<Nome do Projeto>

Documento de Arquitetura de Software

Versão <1.0>

[Observação: O template a seguir é fornecido para uso com o Rational Unified Process (RUP). O texto em azul exibido entre colchetes e em itálico (style=InfoBlue) foi incluído para orientar o autor e deve ser excluído antes da publicação do documento. Um parágrafo digitado após esse estilo será automaticamente definido como normal (style=Body Text).]

[Para personalizar campos automáticos no Microsoft Word (que exibem um fundo cinza quando selecionados), escolha File>Properties e substitua os campos Title, Subject e Company pelas informações apropriadas para este documento. Depois de fechar a caixa de diálogo, para atualizar os campos automáticos no documento inteiro, selecione Edit>Select All (ou Ctrl-A) e pressione F9 ou simplesmente clique no campo e pressione F9. Isso deve ser feito separadamente para Cabeçalhos e Rodapés. Alt-F9 alterna entre a exibição de nomes de campos e do conteúdo dos campos. Consulte a ajuda do Word para obter mais informações sobre como trabalhar com campos.]

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 13/06/2021 | 1.0 | Criação do documento | Pedro Henrique F. Melo |
| 17/06/2021 | 2.0 | Atualização do documento | Pedro Henrique F. Melo |
|  |  |  |  |
|  |  |  |  |

Índice Analítico

1. Introdução 4

1.1 Finalidade 4

1.2 Escopo 4

1.3 Definições, Acrônimos e Abreviações 4

1.4 Referências 4

1.5 Visão Geral 4

2. Representação Arquitetural 4

3. Metas e Restrições da Arquitetura 4

4. Visão de Casos de Uso 5

4.1 Realizações de Casos de Uso 5

5. Visão Lógica 5

5.1 Visão Geral 5

5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura 5

6. Visão de Processos 5

7. Visão de Implantação 5

8. Visão da Implementação 5

8.1 Visão Geral 5

8.2 Camadas 6

9. Visão de Dados (opcional) 6

10. Tamanho e Desempenho 6

11. Qualidade 6

Documento de Arquitetura de Software

# Introdução

O presente documento tem como objetivo descrever o documento de arquitetura do projeto de sistema de gerenciamento de estoque de Centro de Distribuição. Este sistema tem como finalidade gerenciar de forma simplificada um armazém em um centro de distribuição de diferentes itens.

## Finalidade

Este documento demonstra os processos usados para o desenvolvimento do software incluindo suas fontes que justificam as decisões tomadas que resultaram na arquitetura final do sistema do projeto.

## Escopo

O intuito desse papel é oferecer aos usuários finais e desenvolvedores que modificarão este projeto em eventuais atualizações, a informação necessária para implementar tais alterações e informar como esse funciona. Além de auxiliar no entendimento do sistema por novos membros da equipe.

## Definições, Acrônimos e Abreviações

Este projeto possui um documento dedicado a Definições, Acrônimos, Abreviações e Glossário.

