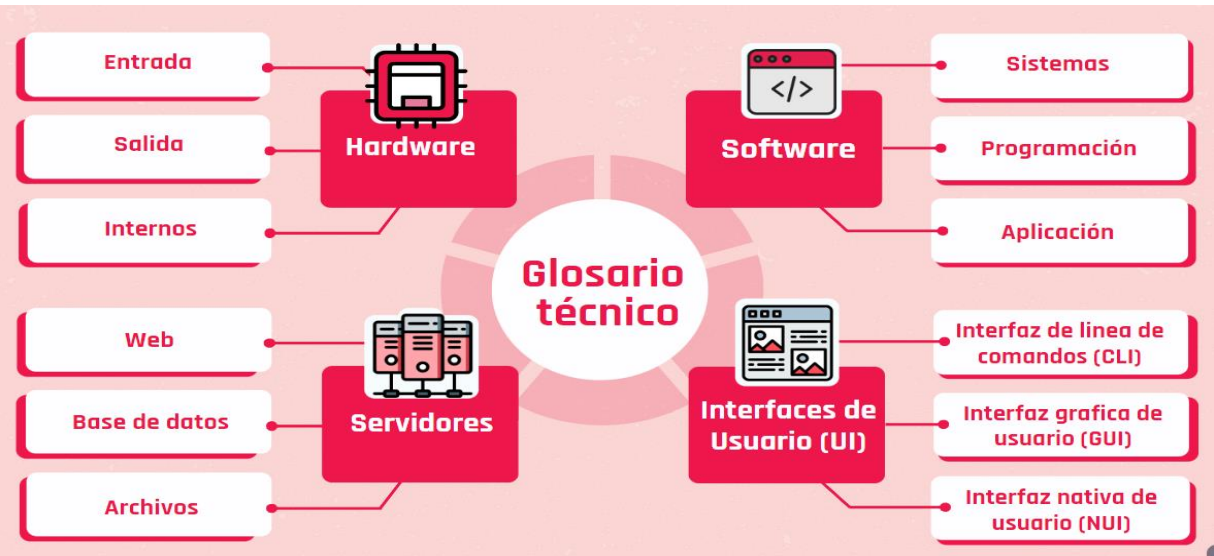
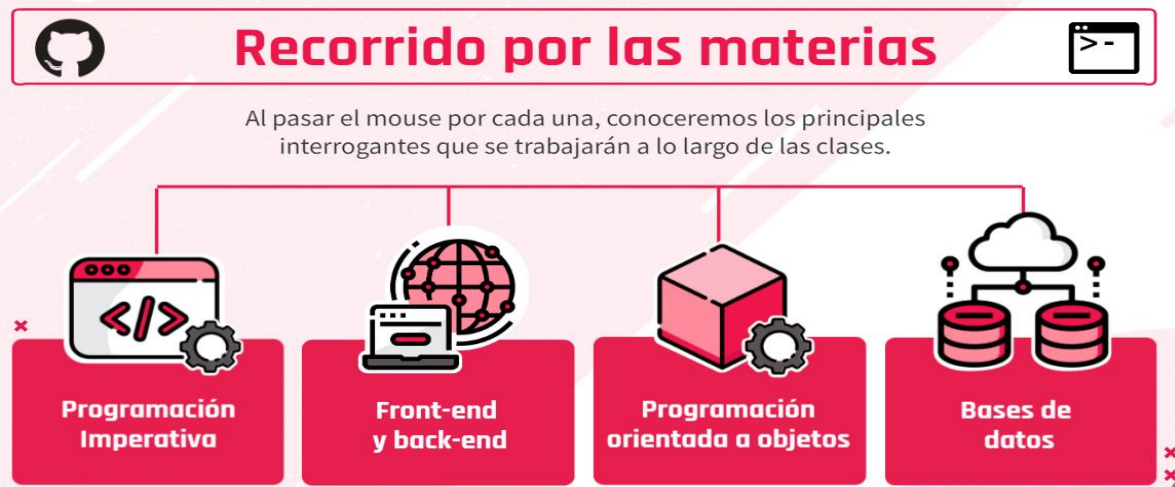


INTRODUCCIÓN A LA INFORMATICA



Glosario técnico

Hardware

Hardware

El hardware o equipo se refiere a las partes físicas, tangibles, de un sistema informático, sus componentes eléctricos, electrónicos, electromecánicos y mecánicos. Ejemplos de hardware son los cables, gabinetes, periféricos de todo tipo y cualquier otro elemento físico involucrado; contrariamente, el soporte lógico e intangible es el llamado software.

