

## Curso Superior de Desenvolvimento de Software Multiplataforma

Alinne Martins Cardoso, 3011392323003

Bruno Henrique Pasqual, 3011392323012

Maycon Sanches B. de Moura, 3011392323013

Pietro Adrian R. Da Silva, 3011392323005

Rafael Martins Cardoso, 3011392323010

Projeto Interdisciplinar

Engenharia de Software II

Desenvolvimento Web II

Banco de Dados Relacional

Técnicas de Programação I

FoodClub

Orientadores: Prof<sup>a</sup> Cristiane Palomar Mercado

Prof<sup>a</sup> Maria Janaína da Silva Ferreira

Prof<sup>o</sup> Jones Artur Gonçalves

Prof<sup>o</sup> Wladimir Zuanazzi

Prof<sup>o</sup> Rodrigo De Paula Diver

Votorantim  
Junho, 2024

## RESUMO

O projeto visa criar uma plataforma que facilite a comunicação entre restaurantes, pequenas e médias empresas. A ideia é permitir que os donos das empresas e seus funcionários escolham facilmente onde desejam pedir o almoço. A plataforma propõe uma “lista de chamada” para que os funcionários possam selecionar suas opções diretamente do menu do restaurante escolhido. Além disso, as empresas terão a opção de fechar prazos maiores com o mesmo restaurante, garantindo preços melhores.

A plataforma busca auxiliar tanto os restaurantes quanto as empresas. Para os restaurantes, oferece visibilidade e a oportunidade de atrair mais clientes, disponibilizando seus produtos online. Já para as empresas, proporciona um gerenciamento fácil e prático do almoço dos funcionários, incluindo a troca de restaurantes e a disponibilidade de novos cardápios.

No que diz respeito às tecnologias empregadas, a plataforma utiliza Angular no frontend, proporcionando uma interface de usuário dinâmica e responsiva. Angular é conhecido por sua eficiência e capacidade de criar aplicativos de página única ricos em recursos. No backend, a plataforma é alimentada por Java, uma linguagem de programação robusta e versátil, juntamente com o MySQL Server para gerenciamento de banco de dados. Essa combinação permite um processamento de dados eficiente e seguro. A escolha dessas tecnologias garante que a plataforma seja escalável, mantendo a performance e a segurança. Com essa infraestrutura tecnológica, a plataforma está bem equipada para atender às necessidades tanto dos restaurantes quanto das empresas, facilitando a comunicação e proporcionando uma experiência de usuário suave e intuitiva.

## SUMÁRIO

1. DESCRIÇÃO DO PROJETO .....	1
1.1. Proposta do Software.....	1
1.2. Justificativa.....	1
1.3. Mapa do Site .....	2
1.4. Logomarca .....	2
2. REQUISITOS DO PROJETO .....	3
2.1. Levantamento de Requisitos.....	3
2.2. Requisitos Funcionais .....	5
2.3. Descrição do Caso de Uso .....	7
2.4. Requisitos Não Funcionais .....	12
3. PROJETO DO SOFTWARE .....	13
3.1. Tecnologias Utilizadas .....	13
3.2. Modelo de dados .....	16
3.2.1 Modelo Conceitual .....	16
3.2.2 Modelo Lógico .....	17
3.3. Diagrama de Classe.....	18
3.4. Diagrama de Atividades .....	19
3.5. Diagrama de Sequência .....	20
3.6. Interfaces com o usuário .....	21
4. ESTRATÉGIA DE TESTES .....	26
5. IMPLANTAÇÃO.....	26
6. REFERÊNCIAS .....	27

## LISTA DE FIGURAS

Figura 1 - Mapa do Site .....	2
Figura 2 - Logomarca .....	2
Figura 3 - Casos de Uso .....	6
Figura 4 - Modelo Conceitual .....	16
Figura 5 - Modelo Lógico .....	17
Figura 6 - Diagrama de Classe .....	18
Figura 7 - Diagrama de Atividades .....	19
Figura 8 - Diagrama de Sequência .....	20

## LISTA DE QUADROS

Tabela 1 - Requisitos Funcionais .....	5
Tabela 2 - Caso de Uso 1 .....	7
Tabela 3 - Caso de Uso 2 .....	7
Tabela 4 - Caso de Uso 3 .....	8
Tabela 5 - Caso de Uso 4 .....	8
Tabela 6 - Caso de Uso 5 .....	9
Tabela 7 - Caso de Uso 6 .....	9
Tabela 8 - Caso de Uso 7 .....	10
Tabela 9 - Caso de Uso 8 .....	10
Tabela 10 - Caso de Uso 9 .....	11
Tabela 11 - Caso de Uso 10 .....	11
Tabela 12 - Requisitos não funcionais .....	12



## **1. DESCRIÇÃO DO PROJETO**

### **1.1. Proposta do Software**

O presente projeto visa criar uma plataforma onde restaurantes, pequenas e médias empresas se comuniquem facilmente para que os donos das empresas e seus funcionários tenham uma maneira simples de escolher o local de onde desejam pedir seu almoço. Com a proposta de criar uma “lista de chamada” para os funcionários escolherem suas opções do dia direto do menu do restaurante escolhido, as empresas terão mais de uma opção de onde poderão escolher o menu assim como também poderão fechar prazos maiores com o mesmo restaurante e, desta forma, garantir preços melhores. Além disso, será possível auxiliar os restaurantes que buscam angariar uma quantidade maior de clientes, disponibilizando seus produtos de forma online.

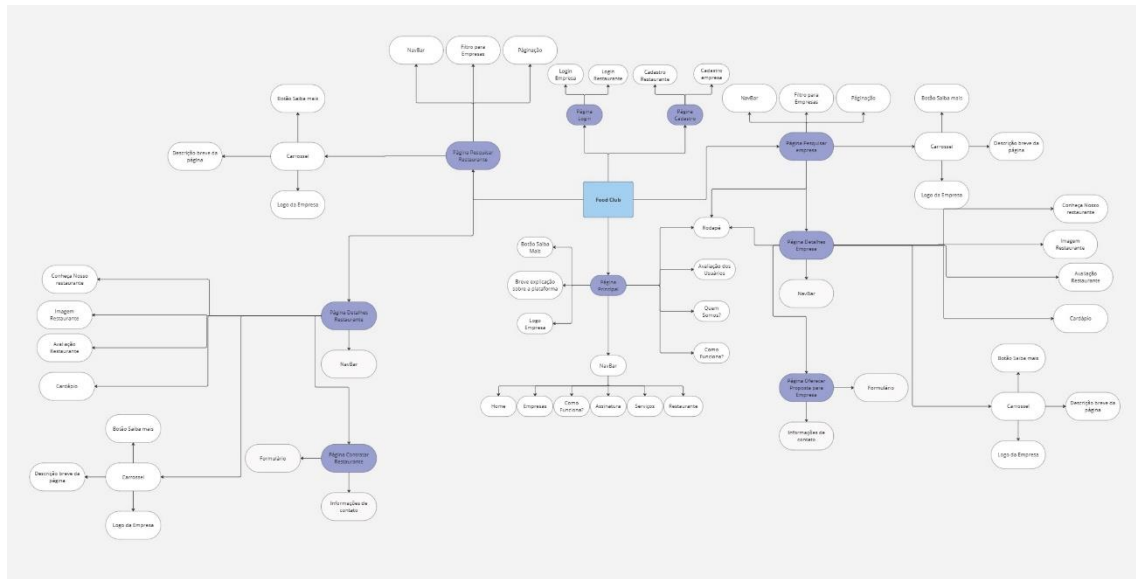
Será criada uma plataforma que ofereça a facilidade de outros aplicativos como iFood e Zé Delivery para um ambiente empresarial, onde a empresa poderá escolher o restaurante que irá atendê-la e o menu deste restaurante escolhido será disponibilizado para os seus funcionários cadastrados. E, por fim, poderão realizar suas escolhas direto de seus celulares, que serão reunidas em um único pedido da empresa e enviado ao restaurante. Com isso, busca-se um fácil gerenciamento de escolhas de restaurantes para as empresas.

### **1.2. Justificativa**

O projeto visa conectar mais restaurantes com micro e médias empresas, dessa forma é possível aumentar a visibilidade dos restaurantes, integrando-os a uma plataforma online. Já para as empresas, o objetivo do projeto é disponibilizar uma forma fácil e prática de gerenciar o controle do almoço dos seus funcionários, a troca de restaurantes e a disponibilidade de novos cardápios para eles.

### 1.3. Mapa do Site

Figura 1 - Mapa do Site



Fonte: Autoria Própria

### 1.4. Logomarca

Figura 2 - Logomarca



Fonte: Autoria Própria



## **2. REQUISITOS DO PROJETO**

### **2.1. Levantamento de Requisitos**

Para o levantamento dos requisitos, recorreremos à observação direta de um dos membros da nossa equipe. Ele enfrenta dificuldades diárias na empresa onde trabalha para fazer o pedido de marmitex para o almoço. Atualmente, o processo é realizado pelo WhatsApp, onde o restaurante envia o cardápio do dia. No entanto, este processo manual tem gerado diversos problemas.

Um dos problemas é que, muitas vezes, o funcionário está em reunião e não pode fazer o pedido naquele momento. Além disso, o restaurante precisa aguardar o fechamento do pedido, o que pode levar algum tempo. Outra dificuldade surge quando alguém da equipe precisa percorrer a empresa perguntando se o funcionário foi trabalhar ou se está fazendo alguma visita externa.

Esses atrasos na comunicação resultam em um grande atraso para a realização do pedido das marmitas. Portanto, é essencial buscar uma solução que otimize esse processo e minimize os contratempos enfrentados atualmente.

A seguir analisamos duas ferramentas para ajudar no levantamento de requisitos:

O iFood é um serviço de delivery online que permite aos usuários fazerem pedidos de refeições diretamente de seus smartphones, sem a necessidade de sair de casa. Para utilizá-lo, basta baixar o aplicativo gratuito disponível para iOS e Android. Após o download, os usuários podem buscar restaurantes por tipo de comida ou localização. Ao selecionar um restaurante, o menu completo é exibido, permitindo que o usuário escolha os itens desejados.

Para os restaurantes, o iFood oferece a possibilidade de cadastro na plataforma. Os requisitos mínimos incluem possuir um CNPJ, entregar refeições prontas para consumo e ter acesso a um computador com conexão à internet. O processo de cadastro é realizado online, e os restaurantes pagam uma porcentagem sobre o valor das vendas realizadas através do aplicativo.

Em resumo, o iFood<sup>1</sup> é uma plataforma que conecta restaurantes e clientes, facilitando o processo de pedidos de delivery. Seja como usuário ou parceiro, o aplicativo oferece uma experiência simples e dinâmica para todos os envolvidos.

O Zé Delivery<sup>2</sup> é um aplicativo de entregas desenvolvido pela Ambev para garantir que nunca falte bebida gelada. Ele foca em entregar bebidas (alcoólicas e não alcoólicas) de forma rápida e conveniente. Veja como funciona:

O usuário baixa o aplicativo no Android ou iOS. Ao fazer um pedido, o Zé Delivery direciona para a loja mais próxima do solicitante. A entrega é feita em até uma hora, e os preços são equivalentes aos dos supermercados ou até mais econômicos. O aplicativo oferece uma variedade de bebidas, incluindo cervejas, vinhos, água, energéticos, refrigerantes, salgadinhos e até gelo. O Zé Delivery atende diversas cidades no Brasil, incluindo São Paulo, Rio de Janeiro, Belo Horizonte, Brasília, Recife e muitas outras.

