

→ Para o Ser Programador

- ↳ SAVAR uma dor - automatizar, ganhar bem, não gostar do que faz
- Compreender fundamento/conceito - ele quer conhecimento
- Bom desenvolvedor + um excelente usuário
- Console e Git melhor amigo
 - ↳ autossave em código, se estruturar como um ORÇAMENTO todo código apresentado todo projeto desenvolvido, não sozinha de um resultado

2 Ferramentas

→ Git e GIt

- ↳ Plataforma de armazenamento e outros
- ↳ comando transferência de dados
- Gerenciamento de

→ Controle VERSÃO / Fonte

- ↳ GERENCIAR alterações código fonte
- ↳ GARANTIR que seja armazenado forma segura, atualizado, consistente todos envolvidos, mantendo históricos distribuído
- ↳ Não é só salvar e manter histórico, vai + além

• Git CV &

↳ Distribuído

- ↳ ORGANIZAR forma segura e histórico - Repos local / Público
- Integridade, alto desempenho, SEGURANÇA, voltar no tempo (Desfazer)
- ↳ GERENCIAR Rep. local e SINCRONIZAR Rep. local e Rep. remoto
- PREMISSAS: P/NOSSA APLICAÇÃO

Obs: Saber muito bem sobre:

↳ comandos Atualização código

||

Envio código Repositório - MERGE (Gerenciar conflitos)

→ Git init - Cria o Repositório (local) auditar / backup

→ Git add -

→ Git commit - m " " → Finalidade do commit - Informar

↳ Push → envio, commit/merge arquivos local / Repositório

→ Criar Repositório local

↳ ~~Dir~~ C:/DEV/WS/

↳ Raiz do diretório onde vai ser a base de diretório onde vai estar sincronizado com Rep. Público - Github

Obs: Nome associado ao Rep. Público



→ Branch

Branch é meu ramo

- ↳ Poderia ser feita do mesmo arquivo
- Create - Git branch nome
- listar

- Git branch
- mudar de branch
- Git checkout nomeBranch
- Delete - add com

→ Git Push

↳ Copia, lá fora branch

→ Dois trajetórias de modificação, iteração

cria uma réplica

Copia de segurança / versões dentro do próprio repositório

- ciclo de vida } branch
- e nomes }

→ P/ mudar / puxar

↳ ou p/ branch principal

- ↳ Git merge nome branch que quero puxar
- ↳ Git push ou pull automaticamente

Obs → No dia a dia existem pessoas responsáveis pelo merge, p/ ou se um ataca a segurança

30 min
atc

→ merge e Pull Request contendo diferenças entre branches

↳ consolidação versão final

≠ Permissões

Ex

- 1º merge principal (selec. a tela) receptor
- 2º execute merge da que contenha mod. checkout

Pull

- ↳ Escolho do repositório, do qual eu quero puxar
- Parte do rep. e cria minha versão.
- É uma branch
- ↳ que eu quero q
- ter permissão e
- a ver é da outra pessoa

→ Parte do rep. de alguém eu puxo p/min (fork)

* Dêuse bem

→ Git IGNORE

↳ ARQUIVOS, EXTENSÕES OU DIRETÓRIOS sejam desconsiderados no Repositório

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

↳ Criar no terminal: touch .gitignore

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

Aula 2 - 14/12/21

resumo

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 repositório p/ o Rep. ou seja

↳ 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 forma local
- 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
mesmo conteúdo

~~Git push~~

→ Git push -u origin main