



**UNIVERSIDADE ESTADUAL DA PARAÍBA  
CAMPUS I - CAMPINA GRANDE  
CENTRO DE CIÊNCIAS E TECNOLOGIA  
DEPARTAMENTO DE COMPUTAÇÃO  
CURSO DE BACHARELADO EM COMPUTAÇÃO**

**SÉRGIO DAVI MARINHO TEOTÔNIO**

**UNIDELIVERY: UMA PLATAFORMA CUSTOMIZÁVEL PARA GERENCIAMENTO  
DE PEDIDOS PARA DELIVERY**

**CAMPINA GRANDE - PB  
2023**

**SÉRGIO DAVI MARINHO TEOTÔNIO**

**UNIDELIVERY: UMA PLATAFORMA CUSTOMIZÁVEL PARA GERENCIAMENTO  
DE PEDIDOS PARA DELIVERY**

Trabalho de Conclusão de Curso apresentado ao Curso de Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de Bacharel em Computação.

**Orientador:** Profa. Ma. Cheyenne Ribeiro Guedes Isidro

**CAMPINA GRANDE - PB  
2023**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

T314u Teotonio, Sergio Davi Marinho.  
Unidelivery [manuscrito] : uma plataforma customizável para gerenciamento de pedidos para *delivery* / Sergio Davi Marinho Teotonio. - 2023.  
30 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2023.

"Orientação : Profa. Ma. Cheyenne Ribeiro Guedes Isidro, Coordenação do Curso de Computação - CCT. "

1. Vendas online. 2. Aplicações web. 3. Plataforma online.  
4. Gerenciamento de pedidos. I. Título

21. ed. CDD 005.3

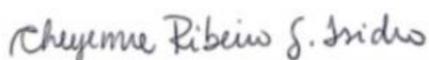
SÉRGIO DAVI MARINHO TEOTÔNIO

UNIDELIVERY: UMA PLATAFORMA CUSTOMIZÁVEL PARA GERENCIAMENTO  
DE PEDIDOS PARA DELIVERY

Trabalho de Conclusão de Curso  
apresentado ao Curso de Ciência da  
Computação do Centro de Ciências e  
Tecnologia da Universidade Estadual da  
Paraíba, como requisito parcial à  
obtenção do título de Bacharel em  
Computação.

Aprovado em: 29 / 08 / 2023.

**BANCA EXAMINADORA**



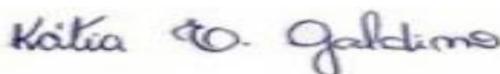
---

Profa. Me. Cheyenne Ribeiro Guedes Isidro (CCT/UEPB)  
Orientador(a)



---

Prof. Me. Edson Holanda Cavalcante Junior (CCT/UEPB)  
Examinador(a)



---

Profa. Dra. Kátia Elizabete Galdino (CCT/UEPB)  
Examinador(a)

## RESUMO

Com o distanciamento físico imposto pela pandemia do COVID-19, muitos estabelecimentos de pequeno e médio porte foram compelidos a adotar meios online para comercializar seus produtos durante esse período. Dentre as opções disponíveis, a utilização de plataformas online para o gerenciamento de pedidos se destacou como uma forma de efetuar vendas online. No entanto, tais soluções apresentavam problemas tais como layouts padronizados que não refletiam a identidade visual do estabelecimento, usabilidade insatisfatória e baixa performance. Já outras soluções utilizadas, tal como o Whatsapp, não possuíam as funcionalidades necessárias para dar suporte ao negócio uma vez que não foram desenvolvidas para tal fim. Seu uso precisou ser adaptado, gerando assim problemas de gerenciamento dos pedidos e pagamentos. Diante desse cenário, buscou-se desenvolver uma ferramenta computacional com o intuito de obter maior adesão por parte dos comerciantes de pequeno porte, reduzindo os custos e simplificando sua manutenção. Esse trabalho apresenta, portanto, o relatório técnico de desenvolvimento dessa ferramenta, chamada Unidelivery, criada com tecnologias disponíveis no mercado, resultando em uma solução com desempenho notável, escalabilidade e mais customização para adequar-se à identidade visual de cada estabelecimento. Além disso, prioriza a facilidade de uso para o usuário final. A solução proposta apresentou ganhos significativos para o estabelecimento que a testou em ambiente real, bem como atraiu maior adesão por parte dos clientes do estabelecimento. A disponibilidade do serviço quase que ininterrupta foi outro destaque, contribuindo para a satisfação dos usuários.

**Palavras-Chave:** vendas online; aplicações web; plataforma online; gerenciamento de pedidos.

## **ABSTRACT**

Due to the physical distancing imposed by the COVID-19 pandemic, many small and medium-sized establishments found themselves compelled to embrace online channels for product marketing during this period. Among the available options, the use of online platforms for order management emerged as a prominent means of conducting online sales. However, such solutions encountered issues such as standardized layouts that failed to reflect the establishment's visual identity, subpar usability, and suboptimal performance. Other solutions, such as WhatsApp, lacked the necessary functionalities to support business operations as they were not originally designed for this purpose. Consequently, their use had to be adapted, leading to challenges in order and payment management. In light of this scenario, there arose a need to develop a computational tool aimed at garnering greater participation from small-scale merchants while simultaneously reducing costs and streamlining maintenance efforts. This work, therefore, presents the technical development report of this tool, known as Unidelivery. Unidelivery was built using readily available market technologies, with the goal of providing a solution characterized by outstanding performance, scalability, and enhanced customization to align with the visual identity of each establishment. Additionally, it places a strong emphasis on user-friendliness for the end user. The proposed solution delivered significant benefits to the establishments that tested it in real-world scenarios and garnered increased adoption from the establishments' clientele. The nearly uninterrupted service availability emerged as another noteworthy feature, contributing to user satisfaction. This research offers valuable insights into how technology can be tailored to meet the specific needs of small and medium-sized businesses in an ever-evolving online sales environment.

**Keywords:** online sales; web applications; online platform; order management.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Arquitetura do sistema.....	17
Figura 2 - Tela de login.....	19
Figura 3 - Tela de catálogos.....	20
Figura 4 - Tela de categorias.....	20
Figura 5 - Tela de itens do catálogo.....	21
Figura 6 - Tela de edição de um item.....	22
Figura 7 - Tela de cadastro de item.....	22
Figura 8 - Tela de informações do estabelecimento.....	23
Figura 9 - Tela de gerenciamento de horários de funcionamento.....	24
Figura 10 - Tela de catálogo e horário de funcionamento.....	24
Figura 11 - Tela para informar dados e o resumo do pedido.....	26
Figura 12 - Tela de Whatsapp do cliente.....	27

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>7</b>
<b>2</b>	<b>METODOLOGIA.....</b>	<b>9</b>
2.1	Levantamento de requisitos.....	9
2.1.1	Requisitos Funcionais.....	10
2.1.2	Requisitos Não-Funcionais.....	13
<b>3</b>	<b>IMPLEMENTAÇÃO DO BACK-END.....</b>	<b>14</b>
3.1	Linguagens.....	14
3.2	Persistência.....	15
3.3	Segurança.....	15
3.4	Disponibilidade.....	16
<b>4</b>	<b>IMPLEMENTAÇÃO DO FRONT-END.....</b>	<b>18</b>
4.1	Funcionalidades.....	18
4.1.1	Telas para o perfil Comerciante.....	19
4.1.2	Telas para o perfil Cliente.....	24
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>28</b>
	<b>REFERÊNCIAS.....</b>	<b>30</b>

## 1 INTRODUÇÃO

Após o crescimento ascendente de vendas online durante e após a pandemia da COVID-19, devido à necessidade de manter o distanciamento físico, pequenos estabelecimentos tiveram que se reinventar através de ferramentas na Internet para comercializar seus produtos e alcançarem um maior número de clientes.

Porém o custo e o tempo necessário para desenvolvimento de um *e-commerce* completo e totalmente sob demanda, é alto demais para a maioria dos empreendedores de pequeno porte, fazendo com que busquem utilizar ferramentas gratuitas, tal como o *WhatsApp*. Este aplicativo mensageiro entre pessoas, lançado em 2009 e que hoje conta com mais de 2 bilhões de usuários ativos, está cada vez mais em uso por parte dos estabelecimentos (Loubak, 2020). No entanto, ao utilizar uma ferramenta que não foi desenvolvida para vendas online, existe um desconforto e confusão durante o processo para anotar um pedido de um cliente ou compartilhar o cardápio/catálogo de itens disponíveis, pois, os clientes precisam especificar o item que desejam, informar o endereço, a forma de pagamento e possíveis observações no pedido. Além disso, a ferramenta só permite o compartilhamento de imagens, documentos ou áudios e não possui uma tratativa específica para o assunto.

