

um apanhado de dicas de como usar Git (bem, ou um pouco melhor)

Sara Mortara, Andrea Sanchez-Tapia & Diogo Rocha

16 de Junho de 2019

sistema de controle de versão **livre** e de **código aberto**

- pegada
- alta performance
- múltiplos fluxos de trabalho
- *branching merging*
- distribuição - *clone*
- área intermediária - *staging area*

o que é GitHub

serviço de hospedagem na internet para controle de versão e colaboração usando **git**

por onde começar

- guia básico de **Git** no **GitHub**:
<https://guides.github.com/activities/hello-world/>
- como usar o **Git** no **RStudio**:
<https://pagepiccinini.com/r-course/lesson-0-introduction-and-set-up/>

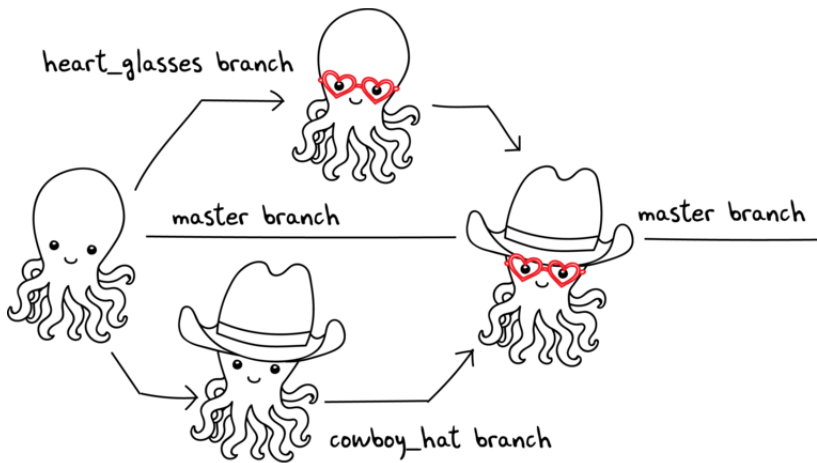
fluxo de trabalho *feature*



em outras palavras

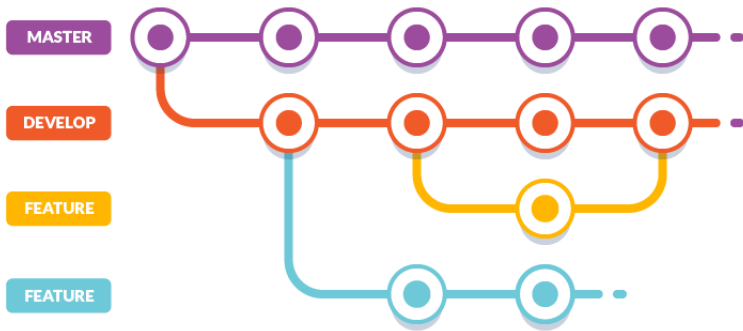
merge it back into the original project.

Here's a simplified version of that workflow:



Each feature is added back into master individually. So if the glasses are finished before the cowboy hat, no problem: those can be

fluxo de trabalho *gitflow*



o exemplo do modleR

https://github.com/Model-R/modelr_pkg

Model-R / **modelr_pkg** Watch 6 Star 5 Fork 1

Code Issues 9 Pull requests 0 Projects 0 Wiki Security Insights Settings

An ecological niche model workflow based on dismo() [Edit](#)

[Manage topics](#)

279 commits 4 branches 0 releases 6 contributors

Branch: master New pull request Create new file Upload files Find File Clone or download

Switch branches/tags

Find or create a branch...

Branches Tags

- andrea_tests
- documentation
- ✓ master
- pROC
- .gitignore

centroid, mindist

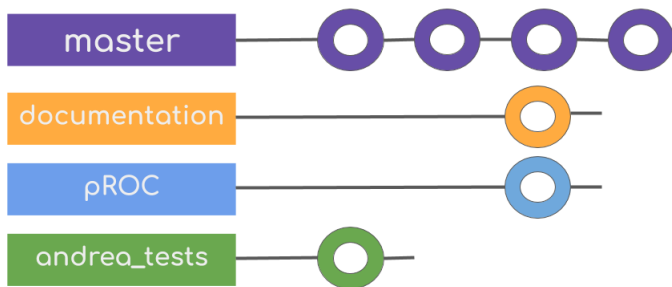
DESCRIPTION

NAMESPACE

README.md

modelr_pkg/tree/master

fluxo de trabalho do pacote modleR



comandos básicos

Para clonar um repositório já existente:

```
git clone https://github.com/Model-R/modelr_pkg.git
```

Para criar um repositório git localmente:

```
git init
```

os meus **quatro** comandos básicos

1. Para atualizar o repo localmente:

```
git pull origin master
```

2. Para adicionar um arquivo com mudanças:

```
git add filename
```

3. O *commit*:

```
git commit -m "uma mensagem informativa para você e  
coleguinhxs"
```

4. Enviando para o repositório:

```
git push origin master
```

boas práticas

boas práticas

- 1 descrever no ***commit*** a mudança

boas práticas

- 1 descrever no ***commit*** a mudança
- 2 trabalhar uma tarefa em cada ***branch***

boas práticas

- 1 descrever no ***commit*** a mudança
- 2 trabalhar uma tarefa em cada ***branch***
- 3 `.gitignore`

boas práticas

- 1 descrever no ***commit*** a mudança
- 2 trabalhar uma tarefa em cada ***branch***
- 3 .gitignore
- 4 usar modo ***diff*** para acompanhar mudanças no código

boas práticas

- 1 descrever no *commit* a mudança
- 2 trabalhar uma tarefa em cada *branch*
- 3 .gitignore
- 4 usar modo *diff* para acompanhar mudanças no código
- 5 reportar erros no código em *issues*