

Luís Fernando da Silva Corrêa

# **Modelagem de Aplicativo de Delivery para Economia Solidária**

Formiga - MG

2025

Luís Fernando da Silva Corrêa

# **Modelagem de Aplicativo de Delivery para Economia Solidária**

Monografia do Trabalho de Conclusão de Curso apresentado ao Instituto Federal Minas Gerais - Campus Formiga, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais

Campus Formiga

Ciência da Computação

Orientador: Prof. Dr. Alexandre Pimenta

Coorientador: Prof. Eduardo Miranda

Formiga - MG

2025

Corrêa, Luís Fernando da Silva

C824m            Modelagem de aplicativo de delivery para economia solidária / Luís Fernando da Silva Corrêa -- Formiga : IFMG, 2025.

111 p. : il.

Orientador: Prof. Dr. Alexandre Pimenta

Co-orientador: Eduardo Miranda

Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – *Campus* Formiga.

1. Delivery solidário. 2.Cooperativas digitais. 3.Economia Local. 4. Plataformas alternativas. 5. Autogestão tecnológica. I. Pimenta, Alexandre. II. Miranda, Eduardo. III. Título.

CDD 004

*Ficha catalográfica elaborada pela Bibliotecária Lívia Renata Santos- CRB/6-2561*

Luís Fernando da Silva Corrêa

## **Modelagem de Aplicativo de Delivery para Economia Solidária**

Monografia do Trabalho de Conclusão de Curso apresentado ao Instituto Federal Minas Gerais - Campus Formiga, como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Trabalho aprovado em 20 de Junho de 2025.

BANCA EXAMINADORA

---

**Prof. Dr. Alexandre Pimenta**  
Orientador

---

**Prof. Eduardo Miranda**  
Coorientador

---

**Prof. Dr. Bruno Ferreira**  
Convidado 1

---

**Prof. Dr. Carlos Bernardes**  
Convidado 2

Formiga - MG  
2025

*Este trabalho é dedicado à minha família e à sociedade, até aqueles com quem não concordo.*

# Agradecimentos

Agradeço, em primeiro lugar, aos meus pais, pelo apoio incondicional e pelos valores que me permitiram chegar até aqui. Expresso minha gratidão a todos os professores que contribuíram para minha formação ao longo do curso, em especial ao meu orientador, Prof. Dr. Alexandre Pimenta, por acreditar na proposta deste trabalho, pela orientação precisa e pelo incentivo constante. Também agradeço ao meu coorientador, Prof. Eduardo Miranda, pela disponibilidade, pelas contribuições valiosas e pelo suporte técnico e acadêmico durante todo o desenvolvimento deste projeto.

*“Duas coisas são infinitas: o universo e a estupidez humana. Mas, em relação ao universo, ainda não tenho certeza absoluta.” (Albert Einstein)*

# Resumo

Este trabalho apresenta a proposta de um modelo de sistema de delivery colaborativo, desenvolvido como alternativa às plataformas privadas tradicionais. O objetivo é oferecer uma solução de fácil replicação por cooperativas, associações comunitárias ou outras formas de organização baseadas na economia solidária, com menor custo operacional e foco na valorização da economia local. A metodologia adotada incluiu análise comparativa de aplicativos existentes, levantamento de requisitos e elaboração de protótipos com base em dados coletados por meio de formulários aplicados a consumidores, empreendedores e entregadores. Como resultado, foi obtida uma documentação visual e conceitual, que pode servir como base para o desenvolvimento de soluções comunitárias mais justas, transparentes e sustentáveis.

**Palavras-chave:** delivery solidário; cooperativas digitais; economia local; plataformas alternativas; autogestão tecnológica.

# Abstract

This work presents the proposal of a collaborative delivery system model, developed as an alternative to traditional private platforms. The goal is to offer a solution that can be easily replicated by cooperatives, community associations, or other forms of organization based on the solidarity economy, with lower operational costs and a focus on strengthening the local economy. The adopted methodology included a comparative analysis of existing applications, requirements gathering, and the creation of prototypes based on data collected through surveys applied to consumers, entrepreneurs, and delivery workers. As a result, visual and conceptual documentation was produced, which can serve as a foundation for the development of fairer, more transparent, and sustainable community-based solutions.

**Keywords:** solidarity delivery; digital cooperatives; local economy; alternative platforms; technological self-management.

# Lista de ilustrações

Figura 1 – Exemplificação do processo A -> B utilizando o pedido ao restaurante	18
Figura 2 – Exemplificação do processo A -> B utilizando o pedido ao motorista de aplicativo . . . . .	19
Figura 3 – Matriz de stakeholders: classificação com base em poder e interesse . .	29
Figura 4 – Modelo entidade-relacionamento representando a base de dados de um sistema escolar. . . . .	30
Figura 5 – Diagrama de casos de uso aplicado a um sistema escolar, evidenciando as funcionalidades associadas a cada perfil de usuário. . . . .	32
Figura 6 – Exemplo de diagrama de sequência representando o fluxo de lançamento de nota por um professor em um sistema escolar. . . . .	33
Figura 7 – Exemplo de diagrama UML da arquitetura geral do sistema, com destaque para a separação entre frontend, backend e acesso ao banco de dados. . . . .	34
Figura 8 – Exemplo ilustrativo da fase de prototipação com esboços de telas . . .	35
Figura 9 – Protótipo de interface de usuário (UI) com elementos visuais e disposição funcional para avaliação de experiência (UX). . . . .	36
Figura 10 – Representação esquemática do modelo cascata, com suas fases sequenciais	37
Figura 11 – Ciclo do modelo iterativo e incremental, evidenciando a entrega contínua e progressiva do software. . . . .	38
Figura 12 – Representação do modelo espiral, destacando seus ciclos iterativos com foco em riscos . . . . .	39
Figura 13 – Diagrama representando os princípios fundamentais do <i>Extreme Programming</i> (XP). . . . .	40
Figura 14 – Quadro Kanban ilustrando o fluxo de trabalho dividido em etapas visuais: Fazer, Fazendo e Feito. . . . .	40
Figura 15 – Fluxo visual de uma sprint no Scrum, incluindo backlog, planejamento, execução e entrega. . . . .	41
Figura 16 – Representação do ciclo contínuo DevOps, integrando desenvolvimento, testes, operação e monitoramento . . . . .	42
Figura 17 – Comparação entre valores de assinatura mensal das plataformas de delivery. . . . .	48
Figura 18 – Percentual de comissão por pedido retido pelas plataformas. . . . .	48
Figura 19 – Comparação das taxas de entrega aplicadas pelas plataformas. . . . .	49
Figura 20 – Taxa de serviço adicionada aos pedidos em plataformas de delivery. . .	49
Figura 21 – Diagrama de Modelo de Domínio do sistema proposto . . . . .	52
Figura 22 – Atores Principais . . . . .	63

Figura 23 – Diagrama de sequência do processo de realização de pedido pelo cliente	65
Figura 24 – Diagrama de sequência do processo de entrega realizada pelo entregador	66
Figura 25 – Diagrama Entidade-Relacionamento (ER) simplificado do sistema . . .	68
Figura 26 – Tela Inicial . . . . .	70
Figura 27 – Tela de Login . . . . .	71
Figura 28 – Tela de Inicio . . . . .	72
Figura 29 – Tela com Lista de Estabelecimentos . . . . .	73
Figura 30 – Tela de Pedidos Recebidos (Restaurante) . . . . .	74
Figura 31 – Tela Recebimento de Pedidos ao Entregador . . . . .	75
Figura 32 – Arquitetura geral da solução de delivery solidário . . . . .	77
Figura 33 – Versionamento e CI/CD . . . . .	78
Figura 34 – Diagramas de Casos de Uso da Plataforma de Delivery Solidário . . . .	80
Figura 35 – Casos de Uso - Cliente, Restaurante . . . . .	80
Figura 36 – Casos de Uso - Entregador, Administrador . . . . .	80

# Lista de tabelas

Tabela 1 – Comparação entre Metodologias de Desenvolvimento . . . . .	42
Tabela 2 – Comparação de funcionalidades entre plataformas de delivery . . . . .	46
Tabela 3 – Comparação de taxas entre diferentes plataformas de delivery . . . . .	47
Tabela 4 – Onde encontrar as informações sobre as taxas e funcionalidades . . . . .	49

# Lista de Abreviaturas e Siglas

**ANPD** Autoridade Nacional de Proteção de Dados

**API** Application Programming Interface

**app** Aplicativo

**AWS** Amazon Web Services

**CCPA** California Consumer Privacy Act

**CD** Continuous Delivery

**CEP** Código de Endereçamento Postal

**CI** Continuous Integration

**CPF** Cadastro de Pessoa Física

**CSS** Cascading Style Sheets

**DevOps** Contração de Development e Operations

**DPIA** Data Protection Impact Assessment

**EUA** Estados Unidos da América

**ER** Entidade-Relacionamento

**FAQ** Frequently Asked Questions

**FTC** Federal Trade Commission

**GDPR** General Data Protection Regulation

**HTML** HyperText Markup Language

**HTTPS** Hypertext Transfer Protocol Secure

**iOS** iPhone Operating System

**LGPD** Lei Geral de Proteção de Dados Pessoais

**MVP** Minimum Viable Product

**PHP** Personal Home Page

**PIX** Nome criado pelo Banco Central, não é sigla

**RESTful** Representational State Transfer

**SCL** Strategic Communication Laboratories

**SIG** Sistema Integrado de Gestão

**SIGAA** Sistema Integrado de Gestão de Atividades Acadêmicas

**SMS** Short Message Service

**SOLID** Single Responsibility Principle. Open-Closed Principle. Liskov Substitution Principle. Interface Segregation Principle. Dependency Inversion Principle

**SSO** Single Sign-On

**TCC** Trabalho de Conclusão de Curso

**TDD** Test-Driven Development

**UE** União Europeia

**UI** User Interface

**UX** User Experience

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
<b>1.1</b>	<b>Análise de Dados e o Caso Cambridge Analytica</b>	<b>20</b>
<b>1.2</b>	<b>Justificativa</b>	<b>21</b>
<b>1.3</b>	<b>Objetivos</b>	<b>22</b>
1.3.1	Objetivo Geral	22
1.3.2	Objetivos Específicos	22
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>23</b>
<b>2.1</b>	<b>LGPD</b>	<b>23</b>
2.1.1	Lei Geral de Proteção de Dados (LGPD) - Brasil	23
2.1.2	Lei de Privacidade do Consumidor da Califórnia (CCPA) - Estados Unidos	24
2.1.3	Regulamento Geral sobre a Proteção de Dados (GDPR) - Europa	25
2.1.4	Métodos Utilizados na LGPD	25
<b>2.2</b>	<b>Engenharia de Software e Princípios Aplicados</b>	<b>27</b>
2.2.1	Levantamento de Requisitos	28
2.2.2	Stakeholders	28
2.2.3	Banco de Dados	29
2.2.4	Diagramas	31
2.2.4.1	Diagrama de Casos de Uso	31
2.2.4.2	Diagrama de Sequência	32
2.2.4.3	Diagrama UML	33
2.2.5	Fase de Prototipação	34
2.2.6	UI/UX	35
2.2.7	Metodologias de Desenvolvimento de <i>Software</i>	36
2.2.7.1	Modelo Cascata ( <i>Waterfall</i> )	36
2.2.7.2	Desenvolvimento Iterativo e Incremental	37
2.2.7.3	Modelo Espiral	38
2.2.7.4	Metodologias Ágeis	39
2.2.7.5	DevOps	41
2.2.8	Comparação entre Metodologias	42
<b>2.3</b>	<b>Trabalhos Relacionados</b>	<b>44</b>
2.3.1	AiqFome	45
2.3.2	iFood	46
2.3.3	UaiRango	46
2.3.4	Funcionalidades dos Aplicativos	46

