**RESUMEN INTRODUCCION A LA INFORMATICA**

**GIT**

Git es un conjunto de programas, de utilidades, de líneas y de comandos que están diseñados para ejecutarse en un entorno de líneas de comando de estilo Unix.

Es un software de control de versiones que registra los cambios realizados sobre un archivo o un conjunto de archivos a lo largo del tiempo. Permite revertir archivos y proyectos enteros a un estado anterior, comparar cambios a lo largo del tiempo.

Quién modificó por última vez algo que puede estar causando un problema.

**GIT BASH**

App de terminal Interfaz con SO Comandos escritos(CLI)

Cliente de git:terminal o bash usado para ejecutar comandos git.

para entornos de Microsoft Windows ofrece una capa de emulación para una experiencia de líneas de comandos de Git..

Bash es el acrónimo en inglés de Bourne Again Shell.

**SHELL**

Aplicación de terminal que se utiliza como interfaz con un sistema operativo mediante comandos escritos.

Bash es una shell predeterminada en Linux y macOS.

Git Bash es un paquete que instala Bash, algunas utilidades comunes de bash y Git en un sistema operativo Windows.

**TERMINAL:**

Programa en todo SO Órdenes al sistema mediante comandos Win + R

**CLI: DIFERENCIAS CON INTERFAZ GRÁFICA DE USUARIO (CUI)**

* Interacción con SI y software
* Conocimiento en comandos y experiencia
* Poca memoria para ejecutar
* Rápida reduce uso de recursos y dependencias
* Flexible, combinar comandos
* Uso en servidores, clusters, desarrollo y seguridad
* Mayor precisión

**CUI:**

* Interacción interfaz gráfica
* Usuarios principiantes, intuitiva
* Mas memoria por contenido gráfico
* Lenta por capa gráfica y navegabilidad
* Poco flexible, operaciones ya establecidas
* Público general, cotidiano
* Menor precisión

**ACCIONES, NAVEGACIÓN E INTERACCIONES CON FICHEROS**

Ejecución de orden abre 3 interfaces estandar:

Archivos hechos En el directorio /dev

**Entrada** (stdin)

archivo por el que una orden recibe su entrada (por defecto, es el teclado).

**Salida**(stdout)

archivo por el que una orden presenta sus resultados (ventana en la que se está ejecutando el intérprete de órdenes).

**Error** (stderr)

una orden presenta los mensajes que va generando cuando ocurre un error

**NAVEGACIÓN:**

Carpeta/ = directivo

Archivo no lleva /

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: mover entre diferentes directorios

cd Image espacios en blanco usar “ “ o \_

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

**FLAGS de list**

Ls -l muestra permisos, usuario q creó el archivo

Ls -lahmuestra permisos, usuario q creó el archivo, almacenamiento, fecha (mas fácil para leer)

Ls -a todos archivos incluso ocultos

Ls -R directorio de otras carpetas y archivos forma árbol

Crear o eliminar carpetas

mkdir / Crear una carpeta (make directory).

rmdir / Borrar una carpeta (remove directory)

Rm -r carpeta / “nombre” borra carpeta

No lleva a papelera, elimina directamente

Crear o eliminar archivos

gedit / Crear y editar archivos de texto.

rm / Borrar archivos.

Touch “nueva carpeta” /archivo.xls

Con un solo comando crea en esa ruta relativa, “\_

Sintaxis:

Argumentos:

Ej.,: directorio a recorrer CD Documentos

Opciones:

Ls -t lista c/archivo último modificados

**Direcciones Relativas:** Ubica la ruta, sin especificar, ahorra tiempo, mayores errores

**Direcciones Absolutas**: ruta completa de la ubicación de la carpeta o archivo en el sistema

/ al comienzo de la ruta

**Tecla tab** autompleta nombre carpeta, archivo

Cd .. volves al directorio desde la carpeta donde estas

Cd .. /carpeta/ subi un nivel, vas a la carpeta q llamaste en este caso carpeta

Posible crear 2 archivos en el mismo comando touch archivo3 archivo.txt

**REPOSITORIO LOCAL GIT**

**Repositorio:** almacén de archivos

**Repositorio local:** tiene todos los archivos (que hayas guardado en él) en nuestra computadora.

**Commits:** paquetes de archivos (timestap -fecha de creación, estado- y autor)

Historial de cambios de proyectos

Confirma agregado de archivos

**REPOSITORIO REMOTO**

viven en la nube, es decir, en GitHub.

1 proyecto = 1 repositorio

**GIT HUB**

plataforma colaborativa que nos va a permitir llevar un control de versión

sobre nuestro código.

**GIT PUSH** antes de **git pull** para subir un repositorio en la nube.

se descarga actualizaciones que tenga el repositorio remoto de GitHub antes de subir mi modificación.

**PEGAR TOKEN** DESPUES DE // @ (dirección de repositorio) github.com/NaNahirr…

**nano/config**  posibilidad de ver token, usuario y modificarlo

diferentes repo tienen su propio token con sus permisos

**CONECTAR LOS REPOSITORIOS:**

Iniciar repositorio, Git init inicia proyecto en carpeta que se conecte con el repositorio.

inidicar al repositorio el usuario, mediante:

1. Git config user.name “mi-usuario”(desde carpeta proyecto)Poner usuario creado en GitHub

2. Git config user.email “micorreo@”

Verificar comandos sin lo ultimo

* Git config --global user.name o email determinar cambios todos hechos por mi siempre.
* git push: envía archivos del repositorio local al remoto
* git push origin main envía archivos del repositorio local al remoto, en la rama main

**AGREGAR ARCHIVOS**

* Commits: seguimiento y control sobre versiones de los archivos
* Git add (nombre arch): suma un solo archivo, correr el comando git add archivo.extension
* Git add . agrega todos los archivos de la carpeta
* Git status: estado de carga del archivo