Com isso, o empreendedor perde bastante tempo para entender o que o cliente quer e acaba congestionando a sua fila de pedidos, além de dificultar o gerenciamento da ordem dos pedidos realizados, em andamento e os já finalizados.

Outra problemática encontrada, se dá devido ao estabelecimento possuir uma logomarca ou cores específicas que definem uma identidade visual ao local. A maioria das plataformas existentes, tais como a *OlaClick*<sup>1</sup>, *Cardapiando*<sup>2</sup> e *MenuDino*<sup>3</sup> possuem um layout padrão independente do estabelecimento, além de não facilitar o processo para escolha de itens mais complexos como montar uma pizza, ou itens que precisam de mais configurações do que uma simples escolha.

Existem meios de integração direta com o aplicativo de mensagens *WhatsApp*<sup>4</sup>, porém o acesso é controlado e gerenciado pela plataforma, o que

---

<sup>1</sup> [OlaClick](https://olaclick.com/pt-br/) - <https://olaclick.com/pt-br/>

<sup>2</sup> [Cardapiando](https://cardapiando.com/) - <https://cardapiando.com/>

<sup>3</sup> [MenuDino](https://www.menudino.com/) - <https://www.menudino.com/>

<sup>4</sup> [WhatsApp](https://www.whatsapp.com) - <https://www.whatsapp.com>

dificulta a integração para empresas de pequeno porte. Algumas plataformas de delivery como o *Ifood*<sup>5</sup> - fundado em 2011, conta com mais de 300 mil restaurantes parceiros e mais de 65 milhões de pedidos por mês na plataforma (Ifood, 2023) - compactuam da ideia de atender pequenos empreendedores, porém, a política de precificação é ruim para a maioria dos estabelecimentos que o utilizam, pois a cobrança de taxas por cada pedido do estabelecimento eleva o preço para o usuário final.

Diante desse contexto, apresentamos neste documento o relatório técnico sobre o desenvolvimento de uma plataforma chamada **Unidelivery**, para controle de pedidos online que possa ser customizada de acordo com a identidade visual do estabelecimento e autogerenciada pelo dono do estabelecimento, utilizando o *WhatsApp* para contato inicial. Tal plataforma está disponível de forma online e não necessita de instalação de aplicativos no aparelho do usuário final. Outra característica é que ela é escalável, desenvolvida utilizando tecnologias *open source* e servidores de pequeno porte, além de serviços de baixo custo e com o foco em facilitar o máximo possível a experiência do usuário.

---

<sup>5</sup> [Ifood](https://www.ifood.com.br) - <https://www.ifood.com.br>

## 2 METODOLOGIA

O desenvolvimento do sistema envolveu as etapas iniciais de planejamento com a realização do levantamento dos requisitos funcionais e não funcionais, apresentados neste capítulo. Em seguida foram definidas as tecnologias a serem utilizadas, e iniciou-se o desenvolvimento do back-end e posteriormente do front-end conforme resultados apresentados nos capítulos seguintes.

### 2.1 Levantamento de requisitos

Para especificar e desenvolver o sistema, foram realizadas diversas reuniões online, através de ferramentas como o Google Meet <sup>6</sup>, com alguns comerciantes que utilizam o *WhatsApp* para se comunicar com o cliente e registrar um pedido. Durante as conversas, detalhamos algumas das quais seriam as principais funcionalidades da aplicação, tais como, fácil acesso e manuseio para o usuário final, como também uma área de administração para permitir que os donos de cada estabelecimento cadastrado possam manter atualizados os valores e a disponibilidade dos itens do seu catálogo. Também conversamos sobre o diferencial entre as aplicações existentes no mercado, e nesse ponto levamos em consideração que a maioria dos sistemas que existem não possibilitam uma leve customização da interface gráfica para cada estabelecimento. Tal falta de personalização deixa alguns dos comerciantes desconfortáveis, pois eles desejam ter algo simples e prático, mas sem perder a sua própria identidade visual.

Alguns problemas em utilizar um aplicativo de mensagens de texto como o *WhatsApp* para se relacionar com o cliente final foram discutidos e pontuados durante esse levantamento de requisitos, por exemplo:

- Erros de digitação entre as partes envolvidas.
- Envio contínuo de imagens e arquivos em PDF com o catálogo/cardápio do estabelecimento.
- Desordem entre a sequência das informações entre endereço, itens e forma de pagamento. Tal desorganização causa problemas para anotar o pedido dependendo do fluxo de mensagens do estabelecimento.

---

<sup>6</sup> [Google Meet](https://meet.google.com) - <https://meet.google.com>

Portanto, o sistema a ser desenvolvido deveria atender pessoas que desejam realizar um pedido em estabelecimento previamente cadastrado. Cada estabelecimento deve possuir uma URL customizada, e ao abrir a URL em um navegador web de qualquer dispositivo móvel ou de mesa, o usuário irá ter acesso ao catálogo de itens do estabelecimento, podendo escolher os itens e respectiva quantidade, informar seu endereço para entrega, forma de pagamento e ter um feedback instantâneo do valor total do pedido. O sistema será responsável por unir todas as informações e ao finalizar o pedido irá iniciar uma janela do aplicativo *WhatsApp* do cliente com a conta do estabelecimento selecionada e um texto com todos os detalhes do pedido previamente escrito e preenchido no campo de texto do aparelho do cliente, restando apenas para o usuário a única ação de enviar a mensagem gerada.

Uma das principais demandas para o desenvolvimento é a capacidade da aplicação se adaptar em cores de acordo com a identidade visual de cada estabelecimento. Ou seja, cada estabelecimento cadastrado deve informar quais as cores principais da sua logo e assim a página do catálogo se adapta e renderiza os elementos com base nas cores selecionadas.

### 2.1.1 Requisitos Funcionais

Com base na análise inicial, foram determinados os requisitos funcionais a seguir:

**RF01 - Consultar estabelecimentos:** Apresentar um campo de pesquisa, listando os estabelecimentos disponíveis para acesso e aplicando o filtro de consulta de acordo com os termos solicitados pelo usuário.

- Perfis com permissão [**Cliente**]

**RF02 - Acessar o catálogo de itens:** Ao selecionar um estabelecimento disponível, a plataforma deverá direcionar o usuário para o menu/catálogo/cardápio ativo para o estabelecimento selecionado. Exibindo todos os itens cadastrados de forma limpa e clara para seleção.

- Perfis com permissão [**Cliente**]

**RF03 - Selecionar item:** Ao selecionar um item do catálogo a plataforma deverá adicioná-lo a um carrinho de compras virtual, possibilitando a remoção e identificação na lista, calculando assim um total inicial do pedido.

- Perfis com permissão [**Cliente**]

**RF04 - Visualizar informações do item:** O item deverá conter informações de título, descrição e preço obrigatoriamente. Porém também será possível visualizar a quantidade em unidades que a compra do item resulta (10 unidades/compra), peso/litros, quantidade de pessoas que o item serve, uma imagem também poderá ser anexada.

- Perfis com permissão [**Cliente, Comerciante**]

**RF05 - Categorias:** É possível categorizar o catálogo, adicionando ou removendo categorias, tais como Bebidas, Pratos Executivos, Sobremesas dentre outros.

- Perfis com permissão [**Comerciante**]

**RF06 - Finalizar pedido:** Iniciar o processo para finalizar o pedido, solicitando informações do cliente, como: Identificação, Endereço e Forma de Pagamento. Após preencher os campos obrigatórios, o sistema apresentará um resumo do pedido e assim o cliente poderá finalizar.

- Perfis com permissão [**Cliente**]

**RF07 - Definir perfis e permissões:** Cada usuário possui um perfil com uma lista de permissões associadas.

- **Administrador** acessa todo o sistema, gerenciando usuários e estabelecimentos.
- **Comerciante** acessa apenas o seu estabelecimento, podendo gerenciá-lo de acordo com a sua necessidade.
- **Cliente** acessa apenas o catálogo do estabelecimento, podendo efetuar um pedido na plataforma.

**RF08 - Gerenciar estabelecimento:** O usuário poderá acessar via login (e-mail e senha) o painel de gerenciamento do seu estabelecimento.

