

 MINERVA SOLUTIONS		Plano de Gerenciamento de Configuração
Data: 06/11/2018	Objetivo Estratégico: Gerenciamento dos dados administrativos do Instituto Ada Lovelace.	
ID: 1008	Nome do Projeto: sigAda - Sistema de Gerenciamento do Ada	
CC: Paulo Pires	Cliente: Instituto Ada Lovelace	
Patrocinador: UFRJ	Gerente de Projeto: Rafael Cardoso	

1. Histórico de Versão

Data	Versão	Descrição	Autor(es)
06/11/2018	1.0.0	Abertura do plano de controle de versões	Eduardo, Rafael e Tainá
15/11/2018	1.1.0	Adição de novos exemplos no tópico 4.1.2 na arte da nomenclatura dos artefatos	Rafael, Eduardo, Tainá e Ricardo

2. Introdução

2.1. Finalidade

Este documento tem como finalidade apresentar as ferramentas e metodologias utilizadas para o gerenciamento de configuração e controle de versionamento.

2.2. Escopo

Este documento especificará padrões de organização, nomenclatura, funcionamento e modo de utilização de todas as ferramentas utilizadas para controle de versionamento da documentação e dos códigos implementados.

2.3. Definições, Acrônimos e Abreviações

Termos	Significados
IDE	Ambiente de Desenvolvimento Integrado (Integrated Development Environment)
GC	Gerente de configuração
CS	Caso de uso

2.4. Visão Geral

Os próximos tópicos deste documento descrevem como o projeto será estruturado em termos de nomenclaturas de artefatos, organização de código fonte e documentação. Além disso, é definido como se dá a gerência da configuração do projeto, qual(is) pessoa(s) estão no cargo e quais são suas responsabilidades. Ao final, é apresentado alguns recursos utilizados para o estudo das ferramentas selecionadas para o versionamento dos artefatos, como tutoriais em vídeo e documentação.

3. Gerenciamento de Configuração de Software

3.1. Organização, Responsabilidades e Interfaces

O gerente de configuração tem um papel importante no projeto, uma vez que assegura que os artefatos gerados estejam corretos e coerentes de acordo com as especificações apresentadas neste documento. Também é ele que é responsável por controlar as mudanças realizadas no projeto.

Gerente de configuração (GC): Rafael Cardoso.

3.2. Ferramentas, Ambiente e Infraestrutura

A gerência da configuração se dará através do sistema operacional Windows, utilizando as ferramentas abaixo, precisando, portanto, de uma conexão com a internet de qualidade.

Tipo	Ferramenta	Versão	Função
Controle de Versão	GitHub	-	Repositório para os códigos e documentação e o mesmo será responsável pelo versionamento
IDE	Microsoft Visual Studio	2017	Utilizado para a implementação do software, tendo integração com o GitHub, para que se possa atualizar o código armazenado nos repositórios
Editor de Texto	GitHub Wiki	-	Utilizado como repositório dos artefatos gerados no projeto
Editor de Texto	Google Docs	-	Utilizado para a criação dos artefatos gerados no projeto

4. O Programa de Gerenciamento de Configuração

4.1 Identificação da Configuração

4.1.1 Métodos de identificação

Os artefatos gerados pelo projeto, referente à documentação, devem seguir o seguinte padrão de nomenclatura para o título do arquivo:

<TIPO_ART>_InstitutoAdaLovelace.<TIPO_EXT>

Identificador	Significado
<TIPO_ART>	Tipo de artefato, onde as iniciais de cada palavra estão em maiúsculo e não há espaço entre as palavras. Ex.: PlanoDeProjeto, PlanoDeIteração
<TIPO_EXT>	Tipo de extensão de arquivo. Ex.: pdf, docx, odt

Para as imagens, como os diagramas e esquemas, será utilizado o seguinte padrão:

<SIGLA_TIPO_IMG>_<DESCRIÇÃO_IMG>_InstitutoAdaLovelace.<TIPO_EXT>

Identificador	Significado
<SIGLA_TIPO_IMG>	Sigla que diz ao que a imagem se refere. Para criar a sigla, usa-se até três caracteres para diferenciar as siglas entre elas, caso seja necessário. Ex.: DA - Diagrama de Atividade DS - Diagrama de Sequência DC - Diagrama de Classe MER - Modelo Entidade-Relacionamento
<DESCRIÇÃO_IMG>	Descrição de poucas palavras do conteúdo da imagem. Tal descrição não contém espaço e as iniciais das palavras estão em maiúsculo. Ex.: CS01FazerLogin CS18
<TIPO_EXT>	Tipo da extensão do arquivo. Ex.: jpg, png, gif

4.1.2 Versionamento dos artefatos

Nomenclatura

O número de versão dos artefatos é dada pela lógica abaixo:

<primeiro>.<segundo>.<terceiro>

Onde, o acréscimo de 1 é feito caso:

- <primeiro> : Uma alteração que modifica completamente o artefato é aprovada pelo cliente.
- <segundo> : Uma alteração é feita no artefato, porém suas características principais são mantidas.
- <terceiro> : É feita uma correção de detalhe no artefato, como uma correção de *bug*, ortográfica, formatação, entre outras.

Comentários de release

A plataforma GitHub disponibiliza que comentários possam ser adicionados a cada alteração das páginas criadas na Wiki e nos códigos armazenados nos repositórios. Utilizaremos deste benefício para fazer um melhor controle do que está ocorrendo naquele artefato em específico, seguindo a estrutura a seguir:

<SIMBOLO> <COMENTARIO>

Identificador	Significado
<SIMBOLO>	Representa a ação que foi realizada naquele artefato. Os símbolos pode ser: <ul style="list-style-type: none">• “+” : Adição de algum elemento.• “-” : Eliminação de algum elemento.• “#” : Alteração/Correção de algum elemento.
<COMENTARIO>	Descrição da localização que sofreu a ação dita pelo símbolo, além disso, pode ser escrito o motivo daquela ação. O texto deve ser claro e sucinto. Caso se tenha feito a ação em mais de um lugar, eles devem ser listados, utilizando como separador o “,”

4.1.3 Estrutura do repositório de versões

O Repositório principal é destinado ao armazenamento da documentação, dos códigos implementado e para a apresentação do projeto. Além do repositório principal é mantido o Wiki que tem como objetivo facilitar a visualização da documentação no GitHub.

Diretório	Conteúdo
.	README do repositório, contendo algumas informações básicas.
./Documentos	Esse diretório contém todos documentos gerado durante o desenvolvimento. Tais documentos estão em md5 e PDF.
./Codigo	Esse diretório contém todos os códigos gerados.
./Apresentacao	Diretório voltado para as apresentações periódicas do projeto.

4.2 Processo de Armazenamento de Mídia

O repositório do projeto deve ser *clonado* pelos integrantes da equipe, bem como armazenado em alguma ferramenta de armazenamento na nuvem, como o Google Drive. Isso é feito para, caso ocorra algum problema com o repositório original, existem cópia dos artefatos guardadas.

5. Treinamento e Recursos

- O canal no YouTube, Felipe Classroom Tutorials, possui um tutorial sobre o Google Docs. Disponível em: <https://www.youtube.com/playlist?list=PLwXXOxvDboeagQBUINBfC_-t9vA9c3bul>
- O GitHub também disponibiliza um tutorial interativo para aprender a como usá-lo. O tutorial pode ser encontrado em: <<https://try.github.io>>
- Diversos conteúdos a respeito do controle de versão e gerência de configuração de forma completa, mas didática pode ser encontrado em: <<http://git-scm.com/doc>>