2.3.5	Taxas Cobradas . . . . .	47
2.3.6	Onde Encontrar as Informações . . . . .	49
<b>3</b>	<b>DESENVOLVIMENTO . . . . .</b>	<b>51</b>
<b>3.1</b>	<b>Análise de Requisitos . . . . .</b>	<b>51</b>
3.1.1	Especificação Técnica . . . . .	53
3.1.2	LGPD . . . . .	54
<b>3.2</b>	<b>Levantamento de Requisitos . . . . .</b>	<b>54</b>
3.2.1	Engenharia de Requisitos . . . . .	54
3.2.2	Entrevistas com Stakeholders . . . . .	55
3.2.3	Metodologia de Elaboração dos Formulários . . . . .	56
3.2.3.1	Formulário Consumidores . . . . .	57
3.2.3.2	Formulário Entregador . . . . .	58
3.2.3.3	Formulário Empreendedores . . . . .	60
3.2.4	Análise de Sistemas Existentes . . . . .	62
3.2.5	Criação de Documentação Detalhada . . . . .	62
3.2.6	Modelagem de Casos de Uso . . . . .	63
3.2.7	Diagrama de Sequência . . . . .	64
3.2.8	Especificação Funcional . . . . .	66
<b>3.3</b>	<b>Design e Implementação . . . . .</b>	<b>66</b>
3.3.1	Design de Banco de Dados . . . . .	67
3.3.2	UI/UX e Especificação da Interface . . . . .	68
3.3.3	Definição de Componentes Modulares . . . . .	75
3.3.4	Escolha da tecnologia . . . . .	77
3.3.5	Atividades de Desenvolvimento . . . . .	77
3.3.6	Estimativa de Custos . . . . .	78
3.3.7	Casos de Uso . . . . .	78
3.3.7.1	Cliente - Cadastrar-se . . . . .	81
3.3.7.2	Cliente - Fazer Login . . . . .	81
3.3.7.3	Cliente - Realizar Pedido . . . . .	81
3.3.7.4	Cliente - Acompanhar Entrega . . . . .	82
3.3.7.5	Cliente - Avaliar Pedido . . . . .	82
3.3.7.6	Entregador ou Restaurante - Solicitar Cadastro . . . . .	82
3.3.7.7	Restaurante - Gerenciar Menu do Dia . . . . .	83
3.3.7.8	Entregador - Aceitar Entrega . . . . .	83
3.3.7.9	Entregador - Atualizar Status da Entrega . . . . .	83
3.3.7.10	Restaurante - Editar Cardápio em Tempo Real . . . . .	84
3.3.7.11	Administrador - Gerenciar Usuários . . . . .	84
3.3.7.12	Administrador - Acompanhar Estatísticas do Sistema . . . . .	84
3.3.7.13	Administrador - Visualizar Relatórios . . . . .	85

3.3.7.14	Administrador - Gerenciar Categorias e Lojas . . . . .	85
3.3.7.15	Administrador - Moderar Avaliações . . . . .	85
3.3.7.16	Administrador - Aprovar Cadastros Pendentes . . . . .	86
3.3.8	Papéis e Adaptação no Projeto . . . . .	86
3.3.8.1	Papéis e Adaptação no Projeto . . . . .	86
3.3.8.2	Exemplo de Aplicação Prática . . . . .	87
<b>4</b>	<b>CONCLUSÃO . . . . .</b>	<b>88</b>
<b>5</b>	<b>TRABALHOS FUTUROS . . . . .</b>	<b>89</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>90</b>
	<b>APÊNDICES . . . . .</b>	<b>94</b>
	<b>APÊNDICE A – RESPOSTAS CONSUMIDORES . . . . .</b>	<b>95</b>
	<b>APÊNDICE B – RESPOSTAS EMPREENDEDORES . . . . .</b>	<b>99</b>
	<b>APÊNDICE C – RESPOSTAS ENTREGADORES . . . . .</b>	<b>107</b>

# 1 Introdução

A pandemia de *COVID-19* acelerou significativamente a digitalização em diversos setores, impulsionando o uso de aplicativos e tecnologias digitais tanto por empresas quanto por consumidores. Com as restrições impostas pelo isolamento social, houve uma necessidade urgente de adaptação às novas realidades, levando a uma transformação digital sem precedentes.

No Brasil, esse movimento foi particularmente evidente. Aplicativos de *delivery*, transporte, bancos digitais, redes sociais e plataformas de videoconferência se tornaram parte do cotidiano de milhões de brasileiros. Segundo a empresa de análise de mercado [app Annie](#), o Brasil foi o quinto país que mais baixou aplicativos em 2021, totalizando mais de 10 bilhões de downloads naquele ano, um reflexo direto da digitalização forçada pela pandemia ([AppsFlyer, 2021](#)).

Outro dado relevante é que, de acordo com a pesquisa da TIC Domicílios 2022, realizada pelo [Cetic.br](#), o uso de aplicativos de entrega no Brasil cresceu de 30% para 52% da população conectada entre 2019 e 2021, sendo mais expressivo nas classes A e B, mas com avanço notável também nas classes C e D. Aplicativos de bancos digitais, por sua vez, foram utilizados por mais de 60% dos internautas brasileiros em 2021, evidenciando uma migração massiva de serviços físicos para o meio digital ([CETIC.br, 2022](#)).

Mesmo no período pós-pandêmico, o hábito do uso de aplicativos se manteve. Um estudo da empresa Globo em parceria com a [Kantar IBOPE](#) mostrou que 79% dos brasileiros continuaram utilizando aplicativos de *delivery* com frequência em 2023, enquanto 67% relataram fazer compras online com regularidade. O comportamento digital tornou-se parte da rotina da população, apontando uma consolidação dos aplicativos como mediadores centrais entre oferta e demanda de produtos e serviços no país ([Globo; Kantar IBOPE, 2023](#)).

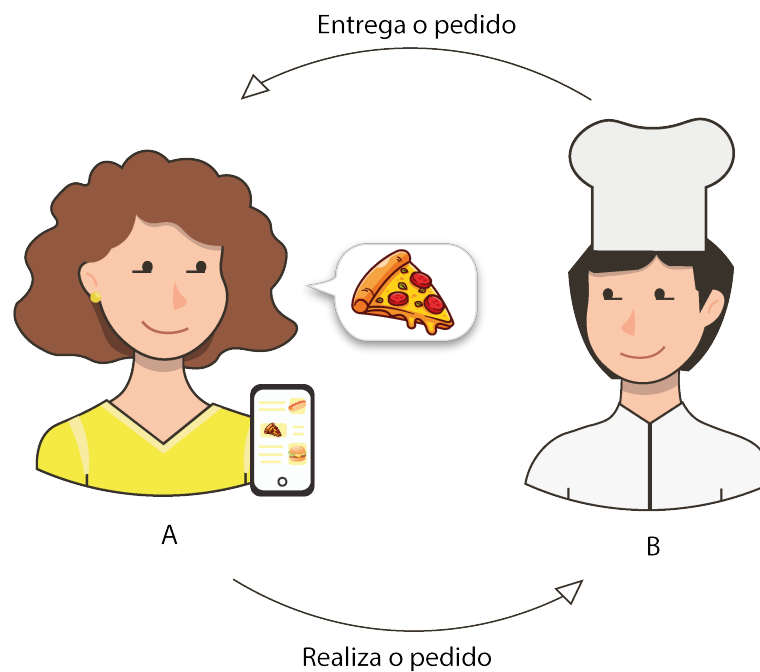
Nesse contexto, este trabalho propõe a modelagem de um sistema de *delivery* baseado em princípios da economia solidária, como autogestão, cooperação e valorização da comunidade. A ideia é fornecer um modelo técnico e organizacional que possa ser adotado por cooperativas de entregadores, associações de bairro, redes de produtores locais ou iniciativas comunitárias autônomas, com foco na redução de custos operacionais, na proteção de dados e na construção de um ecossistema digital mais justo e descentralizado.

O projeto busca demonstrar que é possível estruturar tecnologicamente uma plataforma de entregas capaz de funcionar de forma coletiva e ética, sem a dependência de corporações centralizadoras. Com base em dados reais de usuários e boas práticas de desenvolvimento, foi desenhado um modelo funcional que serve como ponto de partida

para soluções comunitárias de delivery, construídas por e para os próprios participantes da cadeia produtiva.

Nos aplicativos de serviços, geralmente há dois pontos principais: o Ponto A, que representa o consumidor final, e o Ponto B, que é o fornecedor do serviço ou produto. Por exemplo, em um aplicativo de delivery de alimentos, o cliente que solicita uma pizza é o Ponto A, enquanto o restaurante que prepara e fornece a pizza é o Ponto B, Figura 1. Da mesma forma, em aplicativos de transporte, o passageiro que solicita uma corrida é o Ponto A, e o motorista que realiza o transporte é o Ponto B, Figura 2. Essa dinâmica estabelece uma conexão direta entre quem oferece e quem consome, facilitada pela plataforma digital.

Figura 1 – Exemplificação do processo A -> B utilizando o pedido ao restaurante



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

Figura 2 – Exemplificação do processo A -> B utilizando o pedido ao motorista de aplicativo



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

Aplicativos de serviços, operados por empresas privadas, têm como principal objetivo a maximização dos lucros. Para alcançar esse fim, muitas vezes impõem taxas elevadas aos fornecedores que utilizam a plataforma. Essas taxas adicionais levam os fornecedores a aumentarem os preços dos produtos ou serviços oferecidos dentro do aplicativo, resultando em um custo maior para o consumidor final. Assim, a busca incessante pelo lucro por parte das plataformas pode criar um ciclo onde tanto fornecedores quanto consumidores são financeiramente prejudicados.

O presente projeto propõe a criação de um modelo de plataforma de *delivery*, que visa oferecer uma alternativa aos grandes aplicativos privados. A proposta é que o sistema seja estruturado de forma modular, documentada e acessível, permitindo que coletivos, cooperativas ou redes de economia solidária possam replicá-lo e adaptá-lo com facilidade. Ao disponibilizar um modelo replicável e tecnicamente viável, busca-se não apenas a democratização do acesso tecnológico, mas também a valorização da economia local, a redução de taxas operacionais para empreendedores e a prestação de um serviço mais justo e transparente à população. Esse tipo de iniciativa reforça a autonomia das comunidades na gestão de suas próprias ferramentas digitais, promovendo maior equidade e protagonismo social.

## 1.1 Análise de Dados e o Caso Cambridge Analytica

A análise de dados tornou-se um componente central para organizações públicas e privadas na atualidade, sendo aplicada em áreas como negócios, saúde, segurança, educação e política. Seu papel vai além da simples coleta de informações: trata-se de extrair conhecimento e gerar valor estratégico por meio da interpretação de grandes volumes de dados. A tomada de decisão baseada em dados, em vez de suposições, conduz a resultados mais precisos, eficientes e inovadores. Com o avanço das tecnologias de armazenamento e processamento, tornou-se viável capturar, tratar e analisar dados em tempo real, elevando a importância dessa prática na era digital.

No entanto, a capacidade de analisar dados em larga escala também levanta preocupações éticas e jurídicas, especialmente quando há uso indevido de informações pessoais. Um dos casos mais emblemáticos nesse sentido foi o escândalo envolvendo a empresa *Cambridge Analytica* (ANALYTICA, 2025). Fundada em 2013 como subsidiária da SCL Group, a empresa se especializou em combinar mineração e corretagem de dados com estratégias de comunicação política. Sua atuação ganhou notoriedade por desenvolver perfis psicológicos detalhados de eleitores a partir de dados obtidos de redes sociais, permitindo segmentação precisa de mensagens eleitorais.

