Chatsheet comandos más utilizados

pwd
arroja el directorio de trabajo donde se esta posicionado

cd <nombre de carpeta>
cambia de directorio hacia la carpeta indicada

ls
Arroja los archivos y carpetas que se encuentran en el directorio donde estamos parado

ls -t
Arroja los archivos o carpetas que se encuentran en el directorio donde estamos parado, ordenados según última modificación.

ls -R
Arroja la composición del directorio en forma recursiva. Muestra carpetas con su contenido.

mkdir <nombre carpeta>
Crea una carpeta en la ubicación donde estemos parados

mkdir Nombre_carpeta/Nombre_carpeta2
Crea una carpeta dentro de una carpeta previamente creada sin tener que cambiar ubicación a Nombre_carpeta

touch <nombre archivo editable>
Crea un archivo editable de texto. Le podemos asignar al mismo la terminación que deseemos, tal como .txt, .xls, .html, .doc, .ppt, etc.

nano <nombre de archivo>
Accede al archivo de texto y permite editar sus lineas, agregando o sacando información.

grep
Permite busca texto dentro de archivos.

grep -r
Premite arrojar el resultado de la busqueda de forma recursiva, mostrando carpetas y subcaretas.

git config --global user.name <nombre de usuario>
Establece el nombre de usuario que va a trabajar en el repositorio

git config --global user.email <email@email.com>
Establece el email del usuario que va a trabajar en e repositorio

git init
Inicia un repositorio en la carpeta donde estemos ubicados. Crea un archivo .git el cual servirá para gestionar el mismo.

git add <archivo>
Añade el archivo a la stage area, donde los cambios serán reconocidos por Git.

git add .
Añade todos los cambios sin especificar a la stage area.

git commit -m
Añade los archivos y cambios al repositorio local y lo confirma con una frase que sirva de orientación para otros y para el futuro seguimiento.

git status
Arroja el estado de los archivos dentro del stage para verificar que esten agregados, modificados o eliminados.

git push origin <nombre de rama>
Añade los archivos y cambios a la rama principal

git pull origin <nombre de rama>
Trae los cambios y archivos de la rama principal al repositorio local

git branch <nombre de rama>
Crea una nueva rama con su nombre

git checkout <nombre de rama>
Sale de la rama donde estaba parado y se posiciona en la rama que se le indica en el comando.

git remote add origin git@github.com/minombre/archivogit.git
Enlaza el repositorio local con uno remoto

git clone
Clona un repositorio remoto en una carpeta en un repositorio local

git merge
Fusiona mi rama con la rama principal

git rm <archivo>
Remueve archivo en seguimiento

git log
Muestra el historial