

Desenvolvimento Aberto



Fluxo de trabalho distribuído e Comunidades de Software

Igor dos Santos Montagner (igorsm1@insper.edu.br)

Aula 01

- Objetivos e motivações
- Fizemos uma primeira tentativa de PR para se adicionar ao repo da disciplina

Hoje

Expositiva

- Comunidades de software
- Fluxo de trabalho distribuído: modelo de Pull Request
 - `revert`
 - `rebase`
 - `ammend`

Atividades

- Apresentação de projetos de software
- Revisar PR da aula anterior

Comunidades de software

Discussão

Pergunta: quais são os softwares que vocês mais utilizam no dia a dia?

Pergunta 2: quem é o principal desenvolvedor do projeto?

Exemplo: Kernel Linux



Fundação dá suporte ao desenvolvimento do Kernel Linux e à criação de um ecossistema de software de código aberto ao seu redor.

Apoio:

- Financeiro
- Propriedade intelectual
- Infraestrutura
- Serviços e treinamento

Exemplo: Kernel Linux

Empresas que patrocinam desenvolvimento:

1. Intel (13.1%)
2. Indivíduos (8.2 %)
3. Red Hat (7.2%)
4. Linaro (5.6%)
5. Desconhecido (4.1%)
6. IBM (4.1%)
7. Consultores (3.3%)
8. Samsung (3.2%)
9. SUSE (3.0%)
10. Google (2.7%)

Fonte: "2017 Linux Kernel Development Report"

Exemplo: Ubuntu

CANONICAL

Empresa privada controla o desenvolvimento do Ubuntu. Obtem recursos via

- Doações
- Serviços de desenvolvimento e consultoria
- Treinamento

Exemplo: Gitlab



Empresa financiada até agora por VC (Venture Capital) em busca de lucro.

- Opção de planos grátis com software de código aberto
- Opção de planos corporativos com mais funcionalidades e **suporte**
- Instalações locais

Exemplo: Elementary



Distribuição linux comandada por uma empresa "minúscula" focada em experiência de usuário. Recursos são obtidos via

- Doações
- Compras na loja AppCenter

Mystery Donation Lets Elementary Hire Full-Time Staff

The Etsy of Linux distros has some good news to share



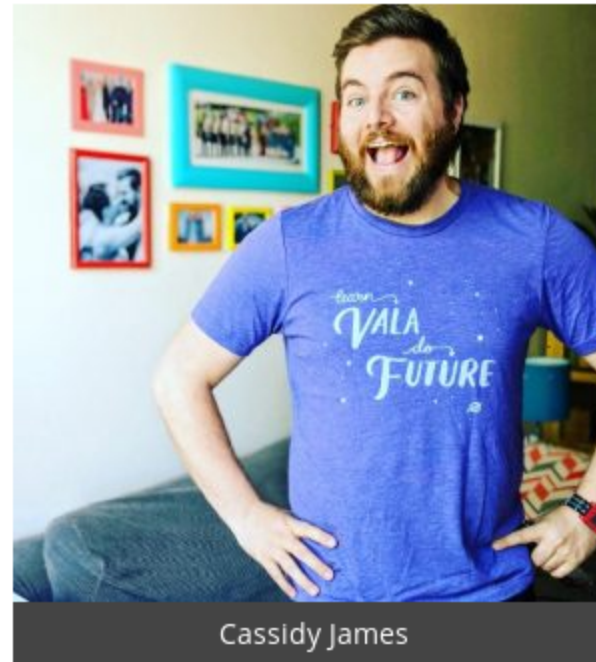
Joey Sneddon Updated Saturday

Community

The GNOME project recently received a \$1 million donation from [an anonymous benefactor](#) — now it seems the altruism is spreading to other open source projects.

The folks behind [elementary OS](#), a popular Ubuntu-based Linux distribution, have revealed they too recently received a “large private contribution” from an otherwise anonymous donor.

So large is the donation they say they’ve been able to hire one of their part-time volunteers full time!