Entrada

Son aquellos componentes que permiten el ingreso de información, en general desde alguna fuente externa o por parte del usuario. Proveen el medio fundamental para transferir hacia la computadora (al procesador) información desde alguna fuente, sea local o remota. También permiten cumplir la tarea esencial de leer y cargar en memoria el sistema operativo y los programas informáticos, los que a su vez ponen operativa la computadora y hacen posible realizar las más diversas tareas.

Entre los periféricos de entrada se puede mencionar: teclado, mouse, escáner, micrófono, cámara web, joystick, lectoras de CD, DVD o BluRay, entre otros.

Salida

Son aquellos que permiten dar salida a la información resultante de las operaciones realizadas por la CPU. Los más comunes de este grupo son los monitores, las impresoras, las consolas, y los altavoces.

Software de sistema

Desvincula al usuario y al programador de los detalles del sistema informático en particular que se use, transparentando el procesamiento referido a las características internas de: memoria, discos, puertos y dispositivos de comunicaciones, impresoras, pantallas, teclados, etc. El software de sistema le procura al usuario y programador adecuadas interfaces de alto nivel, controladores, herramientas y utilidades de apoyo que permiten el mantenimiento del sistema global. Incluye entre otros:

- Sistemas operativos
- Controladores de dispositivos
- Herramientas de diagnóstico
- Herramientas de corrección y optimización
- Servidores
- Utilidades

Software de programación

Es el conjunto de herramientas que permite al programador desarrollar programas de informática, usando diferentes alternativas y lenguajes de programación, de una manera práctica. Incluyen en forma básica:

- Editores de texto
- Compiladores
- Intérpretes
- Enlazadores
- Depuradores
- Entornos de desarrollo integrados (IDE)

Internos

El hardware interno es el conjunto de componentes físicos que forman parte del dispositivo principal, siendo inseparable de este.

En otras palabras, cada parte del hardware interno es una pieza fundamental de cara al funcionamiento correcto del dispositivo. Ya que si faltara alguna de las partes de este, podría bien no ejecutar alguna tarea e incluso directamente no funcionar.

Ejemplos de hardware interno: Placa base, CPU, RAM, GPU, HDD, SSD.

Software de aplicación

Es aquel que permite a los usuarios llevar a cabo una o varias tareas específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido, con especial énfasis en los negocios. Incluye entre muchos otros:

- Aplicaciones para Control de sistemas y automatización industrial
- Aplicaciones ofimáticas
- Software educativo
- Software empresarial
- Bases de datos
- Telecomunicaciones (por ejemplo, Internet y toda su estructura lógica)
- Videojuegos
- Software de diseño asistido (CAD)

Servidores

Servidores

Un **servidor** es una computadora capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia. Se pueden ejecutar en computadoras dedicadas a las cuales se les conoce individualmente como "el servidor" o en la mayoría de los casos una misma computadora puede proveer múltiples servicios y tener varios servidores en funcionamiento.

Interfaces de usuario (UI)

Interfaces de usuario (UI)

La interfaz de usuario es el espacio donde se producen las interacciones entre seres humanos y máquinas. El objetivo de esta interacción es permitir el funcionamiento y control más efectivo del sistema informático desde la interacción con el humano.

Servidor web

Almacena documentos HTML, imágenes, archivos de texto, escrituras, y demás material web compuesto por datos (conocidos colectivamente como contenido) y distribuye este contenido a clientes que lo piden en la red.

Servidor de base de datos

Provee servicios de base de datos a otros programas u otras computadoras, como es definido por el modelo cliente-servidor. También puede hacer referencia a aquellas computadoras (servidores) dedicadas a ejecutar esos programas, prestando el servicio.

