

---

# Documentação de Projeto

para o sistema

# Picnic Caseiros

**Versão 1.0**

Projeto de sistema elaborado pelo(s) aluno(s) Douglas Fernandes, Davi Brandão, Guilherme Costa, Filipe Iannarelli e Raul Goulart como parte da disciplina **Projeto de Software**.

20 de março de 2024

## Tabela de Conteúdo

1.	Introdução .....	1
2.	Modelos de Usuário e Requisitos .....	1
2.1	Descrição de Atores .....	1
2.2	Modelo de Casos de Uso.....	1
2.3	Diagrama de Sequência do Sistema.....	3
3.	Modelos de Projeto .....	3
3.1	Arquitetura .....	3
4.	Diagrama de Componentes e Implantação .....	5
4.1	Diagrama de Classes .....	6
4.2	Diagramas de Sequência .....	6
4.3	Diagramas de Comunicação .....	6
4.4	Diagramas de Estados .....	6
5.	Modelos de Dados.....	7

## Histórico de Revisões

Nome	Data	Razões para Mudança	Versão
Guilherme Costa	20/03/2024	Emissão inicial	1.0

## **1. Introdução**

Este documento agrega a elaboração e revisão de modelos de domínio e de projeto para o sistema Picnic Caseiros. Nosso projeto visa desenvolver um sistema sob medida para a Picnic Caseiros, uma empresa dedicada à produção e venda de bolos caseiros e outros alimentos artesanais. Nosso objetivo é otimizar a gestão de insumos, receitas, vendas e fidelização de clientes, impulsionando o crescimento sustentável e a eficiência operacional.

## **2. Modelos de Usuário e Requisitos**

### **2.1 Descrição de Atores**

Na modelagem desenvolvida para o projeto, foram mapeados dois atores distintos:

- Usuário: pode apenas realizar agendamentos de pedidos pela aplicação;
- Administrador: pode realizar todas as demais tarefas previstas, como gerenciar os insumos, receitas, formas de pagamentos, dentre outros.

### **2.2 Modelo de Casos de Uso**

Nessa seção será apresentado o diagrama de caso de uso do projeto. Neste diagrama são elencados os atores que participam da interação, bem como as suas possíveis ações. Conforme pode ser observado na Figura 1 a seguir, o ator Usuário possui acesso às funcionalidades relacionadas ao agendamento de pedidos. O ator Administrador, que herda as permissões de Usuário, pode interagir com diversas outras funcionalidades, como monitorar o processo de fidelização dos clientes, gerenciar os insumos cadastrados, gerenciar as formas de venda dos produtos, gerenciar as vendas realizadas, gerenciar as receitas, gerenciar a precificação dos produtos e gerenciar fornecedor e clientes. Outro ponto relevante, é que o Administrador também pode gerenciar e emitir relatórios das funcionalidades de fidelização de clientes, gerenciamento de insumos, formas de venda e vendas realizadas.

