Versionamento?

Registro de mudanças em arquivos, que possibilita recuperação ou acesso a versões anteriores

Desenvolvimento de código em colaboração com outros integrantes

O que é GIT?

git é um sistema de versionamento de código, que guarda os registros de versão como snapshots(fotos) do estado do projeto, além da referência/caminho para essa foto.

Git e suas operações locais

A maioria das operações feitas pelo git são locais e por isso boa parte das operações são praticamente instantaneas devido a facilidade de acessar arquivos em seu proprio computador

os comandos serão realizados via CMD

1º git --version : utilizado para ver a versão do git instalada na maquina

2º git config user.name : utilizado para configurar o nome do usuário do repositório

3º git config user.email : utilizado para configurar o email do usuário do repositório

Como fazer um pull de um repositório do github?

4º git clone e\_caminho\_HTTPS

5º git init : é utilizado para preparar o repositorio local para subir versões para o github, quando é iniciado o comando, dentro da pasta repositorio ficará uma pasta com .git

Estados do Git e oque ocorre em cada estado:

Untracked: nesse estado o repositorio ainda não foi mapeado, neste momento a organização ocorre toda localmente

Unmodified: nesse estado o repositorio já consta mapeado, é possivel através do comando:

6º git status

verificar o status do repositorio junto ao git, nesse estado é possivel realizar edições dos arquivos, fazendo com que através o comando

7º git add

possa ser informado ao git que aquela versão alterada poderá ser commitada, após o arquivo ser modificado ele entra no estada de

Modified: Nesse estado ele através do comando já informado git add subira para o estado de

Stage: nesse estado o arquivo poderá ser commitado e em seguida voltara ao estado de Unmodified

8º git diff : comando utilizado para mostrar onde houve alteração nos arquivos

Uma observação, para sair do comando git diff é necessário clica na tecla “q” dentro do console

Com o git diff é possível verificar oque foi alterado por outros usuário no repositório online. Exemplo git diff origin/master

9º git commit -m "mensagem qualquer" : comando utilizado para deixar tudo pronto afim de subir para o github

10º git log : é utilizado para ver o log de todos os commit’s já realizados

Uma observação, para sair do comando git log é necessário clica na tecla “q” dentro do console

11º git restore : é utilizado para que seja restaurada um arquivo que já foi salvo a modificação localmente. Antes de ir para área de stage(antes de da um git add)

Utilizando o git restore, após o arquivo esta na stage, segue dessa forma para voltar atras da modificação: git restore –staged .\nome\_do\_arquivo

12º git remote : este comando mostrará qual o nome do repositório remoto

Agora que já temos o nome do repositório remoto, necessitamos saber o nome da Branch

13º git Branch : mostra a Branch local

14º git push origin master : esse comando todo é para subir as alterações commitadas para o github web

15º git pull : puxa todo repositório online

16º git fetch : diferente do git pull, o git fetch só baixara itens do repositório que não constam em nossa maquina

Forma direta de fazer preparo do repositório local, push de todo repositório e push de alterações mais individuais

Enviar todo o repositório:

# Inicializar repositório local

git init

# Adicionar arquivos ao índice

git add .

# Criar um commit

git commit -m "Primeiro commit"

# Adicionar repositório remoto

git remote add origin https://github.com/Orlando1592/Testenado20.git

# Enviar arquivos para o repositório remoto

git push -u origin master

Enviar alterações mais simples

# Dentro do diretório do seu projeto

cd /caminho/para/seu/projeto

# Adicionar o arquivo ao "staging area"

git add nomeDoArquivo.txt

# Fazer o commit

git commit -m "Primeiro commit"

# Enviar (push) as alterações para o GitHub

git push origin master

oque são branch’s?

São ramificações de um código maior.

Por exemplo uma equipe de desenvolvimento pode ter determinados Devs, cada um desenvolvendo um item diferente do programa como todo, e no final juntar o código e ter o app completo

Como criar Branches?

Branch padrão : Master

17º git branch nome\_do\_branch : este comando cria uma nova branch

18º git checkout nome\_da\_branch\_destino : este comando seleciona a nova Branch, agora será nela que serão feito os commits

Como ver em qual Branch esta?

19º git log --decorate –online

Vamos imaginar que temos 2 branches:

Master e testing

A master esta com 1 commit

A testing esta com 1 commit

Agora, como fazer um merge(junção/mesclagem)?

A branch que eu quero que receba o código é a master, então tenho que esta nela

Git checkout master

Estando nela, vou utilizar o comando

20º git merge testing : com esse comando foi possível puxar os itens contidos no repositório da Branch testint