
Gitflow não morde!

É só um melhor fluxo de trabalho

Visão Geral

Idealizado por Vincent Driessen

Janeiro de 2010

Vantagens

- Não precisa de um novo repositório
- Facilita o entendimento
- Otimiza o fluxo de trabalho
- Bitbucket possui suporte

Problema

Se torna um caos gigantesco se não utilizado corretamente!

Branches

- *Master*



Branches

- *Master*
- *Develop ou Development*



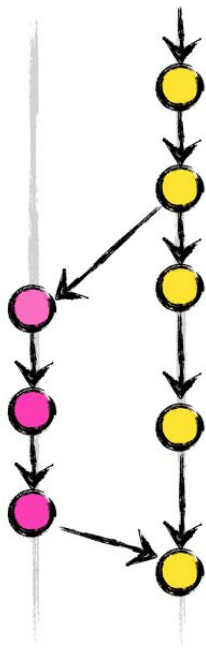
Branches

- Master
- Develop
- Release



feature
branches

develop



Os "galhos" de apoio

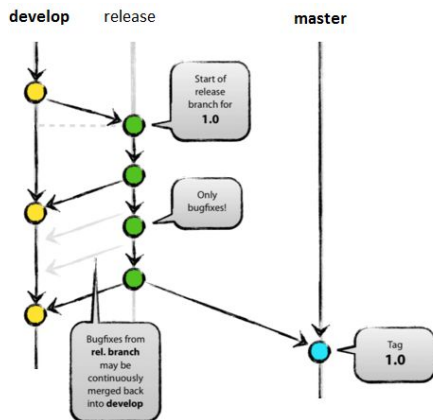
feature/

- Cada HEAD é uma task, subtask ou bug
- Ela pode ser um story ou apenas uma task

bugfix/

- Cada HEAD é uma correção
 - Ela só pode ser um ou mais bugs
- Criadas da development
 - Voltam para a development

Os "galhos" de apoio



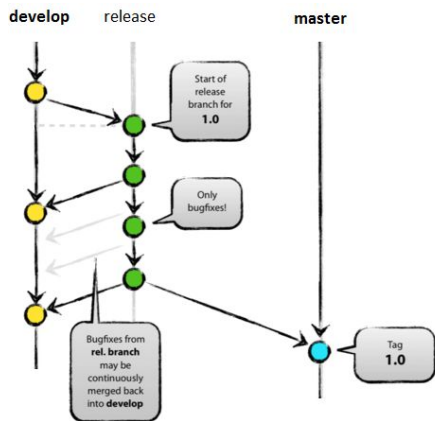
release/x.y.z

- Branch do servidor de staging
- Pode receber envios
- Criada da development
- Volta para development e gera um envio na master

hotfix/

- **Não é a bugfix!**
- Cada hotfix é um ou mais bugs críticos
- Criada da master!
- Volta para master e gera um envio para a develop

Os "galhos" de apoio



release/x.y.z

- Branch do servidor de staging
- Pode receber envios
- Criada da development
- Volta para development e gera um envio na master



hotfix/x.y.z

- Não é a bugfix!
- Cada HEAD é um ou mais bugs críticos
- **Criada da master!**

Volta para master e gera um envio para a develop



Cuidado com nomes

- Simples e compreensível
- Descreva no geral o que será feito
- Coloque sentido no mesmo
- Use o padrão do projeto
- Use o código da issue no Jira mais uma descrição simples (exemplo: feature/RIN-1-login)
 - O código da issue tem que ser exatamente como está no Jira
 - Use sempre “-” para separar palavras. Não use camelCase
- Tudo tem que ser em inglês



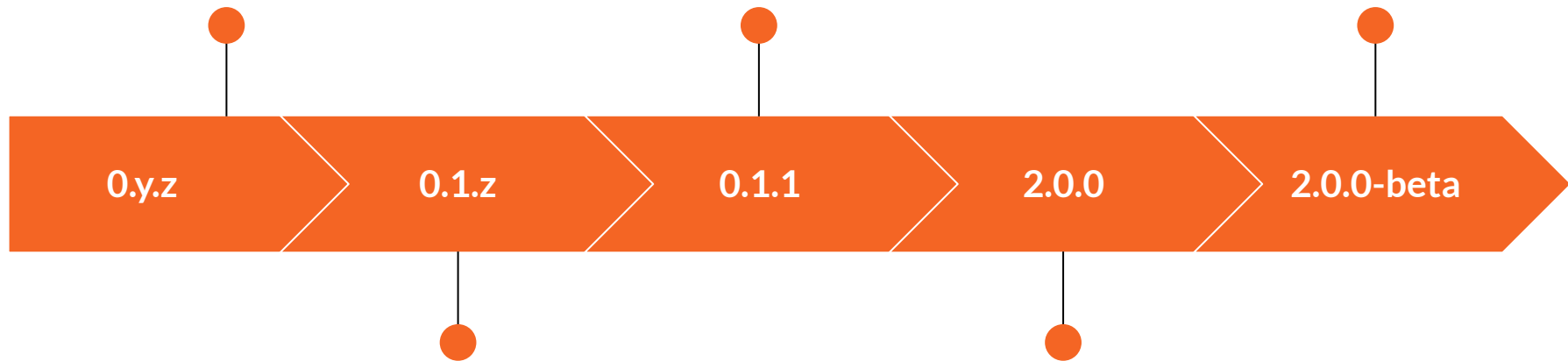
VERSIONAMENTO

Visão geral

No início do desenvolvimento, a versão MAJOR deve ser zero. Versão Maior deve ser incrementada se forem introduzidas mudanças incompatíveis com a documentação

Versão de PATCH deve ser incrementado apenas se mantiver compatibilidade com a documentação e introduzir correção de bugs.

Quando MAJOR, MINOR e PATCH são iguais, a versão de Pré-Lançamento tem precedência menor que a versão normal



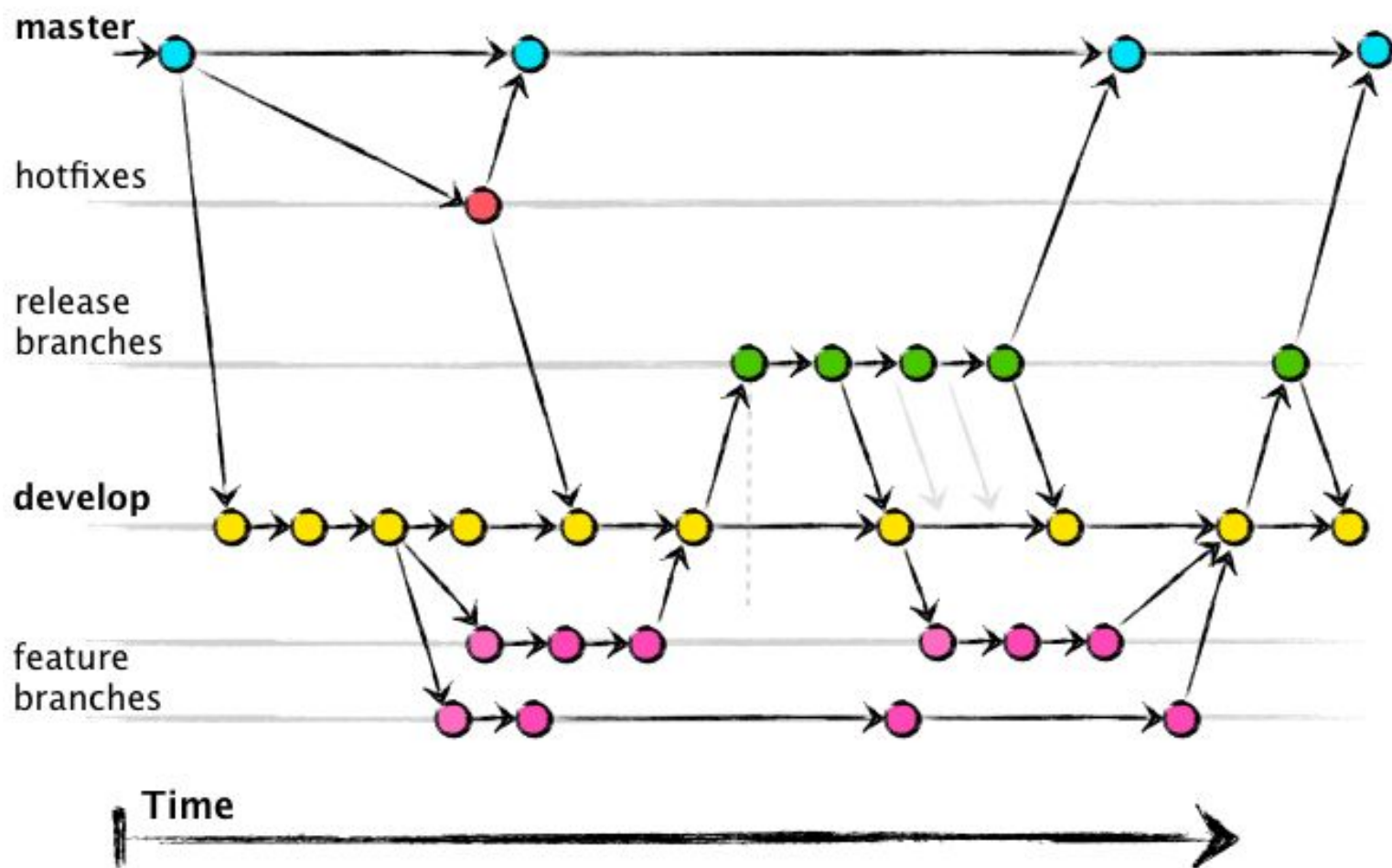
Versão MINOR deve ser incrementada se uma funcionalidade nova e compatível com a documentação for introduzida

A precedência é determinada pela primeira diferença quando se compara cada identificador da esquerda para direita

IIIIIT'S TIIIIIME!

