**II\_7**

**SISTEMAS OPERATIVOS S.O.**

Software q comprende un conjunto de programas que permiten la comunicación entre usuario y dispositivo. Administra los recursos de hardware del sistema. Empieza a funcionar en el momento en que encendemos nuestro dispositivo.

Administra también los dispositivos de entrada y salida, colas de procesos y los recursos de los dispositivos.

**Es el soporte lógico que controla el funcionamiento del equipo físico.**

**SERVIDORES**

Podemos encontrar S.O. como UNIX, RED HAT, WINDOWS SERVER, DEBIAN

**Se diferencian en el uso…** los SO de servidores son multiusuarios a diferencia de las compu domésticas que son monousuario.

**CLASIFICACION SEGÚN LA LICENCIA**

-OPEN SOURCE: permiten modificar, usar y adaptar un SO .Ej UBUNTU Y RED HAT

-Propietary software: no se pueden modificar. Ej WINDOWS

**RECURSOS ADMINISTRADOS POR EL SO**

-Gestionar la **memoria** de acceso aleatorio y ejecutar las app, designando recursos necesarios.

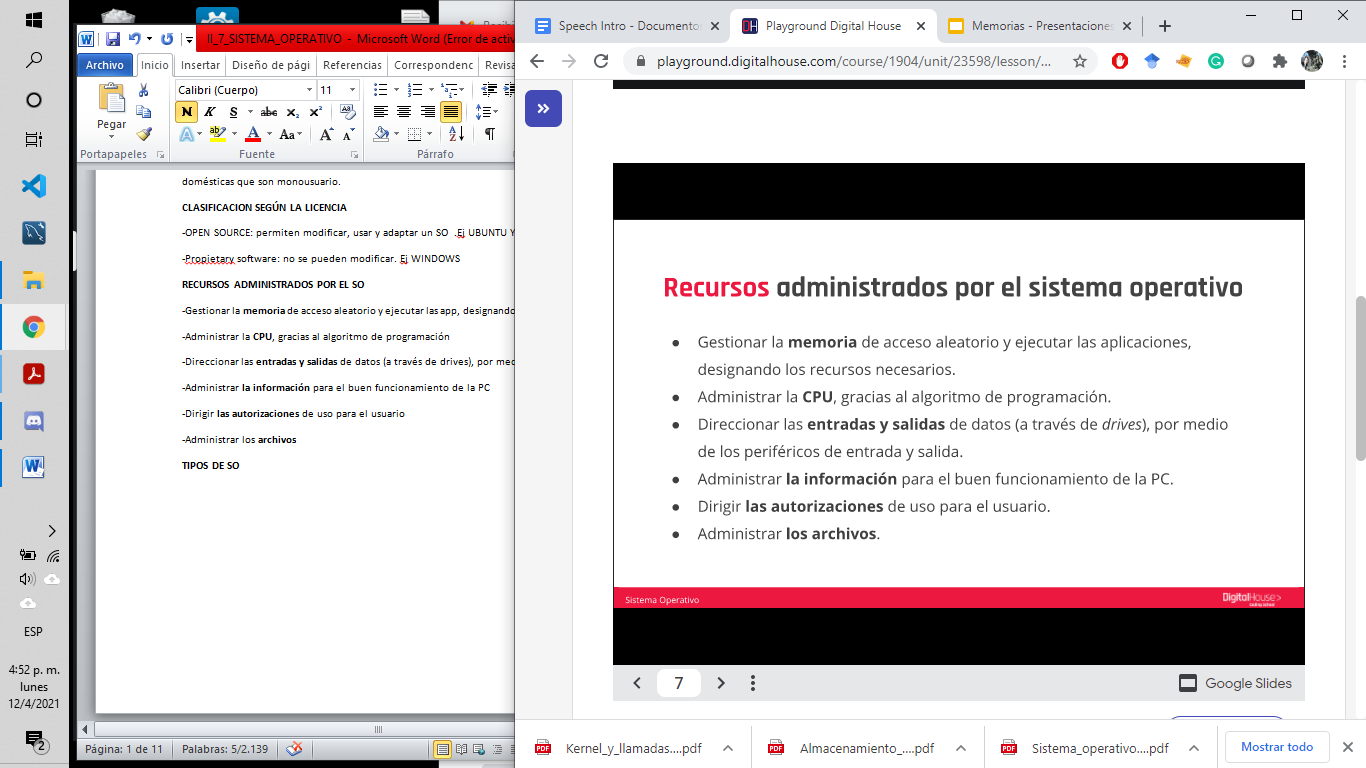
-Administrar la **CPU**, gracias al algoritmo de programación

-Direccionar las **entradas y salidas** de datos (a través de drives), por medio de los periféricos.

