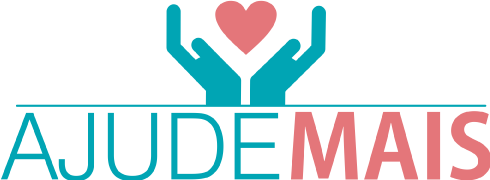
****

**Ajude Mais**

**Plano de Gerência de Configuração**

Monteiro, 22 de março de 2017

**Revisões**

|  |  |  |
| --- | --- | --- |
| ***Data*** | ***Descrição*** | ***Autor*** |
| 10/05/2017 | Adição de baselines e siglas de documentos | Rafael |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Tabela** 1: Controle de mudanças no documento, Ajude Mais, 2017

**CONTEÚDO**

[**1.** **Introdução** 3](#_Toc478159973)

[**1.1.** **Objetivos** 3](#_Toc478159974)

[**1.2.** **Organização do Documento** 4](#_Toc478159975)

[**1.3.** **Papéis e Responsabilidades** 4](#_Toc478159976)

[**2.** **Plano de Configuração** 5](#_Toc478159977)

[**2.1.** **Controle de Configuração** 5](#_Toc478159978)

[**2.1.1.** **Estrutura do Repositório de Gerência de Configuração** 5](#_Toc478159979)

[**3.** **Métodos de Identificação** 7](#_Toc478159980)

[**3.1.** **Documentos** 7](#_Toc478159981)

[**3.2.** **Versionamento** 7](#_Toc478159982)

[**3.3.** **Releases** 8](#_Toc478159983)

[**4.** **Ambiente, Ferramentas e Infraestrutura** 9](#_Toc478159984)

[**4.1.** **Plano de Software, ferramenta e Infraestrutura** 9](#_Toc478159985)

[**5.** **Estrutura de Branches do Projeto** 11](#_Toc478159986)

1. **Introdução**

Este documento descreve o Plano de Gerência de Configuração para o projeto de desenvolvimento do sistema Ajude Mais.

* 1. **Objetivos**

O presente documento tem por objetivo apresentar a organização, nomenclatura e regras de versionamento para a gerência de configuração do projeto de desenvolvimento do sistema Ajude Mais.

Este plano é destinado a todos os integrantes da equipe responsável pelo o desenvolvimento do sistema.

* 1. **Organização do Documento**

Nesta seção deve ser descrita a estrutura do documento, conforme exemplo abaixo:

As seções subsequentes deste documento estão assim organizadas:

* Seção 2, são descrito os papéis e responsabilidades da gerência de configuração;
* Seção 3 é apresentado o plano de configuração onde é definido a estrutura do armazenamento, as configurações bases do projeto, o controle de configuração e as políticas de segurança e acesso aos itens de configuração;
* Seção 4 é descrito a identificação de artefatos e versões do projeto.
* Seção 5 é descrito o plano de software com todas as tecnologias e ferramentas utilizadas.
  1. **Papéis e Responsabilidades**

|  |  |
| --- | --- |
| Papel | Responsabilidade |
| Gerente de Desenvolvimento (GD) | Juntamente com a CCM receber, analisar e aprovar os PFM. |
| Líder de Projeto (LP) | Planejar as atividades de GC juntamente com o Responsável pela Configuração, designar executante, finalizar SM, autorizar a criação das configurações bases conforme descrito na seção Plano de Configuração. |
| Responsável pela Configuração (RC) | Criar e manter infraestrutura corporativa (servidores) de GC; Implementar as políticas de Controle de Acesso ao ambiente de GC, Realizar os backups dos repositório de configuração dos projetos |

**Tabela** 2: Definição dos papéis e responsabilidades no projeto, Ajude Mais, 2017.

1. **Plano de Configuração**
   1. **Controle de Configuração**

Para um bom gerenciamento de versão durante o desenvolvimento do projeto, será usado a ferramenta [Git](http://www.git-scm.com/).

A escolha desta ferramenta deu-se pela sua grande popularidade e utilização no ambiente global de desenvolvimento de software. Empresas como Google, Microsoft, Netflix e Facebook adotam tal ferramenta como a principal neste proposito.

* + 1. **Estrutura do Repositório de Gerência de Configuração**

A seguir será apresentado a estrutura definida para armazenamento dos artefatos do projeto no repositório.