O escândalo veio à tona em 2018, quando foi revelado que a empresa havia coletado e utilizado indevidamente dados pessoais de cerca de 87 milhões de usuários do Facebook. Essas informações foram obtidas através de um aplicativo de teste de personalidade desenvolvido por um acadêmico, que, além de capturar dados dos usuários que o acessavam, também coletava dados de seus contatos sem consentimento explícito. Com esses dados, a *Cambridge Analytica* criou campanhas políticas altamente direcionadas, influenciando eventos como as eleições presidenciais dos Estados Unidos em 2016 e o referendo do Brexit no Reino Unido (GUARDIAN, 2025).

Os principais pontos do escândalo envolvem:

- **Aquisição de dados:** obtenção de informações pessoais sem consentimento através de um aplicativo disfarçado de teste psicológico;
- **Uso indevido de dados:** elaboração de campanhas de microtargeting político com base em perfis comportamentais dos eleitores;
- **Impacto eleitoral:** influência direta em decisões políticas e manipulação da opinião pública;
- **Consequências legais:** processos judiciais em diversos países e uma multa de 5 bilhões de dólares aplicada à Meta/Facebook pela FTC dos Estados Unidos;

- **Encerramento das atividades:** dissolução da empresa em 2018 diante da pressão pública e institucional.

Esse caso tornou-se um marco global sobre os riscos do uso indiscriminado de dados e catalisou o debate sobre privacidade digital, governança da informação e a necessidade urgente de regulamentações robustas. A repercussão do episódio contribuiu diretamente para a criação e fortalecimento de legislações como o [GDPR](#) na União Europeia e a [LGPD](#) no Brasil, refletindo um movimento global por maior responsabilidade no uso de dados pessoais. Dessa forma, a análise de dados segue sendo uma ferramenta poderosa, mas que requer responsabilidade ética, transparência e conformidade legal para que seu potencial seja utilizado em benefício da sociedade.

## 1.2 Justificativa

O modelo desenvolvido neste trabalho foi estruturado com o objetivo de ser facilmente adaptável e replicável por organizações comunitárias, como cooperativas de entregadores, redes de produtores locais e associações de bairro. Ao fornecer um guia detalhado e um passo a passo claro para a implementação do aplicativo, espera-se que essas iniciativas possam adotar e personalizar a solução conforme suas necessidades específicas, promovendo a autogestão e a soberania digital no contexto local.

A privatização de serviços essenciais, como o fornecimento de energia elétrica, tem gerado debates em todo o mundo sobre sua real eficácia. Um exemplo recente é o caso da empresa italiana *Enel*, que assumiu a distribuição de energia em São Paulo após a privatização da Eletropaulo. Desde então, foram registradas diversas falhas no serviço, como longos períodos de apagão e aumentos nas tarifas, gerando insatisfação generalizada na população. Em contraste, outros países têm seguido caminho oposto: segundo reportagem do ([Brasil de Fato, 2024](#)), há uma tendência global de reestatização de serviços de energia, com mais de mil reversões de privatizações ocorrendo em países como Alemanha, França e Argentina, motivadas pela má prestação de serviços por empresas privadas. Enquanto isso, no Brasil, insiste-se em modelos que priorizam o lucro em detrimento da qualidade, mesmo diante de evidências internacionais contrárias.

Nos últimos anos, diversos países europeus têm revertido processos de privatização em setores essenciais, trazendo empresas de volta ao controle público - a esse movimento dá-se o nome de reestatização. Essa tendência é especialmente notável em serviços como água, energia e transporte público.

Na França, por exemplo, várias cidades, incluindo Paris, reverteram a privatização dos serviços de água, retornando à gestão pública devido a preocupações com a qualidade do serviço e aumento de tarifas sob administração privada. ([Transnational Institute, 2017](#))

Na Alemanha, o setor energético tem sido um foco significativo desse movimento. Desde 2000, foram registrados 284 casos de remunicipalização relacionados à energia, refletindo uma insatisfação com a gestão privada e uma busca por maior controle local sobre recursos energéticos. ([Energy Transition, 2017](#))

Além disso, um estudo publicado na *Journal of Economic Policy Reform* aponta que mais de 1.600 casos de remunicipalização foram registrados em mais de 70 países, abrangendo serviços como água, eletricidade e saúde. ([PALM; MALM; HÖRNSTRÖM, 2023](#))

Esses exemplos ilustram uma mudança significativa na gestão de serviços públicos na Europa, onde governos locais e nacionais estão reassumindo o controle de setores fundamentais para garantir qualidade, acessibilidade e responsabilidade social na prestação desses serviços.

A proposta de uma plataforma cooperativa também visa promover uma redução significativa nas taxas cobradas de fornecedores e entregadores, eliminando intermediários que impõem altos percentuais sobre cada transação. Essa diminuição de custos operacionais tende a gerar reflexos diretos na formação do preço final ao consumidor, permitindo que produtos e serviços sejam ofertados a valores mais acessíveis. Em um cenário onde a margem de lucro dos pequenos empreendedores é estreita e o poder de compra da população é cada vez mais pressionado, um modelo que favorece economicamente todas as partes envolvidas se mostra não apenas viável, mas necessário.

## 1.3 Objetivos

### 1.3.1 Objetivo Geral

Modelar um sistema de delivery de forma que ele seja escalável, replicável e de fácil implementação por cooperativas, associações ou redes comunitárias, visando à melhor distribuição dos valores repassados aos participantes da plataforma, à redução das taxas normalmente cobradas por grandes empresas do setor e ao tratamento ético e seguro dos dados pessoais envolvidos, em conformidade com a [LGPD](#).

### 1.3.2 Objetivos Específicos

1. Fazer busca de trabalhos similares
2. Fazer a modelagem do sistema e seus casos de uso
3. Estudo para uso e acesso aos dados de acordo com as novas normas da [LGPD](#)

## 2 Fundamentação Teórica

Este capítulo apresenta os principais conceitos e referências que fundamentam o desenvolvimento do sistema proposto. Inicialmente, discute-se o papel central que a análise de dados ocupa na sociedade contemporânea. Hoje, grande parte das empresas de tecnologia baseia seus modelos de negócio na coleta, processamento e monetização de dados dos usuários, explorando essas informações para personalização de serviços, publicidade direcionada e tomada de decisão estratégica.

Como exemplo emblemático do uso controverso desses dados, destaca-se o caso da *Cambridge Analytica*, empresa que ganhou notoriedade ao utilizar indevidamente informações extraídas de redes sociais, resultando em um escândalo global envolvendo manipulação de processos eleitorais e violação de privacidade.

Diante da repercussão de casos como esse, diversos países passaram a adotar legislações específicas para proteger os dados pessoais de seus cidadãos e regular a atuação de empresas no ambiente digital (2.1). Tais medidas refletem a crescente preocupação com a privacidade e segurança da informação em escala global.

Além da reflexão sobre dados e privacidade, este capítulo também contempla conceitos fundamentais da engenharia de software que embasam o desenvolvimento do sistema proposto. São abordadas práticas como o levantamento de requisitos, a modelagem com diagramas UML, a definição de estrutura de banco de dados e a prototipação de interfaces. Essas ferramentas, aliadas a uma metodologia ágil como o Scrum, proporcionam uma base sólida para a construção de soluções digitais que atendam às necessidades reais dos usuários e respeitem princípios éticos e técnicos.

### 2.1 Lei Geral de Proteção de Dados (LGPD)

A Lei Geral de Proteção de Dados (LGPD) no Brasil, a Lei de Privacidade do Consumidor da Califórnia (CCPA) nos Estados Unidos e o Regulamento Geral sobre a Proteção de Dados (GDPR) na Europa são as principais legislações que regulamentam a proteção de dados pessoais em suas respectivas regiões.

#### 2.1.1 Lei Geral de Proteção de Dados (LGPD) - Brasil

A LGPD (GOV.BR, 2025), Lei nº 13.709/2018, foi sancionada em agosto de 2018 e entrou em vigor em setembro de 2020. Seus principais pontos são:

- Aplicabilidade: Aplica-se a todas as empresas e organizações que coletam ou proces-

sam dados pessoais no Brasil, independentemente de onde estejam localizadas.

- **Dados pessoais e dados sensíveis:** Define dados pessoais como qualquer informação relacionada a uma pessoa identificada ou identificável, e dados sensíveis como aqueles que dizem respeito à origem racial ou étnica, convicção religiosa, opinião política, etc.
- **Direitos dos titulares:** Incluem acesso, correção, exclusão, portabilidade dos dados, e informações sobre o compartilhamento de dados.
- **Consentimento:** O tratamento de dados pessoais deve ser feito com o consentimento do titular, exceto em casos específicos previstos pela lei.
- **Segurança:** As empresas devem adotar medidas de segurança, técnicas e administrativas aptas a proteger os dados pessoais.
- **Autoridade Nacional de Proteção de Dados (ANPD):** Órgão responsável por zelar pela proteção dos dados pessoais e por regulamentar, fiscalizar e aplicar sanções.

### 2.1.2 Lei de Privacidade do Consumidor da Califórnia (CCPA) - Estados Unidos

A **CCPA** ([CALIFORNIA, 2025](#)) foi aprovada em junho de 2018 e entrou em vigor em janeiro de 2020. Embora seja uma legislação estadual, tem impacto nacional e internacional devido à relevância econômica da Califórnia. Seus principais pontos são:

- **Aplicabilidade:** Aplica-se a empresas que fazem negócios na Califórnia e que atendam a certos critérios, como receita bruta anual superior a US\$ 25 milhões, processam dados de 50.000 ou mais residentes da Califórnia, ou obtêm 50% ou mais de sua receita anual da venda de dados pessoais dos residentes.
- **Direitos dos consumidores:** Incluem o direito de saber quais dados pessoais estão sendo coletados, o direito de acesso, o direito de exclusão e o direito de não discriminação por exercerem seus direitos.
- **Venda de dados:** Dá aos consumidores o direito de optar por não ter seus dados vendidos a terceiros.
- **Segurança:** As empresas devem implementar e manter medidas de segurança razoáveis para proteger os dados pessoais.

### 2.1.3 Regulamento Geral sobre a Proteção de Dados (GDPR) - Europa

O [GDPR](#) ([COUNCIL, 2025](#)) entrou em vigor em maio de 2018 e é considerado uma das legislações de proteção de dados mais rigorosas do mundo. Seus principais pontos são:

- **Aplicabilidade:** Aplica-se a todas as empresas que processam dados de residentes da [UE](#), independentemente de onde a empresa esteja localizada.
- **Dados pessoais e dados sensíveis:** Define dados pessoais de forma ampla e inclui categorias especiais de dados sensíveis, como dados de saúde, orientação sexual, dados biométricos etc.
- **Direitos dos titulares:** Incluem o direito de acesso, retificação, exclusão ("direito ao esquecimento"), restrição de processamento, portabilidade dos dados, e o direito de objeção.
- **Consentimento:** Requer consentimento explícito e informado para o processamento de dados pessoais, exceto em casos específicos.
- **Segurança:** Exige que as empresas implementem medidas técnicas e organizacionais para proteger os dados pessoais.
- **Autoridades de Proteção de Dados:** Cada país membro tem uma autoridade independente responsável por monitorar a aplicação do [GDPR](#).

### 2.1.4 Métodos Utilizados na LGPD

A [LGPD](#) do Brasil estabelece uma série de medidas e práticas que organizações e indivíduos devem seguir para garantir a proteção dos dados pessoais. Alguns dos métodos utilizados para cumprir a [LGPD](#):

#### 1. Mapeamento de Dados

- **Inventário de Dados:** Identificar quais dados pessoais são coletados, processados e armazenados.
- **Fluxo de Dados:** Documentar o ciclo de vida dos dados dentro da organização.

