

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO  
RIO GRANDE DO SUL  
CAMPUS CANOAS  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS

FILIPE RITTER GHILARDI

**Pedido Expresso: Sistema web de recebimento de pedidos**

Canoas, 2025.

FILIPPE RITTER GHILARDI

**Pedido Expresso: Sistema web de recebimento de pedidos**

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas pelo Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul – Campus Canoas.

Professora Dra. Carla Odete Balestro Silva  
Orientadora

Canoas, 2025.



**Ministério da Educação**  
**Secretaria de Educação Profissional, Científica e Tecnológica**  
**Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul**  
**Campus Canoas**

**ATA DE DEFESA PÚBLICA DO TRABALHO DE CONCLUSÃO DE CURSO**

Ao 1º dia do mês de dezembro de 2025, às 18h horas, em sessão pública no Auditório 1 do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul, Campus Canoas, na presença da Banca Examinadora presidida pelo(a) Professor(a):

Prof. Dra. Carla Odete Balestro Silva

composta pelos examinadores:

1. Prof. Dr. Mariano Nicolao
2. Prof. Dr. Marcio Bigolin

o(a) aluno(a) Filipe Ritter Ghilardi apresentou o Trabalho de Conclusão de Curso intitulado: **Pedido Expresso: Sistema Web de Recebimento de Pedidos** como requisito curricular indispensável para a integralização do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Após reunião em sessão reservada, a Banca Examinadora deliberou e decidiu pela **APROVAÇÃO** do referido trabalho, divulgando o resultado formalmente ao aluno e demais presentes e eu, na qualidade de Presidente da Banca, lavrei a presente ata que será assinada por mim, pelos demais examinadores e pelo aluno.

Documento assinado digitalmente  
**gov.br** CARLA ODETE BALESTRO SILVA  
Data: 12/12/2025 10:00:07-0300  
Verifique em <https://validar.iti.gov.br>

---

Presidente da Banca Examinadora

Documento assinado digitalmente  
**gov.br** MARIANO NICOLAO  
Data: 14/12/2025 15:40:23-0300  
Verifique em <https://validar.iti.gov.br>

---

Examinador 01

Documento assinado digitalmente  
**gov.br** MARCIO BIGOLIN  
Data: 16/12/2025 17:31:17-0300  
Verifique em <https://validar.iti.gov.br>

---

Examinador 02

Documento assinado digitalmente  
**gov.br** FILIPE RITTER GHILARDI  
Data: 16/12/2025 18:45:39-0300  
Verifique em <https://validar.iti.gov.br>

---

Aluno

## **AGRADECIMENTOS**

Ao Grande Arquiteto do Universo, pela inspiração e força ao longo desta jornada.

Ao meu filho, Bernardo Ghilardi, pelo apoio constante e incentivo nos momentos decisivos.

À Professora Dra. Carla Odete Balestro Silva, pela orientação dedicada e pelas contribuições essenciais para este trabalho.

Aos professores do curso de Análise e Desenvolvimento de Sistemas, pelo conhecimento transmitido e pela formação recebida.

Aos amigos e colegas de turma, pelo companheirismo e apoio durante o percurso acadêmico.

Ao IFRS – Campus Canoas, pela oportunidade e pela experiência proporcionada.

## LISTA DE ABREVIATURAS E SIGLAS

API - *Application Programming Interface* (Interface de Programação de Aplicações)

CSS - *Cascading Style Sheets* (Folhas de Estilo em Cascata)

HTML - *Hypertext Markup Language* (Linguagem de Marcação de Hipertexto)

PDO - *PHP Data Objects* (Objetos de Dados PHP)

PHP - *Hypertext Preprocessor* (Pré-processador de Hipertexto PHP)

SQL - *Structured Query Language* (Linguagem de Consulta Estruturada)

UML - *Unified Modeling Language* (Linguagem de Modelagem Unificada)

CRUD - *Create, Read, Update, Delete* (Criar, Ler, Atualizar, Excluir)

## LISTA DE TABELAS

Tabela 1: Comparação de app tradicional com web .....	19
Tabela 2: Comparação preços de outras plataformas .....	20
Tabela 3: Requisitos .....	28

## LISTA DE FIGURAS

Figura 1: Dashboard de gestão do iFood para restaurantes parceiros .....	13
Figura 2: Fluxo de navegação no iFood: (A) Tela inicial, (B) Busca por restaurantes, (C) Cardápio do estabelecimento .....	14
Figura 3: Dashboard de gestão da Rappy para restaurantes parceiros.....	15
Figura 4: Fluxo de navegação na Rappy: (D) Tela de como era um restaurante.....	15
Figura 5: Dashboard de gestão da Uber Eats para restaurantes parceiros.....	16
Figura 6: Fluxo de navegação na Uber Eats: (F) Tela inicial, (G) Exemplo de restaurante, (H) Exemplo de restaurante.....	17
Figura 7: Dashboard de gestão do Anota AI para restaurantes parceiros .....	18
Figura 8: Fluxo de navegação no Anota AI: (I) Cardápio do estabelecimento e (J) Interface de um produto.....	18
Figura 9: Caso de uso.....	30
Figura 10: Diagrama de classe.....	31
Figura 11: Diagrama de fluxo.....	32
Figura 12: Diagrama de sequência.....	33
Figura 13: Modelo de Projeto (SCRUM) Pedido expresso.....	34
Figura 14: Entidade Relacionamento.....	36
Figura 15: Tela de login.....	37
Figura 16: Tela Dashboard.....	38

Figura 17: Tela de configuração.....	38
Figura 18: Tela gestão de pedidos.....	39
Figura 19: Tela para inserir o pedido direto no balcão.....	39
Figura 20: (K)Menu de navegação: (L) Cardápio de um produto e (M)Adicionar Carrinho.....	40
Figura 21: (N)Inserir número e nome (O) Forma de retirada e (P) Cadastro de endereço caso não tenha .....	41
Figura 22: (Q)Forma de pagamento, (R) Finalizando pedido e (S)Mensagem do cliente para o restaurante usando API.....	42
Figura 23: Tela gestão de produtos.....	43
Figura 24: Tela gestão de pedidos.....	43
Figura 25: Tela gestão de clientes .....	44
Figura 26: Tela gera relatório produtos.....	44
Figura 27: Estrutura de pastas do projeto.....	46
Figura 28: (T)Estrutura de pastas abertas 1,(U)Estrutura de pastas abertas 2 e (V)Estrutura de pastas abertas 3.....	47
Figura 29: Código de verificação de login.....	48
Figura 30: Código de cálculo de taxas de entrega.....	49
Figura 31: Integração com WhatsApp.....	49
Figura 32: Cadastro de funcionários.....	50
Figura 33: Lógica de status operacional.....	51

Figura 34: Teste CRUD funcionários.....	52
Figura 35: Testes cálculo formatos e validações.....	53
Figura 36: Teste de performance.....	54

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>7</b>
1.1 MOTIVAÇÃO	8
<b>2 METODOLOGIA</b>	<b>11</b>
<b>3 ESTADO DA ARTE</b>	<b>13</b>
<b>4 REFERENCIAL TEÓRICO</b>	<b>23</b>
4.1 PHP	23
4.2 HTML5: Estrutura Semântica e Funcionalidades Avançadas	24
4.3 CSS3: Estilização Moderna e Layout Responsivo	25
4.4 Bootstrap: Framework para Desenvolvimento Ágil e Responsivo	26
4.5 jQuery: Interatividade e Manipulação Simplificada	27
4.6 MariaDB no Ambiente de Desenvolvimento com XAMPP	28
4.7 Scrum: Metodologia ágil de desenvolvimento	29
<b>5 DESENVOLVIMENTO</b>	<b>30</b>
5.1 Requisitos	30
5.2 UML	31
5.3 Metodologia Scrum	35
5.4 ENTIDADE-RELACIONAMENTO	36
5.5 Front-end	38
5.6 Back-end	46
5.7 Testes	51
<b>6 CONCLUSÃO</b>	<b>54</b>
<b>REFERÊNCIAS</b>	<b>56</b>

## 1 INTRODUÇÃO

Segundo o diretor executivo da empresa Alphacode, Rafael Franco (Madureira, 2020), a aquisição de um aplicativo próprio pode gerar uma série de benefícios para qualquer negócio, desde a melhora no relacionamento com os clientes até a facilitação das vendas. Porém existem alguns fatores a serem avaliados, como por exemplo, a demanda dos clientes, se querem só um app (aplicativo) ou uma maneira de aumentar a interação com a empresa. Ainda para Franco(2020), outra questão importante é analisar os concorrentes que possuem um aplicativo para não ficar em desvantagem, a ferramenta será um complemento no negócio facilitando os serviços de entrega, reserva e compra que também servirá para o marketing da empresa, mas para isso é recomendável uma avaliação aos custos envolvidos e se realmente possui recursos financeiros para o investimento (Madureira, 2020).

Segundo o empresário da rede de restaurantes Giraffas, Carlos Guerra, em reportagem à UOL, o mercado de entrega de comida é quase um monopólio e assim restaurantes acabam pagando um valor de taxa absurdo para a empresa dominante e é necessário que surjam novas concorrências para a mudança do cenário (Desidério, 2023). O objetivo deste trabalho visa desenvolver um sistema web customizável para venda de comida online que poderá ser adaptado para diversos restaurantes com alteração de nome, endereço e o que for necessário para seu comércio e os clientes podem navegar entre categorias, produtos e o carrinho e fazer o pedido caso tenha cadastro ao contrário será direcionado para fazer o cadastro. O projeto utilizará uma metodologia e abordagem qualitativa de natureza exploratória e para identificação dos problemas a serem amenizados.

As tecnologias utilizadas para o desenvolvimento do sistema foi PHP (*Hypertext Preprocessor*), PDO (*Data Object*), MySQL (*Structured Query Language*), JavaScript, HTML (*Markup Language*) e o CSS (*Cascading Style Sheets*). Para agilizar a implementação, os frameworks Bootstrap e o JQuery também serão utilizados.

No capítulo a seguir, serão exploradas as motivações que levaram à concepção deste projeto, os objetivos a serem alcançados, a metodologia empregada, bem como, a apresentação da aplicação desenvolvida, as conclusões e as referências utilizadas.

## 1.1 MOTIVAÇÃO

Segundo Gustavo Penz, diretor de operações da rede de franquias GoCoffee em entrevista com a *Época Negócios* fala que sem um sistema para auxiliar nos desafios é difícil alcançar lucratividade (Penz, 2021). Natasha Paoli, gerente geral do Cuia Café, fala da importância de um bom sistema para facilitar a forma de administrar, a empresa já utiliza um sistema app delivery para facilitar as vendas (Tecnologia, 2021). Com estas informações é possível identificar os benefícios que um sistema traz para grandes e pequenas.

O diretor logístico do Mercado Livre, Luiz Augusto Vergueiro, em entrevista para o G1, comenta sobre a importância de uma estrutura de logística sustentada pela tecnologia, ressaltando que são as duas coisas convivendo em conjunto para poder fazer com que o pacote chegue com mais eficiência ao seu destino final (Black, 2023).

Madureira (2020), repórter da BBC, entrevistou donos de delivery para saber sobre as realidades de trabalhar com a plataforma do iFood<sup>1</sup>. Eles comentam sobre promoções sem fundamentos que a plataforma aplica aos clientes e sobre a taxa de 27% que precisam pagar sobre cada venda de comida, deixando seus lucros negativos.

Com a presente realidade, é necessário mais alternativas para atender micro empresas que entregam comida, onde empreendedores consigam se estruturar sem gastos excessivos com empresas gerenciadoras de pedidos/entrega.

A motivação para o desenvolvimento desse sistema é baseada nesses problemas, já que pequenas empresas não conseguem avançar e se manter no mercado de negócios, perdendo a motivação e interesse em continuar e a maioria até desistindo. Para amenizar este problema foi desenvolvido um sistema web customizável que poderá ser adaptado para diversos comerciantes de comida.

---

<sup>1</sup> Uma empresa brasileira de tecnologia especializada em delivery online, foi criada pelos sócios Patrick Sigrist, Eduardo Baer, Guilherme Bonifácio e Felipe Fioravante em 15 de maio de 2011, em São Paulo (SP)(O, 2023).

Nessa ferramenta poderá ser inserido nome, endereço, telefone e outras informações da loja manualmente, o sistema funcionará de uma forma automatizada para a venda, onde o cliente acessa um link sendo o mais recomendado porém não obrigatório utilizar o Whatsapp<sup>2</sup> *business* para aproveitar o *chatbot*<sup>3</sup> gratuito que retorna uma mensagem com o link do sistema, o cliente navega entre categorias e produtos, seleciona itens no carrinho e quando for finalizar a compra ele verifica o cadastro através do telefone, se possui finaliza a compra se não possui solicita o cadastro e antes da finalização do pedido, notifica o comerciante com informações da solicitação. Ele pode ser integrado a um app (aplicativo), de comunicação (WhatsApp), por ser popular, o sistema será responsivo e terá um login adm(administrador), para crud <sup>4</sup>de produtos e usuários, o restaurante que adquirir o sistema pagará um valor mensal(não excessivo), para a manutenção serão analisados de forma a fornecer aos usuários a menor taxa possível.

## 1.2 OBJETIVOS

### 1.1.1 Objetivo Geral

Desenvolver um sistema web responsivo de recebimento de pedidos para microempreendedores que vendem comida online.

### 1.1.2 Objetivos Específicos

- Pesquisar restaurantes que utilizam algum sistema para recebimento de pedidos.
- Entrevistar empreendedores para identificar as necessidades
- Fazer o levantamento dos requisitos e modelar o sistema.
- Documentar e implementar um sistema com baixo custo aos usuários.
- Realizar testes funcionais e de aceitação.

---

<sup>2</sup> A empresa WhatsApp foi fundada por Jan Koum e Brian Acton ao fim de 20 anos (entre os dois) a trabalhar na Yahoo A WhatsApp juntou-se ao Facebook em 2014, mas continua a funcionar como uma aplicação independente, com foco na criação de um serviço de mensagens que funciona de forma rápida e fiável em qualquer parte do mundo(Sobre, 2024).

<sup>3</sup> Um chatbot é um programa que simula conversas humanas, permitindo a interação com dispositivos digitais de forma natural. Eles podem variar desde sistemas simples até assistentes complexos que aprendem e se personalizam com o uso (ORACLE, 2024)

<sup>4</sup> CRUD é um acrônimo para as operações fundamentais em qualquer sistema que interage com um banco de dados: Create (Criar), Read (Ler), Update (Atualizar) e Delete (Apagar).

## **2 METODOLOGIA**

Segundo Fonseca (2002), a pesquisa é a principal forma de chegar mais próximo do entendimento de uma realidade a investigar, ela é o processo que permanece inacabado e a pesquisa científica é o resultado de um inquérito ou exame e que tinha o objetivo de resolver o problema. A metodologia deste trabalho é de abordagem qualitativa e de natureza exploratória para melhor compreensão das necessidades do público a ser atingido, ainda para Fonseca (2002), por se preocupar com aspectos que não permitem ser quantificados e focando na compreensão e explicação da dinâmica das relações sociais.

Para o levantamento dos requisitos foram realizadas entrevistas com donos de estabelecimentos de produção de alimentos na região metropolitana de Porto Alegre/RS, que também fornecem os produtos por delivery.

Para fundamentar o desenvolvimento do sistema proposto, foram realizadas entrevistas semiestruturadas com cinco empreendedores do setor alimentício, todos proprietários de estabelecimentos de pequeno porte na região metropolitana de Porto Alegre/RS. Os participantes possuem entre 5 e 12 anos de experiência no ramo, o que confere relevância e diversidade às informações obtidas. As entrevistas ocorreram entre os dias 18 e 22 de agosto de 2025, com o objetivo principal de compreender os fluxos operacionais atuais, identificar dificuldades enfrentadas no atendimento via delivery e levantar expectativas quanto à automação de processos.

Em respeito aos princípios éticos de pesquisa, as identidades dos entrevistados e de suas empresas foram preservadas mediante solicitação. Durante as entrevistas, foram explorados temas essenciais como o processo atual de recebimento e gestão de pedidos, dificuldades operacionais e gargalos em horários de pico, necessidade de atualização dinâmica do cardápio, controle de estoque e comunicação interna, identificação e histórico de

clientes, preferências quanto à automação de mensagens e notificações, além das expectativas em relação à interface e funcionalidades do sistema.

Os relatos obtidos forneceram subsídios valiosos para a etapa de especificação técnica, permitindo identificar requisitos críticos e oportunidades de melhoria que orientaram diretamente o desenvolvimento das funcionalidades do sistema, as quais serão detalhadas no capítulo de Desenvolvimento. Esta abordagem metodológica garantiu que a solução proposta estivesse alinhada com as reais necessidades do público-alvo, incorporando experiências práticas e demandas específicas do contexto operacional dos estabelecimentos alimentícios.

A partir das informações coletadas nas entrevistas, foram elaboradas a documentação definindo os requisitos funcionais e não funcionais, para documentação será utilizado *Unified Modeling Language (UML)*, para modelar casos de uso do sistema e assim ter uma melhor compreensão do seu funcionamento.

No desenvolvimento do sistema foi utilizado SCRUM, por ser baseada em princípios ágeis, ela irá dividir o projeto em ciclos curtos com o objetivo de aumentar a produtividade e possibilitar adaptações (Entenda,2023). Pelo fato do trabalho ser individual as sprints quinzenais será apresentado o andamento do desenvolvimento ao orientador e utilizadas para ajustes no andamento do projeto.

As tecnologias para desenvolver o sistema será PHP (*Hypertext Preprocessor*), PDO (*Data Object*), para facilitar a conexão com o banco MySQL (*Structured Query Language*) e JavaScript, além de marcações e estilos com o Hypertext HTML (*Markup Language*) e o CSS (*Cascading Style Sheets*), para facilitar e agilizar a implementação certos frameworks como o Bootstrap e o JQuery. Para a interação com o usuário foi utilizado uma API (*Application Programming Interface*) do Whatsapp que quando o usuário enviar uma mensagem para o restaurante o chat boot retorna uma mensagem com o link do sistema e uma mensagem instruindo o cliente a entrar no link, onde será direcionado ao sistema de web da loja.

Os testes automatizados foram feitos ao decorrer no desenvolvimento do software com o framework PHPUnit, para identificação de bugs e correção deles, ao decorrer do andamento será executado algumas simulações de funcionamento, o teste de aceitação foi realizado com com o dono de um estabelecimento com 3 anos no mercado de produção de alimentos na região metropolitana de Porto Alegre/RS, que também fornece produtos por delivery e aceitou em colaborar para uma breve aplicação no seu comércio.

### 3 ESTADO DA ARTE

Este capítulo tem como objetivo realizar um levantamento do estado da arte sobre sistemas de delivery, analisando softwares comerciais com propostas similares. Foi conduzida uma análise prática de plataformas existentes no mercado, examinando como elas implementam, em suas estruturas, módulos essenciais.

O mercado de delivery online passou por uma transformação significativa nos últimos anos, impulsionado pela digitalização de serviços e mudanças nos hábitos de consumo. A pandemia de COVID-19 acelerou esse processo, tornando o delivery uma prática comum e essencial para a sobrevivência de muitos negócios locais. O sucesso de plataformas como o iFood pode ser explicado por estratégias baseadas na economia da informação, como o custo de aprisionamento e o feedback positivo (SILVA, 2021).

