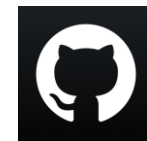


Git e Github

Controlador de versão e projetos



Ementa

- ▶ O que é Git e Github
- ▶ Configuração do Git
- ▶ Criando e clonando repositórios
- ▶ Fazendo fork de repositórios
- ▶ Trabalhando com branches
- ▶ Criando snapshots
- ▶ Merge de branches
- ▶ Git flow
- ▶ SourceTree

O que é o Git e Github

- ▶ **Git:** é um sistema de controle de versão moderno mais utilizado hoje em dia. É um projeto de código aberto desenvolvido em 2005 por Linus Torvalds (Linux).
- ▶ **Github:** é considerado uma ferramenta essencial para engenheiros de software, com uma popularidade sem igual. Atualmente, ele acomoda mais de 25 milhões de usuários. Em suma, o Github é um serviço baseado em nuvem que hospeda um sistema de controle de versão chamado Git, permitindo que os desenvolvedores colaborem e façam mudanças em projetos compartilhados enquanto mantêm um registro detalhado do seu progresso.

Conceitos mais utilizados


- ▶ Repository
- ▶ Branches
- ▶ Commits
- ▶ Pull Request
- ▶ Merge
- ▶ Tags
- ▶ Reset
- ▶ Stash
- ▶ Fetch

Repository


- ▶ É um diretório onde os arquivos do seu projeto ficam armazenados. Ele pode ficar em um depósito do Github ou em seu computador. Você pode armazenar códigos, imagens, áudios, ou qualquer coisa relacionada ao projeto no diretório.

Repository

► Ex.: <https://github.com/bittencourtthulio>

 Product ▾ Solutions ▾ Open Source ▾ Pricing


Search / Sign in Sign up




Thulio Bittencourt
bittencourtthulio

Follow

515 followers · 1 following

 thuliobittencourt

 <http://www.thuliobittencourt.com>

Overview Repositories 31 Projects Packages Stars 7

Popular repositories

TBGConnection Public

Componentes que criam uma camada de abstração para os drivers de conexão do Delphi e permitem que você crie projeto completamente independente do driver de conexão escolhido, com baixo acoplamento,...

