**Comandos de Git y Bash**

**Basicos de bash:**

history

ls/dir: [-al]

cat: [file]

touch: ‘name of file’

pwd

cd

alias –‘name’=’variable’

**Basicos de git:**

git init

git add: [file] , [.]

git status

git config: [--list] , [--help]

git commit: [-m ‘ ”texto” ’] , [-am ‘ “texto” ’]

git log: [file] , [--all] , [--graph] , [--decorate] , [oneline] , [--stat]

git diff: [(codigo de commit codigo de commit)]

git show: [file]

git reset: [codigo de commit] [--hard] [--soft]

git checkout: [codigo de commit] [rama]

git rm –cached: [file]

git restore --staged: [file]

git branch: [rama]

git show\_brach: [--all]

git merge: [(rama rama)]

git tag: [-a ‘nombre de version’ -m 'texto']

**Conexión con Git Hub:**

git pull: [(‘origin’ master)]

git push: [(‘origin’ master)]

git remote: [-v] , [(set-url ‘origin’ ‘url con ssh’]

**Llaves Publicas y Privadas:**

eval $(ssh-agent -s)

~ (en codigo ascii: alt+126)

ssh-add ~/.ssh/id\_rsa

**Git stash:** La idea es que tengo cambias en una rama que no están todavía commiteados y entonces no puedo hacer checkout a otra rama. Entonces git stash guarda temporalmente los cambios existentes y ya no hay cambios que añadir, por lo tanto, podes visitar otras ramas.

git stash : [list]

[pop] :‘aca se recupera lo ‘stasheado’’

[(branch ‘nombre de rama’)] : ‘crea una rama con lo stasheado’

[drop] : ‘elimina el stash guardado’