Documento de Arquitetura de Software

# INFORMAÇÕES SOBRE O PROJETO

|  |  |  |
| --- | --- | --- |
| **Gahima** | | |
| **Gestor do Projeto** | **Gerente de Projeto** | **Desenvolvedor** |
| Igor Horta | João Mago | Gabriel Amaral |
| Igor.horta@gahima.com | [Joao.mago@gahima.com](mailto:Joao.mago@gahima.com) | Gabriel.amaral@gahima.com |
| 3458-8979 | 3485-8965 | 3458-9865 |

|  |
| --- |
| **Objetivo deste Documento** |
| Este documento tem como objetivo descrever as principais decisões de projeto tomadas pela equipe de desenvolvimento e os critérios considerados durante a tomada destas decisões. Suas informações incluem aparte de *hardware* e *software* do sistema. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Histórico de Revisão* | | | | |
| Data | Demanda | Autor | Descrição | Versão |
| [24/05/2020] | [AR001] | [Igor Horta] | Criação do Documento | 0.1 |
|  |  |  | Homologação do Documento. | 1.0 |
|  |  |  |  |  |

Sumário

[INFORMAÇÕES SOBRE O PROJETO 1](#_Toc41242973)

[1. INTRODUÇÃO 3](#_Toc41242974)

[1.1 Finalidade 3](#_Toc41242975)

[1.2 Escopo 3](#_Toc41242976)

[1.3 Definições, Acrônimos e Abreviações 3](#_Toc41242977)

[1.4 Referências 4](#_Toc41242978)

[2. REPRESENTAÇÃO ARQUITETURAL 4](#_Toc41242979)

[3. REQUISITOS E RESTRIÇÕES ARQUITETURAIS 5](#_Toc41242980)

[4. VISÃO DE CASOS DE USO 5](#_Toc41242981)

[4.1 Casos de Uso significantes para a arquitetura 6](#_Toc41242982)

[5. VISÃO LÓGICA 7](#_Toc41242983)

[5.1 Visão Geral – pacotes e camadas 7](#_Toc41242984)

[6. VISÃO DE IMPLEMENTAÇÃO 9](#_Toc41242985)

[6.1 Caso de Uso FB001 Login 9](#_Toc41242986)

[6.1.1 Diagrama de Classe Login 9](#_Toc41242987)

[6.2 Caso de Uso FB002 Catálogo de Roupas 9](#_Toc41242988)

[6.2.1 Diagrama de Classe Produtos 9](#_Toc41242989)

[6.3 Caso de Uso FB003 Efetuar Pagamento 9](#_Toc41242990)

[6.3.1 Diagrama de Classe Pagamento 9](#_Toc41242991)

[6.4 Caso de Uso FB004 Emitir Relatórios 10](#_Toc41242992)

[6.4.1 Diagrama de Classe Relatórios 10](#_Toc41242993)

[7. VISÃO DE IMPLANTAÇÃO 10](#_Toc41242994)

[8. DIMENSIONAMENTO E PERFORMANCE 10](#_Toc41242995)

[8.1 Volume 10](#_Toc41242996)

[8.2 Performance 11](#_Toc41242997)

[9. QUALIDADE 11](#_Toc41242998)

# INTRODUÇÃO

## Finalidade

Este documento fornece uma visão arquitetural abrangente do sistema Gahima usando diversas visões de arquitetura para **representar** diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.

O documento irá adotar uma estrutura baseada na visão “4+1” de modelo de arquitetura [KRU41].



Figura 1 – Arquitetura 4+1

## Escopo

Este Documento de Arquitetura de Software se aplica ao sistema Gahima, que será desenvolvido pelo responsável Gabriel Amaral.

## Definições, Acrônimos e Abreviações

QoS – Quality of Service, ou qualidade de serviço. Termo utilizado para descrever um conjunto de qualidades que descrevem as requisitos não-funcionais de um sistema, como performance, disponibilidade e escalabilidade[QOS].

