

Git e GitHub



Dados do github:

luppocode@outlook.com
Caio123.f

Git

° Para que serve?

VCS - Version Control System

↳ Quando há muitas pessoas no projeto, todos tem acesso ao código atualizado.

↳ Só permite enviar uma alteração, caso a sua seja a última. -- Controle de versões

Baixar git no site oficial!

↳ Seguir os padrões (para o curso)

O gitbash fornece comandos comuns de terminal e cmd.

Primeiros Comandos

git init

↳ Inicia o repositório na pasta local

git status

↳ Mostra a situação da pasta atual.

Quem é você?

git config --local ||

user.name (name)

user.email (email)

-

Para logar na sua conta.

Salvar arquivo

git add <arch> ou <.>

git rm --remove

git commit -m <mensagem>

↳ Salva as alterações com msg.

git log

↳ mostra commits

-p- mostra alterações

Git status

↳ HEAD -- Estado atual do código (onde está)

↳ Working tree -- local real dos arquivos

↳ index -- Onde os commits ficam

devhints.io/git-log

Ignorando arquivos

`git ignore` - arquivo que contém nome de arquivos para **não** monitorar.

Nunca se adicione um código que não funciona. "Quando" é relativo. Mas é interessante a cada significativa.

Compartilhar Código

Criar uma pasta servidor

```
git init --bare
```

↳ diz que ele é "cru"

```
git remote
```

↳ lista repositório que conhece

```
git remote add <pasta>
```

```
git push local master -o branch
```

↳ server

GitHub

Tem que gerar um token pra subir o código:-
go. ↳ faz isso em settings

Branches

Separa linhas de desenvolvimento

↳ master
↳ padrão

git branch <novo> -> cria branch

git checkout <título> -> troca branch
↳ -b cria lista

Unindo trabalhos

git merge <branch pra se unir>

git rebase título -> única linha

Resolvendo Conflito

Só remover informação errada e salvar arquivo
non

Ctrl + Z

↳ Quando se faz muitas alterações

-º git status

↳ Se não comitou, pode dar `git checkout -- <file>`
↳ apaga alterações feitas

↳ Se adicionou

↳ `git reset HEAD <file>`

↳ estado de trabalho

↳ Se deu `commit`

↳ `git log` - pega o Hash do commit

↳ `git revert <hash>`

↳ Vai gerar um commit dizendo
que foi revertido

Git Stash

- > Guarda informações em um local temporário os dados modificados.
- o Para trazer de volta
 - stash list
 - ↳ mostra os índice das modificações
 - stash apply <índice>
 - ↳ aplica o que foi salvo
 - stash drop
 - ↳ remove da lista
 - stash pop
 - ↳ deleta e aplica
- o É útil quando precisar fazer outra tarefa

Viajando no tempo

`git log --oneline` → verificar commit

Para viajar entre os commits, ou seja, retornar o projeto para o estado que estava naquele momento, usa-se:

`git checkout <hash>`

↳ Te coloca em um HEAD destacado projeto

Para salvar alterações do commit desanexado:

→ criar um novo branch e modificar lá

Vendo Alterações

`git diff <hash>..<hash>`

↳ Ver todas alterações em uma faixa de tempo.

`git diff`

↳ Mostra o que foi feito até agora desde o último commit

Tags e Releases

o Para marcar um commit (ex: versão ou outra coisa)

Tag → marca um ponto fixo na aplicação

↳ ñ muda mais

↳ `git tag -a v0.1.0 -m "primeira beta"`

↳ cria tag

↳ `git push local <tag>`

Isso deixa o projeto baixável para quem quiser usar

Por: Caio Lupo Ribeiro