*Repositório*

***+-Sistema***

*| +-000-Requisitos*

*| +-001-Análise e Projeto*

*| +-002-Implementação*

*| +-003-Teste*

*| +-Implementacao*

*| +--Testes-ITXX*

*| +-004-Gerência de Configuração*

***+-Projeto-AjudeMais***

*| +-000-Gerência de Projeto*

*| +-001-Atas de Reunião*

*| +-002-Acompanhamento*

*| +-Iteracao -XX*

*| +-003-Teste*

*| +-Iteracao-XX*

O repositório é dividido em Sistema, para artefatos que pertencem ao sistema independente de quantos projetos irão desenvolvê-lo e Projeto-AjudeMais. Isso é necessário pois vários projetos podem ser executados para o desenvolvimento do mesmo sistema. Os subdiretórios estão organizados pelas disciplinas do processo. A tabela abaixo identifica o que deve ser armazenado em cada subdiretório do projeto.

|  |  |  |
| --- | --- | --- |
| Diretório | Subdiretório | Descrição |
| Sistema | 000-Requisitos | Armazenamento de Documentos de requisitos, documentos de casos de uso, Product backlog, User stories, etc. |
| 001-Análise e Projeto | Documentos referentes à análise e projeto do sistema como documentos de arquitetura, modelos UML, modelo de dados, dicionário de dados, etc. |
| 002-Implementação | Nesse diretório deve estar todo o código relativo ao sistema. |
| 003-Teste | Documento relativo aos cenários de teste para as User Stories. |
| 004-Gerência de Configuração | Armazenamento do plano do plano de gerência de configuração. |
| Projeto | 000-Gerência de Projeto | Armazenamento do plano de projeto. |
| 001-Atas de Reunião | Armazenamento das Atas de reunião do projeto. |
| 002-Acompanhamento | Deve possuir um subdiretório para cada iteração do projeto contendo o plano da iteração, reuniões de acompanhamento da iteração, Reunião de finalização da iteração, Burndown da iteração e timeline da iteração. |
| 003-Teste | Armazenamento dos relatórios de execução de testes. Deve possuir um subdiretório para cada iteração uma vez que os testes são executados por iteração. |

**Tabela** 3**:** Descrição das pastas do projeto, Ajude Mais, 2017

1. **Métodos de Identificação**
   1. **Documentos**

Todos os documentos disponibilizados no repositório devem ser identificados baseados na seguinte nomenclatura:

<ID ARTEFATO>-<NOME ARTEFATO>

Onde:

* <ID ARTEFATO> é a sigla de identificação do artefato conforme Tabela 1.
* <NOME ARTEFATO> é nome de identificação do artefato conforme *Tabela*

|  |  |
| --- | --- |
| ID artefato | Nome artefato |
| PPJ | Plano de Projeto |
| ARQ | Documento de Arquitetura |
| REQ | Documento de Requisitos |
| UCS | Documento de Casos de Uso |
| PBL | Product Backlog |
| TEA | Testes de Aceitação |
| PGC | Plano de Gerência de Configuração |
| PIT | Plano de Iteração |
| DRI | Documento de Riscos |
| ATA | Ata de Reunião |
| RAP | Relatório de Acompanhamento do Projeto |
| BRD | Burndown |
| TEM | Testes Manuais |
| FDI | Feedback da iteração |
| RVI | Roteiro de validação da iteração |
| TLI | Timeline da iteração |
| ADR | Apresentação de resultados |
| DRI | Documento de Retrospectiva |

**Tabela** 4: Identificadores e Nomes dos Artefatos, Ajude Mais, 2017

* 1. **Versionamento**

A abordagem de versionamento do software deverá seguir o seguinte modelo:

*Versão: <Major>.<Minor >.<Patch>*

* <Major> representa a versão principal do software. Ela será incrementada a cada nova release. Neste caso, <Minor> e <Patch> devem ser decrementadas para o valor zero (0).
* <Minor> representa a versão secundária. Ela será incrementada durante a criação e disponibilização de novos recursos no software. Neste caso, <Patch> deve ser decrementada para zero (0).
* <Patch> representa a versão para possíveis correções na versão atual do software. Ela será incrementada, após a correções de possíveis problemas, na mais recente versão criada do software.
  1. **Baseline e Releases**

Sempre que tivermos uma versão estável do software testada e com documentação aprova, mas sem perpectiva de uma realese deve ser gerada uma baseline com o objetivo de recuperarmos os estado em que ficou o código e a documentação. Assim, as baselines devem seguir a seguinte nomenclatura:

*B<NUMERO\_B>-IT<NUMERO\_ITERACAO>*

Sempre que tivermos uma versão estável do software, a qual existe a possibilidade de ser utilizada no ambiente real, deverá ser criada uma release candidate. Nela, todas funcionalidades para versão deverão estar especificadas, implementadas e testadas. Assim, para sua criação deverá ser seguida a seguinte nomenclatura:

*<ID\_VERSAO>-RC<NUMERO\_RC>-IT<NUMERO\_ITERACAO>*

Onde:

* ID\_VERSAO: O identificador da versão que está sendo construída.
* NUMERO\_RC: Número da release candidate.
* NUMERO\_ITERACAO: O número da iteração em que foi gerada a release candidate.

