****

**Plano de Gerência de Configuração**

**Versão 1.2**

**Histórico de Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Descrição | Autor |
| 08/03/2013 | 0.1 | Criação do Documento | Vinícius Medeiros |
| 16/03/2013 | 0.2 | Ajustes no documento | Vinícius Medeiros |
| 26/03/2013 | 1.0 | Primeira versão do documento | Vinícius Medeiros |
| 16/04/2013 | 1.1 | Alterações no documento | Vinícius Medeiros |
| 13/05/2013 | 1.2 | Alterações no documento | Vinícius Medeiros |

Sumário

[**1.** **Introdução** 3](#_Toc353870373)

[**1.1.** **Propósito** 4](#_Toc353870374)

[**1.2.** **Escopo** 4](#_Toc353870375)

[**1.3.** **Definições, Acrônimos e Abreviações** 4](#_Toc353870376)

[**1.4.** **Visão Geral** 4](#_Toc353870377)

[**2.** **Gerenciamento da Configuração do Software** 4](#_Toc353870378)

[**2.1.** **Papéis** 4](#_Toc353870379)

[**2.2.** **Ferramentas, ambiente e infraestrutura** 5](#_Toc353870380)

[**2.2.1.** **Ferramentas para a Gerência de Configuração** 5](#_Toc353870381)

[**2.2.2.** **Ferramentas para o desenvolvimento do sistema** 5](#_Toc353870382)

[**2.2.3.** **Outras Ferramentas** 6](#_Toc353870383)

[**2.2.4.** **Estrutura do Ambiente** 6](#_Toc353870384)

[**2.3.** **Principais Expressões de Gerência de Configuração** 6](#_Toc353870385)

[**2.4.** **Repositório** 7](#_Toc353870386)

[**2.4.1.** **Estrutura do repositório** 8](#_Toc353870387)

[**2.4.2.** **Controle de Acesso** 8](#_Toc353870388)

[**2.5.** **Identificação dos Artefatos** 9](#_Toc353870389)

[**2.6.** **Identificação dos Commits** 9](#_Toc353870390)

[**3.** **Controle de Mudanças** 9](#_Toc353870391)

1. **Introdução**

Este plano estabelece a integridade dos artefatos do SGCONT – Sistema Gerencial de Contabilidade, permitindo o acompanhamento destes itens durante todo o ciclo de vida do projeto, preservando o histórico de evolução do sistema e auxiliando a gerenciar as mudanças e o estado dos itens de configuração e rastrear modificações nos itens de configuração ao longo do tempo.

* 1. **Propósito**

Este documento descreve a organização, nomenclatura e regras para o versionamento de artefatos do projeto SGCONT.

* 1. **Escopo**

Este Plano de Gerenciamento de Configuração é destinado para todos os integrantes da equipe responsável pela manutenção do sistema SGCONT e abrange todas as informações sobre o controle e gerenciamento dos artefatos do projeto.

* 1. **Definições, Acrônimos e Abreviações**

|  |  |
| --- | --- |
| Termo | Significado |
| SVN | Subversion (Sistema de Controle de Versão) |
| Scrum | É um processo ágil que permite manter o foco na entrega do maior valor de negócio, no menor tempo possível. |
| GCS | Abreviação para ‘Gerência de Configuração de Software’ |
| SM | Solicitação de Mudança (Redmine) |

* 1. **Visão Geral**

Os próximos capítulos deste documento estão divididos conforme a tabela abaixo:

|  |  |
| --- | --- |
| Sessão | Descrição |
| 2 | São relacionados os papéis, as responsabilidades das atividades e as ferramentas utilizadas no projeto. |
| 3 | São abordados os detalhes sobre a identificação dos artefatos no projeto, assim como a ferramenta de Controle de Mudanças. |

