

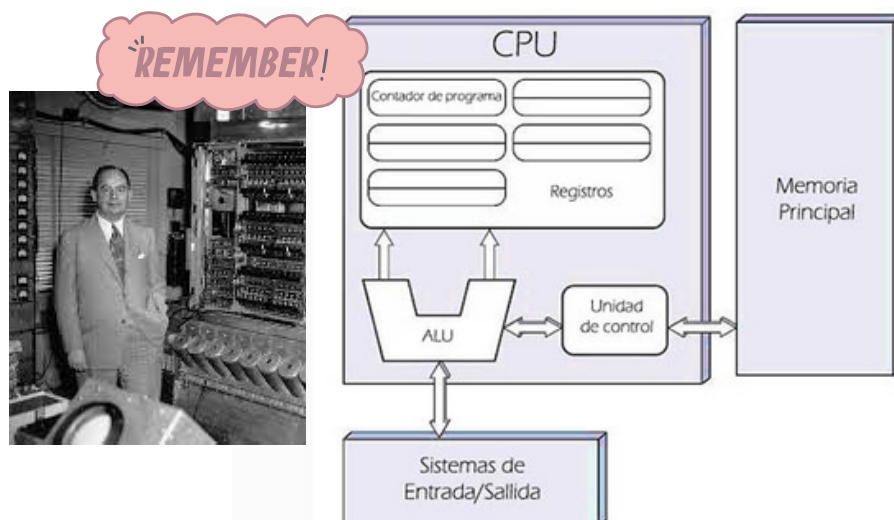
Introducción a la informática

Computadora

Es un dispositivo electrónico que recibe instrucciones y es capaz de ejecutarlas. Esto lo logra gracias a unos recursos físicos llamados Hardware y unas herramientas de interpretación y ejecución llamadas software.

❖❖❖ ¿Cómo funciona?

El funcionamiento de una pc está determinado por su arquitectura, la arquitectura Von Neumann definió un modelo básico necesario para que una pc sea una pc. Esto es posible gracias a que el procesador de la computadora (el cerebro de la pc) puede comunicarse con las distintas partes que componen el cuerpo de la computadora y así, estos funcionan como un solo componente.

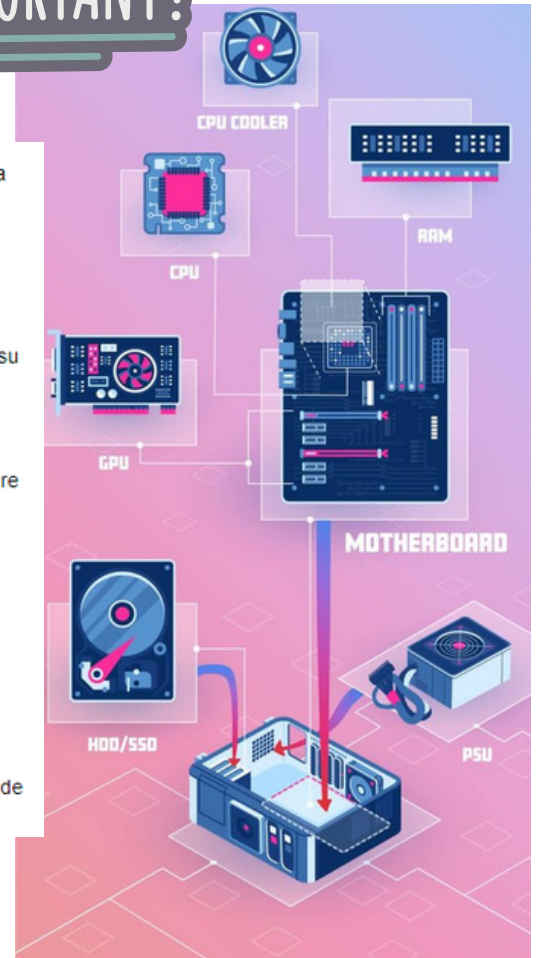


Hardware

IMPORTANT!

