UTFPR-UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Bacharelado em Engenharia de Software - 6º Período

DISCIPLINA: *Gerência De Configuração - SO36B -ES61*

PROFESSOR: *Alexandre L'Erario*

Plano de gerenciamento de configuração

Cassio Pereira

Caroline Marques

Lucas Malheiros

Pedro Henrique Bernardes

Cornélio Procópio

2021

Projeto	[Sigla – Nome]
Gerente de Projetos	[Nome]
Fábrica de Software	[Sigla – Nome]

Cornélio Procópio
2021

Histórico de Revisões

Data	Versão	Descrição	Autor
11/08/2016	0.1	Elaboração do documento.	

SUMÁRIO

1. INTRODUÇÃO

O Plano de Gerenciamento de Configuração descreve todas as atividades do Gerenciamento de Controle de Configuração e Mudança que serão executadas durante o ciclo de vida do produto. Suas atividades envolvem identificar a configuração do software, manter sua integridade durante o projeto e controlar sistematicamente as mudanças, sendo apresentado os objetivos do projeto, suas premissas, itens de dentro e fora do escopo, etapas, cronograma e profissionais envolvidos.

1.1. Objetivos

O objetivo deste documento é criar um padrão a ser seguido por todos os membros da equipe com o intuito de garantir o maior controle do produto no decorrer do projeto, remover a funcionalidade de geração de relatórios referentes às informações contidas no banco de dados de um sistema já implementado, sendo justificada essa alteração pelo fato da funcionalidade não ser mais necessária.

1.2. Escopo

Estão no escopo do projeto as seguintes atividades: Iniciar processo de planejamento do projeto; Concluir planejamento; Implementar alterações; Realizar testes das alterações; Adicionar alterações no ambiente de avaliação; Entregar para avaliação e migrar aplicação para ambiente final.

Como o sistema foi implementado anteriormente a este documento, o escopo do projeto não cobrirá a fase de design (levantamento de requisitos). Por não haver previsão no processo de desenvolvimento, a manutenção também não estará presente.

1.3. Definições, Acrônimos e Abreviações

Termo	Descrição
RUP	*Rational Unified Process*. Processo de engenharia de software da IBM.
MDS	Metodologia de Desenvolvimento de Software.
Baseline	Linha de base. Conjunto de versões de itens de configuração comprovadamente estáveis. Uma *baseline* é usada como base no desenvolvimento da próxima fase do artefato e tem suas mudanças controladas por um processo formal.

1.4. Referências

- *Template* do Plano de Gerenciamento de Configuração, RUP 7.0, IBM.

- Plano do Projeto
  <https://docs.google.com/document/d/1_iZeAaWGVygxQApEOsgOXP8aMST0EhE4/edit?usp=sharing&ouid=103956549708444126467&rtpof=true&sd=true>
- Cronograma do Projeto
  https://github.com/AbyssalSire/gerencia-de-configuracoes/blob/main/artefatos/estudo-viabilidade.pdf

1.5. Evolução

O Plano de Gerenciamento de Configuração deve ser mantido atualizado para refletir o planejamento corrente. Dessa forma, as seguintes situações representam gatilhos para atualização do plano e nova aprovação deste documento:

- Mudança nos itens de configuração;
- Mudança na identificação dos arquivos;
- Mudança na identificação das *Tags*/*Branches*;
- Mudança no padrão de versionamento;

2. Gerência de Configuração de Software

2.1. Organização, Responsabilidades e Interfaces

Funções	Responsabilidades
Gerente	Responsável por iniciar o projeto formal, realizar a descrição formal da modificação a ser realizada, solicitar análise de viabilidade e concluir projeto formal.
Engenheiro	Responsável por analisar a viabilidade do projeto, informar caso o projeto não seja viável, iniciar o planejamento do projeto e concluir o planejamento com a entrega do protótipo.
Programador	Equipe responsável pela implementação das funcionalidades.
Teste	Equipe responsável pela execução dos testes planejados para cada versão do sistema e registro dos defeitos em não conformidades identificadas.
Cliente	Responsável por fazer a requisição inicial da mudança a ser feita e posteriormente por validar as mudanças feitas pela equipe.

2.2. Ferramentas, Ambientes e Infraestrutura

2.2.1. Ferramentas

Termo	Versão	Descrição
PostgreSQL	13.3	Ferramenta para gerenciar o banco de dados, download disponibilizado no endereço: https://www.postgresql.org/
GitHub	-	Sistema web para centralizar o sistema de gerenciamento de versões, agindo como repositório remoto. Disponível em: https://github.com/
Eclipse	2021-03 (4.19.0)	IDE de desenvolvimento em Java com suporte a Add-ons para testes, criação de interface por meio de interfaces visuais, disponível em: https://www.eclipse.org/downloads
JUnit	5.7.2	Ferramenta para realização de testes unitários em softwares escritos em Java, disponível em: https://junit.org/junit5/
Git	2.31.1	Sistema para o gerenciamento local de versões, disponível em: https://git-scm.com/
pgAdmin 4	5.0	Plataforma aberta de desenvolvimento e administração de bancos de dados que utilizam de PostgreSQL, disponível em https://www.pgadmin.org/
GanttProject	2.7.2	

2.2.2. Ambientes

Os ambientes de desenvolvimento serão as máquinas pessoais da equipe, portanto serão mantidos por cada membro, entretanto participarão de ferramentas de ambientes virtuais como Google Meet para reuniões.
As IDEs utilizadas pela equipe serão o Eclipse e Visual Studio Code, para o desenvolvimento dos bancos de dados será utilizado o pgAdmin para a maior facilidade com a integração ao PostgreSQL.
{[(O ambiente que será entregue a equipe de desenvolvimento, deverá ser mantido pela equipe de arquitetura, através de *Virtual Machines* que seguiram os padrões dos ambientes mantidos pela equipe de infraestrutura. As ferramentas de desenvolvimento "*IDEs*" serão de livre escolha do desenvolvedor, desde que a mesma seja uma ferramenta de Software Livre, tais como *Atom*, *Eclipse*, *NetBeans* …)]}

2.2.3. Infraestrutura

2.2.3.1. Desenvolvimento

O desenvolvimento será feito na máquina pessoal dos membros da equipe de desenvolvimento, os códigos liberados pela equipe serão enviados ao GitHub por meio de ferramentas como Git, SourceTree ou diretamente no site.