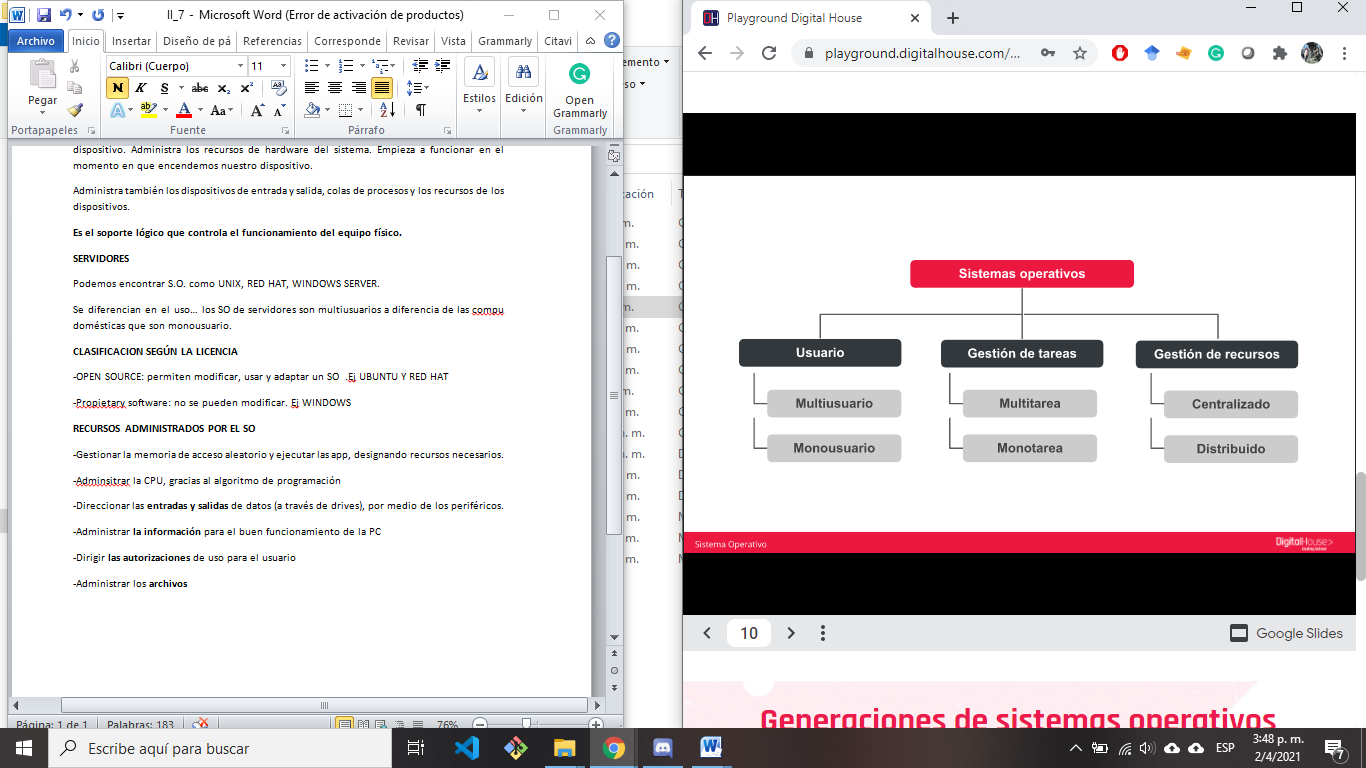
-Administrar **la información** para el buen funcionamiento de la PC

-Dirigir **las autorizaciones** de uso para el usuario

-Administrar los **archivos**



**TIPOS DE SO**



**Multitarea:** puede ejecutar varios procesos al mismo tiempo.

**Centralizado:** sólo permite utilizar los recursos de un solo ordenador.

**Distribuido:** permite ejecutar los procesos de más de un ordenador al mismo tiempo.

**GENERACIÓN DE SO**

|  |
| --- |
|  |
| **Generación Cero (década de 1940)** |
| Las computadoras electrónicas digitales **no tenían sistema operativo**. Los programas, por lo regular, manejaban un bit a la vez, en columnas de switchs mecánicos. Los programas de lenguaje máquina manejaban tarjetas perforadas. |

|  |
| --- |
| **Primera generación (1945–1955)** |
| **Tubos de vacío y tableros enchufables**  Se lograron construir máquinas calculadoras usando tubos de vacío. Estas máquinas eran enormes y ocupaban cuartos enteros con decenas de miles de tubos de vacío, pero eran mucho más lentas que incluso las computadoras personales más baratas de la actualidad. Toda la programación se realizaba en lenguaje de máquina absoluto. |

|  |
| --- |
| **La segunda generación (1955–1965)** |
| **Transistores y sistemas de lote**  Estas máquinas se encerraban en cuartos de computadora con acondicionamiento de aire especial. Para ejecutar un programa, un programador escribía primero el programa en papel (en FORTRAN o ensamblador) y luego lo **perforaba en tarjetas**. Después, llevaba el grupo de tarjetas al cuarto de entrada y lo entregaba a uno de los operadores. Cuando la computadora terminaba el trabajo que estaba ejecutando en ese momento, se separaba la salida impresa y se llevaba al cuarto de salida donde el programador podía buscarla. Luego, el operador tomaba uno de los grupos de tarjeta traídos del cuarto de entrada y lo introducía en el lector. Si se requería el compilador de FORTRAN, el operador tenía que traerlo de un archivero e introducirlo en el lector.  Dado el alto costo del equipo, la solución que se adoptó generalmente fue el sistema por lotes. El principio de este modo de operación consistía en juntar una serie de trabajos en el cuarto de entrada, leerlos y grabarlos en una cinta magnética usando una computadora pequeña y (relativamente) económica.  Después de cerca de una hora de reunir un lote de trabajos, la cinta se rebobinaba y se llevaba al cuarto de la máquina, donde se montaba en una unidad de cinta. El operador cargaba entonces un programa especial, que leía el primer trabajo de la cinta y lo ejecutaba. La salida se escribía en una segunda cinta, en lugar de imprimirse. Cada vez que terminaba un trabajo, el sistema operativo leía automáticamente el siguiente trabajo de la cinta y comenzaba a ejecutarlo. |

**Tercera generación (1965–1970)**

**Tercera generación (1965–1970)**

**Circuitos integrados ( CI ) y multiprogramación**

Las máquinas diferían solo en el precio y el rendimiento (memoria máxima, velocidad del procesador, número de dispositivos de E/S permitidos, entre otros). IBM trató de resolver simultáneamente ambos problemas introduciendo la System/360, puesto que todas las máquinas tenían la misma arquitectura y conjunto de instrucciones, los programas escritos para una máquina podían ejecutarse en todas las demás, al menos en teoría.