- Perfis com permissão [**Comerciante**]

**RF09 - Gerenciar Catálogos:** No painel principal, o usuário terá acesso à lista de catálogos existentes. Para cada catálogo listado, o usuário deverá acessar a lista de itens do mesmo, podendo adicionar um novo item ou alterar informações individuais.

- Perfis com permissão [**Comerciante**]

**RF10 - Gerenciar Item:** Para cada catálogo, deverá existir a possibilidade de adição/edição de um item, respeitando os atributos mencionados em **RF04 Visualizar informações do item.**

- Perfis com permissão [**Comerciante**]

**RF11 - Gerenciar horário:** O usuário poderá alterar e adicionar horários de funcionamento do estabelecimento.

- Perfis com permissão [**Comerciante**]

**RF12 - Gerenciar meios de pagamento:** O usuário poderá alterar e adicionar os meios de pagamento disponíveis para o estabelecimento, escolhendo entre Cartão de Crédito, Cartão de Débito, Pix ou Dinheiro em espécie.

- Perfis com permissão [**Comerciante**]

**RF13 - Cadastrar informações do item:** Deverá ser possível inserir informações de título, descrição e preço. Também será possível informar a quantidade em unidades que a compra do item resulta (10 unidades/compra), peso/litros por item, quantidade de pessoas que o item serve e uma imagem também poderá ser anexada.

- Perfis com permissão [**Comerciante**]

**RF14 - Cadastrar estabelecimento:** Deverá ser possível inserir informações de título, logotipo, endereço e cores do estabelecimento.

- Perfis com permissão [**Administrador**]

### 2.1.2 Requisitos Não-Funcionais

Quanto aos requisitos não-funcionais, foram definidos os seguintes:

**RNF01 - Disponibilidade:** A aplicação deve estar disponível para os usuários 24 horas por dia, 7 dias por semana, com um tempo de inatividade planejado não superior a 2 horas por mês.

**RNF02 - Usabilidade:** A interface do usuário deve ser projetada de forma intuitiva, seguindo os princípios de usabilidade, para facilitar a navegação e interação. A aplicação deve ser responsiva e se adaptar a diferentes tamanhos de tela, garantindo uma experiência consistente em dispositivos móveis.

**RNF03 - Manutenção:** A aplicação deve ser projetada para permitir atualizações frequentes sem causar interrupção no serviço.

### 3 IMPLEMENTAÇÃO DO BACK-END

O software foi desenvolvido utilizando o padrão arquitetural MVC (*Model-View-Controller*), que é uma abordagem comum para separar as responsabilidades e camadas de uma aplicação web.

#### 3.1 Linguagens

A linguagem escolhida para o desenvolvimento foi o JavaScript<sup>7</sup>, com o auxílio do *superset* TypeScript<sup>8</sup>, que possibilita a definição de domínios para os atributos e facilita a organização e construção das camadas da aplicação (Mozilla, 2023). Para habilitar o uso do JavaScript no lado do servidor, optamos pela plataforma Node.js<sup>9</sup>, juntamente com o framework NestJS<sup>10</sup>. O NestJS complementa o Node.js, fornecendo uma estrutura sólida e recursos adicionais para facilitar o desenvolvimento de aplicativos robustos (NestJS, 2023).

A escolha do Node.js como plataforma e do JavaScript como linguagem ocorreu devido à sua capacidade de reduzir custos ao disponibilizar aplicações sem a necessidade de um servidor robusto e, conseqüentemente, mais caro.

Adotando uma abordagem monolítica, todas as funcionalidades e o código do backend são agrupados em um único sistema, expostos por meio de endpoints (URIs) para a web seguindo o padrão REST (Representational State Transfer), que define o uso e restrições da API disponibilizada, assim como também utiliza o protocolo HTTP para comunicação entre as partes do sistema. Para documentar a API optou-se por utilizar o Swagger<sup>11</sup>, que é uma ferramenta para criar, documentar e visualizar APIs, descrevendo seus endpoints (URIs), parâmetros, modelos de integração e exemplos de respostas, em caso de sucesso ou erro na requisição (Swagger, 2023).

---

<sup>7</sup> [JavaScript](https://developer.mozilla.org/en-US/docs/Web/JavaScript) - <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<sup>8</sup> [TypeScript](https://www.typescriptlang.org) - <https://www.typescriptlang.org>

<sup>9</sup> [NodeJS](https://nodejs.org) - <https://nodejs.org>

<sup>10</sup> [NestJS](https://nestjs.com) - <https://nestjs.com>

<sup>11</sup> [Swagger](https://swagger.io) - <https://swagger.io>

### 3.2 Persistência

O MySQL<sup>12</sup> foi escolhido como banco de dados devido às necessidades da aplicação, uma vez que existem muitos relacionamentos entre as entidades definidas. Optou-se por um banco de dados relacional, em vez de um banco não relacional como o MongoDB<sup>13</sup>, porque um banco de dados relacional bem estruturado assegura a integridade dos dados e facilita a recuperação de informações da base de dados por meio de consultas SQL (AWS, 2023). Isso permite uma melhor manipulação e organização dos dados, sem perder eficiência.

Para a manipulação da conexão e o gerenciamento das tabelas no banco de dados, utilizou-se o TypeORM (Object Relational Mapping). Essa ferramenta nos permite marcar os atributos das classes (TypeScript) que representam as entidades do sistema, facilitando a interação com o banco de dados. Além de oferecer um conjunto de recursos que simplificam as operações de persistência de dados e mapeamento objeto-relacional, tornando o desenvolvimento mais eficiente e organizado.

A manipulação de imagens na aplicação é realizada por meio do Amazon S3, um serviço que permite o armazenamento e a recuperação de imagens de maneira ágil e eficaz (Amazon, 2023). Os servidores da Amazon estão localizados no Brasil, o que reduz significativamente o tempo de resposta em comparação com servidores no exterior. Toda a integração foi efetuada utilizando o SDK da AWS para enviar e atualizar imagens no repositório do S3, proporcionando ainda uma opção de baixo custo para armazenamento de dados.

### 3.3 Segurança

Para a segurança, o JSON Web Token<sup>14</sup> (JWT) foi utilizado em todas as requisições que necessitam de validação do perfil de usuário logado e também para proteger recursos da aplicação que manipulam informações do estabelecimento, catálogos, itens e qualquer informação sensível. Ao utilizar JWT ao solicitar o login com e-mail e senha do usuário cadastrado, a aplicação gera um token que pode ser

---

<sup>12</sup> [MySQL](https://www.mysql.com) - <https://www.mysql.com>

<sup>13</sup> [MongoDB](https://www.mongodb.com) - <https://www.mongodb.com>

<sup>14</sup> [JSON Web Token](https://jwt.io) - <https://jwt.io>

decodificado para recuperar informações da sessão. Esse token é armazenado nas aplicações clientes (frontend) e enviado em cada requisição para validar a sessão.

### 3.4 Disponibilidade

Quanto à disponibilização da aplicação, primeiramente foram instanciados três servidores de baixa escala na Digital Ocean<sup>15</sup> utilizando um serviço Droplet, custando aproximadamente \$7 (~R\$34,56) na cotação atual. Cada servidor possui instalado o sistema operacional Ubuntu Server<sup>16</sup>, que é uma distribuição Linux amplamente utilizada, e o Docker<sup>17</sup> que é uma plataforma de virtualização leve que permite empacotar aplicativos e suas dependências em contêineres isolados, facilitando a implantação e execução das aplicações em diferentes ambientes.

O primeiro servidor é responsável por dois containers Docker com instâncias de um repositório NPM<sup>18</sup> e um registro para imagens do Docker. Aqui podemos gerenciar pacotes Node e compartilhar entre as aplicações arquivos comuns para o desenvolvimento. E com o registro de imagens do docker, podemos armazenar e disponibilizar as futuras construções de cada aplicação para executá-las em outro servidor utilizando Docker.

O segundo servidor é responsável por um container Docker que irá disponibilizar o frontend da aplicação virtualizando um servidor Nginx<sup>19</sup>. O Nginx é amplamente utilizado para disponibilizar aplicações web frontend e pode lidar com alto tráfego, balanceamento de carga e fornecer recursos avançados de cache e segurança.

O terceiro e último servidor é responsável por virtualizar dois containers docker contendo imagens do MySQL e do Node para disponibilizar a API da aplicação.

Todos os servidores são gerenciados utilizando o Docker como principal ferramenta para controlar os ambientes da aplicação. Além disso, é necessário configurar as portas e os endereços IP na Digital Ocean para permitir a exposição dos servidores na Internet.