Ainda conforme Silva (2021), Plataformas como iFood, Rappi e Uber Eats dominam o mercado brasileiro, oferecendo soluções completas de logística, pagamento e visibilidade. No entanto, essas plataformas apresentam desafios como altas taxas de comissão, dependência de aplicativos instaláveis e restrição no acesso aos dados dos clientes, o que pode limitar a autonomia dos comerciantes.

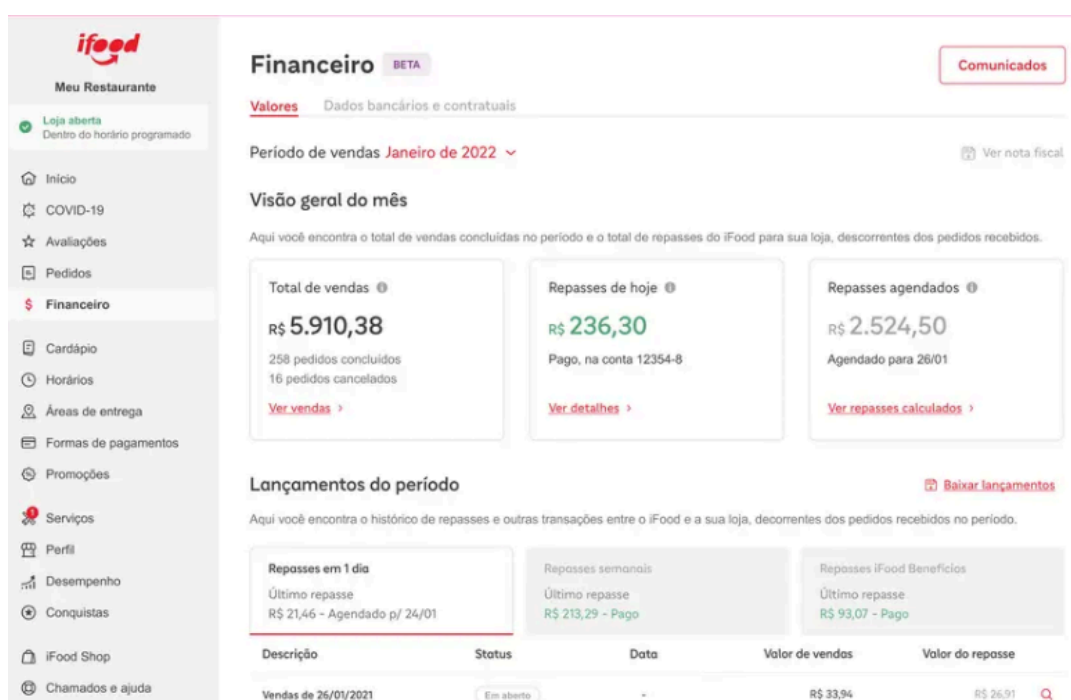
A reportagem publicada pelo G1 mostra como a 99Food se posiciona como um delivery com “alma brasileira”, valorizando tanto a diversidade gastronômica regional quanto a praticidade e economia que fazem parte do cotidiano nacional. Integrada ao ecossistema da DiDi, a plataforma aposta em tecnologia própria para otimizar entregas, reduzir custos e permitir que restaurantes mantenham os mesmos preços do balcão, além de adotar um modelo “ganha-ganha-ganha” que beneficia consumidores, estabelecimentos e entregadores. Com presença crescente em várias capitais e planos de expansão para 100 cidades até 2026, a 99Food reforça sua proposta de unir sabor, conveniência e acessibilidade, tornando-se reflexo da forma como o brasileiro se conecta e consome (G1, 2025).

Conforme descrito no site, o iFood caracteriza-se como uma plataforma digital de delivery que atua como intermediária entre consumidores, restaurantes, varejistas e entregadores, priorizando a simplicidade e a praticidade. Seu modelo de negócios não se restringe à entrega de refeições, abrangendo também serviços de delivery de mercado e

soluções financeiras, o que evidencia a capacidade de diversificação e escalabilidade inerentes a plataformas desse tipo. A empresa defende a tecnologia como ferramenta para otimizar a produtividade e a eficiência das relações entre todos os agentes do ecossistema, um princípio fundamental para o desenvolvimento de sistemas de delivery sustentáveis (IFOOD, 2025).

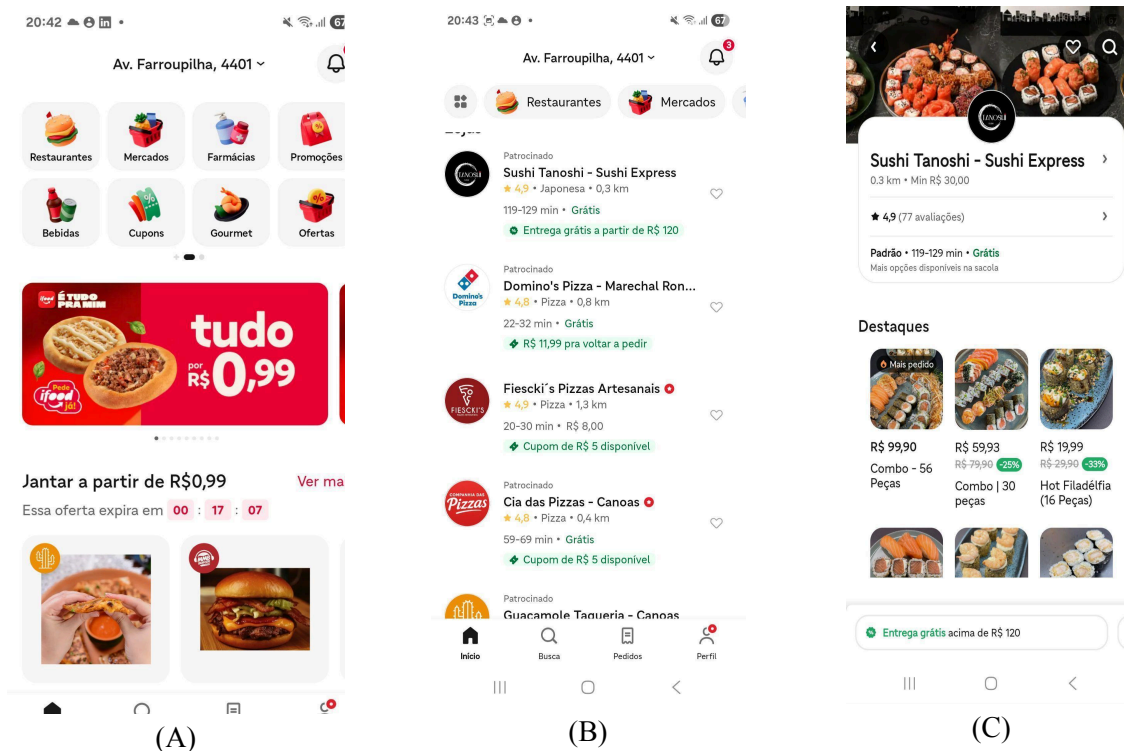
A Figura 1 ilustra o dashboard de gestão utilizado pelos restaurantes parceiros do iFood, para administrar pedidos e operações, enquanto a Figura 2 demonstra a interface de navegação disponível para os clientes finais, apresentando o cardápio digital de um estabelecimento da plataforma.

**Figura 1** - Dashboard de gestão do iFood para restaurantes parceiros.



Fonte: iFood (2025).

**Figura 2** - Fluxo de navegação no iFood: (A) Tela inicial, (B) Busca por restaurantes, (C) Cardápio do estabelecimento.



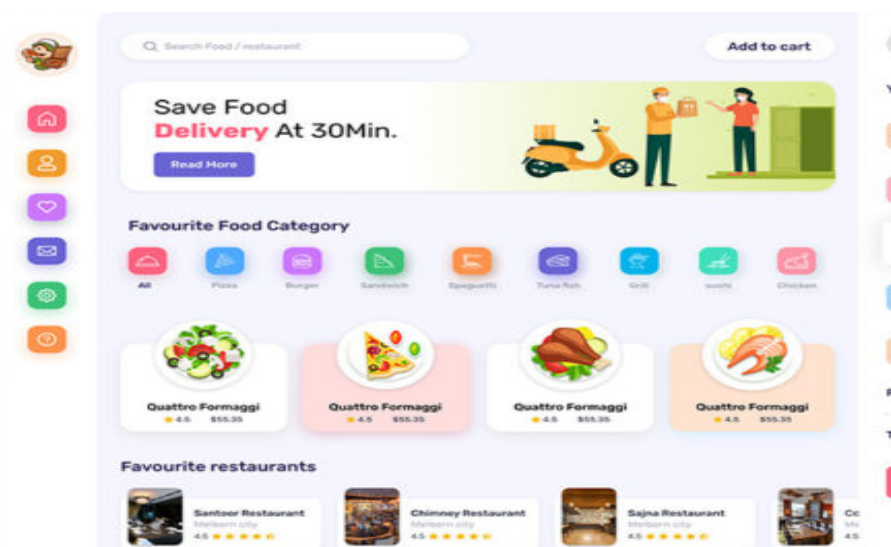
Fonte: iFood (2025).

De acordo com a Rappi (2025), a plataforma posiciona-se no mercado destacando sua capacidade de realizar entregas em menos de sessenta minutos, aliada a um conjunto de ferramentas de autogestão que permitem aos comerciantes controlar horários de funcionamento, gerenciar catálogos de produtos, monitorar vendas em tempo real e acessar estatísticas operacionais para embasar suas decisões. A empresa enfatiza ainda que os estabelecimentos podem criar campanhas personalizadas para aumentar a visibilidade dentro do aplicativo, promovendo assim o crescimento do negócio.

A Figura 3 ilustra o dashboard de gestão utilizado pelos restaurantes parceiros da Rappi, para administrar pedidos e operações, enquanto a Figura 4 demonstra a interface de navegação disponível para os clientes finais, apresentando um exemplo de como era o

cardápio digital de um estabelecimento da plataforma porém agora se encontra indisponível na região.

**Figura 3** - Dashboard de gestão da Rappy para restaurantes parceiros.

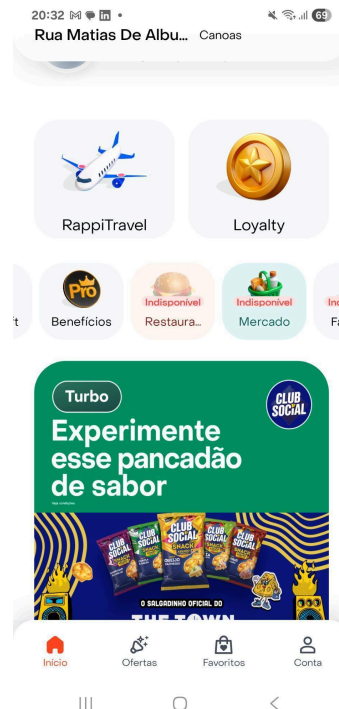


Fonte: Rappy (2025).

**Figura 4** - Fluxo de navegação na Rappy: (D) Tela de como era um restaurante, (D) Atualmente a opção de restaurantes está indisponível como mostra na imagem.



(D)



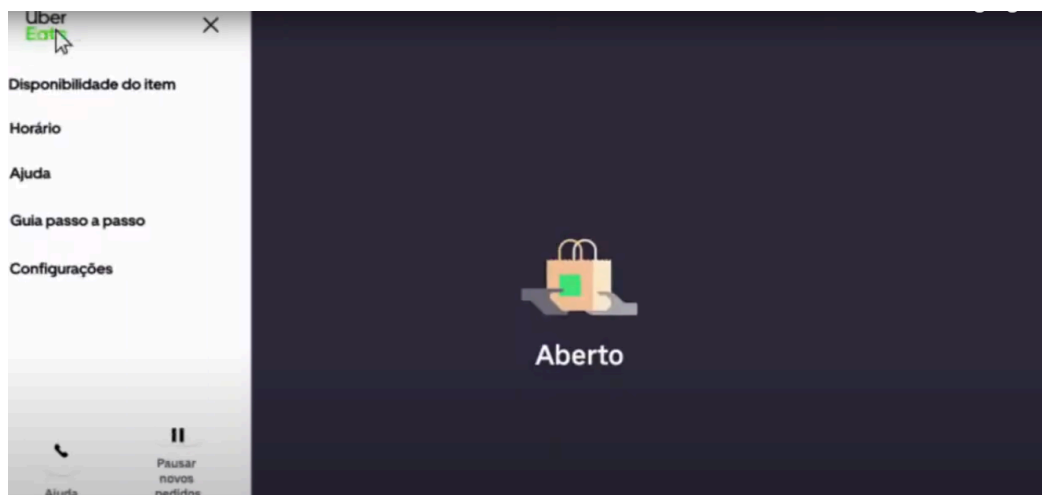
(E)

Fonte: Rappy (2025).

De acordo com reportagem do G1 (2022), a Uber Eats anunciou em 2022 a descontinuação do serviço de delivery de restaurantes no Brasil, mantendo apenas as operações de entrega de itens de supermercado e pacotes através da Cornershop by Uber e Uber Flash. A empresa afirmou que vai focar seus esforços em duas frentes principais: intermediação de entregas de supermercados e o serviço corporativo Uber Direct para entregas no mesmo dia.

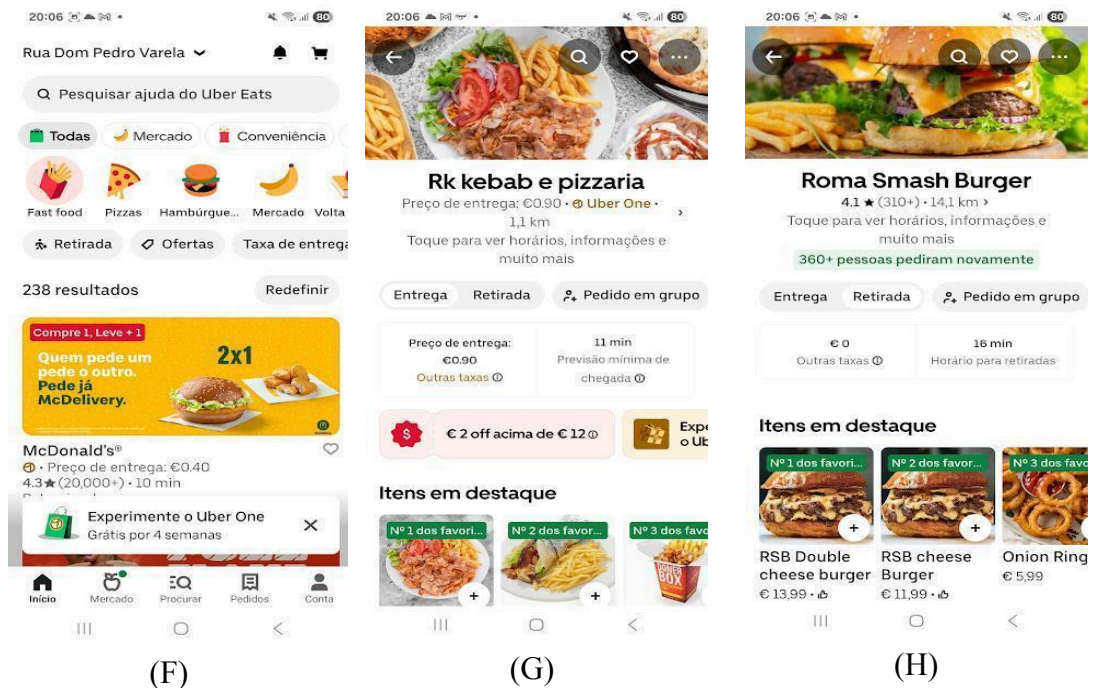
A Figura 6 ilustra o dashboard de gestão que era utilizado pelos restaurantes parceiros da Uber Eats, para administrar pedidos e operações, enquanto a Figura 7 demonstra a interface de navegação disponível para os clientes finais, apresentando um exemplo de como é o cardápio digital de um estabelecimento da plataforma fora do país, onde ainda continua suas atividades.

**Figura 5** - Dashboard de gestão da Uber Eats para restaurantes parceiros.



Fonte: Uber Eats (2025).

**Figura 6** - Fluxo de navegação na Uber Eats: (F) Tela inicial, (G) Exemplo de restaurante, (H) Exemplo de restaurante.

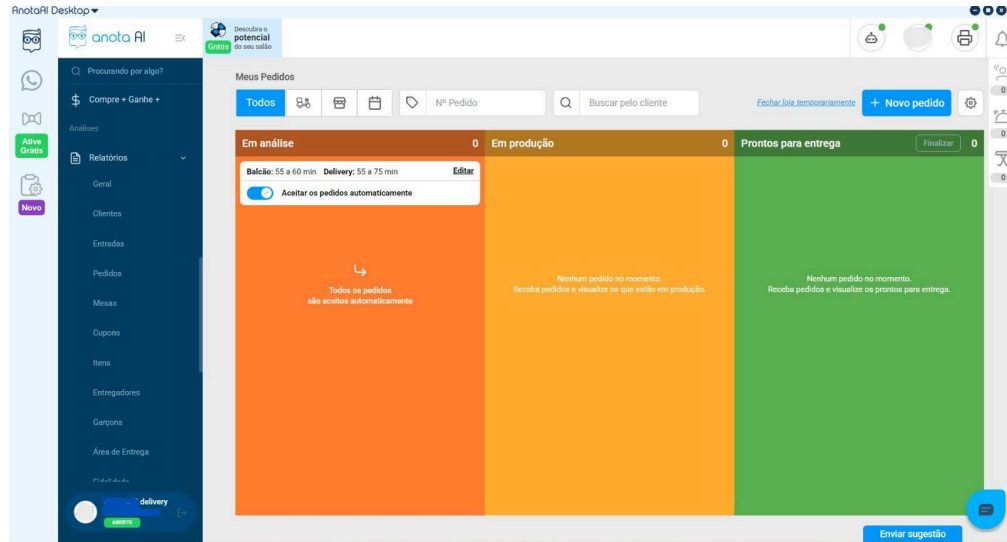


Fonte: Uber Eats (2025).

Soluções como o Anota AI surgem como alternativas viáveis para pequenos empreendedores. O sistema oferece atendimento automatizado via redes sociais (WhatsApp, Facebook e Instagram), cardápio digital, gestão de pedidos, programa de fidelidade, pagamento via Pix e relatórios de vendas (ANOTA AI, 2023). Além disso, permite personalização do atendimento, agendamento de pedidos, integração com sistemas de logística e PDV, e uso de QR Code para cardápios em mesas.

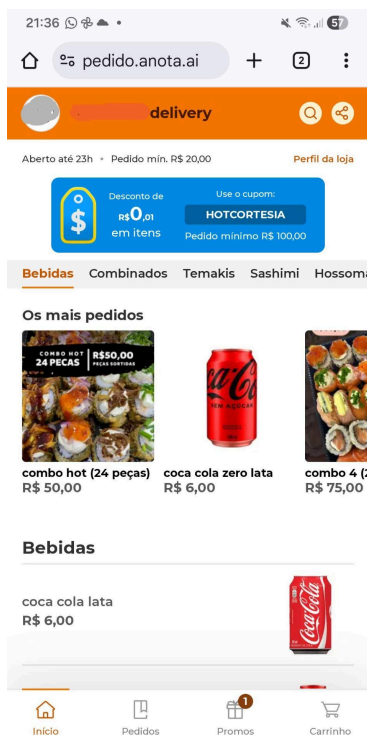
A Figura 7 ilustra o dashboard de gestão utilizado pelos restaurantes parceiros do Anota AI, para administrar pedidos e operações, enquanto a Figura 8 demonstra a interface de navegação disponível para os clientes finais, apresentando o cardápio digital de um estabelecimento da plataforma.

