



IOS DOJO

//01

Git-flow

Fellipe Bravo
iOS Developer

Outubro/2020



Introdução	03
Branches	05
Features	08
Releases	10
Hotfixes	12
Mundo Ideal	14
Let's do it!	16
Referências e Contato	17

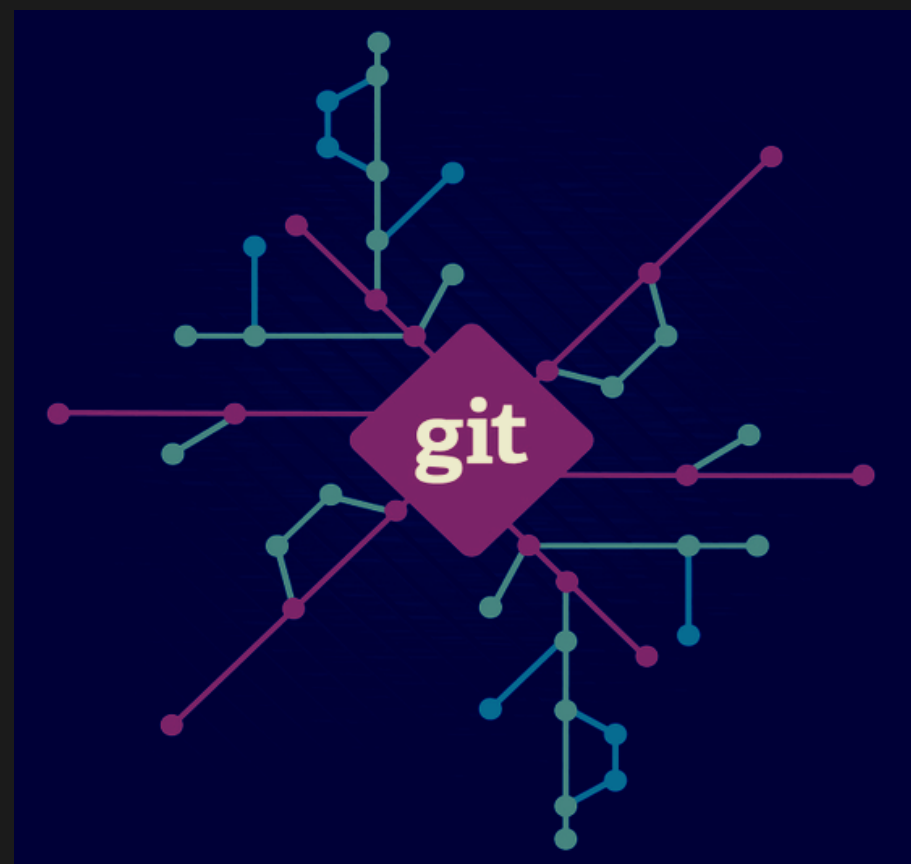
Sumário



IOS DOJO

Introdução

//03



Introdução

Essa apresentação sobre o Git-flow tem o intuito de mostrar um padrão de utilização do git para gerenciar de forma menos complicada as features do projeto, gestão de releases e até mesmo fixes em produção.

Apenas os pontos principais de atuação do git-flow serão mostrados, para maiores informações veja a referência.



IOS DOJO

Branches

//05

DEVELOP

A branch principal do time que possui algumas features desenvolvidas porém que não estão prontas para virarem uma versão estável para produção.



MASTER

A branch intocável. Possui apenas o código estável e nossa cópia do que está em produção hoje. Nenhum desenvolvedor tem o poder de commitar nela e editá-la.



FEATURE

Toda e qualquer implementação nova ou modificação de alguma funcionalidade existente que não seja uma correção de bug.



RELEASE

As releases são versões estáveis do código que possuem diversas features implementadas. Sempre são iniciadas a partir de uma versão da branch develop e não necessariamente serão mergeadas.



HOTFIX

Temos um bug de correção emergencial em produção. Eis que surge o hotfix para nos salvar.