#### 2. Governança e Gestão de Dados

- **Políticas de Privacidade:** Desenvolver e implementar políticas claras sobre o uso de dados pessoais.
- **Treinamento e Conscientização:** Capacitar funcionários sobre as práticas e responsabilidades relativas à proteção de dados.

### 3. Segurança da Informação

- Controle de Acesso: Restringir o acesso aos dados pessoais apenas para pessoas autorizadas.
- Criptografia: Usar criptografia para proteger dados sensíveis tanto em trânsito quanto em repouso.
- Monitoramento e Auditoria: Implementar sistemas de monitoramento e auditoria para detectar e responder a incidentes de segurança.

### 4. Gestão de Consentimento

- Coleta de Consentimento: Obter o consentimento explícito dos titulares dos dados para o processamento de seus dados.
- Gerenciamento de Preferências: Permitir que os titulares dos dados possam gerenciar suas preferências e retirar consentimentos quando desejarem.

### 5. Gerenciamento de Incidentes

- Plano de Resposta a Incidentes: Desenvolver e implementar um plano de resposta a incidentes de segurança de dados.
- Notificação de Violações: Notificar a [ANPD](#) e os titulares dos dados sobre violações de segurança.

### 6. Avaliação de Impacto à Proteção de Dados ([DPIA](#))

- Avaliação de Riscos: Realizar avaliações de impacto à proteção de dados para identificar e mitigar riscos associados ao processamento de dados pessoais.

### 7. Direitos dos Titulares

- Acesso e Transparência: Garantir que os titulares possam acessar seus dados e entender como eles são usados.
- Correção e Exclusão: Permitir que os titulares corrijam dados incorretos e solicitem a exclusão de seus dados.
- Portabilidade de Dados: Facilitar a transferência de dados pessoais de um controlador para outro.

### 8. Conformidade e Auditoria

- Auditorias Internas e Externas: Realizar auditorias periódicas para verificar a conformidade com a [LGPD](#).
- Relatórios de Conformidade: Manter registros detalhados das atividades de processamento de dados para comprovar conformidade.

Esses métodos ajudam as organizações a garantir que estão em conformidade com a [LGPD](#), protegendo os dados pessoais dos indivíduos e evitando penalidades legais.

## 2.2 Engenharia de Software e Princípios Aplicados

A Engenharia de Software é compreendida como “a aplicação de uma abordagem sistemática, disciplinada e quantificável ao desenvolvimento, operação e manutenção de software”, conforme definição do IEEE ([SOCIETY, 2014](#)). Ian Sommerville também define a área como “ramo da engenharia voltado a todos os aspectos da produção de software”, evidenciando sua abrangência multidisciplinar e foco na qualidade ([SOMMERVILLE, 2011](#)).

Essas definições reforçam que o desenvolvimento de software não deve ser conduzido de maneira improvisada, mas sim orientado por processos bem definidos, boas práticas e métricas de controle. Em projetos com múltiplos requisitos e perfis de usuário, como sistemas de delivery ou plataformas digitais com diferentes camadas de interação, a engenharia de software oferece a base metodológica necessária para garantir escalabilidade, robustez, manutenção e alinhamento com as necessidades dos envolvidos.

No contexto deste trabalho, a aplicação de técnicas e princípios da engenharia de software foi essencial para organizar e estruturar o desenvolvimento do sistema. As práticas adotadas incluíram:

- **Metodologias Ágeis**, como o *Scrum*, para promover ciclos iterativos de planejamento, execução e validação das funcionalidades;
- **Prototipação de Interface**, utilizando ferramentas como o *Figma*, para validar fluxos de navegação e usabilidade antes da implementação;
- **Engenharia de Requisitos**, com ênfase no levantamento de informações via formulários aplicados a diferentes públicos-alvo (clientes, entregadores e lojistas), assegurando uma visão ampla das necessidades reais do sistema;
- **Modelagem com Diagramas UML**, abrangendo os aspectos estruturais e comportamentais da aplicação por meio de diagramas de casos de uso, sequência e modelo de domínio;
- **Análise de Stakeholders**, contemplando os principais perfis envolvidos — clientes, entregadores, estabelecimentos e gestores — a fim de compreender diferentes interesses e alinhar os requisitos às expectativas dos usuários.

A adoção desses princípios resultou em uma base sólida para o desenvolvimento do sistema, facilitando a comunicação entre os participantes do projeto, organizando as

etapas e promovendo uma documentação estruturada que pode ser reutilizada, mantida e adaptada para outras realidades.

### 2.2.1 Levantamento de Requisitos

Entre as etapas fundamentais da engenharia de software está o levantamento de requisitos, responsável por identificar, documentar e validar as necessidades dos usuários e demais envolvidos no sistema. Essa atividade inicial é crítica para o sucesso do projeto, pois define o que deve ser desenvolvido e sob quais critérios será considerado funcional ou completo (MORTIER, 2023).

Os métodos utilizados nessa etapa incluem entrevistas, observações, formulários, *workshops* e protótipos interativos. O levantamento de requisitos tem como objetivo não apenas registrar funcionalidades desejadas, mas compreender os processos existentes, os objetivos institucionais e os problemas enfrentados pelos usuários no cotidiano. No caso de plataformas de delivery, por exemplo, pode-se investigar as dificuldades enfrentadas por entregadores, o tempo de espera percebido pelos clientes ou os obstáculos logísticos de comerciantes.

### 2.2.2 Stakeholders

Diretamente ligado ao levantamento de requisitos, em projetos colaborativos, especialmente aqueles que envolvem tecnologia e impacto social, é essencial identificar todos os grupos de pessoas ou organizações que são afetados ou que podem influenciar os resultados. Esses grupos são chamados de **stakeholders**, ou em português, *partes interessadas*. Isso inclui usuários finais (clientes), fornecedores (restaurantes), operadores (entregadores) e até os administradores ou cooperativas responsáveis pela manutenção da plataforma (OMG, 2025).

A correta identificação dos stakeholders é essencial para priorizar requisitos e prever conflitos ou interesses divergentes. Cada grupo possui objetivos e expectativas diferentes: clientes esperam agilidade e simplicidade, entregadores valorizam a eficiência e segurança, enquanto comerciantes buscam visibilidade e controle sobre pedidos e cardápios. Considerar essas perspectivas desde o início favorece o alinhamento entre tecnologia e realidade de uso.

A **matriz de stakeholders** é uma ferramenta visual que ajuda a entender como cada grupo deve ser tratado durante o desenvolvimento do projeto. Ela é organizada com base em dois critérios:

- **Poder:** o quanto esse grupo pode influenciar as decisões e o rumo do projeto;
- **Interesse:** o quanto esse grupo se importa com os resultados ou impactos do projeto.

A Figura 3 mostra uma matriz dividida em quatro quadrantes, cada um com uma estratégia específica de comunicação e engajamento:

1. **Gerenciar estreitamente (alto poder e alto interesse)**: são os stakeholders mais importantes. Precisam ser envolvidos diretamente e com frequência. No projeto, isso se aplica aos **Cientes**, pois eles têm alto interesse nos resultados e também podem influenciar decisões importantes.
2. **Manter satisfeitos (alto poder, mas baixo interesse)**: são grupos que têm influência, mas que não estão tão envolvidos com os detalhes. Devem ser bem tratados para evitar possíveis problemas no futuro. Os **Fornecedores** se encaixam nesse perfil.
3. **Manter informados (alto interesse, mas pouco poder)**: essas partes querem acompanhar o andamento, mesmo que não possam influenciar diretamente. É importante mantê-los atualizados. No caso do projeto, os **Administradores** se enquadram aqui.
4. **Monitorar (baixo poder e baixo interesse)**: são os grupos que não precisam de atenção constante, mas devem ser acompanhados para o caso de mudanças. **Operadores** e **Cooperativas** entram nesta categoria.

Essa classificação permite ao time focar esforços de comunicação e gestão onde eles são mais necessários, otimizando recursos e fortalecendo as relações com os envolvidos.

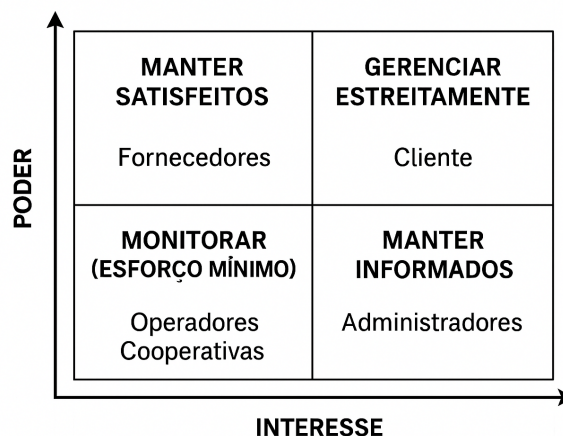


Figura 3 – Matriz de stakeholders: classificação com base em poder e interesse

### 2.2.3 Banco de Dados

Após o levantamento dos requisitos e a identificação dos *stakeholders*, é essencial projetar uma estrutura de dados que atenda às operações fundamentais do sistema escolar.

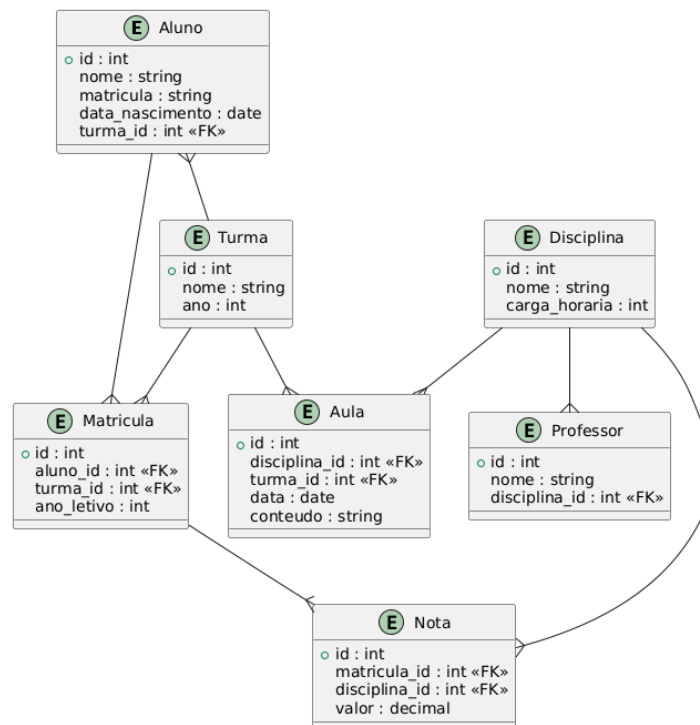
O banco de dados é o componente responsável por armazenar, organizar e disponibilizar essas informações de forma estruturada. Um projeto eficiente contempla três níveis: modelagem conceitual (entidades e relacionamentos), modelagem lógica (estrutura das tabelas) e modelagem física (otimização para desempenho e segurança) (SPRINGER, 2022).

No caso do sistema escolar apresentado, foram definidas entidades como **Aluno**, **Professor**, **Disciplina**, **Turma**, **Matricula**, **Aula** e **Nota**, que se relacionam para formar a base da lógica acadêmica. Por exemplo, a entidade **Matricula** realiza a ligação entre alunos e turmas, enquanto a entidade **Nota** associa avaliações aos alunos de forma integrada com as disciplinas. A entidade **Aula**, por sua vez, representa a ocorrência de uma disciplina em determinada turma, incluindo o conteúdo ministrado em cada data.

Essas relações garantem a integridade dos dados e possibilitam funcionalidades como lançamento de notas, controle de aulas e acompanhamento da vida acadêmica dos estudantes. Técnicas de normalização, uso de chaves estrangeiras e definição de restrições são adotadas para assegurar consistência e evitar redundâncias (MORTIER, 2023).

