



RESUMO Git GitHub

CONFIGURAÇÕES INICIAIS

git config - -global user.name "" - NOME

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

git config - -global core.editor "" - EDITOR DE CODIGO

INICIAR GIT EM UMA PASTA

git init

git config credential.helper store

para cadastrar email e senha pra nao ter que botar toda vez q der um push

Deseja criar pontos na história da produção de seu projeto

git add (nome do arquivo)

adicionar um arquivo que sofreu modificação para um espaço de espera

git add .

envia todos os arquivos para o container

git commit -m "descrição"

vai leva a alteração para o repostorio local ou seja vai criar um ponto na história

git commit -am " "

simplificação do git add + git commit

git commit —amend —no-edit

comitar o arquivo com o arquivo(commit) anterior

git stash

caso faça alguma mudança desnecessária no momento usar o git stash (tipo esconderijo) onde ficara as mudanças

git stash list - listar os arquivos que estão no stash

git stash pop - voltar o arquivo para seu código principal

git stash clear - limpar a stash

git tag (1.0) - criar uma tag para o commit que voce esta

git tag (1.0) -m "release 1.0" - tag com mensagem

git tag -a "0.1.beta" -m "release 0.1.beta" <numero do commit que quer colocar a tag> - colocar tag em um commit anterior

git tag - listar as tags q tem

git tag -d (1.0) - deletar a tag

git push —delete origin 0.1 - deletar a tag do servidor (github)

git push origin main —follow-tags - subir para o servidor os commit junto com suas tags anotadas

git show <id commit> - mostra o que foi feito dentro daquele commit

git reset <nome arquivo> - volta um estado do arquivo se ja estava com git add e usar git reset ira voltar para antes do git add (REVERTE O GIT ADD)

git reset <id do commit> - volta para a branch escolhida

git reset <id commit> —soft - volta para o commit escolhido e retorna os arquivos do commit para o git add (index/stage)

—mixed - volta para commit escolhido e retorna os arquivos para antes do git add (workspace)

—hard - deleta o commit escolhido e todos arquivos nela

git push origin HEAD --force - depois do reset pra atualizar o github

git revert <id commit> - reverte o commit criando um novo commit tirando o que tinha na branch antiga

git clean -n

git clean -f - remover arquivos untracked antes do git add

git clean -df - remover pastas

git rm <nome do arquivo> - remove arquivos ja conhecidos pelo git

git rm <nome do arquivo> -r - remover pastas

git rm <nome do arquivo> —cached - destrackear um arquivo ja conhecido pelo git

git checkout <nome do arquivo> - desfaz as mudanças do arquivo quando estiver no workspace (unstaged area)

git checkout <id commit> ou <tag> - ver como o seu código era naquela versão ir para aquele commit

git checkout <nome da branch> - retorna para a branch escolhida ex head:main

git checkout -b <nome da nova branch> - se retornou para uma versao antiga e quer fazer dela uma branch para resolver algum bug e depois colocar de volta a main

git branch - ver em qual branch se encontra

git merge <nome da branch> - QUANDO ESTIVER NA MAIN e quiser mover uma branch para ela

REBASE INTERATIVO

git rebase -i HEAD~<numero de commits que deseja modificar na historia>

git reflog - mostra tudo que foi feito no projeto

CLONAR REPOSITORIO GITHUB

git clone (link do repositorio pego no github)

git pull - puxar atualizacoes para seu projeto que foram feitas direto no github

GITFLOW

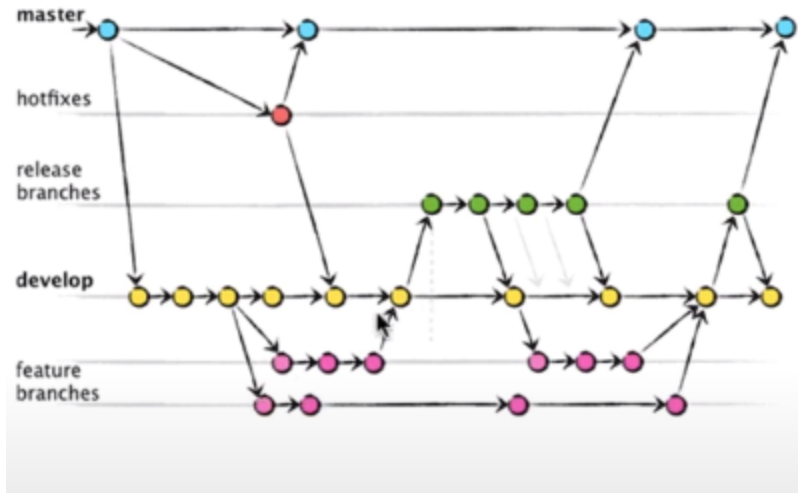
<https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>

MASTER - somente para lançar a versão do produto/feature

DEVELOP - nela vai ser desenvolvida o fluxo de trabalho criando features a partir dela

FEATURE/... - local onde vai ser desenvolvida a funcionalidade e depois mandada para a develop

RELEASE - onde sera entregue para o QA testar ou para o cliente ver se é oq ele quer mesmo depois de validada sera mandada a versão para lançar na master



git flow init - iniciar repositório com git flow nele

git flow feature start <nome da funcionalidade> - começar feature

git flow feature finish <nome da funcionalidade> - finalizar feature mandar para a develop

git flow release start <numero da release e nome> - começar release

git flow release finish <numero da release e nome> - lançar release para main

git flow hotfix start <numero ou nome do hotfix> - iniciar hotfix

git flow hotfix finish <numero ou nome do hotfix> - finalizar hotfix