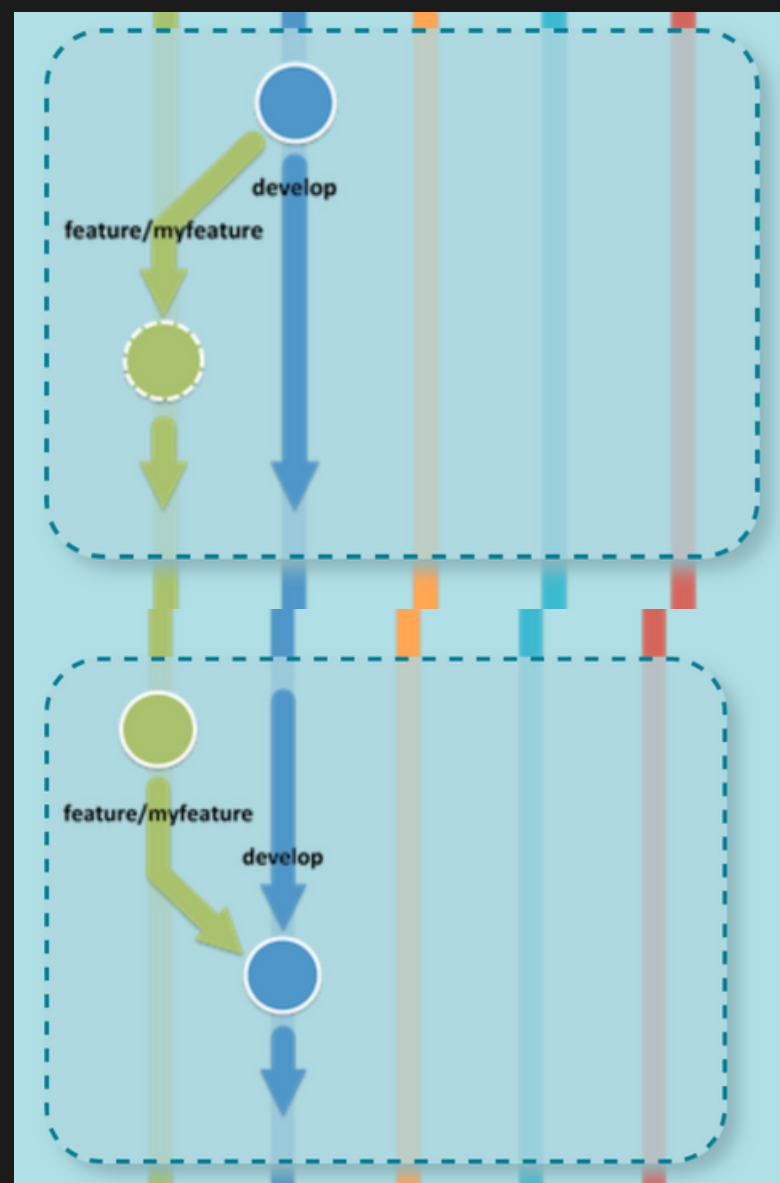




IOS DOJO

Features

//08



Features

As features são as novas funcionalidades desenvolvidas que partem de uma versão em desenvolvimento da branch `develop` e retornam para a mesma.

O seu ciclo de vida normalmente dura por uma sprint inteira ou até mais de uma sprint se for uma feature muito grande.

A equipe pode desenvolver em conjunto em cima de uma mesma feature.



