***Git Essentials (CLI)***

***Config inicial do Git:***

git config --global user.name "*LGR3*"

git config --global user.email *lgr3*[*@cin.ufpe.br*](mailto:pam4@cin.ufpe.br)

***|| OU Clona um repositório remoto*** (previamente criado no GitHub)***:***

git clone *git@github.com:lgr3/Meu\_Repo\_no\_GitHub.git* # https address ou ssh

git clone *usuário@servidor:path-caminho/para/repositório.git* # **SSH *sempre* melhor**

***|| OU Inicia o Git e add remote depois:*** ***# Normalmente, não é bom para iniciantes.***

git init**;** git add -A**;** git commit -m "*Initial commit Projeto XYZ*"**;**

git remote add origin *git@github.com:<github-username>/<repo-name>.git*

git push -u origin master

git remote -v # checa a origem remota do repo

***Fluxo padrão:***

git branch # mostra as branches locais

git branch **-a** # mostra todas as branches locais e remotas

git checkout Dev02 # muda da branch atual para a Dev02

git checkout **-b** Dev03 # cria a nova branch Dev03 e já checkout-a pra ela

***ATUALIZAR EM RELAÇÃO AO REMOTO (Se alguém deu ‘git push’ mesma branch):***

**git remote update** # necessário se mais de um developer trabalha>

**git fetch -a** # >na mesma branch; isto garante sincronicidade.

**git pull /** (ou) **git pull -a** # puxa (download) o que tiver no repo remotor

Caso aja problema para pull # git branch --set-upstream-to=origin/master master

Se houver inconsistências ou infos locais desatualizadas no cmd ‘*git branch -a*’, utilize:

*git remote update* ***--prune*** # deleta restos de informação e branchs do repo local.

***Fluxo de Desenvolvimento; Dev e depois:***

git status# aparecerá em vermelho os arquivos não *staged*

**git add .** # ou **git add -A** # o paramtr **-A** é mais completo.

git status# aparecerá em verde os staged a serem commitados

**git commit -m “**Mensagem titulo do commit**”**

**git push** # já na branch em si # ou: **git push** origin <branch-name>

*MERGE a branch Dev para a Master: (****o ADMIN*** *do Projeto que faz isso a cada Sprint):*

git checkout master

**git merge** Dev03 # melhor usar ***pull request*** no ***GitHub.***

git commit -m “merge Dev-versao-tal para a Master”

**git push** origin master # melhor usar ***pull request*** no ***GitHub.***

***Resetar modificações não commitadas: # comum qnd fzndo testes s/ commitá-los.***

*So, you should reset the changes with:* ***# bastante importante.***

**git reset** --hard HEAD

*Then clean the untracked files and directories with:*

**git clean** -df # Ref: git-scm.com/docs/git-clean

***Git Ignore*** *(.gitignore)****: # Extremamente important/relevante!***

Se trata de um arquivo oculto, na pasta base do repo, onde se diz o que deve-se ignorar.

ex.: pastas /temp e arquivos temporários de compilação ou runtime do seu projeto.

Repositório GitHub oficial de arquivos *.gitignore* atualizados às tecnologias de software:

<https://github.com/github/gitignore>

***EXTRAs:***

**git remote set-url origin** # mudar/setar a url *(https||SSH)* da origin *(repo remoto)*

*=> Commitar mudanças atuais (staged ou não) em outra branch:*

**git stash**  # *empilha as mudanças realizadas até então*

**git checkout <outra-branch>**

**git stash list** # *lista os stashes de mudanças ‘empilhados’*

**git stash pop** # ***pop*** *as modificações stash-eadas p/ a branch atual*

**git cherry-pick *-x*** <branch-name Ou comit-hash>

*# aplicar um commit de uma branch qlqer (último commit ou o hash) a esta branch atual:*

Estudar \*\***git revert\*\*** para desfazer commits/pushs *(parece interessante-útil / necessário)*

***REFERÊNCIAs / REFERENCEs:***

<http://rogerdudler.github.io/git-guide/index.pt_BR.html>

***[Git]:***

[http://git-scm.com/](https://git-scm.com/)

GitHub Flow:

<https://guides.github.com/introduction/flow/> # google image for: Git Flow

[Git Immersion](http://creativecommons.org/licenses/by-nc-sa/3.0):

<http://gitimmersion.com/>

Codecademy:

<https://www.codecademy.com/learn/learn-git>

Git - #1 - Introdução - Rodrigo Branas

<https://www.youtube.com/watch?v=C18qzn7j4SM>

Code School Git:

<https://www.codeschool.com/learn/git>