X-Soft Soluções

**StudyEach**

Plano de Gerência De Configuração

Versão 1.2

**Histórico de Alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Responsável** |
| 28/05/2016 | 1.0 | Criação do documento | João Gabriel |
| 26/06/2016 | 1.1 | Alterações visando atender o Nível F do MPS.BR | Tiago Damascena |
| 01/12/2016 | 1.2 | Alterações para atender ao novo modelo de Gerência de Configuração | Tiago Damascena |
|  |  |  |  |
|  |  |  |  |

# Plano de Gerência de Configuração

Esse documento incorpora ao 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

## Identificação da configuração

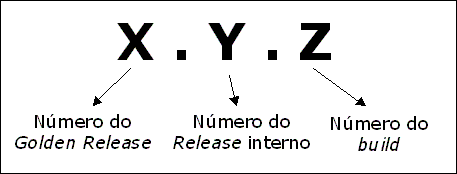
Deve ser seguida a seguinte convenção para nomenclatura de arquivos e artefatos na Estrutura de Diretórios do Produto.

**<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:

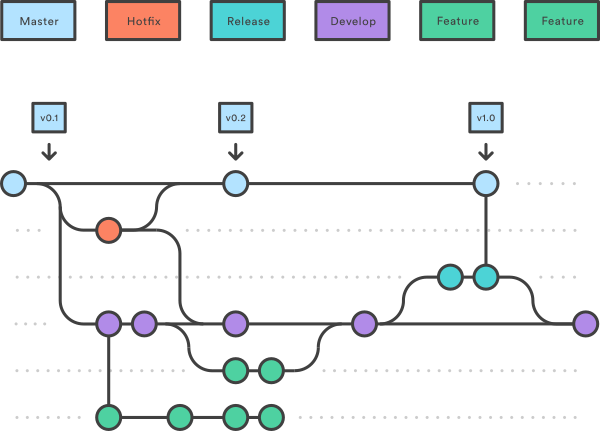


O número inicial do *build* é **0.0.0**. Quando cada *release* interno for lançado, o segundo dígito deve ser incrementado. Desse modo, o primeiro *release* interno é nomeado **0.1.0**. Quando um *build* (que não constitua um release interno) for gerado, deve-se incrementar o terceiro dígito. Desse modo, 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.

## 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-< nome\_da\_tarefa >**

|  |  |
| --- | --- |
| **Parte da Linha** | **Significado** |
| <feature> | Identificação do padrão gitflow para tarefas |
| <nome\_da\_tarefa> | Nome da task que está sendo desenvolvida. |

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

**<id\_tarefa> - <texto>**

|  |  |
| --- | --- |
| **Parte da Linha** | **Significado** |
| <id\_tarefa> | O id da issue no GitHub que essa tarefa implementa, seguindo o padrão #id para facilitar a rastreabilidade na ferramenta. |
| <texto> | Texto explicativo sobre as alterações realizadas no *commit*. |

## Entrega da funcionalidade

Quando o desenvolvimento de uma tarefa for finalizado o desenvolvedor deve solicitar um *pull request* no GitHub da *branch* dessa tarefa na *branch* develop. Esse pull request terá seus testes executados e a padronização de código checada automaticamente e caso passe nas verificações outro membro da equipe poderá avaliar a *branch* e fazer merge ou não.

## Fim de Sprint

Ao fim de cada Sprint deve ser realizada a auditoria de configuração que é explicada na seção 3 desse mesmo documento e a limpeza das branchs que já fizeram merge, tanto de novas funcionalidades quanto de correção.

# 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 |

# Auditoria

Ao final de cada Sprint deve ser feita a auditoria de configuração, visando verificar se os procedimentos e diretrizes estão sendo seguidos de forma correta e adequada, bem como se os itens de configuração e as *baselines* estão íntegras, corretas e consistentes.

A auditoria será realizada pelo Scrum Master através de um checklist com itens pré-definidos. Essa checklist pode ser acessada no link abaixo:

[Checklist de Avaliacao](https://github.com/NikolasMatias/PGP-2016/blob/develop/docs/Project%20Docs/STE_GQ_Checklists/STE_CA_ChecklistDeAvaliacao_Sprint-04.xlsx)