GIT

¿Qué es?
Es un software de control de versiones. Mantiene eficientemente las actualizaciones sobre el código fuente.

Ventajas:

- Compartir proyectos con otras personas
    - Coordinar el trabajo que varias personas realizan sobre los mismos
        - Qué persona realizó la última modificación
- Tener varias versiones de un mismo proyecto
    - Llevar registro de los cambios de los archivos
- Guardar proyectos en la nube
    - Tener backup de nuestros archivos continuamente actualizados
    - No es necesario tener varias copias del proyecto

¿Qué es un repositorio?
Es el lugar donde se irán almacenando los archivos de nuestro proyecto

Tipos de repositorios:
Los repositorios que se alojan en GitHub los llamamos repositorios remotos, mientras que a los que se alojan en nuestra PC los llamamos repositorios locales. Es necesario crear un vínculo entre ambos para poder mantener actualizados los archivos locales que están conectados a ese repositorio en la nube

REPOSITORIO LOCAL
Es un almacen de archivos, donde se irán almacenando en pequeños "paquetes" que llamamos **commits**

- Los commits permiten hacer un seguimiento de los cambios que se van realizando, dado que cada uno de ellos tiene:
    - Fecha de creación (timestamp)
    - Autor
- Antes de comenzar, se debe configurar la cuenta para que el programa identifique quien hace los cambios. Se hace con la cuenta con la que abrimos Github
    - Cuenta:
        - git config user.name "mi-usuario"
        - git config user.name (para corroborar que el usuario se configuró correctamente)
    - Correo:
        - git config user.email "miCorreo@email.com"

        - git config user.email (para corroborar que el correo electrónico se configuró correctamente)
    - También se puede configurar de manera global (solo una vez), para que todos los repositorios que se hagan en ese computador, sean con mi cuenta:
        - git config --global user.name "mi-usuario"
        - git config --global user.email "miCorreo@email.com"
- Cambiar nombre de rama principal repositorios
    - git config --global init.default branch main
- Iniciar repositorio
    - git init ---> Crea un repositorio local (computadora)
- Agregar archivos al repositorio
    - git add archivo.ext
        - git add . o git add * ---> agregar todos los archivos presentes en el repositorio
            - Aunque lo más conveniente es agregar funcionalidad por funcionalidad (de a una)
    - git status
        - Muestra el estado de los archivos
        - Cuando modifico los archivos que ya había agregado, aparecerán en rojo porque git los toma como "nuevos", "sin seguimiento", por lo que debo volver a usar el comando git add
            - Los status pueden ser:
                - U ---> Untracked
                - A ---> Added (por primera vez)
                - M ---> Modified
- Confirmar modificaciones de archivos
    - git commit -m "mensaje"
        - Confirmación de agregado de archivos
        - Podemos ver fecha y autor de las modificaciones
        - Generan puntos en la línea de tiempo del proyecto que nos permiten identificar el estado del mismo hasta ese momento específico y volver sobre los mismos si así lo queremos
        - Se recomiendan hacer commits "pequeños", ya que se pueden visualizar más fácil los cambios. Además, el mensaje del commit permite únicamente 50 caracteres, por lo que no se pueden hacer muchos cambios en uno solo.
            - Buenas prácticas con los commits
                - [FIX] - [ADD] - [MOD] - [FIX] ---> git commit -m "[FIX] funcion1"
        - SHA (Simple Hasing Algorithm) ---> Se refiere al número de commit
    - git log ---> registra un historial de los cambios en el proyecto
    - git blame archivo.ext
        - Se usa para saber quién erró en cierto commit
- Recuperar archivos eliminados

    - git restore .
- Eliminar un repositorio
    - rm -rf .git

Navegación e interacción con ficheros
De tipo informativo:

- pwd ---> mostrar nombre de la carpeta en el que uno se encuentra situado [Print Working Directory]
- dir ---> mostrar los archivos de la carpeta en la que estamos ubicados
- cd ---> cambiar la carpeta de trabajo. A lo que va luego del comando lo llamamos **argumento** [Change Directory]
    - cd .. ---> nos permite retroceder una carpeta
    - cd nombreCarpeta ---> Acceder a la carpeta que le indiquemos
- ls ---> listar el contenido de directorios (ficheros y carpetas) [List]
    - ls -t ---> indicación llamada **opción** o **flag**, que ordena archivos por fecha de modificación
    - ls -R ---> lista todos los archivos y carpetas que hay a partir del punto donde uno está
    - ls -a ---> muestra los archivos ocultos
- cat ---> mostrar contenido de un archivo
    - cat house.txt

Crear o eliminar carpetas

- mkdir ---> crear una carpeta
    - mkdir nombreCarpeta
- rmdir ---> borrar una carpeta

Crear o eliminar archivos

- gedit (Linux)---> crear y editar archivos de texto
- touch (Windows)---> crea un archivo de cualquier tipo (no solo de texto) con el nombre que le indiquemos
    - touch nombreArchivo.ext
    - echo $null > nombreArchivo.ext ---> para Powershell
    - Crear varios archivos al tiempo en una carpeta
        - touch carpeta3/introduccion.txt carpeta3/informatica.txt
- rm ---> borrar
    - Borrar un archivo
        - rm archivo.ext
    - Borrar varios archivos de una carpeta
        - rm carpeta3/eliminar.txt carpeta1/eliminar.txt
    - Borrar un directorio (carpeta)
        - rm directorio -r
- nano ---> editar textos y crear archivos nuevos para no sobreescribir los anteriores

    - nano house.txt
        - Ctrl + W
        - Cambiar nombre
        - Ctrl + M
        - Y
        - Ctrl + X

Copiar archivos y directorios

- cp ---> copiar un archivo o carpeta en el directorio especificado
- cp -r ---> copiar carpetas
- mv
    - mover un archivo o carpeta a otro archivo o carpeta
        - mv auditoria nuevaCarpeta
    - mover varios archivos del mismo tipo a una carpeta
        - mv *.jpg imagenes
    - mover archivos que comiencen con una palabra (o secuencia de letras) a una carpeta
        - mv leer* textos
    - cambiar el nombre de un archivo por el nombre nuevo que le indiquemos
        - mv file1 file2
    - mover un archivo a una carpeta anterior
        - mv house.txt ..
    - mover un archivo a una carpeta (que está posicionada en el mismo lugar)
        - mv house.txt carpeta1
    - mover un archivo desde una carpeta a una carpeta exterior
        - carpeta1 ---> house.txt
        - carpeta2 (vacía)
        - mv house.txt ../carpeta2
            - carpeta1 (vacía)
            - carpeta2 ---> house.txt
    - renombrar varios archivos al tiempo
        - mv carpeta1/crear.txt carpeta1/digital.txt

Abrir archivos

- start ---> abrir archivos en un programa
    - start winword house.txt ---> Word
    - start wordpad house.txt
    - code house.txt ---> VSCode

Regresar a versiones anteriores:

- git checkout funcion.txt

Ayuda consola

- -i ---> pregunta antes de remover
    - rm house.txt -i
        - y
        - n
- -h (--help) ---> ayuda con un comando
- Ctrl + C ---> detiene la ejecución de comandos