**INTRODUCCIÓN A LA INFORMÁTICA**

**GLOSARIO TECNICO**

**HARDWARE**

Partes físicas, tangibles, de un sistema informático, sus componentes eléctricos, electrónicos, electromecánicos y mecánicos. Ej: cables, gabinetes, periféricos de todo tipo y cualquier otro elemento físico involucrado; contrariamente, el soporte lógico e intangible.

**Entrada:** 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 (teclado, mouse, escáner, micrófono, cámara web, joystick, lectoras de CD, DVD o BluRay, etc.)

**Salida:** permiten dar salida a la información resultante de las operaciones realizadas por la CPU (monitores, impresoras, consolas y altavoces).

**Internos**: componentes físicos que forman parte del dispositivo principal, siendo fundamental para el funcionamiento (Placa base, CPU, RAM, GPU, HDD, SDD).

**SERVIDORES**

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.

* **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.
* **Servidor de archivos:** almacena archivos y distribuye a otros clientes en la red.

**SOFTWARE**

Soporte lógico al sistema formal de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware. La interacción entre el software y el hardware hace operativo un ordenador (u otro dispositivo), es decir, el software envía instrucciones que el hardware ejecuta, haciendo posible su funcionamiento.

**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**

Conjunto de herramientas que permite al programador desarrollar programas de informática, usando diferentes alternativas y lenguajes de programación, de una manera práctica (Editores de texto, Compiladores, Intérpretes, Enlazadores, Depuradores, (IDE) Entornos de desarrollo integrados).

**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)

**INTERFACES DE USUARIO (UI)**

Espacio donde se producen las interacciones entre humanos y máquinas, con objetivo de permitir funcionamiento y control más efectivo del sistema informático desde la interacción.

**Interfaz de línea de comandos (CLI)** Interfaces alfanuméricas (intérpretes de comandos) que solo presentan texto.

**Interfaz gráfica de usuario (GUI)** Permiten comunicarse con la computadora de forma rápida e intuitiva representando gráficamente los elementos de control y medida.

**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.

**COMPUTADORA**

Dispositivo electrónico capaz de recibir instrucciones dadas por un usuario por medio de interfaz (hace más sencilla la comunicación entre usuario y máquina), que presenta sistema operativo.

El sistema operativo interpreta y ejecuta con los recursos que dispone, a estos recursos se les conoce como hardware y todo lo que interpreta y ejecuta los deseos del usuario es el software.

**FUNCIONES:**

* Cálculos complejos
* Procesar datos
* Interpretar datos
* Comunicación entre dispositivos
* Comunicación con seres humanos

**HISTORIA**

**1122 China:** invención del Abaco (Instrumento para realizar operaciones aritméticas).

**1642 Francia:** Blaise Pascal inventa la pascalina (primera calculadora metálica intentando ayudar a su padre en los cálculos de su trabajo).

**1801 Francia:** Joseph Jacquard inventa las tarjetas perforadas (una forma de guardar información, para que cualquiera pudiera crear diseños de telas muy complejos).

**Entre 1833 y 1842 Gran Bretaña:** Charles Babbage intento crear la primera computadora.

* Ada Lovelace diseña el primer algoritmo.

**1936 Reino Unido:** Alan Turín, la computadora logra ejecutar programas almacenados debido a la máquina de Turín (creada con la intención de descifrar los mensajes alemanes en la segunda guerra mundial)

**1943 USA:** se creó una de las primeras computadoras conocida como ENIAC, operada por las chicas del refrigerador. Junto con Ada Lovelace fueron consideradas las primeras programadoras de la historia.

**1945 Hungría:** Von Neuman diseño una arquitectura que estableció las pautas actualmente utilizadas de como una computadora debería ser armada para optimizar al máximo sus recursos.

**40’s** Primera generación.

**60’s** Tránsitor – chip.

**70’s** Microchip.

**80’s** Windows Apple Linux.

**2011 IBM:** entramos en la era cognitiva, donde encontramos tecnología como la computadora cuántica (pueden ejecutar en 200 segundos cálculos complejos, otras computadoras 10 mil años)

**¿Qué deseamos a futuro con la inteligencia artificial?** Un agente flexible.

**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. interfaz de línea de comandos con la que vamos a interactuar, se encarga de procesar datos y devolver resultados.  
  
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.  
  
Una consola es un tipo especial de terminal. Es una instrucción específica dada a una aplicación informática para realizar algún tipo de tarea o función. La línea de órdenes permite 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.

Programa presente en todos los sistemas operativos, con el que podemos darle ordenes al sistema o computadora a través de ejecución de comandos, ordenes que el usuario proporciona a un sistema informático mediante texto.

