Pedidos Já

Documento de Arquitetura de Software

Versão 1.0

Índice Analítico

Conteúdo

[Conteúdo 2](#_Toc490234402)

[1. Introdução 3](#_Toc490234403)

[1.1 Finalidade 3](#_Toc490234404)

[1.2 Escopo 3](#_Toc490234405)

[1.3 Definições, Acrônimos e Abreviações 3](#_Toc490234406)

[1.4 Referências 3](#_Toc490234407)

[1.5 Visão Geral 3](#_Toc490234408)

[2. Representação Arquitetural 4](#_Toc490234409)

[3. Metas e Restrições da Arquitetura 4](#_Toc490234410)

[4. Visão de Casos de Uso 4](#_Toc490234411)

[4.1 Realizações de Casos de Uso 5](#_Toc490234412)

[5. Visão Lógica 7](#_Toc490234413)

[5.1 Visão Geral 7](#_Toc490234414)

[5.2 Pacotes de Design Significativos do Ponto de Vista da Arquitetura 7](#_Toc490234415)

[5.3 Diagrama de componentes 8](#_Toc490234416)

[6. Visão de Processos 9](#_Toc490234417)

[7. Visão de Implantação 9](#_Toc490234418)

[8. Visão da Implementação 10](#_Toc490234419)

[9. Qualidade 10](#_Toc490234420)

[10. Exceções 11](#_Toc490234421)

Documento de Arquitetura de Software

# Introdução

Esse documento provê uma visão de alto nível dos objetivos da arquitetura, dos estilos arquiteturais e componentes que foram selecionados para poder estruturar as funcionalidades propostas pelos casos de usos levantados do Pedidos Já.

## Finalidade

Este documento oferece uma visão geral arquitetural abrangente do Sistema Pedidos Já, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao utilizando a linguagem de modelagem unificada (UML – *Unified Modeling Language*).

## Escopo

Este Documento de Arquitetura de Software é aplicado ao Sistema Pedidos Já, que será desenvolvido pelos alunos Daniel Lima Oliveira e Everton Mendonça Lima do curso de Sistemas de Informação da Universidade Federal de Sergipe – Campus Itabaiana, como projeto da disciplina Engenharia de Software II.

## Definições, Acrônimos e Abreviações

* UC – User Case.

## Referências

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

* Modelo de Casos de Uso
* Descrição de Casos de Uso
* Documento de Requisitos Funcionais e Não Funcionais
* *Rational Unified Process*

## Visão Geral

Com o objetivo de contemplar todos os aspectos da arquitetura, estratificamos o documento nas seguintes subseções:

* Subseção 02: Descreve a representação da arquitetura do software;
* Subseção 03: Descreve as restrições arquiteturais do sistema;
* Subseção 04: Descreve os requisitos funcionais que causam significante impacto na arquitetura;
* Subseção 05: Descreve a visão lógica da arquitetura;
* Subseção 06: Descreve a visão de processos;
* Subseção 07: Descreve a visão de implantação;
* Subseção 08: Descreve a visão de implementação;
* Subseção 09: Descreve as principais características de dimensionamento do software que têm um impacto na arquitetura;
* Subseção 10: Mostra a hierarquia de exceções.

# Representação Arquitetural

Este documento apresenta a arquitetura como um arranjo das visualizações, mencionadas acima. Essas visões são apresentadas como Modelos do StarUML e utiliza a Linguagem de Modelagem Unificada (UML – *Unified Modeling Language).*

Para representar a arquitetura do software, foram utilizados como base os seguintes estilos arquiteturais:

* Camadas;
* Cliente-Servidor;
* Repositório de banco de dados.

