

Git

Comandos

git init – cria um repositório git vazio

git status – visualiza as alterações que tiveram

git add “nome do arquivo” – adiciona um arquivo

git add . – adiciona todos os arquivos de uma vez

git commit – cria uma versão do código

- -m “” envia uma mensagem junto

git push – envia a versão do código para a nuvem (é necessário cria-lo anteriormente, como no github, por exemplo)

- git remote add origin link – define para onde o código será enviado (dará um erro se não souber para qual branch enviar, inicialmente é enviado para a master)

git reflog – mostra quais versões foram adicionadas (a versão mais atual fica no topo)

git reset --hard id – retorna para um commit específico

git branch - verifica quais branches estão criadas

- branch que apresentar o * é a Branch em que estaremos trabalhando no momento

git branch nome – cria uma nova branch

git checkout nome – troca para a branch que quer usar

git merge nome – mergear o código, puxar de uma branch para outra Branch



- o **ATENÇÃO!!!!**
- o ANTES de realizar o merge é necessário trazer todas as atualizações existentes no servidor para a sua máquina local

git pull – traz as atualizações que existem no servidor

Passo a passo

1. Git pull da Branch principal
2. Gerar uma nova branch a partir da Branch principal
3. Trabalhar e adicionar novas funcionalidades na nova Branch que criou
4. Finalizar o trabalho na Branch temporária
5. Git checkout na Branch principal
6. Git pull
7. Mergiar o Código da Branch temporária com a Branch principal (depois de testar)
8. Git push da Branch principal

Nomenclaturas

commits: versões do código

branches: caminhos diferentes

- A, B, C (caminhos)
 - A: versão 1, versão 2, versão 3 (principal, apenas código que funcionam)
 - B: versão 4, versão 5 (branches paralelas)
 - C: versão 6, versão 7 (branches paralelas)

merges: passar as alterações entre branches