IOS DOJO

Releases

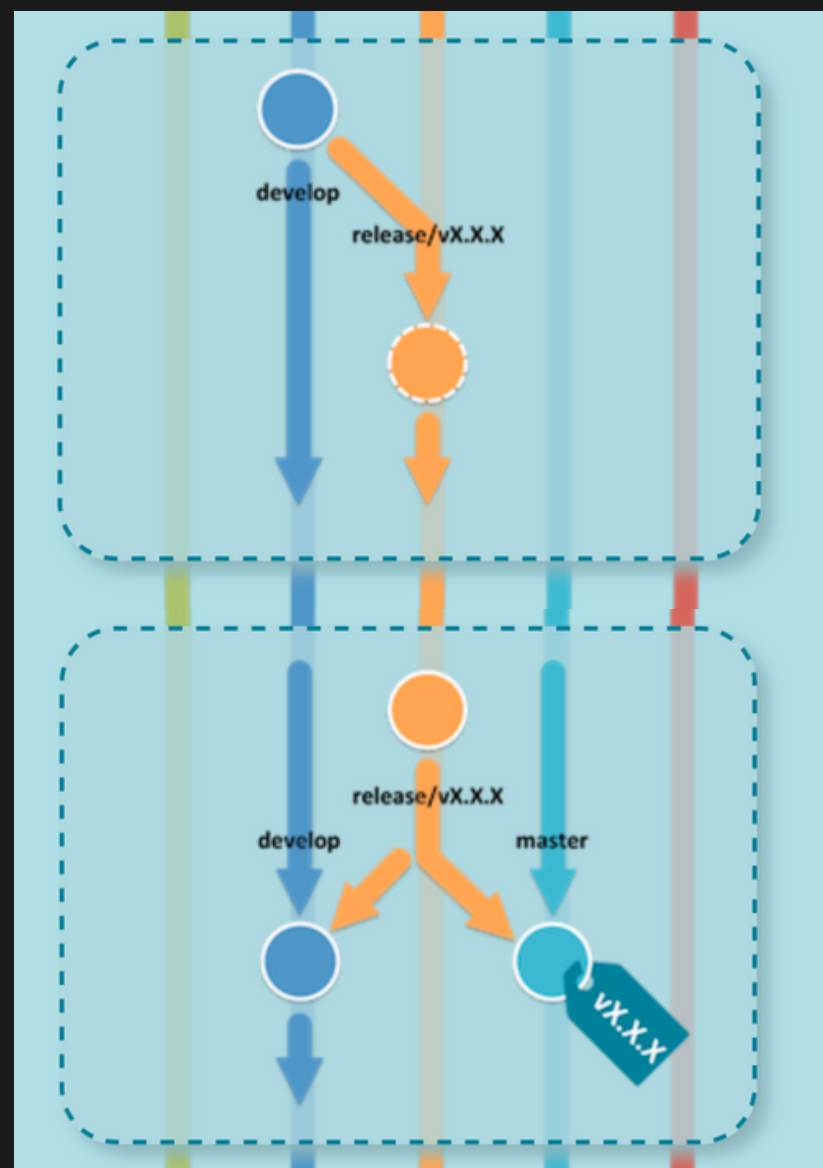
//10

Releases

As releases são as versões estáveis de um conjunto de features mergeadas na develop.