Em resumo, o Zé Delivery facilita o pedido de bebidas geladas, garantindo que todos possam realizar seu pedido onde e quando quiserem.

---

<sup>1</sup> conforme disponível em: **iFood**. Disponível em: <<https://www.ifood.com.br/?toHome=true>>. Acesso em: 17 maio. 2024.

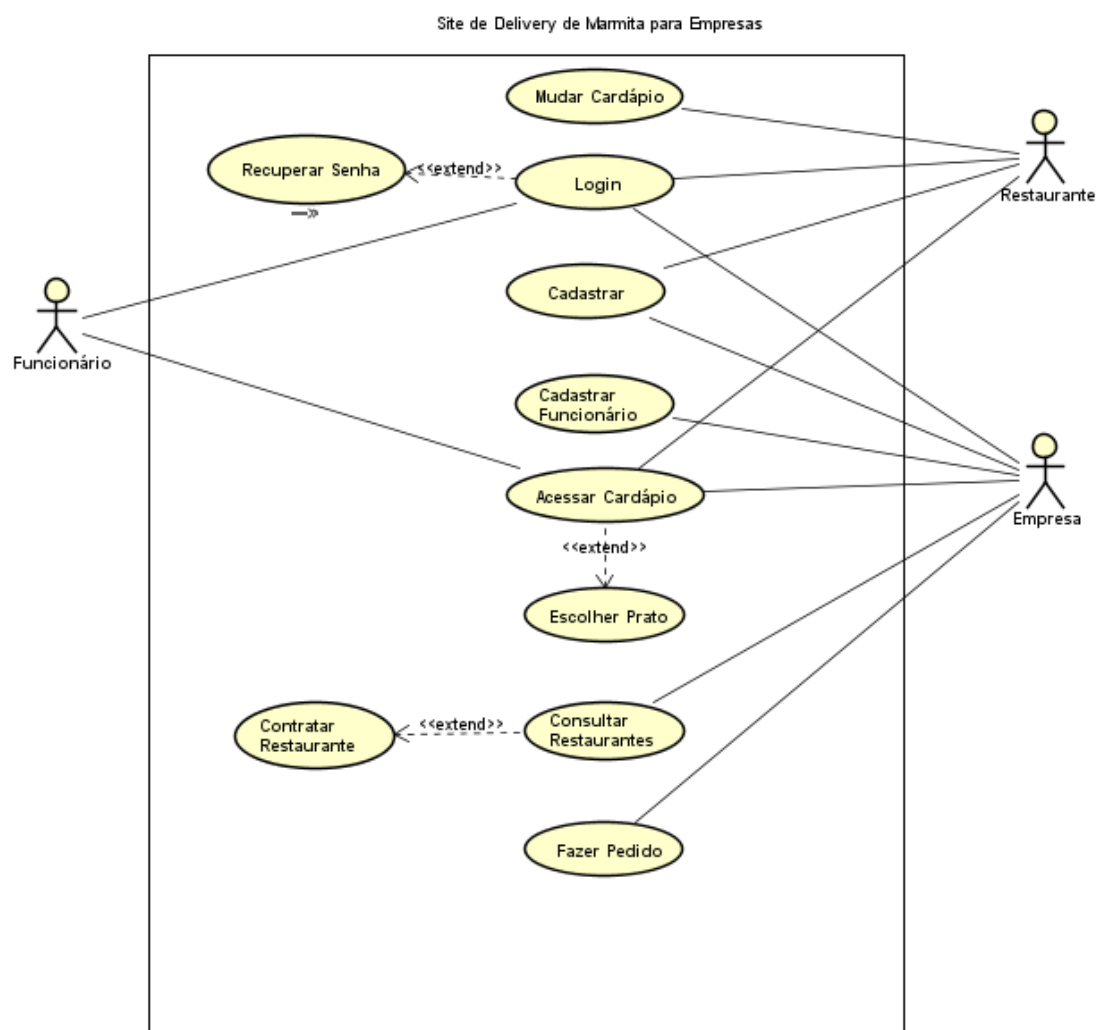
<sup>2</sup> conforme disponível em: **Zé Delivery: Entrega de bebidas geladas a preço baixo**. Disponível em: <<https://www.ze.delivery/>>. Acesso em: 17 maio. 2024.

## 2.2. Requisitos Funcionais

Tabela 1 - Requisitos Funcionais

	Nome do Requisito	Descrição	Página Associada
RF1	Página de Cadastro	Os usuários devem poder acessar a página de cadastro para criar uma conta	Página Inicial
RF2	Formulário de Cadastro	A página de cadastro deve conter um formulário para inserir informações pessoais	Página de Cadastro
RF3	Recuperação de Senha	Os usuários devem ter a opção de recuperar a senha esquecida	Página de Login
RF4	Navegação no Site	Os usuários podem navegar facilmente entre as páginas do site	Todas as Páginas
RF5	Acesso à Página Inicial	Os usuários logados podem acessar a página inicial do site	Página Inicial
RF6	Cadastrar Cardápio	O Restaurante deverá cadastrar o cardápio para a Empresa e os Funcionários visualizarem	Página do Restaurante
RF7	Exibição do Cardápio	O Restaurante deverá disponibilizar o cardápio para a Empresa visualizar	Página do Restaurante
RF8	Exibição do Cardápio	O Funcionário poderá visualizar os pratos do cardápio que a Empresa escolheu	Página de escolha dos pratos
RF9	Busca de Restaurantes	Deve haver uma funcionalidade de busca para facilitar na localização do Restaurante desejado	Página dos restaurantes
RF10	Busca de Empresas	Deve haver uma funcionalidade de busca para facilitar na localização da Empresa desejada	Página das empresas
RF11	Adicionar Avaliações	As Empresas registradas podem adicionar avaliações aos Restaurantes	Página do restaurante
RF12	Finalização da Compra	Deve haver uma opção para as Empresas finalizarem o pedido	Página de fazer pedido
RF13	Escolha dos pratos	Deve haver uma opção para os Funcionários escolherem os pratos	Página de escolha dos pratos
RF14	Exibição dos pratos	Deve haver uma opção para os Funcionários visualizarem os pratos	Página de escolha dos pratos

Figura 3 - Casos de Uso



Fonte: Autoria Própria

## 2.3. Descrição do Caso de Uso

Tabela 2 - Caso de Uso 1

<b>Caso de uso</b>	Mudar o cardápio
<b>Ator principal</b>	Restaurante
<b>Ator secundário</b>	Sistema
<b>Pré-condições</b>	O restaurante já deve ter sido registrado e deverá ter realizado o login
<b>Pós-condições</b>	O cardápio do restaurante será alterado ou criado
<b>Fluxo Principal</b>	
<b>Ações do ator</b>	<b>Ações do sistema</b>
<b>Alterar/Cria menu</b>	
Acessa a opção de alterar o cardápio	
	Mostra a tela de gerenciamento de cardápio
Realiza as alterações do cardápio	
	Valida as informações fornecidas e dá os retornos apropriados
	Salva as alterações

Tabela 3 - Caso de Uso 2

<b>Caso de uso</b>	Login
<b>Ator principal</b>	Restaurante, Empresa e Funcionário
<b>Ator secundário</b>	Sistema
<b>Pré-condições</b>	Os usuários já devem ter sido registrados, e devem estar deslogado
<b>Pós-condições</b>	O usuário irá logar com sua conta
<b>Fluxo Principal</b>	
<b>Ações do ator</b>	<b>Ações do sistema</b>
<b>Login (restaurante)</b>	
Entrar na página inicial	
Acessar a opção de login	
Caso seja uma Empresa, clicar na opção empresa, caso seja um Restaurante, clicar na opção Restaurante, caso seja um Funcionário, clicar na opção Funcionário	
	Mostrar os campos para realizar o login (usuário e senha)
Preenche as informações de login e pressiona "logar"	
	Valida as informações, caso esteja correto entra na aplicação com o usuário passado, caso contrário exibe a mensagem de erro e mostra as opções adequadas (recuperar senha, digite novamente)
Entra na aplicação na tela inicial do usuário como restaurante	

Tabela 4 - Caso de Uso 3

<b>Caso de uso</b>	Recuperar senha
<b>Ator principal</b>	Empresa, Restaurante e Funcionário
<b>Ator secundário</b>	Sistema
<b>Pré-condições</b>	Estar cadastrado
<b>Pós-condições</b>	Fazer login
<b>Fluxo Principal</b>	
<b>Ações do ator</b>	<b>Ações do Sistema</b>
<b>Recuperar senha</b>	
Entra na tela de Login	
Seleciona recuperar senha	
	Abre a tela de recuperação de senha
Coloca e-mail cadastrado e envia	
	Envia código para o número de celular cadastrado nesse e-mail
Insere código	
	Envia nova senha para o e-mail cadastrado
Loga no sistema com a nova senha gerada	

Tabela 5 - Caso de Uso 4

<b>Caso de uso</b>	Cadastrar
<b>Ator principal</b>	Empresa, Restaurante
<b>Ator secundário</b>	Sistema
<b>Pré-condições</b>	O usuário não ser já cadastrado no sistema
<b>Pós-condições</b>	Irá criar um cadastro no sistema
<b>Fluxo Principal</b>	
<b>Ações do ator</b>	<b>Ações do sistema</b>
<b>Cadastrar restaurante</b>	
Acessar a tela inicial	
	Sistema irá exibir as opções de login ou cadastro
Acessar a tela de cadastro	
	Exibe a tela de cadastro exigindo algumas informações para realizar o cadastro
Usuário preenche as informações exigidas, aceita os termos de uso e pressiona o botão para realizar o cadastro	
	Realiza validações das informações enviadas
	Retorna o feedback para o usuário exibindo a mensagem de que o restaurante foi cadastrado com sucesso, caso contrário exibe uma mensagem de erro que mostra os campos que inválidos
	Os dados informados são salvos dentro do banco de dados

Tabela 6 - Caso de Uso 5

<b>Caso de uso</b>	Cadastrar Funcionário
<b>Ator principal</b>	Empresa
<b>Ator secundário</b>	Sistema
<b>Pré-condições</b>	O funcionário deve estar vinculado a empresa
<b>Pós-condições</b>	Escolher a refeição
<b>Fluxo Principal</b>	
<b>Ações do ator</b>	<b>Ações do Sistema</b>
<b>Cadastrar funcionário</b>	
Acessar a tela de cadastro de funcionários	
	Exibir tela de login das empresas caso não esteja logado
	Caso esteja logado, exibir janela com os campos detalhados das informações necessárias para o cadastro
Preencher os campos necessários, com informações que vinculam o funcionário a empresa cadastrada.	
	O sistema salva o funcionário e exibe uma janela informando que o funcionário foi cadastrado com sucesso, com botões de finalização ou continuar cadastrando.
Clicar em finalizar / continuar cadastrando	

Tabela 7 - Caso de Uso 6

<b>Caso de uso</b>	Fazer Pedido
<b>Ator principal</b>	empresa
<b>Ator secundário</b>	Sistema
<b>Pré-condições</b>	Ter os pedidos consolidado dos funcionários
<b>Pós-condições</b>	Acesso a tela de gerenciamento de pedidos
<b>Fluxo Principal</b>	
<b>Ações do ator</b>	<b>Ações do Sistema</b>
<b>Acessar Cardápio</b>	
Acessar a tela de pedidos	
	Exibir tela de login das empresas caso não esteja logado
	Caso esteja logado, exibir janela com os pedidos consolidado dos funcionários
Verifica os pedidos enviados pelos funcionários e confirma envio para o restaurante	
	Exibir uma janela informando que o pedido foi enviado

Tabela 8 - Caso de Uso 7

<b>Caso de uso</b>	Escolher prato
<b>Ator principal</b>	Funcionário
<b>Ator secundário</b>	Sistema
<b>Pré-condições</b>	O funcionário deve estar vinculado a empresa e ter usuário e senha para acesso.
<b>Pós-condições</b>	Acesso a página com o cardápio pré-selecionado pela empresa
<b>Fluxo Principal</b>	
<b>Ações do ator</b>	<b>Ações do Sistema</b>
<b>Acessar Cardápio</b>	
Acessar a tela do cardápio para os funcionários	
	Exibir tela de login das empresas caso não esteja logado
	Caso esteja logado, exibir janela com as opções de cardápio pré-definidas entre empresa e restaurante
Selecionar a refeição do dia seguinte e clicar em enviar pedido.	
	Exibir uma janela informando que o pedido foi enviado
Clicar em sair do cardápio, ou voltar para tela inicial	
	Voltar para tela inicial, ou encerrar seção.