---

<sup>15</sup> [Digital Ocean](https://www.digitalocean.com) - <https://www.digitalocean.com>

<sup>16</sup> [Ubuntu Server](https://ubuntu.com/download/server) - <https://ubuntu.com/download/server>

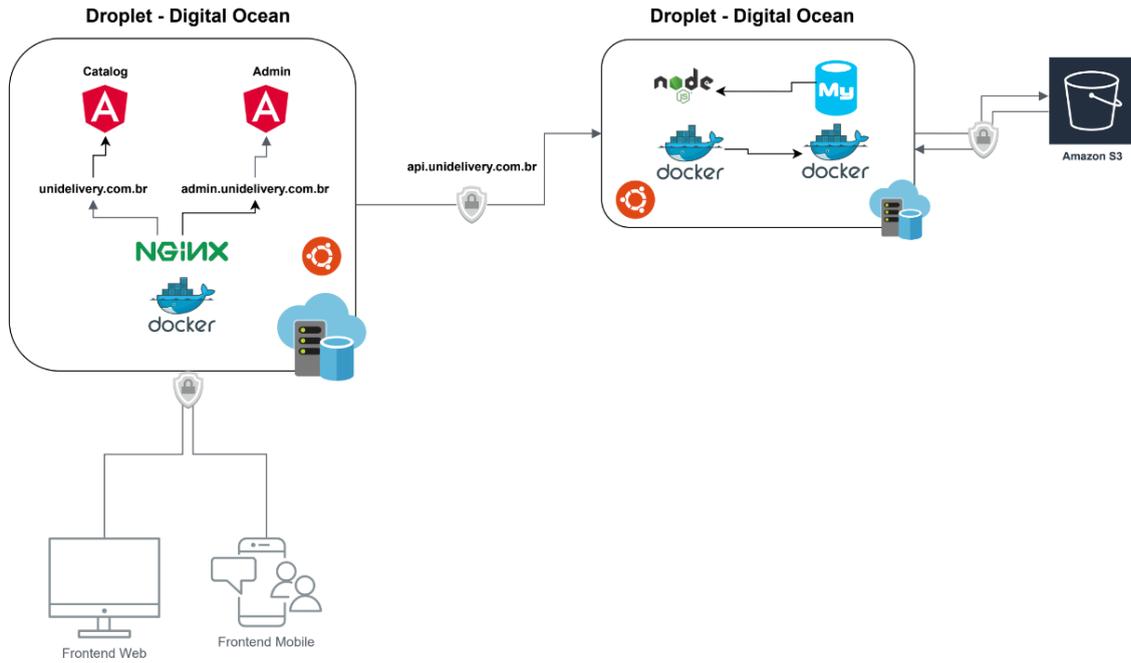
<sup>17</sup> [Doker](https://www.docker.com) - <https://www.docker.com>

<sup>18</sup> [NPM](https://www.npmjs.com/) - <https://www.npmjs.com/>

<sup>19</sup> [NGINX](https://nginx.org) - <https://nginx.org>

A visão geral da arquitetura aqui apresentada pode ser vista na Figura 1 a seguir.

**Figura 1** - Arquitetura do sistema



**Fonte:** elaborado pelo autor (2023)

## 4 IMPLEMENTAÇÃO DO FRONT-END

No frontend, a aplicação foi desenvolvida utilizando o Angular<sup>20</sup>, um framework SPA (*Single Page Application*) JavaScript popular para construir interfaces de usuário interativas. O Angular fornece recursos para criar aplicativos web complexos, como vinculação de dados, gerenciamento de estado e roteamento.

O fluxo da aplicação é simples e prático: o cliente acessa um link personalizado com o nome do estabelecimento no final da URL, utilizando o navegador do seu smartphone ou computador. Esse link aponta para um domínio na Internet, que, por sua vez, redireciona para o endereço IP do servidor responsável por fornecer a aplicação Angular. Ao executar a aplicação, o sistema identifica o catálogo disponível para aquele estabelecimento e exibe na tela informações e uma lista dos itens disponíveis para venda.

O usuário seleciona os itens desejados, preenche os dados de contato, informa o endereço e a forma de pagamento que serão utilizados para a entrega do produto. Após preencher todos os dados necessários, o sistema gera um link que redireciona para o aplicativo WhatsApp do cliente. Esse link é automaticamente preenchido com todas as informações do pedido para o estabelecimento, tornando o processo ágil e conveniente. Com isso, cabe apenas ao cliente enviar a mensagem, ficando responsável pelo envio final e facilitando o contato direto com o estabelecimento para concluir o processo de pedido.

Toda a comunicação para recuperar informações do estabelecimento, itens do catálogo, imagens e taxas de entrega do sistema, é feito através da internet e protocolo HTTPS para comunicação com a API do back-end.

Toda a prototipagem das telas foi elaborada utilizando a ferramenta Figma<sup>21</sup>, pois facilita a definição das cores e disposição dos elementos na tela, além de permitir a simulação do fluxo do usuário.

### 4.1 Funcionalidades

As telas produzidas para atender aos requisitos inicialmente definidos são apresentadas a seguir de acordo com o perfil do usuário.

---

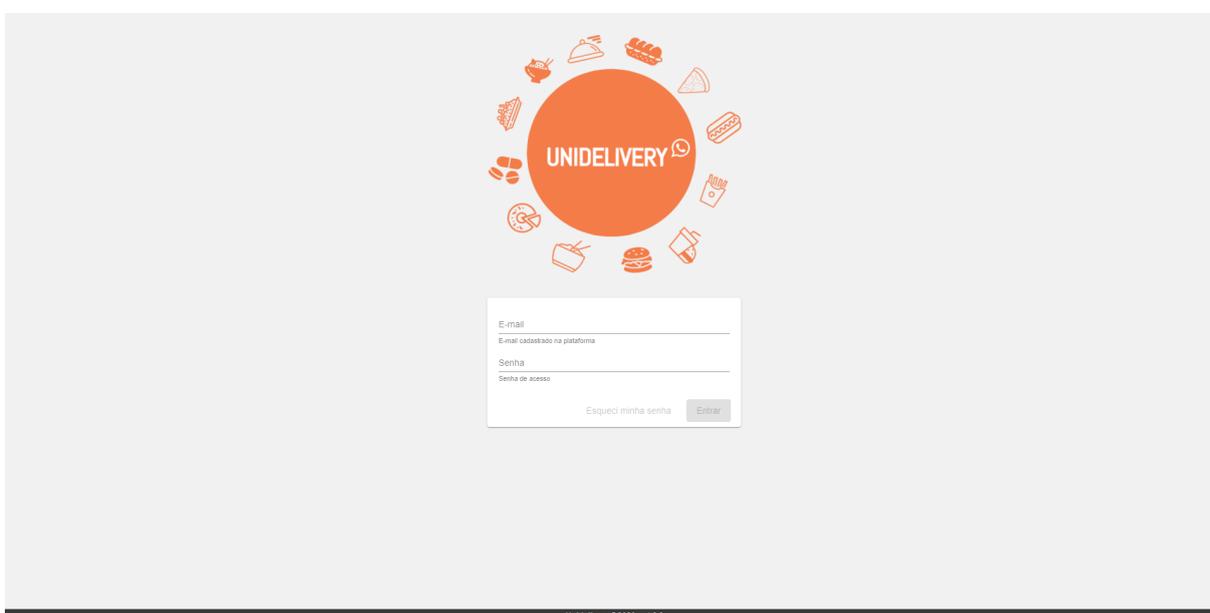
<sup>20</sup> [Angular](https://angular.io) - <https://angular.io>