**ARCHIVOS MODIFICADOS:**

Toma como nuevos sin seguimiento

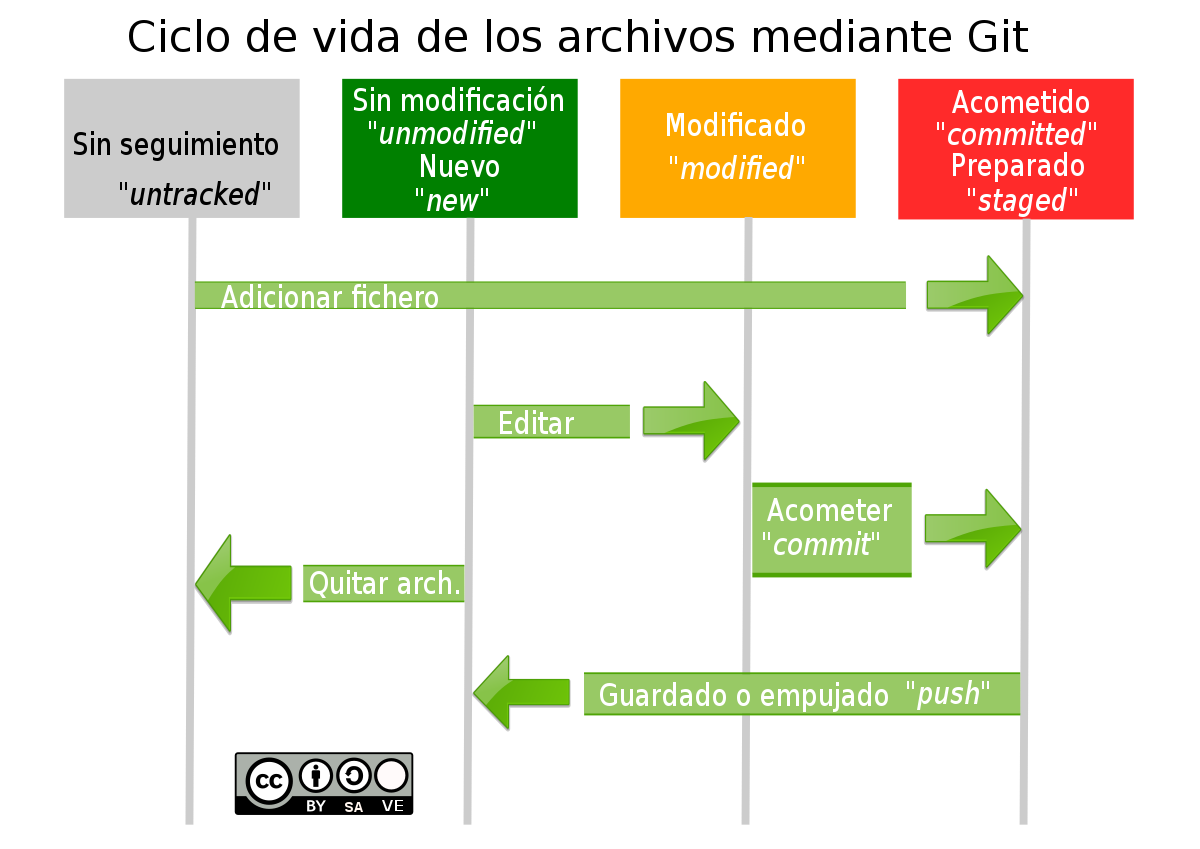
Ejecutar git add o git add . para correrlos

**Ejecutar Commit o commitear:**

1. Agregar los archivos modificados al repo

2. Tipear git commit -m”primer commit del proyecto”(anotación del status)

Linea de tiempo.



**ciclo de vida de archivo:**

1. Creación del archivo, index.html (está en estado **untracked**).
2. Cambiamos el estado del archivo a **seguimiento** a través del comando *git add* index.html (especificando nombre del archivo index.html para este caso).
3. Si el archivo es modificado, deja de estar en seguimiento y pasa a un estado de **modificado,** ya que Git detecta que hubo cambios.
4. Guardamos el archivo con ctrl+g (o ctrl + s) y luego escribimos *git add* index.html (especificando nombre del archivo index.html para este caso), el archivo pasa al **stage area**.
5. El archivo está listo para realizar el *commit*, que es cuando se genera el punto histórico, a través del comando git commit.

Para especificar que git **no realice más seguimiento** a un archivo, usamos el comando **git *rm --cached*** donde si bien el archivo como tal no se ha eliminado, Git procede a ignorarlo y pasa a un estado de untracked.