Los 360 y los sistemas operativos de tercera generación parecidos a él producidos por otros fabricantes de computadoras lograron satisfacer a sus clientes en un grado razonable y también popularizaron varias técnicas clave que no existían en los sistemas operativos de la segunda generación. Tal vez la más importante de ellas haya sido la multiprogramación.

El problema era el tiempo de espera, la solución a la que se llegó fue dividir la memoria en varias secciones, con un trabajo distinto en cada partición. Mientras un trabajo estaba esperando que terminara su E/S, otro podía estar usando la CPU. Si se podían tener en la memoria principal suficientes trabajos a la vez, la CPU podía mantenerse ocupada casi todo el tiempo. También, tenían la capacidad de leer trabajos de las tarjetas al disco tan pronto como se llevaban al cuarto de computadoras. Luego, cada vez que un trabajo terminaba su ejecución, el sistema operativo podía cargar uno nuevo del disco en la partición que había quedado vacía y ejecutarlo.

**Cuarta generación (1980– a nuestros días)**

1

**Cuarta generación (1980– a nuestros días)**

**Computadoras personales**

Con la invención de los circuitos integrados a gran escala (LSI), chips que contienen miles de transistores en un cm2 de silicio, nació la era de la computadora personal.

Dos sistemas operativos dominaron inicialmente el campo de las computadoras personales y las estaciones de trabajo: MS-DOS de Microsoft y UNIX. MS-DOS se usaba ampliamente en la IBM PC y otras máquinas basadas en la CPU Intel 8088 y sus sucesoras. Más tarde, la Pentium y Pentium Pro. Aunque la versión inicial de MS-DOS era relativamente primitiva, versiones subsecuentes han incluido características más avanzadas, muchas de ellas tomadas de UNIX. El sucesor de Microsoft para MS-DOS, Windows, originalmente se ejecutaba encima de MS-DOS, pero a partir de 1995 se produjo una versión autosuficiente de WINDOWS.

El otro competidor importante es UNIX, que domina en las estaciones de trabajo y otras computadoras del extremo alto, como los servidores de red. **UNIX es popular sobre todo en máquinas basadas en chips RISC de alto rendimiento.**

**QUIZ**

[1](https://playground.digitalhouse.com/)

**[¿Un sistema operativo es el soporte físico que controla el funcionamiento del equipo lógico?](https://playground.digitalhouse.com/)**

Opción 2: Falso.

**Corrección**  ¡Buen trabajo!

[2](https://playground.digitalhouse.com/)

**[Los recursos fundamentales que administra el sistema operativo son:](https://playground.digitalhouse.com/)**

Opción 1: El procesador.

Opción 4: La información.

Opción 3: La memoria.

**Corrección**  ¡Buen trabajo!

[3](https://playground.digitalhouse.com/)

**[El sistema operativo permite que varios usuarios ejecuten simultáneamente sus programas es:](https://playground.digitalhouse.com/)**

Opción 2: Multiusuario.

**Corrección**  ¡Buen trabajo!

[4](https://playground.digitalhouse.com/)

**[¿Los transistores y sistemas de lote con qué generación/es corresponden?](https://playground.digitalhouse.com/)**

Opción 2: Segunda generación.

**Corrección**  ¡Buen trabajo!

[5](https://playground.digitalhouse.com/)

**[¿Cuál de la siguiente lista es un sistema operativo para servidores?](https://playground.digitalhouse.com/)**

Opción 1: Red Hat.

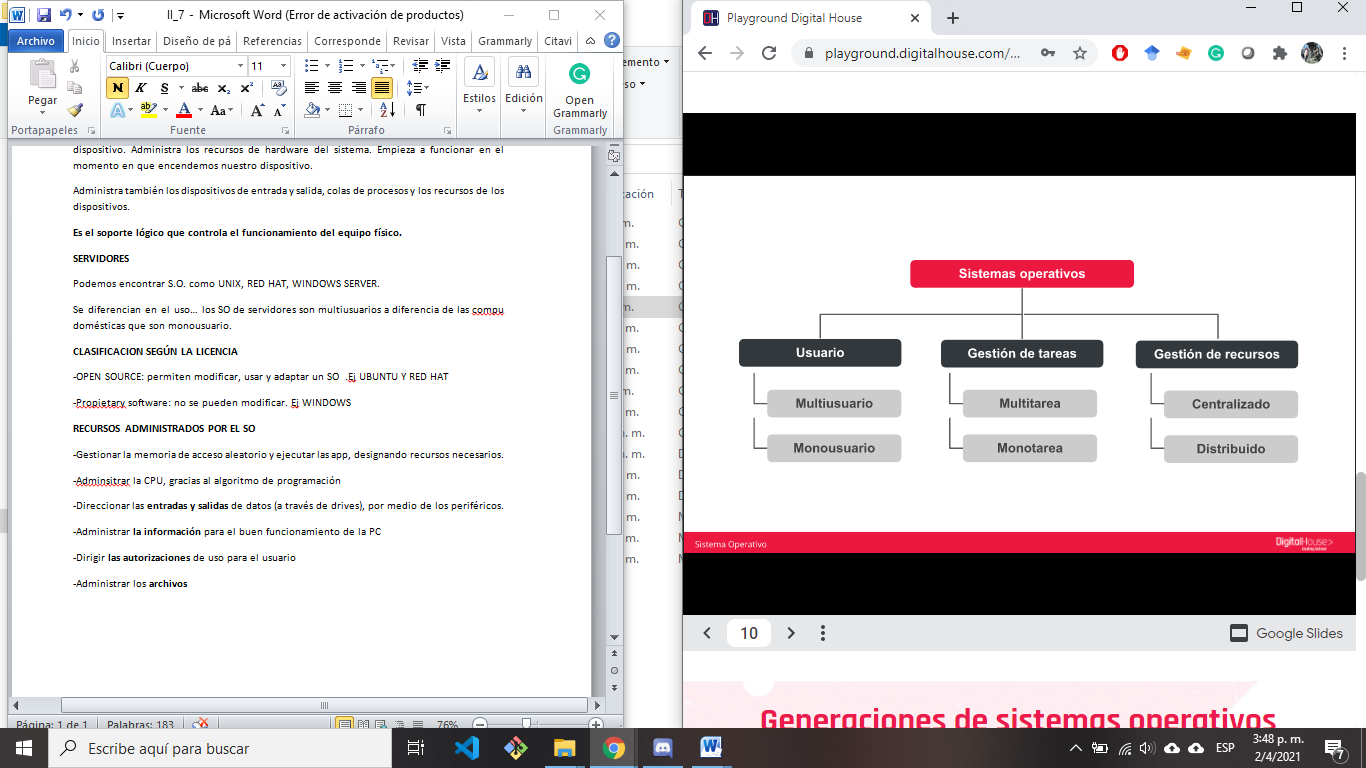
Opción 2: Microsoft Windows Server.