<sup>21</sup> [Figma](https://www.figma.com) - <https://www.figma.com>

### 4.1.1 Telas para o perfil Comerciante

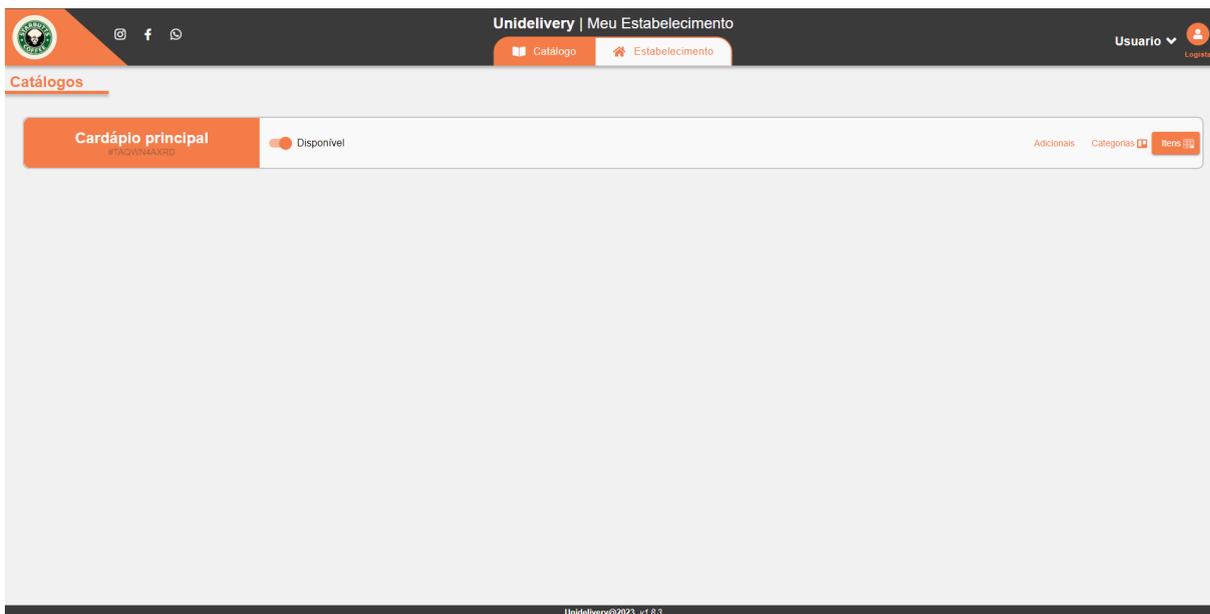
Para administrar um estabelecimento e alterar informações dos itens, o dono do estabelecimento, usando seu e-mail e senha cadastrado pelo administrador do sistema (via banco de dados), deve acessar a página para efetuar login e gerenciar as informações. A tela inicial para realização do login pode ser vista na Figura 2 a seguir.

**Figura 2** - Tela de login



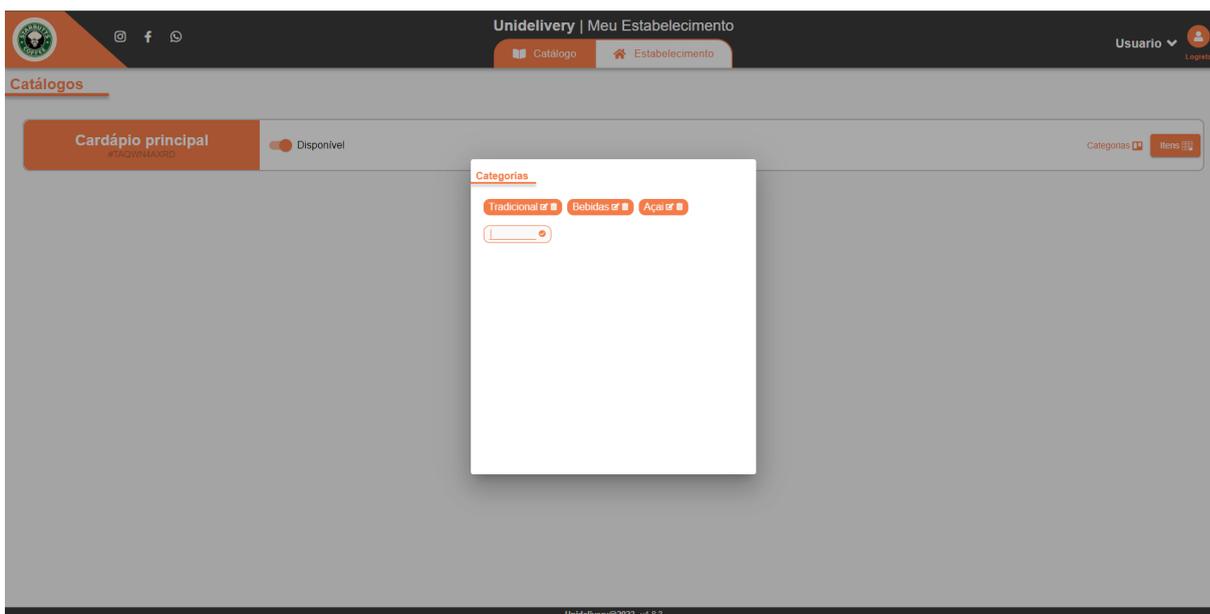
**Fonte:** elaborado pelo autor (2023)

Após efetuar login o usuário é direcionado para tela de catálogos (Figura 3), listando todos os catálogos que ele tem cadastrado no sistema e algumas informações do seu estabelecimento.

**Figura 3 - Tela de catálogos**

Fonte: elaborado pelo autor (2023)

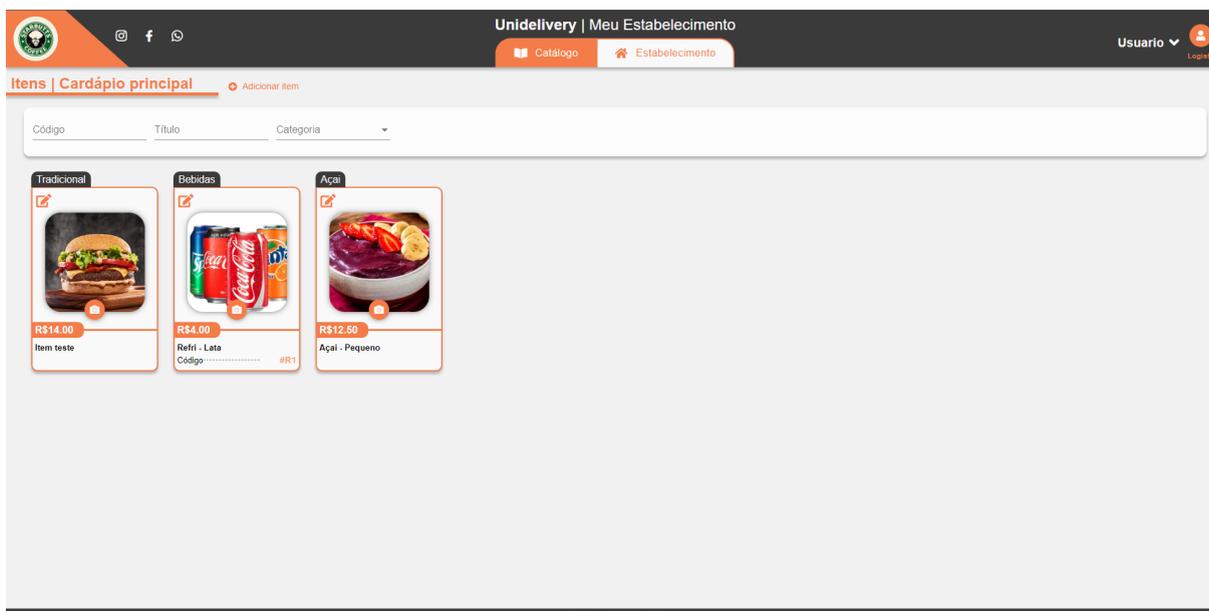
Na tela de Catálogos, ao clicar no botão Categorias, a tela para gerenciar as categorias disponíveis no catálogo irá aparecer (Figura 4), e o usuário poderá editar ou remover uma categoria existente, como também cadastrar uma nova categoria.

**Figura 4 - Tela de categorias**

Fonte: elaborado pelo autor (2023)

Ainda na tela de Catálogos, ao clicar no botão Itens, a tela para gerenciar os itens disponíveis no catálogo irá aparecer (Figura 5), e o usuário poderá editar um item existente, como também cadastrar um novo item.