Tabela 9 - Caso de Uso 8

<b>Caso de uso</b>	Consultar restaurantes
<b>Ator principal</b>	Qualquer pessoa
<b>Ator secundário</b>	Sistema
<b>Pré-condições</b>	Estar com o site ou aplicativo aberto
<b>Pós-condições</b>	
<b>Fluxo Principal</b>	
<b>Ações do ator</b>	<b>Ações do Sistema</b>
<b>Consultar restaurantes</b>	
Acessar a tela com os restaurantes cadastrados	
	Exibir tela de login dos restaurantes cadastrados
Selecionar filtros ou clicar em um restaurante	
	Abrir uma janela referente ao restaurante clicado, com informações relevantes ao tipo de usuário



Tabela 10 - Caso de Uso 9

<b>Caso de uso</b>	Contratar restaurante
<b>Ator principal</b>	Empresa
<b>Ator secundário</b>	Sistema//restaurante
<b>Pré-condições</b>	Estar logado, ter restaurantes cadastrados
<b>Pós-condições</b>	Efetuar pagamento
<b>Fluxo Principal</b>	
<b>Ações do ator</b>	<b>Ações do Sistema</b>
<b>Contratar restaurante</b>	
	Chama o caso de uso Consultar Restaurantes
Seleciona um restaurante	
	Exibe dados do restaurante selecionado
Avalia se o restaurante atende os requisitos da empresa	
Caso sim, clica na opção que concorda com os termos do contrato para realizar pedidos	
	Sistema confirma dados para contrato e envia ao restaurante

Tabela 11 - Caso de Uso 10

<b>Caso de uso</b>	Acessar Cardápio
<b>Ator principal</b>	Empresa, funcionário, restaurante
<b>Ator secundário</b>	Sistema
<b>Pré-condições</b>	Restaurante ter criado um cardápio
<b>Pós-condições</b>	
<b>Fluxo Principal</b>	
<b>Ações do ator</b>	<b>Ações do sistema</b>
<b>Acessar Cardápio</b>	
Acessar a tela dos restaurantes para visualizar os cardápios	
	Apresentar os cardápios disponibilizados pelos restaurantes

## 2.4. Requisitos Não Funcionais

Tabela 12 - Requisitos não funcionais

	Nome do Requisito	Descrição	Página Associada
RNF1	Escalabilidade	O sistema deve ser capaz de lidar com 100.000 usuários simultâneos.	Todas as Páginas
RNF2	Navegadores	Suporte aos principais navegadores, como Chrome, Firefox, Safari e Edge.	Todas as Páginas
RNF3	Dispositivos	Layout responsivo para diferentes tamanhos de tela (desktop, tablet, mobile).	Todas as Páginas
RNF4	Interface Intuitiva	A interface do usuário deve ser fácil de usar e entender.	Todas as Páginas
RNF5	Facilidade de Manutenção	O sistema deve ser modular e fácil de manter.	Todas as Páginas
RNF6	Atualizações sem Interrupções	Atualizações do sistema devem ser feitas sem interromper o serviço.	Todas as Páginas
RNF7	Proteção de Dados	Cumprir regulamentações de proteção de dados, como LGPD.	Todas as Páginas
RNF8	Tolerância a Falhas	O sistema deve ser capaz de lidar com falhas sem afetar gravemente o serviço.	Todas as Páginas
RNF9	Desempenho do Banco de Dados	O banco de dados deve ser otimizado para consultas rápidas e eficientes.	Todas as Páginas
RNF10	Manutenção e Atualização	Garantir que o sistema seja fácil de manter e atualizar.	Todas as Páginas
RNF11	Controle de Acesso	Garantir que apenas usuários autorizados tenham acesso a determinadas áreas.	Todas as Páginas
RNF12	Tempo de Resposta	Garantir que o tempo de resposta do servidor seja inferior a 200 ms.	Todas as Páginas
RNF13	Cache de Páginas	Implementar estratégias eficientes de caching para melhorar o desempenho.	Todas as Páginas

## 3. PROJETO DO SOFTWARE

### 3.1. Tecnologias Utilizadas

#### **HTML (Hypertext Markup Language):**

<sup>3</sup>HTML é a linguagem padrão para criação e estruturação de páginas web. Onde utilizamos para definir a estrutura e o conteúdo do nosso site, incluindo textos, imagens, vídeos e links.

#### **CSS (Cascading Style Sheets):**

<sup>4</sup>CSS é uma linguagem de estilo que permite definir o visual e o layout das páginas HTML. Onde utilizamos para estilizar e personalizar o design das páginas, garantindo uma experiência visual atraente e consistente.

#### **JavaScript:**

<sup>5</sup>JavaScript é uma linguagem de programação amplamente utilizada para adicionar interatividade e dinamismo às páginas web. Onde utilizamos para tornar a interface do usuário mais interativa e responsiva, adicionando funcionalidades específicas que melhoram a experiência do usuário.

#### **TypeScript:**

<sup>6</sup>TypeScript é um superset de JavaScript que adiciona recursos de tipagem estática à linguagem. Onde utilizamos para escrever código mais robusto e escalável, especialmente em aplicações front-end complexas.

---

<sup>3</sup> <https://html.spec.whatwg.org/>

<sup>4</sup> <https://www.w3.org/Style/CSS/>

<sup>5</sup> <https://tc39.es/ecma262/>

<sup>6</sup> <https://www.typescriptlang.org/>

**Angular:**

<sup>7</sup>Angular é um framework front-end desenvolvido pelo Google, que facilita a construção de aplicações web escaláveis e dinâmicas. Onde utilizamos para criar a camada de apresentação da aplicação web, implementando componentes reutilizáveis e interações complexas entre os elementos da interface.

**MySQL Server:**

<sup>8</sup>MySQL é um sistema de gerenciamento de banco de dados relacional amplamente utilizado em aplicações web. Onde utilizamos como o banco de dados principal para armazenar informações como cadastro de restaurantes, cardápios, usuários e pedidos.

**Java:**

<sup>9</sup>Java é uma linguagem de programação amplamente adotada para o desenvolvimento de aplicações empresariais robustas e escaláveis. Onde utilizamos para desenvolver a lógica de negócio da aplicação, incluindo o processamento de pedidos, a gestão de usuários e a integração com o banco de dados.

**Spring Boot:**

<sup>10</sup>Spring Boot é um framework Java que simplifica o desenvolvimento de aplicações web e serviços RESTful. Onde utilizamos para construir a camada de serviços RESTful, que será responsável por expor funcionalidades do sistema para o front-end Angular e outras aplicações cliente.

**Supabase:**

<sup>11</sup>Supabase é uma plataforma de desenvolvimento de aplicativos BaaS (Backend as a Service) de código aberto, que oferece banco de dados, autenticação e armazenamento de arquivos. Ele permite criar aplicativos web e

---

<sup>7</sup> <https://angular.io/>

<sup>8</sup> <https://dev.mysql.com/doc/>

<sup>9</sup> <https://docs.oracle.com/en/java/index.html>

<sup>10</sup> <https://docs.oracle.com/en/java/index.html>

<sup>11</sup> <https://supabase.com/>

móveis sem necessidade de escrever código, suporta várias formas de autenticação, incluindo social e de terceiros, e possibilita a modelagem de dados eficiente para projetos como aplicativos de blog. Além disso, o Supabase é acessível para desenvolvedores de todos os níveis e suas ferramentas são em sua maioria gratuitas.

### **ngx-toastr**

O <sup>12</sup>ngx-toastr é uma biblioteca npm amplamente utilizada em projetos Angular para exibir notificações de toasts (mensagens flutuantes) de forma elegante e não intrusiva. Com ele, os desenvolvedores podem informar os usuários sobre ações bem-sucedidas, erros ou outras mensagens importantes, melhorando a experiência do usuário em suas aplicações web.

### **ngx-mask**

O <sup>13</sup>ngx-mask é uma biblioteca para Angular que permite aos desenvolvedores formatar e validar facilmente dados de entrada em suas aplicações. Ele oferece um conjunto de diretivas e pipes que podem ser usados para aplicar máscaras a campos de entrada, garantindo que os dados inseridos pelos usuários estejam em um formato específico.

---

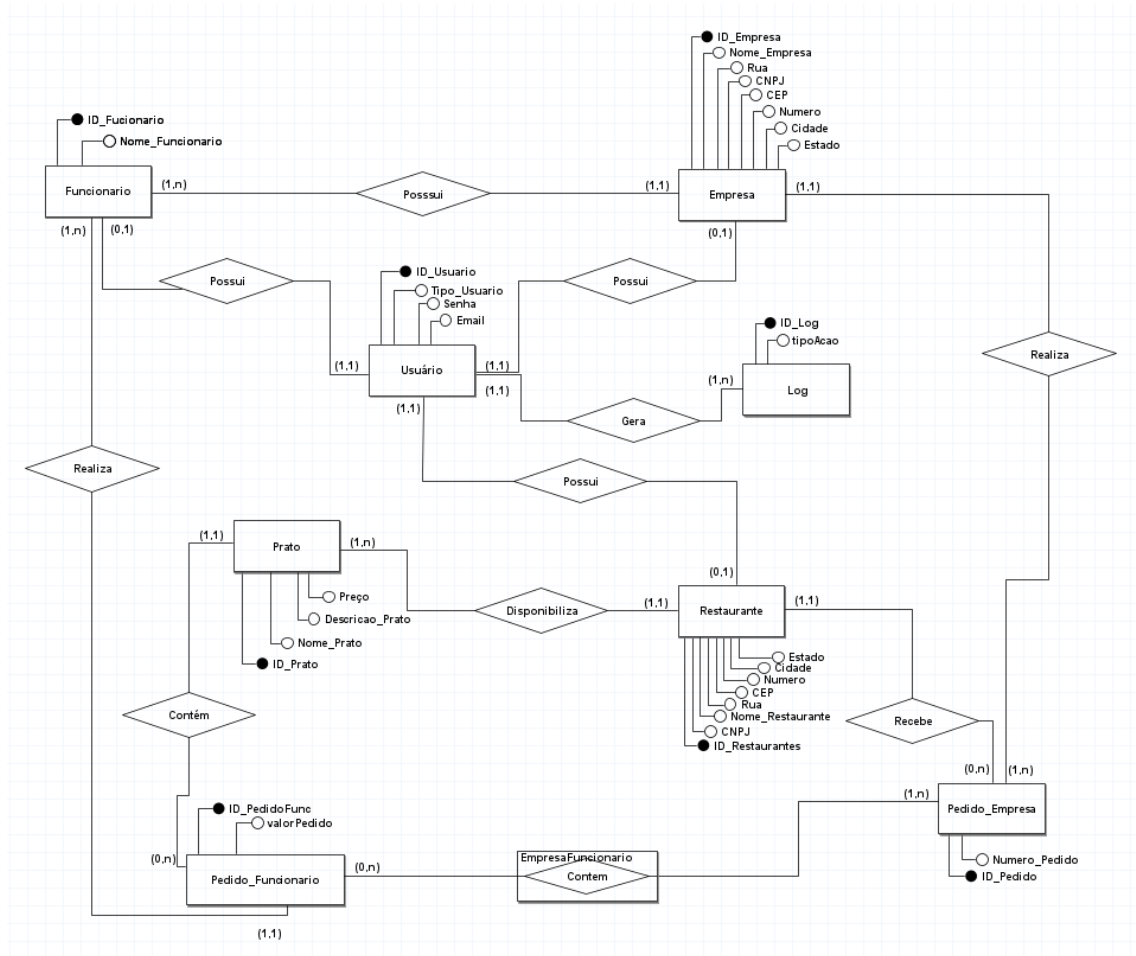
<sup>12</sup> <https://www.npmjs.com/package/ngx-toastr>