Opción 3: Debian.

**Corrección**  ¡Buen trabajo!

**CLASIFICACIÓN DE LOS SO**

**TIPOS DE SO**



GESTIÓN DE TAREAS

**Monotarea:** sólo una tarea a la vez sin que se pueda interrumpir. Ej Windows me y Windows vista. Para imprimir un archivo sólo se puede hacer eso hasta q termine. EJ DOS

**Multitarea:** puede ejecutar varios procesos al mismo tiempo. Windows, Unix, Linux, Mac OSX

USUARIOS

**Monousuario:** EJ Windows (hasta Me) - DOS

**Multiusuario:** por **medio de terminales conectadas a la compu o de manera remota.** pueden ser usuarios de manera remota. Ej UNIX, LINUX, SOLARIS. Windows (a partir de XP), Unix, Linux , Mac OSX

ESTRUCTURA INTERNA

**Estructura monolítica:** constituido x un solo programa con una serie de rutinas internas entrelazadas entre sí, para comunicarse entre ellas. Suelen ser hechos a medida, por lo que son muy rápidos pero no tienen flexibilidad para soportar distintos tipos de app. Ej: VMS, Linux, Multics, Windows (hasta Me)

**Estructura jerárquica:** una parte del sistema tiene subpartes, se subdivide en capas o jerarquías. Ej Unix, Multics

**Máquina virtual:** objetivo de integrar distintos SO dando la sensación de ser varias máquinas diferentes. Ej Microsoft Hyper-V, VMware, VirtualBox, QEMU, Kernel-Based Virtual machine

**Cliente-servidor:** es de propósito gral. Desventaja: no resuelve problemas de compartir información

GESTION DE RECURSOS

**Centralizado:** sólo permite utilizar los recursos de un solo ordenador. Ej Windows, Unix, Linux, Mac OSX

**Distribuido:** permite ejecutar los procesos de más de un ordenador al mismo tiempo. Ej Novell Netware, Windows Server, Cisco IOS, Unix, Linux

**QUIZ**

[1](https://playground.digitalhouse.com/)

**[Los sistemas operativos que nos permiten realizar varias tareas al mismo tiempo son:](https://playground.digitalhouse.com/)**

Opción 2: Multitarea.

**Corrección**  ¡Buen trabajo!

[2](https://playground.digitalhouse.com/)

**[¿Cuál es la estructura del sistema operativo que está constituido por un solo programa compuesto de una serie de rutinas entrelazadas entre sí?](https://playground.digitalhouse.com/)**

Opción 3: Estructura monolítica.

**Corrección**  ¡Buen trabajo!

[3](https://playground.digitalhouse.com/)

**[¿Cuál es el sistema operativo que tiene la estructura más reciente de todos?](https://playground.digitalhouse.com/)**

Opción 4: Cliente-servidor.

**Corrección**  ¡Buen trabajo!

[4](https://playground.digitalhouse.com/)

**[El objetivo de los sistemas operativos de \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ es el de integrar distintos sistemas operativos dando la sensación de ser varias máquinas diferentes.](https://playground.digitalhouse.com/)**

Opción 2: Máquina virtual.

**Corrección**  ¡Buen trabajo!

**KERNEL Y LLAMADAS AL SISTEMA**

**KERNEL**: parte del SO (cerebro) encargado de interactuar entre las distintas apps y sus necesidades con los distintos recursos que requiere para ejecutarlos.

CAPA FUNDAMENTAL DEL SO, ES EL ENCARGADO DE COMUNICAR Y ADMINISTRAR LOS RECURSOS DE LA PC, COMO LA RAM O EL USO DE LA CPU.

-Asigna y prioriza recursos de hardware.

-Asigna prioridad según las necesidades del SO.

Las interacciones se llevan a cabo a través de Llamadas al sistema. son el método que tienen las app para solicitar un servicio o recurso.

Kernel está dentro del SO y es una parte esencial del mismo.

