**Códigos git**

**git clone** <link do reposorio> ou <diretorio>

**git clone** –branch new-feature <repositório> - clona a partir do branch espeficico

**git log - -**visualizar historico

**git log - -oneline** – visualizar historicos simplificado

**git log -p –** visualizar historicos detalhado

**git log - -author= “nome do usuário”** – visualizar autor daquele commit

**git log –since=1.monht.ago –until=1.day.ago**  - visulizar commit por data especifica

**git log –pretty=”formta:%h %s”** – visualizar o commit formatado do seu jeito

**git pull** – visualizar (puxa) o repositório

**git status** – Visualizar o status atual do repositório

**git push** – Empurrar uma atualização para o repositório

**git restore - -source “codigo copiado”** – atualiza para a versão escolhida

/**Modificou 1º add e depois commit**/

**git add –** adiciona um Att para commitar se colocar-mos git add . ele ira adicionar todos os arquivos modificados

**git commit –** confirmar um atualização feita (-m “sfa” – serve para escrever uma mensagem)

//

**git checkout -b <nome do branch> -** Cria uma nova branch e já fica selecionado

**git branch –** Visualiza o branch atual e também é possivel criar uma nova branch com o comando git branch {nome}

**git switch <nome do branch> -** Muda de branch

**git fetch –** Mudar para um branch remoto;

em seguida digite:

**git switch –c <nome-do-branch-local> --track <remoto-origin>/<nome-do-branch-remoto>**

**git merge <nome do repositório> -** Puxa modificações para o repositório que deseja, após esse comando da um **push** para que seja feita a alteração

**git config - -local user.name “nome” –** inseri o nome do Dev no repositório local

**git config - -global user.name “nome” –** inseri o nome do Dev no repositório global

**git config user.name ou user.email –** user.name mostra o usuário e o user.email mostra o email cadastrado fazer um por vez

**git bare –bare –** cria um repositório puro que só contém alterações dos arquivos não contém a cópia dos arquivos fisicos **permitindo adicionar esse repositório como remoto em outro (ele não vai servir para edição final)**

**git remote –** Lista após o enter todos os repositórios remotos que o repositório local conhece(por enquanto não foi adicionado nenhum então ele não mostra nenhum)

**git remote add “nome e preferencia” (endereço na rede, URL ou nome pasta) –** Adiciona o repositório remoto a pasta main

**git remote –v –** Mostra a pasta do repositório remoto e a opção tanto de buscar(fetch) quando enviar (push)

**git remote rename (nome) (nome atualizado) -**  renomeia o repositório remoto que deseja trabalhar

**git clone -**  Trazer o repositório remoto para estação de trabalho

**git clone (endereço, URL, nome pasta etc) nome da pasta desejada** – e enter

**git push (repositório) (main ou qual estiver setado) -** empurra para o repositório escolhido do repositório especificado

**git pull (de onde) (pra onde) –** Puxa o repositório do local para sua estação de trabalho

**git merge {repositório da att} –** de uma ramificação da branch puxa a atualização feita para o repositório que você estiver

**git rebase {nome branch} –** Atualiza os dados da branch referenciada para a frente da branch main deixando a branch main no final da linha gerando uma única linha sem ficar gerando muti commit.

**Git log - -graph –** Mostra as linhas especificas de desenvolvimento

**OBS:** Caso façamos modificações em dois branchs teremos um conflito que devemos aprovar no software de edição VSCode etc.....

E caso não tenhamos atualizado os dados modificados e tentarmos efetuar alguma att na main ou master precisamos antes puxa as atualizações da branch atualizada senão não sera possivel atuallizar essa branch

Então sempre antes de começar a trabralhar verificar com git pull se ouve alguma alteração no projeto

**OBS:** A modificaçao não foi adicionada ainda então execute o comando abaixo

**Git checkout - - {file} –** Desfaz a modificação feita caso você não goste.

**Git restore {file} -** Desfaz a modificação feita caso você não goste

**OBS:** A modificaçao já foi adicionada então execute o comando abaixo

**Git restore - -staged {file} –** Retira a modificaçã feita precisando fazer o restore ou checkout para apagar a modificação

. **OBS:** A modificaçao já foi adicionada e commitada então execute o comando abaixo

**Git revert {rash} –** reverte a commit feita

**Git stash –** Salva modificação feita sem gera log para pode acessar mais a frente

**Git stash list –** Mostra a lista salva

**2 Opções para retornar ao trabalho salvo**

**Git stash applay {numero da stash} –** Aplica as modificações mais elas ainda se mantem na stash necessitando fazer depois um **git stash drop** para remover a stash salva

**Git stash pop –** Pega tiragit log

o último item salvo e faz o merge e já tira do stash

**Viajar no tempo**

Utilizar o comando checkout + a hash do commit

**Git checkout {hash do commit que deseja voltar} –** volta para a atualização da has inserida

**OBS:** Ao entra nessa area você pode fazer alguns testes modificações que não serão modificadas na branch main ou master por que essa branch que você estará trabalhando é exclusiva se vc fizer modificação e voltar para branch main tudo estará normal lá.

Caso queira salvar as modificações feitas ai é só criar um novo branch que suas modificações ficarão anexadas como de costume.

**Ver alterações feitas**

**Git diff –** Mostra o que foi alterado mais não commitado ainda a partir do momento que o arquiv foi adicionado ai o git diff não mostra mais a modificação

**Git diff {hash}..{hash} –** Você consegue vizualizar todas as alterações feitas do 1º endereço até o 2º endereço

**Gerar Versão**

Quando terminado um realese para entregar ao cliente podemos cravar um tag como versão tipo v1.0

**Git tag –a {nome da tag}-** Adiciona uma tag da versão o nome não tem regra mais é bom colocar ex v0.1.0

Se quiser adicionar uma mensagem depois do nome da tag adicione –m “e frase que você quer”

**Git tag –** mostra as tags disponiveis no seu projeto

**Git push meu-local {tag desejada}-** Empurra a tag da versão escolhida para o repositorio