**Figura 7** - Dashboard de gestão do Anota AI para restaurantes parceiros.

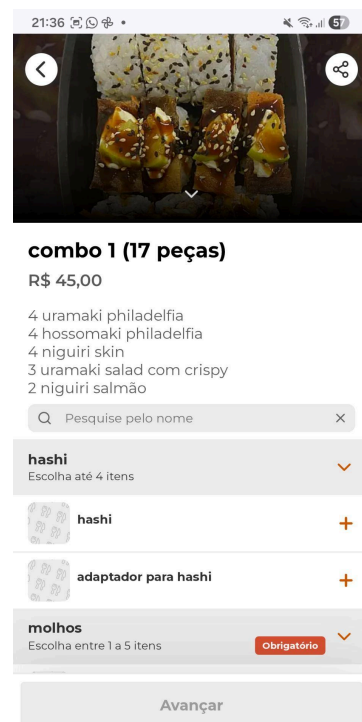


Fonte: Anota AI (2025).

Figura 8 - Fluxo de navegação no Anota AI: (I) Cardápio do estabelecimento e (J)Interface de um produto.



(I)



(J)

Fonte: Anota AI (2025).

### 3.1 Comparação entre Modelos de Delivery

A seguir, apresenta-se uma tabela comparativa entre sistemas tradicionais e soluções web adaptadas, destacando aspectos como custo, autonomia e personalização.

Tabela 1- Comparação de app tradicional com web

<b>Característica</b>	<b>Sistemas Tradicionais</b>	<b>Sistemas Web</b>
Instalação	Necessária via app	Acesso direto pelo navegador
Custo	Alto (comissões e mensalidades)	Baixo
Acesso aos dados do cliente	Limitado	Total
Personalização	Restrita	Alta
Autonomia do comerciante	Baixa	Alta

**Fonte:** Elaborado pelo autor (2025)

O sistema desenvolvido neste trabalho surge para preencher uma lacuna importante identificada no mercado de delivery digital, a falta de opções verdadeiramente acessíveis para pequenos comerciantes. A análise do cenário atual mostra um mercado predominantemente dominado por grandes plataformas que, mesmo quando estabelecem parcerias como no caso do Anota AI com o iFood, mantêm um modelo de negócios que não prioriza a sustentabilidade financeira dos pequenos negócios. Não foi identificada nenhuma solução com foco específico em reduzir custos e garantir autonomia para esses estabelecimentos. É nesse contexto que nosso sistema se apresenta, utilizando uma abordagem web adaptada para dispositivos móveis que combina usabilidade, autonomia operacional e custo reduzido. Mais

do que uma simples alternativa, nossa proposta oferece aos pequenos comerciantes uma opção estratégica para ingressar no delivery digital sem comprometer sua rentabilidade ou perder o controle sobre seu próprio negócio.

Tabela 2 - Comparação preços de outras plataformas

<b>Plataforma</b>	<b>Comissão sobre pedidos</b>	<b>Mensalidade fixa</b>	<b>Taxa de entrega</b>
<b>iFood</b>	23% a 27% sobre pedidos (dependendo do plano)	R\$ 100 a R\$ 130/mês	Variável, geralmente paga pelo cliente; restaurantes podem arcar em promoções
<b>Uber Eats</b>	30% (entrega pela Uber) ou 20% (autoentrega)	R\$ 500 taxa inicial de adesão	Incluso na comissão ou variável conforme modalidade
<b>99Food</b>	Comissão variável (em torno de 15% a 20%)	Promoções de isenção; sem mensalidade em alguns planos	Taxa de entrega depende da distância, se optar pelo plano com entregadores da 99Food
<b>AnotaAi</b>	Não cobra comissão sobre pedidos	R\$ 200 a R\$ 499/mês (dependendo do plano)	Não há taxa de entrega (restaurante gerência próprio delivery)
<b>Pedido expresso</b>	Não cobra comissão sobre pedidos	R\$ 100 a R\$ 200	Não há taxa de entrega (restaurante gerência próprio delivery)

**Fonte:** Elaborado pelo autor (2025)

## 4 REFERENCIAL TEÓRICO

Este capítulo tem como objetivo descrever a seleção das tecnologias, linguagens, ferramentas e metodologias adotadas para o desenvolvimento do sistema Pedido Expresso. A escolha de cada item foi pautada por critérios como custo benefício, adequação ao projeto e escalabilidade, alinhado-se diretamente com o objetivo de criar uma solução viável para pequenos comerciantes.

### 4.1 PHP

O PHP é uma linguagem open-source de scripting do lado do servidor voltada ao desenvolvimento de páginas e aplicações web dinâmicas. Segundo a documentação oficial (PHP, s.d.), a linguagem foi criada em 1994 por Rasmus Lerdorf como um conjunto de scripts (PHP/FI) e evoluiu de uma ferramenta simples para uma linguagem completa ao longo das décadas. Essa trajetória de desenvolvimento é marcada por marcos importantes: a transição para o PHP 3 (1997–1998) introduziu a Zend Engine, preparando o terreno para recursos modernos; o PHP 5 consolidou a Programação Orientada a Objetos (POO); o PHP 7 promoveu um salto de desempenho a partir do projeto phpng; e o PHP 8 trouxe o compilador JIT, melhorias de tipagem e recursos sintáticos contemporâneos (MORAES, 2025).

De acordo com Kinsta (2025), o PHP permanece central no ecossistema web, com uso maciço em Sistemas de Gerenciamento de Conteúdo (CMS), como o WordPress, e forte adoção em frameworks modernos como o Laravel, o que favorece a produtividade, segurança e boas práticas em projetos atuais. Entre as características técnicas que sustentam sua relevância, destacam-se a facilidade de desenvolvimento para a web, a ampla documentação oficial, a compatibilidade com múltiplos servidores e sistemas operacionais e o suporte robusto à POO. Conforme explica Nixon (2018), a linguagem também possui integração natural com MySQL por meio de extensões modernas (PDO/MySQLi) e um vasto ecossistema de bibliotecas. A adoção de boas práticas é amplamente documentada na literatura técnica recente sobre “PHP moderno” (LOCKHART, 2015).

Do ponto de vista da segurança e do desempenho, o manual oficial (PHP, s.d.) apresenta diretrizes para sessões, arquivos, bancos de dados e ocultação de informações sensíveis. Ademais, Moraes (2025) aponta que, nas versões PHP 7 e 8, há ganhos expressivos de desempenho e reforços de tipagem que reduzem classes de erros comuns. Guias

introdutórios atualizados reforçam práticas seguras e mostram o PHP como uma tecnologia ainda relevante no mercado e de rápida curva de aprendizado para iniciantes (HOSTGATOR BRASIL, 2025; KINSTA, 2025).

Para o desenvolvimento de um sistema de delivery, como o proposto neste trabalho, o PHP se justifica por uma conjunção de fatores. Ele oferece rapidez de desenvolvimento e custo acessível (KINSTA, 2025), amparado por uma documentação extensa e uma comunidade ativa (PHP, s.d.). Segundo a DevMedia (s.d.), a linguagem dispõe de frameworks e bibliotecas maduros para funcionalidades essenciais, como autenticação, APIs REST e integração com bancos relacionais. Por fim, a performance adequada proporcionada pelas versões 7 e 8, com seus recursos modernos, assegura a robustez necessária para a aplicação (MORAES, 2025).

#### 4.2 HTML5: Estrutura Semântica e Funcionalidades Avançadas

O HTML5 representa a quinta grande revisão da linguagem de marcação que constitui a base da Web. Sua evolução está diretamente ligada à necessidade de tornar a web mais semântica, acessível e funcional para aplicações modernas. Segundo a especificação da WHATWG (2025), o HTML foi originalmente criado para descrever documentos científicos, mas sua flexibilidade permitiu que fosse adaptado ao longo dos anos para representar diversos tipos de conteúdo e aplicações, como sistemas de compras, jogos online e editores de texto.

Durante os anos 1990, o HTML passou por diversas revisões, culminando na versão 4.01 em 1999. No início dos anos 2000, o W3C tentou substituir o HTML por uma linguagem baseada em XML, chamada XHTML. No entanto, essa abordagem não teve ampla adoção, o que levou à criação do grupo WHATWG em 2004, formado por empresas como Apple, Mozilla e Opera, com o objetivo de continuar o desenvolvimento do HTML de forma mais prática e compatível com a realidade da Web.

O HTML5 surgiu como resposta a essa necessidade de evolução, incorporando elementos semânticos como `<article>`, `<section>`, `<header>`, `<footer>` e `<nav>`, que ajudam a estruturar o conteúdo de forma lógica e acessível. De acordo com a MDN Web Docs (2025), esses elementos melhoram a legibilidade do código e facilitam a navegação por tecnologias assistivas, como leitores de tela.

Além disso, o HTML5 trouxe suporte nativo a recursos multimídia, como áudio e vídeo, por meio das tags <audio> e <video>, eliminando a dependência de plugins externos. Também introduziu o elemento <canvas>, que permite a renderização de gráficos e animações diretamente no navegador. A MDN (2025) destaca que essas funcionalidades são integradas a APIs modernas, como Web Audio API e HTMLMediaElement, ampliando as possibilidades de interatividade nas aplicações web.

Outro avanço importante foi a melhoria nos formulários, com novos tipos de entrada como email, date e range, além de validações nativas que simplificam o desenvolvimento e melhoram a experiência do usuário. Segundo o W3C (2014), o HTML5 define critérios claros de conformidade e introduz elementos baseados em práticas consolidadas, promovendo maior interoperabilidade entre navegadores.

#### 4.3 CSS3: Estilização Moderna e Layout Responsivo

As folhas de estilo em cascata (CSS) surgiram em 1994, quando Håkon Wium Lie, trabalhando no CERN, percebeu a necessidade de uma linguagem que permitisse controlar a apresentação visual dos documentos na Web. Na época, o HTML não oferecia recursos para definir estilos como fontes, cores ou layouts, o que gerava frustração entre os autores de páginas (W3C, 2016).

A primeira recomendação oficial do CSS foi publicada pelo W3C em 1996, com o objetivo de separar o conteúdo da apresentação. Desde então, a linguagem evoluiu significativamente, passando por diversas versões até chegar ao CSS3, que introduziu uma abordagem modular. Essa modularização permitiu que diferentes partes da especificação fossem desenvolvidas e atualizadas de forma independente, facilitando a implementação por navegadores e promovendo maior flexibilidade (W3C, 2001).

O CSS3 trouxe avanços importantes para o desenvolvimento web moderno. Entre os principais recursos estão os seletores avançados, transições, animações, gradientes, e os sistemas de layout Flexbox e Grid, que permitem criar estruturas responsivas e adaptáveis a diferentes tamanhos de tela. Segundo a MDN Web Docs (2025), essas ferramentas possibilitam o desenvolvimento de interfaces dinâmicas e acessíveis, otimizadas para múltiplos dispositivos.

Outro destaque do CSS3 são as media queries, que permitem aplicar estilos específicos com base nas características do dispositivo, como largura da tela, resolução e orientação. Essa funcionalidade é essencial para o design responsivo, garantindo que o conteúdo seja exibido de forma adequada em celulares, tablets e desktops (MDN, 2025).

Além disso, o CSS3 passou a incluir módulos como Box Model, Fonts, Backgrounds and Borders, e User Interface, cada um com especificações próprias e atualizações contínuas. O W3C (2025) mantém um snapshot anual com os módulos estáveis e em desenvolvimento, permitindo que desenvolvedores acompanhem a evolução da linguagem e adotem recursos com maior segurança.

#### 4.4 Bootstrap: Framework para Desenvolvimento Ágil e Responsivo

As folhas de estilo em cascata (CSS) surgiram em 1994, quando Håkon Wium Lie, trabalhando no CERN, percebeu a necessidade de uma linguagem que permitisse controlar a apresentação visual dos documentos na Web. Na época, o HTML não oferecia recursos para definir estilos como fontes, cores ou layouts, o que gerava frustração entre os autores de páginas (W3C, 2016).

A primeira recomendação oficial do CSS foi publicada pelo W3C em 1996, com o objetivo de separar o conteúdo da apresentação. Desde então, a linguagem evoluiu significativamente, passando por diversas versões até chegar ao CSS3, que introduziu uma abordagem modular. Essa modularização permitiu que diferentes partes da especificação fossem desenvolvidas e atualizadas de forma independente, facilitando a implementação por navegadores e promovendo maior flexibilidade (W3C, 2001).

O CSS3 trouxe avanços importantes para o desenvolvimento web moderno. Entre os principais recursos estão os seletores avançados, transições, animações, gradientes, e os sistemas de layout Flexbox e Grid, que permitem criar estruturas responsivas e adaptáveis a diferentes tamanhos de tela. Segundo a MDN Web Docs (2025), essas ferramentas possibilitam o desenvolvimento de interfaces dinâmicas e acessíveis, otimizadas para múltiplos dispositivos.

Outro destaque do CSS3 são as media queries, que permitem aplicar estilos específicos com base nas características do dispositivo, como largura da tela, resolução e orientação. Essa funcionalidade é essencial para o design responsivo, garantindo que o conteúdo seja exibido de forma adequada em celulares, tablets e desktops (MDN, 2025).

Além disso, o CSS3 passou a incluir módulos como Box Model, Fonts, Backgrounds and Borders, e User Interface, cada um com especificações próprias e atualizações contínuas. O W3C (2025) mantém um snapshot anual com os módulos estáveis e em desenvolvimento, permitindo que desenvolvedores acompanhem a evolução da linguagem e adotem recursos com maior segurança.

#### 4.5 jQuery: Interatividade e Manipulação Simplificada

Segundo o blog oficial do jQuery (METHVIN, 2016), a biblioteca foi criada por John Resig em janeiro de 2006, durante o evento BarCamp NYC. Na época, já existiam diversas bibliotecas JavaScript tentando resolver problemas semelhantes, mas o jQuery se destacou por sua simplicidade e pela forte comunidade que se formou em torno do projeto. A proposta era clara: “Write less, do more”, ou seja, escrever menos código para fazer mais. Com o tempo, o jQuery se tornou a biblioteca JavaScript mais utilizada no mundo, sendo incorporada em milhões de sites e aplicações.

De acordo com a documentação oficial (JQUERY, 2025), o jQuery é uma biblioteca rápida, leve e rica em recursos, que simplifica tarefas comuns como manipulação do DOM, tratamento de eventos, animações e requisições AJAX. A API do jQuery é fácil de usar e funciona de forma consistente em diversos navegadores, o que contribuiu para sua ampla adoção.

A documentação técnica (JQUERY API, 2025) destaca que o jQuery oferece métodos eficientes para manipular elementos HTML, como `.html()`, `.append()`, `.remove()`, `.attr()` e `.css()`. Esses métodos permitem alterar conteúdo, atributos e estilos de forma simples e direta, reduzindo a complexidade do código JavaScript tradicional.

Para a equipe do jQuery (OPENJS FOUNDATION, 2025), a biblioteca continua relevante mesmo com o surgimento de frameworks modernos, pois oferece uma abordagem

versátil e extensível para o desenvolvimento web. Além disso, o jQuery possui uma pegada leve (cerca de 30kB minificado), suporte a seletores CSS3 e compatibilidade com os principais navegadores, incluindo versões antigas como o Internet Explorer 9.

Segundo Willison (2024), cerca de 90% dos sites ainda utilizam jQuery, embora muitos estejam em versões desatualizadas. A OpenJS Foundation recomenda manter a biblioteca atualizada para garantir segurança, compatibilidade e adoção das melhores práticas. O plugin jQuery Migrate facilita esse processo, permitindo que aplicações antigas sejam adaptadas às versões mais recentes.

#### 4.6 MariaDB no Ambiente de Desenvolvimento com XAMPP

O MariaDB é um sistema de gerenciamento de banco de dados relacional (SGBDR) de código aberto, desenvolvido como um fork do MySQL em 2009 por Michael “Monty” Widenius. A criação do MariaDB teve como objetivo preservar a liberdade de desenvolvimento e garantir a continuidade de uma plataforma aberta e comunitária após a aquisição do MySQL pela Oracle Corporation (MARIADB CORPORATION, 2025).

Segundo a documentação oficial da MariaDB Corporation, o sistema oferece uma arquitetura robusta e escalável, com suporte a recursos avançados como Stored Procedures, Triggers, Views, Common Table Expressions (CTEs) e Window Functions, que permitem a construção de consultas complexas e otimizadas para aplicações modernas (MARIADB CORPORATION, 2025).

A MariaDB Foundation destaca que o sistema mantém alta compatibilidade com o MySQL, permitindo que scripts, comandos e clientes sejam reutilizados com mínima adaptação. Isso facilita a migração entre os sistemas e amplia a adoção do MariaDB em projetos que originalmente utilizavam o MySQL (MARIADB FOUNDATION, 2025).

No contexto do desenvolvimento web, o MariaDB é frequentemente utilizado em conjunto com o XAMPP, um pacote gratuito e multiplataforma que inclui Apache, PHP e MariaDB. O XAMPP é mantido pela organização Apache Friends e tem como objetivo facilitar a criação de ambientes locais para desenvolvimento de aplicações web, especialmente em PHP (APACHE FRIENDS, 2025).

Desde 2015, o MariaDB substituiu o MySQL como banco de dados padrão no XAMPP, permitindo que desenvolvedores iniciem rapidamente seus projetos sem a necessidade de

configurações complexas. Essa integração é ideal para ambientes acadêmicos e protótipos, como o sistema de delivery desenvolvido neste trabalho (APACHE FRIENDS, 2025).

#### 4.7 SCRUM: METODOLOGIA ÁGIL DE DESENVOLVIMENTO

O Scrum é uma metodologia ágil voltada para o gerenciamento e desenvolvimento de projetos complexos, especialmente na área de software. Criado por Ken Schwaber e Jeff Sutherland, o Scrum propõe uma abordagem iterativa e incremental, com foco na entrega contínua de valor ao cliente e na adaptação constante às mudanças (SCHWABER, SUTHERLAND, 2020).

Segundo a documentação oficial, o Scrum é composto por papéis, eventos e artefatos. Os papéis principais são o Product Owner, responsável por definir e priorizar os requisitos do produto; o Scrum Master, que atua como facilitador do processo; e o Time de Desenvolvimento, encarregado de implementar as funcionalidades. Os eventos incluem a Sprint, que é um ciclo de trabalho com duração fixa (geralmente de duas a quatro semanas), a Daily Scrum (reunião diária de alinhamento), a Sprint Planning, a Sprint Review e a Sprint Retrospective (SUTHERLAND, 2014).

Durante o desenvolvimento do sistema de delivery online, embora nem todos os elementos do Scrum tenham sido aplicados formalmente, a metodologia serviu como referência organizacional. A divisão do projeto em etapas, a priorização de funcionalidades e a realização de testes incrementais refletem práticas inspiradas no modelo ágil.

Segundo Pressman, metodologias ágeis como o Scrum são especialmente eficazes em projetos que exigem rapidez, colaboração contínua com o cliente e capacidade de resposta a mudanças, características presentes no contexto acadêmico e em sistemas voltados para o mercado digital (PRESSMAN, 2016).