Windows tiene su propio kernel

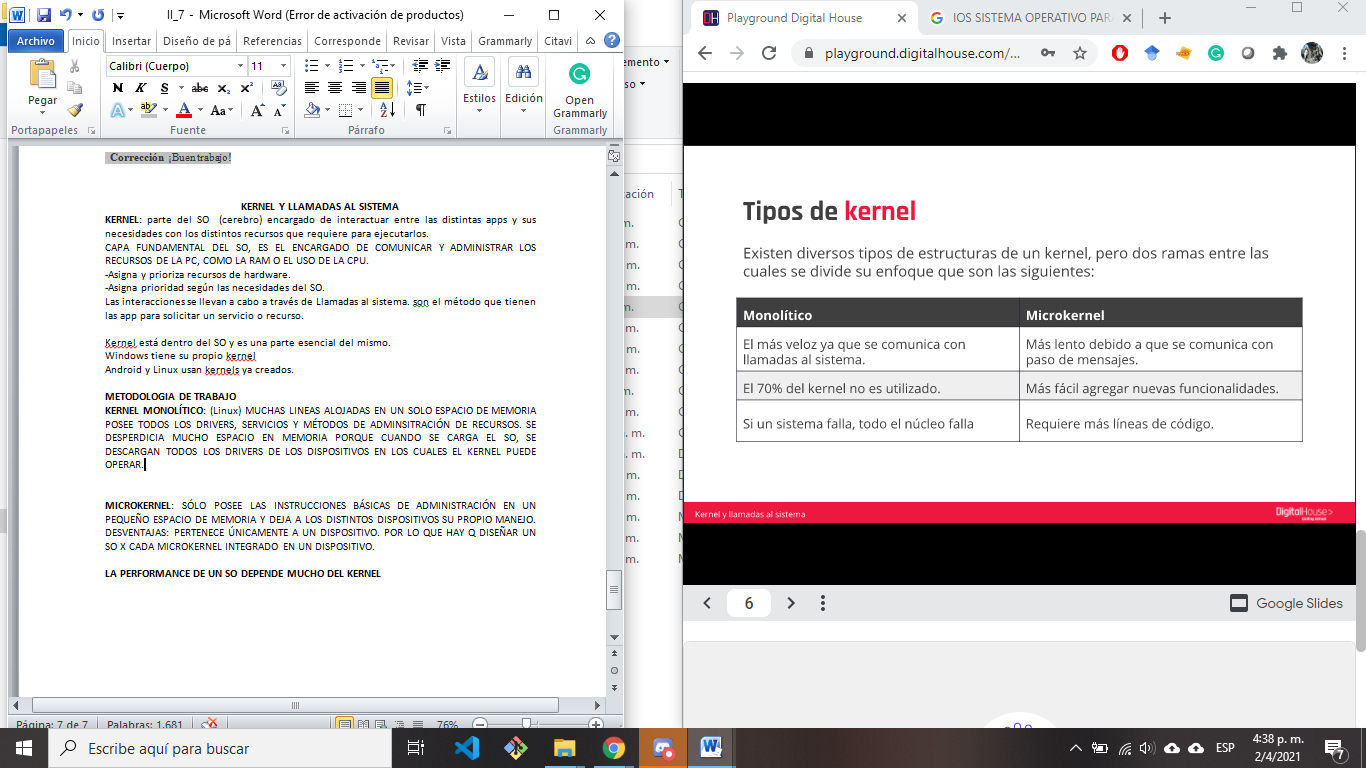
Android y Linux usan kernels ya creados.

