

* Dêuse bem

→ Git IGNORE

↳ ARQUIVOS, EXTENSÕES OU DIRETÓRIOS sejam desconsiderados no Repositório

↳ Obs. tem que estar na pasta onde tem arquivo oculto .git, onde foi inicializado

↳ Criar no terminal: touch .gitignore

↳ J. o arquivo texto p/ colocar dentro e ser ignorado

Aula 2 - 14/12/21

revisão

Atividade

↳ Serviço Repositório

↳ token

GitHub

↳ Serviço em nuvem desenvolvedores, confirme aula 1

→ Git Add .
→ Git commit -m " " } modificações

↳ Criação Repositório: compatibilidade

↳ Nome associado finalidade do Projeto, representado p/ diretório

1º Rep. clonar o nome d. diretório se é nome do Rep

2º Git. init na pasta no dir. p/ o documento p/ o Rep. ou vice versa

↳ Obs. Pega a mesma ideia do nome antes de criar algo de

vel e que é relevante

- No Git mostram 2 opções comandos, porém quando é feita o Dir, já tá feito a 1ª opção
- Então já feito p/ 2º p/ sincronizar com URL, até então tudo era de terminal
- Por Git init

→ Git Remote Add ORIGIN URL → Rep Remote

Atenção
nos parâmetros

→ Git branch -m main

↳ Ramificação Projeto (clonar do mesmo projeto / alterar hist. dados) versões
sem conteúdo

~~Git commit~~

→ Git push -u origin main

- ↳ Depois da modificação sera: git add -
ri commit
git push

↳ selection a pasta dir no get bash

C₃ volta a ter o director, volta ser REPOSICIONADO

↳ Pensa logo em Git pull

↳ P/S INCRONIZAR

↳ Avaliar: Esforço e importância ~~do~~ valor da mod. f. cap.
do conteúdo algo que impacta

valor da mod. Ficação no Repositório

dme
 ↳ markdown - Representação HTML
 ↳ serve de Portfolio para PROSGETO

УКОН К

- minor crystal

↳ Problema - Agilizar o desenvolvimento / organizar a interação de 'Nóds' e valores entre eles em um mesmo contexto

→ Branch

Branch → meu uso

- ↳ Repositório
- Criar - Git branch nome do branch
- listar
 - Git branch
- mudar de branch
 - Git checkout nomeBranch
 - Diferença - add commit
- Git push

↳ Copiar, criar branch

→ Dois trajetórias de modificação, integração

- criar uma réplica
- copiar de segurança / versões de volta do próprio repositório
 - ciclo de vida } branch
 - e nomes }

→ P/mudar/puxar

- ↳ Vou p/branch principal main
 - Git merge nome branch que quero puxar
 - Git push ou pull anteriormente

Obs → No dia a dia existem pessoas responsáveis pelo merge, p/ver se não altera a segurança

30 min
até 1

→ merge e Pull Request contendo diferenças entre branches

↳ consolidação versão final organizada

≠ Permissões

- Ex
- 1º merge pelo dev (selec. da tela)
 - 2º executar merge da que contém mod. checkout

Pull

- ↳ Espelho do repositório, do qual eu faço parte do rep. e crio minha versão.
- ↳ É uma branch, a parte do rep. de alguém eu puxo p/min (fork)
- ↳ Aquele a quem puxei permissão e a qual é da outra pessoa

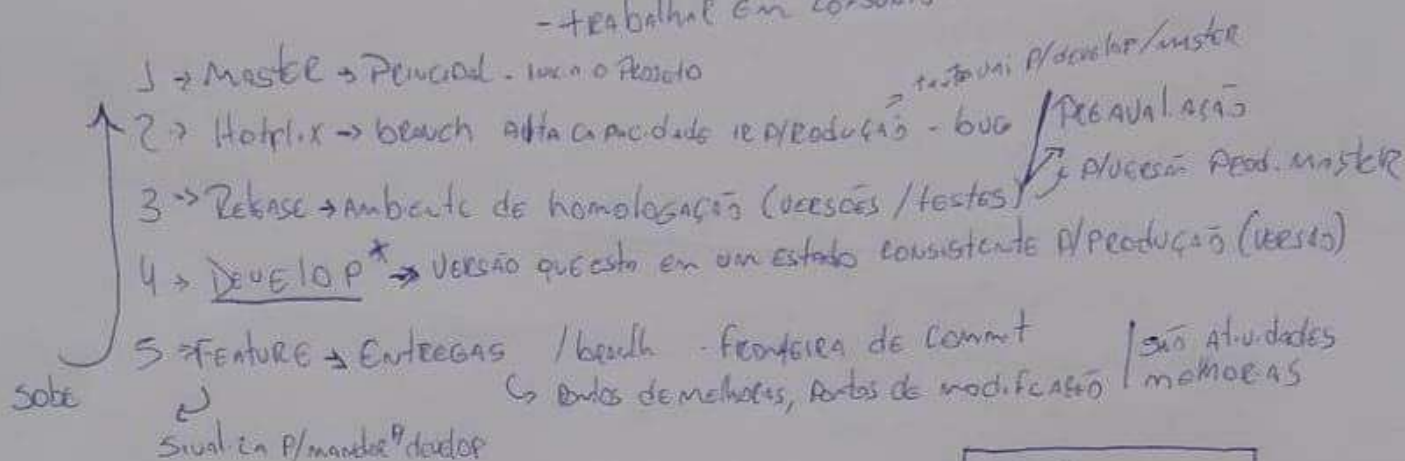
Git Flow

↳ ideia baseada no fluxo de trabalho baseado em branches bem definidas

Ramificações

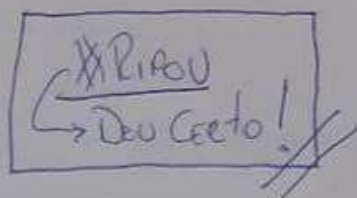
↳ fronteiras as melhorias

Obs: Desenvolvedor NUNCA acessa a master, fronteira do G e Develop
- trabalhe em conjunto -



Obs: master e Develop - São eternas

↳ + sofre alterações



Git Flow

↳ Segurança e integridade do código