# Faculdade de Juazeiro do Norte - FJN

Sistema de Controle de Estoque

Plano de Gerenciamento de Configuração

**Equipe:**

Ananda Rafaele Pinheiro Silva

Andressa Alves de Oliveira Souza

Cícero Tiago Galdino Bento Júnior

Jonas dos Santos Aquino

Jhonatan Bezerra Ferreira

Histórico de Revisões

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 09/10/2018 | 0.1 | Elaboração do documento. | Cicero Tiago |
| 10/10/218 | 0.2 | Gerenciamento de Configuração de software e fluxos de controles de mudanças. | Jonas e Jhonatan |
|  |  |  |  |

SUMÁRIO

[1. Introdução 4](#_Toc526971561)

[1.1. Objetivos 4](#_Toc526971562)

[1.2. Escopo 4](#_Toc526971563)

[2. Gerência de Configuração de Software 4](#_Toc526971564)

[2.1. Ferramentas 4](#_Toc526971565)

[2.2. Fluxos de controle de versão 4](#_Toc526971566)

[2.3. Regras para versionamento 4](#_Toc526971567)

[3. Fluxos de controles de mudanças 5](#_Toc526971568)

# Introdução

O Plano de Gerenciamento de Configuração descreve todas as atividades do Gerenciamento de Controle de Configuração e Mudança que serão executadas durante o ciclo de vida do produto. Suas atividades envolvem identificar a configuração do software, manter sua integridade durante o projeto e controlar sistematicamente as mudanças.

## Objetivos

O objetivo deste documento é criar um padrão a ser seguido por todos os membros da equipe com o intuito de garantir o maior controle do produto no decorrer do projeto.

## Escopo

Este Plano de Gerenciamento de Configuração é destinados para todos os integrantes da Fábrica de Software Origami, e abrange todo o controle e gerenciamento da configuração do projeto.

# Gerência de Configuração de Software

## Ferramentas

|  |  |  |
| --- | --- | --- |
| Termo | Versão | Descrição |
| *Git* | 2.19.0 | Ferramenta de administração dos repositórios distribuídos. |
| *GitHub* |  | Ferramenta de controle de versão junto com o *Git, também utilizado como gerenciamento de mudanças e versionamento.* |

## Fluxos de controle de versão

O controle de versão adotado é de caráter distribuído, sendo assim cada desenvolvedor possui um repositório local, onde a cada revisão o desenvolver realizará um *commit/push* no repositório central, *GitHub* na *Branch* de desenvolvimento, no final de cada *Sprint,* as revisões são mescladas na *Branch master*, gerando assim uma versão, que essa é identificada utilizando a convenção numérica X.X.X onde o primeiro número identifica a Versão e segundo a Release e o terceiro a Build.

## Regras para versionamento

Será considerado uma Versão do sistema, quando um grupo de funcionalidades básicas atenderem os requisitos pré-estabelecidos pelo cliente, que é um software utilizável e testado para ser lançado no ambiente de produção, a primeira versão vai começar do numeral 1 seguindo em ordem crescente para próximas versões

Para a *Release* vai ser considerando funcionalidades adicionadas nas próximas Sprints, seguindo a mesma ordem crescente.

# Fluxos de controles de mudanças

Para o controle de mudanças será utilizado da *Issues* do *Github* que permite aos usuários reportarem problemas/bugs, sendo assim para solicitações de mudanças será aberta uma *Issues* e será rotulada e atribuída para o responsável.

1. Abertura *da Issues:*Ao clicar em novo *Issue* você pode escolher o tipo de problema que deseja reportar ou selecionar vários tipos de problemas, deve digitar um título e uma descrição para o seu problema.
2. Rotulação*:*Pode catalogar por cores para ajudar a filtrar seus problemas.
3. Atribuição*:*Pode atribuir o problema para algum membro da equipe.
4. AnáliseÉ feita a análise da solicitação de mudança, e verificado se realmente é algo que deverá ser realizado, caso sim é feito o procedimentos necessários para acatar a mudança.
5. FechamentoApós a conclusão da tarefa referente a solicitação de mudança, é mudado o status da *Issues* para fechado, indicando o termino do fluxo.