

Clase 3: 18/02/2022

- VSCode
- Git Bash (Terminal) – Comandos
 - pwd muestra en la carpeta que estamos
 - cd .. ir para atras
 - cd ir a directorio
 - mv mover archive
 - mkdir crear directorio
 - touch crear un archivo
 - rm remover archive
 - rmdir remover directorio vacio
 - rm -r remover directorio con archivos
 - ls listar archivos de la carpeta
 - ls -a listar archivos de la carpeta incluso los ocultos
 - ls -R muestra los archivos y carpetas de forma recursiva (todo)
 - cp copiar un archivo o carpeta
 - clear limpiar terminal
 - ctrl + l limpiar terminal
 - "" para nombrar archivos o carpetas con espacios en el medio

Clase 4: 21/02/2022

- C4A
 - git init //crea repositorio
 - git config user.name "nombre de usuario"
 - git config user.email "mail"
 - git add "archivo"
 - git add . //agrega todos los archivos que no se agregaron antes
 - git status
 - git commit -m "comentario"
 - git log
 - git rm --cached //git deja de hacer seguimiento

- Untracked //sin seguimiento
- Added o Tracked //agregados con git add
- Modified //archivo agregado modificado. Hay que agregarlo de vuelta y comitearlo

Clase 5: 23/02/2022

- C5A
- Subir archivos a Git Hub
 - GitHub //repositorio en la nube
 - git remote add origin <https://github.com/LKN10/mercadoAbierto.git> //asi comunicamos nuestro repositorio local con uno en Git Hub

- Una vez enlazados los repositorios, cada vez q agreguemos un archivo o lo modifiquemos en el local, hacemos lo siguiente para agregarlo en la nube.
 - `git add "archivo"`
 - `git commit -m "comentario"`
 - `git push origin main`
 - `git remote add origin`
<https://github.com/LKN10/mercadoAbierto.git>
- Descargar archivos de GitHub
 - `git clone "URL donde esta el repositorio"` //cuando no existen previamente en la pc
 - `git pull origin main` //para actualizar un repositorio desde GitHub que ya estaba en la pc
 - `git remote -v` //
- Nueva Rama o branch
 - `git branch` // Enumera todas las ramas de tu repositorio, es similar a `git branch --list`.
 - `git branch <branch>` // Crea una nueva rama llamada <branch>.
 - `git branch -d <branch>` // Elimina la rama llamada <branch>. Git evita que eliminemos la rama si tiene cambios que aún no se han fusionado con la rama Main.
 - `git branch -D <branch>` // Fuerza la eliminación de la rama especificada, incluso si tiene cambios sin fusionar.
 - `git checkout nombre_rama` // Para moverse de una rama a otra.