2.2.3.2. Homologação

O ambiente de homologação será a própria máquina pessoa dos membros da equipe de desenvolvimento, com a área gestora de códigos fontes sendo o repositório no GitHub

2.2.3.3. Treinamento

Nenhum ambiente de treinado especificado, entretanto poderá ser utilizado de repositórios Git e das máquinas pessoais dos membros em treinamento,

2.2.3.4. Produção

O ambiente de release será o branch “Main” do repositório no GitHub: https://github.com/AbyssalSire/gerencia-de-configuracoes .

3. O PROGRAMA DE GERENCIAMENTO DE CONFIGURAÇÃO

3.1. Identificação da Configuração

3.1.1. Métodos de Identificação

O detalhamento para a convenção para rotular os artefatos na estrutura de pastas do produto, será detalhada no documento PAP do projeto, que estará disponível no diretório de Gerencia de Configuração. Abaixo segue uma tabela com os acrônimos e significados.

Acrônimos	Significado
ARQ	Documento de Arquitetura
IMP	Documento de Implantação
PGC	Plano de Gerenciamento de Configuração
PAP	Documento de Permissões de Pastas e Acessos por Perfil
CBL	Documento de Controle de *BaseLines*
NEG	Documento de Negocio
PPR	Plano do Projeto
PPF	Planilha de Contagem de Ponto de Função
PNE	Documento de Processo de Negócio
CRT	Checklist de Revisão Técnica
RRT	Relatório de Revisão Técnica
PLT	Plano de Teste
PRT	Plano de Resultado de Teste
RTE	Roteiros de Teste
EUC	Especificação de Caso de Uso

3.1.2. Baselines do Projeto

As baselines serão definidas a cada mudança de fase do projeto, e uma de encerramento.

Fase	Itens de Configuração
Fase 1	Documento de Arquitetura

	Documento de Implantação
	Plano de Gerenciamento de Configuração
Fase 2	Documento de Permissões de Pastas e Acessos por Perfil
	Documento de Controle de *BaseLines*
	Documento de Negócio
Fase 3	Plano do Projeto
	Planilha de Contagem de Ponto de Função
	Documento de Processo de Negócio
Fase 4	Checklist de Revisão Técnica
	Relatório de Revisão Técnica
	Plano de Teste
Encerramento	Todos os Itens de configuração gerados nas fases anteriores
	Termo de encerramento

3.1.3. Estrutura do Repositório

O repositório está estruturado conforme o padrão a seguir:

- Baseline - pasta chamada “Software-manga-desktop-base”:
  - bin - gerada automaticamente;
  - lib - contendo os arquivos .jar utilizados no projeto;
  - src - contendo os códigos:
    - banco de dados - contém arquivo com as configurações necessárias para o funcionamento do banco
    - control - arquivos de controle, sendo eles o de conexão ao banco de dados (JDBCUtil.java), arquivo de iniciação do programa (Main.java), acesso à tabela Mangás do banco de dados (MangaDAO.java) e acesso à tabela Usuário do banco de dados (UsuarioDAO.java);
    - model - arquivos responsáveis pelo tratamento dos dados manipulados do banco de dados e possibilitando a utilização por eles nos outros arquivos;
    - view - arquivos de interface gráfica.
  - Descricao_E_Mudancas.txt - arquivo contendo uma descrição básica do software e as mudanças feitas nele ao longo do tempo;
  - criador.sql - código SQL para a criação do banco de dados.
- Artefatos gerados - pasta “artefatos”

3.2. Controle de Configuração e Mudança

3.2.1. Processo de Solicitações de Mudança

A solicitação de mudanças deve ser iniciada pelo cliente por meio dos canais de comunicação disponíveis com a equipe, após o contato inicial o Gerente de Projeto envia para o cliente uma requisição com dados mais categorizados da mudança para poder gerar os requisitos.

O gerente então planeja um processo de desenvolvimento, para que o Engenheiro possa projetar a mudança a ser implementada, seguida da entrega do protótipo ao Cliente para a autorização do desenvolvimento das funcionalidades apresentadas. Após o aval do cliente a mudança então entra para o processo de produção.

3.3. Estimativa do Status de Configuração

3.3.1. Processo de Armazenamento e Liberação do Projeto

O processo de garantia de funcionalidade é feito de forma online e nas seguintes etapas de backup e processo: base de dados, versionamento e cloud.

A base de dados possui um versionamento através de um sistema de migrações, que é o uso de um serviço de migração de banco de dados para migrar dados de um banco de origem para um ou mais bancos de destino, através desse processo é possível identificar falhas em um novo versionamento e se restaurar a versão anterior, a fim de recuperar o funcionamento adequado da aplicação.

O versionamento se dá através da ferramenta git, que gera versões de código em branchs, caso ocorra alguma versão ou erro irreversível é possível retomar a versão anterior a fim de retomar o funcionamento do aplicativo.

Em cloud, é possível utilizar as ferramentas disponibilizadas pelo provedor de serviço host, para em situações de bug, parar a aplicação ou voltar uma versão de deploy da mesma.