



# Política de Git

## - Política de Commits

### 1 - Faça commits simples

Dividir os commits em pequenos implementando uma funcionalidade

### 2 - Escrever os commits em Ingles

Para deixar o projeto acessível para todos, o idioma padrão tanto para o código quanto para as coisas relacionadas, logo os commits em ingles (**caso o grupo queira manter em portugues, iremos alinhar isso**)

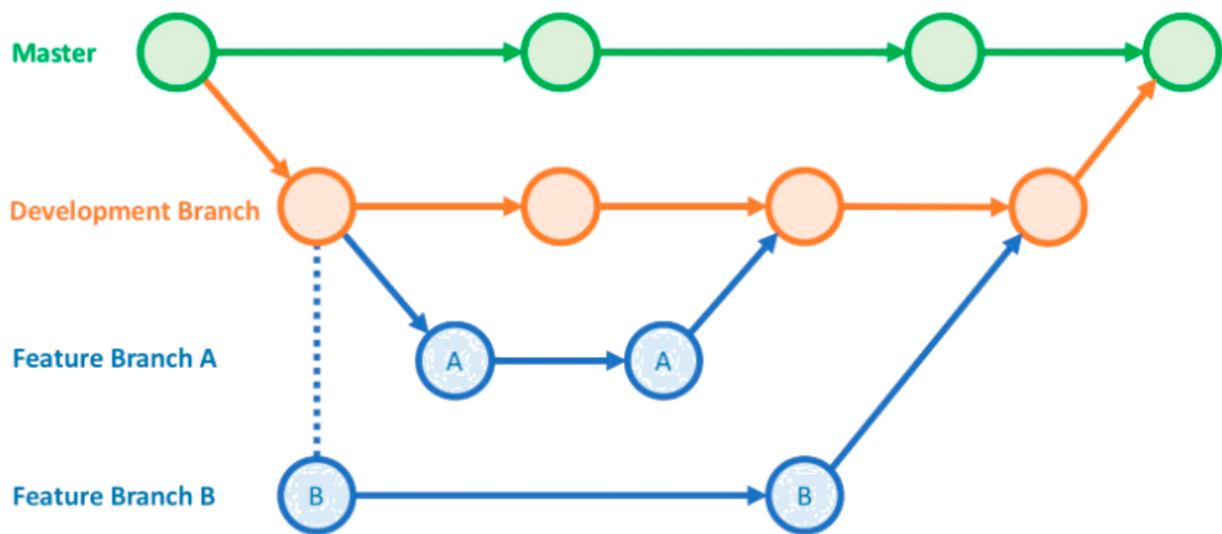
#### Formato

```
#<numero-da-issue>-<corpo_do_commit>
```

**Padrão Snake case: Utilizamos underline no lugar do espaço para separar as palavras.**

Obs: Ao longo do projeto, vão entendendo na prática convenções de nomenclaturas

## - Política de Branch



### 1 - Branches principais

O repositório terá duas branches principais: *master* e *develop*

A branch *master* possui sempre a versão mais estável da aplicação, ou seja a versão que deve estar pronta para produção

Já a branch *develop* possui a última versão desenvolvida da aplicação. Pode ser considerada a branch de integração, onde os PRS serão feitos nela.

### 2 - Pull Request

Toda nova *feature* desenvolvida deve ser integrada à branch *develop* por meio de um processo de **Pull Request**, que consiste em revisar o código a ser integrado, além de validar a integridade da nova funcionalidade utilizando das ferramentas de integração contínua.

## - Política de Issues

### 1.1 - Título da issue

Descreva o problema de forma sucinta, sendo explícito na descrição do título. Um padrão a ser seguido é **[USX]+[VERBO]+[SUBSTANTIVO]**

### 1.2 - Descrição da issue

Todas as Issues devem possuir uma descrição detalhada de o que deve ser feito e como deve ser feito.

Em caso de User Story, a issue deve seguir um padrão onde deve responder as seguintes perguntas:

Eu como **[Papel]**, quero **[Meta]** para que eu possa **[Motivo]**.

**Exemplo:** “Eu, como **desenvolvedor**, desejo **realizar os testes funcionais das classificações**, para **garantir o devido funcionamento da aplicação**”

Após a descrição detalhada, o corpo da issue deve haver um checklist de todas as operações que deve ser realizada.

Exemplo: **US18** - **Criar resumo de política de git**

---

## Alguns comandos do git que serão importantes pra matéria (Adicional)

### 1 - Adicionar arquivos que desejam que suba

```
git add <nome-dos-arquivos-a-serem-adicionados>
```

ou

```
git add .
```

No começo, evitem de usar esse para aprender como realizar commit.

Muito cuidado também que as vezes vocês podem realizar a subida de um arquivo que nao seja necessário ou/e que quebre a aplicação

## 2 - Realizar o commit

```
git commit -m "[numero-da-issue]-[mensagem-do-commit]"
```

## 3 - Realizar o push

```
git push origin <nome-da-branch>
```

## 3 - Realizar o pull

```
git pull origin <nome-da-branch>
```

ou

```
git push
```

## 4 - Criar nova branch

```
git branch <nome-da-branch>
```

## 5 - Realizar a mudança de branch

```
git checkout <nome-da-branch>
```

### *PLUS*

```
git checkout -b <nome-da-branch>
```

\* Esse comando realiza a criação direta da branch e já muda automaticamente para ela