## 5 DESENVOLVIMENTO

O desenvolvimento do sistema foi realizado com base nos requisitos levantados e nas tecnologias definidas no referencial teórico. Esta etapa descreve as fases práticas da construção do sistema, desde a análise dos requisitos até a implementação do front-end e back-end, utilizando metodologias ágeis para garantir flexibilidade e eficiência. O objetivo é apresentar de forma clara as decisões técnicas, ferramentas utilizadas e os processos que levaram à entrega do produto final.

### 5.1 REQUISITOS

Os requisitos funcionais descrevem as funcionalidades que o sistema deve oferecer para atender às necessidades do usuário. Para o sistema de delivery, foram definidos os seguintes requisitos principais

Tabela 3 - Requisitos

<b>Tipo de Requisito</b>	<b>Código</b>	<b>Descrição</b>
<b>Funcional</b>	RF 001	Listar produtos por categorias
	RF 002	Cadastro, exclusão e alteração de produto
	RF 003	Cadastro de cliente e administrador
	RF 004	Inserir/excluir produtos no carrinho
	RF 005	Botão de loja aberta/fechada
	RF 006	Botão de produto indisponível para informar o cliente

	RF 007	Atualização automática da tela na entrada de pedidos
<b>Não Funcional</b>	RNF 001	Login restrito para acesso ao sistema
	RNF 002	Sistema web compatível com computadores pessoais
	RNF 003	Identificação do cliente pelo número de telefone após cadastro
	RNF 004	Tela de cadastro/alteração de produto acessível apenas ao administrador
	RNF 005	Botão de loja aberta/fechada acessível apenas ao administrador
	RNF 006	Cálculo automático dos produtos no carrinho
	RNF 007	Envio automático ou semiautomático de mensagem ao cliente na entrega
	RNF 008	Exibição de informações do pedido e cliente na loja

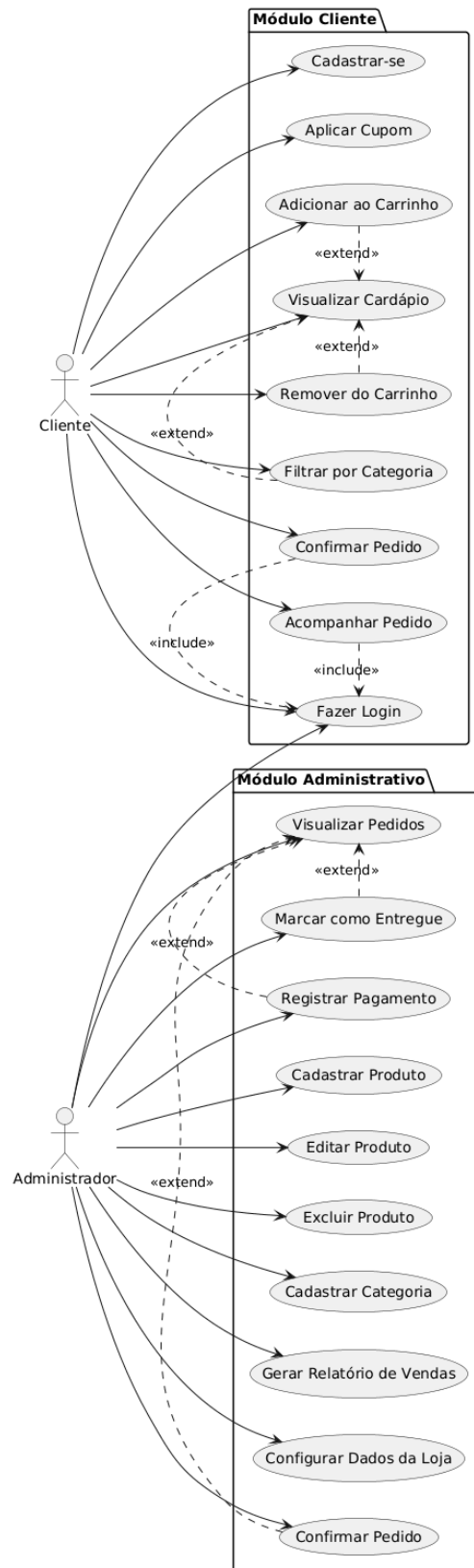
**Fonte:** Elaborado pelo autor (2025)

## 5.2 UML

O caso de uso da Figura 9 representa as interações entre os usuários e o sistema. Foi elaborado o diagrama para ilustrar as principais funcionalidades e facilitar o desenvolvimento.

Figura 9 - Caso de uso

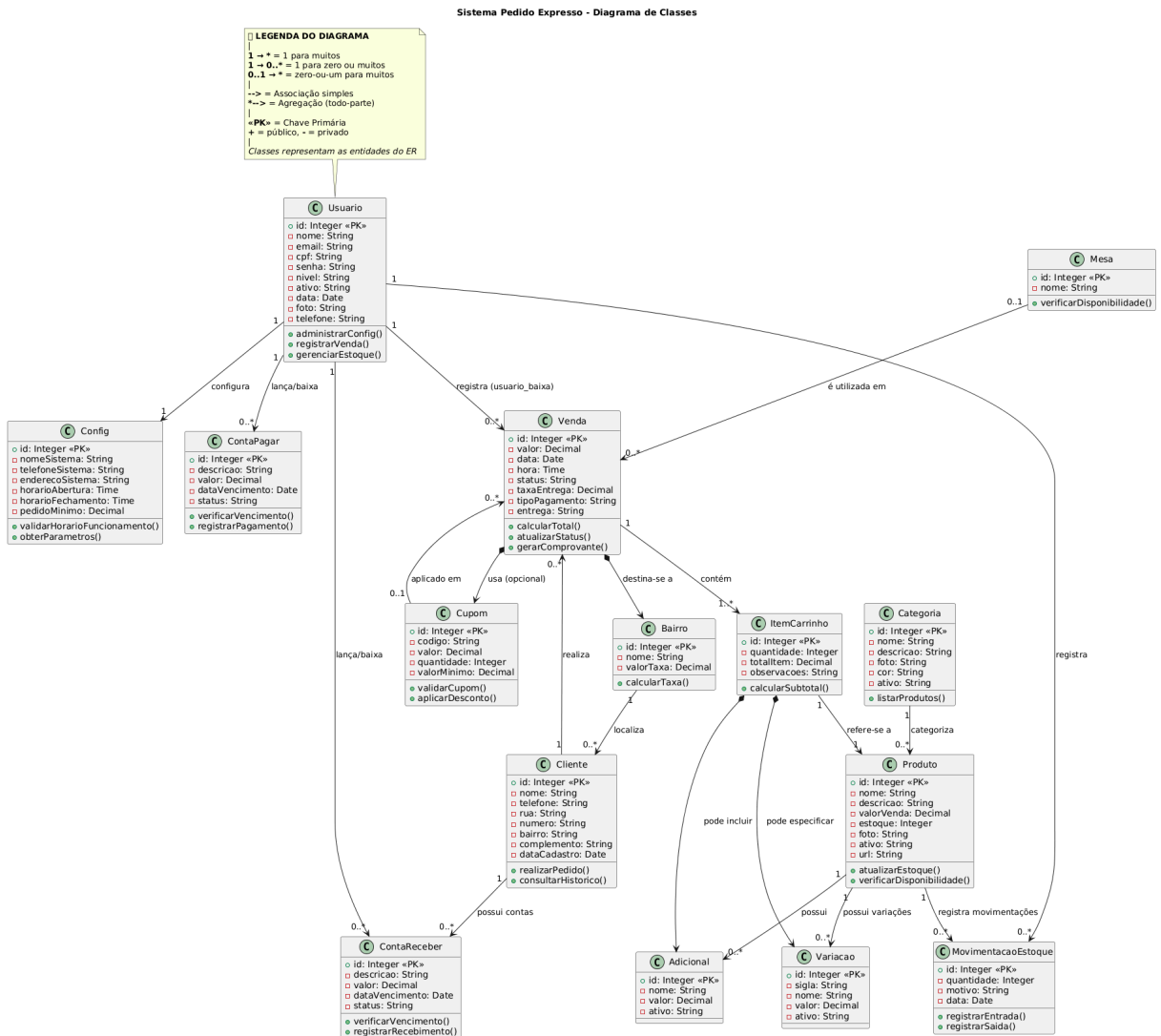
Sistema Pedido Expresso - Diagrama de Caso de Uso



Fonte: Elaborado pelo autor (2025)

O Diagrama de classe da Figura 10 representa o início da estrutura para começar a modelagem do sistema.

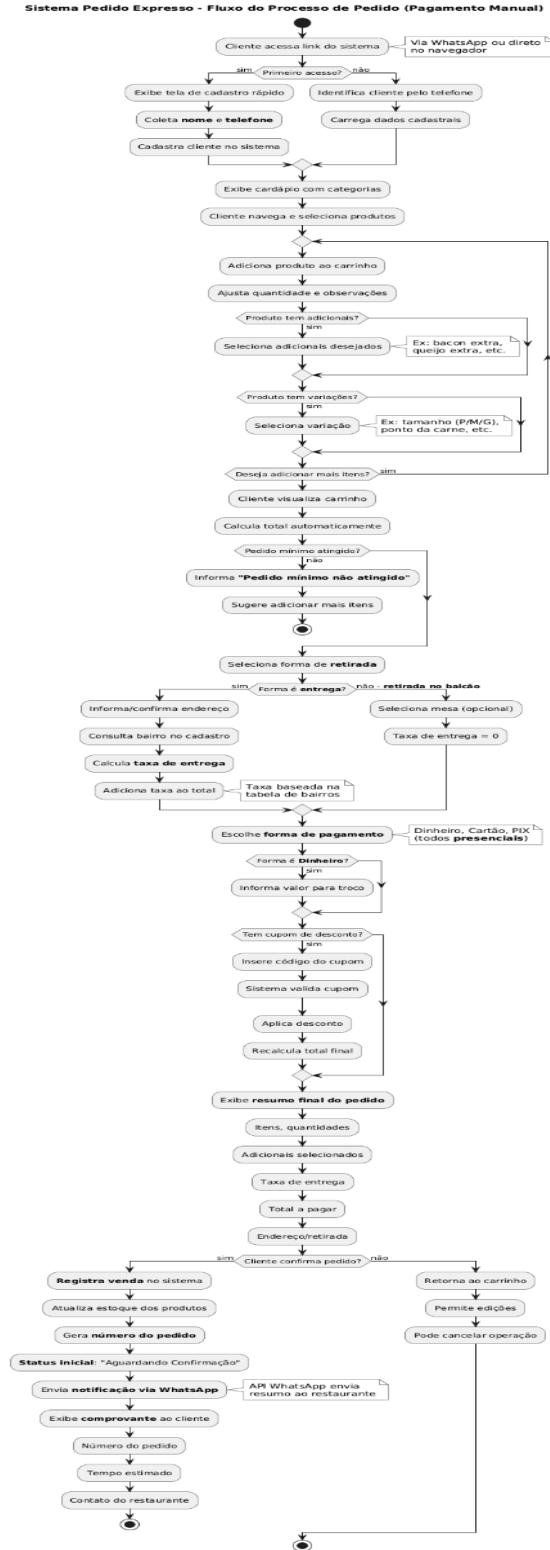
**Figura 10 - Diagrama de classe**



Fonte: Elaborado pelo autor (2025)

O Diagrama de fluxo da figura 11 representa as interações entre o cliente e o sistema.

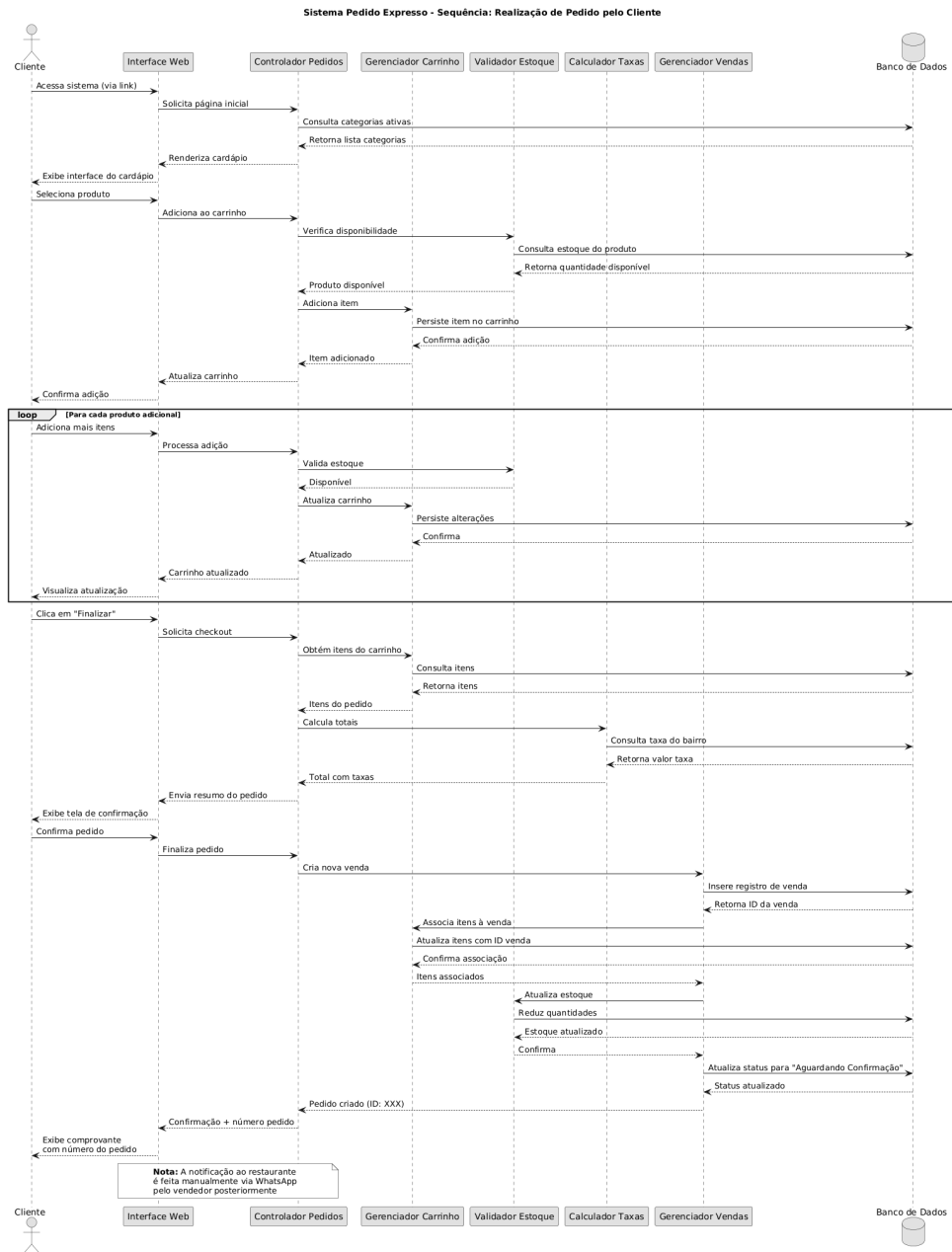
Figura 11 - Diagrama de fluxo



Fonte: Elaborado pelo autor (2025)

O Diagrama de seqüência da Figura 11 exemplifica como o sistema irá se comportar durante o acesso de um cliente em compras.

Figura 12 - Diagrama de seqüência

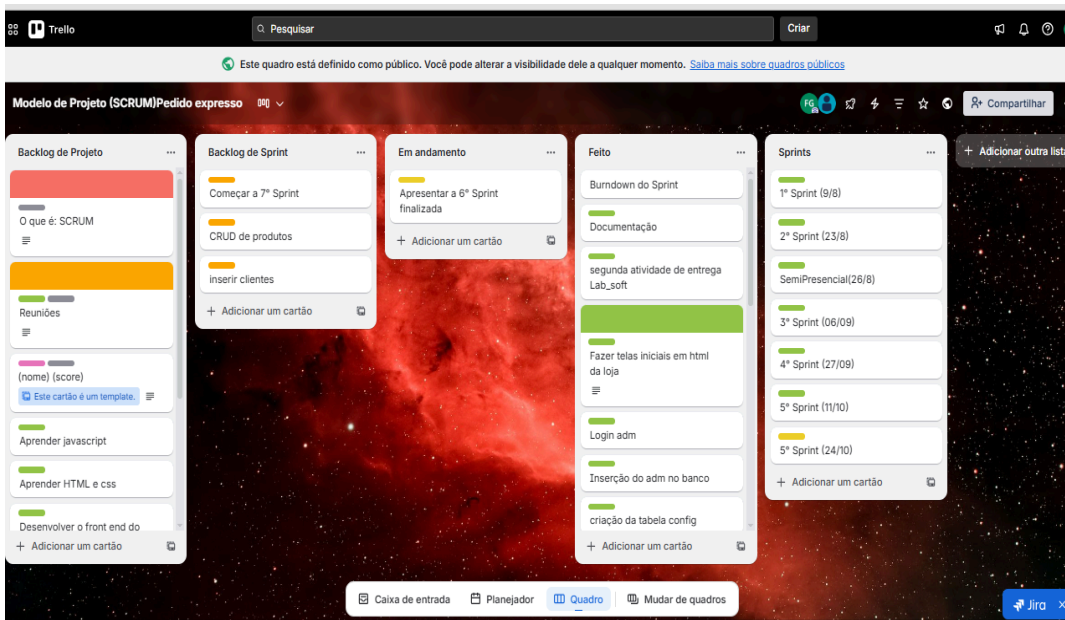


Fonte: Elaborado pelo autor(2025)

### 5.3 METODOLOGIA SCRUM

A metodologia Scrum foi aplicada de forma adaptada para o desenvolvimento individual, organizando o projeto em sprints semanais com ciclos de planejamento, desenvolvimento e revisão contínua. Utilizando o Trello como ferramenta principal para gerenciamento do backlog e acompanhamento visual das tarefas, foi possível manter um fluxo de trabalho organizado e interativo. Essa abordagem ágil permitiu priorizar as funcionalidades mais críticas do sistema, realizar ajustes rápidos baseados no progresso e garantir uma evolução constante do produto, com entregas de valor incrementais a cada ciclo de desenvolvimento.