Após aprovação de uma release candidate pelo cliente, deve ser gerada uma nova release, representando uma versão usável do software. Desta forma, as releases de iteração devem seguir a seguinte nomenclatura:

*<ID\_VERSAO>-IT<NUMERO\_ITERACAO>*

Onde:

* ID\_VERSAO: O identificador da versão que está sendo construída.
* NUMERO\_ITERACAO: O número da iteração em que foi gerada a release.

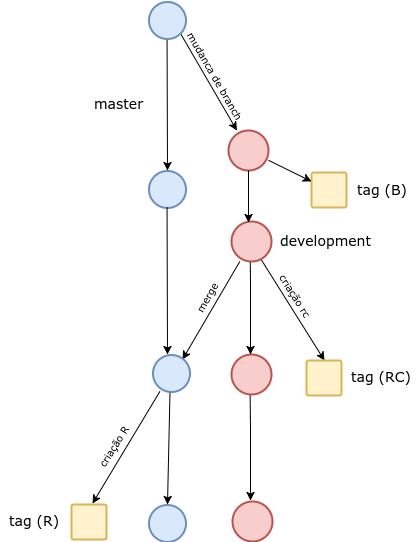
1. **Ambiente, Ferramentas e Infraestrutura**
   1. **Plano de Software, ferramenta e Infraestrutura**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome do Item | Propósito | Ambiente | Release/Versão |
| Git | Controle de Repositório | Desenvolvimento | 2.12.1 |
| MS-Office | Documentos do Word | Todos | 2013 ou 2016 |
| Astah UML | Diagramação UML | Todos | 7.x |
| Eclipse | IDE de desenvolvimento. | Desenvolvimento | Spring tool suite ou eclipse Mars ou eclipse Neon |
| Android Studio | IDE de desenvolvimento. | Desenvolvimento | 2.x |
| WebStorm | IDE desenvolvimento Web | Desenvolvimento | 2017.1 |
| Postgres | Banco de Dados Relacional | Desenvolvimento | 9.x |
| PgAdmin | Interface para gerenciamento de banco de dados Postgres. | Desenvolvimento | 3.x |
| HSQL | Banco dados relacional para testes. | Desenvolvimento | 2.3.x |
| Atom | Editor de texto | Desenvolvimento | 1.13.x |
| Java JDK | Kit básico para o desenvolvimento em Java. | Desenvolvimento | 1.8 |
| SDK Android | Kit básico para o desenvolvimento na plataforma Android. | Desenvolvimento | 2.x |
| Hibernate | Framework de persistência. | Desenvolvimento | 5.2.x |
| Spring Boot | Facilitador de configuração de projetos Spring. | Desenvolvimento | 1.5.x |
| Spring Data | Biblioteca responsável por gerenciar persistência | Desenvolvimento | 1.10.x |
| Spring MVC | Projeto Spring que faz parte do Spring Framework | Desenvolvimento | 1.5.x |
| Bean Validation | Especificação que permite validar objetos com facilidade em diferentes camadas da aplicação | Desenvolvimento |  |
| Spring Security | Framework de autenticação e controle de acesso. | Desenvolvimento | 4.x |
| Spring Tests | Projeto Spring que faz parte do Spring Framework | Desenvolvimento | 1.5.x |
| Junit | Framework para escrita de testes | Desenvolvimento | 5.x |
| DBunit | Extensão JUnit para testes de banco de dados. | Desenvolvimento | 2.5.x |
| JasperReports | Biblioteca responsável em gerar relatórios em PDF. | Desenvolvimento | 6.2.x |
| Facebook API | Biblioteca para comunicação com Facebook | Desenvolvimento | - |
| Google Maps API | Biblioteca para lidar com comunicação com Google Maps. | Desenvolvimento | - |
| AngularJS | MVW Framework Javascript | Desenvolvimento | 1.6.x |
| Node JS | Plataforma para trabalhar com Javascript no backend. |  |  |
| Bootstrap | Biblioteca de componentes para Web | Desenvolvimento | 3.x |

**Tabela** 5: Plano de Software, ferramentas e Infraestrutura, Ajude Mais, 2017.

1. **Estrutura de Branches do Projeto**

A estratégia de branches adotada será de acordo com seguinte modelo:



* A *branch master* deverá conter os artefatos da release mais recente. Após a conclusão de cada iteração é recomendado a realização de um merge da *branch development* para *branch master* dos artefatos produzidos, já com uma release candidate criada.
* A *branch development* será o local de produção dos artefatos do projeto, artefatos como, desenvolvimento de novas funcionalidades, testes, atualização de documentação e etc.
* Uma *tag*  deve ser criada sempre que existir uma versão do software usável. Considerando a criação de uma *tag* antes da aprovação pelo cliente (RC), e outra depois de sua aprovação (R). Já se não ocorrer nenhuma destas ocasiões é recomendado gerar um Tag (B) para registrar o código implementado durante a iteração.