

Git and Github

para poetas e hobbysta

By Wesley cantarino

Universidade Federal Juiz de fora

February 23, 2020



Visão geral

1 Introdução

2 Primeiros passos

3 Main

4 Referências



Introdução



Apresentação

- 1 O criador do Git (e também do linux) recomenda todo usuário começar pelos comandos básicos e ir avançando gradualmente com o tempo.

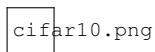


Figure: Criador do Git

- 2 Já Bob leu no stater overflow e acha que esta fluente.

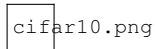


Figure: Bob não usa Git

Não sejam um Bob.



Apresentação

Afinal .. o que é Git?

- 1 é um sistema de controle de versões distribuído, usado principalmente no desenvolvimento de software, mas pode ser usado para registrar o histórico de edições de qualquer tipo de arquivo.

Afinal .. o que é GitHub?

- 1 GitHub é uma plataforma de hospedagem de código-fonte com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo

Lembre que Git != Github mas juntos formam um casal perfeito <3



Primeiros passos



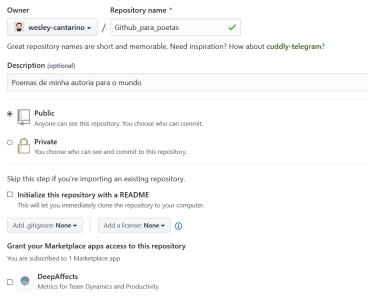
Init

- 1 Após baixar e instalar Git e criar sua conta no Github.
- 2 Neste curso adotaremos Git para Windows.
- 3 Existe concorrentes Github assim como também do git mas todos seguem a mesma filosofia.



Init

- 1 Open github.com
- 2 Sign in or Sign up (in para criar conta e up para logar).
- 3 No clique no icone de "mais" no canto superior direito e vá em "new repository"



Owner Repository name *

wesley-cantarino / Github_para_poetas ✓

Great repository names are short and memorable. Need inspiration? How about [cuddly-telegram?](#)

Description (optional)

Poemas de minha autoria para o mundo

☒ Public
Anyone can see this repository. You choose who can commit.

☐ Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add gitignore: **None** Add a license: **None** ⓘ

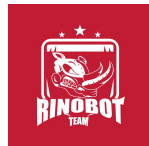
Grant your Marketplace apps access to this repository
You are subscribed to 1 Marketplace app.

☐ DeepAffects
Metrics for Team Dynamics and Productivity

4

Figure: .

- 5 Click em criar repository e vualá.
- 6 Copie o link https.



Clone

- 1 Em uma pasta qualquer de sua preferência.
- 2 Click com botão direito e abra "Git bash here"
- 3 Nesta janela é onde a mágia do Git acontece. É nela que vamos trabalhar daqui em diante e todo sempre.
- 4 Digitar "git clone /seu https/"
Por padrão o git cria uma pasta com mesmo nome do seu repositório mas você pode mudar digitando o nome da pasta logo após colar seu https.



Main



Principais tópicos recorrentes

1 Criar arquivo de texto (bloco de notas) chamado "poemas.txt"

2 `figuras/poema.png`

Figure: Pode criar o poema do seu jeito.



Primeiro commit

No git bash here, digitar:

- 1 `git status`
O arquivo poemas.txt aparecerá em vermelho pois ele tem alterações deste a última versão salva no git.
- 2 `git add .`
Adicionar/rastrear todos os arquivos.
- 3 `git commit -m "titulo do commit" -m "mensagem opcional"`

Melhor forma de entender como git funciona e trata seus arquivos é imaginar todo seu projeto como uma um jogo que vai sendo contruida de acordo com seu trabalho. Quando damos commit nós salvamos aquele ponto da história.

- 1 `git push origin master`
Para enviar no github onde master é sua branch.
Se olharmos no Github nossas alterações vão estar lá.

É o ciclo sem fim... rei leão.



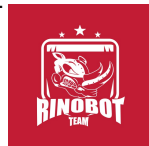
Branch



figuras/poema.png

Figure: ramificações

- Quando queremos desenvolver algo novo ou resolver um bug difícil podemos usar o conceito de branch para nos ajudar. Branch nada mais são do que ramificações de um(s) arquivo(s) principal (principais).
- Exemplo do uso de branch é pensar no desenvolvimento de um aplicativo mobile onde sem tem três branch. Uma é a master onde tem a aplicação rodando na sua última versão e estável, branch desenvolvimento que é onde os programadores testam novos recursos e a branch correção onde os programadores tentam corrigir os bugs e mal funcionamentos reportados na sua versão estável (master).



Branch

- Criando nova branch
git checkout -b "nome do branch"
- Faça alguma alteração no texto e de commit.
Lembre-se que agora o comando git push origin não é mais na master e sim no nome da sua branch.



Merge

- Você como desenvolvedor, após criar uma branch específica para resolver um bug. Seu trabalho agora é juntar a sua solução na versão estável do seu produto. com isso, você deve usar o comando `git merge <branch name>`



Comandos recorrentes

vou colocar ainda, ou não ... sei lá



Referências



Sites

vou colocar ainda, ou não ... sei lá



livros

- 1 Pro git - livro gratuito e completo.
<https://git-scm.com/book/en/v2>



Obrigado

1 contatos
github.com/wesley-cantarino

