

1 Plano de Gerência e Configuração de Software

O Plano de Gerência e Configuração de Software tem por objetivo apresentar padrões, políticas, ferramentas, instruindo sobre o ambiente de desenvolvimento de software e qualquer atividade de configuração necessária.

1.1 Políticas

1.1.1 Política de Commits

Os commits devem ser atômicos e seu comentário deve descrevê-lo de forma sucinta. O texto deve descrever o que foi produzido, de forma resumida e em português. Caso o commit não seja destinado para a conclusão da funcionalidade ou documento, deve-se iniciar com o verbo no gerúndio, no entanto, se o commit é destinado a conclusão da funcionalidade ou documento, deve-se iniciar com o verbo no particípio. Além disso, deve conter o número de sua issue correspondente, no seguinte formato:

Repositórios

Texto começando com letra maiúscula, verbo no gerúndio ou particípio

Exemplo: ***commit* destinado à conclusão**

Criada estrutura de usuário.

***commit* não destinado à conclusão**

Criando estrutura de usuário.

1.1.2 Política de Branches

Serão utilizados os princípios do **Gitflow** que ajudarão no controle do que está sendo produzido pela a equipe, onde, ao mesmo tempo falhas serão corrigidas, novas funcionalidades serão implementadas, garantindo o funcionamento do código de produção. O Gitflow foi criado em 2010, é considerado um ótimo modelo de *branching*. É um modelo fortemente baseado em branches, mas focados em entregas de projetos, ele define os papéis de cada branch e como elas devem interagir. Apesar dele ser um pouco mais complexo que outros workflows, ele disponibiliza um framework robusto para gerenciar projetos mais complexos.

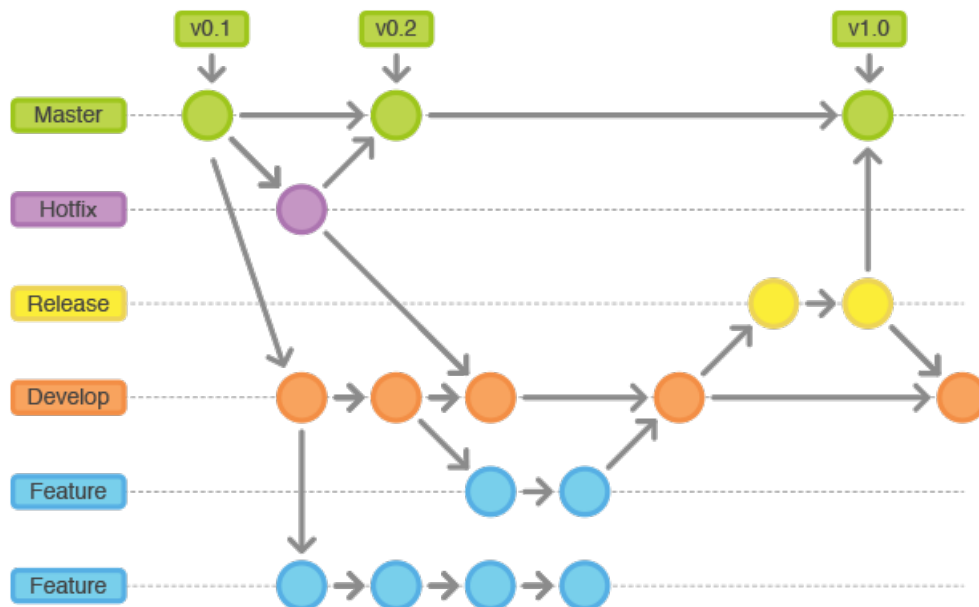


Figure 1: Exemplo do funcionamento do GitFlow

A *master* será a *branch* estável do projeto, sendo ela proveniente da *devel* por meio de aprovação de *pull request* ao fim de cada *release*. Nenhum membro será autorizado a fazer *commits* diretamente na *master* ou na *develop*

As branches auxiliares são destinadas a implementação de funcionalidades, realização de histórias técnicas e conserto de bugs. Cada uma dessas atividades terá sua própria branch, criada a partir da *develop*, as *hotfix* são as branches criadas a partir da *master* e servem para resolver de forma rápida os bugs em produção. Terão como padrão de nomenclatura:

[Identificador da atividade]-[Nome issue associada a atividade]

Exemplos:

feature/TS03-Configurar-Ambientes

hotfix/BUG-Duplicação-no-Banco

feature/US01-Implementar-Login

doc/DOC-diagrama-classe

Após o fim do desenvolvimento nas branches auxiliares elas devem ser incorporadas a *develop* por meio de *pull request*.