A Figura 4 apresenta a modelagem conceitual do banco de dados escolar, destacando as principais entidades e suas inter-relações.

Figura 4 – Modelo entidade-relacionamento representando a base de dados de um sistema escolar.



## 2.2.4 Diagramas

A representação visual do sistema é uma das ferramentas mais utilizadas em engenharia de software, especialmente na fase de modelagem. Diagramas facilitam a comunicação entre desenvolvedores, designers, analistas e stakeholders ao traduzirem conceitos abstratos em esquemas compreensíveis. Entre os diversos tipos de diagramas disponíveis na UML, destacam-se os de casos de uso, sequência e classes.

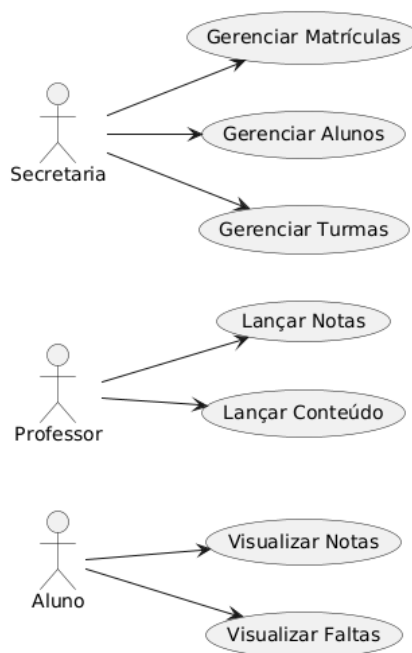
### 2.2.4.1 Diagrama de Casos de Uso

O diagrama de casos de uso é uma ferramenta essencial da UML para representar, de forma visual e simplificada, as interações entre os diferentes perfis de usuários (atores) e o sistema. Essa técnica é fundamental na fase de levantamento e análise de requisitos, pois permite comunicar claramente quais funcionalidades são esperadas de acordo com cada papel envolvido (PARADIGM, 2024).

Na Figura 5, observa-se um exemplo de aplicação de casos de uso em um sistema escolar. Três atores principais são representados: *Secretaria*, *Professor* e *Aluno*. A *Secretaria* é responsável por operações administrativas, como o gerenciamento de matrículas, alunos e turmas. Já o *Professor* possui funcionalidades voltadas à rotina pedagógica, como lançar notas e conteúdos das aulas. O *Aluno*, por sua vez, tem acesso à visualização de suas notas e faltas, promovendo transparência e acompanhamento de seu desempenho.

Esse tipo de diagrama contribui significativamente para a delimitação do escopo funcional da aplicação e para o alinhamento entre desenvolvedores, analistas e usuários finais, especialmente em ambientes acadêmicos com múltiplos perfis de uso.

Figura 5 – Diagrama de casos de uso aplicado a um sistema escolar, evidenciando as funcionalidades associadas a cada perfil de usuário.



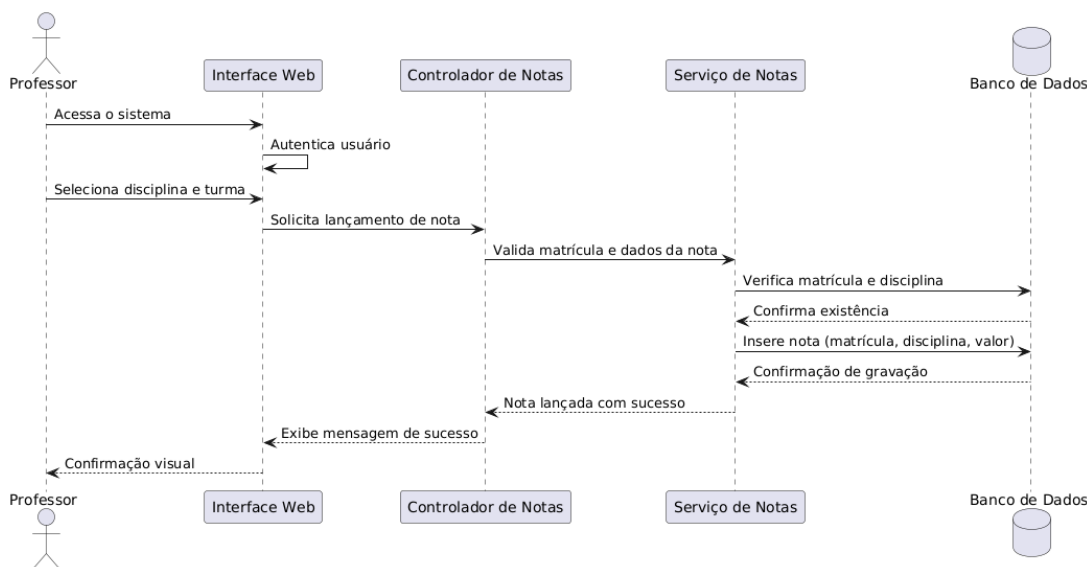
#### 2.2.4.2 Diagrama de Sequência

Complementando os diagramas de casos de uso, o diagrama de sequência representa o comportamento dinâmico de um sistema ao longo do tempo, evidenciando a troca de mensagens entre atores e componentes internos. Essa representação permite compreender o encadeamento lógico das ações, identificar responsabilidades de cada módulo e antecipar falhas no fluxo de comunicação (SPRINGER, 2022).

Na Figura 6, é apresentado um exemplo de diagrama de sequência aplicado ao cenário de um sistema escolar. O fluxo demonstra o processo de lançamento de notas por um professor, desde o acesso ao sistema e seleção de turma e disciplina, até a validação e registro da nota no banco de dados.

Esse tipo de diagrama é especialmente útil para validar o comportamento esperado do sistema, documentar a lógica de negócios e alinhar a equipe técnica quanto às dependências entre componentes. Além disso, contribui para a clareza na implementação e facilita a comunicação entre desenvolvedores e analistas de requisitos.

Figura 6 – Exemplo de diagrama de sequência representando o fluxo de lançamento de nota por um professor em um sistema escolar.



### 2.2.4.3 Diagrama UML

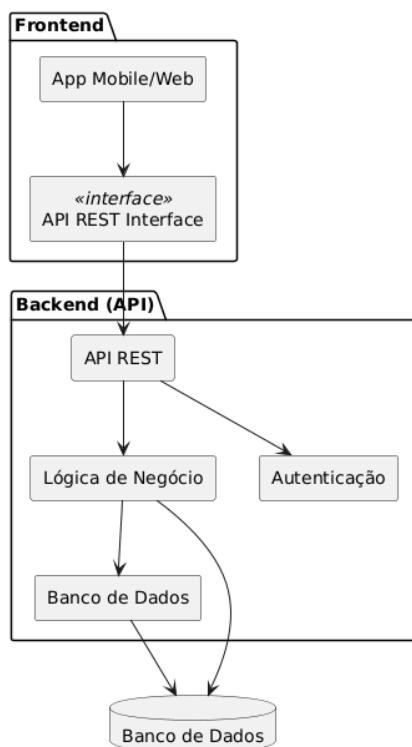
A *UML* (Unified Modeling Language) é um padrão amplamente adotado na engenharia de software para modelagem visual de sistemas. Seu principal objetivo é fornecer uma notação comum que possa ser compreendida por desenvolvedores, analistas, gerentes e stakeholders, promovendo uma comunicação clara e precisa sobre a estrutura e o comportamento de uma aplicação (RICCIARDI, 2024).

A *UML* compreende diversos tipos de diagramas, sendo organizados em três categorias principais: estruturais (como o diagrama de classes), comportamentais (como o diagrama de casos de uso e de atividades) e de interação (como o diagrama de sequência). Cada diagrama atende a uma necessidade específica dentro do processo de desenvolvimento, auxiliando na documentação, validação e manutenção do sistema.

Na Figura 7, apresenta-se um exemplo prático de arquitetura baseada em *UML* para o sistema de delivery proposto. O diagrama mostra a separação entre os componentes do *Frontend*, que representa o aplicativo web ou mobile utilizado pelos usuários, e o *Backend*, responsável pelo processamento de requisições, regras de negócio, autenticação e persistência de dados. A comunicação entre as camadas se dá via uma *API REST*, conforme indicado pela interface.

Esse tipo de representação é essencial para garantir a organização modular do projeto, favorecendo a escalabilidade, a reutilização de componentes e a compreensão global da arquitetura por todas as partes envolvidas no desenvolvimento.

Figura 7 – Exemplo de diagrama UML da arquitetura geral do sistema, com destaque para a separação entre frontend, backend e acesso ao banco de dados.



### 2.2.5 Fase de Prototipação

Antes da implementação completa, a prototipação permite validar rapidamente a usabilidade e a lógica das interfaces do sistema. A fase de prototipação é uma etapa fundamental no processo de desenvolvimento de sistemas. Essa prática consiste em criar versões preliminares (protótipos) das telas ou fluxos para análise e testes com usuários reais ou simulados. É especialmente valiosa em projetos com forte componente visual, como aplicativos móveis, onde a experiência do usuário é decisiva (ABRAHAMSSON et al., 2017).

Prototipar ajuda a antecipar ajustes de navegação, disposição de elementos e fluxos funcionais. Além disso, reduz retrabalho e facilita a iteração com base em *feedbacks*.

Nessa fase, as ideias começam a ganhar forma por meio da criação de representações visuais do sistema, chamadas de **protótipos**. O objetivo é simular, de maneira rápida e visual, como será a interface do sistema, suas funcionalidades principais e o fluxo de interação com o usuário.

Protótipos podem ser simples esboços feitos à mão (também chamados de *wireframes*) ou versões mais elaboradas desenvolvidas em ferramentas digitais. O mais importante é que eles permitam visualizar e testar conceitos antes que o sistema seja realmente implementado. Isso reduz custos e ajuda a evitar retrabalho, pois facilita o entendimento

entre desenvolvedores, usuários e demais envolvidos no projeto.

A Figura 8 mostra um exemplo de prototipação de baixa fidelidade. São apresentados três esboços de telas de um aplicativo, desenhados de forma simples em um quadro branco. Cada tela contém elementos básicos, como áreas reservadas para imagens, listas, botões e textos. As anotações com *post-its* ajudam a organizar ideias, destacar funcionalidades ou indicar melhorias sugeridas.

Essa abordagem permite testar hipóteses, coletar *feedback* de usuários e ajustar o projeto com agilidade, mesmo antes de qualquer linha de código ser escrita.



Figura 8 – Exemplo ilustrativo da fase de prototipação com esboços de telas

### 2.2.6 UI/UX

*UI (User Interface)* e *UX (User Experience)* são disciplinas complementares, mas distintas. A *UI* foca no layout visual, na estética e na organização dos elementos na tela. Já a *UX* considera aspectos subjetivos da experiência do usuário ao interagir com o sistema, como fluidez, confiança, clareza e satisfação (PRATAMA; CAHYADI, 2020).

Integrar práticas de *UI/UX* desde as fases iniciais do projeto garante maior aceitação do produto final, além de reduzir a curva de aprendizado e melhorar os indicadores de retenção. Segundo Kashfi et al. (2016), incluir *UX* no ciclo de vida do software resulta em produtos mais eficazes, com menor custo de manutenção (KASHFI; NILSSON; FELDT, 2016).

Um exemplo prático de aplicação de princípios de *UI/UX* pode ser observado na

Figura 9, que apresenta um protótipo de tela inicial de aplicativo. A imagem demonstra o uso de cores contrastantes, hierarquia visual clara e elementos interativos posicionados de forma acessível, contribuindo para uma navegação intuitiva e uma primeira impressão positiva do sistema.

Figura 9 – Protótipo de interface de usuário (UI) com elementos visuais e disposição funcional para avaliação de experiência (UX).