Componentes Internos

- ↓ **CPU** : la unidad central de procesamiento ejecuta instrucciones que componen un programa de computadora. Este es el núcleo de tu PC.
- ↓ **GPU** : la unidad de procesamiento gráfico procesa imágenes y es el componente más importante de una PC para juegos.
- ↓ **PLACA MADRE** : MOBO proporciona conectividad entre los componentes de hardware de su computadora.
- ↓ **PSU** : la unidad de fuente de alimentación es el componente responsable de alimentar su sistema. PLACA MADRE: MOBO proporciona conectividad entre los componentes de hardware de su computadora.
- ↓ **ALMACENAMIENTO** : su SSD / HDD es responsable de almacenar su sistema operativo, aplicaciones, juegos y archivos multimedia.
- ↓ **RAM** : la memoria de acceso aleatorio son datos temporales que la PC está utilizando activamente en el presente.
- ↓ **CARCASA DE PC** : una carcasa de computadora, también conocida como chasis, torre o gabinete de computadora, alberga toda su construcción y, al mismo tiempo, proporciona flujo de aire.



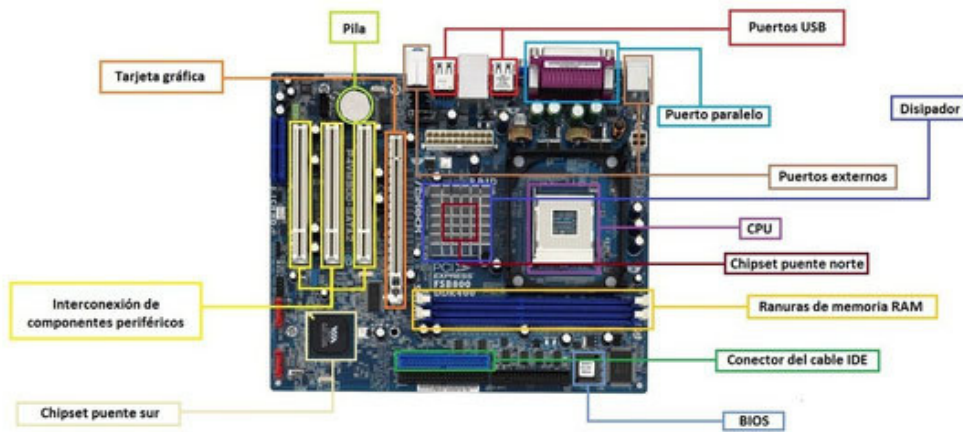
Periféricos

Son los dispositivos que conectamos a la pc para añadir funcionalidades. Estos pueden ser de Entrada, Salida o Mixtos.



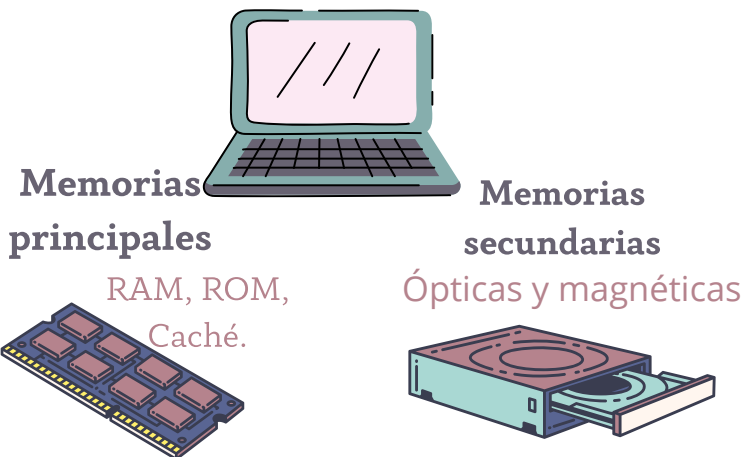
IMPORTANT!