CREACION DE UN REPOSITORIO EN GITHUB

1. new repo, poner nombre ej: BranchRepo.
2. crear readme.md.
3. hacer git clone de eso.
4. desde escritorio apretar git bash.
5. git clone con el https con el token.
6. despues del token poner @github.
7. y se clona el repo con el archivo readme adentro.
8. git –config.
9. crear dir, dir2
10. touch repoBranch/dr1/estilo.css
11. git add .
12. git commit -m”agregando dip1”
13. git config user name
14. git config user email
15. git push origin name
16. en repo de la web se tiene q ver la carperta creada.
17. para crear branch--.
18. git branch nuevaRama.
19. para subir git push -u nuevaRama.
20. si uno git branch solo veo la cantidad de rama q tiene el repo.
21. para ver la que tengo se ve en verde.
22. para pasar a una rama git checkout nuevaRama.
23. en visual studio code en terminal es más visible las ramas.
24. para cambiar entre ellas git switch nuevaRama.
25. cambia a esa rama
26. si no se commitea el archivo de una rama, no queda ubicado en ninguna de las 2.
27. eso se ve con git checkout en ambas ramas, nuevaRama o main.
28. cuando se commitea queda en una sola rama.
29. git touch ejemplo.
30. git checkout nuevaRAMA.
31. git commit -m “agregando nueva rama”.
32. git checkout main, el archivo agregado en nuevaRama no se ve en main.
33. con merch se unen las ramas locales con sus ejemplos en remoto en GtiHub.
34. solo lo hace la persona autorizada.
35. con git log veo los commtis hechos en cada rama, su historial.
36. traje lo que estaba en nuevaRama a main, se fusiona una con la otra, se copia el contenido de una en otra.
37. cuando se hacen nuevas funcionalidades a un proyecto se abre esta nueva rama branch.
38. conflictos cuando haces 2 cambios en 2 ramas, deja definición a programador, se aceptan ambos cambios.
39. con git push origin main,
40. diferencia entre git pull y merge, pull traemos cambios de repo remoto y se hace antes de subir el archivo modificado, con push subimos el archivo local
41. merge, check out, etc son propios del repo local.
42. **git check out** recupera archivo borrado, **git swift** cambia entre ramas
43. comando **git push -u origin** sube la nueva rama a git hub, se puede mover entre ambas
44. comando **git branch -a**  muestra todas las ramas, todo lo que tenga origin está en el repo remoto.

proyecto concatenación de un proyecto

es una rama--2 branch-2 concatenación de commits en el tiempo

el nombre de la rama principal de todo proyecto es main

siempre se crea en primera instancia

un commit arma estación en el tiempo, viaje del pasado al futuro

último commit es el head “del presente”

para traer algo del pasado al presente

debemos traer un commit del pasado mediante los hash (identificación única de los commit conjunto de números)

se puede traer con los reset al presente

se puede modificar el head trayendo un anterior commit.

**Main:**

* rama donde funciona el programa, necesita varias validaciones.
* se toca menos.
* se hace una copia llamada **dev**, y sobre esa se trabaja.
* para generar cambios debo crear una **nueva rama con git branch.**
* asi traigo todo su historial.
* se copian todos los commits del repositorio.
* para juntar las actualizaciones con la rama principal usar comando **merch.**
* asi se unen las 2 cosas, lo que no esté en una rama, está en la otra.

**RAMA MASTER A MAIN**

desde el 1 de octubre de 2020, cambió rama master a nombre main

podemos configurarlo por defecto con este comando

**config --global init.defaultBranch main**

**RESOLVER CONFLICTOS**

aparece con el error:

*error: failed to push some refers to y la url del repo*

alguien hizo cambios en el repo antes que vos, está desactualizado el repo remoto

git pull origin main o nombre rama descarga contenido que falta actualizar.

*CONFLICT CONTENT Merge conflict in …*

cambios se superponen en la misma línea de código.

**3 Soluciones posibles:**

* Dejamos nuestro código, versión local, el código que viene de GitHub o Unimos ambos.
* hay que borrar las tres líneas de texto que nos agrega Git para identificar la zona del conflicto.
* Si no hay conflictos entre lo que teníamos y lo que trajimos, Git va a fusionar todos
* los cambios de forma automática.
* "líneas divisorias de conflictos". La línea ======= es el "centro" del conflicto. Todo el contenido entre el centro y la línea <<<<<<< HEAD es contenido que existe en la rama principal actual a la que apunta la referencia HEAD. Por el contrario, todo el contenido entre el centro y >>>>>>> new\_branch\_to\_merge\_later es contenido que está presente en nuestra rama de fusión.
* La forma más directa de resolver un conflicto de fusión es editar el archivo conflictivo usar git add para preparar el nuevo contenido fusionado y commit.

**git diff** yuda a encontrar diferencias entre los estados de un repositorio/unos archivos. Esto es útil para predecir y evitar conflictos de fusión.

**git checkout** deshacer cambios en los archivos o para cambiar ramas.

**git reset --mixed** deshacer cambios en el directorio de trabajo y el entorno de ensayo.

**CREAR RAMA**

1. **git branch SOLO** enumera cambiar el nombre y eliminar ramas. No permite cambiar entre ramas o volver a unir un historial bifurcado.
2. **git branch <namerama>** crea rama “namerama”
3. **git branch -d <namerama>** elimina esa rama si tiene cambios aun no fusionado con main
4. **git branch -D <namerama>**fuerza eliminación aunque tenga cambios sin fusionar
5. **git checkout** <name> mover si no hay cambios hechos
6. **git push origin<namerama>**
7. **git pull <namerama>**trae cambios de esa rama

**MÓDULO 2,**

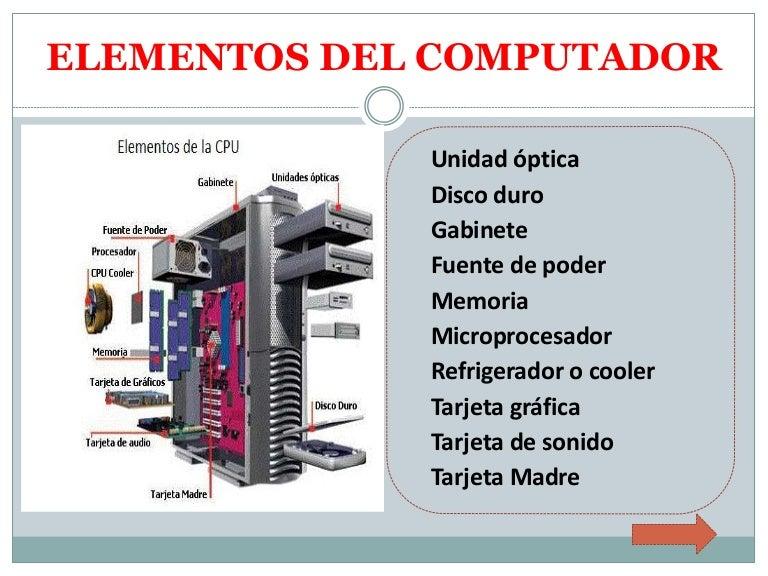
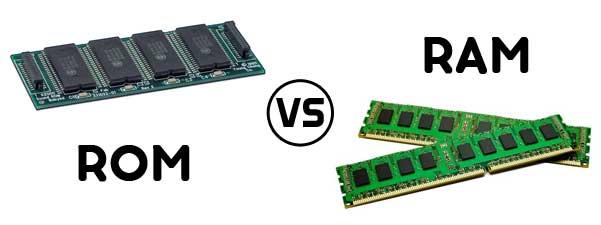
**COMPONENTES DE UNA COMPUTADORA**

