

**PONTIFÍCIA UNIVERSIDADE CATÓLICA  
DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-Graduação Lato Sensu em Arquitetura de  
Software Distribuído**

**Projeto Integrado**

**Relatório Técnico**

**Sistema de Gestão de Delivery**

**Julio Cezar da Silva**

**Belo Horizonte  
2023**

**Julio Cezar da Silva**

## **Sistema de Gestão de Delivery**

Trabalho de Conclusão de Curso de Especialização em Arquitetura de Software  
Distribuído como requisito parcial à obtenção do título de especialista.

Orientador: Prof. Luiz Alberto

**Belo Horizonte  
2023**

## Sumário

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Objetivo</b>	<b>4</b>
2.1	Objetivos Específicos . . . . .	4
<b>3</b>	<b>Restrições Arquiteturais</b>	<b>5</b>
<b>4</b>	<b>Requisitos Funcionais</b>	<b>6</b>
4.1	Módulo de produtos . . . . .	6
4.1.1	RF-1: Cadastro de produtos . . . . .	6
4.1.2	RF-2: Cadastro de categoria . . . . .	6
4.1.3	RF-3: Cadastro de complementos . . . . .	6
4.1.4	RF-4: Cadastro de combos . . . . .	6
4.2	Módulo de clientes . . . . .	6
4.2.1	RF-5: Cadastro de clientes . . . . .	6
4.2.2	RF-6: Cadastro de endereços de clientes . . . . .	7
4.3	Módulo de pedidos . . . . .	7
4.3.1	RF-7: Cadastro de pedidos delivery . . . . .	7
4.3.2	RF-8: Cadastro de pedidos mesa . . . . .	7
4.3.3	RF-9: Cadastro de pedidos cliente . . . . .	7
4.4	Módulo de estoque . . . . .	7
4.4.1	RF-10: Cadastro de ingredientes . . . . .	7
4.4.2	RF-11: Relatório de estoque . . . . .	7
4.5	Módulo financeiro . . . . .	8
4.5.1	RF-12: Formas de pagamento . . . . .	8
4.5.2	RF-13: Pagamento Online . . . . .	8
4.6	Módulo autenticação/autorização . . . . .	8
4.6.1	RF-14: Níveis de acesso . . . . .	8
4.6.2	RF-15: Autenticação . . . . .	8
<b>5</b>	<b>Requisitos Não Funcionais</b>	<b>9</b>
5.1	Manutenabilidade . . . . .	9
5.1.1	RNF-1: Desenvolvimento em camadas . . . . .	9
5.2	Usabilidade . . . . .	9
5.2.1	RNF-2: Padrão de interfaces . . . . .	9
5.3	Segurança . . . . .	9
5.3.1	RNF-3: Banco de dados . . . . .	9
5.3.2	RNF-4: Autenticação/Autorização . . . . .	9
5.4	Interoperabilidade . . . . .	9
5.4.1	RNF-5: Comunicação com sistema de pagamento . . . . .	9
5.5	Desempenho . . . . .	10
5.5.1	RNF-6: Acesso após Autenticação . . . . .	10
<b>6</b>	<b>Mecanismos Arquiteturais</b>	<b>11</b>

<b>7</b>	<b>Modelagem Arquitetural</b>	<b>12</b>
7.1	Diagrama de Contexto . . . . .	12
7.2	Diagrama de Container . . . . .	13
7.3	Diagrama de Componentes . . . . .	14
<b>8</b>	<b>Vídeo Apresentação 1</b>	<b>15</b>

## 1. Introdução

O mercado de delivery tem crescido de forma significativa no Brasil, o mesmo foi impulsionado ainda mais devido a pandemia de COVID-19.

Segundo a Associação brasileira de Bares e Restaurantes(Abrasel), o setor de alimentação movimentou cerca de R\$ 200 bilhões em 2019, sendo que destes, 11% foi representado pelo delivery. Em 2020 com a pandemia, o mercado cresceu para 20% do faturamento total do setor[1].

Durante a pandemia tivemos um grande aumento no uso de plataformas de delivery, principalmente do ifood[2] que hoje é a principal plataforma de delivery no Brasil. Essas plataformas possuem uma taxa altíssima, mesmo as altas taxas aplicadas, que variam de 12% a 27%, além de uma mensalidade em torno de R\$ 100,00[5].

