

Picnic Caseiros

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.

Introdução

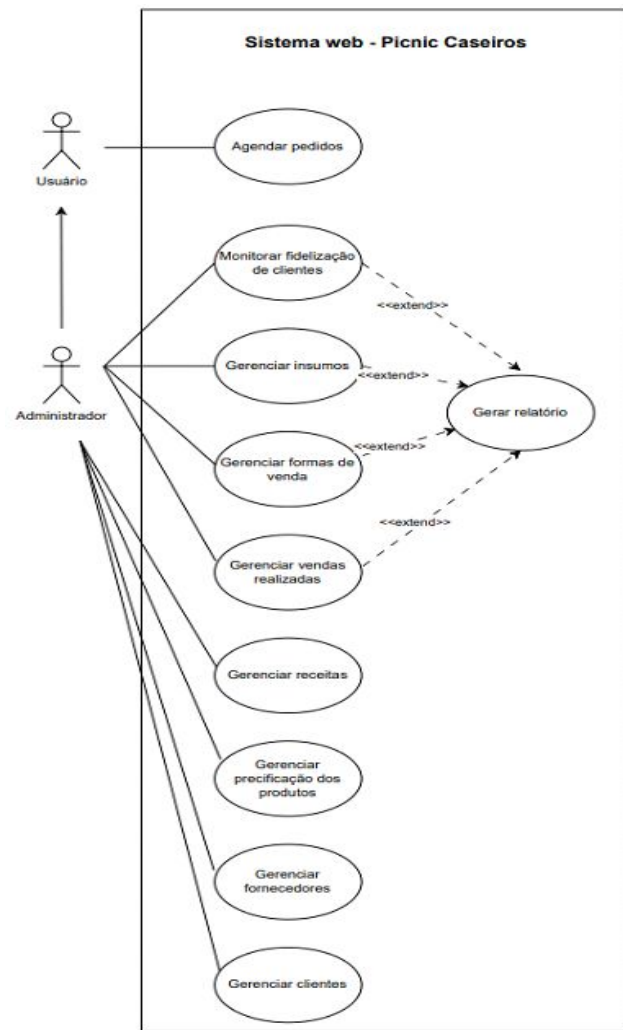
Nosso projeto visa desenvolver um sistema sob medida para a Picnic Caseiros, uma empresa dedicada à produção e venda de bolos caseiros e outros alimentos artesanais. Nosso objetivo é otimizar a gestão de insumos, receitas, vendas e fidelização de clientes, impulsionando o crescimento sustentável e a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

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.

Nesta seção será apresentado e contextualizado o diagrama de caso de uso do projeto



Descrição do Diagrama de Casos de Uso

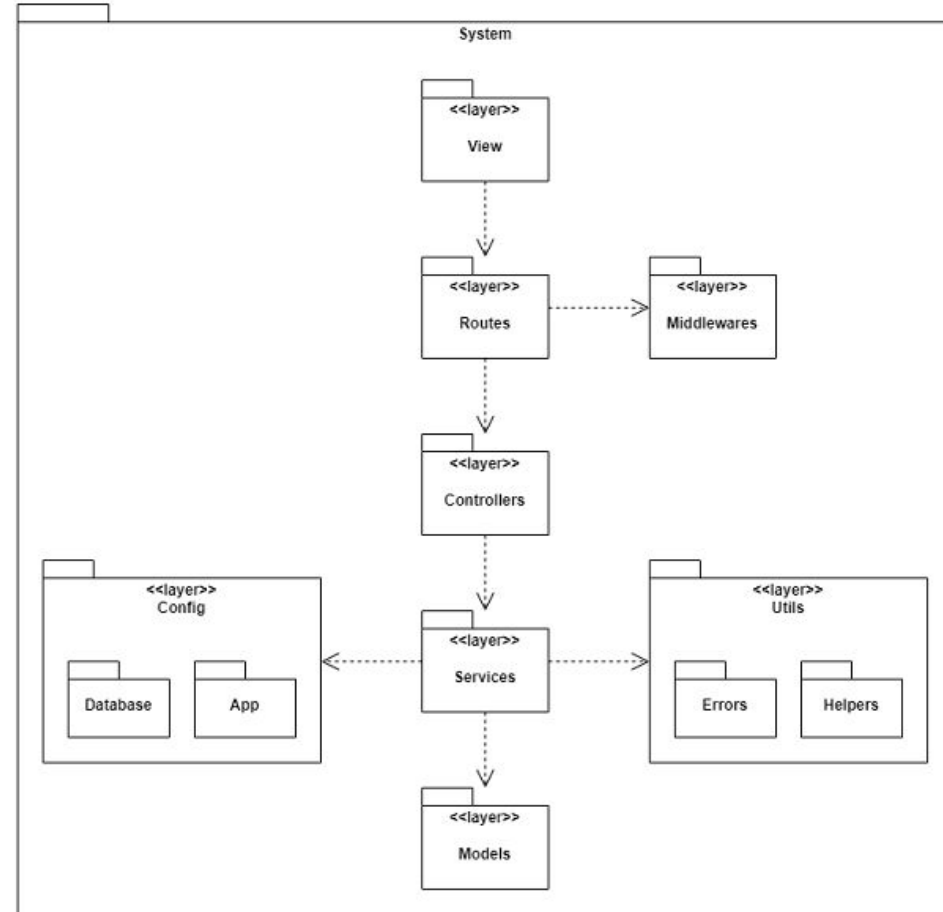
O ator Administrador (herda as permissões de Usuário), pode interagir com funcionalidades como:

- 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
- gerenciar fornecedor e clientes.