Pascal ☆ 31 🍴 18

TBGFirebaseConnection Public

Componente para Conexão e Operações com o Google Firebase

Pascal ☆ 23 🍴 15

ProjectHorseSimpleORM Public

Pascal ☆ 20 🍴 10

BlogFontes Public

Fontes dos Artigos do Blog

Pascal ☆ 19 🍴 27

DelphiTips Public

Pascal ☆ 15 🍴 11

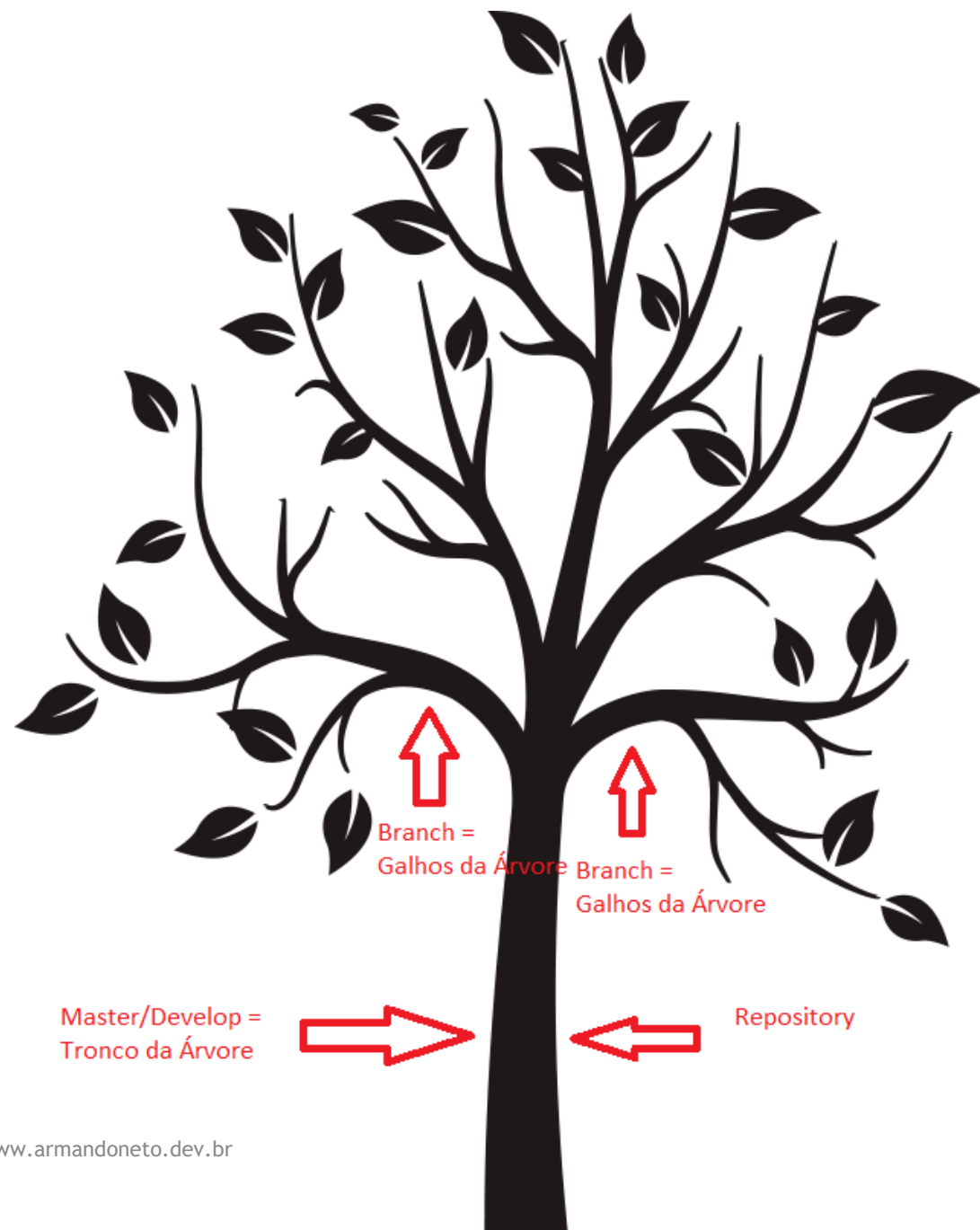
translator4d Public

Pascal ☆ 14 🍴 6

1 contribution in the last year

Branches

- ▶ Branches são ramificações, você gera versões diferentes de um repositório. Quando você modifica o projeto nas branches de recursos, um desenvolvedor pode ver como isso vai afetar o projeto principal na hora que tudo for integrado.



Commits

- ▶ Os Commits é como as mudanças salvas no GitHub são chamadas. Cada vez que o desenvolvedor muda o arquivo do branch de recurso, ele terá que executar um Commit para mantê-lo.

Pull Request

- ▶ Para propor as mudanças que você acabou de fazer para outros desenvolvedores trabalhando no mesmo projeto, você deve criar um Pull Request. São eles que fazem ser tão fácil de trabalhar junto em projetos, já que eles são a principal ferramenta de colaboração no GitHub.
- ▶ Pull Request permitem que você veja as diferenças entre o projeto original e o seu branch de recurso. É assim que você pede para os seus pares revisá-las. Se os outros desenvolvedores aprovarem as modificações, eles podem executar um Merge Pull Request (solicitação de mesclagem). Isso irá aplicar as mudanças para o projeto principal.