De acordo com o Instituto FoodService Brasil(IFB), no Brasil espera-se que o delivery cresça em torno de 7,5% em 2023, fazendo com que os estabelecimentos invistam em ampliar e aprimorar este serviço de entrega de comida[4].

A transformação digital[3] está presente nas nossas atividades diárias, e não está limitada ao uso pessoal, o setor de delivery está cada vez mais preparado digitalmente, e os estabelecimentos que não adotarem as tecnologias que possam facilitar seus processos, controles e integração com os clientes ficarão para trás no mercado.

Com a utilização de tecnologias como sistemas gerenciais, o estabelecimento evita e minimiza erros e prejuízos nos seus processos, tem um melhor controle de estoque, financeiro e diversas rotinas do seu comércio. Além de otimizar o tempo tanto de preparação com as comandas chegando automaticamente via sistema, sem a movimentação de papel que pode se perder no processo, quanto a agilidade em atender o cliente e cadastrar seu pedido, algo que pode ser inclusive feito pelo próprio cliente direto de sua casa.

## 2. Objetivo

O objetivo do projeto é apresentar uma arquitetura de um sistema para estabelecimentos de alimentação, de modo que os mesmos possam ter um controle dos seus produtos, clientes e da sua rotina. Além de um sistema que permita aos clientes efetuarem pedidos de forma online.

Portanto, com objetivo de desenvolver a modelagem de um software no qual os estabelecimentos possam utilizar e oferecer através do mesmo diversos benefícios aos seus clientes, como promoções, fidelização do cliente, mensagens diretas ao smartphone do cliente, até mesmo preços mais baixos desde que o estabelecimento faça um bom marketing para atrair clientes que utilizem plataformas de pedidos como ifood.

O cliente poderá acessar o software através de um aplicativo para smartphone, seja uma progressive web page, ou mesmo um aplicativo nativo, o que será definido no decorrer das análises e projeto arquitetural.

### 2.1. Objetivos Específicos

Os objetivos específicos são:

- Permitir ao estabelecimento ter um controle dos seus cadastros de clientes, produtos, promoções, etc.
- Oferecer ao cliente do estabelecimento um sistema facilitador para que o mesmo possa efetuar pedidos diretamente da sua casa.
- Facilitar ao estabelecimento oferecer promoções diretamente ao cliente em casa, através do próprio sistema, isto é, enviar promoções para o cliente, mensagens de aniversário ou que mais convier ao estabelecimento.
- Facilitar o controle de estoque do estabelecimento.
- Facilitar o controle financeiro do estabelecimento.
- Permitir o fácil gerenciamento do estabelecimento através de diversos relatórios.
- Oferecer ao cliente a opção de pagar online, para isso deverá ser feito integrações com os principais players de pagamento, como pagseguro, mercadopago, etc.
- Permitir ao estabelecimento receber pedidos de plataformas de delivery como ifood.

### 3. Restrições Arquiteturais

Table 3.1: Restrições

ID	Descrição
RA001	O software deverá ser desenvolvido em Java utilizando o framework Spring
RA002	As APIs devem seguir o padrão RestFul
RA003	O software deverá integrar com plataformas de delivery, de forma a permitir os clientes pedirem por ela, como ifood.
RA004	O software deverá integrar com pelo menos um sistema de pagamento online, como pagseguro, mercadopago.
RA005	O software administrativo do estabelecimento deverá ter níveis de acesso para proteger dados sensíveis, como endereço e outras informações pessoais dos clientes

## 4. Requisitos Funcionais

Abaixo listamos os requisitos funcionais da aplicação[6]. Os requisitos terão níveis de dificuldade e prioridades definidos em B - Baixa, M - Média e A - Alta.

### 4.1. Módulo de produtos

#### 4.1.1. RF-1: Cadastro de produtos

O sistema deve permitir que o colaborador cadastre produtos com nome, categoria, preço de custo por tamanho, preço de venda por tamanho, unidade de medida, descrição, foto, se será feito controle de estoque, ficha técnica, complementos.

*Dificuldade: M - Prioridade: A*

#### 4.1.2. RF-2: Cadastro de categoria

O sistema deve permitir cadastrar a categoria de produtos, definindo possíveis tamanhos dos produtos nessa categoria ou tamanho única, por exemplo: Refrigerantes, definir os tamanhos 350ml, 500ml, 1l, 1.5l, 2l, etc.