COMPONENTES DE LA PLACA BASE



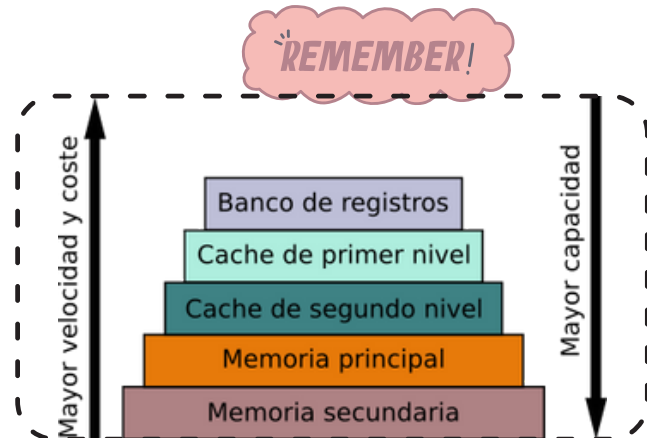
Memoria

Hay dos tipos de memorias: principales y secundarias. Las principales son las indispensables para el funcionamiento de la pc, mientras que las secundarias complementan el sistema.



Almacenamiento

SSD(disco Sólido) y HDD (disco mecánico)



Medidas






1 Bit = Binary Digit
 8 Bits = 1 Byte
 1024 Bytes = 1 KB (Kilo Byte)
 1024 KB = 1 MB (Mega Byte)
 1024 MB = 1 GB(Giga Byte)
 1024 GB = 1 TB(Terra Byte)
 1024 TB = 1 PB(Peta Byte)
 1024 PB = 1 EB(Exa Byte)
 1024 EB = 1 ZB(Zetta Byte)
 1024 ZB = 1 YB (Yotta Byte)
 1024 YB = 1 (Bronto Byte)
 1024 Brontobyte = 1 (Geop Byte)
 Geop Byte is The Highest Memory

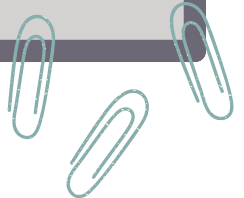
Caraterísticas

Las principales características de la memoria (RAM) son: La velocidad (MHZ), la capacidad(GB), la latencia(ciclos de reloj, o cada cuánto se mandan los datos), y el voltaje o energía que utiliza.



Glosario

- **Hardware:** partes físicas del computador
 - De **entrada:** medio para ingresar información 
 - De **salida:** medio para la salida de la información 
 - Interno:** componentes que son inseparables del dispositivo
- **Software:** soporte lógico que interactúa con el hardware gracias al sistema operativo.
 - Sistema:** brinda una interfaz para el usuario 
 - Programación:** brinda una interfaz para el desarrollador
 - App:** sistema para ejecutar tareas
- **Interfaces de usuario (UI):** espacio donde el humano interactúa con el computador.
 - Interfaz de línea de comandos (CLI):** terminal, interfaz solo de texto 
 - Interfaz gráfica de usuario (GUI):** interfaz gráfica para ejecutar comando o acciones
 - Interfaz nativa de usuario (NUI):** panel de control manejado a través del tacto o movimiento
- **Servidor:** computadora que atiende a las peticiones de un cliente y devuelve una respuesta
 - Web:** almacena el contenido colectivo y lo distribuye entre clientes de la red 
 - Base de Datos:** provee datos mediante el modelo cliente-servidor
 - Archivos:** almacena y distribuye archivos



Terminal

Es un contenedor de un Shell. Un shell es un programa que interpreta comandos gracias a una interfaz de línea de comandos. También llamado CLI.



Consola

Es un tipo de terminal. La escritura en la Consola se hace a través de la línea de comandos. Esta permite: mostrar o crear archivos, o datos.

Los IDE son editores de código donde podemos iniciar un terminal. También podemos usar el shell de nuestro sistema operativo



Git



Sistema de control de versiones. Este es un software que debemos instalar en nuestra computadora y posteriormente, debemos iniciarlo en nuestra carpeta o repositorio local.

Git Commands

\$ git init

Crea un repositorio local

\$ git remote add origin <url>

Crea un repositorio remoto

\$ git add .