# Metas e Restrições da Arquitetura

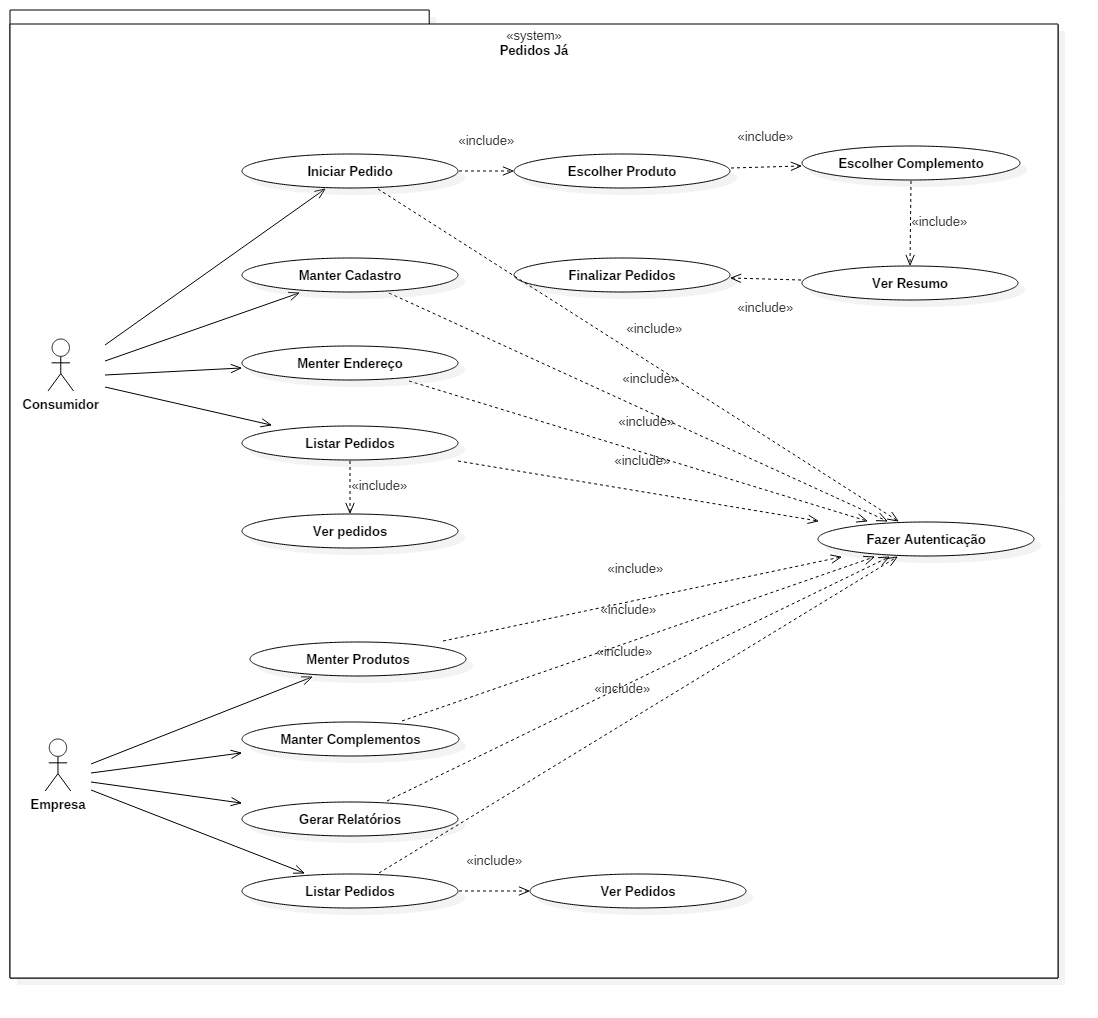
Para que o software tenha um comportamento esperado por nossos *stakeholders*, definimos as seguintes restrições:

* O Sistema deverá ser *Cross-Browser*, ou seja, as funcionalidades por ele oferecidas deverão funcionar plenamente nos mais diversos browsers disponíveis no mercado, a exemplo, do *Mozilla Firefox*, *Google Chrome, Opera, Safari, Netscape* e *Internet Explorer*, em suas versões mais recentes;
* O sistema será implementado utilizando os recursos inicialmente disponibilizados pelo *.NET Framework* e ASP.NET MVC, tendo como linguagem de programação principal o C#;
* O sistema terá os seus dados persistidos no banco de dados PostgreSQL;
* Apenas usuários autenticados podem ser acessar o sistema;
* O Sistema será multiplataforma, funcionado principalmente nos sistemas operacionais das plataformas Windows, Linux e MAC OS, em suas versões mais recentes, principalmente;
* O Sistema terá como *Design Pattern* o modelo MVC.

