

* GIT E GITHUB - COMANDOS AVANÇADOS *

- HISTÓRICO DE COMMITS COM GIT LOG E GIT DIFF

~ git log --oneline

Mostra todos os commits

-p

Mostra commits com mais detalhes

--graph

Mostra linha do tempo dos commits

--format "..."

* DETALHAR (%h, %am, %an, %n)

--help

AJUDA SOBRE O COMANDO

SAIR DO LOG: TECLA Q

* EXISTE MAIS PLACE HOLDERS, EXEMPLO:

%H (HASH COMPLETO)

%e (EMAIL DO AUTOR)

~ git diff HASH .. HASH

Mostra as alterações comparando

os commits, mas podendo limitar

os espaços com commits usando

valores HASH.

↳ CHAVE COMMIT

- TRABALHANDO COM BRANCHS

~ git branch NOME BRANCH

OR

~ git switch -c NOME BRANCH

~ git checkout NOME BRANCH

↳ OS DOIS COMANDOS

↳ CRIA E ALTERA O BRANCH

JUNTOS (CRIA E ALTERA)

~ git branch -d NOME BRANCH

↳ PARA EXCLUIR UMA BRANCH

~ git rebase main

BRANCH C/ A VERSÃO
+ ATUALIZADA DA MAIN

- ~ git merge **NOME BRANCH** UNIR BRANCH COM A ~~MATE~~ MAIN
- ~ git push origin : **NOME BRANCH** REMOVE BRANCH

- TRABALHANDO COM VERSOES

→ SERVER PI SALVAR SEU PROGRESSO PI SER FINALIZADO DEPOIS

- ~ git stash GUARDAR ESTADO
- ~ git stash pop PEGAR ESTADO
- ~ git stash list LISTAR ESTADOS
- ~ git stash clear LIMPAR
- ~ git push -m "memoregem" COMMIT
- ~ git restore . RESTAURAR CODIGO
- NOME ARQUIVO**

- GERANDO ENTREGAS TAGS = VERSOES NO CODIGO

→ CRIAR E COMMITAR TAG

- ~ git tag -a **NOME VERSAO** -m "COMMIT"
- ~ git tag -d **NOME VERSAO** REMOVER TAG
- ~ git push origin --tags ou **NOME VERSAO** SUBIR A TAG

- CONFERINDO ALTERAÇÕES

- ~ git log nome **NOME ARQUIVO**
- ~ git show **HASH COMMIT**
(OPCIONAL)
- ~ git log **NOME BRANCH** HASH DA BRANCH
- ~ git cherry-pick hash HASH DE COMMIT DE OUTRA BRANCH