## 2.2.7 Metodologias de Desenvolvimento de *Software*

O desenvolvimento de software é uma atividade complexa que requer planejamento estruturado e execução disciplinada. Para lidar com os desafios dessa prática, foram criadas diferentes metodologias de desenvolvimento, cada uma com características específicas para lidar com os requisitos de escopo, prazo, custo, qualidade e flexibilidade. Esta seção apresenta uma visão geral das metodologias mais utilizadas na indústria e na academia, destacando suas vantagens, desvantagens e contextos de aplicação.

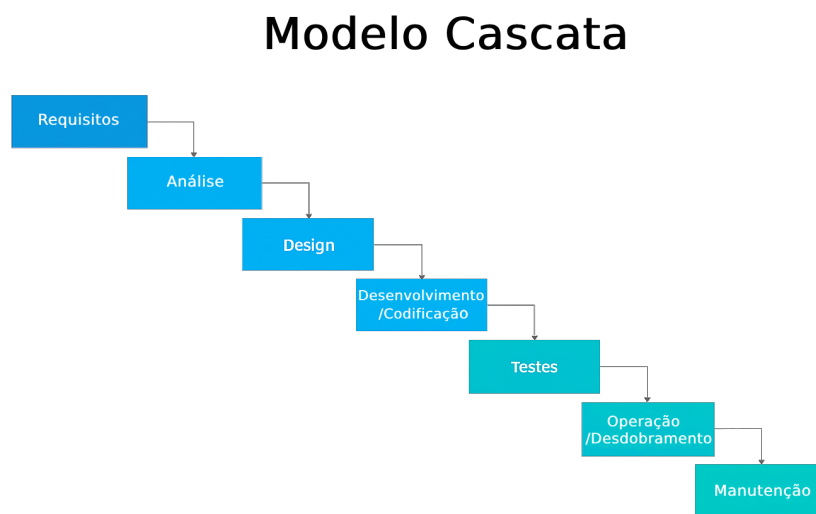
### 2.2.7.1 Modelo Cascata (*Waterfall*)

O modelo cascata é uma das metodologias mais tradicionais e foi formalizado por *Winston Royce* em 1970. Ele segue uma sequência linear de fases: levantamento de requisitos, análise, design, implementação, testes, implantação e manutenção (PRESSMAN; MAXIM, 2016). Cada etapa deve ser concluída antes do início da seguinte, o que torna o processo rígido, porém mais controlado.

Este modelo é indicado para projetos bem definidos, com requisitos estáveis e pouco sujeitos a mudanças. Contudo, a ausência de flexibilidade e a detecção tardia de falhas representam suas maiores limitações (SOMMERVILLE, 2011).

A Figura 10 ilustra visualmente a estrutura do modelo cascata, destacando a linearidade e a dependência entre as etapas do processo de desenvolvimento de software.

Figura 10 – Representação esquemática do modelo cascata, com suas fases sequenciais

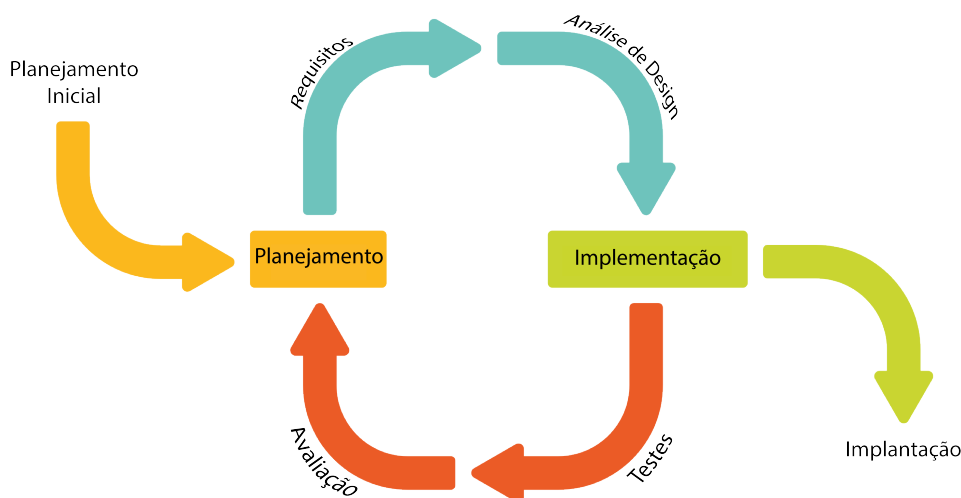


#### 2.2.7.2 Desenvolvimento Iterativo e Incremental

O modelo iterativo-incremental propõe a entrega do *software* em partes funcionais e progressivamente mais completas. A cada iteração, uma versão funcional e evoluída do sistema é disponibilizada, possibilitando validações contínuas com os usuários finais e promovendo ajustes rápidos baseados em feedbacks (BOEHM, 1988).

Esse modelo se destaca por sua capacidade de reduzir riscos e aumentar a flexibilidade durante o desenvolvimento, sendo a base conceitual para diversas metodologias ágeis. A Figura 11 ilustra o ciclo iterativo típico, destacando as fases de planejamento, implementação, testes e avaliação recorrente, que se repetem até que o produto final atenda plenamente aos requisitos do cliente.

Figura 11 – Ciclo do modelo iterativo e incremental, evidenciando a entrega contínua e progressiva do software.



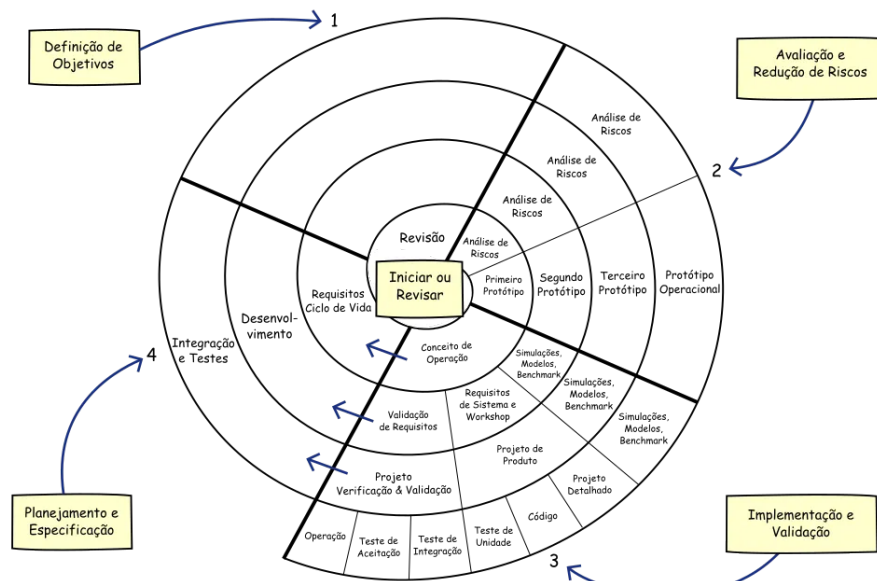
### 2.2.7.3 Modelo Espiral

Proposto por *Barry Boehm*, o modelo espiral combina elementos do modelo cascata com análise de risco em cada ciclo. Seu principal diferencial é a abordagem iterativa em espiral, na qual cada volta representa uma fase do desenvolvimento com foco progressivo na definição, implementação, verificação e validação do sistema (BOEHM, 1988).

Essa estrutura permite revisões frequentes, favorecendo o refinamento de requisitos e o controle de riscos técnicos e gerenciais. Por isso, o modelo espiral é altamente indicado para projetos de grande porte, complexidade elevada e com alto grau de incerteza. Ele também se destaca por incorporar feedbacks contínuos ao processo, contribuindo para maior flexibilidade e mitigação de erros ao longo do ciclo de vida do software.

A Figura 12 ilustra o funcionamento desse modelo, evidenciando como o desenvolvimento avança em ciclos concêntricos, integrando atividades como definição de objetivos, avaliação de riscos, planejamento e validação.

Figura 12 – Representação do modelo espiral, destacando seus ciclos iterativos com foco em riscos

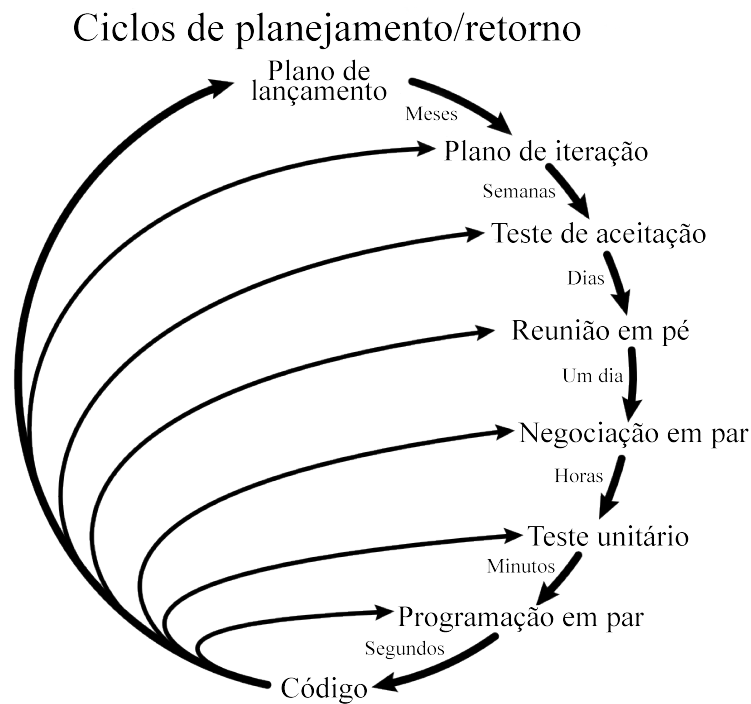


#### 2.2.7.4 Metodologias Ágeis

As metodologias ágeis surgiram como resposta às limitações dos modelos tradicionais. Elas priorizam interações humanas, entrega rápida de software funcional e adaptação contínua às mudanças. O Manifesto Ágil, publicado em 2001, foi um marco na transformação da indústria de software (BECK et al., 2001).

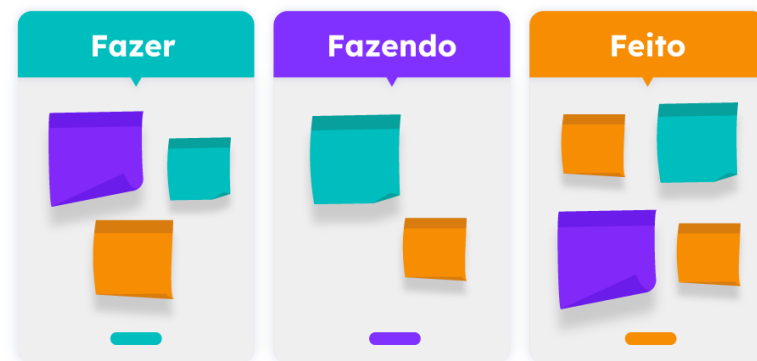
- *Extreme Programming* (XP): enfatiza boas práticas técnicas como programação em par, desenvolvimento orientado a testes (TDD), integração contínua e pequenas releases. Foi proposto por Kent Beck e busca elevar a qualidade do software e a capacidade de resposta a mudanças (BECK; ANDRES, 2004). A Figura 13 apresenta uma visão esquemática das práticas e valores centrais do XP.

Figura 13 – Diagrama representando os princípios fundamentais do *Extreme Programming* (XP).