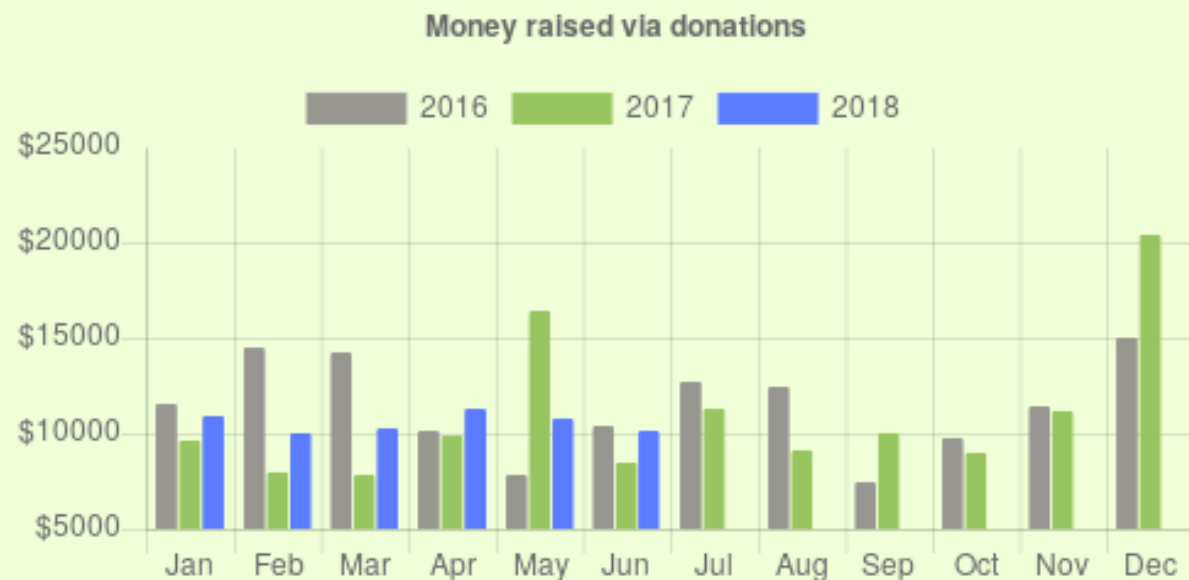
Cassidy James

Exemplo: Linux Mint



Distribuição linux cujo objetivo é que tudo funcione direto na instalação. É considerada boa para iniciantes.