## Referências

[KRU41]: The “4+1” view model of software architecture, Philippe Kruchten, November 1995, <http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf>

# REPRESENTAÇÃO ARQUITETURAL

Este documento irá detalhar as visões baseado no modelo “4+1” [KRU41], utilizando como referência os modelos definidos na MDS. As visões utilizadas no documento serão:

|  |  |  |  |
| --- | --- | --- | --- |
| Visão | Público | Área | Modelo da MDS |
| Lógica | Analistas | Realização dos Casos de Uso |  |
| Processo | Integradores | Performance, Escalabilidade, Concorrência |  |
| Implementação | Programadores | Componentes de Software |  |
| Implantação | Gerência de Configuração | Nodos físicos |  |
| Caso de Uso | Todos | Requisitos funcionais |  |
| Dados | Especialistas em dados  Administradores de dados | Persistência de dados |  |

Tabela 1 – Visões, Público, Área e Artefatos da MDS

# REQUISITOS E RESTRIÇÕES ARQUITETURAIS

Esta seção descrever os requisitos de software e restrições que tem um impacto significante na arquitetura.

|  |  |
| --- | --- |
| Requisito | Solução |
| Linguagem | Linguagem Backend Java, Frontend HTML+PHP+Javascript, banco de dados SQL Server |
| Plataforma | Software Eclipse com conexão de servidor XAMP para banco de dados+PHP. |
| Segurança | Integração para meio de pagamento e autenticação via senha para operações sensíveis no sistema |
| Persistência | Os dados serão persistidos em banco de dados relacional SQL Servidor com backup semanal |

Tabela 2 – Exemplo de requisitos e restrições

# VISÃO DE CASOS DE USO

Esta seção lista as especificações centrais e significantes para a arquitetura do sistema.

Lista de casos de uso do sistema:

* **Caso de Uso FB001 Login**
* **Caso de Uso FB002 Catálogo de Roupas**
* **Caso de Uso FB003 Efetuar Pagamento**
* **Caso de Uso FB004 Efetuar Entrega**
* **Caso de Uso FB005 Relatório de Entregas**