O fluxo completo passo a passo



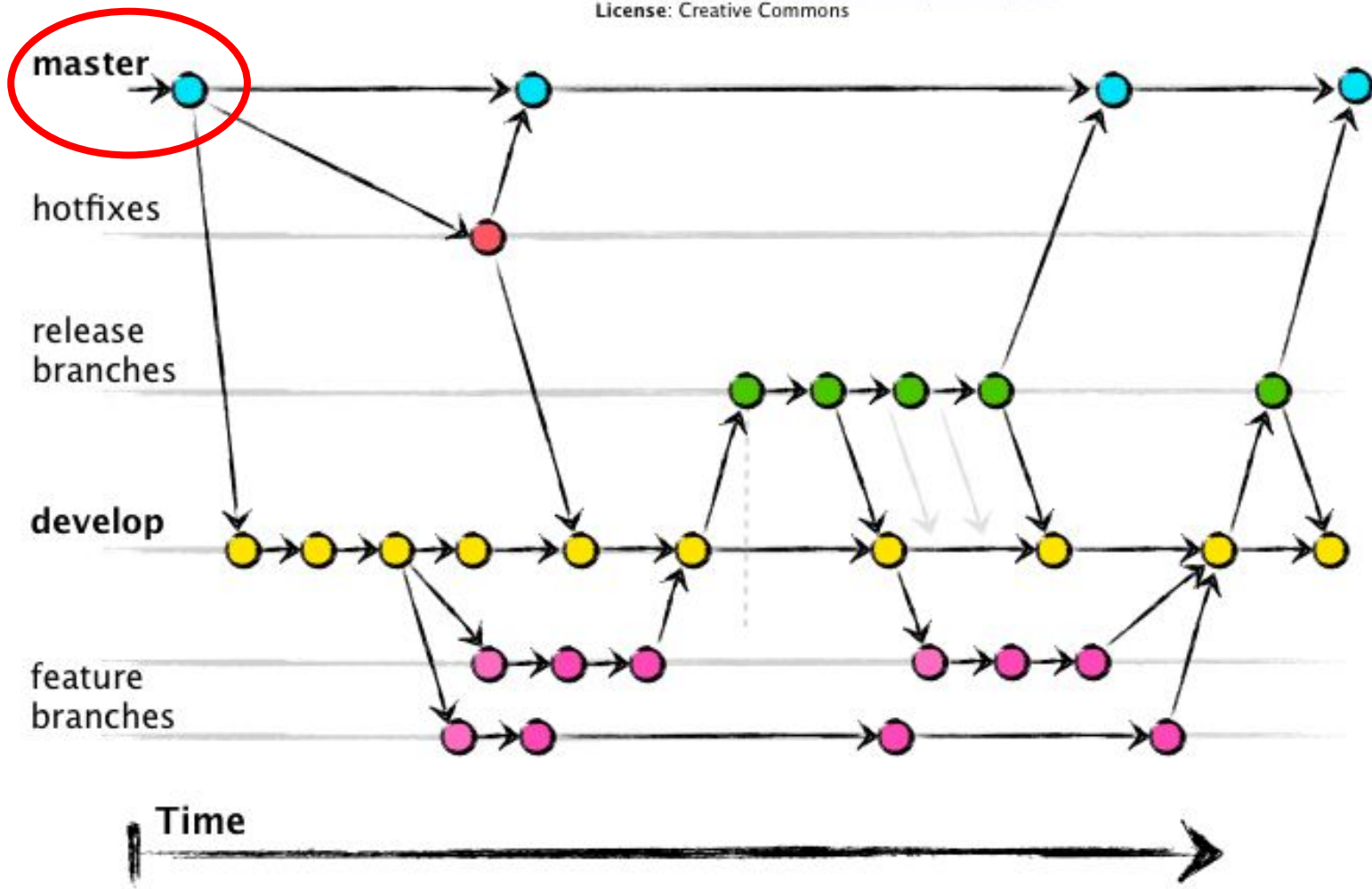


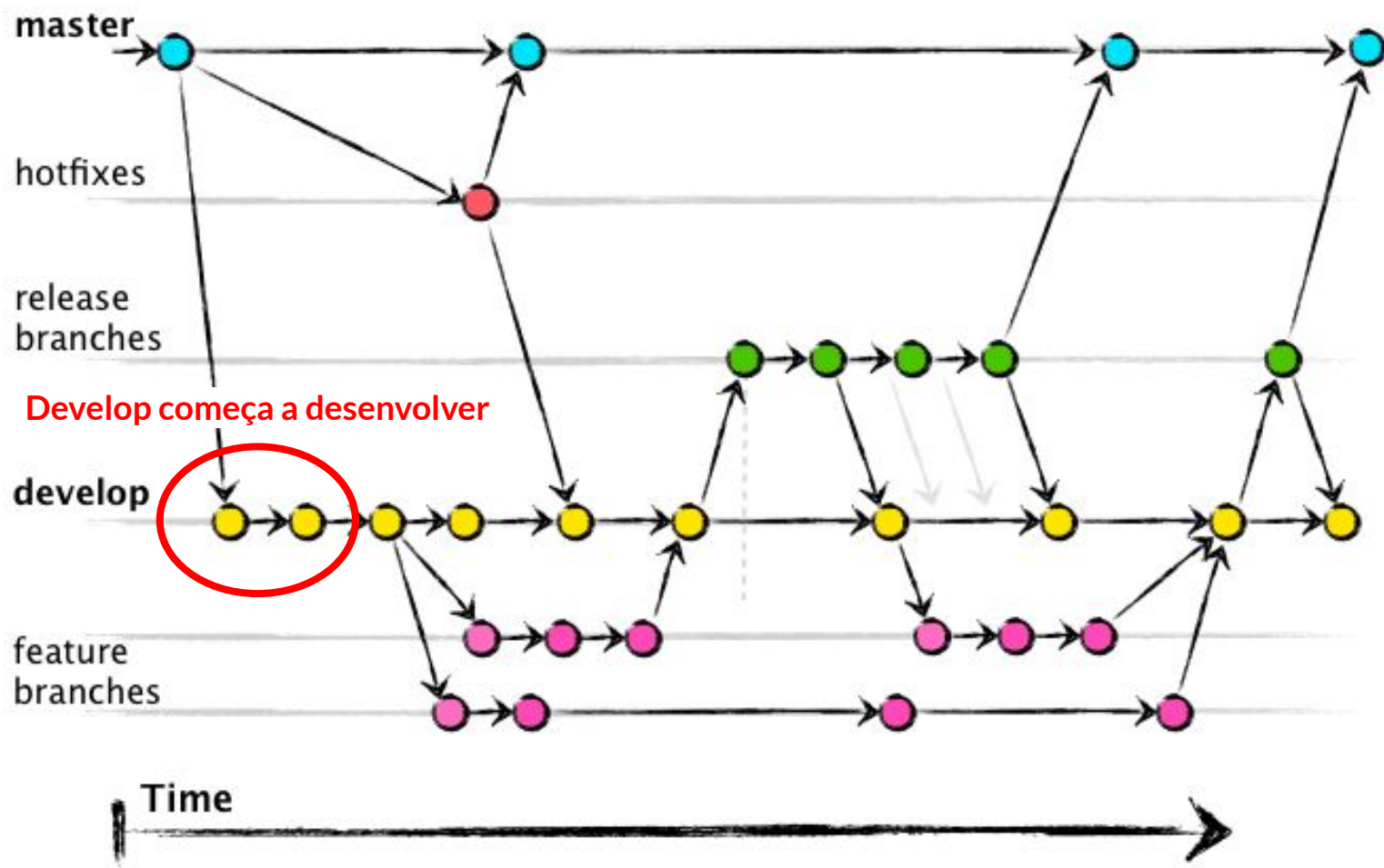
Versão 0.1.0 - Initial Commit

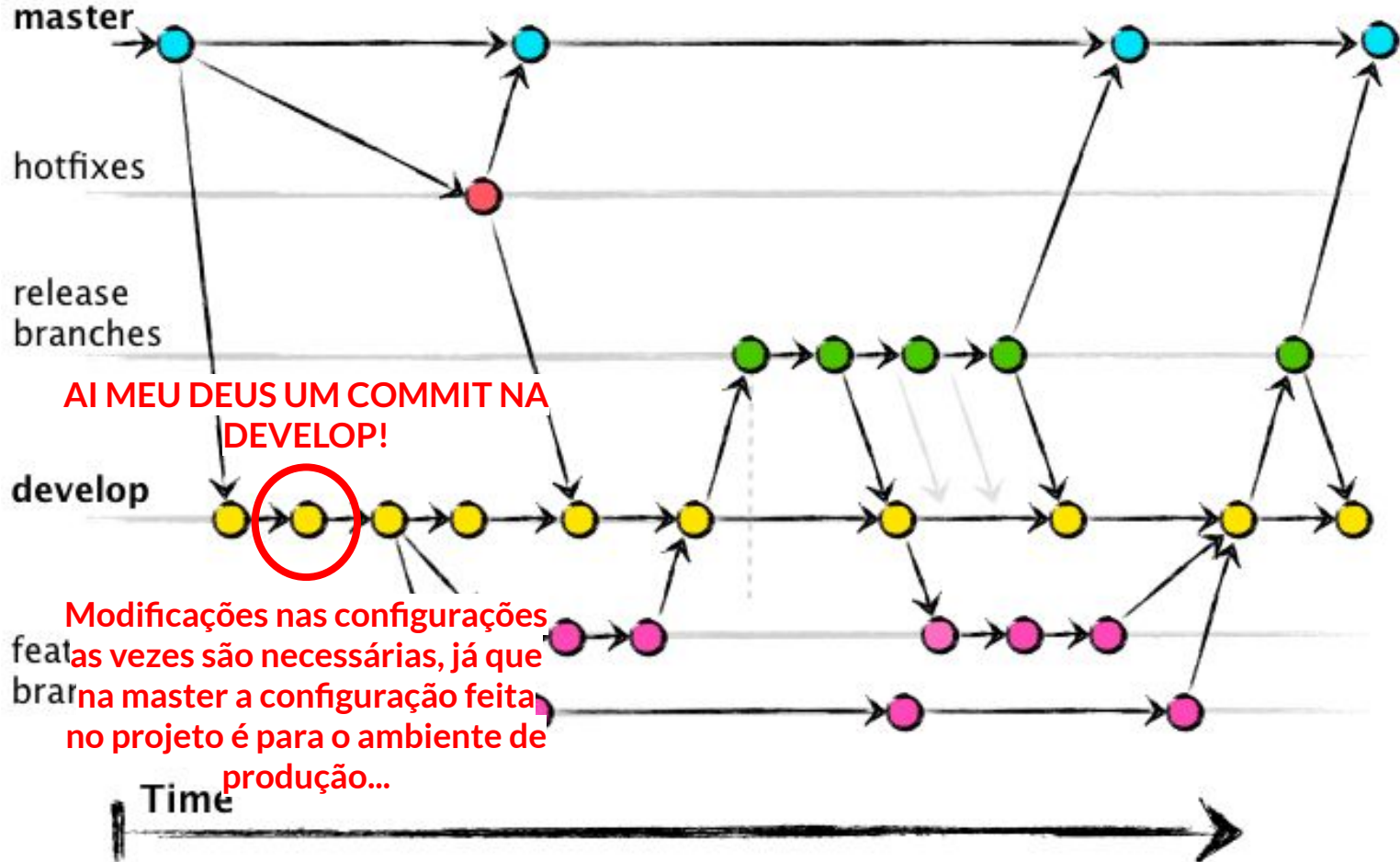
Author: Vincent Driessen

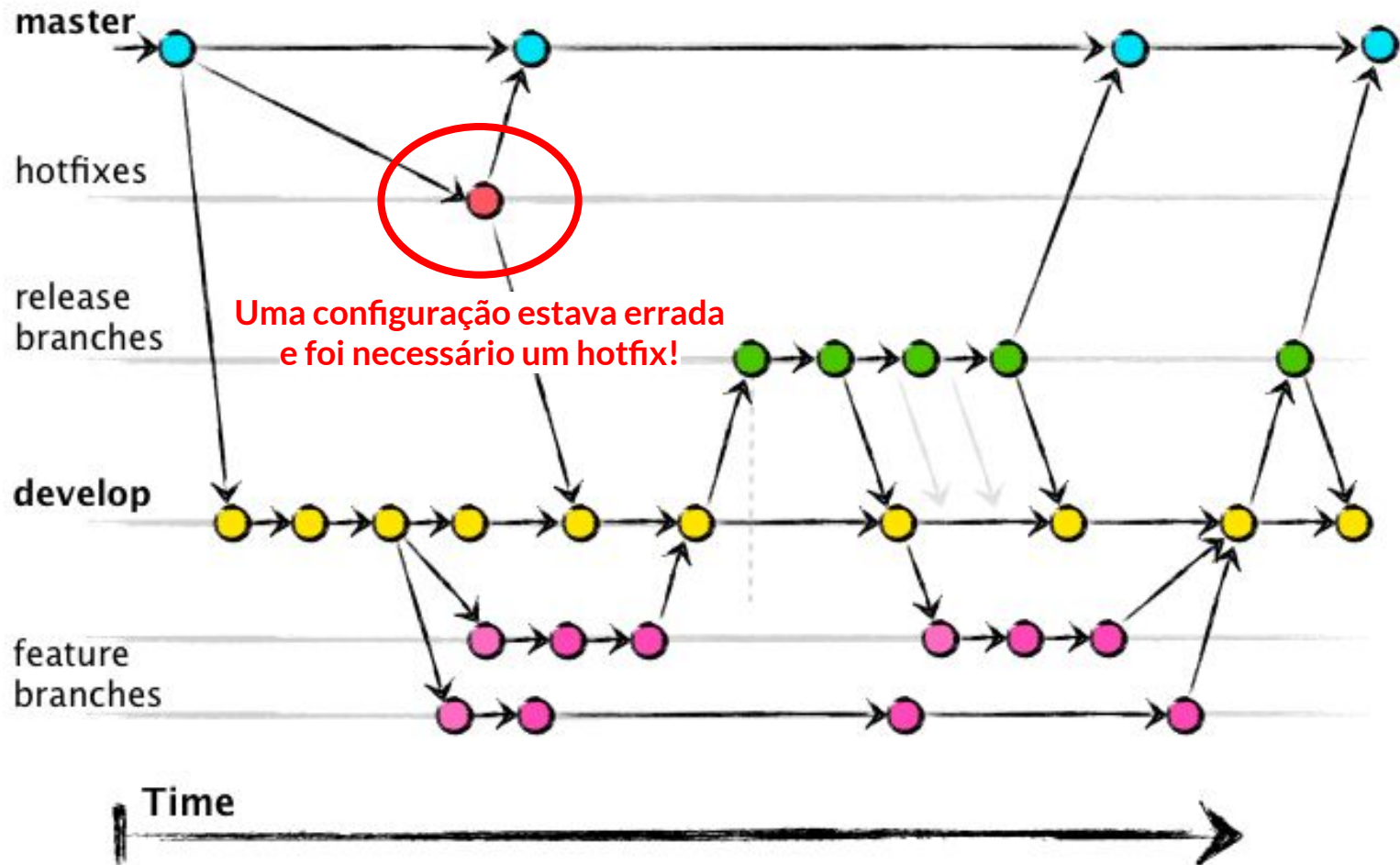
Original blog post: <http://nvie.com/archives/323>