*Dificuldade: M - Prioridade: A*

#### 4.1.3. RF-3: Cadastro de complementos

O sistema deverá permitir cadastrar complementos e seus valores, os mesmos deverão ser ligados ao tamanho do produto no cadastro de produtos. Exemplo: Acrescimo de Bacon pizza M, Acrescimo de Bacon pizza G e assim por diante.

*Dificuldade: B - Prioridade: M*

#### 4.1.4. RF-4: Cadastro de combos

O sistema deverá permitir o cadastro de combos, juntando vários produtos cadastrados em um combo e definindo o valor para o combo.

*Dificuldade: A - Prioridade: M*

### 4.2. Módulo de clientes

#### 4.2.1. RF-5: Cadastro de clientes

O sistema deve permitir cadastrar os clientes com nome, email, data nascimento, telefone principal, cpf ou cnpj, identidade ou inscrição estadual, endereço principal(endereço, número, complemento, bairro, cidade, cep, valor do frete).

*Dificuldade: B - Prioridade: A*



#### **4.2.2. RF-6: Cadastro de endereços de clientes**

O sistema deverá permitir cadastrar mais endereços para o cliente, com nome do endereço e os campos definidos no requisito RF-004.

***Dificuldade: B - Prioridade: M***

### **4.3. Módulo de pedidos**

#### **4.3.1. RF-7: Cadastro de pedidos delivery**

O sistema deve permitir cadastrar pedidos para os clientes, selecionando cliente, produtos, endereço de entrega, forma de pagamento, status do pedido (Aberto, Em preparo, Pronto/Para retirar, Saiu para Entregar, Finalizado/Entregue), lançamento de desconto ou taxa de serviço.

***Dificuldade: A - Prioridade: A***

#### **4.3.2. RF-8: Cadastro de pedidos mesa**

O sistema deve permitir cadastrar pedidos para os clientes no estabelecimento, selecionando cliente, produtos, forma de pagamento, status do pedido (Aberto, Em preparo, Pronto/Para retirar, Saiu para Entregar, Finalizado/Entregue), lançamento de desconto ou taxa de serviço, divisão de valores.

***Dificuldade: A - Prioridade: A***

#### **4.3.3. RF-9: Cadastro de pedidos cliente**

O sistema deve permitir que o próprio cliente faça seu pedido online, selecionando os produtos, endereço de entrega.

***Dificuldade: A - Prioridade: A***

### **4.4. Módulo de estoque**

#### **4.4.1. RF-10: Cadastro de ingredientes**

O sistema deve permitir o gerenciamento do estoque através do cadastro de ingredientes, e nos produtos que controlam estoque ter a ficha técnica dos mesmos ligadas aos ingredientes. O cadastro deverá ter informações como nome do ingrediente, unidade de medida, data de vencimento, preço de custo.

***Dificuldade: A - Prioridade: M***

#### **4.4.2. RF-11: Relatório de estoque**

O sistema deve possuir relatório que mostre o estoque de produtos e os produtos com vencimento próximo, de modo a evitar perdas.

***Dificuldade: A - Prioridade: M***

#### **4.5. Módulo financeiro**

##### **4.5.1. RF-12: Formas de pagamento**

O sistema deve permitir o gerenciamento das formas de pagamento, para que possam ser selecionadas no lançamento do pedido. Exemplo: Cartão de Crédito, Cartão de Débito, Dinheiro, Pix, etc. Também as respectivas bandeiras quando necessário.

***Dificuldade: A - Prioridade: M***

##### **4.5.2. RF-13: Pagamento Online**

O sistema deve fazer integração com pelo menos 1 meio de pagamento online como pagseguro ou mercado pago.

***Dificuldade: A - Prioridade: B***

#### **4.6. Módulo autenticação/autorização**

##### **4.6.1. RF-14: Níveis de acesso**

O sistema deve permitir o cadastro/gerenciamento de níveis de acesso, a princípio dois níveis serão suficientes: Usuário e Administrador. Os usuários poderão fazer lançamento de pedidos, cadastros de clientes. Os administradores terão acesso irrestrito ao sistema.

***Dificuldade: B - Prioridade: A***

##### **4.6.2. RF-15: Autenticação**

O sistema deve ser fechado, não permitido o acesso a ele, somente para usuários cadastrados através de autenticação. Clientes externos deverão poder acessar um cardápio online e ao efetuar pedido fazer o devido cadastro.