Agrega un archivo a Staging

\$ git commit -m "Message"

Comitea el repo

\$ git push -u origin <branch>

Pushea un archivo al repo (origin)

\$ git pull

Actualiza el repo local

\$ git clone <url>

Clona un repo

\$ git log

Muestra los commit de una rama

\$ git reset <file>

Remueve archivos del staging

\$ git stash

Almacena los archivos trackeados modificados

\$ git branch

lista las ramas locales

\$ git checkout -b <branch>

Crea una rama y te mueve a ella

\$ git merge <branch>

Unifica la rama con main

\$ git checkout <branch>

Para moverse entre ramas

\$ git branch -d <branch>

Elimina una rama

\$ git branch -m <old> <new>

Cambia el nombre de la rama

\$ git diff

Muestra los cambios en staging

\$ git diff --staged

Muestra los cambios en staging

\$ git status

Lista los archivos comiteados

\$ git <command> --help

Muestra ayuda para un comando específico

REMEMBER!

es un repositorio
be, basado en Git





Git Bash



Comandos más usados:

```
MINGW64:/c/Users/39495
39495@LAPTOP-VRM155Q5 MINGW64 ~
$ ls Listar los archivos de un directorio o carpeta
$ mkdir (nombre) crear directorio o carpeta
$ touch (nombre.txt) crear un archivo
$ rmdir (nombre) eliminar un directorio o carpeta
$ rm (nombre.txt) eliminar un directorio o carpeta
$ mv (ruta/nombre) renombrar un archivo existente o mover un archivo a un
nuevo directorio o carpeta
$ cp (ruta/nombre) copiar una carpeta o archivo
$ cd (ruta) moverse entre directorios ("./" o "../")
$ clear limpiar la pantalla
$ find buscar un directorio o carpeta
$ grep buscar un archivo
```



Conectar un repo local a Github

- Ejecutar un git init en el repositorio

- Indicar el usuario:

git config user.name (usuario github)

git config user.mail (email)

IMPORTANT!

