

GitFlow

Uma forma de **organizar nossos branches** é **através do GitFlow**. Esse modelo foi sugerido por Vicent Driessen (<http://nvie.com/posts/a-successful-git-branching-model/>) em **2010** e é bem popular devido a uma extensão que **leva o mesmo nome**. A grande vantagem nessa extensão é a padronização na forma que a equipe irá **trabalhar no repositório**. O uso da extensão é menos propenso a falhas pois a equipe usa comandos mais amigáveis para manipular, criar e combinar as branches.

Esse modelo resulta na seguinte organização de branches:



Master: É a branch estável do projeto. Nela é onde temos o código pronto para ser enviado para produção e nela é que vão as tags do projeto.

Develop: A branch develop é baseada no master e a partir dela são criadas as novas funcionalidades do sistema, que são as branches de features.

Feature: É aqui onde as novas funcionalidades são criadas e depois são combinadas com develop.

Release: Após uma série de Features serem entregues, existe um passo de release antes de combinar o conteúdo de develop com a master. Alguns commits de preparação podem ser criados nessa etapa para garantir a qualidade do código gerado. Após finalizado, essas alterações são combinadas de volta em ambas as branches develop e master, e uma tag de release é criada automaticamente na master.

Hotfix: Ao notar um bug em produção, o processo de correção geralmente acontece de maneira segregada das features convencionais. Um hotfix é criado

a partir do master e combinado devolta em dois sentidos com o master, para garantir que o bug seja resolvido, e com o develop, para assim evitar uma possível regressão do bug.

Mais detalhes sobre o fluxo dessas estratégias e os comandos para executar essas ações podem ser consultados nessa página (http://danielkummer.github.io/git-flow-cheatsheet/index.pt_BR.html).