# Visão de Casos de Uso

Nessa seção serão listados os casos de uso responsáveis por nos fornecer num alto-nível de abstração as principais funcionalidades e comportamentos esperados do sistema. Esses casos de uso são:

* UC 01 - Fazer Autenticação;
* UC 02 - Listar Pedidos Empresa (Empresa);
* UC 03 - Gerar Relatórios (Empresa);
* UC 04 - Manter Produtos (Empresa);
* UC 05 - Manter Complementos (Empresa);
* UC 06 - Manter Cadastro;
* UC 07 - Ver Pedido;
* UC 08 - Listar Pedidos;
* UC 09 - Manter Endereços;
* UC 11 - Iniciar Pedido;
* UC 12 - Escolher Produtos;
* UC 13 - Escolher Complementos
* UC 14 - Ver Resumo
* UC 15 - Finalizar Pedido
* UC 16 - Ver Pedido Empresa (Empresa)

****

**Figura 1 – Diagrama de Casos de Uso**

## Realizações de Casos de Uso

A seguir temos uma breve descrição de cada caso de uso, utilizado na descrição das principais funcionalidades do sistema. São eles:

* UC 01 - Fazer Autenticação – Autenticação no sistema;
* UC 02 - Listar Pedidos Empresa (Empresa) – Empresa lista os pedidos feitos a ela;
* UC 03 - Gerar Relatórios (Empresa) – Empresa gera relatório de vendas;
* UC 04 - Manter Produtos (Empresa) – Empresa cadastra, altera, consulta e remove produtos;
* UC 05 - Manter Complementos (Empresa) - Empresa cadastra, altera, consulta e remove complementos;
* UC 06 - Manter Cadastro – Usuário altera dados pessoais;
* UC 07 - Ver Pedido – Usuário ver resumo do pedido já feito;
* UC 08 - Listar Pedidos – Usuário lista os pedidos feitos;
* UC 11 - Iniciar Pedido – Usuário inicia o pedido;
* UC 12 - Escolher Produtos – Usuário escolhe os produtos do pedido;
* UC 13 - Escolher Complementos – Usuário escolhe complementos para os produtos escolhidos;
* UC 14 - Ver Resumo – Usuário ver o resumo do pedido antes de confirmá-lo;
* UC 15 - Finalizar Pedido – Usuário finaliza o pedido;
* UC 16 - Ver Pedido Empresa (Empresa) – Empresa vê um pedido específico feito a ela.

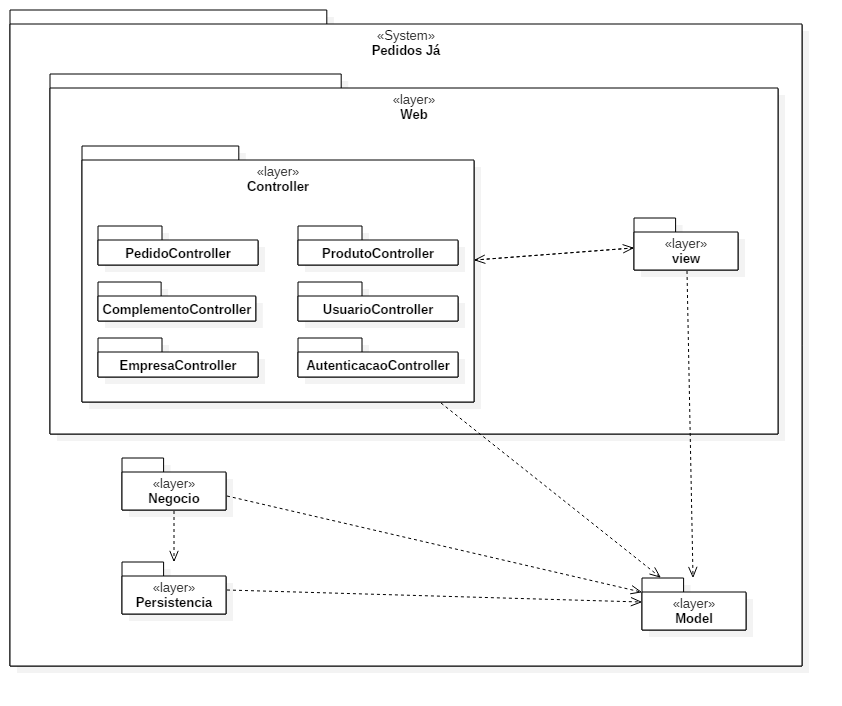
# Visão Lógica

## Visão Geral

A visão lógica do Sistema Pedidos Já é composta principalmente por 3 pacotes:

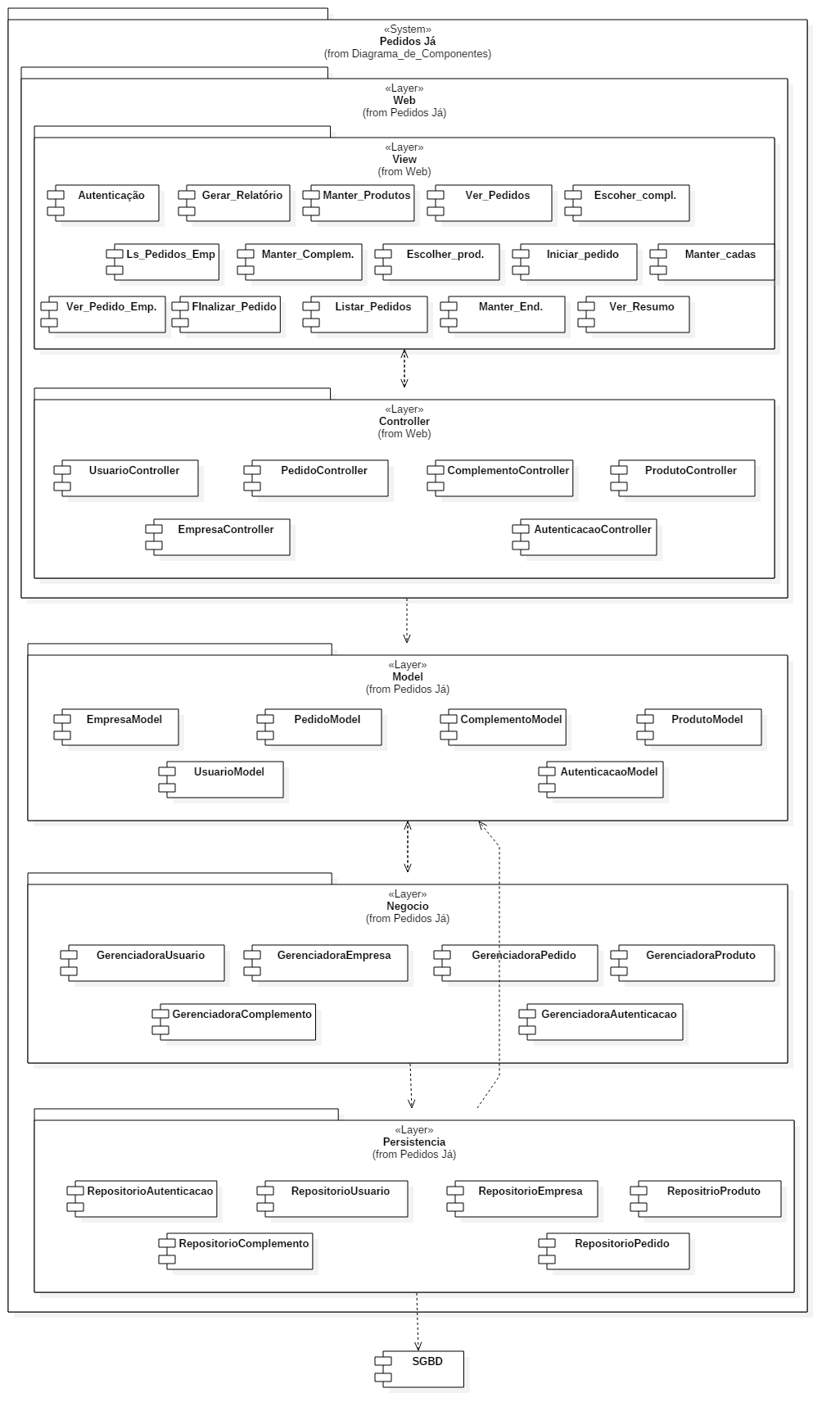
* Model: Esse pacote representa a implementação da parte lógica do domínio da aplicação. Aqui serão armazenadas classes que representam as entidades da aplicação e classes responsáveis pela persistência dos dados;
* Views: Nesse pacote são armazenados os componentes responsáveis por prover a interface gráfica da aplicação;
* Controller: Nesse pacote são armazenados os componentes do tipo Controller. Tais componentes são responsáveis por relacionar os Models às Views, de acordo com a interação do usuário.