Interfaz de linea de comandos (CLI)

Interfaces alfanuméricas (intérpretes de comandos) que solo presentan texto.

Interfaz grafica de usuario (GUI)

Permiten comunicarse con la computadora de forma rápida e intuitiva representando gráficamente los elementos de control y medida.

Servidor de archivos

Es el que almacena varios tipos de archivos y los distribuye a otros clientes en la red.

Interfaz nativa de usuario (NUI)

Pueden ser táctiles, representando gráficamente un "panel de control" en una pantalla sensible al tacto que permite interactuar con el dedo de forma similar a si se accionara un control físico; pueden funcionar mediante reconocimiento del habla, como, por ejemplo, Siri; o mediante movimientos corporales, como es el caso de Kinect.

Introducción a la terminal

Cuando hablamos de terminal nos referimos a un software o programa contenedor que ejecuta un shell. Hace décadas, este era un dispositivo físico que consistía en poco más que un monitor y un teclado. Pero, ¿qué es un shell? Es la interfaz de línea de comandos con la que vamos a interactuar, se encarga de procesar datos y devolver resultados.

Entonces, una terminal es una interfaz que nos sirve para comunicarnos con una computadora, un teclado para entrada de datos y una pantalla para mostrar únicamente caracteres alfanuméricos (sin gráficos).

Y cuando hablamos de consola, ¿a qué nos referimos? Una consola es un *tipo especial de terminal*. Para empezar a escribir en la consola es necesario hacerlo a través de líneas de comandos, es decir, una instrucción u orden que nosotros (usuarios) le proporcionamos a un sistema informático. Es una instrucción específica dada a una aplicación informática para realizar algún tipo de tarea o función. La línea de órdenes permiten a los usuarios dar instrucciones por medio de textos sencillos como mostrar archivos, crear archivos, mostrar datos, llamar procesos, entre otros.

Ahora bien, cada sistema operativo incorpora un determinado número de comandos básicos, que permiten ejecutar las tareas más simples con órdenes directas. Esos comandos son propios y generalmente varían según el sistema operativo.

Veamos en mayor profundidad estos conceptos y los diversos comandos que podemos utilizar.

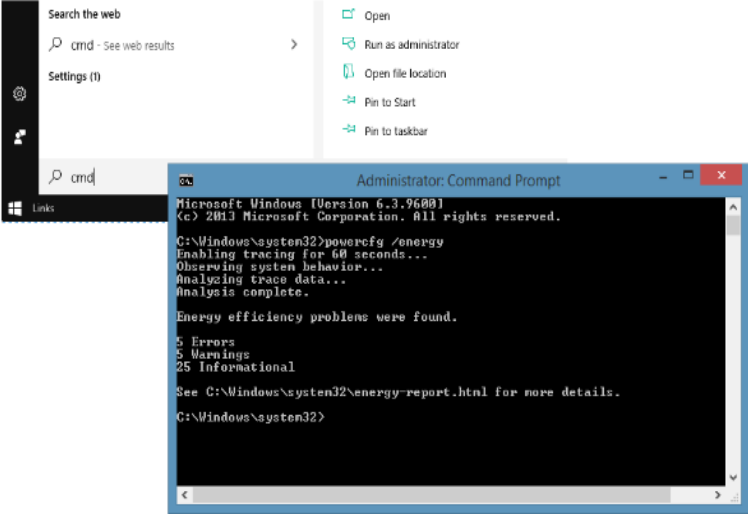
La terminal

Es un **programa** que está presente en **todos** los **sistemas operativos**. Usando la terminal podemos darle **órdenes** al sistema.

Todo eso que hacemos usando el mouse lo podemos lograr a través de comandos en la terminal, como por ejemplo, **crear** una **carpeta** o un **archivo**, mover un archivo de lugar y mucho más.