<sup>13</sup> <https://www.npmjs.com/package/ngx-mask>

## 3.2. Modelo de dados

### 3.2.1 Modelo Conceitual

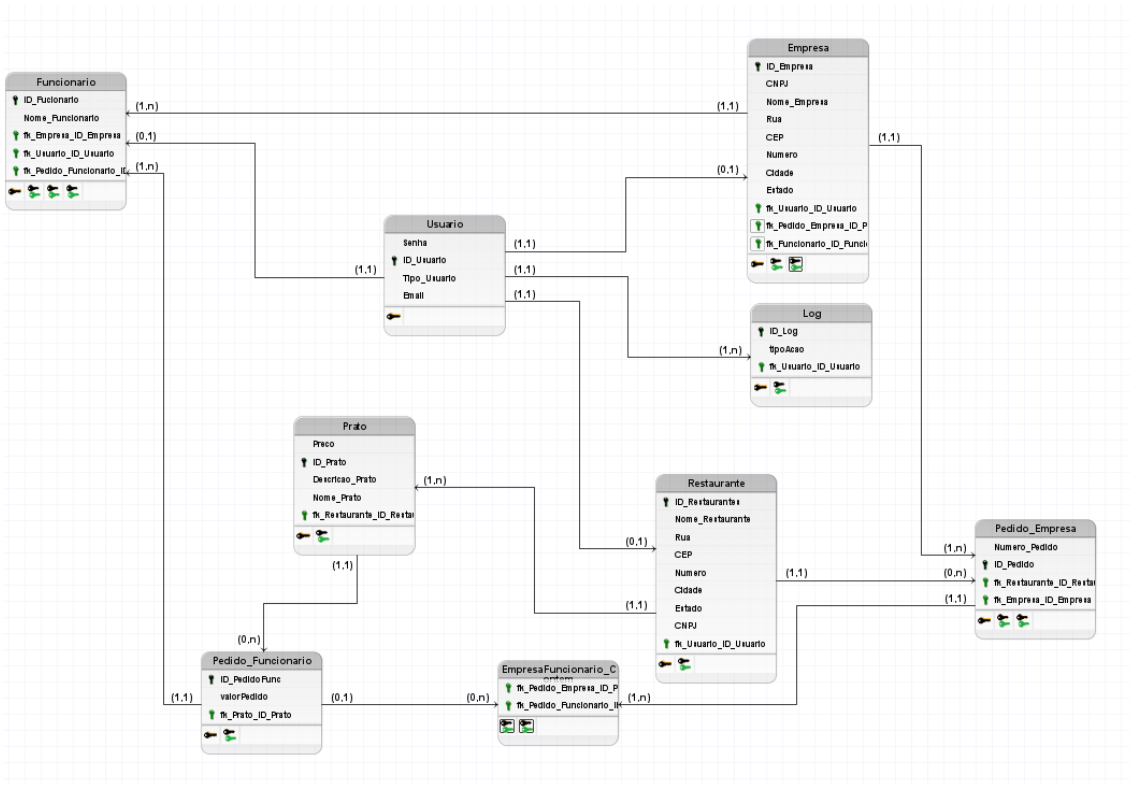
Figura 4 - Modelo Conceitual



Fonte: Autoria Própria

### 3.2.2 Modelo Lógico

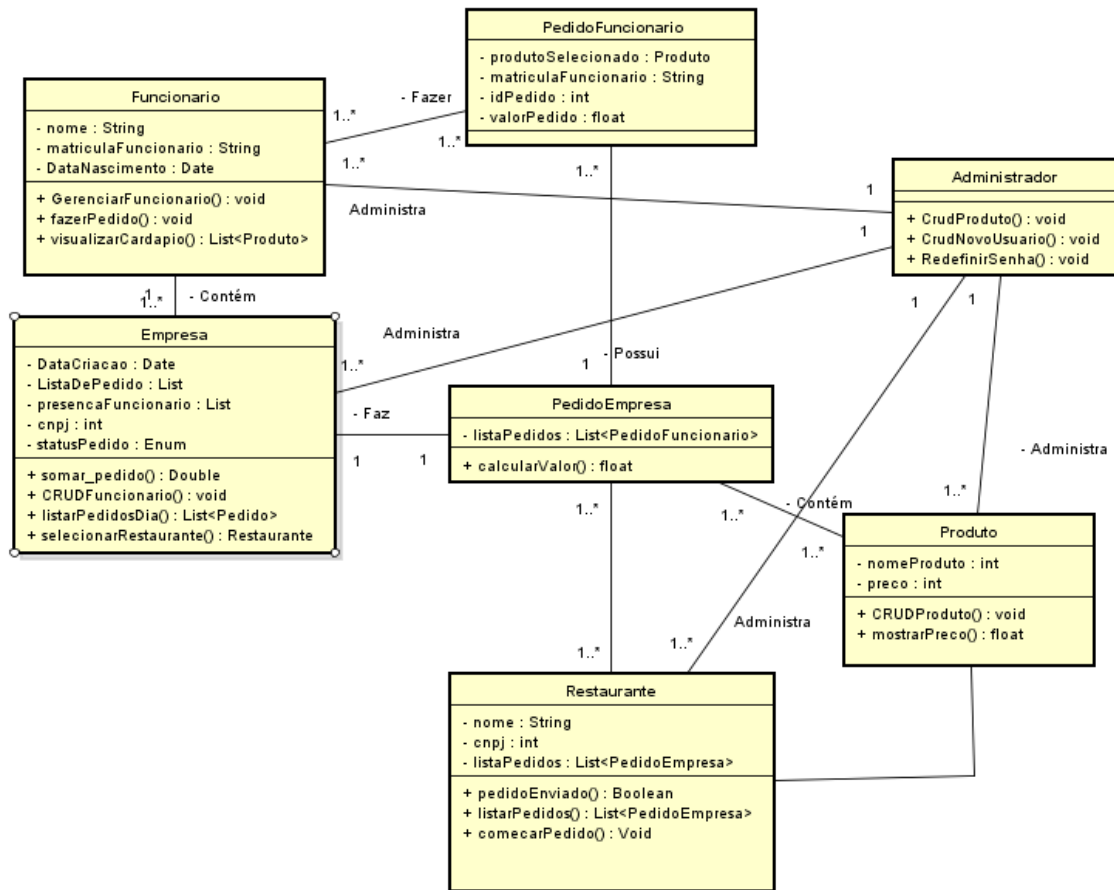
Figura 5 - Modelo Lógico



Fonte: Autoria Própria

### 3.3. Diagrama de Classe

Figura 6 - Diagrama de Classe

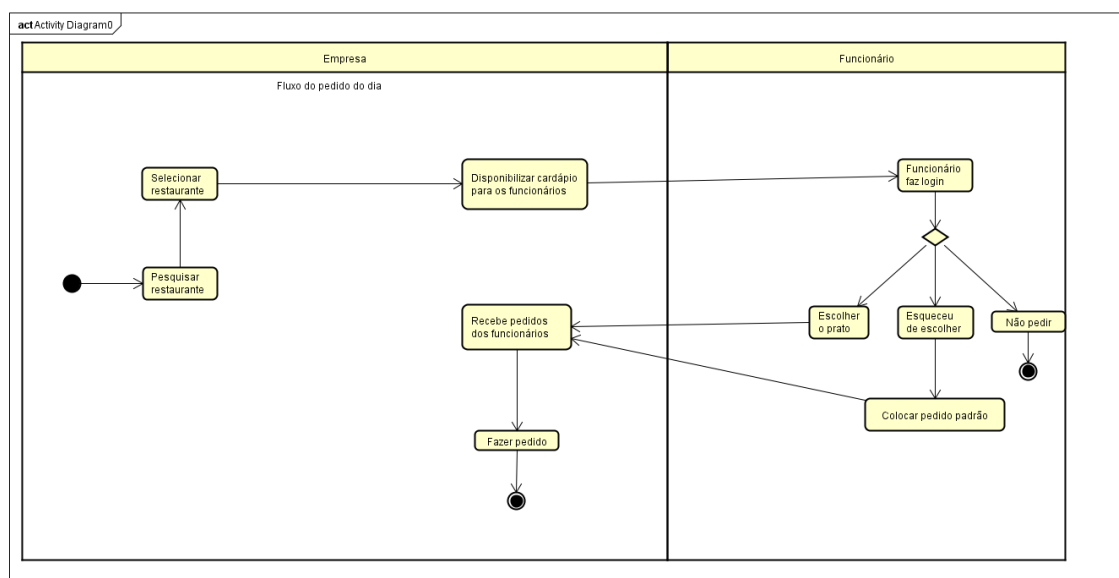
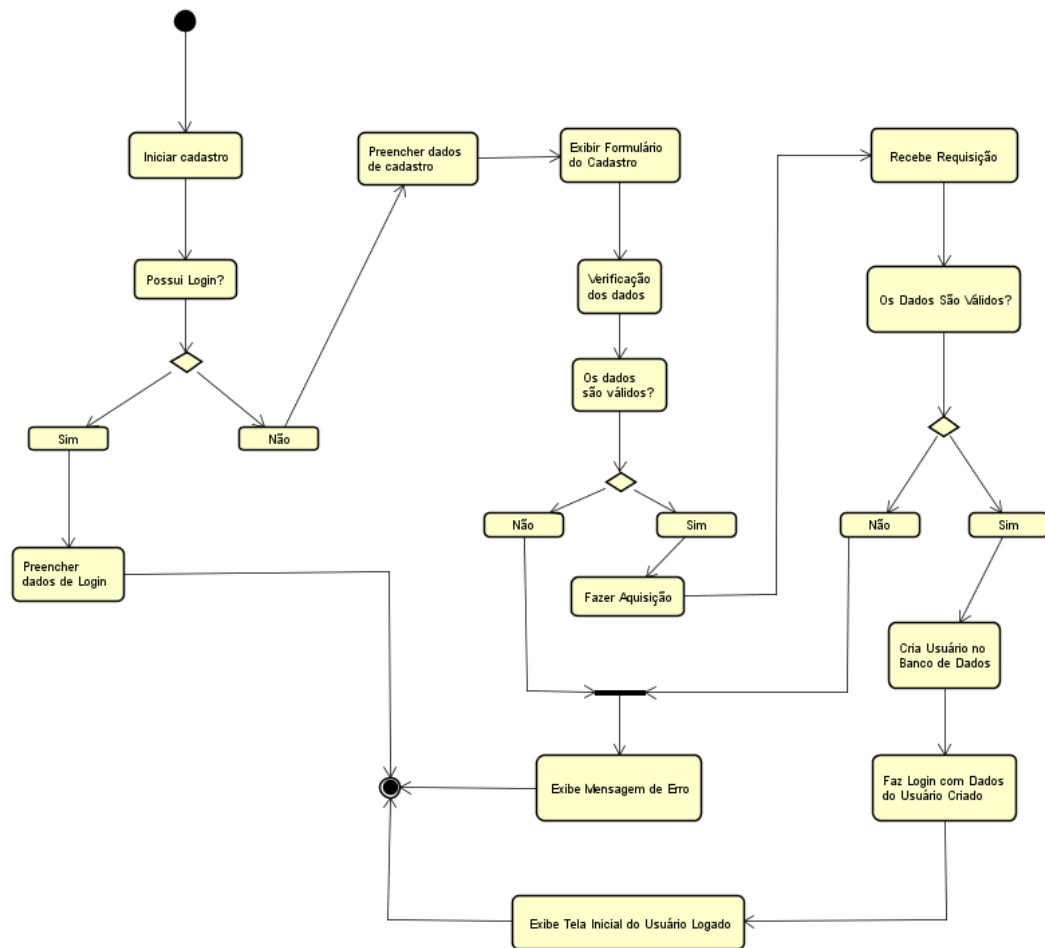