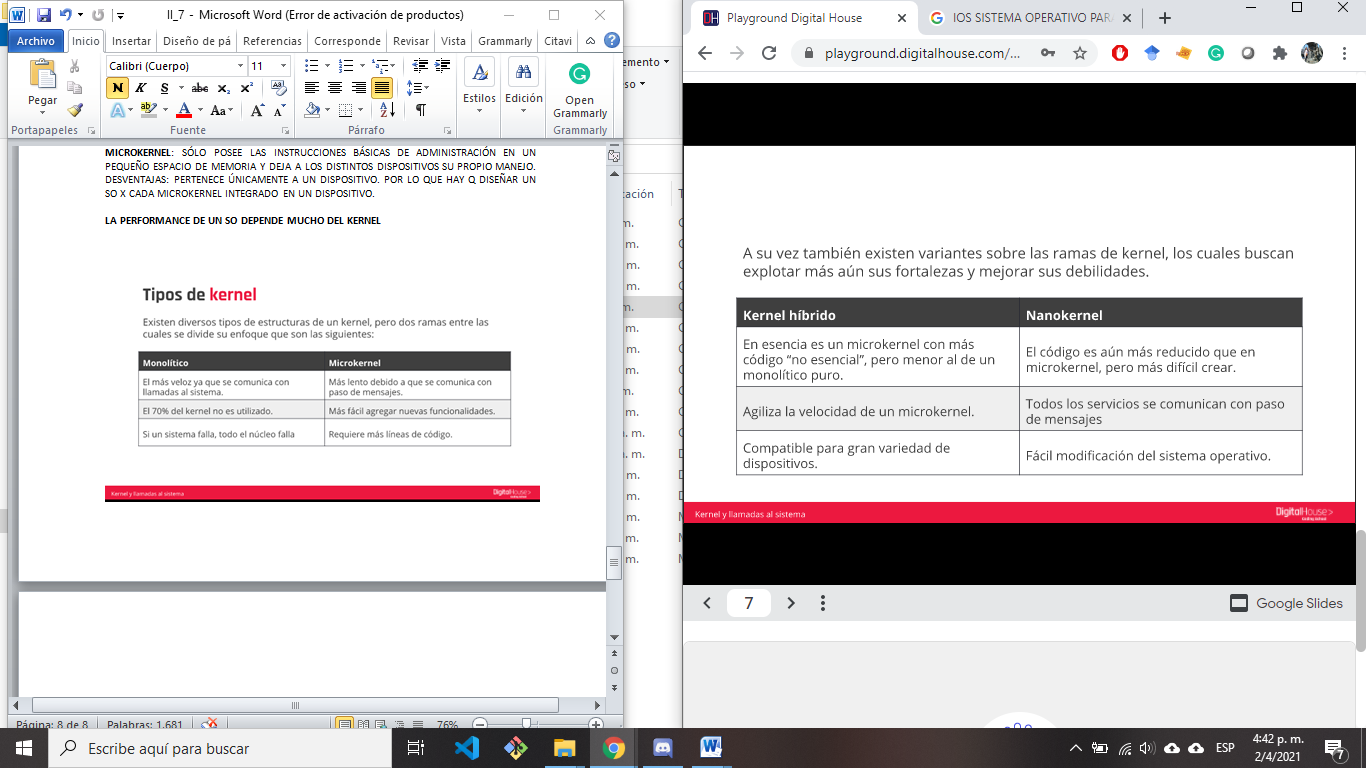
**METODOLOGIA DE TRABAJO**

**KERNEL MONOLÍTICO**: (Linux) MUCHAS LINEAS ALOJADAS EN UN SOLO ESPACIO DE MEMORIA POSEE TODOS LOS DRIVERS, SERVICIOS Y MÉTODOS DE ADMINSITRACIÓN DE RECURSOS. SE DESPERDICIA MUCHO ESPACIO EN MEMORIA PORQUE CUANDO SE CARGA EL SO, SE DESCARGAN TODOS LOS DRIVERS DE LOS DISPOSITIVOS EN LOS CUALES EL KERNEL PUEDE OPERAR.

**MICROKERNEL**: SÓLO POSEE LAS INSTRUCCIONES BÁSICAS DE ADMINISTRACIÓN EN UN PEQUEÑO ESPACIO DE MEMORIA Y DEJA A LOS DISTINTOS DISPOSITIVOS SU PROPIO MANEJO. DESVENTAJAS: PERTENECE ÚNICAMENTE A UN DISPOSITIVO. POR LO QUE HAY Q DISEÑAR UN SO X CADA MICROKERNEL INTEGRADO EN UN DISPOSITIVO.

