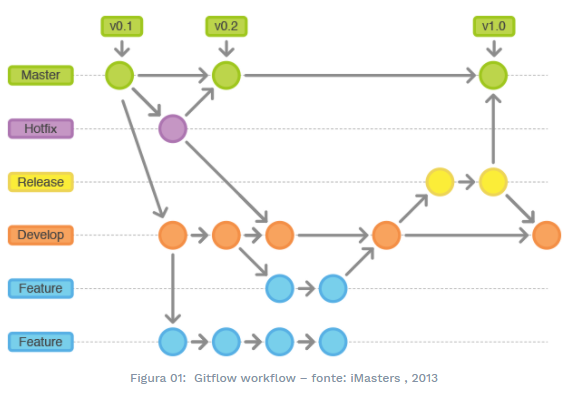
# GitFlow

Quando falamos de desenvolvimento, onde temos uma equipe atuando, seja um time de desnvolvimento ou time de dados ou até mesmo um time de infraestrutura, atuando sobre o mesmo código precisamos de um gerenciamento, pois temos várias etapas como requisitos do sistema, arquitetura, padrões de codificação, gerenciamento de configuração, deploy, etc. Além disso, o código está sempre em constante mudanças e com isso surge a necessidade de controlar tudo isso, e uma das formas é utilizando o GitFlow.

Se imaginarmos vários times trabalhando de forma concorrente e algumas vezes em tarefas conflitantes que exigem uma boa gestão do código para não se perder trabalho já feito ou problemas na consolidação de uma versão aceitável do código, o Gitflow, que é um framework criado para trabalhar em conjunto com o sistema de controle de versão Git pode nos ajudar a melhorar a organização dos times.

Ele se baseia em criação de branches onde cada uma dela possui um papel definido e também defini como cada uma deve interagir com a outra. Elas são Historic Branches, Feature Branches, Release Branches e Maintenance Branches.

* **Historic Branches:​** Ao invés de trabalhar apenas com o branch master, esse workflow utiliza dois branches principais para guardar histórico do projeto. O branch master guarda o histórico oficial das entregas, já o branch develop serve como integração entre todos os branches de funcionalidades (feature branches).
* **Feature Branches:​** Cada funcionalidade deve ter seu próprio branch, e ele deve ser criado a partir do branch develop. Quando uma funcionalidade for concluída, ela é mesclada (merged) novamente com o seu branch pai. As features nunca devem interagir diretamente com o master.
* **Release Branches:​** Quando o branch develop estiver com funcionalidades suficientes para uma entrega, nós criamos um branch de entrega (release branch). Com isso, nós damos início ao próximo ciclo de entrega, ou seja, nenhuma nova funcionalidade pode ser incluída a partir desse momento. Quando estivermos prontos para realizar a entrega, o release é mesclada com os branches master e develop.
* **Maintenance Branches:​** Também conhecidos como hotfix. Eles são usados para corrigir rapidamente algum problema em produção. Este é o único branch que deve ser criado a partir do master. Assim que a correção for finalizada, o branch é fechado e mesclado com o master e develop, mantendo assim as linhas completamente atualizadas.



Um problema que o GitFlow nos ajuda a resolver ocorreria quando o projeto em produção e precisa de uma correção por conta de um bug que precisa ser resolvido imediatamente e se existisse apenas a branch master, tal problema só poderia ser corrigido após o desenvolvimento das novas funcionalidades em questão.

Esse tipo de problema é um grande impeditivo para a escalabilidade dos times, ainda mais por conta das tecnologias tais como microservices, data pipelines entre outras. Os times de alta performance precisam de flexibilidade e segurança na organização dos artefatos de códigos gerados.