

Versionamento

O versionamento é feito com [git](#).

O repositórios podem sera acessados na aba [Repos](#). Antes de começar a trabalhar com os repositórios é preciso [criar e adicionar em sua conta uma chave SSH](#).

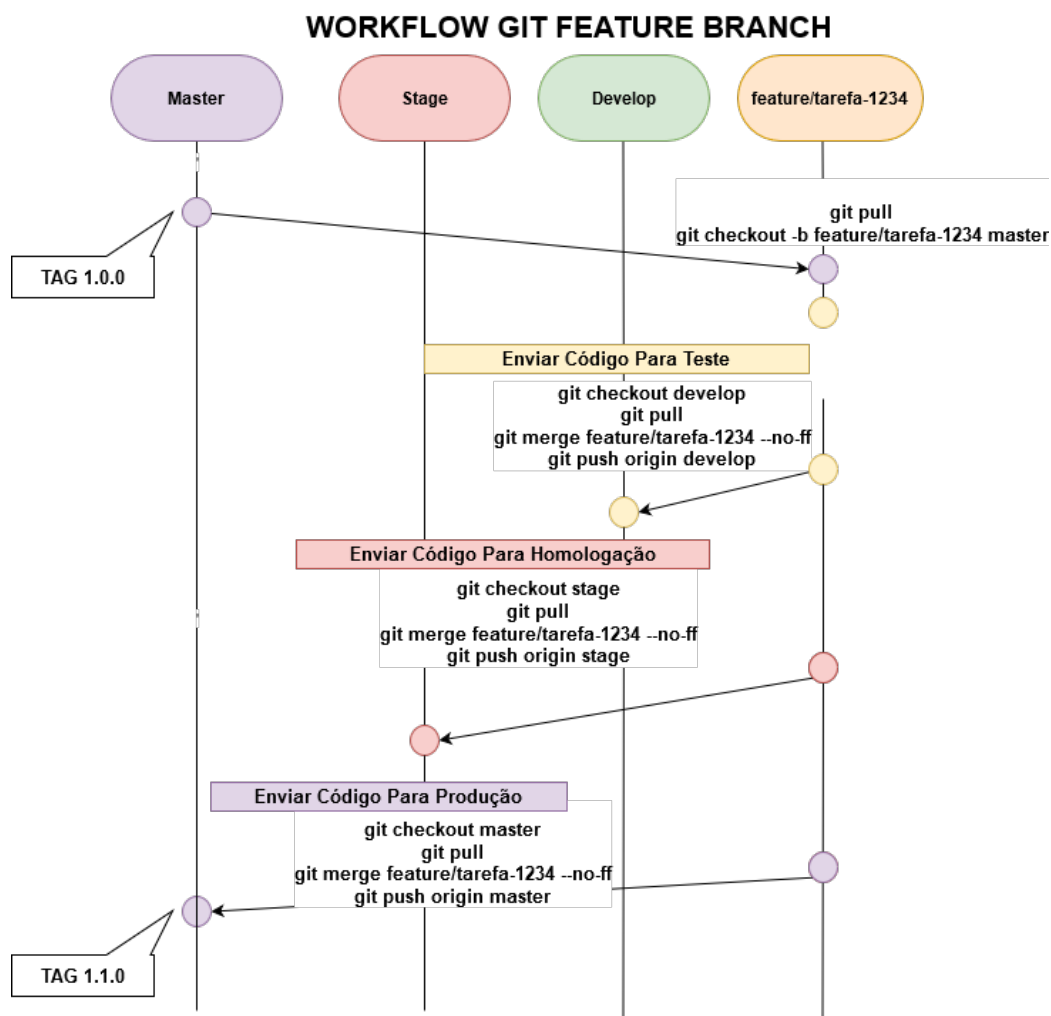
Clonando o Repositório

Acesse o repositório desejado no menu [Repos](#), em seguida clique em 'clone', marque o protocolo SSH e copie o endereço resultante. Em seguida, no diretório desejado, execute o comando `git clone` informando o endereço do respositório copiado. Por exemplo, para clonar o repositório do front do PRDI:

```
$ git clone git@ssh.dev.azure.com:v3/bbts-lab/PRDI/prdi
```

Fluxo de Trabalho

Para este projeto, deve ser seguido o **feature branch**.



Passo 1

Para cada nova funcionalidade, o desenvolvedor deverá realizar um clone da branch de produção com o seguinte padrão de nome
tarefa-NUMERO_TAREFA:

```
$ git checkout -b tarefa-12345 master
```

Passo 2

Realizar as alterações necessárias e commitar em sua branch. Os commits devem conter o código da tarefa (útil para acompanhar no card) e a descrição do que foi feito:

```
$ git add .
```

```
$ git commit -am "Tarefa #12345: minha alteração"
```

```
$ git push origin tarefa-12345
```

Passo 3

Ao finalizar sua atividade e seus devidos testes, o desenvolvedor deverá realizar o merge para a branch develop, onde os testadores deverão realizar mais alguns testes antes de colocar a atividade em homologação:

```
$ git checkout develop
```

```
$ git pull
```

```
$ git merge tarefa-12345 --no-ff
```

```
$ git push origin develop
```

Passo 4

Após os testadores aprovarem a atividade, o mesmo processo deverá ser realizado para a branch stage, onde o Product Owner irá verificar se o que foi feito está de acordo com o esperado:

```
$ git checkout stage
```

```
$ git pull
```

```
$ git merge tarefa-12345 --no-ff
```

```
$ git push origin stage
```

Passo 5

Se houverem bugs, o desenvolvedor deverá realizar os ajustes na branch daquela tarefa e refazer os passos 3 e 4.

Passo 6

Ao término da Sprint, tudo o que foi homologado pelo Product Owner deverá ser feito o merge para a branch master:

```
$ git checkout master
$ git pull
$ git merge tarefa-12345 --no-ff
$ git push origin master
```

Obs: Não esquecer de alterar a versão do sistema seguindo o [versionamento semântico](#) .

HotFixes

Se houverem bugs em produção, o desenvolvedor deverá criar uma branch a partir da master com o seguinte padrão de nome: `hotfix-NUMERO_TAREFA`. Se a correção tiver a análise da necessidade de homologação, deverão ser realizados os passos 3 e 4.

Se for algo muito simples como: troca de label, acentuação, cor, etc. poderão ser realizados sem a necessidade de homologação, porém deverão constar nas outras branches!

Atenção

Não execute os comandos abaixo:

```
$ git merge develop
$ git merge stage
```

Não faça commit direto nas branch's develop, stage e master.