**LA PERFORMANCE DE UN SO DEPENDE MUCHO DEL KERNEL**

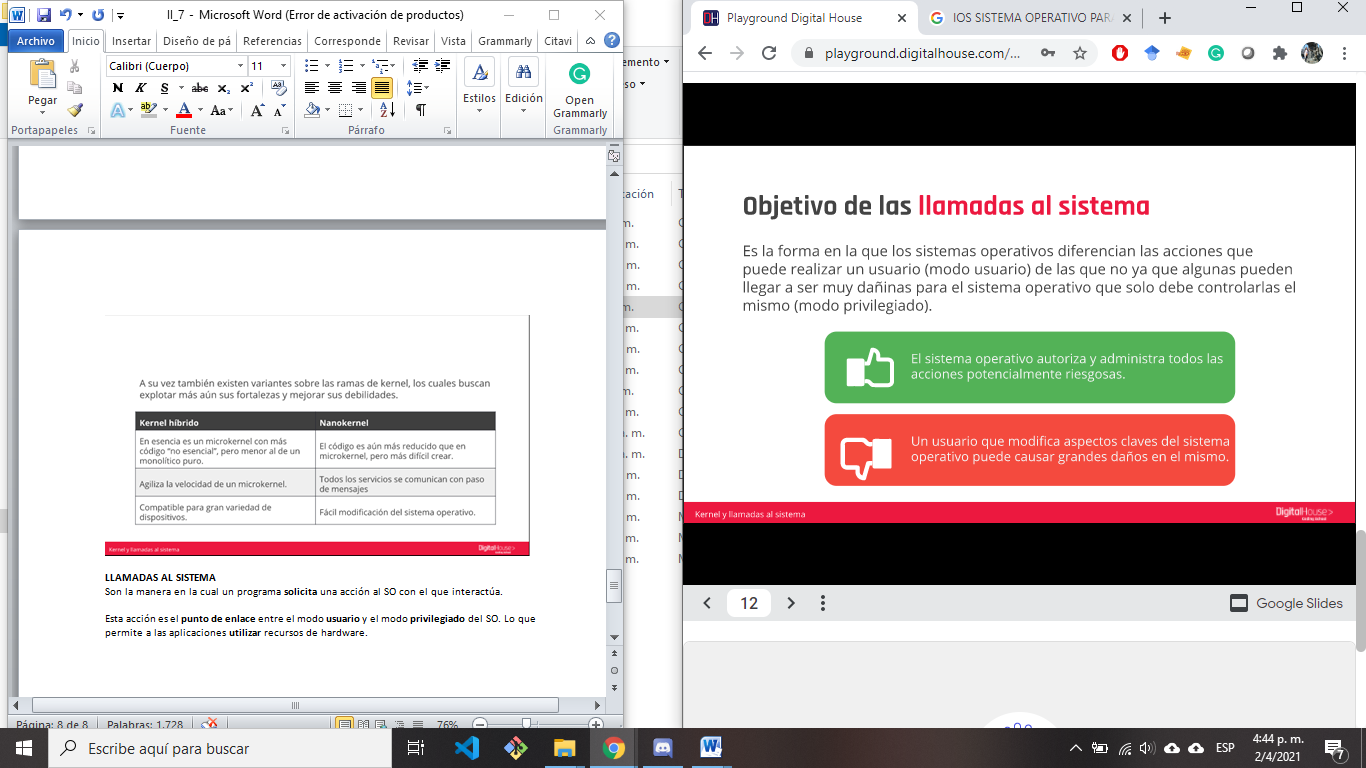


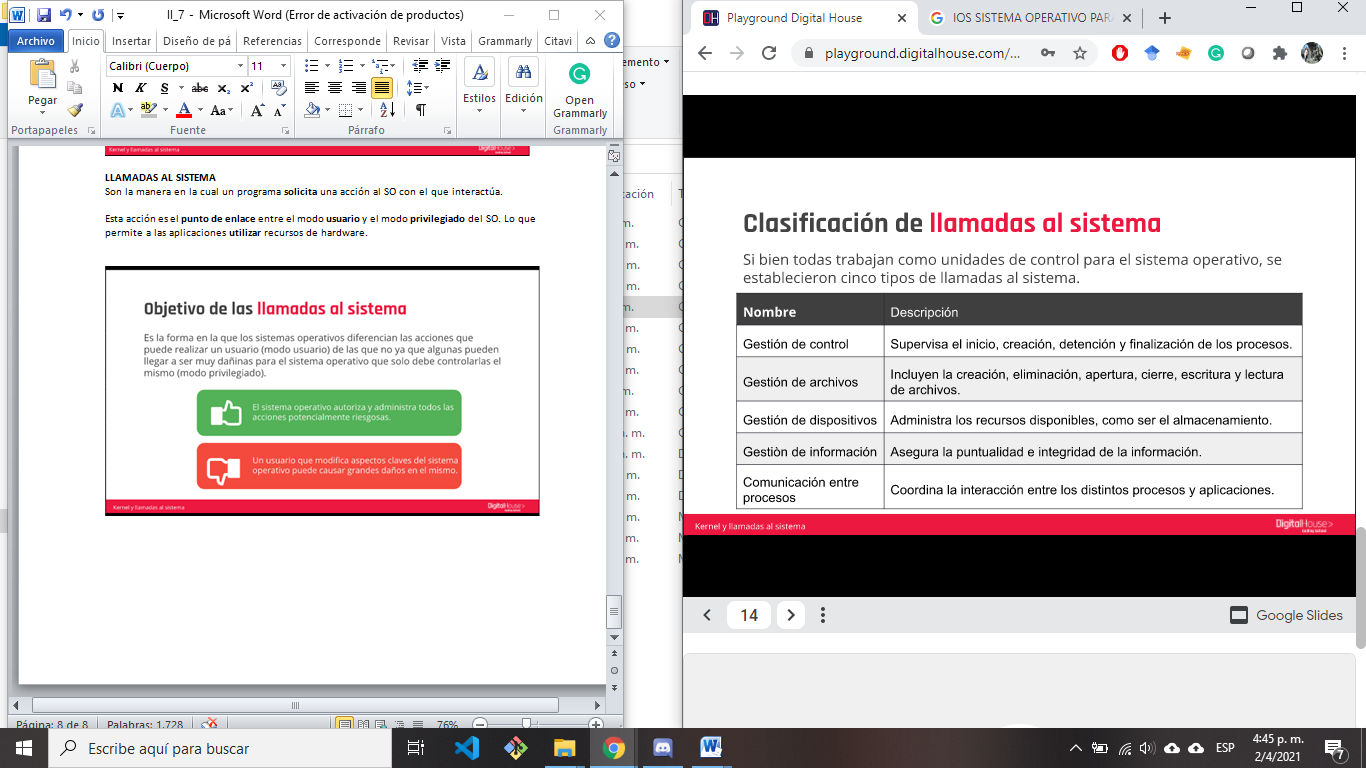


**LLAMADAS AL SISTEMA**

Son la manera en la cual un programa **solicita** una acción al SO con el que interactúa.

Esta acción es el **punto de enlace** entre el modo **usuario** y el modo **privilegiado** del SO. Lo que permite a las aplicaciones **utilizar** recursos de hardware.





QUIZ

[1](https://playground.digitalhouse.com/)

[¿Qué es una llamada al sistema?](https://playground.digitalhouse.com/)

Mecanismo para solicitar una acción del sistema operativo.

**Corrección**   ¡Excelente! Es la manera que tienen las aplicaciones para interactuar con el sistema operativo.

[2](https://playground.digitalhouse.com/)

[¿Cómo se dividen los kernels?](https://playground.digitalhouse.com/)

Monolítico.

Microkernel

**Corrección**  ¡Excelente! Estos son los dos grandes pilares.

[3](https://playground.digitalhouse.com/)

[¿Conectar un mouse USB genera una llamada al sistema?](https://playground.digitalhouse.com/)

Opción 2: Verdadero

**Corrección**   ¡Correcto! la detección de hardware es uno de los ejemplos más visibles de las llamadas al sistema.

[4](https://playground.digitalhouse.com/)

[El microkernel de una PC de escritorio funciona en una notebook.](https://playground.digitalhouse.com/)

Opción 1: Verdadero

**Corrección**  ¡Muy bien! Siempre y cuando el hardware sea de las mismas características podrá funcionar correctamente.

[5](https://playground.digitalhouse.com/)

[La gestión de control otorga permisos de lectura y escritura en archivos](https://playground.digitalhouse.com/)

Opción 2: Falso

**Corrección**  ¡Muy bien! El gestor de archivos tiene la responsabilidad sobre los mismos en el sistema operativo.

**CLASE 5/04**

**DUDA**: PORQUE EL MICROKERNEL DE UNA PC DE ESCRITORIO FUNCIONA EN UNA NOTEBOOK?

Una variante del microkernel, es el kernel híbrido.

