

# **UNICESUMAR – UNIVERSIDADE CESUMAR**

CENTRO DE CIÊNCIAS EXATAS TECNOLÓGICAS E AGRÁRIAS CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

# PLANO DE GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE

Danielle Sayuri Kitagawa

# PLANO DE GERENCIAMENTO DE CONFIGURAÇÃO DE SOFTWARE

# 1.INTRODUÇÃO

O objetivo deste plano de gerenciamento é definir as ferramentas a serem utilizadas no decorrer do desenvolvimento do projeto, de forma a compreender as políticas a serem seguidas, utilizando ferramentas que garantam o gerenciamento organizacional e o controle do planejamento das atividades.

#### 2. FERRAMENTAS

#### 2.1 Redmine

Ferramenta de gerenciamento de controle, como as issues, solicitações de mudanças e rastreamento das funcionalidades.

#### 2.2 Github

Ferramenta para a hospedagem do repositório do código-fonte e o controle das versões com o Git.

### 2.3 Google Docs

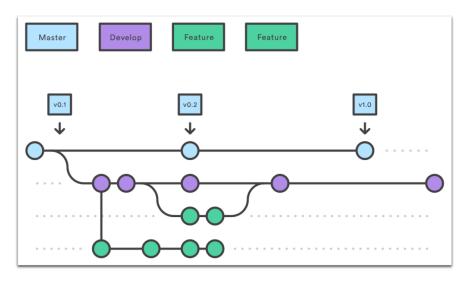
Ferramenta de gerenciamento dos artefatos de especificação

#### 3. FLUXO DO PROJETO

Deve ser feito três projetos no Redmine, sendo um específico para solicitação de mudanças, outro para as features e para as atividades de desenvolvimento, tendo fluxos diferentes para cada tipo de issue.

#### 4. POLÍTICAS

#### 4.1 Branch



Fonte: Atlassian, 2020.

O modelo a ser utilizada é o GitFlow, que possui a branch Master e a Develop, sendo as principais para o desenvolvimento do projeto. Sendo a primeira, relacionado a versão estável do produto, e a segunda é a base do desenvolvimento do projeto, sendo que todas as features são feitas de acordo

com essa branch. Já as features são a evolução do projeto, as funcionalidades a serem implementadas.

#### 4.2 Commit

Os commits devem seguir o seguinte padrão, de forma que inicie com a abreviação padrão do sistema juntamente com o número da pendência, e uma descrição do que foi feito. Dessa forma, a pendência estará ligada a issue no Redmine.

Exemplo: "XX-01 Primeiro Commit"

Nome do sistema – Número da pendência – Mensagem do que está sendo feito

## 4.3 Solicitação de aprovação de mudanças (Pull Request)

O responsável deverá solicitar um pull request para a integração da feature a branch da develop, para isso um membro da equipe deverá revisar as alterações, sendo possível comentar, aprovar ou também solicitar melhorias antes de aprovar. Antes da solicitação do merge, também será feita uma nova revisão.

### 4.4 Liberação da Release

Para a liberação da release, o Product Owner(PO) deverá registrar em um documento as seguintes informações, a data de liberação, a versão, o autor e também as notas da revisão.

## 4.5 Tipos de Issues (Tracker)

As issues podem ser classificadas com os tipos:

- **4.5.1 Support (Atendimento):** Representam um atendimento prestado a um cliente, ou seja, são as dúvidas/problemas relatados pelo cliente
- 4.5.2 Improvement (Melhoria): Representam a criação ou a evolução de uma funcionalidade
- **4.5.3 Technical Debt (Dívida Técnica):** Representam a defasagem técnica de alguma funcionalidade do sistema, como uma biblioteca desatualizada.
- **4.5.4 Bug (Falha):** Representam uma falha em uma funcionalidade existente, que não está de acordo com os requisitos propostos.
- **4.5.5 Task (Tarefa):** Representam ajustes, mas que não estão relacionados a um software entregável, mas a infraestrutura ou algo relacionado ao produto.

#### 5. Status das Issues

Para as Issues, deverá conter os seguintes status: backlog, em análise técnica, para desenvolvimento, em desenvolvimento, para auditoria, em auditoria, para teste, em teste, para publicar, em publicação, concluída, bloqueada e excluída.