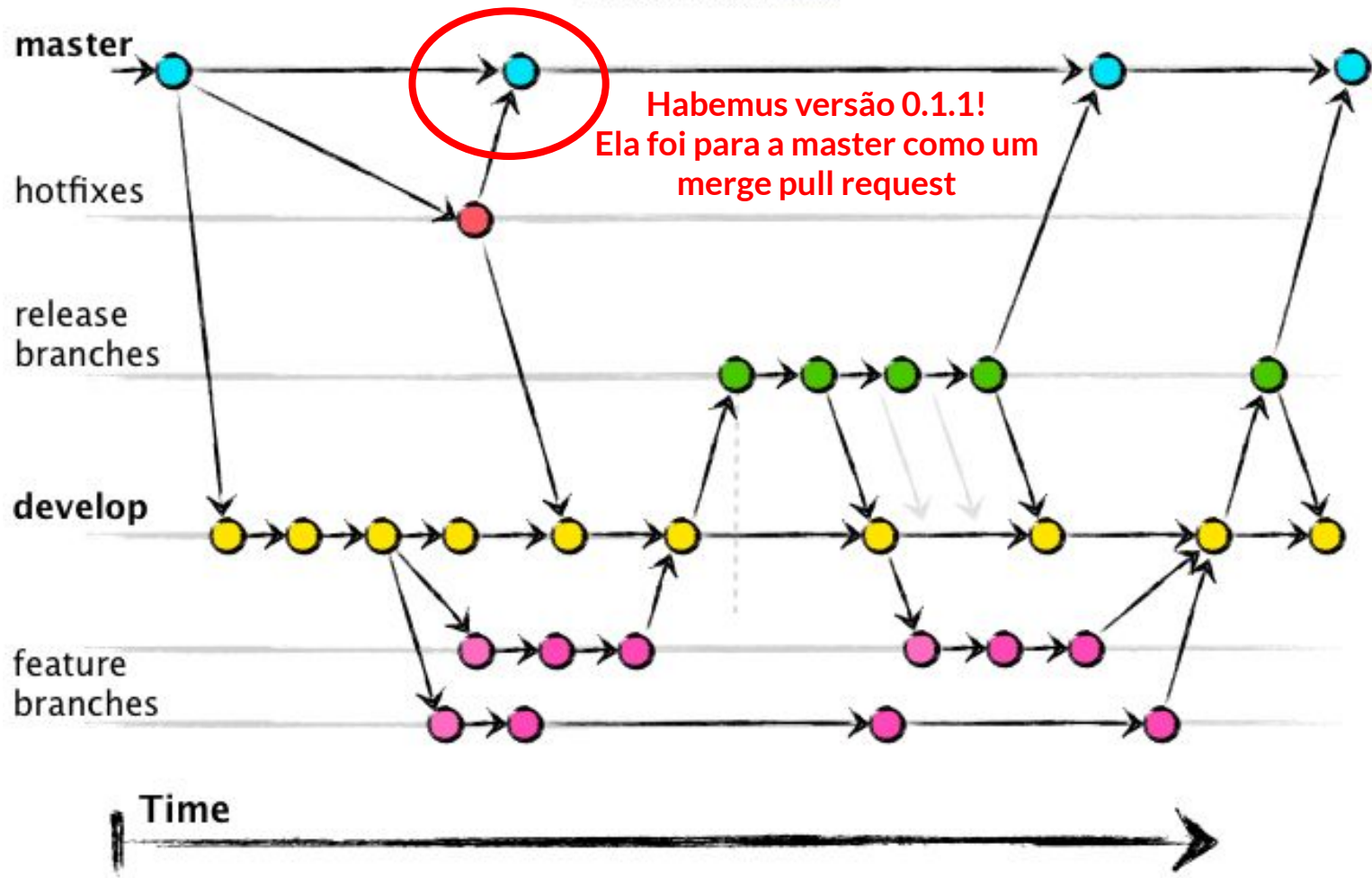
License: Creative Commons

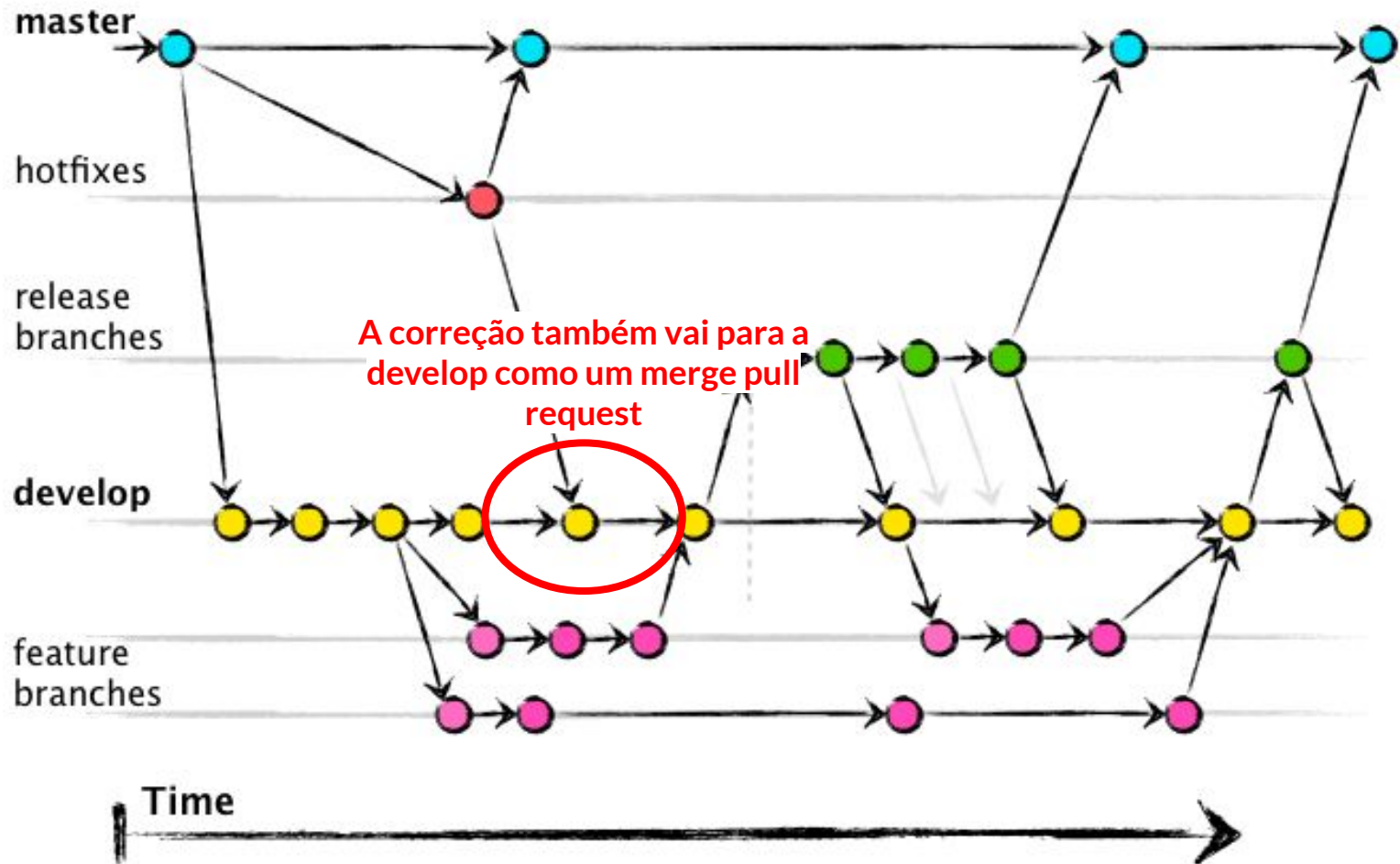


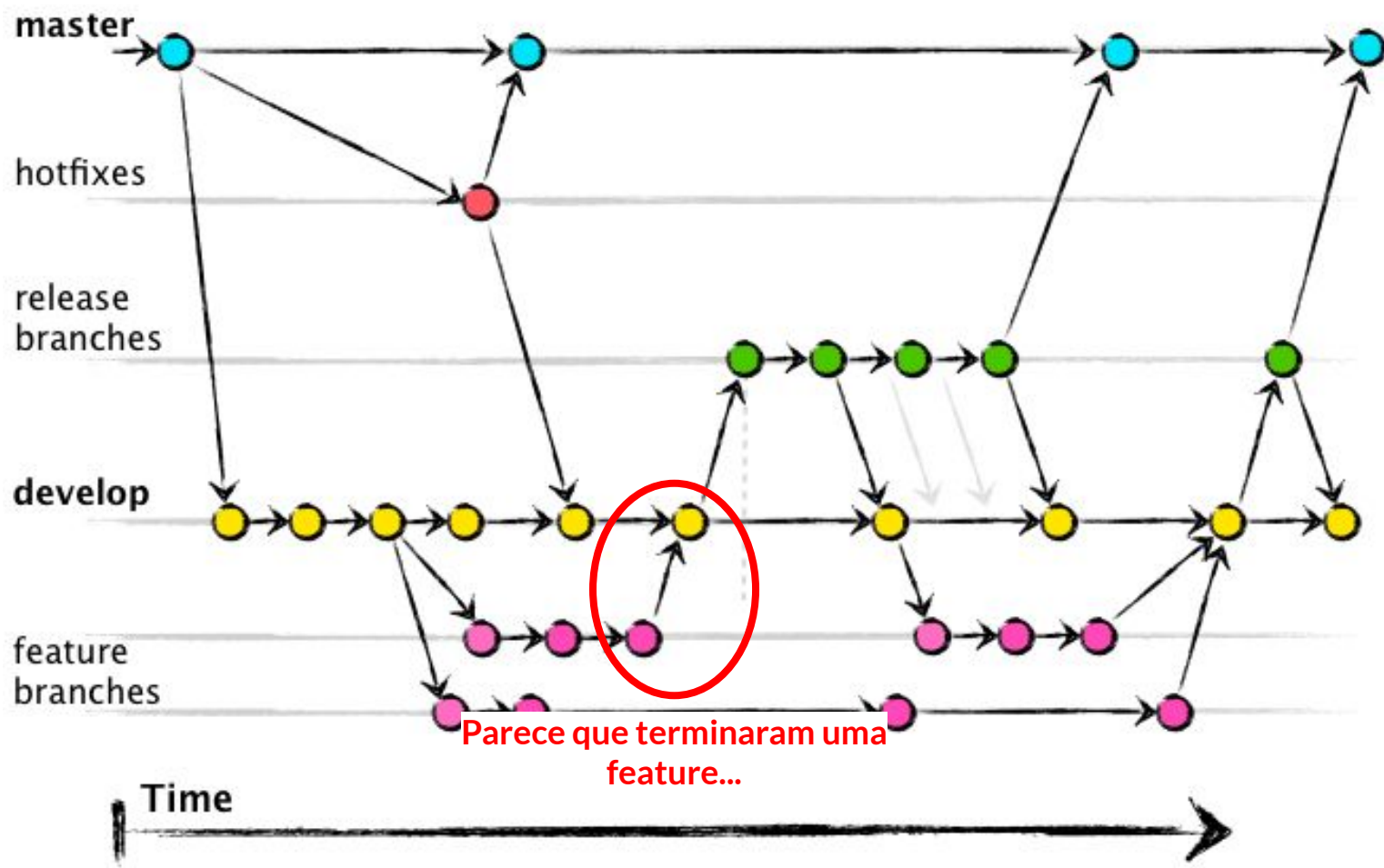


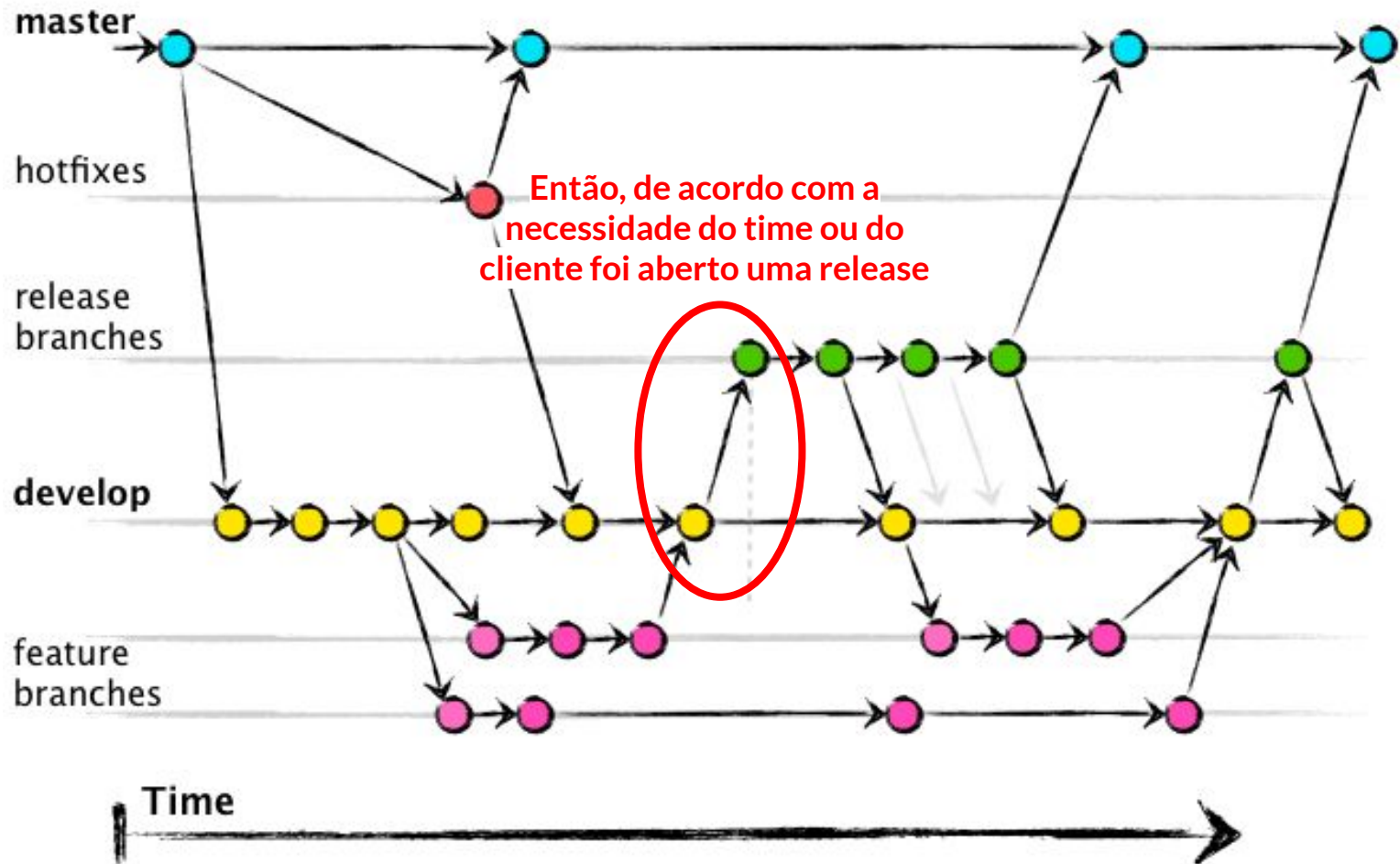


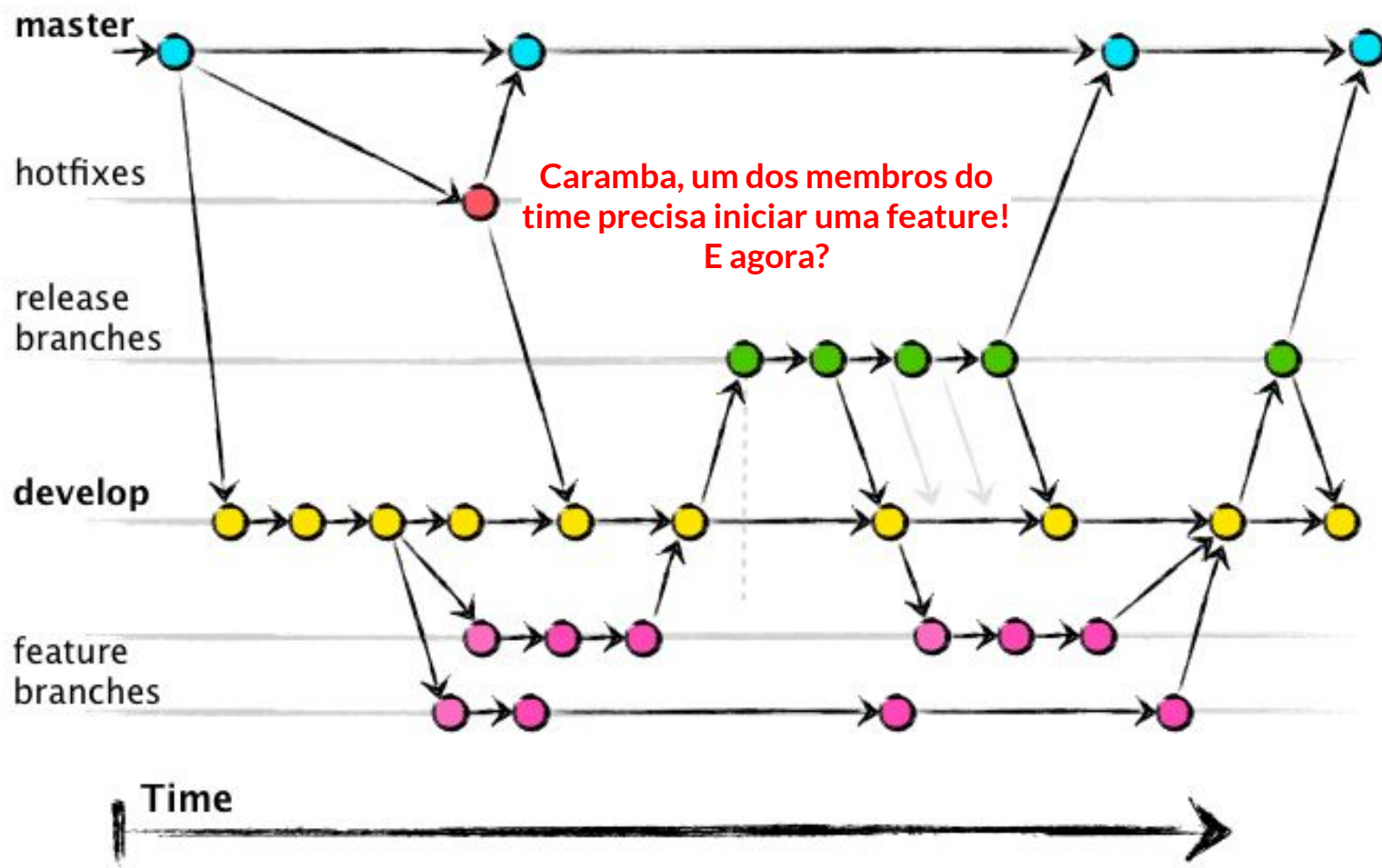


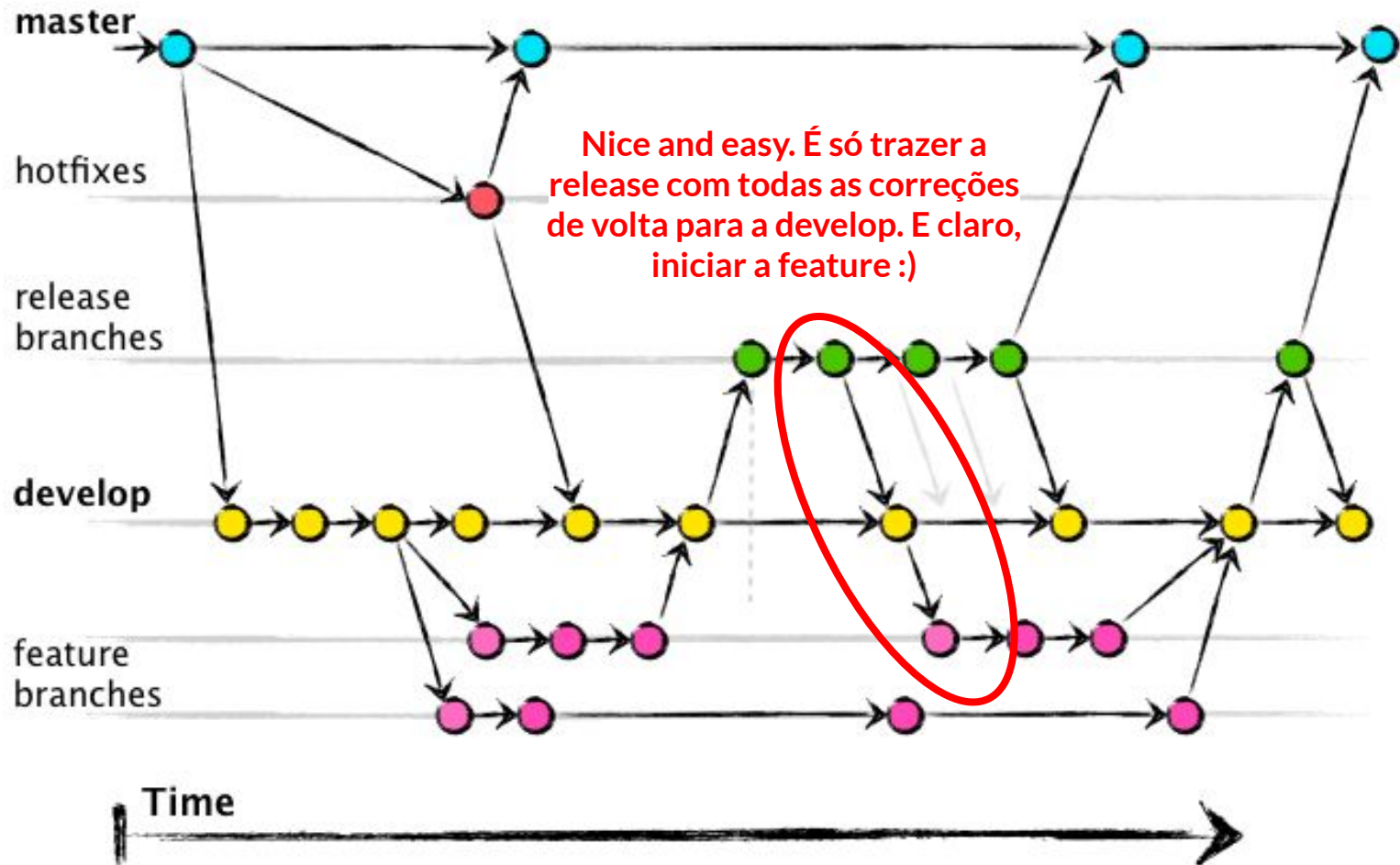


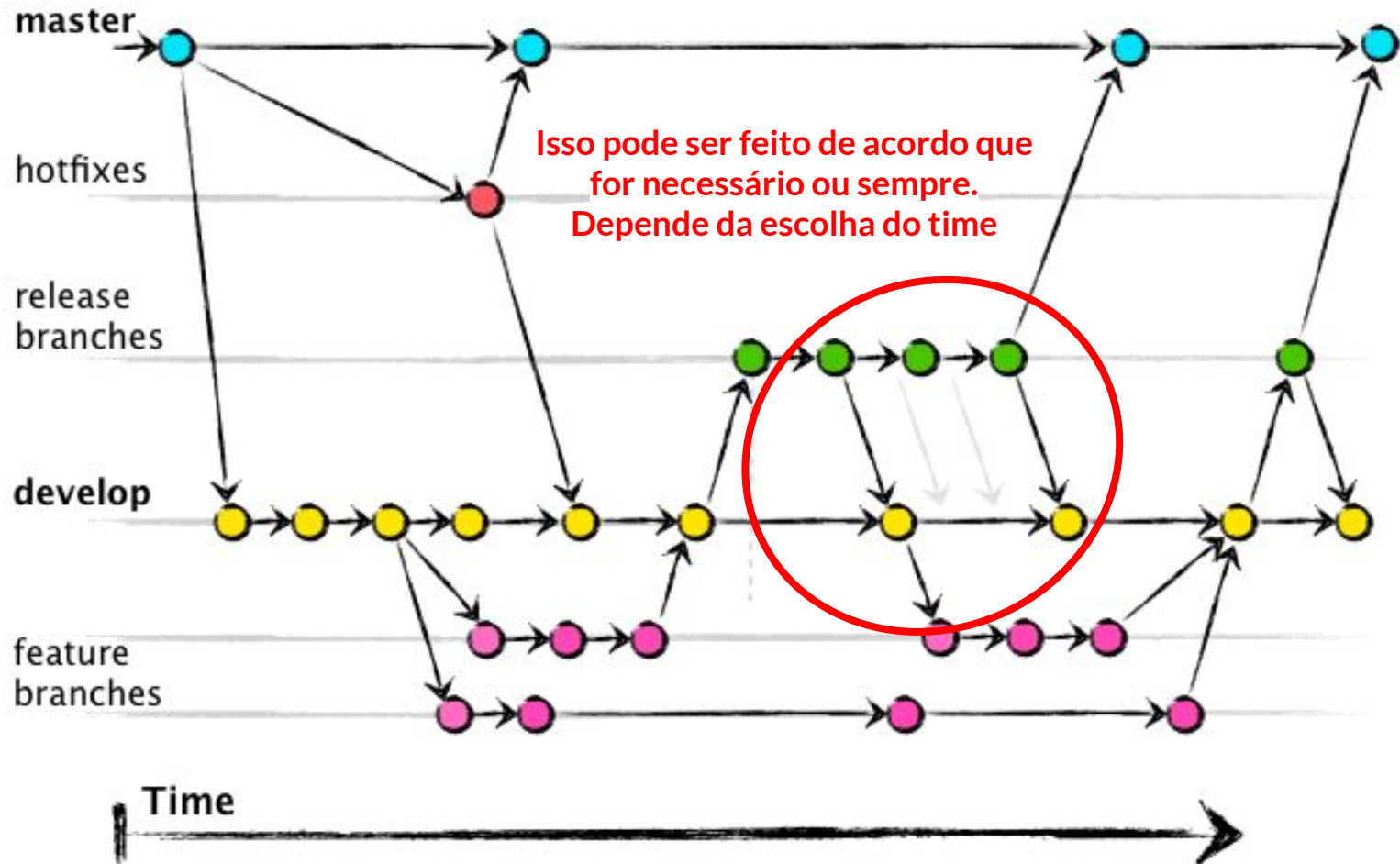


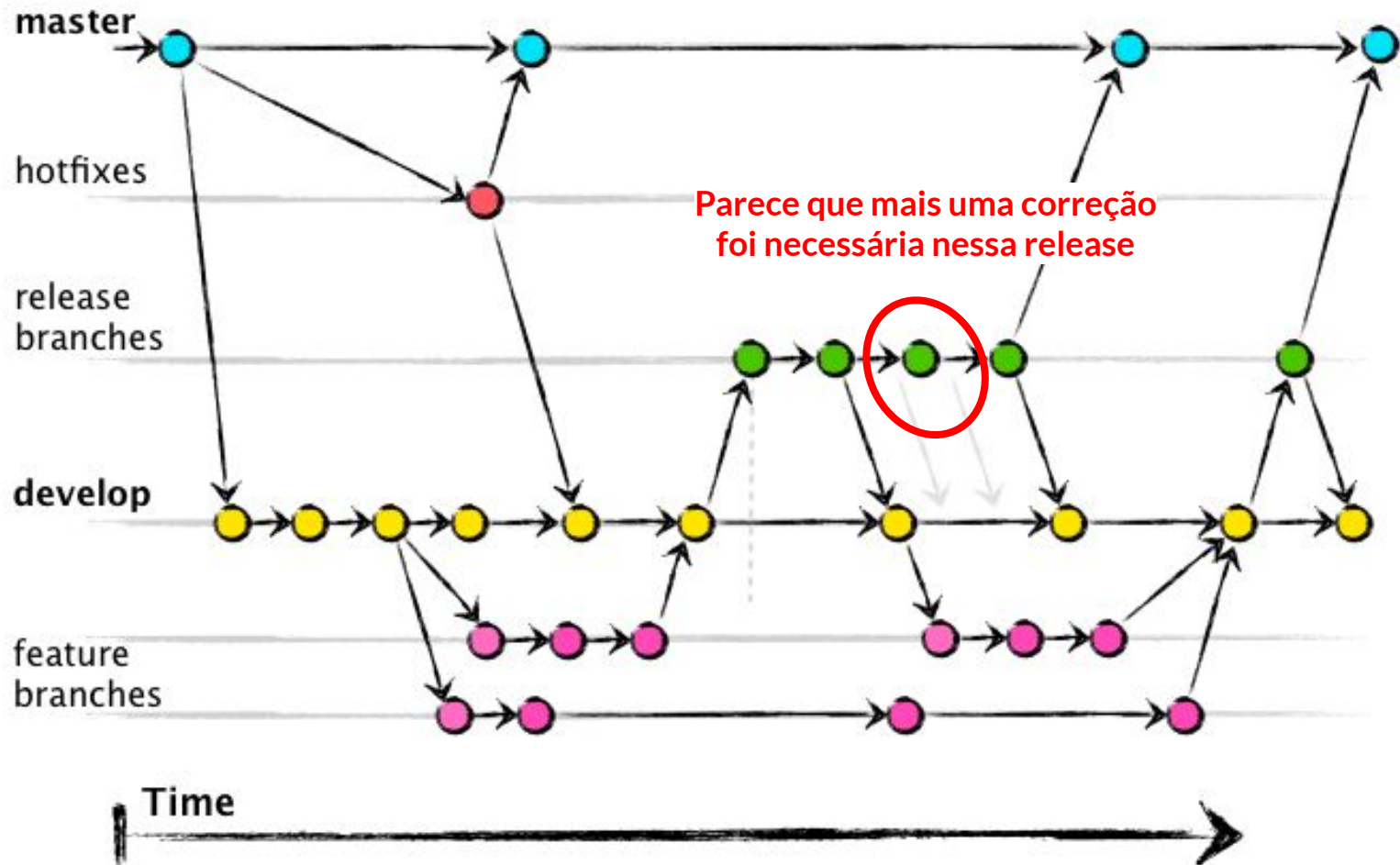










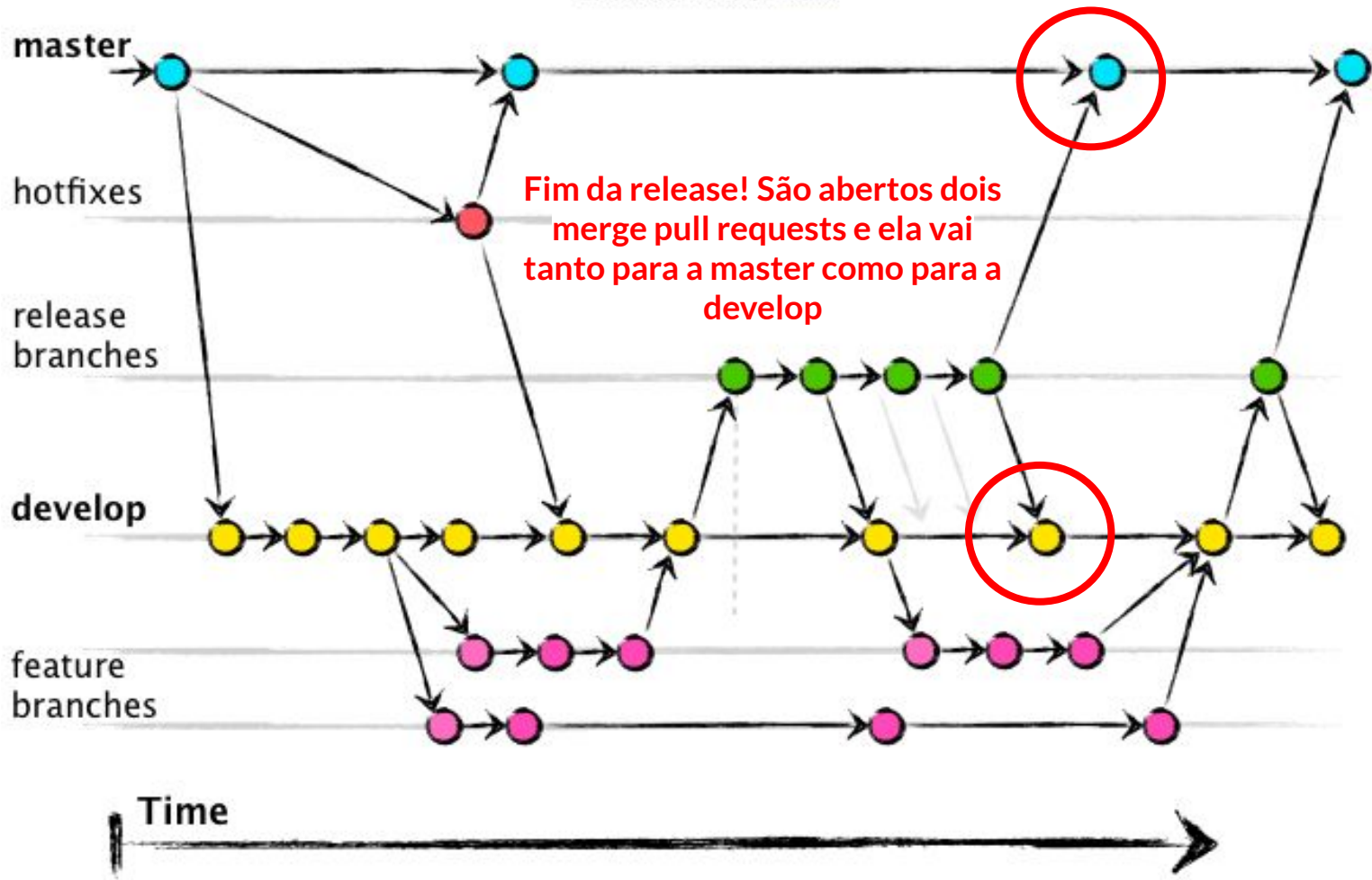


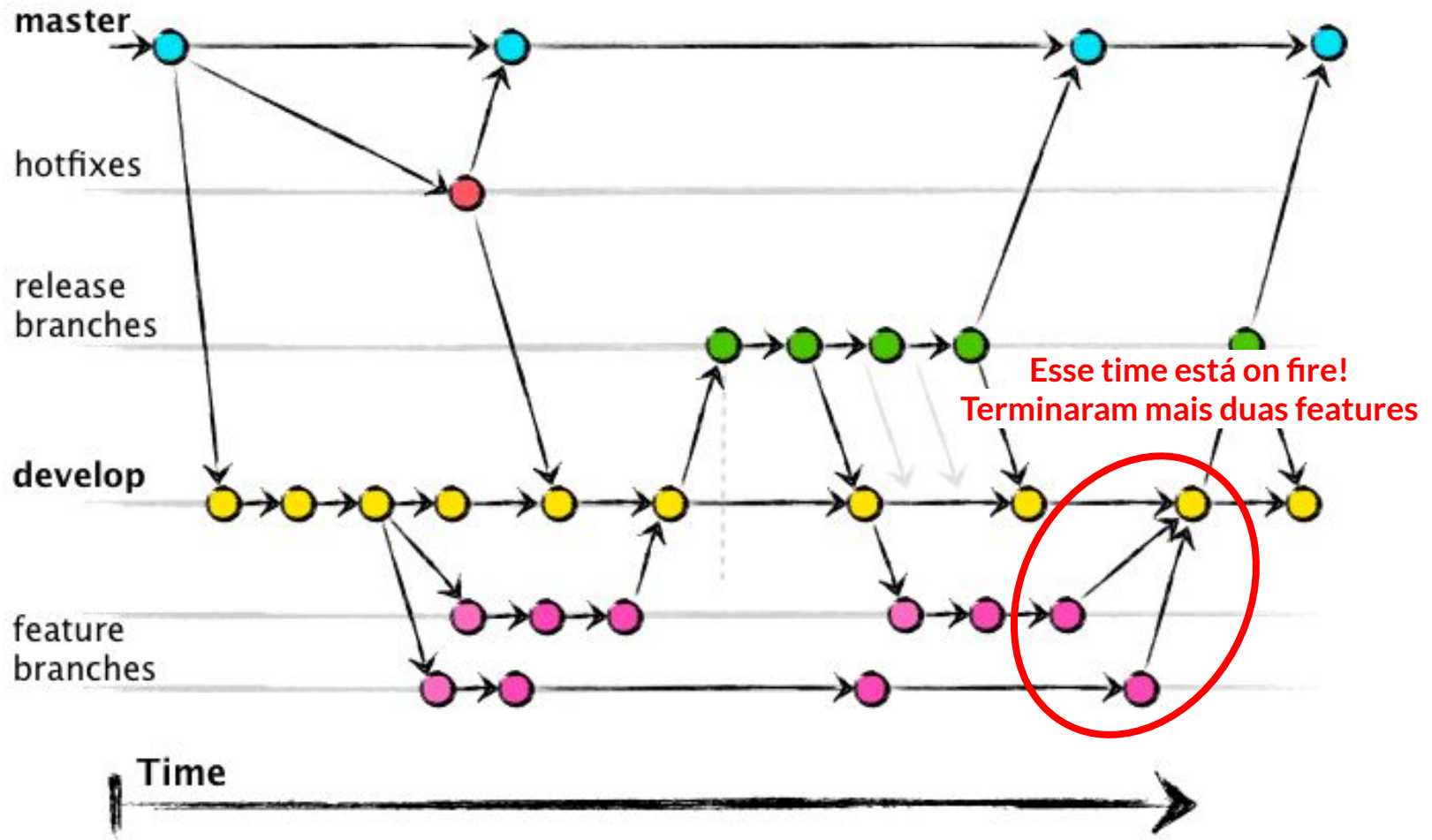
Author: Vincent Driessen

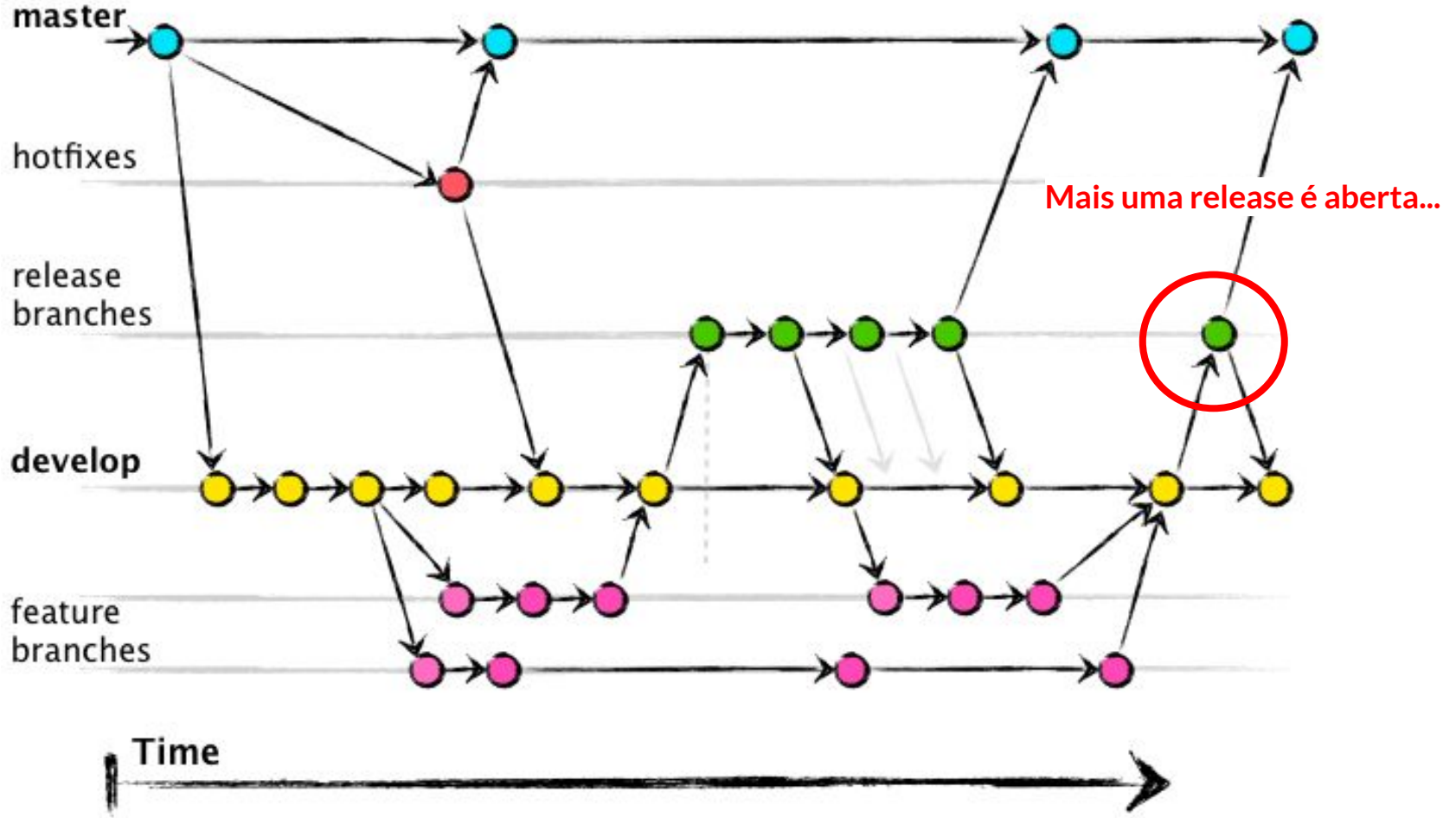
Original blog post: <http://nvie.com>

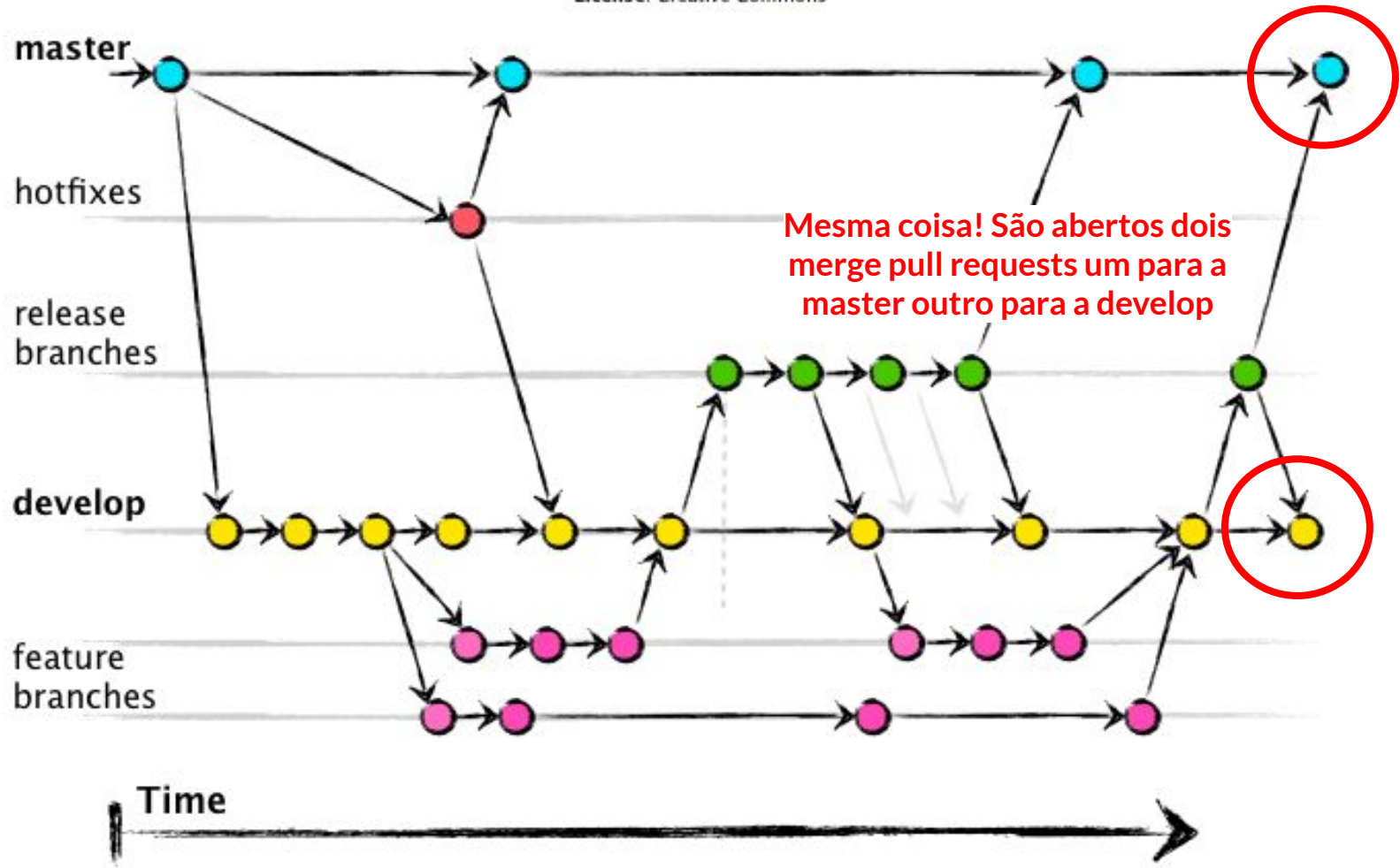
License: Creative Commons

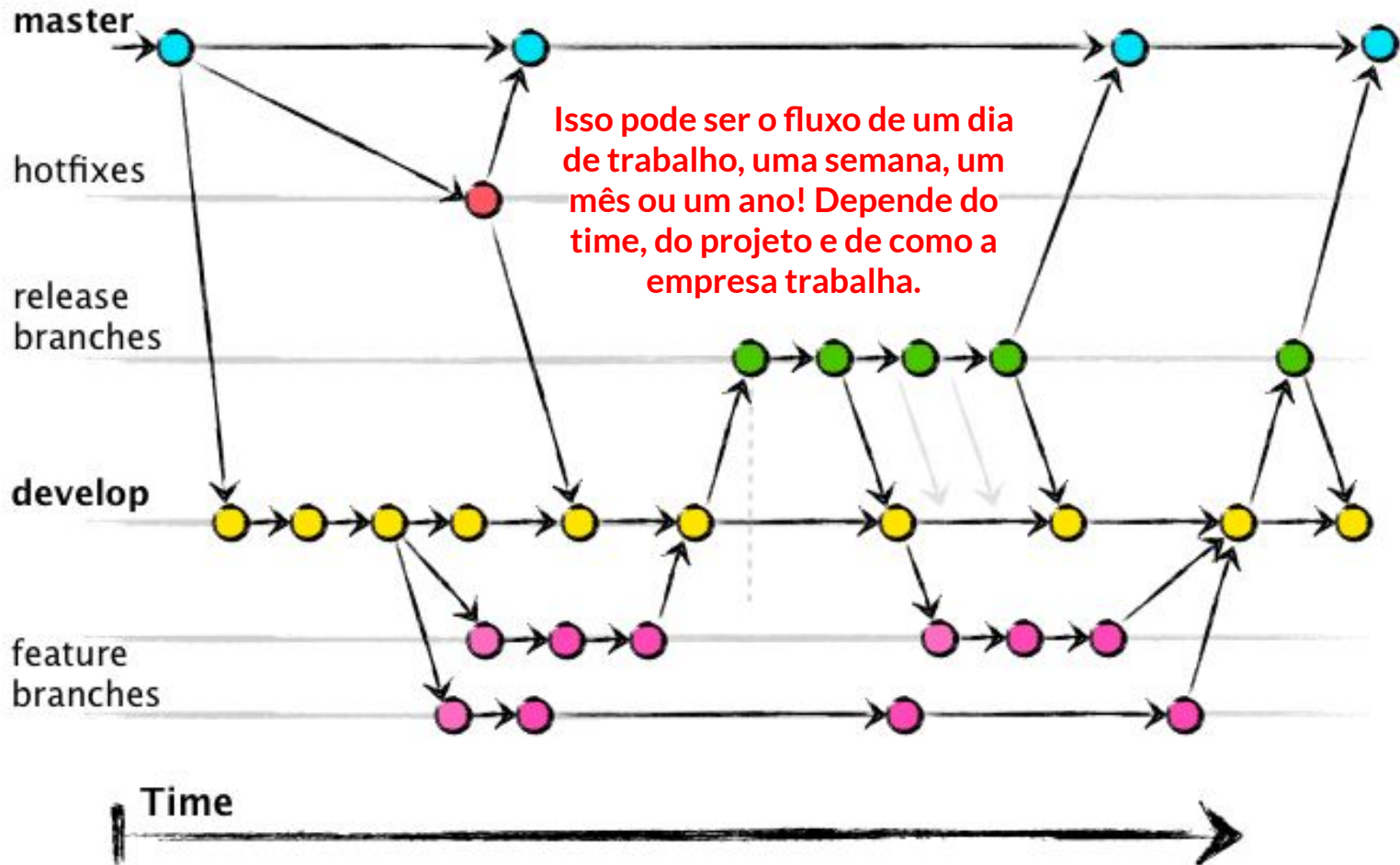
Habemus versão 0.2.0!







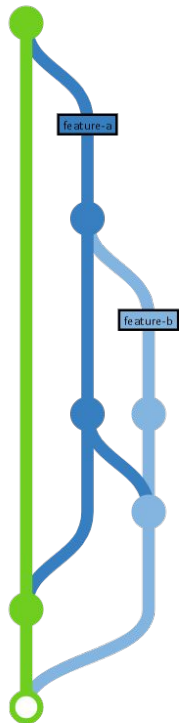




DICAS

Elas podem ser úteis ~~com elas~~

develop

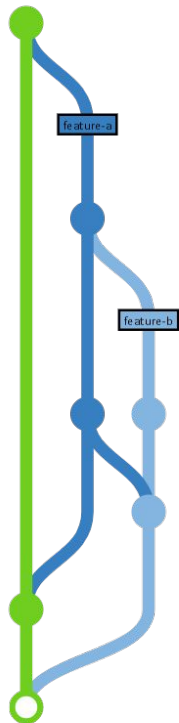


E se eu precisar de uma feature para fazer a minha?

É fácil, vamos lá:

- Crie sua feature/branch-B da feature/branch-A;
- Trabalhe na sua feature normalmente;
- Se for adicionado mais alguma coisa na feature/branch-A é só dar um rebase;
- Termine seu trabalho na feature/branch-B e espere até que a feature/branch-A seja enviada para develop;
- Se quiser faça um rebase da sua feature com a develop;
- Então é só abrir um pull request para a develop. Pronto!

develop

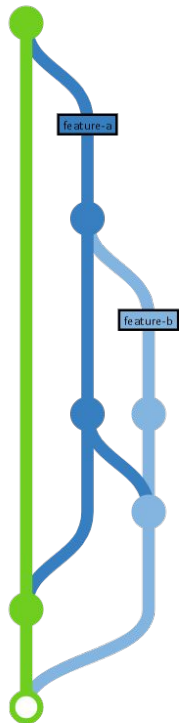


Não é tão fácil assim né? E eu particularmente odeio rebase!

Com vocês o "caminho melhor":

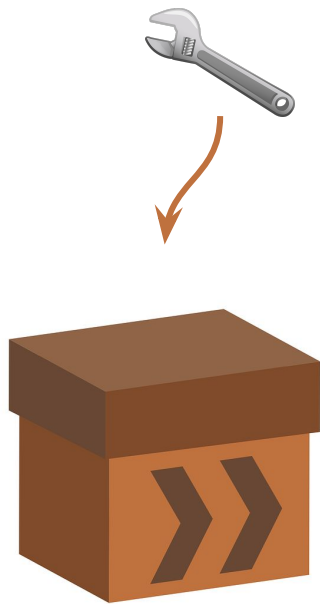
- Crie sua feature/branch-B da feature/branch-A;
- Trabalhe na sua feature normalmente;
- Se for adicionado mais alguma coisa na feature/branch-A é só dar um rebase **merge**;