Fonte: Autoria Própria



### 3.4. Diagrama de Atividades

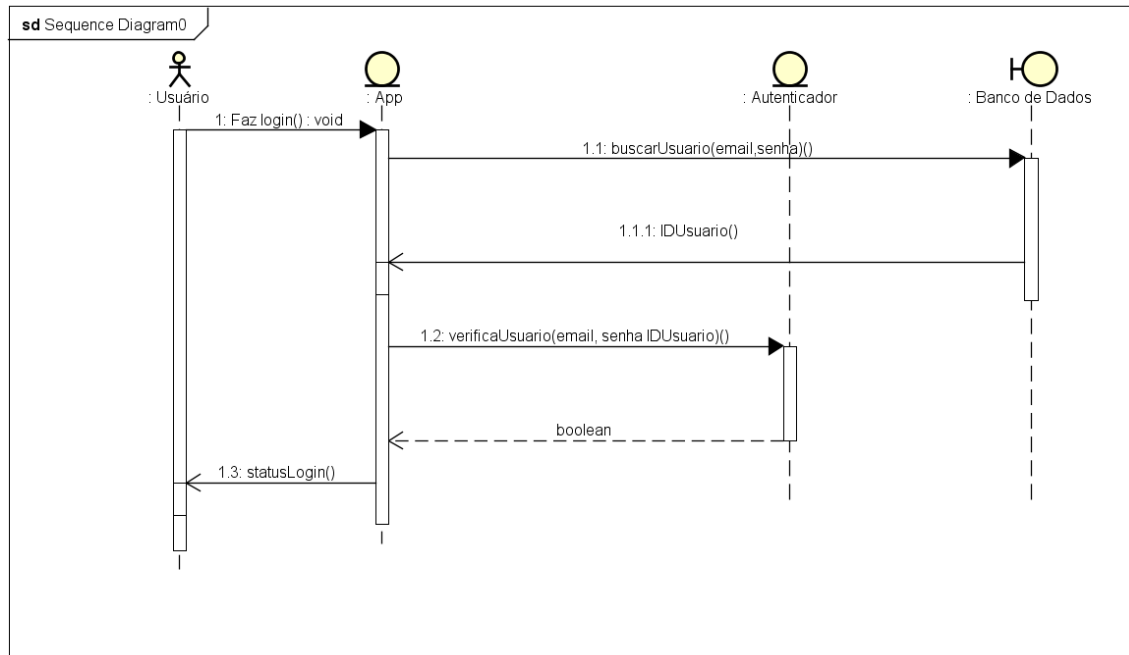
Figura 7 - Diagrama de Atividades



Fonte: Autoria Própria

### 3.5. Diagrama de Sequência

Figura 8 - Diagrama de Sequência

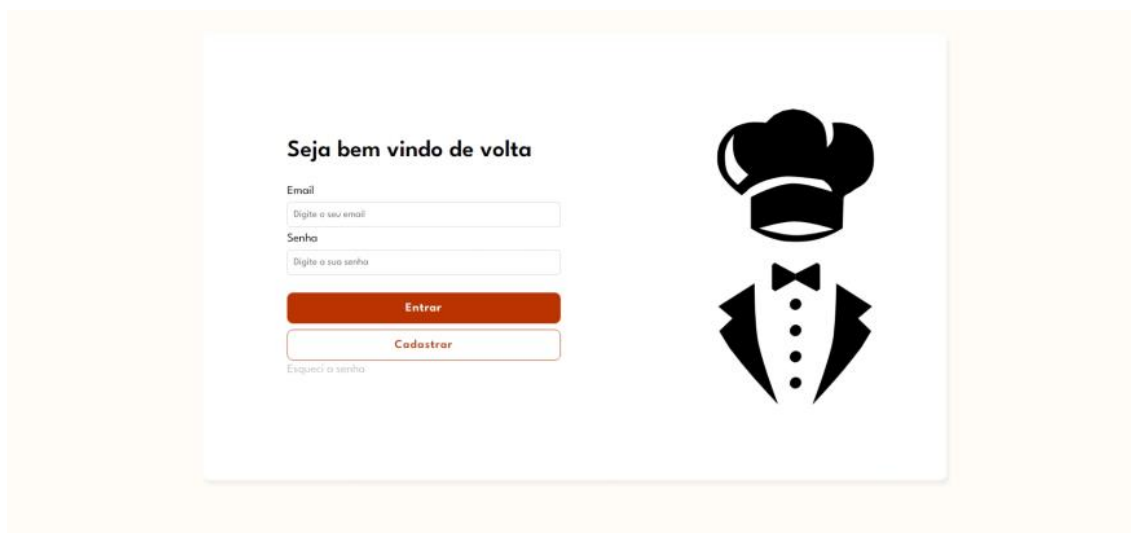


Fonte: Autoria Própria

3.6. Interfaces com o usuário

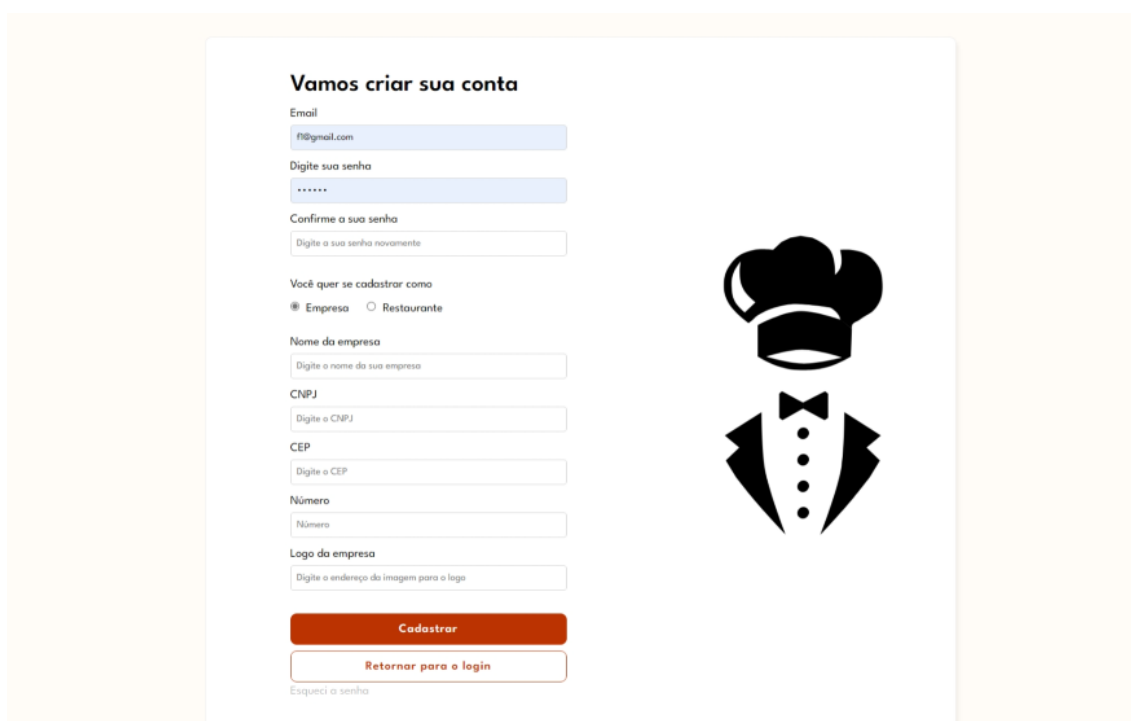


Figura 9 - Home



The login page features a white card on a light orange background. On the left, the heading "Seja bem vindo de volta" is followed by input fields for "Email" and "Senha", both with placeholder text "Digite o seu email" and "Digite a sua senha" respectively. Below these are two buttons: a red "Entrar" button and a white "Cadastrar" button. A link "Esqueci a senha" is at the bottom left. On the right, a black silhouette of a chef's hat and tuxedo is displayed.

Figura 10 - Página de Login



The registration page features a white card on a light orange background. On the left, the heading "Vamos criar sua conta" is followed by several input fields: "Email" (with placeholder "f@gmail.com"), "Digite sua senha" (with placeholder "\*\*\*\*\*"), "Confirme a sua senha" (with placeholder "Digite a sua senha novamente"), "Nome da empresa" (with placeholder "Digite o nome da sua empresa"), "CNPJ" (with placeholder "Digite o CNPJ"), "CEP" (with placeholder "Digite o CEP"), "Número" (with placeholder "Número"), and "Logo da empresa" (with placeholder "Digite o endereço da imagem para o logo"). There are radio buttons for "Empresa" (selected) and "Restaurante". Below the inputs are two buttons: a red "Cadastrar" button and a white "Retornar para o login" button. A link "Esqueci a senha" is at the bottom left. On the right, a black silhouette of a chef's hat and tuxedo is displayed.