1.1.3 Política de Aprovação do Código

Para a aprovação do código, o *pull request* deve ser revisado por ao menos 1 membro da equipe, a nomenclatura da *branch* e dos *commits* devem estar de acordo com as definições deste documento, o código deve estar escrito seguindo a folha de estilo, a build não pode apresentar erros e o *pull request* deve seguir o template do *community*.

1.2 Uso de Issues

As *issues* serão criadas com o objetivo de mapear e descrever todo o trabalho a ser desenvolvido durante o projeto, possibilitando controle e transparência do que está sendo feito. Com isso, conseguiremos manter o rastro de tudo que foi planejado e efetuado.

As issues vão conter identificadores e *labels*, para que se possa indicar sua natureza. Os identificadores definidos para o projeto serão:

[**EPIC**] - Utilizado para as issues que representam épicos.

[**US**] - Utilizado para as issues que representam histórias de usuário.

[**TS**] - Utilizado para as issues que representam histórias técnicas.

O formato padrão de nomenclatura para essas issues é:

[Identificador Número-da-issue] nome-definido-pela-equipe-para-issue

Exemplo:

[US01] Prototipação

[**REFACTOR**] - Utilizado para issues que representam refatoração.

[**BUG**] - Utilizado para issues que representam correção de bugs.

[**DOC**] - Utilizado para as issues que representam tarefas de documentação.

[**TRAINING**] - Utilizado para issues que representam atividades de estudo e treinamento.

[**QUESTION**] - Utilizado para issues que representam perguntas que a comunidade deseja fazer aos mantenedores.

[**SUGGESTION**] - Utilizado para issues que representam sugestões que a comunidade deseja fazer aos mantenedores.

O formato padrão de nomenclatura para essas issues é:

[Identificador] Nome definido para a issue pela equipe

Exemplo:

[BUG] Duplicação no Banco

1.3 Ferramentas

Ferramenta	Descrição
<i>Git</i>	<i>Ferramenta de versionamento</i>
GitHub	Ferramenta de hospedagem de repositórios
ZenHub	Ferramenta de gerenciamento de equipe
React Native	Framework para a criação de aplicativos mobile
Travis CI	Ferramenta de integração contínua
VS Code	Ferramenta de construção e edição de código fonte
Android Studio	Ferramenta para emulação do aplicativo

1.3.1 Integração das Ferramentas

Considerando o processo de desenvolvimento do software, o código fonte em Javascript (React Native) é editado através da IDE VS Code e sua emulação é realizada através do Android Studio. Já seu versionamento é feito utilizando-se o GitHub e, sempre que possível, é sincronizado o trabalho realizado localmente com o repositório remoto hospedado no GitHub. A partir deste ponto entra em cena o Travis CI, pois após cada alteração no repositório remoto o Travis gera uma nova build do projeto. Além de realizar a build, o Travis também possui outras funções que em alguns casos fazem parte da build e em outros casos são eventos pós build ou pré build, e comunica qualquer problema que ocorra nesse processo, impedindo que código quebrado se junte as versões estáveis do projeto.

1.4 Referências

PMI. Um guia do conhecimento em gerenciamento de projetos. Guia PMBOK® 5a. ed. - EUA: Project Management Institute, 2013

Semantic Versioning 2.0.0 . Semantic Versioning Specification (SemVer). Disponível em (<http://semver.org/>)

PlataformaJogosUnB. Plano de Gerenciamento de Configuração de Software. Disponível em (<https://github.com/fga-gpp-mds/2017.1-PlataformaJogosUnB/wiki/Plano-de-Gerenciamento-de-Configura>)