**PS-MCTIC**

**Guia Operacional – Políticas de GC**

**Versão 1.2**

|  |
| --- |
| **Histórico de Revisões** |
| **Versão** | **Data** | **Descrição** | **Autor** |
| 1.0 | 19/01/2015 | Padronização do formato do Guia Operacional de GC já existente no MCTI. | RSI Informática |
| 1.1 | 28/01/2015 | Inclusão da referência do PS-MCTI | RSI Informática |
| 1.2 | 28/03/2017 | Atualização da sigla MCTI para MCTIC e descrição pela fusão do Ministério das Comunicações. | CTIS |

Sumário

[1. INTRODUÇÃO 4](#_Toc478459628)

[2. IDENTIFICAÇÃO DA CONFIGURAÇÃO 4](#_Toc478459629)

[2.1. Conceitos básicos: 4](#_Toc478459632)

[2.2. Identificação de Itens de Configuração (IC) 4](#_Toc478459633)

[2.3. Identificação de Configurações Base (CB) para Documentos 5](#_Toc478459634)

[2.4. Identificação de Configurações Base (CB) para Código Fonte 6](#_Toc478459635)

[3. CONTROLE DE VERSÃO 6](#_Toc478459636)

[3.1. Controle de Versões de Documentos 7](#_Toc478459638)

[3.2. Controle de Versões de Código Fontes 8](#_Toc478459639)

[4. ESTRUTURA DE ARMAZENAMENTO E POLÍTICAS DE ACESSO 9](#_Toc478459640)

[4.1. Estrutura de Diretórios do Repositório de “Desenvolvimento” 9](#_Toc478459643)

[4.2. Estrutura de Diretórios do Repositório de “Gestão” 10](#_Toc478459644)

[5. COMANDOS BÁSICOS NO REPOSITÓRIO DE “DESENVOLVIMENTO” 10](#_Toc478459645)

[6. ROTINAS DE BACKUP E RESTORE 11](#_Toc478459646)

[7. REFERÊNCIAS 11](#_Toc478459647)

# INTRODUÇÃO

Este guia tem por finalidade descrever requisitos e procedimentos necessários para as atividades de Gestão de Configuração (GC). O guia define método para geração e controle de Itens de Configuração (IC) e Configurações Base (CB) e apoia a execução da atividade Planejar Gestão de Configuração do Processo de Software (PS-MCTIC).

# IDENTIFICAÇÃO DA CONFIGURAÇÃO

Identificação da configuração aplica-se a todo produto de software, seja ele: código, documentos, planilhas, procedimentos, binários, etc.

1.
2.

## Conceitos básicos:

**Item de Configuração (IC):** “Cada um dos elementos de informação que são criados durante o desenvolvimento de um produto de software, ou que para este desenvolvimento sejam necessários, que são identificados de maneira única e cuja evolução é passível de rastreamento” (Pressman, 1992).

**Configurações-Base (CB):** Um conjunto bem definido de itens de configuração que representam um estágio do desenvolvimento.

##  Identificação de Itens de Configuração (IC)

A Identificação única de Itens de Configuração é de extrema importância e desta forma facilita a localização e identificação da versão correta de qualquer Item de Configuração do projeto.

Todos os produtos gerados durante o ciclo de vida do projeto são Itens de Configuração (ICs).

Os ICs que serão controlados, com exceção de códigos fonte, devem utilizar a seguinte estrutura de rótulo de identificação única:

**<Sigla do Sistema>** - **<Sigla do Artefato>** - **<Nome do Item>**

Onde:

**<Sigla do Sistema>** corresponde à sigla que identifica o Sistema;

**<Sigla do Artefato>** corresponde à sigla que identifica o artefato (Opcional);

**<Nome do Item>** corresponde ao nome que identifica o item.

Exemplo:

* **SIGMCT - Caso de Desenvolvimento**
* **SIGMCT - UC - Efetuar Login**

Os ICs correspondentes a códigos fontes deverão ter sua nomenclatura definida e documentada no documento Guia de Implementação.

## Identificação de Configurações Base (CB) para Documentos

Foram definidos dois tipos de configurações base para agrupar os produtos gerados durante o ciclo de vida do projeto.