**HARDWARE**

parte “dura” de un sistema informático, los componentes electrónicos en sí, como placas de video, memorias, teclados, todo componente tangible que utilizamos para interactuar con nuestro ordenador.

**SOFTWARE**

instrucciones necesarias para realizar una determinada función, como, por ejemplo, los sistemas operativos, los juegos, programas de usuario y demás

****

1. **placa madre:** placa principal todos los demás dispositivos se conectan, de manera directa (como los circuitos eléctricos interconectados), como indirecta (a través de puertos USB u otro
2. **procesador:** CPU, cerebro, “dirige” interpreta y ejecuta instrucciones/operaciones mediante operaciones aritméticas y lógicas. está en la placa madre, tiene disipador de calor
3. **RAM Y ROM:**
4. **Placa video y sonido**:1ra o tarjeta gráfica muestra imágenes refiere al GPU, la 2dar reproduce sonidos y recibe
5. **dispositivos almacenamiento:** secundario:no necesita la PC para su funcionamiento permanente disco rígido o duro.
6. **fuente de alimentación**: transforma energía, la envía a placa madre y resto
7. **- componentes externos:** no imprescindibles para funcionamiento de la PC
8. **periféricos:** añaden funciones y operaciones, pero no esenciales.
   1. **entrada:** ingreso de datos (micro, teclado, cámara, táctil, ratón, lectora DVD.)
   2. **salida:** extrae monitor, parlante, proyector, impresora 3D
   3. **mixtos:** ambas (multifunción, pantalla táctil, modem, tarjeta red, lectograbadora).

**ARQUITECTURA DE COMUNICACIÓN ENTRE COMPONENTES**

****

**MODELO VON NEUMANN**

Primeras pcs, cálculos específicos

después de la PGM surge la necesidad de crear maquina con propósito general 100% electrónica.

Modelo conceptual, forma lógica de cómo funciona una pc; como la pc tiene que esta armada de forma correcta gestiona bien recursos principales y secundarios.

divide el funcionamiento del pc en 3 bloques (CPU, memoria principal y salida con la memoria secundaria)

**CPU** (central processing unit)

unidad central de procesamiento es el hardware dentro de un ordenador u otros dispositivos programables, su trabajo es interpretar las instrucciones de un programa informático.

cerebro, ejecuta procesos y toma decisiones.

ejecuta instrucciones almacenadas como números binarios organizados en la memoria principal, por lo que, cuanto más potente sea tu procesador, más rápido podrá hacer las operaciones y más rápido funcionará tu dispositivo en general.

Es encargado de leer, interpretar y procesar las instrucciones primero del Sistema Operativo, y después de los programas o aplicaciones que tenés instalados en el ordenador.

Lee datos e instrucciones de la memoria RAM, requiere información desde perifericos de entrada y se comunica con periféricos de salida para mostrar los resultados.

A través de su bus frontal se conecta con el puente norte, que conecta parte principal con el resto de la pc.

para ejecutar un proceso, la info tiene que estar en la memoria principal o RAM, cargada ahí

CPU manda instrucción de la info que necesito ejecutar dentro de la RAM

si la encuentra dentro de la RAM y el puente norte, se ejecuta

si no está la info en memoria principal, o la RAM se busca en memoria secundaria.

envía la instrucción mediante bus interno llega al puente sur (partes menos importantes o secundarias de la pc).

en memoria secundaria, busca información si encuentra la info, la envía a memoria principal y de ahí al CPU para ser ejecutado.

tiene generalmente 2 núcleos: ejecuta más tareas al mismo tiempo.

cada núcleo ejecuta un proceso a la vez

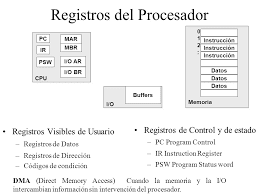
procesos: programas que se están ejecutando.

si tenes un solo núcleo tomas una parte y la ejecutas, avanzas al otro una vez q se haya ejecutado el anterior.

**componentes:**

**ALU** unidad aritmética lógica. +, -\* y /, AND OR o NOT.

**registros de cpu:** ayuda para automatizar tareas, tipo de memoria rápidas, dentro del procesador, usa unidad aritmetico-logica.

Memoria de muy alta velocidad, que se utiliza en los procesadores para acceder a información importante de manera rápida. 

La CPU tiene 5 registros internos.PC: Program counter

IR: Instructions register MAR: Memory address register, MDR: Memory data register y Accumulator.

**contador programa**: donde están los programas, cómo se almacenan en memoria con índice.

**unidad de control:** controla funcionamiento ALU, sistema de entrada y salida, ayuda a memoria secundaria.

Mueve datos y programas desde y hacia la memoria, ejecuta instrucciones del procesador.