**Ventajas**

* Los comandos son más rápidos de ejecutar que en una interfaz gráfica (velocidad).
* Los comandos siempre funcionan de la misma manera y nos permite trabajar con cualquier versión (compatibilidad).
* La línea de comandos es una interfaz limpia (sin distracciones), permite personalizar.
* Mas productividad.

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

* **La entrada** (stdin) archivo por el que una orden recibe su entrada (teclado).
* **La salida** (stdout) archivo por el que una orden presenta sus resultados (pantalla o más concretamente la ventana en la que se está ejecutando el intérprete de órdenes).
* **El error** (stderr) archivo por el que una orden presenta los mensajes que va generando cuando ocurre un error (por defecto, también es la pantalla).

Todos estos archivos se crean en el directorio /dev.

**VISUAL STUDIO CODE O VSCODE**

Es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft. Un IDE es un conjunto de herramientas diseñadas para facilitar creación y desarrollo de nuestros programas o aplicaciones.

**GIT BASH**

Aplicación de terminal que se utiliza como interfaz con un sistema operativo mediante comandos estrictos (CLI: interfaz de línea de comandos).

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

**Crear o eliminar carpetas**

* mkdir / Crear un carpeta (make directory).
* 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)

**COMANDOS**

Son instrucciones codificadas para ser un sistema operativo.

Las computadoras nos proveen comandos que podemos ejecutar en la terminal, para enviarles estas instrucciones de forma clara y precisa.

* PWD: imprimir directorio de trabajo.
* CD: cambiar de directorio (se llama argumento = la ruta que se escribe después del comandó).
* LS -T: los ordena por fecha de modificación.

**RUTAS:** Absolutas (/=raiz) y Relativas (=raiz)

**¿A qué llamamos líneas de comando?**

* Interfaz de usuario limpia y sin distracciones.
* Interfaz que permite personalizarla.

**Búsqueda en la terminal**

**CON FIND Y GREP**

**Búsqueda en la terminal**

**CON FIND Y GREP**

**BÚSQUEDA EN LA TERMINAL CON**

**find (encontrar archivos en un determinado directorio a partir de diversas reglas de búsqueda)**

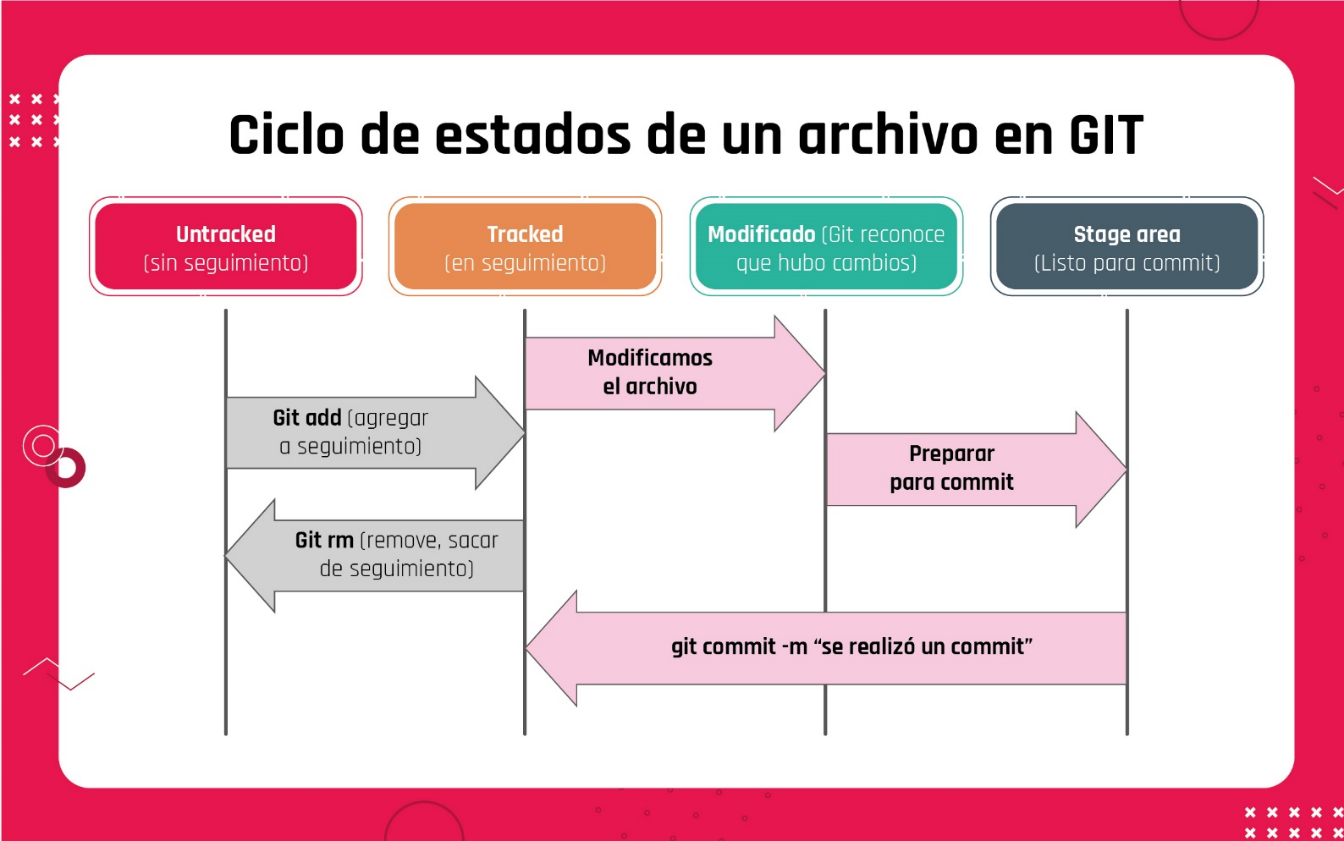
* **Buscar archivos o carpetas:** se hace por nombre completo o que coincidan con ciertas letras o frases, archivos de ciertos usuarios, creados en cierta fecha, etc.
* **Por nombre o tamaño:** podemos utilizar la opción -name para buscar por nombre o la opción -size si queremos buscar por tamaño de archivos.
* **Sensibilidad a mayúsculas:** Tanto en find como en grep, podemos utilizar la opción -i para que no se tenga en cuenta la sensibilidad a mayúsculas.
* **Carpeta actual:** En find y en grep podemos utilizar. (punto), el cual permite buscar a partir del directorio donde estamos ubicados actualmente.