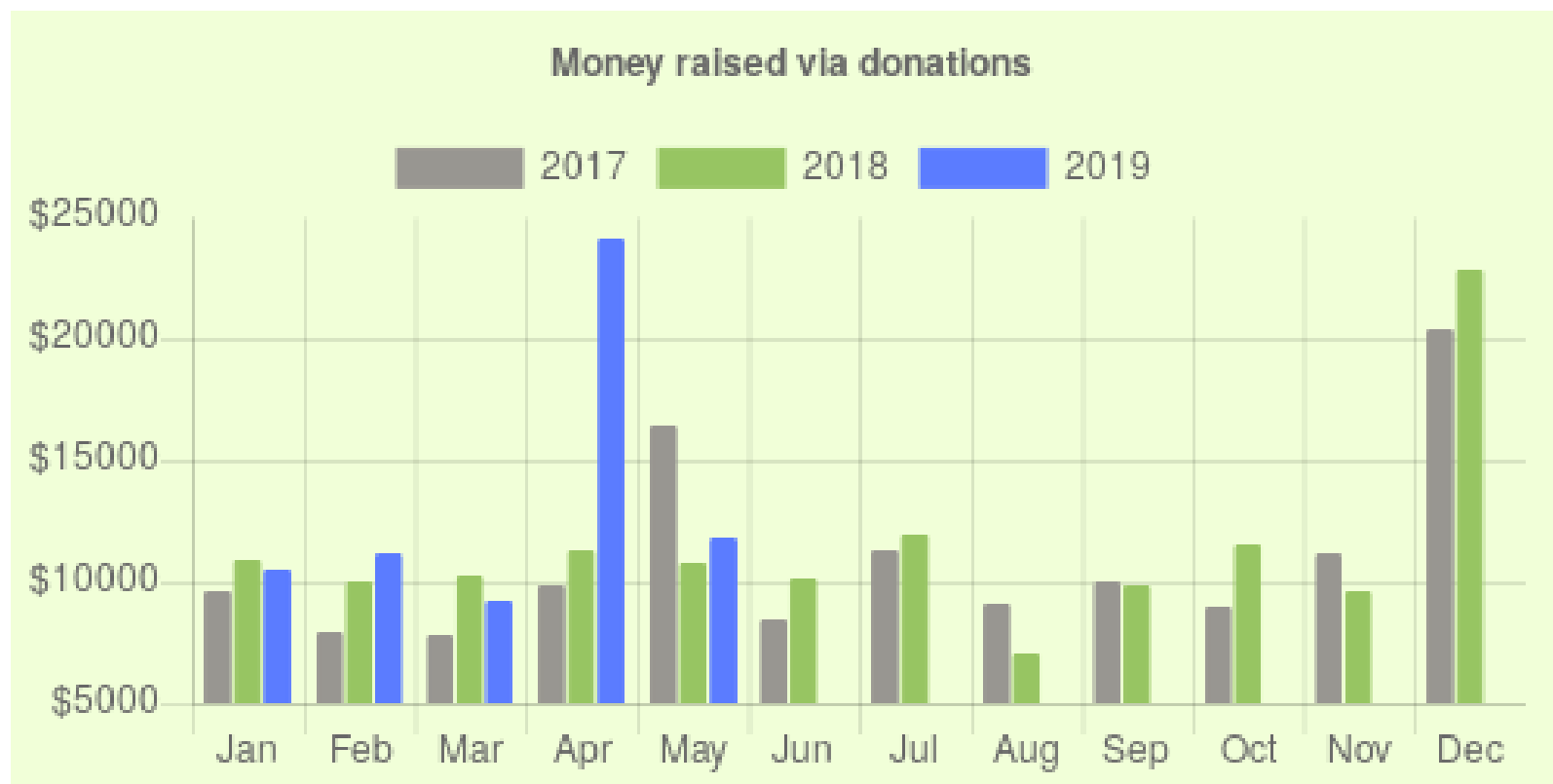
Donating to Linux Mint























Why donate money? Linux Mint is free of cost and doesn't generate any direct sort of income. It is funded by advertising, sponsoring and donations and although it is financially supported by its own community of users, its ambition is to compete with projects which are backed by large companies such as Microsoft, Apple, RedHat, Novell and Canonical.

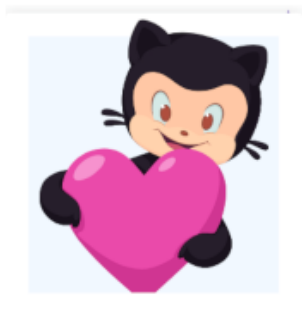
How is the money used? The money is used to finance the needs of the distribution, to pay for hosting, to advertise on other websites, to purchase equipment and to fund the work of the people behind it.

How much should you give? The donations we receive range from a single dollar to a couple of hundreds. The important thing for people is to give something, and for us it's to generate the income that will let us work and succeed on this project, month after month. If you want to help, and if you can do so, please make a donation to the project and encourage others to do so.



2019-07-29		Matei M.	\$16
2019-07-29		able heads (Jan S.)	\$163 (8th donation)
2019-07-29		Rolf H.	\$22
2019-07-29		Zaahir L.	\$26
2019-07-29		Lawrence K.	\$10
2019-07-29		Vincent B.	\$22
2019-07-29		施 中右	\$2
2019-07-29		Geoffrey B.	\$20
2019-07-29		Colin B.	\$50 (3rd donation)
2019-07-29		Vince M.	\$10 (2nd donation)
2019-07-29		David M.	\$10
2019-07-29		Josemi	\$11 (2nd donation)
2019-07-29		Dionysios K.	\$5
2019-07-29		Michael R.	\$4
2019-07-29		Ronny D.	\$5
2019-07-29		Gerald F.	\$40 (2nd donation)
2019-07-29		Ioannis P.	\$33
2019-07-29		Armin R.	\$30 (2nd donation)
2019-07-29		鈴木 和夫	\$10 (2nd donation)
2019-07-29		Jon L. L.	\$65 (5th donation)