**registro:** area de almacenamiento de alta velocidad, datos almacenarse para procesarse

**registro de direcciones:** ubicación en la memoria de los datos a acceder.

**registro de datos:** de memoria, datos q van a la memoria

**contador programa o program counter** calcula ciclos de ejecución.

**caché de la CPU:**

se divide en un total de tres niveles generales al que podemos sumar un cuarto que no resulta nada común.

La diferenciación entre memoria caché L1, L2 y L3 (también L4) obedecen a un orden de jerarquía establecido por cercanía al procesador, velocidad y capacidad

**ciclo de la máquina:**

buscar, decodificar, ejecutar y almacenar, hace program counter.

instrucciones obtenidas de la memoria, CPU decodifica y las ejecuta. resultado almacenado en la memoria.

**memoria principal:**  acceso mecanismos de entrada y salida

**características:**

**-zócalo**

fija y conecta el procesador. hace comunicaciones con el exterior

según numero de conexiones, mas elementos integrados. Mayor ancho de banda a otros elementos. celu y laptop no los usan.

**-núcleo/core:**

subprocesador en si mismo.

sigle-core:no realiza mas de una tarea al mismo tiempo.

dual-core, quad-core; mas divisiones mas tareas al mismo tiempo.

**-hilo/threads de cpu** flujo de control de programa. ayuda a cómo administrar una tarea.

“tiempos de espera” aprovechen mejor.

divide la tarea en porciones, alternando generando la sensación que se hacen al mismo tiempo.

con procesadores multihilos: ejecuta porciones de programa y las ejecuta en paralelo; haciéndolas más rápido.

**memoria caché:**

mas rápida, acelera el acceso a memoria del procesador, guarda datos e instrucciones más usados.

**frecuencia:**

reloj que sincroniza componentes y limita operaciones que realiza en determinado tiempo.

números de cambios que se hacen en un determinado segundo.

IPC, instrucciones por ciclo mide instrucciones que realiza el procesador en cada ciclo de su reloj interno

Mientras que una CPU con una velocidad de reloj más rápida puede completar más ciclos en un segundo, una CPU con un IPC más alto, pero con una frecuencia más baja, puede ser capaz de completar más tareas en un segundo.

**características procesadores**

* latencia: el tiempo que se demora el procesador en ejecutar sola una acción.
* frecuencia o velocidad: cuanto tarda para llevar adelante una determinada cantidad de tareas, varias acciones o ciclo completo medida en GHz.
* mayor frecuencia más acciones que puede hacer.
* voltaje: tensión que necesita el procesador para funcionar.
* latencia de los procesadores.
* capacidad: cuanto debo tener para levantar la tarea.
* overclock: subís una frecuencia base a un nivel tolerable a nivel más rápido, funciona, pero acorta la vida útil del procesador, determinado por la BIOS si te deja levantar el video.

**GPU:**

mismas acciones que la cpu pero solo acciones gráficas, si es algo gráfico lo hace la gpu

tecnología corse(mas nucleos cpu ejecuta muchos procesos en paralelo).

tarjeta grafica de pc.

unidad de procesamiento gráfico es un coprocesador dedicado al procesamiento de gráficos para aligerar la carga de trabajo del procesador central en aplicaciones, como los videojuegos o aplicaciones 3D interactivas.

**Puente Norte**

Chip que controla las funciones de acceso desde y hasta el CPU, PCI-Express, memoria RAM, vídeo integrado (dependiendo de la placa) y el puente sur.

**Puente Sur**

Chip que se encarga de coordinar los diferentes dispositivos de entrada y salida y algunas otras funcionalidades de baja velocidad. No está conectado a la CPU y se comunica con ella indirectamente a través del puente norte.

**USB**

El bus universal en serie es utilizado como estándar para conexión de periféricos. Se puede conectar con el teclado, el mouse, la memoria USB, el escáner, celular, la impresora, el módem, la grabadora de DVD externa, el disco duro externo, entre otros.

**Memoria Secundaria:**

Es un tipo de almacenamiento masivo y permanente con mayor capacidad para almacenar datos e información que la memoria primaria (RAM) que es volátil, aunque la memoria secundaria es de menor velocidad.

**MEMORIA PRINCIPAL:**

funcionamiento de la computadora, sin ella no prende

registros, ram, caché y rom.

**ROM**

read only memory o memoria de solo lectura. solo puede ser leída, no escrita. Guarda las instrucciones necesarias para que la computadora pueda iniciarse.

chip con configuraciones de la pc para funcionar

se cambia a través de la BIOS, que tiene una pila 2032 esa mantiene la config de la rom o cpu cuando no hay energia.

se puede actualizar el rom y la bios, tarea sensible, si se modifica mal la ROM, la pc no va a funcionar mas.

BIOS: software con que se modifica la ROM,

**CACHE**

la CPU y la memoria RAM. La CPU copia los datos más relevantes que va a utilizar de la memoria RAM para acceder a ellos más rápidamente.

**RAM**

Carga todas las instrucciones que ejecuta la CPU y otras unidades del computador, además de contener los datos que manipulan los distintos programas. Accede a las instrucciones y guarda los resultados inmediatos. Es volátil.

velocidad de la pc.

almacena programas que se van a ejecutar en el procesador.

a medida que se ejecutan los procesos se mueren y comienzan otros alojados en la memoria secundaria.

**SLOT**

Conecta ram con CPU muchos pines, placa madre puede tener mas de uno

**Acceso de la CPU a la RAM**

* dual channel: acceso simultáneo a dos módulos de memoria.
* todos los módulos de memoria deben tener la misma capacidad, velocidad, frecuencia, latencia y fabricante.
* si pongo una RAM que tenga menos velocidad que la otra, van a trabajar las 2 al mismo nivel
* Las velocidades se suman > Si la velocidad de cada módulo es de 1600 Mhz, la velocidad total será de 3200 Mhz.
* La capacidad se suma > Si cada módulo tiene una capacidad de 8 GB, la capacidad total será de 16 GB.
* single channel: utiliza una única señal a un ancho de banda y frecuencia determinada.

