Git and Github

Comandos:

Comandos para se cadastrar no git

git config –local user. name “seu nome” (Responsável por cadastrar seu nome)
git config –local user.email “seu email” (Responsável por cadastrar seu email)
git config user.name (Responsável por conferir o nome cadastrado)
git config user.email (Responsável por conferir o email cadastrado)

Comando para iniciar o git

git init (Responsável por iniciar o git em um arquivo/pasta)

Comando para adicionar arquivo no git

git add (nome do arquivo) (adiciona o arquivo que você colocou o nome)
git add . (adiciona todos os arquivos de uma vez)

Comando para fazer commit

git commit -m “descrição do commit” (Faz o commit)

Comando para ver os status

git status (exibe as condições do diretório)

Comandos para enviar para o GitHub

git remote add origin (URL do repositório)
git branch -M main
git push -u origin main

obs: depois que se coloca o “-u” os próximos comandos só precisa escrever git push

Comandos para ver os históricos dos commits

git log (exibe os commits feitos) - aperte “Q” para sair do git log
git log–oneline (exibe os commits feitos resumidos)
git -p (exibe o histórico de commits mostrando o que foi alterado)
git log–graph (exibe o histórico em forma de gráfico de linhas)

Comando para receber alteração direto do GitHub

git pull (Traz os arquivos que estão no GitHub)

Comando para clonar repositório

git clone (URL do repositório) (clona um repositório já existente)

Comando para desfazer uma modificação

git checkout (nome do arquivo)

Branch

Comando para verificar quantas branches tem no projeto

git branch (exibe as branches)

Comando para criar uma branch

git branch(nome da branch) (cria uma nova branch)

Comando para mudar de branch

git checkout (nome da branch) (muda para outra branch)
git checkout -b (nome da branch não existente) (cria uma branch e já muda para ela)

Comando para juntar os commits de duas branches

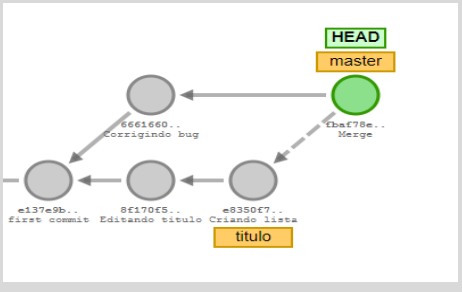
obs: vá na branch que você quer juntar os commits primeiro

git merge (nome da outra branch que você quer os commits) (cria um novo commit juntando branches)
git rebase (mesmo objetivo do git merge)

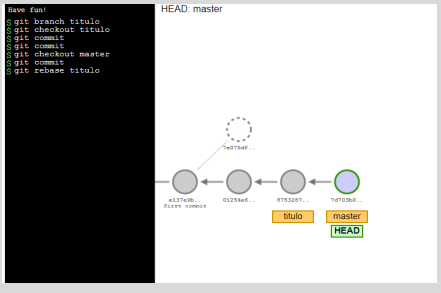
diferença entre git merge e git rebase:
git merge
vai criar um commit novo com a junção dos commits das branches

git rebase
não vai criar um commit com a junção dos commits da branches, ele vai simplesmente transferir o último commit da branch para a outra branch

git merge



git rebase



Comandos para desfazer alterações

para desfazer modificação antes de usar o git add
git restore (nome do arquivo)
git checkout –(nome do arquivo)

para desfazer modificação depois de usar o git add
git restore –staged (nome do arquivo)

para desfazer um commit feito
git revert a1693e5

Stash

O stash serve para guardar uma alteração para usá-la depois sem precisar fazer um commit.

Comando para criar um stash

git stash (cria um stash)

Comando para ver os stash que possuímos

git stash list (mostra os stash que temos)

Comando para usar o stash

git stash apply 0 (executa o stash chamamos)

obs: o número dos stashes começam por zero.

Comando para apagar os stashes existentes

git stash drop (apaga o último stash)

Comando para executar um stash e apagá lo em seguida

git stash pop (executa e apaga o stash)

Comando para voltar em commits antigos

git checkout a1693e5 (volta ao commit antigo)

obs: qualquer alteração feita no commit antigo não vai ser salvo, para que seja salva tem que se criar uma branch naquele commit.

Comando para ver as alterações feitas de um commit até o outro

git diff a1693e5..fad42fc (comando que mostra a diferença entre os commits)

Tags

As tags vão servir para organizar nosso repositório, pois assim fica mais fácil de saber qual é aquela versão do código

Comando para criar uma tag

git tag -a v0.1.0 -m “primeira versão (BETA)”

Comando que mostra as tags existentes

git tag

Comando para mandar a tag ao GitHub

git push origin v0.1.0

Links úteis

https://git-school.github.io/visualizing-git/#free (visualizar como se comporta os commits e as branches)

Documentação feita por Levi Martins de Andrade - https://github.com/Levi-Martins