## Referências

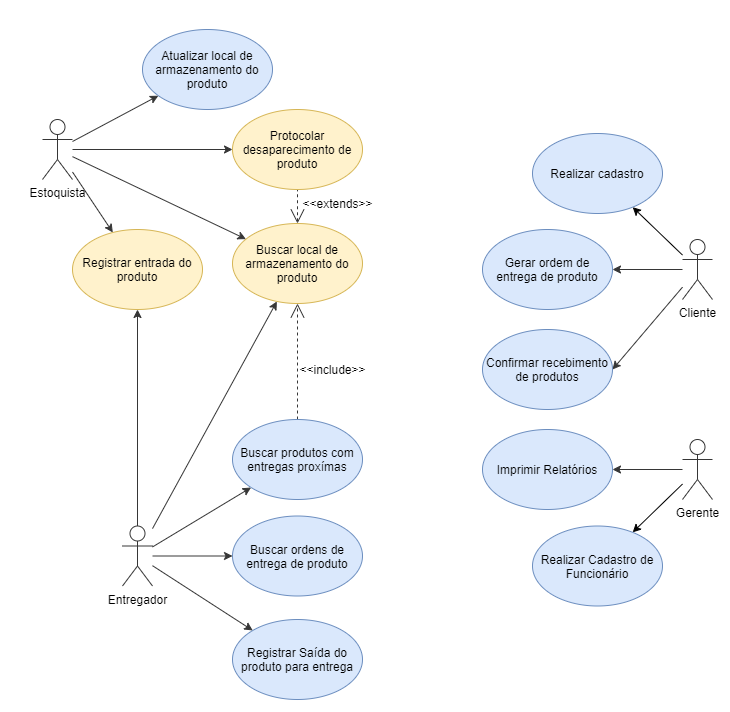
# Representação Arquitetural

Este documento apresenta a arquitetura como uma série de visualizações; visualização caso de uso, visualização lógica, visualização do processo e visualização da implementação.

# Metas e Restrições da Arquitetura

* Linguagem de programação Java (versão JDK 16.0.1)
* Computadores devem ser conectados em rede local com intuito de manter plena funcionalidade do sistema
* Login e Senhas devem ser mantidos em segurança
* Utilização do paradigma Orientado a Objetos para o desenvolvimento

# Visão de Casos de Uso



Nome do Diagrama: Casos de Uso Significativos para a Arquitetura

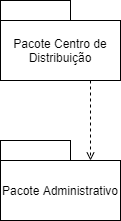
# Visão Lógica

**A visualização lógica do sistema de registro em curso é composta de 2 pacotes principais: Administrativo; Centro de Distribuição.**

**O pacote Administrativo é responsável pela criação de protocolos notificando as mudanças feitas, por usuários ao sistema;**

**O pacote Centro de Distribuição, é responsável pela modificação de dados no sistema além de gerenciar relatórios semanais.**

## Visão Geral



## Pacotes de Design Significativos do Ponto de Vista da Arquitetura

[Para cada pacote significativo, inclua uma subseção com o respectivo nome, uma breve descrição e um diagrama com todos os pacotes e classes significativos nele contidos.

Para cada classe significativa no pacote, inclua o respectivo nome, uma breve descrição e, opcionalmente, uma descrição de algumas das suas principais responsabilidades, operações e atributos.]

# Visão de Processos

[Esta seção descreve a decomposição do sistema em processos leves (threads simples de controle) e processos pesados (agrupamentos de processos leves). Organize a seção em grupos de processos que se comunicam ou interagem. Descreva os modos principais de comunicação entre processos, como transmissão de mensagens e interrupções.]

# Visão de Implantação

[Esta seção descreve uma ou mais configurações da rede física (hardware) na qual o software é implantado e executado. Ela é uma visão do Modelo de Implantação. No mínimo, para cada configuração, ela deve indicar os nós físicos (computadores, CPUs) que executam o software e suas interconexões (barramento, LAN, ponto a ponto, etc.) É incluído também um mapeamento dos processos da **Visão de Processos** nos nós físicos.]

# Visão da Implementação

[Esta seção descreve a estrutura geral do modelo de implementação, a divisão do software em camadas e os subsistemas no modelo de implementação e todos os componentes significativos do ponto de vista da arquitetura.]

## Visão Geral

[Esta subseção nomeia e define as diversas camadas e o seu conteúdo, as regras que determinam a inclusão em uma camada específica e as fronteiras entre as camadas. Inclua um diagrama de componentes que mostre os relacionamentos entre as camadas. ]

## Camadas

[Para cada camada, inclua uma subseção com o respectivo nome, uma lista dos subsistemas localizados na camada e um diagrama de componentes.]

# Tamanho e Desempenho

O software, como está, é capaz de gerenciar a modificação de dados de 1 usuário com a criação de tabelas geracionais temporários.

# Qualidade

O padrão de arquitetura adotado no projeto tem como finalidade garantir uma melhor fundação de código-fonte, com o intuito de disponibilizar alterações/personalização/atualização de forma simples, bem como a portabilidade do mesmo, capaz de rodar em maquinas de diferentes portes que possuem capacidade de entender a linguagem/bibliotecas utilizadas no software.