

Plano de Gerência de Configuração



Space Cats Lab

Sumário

| | |
|---|----------|
| 1 Objetivo de Plano de Configuração | 3 |
| 2 Processo de Gerência de configuração | 3 |
| 3 Definições, Abreviações e Acrônimos | 4 |
| 4 Ferramentas | 5 |
| 5 Gerenciamento da configuração..... | 6 |
| 5.1 Nomenclatura e Descrição das Branches. | 6 |
| 6 Padrões | 7 |
| 6.1Commits..... | 7 |
| 6.2 Documentos..... | 7 |
| 7 Solicitação de mudança (CR) | 8 |
| 8 Ciclo de Vida..... | 8 |
| 8.1 Alto nível..... | 8 |
| 8.2 Baixo nível..... | 9 |

1 Objetivo de Plano de Configuração

Esse plano de configuração tem como objetivo a estabilização de normas e estratégias que irão ser adotadas para o desenvolvimento de todo o projeto da SPACE CATS LAB. Todas as ferramentas, tecnologias e normas contidas nesse plano devem ser adotadas para manter a integridade do projeto durante seu ciclo de vida.

Todos os processos a seguir foram escolhidos visando uma melhor performance e entendimento para todos os processos da fábrica.

2 Processo de Gerência de configuração

Para todo novo projeto da fábrica Space Cats Lab deverá ser dado um nome com as seguintes regras: SLC_<NOME_DO_PROJETO>_<ANO_DO_PROJETO>, Sendo da seguinte forma:

- SLC = Iniciais da fábrica de software
- Nome do projeto = Deverá conter no máximo 7 caracteres alfabéticos.
- Ano do projeto = Ano em que o projeto foi aceito.

3 Definições, Abreviações e Acrônimos

Segue abaixo tabela com as definições, abreviações e acrônimos que serão utilizadas durante todo este documento.

| | |
|-----------------------|--|
| TC- Tipo do Commit | Abreviação utilizada para identificar o tipo do commit que será enviado. |
| NI- Número da issue | Abreviação utilizada para identificar a issue à qual pertence o commit. |
| Breve Descrição | Acrônimo utilizado para descrever em poucas palavras as mudanças realizadas no código. |
| Branch | É uma nomenclatura utilizada para identificar um ramo em um repositório git. |
| Sprint | Nomenclatura utilizada para identificar ciclos de trabalho na metodologia ágil. |
| Commit | Nomenclatura utilizada para sinalizar um envio de um código para o repositório git. |
| Release | É uma entrega planejada de uma parte do projeto, essa parte deve ser funcional. |
| CR – Changing Request | Abreviação usada para indicar uma solicitação de mudança nos requisitos do projeto. |
| Issues | Ferramenta do Github que permite o gerenciamento de tarefas. |
| P.B Product Backlog | Abreviação usada para indicar os requisitos do projeto. |

4 Ferramentas

Com o objetivo de otimizar o desenvolvimento do projeto as seguintes ferramentas e tecnologia foram adotadas como obrigatórias para a execução do projeto.

As ferramentas e tecnologias são as seguintes:

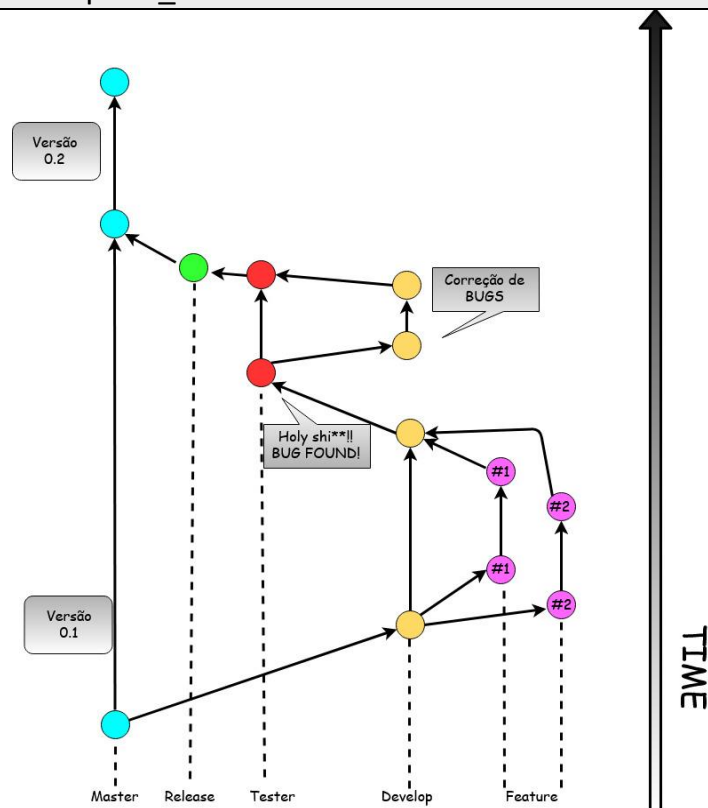
| Ferramenta | Objetivo |
|------------------------------------|---|
| Visual Studio Code | IDE utilizada para desenvolvimento das aplicações WEB. |
| REACT | Biblioteca JAVASCRIPT utilizada para desenvolvimento de interface WEB. |
| NODEJS | Tecnologia utilizada para criação de backend para desenvolvimento WEB. |
| TELEGRAM | Ferramenta de comunicação instantânea para membros do projeto. |
| DISCORD | Ferramenta de comunicação para membros do projeto. |
| GIT | Ferramenta utilizada para controle de versão. |
| GITHUB | Ferramenta remota que utiliza o serviço de controle de versão e gerenciamento de projeto. |
| MARIADB | Sistema de gerenciamento de banco de dados utilizado para criação de banco de dados relacional. |
| WIX | Ferramenta para criar o site da fábrica para apresentações da disciplina. |
| DRAW.IO | Ferramenta utilizada para criação de UML. |
| HEFLO | Ferramenta utilizada para a modelagem de processos. |
| ADOBE XD | Ferramenta utilizada para criação de protótipos para páginas WEB. |
| GOOGLE DRIVE | Drive de armazenamento em nuvem para todos os arquivos da empresa. (*) |
| GMAIL | Plataforma de correio eletrônico adotada pela fábrica. |