**Figura 13** - Modelo de Projeto (SCRUM)Pedido expresso



**Fonte:** Elaborado pelo autor (2025)

#### 5.4 ENTIDADE-RELACIONAMENTO

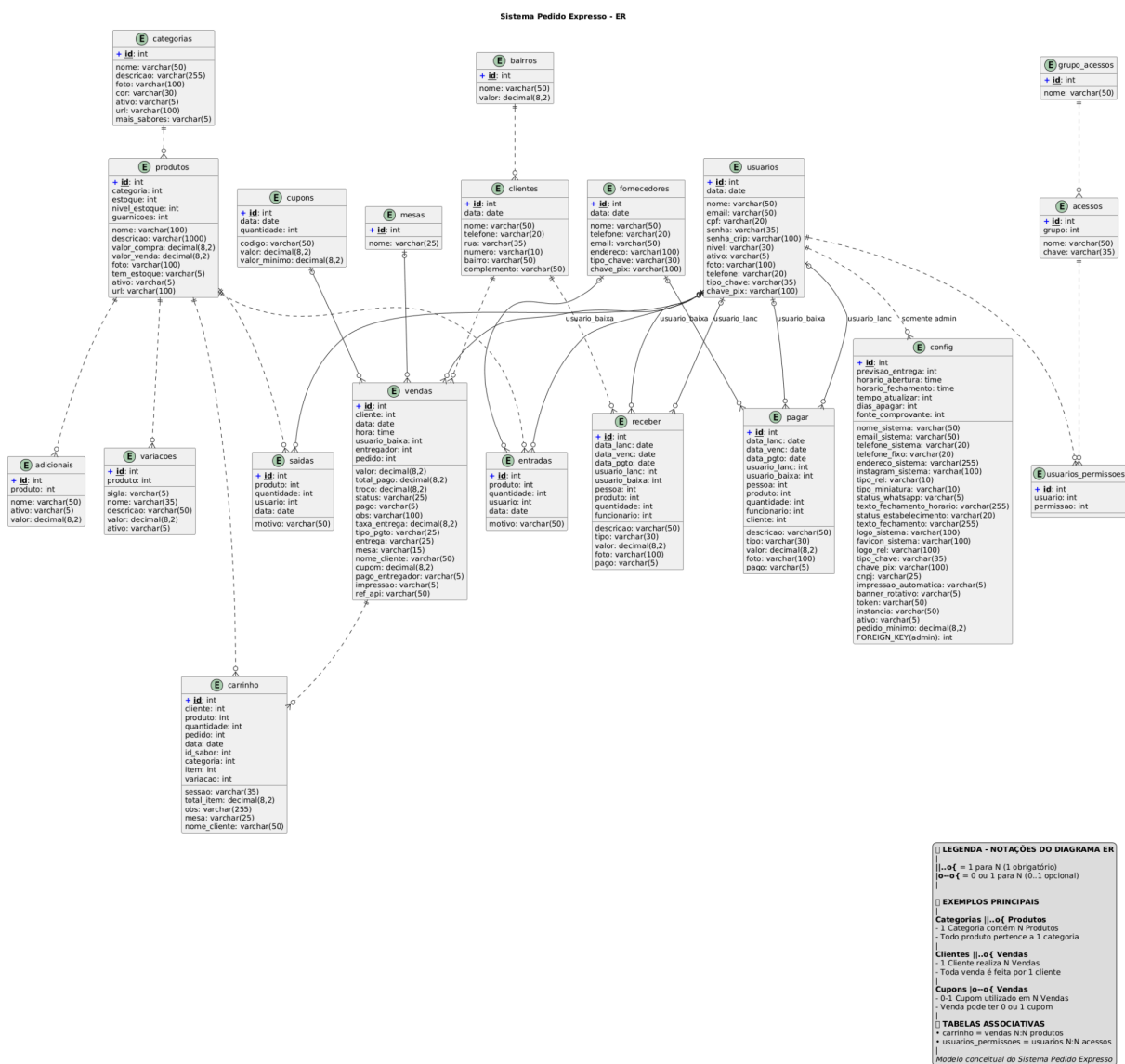
O modelo entidade-relacionamento (ER) apresentado na Figura 14 representa a estrutura final do banco de dados do sistema Pedido Expresso, resultante de um processo iterativo de desenvolvimento onde o esquema de dados foi sendo incrementalmente refinado e expandido ao longo do ciclo de desenvolvimento.

Inicialmente concebido com entidades básicas como Cliente, Produto e Venda para atender aos requisitos fundamentais do delivery, o modelo evoluiu para incorporar entidades especializadas como Config para parametrização do estabelecimento, Bairros para cálculo de taxas de entrega, Cupons para gestão promocional, e Usuários com sistema de permissões hierárquicas.

Essa evolução reflete a abordagem ágil adotada no projeto, onde novas funcionalidades identificadas durante o desenvolvimento demandaram a criação de entidades adicionais e o refinamento dos relacionamentos. O diagrama final captura essa maturidade, apresentando relacionamentos um-para-muitos (Cliente-Venda), muitos-para-muitos através de tabelas de junção (Carrinho), e um-para-um para controle de acesso administrativo.

A modelagem resultante suporta integralmente tanto as operações básicas de pedido e entrega quanto funcionalidades avançadas de gestão, relatórios e personalização do sistema, demonstrando a escalabilidade do projeto de banco de dados para acomodar requisitos emergentes.

Figura 14 - Entidade Relacionamento



Fonte: Elaborado pelo autor(2025)

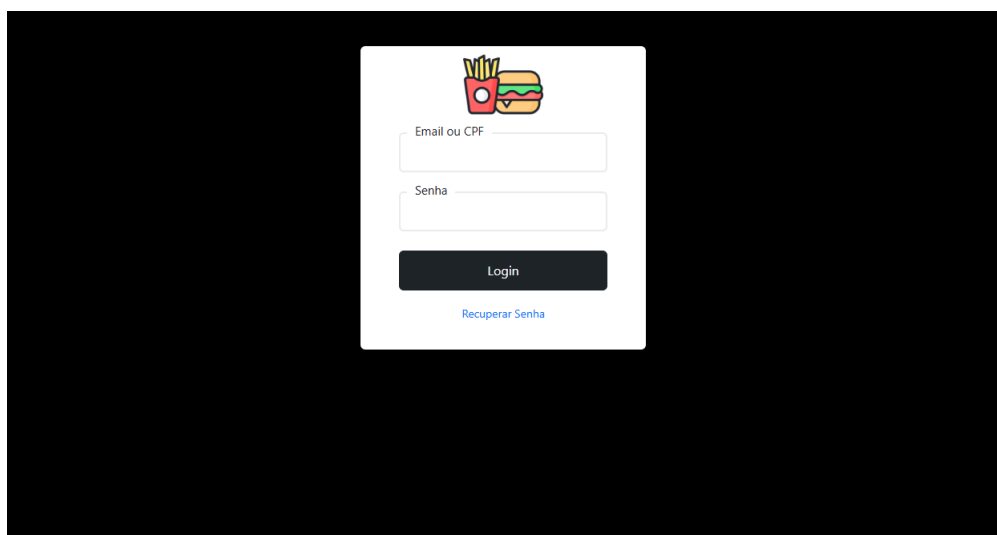
O sistema Pedido Expresso estrutura-se em um modelo relacional com entidades centrais interconectadas: Clientes realizam Pedidos através do Carrinho, que agrega múltiplos Produtos organizados por Categorias. O módulo administrativo gerencia Usuários com sistema hierárquico de permissões, enquanto entidades de suporte como Fornecedores, Bairros (para cálculo de taxas) e Cupons complementam o ecossistema. Os relacionamentos abrangem desde associações 1:N (Cliente→Pedidos) até N:N mediadas por tabelas de junção (Produtos↔Pedidos via Carrinho), formando uma arquitetura de

dados coesa que suporta todas as operações do delivery, desde o catálogo até a entrega e gestão financeira.

## 5.5 FRONT-END

A interface do usuário do sistema Pedido Expresso<sup>5</sup> foi desenvolvida com foco na experiência do cliente, utilizando uma *stack* tecnológica composta por HTML5, CSS3, Bootstrap 5 e jQuery. Esta combinação permitiu a criação de uma interface responsiva que se adapta automaticamente a diferentes dispositivos, desde smartphones até desktops, assegurando usabilidade tanto para clientes finais quanto para administradores.

**Figura 15** - Tela de login

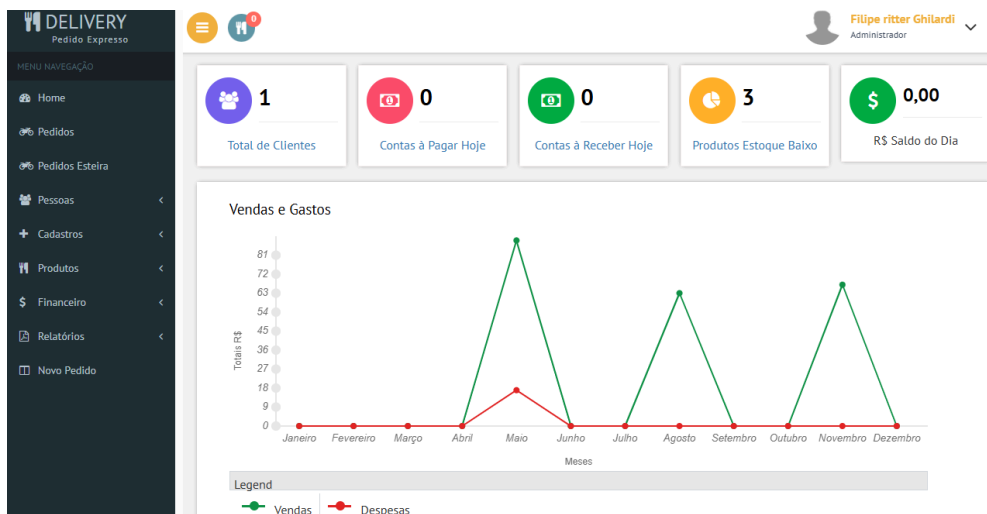


**Fonte:** Elaborado pelo autor (2025)

---

<sup>5</sup> Pedido Expresso, Sistema automatizado para gestão de delivery. Desenvolvimento: Filipe Ghilardi.

Figura 16 - Tela Dashboard



Fonte: Elaborado pelo autor(2025)

Figura 17 - Tela de configuração

The configuration page includes the following fields and settings:

- Nome do Projeto:** Pedido Expresso
- Email Sistema:** filipeghilardi@gmail.com
- Telefone Sistema:** (51) 99323-9752
- Telefone Fixo:** Telefone Fixo
- Endereço (Rua Número Bairro e Cidade):** Rua X Número 150 - Bairro Centro
- Instagram:** https://www.instagram.com/filipeghilardi
- Tipo Relatório:** PDF
- Mostrar Miniaturas:** Cores
- Status Whatsapp:** Sim (Manual)
- Previsão Entrega Minutos:** 50
- Abertura:** 08:00
- Fechamento:** 21:00
- Texto Fechamento Fora do Horário:** Estamos Fechados, funcionamos das 09:00 as 00:00
- Estabelecimento:** Aberto
- Texto Fechamento (Imprevisto):** Fechamos por falta de produtos
- CNPJ da Loja:** CNPJ caso tenha
- Atualizar Pedidos:** 30
- Tipo Chave Pix:** Telefone
- Chave Pix:** 5199323952
- Dias Limpar Banco:** 30
- Impressão Automática:** Não
- Fonte Comprovante:** 11
- Banner Rotativo:** Não
- Token Api Whatsapp:** Token Whatsapp
- Instância Api Whatsapp:** Instancia Whatsapp
- Pedido Mínimo:** 12.00
- Logo (\*PNG):** Escolher arquivo | Nenhum...escolhido
- Ícone (\*Png):** Escolher arquivo | Nenhum...escolhido

Fonte: Elaborado pelo autor(2025)

Figura 18 -Tela gestão de pedidos

Todos(8) / Iniciado(6) / Preparando(0) / Em Rota Entrega(2)

Mostrar 10 registros

Buscar:

Cliente	Valor	Total Pago	Troco	Forma PGTO	Status	Hora	Ações
Pedido (3) / Fernanda (Delivery)	R\$ 40,00	R\$ 100,00	R\$ 60,00	Dinheiro	Entrega →	10:06:07	🔍 🗑️ 📄
Pedido (4) / Cleonice (Delivery)	R\$ 35,00	R\$ 35,00	R\$ 0,00	Dinheiro	Entrega →	10:10:08	🔍 🗑️ 📄
Pedido (5) / Luiz Fernando (Consumir Local)	R\$ 30,00	R\$ 30,00	R\$ 0,00	Cartão de Crédito	Iniciado →	10:13:05	🔍 🗑️ 📄
Pedido (6) / Jean (Consumir Local)	R\$ 46,90	R\$ 46,90	R\$ 0,00	Cartão de Débito	Iniciado →	10:19:17	🔍 🗑️ 📄
Pedido (7) / Paulo (Delivery)	R\$ 55,00	R\$ 55,00	R\$ 0,00	Cartão de Débito	Iniciado →	10:22:19	🔍 🗑️ 📄
Pedido (8) / Bruno (Retirar)	R\$ 35,00	R\$ 35,00	R\$ 0,00	Pix	Iniciado →	10:22:58	🔍 🗑️ 📄
Pedido (9) / Thais (Delivery)	R\$ 43,00	R\$ 43,00	R\$ 0,00	Cartão de Crédito	Iniciado →	10:28:45	🔍 🗑️ 📄
Pedido (10) / Joa (Consumir Local)	R\$ 35,00	R\$ 35,00	R\$ 0,00	Pix	Iniciado →	10:32:17	🔍 🗑️ 📄

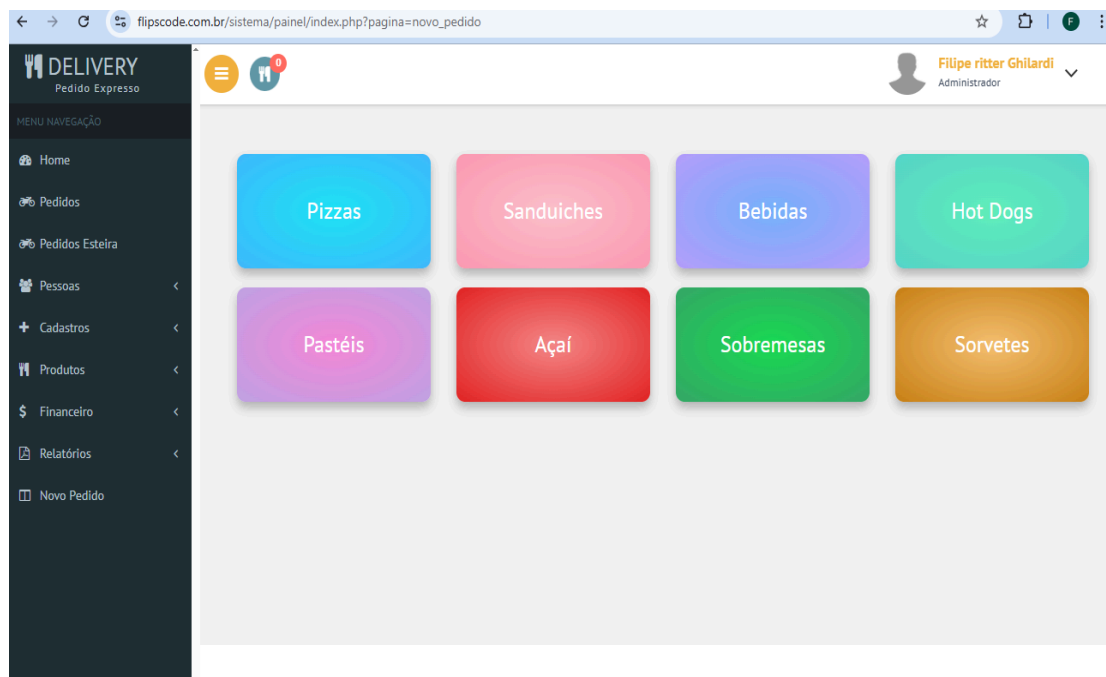
Listando 1 / 8 de 8 registros

Anterior 1 Próximo

Total de Pedidos: 8

Fonte: Elaborado pelo autor (2025)

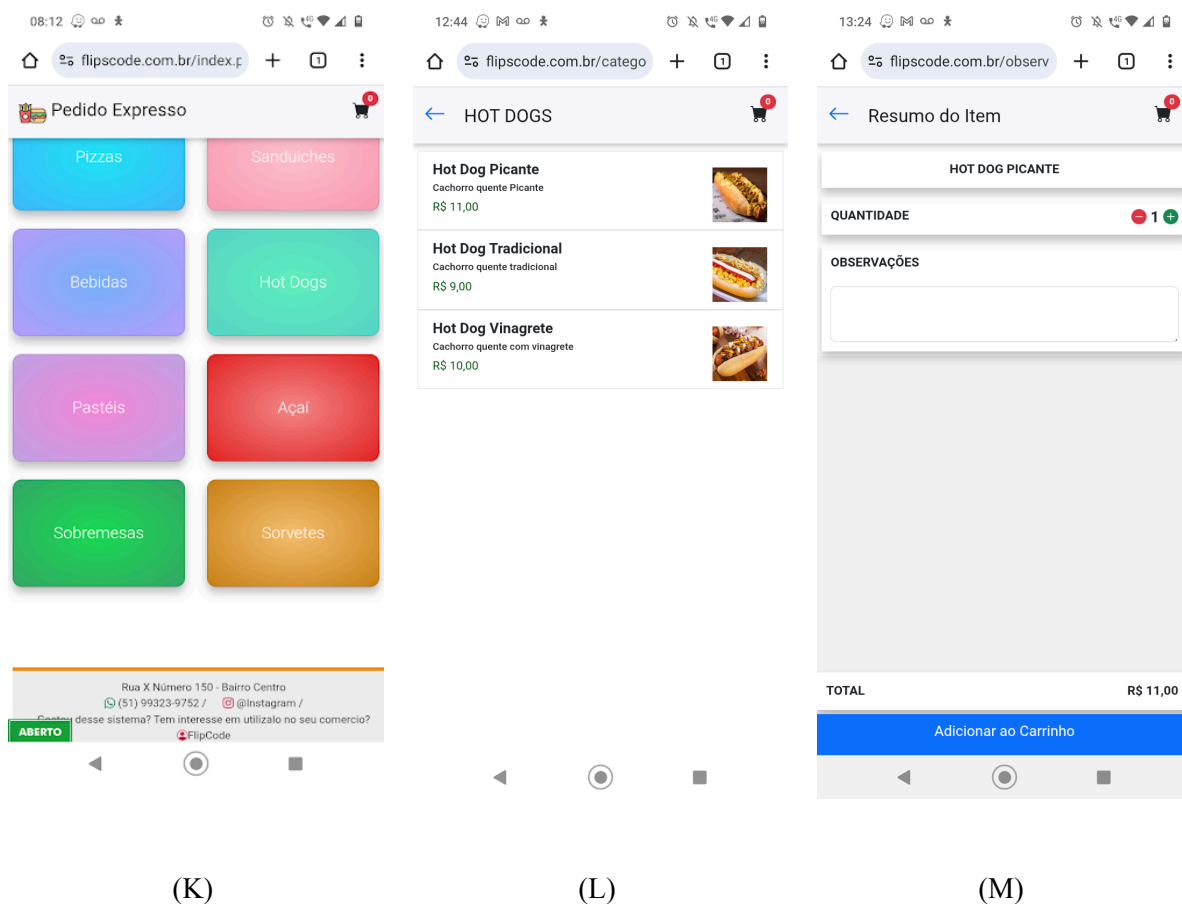
Figura 19 - Tela para inserir o pedido direto no balcão



Fonte: Elaborado pelo autor (2025)