1. **Gerenciamento da Configuração do Software**
   1. **Papéis**

|  |  |  |  |
| --- | --- | --- | --- |
| Papéis | | Equipe | Responsabilidades |
| Gerência de Projeto | | Sávio Luiz | - Gerenciar integrantes  - Estimar prazos |
| Requisitos | | Vivianne Sousa  Rômulo Aurélio | - Gerar documentação |
| Solução Técnica | Arquitetura | Mariana Victor  Vinícius Medeiros | - Definir ferramentas  - Configurar ambiente  - Definir estrutura de desenvolvimento |
| Análise & Projeto | Vivianne Sousa  Sávio Luiz | - Gerar documentação  - Análise de regras de negócio |
| Implementação | Vivianne Sousa  Mariana Victor  Rômulo Aurélio | **-** Desenvolver sistema |
| Gerência de Configuração | | Vinícius Medeiros | - Controle de versão  - Controle de mudanças  - Auditoria do repositório |
| Testes | | Sávio Luiz  Rômulo Aurélio | - Criar Planos de testes  - Realizar testes  - Desenvolver testes automáticos |

* 1. **Ferramentas, ambiente e infraestrutura**

* + 1. **Ferramentas para a Gerência de Configuração**

|  |  |  |  |
| --- | --- | --- | --- |
| Ferramenta | Tipo | Descrição | Versão |
| Servidor Google Code | Web | Armazenar código fonte, documentos, artefatos e versões do sistema. | N/D |
| Subversion | Controle de versões | Sistema utilizado para controlar o versionamento dos artefatos do projeto. | 1.7.9 |
| TortoiseSVN | Ferramenta de acesso ao SVN | Software utilizado como Cliente do Subversion para Windows. | 1.7.11 |
| Redmine | Controle de Mudanças | Sistema utilizado para gerenciar as mudanças ocorridas no projeto. | N/D |

* + 1. **Ferramentas para o desenvolvimento do sistema**

|  |  |  |
| --- | --- | --- |
| Tipo | Ferramenta | Versão |
| Plataforma de Desenvolvimento | Eclipse IDE | Juno |
| Framework: Hibernate 4, Spring 3, JSF 2, Primefaces 3.5 | N/D |
| Linguagem: Java (J2EE) | JDK 1.7 |
| Banco de Dados | PostgreSQL | 9.2.3 |
| Relatórios | API: JasperReports / IDE: IReport | N/D |
| Servidor de Aplicação | Apache Tomcat | 7.0.33 |
| Comunicação | Gtalk / Gmail / Hangout | Beta |

* + 1. **Outras Ferramentas**

|  |  |
| --- | --- |
| Ferramenta | Descrição |
| Microsoft Word | Utilizado para o desenvolvimento de documentos, casos de uso e demais artefatos do projeto. |
| Microsoft Project | Utilizado para a realização do acompanhamento das atividades de cada participante do projeto. |
| Google Docs | Utilizado para colaborar com o processo de atribuição e gerenciamento de atividades. |

* + 1. **Estrutura do Ambiente**

|  |  |  |
| --- | --- | --- |
| Ambiente | Descrição | Transição |
| Desenvolvimento | É o ambiente que servirá para o desenvolvimento do sistema, correção, modificação e etc. | O componente atingirá a maturidade quando os requisitos forem supridos e testados pelos desenvolvedores através de testes de carga, unitários e etc. |
| Banco de Dados | É o ambiente onde conterá o Banco de dados. | Ambiente que conterá o Banco de dados do sistema. |
| Testes | É o ambiente onde serão realizados testes em cima de: Funcionalidades, correções e evoluções do sistema. | Ambiente onde serão realizados os testes. |

* 1. **Principais Expressões de Gerência de Configuração**

O processo de Gerência de Configuração utiliza termos técnicos que serão bastante ouvidos e vivenciados no dia-a-dia.

- **Trunk** –Linha principal do projeto. Nela estão presentes os fontes mais estáveis do projeto.

- **Branch** – Ramificações do projeto. Caso haja a necessidade de serem efetuadas alterações impactantes, um novo branch é gerado, permitindo o desenvolvimento paralelo do sistema, evitando, com isso, conflito entre estas alterações e a linha principal.