Merge

- ▶ O comando `git merge` é usado para mesclar uma ramificação no ramo ativo:

`git merge <branch-name>`

Tags

- ▶ A marcação (tag) é usada para marcar compromissos específicos, exemplo: versão liberada, versão específica - Normalmente tags não sofrem alterações, são versões fechadas - backups

git tag 1.1.0 <insert-commitID-Here>

Reset

- ▶ Para redefinir o índice e o diretório de trabalho para o estado do último commit:

git reset --hard HEAD

Stash

- ▶ Provavelmente um dos menos conhecidos comandos do git básico que ajuda a salvar as mudanças que não devem ser cometidos imediatamente, mas em uma base temporária:

git stash

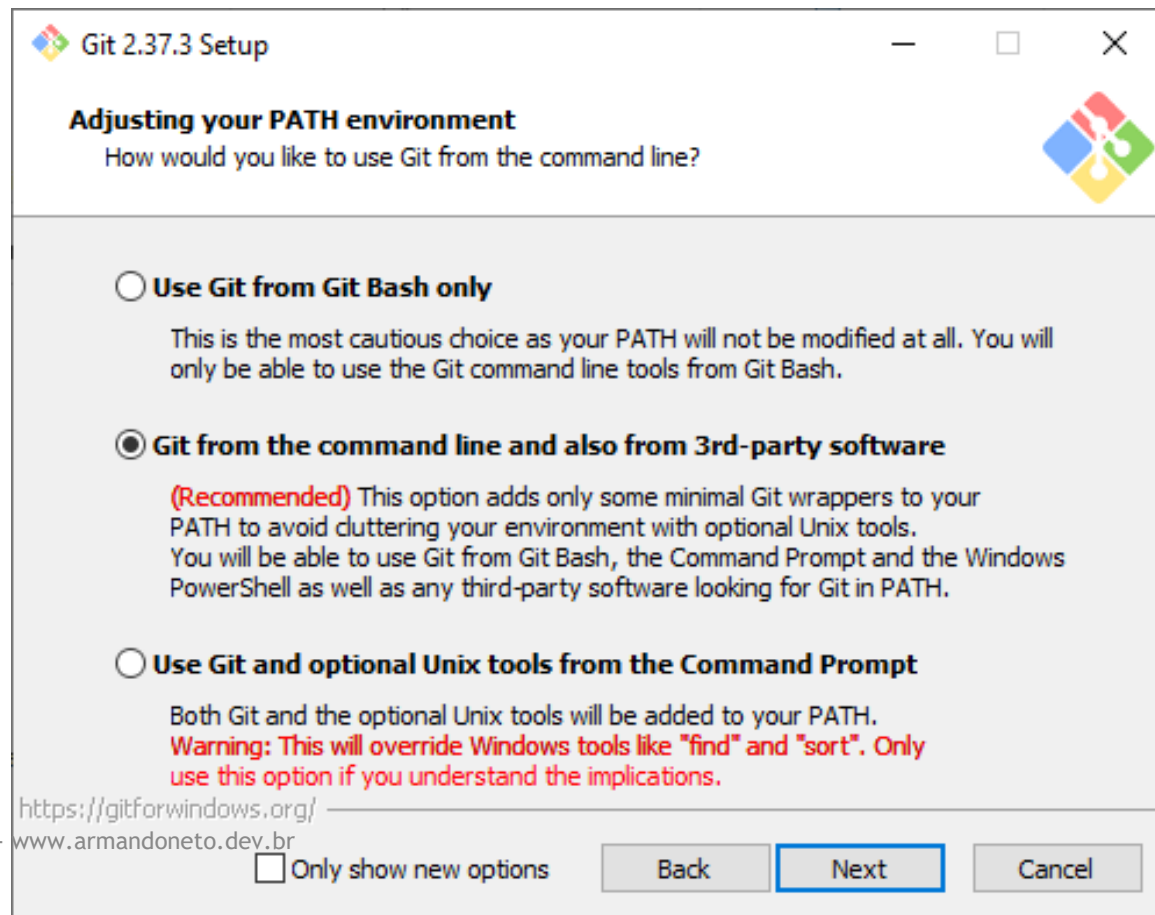
Fetch

- ▶ Permite que um usuário obtenha todos os objetos do repositório remoto que atualmente não residem no diretório de trabalho local:

