**Estudo GIT – João e Cayo**

**Comandos Essenciais**

Vê e-mail e usuário que está usando – git config - - list

Iniciar repositório Local – git init

Trazer repositório existente localmente – git clone

Juntar Repositório Remoto com Repósitorio Local – git remote add origin URL

Limpar seu terminal GIT – clear

Clonar meu repositório Remoto no repositório local – git clone URL

Clonar meu repositório sendo numa branch diferente – git clone -b <nome da branch> <URL>

Trazer o que tenho no meu remoto para o meu repositório local sem clonar – git pull origin máster

Saber o que esta acontecendo no seu git – git status

Para rastrear os arquivos não rastreados – git add \* = todos os arquivos ou git add <nome do arquivo>

Comitar os arquivos quer dizer que meu arquivo ou código modificado no arquivo esta salvando – git commit -m “comentário”

Mandar o que eu alterei no repositório local para o repositório remoto – git pull origin máster

Criar branch localmente - git checkout -b “nome da nova branch”

Mudar de branch – git checkout “nome da branch”

Arquivos que não precisa subir para o git , cria-se uma extensão - .gitignore

\*Git FLow é o fluxo de branchs

\*Git fork é vamos dizer um clone no repositório de outra pessoa feita no git remote, dai você faz uma alteração e pra que a pessoa aceite você envia um pull request pra vê se ela aceita.

Rodar projeto sem o ReadME – git push -u origin máster

Adicionar ReadMe no terminal – git add “ReadMe.md”

Observações:

\*README de um projeto é preenchido pelo um analista de negócios.

\*Sempre usar o Origin , comando padrão para identificar o repositório remote;

\* Git Pull – Trazer

\*Untracked diz arquivos não rastreados;

\*Após alterar um arquivo, novamente tem que rastreá-lo;

\* O git só salva uma alteração no código se você comitar.

\*Colocar na mensagem do comitt , algo objetivo e descritivo do que você está fazendo.

\*Git Push – Mandar ao remoto

\*Não é necessário sempre da o push após o commit, mas sim após terminar o seu objetivo.

\* Um projeto pode ter mas de uma branch , pois branch é a arvore de tipos diferentes tipos, onde cada pessoa vai ser designada a uma branch.

\*Quando cria uma branch localmente, automaticamente a ferramenta git troca a branch para criada.

\*Não usar espaços e também não usar acentos.

\*Master é a branch já para a produção.

\*Pull requast é de suma importância , para que após avaliação, se aprovado pode subir para a produção.

\*Develop branch criada da master

\*Release branch de desenvolvimento

\*feature branch criada da develop para alterações

\* Fluxo = Master -> Develop -> feature -> Develop -> release -> máster

\* Hotfixes bug na máster

\* Se for mudar de git hub para o lab é necessário a apagar as credenciais no seu computador, em Gerenciar credências.