

CAREFOUR

Plano de Gerenciamento de Configuração

Versão 1.0

Histórico da Revisão

Data	Versão	Descrição	Autor
<dd/mm/aa>	<x.x>	<detalhes>	<nome>

Plano de Gerenciamento de Configuração

1. Introdução

A aplicação deste plano garante a integridade de códigos-fonte e demais produtos do sistema CAREFOUR, permitindo o acompanhamento destes itens durante todo o ciclo de vida do projeto, e preservando o histórico de evolução do sistema. Este Plano de Gerência de Configuração auxilia os componentes do grupo de desenvolvimento a:

- gerenciar o estado dos itens de configuração do sistema;
- controlar as mudanças em itens de configuração;
- rastrear modificações nos itens de configuração ao longo do tempo.

Apesar de serem englobadas pela Gerência de Configuração, as seguintes informações não serão tratadas neste plano, e deverão ser estabelecidas em momento futuro:

- itens de configuração de hardware;
- itens configuração de software;
- itens de configuração de rede;
- políticas de controle de mudanças.

1.1 Finalidade

O Plano de Gerência de Configuração tem aplicação direta nas atividades dos componentes do grupo de desenvolvimento e manutenção do sistema. No entanto, as informações aqui contidas servirão de referências para profissionais de outras áreas como: produção e operação, testes, homologação, documentação, entre outras.

1.2 Escopo

Este documento descreve toda a infra-estrutura utilizada durante o desenvolvimento do projeto CAREFOUR.

1.3 Definições, Acrônimos e Abreviações

Esta seção explica o conceito de alguns termos importantes que serão mencionados no decorrer deste documento. Estes termos são descritos na tabela a seguir, estando apresentados por ordem alfabética.

Termo	Descrição
BaseLine	Conjunto de artefatos que recebe uma aprovação de estabilidade. Um baseline é usado como uma base no desenvolvimento das próximas fases dos artefatos e tem suas modificações controladas por um processo.
CR	Solicitação de Mudança (Change Request)
CVS	Sistema de Controle de Versão (Control Version System)
SCMP	Plano de Gerência de Configuração de Software (Software Configuration Management Plan)

1.4 Referências

- DATASUS - Plano de Gerência de Configuração de Software, versão 02
- PRODATER - Plano de Gerenciamento de Configuração, versão 1.0
- SIGEQ - Plano de Gerenciamento de Configuração, versão 1.5
- SYSCO - Plano de Gerência de Configuração de Software, versão 0.01
- VENSSO - Plano de Gerência de Configuração de Software, versão 1.0

1.5 Visão Geral

As próximas seções deste documento estão divididas conforme a tabela abaixo.

Seção	Descrição
2	São relacionados os papéis, as responsabilidades das atividades e as ferramentas relacionadas ao projeto.
3	Descreve os métodos de identificação dos artefatos do projeto e apresenta como serão criadas e controladas as <i>Baselines</i> .
4	São abordados os detalhes sobre quando o Plano de Gerenciamento de Configuração deve ser atualizado.

2. Gerenciamento de Configuração de Software

2.1 Organização, Responsabilidades e Interfaces

2.1.1 Organização

A equipe do sistema irá manter os itens de configuração no ambiente de desenvolvimento do repositório do sistema. As atividades de manutenção dos itens de configuração envolvem a inclusão, exclusão e a modificação dos itens de configuração do sistema.

2.1.2 Responsabilidades

As responsabilidades da equipe de gestão de configuração envolvem as seguintes itens:

- estruturar o ambiente de desenvolvimento no repositório do sistema e identificar e controlar os itens de configuração;
- estabelecer as versões do sistema, definindo números únicos para cada versão e rotulando os itens de configuração que a compõe com referências aos números de versão;
- apoiar a equipe do sistema no uso das ferramentas de gerência de configuração;
- elaborar registro das alterações implementadas em cada versão disponibilizada do sistema (changelog).

2.2 Ferramentas, Ambiente e Infra-estrutura

2.2.1 Ferramentas

A ferramenta que será utilizada para realizar a gestão de configuração do sistema é a ferramenta de controle de versão GitHub. GitHub é um Serviço de Web Hosting Compartilhado para projetos que usam o controle de versionamento Git. É escrito em Ruby on Rails pelos desenvolvedores da Logical Awesome (Chris Wanstrath, PJ Hyett e Tom Preston - Wernder). O GitHub possui planos comerciais e gratuitos para projetos de código aberto. Este site possui funcionalidades de uma rede social como feeds, followers, wiki e um gráfico que mostra como os desenvolvedores trabalham as versões de seus repositórios.

O repositório tem como Raiz a pasta TP_CareFour e dentro desta pasta os arquivos serão divididos pelas suas funcionalidades.

2.2.2 *Ambiente*

Local de trabalho da equipe do sistema, onde ocorre a elaboração e atualização dos itens de configuração. Toda a equipe tem acesso aos itens de configuração deste ambiente, sendo possível a implementação de controle de permissões por diretório, caso seja necessário. Inicialmente, o ambiente de desenvolvimento do repositório é representada por 1 diretório apenas

- /documentos :
Pasta onde constaram os documentos gerencias referentes a Gestão de Configuração do sistema.
- /testes:
Pasta onde encontraram os documentos referentes aos planos de testes.
- /Codigo:
Pasta responsavel pela postagem do codigo utilizado no desenvolvimento do sistema.

2.2.3 *Infra-estrutura*

Cada integrante da equipe de Gestão de configuração deve possuir uma copia do repositório de dados em uma ambiente local para realizar as atividades de controle de versão. Deve também possuir acesso à internet e acesso ao GitHub para que possam realizar as alterações necessarias. É de total responsabilidade dos integrantes da equipe manter os repositórios locais para que possam executar seu trabalho de forma segura e eficiente.

2. O Programa de Gerenciamento de Configuração

3.1 Identificação da Configuração

3.1.1 *Métodos de Identificação*

Todos os itens de configuração do sistema, devem ser identificado de acordo com a nomenclatura abaixo:

<ARTEFATO>_<DATA>

Em que:

<ARTEFATO>: é a identificação do artefato (ver Tabela de Identificação)

<DATA> é a data da criação artefato, ou seja, a data da primeira versão do artefato em questão. Para o formato da data use DDMMAAAA, como por exemplo 11032014, que corresponde ao dia 11 de março de 2014.

Tabela de Identificação

Artefato	Identificação
Caso de Teste	TST
Modelo de classes	CMD
Modelo Entidade-Relacionamento	ER
Plano de Gerência de Configuração de Software	SCM
Plano de Testes de Software	STP

3.1.2 *Baselines do Projeto*

As Baselines do projeto podem ser definidas usando sempre usando uma <SEQ> se referindo ao número da versão lançada de determinado componente do software. É gerado um número sequencial incrementando a cada vez que uma solicitação e mudança impacta em versões do produto.

O número da versão dos artefatos deve seguir o seguinte padrão:

X.YY

X é um número decimal que representa uma versão final do artefato;

YY é um número hexadecimal que representa um draft da versão X do artefato.

O número de versão dos artefatos muda de acordo com tais regras:

- A versão inicial do artefato deve ser 0.01;
- Toda vez que houver uma modificação no artefato, o YY deve ser incrementado;
- Quando a versão X for incrementada, YY retorna para 00.

4. Marcos

10/04/2014 – Entrega do plano de testes

26/05/2014 - Execução dos testes