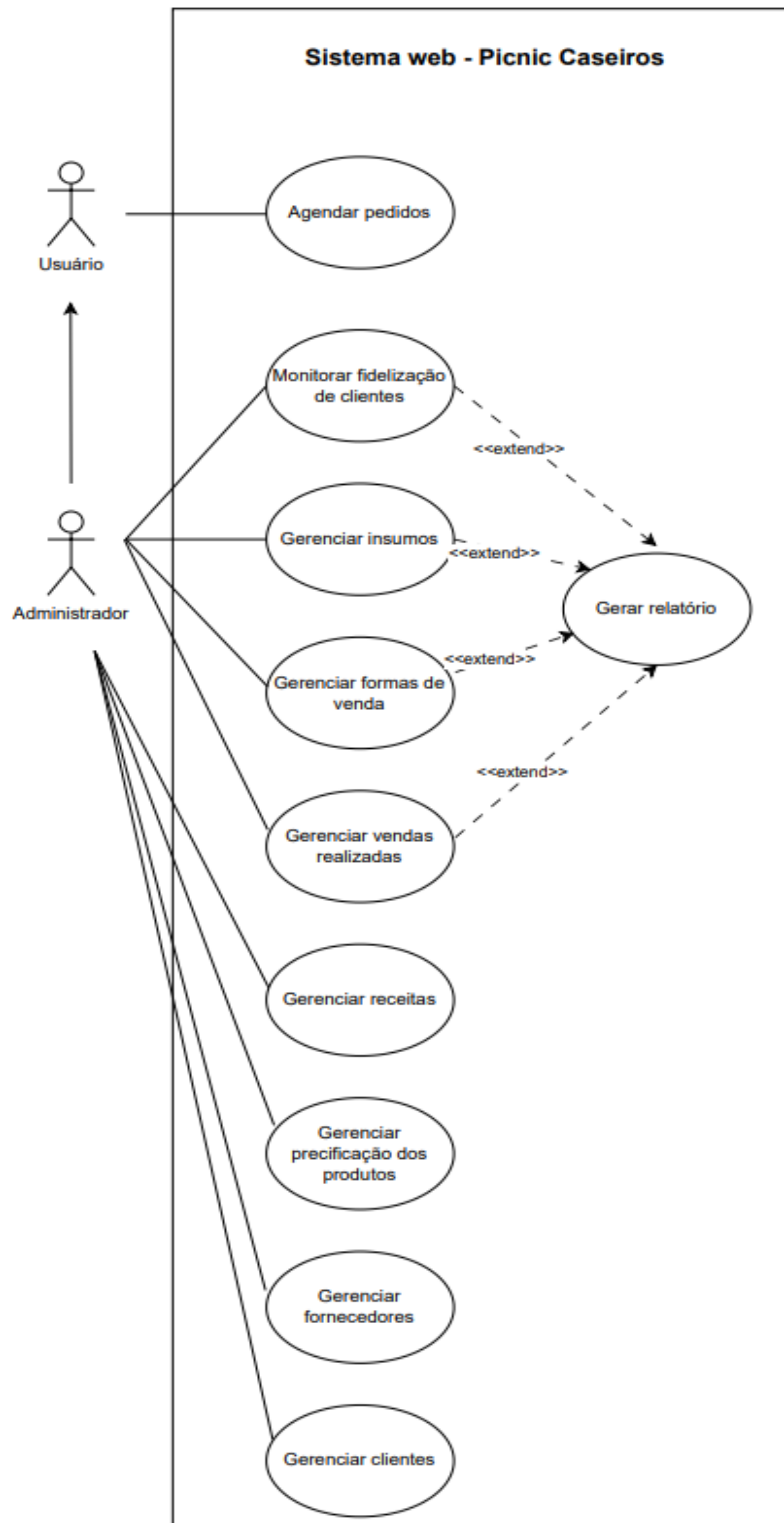


Figura 1 - Diagrama de casos de uso do projeto

## 2.3 Diagrama de Sequência do Sistema

Nesta subseção é apresentado o diagrama de sequência do sistema de pelo menos, 3 Casos de Uso ou Histórias de Usuário descritos na Seção 2.3.

Formato para cada contrato de operação

Contrato	
Operação	
Referências cruzadas	
Pré-condições	
Pós-condições	

## 3. Modelos de Projeto

### 3.1 Arquitetura

No framework Laravel, adotado para desenvolvimento da aplicação, a arquitetura em camadas é organizada de forma a promover uma estrutura modular e escalável para o desenvolvimento de aplicativos da web, facilitando a manutenção e teste de código. Essa arquitetura consiste em várias camadas interconectadas:

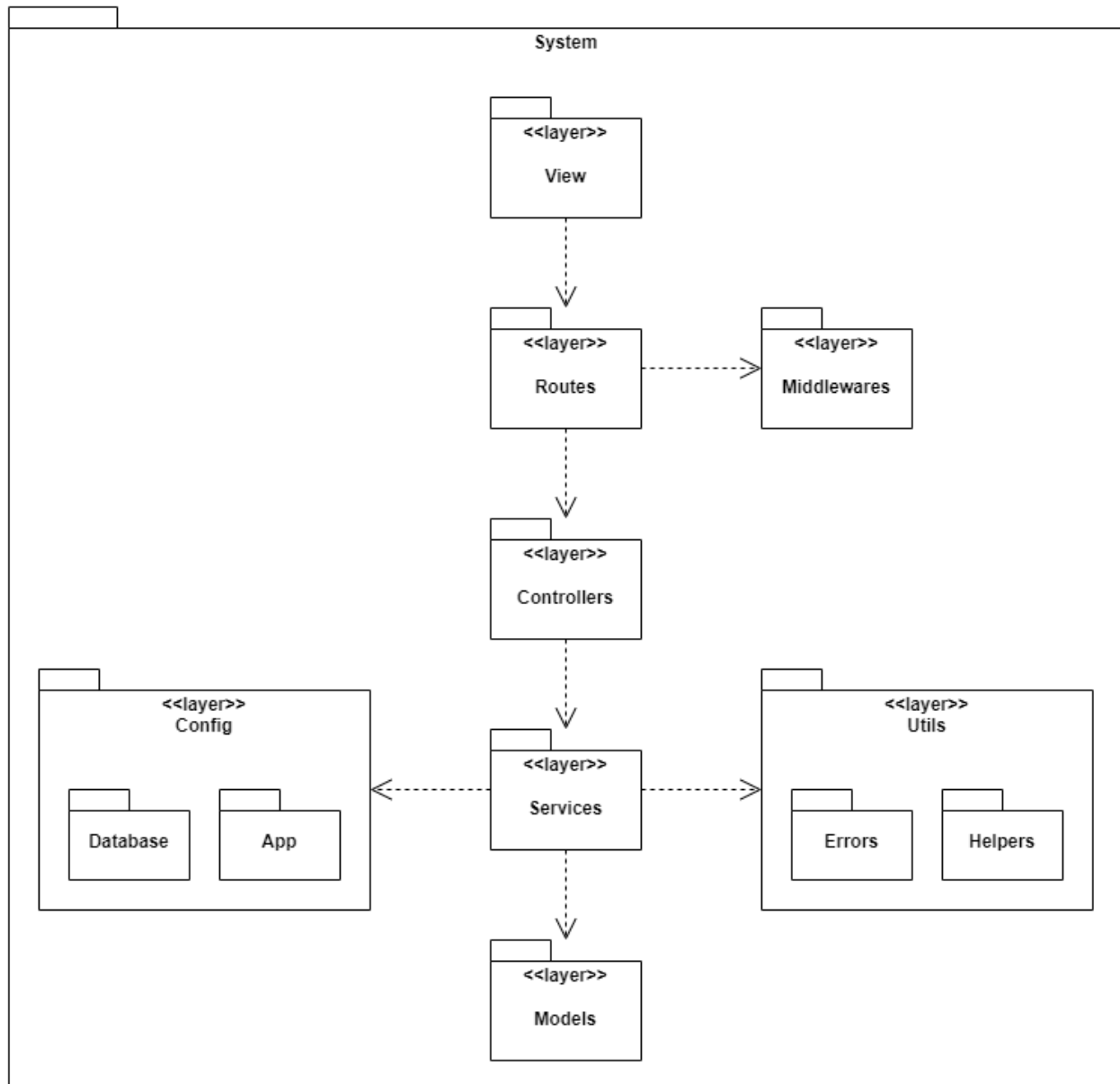
A camada de Views é responsável pela apresentação da interface do usuário e é onde o conteúdo é renderizado para o navegador. As views podem conter HTML, CSS, JavaScript e elementos dinâmicos incorporados usando a sintaxe Blade do Laravel. As Routes definem as URLs do aplicativo e direcionam essas URLs para os respectivos controladores. As rotas podem ser definidas de forma explícita no arquivo de rotas ou usando convenções de nomenclatura RESTful.

Os Middlewares são camadas intermediárias que podem ser aplicadas a rotas individuais ou a grupos de rotas. Eles fornecem uma maneira de filtrar as solicitações HTTP antes que elas alcancem os controladores, permitindo a execução de ações como autenticação, autorização e manipulação de requisições. Os Controllers são responsáveis por receber as requisições do cliente, processá-las e retornar uma resposta apropriada. Eles agem como intermediários entre as rotas e os serviços de negócios, coordenando a lógica de aplicativo e manipulando os dados recebidos das requisições.