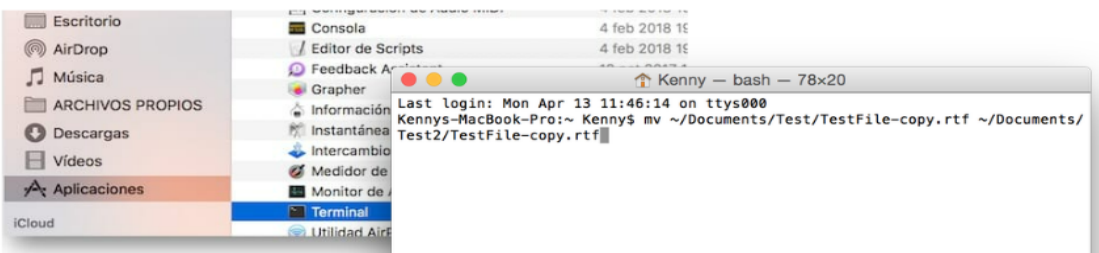
Terminal en Windows

Para abrir la terminal en Windows, podemos presionar las teclas Win+R y, en la ventana emergente, escribir el comando "cmd.exe". También podemos hacer uso del buscador y escribir la palabra "cmd" o "PowerShell".



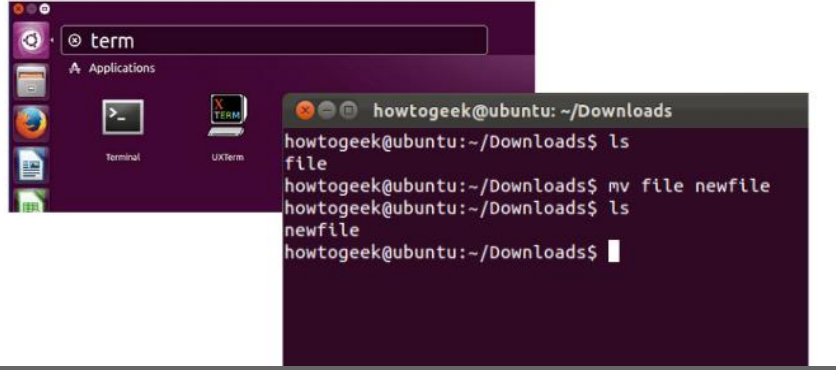
Terminal en Mac

Para abrir la terminal en Mac, podemos ir al Finder y pulsar en Archivo > Nueva ventana del Finder (⌘N). Luego, en el menú lateral izquierdo, hacer clic en Aplicaciones y buscar en el listado la terminal.



Terminal en Linux

Para abrir la terminal en Linux, podemos presionar las teclas Ctrl+Alt+T o simplemente ingresar la palabra "terminal" en el buscador que ofrece el sistema operativo.



Interfaz de línea de comandos (CLI)
Command line interface



Iterfaz gráfica de usuario (CUI)
Graphical user interface



VS.

Permite al usuario ingresar comandos para interactuar con el software y el sistema operativo.	Permite al usuario interactuar con los programas a través de una interfaz gráfica.
Requiere conocimientos técnicos de comandos y experiencia.	Es amigable, intuitiva y fácil de usar para usuarios principiantes.
Necesita poca memoria para su ejecución.	Requiere más memoria para ejecutar sus componentes gráficos.
Es muy rápida, ya que reduce al mínimo el uso de recursos y dependencias.	Ejecución más lenta debido a la capa gráfica y navegabilidad.
Flexible, permite combinar comandos.	Poco flexible, solo realiza las operaciones para la cual fue diseñada.
Se utiliza en servidores, clusters, desarrollo y seguridad.	Está orientado al público general y al uso cotidiano.
Mayor precisión.	Menor precisión.

Terminal ¿Qué podemos hacer con la terminal?

Somos libres de usar el sistema de ventanas de Linux, pero creemos que utilizar la terminal nos será muy provechoso.

Cada vez que ejecutamos una orden, el sistema operativo abre automáticamente tres interfaces de tipo estándar:

Entrada +

Salida +

Error +

Todos estos archivos se crean en el directorio /dev.

Entrada

La entrada estándar (stdin) se refiere al archivo por el que una orden recibe su entrada (por defecto, es el teclado)

Salida

La salida estándar (stdout) se refiere al archivo por el que una orden presenta sus resultados (por defecto, es la pantalla o más concretamente la ventana en la que se está ejecutando el intérprete de órdenes).

Error

El error estándar (stderr) se refiere al archivo por el que una orden presenta los mensajes que va generando cuando ocurre un error (por defecto, también es la pantalla).