***Dificuldade: A - Prioridade: A***

## 5. Requisitos Não Funcionais

Abaixo listamos os requisitos não funcionais da aplicação em termos arquiteturais[6].

### 5.1. Manutenabilidade

#### 5.1.1. RNF-1: Desenvolvimento em camadas

O sistema deve ser desenvolvido utilizando uma arquitetura em camadas, de modo a facilitar a manutenção do mesmo. A princípio podemos utilizar o modelo MVC(Model View Controller).

*Prioridade: M*

### 5.2. Usabilidade

#### 5.2.1. RNF-2: Padrão de interfaces

O sistema deve ser desenvolvido de modo a manter um padrão na interface com o usuário, de modo a facilitar a familiarização do usuário com as telas do sistema. O sistema deverá funcionar também de maneira responsiva, de modo a ser fácil de se utilizar em dispositivos móveis quanto em computadores.

*Prioridade: A*

### 5.3. Segurança

#### 5.3.1. RNF-3: Banco de dados

O sistema deve proteger os dados sensíveis dos clientes e usuários através de criptografia de senhas no banco de dados, acesso restrito ao banco de dados, o mesmo não deverá ter acesso externo a internet.

*Prioridade: A*

#### 5.3.2. RNF-4: Autenticação/Autorização

O sistema deverá ter mecanismos de autenticação e autorização para somente determinados níveis acessar informações mais sensíveis.

*Prioridade: A*

### 5.4. Interoperabilidade

#### 5.4.1. RNF-5: Comunicação com sistema de pagamento

O sistema deve ser capaz de se comunicar/integrar com pelo menos um meio de pagamento externo como pagseguro através de uma API.

*Prioridade: M*

## **5.5. Desempenho**

### **5.5.1. RNF-6: Acesso após Autenticação**

O sistema deve ser capaz de exibir sua tela inicial após a inserção dos dados de acesso em menos de 5 segundos.

***Prioridade: M***

## 6. Mecanismos Arquiteturais

Abaixo listamos os mecanismos que irão compor a arquitetura do software proposto.

Table 6.1: Módulo Cadastros

Análise	Design	Implementação
Mapeamento objeto-relacional	Utilização do padrão de projeto Data Access Object (DAO)	Utilização do Hibernate como framework ORM
Gerenciamento de dados estruturados	Modelagem de banco de dados relacional	Utilização do PostgreSQL como SGBD
Autenticação de usuários	Utilização de algoritmo de criptografia unidirecional para armazenamento de senhas, como SHA-256 ou bcrypt, e implementação de um mecanismo de validação de login e senha no servidor.	Utilização de um framework de autenticação, como Spring Security
Integração entre diferentes sistemas	Utilização de protocolos e formatos de dados interoperáveis, como JSON	Utilização de APIs RESTful para comunicação entre os sistemas.
Registro de atividades do sistema (log)	Utilização de frameworks de logging	Log4j
Gerenciamento de versões de código	Utilização de um sistema de controle de versão	GitHub
Documentação das APIs	Utilização ferramentas de documentação	Swagger
Teste do Software	Utilização de testes unitários, trabalhar com a metodologia TDD	JUnit
Front-end	Interface de comunicação com o usuário do sistema	Angular
Back-end	Utilização de arquitetura MVC e injeção de dependências	Spring framework

## 7. Modelagem Arquitetural

Apresentamos abaixo a modelagem arquitetural do sistema de delivery, utilizaremos para nossa modelagem o modelo C4.

### 7.1. Diagrama de Contexto

Começaremos pelo diagrama de contexto, nele temos o nosso sistema de delivery Pop Food que é responsável por gerenciar todo o estabelecimento, e os atores como clientes e atendentes/administradores que fazem pedido de comida e são responsáveis pela administração dentro do software respectivamente. Podemos verificar ainda os sistemas externos como PagSeguro e Ifood para pagamento e pedidos online respectivamente.