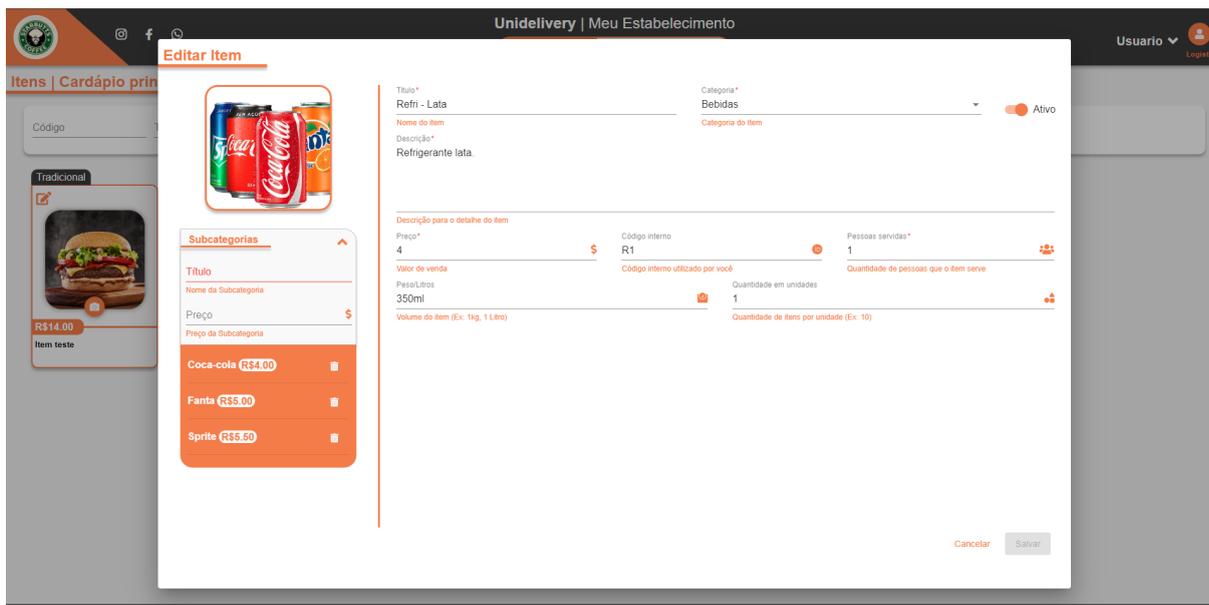
**Figura 5** - Tela de itens do catálogo



**Fonte:** elaborado pelo autor (2023)

Na tela de itens do catálogo, ao clicar no ícone para editar um item, a tela de edição correspondente será exibida, conforme mostrado na Figura 6. O usuário comerciante terá a possibilidade de modificar as informações do item, permitindo atualizar o catálogo com os dados mais recentes e precisos relacionados ao produto em questão. Essa funcionalidade oferece ao usuário o controle total sobre as informações exibidas, garantindo que o catálogo esteja sempre atualizado e refletindo as mudanças desejadas pelo estabelecimento.

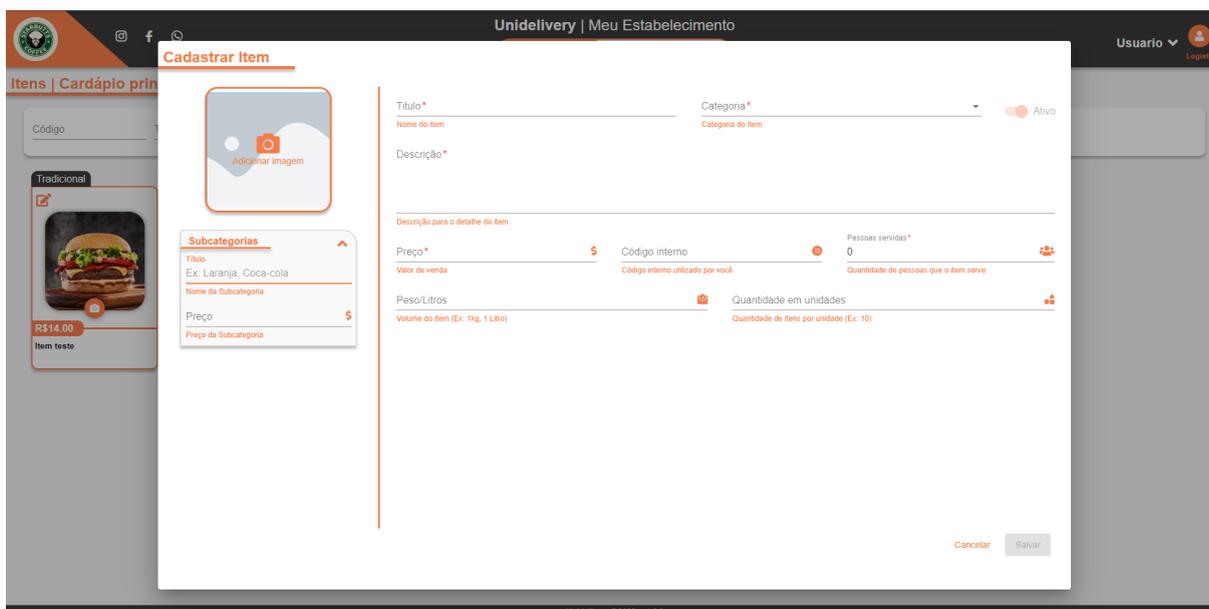
**Figura 6 - Tela de edição de um item**



Fonte: elaborado pelo autor (2023)

Ao clicar no botão para cadastrar um item, a tela de cadastro será exibida conforme mostrado na Figura 7. Nessa tela, o item estará em branco, aguardando que o comerciante preencha todas as informações necessárias para disponibilizá-lo no catálogo. Essa funcionalidade permite ao usuário adicionar novos itens de forma simples e intuitiva, garantindo que o catálogo esteja sempre atualizado com os produtos mais recentes do estabelecimento.

**Figura 7 - Tela de cadastro de item**



Fonte: elaborado pelo autor (2023)

A partir de qualquer das telas também é possível clicar na aba Estabelecimento, no topo da página, e o comerciante será direcionado para a tela de informações e configurações do seu estabelecimento.

**Figura 8** - Tela de informações do estabelecimento

The screenshot shows the 'Meu Estabelecimento' page in the Unidelivery system. The page is divided into several sections:

- Header:** Unidelivery | Meu Estabelecimento, with navigation tabs for 'Catálogo' and 'Estabelecimento'.
- Informações:**
  - CNPJ: 42.787.579/0012-1, Hash: #meuestabelecimento
  - Nome: Meu Estabelecimento
  - CEP: 58.135-000, UF: PB, Cidade: Esperança, Bairro: Centro, Logradouro: Sete de Setembro, Nº: 35
- Plano Unidelivery:**
  - Título: Plano Inicial, Data adesão: 21, maio 2023, Data expiração: 21, maio 2024
  - Descrição: Plano com imagens, Ativo: Sim
  - Catálogos: 2, Itens/Catálogo: 150, Categorias: 15, Subcategorias: 3
- Meios de Pagamento:**
  - Cartão de Crédito (checked)
  - Cartão de Débito (checked)
  - Dinheiro (checked)
  - Pix (unchecked)

Fonte: elaborado pelo autor (2023)

Por fim, é possível ao comerciante acessar a tela para gerenciar o horário do estabelecimento a partir da tela mostrada na Figura 9.

**Figura 9** - Tela de gerenciamento de horários de funcionamento

The screenshot shows the 'Meu Estabelecimento' page with the 'Horário de Funcionamento' section active. The page layout is similar to Figure 8, but the 'Meios de Pagamento' section is replaced by a schedule grid.

**Horário de Funcionamento:**

Domingo	Segunda-feira	Terça-feira	Quarta-feira	Quinta-feira	Sexta-feira	Sábado
00	00	00	00	00	00	00
01	01	01	01	01	01	01
02	02	02	02	02	02	02
03	03	03	03	03	03	03
04	04	04	04	04	04	04
05	05	05	05	05	05	05
06	06	06	06	06	06	06
07	07	07	07	07	07	07
08	08	08	08	08	08	08
09	09	09	09	09	09	09
10	10	10	10	10	10	10
11	11	11	11	11	11	11
12	12	12	12	12	12	12
13	13	13	13	13	13	13
14	14	14	14	14	14	14
15	15	15	15	15	15	15
16	16	16	16	16	16	16
17	17	17	17	17	17	17
18	18	18	18	18	18	18
19	19	19	19	19	19	19
20	20	20	20	20	20	20
21	21	21	21	21	21	21
22	22	22	22	22	22	22
23	23	23	23	23	23	23

The grid shows a 24-hour cycle for each day of the week. The 'Domingo' column has a red bar from 00:00 to 23:00, indicating the operating hours. The 'Segunda-feira' column has a red bar from 00:00 to 00:00, indicating it is closed. The other days (Terça-feira to Sábado) are currently empty, indicating they are also closed.