- Termine seu trabalho na feature/branch-B e espere até que a feature/branch-A seja enviada para develop;
- ~~Se quiser faça um rebase da sua feature com a develop;~~
- Então é só abrir um pull request para a develop. Pronto!

develop



Qual a vantagem de um para o outro?

- O primeiro caminho é o que o git recomenda. Para eles não pode existir uma feature de uma feature. Tudo vai ser feito a partir de uma branch principal;
- O segundo caminho além de preservar o histórico ainda reduz a quantidade de erros que podem acontecer (é sério, rebase é um negócio muito chato até pra quem conhece git a alguns bons anos);
- Independente de qual utilizar, use-o até o fim. É importante ser fiel ao seu time.



E um Hotfix durante um Release?

- **Como diz Vincent em seu artigo enquanto fala sobre os hotfixes:** *A única exceção para essa regra é: quando uma branch de release existir, o hotfix deve ser “merged” na branch de release ao invés da develop. [...] Se alguma coisa precisa ser prontamente enviada para a develop e não pode esperar pelo release você pode, de forma segura, dar um merge do hotfix na develop também.*

COMANDOS

Esses sim vão ser úteis com certeza!



Antes de tudo: escolha uma

Saiba que existem duas formas de se trabalhar com o gitflow. Utilizando os comandos *cru*s do git ou utilizando o *git-flow cheatsheet*!

- **Comando cru:** você não precisa de nada além do git para dar início ao projeto com gitflow. Basta seguir a lógica que foi descrita anteriormente que tudo irá - talvez - correr bem.