Figura 11 - Página de Cadastro - Empresa

### Vamos criar sua conta

Email

Digite sua senha

Confirme a sua senha

Você quer se cadastrar como  
☐ Empresa ☒ Restaurante

Nome do restaurante

CNPJ

CEP

Número

Logo da empresa

**Cadastrar**

**Retornar para o login**

[Esqueci a senha](#)




Figura 12 - Página de Cadastro - Restaurante








    	<h3>Restaurantes disponíveis</h3>
	Tatiana marmitex
	Dona Maria Marmitex
	Colher de pau

Figura 13 - Lista de Restaurantes

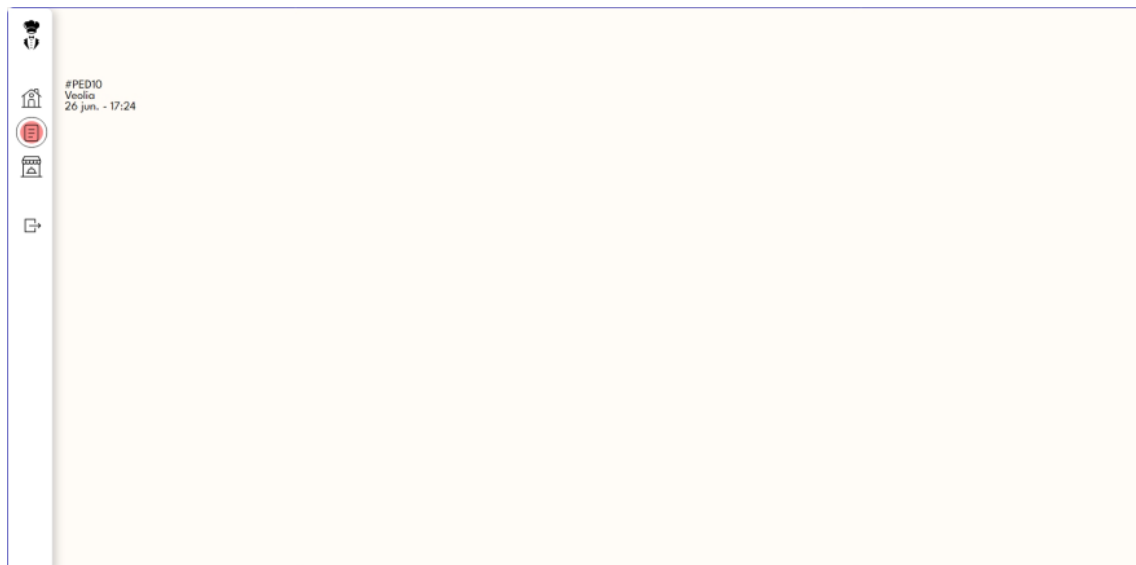


Figura 14 - Lista de Pedidos

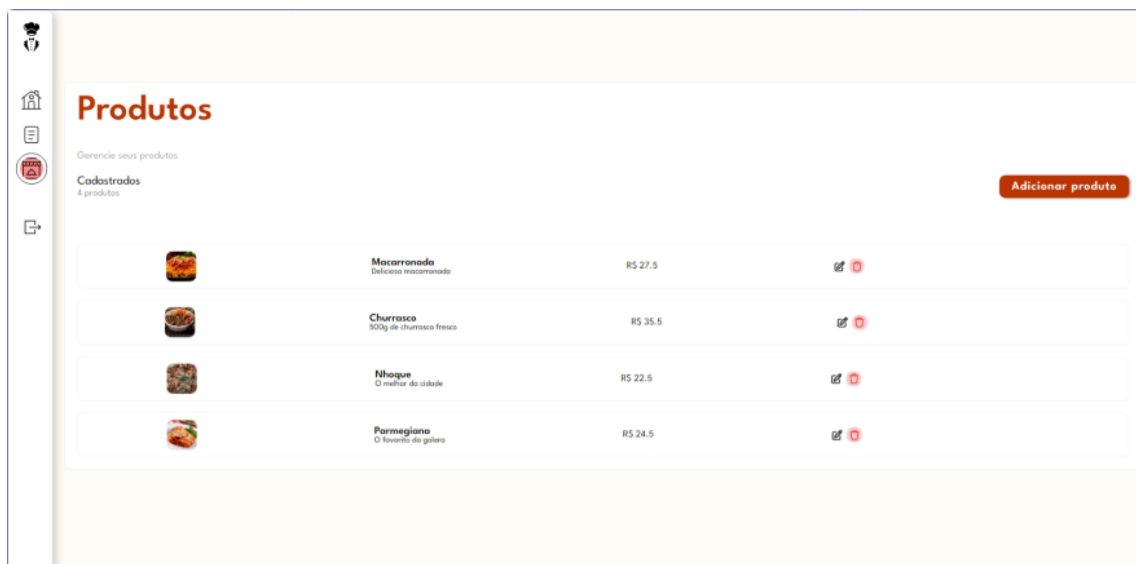
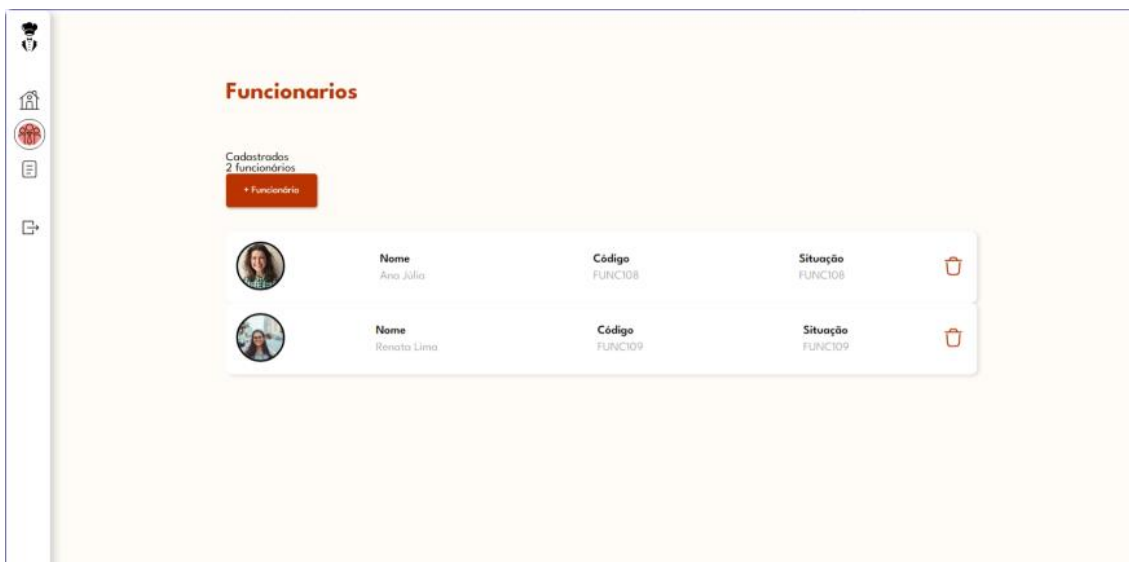


Figura 15 - Lista de Produtos/Pratos







	Nome	Código	Situação	
	Ana Júlia	FUNC108	FUNC108	
	Renata Lima	FUNC109	FUNC109	

Figura 16 - Lista de Funcionários da Empresa

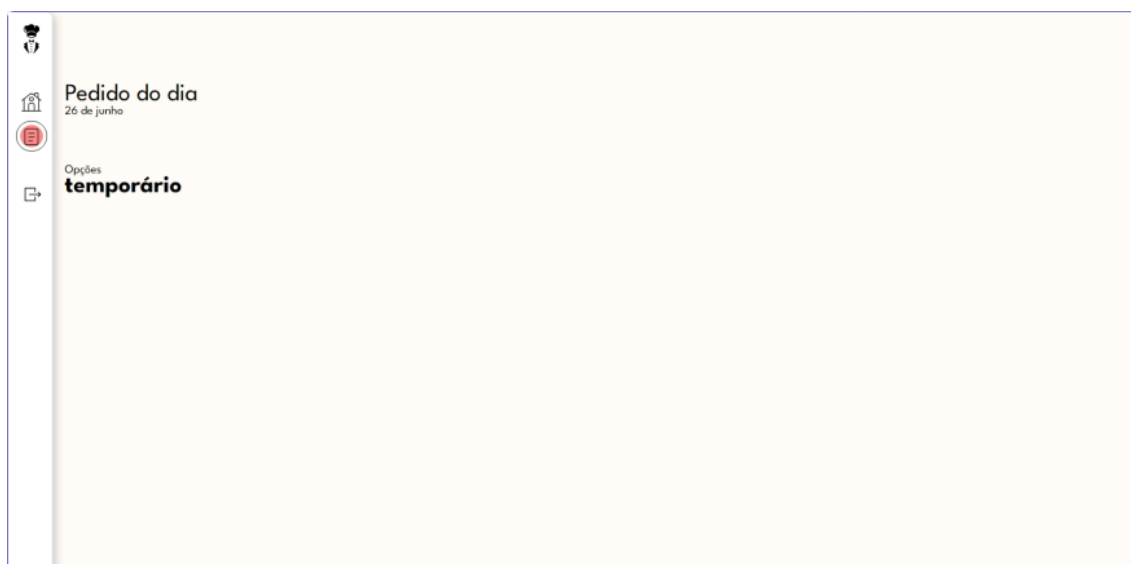


Figura 17 - Seção de Opções de Pedido do Funcionário

## 4. ESTRATÉGIA DE TESTES

Escrever uma classe de teste com JUnit envolve o uso de diversos métodos e anotações da API, criando pequenos cenários para verificar se o comportamento de uma unidade de trabalho está conforme o esperado.

Geralmente, a criação dos códigos de teste envolve a implementação de alguns ou todos os passos a seguir:

- Reunir e inicializar um conjunto de objetos, se necessário.
- Fazer com que esses objetos executem suas tarefas dentro de um contexto específico.
- Assegurar que o resultado obtido corresponda ao esperado.

Além dos testes automatizados com JUnit, a realização de testes de usabilidade é essencial para garantir uma experiência de usuário satisfatória em sites. O teste de usabilidade avalia a facilidade com que os usuários conseguem navegar, compreender e interagir com o site.

Esse tipo de teste geralmente envolve a observação de usuários reais enquanto eles executam tarefas específicas no site, identificando dificuldades e pontos de frustração. Feedback qualitativo e quantitativo é coletado para informar melhorias no design e na funcionalidade do site.

O objetivo é assegurar que o site seja intuitivo, eficiente e agradável de usar, melhorando assim a satisfação do usuário e a eficácia do site em atingir seus objetivos.

## 5. IMPLANTAÇÃO

Foi criado um repositório no GitHub para que pudéssemos contribuir em grupo, e encontra-se no link a seguir: <https://github.com/Bruno-Pasqual/foodClub>.

