



`git config color.ui true` (para colorear la interfaz)

`cd` (dirigir con el terminal a una ubicación o carpeta)

`ls` (sirve para ver la lista de archivo)

`git merge nombredelarama` (Sirve para fusionar una rama con un commit master)

`git checkout nombre-a-donde-se-quiere-dirigir` (Sirve para fusionar una rama con un commit master)

`-m` (Sirve adjuntar un mensaje de guia)

`rm -Rf` (Sirve borrar un directorio)

`git branch nombre-de-la-rama` (Sirve para crear una rama)

`git status` (estado del proyecto)

`git log` (visualizar los commit creados)

`git commit -m "mensaje-que-deseas-poner"` (Crear commit )

`git add -A` (agregar archivos al commit por crear )

`git config --global alias.choque2015 "log --oneline --graph --all"` (agregar un alias)

`git "nombredelalias"` (observar el proyecto desde alias)

`git master`(Dirigir a la rama principal)

`git cloning`(clonar)

`git --version`(ver la version de git)

`git reset --hard` (borrar por completo todos los cambios hechos)

`git reset --soft` (te borra el proceso pero te mantiene los commits)

`git reset --mixer` (No te borra el contenido pero si la creacion del commit)

`git init` (crea un repositorio nuevo)





```
git remote add origin (conectar remotamente nuestro contenedor principal)
git remote add upstream (conectarse con el repositorio principal clonado para posteriormente dar propuesta)
git fetch origin (para descargar los cambios del contenedor principal)
git merge origin (mezclar cambios del contenedor principal con el que tenemos en local)
git push origin master (subir lo archivos a nuestro repositorio principal)
git remote -v (para visualizar las conecciones que tenemos vinculados)
$ ssh-keygen (agregar una llave ssh)
$ cd .ssh(entrar a la carpeta donde se aloja el ssh)
$ cat id_rsa.pub(ver el contenido)
http://nombre-del-usuario.github.io/nombre-del-repositorio(esto poner en edit para cargar un deployment del ejercicio)
cd .git/hooks (para entrar ala carpeta donde se alojan los hooks)
touch post-commit (para crear el archivo post-commit , este nos ejecutara comandos automaticamente despues del commit)
touch nombre.sh (Nos crea archivos sh scripts que sirve para automatizar comandos)
#!/bin/sh (para decirle al archivo sh o hooks que se ejecute dicho tipo de archivos)
chmod +x post-commit (para autorizar que se ejecute dicho hook)
ssh direccion-remota-via-ssh (para conectar remotamente el servidor con el local via ssh)
git pull origin master (actualiza la informacion de los archivos del servidor)
vim [nombre-del-archivo] (abre el archivo en el editor de texto de la terminal)
cd .ssh (entrar al archivo ssh para conseguir los keys necesarios de conexión)
sh nombredelservidor 'bash -s' < deploy.sh^ (para decirle al servidor que ejecute deploy.sh)
```