Outras fontes



Diversas outras maneiras de conseguir suporte e financiamento

Razões para financiar (empresa)

- "Adquirir" especialistas em uma tecnologia
- Vender suporte ou consultoria
- Marketing
- Licença de software "obriga"

Nem sempre o que é financiado termina em um formato "aproveitável" por terceiros (ex: Android/Linux, TiVO)

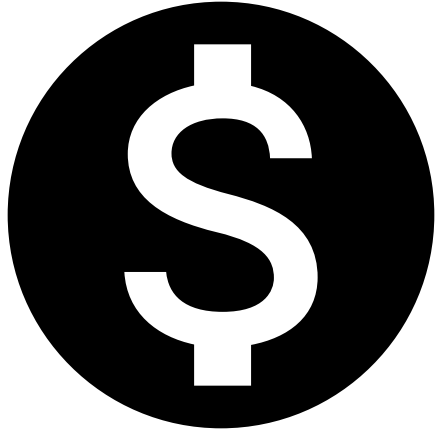
Razões para contribuir (indivíduo)

- Profissionais
 - Reputação / Currículo
 - Necessidade de negócio
- Pessoais
 - Altruísmo / Ideologia
 - Sensação de realização pessoal
 - Exercício de criatividade
 - Resolver um problema de seu interesse

Como começar a contribuir?



Atividade: Explorando o entorno de um projeto



- Pesquisar como as comunidades de alguns projetos estão organizadas
- Produzir uma apresentação resumindo suas descobertas