Os Services são classes que encapsulam a lógica de negócios do aplicativo. Eles fornecem uma interface limpa para acessar e manipular os dados do aplicativo, promovendo a reutilização e a manutenção do código. Os Models representam e gerenciam os dados do aplicativo. Eles mapeiam

diretamente para as tabelas do banco de dados e fornecem métodos para realizar operações CRUD (Create, Read, Update, Delete) nos dados.

A camada de Utils engloba classes utilitárias, como classes de manipulação de erros e classes de helper, que fornecem funcionalidades auxiliares para o desenvolvimento do aplicativo. A camada de Config inclui arquivos de configuração que definem várias configurações do aplicativo, como configurações de banco de dados, configurações de ambiente e configurações gerais do aplicativo.



## 4. Diagrama de Componentes e Implantação

Serão apresentados nesta seção os diagramas de componente e implantação do projeto.

O primeiro deles, o diagrama de componentes (Figura 2), indica os principais componentes do sistema, e como eles são relacionados em termos de dependência um do outro, além das interfaces que utilizam e disponibilizam. O diagrama de implantação (Figura 3), por sua vez, representa os recursos físicos e de software que são necessários para que o sistema seja implementado corretamente.

Conforme pode ser observado na Figura 2, o principal componente do diagrama de componentes é o servidor web, que vai se relacionar diretamente com diversos outros componentes, como *controllers*, *services*, *models* e *configuration*. Portanto, é nesse componente que estará a comunicação com o banco de dados e disponibilização de API externa para a aplicação web.