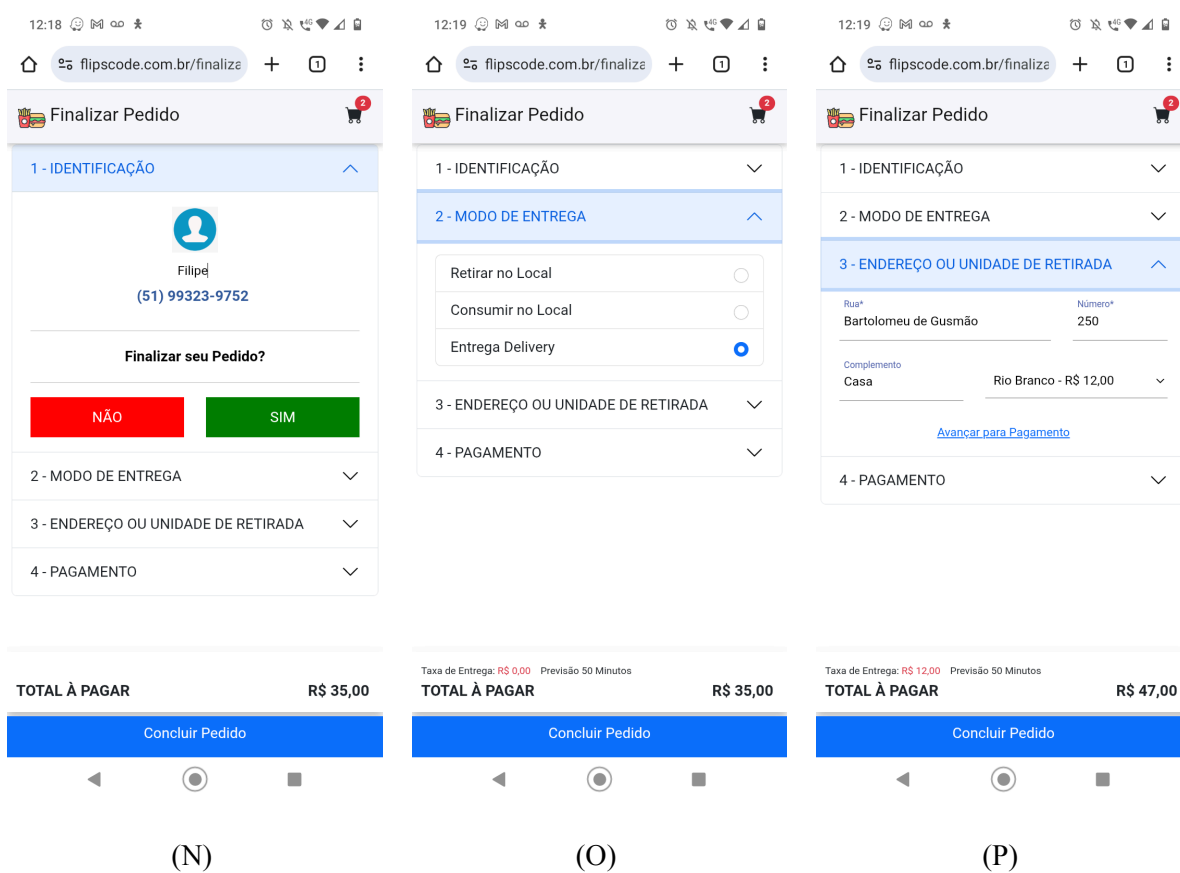
Foram implementadas páginas estratégicas para cada etapa da jornada do usuário: a vitrine virtual para exibição do cardápio com categorias organizadas e filtros intuitivos; o carrinho de compras com atualização em tempo real de valores; e o sistema de checkout com formulários otimizados para finalização do pedido.

**Figura 20** - (K)Menu de navegação: (L) Cardápio de um produto e (M)Adicionar Carrinho.



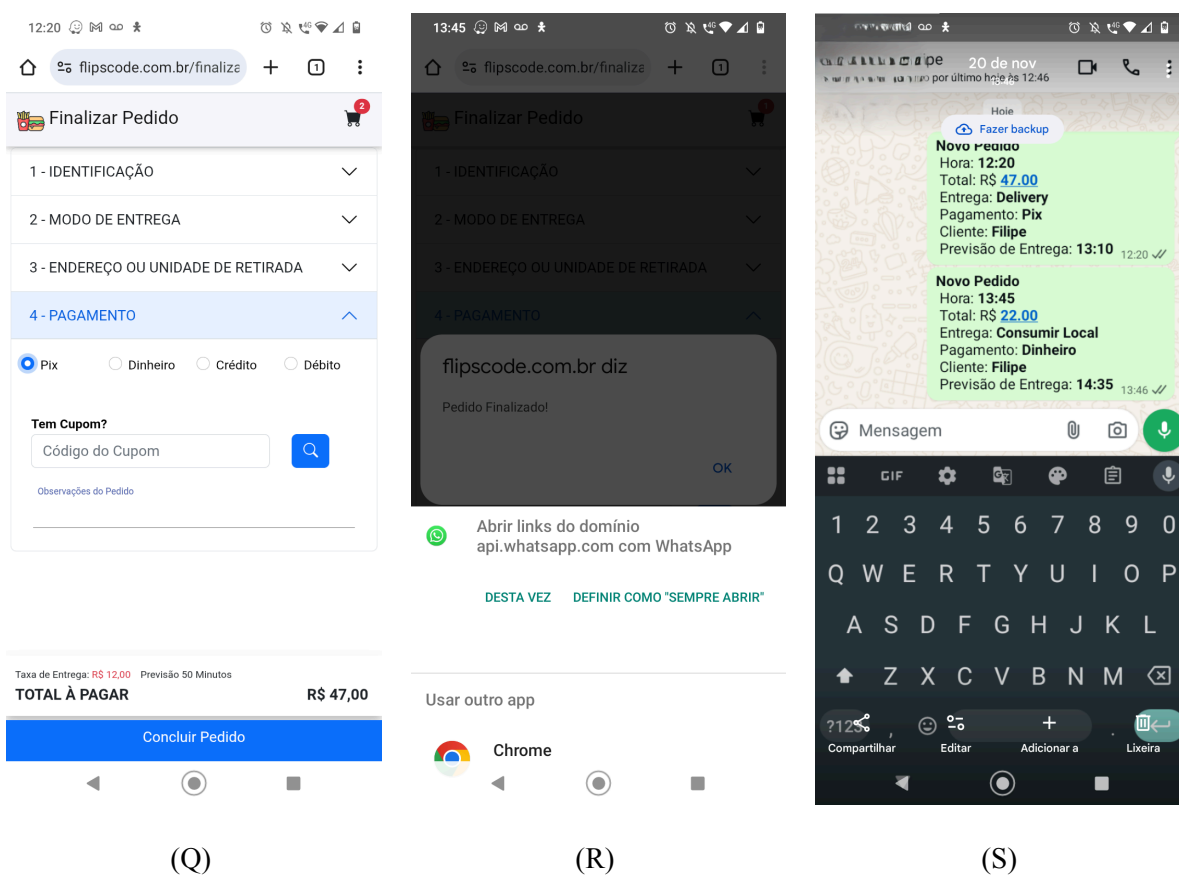
Fonte: Elaborado pelo autor (2025)

**Figura 21** - (N) Inserir número e nome (O) Forma de retirada e (P) Cadastro de endereço caso não tenha.



Fonte: Elaborado pelo autor (2025)

**Figura 22 - (Q)Forma de pagamento, (R) Finalizando pedido e (S) Mensagem do cliente para o restaurante usando API.**



**Fonte:** Elaborado pelo autor (2025)

Para o painel administrativo, desenvolveu-se interfaces completas de gestão de produtos, pedidos, clientes e relatórios.

**Figura 23** -Tela gestão de produtos

DELIVERY  
Pedido Expresso

MENU NAVEGAÇÃO

- Home
- Pedidos
- Pedidos Esteira
- Pessoas
- Cadastros
- Produtos
- Financeiro
- Relatórios
- Novo Pedido





























































Flipscode.com.br/sistema/painel/index.php?pagina=produtos

Filipe Ritter Ghilardi  
Administrador

+ NOVO PRODUTO

Mostrar 10 registros

Buscar:

Nome	Categoria	Valor Compra	Valor Venda	Estoque	Ações
Enérgico Red Bull	Bebidas	R\$ 0,00	R\$ 13,00	33	     
Sorvete de Chocolate	Sorvetes	R\$ 0,00	R\$ 9,00	0	     
Sorvete de Baunilha 30...	Sorvetes	R\$ 0,00	R\$ 8,00	0	     
Pudim de Leite Condensado	Sobremesas	R\$ 0,00	R\$ 16,00	0	     
Pave de Maracujá	Sobremesas	R\$ 0,00	R\$ 16,00	0	     
Mousse de Chocolate	Sobremesas	R\$ 0,00	R\$ 18,00	0	     
Burguer Picanha	Sanduíches	R\$ 0,00	R\$ 30,00	0	     
Burguer Costelinha	Sanduíches	R\$ 0,00	R\$ 25,00	0	     
Burguer Cheddar	Sanduíches	R\$ 0,00	R\$ 23,00	0	     
Pizza Peperoni	Pizzas	R\$ 0,00	R\$ 0,00	0	     

Fonte: Elaborado pelo autor (2025)

**Figura 24** -Tela gestão de pedidos

flipscode.com.br/sistema/painel/index.php?pagina=pedidos\_esteira

DELIVERY Pedido Expresso

Filipe Ritter Ghilardi Administrador

Todos(10) / Iniciado(8) / Preparando(0) / Em Rota Entrega(2)

Pedido	Horário	Nome	Status	Valor	Forma de Pagamento
Pedido 3	10:06	Fernanda Ritter (Delivery)	Status Entregue	R\$ 100,00 Troco: 60,00	(Dinheiro)
Pedido 4	10:10	Cleonice (Delivery)	Status Entregue	R\$ 35,00	(Dinheiro)
Pedido 5	10:13	Luiz Fernando (Consumir Local)	Status Iniciado	R\$ 30,00	(Cartão de Crédito)
Pedido 6	10:19	Jean baitezab (Consumir Local)	Status Iniciado	R\$ 46,90	(Cartão de Débito)
Pedido 7	10:22	Paulo (Delivery)	Status Iniciado	R\$ 55,00	(Cartão de Débito)
Pedido 8	10:22	Bruno Ritter (Retirar)	Status Iniciado	R\$ 35,00	(Pix)
Pedido 9	10:28	Thais (Delivery)	Status Iniciado	R\$ 43,00	(Cartão de Crédito)
Pedido 10	10:32	Joa (Consumir Local)	Status Iniciado	R\$ 35,00	(Pix)
Pedido 11	12:20	Filipe (Delivery)	Status Iniciado	R\$ 47,00	(Pix)
Pedido 12	13:45	Filipe (Consumir Local)	Status Iniciado	R\$ 50,00 Troco: 28,00	(Dinheiro)

Fonte: Elaborado pelo autor (2025)

Figura 25 -Tela gestão de clientes

flipscode.com.br/sistema/painel/index.php?pagina=clientes

DELIVERY Pedido Expresso

Filipe Ritter Ghilardi Administrador

+ NOVO CLIENTE

Mostrar 10 registros

Buscar:

Nome	Telefone	Localidade	Cadastro	Ações
Bruno	(51) 99322		20/11/2025	
Paulo	(51) 12323	Mathias	20/11/2025	
Jean	(51) 9981		20/11/2025	
Luiz Fernando	(51) 98950		20/11/2025	
Cleonice	(51) 99178	Mato Grande	20/11/2025	
Fernanda	(51) 99757	Mato Grande	20/11/2025	
Test senha	(51) 99323	Rio Branco	20/11/2025	
Cliente x	(51) 99795	Guajuviras	20/11/2025	
Teste	(51) 99795	Mato Grande	04/11/2025	

Listando 1 / 9 de 9 registros

Anterior 1 Próximo

Fonte: Elaborado pelo autor (2025)

Figura 26 -Tela gera relatório produtos

## Relatório de Produtos

QUINTA, 20 DE NOVEMBRO DE 2025



Nome	Categoria	Valor Compra	Valor Venda	Estoque
Açaí Pote 1 Litro	Açaí	R\$ 0,00	R\$ 20,00	0
Água Mineral 500 ML	Bebidas	R\$ 0,00	R\$ 6,00	0
Burger Cheddar	Sanduíches	R\$ 0,00	R\$ 23,00	0
Burger Costelinha	Sanduíches	R\$ 0,00	R\$ 25,00	0
Burger Picanha	Sanduíches	R\$ 0,00	R\$ 30,00	0
Coca Cola Lata 350 ML	Bebidas	R\$ 3,33	R\$ 1,00	68
Enérgico Red Bull	Bebidas	R\$ 0,00	R\$ 13,00	32
Hot Dog Picante	Hot Dogs	R\$ 0,00	R\$ 11,00	0
Hot Dog Tradicional	Hot Dogs	R\$ 0,00	R\$ 9,00	0
Hot Dog Vinagrete	Hot Dogs	R\$ 0,00	R\$ 10,00	0
Mousse de Chocolate	Sobremesas	R\$ 0,00	R\$ 18,00	0
Pastel de Carne	Pastéis	R\$ 0,00	R\$ 7,00	0
Pastel de Queijo	Pastéis	R\$ 0,00	R\$ 8,00	0
Pastel Napolitano	Pastéis	R\$ 0,00	R\$ 7,00	0
Pave de Maracujá	Sobremesas	R\$ 0,00	R\$ 16,00	0

**Fonte:** Elaborado pelo autor (2025)

O Bootstrap foi fundamental para garantir consistência visual através de seu sistema de grid, componentes pré-estilizados e utilitários de responsividade. O jQuery complementou com interatividade para ações como adição de itens ao carrinho, cálculo de totais, validação de formulários e gerenciamento de modais.

A interface priorizou princípios de Design com navegação simplificada, hierarquia visual clara. O acompanhamento de pedidos é realizado através do WhatsApp, onde os clientes podem receber uma notificação sobre o status de suas entregas, integrando a comunicação direta com a experiência digital.

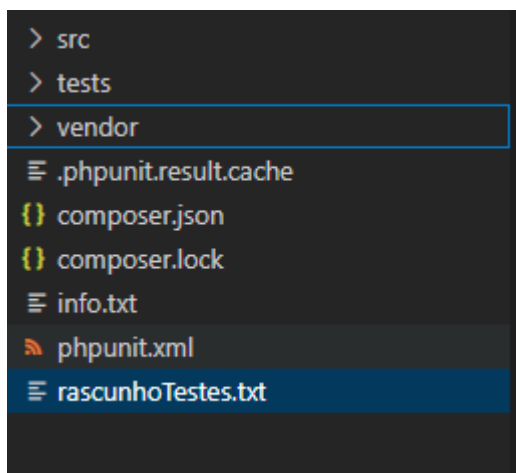
### 5.6 BACK-END

O sistema foi desenvolvido com organização clara de pastas e arquivos, conforme mostrado nas imagens a seguir. A estrutura separa o código front-end do back-end, com páginas específicas para cada função do sistema.

Os códigos implementados incluem desde funcionalidades básicas como login e cadastros, até integrações mais complexas com WhatsApp e cálculos automáticos de taxas de entrega.

A Figura 27 apresenta a arquitetura global do projeto, evidenciando a adoção de padrões de organização modernos. A estrutura inclui o diretório `src/` contendo o código-fonte principal, `tests/` para suíte de testes automatizados, e `vendor/` para gerenciamento de dependências via Composer. A presença de arquivos de configuração como `phpunit.xml` e `composer.json` demonstra a implementação de ferramentas de desenvolvimento profissional, assegurando a qualidade do código.

**Figura 27** - Estrutura de pastas do projeto



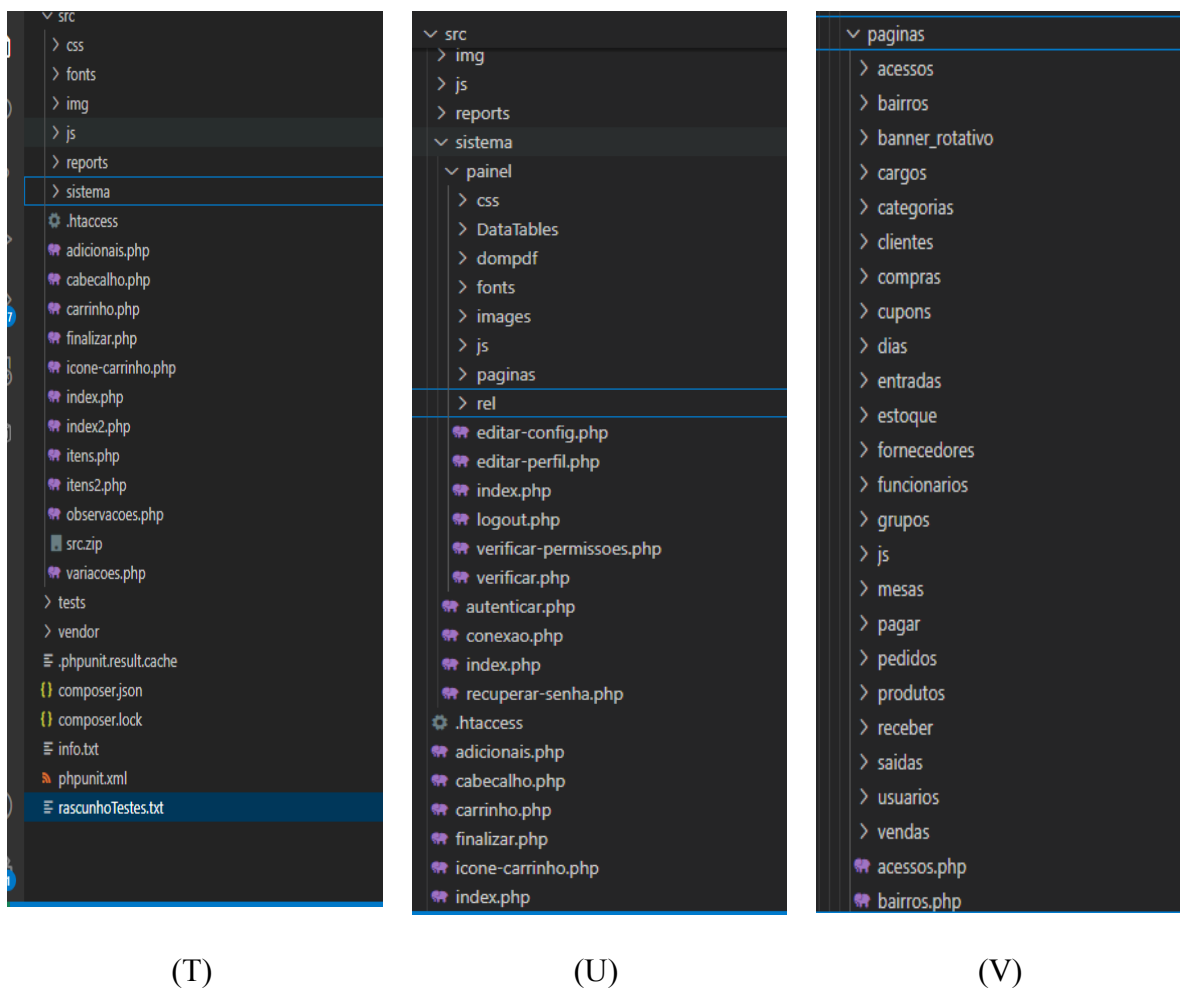
**Fonte:** Elaborado pelo autor(2025)

As Figuras 28 detalham a arquitetura modular do sistema, exemplificando a clara separação de responsabilidades. O front-end organiza-se em `css/`, `fonts/`, `img/` e `js/`, enquanto o back-end estrutura-se no diretório `sistema/painel/` com subdivisões funcionais. A organização em `paginas/` categoriza módulos administrativos (acessos, bairros, vendas, etc.), facilitando a

navegação e manutenção do código. Esta segmentação reflete princípios de arquitetura de software que promovem coesão e baixo acoplamento.