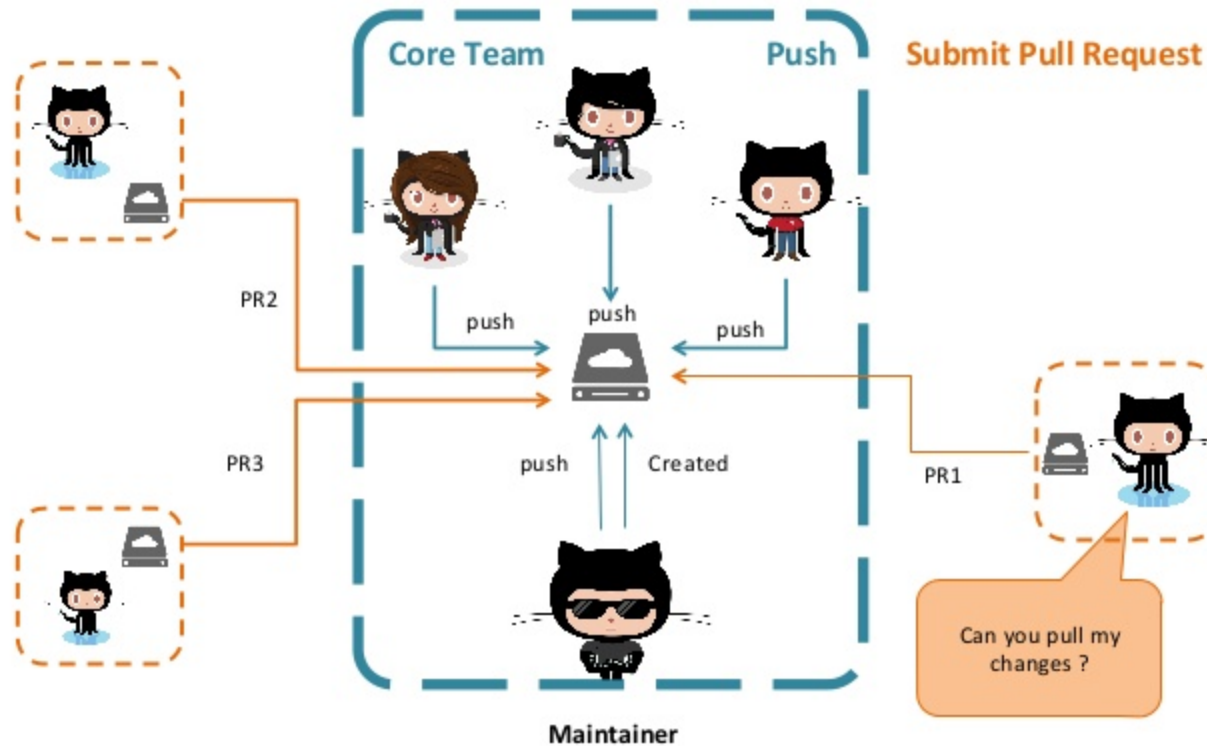
Limite: 18:50

Fluxo de trabalho

Sistemas de controle de versão distribuídos

Modelo 1: Pull Request

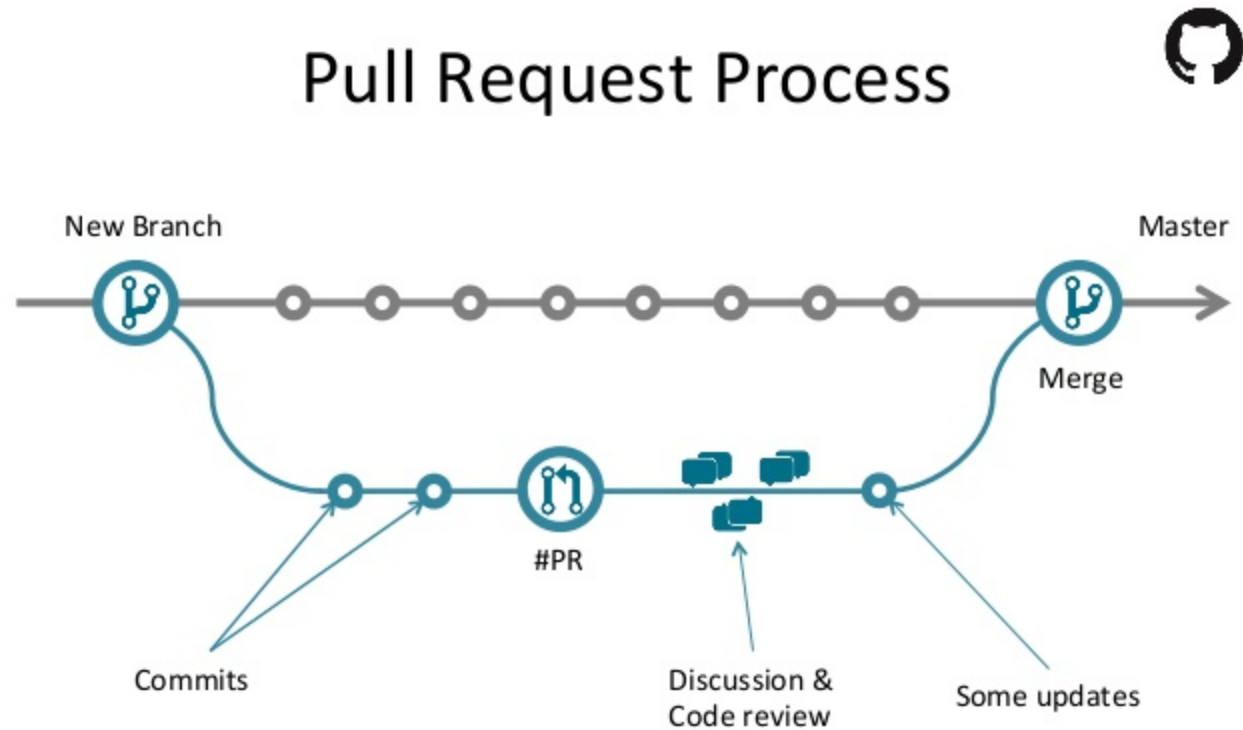
Pull Request Model



Sources : <https://www.iconfinder.com> & <https://octodex.github.com/>

Fonte: <https://www.slideshare.net/abderrahmanebenbachir/continuous-integration-in-github>