**Características:**

**Velocidad:** tiempo que tarda la RAM en recibir una solicitud del procesador y acceder a la información. La velocidad de una memoria está medida en megahertz (Mhz) o millones de ciclos por segundo.

**Capacidad:** cantidad de datos que se pueden almacenar en una RAM. La capacidad se mide en gigabytes (GB).

**Latencia:** ciclos de reloj que transcurren entre una petición y su respuesta. Mide en nanosegundos en CPU, RAM Y CACHÉ. milisegundos en disco rígido, SSD e internet.

**Voltaje:** energía consumida por el módulo de RAM.

**TIPOS:**

1. **V RAM** optimizada para adaptadores de vídeo. Tienen dos puertos para que los datos de vídeo puedan escribirse al mismo tiempo.
2. **DDR RAM** 2000, usada después del 2002 Operaba a 2.5V y 2.6V y su densidad máxima era de 128 Mb (por lo que no había módulos con más de 1 GB) con una velocidad de 266 MT/s (millones de trasnferencia por segundo) (100-200 MHz).
3. **DDR2 RAM:** 2004 voltaje de 1.8 voltios, un 28% menos que DDR. Se dobló su densidad máxima hasta los 256 Mb (2 GB por módulo). Lógicamente la velocidad máxima también se multiplicó, llegando a 533 MHz.
4. **DDR3 RAM** 2007 Implementaron los perfiles XMP módulos de memoria operaban a 1.5V y 1.65V, con velocidades base de 1066 MHz pero que llegaron mucho más allá, y la densidad llegó hasta a 8 GB por módulo.
5. **DDR4 RAM** 2014 Se reduce el voltaje hasta 1.05 y 1.2V, aunque muchos módulos operan a 1.35V. La velocidad se ha visto notablemente incrementada, pero su base comenzó en los 2133 MHz, hay módulos de 32 GB
6. **DDR5 RAM** 2020 nchos de banda de hasta 6.4 Gbps en sus modelos iniciales, es la primera memoria DDR de doble canal en un solo chip. Su frecuencia base es de 4800 MHz y, además, su consumo baja por la clásica reducción de voltaje, esta vez a 1.1 V. Su capacidad de almacenamiento máxima en un módulo de memoria es de 128 GB.

**UNIDADES DE MEDIDA**

1. BIT= 1 o 0
2. BYTE: 8 bits en fila
3. KBYTE: 1024 BYTE en valor binario
4. MEGABYTE 1024 KB
5. GIGABYTE 1024 MB
6. 1 terabyte: 102 4GB
7. 1 petabyte: 1024 TB
8. 1 exabyte: 1024 PB
9. 1 zettabyte: 1024 EB
10. 1 yotta: 1024 ZB

**SISTEMA NUMÉRICO:**

símbolos y reglas de generación que construyen todos los números válidos del sistema

**no posicional:** valor de los símbolos fijo, no depende de la posición

**posicional:** depende del valor y lugar asignado del símbolo

**digito:** símbolos diferentes que constituyen sistema de numeración

**base**: cantidad de dígitos. Ej: 3434 = 3000+ 400+ 30+4 10 símbolos del 0 al 9

**FUNCIONAMIENTO DE PC**

* primero cargado en la memoria principal el proceso.
* pasa a memoria caché -3 niveles segun el procesador, l1, l2, l3- procesador de gama alta.
* programa entra a la caché l3, se disminuye el programa progresivamente a L1
* en L1 entra en procesador, que ejecuta la acciones en el registro-procesador más chiquito-.
* l3, l2, l1 discriminación o filtra lo q se va a procesar en el momento
* generalmente una caché por núcleo, por eso l1 es más chiquita, son 4
* l2 es por sector, va de 2 en 2 núcleo de los fijos.
* l3, caché de todo el procesador
* l4 misma tarea pero en procesadores de masd de 4 nucleos
* 2 nucleos fisicos = 4 procesadores lógicos
* 2 procesadores suelen ser un núcleo
* si ejecuta el procceso, sale y muere, sino en fila de espera y vuelve a ingresa en su tiempo

**COSTO DE VELOCIDAD POR BIT**

si es más rapida la memoria, tiene menos almacenamiento.

mas almacenamiento, mas lento.

mientras menos almacenamiento mas rápidas.

**Secundaria:**

dispositivos capaces de almacenar datos. Pueden ser internos, como el disco duro, o extraíbles, como los discos flexibles y DVDs.

tipo de almacenamiento masivo y permanente (no volátil) con mayor capacidad para almacenar datos e información.

hace al usuario, si no tengo el disco de memoria secundaria, anda la pc.

TIPOS:

1. MAGNÉTICO:
   1. cinta magnetica: graba en pistas sobre una banda plástica con un material magnetizado, generalmente óxido de hierro o algún cromato. audio,video datos
   2. diskette: transporta información de una PC a otra, su capacidad podía llegar hasta 2,88 Mb . Los más utilizados eran los de 3 ½
   3. disco duro: por uno o varios platos rígidos introducidos en una caja hermética y unidos por un eje común que gira a gran velocidad. 2 caras lectura/escritura
2. óptico:
   1. cd sean imágenes, vídeos, audio, documentos hasta 80 minutos de audio= 700 MB de datos.
   2. dvd. disco digital versátil desde 1,46 GB en un DVD de un solo lado y una capa a 17,08 GB en un DVD de dos lados y dos capas. Las variaciones de DVD también consisten en DVD-R, DVD+R, DVD-RW, DVD+RW y DVD-Ram que describen la manera en la que el contenido de medios se almacena en el disco.
   3. blu-ray mayor que el resto, una sola capa almacena unos 25 GB de información.puede admitir varias, multiplicando su capacidad.Hay de hasta 100 GB
3. sólido:
   1. flash: Mem SD tarjeta, letura/estrictura muchas posiciones por impulsos eléctricos
   2. pendrive: tiene memoria flash,varia según modelo hasta 256 GB
   3. SSD, estado sólido utilizan el bus SATA o el PCIe del ordenador (discos ssd M2), siendo los últimos más rápidos que los primeros dado que un SSD normal encuentra un cuello de botella en el bus SATA ya que un SSD ofrece velocidades superiores a las que ofrece el bus SATA 3.