Imagem dividida em três partes para facilitar a visualização da estrutura semi aberta.

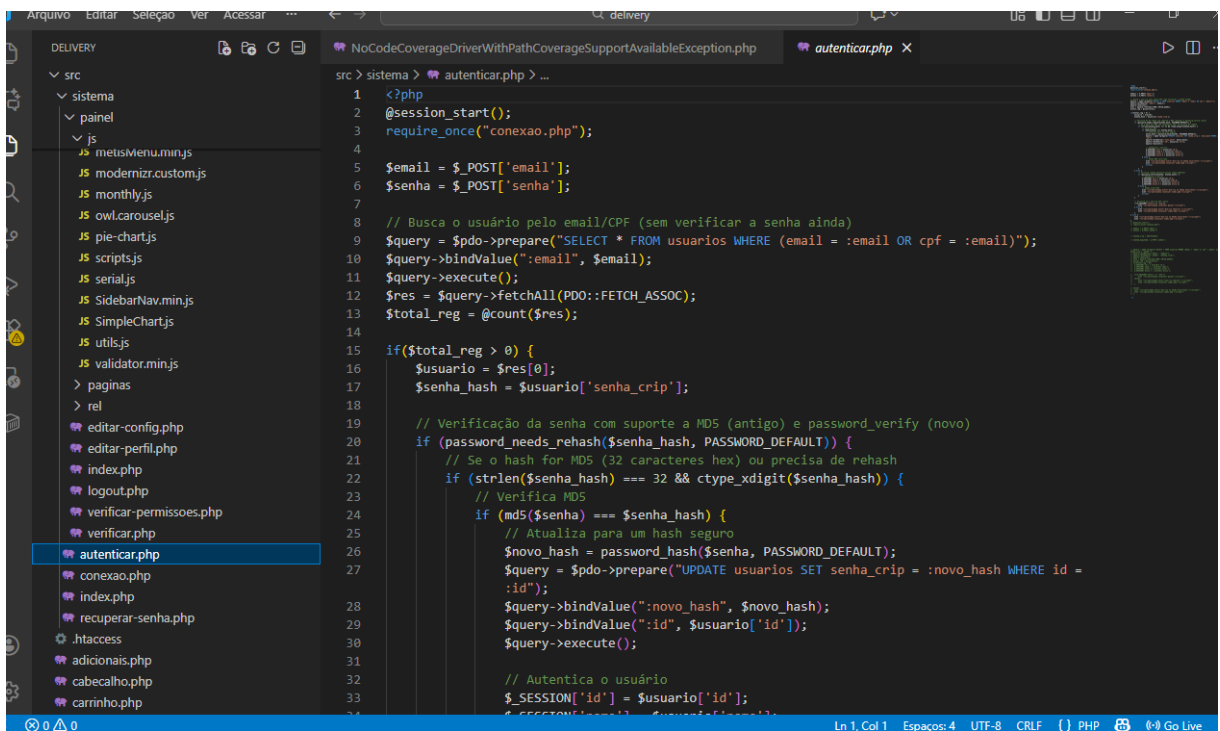
**Figura 28** - (T)Estrutura de pastas abertas 1,(U)Estrutura de pastas abertas 2 e (T)Estrutura de pastas abertas 3.,



Fonte: Elaborado pelo autor (2025)

A Figura 29 mostra o módulo de autenticação, destacando a implementação de um mecanismo de segurança evolutivo. O sistema realiza migração transparente de hashes MD5 legados para o algoritmo bcrypt via `password_hash()`, garantindo conformidade com práticas modernas de segurança. A dupla verificação suportando tanto credenciais históricas quanto o novo padrão, assegura transição sem impacto na experiência do usuário, enquanto validações contra injeção SQL via prepared statements fortalecem a resistência a ataques.

Figura 29 - Código de verificação de login



```
1 <?php
2 @session_start();
3 require_once("conexao.php");
4
5 $email = $_POST['email'];
6 $senha = $_POST['senha'];
7
8 // Busca o usuário pelo email/CPF (sem verificar a senha ainda)
9 $query = $pdo->prepare("SELECT * FROM usuarios WHERE (email = :email OR cpf = :email)");
10 $query->bindValue(":email", $email);
11 $query->execute();
12 $res = $query->fetchAll(PDO::FETCH_ASSOC);
13 $total_reg = @count($res);
14
15 if($total_reg > 0) {
16     $usuario = $res[0];
17     $senha_hash = $usuario['senha_crip'];
18
19     // Verificação da senha com suporte a MD5 (antigo) e password_verify (novo)
20     if (password_needs_rehash($senha_hash, PASSWORD_DEFAULT)) {
21         // Se o hash for MD5 (32 caracteres hex) ou precisa de rehash
22         if (strlen($senha_hash) === 32 && ctype_xdigit($senha_hash)) {
23             // Verifica MD5
24             if (md5($senha) === $senha_hash) {
25                 // Atualiza para um hash seguro
26                 $novo_hash = password_hash($senha, PASSWORD_DEFAULT);
27                 $query = $pdo->prepare("UPDATE usuarios SET senha_crip = :novo_hash WHERE id = :id");
28                 $query->bindValue(":novo_hash", $novo_hash);
29                 $query->bindValue(":id", $usuario['id']);
30                 $query->execute();
31
32                 // Autentica o usuário
33                 $_SESSION['id'] = $usuario['id'];
34                 $_SESSION['nome'] = $usuario['nome'];
35             }
36         }
37     }
38 }
```

**Fonte:** Elaborado pelo autor (2025)

A próxima imagem demonstra o algoritmo de cálculo de taxas de entrega, implementando lógica de negócio baseada em geolocalização. O sistema consulta dinamicamente a tabela bairros para determinar valores personalizados por região, com tratamento condicional para diferentes modalidades. A formatação monetária padronizada e o cálculo em tempo real do valor final exemplificam a atenção à experiência do usuário durante o processo.

**Figura 30** - Código de verificação de login

```

src > js > ajax > calcular-frete.php > ...
1 <?php
2 @session_start();
3 require_once('../sistema/conexao.php');
4
5 $bairro = $_POST['bairro'];
6 $total_compra = $_POST['total_compra'];
7 $entrega = $_POST['entrega'];
8
9
10
11 $query = $pdo->query("SELECT * FROM bairros where nome = '$bairro'");
12 $res = $query->fetchAll(PDO::FETCH_ASSOC);
13 if(@count($res) > 0){
14 $taxa_entrega = $res[0]['valor'];
15 }else{
16     $taxa_entrega = 0;
17 }
18
19 if($entrega != 'Delivery'){
20     $taxa_entrega = 0;
21 }
22 $taxa_entregaF = number_format($taxa_entrega, 2, ',', '.');
23
24 $final_compra = $total_compra + $taxa_entrega;
25 $final_compraF = number_format($final_compra, 2, ',', '.');
26
27 echo $taxa_entregaF . '-'. $final_compraF;
28 ?>
29
30

```

Fonte: Elaborado pelo autor (2025)

A Figura 31 evidencia a integração estratégica com WhatsApp como canal de comunicação pós-venda. A implementação gera mensagens estruturadas automaticamente, incluindo detalhes críticos do pedido (hora, total, forma de pagamento, previsão de entrega).

**Figura 31** - Integração com WhatsApp

```

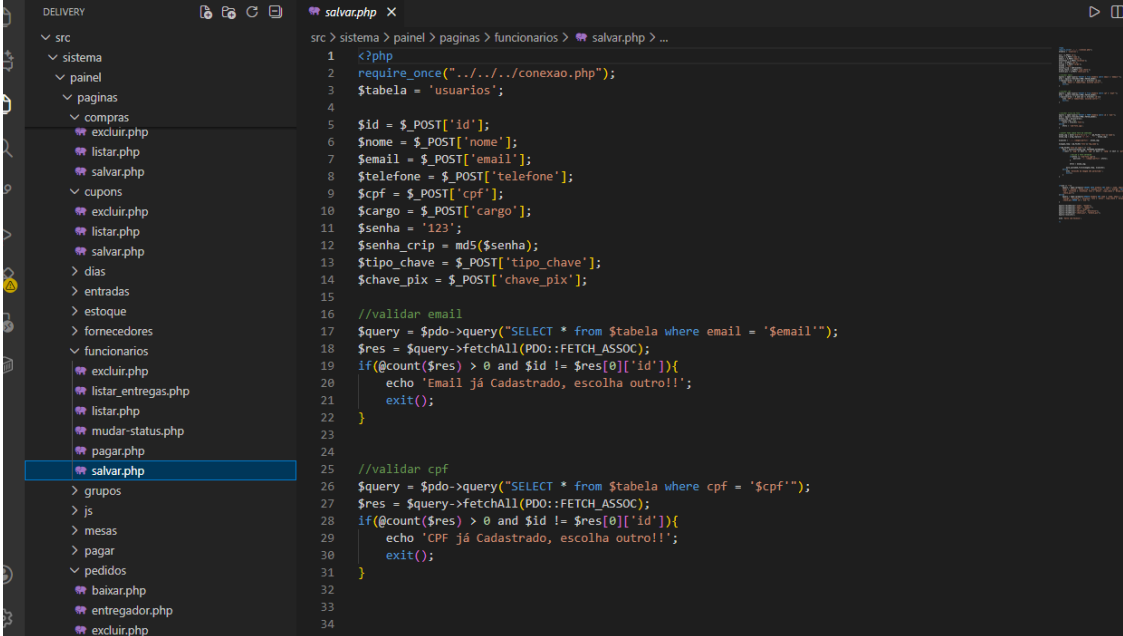
src > finalizar.php > ...
413 <script type="text/javascript">
414
415 function finalizarPedido(){
416     success:function(result){
417         setTimeout(()=>{
418             //
419             if(id_usuario == ""){
420                 window.location='index.php';
421             }else{
422                 window.location='sistema/panel/index.php?pagina=novo_pedido';
423             }
424         },500);
425
426         if(pedido_whatapp == 'Sim'){
427             let a= document.createElement('a');
428             //a.target= '_blank';
429             a.href= 'http://api.whatsapp.com/send?1=pt_BR&phone=?=$whatsapp_sistema?&text=
430 *Novo Pedido* %0A Hora: *' + hora + '* %0A Total: R$ *' + total_compra_finalF +
431 '* %0A Entrega: *' + entrega + '* %0A Pagamento: *' + pagamento + '* %0A Cliente:
432 *' + nome_cliente + '* %0A Previsão de Entrega: *' + result + '*';
433             a.click();
434         }else if(pedido_whatapp == 'Api'){
435             /*

```

Fonte: Elaborado pelo autor (2025)

A Figura 32 exemplifica a implementação padronizada de operações CRUD no sistema, destacando mecanismos robustos de validação de dados. As verificações de unicidade para email e CPF previnem inconsistências no banco de dados, enquanto o uso de prepared statements mitiga riscos de injeção SQL.

**Figura 32** - Cadastro de funcionários



```
1 <?php
2 require_once("../../conexao.php");
3 $tabela = 'usuarios';
4
5 $id = $_POST['id'];
6 $nome = $_POST['nome'];
7 $email = $_POST['email'];
8 $telefone = $_POST['telefone'];
9 $cpf = $_POST['cpf'];
10 $cargo = $_POST['cargo'];
11 $senha = '123';
12 $senha_crip = md5($senha);
13 $tipo_chave = $_POST['tipo_chave'];
14 $chave_pix = $_POST['chave_pix'];
15
16 //validar email
17 $query = $pdo->query("SELECT * from $tabela where email = '$email'");
18 $res = $query->fetchAll(PDO::FETCH_ASSOC);
19 if(@count($res) > 0 and $id != $res[0]['id']){
20     echo 'Email já Cadastrado, escolha outro!';
21     exit();
22 }
23
24 //validar cpf
25 $query = $pdo->query("SELECT * from $tabela where cpf = '$cpf'");
26 $res = $query->fetchAll(PDO::FETCH_ASSOC);
27 if(@count($res) > 0 and $id != $res[0]['id']){
28     echo 'CPF já Cadastrado, escolha outro!';
29     exit();
30 }
31
32
33
34
```

**Fonte:** Elaborado pelo autor (2025)

A Figura 33 apresenta a lógica complexa de determinação do status operacional do estabelecimento. O algoritmo considera múltiplos fatores: status manual do administrador, dia da semana através de consulta à tabela dias, e faixa horária comercial.

**Figura 33** - Cadastro de funcionários

```

1 <?php require_once("cabecalho.php");
2
3 $img = 'aberto.png';
4
5 if($status_estabelecimento == "Fechado"){
6     $img = 'fechado.png';
7 }
8
9
10 $data = date('Y-m-d');
11 //verificar se está aberto hoje
12 $diasemana = array("Domingo", "Segunda-Feira", "Terça-Feira", "Quarta-Feira", "Quinta-Feira",
13 "Sexta-Feira", "Sábado");
14 $diasemana_numero = date('w', strtotime($data));
15 $dia_procurado = $diasemana[$diasemana_numero];
16
17 //percorrer os dias da semana que ele trabalha
18 $query = $pdo->query("SELECT * FROM dias where dia = '$dia_procurado'");
19 $res = $query->fetchAll(PDO::FETCH_ASSOC);
20 if(@count($res) > 0){
21     $img = 'fechado.png';
22 }
23
24 $hora_atual = date('H:i:s');
25
26 //nova verificação de horarios
27 $start = strtotime( date('Y-m-d' . $horario_abertura) );
28 $end = strtotime( date('Y-m-d' . $horario_fechamento) );
29 $now = time();
30 if ( $start <= $now && $now <= $end ) {
31
32 }else{
33     $img = 'fechado.png';
34 }

```

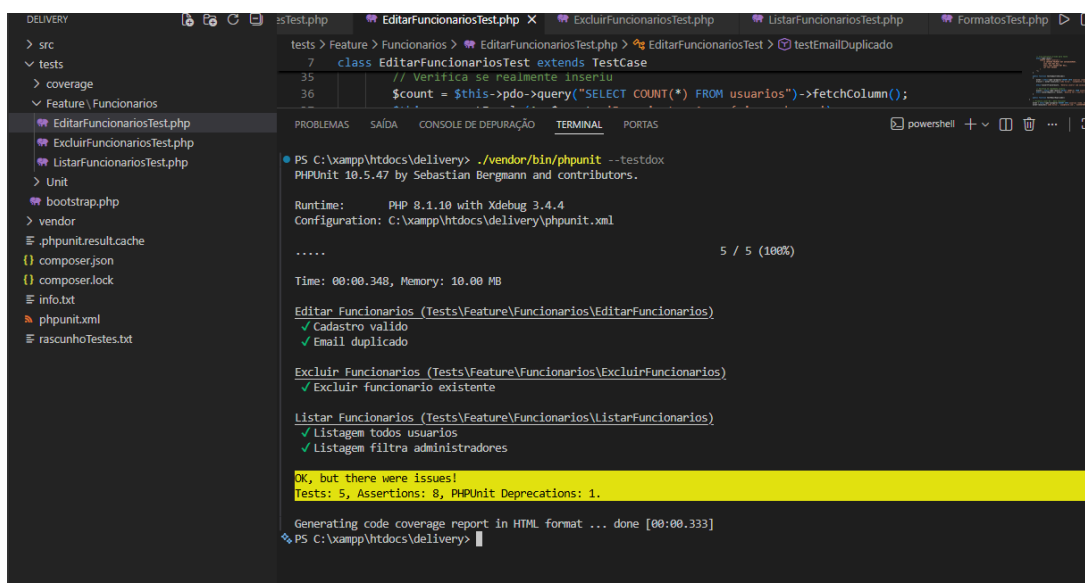
Fonte: Elaborado pelo autor (2025)

## 5.7 TESTES

A implementação do sistema foi submetida a uma bateria de testes para verificação de sua qualidade e funcionalidade. Abaixo são detalhados alguns dos testes representativos que foram executados, incluindo testes unitários, funcionais e de performance, os quais comprovam a robustez da solução desenvolvida.

Os testes CRUD (Create, Read, Update, Delete) para a funcionalidade de funcionários foram executados com sucesso, conforme demonstrado na imagem. O teste `EditarFuncionariosTest` validou dois cenários críticos: cadastro válido e tratamento de email duplicado. A ferramenta PHPUnit foi utilizada para automação dos testes, apresentando um resultado positivo com 5 testes executados e 8 asserções validadas. A cobertura de código em HTML foi gerada automaticamente, permitindo analisar visualmente as partes do sistema que foram testadas. Este processo assegura que as operações básicas de gerenciamento de funcionários funcionem corretamente e previne regressões no código.

Figura 34 - Teste CRUD funcionários



```
tests > Feature > Funcionarios > EditarFuncionariosTest.php > EditarFuncionariosTest > testEmailDuplicado
7 class EditarFuncionariosTest extends TestCase
35 // Verifica se realmente inseriu
36 $count = $this->pdo->query("SELECT COUNT(*) FROM usuarios")->fetchColumn();
..... 5 / 5 (100%)

Time: 00:00.348, Memory: 10.00 MB

Editar Funcionarios (Tests\Feature\Funcionarios\EditarFuncionarios)
✓ Cadastro valido
✓ Email duplicado

Excluir Funcionarios (Tests\Feature\Funcionarios\ExcluirFuncionarios)
✓ Excluir funcionario existente

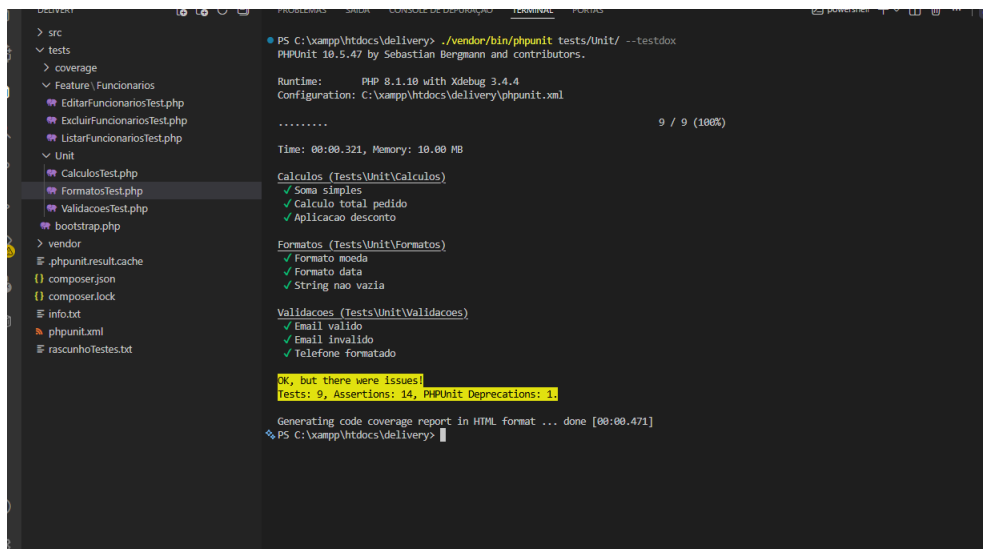
Listar Funcionarios (Tests\Feature\Funcionarios\ListarFuncionarios)
✓ Listagem todos usuarios
✓ Listagem filtra administradores

OK, but there were issues!
Tests: 5, Assertions: 8, PHPUnit Deprecations: 1.

Generating code coverage report in HTML format ... done [00:00.333]
PS C:\xampp\htdocs\delivery>
```

Fonte: Elaborado pelo autor (2025)

Os testes unitários foram executados com sucesso para três componentes principais do sistema: Cálculos, Formatos e Validações. Foram realizados 9 testes com 14 asserções, abrangendo funcionalidades essenciais como cálculos de pedidos com descontos, formatação de moeda e datas, e validação de emails e telefones. A suite de testes unitários garante que as operações matemáticas, formatação de dados e validações de entrada funcionem corretamente de forma isolada, proporcionando confiabilidade nas operações básicas do sistema antes da integração com outros componentes.