Modelo 1: Pull Request



Sources : <https://www.iconfinder.com> & <https://octodex.github.com/>

Fonte: <https://www.slideshare.net/abderrahmanebenbachir/continuous-integration-in-github>

Fluxo de Trabalho: outros modelos

1. Enviar patch via lista de e-mails (ex. Linux)
2. Ferramentas de *code review* (ex. KDE - Phabricator)
3. Patch é anexado no Bug Tracker (ex. Haiku, KDE)
4. Não recebe contribuições externas

Revisão dos PRs da aula passada

Erros comuns

Erro 1: arquivos extras no PR

Erro: modificar arquivos não relacionados a sua mudança proposta.

Ideal:

1. Um Pull Request deverá conter somente os arquivos relacionados a modificação proposta.
2. Cada modificação proposta está em um Pull Request separado

Cria usuário 'antoniojaj' #22

[Edit](#)

 **Open** AntonioAndraues wants to merge 1 commit into [Insper:master](#) from [AntonioAndraues:issue-4](#) 

 Conversation **1**

 Commits **1**

 Checks **0**

 Files changed **8**

+10 -1 

Changes from **all commits**  [File filter...](#)  **x Clear filters** [Jump to...](#)  

0 / 8 files viewed 


[Review changes](#) 

> **BIN +10 KB** .DS_Store 


☐ Viewed ...

> **3**  .gitignore 


☐ Viewed ...

> **BIN +1.76 KB** __pycache__/utils.cpython-36.pyc 



☐ Viewed ...

> **BIN +1.13 KB** skills/__pycache__/__init__.cpython-36.pyc 



☐ Viewed ...

> **BIN +3.44 KB** students/__pycache__/__init__.cpython-36.pyc 

☐ Viewed ...

> **6**  students/antoniojaj 

☐ Viewed ...

> **1**  students/antoniojaj-achievements 

☐ Viewed ...

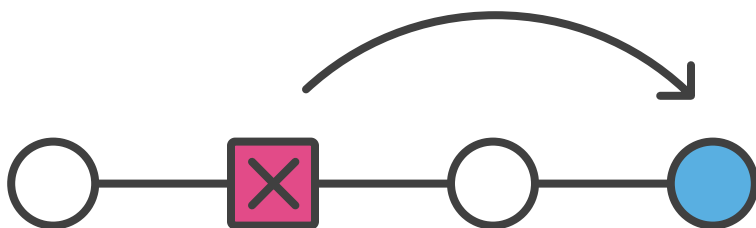
> **1**  students/igorsm1-achievements 

☐ Viewed ...

 **ProTip!** Use [n](#) and [p](#) to navigate between commits in a pull request.

Erro 1: arquivos extras no PR

Como consertar: `git revert` cria um commit que desfaz alterações anteriores.



Opções extras:

- `--no-commit`: só adiciona as modificações, mas não faz o commit. Permite desfazer parcialmente um commit.

Fonte: <https://www.atlassian.com/git/tutorials/>


Erro 2: PR contém mais de um commit

Erro: o PR está espalhado em vários commits.

Ideal:

1. Colocar toda sua modificação em um só commit ajuda a entender o que foi modificado no PR.
2. Mantém histórico limpo, com um commit por bug consertado ou funcionalidade adicionada.
3. Facilita encontrar erros e desfazer a modificação caso ela seja problemática (usando ``git revert``)


Erro 2: PR contém mais de um commit





eduardotp1 commented yesterday

First-time contributor + 😊 ...



No description provided.





