

Versionamento - Git E Github Conteúdos

Criar acesso no GIT conta de usuário

git config --global user.name "nome desejado"

git config --global user.email "email desejado"

Visualizar dados = git config user.name

Visualizar dados = git config user.email

Comandos Básicos Git

git init = inicializar repositório

git remote = gerenciar repositórios remotos

Clonar repositório próprio

git clone <URL_do_Repositório>

git clone = faz um cópia de um repositório já existente

git clone = utilizado para repositório que são meus

Forkar repositório externo

git fork <URL_do_Repositório>

git fork = faz um cópia de arquivos, mas precisa de autorização de pull request

git fork = utilizado para repositório que não sou dono externos

fork repositório externo (requer autorização)

Adicionar mudanças em local específico

git add = adicionar mudanças no staged, resolver conflitos e rastrear arquivos

git add * = adicionar todo conteúdo do arquivo

git add local.tipo = adicionar a local específico

Add repositório remoto

git remote add origin + endereço = URL

Salvar as mudanças na área de Staged

git commit = salvar as mudanças na área de Staged

git commit -am "mensagem" = adicionar e comitar em um comando, duas operações

Subir para repositório remoto

git push = enviar as alterações do repositório local para remoto

git push origin remoto/main

Sincronizar repositórios

git pull = sincronizar o repositório local com remoto

Criar branches para trabalhos em um mesmo Projeto

git branch = gerenciar ramificações do repositório e criar galhos pra trabalhos em equipe

git checkout -b nome desejado, cria uma branch

git branch nome desejado, também cria uma branch

git branch -l = listar branch

deletar branch = git branch -D nome.tipo . obs = D maiúsculo

git checkout = navegar entre branches, em qualquer linha de tempo, atual ou passado
git checkout nome da branch , para navegar para branch desejada
alterar algo com commit já adicionado = git checkout rash desejada (identificação do commit, que pode ser encontrado com git log)

Mesclar Repositórios

git merge = Mesclar - unir branches
git merge nome da branch que deseja mesclar

Listar repositório remoto

git remote -v

Listar conteúdo de uma pasta com detalhes

ls -lha

Criar arquivo

echo "nome" > nome.tipo = criar texto no nome e aponta > cria um arquivo.tipo

Visualizar conteúdo do arquivo

cat < arquivo.tipo = visualizar conteúdo do arquivo

Criar arquivo/pasta

touch nome.tipo
mkdir nome.tipo

Navegar para dentro/fora de um diretório

cd nome.tipo navega/entra no arquivo/pasta
cd .. sai do arquivo/pasta

Git log entender e relacionar commits

git log = ver histórico de commit
git log --decorate = mais detalhes do commit
git log --author="nome" = os commit do do mesmo autor
git shortlog = apresenta os commits em ordem crescente
git shortlog -sn = mostra número de commits por autor
git log --graph = mostra graficamente os commits

Analisar o códigos antes de subir

git diff = mostra alterações no arquivo antes do commit
git diff --name-only = mostra local

Verificar sobre alterações de quem e quando

git blame nome.tipo = mostra que fez alterações

Passo a Passo Local dos arquivos com GIT

untracked = git não conhece a existência do arquivo, o mesmo somente foi criado
unmodified = arquivo existe no git mas não modificado
modified = arquivo modificado no git
staged = arquivo pronto para commit
obs: após commit o arquivo volta a ser unmodified

git show + rash do commit = mostra o que foi realizado através desse commit

Resetar erros/alterações

git checkout nome.tipo = retorna a mudança, antes de add

git reset HEAD nome.tipo = remover após add

Após commit voltar commit e as alterações de 3 formas:

git reset --soft = voltar commit ficando na área do staged

git reset --mixed = voltar commit antes do staged = unmodified

git reset --hard = ignora tudo que foi feito, apaga . Cuido ao usar hard....

git reset --soft/mixed + rash do commit anterior ao que possui erro

git ignore = ignora arquivos mencionados , oculta arquivos => nome.ignore e no local:

* .tipo arquivo ou arquivo

git stash = gera um arquivos antes de ser commit, algo a ser modificado

git stash apply = para retornar a trabalhar nas modificações

git stash list = list os stash

git stash clear = limpa os stash

Criar atalhos

Para comando status

git config --global alias.s status

git s, mostrará o status, pode ser aplicado para todos os comandos

tags = melhorar organizar e demonstrar alterações no repositórios

git tag -a versão -m "mensagem da tag de alteracao"

revert - modificações sem usar o reset, assim não perder históricos

git revert + rash do commit