## Casos de Uso significantes para a arquitetura

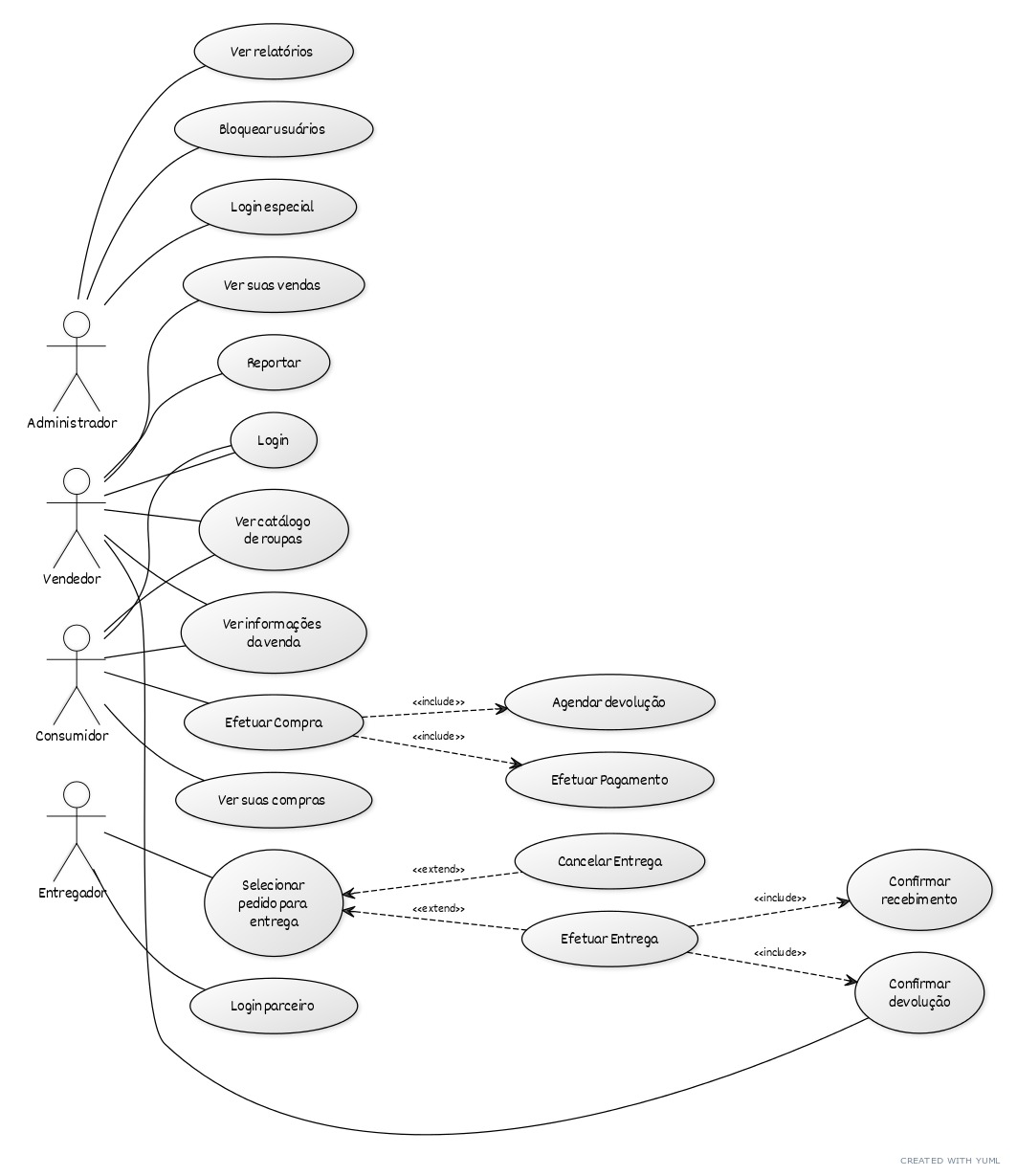


Figura 2 – Exemplo de Diagrama com os casos de uso significativos e atores

# VISÃO LÓGICA

Todas as etapas lógicas do sistema ocorrerão dentro do app Gahima. As classes importantes do sistema são:

* efetuarLogin;
* cadastrarProduto;
* listaProdutoCarrinho;
* listaProdutoEstoque;
* registrarPagamento;
* validarPagamento;
* emitirRelatorioAlugados;
* emitirRelatorioEntregas;
* emissaoRelatorioRepasse;

Os pacotes serão login, produto, pagamento e relatorio com as seguintes classes:

* login (efetuarLogin;)
* produto (cadastrarProduto; listaProdutoCarrinho; listaProdutoEstoque)
* pagamento (registrarPagamento; validarPagamento;)
* relatorio (emitirRelatorioAlugados; emitirRelatorioEntregas; emitirRelatorioRepasse

## Visão Geral – pacotes e camadas

Texto preto sobre fundo branco

Descrição gerada automaticamente

Figura 2 – Exemplo de Diagrama de Camadas da Aplicação

Uma imagem contendo screenshot, texto

Descrição gerada automaticamente

Figura 3 – Exemplo de Diagrama de Pacotes da Aplicação

# VISÃO DE IMPLEMENTAÇÃO

## Caso de Uso FB001 Login

* **Caso de Uso FB003 Efetuar Pagamento**
* **Caso de Uso FB005 Relatório de Entregas**

### Diagrama de Classe Login

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Figura 6.1.1 – Diagrama de Classe Login

## Caso de Uso FB002 Catálogo de Roupas

### Diagrama de Classe Produtos

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Figura 6.2.1 – Diagrama de Classe Produtos

## Caso de Uso FB003 Efetuar Pagamento

### Diagrama de Classe Pagamento

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Figura 6.3.1 – Diagrama de Pagamento

## Caso de Uso FB004 Emitir Relatórios

### Diagrama de Classe Relatórios

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente

Figura 6.3.1 – Diagrama de Pagamento

# VISÃO DE IMPLANTAÇÃO

Descrever os nodos físicos, as configurações e os artefatos que serão implantados.

Uma imagem contendo screenshot, texto

Descrição gerada automaticamente

Figura 7 – Exemplo de Diagrama de Implantação Java com banco de Dados SQL Server

# DIMENSIONAMENTO E PERFORMANCE

## Volume

Enumerar os itens relativos ao volume de acesso aos recursos da aplicação:

* Número de estimado usuários: 15
* Número estimado de acessos diários: 80
* Número estimado de acessos por período: 45m
* Tempo de sessão de um usuário: 1h

## Performance

Enumerar os itens referentes à resposta esperada do sistema:

* Tempo máximo para a execução de determinada transação: 6s

# QUALIDADE

Enumerar os itens de qualidade de software [QOS] significativos para a aplicação:

|  |  |  |
| --- | --- | --- |
| Item | Descrição | Solução |
| Escalabilidade | Escalável a estrutura de armazenamento em Cloud Computing | No futuro será escalável através de plataformas SaaS como Amazon e Ibm. |
| Confiabilidade, Disponibilidade | Banco de dados normalizado na forma 3N de relacionamento entre entidades, | Utilização da versão SQLServer mais recente |
| Segurança | Senha armazenada via criptografia e validação de pagamentos online. | Implementação através de APIs padrão para validação de segurança de login e métodos de pagamento online |