eduardotp1 added some commits yesterday

 Primeiros Passos



f4ba5da

 Primeiros Passos


dbda5ee

 Fixes issue #5


fb96798

 Fixes issue #5

ffebac5



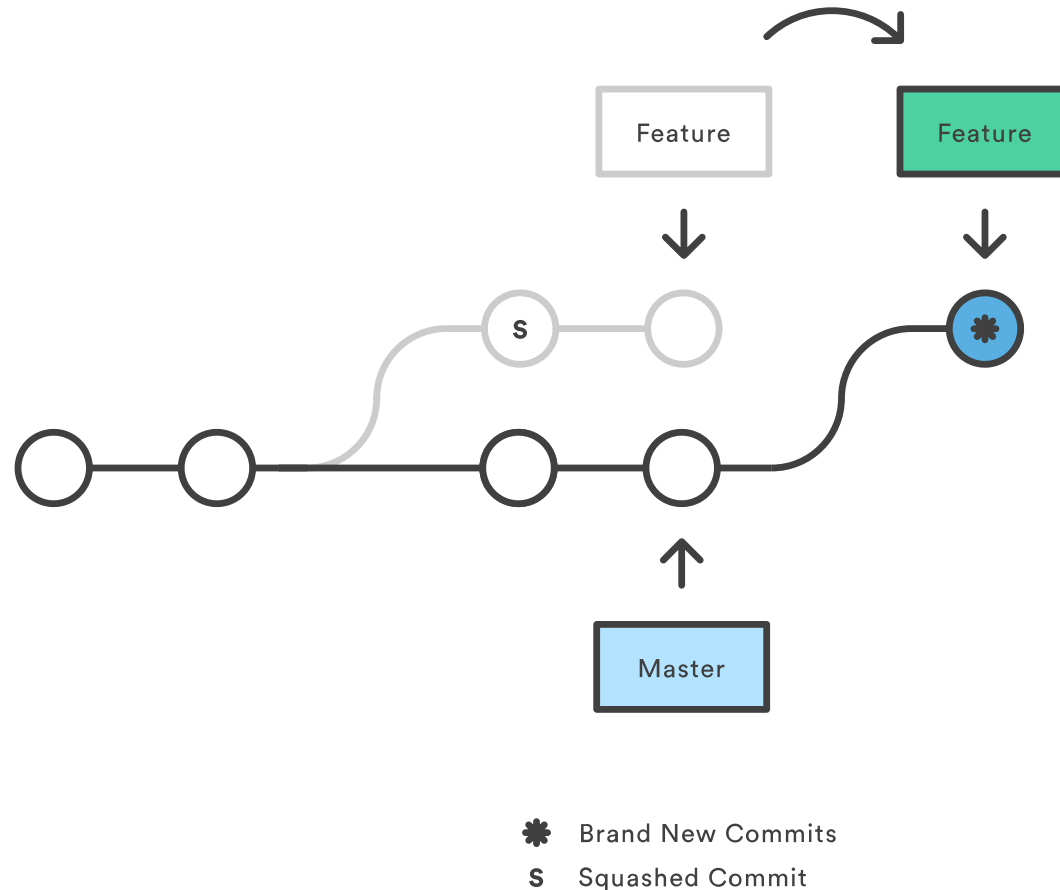
eduardotp1 closed this yesterday



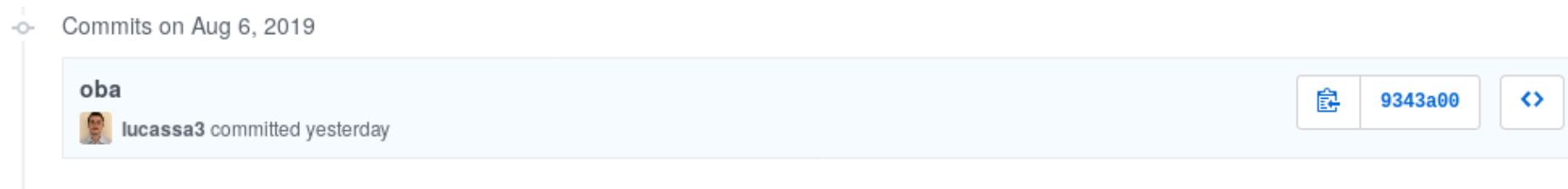
eduardotp1 reopened this yesterday

Erro 2: PR contém mais de um commit

Como consertar: `git rebase -i master` permite remover e juntar commits do branch atual que não estejam no `master`.



Erro 3: Mensagens de commit não descritivas



Como consertar: `git commit --amend` permite modificar o último commit feito.

Atividade

corrigir seu PR e cumprir a última parte do roteiro da última aula.



Comunidades de software