Além disso, está hospedado gratuitamente no Netlify: <https://lfoodclub.netlify.app/>



## 6. REFERÊNCIAS

SOMMERVILLE, Ian. **Engenharia de Software**. 8ª ed. Pearson, 2007

**Anexos.**

## Apêndice – inserir script do banco

### Script de Banco de Dados (Ctrl + Clique)

```
CREATE TABLE Usuario (
```

```
    ID_Usuario INT PRIMARY KEY,
```

```
    Senha VARCHAR(50),
```

```
    Tipo_Usuario VARCHAR(20),
```

```
    Email VARCHAR(50)
```

```
);
```

```
CREATE TABLE Empresa (
```

```
    ID_Empresa INT PRIMARY KEY,
```

```
    CNPJ VARCHAR(18),
```

```
    Nome_Empresa VARCHAR(50),
```

```
    Rua VARCHAR(100),
```

```
    CEP VARCHAR(10),
```

```
    Numero INT,
```

```
    Cidade VARCHAR(50),
```

```
    Estado VARCHAR(2),
```

```
    fk_Usuario_ID_Usuario INT,
```

```
    CONSTRAINT fk_Empresa_Usuario FOREIGN KEY  
(fk_Usuario_ID_Usuario) REFERENCES Usuario(ID_Usuario)
```

```
);
```

```
CREATE TABLE Funcionario (
```

```
    ID_Funcionario INT PRIMARY KEY,
```

```
Nome_Funcionario VARCHAR(50),  
  
fk_Empresa_ID_Empresa INT,  
  
fk_Usuario_ID_Usuario INT,  
  
CONSTRAINT fk_Funcionario_Empresa FOREIGN KEY  
(fk_Empresa_ID_Empresa) REFERENCES Empresa(ID_Empresa),  
  
CONSTRAINT fk_Funcionario_Usuario FOREIGN KEY  
(fk_Usuario_ID_Usuario) REFERENCES Usuario(ID_Usuario)  
  
);
```

```
CREATE TABLE Restaurante (  
  
    ID_Restaurante INT PRIMARY KEY,  
  
    Nome_Restaurante VARCHAR(50),  
  
    Rua VARCHAR(100),  
  
    CEP VARCHAR(10),  
  
    Numero INT,  
  
    Cidade VARCHAR(50),  
  
    Estado VARCHAR(2),  
  
    fk_Usuario_ID_Usuario INT,  
  
    CONSTRAINT fk_Restaurante_Usuario FOREIGN KEY  
(fk_Usuario_ID_Usuario) REFERENCES Usuario(ID_Usuario)  
  
);
```

```
CREATE TABLE Prato (  
  
    ID_Prato INT PRIMARY KEY,  
  
    Preco DECIMAL(10, 2),  
  
    Descricao_Prato VARCHAR(255),  
  
    Nome_Prato VARCHAR(50),
```

```

fk_Restaurante_ID_Restaurante INT,

CONSTRAINT fk_Prato_Restaurante FOREIGN KEY
(fk_Restaurante_ID_Restaurante) REFERENCES
Restaurante(ID_Restaurante)

);

```

```

CREATE TABLE Pedido_Empresa (

    Numero_Pedido INT PRIMARY KEY,

    ID_Pedido INT,

    fk_Restaurante_ID_Restaurante INT,

    fk_Empresa_ID_Empresa INT,

    CONSTRAINT fk_Pedido_Empresa_Restaurante FOREIGN KEY
(fk_Restaurante_ID_Restaurante) REFERENCES
Restaurante(ID_Restaurante),

    CONSTRAINT fk_Pedido_Empresa_Empresa FOREIGN KEY
(fk_Empresa_ID_Empresa) REFERENCES Empresa(ID_Empresa)

);

```

```

CREATE TABLE Pedido_Funcionario (

    ID_PedidoFunc INT PRIMARY KEY,

    valorPedido DECIMAL(10, 2),

    fk_Prato_ID_Prato INT,

    CONSTRAINT fk_PedidoFunc_Prato FOREIGN KEY (fk_Prato_ID_Prato)
REFERENCES Prato(ID_Prato)

);

```

```

CREATE TABLE EmpresaFuncionario (

    ID INT PRIMARY KEY,

```

```

fk_Pedido_Empresa_ID_Pedido INT,

fk_Pedido_Funcionario_ID INT,

CONSTRAINT fk_EmpresaFunc_Pedido_Empresa FOREIGN KEY
(fk_Pedido_Empresa_ID_Pedido) REFERENCES
Pedido_Empresa(Numero_Pedido),

CONSTRAINT fk_EmpresaFunc_Pedido_Funcionario FOREIGN KEY
(fk_Pedido_Funcionario_ID) REFERENCES
Pedido_Funcionario(ID_PedidoFunc)

);

CREATE TABLE Log (

ID_Log INT PRIMARY KEY,

tipoAcao VARCHAR(50),

fk_Usuario_ID_Usuario INT,

CONSTRAINT fk_Log_Usuario FOREIGN KEY (fk_Usuario_ID_Usuario)
REFERENCES Usuario(ID_Usuario)

);

```

-- Inserindo usuários

```

INSERT INTO Usuario (ID_Usuario, Senha, Tipo_Usuario, Email) VALUES (1,
'senha123', 'Admin', 'admin1@empresa.com');

```

```

INSERT INTO Usuario (ID_Usuario, Senha, Tipo_Usuario, Email) VALUES (2,
'senha456', 'Funcionario', 'func1@empresa.com');

```

```

INSERT INTO Usuario (ID_Usuario, Senha, Tipo_Usuario, Email) VALUES (3,
'senha789', 'Restaurante', 'rest1@restaurante.com');

```

```

INSERT INTO Usuario (ID_Usuario, Senha, Tipo_Usuario, Email) VALUES (4,
'senha234', 'Admin', 'admin2@empresa.com');

```

```

INSERT INTO Usuario (ID_Usuario, Senha, Tipo_Usuario, Email) VALUES (5,
'senha567', 'Funcionario', 'func2@empresa.com');

```

```
INSERT INTO Usuario (ID_Usuario, Senha, Tipo_Usuario, Email) VALUES (6, 'senha890', 'Restaurante', 'rest2@restaurante.com');
```

```
INSERT INTO Usuario (ID_Usuario, Senha, Tipo_Usuario, Email) VALUES (7, 'senha345', 'Admin', 'admin3@empresa.com');
```

```
INSERT INTO Usuario (ID_Usuario, Senha, Tipo_Usuario, Email) VALUES (8, 'senha678', 'Funcionario', 'func3@empresa.com');
```

```
INSERT INTO Usuario (ID_Usuario, Senha, Tipo_Usuario, Email) VALUES (9, 'senha901', 'Restaurante', 'rest3@restaurante.com');
```

```
INSERT INTO Usuario (ID_Usuario, Senha, Tipo_Usuario, Email) VALUES (10, 'senha012', 'Admin', 'admin4@empresa.com');
```

-- Inserindo empresas

```
INSERT INTO Empresa (ID_Empresa, CNPJ, Nome_Empresa, Rua, CEP, Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (1, '12.345.678/0001-99', 'Empresa A', 'Rua 1', '12345-678', 100, 'Cidade A', 'AA', 1);
```

```
INSERT INTO Empresa (ID_Empresa, CNPJ, Nome_Empresa, Rua, CEP, Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (2, '98.765.432/0001-11', 'Empresa B', 'Rua 2', '87654-321', 200, 'Cidade B', 'BB', 4);
```

```
INSERT INTO Empresa (ID_Empresa, CNPJ, Nome_Empresa, Rua, CEP, Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (3, '12.345.678/0002-88', 'Empresa C', 'Rua 3', '12345-679', 300, 'Cidade C', 'CC', 7);
```

```
INSERT INTO Empresa (ID_Empresa, CNPJ, Nome_Empresa, Rua, CEP, Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (4, '98.765.432/0002-22', 'Empresa D', 'Rua 4', '87654-322', 400, 'Cidade D', 'DD', 1);
```

```
INSERT INTO Empresa (ID_Empresa, CNPJ, Nome_Empresa, Rua, CEP, Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (5, '12.345.678/0003-77', 'Empresa E', 'Rua 5', '12345-680', 500, 'Cidade E', 'EE', 4);
```

```
INSERT INTO Empresa (ID_Empresa, CNPJ, Nome_Empresa, Rua, CEP, Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (6, '98.765.432/0003-33', 'Empresa F', 'Rua 6', '87654-323', 600, 'Cidade F', 'FF', 7);
```

```
INSERT INTO Empresa (ID_Empresa, CNPJ, Nome_Empresa, Rua, CEP,
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (7,
'12.345.678/0004-66', 'Empresa G', 'Rua 7', '12345-681', 700, 'Cidade G', 'GG',
1);
```

```
INSERT INTO Empresa (ID_Empresa, CNPJ, Nome_Empresa, Rua, CEP,
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (8,
'98.765.432/0004-44', 'Empresa H', 'Rua 8', '87654-324', 800, 'Cidade H', 'HH',
4);
```

```
INSERT INTO Empresa (ID_Empresa, CNPJ, Nome_Empresa, Rua, CEP,
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (9,
'12.345.678/0005-55', 'Empresa I', 'Rua 9', '12345-682', 900, 'Cidade I', 'II', 7);
```

```
INSERT INTO Empresa (ID_Empresa, CNPJ, Nome_Empresa, Rua, CEP,
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (10,
'98.765.432/0005-55', 'Empresa J', 'Rua 10', '87654-325', 1000, 'Cidade J', 'JJ',
1);
```

-- Inserindo funcionários

```
INSERT INTO Funcionario (ID_Funcionario, Nome_Funcionario,
fk_Empresa_ID_Empresa, fk_Usuario_ID_Usuario) VALUES (1, 'Funcionario
A', 1, 2);
```

```
INSERT INTO Funcionario (ID_Funcionario, Nome_Funcionario,
fk_Empresa_ID_Empresa, fk_Usuario_ID_Usuario) VALUES (2, 'Funcionario
B', 2, 5);
```

```
INSERT INTO Funcionario (ID_Funcionario, Nome_Funcionario,
fk_Empresa_ID_Empresa, fk_Usuario_ID_Usuario) VALUES (3, 'Funcionario
C', 3, 8);
```

```
INSERT INTO Funcionario (ID_Funcionario, Nome_Funcionario,
fk_Empresa_ID_Empresa, fk_Usuario_ID_Usuario) VALUES (4, 'Funcionario
D', 4, 2);
```

```
INSERT INTO Funcionario (ID_Funcionario, Nome_Funcionario,
fk_Empresa_ID_Empresa, fk_Usuario_ID_Usuario) VALUES (5, 'Funcionario
E', 5, 5);
```

```
INSERT INTO Funcionario (ID_Funcionario, Nome_Funcionario,
fk_Empresa_ID_Empresa, fk_Usuario_ID_Usuario) VALUES (6, 'Funcionario F',
6, 8);
```

```
INSERT INTO Funcionario (ID_Funcionario, Nome_Funcionario,
fk_Empresa_ID_Empresa, fk_Usuario_ID_Usuario) VALUES (7, 'Funcionario
G', 7, 2);
```



```
INSERT INTO Funcionario (ID_Funcionario, Nome_Funcionario,  
fk_Empresa_ID_Empresa, fk_Usuario_ID_Usuario) VALUES (8, 'Funcionario  
H', 8, 5);
```

```
INSERT INTO Funcionario (ID_Funcionario, Nome_Funcionario,  
fk_Empresa_ID_Empresa, fk_Usuario_ID_Usuario) VALUES (9, 'Funcionario I',  
9, 8);
```

```
INSERT INTO Funcionario (ID_Funcionario, Nome_Funcionario,  
fk_Empresa_ID_Empresa, fk_Usuario_ID_Usuario) VALUES (10, 'Funcionario  
J', 10, 2);
```