**Tipos de Configurações Base:**

* **DES - Desenvolvimento** = Configuração base de desenvolvimento. Conjunto de Itens de Configuração em desenvolvimento. Essa configuração base é composta de ICs entregues.
* **DIS - Distribuição** = Configuração base de distribuição. Conjunto de Itens de Configuração homologados. Essa configuração base é composta de ICs homologáveis.
	+ **T - Total =** Conjunto de Itens de Configuração homologados em sua totalidade.
	+ **P - Parcial =** Conjunto de Itens de Configuração homologados parcialmente, alguns ICs pendentes de homologação.

As configurações base devem ser agrupadas por Ordem de serviço (OS) e ter o rótulo identificador conforme descrito abaixo:

**<E>-<xxx>-<Tipo>-<Forma>**

**E**: Entrega.

**xxx**: Sequencial referente ao Termo de Entrega de Produto.

**Tipo**: Tipo de configuração base.

**Forma**: Total ou Parcial (Utilizado somente para o Tipo de configuração base **DIS**)

Exemplo:

**E-001-DES** (Entrega 001)

**E-001-DIS-P** (Entrega 001 parcialmente homologada)

**E-002-DES** (Entrega 002)

**E-002-DIS-T** (Entrega 002 totalmente homologada)

A Figura abaixo mostra a estrutura do repositório para baseline de documentos:



## Identificação de Configurações Base (CB) para Código Fonte

No caso de produtos empacotados como códigos de programas, foi definida a seguinte configuração base para o rótulo do pacote todo.

A composição básica da nomenclatura será conforme descrito abaixo:

**<PRINCIPAL>.<MELHORIAS E FUNCIONALIDADES>.<MANUTENÇÕES>-<NUMERO OS>**

* **PRINCIPAL:** Oprimeiro número representa a versão atual do seu produto, ao alterar este primeiro número significa que uma grande mudança ocorreu, nova interface, novas soluções.
* **MELHORIAS E FUNCIONALIDADES:** O segundo número representa melhorias e novas pequenas funcionalidades do produto atual.
* **MANUTENÇÕES:** O ultimo número, representa a correção de bugs, não contém novas funcionalidades planejadas, apenas estabilização do produto.
* **NUMERO OS:** Numero da ordem de serviço

Exemplo:

**Primeira Release: 1.0.0-OS001 (Release para atender a demanda OS001)**

**Segunda Release: 1.0.1-OS001 (Correção para atender a demanda OS001)**

**Terceira Release: 1.0.2-OS002 (Correção para atender a demanda OS001)**

**Quarta Release: 1.1.0-OS003 (Evolução para atender a demanda OS003)**

**Quinta Release: 2.0.0-OS010 (Mudança de tecnologia para atender a demanda OS010)**

A Figura abaixo mostra a estrutura do repositório para baseline de documentos:

****

# CONTROLE DE VERSÃO

O controle de versões dos ICs “não entregáveis” é realizado pela própria ferramenta de controle de versões e terá histórico das alterações documentadas no comentário de cada versão criada.

1.

## Controle de Versões de Documentos

Os itens de configuração (documentação) serão versionados da seguinte forma:

1. Enquanto o item de configuração estiver em elaboração terá sua versão inicial **1.0**. Após a elaboração o mesmo passará pelo processo de **revisão interna** antes da entrega, não sofrendo ajustes na versão, mesmo que tenha alterações a serem feitas.
2. Após a entrega, o item de configuração será verificado, caso haja necessidade de ajustes para atender a **mesma demanda**, o Item de Configuração será versionado somente à **segunda casa decimal**, até que o mesmo seja homologado.
3. Em caso de necessidade de alteração do item de configuração para atendimento de **outra demanda**, o mesmo deve ser versionado à **primeira casa decimal**, automaticamente **zerando a segunda casa decimal**, passando pelo processo de **revisão interna** antes da entrega, não sofrendo ajustes na versão, mesmo que tenha alterações a serem feitas. O item de Configuração passará pelo mesmo processo descrito no **item 2**, até a sua devida homologação.

Exemplo:

* Primeira versão: 1.0 (Elaboração para atender a demanda OS001)
* Segunda versão: 1.1 (Correção para atender a demanda OS001)
* Terceira versão: 1.2 (Correção para atender a demanda OS001)
* Quarta versão: 2.0 (Alteração para atender a demanda OS003)
* Quinta versão: 3.0 (Alteração para atender a demanda OS010)

### Fluxo de Versionamento de Documentos



## Controle de Versões de Código Fontes

### Desenvolvimento em Paralelo

O desenvolvimento paralelo é derivação a partir de determinada versão de um arquivo que pode gerar caminhos paralelos ao caminho principal de desenvolvimento. Isso tem utilidade quando se deseja incorporar características particulares em determinadas versões e não em outras, que poderão ser incorporadas ou não às demais versões, conforme figura abaixo.