**SISTEMAS OPERATIVOS**

**Caracteristicas y funciones:**

* suite de aplicaciones.
* kernel núcleo del SO.
* kernel cumple acciones criticas .
* kernel como un superusuario o root.
* según su funcionamiento y robustez se define el SO.
* software que maneja hardware.
* soporte lógico que controla el funcionamiento del equipo físico.
* programas que permite manejar la memoria, disco, medios de almacenamiento de información y los diferentes periféricos o recursos de nuestra computadora.
* Gestiona la memoria de acceso aleatorio y ejecutar aplicaciones designando recursos necesarios.
* administra la CPU .
* direcciona entradas y salidas de datos mediante drivers (y los periféricos de entrada y salida).
* dirige autorizaciones de uso para usuario.
* administra los archivos.
* administra información para buen funcionamiento de la pc.
* dirige autorizaciones de uso para usuario.
* Desde el usuario: programas y funciones que ocultan los detalles del hardware, ofreciendo al usuario una vía sencilla y flexible de acceso al mismo.
* administra los recursos ofrecidos por el hardware y actúa como un intermediario entre la computadora y su usuario.

Servidores:

* Con SO derivados de UNIX, red hat o Windows Server.
* multiusuarios al mismo tiempo trabajando sobre el mismo núcleo.
* varían según hardware y función de cada SO.

**TIPOS:**

*Administración de usuario:*

Multiusuario

varios usuarios ejecuten simultáneamente sus programas

por medio de terminales conectadas a la computadora o por sesiones remotas en una red de comunicaciones

UNIX, LINUX O SOLARIS, ANDROID(funcionalidad desactivada)

Monousuario

ejecuta los programas de un usuario a la vez.

no importa cuántos procesadores tenga la computadora o cuántas tareas realice el usuario, solo podrá dar servicio a uno

Windows 1.0

*Administración o gestión de tareas:*

Multitarea:

varios procesos al mismo tiempo, la mayoría multitarea y multiusuario

Monotarea:

un proceso a la vez sin interrumpir, sistemas operativos más primitivos, como DOS

*gestión de recursos.*

Centralizado:

permite utilizar los recursos de un solo ordenador.

Distribuido:

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

*según licencia.*

Proprietary software:

limitaciones del propietario-Windows).

Open source:

a voluntad de usuario -Ubuntu o Red Hat).

**Estructura interna:**

***Monolítica:***

un solo programa con rutinas entrelazadas entre sí, se comunican entre ellas

rápidos pero no tienen flexibilidad para soportar diferentes tipos de aplicaciones

***Jerárquica:***

subdivide en capas o anillos definidas y con una clara interfaz respecto al resto de los recursos.

***Máquina virtual:***

integran diferentes SO como si fueran maquinas diferentes

**Recursos que administra:**

1. RAM y apps
2. CPU, por los algoritmos de progración
3. entradas y salidas de datos, a través de drivers y periféricos
4. información para buen funcionamiento PC
5. autorizaciones de uso usuario
6. archivos

**ARQUITECTURA CLIENTE – SERVIDOR**

Características:

* procesa información de modo distribuido
* en distintos lugares y accede a recursos compartidos.
* acceso transparente, multiplataforma y multi arquitectura.
* Independencia.
* protocolos asimétricos, el servidor se limita a escuchar en espera de que un cliente inicie solicitud.
* facilita escalabilidad, fácil añadir nuevos clientes a la infraestructura (escala horizontal) o aumentar potencia del o los servidores; aumentando número o capacidad de cálculo-escala vertical-.

*cliente:*

PC con prestaciones ajustadas, término front-end es un proceso que solicita servicios del servidor mediante una petición del usuario.

este proceso interactúa con el cliente mediante GUI

envía solicitud a servidor.

*servidor:*

ordenador con prestaciones elevadas

proceso que tiene recursos y servicios que el cliente le solicita, back end.

según el tipo que sea, será el tipo de arquitectura

programas y datos centralizados, facilita integridad y mantenimiento.

oye solicitud (request) la verifica y es correcta la procesa

cuando tenga el resultado(response) solicitado, envía respuesta.

*datacenter:*

inmuebles para albergar sistemas de información o comunicación protegidos de incendios, accesos indebidos e interrupción de energía

utiliza sistemas de red, como Windows o Linux

*middleware:*

* parte del software que se encarga del transporte de los mensajes
* traduce mensaje enviado por cliente
* ejecutado en ambos lados
* posee DBMS, permite crear, actualizar datos
* API interfaz de aplicaciones
* independiza a los clientes y los servidores
* El cliente recibe el resultado que solicitó.
* Realiza las comprobaciones oportunas —en caso de ser necesarias— y, si ese es el objetivo final, se lo muestra al usuario.
* más control sobre el negocio, permite obtener información desde diferentes orígenes y ofrecer de manera conjunta.
* sistemas débilmente acoplados, interacción mediante envío de mensajes.

**Características de SO en el servidor:**

1. soporte de red: para brindar conectividad
2. amplia compatibilidad con el hardware: se debe priorizar el uso de S.O. actualizados y con un soporte importante de controladores.