Navegación e interacciones con ficheros

Del tipo informativo

pwd / Mostrar el nombre de la carpeta en el que uno se encuentra situado (print working directory).

cd / Cambiar la carpeta de trabajo: Con este comando nos podemos mover entre diferentes directorios. Si queremos ir a un directorio en particular.(para no devolver una por una lo puedes hacer así **cd ../../..**/ dependiendo de los niveles)

ls / Listar el contenido de directorios (list): Este comando lista los ficheros y carpetas.

Crear o eliminar carpetas

mkdir / Crear un carpeta (make directory).

mkdir nombrecarpeta

rmdir / Borrar un carpeta (remove directory)

Crear o eliminar archivos

gedit / Crear y editar archivos de texto.

rm / Borrar archivos.

Copiar archivos y directorios

cp / Copiar un archivo o carpeta en el directorio especificado (copy)

cp -r / Copiar carpetas

Lmv / Mover un archivo o carpeta a un archivo o carpeta (move)

3

Mostrar	
\$ ls	Lista archivos en el directorio o carpeta
\$ ls -a	Lista todos los archivos, incluyendo los archivos ocultos
\$ ls -l	Muestra toda la información de una carpeta: usuario, grupo, permisos, tamaño, fecha y hora de creación.
\$ ls -R	Muestra las carpetas y los archivos contenidos en ellos de manera recursiva
\$ pwd	Muestra la carpeta en la que se está trabajando actualmente
more [Nombre del archivo]	Muestra el contenido de un archivo

Copiar/Mover/Renombrar		
\$ mv [ruta/archivo1] [ruta/archivo2]	Renombra archivos (archivo2 no debe existir o será sobrescrito)	
\$ mv [ruta/carpeta1] [ruta/carpeta2]	Renombra la carpeta1 como carpeta2 (carpeta 2 no debe existir)	
\$ mv [ruta/carpeta1] [ruta/carpeta2] mv auditoria.xls auditoria	Mueve contenido de carpeta1 a carpeta2 (carpeta 2 debe existir)	
\$ cp [ruta/archivo1] [ruta/archivo2] \$ cp [ruta/archivo1] [ruta/archivo2]	Copia un archivo o carpeta	
opción: -r	Indica que copie recursivamente el contenido de las subcarpetas	

Navegación entre carpetas	
\$ cd ..	Sube un nivel de carpeta
\$ cd	Cambia de carpeta
\$ cd/chosen/	Cambia a una carpeta específica

directory	
-----------	--

Otros comandos	
\$ clear	Limpia la pantalla de la terminal
\$ comando --help	Muestra ayuda del comando
Git --version	Ver versión del git

Atajos de teclado	
\$ ctrl + c	Finaliza un proceso vigente que está corriendo en la terminal
\$ ctrl + l	Limpia la pantalla de la termina

Caracteres especiales	
"" (comillas)	Nos permiten utilizar términos que consistan en más de una palabra
.(el punto)	Permite hacer referencia al directorio donde estamos ubicados actualmente

Crear	
\$ mkdir [Carpeta]	Crea una nueva directorio o carpeta
\$ touch [Nombre del archivo]	Crea un nuevo archivo

Eliminar	
\$ rm [Nombre del archivo]	Elimina un archivo
\$ rmdir [Nombre de la carpeta]	Elimina una carpeta vacía
\$ rm -r [Nombre de la carpeta]	Elimina una carpeta y su contenido

<h2>Find</h2> <p>find: Busca archivos o carpetas, podemos hacerlo por nombre completo o que coincidan con ciertas letras o frases, archivos de ciertos usuarios, creados en cierta fecha y un montón de opciones más.</p>	<p>Cuando utilizamos find, podemos utilizar la opción -name para buscar por nombre o la opción -size si queremos buscar por tamaño de archivos.</p>	<p>Tanto en find como en grep, podemos utilizar la opción -i para que no se tenga en cuenta la sensibilidad a mayúsculas.</p>

<p>En find y en grep podemos utilizar .(punto), el cual permite buscar a partir del directorio donde estamos ubicados actualmente.</p>		


Grep

grep: Nos permite buscar texto en los archivos, puede ser el nombre de una funcionalidad que hemos diseñado, el de una variable que hayamos definido o cualquier tipo de texto.