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



**Figura 2 – Diagrama de Pacotes**

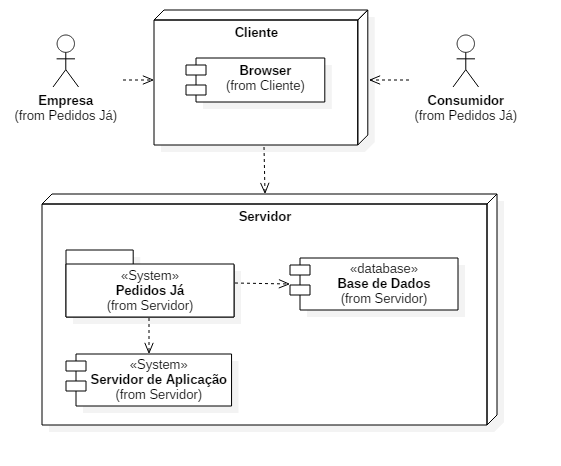
## Diagrama de componentes



**Figura 3 – Diagrama de Componentes**

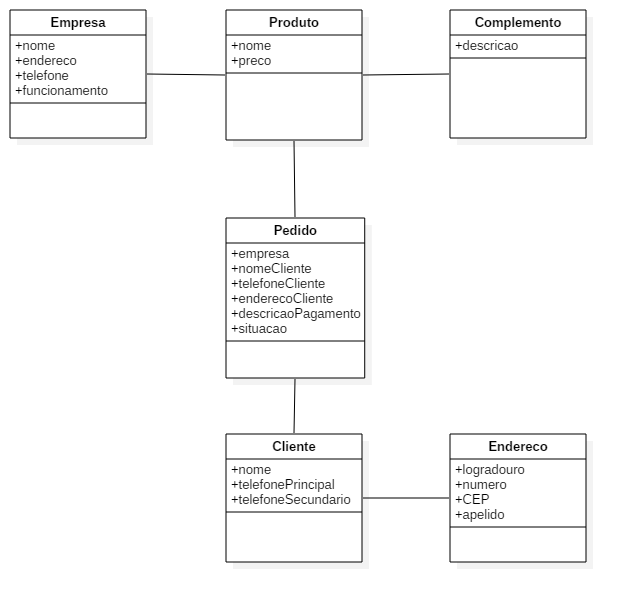
# Visão de Processos

# Visão de Implantação



**Figura 4 – Diagrama de Implantação**

# Visão da Implementação



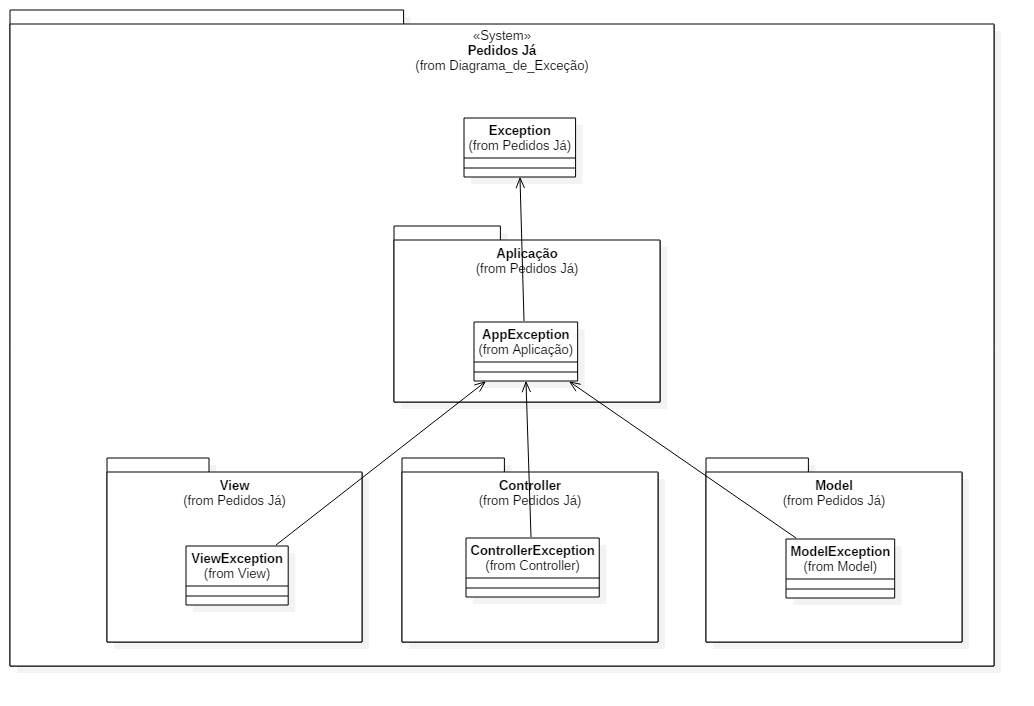
**Figura 5 – Diagrama de Classes**

# Qualidade

O sistema possibilitara ao usuário módulo de autenticação, design minimalista, responsivo e simplificado facilitando a usabilidade.

# Exceções

O tratamento de exceções se dará preferencialmente em camadas. O sistema possuirá tratamento de exceções específicas definidas pela *Application.Exception* que utilizará os tratamentos de exceções genéricos definidos em *System.Exception.*



**Figura 6 – Diagrama da Arquitetura de Exceções**