**Figura 35** - Testes cálculo formatos e validações

```
> src
> tests
  > coverage
    > Feature\Funcionarios
      EditarFuncionariosTest.php
      ExcluirFuncionariosTest.php
      ListarFuncionariosTest.php
    > Unit
      CalculosTest.php
      FormatosTest.php
      ValidacoesTest.php
      bootstrap.php
  > vendor
    phpunit.result.cache
    composer.json
    composer.lock
    info.txt
    phpunit.xml
    rascunhoTestes.txt

PS C:\xampp\htdocs\delivery> ./vendor/bin/phpunit tests/Unit/ --testdox
PHPUnit 10.5.47 by Sebastian Bergmann and contributors.

Runtime:       PHP 8.1.10 with Xdebug 3.4.4
Configuration: C:\xampp\htdocs\delivery\phpunit.xml

.....
Time: 00:00.321, Memory: 10.00 MB

Calculos (Tests\Unit\Calculos)
  ✓ Soma simples
  ✓ Calculo total pedido
  ✓ Aplicacao desconto

Formatos (Tests\Unit\Formatos)
  ✓ Formato moeda
  ✓ Formato data
  ✓ String nao vazia

Validacoes (Tests\Unit\Validacoes)
  ✓ Email valido
  ✓ Email invalido
  ✓ Telefone formatado

OK, but there were issues!
Tests: 9, Assertions: 14, PHPUnit Deprecations: 1.

Generating code coverage report in HTML format ... done [00:00.471]
PS C:\xampp\htdocs\delivery>
```

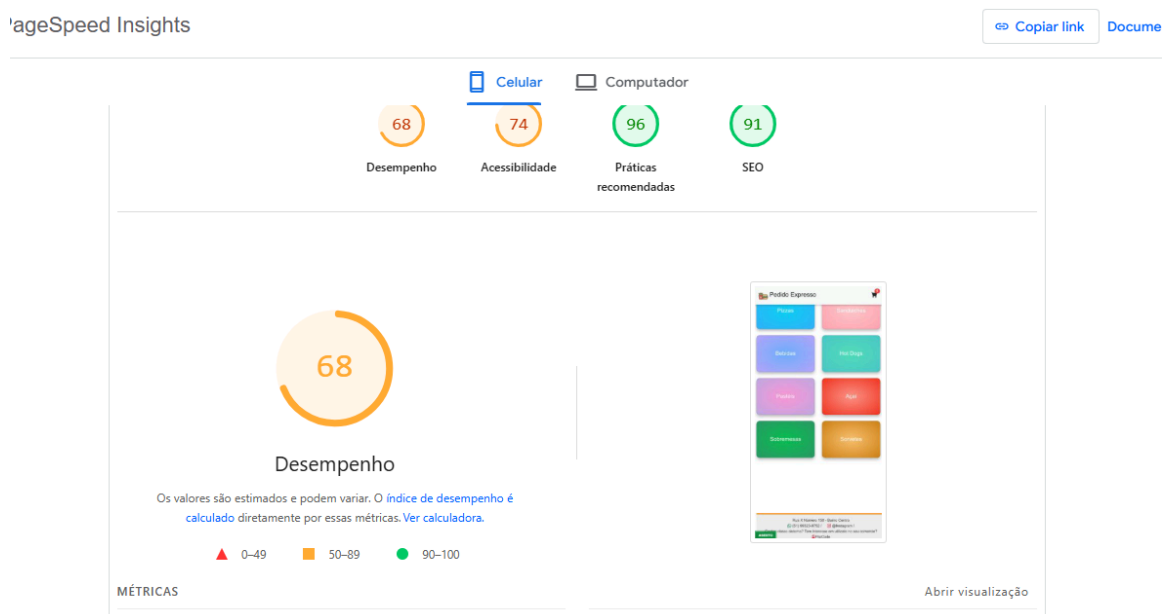
**Fonte:** Elaborado pelo autor (2025)

A análise de performance realizada através do PageSpeed Insights<sup>6</sup> revelou indicadores importantes sobre a qualidade técnica da aplicação. O sistema obteve pontuação de 68 em desempenho, 74 em acessibilidade, 96 em práticas recomendadas e 91 em SEO. Estes resultados demonstram que a aplicação atende bem aos critérios de SEO e boas práticas de desenvolvimento, porém há oportunidades de melhoria no desempenho geral. As métricas coletadas servem como base para otimizações futuras, especialmente no carregamento de páginas e experiência do usuário final.

---

<sup>6</sup> PageSpeed Insights é uma ferramenta para análise de Core Web Vitals e métricas de performance.

**Figura 36** - Teste de performance



**Fonte:** Elaborado pelo autor (2025)

### Teste de aceitação com usuário final

O teste prático durou um dia de operação no restaurante de um voluntário, após o uso do sistema foi perguntado ao dono do estabelecimento o que ele achou.

O sistema demonstrou potencial significativo para otimização do atendimento. O cardápio digital mostrou-se intuitivo, permitindo rápida adaptação, enquanto a organização automática dos pedidos eliminou inconsistências comuns no método anterior via WhatsApp.

Observou-se que pessoas de idade mais avançada mantiveram preferência pelo atendimento telefônico tradicional, indicando a necessidade de canais paralelos durante a transição tecnológica. A integração com WhatsApp mostrou-se particularmente eficaz, mantendo a comunicação direta com clientes, porém de forma estruturada.

A experiência, embora breve, evidenciou a capacidade de modernizar operações mantendo a essência do atendimento personalizado.

## 6 CONCLUSÃO

O presente trabalho atingiu com êxito seu objetivo principal de desenvolver um sistema de delivery online adaptado para dispositivos móveis, atendendo às crescentes demandas do mercado por soluções que aliam praticidade e eficiência na gestão de pedidos. A relevância do tema se confirma pelo crescimento exponencial do comércio eletrônico e pela necessidade premente de plataformas que otimizem processos em um cenário onde mobilidade e conectividade são determinantes para o sucesso comercial.

Durante o desenvolvimento, aplicaram-se conceitos fundamentais de Engenharia de Software, desde o levantamento de requisitos até a modelagem de dados e implementação de interfaces responsivas. A combinação de HTML, CSS, Bootstrap e jQuery no front-end resultou em uma interface amigável e adaptável a diferentes dispositivos, enquanto PHP e MySQL garantiram robustez e integração eficiente no back-end, complementados pela hospedagem na HostGator que assegurou disponibilidade e estabilidade essenciais para aplicações de e-commerce.

A adoção da metodologia Scrum permitiu entregas incrementais e ajustes ágeis, demonstrando especial eficácia na superação de desafios como a definição de taxas de entrega sem integração com APIs externas e a otimização do carregamento das páginas. Os testes realizados atestaram a conformidade do sistema com os requisitos funcionais e não funcionais, validando tanto a funcionalidade quanto a usabilidade em diversos contextos de uso.

Os impactos e benefícios da solução são significativos: para estabelecimentos, significa maior controle operacional, redução de erros e otimização de tempo; para clientes, representa comodidade, agilidade e transparência no acompanhamento de pedidos. Economicamente, a plataforma reduz custos com processos manuais e amplia a competitividade de negócios locais, enquanto socialmente contribui para a democratização do acesso à tecnologia, permitindo que pequenos empreendedores compitam com grandes plataformas.

Academicamente, este projeto consolida a aplicação prática de conceitos de Desenvolvimento Web, Banco de Dados e Metodologias Ágeis, reforçando a integração entre teoria e prática. Profissionalmente, proporcionou experiência em tecnologias de mercado, preparando o autor para projetos de maior complexidade. Os desafios enfrentados - desde a

definição arquitetural até a implementação responsiva e a ausência de integração com APIs externas - transformaram-se em oportunidades de aprendizado e aprimoramento de habilidades técnicas e gerenciais.

Como perspectivas futuras, destacam-se a integração com sistemas de pagamento online, implementação de cálculo dinâmico de rotas via APIs de geolocalização, desenvolvimento de aplicativo nativo e, potencialmente, a adoção de recursos de inteligência artificial para recomendação de produtos e atendimento automatizado via chat.

Em síntese, o projeto não apenas atingiu seus objetivos propostos, demonstrando viabilidade técnica e funcional, mas também consolidou-se como ferramenta com potencial de expansão e aprimoramento contínuo. A experiência adquirida reforça a relevância da busca por soluções criativas adaptadas às necessidades de mercado, evidenciando que a tecnologia, quando aplicada estrategicamente, transforma realidades e gera valor para todos os envolvidos, contribuindo simultaneamente para o avanço tecnológico e acadêmico.

## REFERÊNCIAS

ANOTA AI. Sistema para delivery: conheça tudo sobre a Anota AI. Disponível em: [https://anota.ai/blog/sistema-para-delivery/]. Acesso em: 26 ago. 2025.

APACHE FRIENDS. Download XAMPP. Disponível em: [https://www.apachefriends.org/download.html]. Acesso em: 31 de outubro de 2025.

BOOTSTRAP. History · Bootstrap. Disponível em: [https://getbootstrap.com/docs/4.0/about/history/]. Acesso em: 30 de outubro de 2025.

BOOTSTRAP. Components · Bootstrap v5.0. Disponível em: [https://getbootstrap.com/docs/5.0/customize/components/]. Acesso em: 30 de outubro de 2025.

BOOTSTRAP. Grid system · Bootstrap. Disponível em: [https://getbootstrap.com/docs/4.0/layout/grid/]. Acesso em: 30 de outubro de 2025.

CNN, Entenda como funciona e quando usar a metodologia Scrum. CNN,2023. Disponível:[https://www.cnnbrasil.com.br/tecnologia/scrum/ ] Acesso em: 30 de julho de 2025.

DEV MEDIA. Os melhores artigos sobre PHP: do básico ao avançado. Disponível em: [https://www.devmedia.com.br/artigos/php]. Acesso em: 9 outubro 2025.

DESIDERIO, Mariana. Delivery tem taxa exorbitante e é quase monopólio do iFood, diz Giraffa. uol,2023. Disponível em:[https://economia.uol.com.br/noticias/redacao/2023/08/14/giraffas.htm]. Acesso em: 30 de julho de 2025.

EPOCANEGOCIOS, Tecnologia leva mais eficiência aos restaurantes Globo, 2021. Disponível em: [https://epocanegocios.globo.com/Empreendedorismo/noticia/2021/10/tecnologia-leva-mais-e-eficiencia-aos-restaurantes.html] . Acesso em: 30 de julho de 2025.

FONSECA, João. Metodologia da pesquisa científica. 2002. Disponível em: [https://www.google.com.br/books/edition/Apostila\_de\_metodologia\_da\_pesquisa\_cien/oB5x2SChpSEC?hl=pt-BR&gbpv=1&dq=metodologia+de+pesquisa&printsec=frontcover] Acesso em: 30 de julho de 2025.

G1,Black Friday: lojas online apostam na inteligência artificial para acelerar a entrega dos produtos .Jornal da Globo, 2023. Disponível em:[https://g1.globo.com/jornal-da-globo/noticia/2023/11/25/black-friday-lojas-online-apostam-na-inteligencia-artificial-para-acelerar-a-entrega-dos-produtos.ghtml]. Acesso em: 30 de julho de 2025.

G1. Uber Eats deixa de funcionar no Brasil a partir desta terça,2022. Disponível em: [https://g1.globo.com/tecnologia/noticia/2022/03/07/uber-eats-deixa-de-funcionar-no-brasil-a-partir-desta-terca.ghtml]. Acesso em:28 de setembro de 2025.

**G1.** Por que a 99Food é a cara do brasileiro? G1, 14 nov. 2025. Disponível em: [https://g1.globo.com/especial-publicitario/99-food/noticia/2025/11/14/por-que-a-99food-e-a-cara-do-brasileiro.ghtml]. Acesso em: 20 de novembro de 2025.

**HOSTGATOR BRASIL.** PHP do Zero: Guia Completo para Iniciantes (2025). Disponível em: [https://www.hostgator.com.br/blog/php-guia-basico/]. Acesso em: 9 outubro 2025.

**I FOOD.** O que é o iFood? Conheça a história e a operação da empresa. Ifood, 2023. Disponível em: [https://institucional.ifood.com.br/noticias/o-que-e-o-ifood/]. Acesso em: 30 de julho de 2025.

**I FOOD.** O que é o iFood?, 2025. Disponível em: [https://institucional.ifood.com.br/institucional/o-que-e-o-ifood/]. Acesso em: 28 de setembro de 2025.

**KINSTA.** 27 melhores tutoriais para aprender PHP em 2025. Disponível em: [https://kinsta.com/pt/blog/tutoriais-de-php/]. Acesso em: 13 outubro 2025.

**LOCKHART, Josh.** Modern PHP: New Features and Good Practices. Sebastopol: O'Reilly Media, 2015.

**MADUREIRA, Daniele.** Como apps de entrega estão levando pequenos restaurantes à falência. BBC, 2020. Disponível em: [https://www.bbc.com/portuguese/geral-51272233]. Acesso em: 30 de julho de 2025.

**MARIADB CORPORATION.** MariaDB Documentation. Disponível em: [https://mariadb.com/docs/]. Acesso em: 31 de outubro de 2025.

**MARIADB FOUNDATION.** MariaDB Server Documentation. Disponível em: [https://mariadb.org/documentation/]. Acesso em: 31 de outubro de 2025.

**MORAES, André.** A evolução do PHP: da ferramenta de página pessoal à linguagem de programação moderna e eficiente. Disponível em: [https://www.dio.me/articles/a-evolucao-do-php-da-ferramenta-de-pagina-pessoal-a-linguagem-de-programacao-moderna-e-eficiente-917d72be918f]. Acesso em: 13 outubro 2025.

**MOZILLA DEVELOPER NETWORK (MDN).** Design Responsivo. Disponível em: [https://developer.mozilla.org/pt-BR/docs/Learn\_web\_development/Core/CSS\_layout/Responsive\_Design]. Acesso em: 28 de outubro de 2025.

**MOZILLA DEVELOPER NETWORK (MDN).** Responsive web design. Disponível em: [https://developer.mozilla.org/en-US/docs/Learn\_web\_development/Core/CSS\_layout/Responsive\_Design]. Acesso em: 28 de outubro de 2025.

**MOZILLA DEVELOPER NETWORK (MDN).** HTML: HyperText Markup Language. Disponível em: [https://developer.mozilla.org/pt-BR/docs/Web/HTML]. Acesso em: 27 de outubro de 2025.

MOZILLA DEVELOPER NETWORK (MDN). Media technologies on the web. Disponível em: [<https://developer.mozilla.org/en-US/docs/Web/Media>]. Acesso em: 27 de outubro de 2025.

NIXON, Robin. Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. Sebastopol: O'Reilly Media, 2018.

ORACLE.O que é um chatboot . Oracle, 2024. Disponível em:[<https://www.oracle.com/br/chatbots/what-is-a-chatbot/>].Acesso em: 30 de julho de 2025.

PRESSMAN, Roger S. Engenharia de Software. 8. ed. São Paulo: McGraw-Hill, 2016.

PHP. Manual do PHP. Disponível em: [[https://www.php.net/manual/pt\\_BR/index.php](https://www.php.net/manual/pt_BR/index.php)]. Acesso em: 15 outubro 2025.

PHP. História do PHP – Manual. Disponível em: [[https://www.php.net/manual/pt\\_BR/history.php.php](https://www.php.net/manual/pt_BR/history.php.php)]. Acesso em: 15 outubro 2025.

RAPPI.Bem-vindo à Minha Loja, 2025. Disponível em: [[https://mitienda.rappi.com.br/?\\_ga=2.58577450.1830580252.1759102853-684272524.1759102853&\\_gl=1](https://mitienda.rappi.com.br/?_ga=2.58577450.1830580252.1759102853-684272524.1759102853&_gl=1)]. Acesso em: 28 de setembro de 2025.

SCHWABER, Ken; SUTHERLAND, Jeff. The Scrum Guide. Scrum.org, 2020. Disponível em: [<https://scrumguides.org>]. Acesso em: 01 de novembro de 2025.

SILVA, Dayvisson Soares da. App de entrega de comida e o mercado digital conectado: um estudo de caso da empresa Ifood sobre a ótica da economia da informação. 2021. Disponível em: [<https://www.repositorio.ufal.br/handle/123456789/10461>]. Acesso em: 26 ago. 2025.

SUTHERLAND, Jeff. Scrum: a arte de fazer o dobro do trabalho na metade do tempo. Rio de Janeiro: Leya, 2014.

TERRA, Como saber se é hora de sua pequena empresa ter um app? Terra,2024. Disponível em: [<https://www.terra.com.br/economia/como-saber-se-e-hora-de-sua-pequena-empresa-ter-um-app,e1d4d37c4c604fb655e21a73c64e986eezkuansm.html>]. Acesso em: 30 de julho de 2025.

WHATSAPP. Sobre o whatsapp?. Whatsapp, 2024. Disponível em:[[https://www.whatsapp.com/about?lang=pt\\_BR#:~:text=Sobre%20o%20WhatsApp&text=O%20WhatsApp%20%C3%A9%20gratuito2,maneira%20simples%2C%20segura%20e%20confi%C3%A1vel.&text=E%20sim%2C%20o%20nome%20WhatsApp,'E%20a%20%C3%AD%3F](https://www.whatsapp.com/about?lang=pt_BR#:~:text=Sobre%20o%20WhatsApp&text=O%20WhatsApp%20%C3%A9%20gratuito2,maneira%20simples%2C%20segura%20e%20confi%C3%A1vel.&text=E%20sim%2C%20o%20nome%20WhatsApp,'E%20a%20%C3%AD%3F)"].Acesso em: 30 de julho de 2025.

WHATWG. HTML Living Standard. Disponível em: [<https://html.spec.whatwg.org/multipage/>]. Acesso em: 27 de outubro de 2025.

WORLD WIDE WEB CONSORTIUM (W3C). A brief history of CSS until 2016. Disponível em: [<https://www.w3.org/Style/CSS20/history.html>]. Acesso em: 28 de outubro de 2025.

WORLD WIDE WEB CONSORTIUM (W3C). Introduction to CSS3. 2001. Disponível em: [<https://www.w3.org/TR/2001/WD-css3-roadmap-20010523/>]. Acesso em: 28 de outubro de 2025.

WORLD WIDE WEB CONSORTIUM (W3C). CSS Snapshot 2025. Disponível em: [<https://www.w3.org/Style/CSS/Overview.en.html>]. Acesso em: 28 de outubro de 2025.

WORLD WIDE WEB CONSORTIUM (W3C). HTML5 – A vocabulary and APIs for HTML and XHTML. 2014. Disponível em: [<https://www.w3.org/TR/2014/REC-html5-20141028/>]. Acesso em: 27 de outubro de 2025.