O ator Usuário possui acesso às funcionalidades relacionadas ao agendamento de pedidos.

Arquitetura

Framework Laravel foi adotada para o desenvolvimento da aplicação.



Descrição da Arquitetura

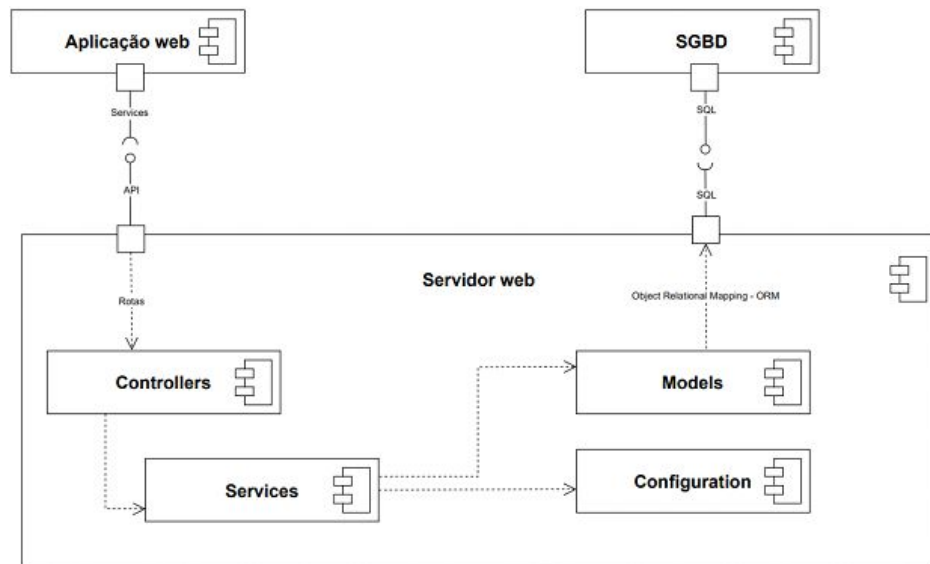
Na framework adotada, a arquitetura em camadas é organizada de forma a promover uma estrutura modular e escalável para o desenvolvimento de aplicativos da web, facilitando a manutenção e teste de código. Essa arquitetura consiste em várias camadas interconectadas

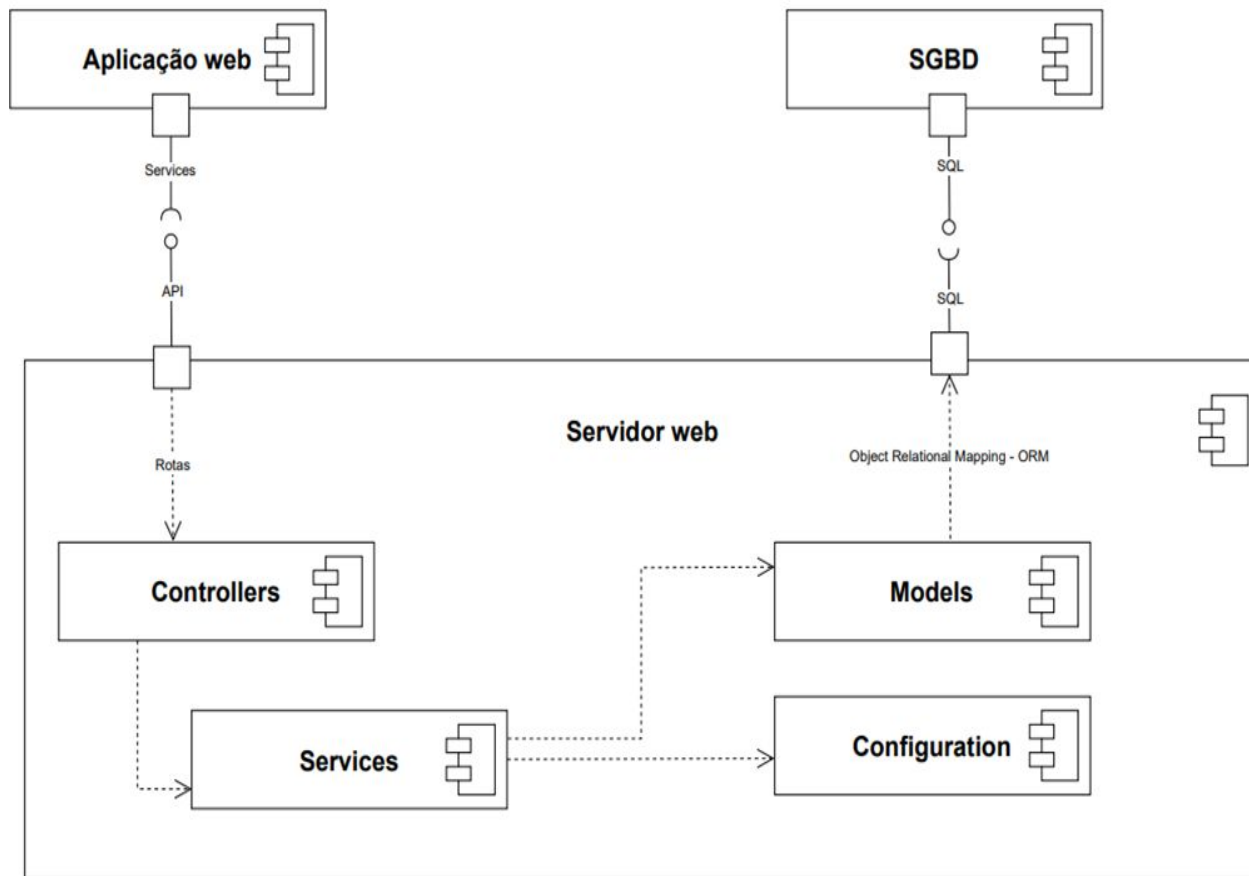
- A camada de Views é responsável pela apresentação da interface do usuário e é onde o conteúdo é renderizado para o navegador.
- As Routes definem as URLs do aplicativo e direcionam essas URLs para os respectivos controladores.
- Os Middlewares são camadas intermediárias que podem ser aplicadas a rotas individuais ou a grupos de rotas.
- Os Controllers são responsáveis por receber as requisições do cliente, processá-las e retornar uma resposta apropriada.