 - **Comandos git-flow cheatsheet:** precisa ser instalado utilizando brew, port, apt-get ou no windows utilizando *Cygwin*. Depois de instalado possui uma série de comandos que facilita seguir todo o fluxo anteriormente descrito.
-

Agora vamos aos trabalhos

Dependendo do lado que escolheu, você saberá se foi uma sábia ou não escolha...

Comandos descaradamente copiados de:

<https://gist.github.com/JamesMGreene/cdd0ac49f90c987e45ac>

Initialize

gitflow	git
<code>git flow init</code>	<code>git init</code>
	<code>git commit --allow-empty -m "Initial commit"</code>
	<code>git checkout -b develop master</code>

Connect to the remote repository

gitflow	git
N/A	<code>git remote add origin git@github.com:MYACCOUNT/MYREPO</code>

Features

Create a feature branch

gitflow	git
<code>git flow feature start MYFEATURE</code>	<code>git checkout -b feature/MYFEATURE develop</code>

Share a feature branch

gitflow	git
<code>git flow feature publish MYFEATURE</code>	<code>git checkout feature/MYFEATURE</code>
	<code>git push origin feature/MYFEATURE</code>

Get latest for a feature branch

gitflow	git
<code>git flow feature pull origin MYFEATURE</code>	<code>git checkout feature/MYFEATURE</code>
	<code>git pull --rebase origin feature/MYFEATURE</code>

Finalize a feature branch

gitflow	git
<code>git flow feature finish MYFEATURE</code>	<code>git checkout develop</code>
	<code>git merge --no-ff feature/MYFEATURE</code>
	<code>git branch -d feature/MYFEATURE</code>

Push the merged feature branch

gitflow	git
N/A	<code>git push origin develop</code>
	<code>git push origin :feature/MYFEATURE</code> <i>(if pushed)</i>

Releases

Create a release branch

gitflow	git