**grep (Búsqueda global de expresiones regulares)**

* **Buscar en archivos:** 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.
* **Número de línea:** Cuando utilizamos grep, la opción -n nos muestra la línea donde encuentra la coincidencia dentro de el/los archivo/s.
* **Subcarpetas:** Cuando utilizamos grep, la opción -r, nos permitirá buscar en todas las subcarpetas recursivamente.
* Tanto en find como en grep, podemos utilizar “”(comillas) que nos permiten buscar un término que consista en más de una palabra.
* **Nombres con espacios:** tanto en find como en grep, podemos utilizar “” (comillas) que nos permiten buscar un término que consista en más de una palabra.



**git rm --cached:** especifica a git no realice más seguimiento al archivo, donde el archivo como tal no se ha eliminado, Git procede a ignorarlo y pasa a un estado de untracked.



**GIT**

Software de control de versiones en la nube con servicios como Github, para mantenimiento y confiabilidad de aplicaciones cuando tienen un gran número de archivos de código fuente, su propósito es llevar registro o historial de los cambios de los archivos y coordinar el trabajo que varias personas realizan sobre los mismos (comparten los archivos de una manera fácil), sin necesidad de crear varias copias, pudiendo tener todos los cambios registrados e incluso volver a la versión inicial

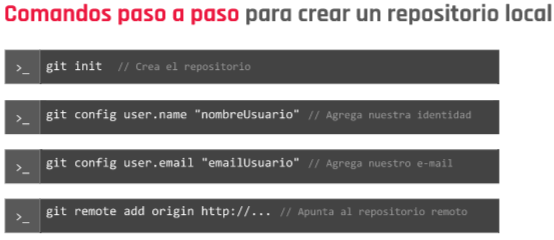
Mantiene eficientemente las actualizaciones sobre el código fuente, mantener un control de las versiones de los archivos y quien realiza las modificaciones.

**REPOSITORIO**

Almacén de archivos en pequeños paquetes llamados commits, historial de cambios y seguimiento, estos paquetes permiten ir haciendo un seguimiento de las modificaciones.