git fetch origin

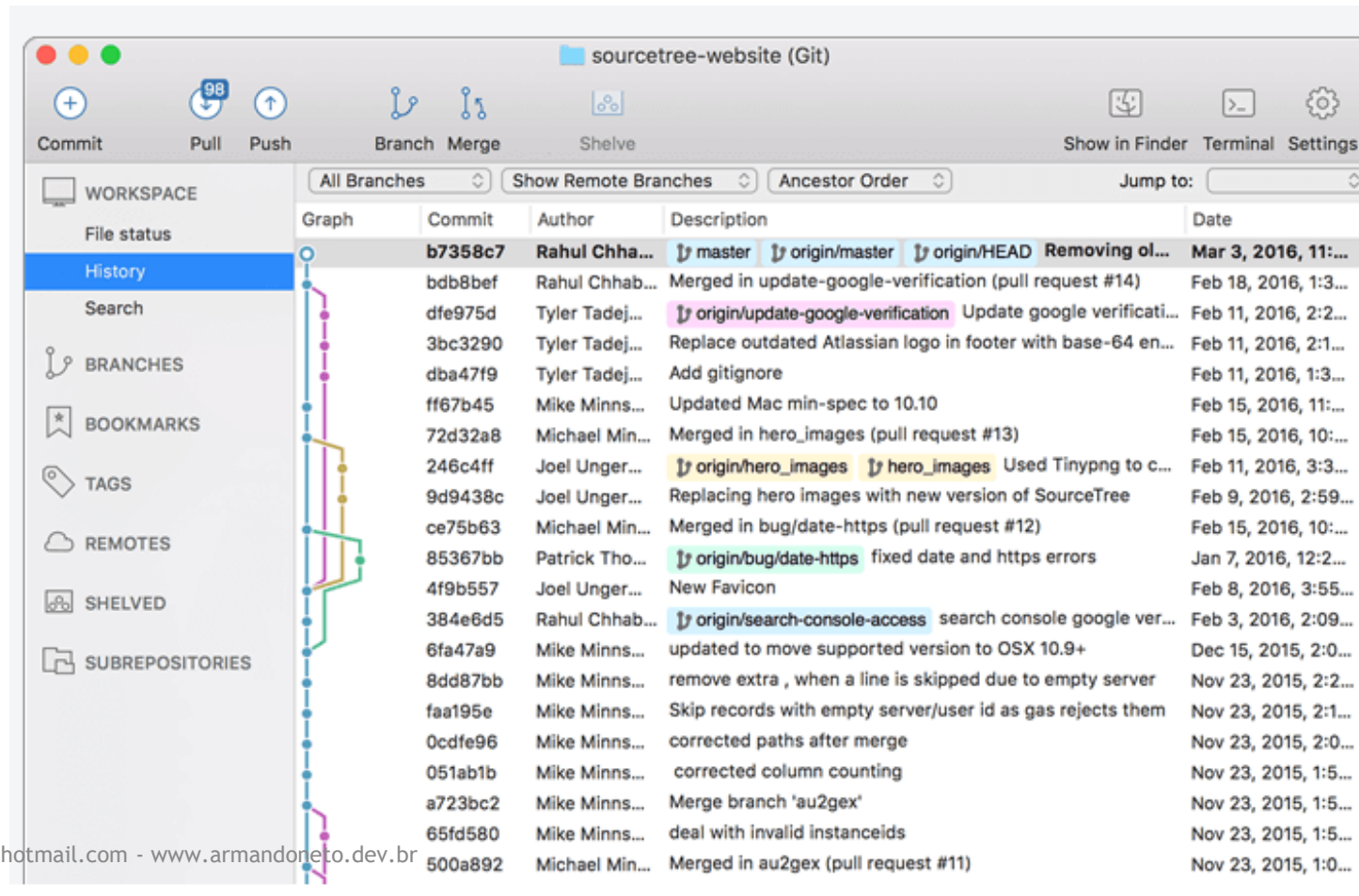
Configuração do Git

► <https://git-scm.com/downloads>





SourceTree

► <https://www.sourcetreeapp.com/>



Criando e clonando repositórios


 Search or jump to...  [Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *


Repository name *

 armandocorrea ▾


 /

Great repository names are short and memorable. Need inspiration? How about [ubiquitous-octo-disco?](#)