<code>git flow release start 1.2.0</code>	<code>git checkout -b release/1.2.0 develop</code>

Share a release branch

gitflow	git
<code>git flow release publish 1.2.0</code>	<code>git checkout release/1.2.0</code>
	<code>git push origin release/1.2.0</code>

Get latest for a release branch

gitflow	git
<code>N/A</code>	<code>git checkout release/1.2.0</code>
	<code>git pull --rebase origin release/1.2.0</code>

Finalize a release branch

gitflow	git
<code>git flow release finish 1.2.0</code>	<code>git checkout master</code>
	<code>git merge --no-ff release/1.2.0</code>
	<code>git tag -a 1.2.0</code>
	<code>git checkout develop</code>
	<code>git merge --no-ff release/1.2.0</code>
	<code>git branch -d release/1.2.0</code>

Push the merged feature branch

gitflow	git
N/A	<code>git push origin master</code>
	<code>git push origin develop</code>
	<code>git push origin --tags</code>
	<code>git push origin :release/1.2.0</code> <i>(if pushed)</i>

Hotfixes

Create a hotfix branch

gitflow	git
<code>git flow hotfix start 1.2.1 [commit]</code>	<code>git checkout -b hotfix/1.2.1 [commit]</code>

Finalize a hotfix branch

gitflow	git
<code>git flow hotfix finish 1.2.1</code>	<code>git checkout master</code>
	<code>git merge --no-ff hotfix/1.2.1</code>
	<code>git tag -a 1.2.1</code>
	<code>git checkout develop</code>
	<code>git merge --no-ff hotfix/1.2.1</code>
	<code>git branch -d hotfix/1.2.1</code>

Push the merged hotfix branch

gitflow	git
N/A	<code>git push origin master</code>
	<code>git push origin develop</code>
	<code>git push origin --tags</code>
	<code>git push origin :hotfix/1.2.1</code> <i>(if pushed)</i>

