

# GitFlow

O *Gitflow* é um modelo de fluxo de trabalho para a gestão de branches no Git

## Explicando sobre seus componentes:

### Branches Principais:

**Master:** Contém a versão de produção do código. Cada commit na master representa uma versão estável e pronta para ser lançada.

**Develop:** Serve como a principal branch de desenvolvimento. Todo o trabalho em andamento é eventualmente integrado nesta branch, que sempre reflete o estado atual do desenvolvimento.

### Branches de Suporte:

**Feature branches:** Usadas para desenvolver novas funcionalidades ou melhorias. São derivadas da branch develop e, uma vez concluídas, são mescladas de volta na develop.

**Release branches:** Preparadas para lançar uma nova versão. Elas são criadas a partir da develop quando o desenvolvimento atinge um estado de "congelamento" para lançamento. Permite a realização de testes e correções de bugs finais sem interromper o trabalho contínuo na develop. Após o lançamento, são mescladas na master e na develop.

**Hotfix branches:** Criadas a partir da master para corrigir bugs críticos em produção. Depois de resolvidos, os hotfixes são mesclados de volta na master e na develop.

## Funcionamento

No Início do projeto é criado as Branchs Master e develop

**Quando uma nova funcionalidade precisa ser desenvolvida é criada uma feature branch a partir da develop.**

git checkout develop -> Feito para assegurar que você está na nova branch.

git checkout -b feature/nova-funcionalidade -> Cria a branch que vai ser utilizada para a nova aplicação.

**Feito o trabalho nessa branch ela é mesclada novamente na develop através do merge por exemplo:**

git checkout develop -> Feito para entrar novamente na branch de desenvolvimento  
git merge feature/nova-funcionalidade -> faz um merge com a branch que estava sendo trabalhada na develop  
git branch -d feature/nova-funcionalidade -> é deletada a branch que estava sendo utilizada para trabalho

**Quando o desenvolvimento atinge um ponto onde pode ser lançado, é criado uma branch de lançamento, a partir da develop.**

git checkout develop  
git checkout -b release/1.0.0

**A partir daqui, a release branch permite ajustes finais e correções de bugs sem interromper o trabalho contínuo na develop.**

**Quando a versão está pronta, a release branch é mesclada na master e um novo tag é criado para marcar a versão.**

git checkout master  
git merge release/1.0.0  
git tag -a 1.0.0 -> Aqui basicamente é feito uma anotação de quem fez o merge, contém mensagem, informações sobre autor e data e assinatura GPG

**A release branch também é mesclada de volta na develop.**

git checkout develop  
git merge release/1.0.0  
git branch -d release/1.0.0

**Quando um bug crítico é encontrado em produção, uma hotfix branch é criada a partir da master.**

git checkout master  
git checkout -b hotfix/1.0.1

**Após corrigir o bug, a hotfix branch é mesclada de volta na master e na develop, garantindo que a correção esteja presente em ambas as branches.**

## **Não Faz parte do GitFlow mas é interessante ter para registro**

### **Pull Requests:**

Um Pull Request é uma proposta de integração de alterações de uma branch para outra. Ele permite que outros desenvolvedores revisem, comentem e aprovem as mudanças antes que elas sejam mescladas no código principal. O processo de revisão ajuda a garantir a qualidade do código e a facilitar a colaboração entre os membros da equipe.

**Quando uma nova funcionalidade é desenvolvida em uma feature branch, o desenvolvedor cria um Pull Request para mesclar essa branch na develop.**

**O Pull Request permite que outros membros da equipe revisem o código, façam comentários e solicitem mudanças antes que a funcionalidade seja integrada na branch principal de desenvolvimento.**

```
git checkout develop  
git checkout -b feature/nova-funcionalidade # Desenvolvimento da funcionalidade  
git push origin feature/nova-funcionalidade
```

**Em seguida, o desenvolvedor cria um Pull Request na interface do GitHub (ou outra plataforma) para mesclar feature/nova-funcionalidade na develop.**

**Sendo aprovado o merge pode ocorrer automaticamente dependendo das configurações ou manualmente.**