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

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

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

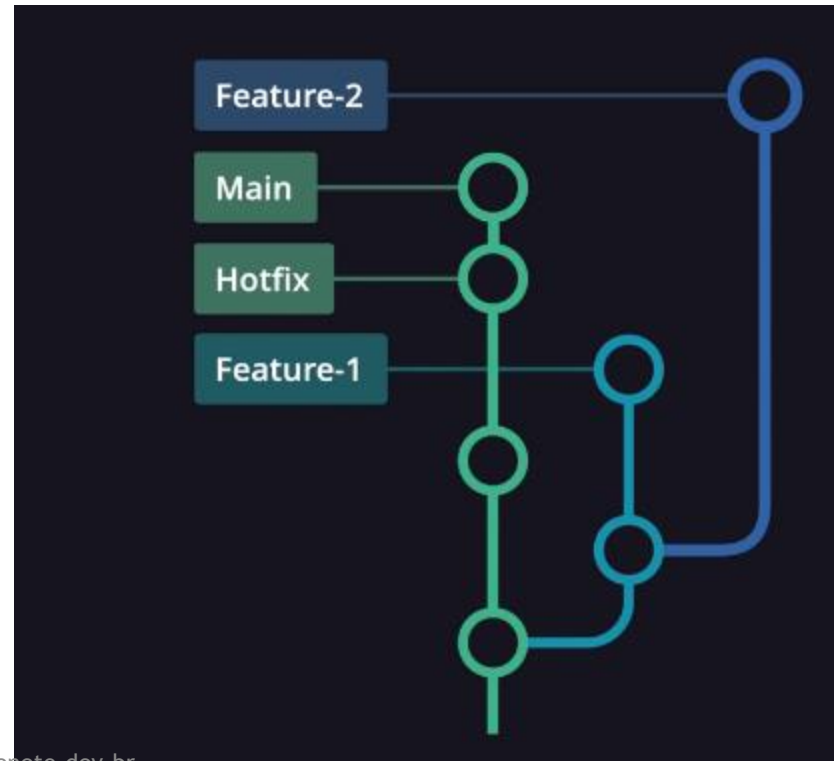
.gitignore template: None ▾

Fazendo fork de repositórios

- ▶ Uma bifurcação é uma cópia de um repositório que você gerencia. As bifurcações permitem que você faça alterações em um projeto sem afetar o repositório original. Você pode buscar atualizações ou enviar alterações para o repositório original com pull requests.

Trabalhando com branches

- ▶ São cópias de trabalho isoladas de um determinado ponto de alteração do branch alvo