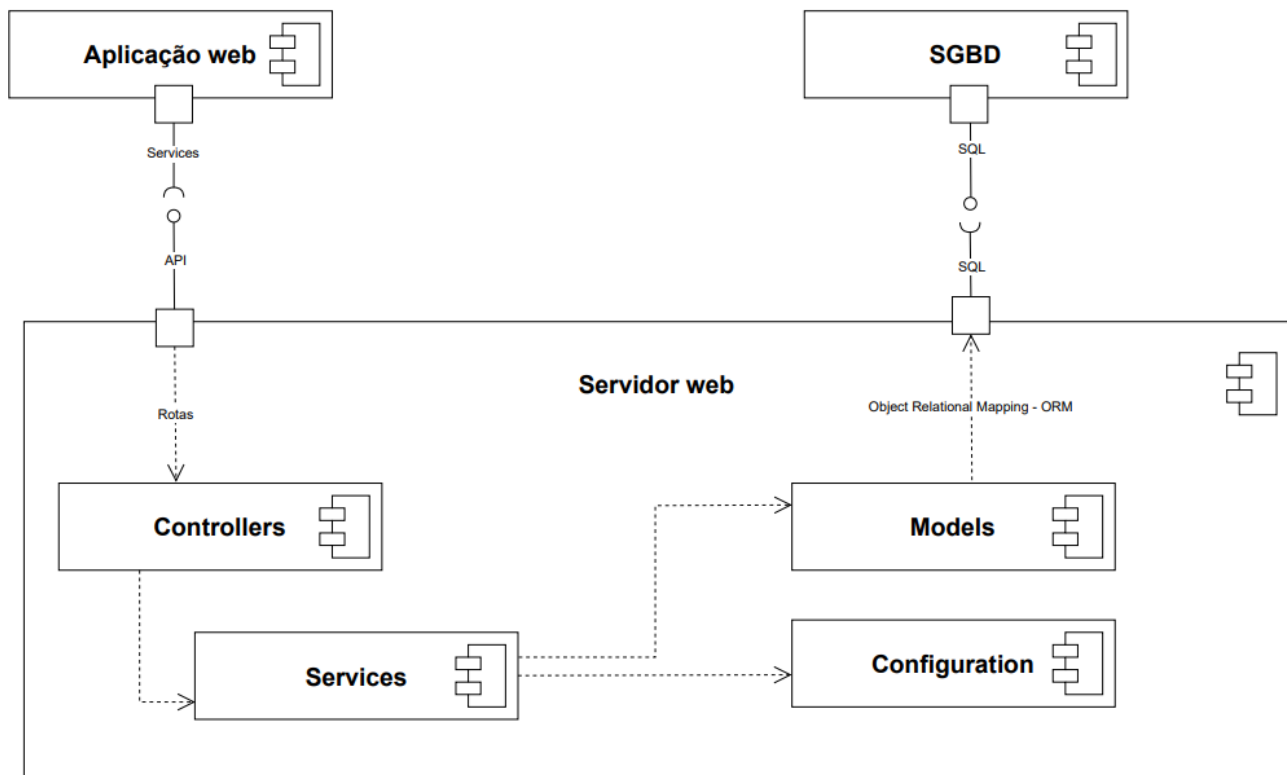


Figura 2 - Diagrama de componentes

Na figura a seguir (Figura 3), é apresentado o diagrama de implantação do sistema. Neste diagrama são apresentados os pontos de processamento do projeto, e onde está sendo utilizado. Conforme ilustrado nessa figura, o navegador web (acessado diretamente pelo usuário) faz requisições ao nosso *back-end*, que é encapsulado por uma máquina virtual em um serviço de nuvem, por meio de protocolos HTTPs. O *back-end*, por sua vez, está relacionando com o banco de dados, fazendo as atualizações / modificações necessárias de acordo com as requisições recebidas.

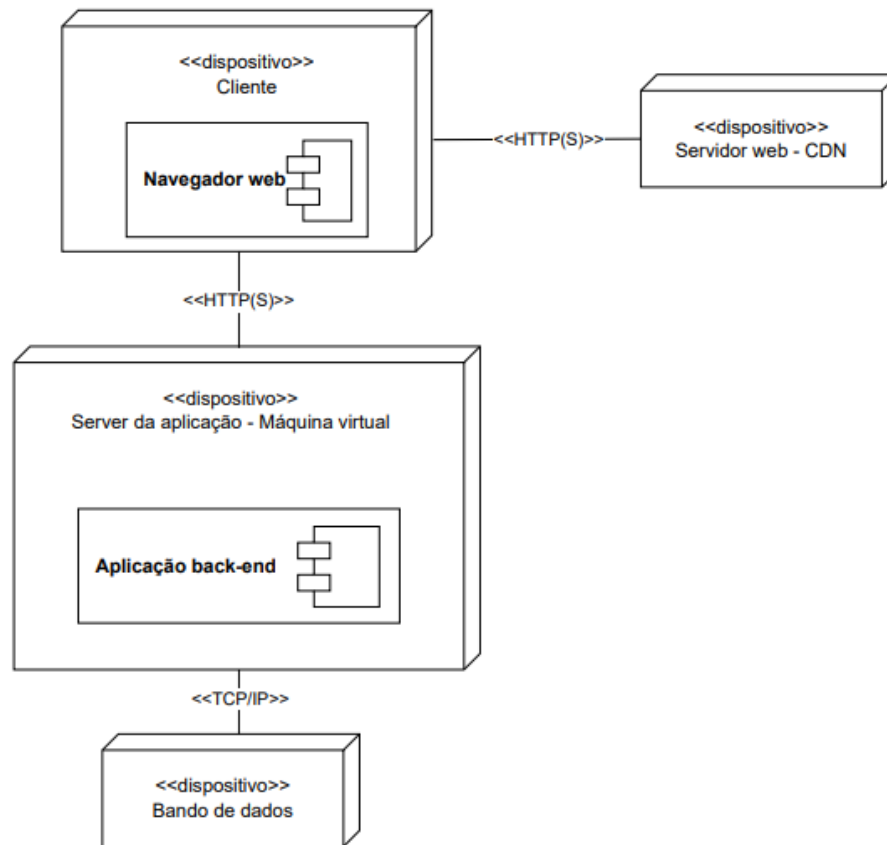


Figura 3 - Diagrama de implantação

## 4.1 Diagrama de Classes

Diagrama de classes do sistema.

## 4.2 Diagramas de Sequência

Diagramas de sequência para realização de casos de uso.

## 4.3 Diagramas de Comunicação

Diagramas de comunicação para realização de casos de uso.

## 4.4 Diagramas de Estados

Diagramas de estados do sistema.



## **5. Modelos de Dados**

Deve-se apresentar os esquemas de banco de dados e as estratégias de mapeamento entre as representações de objetos e não-objetos.