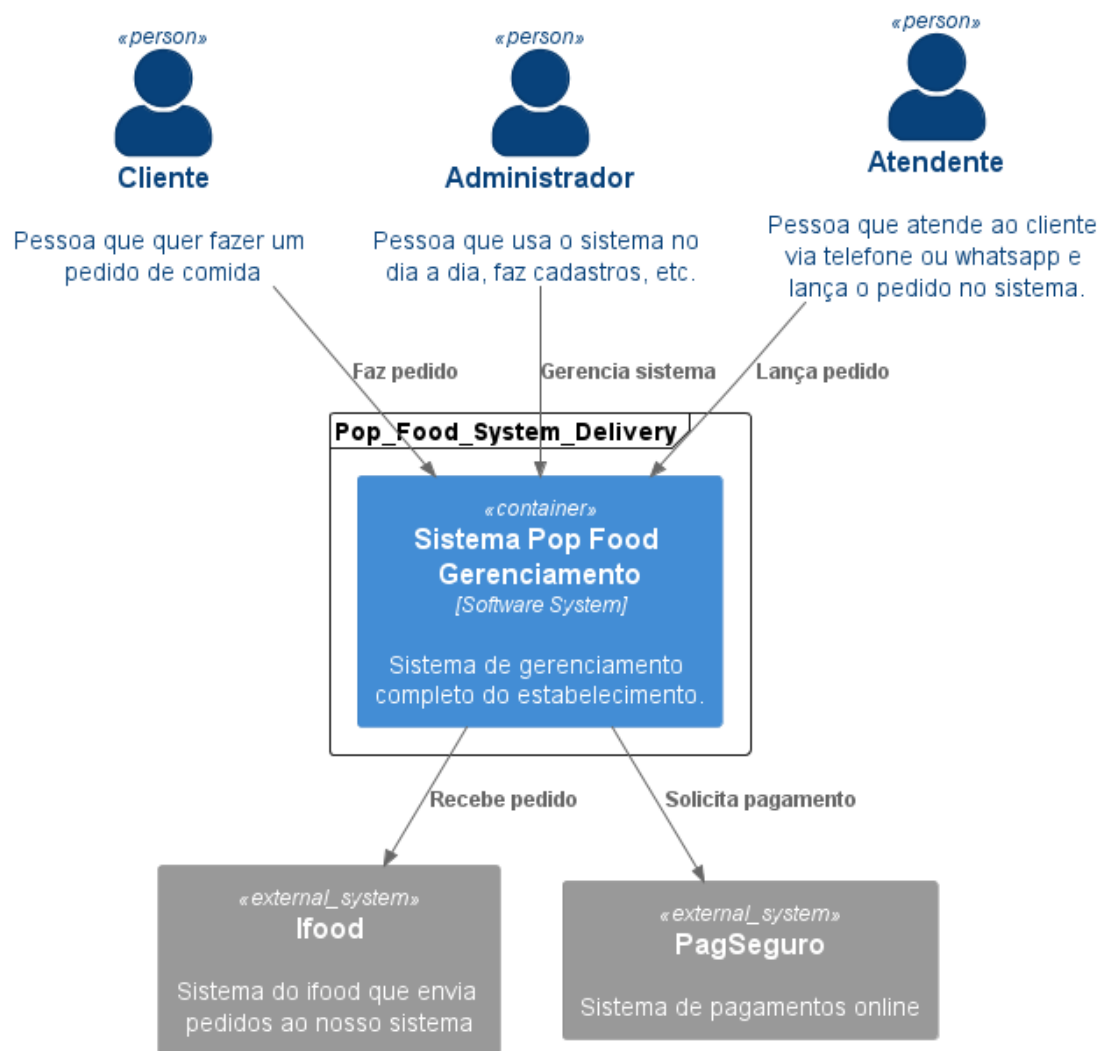


Figure 7.1: Diagrama de Contexto

Imagem resolução maior

## 7.2. Diagrama de Container

Mostramos agora o diagrama de container, nele podemos observar que teremos dois containers de frontend, ambos utilizando a tecnologia Angular, o frontend do Cliente é responsável por apresentar as telas do sistema que os clientes utilizarão para se cadastrar, ver o menu de opções, efetuar pedidos e pagamentos online.

No frontend de administração será disponibilizado as telas de login e administração do sistema, como cadastro de produtos, possibilidade de lançamento de pedidos, cadastro de clientes, etc.

Neste diagrama vemos o container do back end, que utilizará spring boot e será responsável pelas regras de negócio e integração com banco de dados e sistemas externos como iFood e PagSeguro.