- **Tag** – São todas as versões publicadas do sistema (etiquetas).

- **Versão Oficial e Release** –Sempre que forem liberadas versões para os clientes, estas poderão ser OFICIAIS ou RELEASES. As versões oficiais são todas aquelas planejadas, com data pré-definida e alterações já previstas. Já os releases contêm apenas correções, e apesar de possuir datas definidas, podem sofrer alterações, já que dependem da urgência dessas.

- **Repositório** –Local de armazenamento de todos os artefatos do Sistema. Será detalhado no tópico 2.3.

Além dos termos supracitados, há também aqueles utilizados na ferramenta de controle de versão (Tortoise ou RapidSVN). São eles:

- **Update ­**– Processo de atualização da cópia de trabalho local através do download das mudanças efetuadas por outros usuários no repositório.

- **Commit** –Processo de envio das modificações feitas pelo usuário na cópia de trabalho local para o repositório.

- **Merge** – Processo de fusão de modificações feitas por diferentes usuários em um mesmo arquivo. O merge pode ser realizado automaticamente, quando simplesmente efetuado o update em um arquivo modificado localmente; ou manualmente, caso seja necessária a fusão de repositórios diferentes.

- **Add** – Usado para agregar um artefato não-versionado à cópia de Trabalho local.

- **Switch** – Utilizado para efetuar a mudança do caminho do Repositório apontado na cópia de trabalho local.

- **Show Log** – Utilizado para verificar o histórico do Repositório. Através dessa opção é possível visualizar desde um arquivo específico até mesmo todo o repositório.

* 1. **Repositório**

O repositório do SGCONT, assim como já levantado superficialmente no tópico 2.3 deste capítulo, está subdividido em três grandes blocos: TRUNK, BRANCHES e TAGS. Essa divisão, além de seguir a padronização proposta pelo SVN, organiza significativamente o controle de versões, já que facilita a busca de informações sobre mudanças e permite um maior controle sobre todas as versões de arquivos. Essa divisão objetiva ainda estabelecer às linhas de desenvolvimento regras, como onde será feita a alteração, quais grupos poderão acessar aquela linha, etc.

* + 1. **Estrutura do repositório**

Em uma fábrica de *software*, o repositório é responsável por armazenar todos os artefatos produzidos, sejam eles documentos, imagens ou arquivos de desenvolvimento (classes em Java, relatórios compiláveis, etc). Através do processo de versionamento, um determinado arquivo – alocado no repositório – pode ser manipulado por diferentes usuários, sendo cada conjunto de alterações salva como uma nova revisão.

Abaixo segue estrutura do repositório SGCONT:

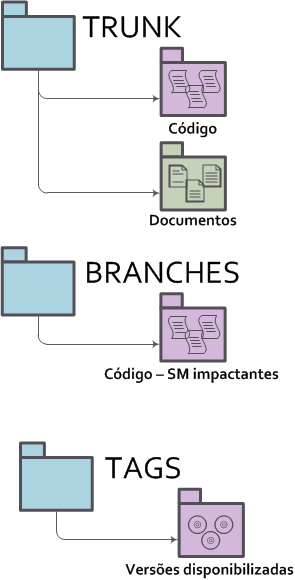
****

Figura 1: Organização do Repositório SGCONT

A estrutura acima define onde os artefatos estarão situados no repositório. A linha de desenvolvimento principal – de onde as versões serão geradas – será o Trunk. Neste diretório também estarão presentes toda a documentação do projeto. Na pasta Branches haverá todas as linhas de desenvolvimento para construção de Solicitações de Mudança impactantes. A pasta Tags conterá todas as versões publicadas.

