Git – Dictionary

* Working tree

Working Directory: Where you are working with your repository, before a command git add.

Staging Area: After the command git add, all the added files will be like a waiting room, waiting the next commit of the branch.

Git Directory: This is the official repository, where all the files will be since the last commit made in the project.

* Essentials configurations

To start to use Git bash without problems, set this two configs.

git config – global user.name “Username” = To set your username

git config – global user.email “e-mail” = To set your e-mail

Git codes

git init = Create a repository

git init –bare = Create a repository with local share ( Obs: your current directory or folder will be used as a “.git”, creating a repository)

git status = Working Directory and Staged Area’s status

git clone file: ////<Local shared directory> <name>= Clone the folder of shared repository on your current local; Optional: If you want to put the cloned folder in a sub folder, just put a name after the shared repository

git rm –cached <file> = Unstage the selected file; It works with a dot (All)

git push <remote> <URL or branch/master> = Update the shared repository with your current commit

git pull <remote> <URL or branch/master> = Copy the shared repository to your current branch

git fetch <remote> <URL or branch/master> = Copy the shared repository to the selected branch

git remote = List all remotes (Default name is “origin”)

git remote add origin (remote repository URL)

ssh-keygen = to generate a key to connect your local machine with a GitHub’s account

1. Git add

git add <file> = Add the selected file to Staging area

git add <\*.type> = Add the selected type file to Staging area

git add . = Add all the files to the Staging Area

1. Git commit

git commit –m “Message” = Add the added files of Staging Area to Git Directory with a message about this commit

git commit –a –m “Message” = Add all the flies of Working Directory to the Staging Area and commit with a message about this commit to Git Directory at once

1. Checkout

git checkout <file> = Discard changes of your files on the Working Directory

git checkout . = With the same idea of “git add .”, discard all the changes of the Working Directory

git checkout <branch> = Change to selected branch

git checkout --b <branch> = If you want to create a new branch and change to it quickly

1. Log

git log = List all your commits with informations about who committed, when committed and Message commit

git log –p = List all your commits with more informations than “Git log”, including a list of modifications on files of each commit

git log --pretty=oneline = List all the commits with just Message and your Key

gitk = Open a Git Gui for more informations about your versions and commits

1. Diff

git diff = Check all the changes of Working Directory’s files

git diff --staged = Check all the changes of Staged Area’s files

1. Tag

git tag = List all the tags

git tag –a <name of tag> <key code of commit> –m “Message” = Create a tag on the actual commit or with the selected commit including key of commit

git tag –d <tag> = Delete the tag

git show <tag> = Go to commit where the selected tag is marked

1. Branch

git branch <name>

git branch = To list all of branch of your repository

git branch –d <branch> = To delete the selected branch after the command “git merge”

git branch –D <branch> = To delete the selected branch without use the command “git merge”

git merge <branch> = Merge the changes of a branch in other branch

1. CD

cd <directory or folder> = Change directory where git is located

cd .. = Return to directory before