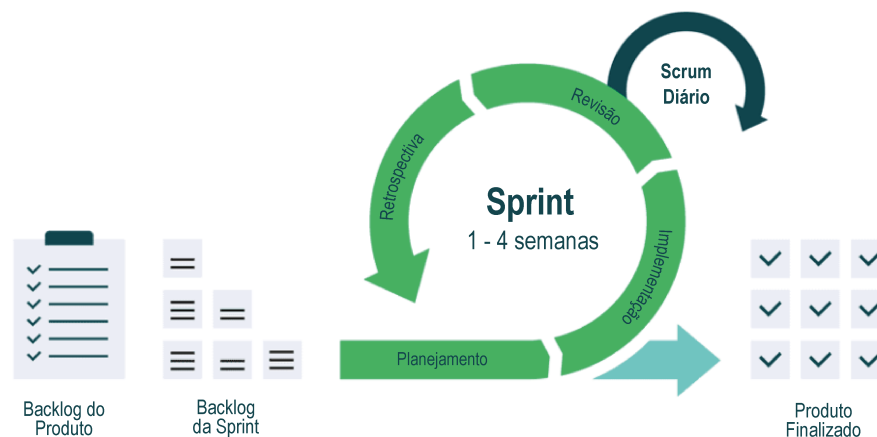
- Kanban: é uma abordagem visual para gerenciamento de fluxo de trabalho. Os quadros Kanban permitem visualizar o progresso das tarefas e identificar gargalos. A Figura 14 apresenta um exemplo de quadro Kanban simplificado, dividido em colunas que representam os estágios de produção, como “Fazer”, “Fazendo” e “Feito”. Essa visualização facilita o acompanhamento contínuo e promove transparência no processo (ANDERSON, 2010).

Figura 14 – Quadro Kanban ilustrando o fluxo de trabalho dividido em etapas visuais: Fazer, Fazendo e Feito.



- Scrum: é uma das metodologias ágeis mais populares. Estrutura o trabalho em *sprints* de curta duração (geralmente de 2 a 4 semanas), com reuniões diárias (*Daily Scrum*), revisões e retrospectivas. Os papéis principais são *Product Owner*, *Scrum Master* e *Development Team* (SCHWABER; SUTHERLAND, 2017). A Figura 15 representa o ciclo de desenvolvimento de uma sprint no contexto do Scrum.

Figura 15 – Fluxo visual de uma sprint no Scrum, incluindo backlog, planejamento, execução e entrega.



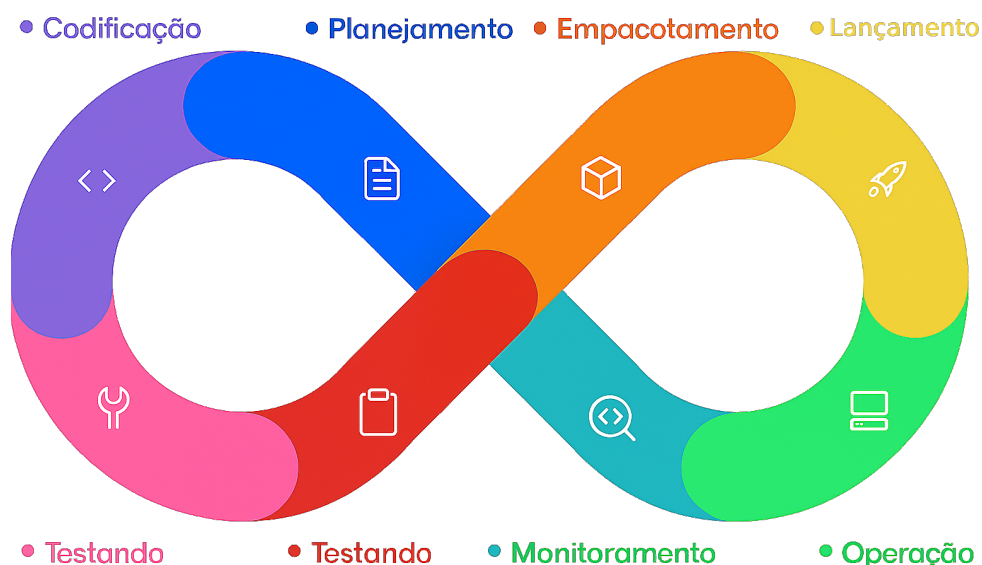
### 2.2.7.5 DevOps

**DevOps** não é uma metodologia tradicional, mas uma filosofia de trabalho que promove a colaboração entre as equipes de desenvolvimento (*Dev*) e operações (*Ops*). Seu objetivo central é acelerar o ciclo de entrega de software por meio da automação, integração contínua (**CI**), entrega contínua (**CD**) e monitoramento constante dos sistemas em produção (KIM; BEHR; SPAFFORD, 2016).

Essa abordagem é especialmente eficaz em ambientes que demandam flexibilidade, alta disponibilidade e atualizações frequentes, características comuns em aplicações web e móveis. Ao incorporar práticas como versionamento automatizado, testes integrados e monitoramento proativo, o **DevOps** permite reduzir o tempo entre a concepção da funcionalidade e sua entrega final ao usuário, mantendo altos níveis de qualidade e confiabilidade.

A Figura 16 representa o ciclo contínuo do **DevOps**, destacando suas fases principais: planejamento, codificação, construção, testes, lançamento, operação e monitoramento — todas interligadas em um fluxo iterativo e contínuo.

Figura 16 – Representação do ciclo contínuo DevOps, integrando desenvolvimento, testes, operação e monitoramento



### 2.2.8 Comparação entre Metodologias

Tabela 1 – Comparação entre Metodologias de Desenvolvimento

Metodologia	Flexibilidade	Risco de Mudança	Indicada para
Cascata	Baixa	Alto	Requisitos Estáveis
Iterativa	Média	Médio	Projetos Médios
Espiral	Alta	Baixo	Projetos Críticos
Scrum	Alta	Baixo	Projetos Ágeis
XP	Alta	Baixo	Projetos com foco técnico
Kanban	Muito Alta	Muito Baixo	Suporte contínuo
DevOps	Alta	Baixo	Integração e Entrega Contínua

Existem diversas metodologias de desenvolvimento de software adotadas ao longo das últimas décadas, cada uma com abordagens, fases e prioridades distintas. A escolha da metodologia correta depende de fatores como escopo, prazo, equipe envolvida e grau de complexidade do projeto. Neste trabalho, optou-se por utilizar a metodologia ágil Scrum, e nesta seção serão apresentadas comparações com outras metodologias tradicionais e ágeis, com justificativas para essa escolha.

- O modelo em cascata (ou *Waterfall*) é uma das metodologias mais antigas e estruturadas. Nele, o processo é dividido em fases sequenciais e rígidas: levantamento de requisitos, análise, design, implementação, testes e manutenção. Cada fase só se inicia após a conclusão da anterior (PRESSMAN; MAXIM, 2016). Embora esse modelo traga uma forte documentação e controle, sua rigidez não se adapta bem a

projetos com requisitos voláteis ou com necessidade de validação contínua, como é o caso de sistemas voltados à experiência do usuário, como aplicativos de delivery. Por essa razão, o modelo cascata foi descartado, pois não atende à natureza iterativa e dinâmica deste projeto, que envolve diferentes perfis de usuário e necessidade constante de adaptação da interface e funcionalidades com base em feedbacks reais.

- O RUP é um processo iterativo de desenvolvimento que incorpora boas práticas de engenharia de software, como uso de casos de uso, modelagem UML e gerenciamento de requisitos. Ele é dividido em quatro fases: iniciação, elaboração, construção e transição (KRUCHTEN, 2004). Embora o RUP permita iterações e adaptação progressiva, ele exige uma carga documental significativa e uma estrutura organizacional mais robusta, o que pode ser inviável em projetos acadêmicos ou desenvolvidos por equipes reduzidas. A escolha pelo Scrum também se justifica frente ao RUP por sua simplicidade operacional e por exigir menos overhead na gestão do projeto, mantendo foco nas entregas incrementais e nas interações diretas entre os envolvidos.
- O XP é uma metodologia ágil voltada principalmente ao desenvolvimento técnico, enfatizando práticas como programação em par, testes automatizados e integração contínua. Apesar de eficiente em ambientes altamente técnicos e colaborativos, o XP é mais difícil de aplicar em projetos acadêmicos ou conduzidos por um único desenvolvedor, pois várias de suas práticas exigem trabalho em dupla e forte interação presencial (BECK; ANDRES, 2004). Portanto, o XP foi considerado inadequado ao contexto deste trabalho, que envolveu desenvolvimento individual com apoio de formulários e validações assíncronas.

## Escolha pelo Scrum

O Scrum é uma metodologia ágil que estrutura o desenvolvimento em ciclos curtos e repetitivos chamados sprints. Cada sprint entrega um incremento funcional do sistema, permitindo inspeção contínua, adaptação e integração de feedbacks. O Scrum prioriza a colaboração com stakeholders, comunicação constante e flexibilidade para mudanças ao longo do desenvolvimento (SCHWABER; SUTHERLAND, 2017).

A escolha pelo Scrum se deu por várias razões:

- **Adaptação rápida às mudanças:** os requisitos do sistema puderam ser revistos e reordenados de acordo com os testes com usuários;
- **Desenvolvimento incremental:** funcionalidades foram implementadas e validadas em ciclos curtos;
- **Redução de retrabalho:** a prototipação e as entregas parciais evitaram a construção de funcionalidades inadequadas;

- **Baixo custo de gestão:** o Scrum é leve em termos de documentação obrigatória e processos formais, o que se alinha ao escopo acadêmico e às limitações de tempo e equipe.

Dessa forma, o Scrum mostrou-se a metodologia mais adequada para o desenvolvimento deste sistema de delivery, possibilitando entregas contínuas, testes incrementais com stakeholders reais e constante melhoria da solução.

O Scrum é uma estrutura de desenvolvimento ágil que organiza o trabalho em ciclos curtos chamados *sprints*, normalmente com duração de duas a quatro semanas. Em cada *sprint*, entrega-se um incremento funcional do produto, o que permite a validação contínua e o ajuste de funcionalidades de acordo com o *feedback* recebido (SCHWABER; SUTHERLAND, 2017).

Dentre os fatores que motivaram a escolha do Scrum para este projeto, destacam-se:

- A entrega contínua de valor, essencial para validar decisões técnicas e funcionais com stakeholders (neste caso, usuários simulados como entregadores, empreendedores e consumidores entrevistados).
- A priorização iterativa, que permitiu focar primeiro nas funcionalidades essenciais, como o cadastro de pedidos e sistema de rastreamento.
- A capacidade de adaptação, fundamental diante de possíveis mudanças no escopo durante o ciclo de desenvolvimento acadêmico.

Para o desenvolvimento da solução proposta neste trabalho, um aplicativo de *delivery* para economia solidária, foi adotada a metodologia Scrum, uma das abordagens ágeis mais amplamente utilizadas na engenharia de *software* moderna. A escolha por essa metodologia justifica-se pelo seu foco em entregas rápidas, iteração constante com os usuários e adaptabilidade a mudanças, o que se alinha perfeitamente ao contexto de desenvolvimento acadêmico e às limitações de tempo e recursos disponíveis.

## 2.3 Trabalhos Relacionados

Nesta seção, são analisadas as principais plataformas de *delivery* que atuam no mercado brasileiro, com foco em suas características operacionais, políticas de cobrança, usabilidade e impacto sobre consumidores, entregadores e empreendedores. O objetivo é compreender como essas plataformas se estruturam e quais são os seus diferenciais, limitações e implicações econômicas e sociais.

Diversos trabalhos acadêmicos propuseram soluções tecnológicas voltadas para sistemas de delivery, especialmente com foco em pequenos negócios, plataformas cooperativas

ou eficiência logística. Essas propostas servem como base comparativa para a estruturação do sistema apresentado neste trabalho.

Dois trabalhos acadêmicos foram escolhidos como referências na construção de sistemas de entrega locais, com abordagens que priorizam acessibilidade, baixo custo e foco em pequenos empreendedores.