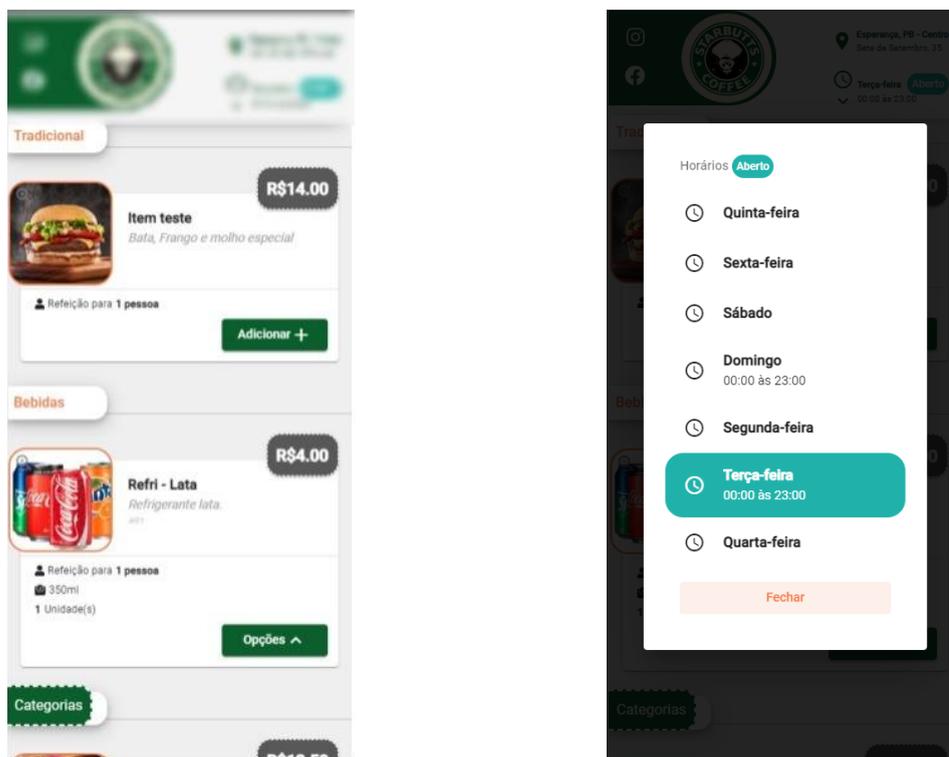
Fonte: elaborado pelo autor (2023)

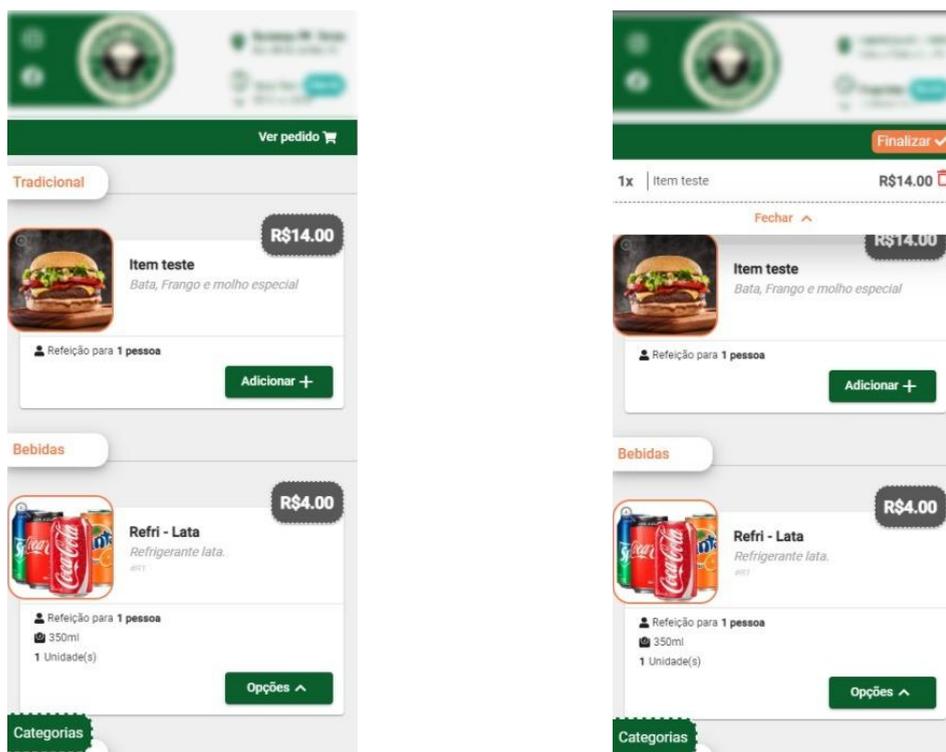
#### 4.1.2 Telas para o perfil Cliente

O cliente poderá acessar o catálogo disponível online através do navegador web do seu smartphone utilizando a URL [http://unidelivery.com.br/URL\\_DO\\_ESTABELECIMENTO](http://unidelivery.com.br/URL_DO_ESTABELECIMENTO). Graças à responsividade da aplicação web, a interface se adapta automaticamente ao dispositivo do usuário, garantindo uma exibição organizada e otimizada dos itens na tela. Dessa forma, o cliente terá uma experiência de navegação agradável e eficiente, independentemente do tamanho da tela do seu dispositivo móvel.

A seguir, são apresentadas as telas destinadas aos usuários que acessam a plataforma por meio de dispositivos móveis. Na Figura 10, encontram-se as telas iniciais do catálogo, bem como a tela de exibição dos horários de funcionamento do estabelecimento. Quando um cliente seleciona um item, a aplicação automaticamente cria um carrinho de compras virtual, e o botão "Ver Pedido" é exibido. Ao selecioná-lo, o cliente pode expandir a lista, visualizando detalhes dos itens escolhidos, enquanto o botão "Finalizar" fica disponível para prosseguir com o pedido.

**Figura 10** - Tela de catálogo e horário de funcionamento





Fonte: elaborado pelo autor (2023)

Ao clicar no botão para finalizar o pedido, no canto superior direito da tela, a tela de preenchimento dos dados necessários será aberta e é permitido ao cliente inserir todas as informações requeridas e seguir um processo passo-a-passo até chegar ao resumo do pedido. Durante esse fluxo o cliente terá a oportunidade de fornecer informações para contato, endereço e meio de pagamento, conforme mostrado na Figura 11.

Uma vez que o primeiro pedido é concluído, os dados do cliente ficam armazenados na sessão do navegador, o que agiliza o processo para o próximo pedido realizado na plataforma. Essa funcionalidade torna a experiência do cliente mais conveniente, pois ele não precisa inserir repetidamente as mesmas informações em futuras compras, melhorando a eficiência e a satisfação geral do usuário.

Figura 11 - Tela para informar dados e o resumo do pedido

The figure displays four screenshots of a mobile application interface, arranged in a 2x2 grid, showing the process of entering and confirming order details.

**Top Left: Informações**  
This screen is titled "Informações" and includes a "Cancelar" button with a close icon. It features a progress bar with four steps: the first is active (orange), and the others are greyed out. Below the progress bar, there are input fields for "Nome Cliente", a masked "WhatsApp" number, and a text area for "Observações". A "Confirmar" button with a checkmark is at the bottom.

**Top Right: Endereço**  
This screen is titled "Endereço" and includes a "Cancelar" button with a close icon. It features a progress bar with four steps: the first is active (orange), and the others are greyed out. Below the progress bar, there are dropdown menus for "Cidade\*" (Esperança) and "Bairro\*" (Centro). It also shows "Taxa de entrega: R\$3.00", "Rua: Manoel Henrique", "Número: 795", and "Complemento: 1 Andar". A "Ponto de Referência" field contains "Proximo a estação". A "Confirmar endereço" button with a checkmark is at the bottom, along with a "Voltar" button.