* + 1. **Identificação dos Artefatos do Repositório**

A fim de padronizar os artefatos criados no repositório, devem-se usar os seguintes modelos:

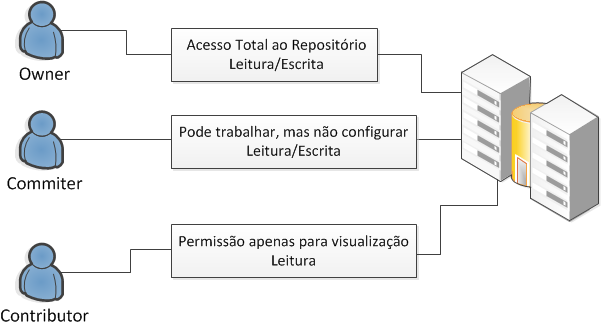
**- Branch** – Todos os branches – criados sempre que houver uma Solicitação de Mudança impactante, conforme supracitado – devem usar o seguinte padrão:

<NÚMERO\_DA\_SM>-<NOME\_FUNCIONALIDADE>-<DATA(AAAAMMDD)>

**- Tag –** Possuirão a sua nomenclatura orientada pela

* + 1. **Controle de Acesso**

Cada repositório – ao ser criado – precisa ser configurado para permitir que o SVN saiba quem poderá acessá-lo. Para isso, o Google Code permite que cada usuário seja definido conforme:



**Repositório**

Figura 2: Organização do Repositório por grupo de acesso.

* 1. **Identificação dos Artefatos**

A fim de manter a organização dos artefatos do SGCONT, todos os documentos seguirão a um determinado padrão de nomenclatura. Abaixo segue a estrutura a ser adotada:

<TIPO\_DOCUMENTO>-<PROJETO>-<DATA\_CRIACAO>.<EXTENSÃO\_ARQUIVO>

Ex.: ATA-SGCONT-04032013.doc

* 1. **Identificação dos Commits**

Os commits deverão ser identificados a partir das solicitações de mudança, que serão atreladas a todas as submissões ao repositório. Para que haja a rastreabilidade entre os artefatos e as solicitações, todos os commits seguirão o seguinte padrão:

<RM>: <DESCRICAO\_DA\_ALTERACAO>

1. **Controle de Mudanças**

O processo de Gerência de Configuração também é responsável pelo controle de mudanças. Para entender o processo de mudança, deve-se entender que cada *commit* realizado deverá satisfazer a uma solicitação de mudança. No âmbito do SGCONT, o controle de mudança é realizado com o auxilio da ferramenta Redmine, sendo cada solicitação aberta intitulada RM. Abaixo segue imagem do Redmine, utilizado no projeto SGCONT:

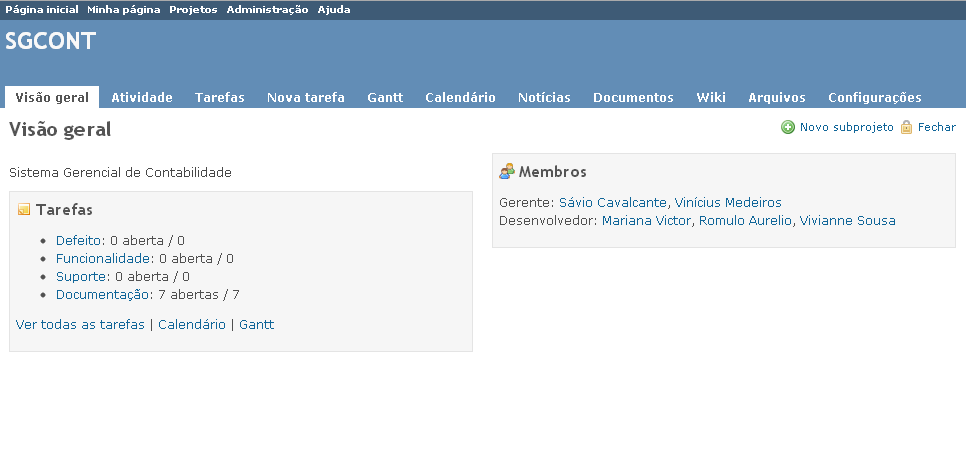


Figura 3: Visão geral do Redmine