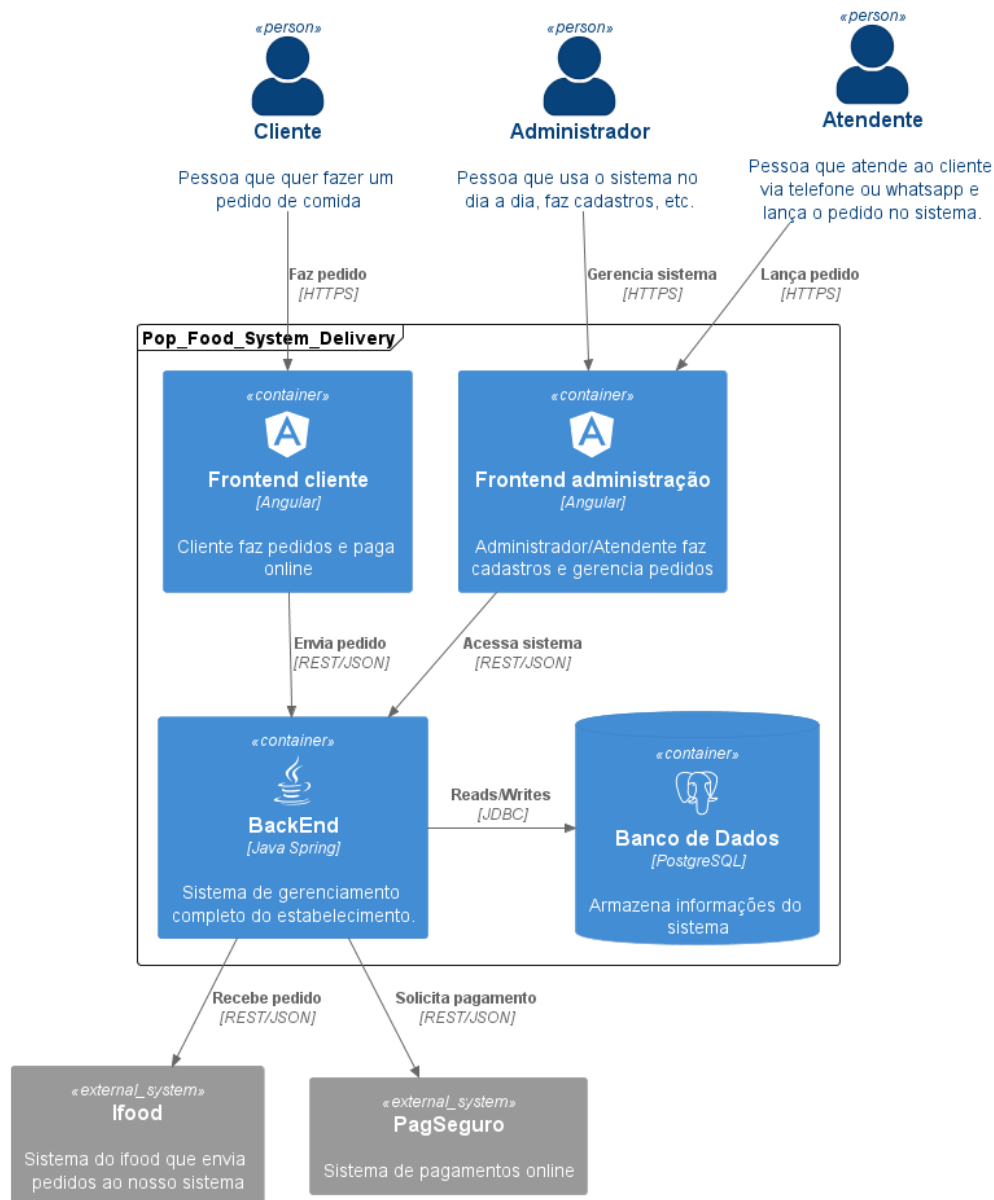


Figure 7.2: Diagrama de Container

Imagem resolução maior

### 7.3. Diagrama de Componentes

Entrando mais em detalhes, temos agora o diagrama de componentes, nele podemos ver as abstrações em caixa preta dos principais componentes do sistema.

