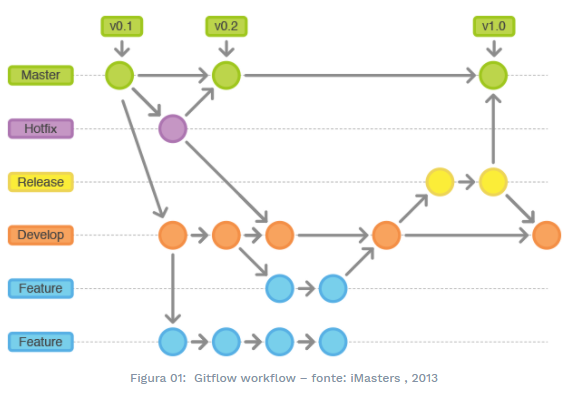
# GitFlow

Quando falamos de desenvolvimento, onde temos uma equipe, seja de desenvolvimento, dados ou até mesmo 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 mudança e com isso surge a necessidade de controlar todas essas mudanças. Uma das formas de fazer esse controle é utilizando o GitFlow.

Podemos imaginar diversos times trabalhando de forma concorrente e algumas vezes em tarefas conflitantes que exigem uma boa gestão do código para que o trabalho já feito não se perca ou para não haver 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 essa organização dos times.

Ele se baseia em criação de branches onde cada uma delas possui um papel definido, assim como define a forma que 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 e bugfix. 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.



Caso exista apenas a branch master, quando um projeto, já em produção, precisa de uma correção por conta de um bug que deve ser resolvido imediatamente tal problema só poderia ser corrigido após o desenvolvimento das novas funcionalidades em questão. Esse é um problema que pode ser facilmente resolvido pelo GitFlow.

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

Em resumo, o GitFlow é um forte aliado das equipes quando se fala em desenvolvimento de código, permitindo manutenções emergenciais, desenvolvimento de novas funcionalidades e rápida implementação sem deixar de lado o controle e versionamento das alterações realizadas. Além disso permitindo que a equipe trabalhe paralelamente sem interferir no trabalho que o outro está fazendo.