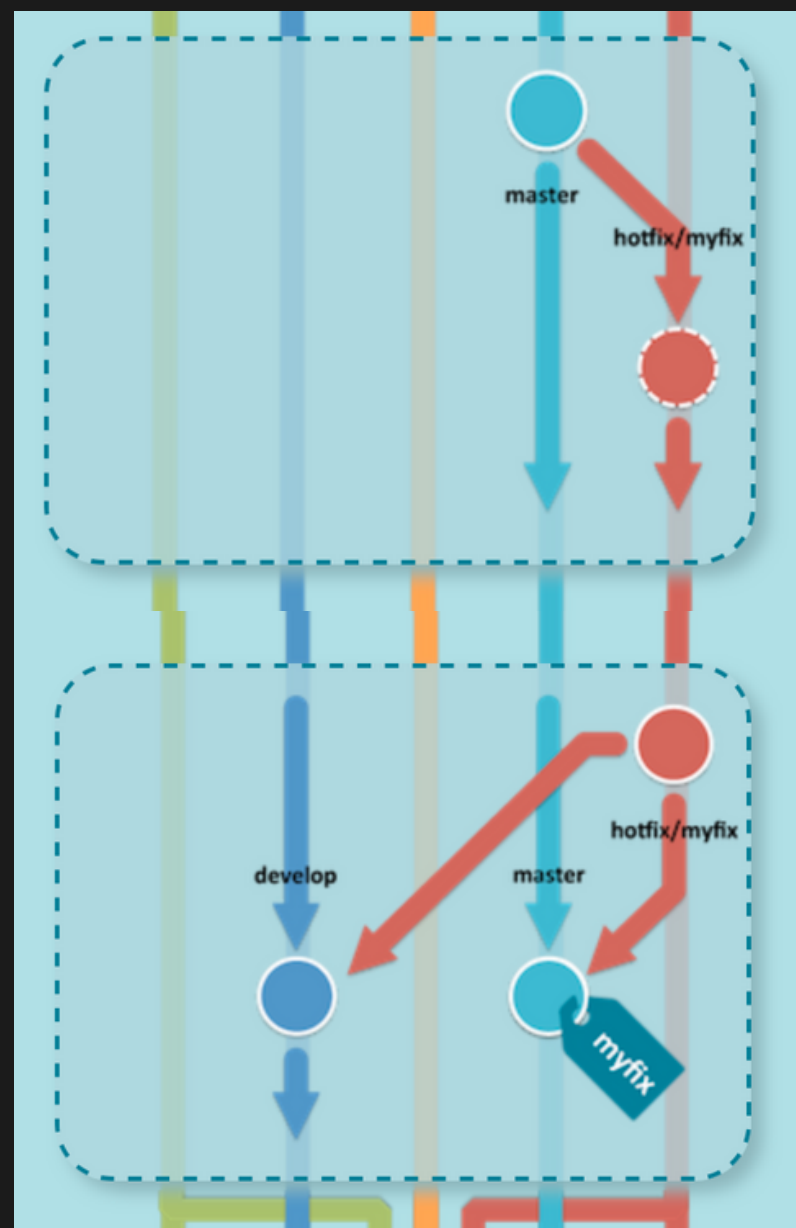
Normalmente podem ser consideradas Release Candidates (RC) para uma subida em homologação, Seu ciclo de vida é mais curto que o da feature, ou a versão está pronta para produção, ou não.

No momento em que uma release é finalizada, um merge na **develop** e na **master** são feitos. Garantindo, respectivamente, que todos que iniciarem features terão a última versão estável do projeto e que exista um 'backup' estável e taggeado.





Hotfixes



//13

Hotfixes

Somos humanos e estamos todos sujeitos ao erro. Subimos para produção uma versão que está com um bug terrível, um crash que seja. Aí entram as hotfixes para garantir que a correção será aplicada tanto no código 'ex-estável' da **master** e da **develop**.

Seu ciclo de vida é extremamente curto: dado que a correção foi feita, essa branch será mergeada em ambas branches principais e fim. Novamente taggeamos a querida **master**.