Para un aprovechamiento pleno de las características del servidor es que el S.O. sea capaz de exprimir al máximo las características técnicas del hardware en donde se ejecuta

1. seguridad: con todos los parches/actualizaciones y políticas estrictas de acceso para prevenir accesos no autorizados o ataques.

instalación de Firewalls (por software o hardware), antivirus y respaldo de la info.

1. tolerancia a fallos:generación de granja de servidores interconectados, que operen como una gran unidad de proceso, dando la posibilidad que ante la caída de uno, otro puede tomar su rol y responsabilidad

***Diferencias con SO de Clientes:***

1. manejo diferente del Hardware: los S.O. de estaciones de trabajo no pueden aprovechar todo el hardware disponible, como, por ejemplo, el manejo de memoria RAM
2. Características soportadas:funcionalidades que nativamente un S.O. de estación de trabajo no es capaz de brindar, su versión de kernel están limitadas o deshabilitadas —casos tales como la virtualización en algunas versiones de Windows 10—
3. Soporte:los S.O. de estación de trabajo, el soporte/cobertura es para un uso específico, si sobre esa base quisiéramos desplegar una arquitectura, vamos a carecer de soporte técnico ya que el fabricante nos indicará que para ese propósito esta la version “Server” del producto.

***SO de los servidores:***

windows 75%incluye:

Windows Server, desde versión 2003 a 2019

Unix 25% incluye:

GNU Linux, FreeBSD, macOsServer

**Windows:**

lanzado en 1985. Al principio funcionó como una interfaz gráfica encima de MS-DOS. Luego las características de MS-DOS se integraron en Windows 95.

posee 88,14% del mercado.

Uno de los mejores antivirus: Windows defender

china 90% copias ilegales

|  |  |
| --- | --- |
| PROS  integración, compatible mayoría apps  soporte  facilidad de uso | CONTRAS  seguridad: tiene más malware  costo: es caro  tamaño: algunos mucho espacio |

**MAC**

fue lanzado en 1984. Comenzó a utilizar una interfaz gráfica desde su inicio. T

odas sus actualizaciones o versiones reciben un nombre en clave para los usuarios, Catalina 2019, Big Sure 2020

2do más usado nivel mundial, basado en Unix

Solo personal de Apple accede al código fuente. Fácil sincronización con dispositivos Apple

|  |  |
| --- | --- |
| PRO  Estabilidad entre hard y soft, casi no tiene fallos  seguridad, menos vulnerable para malware  diseño: estricto y fácil | CONTRAS  Costo  Compatibilidad: muchas apps no compatibles  No apto para gamers |

**GNU/LINUX**

núcleo (kernel) se conoce como Linux, creado por el ingeniero de software finlandés Linus Torvalds.

múltiples componentes y herramientas del proyecto GNU, como un ambiente de escritorio gráfico, editor de imágenes, bibliotecas para lenguajes de programación, compiladores, entre otros

286 distribuciones

1,34% mercado lo usa

97% de pcs más rápidas lo usan

|  |  |
| --- | --- |
| PROS  costo, crear, modificar y redistribuir gratis  pocos virus  estable, amplio uso en servidores | CONTRAS  complejidad para entender y manejar  compatibilidad: parches y conversiones para usar muchos programas  portabilidad de distribuciones: no es prioritaria |

**SERVICIOS**

Puerta de entrada que el SO ofrece para usar y ejecutar programas, archivos o asignar recursos

Los más usados de forma nativa o instalando un software son:

-publicación web

-base de datos

-correo

-archivos

-red

-dominio

**Publicación web**

despacha contenido web al usuario

el sistema hace una búsqueda DNS para encontrar en cuál servidor está alojado el sitio en cuestión.

le pide el contenido del sitio web y, acto seguido, el webserver procesa este pedido y envía dicho contenido al navegador,

Ejemplos: APACHE, codigo abierto mas popular, en Linux multiplataforma

Internet Indormation Services de Windows, Nginx abierto alto rendimiento multiplataforma,

LiteSpeed propiedario, Linux.

**Base de datos**

Database server o RDBMS (Relational DataBase Management Systems) en caso de bases de datos relacionales, oftware de servidor que permiten la organización de la información mediante el uso de tablas, índices y registros.

Extendido el uso de bases de datos no relacionales, conocidas popularmente como No-SQL.

Ejemplos: MySQL lider en webs y apps, multiplataforma

PostgreSQL open source, orientado a objetos, multiplataforma relacional

Microsoft SQL SERVER compatibilidad con lenguaje ASP/ASP.NET, multiplataforma relacional

MongoDB libre guarda en estructuras BSON, multi y no relacional.

**Correo**

na de las aplicaciones más populares en usar el protocolo TCP/IP

MTA mail transport Agent, agente de transferencia de correo (ej. Qmail, Exim, Cyrus entre otros) y MDA mail delivery Agent, recibe mail y lo lleva al inbox (dovecot, procmail y maildrop)

**Archivos**

proporciona a los usuarios un lugar de almacenamiento centralizado para los archivos en sus propios soportes de datos, disponible para todos los clientes autorizados.

Propio de cada sistema oerativo. mas usados: CIFS/Samba(Linux), NTFS(Microsoft) y NFS(UNIX).

**Red**

propios para satisfacer ruteo, firewall o proxy, se aplica más tomando como base sistemas Linux, soluciones: PFSense, OPNSense y DD-WRT

**dominio**

Servidor que responde a las solicitudes de autenticación y verifica a los usuarios en las redes informáticas.

Cuando los usuarios inician sesión en su dominio, el DC verifica su nombre de usuario, contraseña y otras credenciales para permitir o denegar el acceso.