El microkernel carga poca información.

Nanokernel es un proyecto de investigación de una universidad.

El Kernel no se encarga de la interfaz de usuario.

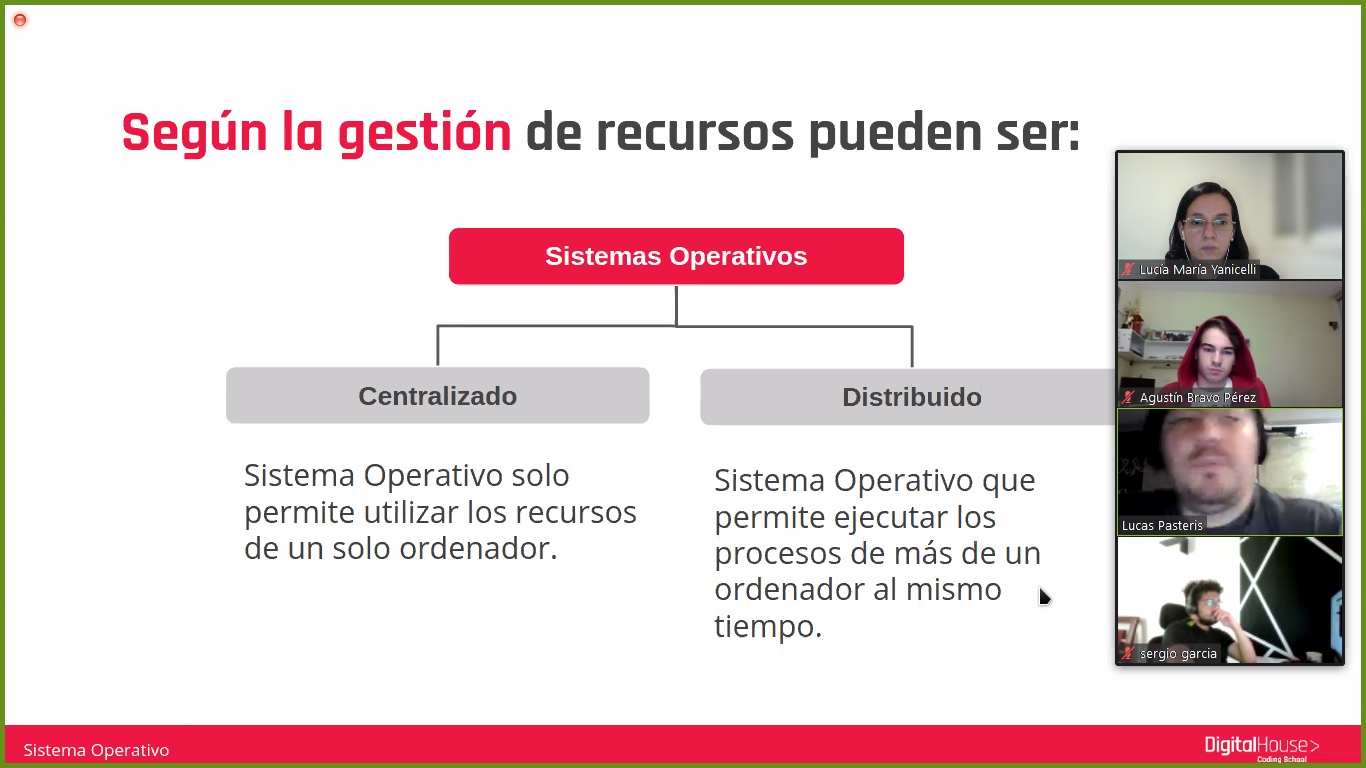
Las tareas más críticas las maneja el Kernel y a las demás el resto del SO.

**Kahoot**

**Modo Privilegio:** se pueden ejecutar los grandes cambios q afectan a la funcionalidad del SO. Podemos generar actualizaciones, borrar carpetas del SO. El kernel trabaja sobre el modo privilegio y le da el poder de ejectuar comandos y acciones fuera de lo común vs **Modo root:** es la forma que se denomina al entorno privilegiado en MAC o LINUX. SUPER USUARIOS.

Centralizado: tiene todos los recursos en una sola computadora

Distribuido: el sistema de storage, almacenamiento x un lado, memoria por otro lado (físicamente). Puede manejar los recursos de los distintos lugares.



**REPASO**

**SO**: el intermediario entre el usuario y la máquina. Conjunto de programas. Hace el soporte lógico que controla el funcionamiento del equipo físico.

El usuario root de Linux tiene más permisos que el administrador de Windows.

TIPOS:

USUARIO:

MULTIUSUARIO VS MONOUSUARIO: podría ser Android en el celular donde sólo podés tener un usuario. Al final dicen que es multiusuario.

ESTRUCTURA INTERNA: monolítica, jerárquica, máquina virtual y cliente-servidor (para responder los pedidos de distintas máquinas)

GESTIÓN DE TAREAS: multitarea: multinucleo vs monotarea: mononúcleo

GESTIÓN DE RECURSOS: centralizados vs distribuidos

LINUX puede ser cliente-servidor, jerarquico, multiusuario, distribuido

PREGUNTA

Que hace el comando mv-r carpeta carpeta1?

mv-r erróneo. Sólo mv para renombrar una carpeta.