Descrição da Arquitetura

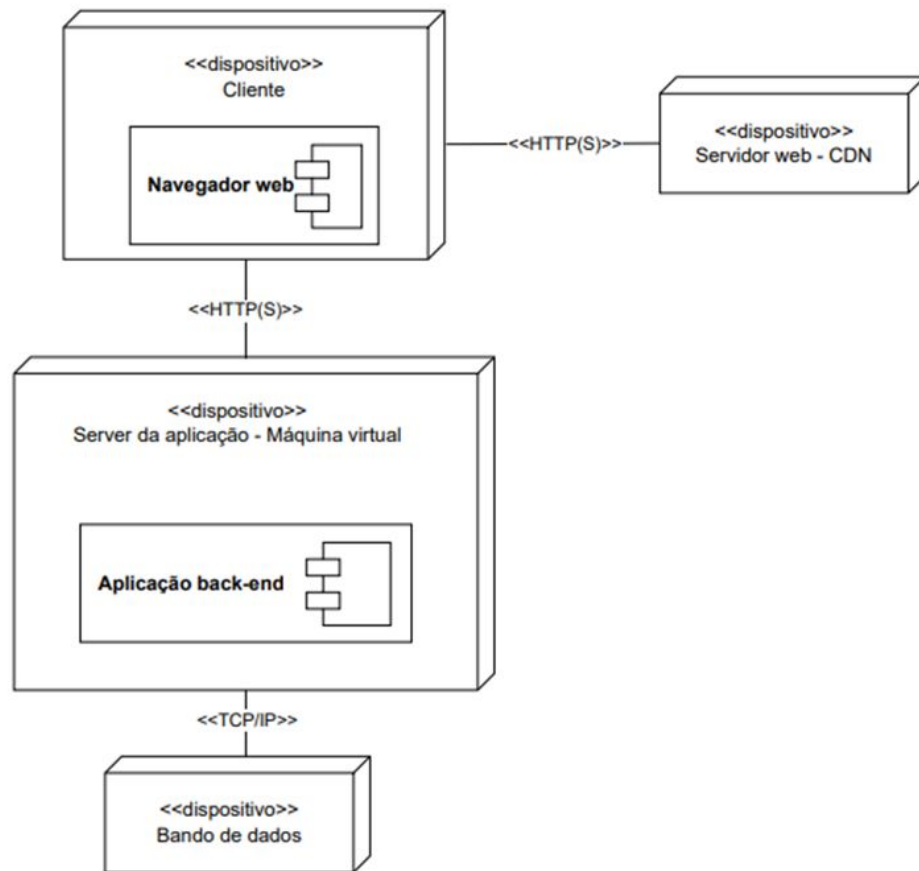
- Os Services são classes que encapsulam a lógica de negócios do aplicativo
- Os Models representam e gerenciam os dados do aplicativo.
- A camada de Utils engloba classes utilitárias (erros, helper etc.)
- A camada de Config definem várias configurações do aplicativo (configurações de banco de dados, configurações de ambiente e configurações gerais do aplicativo)

Diagrama de Componentes e Implantação





O principal componente deste diagrama 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.



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 relacionado com o banco de dados, fazendo as atualizações e modificações necessárias de acordo com as requisições recebidas.

Obrigado!

