eGarçom ®

Cardápio Eletrônico

**Documento de Arquitetura de Software**

**Versão 1.0**

**Histórico da Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 29/03/12 | 1.0 | Composição Inicial do documento de Arquitetura | Jairo |

Sumário

[1. Introdução 4](#_Toc292908471)

[1.1 Finalidade 4](#_Toc292908472)

[1.2 Escopo 4](#_Toc292908473)

[1.3 Definições, Acrônimos e Abreviações 4](#_Toc292908474)

[1.4 Referências 4](#_Toc292908475)

[1.5 Visão Geral 5](#_Toc292908476)

[2. Representação Arquitetural 5](#_Toc292908477)

[3. Metas e Restrições da Arquitetura 5](#_Toc292908478)

[4. Visão de Casos de Uso 5](#_Toc292908479)

[4.1. Diagramas de Caso de Uso 6](#_Toc292908480)

[4.2. Descrição de Caso de Uso 6](#_Toc292908481)

[5. Visão Lógica 8](#_Toc292908482)

[5.1 Visão Geral 8](#_Toc292908483)

[5.2 Pacotes 9](#_Toc292908484)

[5.3 Descrição dos pacotes 11](#_Toc292908485)

[5.4 Diagrama de Componentes 11](#_Toc292908486)

[6. Visão de Processos 13](#_Toc292908487)

[7. Visão de Implantação 13](#_Toc292908488)

[8. Visão de Implementação 13](#_Toc292908489)

[9. Visão de Dados 13](#_Toc292908490)

[10. Tamanho e Desempenho 13](#_Toc292908491)

[11. Qualidade 14](#_Toc292908492)

[12. Exceções 14](#_Toc292908493)

# Introdução

## Finalidade

Este documento tem a finalidade de fornecer uma visão arquitetural do sistema, usando as visões de arquitetura mencionadas nesse documento para representação dos aspectos do sistema, com a intenção de ajudar capturar e transmitir as decisões significativas da arquitetura tomadas em relação ao sistema.

## Escopo

Este Documento de Arquitetura de Software se aplica ao Cardápio Eletrônico, que será desenvolvido pelo aluno Jairo Charnoski, do curso de Sistemas de Informação da Universidade Federal de Sergipe, sendo este um projeto da disciplina de Engenharia de Software II.

## Definições, Acrônimos e Abreviações

Todas as definições juntamente com os termos, acrônimos e abreviações necessárias à adequada interpretação e entendimento deste documento podem ser encontradas no documento de Glossário do Projeto (Glossário Atividade), pode ser acessado através do endereço: <https://projeto-academia.googlecode.com/svn/trunk/Modelagem_Negocio/Modelo%20de%20Casos%20de%20Uso/>.

## Referências

Os seguintes documentos foram utilizados como referência para a elaboração do documento arquitetura:

* Documento de Requisitos da Sistema da Academia - <https://projeto-academia.googlecode.com/svn/trunk/Requisitos/>
* Modelagem de Negócio do Sistema da Academia - <https://projeto-academia.googlecode.com/svn/trunk/Modelagem_Negocio/>
* Modelo de Análise - <https://projeto-academia.googlecode.com/svn/trunk/Analise_Projeto/>
* <http://www.wthreex.com/rup/portugues/webtmpl/templates/a_and_d/rup_sad.htm#1.3>

## Visão Geral

A seção 2 descreve o que a arquitetura do software representa para o sistema. A seção 3 descreve os requisitos de software, na seção 4 são listados os casos de uso que representam alguma funcionalidade significante para o sistema. As seções 5,6,7,8,9 apresentam respectivamente a visão lógica, de processo, de implantação, de implementação e de dados do sistema.

A seção 10 apresenta uma descrição das principais características de tamanho e performance que impactam a arquitetura, bem como as metas de performance desejadas. Finalmente, a seção 11 apresenta uma descrição de como a arquitetura de software contribui para a qualidade geral do sistema.

# 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. Essas visões são apresentadas como Modelos do Rational Rose e utilizam a Linguagem Unificada de Modelagem (UML).

O estilo arquitetural a ser utilizado no desenvolvimento do projeto será o de camadas, onde cada camada pode ser composta por vários componentes que interagem entre si ou entre os componentes de camadas adjacentes. Serão utilizados também outros estilos arquiteturais, tais como: Arquitetura baseada em Componentes e Client-Server.

# Metas e Restrições da Arquitetura

Existem algumas restrições de requisito não-funcionais e de sistema principais que têm uma relação significativa com a arquitetura.

São elas:

• Será uma aplicação Mobile Android;

• A linguagem de desenvolvimento utilizada para o sistema será o JAVA;

•O Sistema Gerenciador de Banco de Dados escolhido para suportar a aplicação será o SQLite.

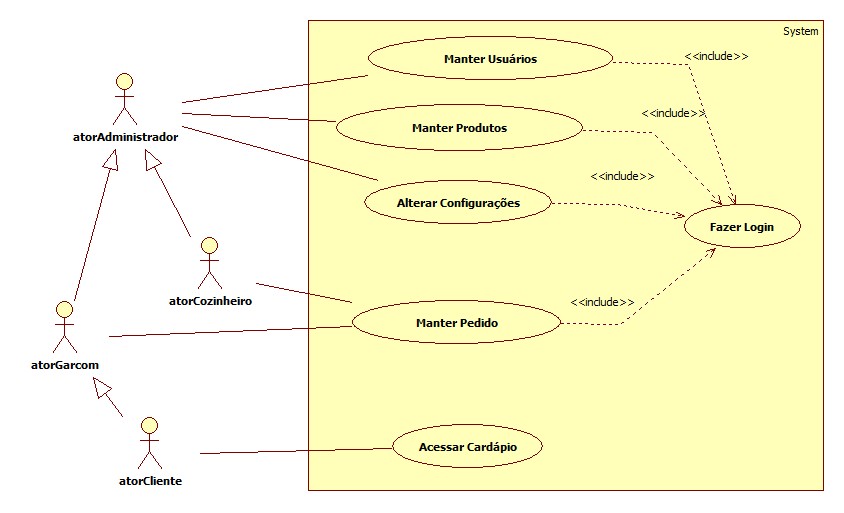
# Visão de Casos de Uso

Os casos de uso do sistema de controle da Academia estão listados a seguir:

* CSU01 - Fazer Login;
* CSU02 - Manter Usuários;
* CSU03 - Manter Produtos;
* CSU04 - Alterar Configurações;
* CSU05 - Manter Pedido;
* CSU06 - Acessar Cardápio;

## 4.1. **Diagramas de Caso de Uso**

O diagrama a seguir representa os casos de uso do sistema.



## 4.2. Descrição de Caso de Uso

1. CSU01 - Fazer Login

O sistema solicita que o usuário digite o nome e a senha, em seguida verifica se os campos solicitados são válidos.

1. CSU02 - Manter Usuários

O Administrador poderá cadastrar um novo Usuário na base de dados, atualizar dados ou excluir o mesmo. Para efetuar o cadastro é necessário nome, endereço, telefone, email e data de nascimento. Uma mensagem deve ser exibida caso o usuário já possua cadastro na base de dados do sistema.

1. CSU03 - Manter Produtos

O Administrador poderá cadastrar um novo Produto na base de dados, atualizar dados ou excluir o mesmo. Para efetuar o cadastro é necessário nome, descrição. Uma mensagem deve ser exibida caso o produto já possua cadastro na base de dados do sistema.

1. CSU04 - Alterar Configurações

O Administrador poderá alterar configurações do sistema.

1. CSU05 - Manter Pedido

O Garçom e o Cozinheiro poderão cadastrar um novo pedido na base de dados, alterar ou excluir os dados de um pedido. Para efetuar esse cadastro será necessário.

1. CSU06 - Acessar Cardápio

O Cliente poderá acessar o cardápio.

# Visão Lógica

Uma descrição da visualização lógica da arquitetura. Descreve as classes mais importantes, sua organização em pacotes de serviço e subsistemas e a organização desses subsistemas em camadas. Também descreve as realizações de casos de uso mais importantes como, por exemplo, os aspectos dinâmicos da arquitetura.

## Visão Geral

A visualização lógica do eGarçom é composta de 3 pacotes principais: apresentação, negócios e dados.



## Pacotes

Os pacotes propostos para o sistema estão listados a seguir:



## Descrição dos pacotes

* apresentacao - Esta camada de aplicativo tem todas as classes de limite que representam as telas do aplicativo visualizadas pelo usuário, é através dela que são feitas as requisições com a camada de negócios.
* negocio - A camada de Negócios tem todas as classes de controlador que representam os gerenciadores de caso de uso que direcionam o comportamento do aplicativo, ou seja, onde deve constar todas as regras ou lógica de negócio do sistema, é através dela que são feitas as requisições com a camada de dados.

dados - Neste pacote ficam as classes responsáveis pela manipulação da estrutura física de armazenamento dos dados. Recebe as requisições da camada de negócios e seus métodos executam essas requisições em um banco de dados.

## Diagrama de Componentes

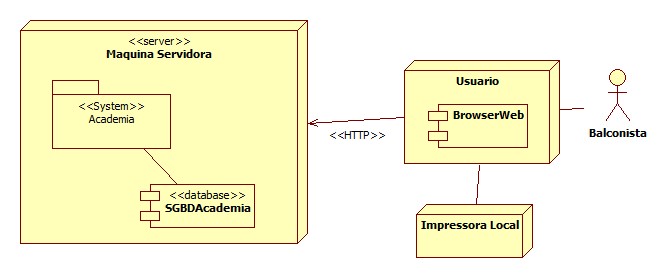


# Visão de Processos

# Visão de Implantação

Esta visão define o ambiente de implantação onde a aplicação será instalada.

Diagrama de Implantação da Arquitetura:



# Visão de Implementação

Ver Guia de Análise e Design - <https://projeto-eGarcon.googlecode.com/svn/trunk/Analise_Projeto/>

# Visão de Dados

# Tamanho e Desempenho

Estruturando o sistema em camadas e a programação através de classes, objetivamos obter uma boa performance do sistema e trataremos a programação para que não haja processamentos desnecessários. Segue a descrição das principais características de tamanho e desempenho do sistema:

* O sistema deve permitir o acesso simultâneo aos usuários restritos ao sistema e a operação deverá ser continuada e ininterrupta.
* O tempo de resposta para carregar uma activity não deve exceder 45 segundos sob condições normais.

# Qualidade

Segue a descrição de como a arquitetura de software contribui para a qualidade geral do sistema:

* Modularidade: o sistema deve ser desenvolvido em camadas, havendo uma interface de comunicação bem definida entre as mesmas.
* Manutenibilidade: O sistema deve adotar padrões de documentação e codificação bem definidos. O código fonte do sistema deve estar bem documentado, ser robusto e fácil de debugar e manter.
* Reusabilidade: a arquitetura do sistema deve ser tal que permita a utilização de classes e componentes de outros projetos, favorecendo o tempo de produção e a qualidade do produto gerado.

# Exceções

