Uso de la terminal

Indice

Comandos de terminal:

- Prompt default de mac
  - [maquina] : [cwd = current working directory] [user]$
  - ~ es home
- Comandos:
  - cd [directorio]
    - Ingresar en directorio
  - cd ..
    - Salir del directorio actual un nivel
  - cd
    - Si se ejecuta el comando solo, devuelve al home
  - ls
    - Enlista contenido del directorio actual
  - mv [fuente] [destino]
    - Mueve el archivo de directorio a otro o,
    - Renombra un archivo o directorio
  - mkdir
    - Crear directorios
  - cp [fuente] [destino]
    - copiar archivos
  - cp -R [fuente] [destino]
    - copiar directorios
  - pwd
    - “Print working directory”, nos muestra la ruta completa del directorio actual
  - rm [archivo]
    - elimina archivos
  - rm -r [directorio]
    - elimina directorio
  - touch
    - Crea archivos

    - open .
        - Abre el directorio actual en el finder

Importante: al trabajar en la terminal, siempre estamos dentro de un directorio. Todos los comandos se ejecutan relativos al directorio en el que estemos.

Comandos de git

- git clone [direccion del repo] [carpeta destino]
    - clonar el repositorio localmente
- git status
    - ver cambios dentro del repositorio local
- git add [archivo]
    - pone bajo seguimiento a los archivos agregados, para el próximo commit
- git commit -m “[mensaje]”
    - confirma los cambios realizados, el mensaje debe explicar cuáles dichos cambios
- git push origin [branch]
    - envía nuestros cambios confirmados al servidor de git.
    - Para enviar a la rama master, se envía como:
        - git push origin master
    - Si se tiene un solo branch en el repositorio, se puede enviar como:
        - git push
- git pull
    - En caso de estar trabajando en grupo, se debe correr el comando **git pull** antes de empezar a trabajar. Este comando trae a nuestra copia local, los cambios que se hayan generado en el repositorio y actualiza nuestro código local. Este comando además permite actualizar el código local, en caso de estar trabajando en dos computadoras diferentes.

Branches

- git checkout -b [nombre de nuevo branch]
    - Crea un fork de nuestro repo actual, con los cambios realizados y los envía al nuevo branch creado.
- git checkout [nombre del branch]
    - Cambia el branch en uso. Por ejemplo, si tenemos 2 branches, “**master”** y “**menutheming”**, podemos pasar de uno a otro de la siguiente manera:
    - git checkout master (nos pasa a la rama master)
    - git checkout menutheming (nos pasa a la rama menutheming)
    - Otros usos del comando checkout se verán más adelante
- git branch

    - Nos muestra el branch en el que estamos trabajando actualmente
- git push -u origin [branch]
    - Envía nuestros cambios confirmados hacia el branch que creamos. El parámetro **-u** define el branch actual por defecto para los siguientes **push.** De esta manera podemos hacer **git push** y se enviarán al branch recientemente creado.
    - Importante: recomiendo el usto de **git push -u origin [branch]** para evitar conflictos en el repositorio.