Cuando utilizamos **grep**, la opción **-n** nos muestra la línea donde encuentra la coincidencia dentro de el/los archivo/s.

Cuando utilizamos **grep**, la opción **-r**, nos permitirá buscar en todas las subcarpetas recursivamente.

<p>Tanto en find como en grep, podemos utilizar ""(comillas) que nos permiten buscar un término que consista en más de una palabra.</p>		


GIT		
Introducción a Git	<div><div>Git init</div><div><p>Repositorio local</p><p>Repositorios local => vive en tu computadora remoto => se hospeda en GitHub</p></div></div>	
¿Qué es en sí Git?		
Software de control de versiones		
Mantiene eficientemente las actualizaciones sobre el código fuente		
Git init	crear repositorio local	
Git config user.name “mi-usuario”	agregar tu nombre usuario	
Git config user.name	Versificar si el nombre de usuario fue agregado correctamente.	
Git confiuser.email”miCorreo@email.com”	Saber correo electrónico con el que me registre.	
Git config user.email	Verificar si el correo fue agregado correctamente.	
Git add nombre del archivo	Agregar archivo por archivo	
Git add .	Agregar todos los archivos	
Git status	Seguimiento del estado de los archivos	
Git commit -m “mensaje”	Comitea los cambios hechos	
Git log	Registra historial de cambios de proyectos	
Ls -a	Muestra archivos ocultos	

Git checkout numero_commit	Devolverme al commit deseado	
Git restore .	Restaura las carpetas eliminadas	

Git hub

 <p>lugar en la nube donde se hospedan proyectos de programación</p>		
URL		
<code>https://github.com/mi-usuario/nombre-del-repo.git</code>		

Subiendo archivos a git hub

 <code>\$ git remote add origin https://github.com/mi-usuario/nombre-del-repo.git</code>	Para conectar el repositorio local con el repositorio remoto.	
<div>7</div> <div><code>\$ git push origin master</code></div> <div>(depende de las rama)</div>	Para subir los cambios al repositorio remoto.	

Bajando archivos de git hub

<code>\$ git clone https://github.com/usuario/repositorio.git</code> (la url del repositorio remoto)	Para clonar repositorios remotos	https://git-scm.com/docs documentación de git
Git temote -v	Para verificar que esté conectado los repositorios remoto y local	
<code>\$ git pull origin main</code>	Para actualizar los cambios realizados	

Ramas		
git branch	<p>git branch</p> <p>Enumera todas las ramas de tu repositorio, es similar a git branch --list.</p> <ul style="list-style-type: none">• git branch <branch> <p>Crea una nueva rama llamada <branch>.</p> <ul style="list-style-type: none">• git branch -d <branch> <p>Elimina la rama llamada <branch>. Git evita que eliminemos la rama si tiene cambios que aún no se han fusionado con la rama Main.</p> <ul style="list-style-type: none">• git branch -D <branch> <p>Fuerza la eliminación de la rama especificada, incluso si tiene cambios sin fusionar.</p>	
Git checkout	<p>Para moverse de una rama a otra, se ejecuta el comando</p> <ul style="list-style-type: none">• git checkout nombre_rama <p>Generalmente, Git solo permitirá que nos movamos a otra rama si no tenemos cambios. Si tenemos cambios, para cambiarnos de rama, debemos:</p> <ol style="list-style-type: none">1. Eliminarlos (deshaciendo los cambios).2. Confirmarlos (haciendo un git commit).	
Guardar cambios y subirlos al repositorio remoto	<p>Una vez que terminamos de realizar los cambios que queremos en nuestra branch, ejecutamos los mismos comandos que vimos hasta ahora: git add, git commit, git status y git log. Pero cuando queramos subir esos cambios, debemos utilizar git push con el nombre de la rama en que estamos posicionados:</p>	<p>git push origin <branch></p> <p>Así también, para traer los cambios de esa rama utilizamos el git pull agregando desde donde queremos traer los cambios:</p> <ul style="list-style-type: none">• git pull origin <branch>