Independente de qual escolheu, haverá apenas uma visualização "bonitinha" de todo do fluxo. Essa:

```
* < d2ed106 (HEAD -> master, tag: v0.8, origin/master) Merged in development (pull request #10)
* | d075de4 (origin/development, development) Merged in hotfix/notifications-datetime (pull request #8)
* | \
* | db96275 Merged in bugfix/credentials-error (pull request #7)
* | *
* | 6809364 (bugfix/credentials-error) Changes configurations
* | 944fb57 Fixes files view error
* | \
* | aee935d Merge branch 'bugfix/uploads-not-shared' into development
* | *
* | 158c11a (bugfix/uploads-not-shared) Adds uploads folders to linked dirs
* | \
* | 707488e Add Capistrano and deploy settings
* | 3822524 (tag: v0.7.1) Merged in hotfix/notifications-datetime (pull request #9)
* | \
* | \
* | fe64d74 (hotfix/notifications-datetime) Adds correct time zone to application
* | \
* | \
* | 989ba4b (tag: v0.7) Merged in development (pull request #6)
* | \
* | \
* | 7833637 Fixes user avatar error
* | flccc00 (tag: v0.6) Merged in development (pull request #5)
* | \
* | \
* | 0297240 Adds uploaded files download methods
* | a651fe7 Change uploaded files location
* | dd6a426 Adds blank error message to multiple files upload
* | abefc1e Fixes password change error
* | 11b55c1 (tag: v0.5) Merged in development (pull request #4)
* | \
* | \
* | a60de77 Adds notifications create calls
* | e1a573d Adds notifications responses and methods
* | bf4beb0 Change templates to work with polymorphic relations
* | c759dc8 Adds id to user object in authentication token
* | db6f1bf Adds client block to suppliers and notifications
* | 7e9e6f4 Adds client block to stages
* | 1eeeb00 Adds client block to categories
* | ec06baf Adds update password and fixes client filter
* | 2d3164b Fixes attachments create bug
* | 48949e6 Fixes client faqs response filter
* | 014b798 Fixes client documents response filter
* | 7d6bebf Fixes client pictures response filter
ESC
```

```
< d2ed106 (HEAD -> master, tag: v0.8, origin/master) Merged in development (pull request #10)
```

```
* d075de4 (origin/development, development) Merged in hotfix/notifications-datetime (pull request #8)
```

```
* db96275 Merged in bugfix/credentials-error (pull request #7)
```

```
* 6809364 (bugfix/credentials-error) Changes configurations
```

```
* 944fb57 Fixes files view error
```

```
* aee935d Merge branch 'bugfix/uploads-not-shared' into development
```

```
* 158c11a (bugfix/uploads-not-shared) Adds uploads folders to linked dirs
```

```
* 707488e Add Capistrano and deploy settings
```

```
* 3822524 (tag: v0.7.1) Merged in hotfix/notifications-datetime (pull request #9)
```

```
* fe64d74 (hotfix/notifications-datetime) Adds correct time zone to application
```

```
* 90b54b (tag: v0.7) Merged in development (pull request #6)
```

```
* 783f1f1
```

```
* 0297240 Adds uploaded files download methods
```

```
* a651fe7 Change uploaded files location
```

```
* dd6a426 Adds blank error message to multiple files upload
```

```
* abefc1e Fixes password change error
```

```
* 11b55c1 (tag: v0.5) Merged in development (pull request #4)
```

```
* a60de77 Adds notifications create calls
```

```
* e1a573d Adds notifications responses and methods
```

```
* bf4beb0 Change templates to work with polymorphic relations
```

```
* c759dc8 Adds id to user object in authentication token
```

```
* db6f1bf Adds client block to suppliers and notifications
```

```
* 7e9e6f4 Adds client block to stages
```

```
* 1eeeb00 Adds client block to categories
```

```
* ec06baf Adds update password and fixes client filter
```

```
* 2d3164b Fixes attachments create bug
```

```
* 48949e6 Fixes client faqs response filter
```

```
* 014b798 Fixes client documents response filter
```

```
* 7d6bebf Fixes client pictures response filter
```

```
ESC
```

O comando é: `git log --all --decorate --oneline --graph`

Mas existe uma outra saída

Você não precisa necessariamente escolher entre a pílula azul ou vermelha. Pega as duas e manda pra dentro! Viva o roxo!

PS. Não é propaganda do NuBank...



- **GitKraken:** compatível com Windows, Linux e Mac. Esse garotão pode ser instalado em sua máquina e vai gerar uma interface muito legal para todos os seus repositórios locais. Além disso, se você tiver o git-flow cheatsheet instalado, ele irá automaticamente detectar isso e gerar uma aba de botões com os comandos dele. Dá uma olhada:
-

FileEditViewHelp

nn-back

master

Undo

Redo

Pull

Push

Branch

Stash

Pop

Glo

Viewing 16/16

Show All

Filter (Ctrl + Alt + f)

LOCAL5/5

REMOTE2/2

PULL REQUESTS0

TAGS9/9

SUBMODULES0

master+1

development

bugfix/credentials-...

bugfix/uploads-not-...

hotfix/notifications-...

v0.7.1

v0.7

v0.6

v0.5

v0.4

v0.3

v0.2

v0.1

Merged in development (pull request #10)5 days ago

Merged in hotfix/notifications-datetime (pull request #9)

Merged in hotfix/notifications-datetime (pull request #8)

Merged in bugfix/credentials-error (pull request #7)

Changes configurations

Fixes files view error

Merge branch 'bugfix/uploads-not-shared' into development

Adds uploads folders to linked dirs

Adds correct time zone to application

Add Capistrano and deploy settings6 days ago

Merged in development (pull request #6)

Fixes user avatar error

Merged in development (pull request #5)

Adds uploaded files download methods

Change uploaded files location

Adds blank error message to multiple files upload

Fixes password change error

Merged in development (pull request #4)a week ago

Adds notifications create calls

Adds notifications responses and methods

Change templates to work with polymorphic relations

Adds id to user object in authentication token

Adds client block to suppliers and notifications

Adds client block to stages

Adds client block to categories

Adds update password and fixes client filter

Fixes attachments create bug

Fixes client faqs response filter

commit: d2ed10

Merged in development (pull request #10)

Development

José Antonio

parent: 382252, d075de

authored 28/2/2019 @ 17:39

5 modified

3 added

12

PathTree

View all files

app/controllers/application_controller.rb

app/uploaders/application_uploader.rb

Capfile

config/deploy.rb

config/deploy/staging.rb

config/initializers/knock.rb

Gemfile

Gemfile.lock

Update available (4.2.2)

deb

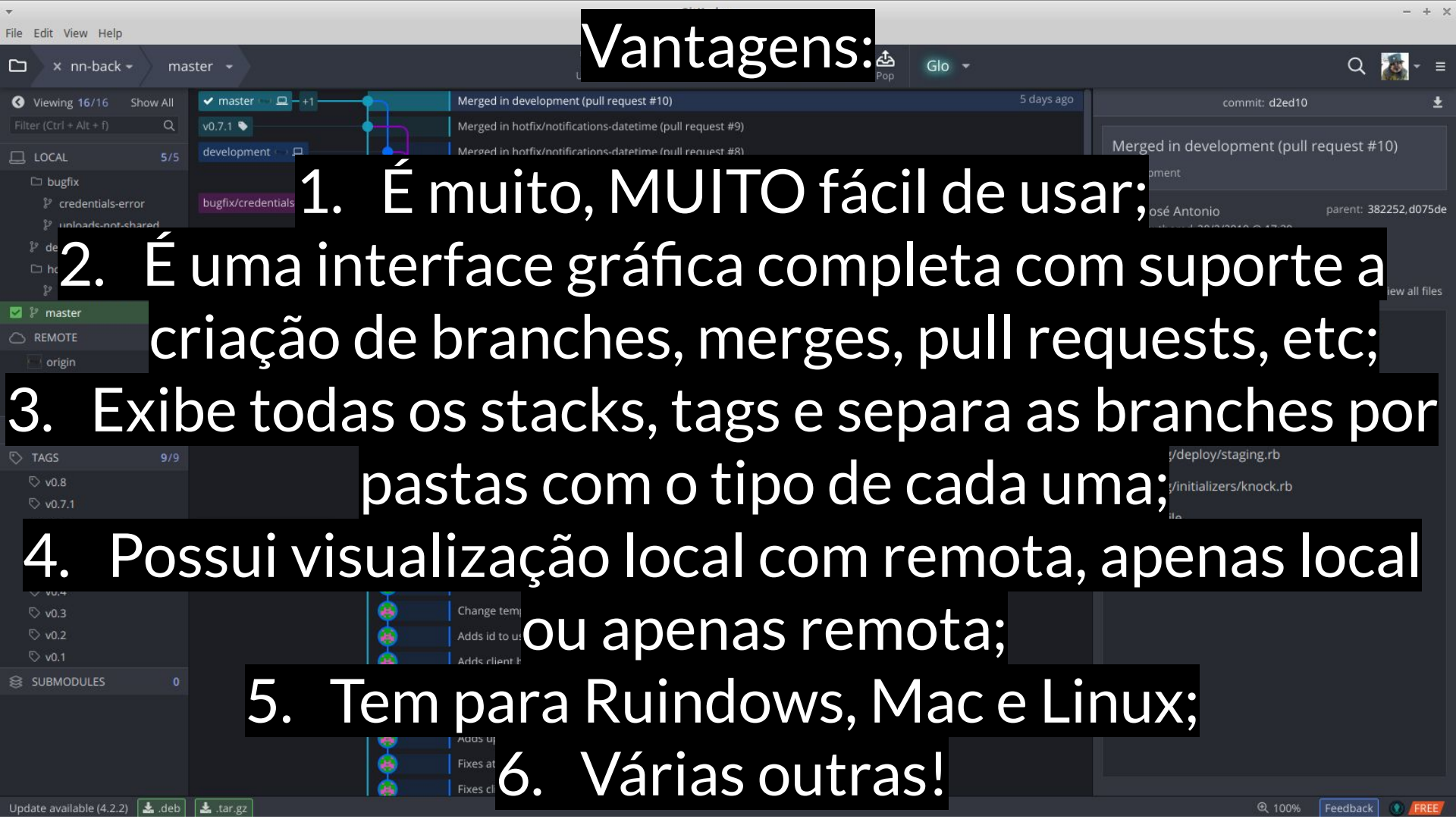
tar.gz

100%

Feedback

FREE

Vantagens:



1. É muito, MUITO fácil de usar;

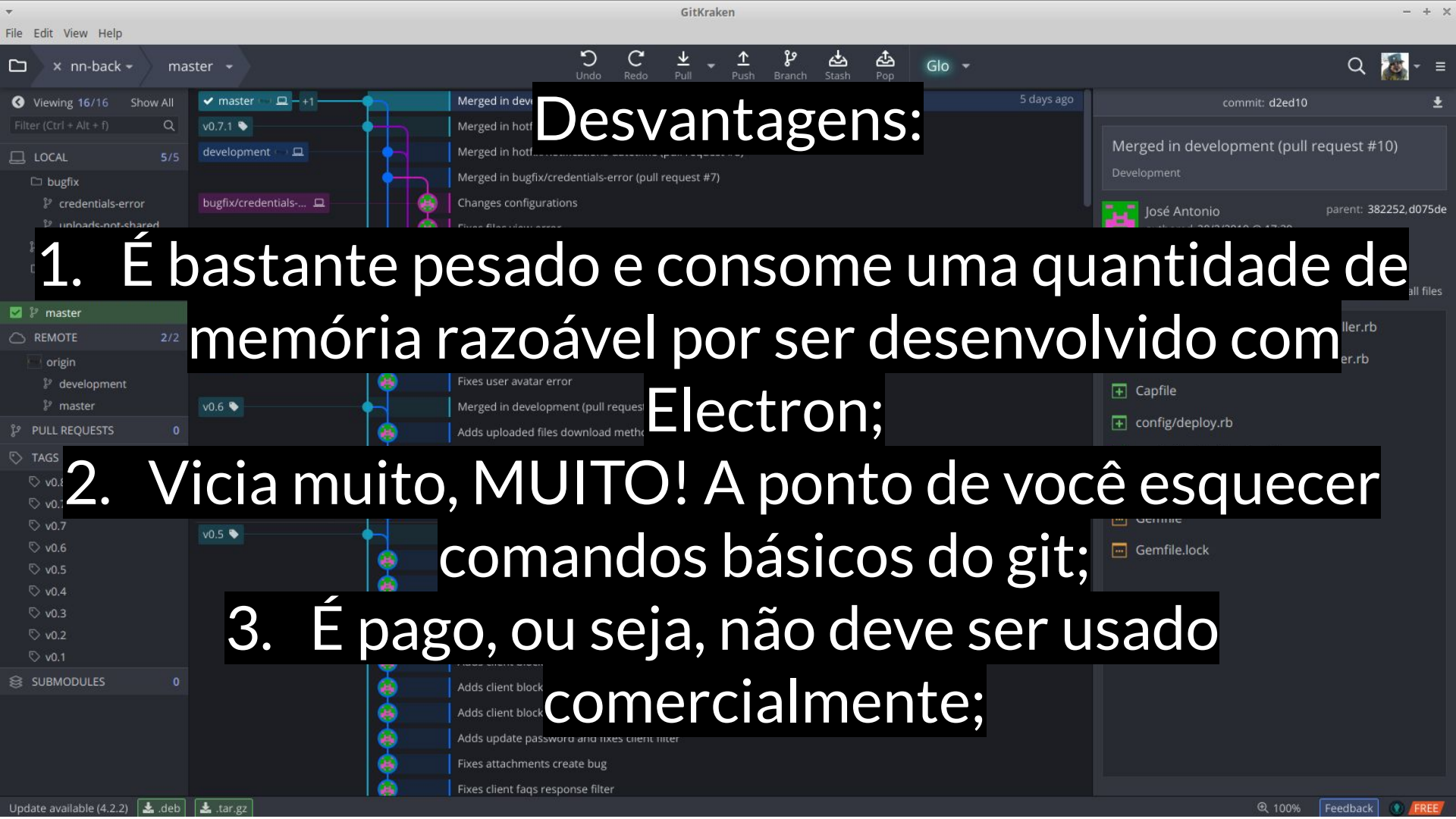
2. É uma interface gráfica completa com suporte a criação de branches, merges, pull requests, etc;

