O que é um branch?

* branch é a forma que o git separa as versões dos projetos
* quando um projeto é criado ele inicia na branch master / main, estamos trabalhando nela até esse ponto do curso
* Geralmente cada nova feature de um projeto fica em um branch separado / novo
* após a finalização das alterações os branchs são unidos para ter o código-fonte final

Criando e visualizando os branches

* Para visualizar os branches disponíveis basta digitar git branch
* Para criar uma branch você precisa utilizar o comando git branch <nome>
* Essas duas operações são muito utilizadas no dia a dia do dev

Deletando branches

* Podemos deletar um branch com a flag –d ou –delete
* Não é comum deletar um branch, normalmente guardamos o histórico do trabalho
* Geralmente se usa o delete quando o branch foi criado errado

Mudando de branch

* Para mudar para outra branch utilizamos git checkout <nome da branch>
* Podemos mudar para outro branch utilizando o comando git checkout -b <nome>
* Esse código cria um novo branch e já muda para ele caso não exista
* Este comando também é utilizado para dispensar mudanças de um arquivo
* Alterando o branch podemos levar alterações que não foram commitadas junto, tome cuidado!
* Sempre commitar antes de mudar de branch

Unindo branches

* O código de dois branches distintos pode ser unido pelo comando git merge <nome>
* Outro comando para a lista dos mais utilizados
* Normalmente é por meio dele que recebemos as atualizações de outros devs
* Nunca utilizar o merge estando na master/main,

Stash

* Podemos salvar as modificações atuais para prosseguir com uma outra abordagem de solução e não perder o código
* O comando para esta ação é o git stash
* Após o comando o branch será resetado para a sua versão de acordo com o repo

Recuperando Stash

* Podemos verificar as stashs criadas pelo comando git stash list
* git stash apply <0,1,2,3…> aplica a stash novamente ao branch
* E também podemos recuperar a stach com o comando git stash <nome>
* Desta maneira podemos continuar de onde paramos com os arquivos adicionados a stash

Removendo a Stash

* Para limpar totalmente as stash de um branch podemos utilizar o comando git stash clear
* Caso seja necessário deletar uma stash específica podemos utilizar git stash drop <nome>

Utilizando tags

* Podemos criar tags nos branches por meio do comando git tag -a <nome> -m “<msg>”
* A tag é diferente da stash, serve como um checkpoint de um branch
* É utilizada para demarcar estágios do desenvolvimento de algum recurso
* cria um ponto de recuperação para avançar ou retroceder

Verificando e alterando tags

* Podemos verificar as tags com o comando git tag
* Podemos verificar uma tag com o comando git show <nome>
* Podemos trocar de tags com o comando git checkout <nome>
* Desta maneira podemos retroceder ou avançar em checkpoints de um branch

Enviando e compartilhando tags

* As tags podem ser enviadas para o repositório de código, sendo compartilhada entre os devs
* O comando é git push origin <nome\_da\_tag>
* Ou se você quiser enviar todas as tags git push origin –tags