revoluciones. Apple presenta una gran innovación en hardware y software con el iPhone, lo mismo que su principal competidor, Android, basado en Linux; ambos predominan en el mercado móvil.



Tema **3**

Arquitectura de un sistema operativo

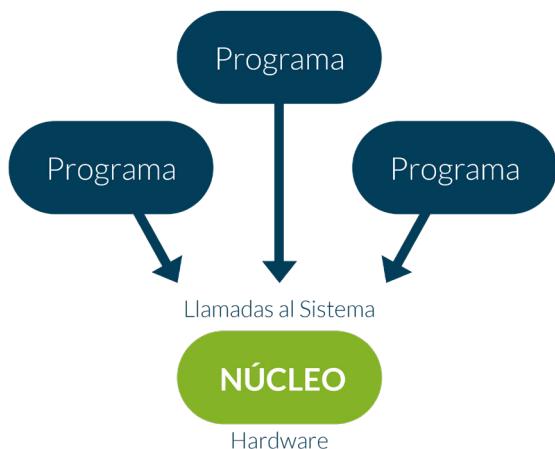
De acuerdo con la forma en que se distribuye la gestión de los recursos, el sistema operativo puede manejar una arquitectura modular, comúnmente subdividida en:

- Control de programas y bibliotecas
- Administrador de archivos y memoria
- Gestión de procesos e interrupciones
- Controladores de dispositivos

De acuerdo con Tanenbaum (2003), dicho modelo arquitectónico del software se puede estructurar de acuerdo con la cantidad de funcionalidades implementadas en el núcleo: monolítica, jerárquica, máquina virtual y por funciones.

Estructura monolítica

El sistema operativo se ejecuta en un exclusivo bloque, donde los procedimientos son llamados por un único programa principal. Se puede visualizar este proceso como una pila de datos, es decir, una estructura dinámica que acumula nuevos elementos, pero todos ellos ubicados en un mismo contenedor. MS DOS de Microsoft y UNIX tienen esta estructura.



Estructura jerárquica o por capas

El sistema operativo se distribuye en seis niveles, estructurados en cascada, es decir, una capa depende exclusivamente de la inferior (Dijkstra, 1983).

Estos niveles o capas son:

- **Capa 0:** nivel de hardware que administra la asignación de tareas al procesador.
- **Capa 1:** segmento de control o gestión de memoria.
- **Capa 2:** procesador secuencial.
- **Capa 3:** control de dispositivos E/S.
- **Capa 4:** programas de usuario.
- **Capa 5:** operador o interfaz de usuario.



Estructura de máquina virtual

Es una estructura que permite que un sistema operativo funcione, aisladamente, sobre otro sistema operativo. El hardware es compartido por múltiples sistemas operativos simultáneamente. Se cuenta, entonces, con un anfitrión (sistema operativo principal) y software de virtualización o hipervisor.

Ejemplos: *VMware*, *VirtualBox*.

El sistema operativo Linux ofrece un modelo alternativo a las máquinas virtuales, conocido como contenedores, donde se comparte el kernel, optimizando el uso del hardware, como docker.



Estructura por funciones

La estructura se define según el tipo de servicio que solicita acceso al recurso. Las capas superiores (aplicaciones) contienen las instrucciones para administración de recursos, como sucede en el modelo cliente-servidor.





Esta licencia permite a otros distribuir, remezclar, retocar, y crear a partir de esta obra de manera no comercial y, a pesar que sus nuevas obras deben siempre mencionar a la IU Digital y mantenerse sin fines comerciales, no están obligados a licenciar obras derivadas bajo las mismas condiciones.