-- Inserindo restaurantes

```
INSERT INTO Restaurante (ID_Restaurante, Nome_Restaurante, Rua, CEP,  
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (1, 'Restaurante A',  
'Rua 1', '12345-678', 100, 'Cidade A', 'AA', 3);
```

```
INSERT INTO Restaurante (ID_Restaurante, Nome_Restaurante, Rua, CEP,  
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (2, 'Restaurante B',  
'Rua 2', '87654-321', 200, 'Cidade B', 'BB', 6);
```

```
INSERT INTO Restaurante (ID_Restaurante, Nome_Restaurante, Rua, CEP,  
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (3, 'Restaurante C',  
'Rua 3', '12345-679', 300, 'Cidade C', 'CC', 9);
```

```
INSERT INTO Restaurante (ID_Restaurante, Nome_Restaurante, Rua, CEP,  
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (4, 'Restaurante D',  
'Rua 4', '87654-322', 400, 'Cidade D', 'DD', 3);
```

```
INSERT INTO Restaurante (ID_Restaurante, Nome_Restaurante, Rua, CEP,  
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (5, 'Restaurante E',  
'Rua 5', '12345-680', 500, 'Cidade E', 'EE', 6);
```

```
INSERT INTO Restaurante (ID_Restaurante, Nome_Restaurante, Rua, CEP,  
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (6, 'Restaurante F',  
'Rua 6', '87654-323', 600, 'Cidade F', 'FF', 9);
```

```
INSERT INTO Restaurante (ID_Restaurante, Nome_Restaurante, Rua, CEP,  
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (7, 'Restaurante G',  
'Rua 7', '12345-681', 700, 'Cidade G', 'GG', 3);
```

```
INSERT INTO Restaurante (ID_Restaurante, Nome_Restaurante, Rua, CEP,  
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (8, 'Restaurante H',  
'Rua 8', '87654-324', 800, 'Cidade H', 'HH', 6);
```

```
INSERT INTO Restaurante (ID_Restaurante, Nome_Restaurante, Rua, CEP,
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (9, 'Restaurante I',
'Rua 9', '12345-682', 900, 'Cidade I', 'II', 9);
```

```
INSERT INTO Restaurante (ID_Restaurante, Nome_Restaurante, Rua, CEP,
Numero, Cidade, Estado, fk_Usuario_ID_Usuario) VALUES (10, 'Restaurante
J', 'Rua 10', '87654-325', 1000, 'Cidade J', 'JJ', 3);
```

-- Inserindo pratos

```
INSERT INTO Prato (ID_Prato, Preco, Descricao_Prato, Nome_Prato,
fk_Restaurante_ID_Restaurante) VALUES (1, 25.50, 'Descricao Prato A', 'Prato
A', 1);
```

```
INSERT INTO Prato (ID_Prato, Preco, Descricao_Prato, Nome_Prato,
fk_Restaurante_ID_Restaurante) VALUES (2, 30.00, 'Descricao Prato B', 'Prato
B', 2);
```

```
INSERT INTO Prato (ID_Prato, Preco, Descricao_Prato, Nome_Prato,
fk_Restaurante_ID_Restaurante) VALUES (3, 35.00, 'Descricao Prato C', 'Prato
C', 3);
```

```
INSERT INTO Prato (ID_Prato, Preco, Descricao_Prato, Nome_Prato,
fk_Restaurante_ID_Restaurante) VALUES (4, 40.00, 'Descricao Prato D', 'Prato
D', 4);
```

```
INSERT INTO Prato (ID_Prato, Preco, Descricao_Prato, Nome_Prato,
fk_Restaurante_ID_Restaurante) VALUES (5, 45.00, 'Descricao Prato E', 'Prato
E', 5);
```

```
INSERT INTO Prato (ID_Prato, Preco, Descricao_Prato, Nome_Prato,
fk_Restaurante_ID_Restaurante) VALUES (6, 50.00, 'Descricao Prato F', 'Prato
F', 6);
```

```
INSERT INTO Prato (ID_Prato, Preco, Descricao_Prato, Nome_Prato,
fk_Restaurante_ID_Restaurante) VALUES (7, 55.00, 'Descricao Prato G', 'Prato
G', 7);
```

```
INSERT INTO Prato (ID_Prato, Preco, Descricao_Prato, Nome_Prato,
fk_Restaurante_ID_Restaurante) VALUES (8, 60.00, 'Descricao Prato H', 'Prato
H', 8);
```

```
INSERT INTO Prato (ID_Prato, Preco, Descricao_Prato, Nome_Prato,
fk_Restaurante_ID_Restaurante) VALUES (9, 65.00, 'Descricao Prato I', 'Prato
I', 9);
```

```
INSERT INTO Prato (ID_Prato, Preco, Descricao_Prato, Nome_Prato,  
fk_Restaurante_ID_Restaurante) VALUES (10, 70.00, 'Descricao Prato J',  
'Prato J', 10);
```

-- Inserindo pedidos de empresas

```
INSERT INTO Pedido_Empresa (Numero_Pedido, ID_Pedido,  
fk_Restaurante_ID_Restaurante, fk_Empresa_ID_Empresa) VALUES (1, 1, 1,  
1);
```

```
INSERT INTO Pedido_Empresa (Numero_Pedido, ID_Pedido,  
fk_Restaurante_ID_Restaurante, fk_Empresa_ID_Empresa) VALUES (2, 2, 2,  
2);
```

```
INSERT INTO Pedido_Empresa (Numero_Pedido, ID_Pedido,  
fk_Restaurante_ID_Restaurante, fk_Empresa_ID_Empresa) VALUES (3, 3, 3,  
3);
```

```
INSERT INTO Pedido_Empresa (Numero_Pedido, ID_Pedido,  
fk_Restaurante_ID_Restaurante, fk_Empresa_ID_Empresa) VALUES (4, 4, 4,  
4);
```

```
INSERT INTO Pedido_Empresa (Numero_Pedido, ID_Pedido,  
fk_Restaurante_ID_Restaurante, fk_Empresa_ID_Empresa) VALUES (5, 5, 5,  
5);
```

```
INSERT INTO Pedido_Empresa (Numero_Pedido, ID_Pedido,  
fk_Restaurante_ID_Restaurante, fk_Empresa_ID_Empresa) VALUES (6, 6, 6,  
6);
```

```
INSERT INTO Pedido_Empresa (Numero_Pedido, ID_Pedido,  
fk_Restaurante_ID_Restaurante, fk_Empresa_ID_Empresa) VALUES (7, 7, 7,  
7);
```

```
INSERT INTO Pedido_Empresa (Numero_Pedido, ID_Pedido,  
fk_Restaurante_ID_Restaurante, fk_Empresa_ID_Empresa) VALUES (8, 8, 8,  
8);
```

```
INSERT INTO Pedido_Empresa (Numero_Pedido, ID_Pedido,  
fk_Restaurante_ID_Restaurante, fk_Empresa_ID_Empresa) VALUES (9, 9, 9,  
9);
```

```
INSERT INTO Pedido_Empresa (Numero_Pedido, ID_Pedido,  
fk_Restaurante_ID_Restaurante, fk_Empresa_ID_Empresa) VALUES (10, 10,  
10, 10);
```

-- Inserindo pedidos de funcionários

```
INSERT INTO Pedido_Funcionario (ID_PedidoFunc, valorPedido,
fk_Prato_ID_Prato) VALUES (1, 25.50, 1);
```

```
INSERT INTO Pedido_Funcionario (ID_PedidoFunc, valorPedido,
fk_Prato_ID_Prato) VALUES (2, 30.00, 2);
```

```
INSERT INTO Pedido_Funcionario (ID_PedidoFunc, valorPedido,
fk_Prato_ID_Prato) VALUES (3, 35.00, 3);
```

```
INSERT INTO Pedido_Funcionario (ID_PedidoFunc, valorPedido,
fk_Prato_ID_Prato) VALUES (4, 40.00, 4);
```

```
INSERT INTO Pedido_Funcionario (ID_PedidoFunc, valorPedido,
fk_Prato_ID_Prato) VALUES (5, 45.00, 5);
```

```
INSERT INTO Pedido_Funcionario (ID_PedidoFunc, valorPedido,
fk_Prato_ID_Prato) VALUES (6, 50.00, 6);
```

```
INSERT INTO Pedido_Funcionario (ID_PedidoFunc, valorPedido,
fk_Prato_ID_Prato) VALUES (7, 55.00, 7);
```

```
INSERT INTO Pedido_Funcionario (ID_PedidoFunc, valorPedido,
fk_Prato_ID_Prato) VALUES (8, 60.00, 8);
```

```
INSERT INTO Pedido_Funcionario (ID_PedidoFunc, valorPedido,
fk_Prato_ID_Prato) VALUES (9, 65.00, 9);
```

```
INSERT INTO Pedido_Funcionario (ID_PedidoFunc, valorPedido,
fk_Prato_ID_Prato) VALUES (10, 70.00, 10);
```

-- Inserindo EmpresaFuncionario

```
INSERT INTO EmpresaFuncionario (ID, fk_Pedido_Empresa_ID_Pedido,
fk_Pedido_Funcionario_ID) VALUES (1, 1, 1);
```

```
INSERT INTO EmpresaFuncionario (ID, fk_Pedido_Empresa_ID_Pedido,
fk_Pedido_Funcionario_ID) VALUES (2, 2, 2);
```

```
INSERT INTO EmpresaFuncionario (ID, fk_Pedido_Empresa_ID_Pedido,
fk_Pedido_Funcionario_ID) VALUES (3, 3, 3);
```

```
INSERT INTO EmpresaFuncionario (ID, fk_Pedido_Empresa_ID_Pedido,
fk_Pedido_Funcionario_ID) VALUES (4, 4, 4);
```

```
INSERT INTO EmpresaFuncionario (ID, fk_Pedido_Empresa_ID_Pedido,
fk_Pedido_Funcionario_ID) VALUES (5, 5, 5);
```

```
INSERT INTO EmpresaFuncionario (ID, fk_Pedido_Empresa_ID_Pedido,  
fk_Pedido_Funcionario
```

-- Selecionar todos os dados da tabela de Usuários

```
SELECT * FROM Usuario;
```

-- Selecionar todos os dados da tabela de Empresas

```
SELECT * FROM Empresa;
```

-- Selecionar todos os dados da tabela de Funcionários

```
SELECT * FROM Funcionario;
```

-- Selecionar todos os dados da tabela de Restaurantes

```
SELECT * FROM Restaurante;
```

-- Selecionar todos os dados da tabela de Pratos

```
SELECT * FROM Prato;
```

-- Selecionar todos os dados da tabela de Pedidos de Empresa

```
SELECT * FROM Pedido_Empresa;
```

-- Selecionar todos os dados da tabela de Pedidos de Funcionários

```
SELECT * FROM Pedido_Funcionario;
```

-- Selecionar todos os dados da tabela de EmpresaFuncionario

```
SELECT * FROM EmpresaFuncionario;
```

-- Selecionar todos os dados da tabela de Logs

```
SELECT * FROM Log;
```

-- Selecionar todos os funcionários de uma empresa específica (ex: Empresa com ID 1)

```
SELECT * FROM Funcionario WHERE fk_Empresa_ID_Empresa = 1;
```

-- Selecionar todos os pratos de um restaurante específico (ex: Restaurante com ID 1)

```
SELECT * FROM Prato WHERE fk_Restaurante_ID_Restaurante = 1;
```

-- Selecionar todos os pedidos feitos por uma empresa específica (ex: Empresa com ID 1)

```
SELECT * FROM Pedido_Empresa WHERE fk_Empresa_ID_Empresa = 1;
```

-- Selecionar todos os pedidos feitos por um funcionário específico (ex: Funcionario com ID 1)

```
SELECT * FROM Pedido_Funcionario WHERE ID_PedidoFunc IN (SELECT  
fk_Pedido_Funcionario_ID FROM EmpresaFuncionario WHERE  
fk_Pedido_Empresa_ID_Pedido IN (SELECT ID_Pedido FROM  
Pedido_Empresa WHERE fk_Empresa_ID_Empresa = 1));
```

-- Selecionar logs de ações de um usuário específico (ex: Usuário com ID 1)

```
SELECT * FROM Log WHERE fk_Usuario_ID_Usuario = 1;
```