1 proyecto =1 repositorio

**REPOSITORIO LOCAL.**



Desde la terminal me paro en la carpeta y hago un git init (genera repositorio local)

* git config – global user.name
* git config user.name (ver el nombre de usuario, igual email).
* Ls -a (muestra archivos ocultos—muestra el archivo que se genera al iniciar el repositorio)

**REPOSITORIO REMOTOS**

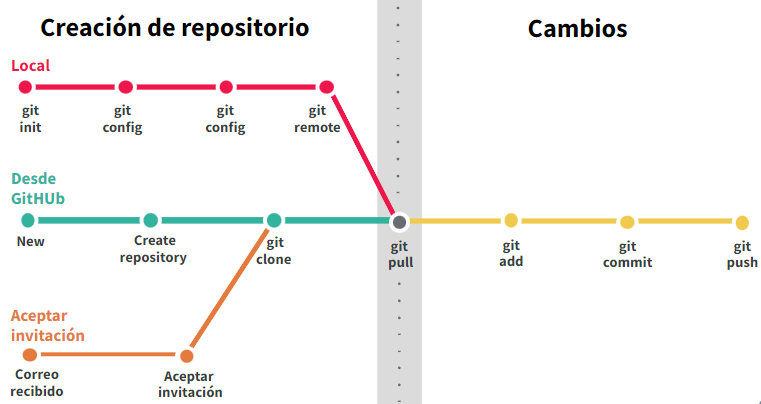


**AGREGAR ARCHIVOS**

1. Git add . (agrega todos los archivos).
2. Git status (seguimiento del estado del estado de los archivos respecto al repositorio)
3. Git commit -m “mensaje” (commit todos los cambios)
4. Git log (historial de todos los commit)

**COMMITS**

Paquetes que nos van a permitir hacer seguimiento o generar un historial de cambios que realizamos en un orden cronológico del proyecto, tienen fecha de creación, autor y mensaje, con ellos podemos revertir cambios y volver a un estado anterior.



**GITHUB**

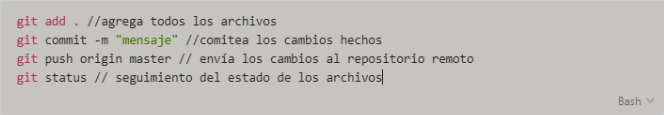
Sistema en la nube donde se hospedan proyectos de programación, se encuentran los repositorios remotos que permiten alojar versiones en línea y compartidas de archivos en sitio Web (GitHub), crea repositorio remoto (origin o MAIN) con URL única (tiene mi nombre y nombre del archivo).

* **Una rama:** línea paralela para agregar funcionalidades sin afectar a la rama principal.

**CONECTANDO NUESTRO REPOSITORIO LOCAL A GITHUB**

1. Tener una cuenta en GitHub.
2. Un repositorio local en la carpeta que queramos conectar el repositorio (git init).
3. git config user.name “mi usuario” (escribimos nuestro nombre de usuario).
4. git config user.email “[miCorreo@email.com](mailto:miCorreo@email.com)” (escribimos nuestra dirección de correo).
5. Git remote add main URL.. (indica el repositorio remote al que se deba asociar).
6. Git remote -v (muestra vinculación repositorios).

**SUBIENDO ARCHIVOS DEL REPOSITORIO LOCAL AL REMOTO**

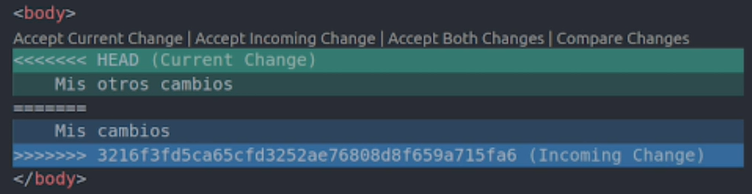


**BAJANDO ARCHIVOS DEL REPOSITORIO REMOTO AL LOCAL**

1. git clone "URL del proyecto" (descargar por primera vez un repositorio, solo una vez).
2. git pull origin master (actualizar el código en tu repositorio local).

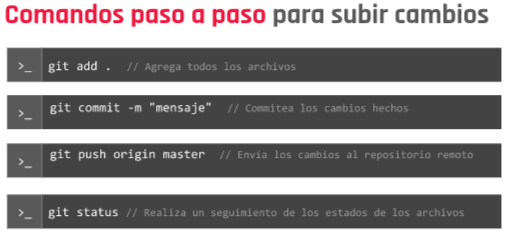
**RESOLVIENDO CONFLICTOS**

* Signos < y = serán nuestra versión local
* Signos = y > será lo que nos llega desde Github

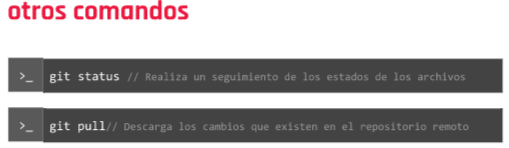


Para evitar conflictos o que sean pequeños, hacer commits pequeños y subirlos frecuentemente.

**SUBIR CAMBIOS DESPUES DE MODIFICACION ORESOLVER EL CONFLICTO**

****

**OTROS COMANDOS**

****