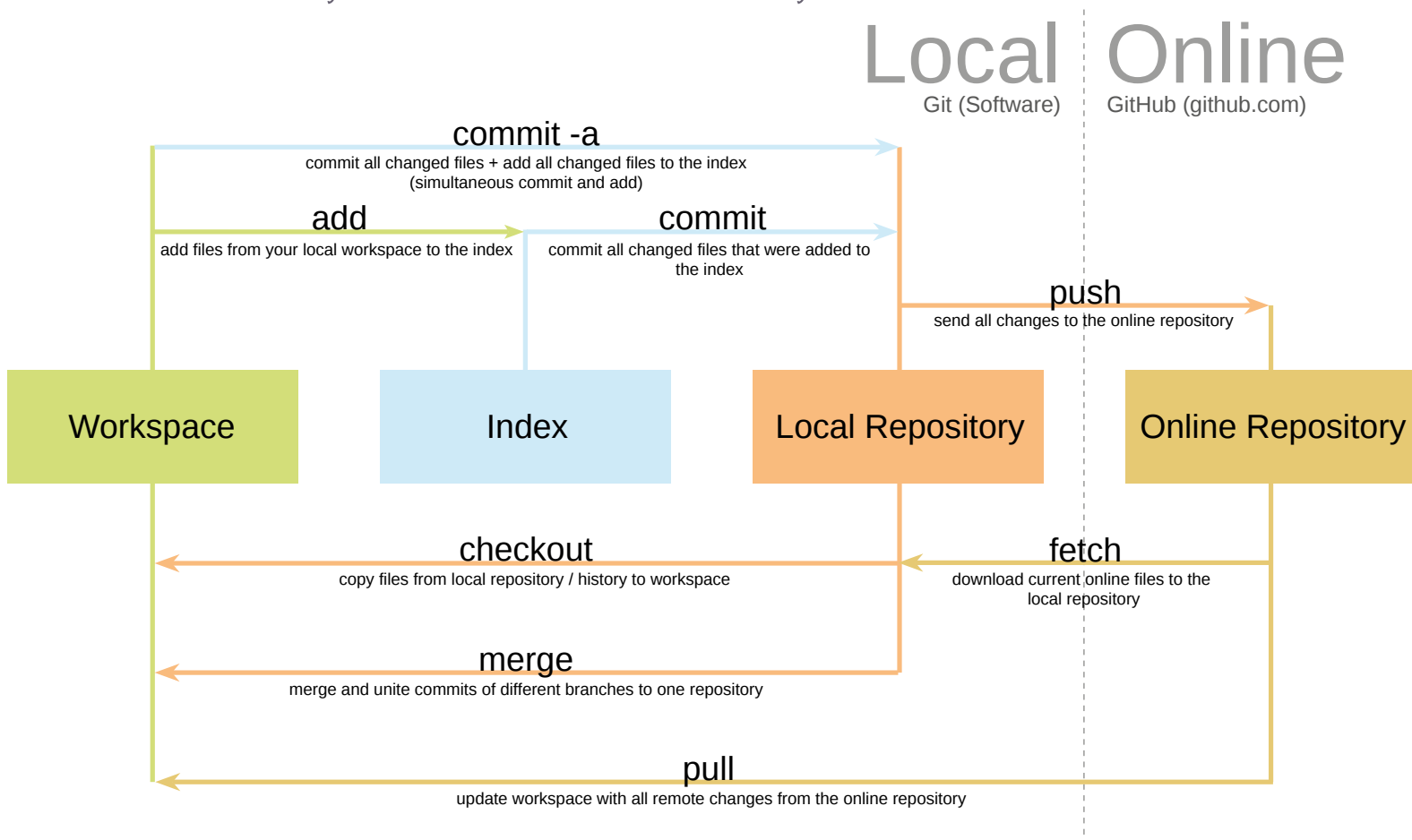
Estados de un archivo en Git

Un archivo entra en seguimiento cuando ejecutamos el comando `git add`. Al ser modificado, este cambia de estado y git nos indica que ha sido modificado.

Cuando ejecutamos `git commit`, creamos un punto en el historial de versiones de ese archivo. Luego podemos subir las modificaciones al repositorio en la nube con `git push` para conservar el estado o versión de ese archivo.

Si queremos que Git deje de hacer seguimiento de un archivo, ejecutamos el comando `git rm --cached`; y el archivo pasa al estado untracked.

Comandos y estados de un archivo en GIT y Github:





Ramas y conflictos

Un conflicto surge cuando dos personas modifican un mismo archivo. El que primero haya subido el archivo no tendrá problemas, sin embargo, el segundo recibirá un mensaje de error al realizar un GIT Push.

```
MINGW64/c/Users/lcruz/Desktop/git-demo
$ git config --global user.name "Your Name"

lcruz@enlace MINGW64 ~/Desktop/git-demo (pagina_principal)
$ clear

lcruz@enlace MINGW64 ~/Desktop/git-demo (pagina_principal)
$ git add .

lcruz@enlace MINGW64 ~/Desktop/git-demo (pagina_principal)
$ git commit -m "Esto dara un error"
[pagina_principal 53488cc] Esto dara un error
1 file changed, 1 insertion(+)

lcruz@enlace MINGW64 ~/Desktop/git-demo (pagina_principal)
$ git push origin pagina_principal
To https://github.com/lcruzsv/git-demo.git
! [rejected]        pagina_principal -> pagina_principal (fetch first)
error: failed to push some refs to 'https://github.com/lcruzsv/git-demo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Para solucionarlo, la segunda persona debe revisar el archivo y descubrir los cambios. Así, podrá definir junto con la primera si los cambios realizados son válidos

```
git-demo
├── .git
├── index.html
└── README.md

index.html
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Prueba de Git</title>
6    </head>
7    <<<<<< HEAD
8      <body class="error">
9        =====
10       <body class="body">
11         >>>>>> ae462fd6992e1f5bde3099aa50878ef00aeb79bf
12         Esta es una prueba
13       </body>
14     </html>
15
```

Una vez se decida cuál archivo es válido, se comitea el archivo final y se une al main con un Git Merge.