1. **-- git flow init**

-- master e develop são fixas, se manterão durante todo o projeto

-- branches "supporting" prefixo(feature/): vão sendo excluídas automaticamente, no decorrer do projeto;

-- + feature/; bugfix/; release/(homologaçao); hotfix/; support/; version tag (optional); hook and filters(yes)

* **-- git branch: exibe as duas branches criadas: develop e master;**

***Exemplo: calculadora***

1. **-- git flow feature start sum:**

cria uma nova branch FEATURE para o desenvolvimento da funcionalidade "soma", ativando-a automaticamente

Desenvolver as funcionalidades necessarias.

* **-- touch sum**

Ao finalizar a funcionalidade:

* 1. **-- git add .**
  2. **-- git commit -m "adiciona funcionalidade de soma"**
  + **(-- git branch [para visualizar as branches])**

1. **-- git flow feature finish sum**

Mescla (realiza o merge automaticamente) a branch feature/sum à branch develop, excluindo automaticamente a branch feature/sum

1. **-- git flow feature start sub**

cria uma nova branch FEATURE para o desenvolvimento da funcionalidade "sub", ativando-a automaticamente

Desenvolver as funcionalidades necessárias.

* + 1. -- touch sub

Ao finalizar a funcionalidade:

* 1. **-- git add .**
  2. **-- git commit -m "adiciona funcionalidade de subtracao"**
* (-- git branch [para visualizar as branches])

1. **-- git flow feature finish sub**

Mescla (realiza o merge automaticamente) a branch feature/sub à branch develop, excluindo automaticamente a branch feature/sub

* -- ls

lista as funcionalidades presentes em develop, até então

**PASSANDO PARA A BRANCH RELEASE**

1. **-- git flow release start 0.1.0**

Sobe a branch develop para a branch de homologação release, com a tag 0.1.0

* **-- git branch**

visualiza a nova branch e as demais;

**PASSANDO PARA A BRANCH MASTER**

1. **-- git flow release finish 0.1.0**

Realiza o merge da branch release/0.1.0 com a master;

Abre o editor de texto para solicitar o pull request

Inserir texto com a descrição das funcionalidades

apaga automaticamente a branch release/0.1.0, ativando a master

* 1. **--git tag**

visualiza a tag da branch atual

**OBSERVAÇÕES**

Ao atualizar finalizar uma branch RELEASE, a atualização da MASTER e da DEVELOP ocorrem AUTOMATICAMENTE, não necessitando de merge ou pull por parte do usuario;

O mesmo fluxo serviria para branches HOTFIX, considerando que as finalizações de HOTFIX tambem irão atualizar MASTER e DEVELOP;

HOTFIX e Develop são as unicas branches criadas a partir da MASTER