(*) Códigos não devem ser alocados nesse Drive, todos os códigos devem ser hospedados no GITHUB, conforme item 6 desse Plano.

5 Gerenciamento da configuração

5.1 Nomenclatura e Descrição das Branches.

| Branch | Descrição |
|----------|--|
| master | Branch inicial de todo projeto, local onde será salvo as versões oficiais do projeto. Apenas o Gerente de Projetos pode enviar itens para essa Branch. |
| realease | Branch que irá conter o resultado funcional/final ao fim da sprint. |
| tester | Branch que irá realizar os testes das features da sprint. |
| develop | Branch onde irá conter a versão mais atualizada do código, todas as Branchs de features serão ramificadas tendo ela como base. |
| features | Toda feature deve ser criada a partir da Branch develop. Cada Issues deverá possuir sua própria Branch com o seguinte padrão "develop <ID_ISSUE>" |



6 Padrões

Essa seção descreve os padrões de identificação adotados para o desenvolvimento do projeto.

6.1 Commits e códigos.

Os commits devem ser escritos da seguinte forma: TC:<BREVE_DESCRIÇÃO>#NI

Abaixo contém uma lista com os 4 tipos de commit disponíveis.

| TC | Descrição |
|------------|-----------------------------|
| 'style' | Para mudanças de estilo. |
| 'refactor' | Para refatoração de código. |
| 'test' | Para testes de código. |
| 'chore' | Para finalização de issue. |

Todos os códigos do projeto devem ser enviados para a branch adequada no repositório remoto da fábrica assim que finalizado.

6.2 Documentos

Todos os documentos relacionados ao projeto, com exceção do código, devem ser armazenados da seguinte forma: TD:<NOME_DO_DOCUMENTO>SCL_2019.

Abaixo contém uma lista dos tipos de documento disponíveis.

| | |
|------|-------------------------|
| IMG | Para Imagens. |
| DOCX | Para documentos Word. |
| PPT | Para Slides. |
| PDF | Para arquivos PDF. |
| TXT | Para arquivos em Texto. |
| MOV | Para arquivos em Vídeo. |

Todos os documentos relacionados ao projeto devem ser armazenados na pasta “Documentos” disponível no DRIVE da fábrica.

7 Solicitação de mudança (CR) e Issues

Todas as mudanças relacionadas ao projeto devem ser registradas contendo o nome do autor da mudança, a descrição da mudança e a data de requerimento. Todas as alterações realizadas no product backlog devem ser registradas através da [ferramenta de issues](#) do Github.

As solicitações de mudança recebidas no meio de uma Sprint devem ser direcionadas ao P.B e aguardar a próxima Sprint para serem implementadas.

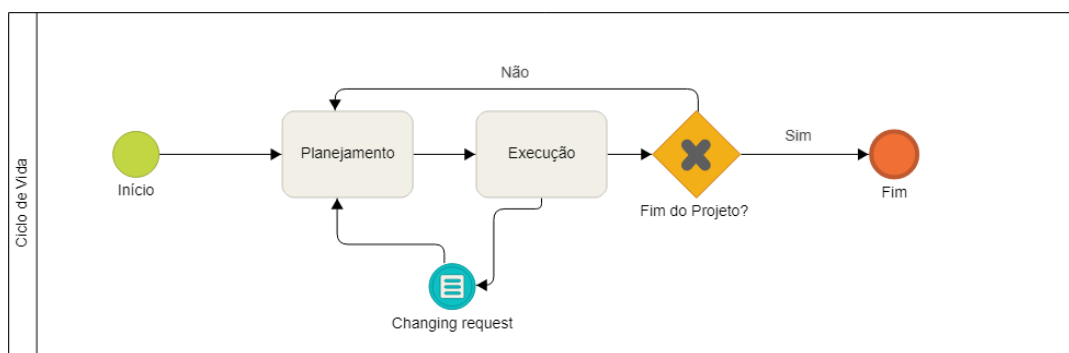
Todas as issue devem conter os seguintes campos:

| NOME | DESCRIÇÃO |
|----------|---|
| Title | Breve descrição do problema. |
| Content | Descrição detalhada do problema. |
| Assignee | Responsável pela resolução. |
| Labels | Esse campo refere-se ao tipo de Issues e sua prioridade. Os tipos de Issue são: Defeito - Issue de defeito; Melhoria - Issue de melhoria; Criação - Issue de Criação. |

8 Ciclo de Vida

8.1 Alto nível

O ciclo de vida do projeto é do tipo interativo-incremental e deve seguir o seguinte diagrama BPMN.



HEFLO

8.2 Baixo nível

O diagrama BPMN abaixo mostra com clareza todas as etapas do ciclo de vida junto com os setores responsáveis por cada uma delas.