3. Exibe todas os stacks, tags e separa as branches por pastas com o tipo de cada uma;

4. Possui visualização local com remota, apenas local ou apenas remota;

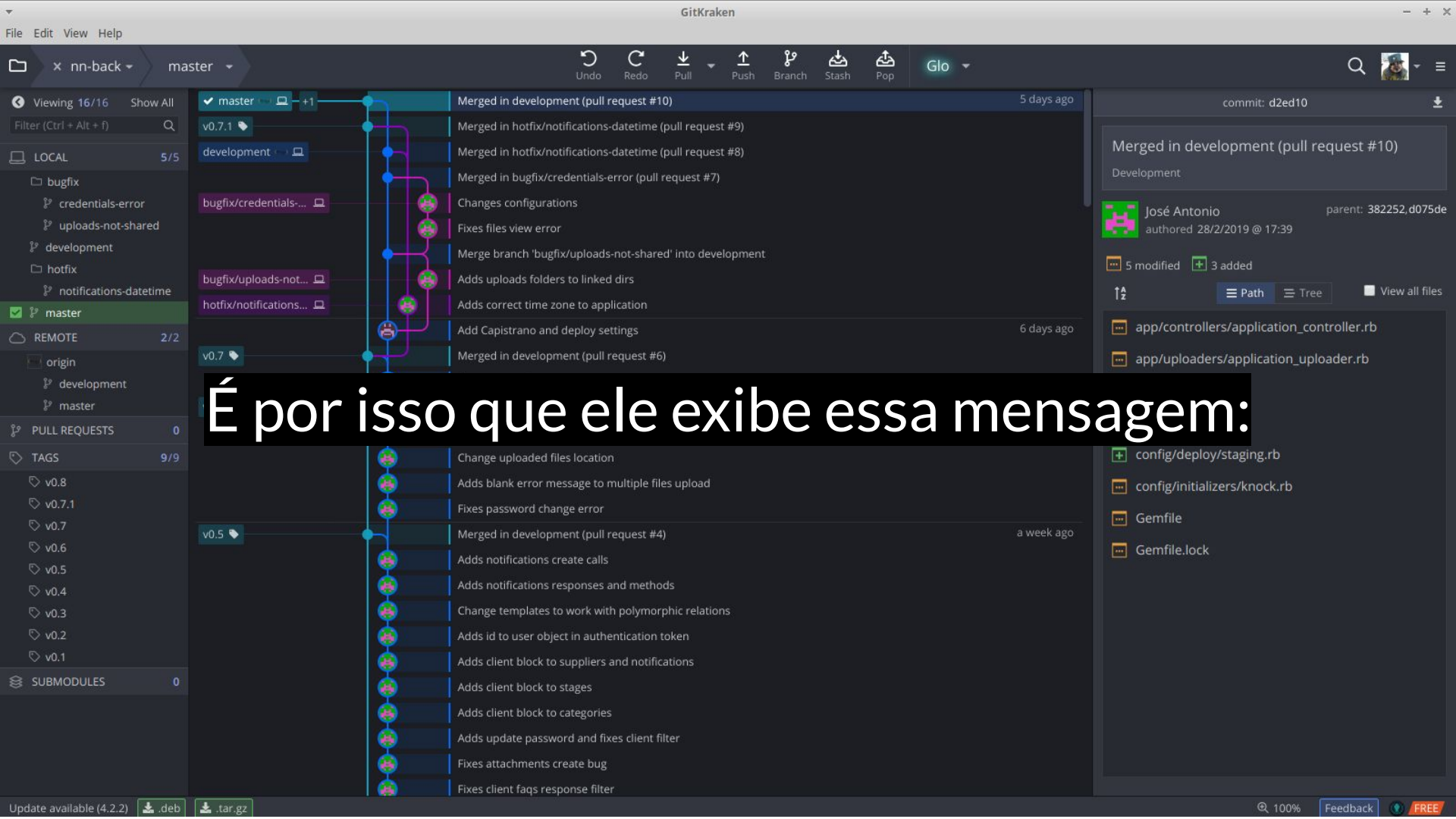
5. Tem para Ruindows, Mac e Linux;

6. Várias outras!

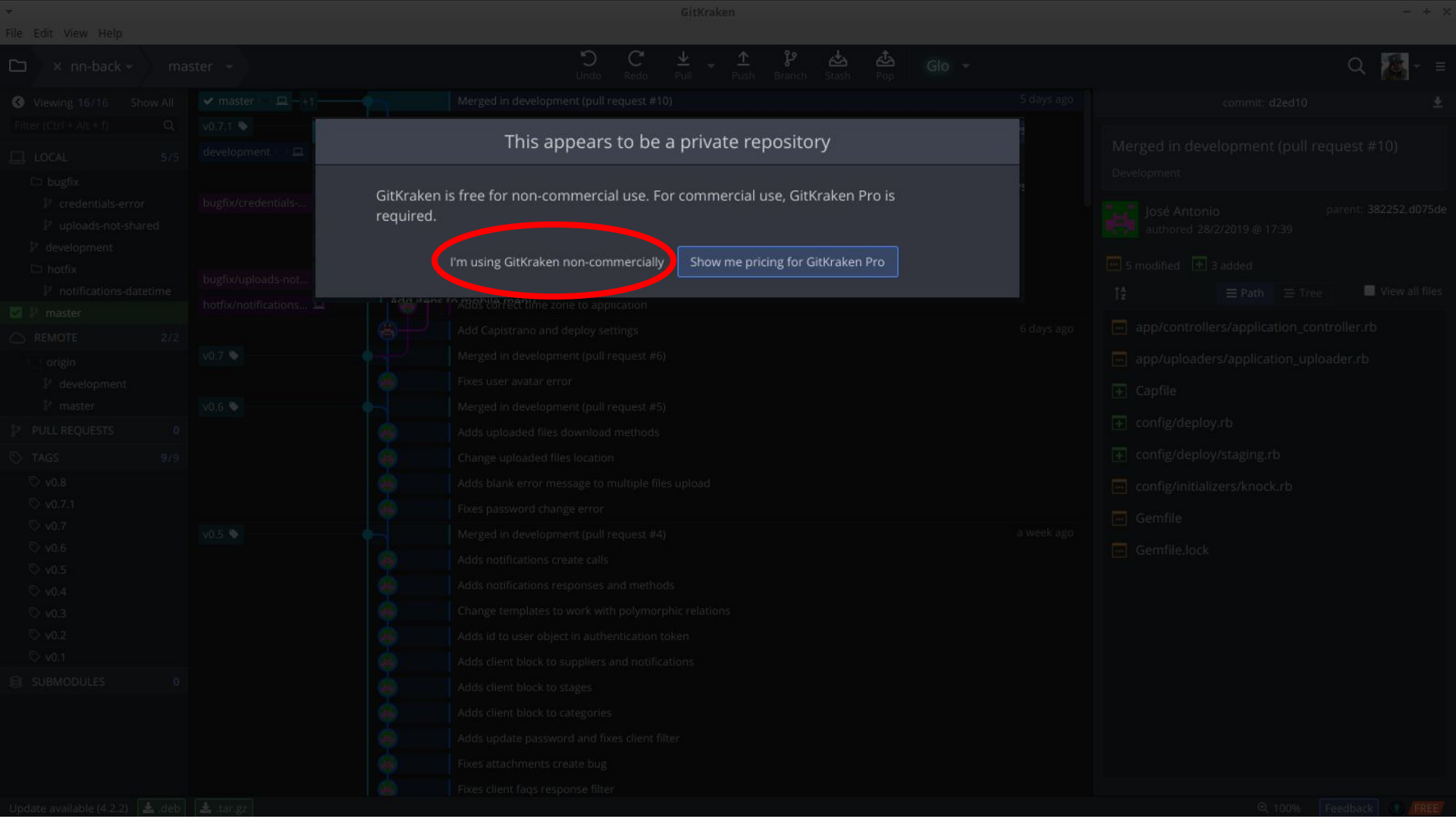


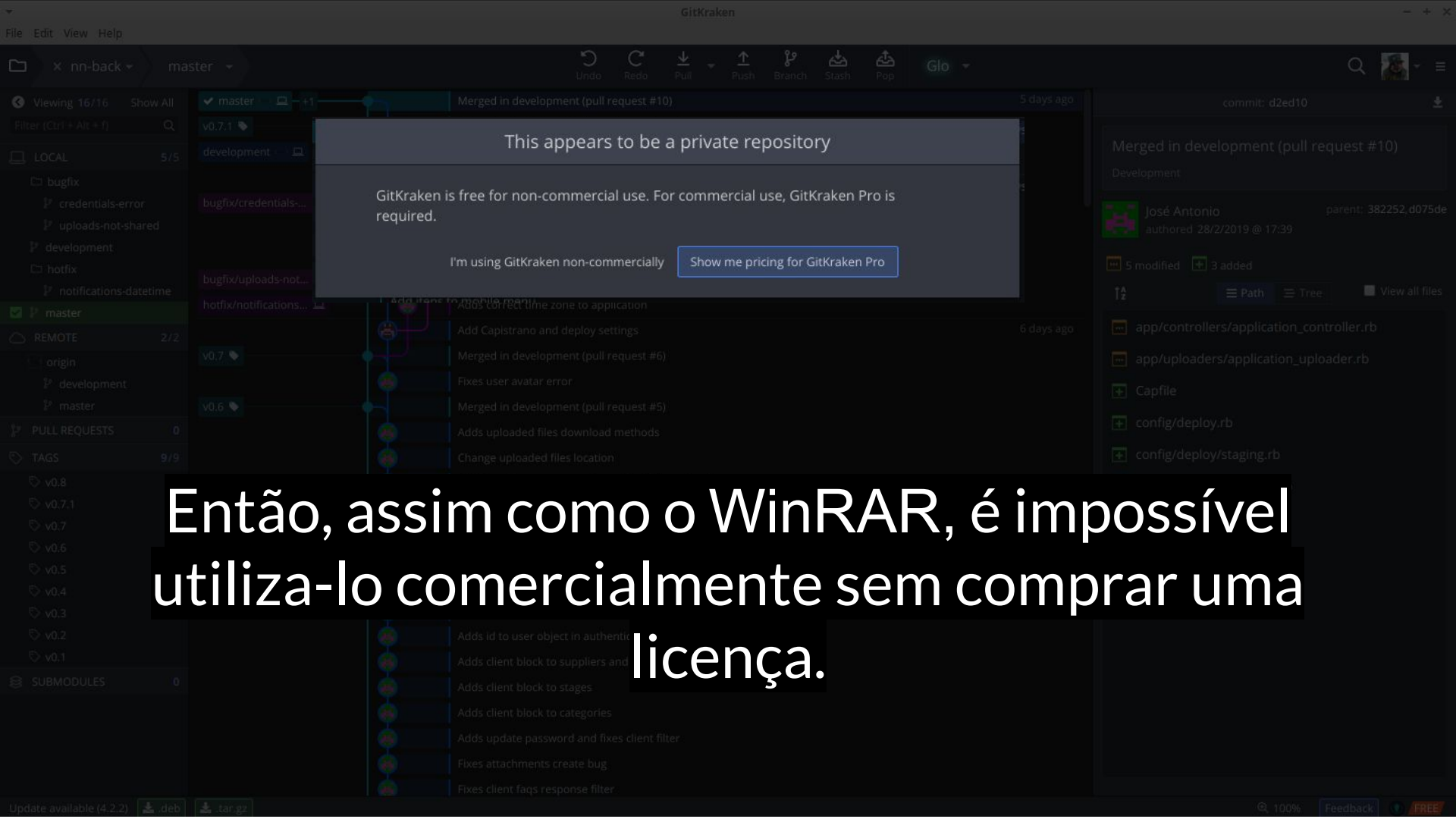
Desvantagens:

1. É bastante pesado e consome uma quantidade de memória razoável por ser desenvolvido com Electron;
2. Vicia muito, MUITO! A ponto de você esquecer comandos básicos do git;
3. É pago, ou seja, não deve ser usado comercialmente;



É por isso que ele exibe essa mensagem:





Então, assim como o WinRAR, é impossível utiliza-lo comercialmente sem comprar uma licença.

COMMITTS

Como eles devem ser escritos!

As 7 regras para um commit:

1. Separe o assunto do corpo com uma linha em branco
 2. Limite o assunto para 50 caracteres
 3. Capitalize a linha do assunto
 4. Não termine a linha do assunto com um ponto
 5. Use o modo imperativo na linha do assunto
 6. Limite o corpo para 72 caracteres
 7. Use o corpo para explicar o o quê e por quê vs como
-

Do this:

- Capitalize a linha do assunto
 - Accelerate to 88 miles per hour
- Não termine com um ponto
 - Open the pod bay doors
- Use o imperativo
 - Update getting started documentation
-

Don't do this:

- Capitalize a linha do assunto
 - accelerate to 88 miles per hour
- Não termine com um ponto
 - Open the pod bay doors.
- Use o imperativo
 - Changing behavior of X
-

Uma boa mensagem de commit deve sempre encaixar na frase “**Se aplicado, este commit ...**”

Por exemplo:

Se aplicado, este commit **refatora o subsistema X**

Se aplicado, este commit **atualiza a documentação de como configurar**

Se aplicado, este commit **remove métodos depreciados**

Se aplicado, este commit **lança versão 1.0.0**

If applied, this commit will **merge pull request #123 from user/branch**

Não funciona para formas não imperativas:

Se aplicado, este commit **corrigido bug com Y**

Se aplicado, este commit **mudando o comportamento de X**

Se aplicado, este commit **mais correções da coisa quebrada**

Se aplicado, este commit **novos métodos da API**

Finalizando

- Lembre-se: é errando que se aprende. Ninguém vai te matar por ter aberto um pull request de uma hotfix para a develop.
- Treinar é tudo de bom! Tente utilizar na faculdade, projetos pessoais, ou simplesmente treine em casa modificando um simples arquivo em um repositório local.

Obrigado!