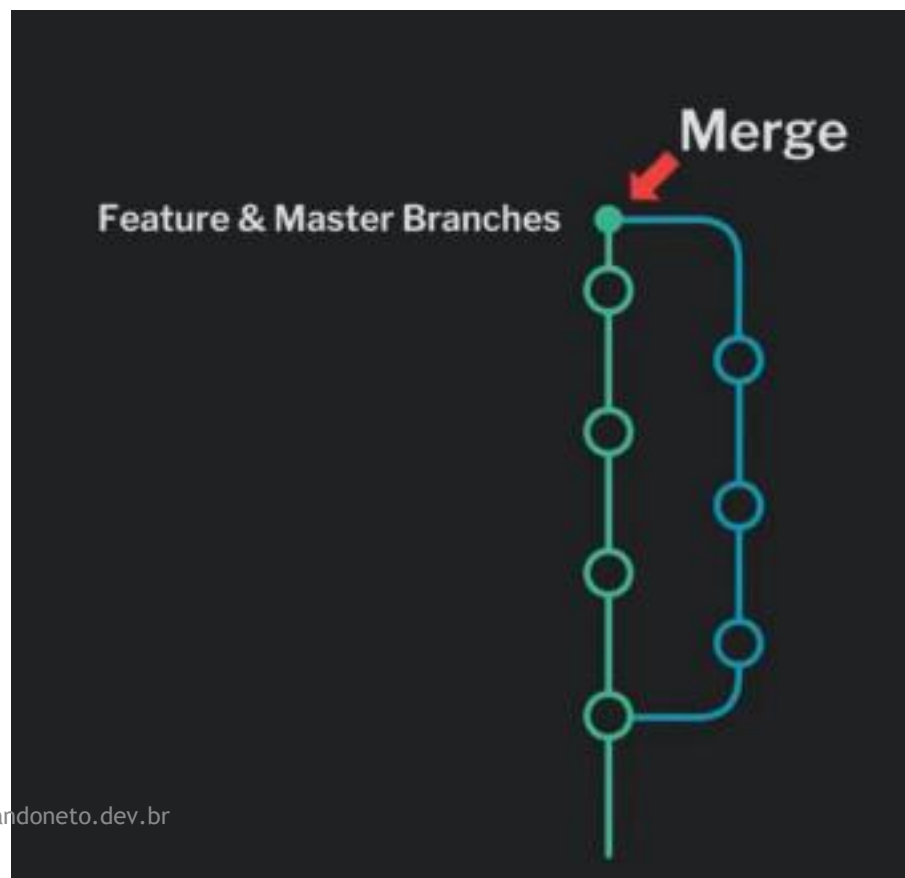


Criando snapshots

- ▶ O instantâneo está para um repositório como a captura de tela está para um vídeo. É o conteúdo (arquivos e pastas) de um repositório em algum momento. Ao fazer commit, você salva o diretório de trabalho atual do seu repositório como um novo snapshot (commit = snapshot + metadata).

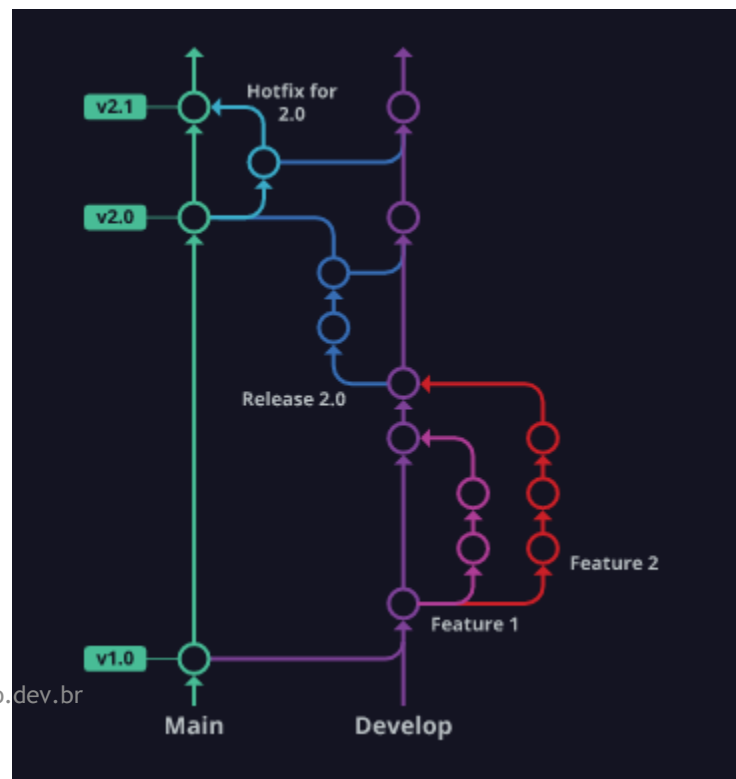
Merge de branches

- ▶ Ao trabalhar no Git, a ação de mesclagem é usada para combinar alterações de uma ramificação para outra, como `Git merge to master`.



Git flow

- ▶ Git flow é uma estratégia popular de ramificação do Git destinada a simplificar o gerenciamento de versões e foi introduzida pelo desenvolvedor de software Vincent Driessen em 2010. Fundamentalmente, o Git flow envolve isolar seu trabalho em diferentes tipos de ramificações do Git



Exercícios

Versionar os projetos do +Devs2Blu

Exercícios

Formar duplas e um evoluir o programa de outro dev