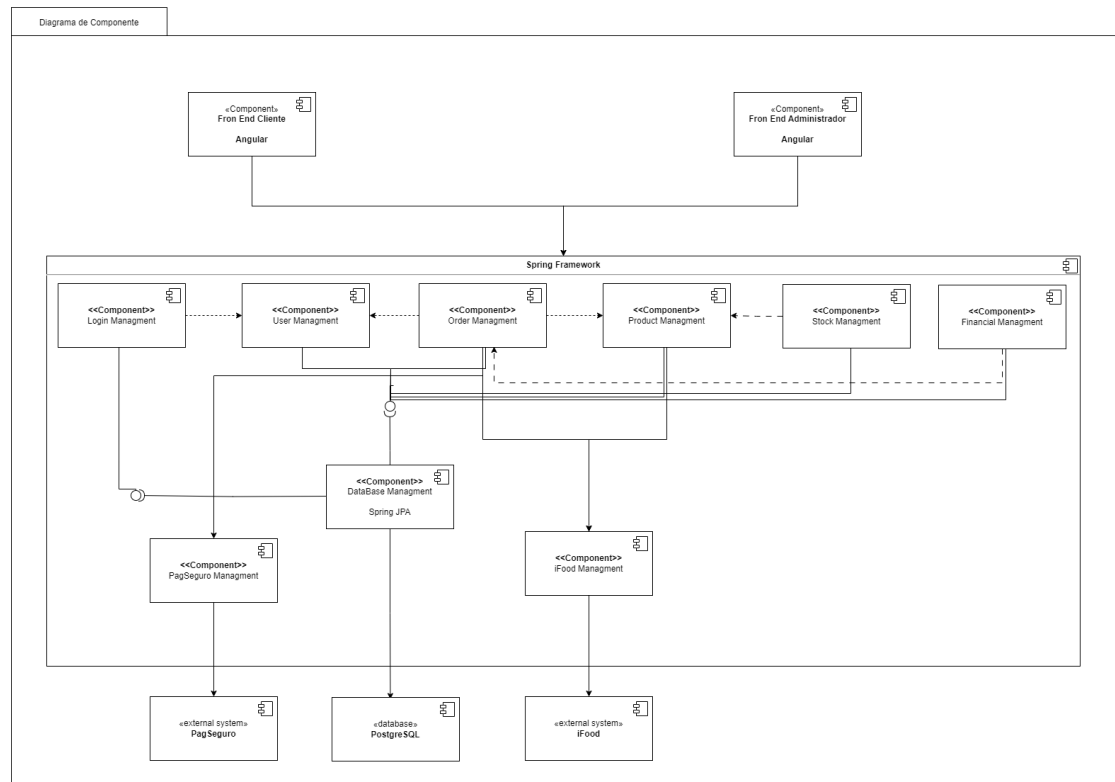


Figure 7.3: Diagrama de Componentes

Imagem resolução maior



## **8. Vídeo Apresentação 1**

Vídeo Apresentação 1

## Bibliography

- [1] Abrasel. Impacto da covid no setor de alimentação. <https://www.abrasel.com.br/noticias/5242-nota-abrasel-sobre-impacto-do-covid-19-no-setor-de-alimentacao-fora-do-lab.html>, 2023. [Acessado em 23 de janeiro de 2023].
- [2] FGV. O 'boom' das plataformas de delivery no brasil e suas consequências peculiares. <https://portal.fgv.br/artigos/boom-plataformas-delivery-brasil-e-suas-consequencias-peculiares>, 2023. [Acessado em 23 de janeiro de 2023].
- [3] Redação IFB. A transformação digital em bares e restaurantes. <https://foodbizbrasil.com/negocios/a-transformacao-digital-em-bares-e-restaurantes/>, 2023. [Acessado em 05 de fevereiro de 2023].
- [4] Lucas Muribeca. Delivery de alimentos deve aumentar 7,5 <https://redepara.com.br/Noticia/231697/delivery-de-alimentos-deve-aumentar-7-5-em-2023-no-brasil#:~:text=De%20acordo%20com%20o%20Instituto,restaurante%20Engenho%20Ded%C3%A9%2C%20em%20Bel%C3%A9m>, 2023. [Acessado em 05 de fevereiro de 2023].
- [5] Saipos. Quem paga a taxa ifood? <https://saipos.com/ifood/taxa-ifood>, 2023. [Acessado em 23 de janeiro de 2023].
- [6] Ian Sommerville. *Engenharia de Software*. Pearson Prentice Hall, São Paulo, 9. ed. edition, 2011.