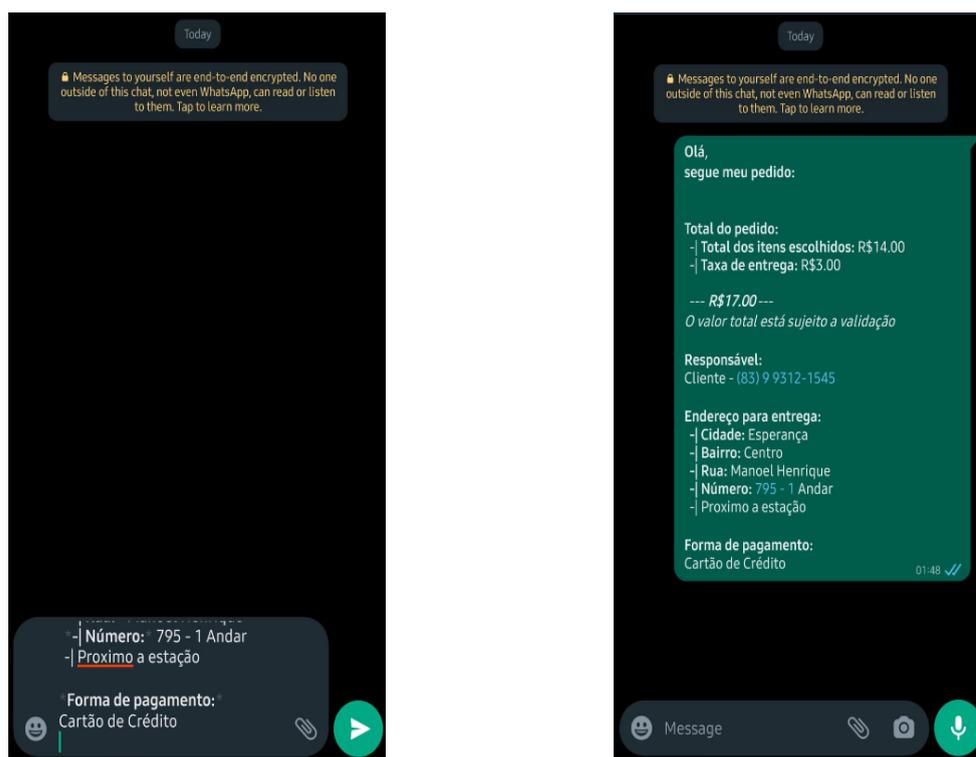
**Bottom Left: Pagamento**  
This screen is titled "Pagamento" and includes a "Cancelar" button with a close icon. It features a progress bar with four steps: the first two are active (orange), and the last two are greyed out. Below the progress bar, a dashed red box contains a summary: "Pedido R\$14.00", "Taxa de entrega R\$3.00", and "Total | R\$17.00". Below this, there is a "Pagamento" dropdown menu set to "Cartão de Crédito". A "Confirmar pagamento" button with a checkmark is at the bottom, along with a "Voltar" button.

**Bottom Right: Resumo**  
This screen is titled "Resumo" and includes a "Cancelar" button with a close icon. It features a progress bar with four steps: the first three are active (orange), and the last one is greyed out. Below the progress bar, it shows "Responsável:" with a masked name, "Cartão de Crédito", and address details: "Cidade: Esperança", "Bairro: Centro", "Rua: Manoel Henrique", and "Número: 795 - 1 Andar". It also shows "Ponto de referência: Proximo a estação". A list of items includes "1x Item teste R\$14.00". A dashed red box at the bottom contains the same summary as the "Pagamento" screen: "Pedido R\$14.00", "Taxa de entrega R\$3.00", and "Total | R\$17.00". A "Confirmar" button is at the bottom, along with a "Voltar" button.

Fonte: elaborado pelo autor (2023)

Ao clicar em "Confirmar" na tela de Resumo do pedido, um link direto para o WhatsApp é criado, solicitando que o navegador abra o aplicativo WhatsApp do cliente. Devido à natureza específica do link para o WhatsApp, o dispositivo reconhece-o e abre o aplicativo do WhatsApp com todos os dados do pedido preenchidos automaticamente no campo de mensagem. Além disso, o contato do estabelecimento é exibido, permitindo ao cliente enviar o pedido com facilidade. Basta que o cliente envie a mensagem, concluindo o processo de forma simples e conveniente, conforme apresentado nas telas da Figura 12. A tela da esquerda mostra a abertura do WhatsApp do cliente com o resumo do pedido pronto para ser enviado, e a tela da direita já mostra o pedido realizado com a mensagem enviada para o estabelecimento.

**Figura 12** - Tela de Whatsapp do cliente



Fonte: elaborado pelo autor (2023)

## 5 CONCLUSÃO

O objetivo do Unidelivery é atender às necessidades simples dos estabelecimentos de menor porte, proporcionando uma forma organizada e facilitada para que o comerciante anote os pedidos e para que o cliente veja e escolha os itens de acordo com suas necessidades e disponha do tempo necessário para isso. A plataforma busca evitar que o cliente se perca na ordem de envio das informações ao estabelecimento, mantendo uma sequência clara: escolher os itens, preencher os dados de contato, endereço e forma de pagamento, com a possibilidade de validar tudo antes de enviar o pedido final.

Após a implementação do Unidelivery conforme aqui relatado, foram recebidos diversos feedbacks do estabelecimento. Alguns clientes, por exemplo, continuaram digitando e solicitando pedidos via mensagem de texto no WhatsApp. Porém, a maioria dos clientes aprovou a plataforma, encontrando facilidade para navegar e escolher seus itens, o que os levou a abandonar o envio de mensagens para o WhatsApp do estabelecimento. Em vez disso, passaram a fazer seus pedidos diretamente pelo Unidelivery, o que permitiu ao comerciante receber o pedido pronto como a primeira interação com o cliente.

A plataforma ainda requer alguns ajustes para melhorar sua funcionalidade, tais como o gerenciamento dos pontos de entrega e seus respectivos valores, além do cadastro de promoções e códigos promocionais para aumentar a aderência do público. Entretanto, a principal melhoria ocorreu um ano após o lançamento, com a implementação de um painel para o gerenciamento de pedidos. Esse painel permitiu que o cliente não precisasse mais voltar ao WhatsApp para enviar uma mensagem de texto com o pedido; agora, ele pode enviá-lo diretamente pelo Unidelivery ao clicar em finalizar o pedido. Por sua vez, o estabelecimento recebe uma notificação e pode visualizar todas as informações do pedido, como também todos os cálculos de itens e taxa de entrega. Essa melhoria se mostrou altamente eficaz, reduzindo significativamente a incidência de pedidos fraudulentos feitos por clientes mal-intencionados, que costumavam manipular a mensagem de texto gerada antes de enviá-la ao estabelecimento através do aplicativo WhatsApp.

Esse processo adicional exigia do comerciante a validação dos valores dos itens e do total do pedido antes de aceitá-lo. Com o painel, esse esforço extra foi minimizado, tornando o processo mais eficiente e evitando possíveis erros. Para a

construção do painel de pedidos, foi necessário implementar o WebSocket e estabelecer uma comunicação eficaz entre a API e o estabelecimento, permitindo a notificação imediata quando um novo pedido é enviado. A implementação transcorreu sem contratempos, uma vez que a aplicação suportou facilmente a conexão WebSocket, incluindo o mapeamento de novas tabelas no banco de dados e a integração suave das novas funcionalidades nas diversas camadas da aplicação.

Em uma conversa recente com o proprietário do estabelecimento que se ofereceu para testar a plataforma, recebemos um feedback positivo de que o sistema tem sido eficaz na gestão de pedidos. Até a presente data, 21 de agosto de 2023, o estabelecimento continua utilizando o Unidelivery em sua versão mais recente disponível.

## REFERÊNCIAS

AMAZON. Armazenamento de objetos construído para armazenar e recuperar qualquer volume de dados de qualquer local. **Amazon S3**, 2023.

Disponível em:

<[https://aws.amazon.com/pt/s3/?nc1=h\\_ls](https://aws.amazon.com/pt/s3/?nc1=h_ls)>. Acesso em: 31 de ago. de 2023.

AWS. Qual é a diferença entre o MongoDB e o MySQL?. **AWS**, 2023.

Disponível em:

<<https://aws.amazon.com/pt/compare/the-difference-between-mongodb-vs-mysql/>>.

Acesso em: 31 de ago. de 2023.

IFOOD. O que é o iFood? Conheça a história e a operação da empresa. **Ifood News**, 2023.

Disponível em:

<<https://www.news.ifood.com.br/o-que-e-o-ifood/>>. Acesso em: 21 de ago. de 2023.

LOUBAK, Ana Letícia. WhatsApp ultrapassa 2 bilhões de usuários em todo o mundo. **Techtudo**, 2020. Disponível em:

<<https://www.techtudo.com.br/noticias/2020/02/whatsapp-ultrapassa-2-bilhoes-de-usuarios-em-todo-o-mundo.ghtml>>. Acesso em: 21 de ago. de 2023.

MOZILLA. Javascript. Disponível em

<<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>. Acesso em: 21 de ago. de 2023.

NESTJS. A progressive Node.js framework for building efficient, reliable and scalable server-side applications. Disponível em: <<https://nestjs.com/>>. Acesso em: 21 de ago. de 2023.

SWAGGER. API Design. 2023 Disponível em:

<<https://swagger.io/solutions/api-design/>> Acesso em: 21 de ago. de 2023.