O código a ser entregue será controlado através da criação de um novo caminho paralelo de evolução/correções (***branches***) a partir do caminho principal. O código em homologação (caminho principal) deve estar integrado na pasta (***Trunk***).

### Fluxo de Codificação em Paralelo



# ESTRUTURA DE ARMAZENAMENTO E POLÍTICAS DE ACESSO

Para manter os IC dos projetos existe um repositório de Desenvolvimento e um repositório de Gestão:

1. A estrutura do repositório de “**Desenvolvimento**” é criada e mantida pela ferramenta **Tortoise SVN**.
2. A estrutura do repositório de “**Gestão**” é criada e mantida pela ferramenta **EPM**.

**Importante**:

1. O detalhamento **interno** da estrutura pré-estabelecida é liberado de acordo com a necessidade da equipe de desenvolvimento.
2. A pasta “11-Documentos nao Padronizados” deve ser utilizada para armazenar documentos diversos que não precisam de controle formal como, documentos de apoio recebido dos clientes, etc.
3.
4.

##  Estrutura de Diretórios do Repositório de “Desenvolvimento”

 

Esta pasta tem o objetivo de manter todos os documentos não padronizados. Esses documentos não são controlados e, portanto, não precisam seguir o padrão de nomeação de IC.

### Política de Acesso

|  |
| --- |
| **ESTRUTURA DE DIRETÓRIOS** |
| **Diretório** | **Usuário** | **Acesso** |
| branches | Equipe de projeto | R |
| Equipe de GC | R, W |
| Fábrica contratada | R |

(Acesso) direitos de acesso: R (read), W (escrita).

|  |
| --- |
| **ESTRUTURA DE DIRETÓRIOS** |
| **Diretório** | **Usuário** | **Acesso** |
| tags | Equipe de projeto | R |
| Equipe de GC | R, W |
| Fábrica contratada | R |

(Acesso) direitos de acesso: R (read), W (escrita).

|  |
| --- |
| **ESTRUTURA DE DIRETÓRIOS** |
| **Diretório** | **Usuário** | **Acesso** |
| trunk | Equipe de projeto | R |
| Equipe de GC | R, W |
| Fábrica contratada | R |

(Acesso) direitos de acesso: R (read), W (escrita).

## Estrutura de Diretórios do Repositório de “Gestão”

A Definir

### Política de Acesso

A Definir

# COMANDOS BÁSICOS NO REPOSITÓRIO DE “DESENVOLVIMENTO”

De maneira geral, as ferramentas de controle de versão organizam os acessos ao repositório por meio de um conjunto de operações básicas conhecidas como **Check Out - Update - Commit**.

1. **Check out** – obtém cópia de trabalho local da versão desejada do IC para modificação.
2. **Update –**atualiza cópia de trabalho local.
3. **Get Lock –** bloqueia o arquivo, impedindo o commit de outro usuário para o arquivo bloqueado.
4. **Release Lock –** desbloqueia o arquivo, permitindo o commit de outro usuário para o arquivo.
5. **Commit** – cria nova versão do IC a partir de cópia de trabalho modificada (obtida por check out).

Nesta operação o usuário deve obrigatoriamente registrar um comentário que descreva o que foi alterado ou incluído:

**<Numero Ordem de Serviço > - <Descrição da alteração>**

Exemplo:

**OS001 - Alteração da funcionalidade decorrente da inclusão da regra de negócio XX.**

**Boas práticas:**

1. Sempre antes de modificar um arquivo na área de trabalho local, sempre fazer o comando Update.

A operação de ***Commit*** deve ser realizada pelo menos uma vez ao dia. É recomendado que seja realizada no final do dia de trabalho.

# ROTINAS DE BACKUP E RESTORE

Os procedimentos de backup e restore dos repositórios devem ser executados conforme descritos na metodologia adotada pelo MCTIC - Ministério da Ciência, Tecnologia e Inovação.

# REFERÊNCIAS

* PS-MCTIC – Processo de Software do Ministério da Ciência, Tecnologia, Inovações e Comunicações
* PS-MCTIC – ITQ – Gestão de Configuração
* PS-MCTIC – GO – Ferramenta de GC