



Padrão de Utilização do Git

v1.0.0

Histórico de Revisão

Data	Versão	Descrição	Autor
11/11/2018	1.0.0	Criação do documento	Samuel Jhonata

Sumário

Considerações iniciais	2
Repositório	2
Branches	2
Controle de Acesso	3
Fluxo de uso	3
Considerações Finais	4



Considerações iniciais

Será utilizado o **GitHub** como controlador de versão desse projeto. Recomenda-se o uso do *GitKraken* ou a versão gráfica para desktop do *GitHub* para facilitar o uso da ferramenta.

Para um melhor funcionamento do projeto, o Git deverá ser usado da maneira estabelecida neste documento, estabelecida a partir de inspiração no **Git Flow**, modelo muito utilizado no mercado.

Repositório

O repositório a ser usado tem o nome '**appColeta**', e deve ter a seguinte árvore de arquivos:

- appColeta
 - Analise e Design
 - Analise
 - Arquitetura
 - Design GUI
 - Desenvolvimento
 - Mobile
 - Web
 - Gerenciamento
 - Configuração
 - Processo
 - Projeto
 - Qualidade
 - Testes

A partir dos níveis descritos, podem haver outros níveis.

Branches

Serão utilizados os seguintes *branches* para o desenvolvimento do projeto:

- **master**: irá conter todo código já testado, versionado que já poderá ser entregue ao cliente, além dos arquivos de gerenciamento mais recentes e os de modelagem já testados;
- **model**: irá ser usado para fazer a modelagem do sistema, seus testes e suas correções;
- **bugs**: irá ser usado para fazer correções em versões que já foram lançadas.
 - deverão ser criados quando necessário, com o nome: **bug_v<versão>**. Ex: '**bug_v0.1.0**', onde a versão deverá ser a referência à versão que está sendo corrigida.
- **release branches**: irá ser usado para lançar uma nova versão estável
 - deverão ser criados com o nome: **release_v<versão>**. Ex: '**release_v0.1.0**', onde a versão deverá ser a referência à versão que será lançada.



Equipe 2 - Aplicativo Coleta Seletiva

- **develop**: deve sempre conter o código mais atual, onde as *branches* de *features* serão ramificadas tendo ela como base;
- **features branches**: irá ser usada para novas implementações, a partir da *develop*
 - **front_end**: irá ser utilizada para desenvolver o *front-end* do projeto.
 - **back_end**: irá ser utilizada para desenvolver o *back-end* do projeto.

Controle de Acesso

O **master** deverá ser acessado apenas pelo Product Manager (PM), Diretor de Projetos ou alguém que seja previamente autorizado por eles.

Model deve ser utilizado apenas pelo PM e pelos Analistas/Testadores.

Os demais *branches* podem ser acessados por qualquer um da equipe, porém, é importante que cada papel procure utilizar sempre o seu branch da forma correta.

Fluxo de uso

O fluxo de uso deve seguir, conforme a **Figura 1** abaixo:

Fluxo Git

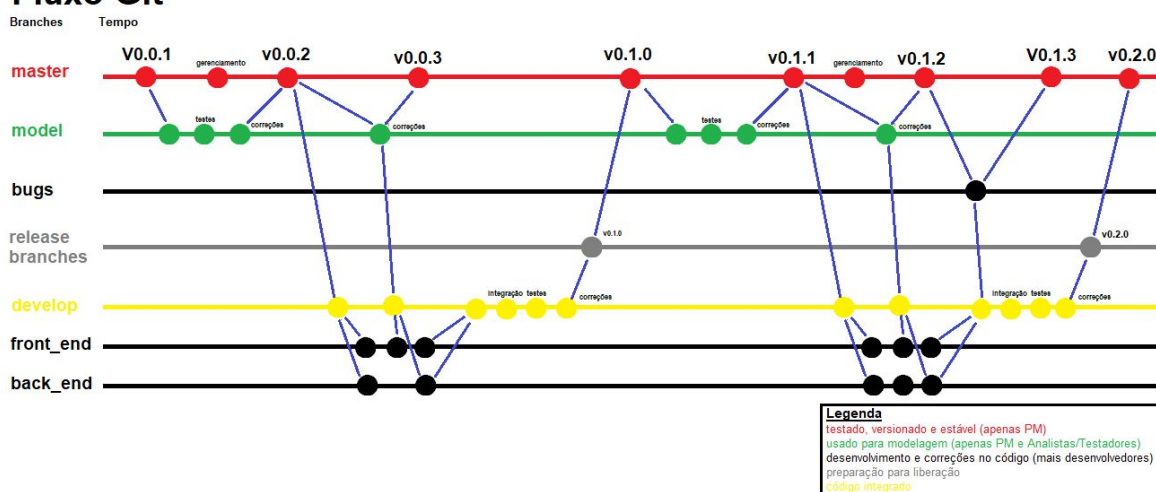


Figura 1: Fluxo de uso do Git

De forma geral, o **master** deve conter a última versão estável, recebendo modificações gerenciais do PM diretamente. O **model** deve ser utilizado para modificações do modelo, sendo ramificado a partir do *master*, fazendo *merge* diretamente no *master* e replicando as alterações no *develop*. Os *branches* de correções de **bugs** devem ser criados a partir de uma ramificação do *master*, fazendo o *merge* após as correções diretamente no *master*. Os *branches* de **release** devem ser criados e ramificados a partir do *develop*, depois de já integrado e testado, criando uma nova versão, fazendo *merge* no *master*. O **develop** deve ser ramificado a partir do *master*, sendo atualizado sempre que houver mudanças já testadas do *model* ou correção de *bugs*. Os *branches* de **features front_end** e **back_end** devem ser atualizados a partir do *develop* a cada início de implementação, ou quando houver atualização no *develop*, durante a implementação da nova funcionalidade.



Considerações Finais

Seja organizado, sempre utilize o padrão determinado e respeite as regras do fluxo, para uma melhor qualidade do desenvolvimento do projeto.