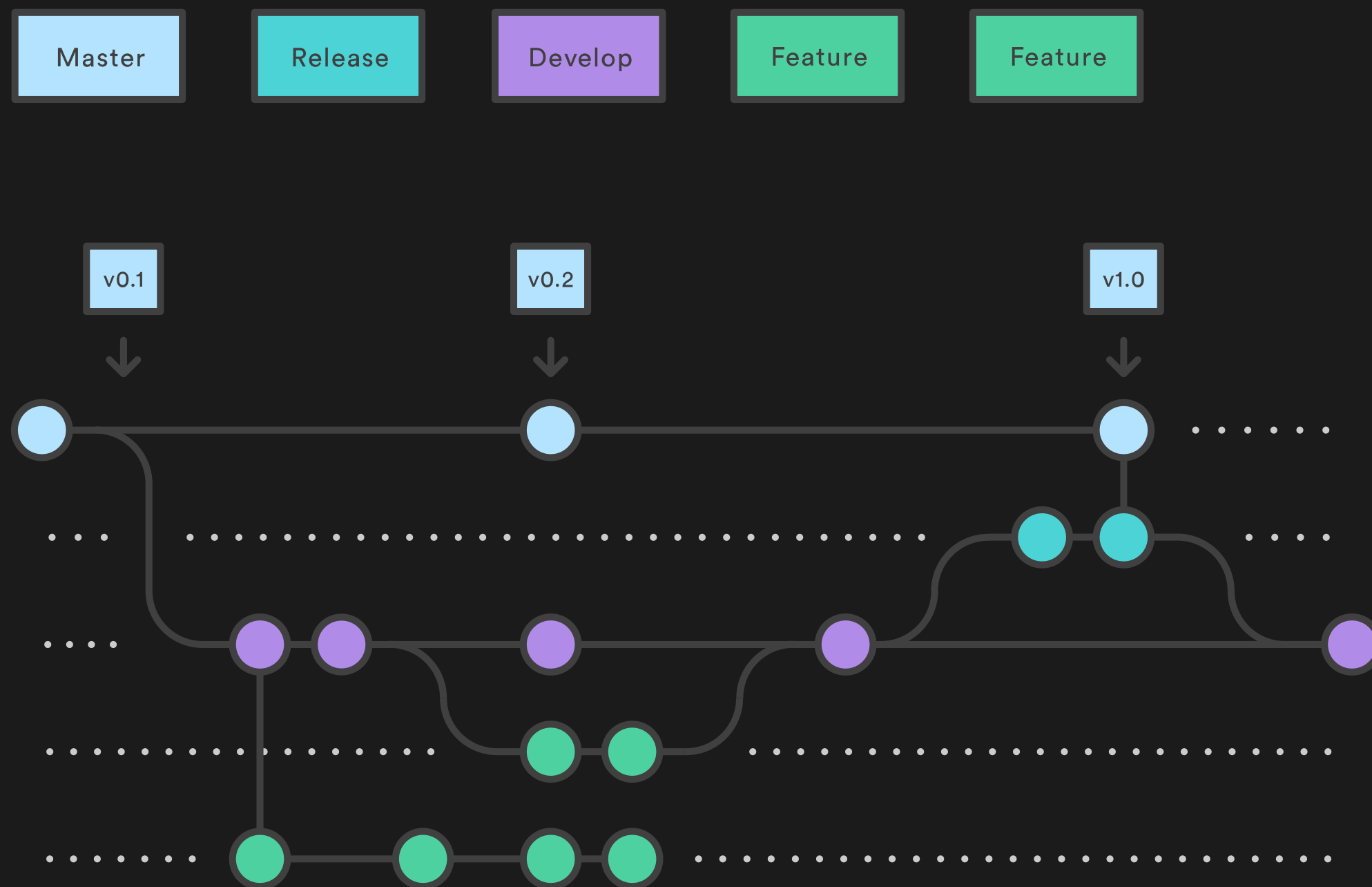


Mundo Ideal



IOS DOJO

Mundo Ideal



//15



IOS DOJO

Let's do it!

Acessem: <https://github.com/insidemymybrain/git-flow-dojo>

//16

//17

PERGUNTAS E
COMENTÁRIOS

Referências e Contato

CHEATSHEET DO GIT-FLOW, [HTTP://DANIELKUMMER.GITHUB.IO/GIT-FLOW-CHEATSHEET/INDEX.PT_BR.HTML](http://danielkummer.github.io/git-flow-cheatsheet/index.pt_br.html)

A SUCCESSFUL GIT BRANCHING MODEL, [HTTPS://NVIE.COM/POSTS/A-SUCCESSFUL-GIT-BRANCHING-MODEL/](https://nvie.com/posts/a-successful-git-branching-model/)

GITFLOW WORKFLOW, [HTTPS://WWW.ATLASSIAN.COM/GIT/TUTORIALS/COMPARING-WORKFLOWS/GITFLOW-WORKFLOW](https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow)

INSTALAÇÃO GIT-FLOW NO TERMINAL, [HTTPS://GITHUB.COM/PETERVANDERDOES/GITFLOW-AVH/WIKI/INSTALLATION](https://github.com/petervanderdoes/gitflow-avh/wiki/Installation)

WORKPLACE

Fellipe Bravo

LINKEDIN

fellipe.bravo

E-MAIL

fellipe.pimente@b2wdigital.com