X-Soft Soluções

**StudyEach**

Plano de Gerência De Configuração

Versão 1.0

**Histórico de Alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Responsável** |
| 28/05/2016 | 1.0 | Criação do documento | João Gabriel |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Plano de Gerência de Configuração

Esse documento incorpora ao Plano de Projeto o Plano de Gerência de Configuração, que documenta as atividades e padrões de gerência de configuração a serem adotados durante o projeto.

## Política de Configuração

Nessa seção, são definidos os itens de configuração sob a responsabilidade da gerência de configuração e a nomenclatura para os identificadores dos itens. São apresentados, também a política para numeração de versões e para os *baselines* do projeto.

### Itens de configuração

A tabela abaixo identifica os itens que estarão sob a gerência de configuração, apresentando uma breve descrição de cada um.

|  |  |
| --- | --- |
| **Item de Configuração** | **Descrição** |
| Programas | Representam o código fonte responsável pela aplicação |
| Configuração | Representam todos os arquivos de configuração para que a aplicação seja executada |
| Artefato metodologia | Representam todos os documentos e arquivos gerados pelo projeto. |

### Identificação da configuração

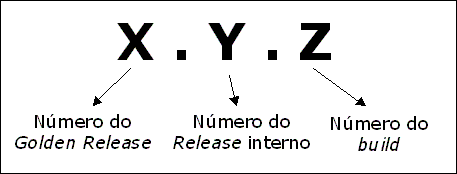
Deve ser seguida a seguinte convenção para nomenclatura de arquivos e artefatos na Estrutura de Diretórios do Produto.

**<STE>\_<AAA>\_<TextoLivre>.<EXT>**

|  |  |
| --- | --- |
| **Parte da Linha** | **Significado** |
| <STE> | Identifica o sistema. “STE - StudyEach” |
| <AAA> | Acrônimo de três letras dos vários tipos de artefatos definidos previamente na seção acrônimos deste documento. |
| <TextoLivre> | Texto Livre para a melhor identificação do documento. |
| <EXT> | Extensão do arquivo do documento. |

### Numeração das versões

Deve-se seguir o seguinte padrão para a numeração dos builds:



O número inicial do *build* é **0.0.0**. Quando cada *release* interno for lançado, o segundo dígito deve ser incrementado. Desse modo, o primeiro *release* interno é nomeado **0.1.0**. Quando um *build* (que não constitua um release interno) for gerado, deve-se incrementar o terceiro dígito. Desse modo, o primeiro *build* lançado depois do primeiro *release* interno é nomeado **0.1.1**. O segundo *release* interno será **0.2.0**. Quando for lançado o primeiro *Golden* *Release*, deve-se incrementar o primeiro dígito e zerar os demais dígitos, o que no caso resulta na versão de número **1.0.0**.

Cada final de Sprint gera um *release*. Builds podem ocorrer durante o Sprint quando a finalização de uma tarefa necessitar do mesmo ou impactar em demais tarefas.

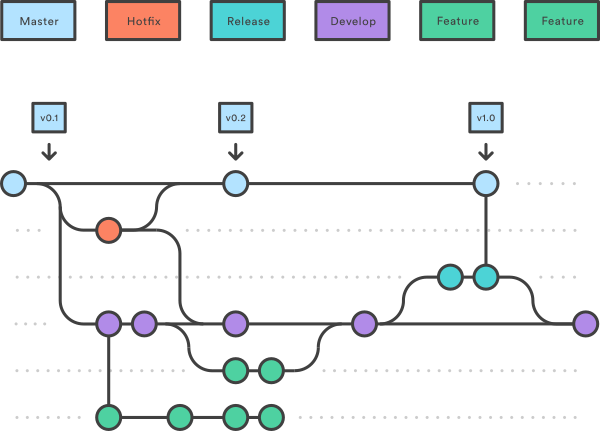
### Procedimentos para Integração Contínua

Essa seção define, essencialmente, como se dará o processo de construção de builds no projeto. Um build é uma versão operacional da aplicação que demonstra um subconjunto dos requisitos da aplicação final a ser desenvolvida, não constituindo necessariamente um release interno. Costuma-se dizer que a geração de builds atua como um medidor do batimento cardíaco do projeto.

Será utilizado no projeto o conceito de builds periódicos, onde a demanda de builds será definida, pela necessidade da realização de testes de unidade durante o desenvolvimento.

### Numeração e identificação de *branchs* e *commits*

Durante a execução desse projeto, todos os *branchs* gerados ou *commits* realizados devem seguir o padrão GitFlow. Uma breve descrição do padrão pode ser observada na imagem a seguir:



Os *branchs* devem ser nomeados de acordo com a tarefa que representa sua existência, seguindo o seguinte padrão:

**feature/TASK-<id\_da\_tarefa>**

|  |  |
| --- | --- |
| **Parte da Linha** | **Significado** |
| <feature> | Identificação do padrão gitflow para tarefas |
| <id\_da\_tarefa> | Identificador único da tarefa que gerou a task. |

Todo *commit* deve identificar o *branch* que originou as alterações a fim de facilitar os processos de revisão. Os textos de *commit* devem ser identificados da seguinte forma:

**<nome\_do\_branch> <texto>**

|  |  |
| --- | --- |
| **Parte da Linha** | **Significado** |
| <nome\_do\_branch> | Nome do branch relacionado ao *commit*. |
| <texto> | Texto explicativo sobre as alterações realizadas no *commit*. |

## Organização e Responsabilidades

|  |  |
| --- | --- |
| **Nome** | **Responsabilidades** |
| Gerente do Projeto | Executar todas as atividades do processo de Gerência de Configuração previstos na IEEE 828-2012 para os itens de configuração definidos |

## Atividades e Cronograma Macro

|  |  |  |
| --- | --- | --- |
| **Atividade** | **Responsabilidade** | **Data Término** |
| Geração do *release* 0.1.0 | Gerente do Projeto | 13/05/2016 |
| Geração do *release* 0.2.0 | Gerente do Projeto | 20/05/2016 |

## Estrutura do repositório

A estrutura de armazenamentos dos itens de configuração no repositório do projeto está definida no diagrama a seguir. A estrutura deve ser seguida na inserção e/ou atualização de qualquer item.