O trabalho de Farias e Barros (2014), desenvolvido no âmbito da UTFPR, apresenta um sistema de delivery para restaurantes de pequeno porte, com o objetivo de reduzir custos operacionais e eliminar intermediários. A proposta baseia-se em uma arquitetura web simples e acessível, com controle de pedidos, produtos e entregadores (FARIAS; BARROS, 2014). Embora a tecnologia utilizada esteja mais desatualizada, os princípios de estrutura enxuta e independência de plataformas privadas permanecem relevantes.

Já Silva (2023), em sua monografia na Universidade Federal de Ouro Preto, propôs um aplicativo de entregas voltado à economia local, integrando comerciantes e entregadores por meio de uma interface simplificada e intuitiva. O projeto enfatiza a descentralização, permitindo que diferentes perfis de usuário gerenciem pedidos, entregas e produtos em tempo real (SILVA, 2023). Esse trabalho contribuiu diretamente para a estrutura modular e o controle dinâmico do cardápio proposto no presente projeto.

Ambos os trabalhos serviram como base conceitual e comparativa, reforçando a viabilidade de sistemas de delivery cooperativos, replicáveis e orientados a comunidades autônomas.

A seguir, será apresentada uma análise individual das plataformas *AiqFome* (2.3.1), *iFood* (2.3.2) e *UaiRango* (2.3.3), observando aspectos como taxas cobradas, funcionalidades oferecidas e experiência do usuário em cada uma delas.

### 2.3.1 AiqFome

O *AiqFome* é um aplicativo brasileiro de delivery de alimentos que conecta usuários a uma variedade de restaurantes locais. Fundado em 2007 em Maringá, Paraná, o *AiqFome* tem se expandido para várias cidades do Brasil, oferecendo uma plataforma conveniente para pedir refeições de restaurantes próximos. O aplicativo se destaca por sua interface amigável e pela variedade de opções de restaurantes e tipos de cozinha, permitindo que os usuários façam pedidos para entrega ou retirada. (AIQFOME-SOBRE, 2025)

O *AiqFome* é uma solução prática para quem deseja pedir comida de forma rápida e conveniente, com uma ampla seleção de restaurantes e uma interface fácil de usar.

### 2.3.2 iFood

O *iFood* é uma plataforma de entrega de alimentos popular no Brasil, fundada em 2011. Ele conecta consumidores a uma variedade de restaurantes, permitindo que os usuários façam pedidos de refeições por meio de um aplicativo móvel ou site. A empresa é uma das líderes no mercado de *delivery* de comida na América Latina e oferece uma ampla gama de opções culinárias, desde fast food até pratos *gourmet*. (IFOOD-SOBRE, 2025)

Essas informações podem variar ao longo do tempo, portanto, é sempre recomendável verificar diretamente nas plataformas oficiais do *iFood* para obter os dados mais recentes.

### 2.3.3 UaiRango

O *UaiRango* é um aplicativo de entrega de alimentos que conecta consumidores a restaurantes e outros estabelecimentos de alimentação. O serviço é bastante popular em várias cidades do Brasil, especialmente em Minas Gerais. A plataforma permite que os usuários façam pedidos de comida para entrega em domicílio ou retirada diretamente no restaurante. (UAIRANGO-SOBRE, 2025)

O *UaiRango* se destaca pela conveniência e variedade de opções que oferece aos consumidores, tornando a experiência de pedir comida mais prática e acessível. As taxas cobradas são geralmente competitivas com outras plataformas de *delivery*, mas é sempre bom verificar os valores exatos antes de confirmar um pedido.

### 2.3.4 Funcionalidades dos Aplicativos

Tabela 2 – Comparação de funcionalidades entre plataformas de *delivery*

Funcionalidade	AiqFome	iFood	UaiRango
Interface intuitiva	Sim	Sim	Sim
Variedade de restaurantes	Alta	Muito alta	Alta
Rastreamento de pedidos	Sim	Sim	Sim
Promoções e cupons	Sim	Sim	Sim
Avaliações e comentários	Não informado	Sim	Sim
Assinatura premium	Não	Sim (iFood Plus)	Não
Opções de pagamento	Cartão, débito e on-line	Diversas, incluindo vales	Diversas

**Fonte:** Elaborado pelo autor com base nos sites oficiais das plataformas (2025).

A tabela de funcionalidades mostra que as três plataformas oferecem um conjunto básico semelhante de recursos, como interface intuitiva, rastreamento de pedidos e promoções. No entanto, o *iFood* se destaca por contar com uma variedade maior de métodos de

pagamento e recursos adicionais como assinatura premium (*iFood Plus*), além de avaliações e comentários, o que contribui para uma experiência mais completa. Já o *AiqFome* e o *UaiRango* focam mais na simplicidade e acessibilidade das funcionalidades básicas.

### 2.3.5 Taxas Cobradas

Tabela 3 – Comparação de taxas entre diferentes plataformas de delivery

Plataforma	Assinatura	P. Pedidos <sup>1</sup>	Por Pedido	Entrega <sup>2</sup>	Serviço <sup>3</sup>
AiqFome	R\$89,90	0,87	12%	+10%	2,99%
iFood	R\$130,00	1,00	12%	+11%	3,20%
UaiRango	Não possui	Não possui	8% <sup>4</sup>	Não possui	Não possui

**Notas:**

1. Taxa extra para pedidos com valor abaixo de 20 reais.
2. Apenas para estabelecimentos com entrega pela plataforma.
3. Percentual adicional aplicado aos pedidos pagos na plataforma.
4. Caso seja exclusivo da plataforma, senão 11%.

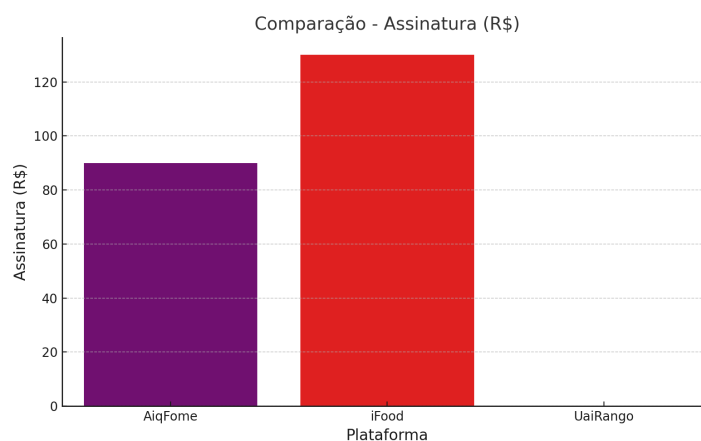
**Fonte:** Elaborado pelo autor com base nos sites oficiais das plataformas (2025).

A comparação das taxas revela que o *UaiRango* é a plataforma mais acessível financeiramente, pois não cobra assinatura, nem taxas para pequenos pedidos ou serviço, e mantém a menor taxa por pedido (8% para estabelecimentos exclusivos). Em contrapartida, o *iFood* é a plataforma com os maiores custos operacionais, incluindo a assinatura mensal mais alta e a maior taxa de serviço. O *AiqFome* apresenta valores intermediários, sendo uma opção mais viável que o *iFood*, mas ainda com mais encargos que o *UaiRango*.

Para melhor visualização e análise comparativa, foram gerados gráficos representando os principais custos operacionais das três plataformas de delivery mais relevantes no cenário nacional: *AiqFome*, *iFood* e *UaiRango*. Esses gráficos foram desenvolvidos a partir das informações coletadas sobre valores de assinatura, taxas para pequenos pedidos, comissões por pedido, taxas de entrega e taxas de serviço.

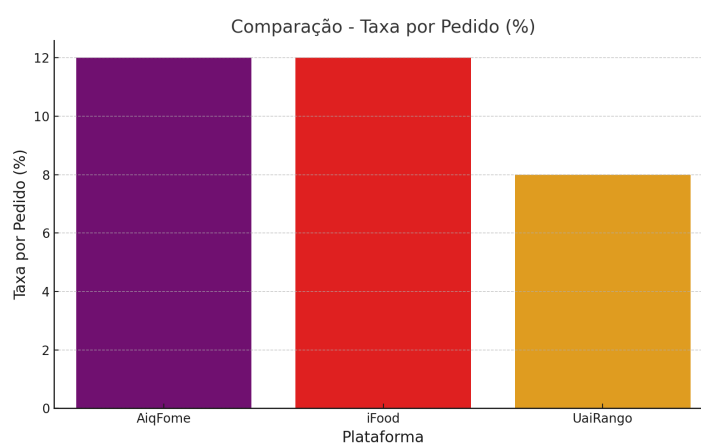
A seguir, são apresentados os gráficos comparativos que auxiliam na compreensão do impacto financeiro que cada plataforma pode gerar para estabelecimentos parceiros e usuários finais.

Figura 17 – Comparação entre valores de assinatura mensal das plataformas de delivery.



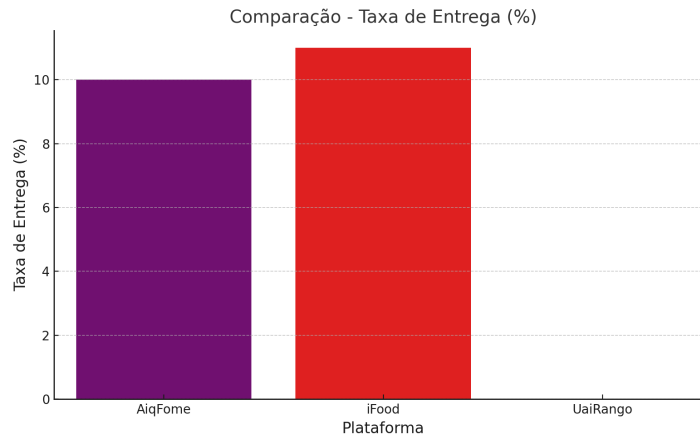
Fonte: Elaborado pelo autor (2025).

Figura 18 – Percentual de comissão por pedido retido pelas plataformas.



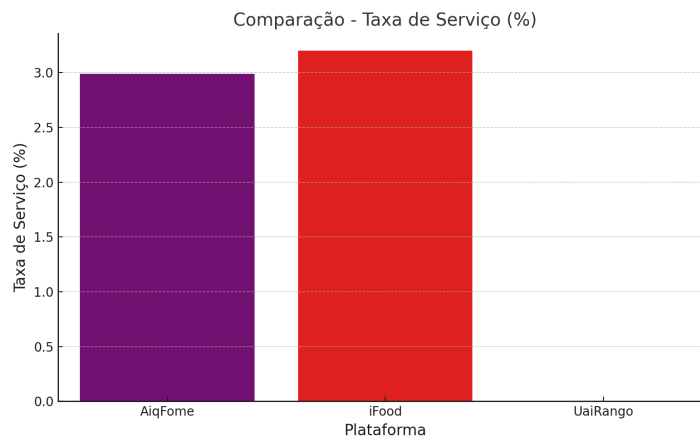
Fonte: Elaborado pelo autor (2025).

Figura 19 – Comparação das taxas de entrega aplicadas pelas plataformas.



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

Figura 20 – Taxa de serviço adicionada aos pedidos em plataformas de delivery.



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

### 2.3.6 Onde Encontrar as Informações

Tabela 4 – Onde encontrar as informações sobre as taxas e funcionalidades

Plataforma	Ajuda	Termos
AiqFome	( <a href="#">AIQFOME-AJUDA, 2025</a> )	( <a href="#">AIQFOME-TERMOS, 2025</a> )
iFood	( <a href="#">IFOOD-AJUDA, 2025</a> )	( <a href="#">IFOOD-TERMOS, 2025</a> )
UaiRango	( <a href="#">UAIRANGO-AJUDA, 2025</a> )	( <a href="#">UAIRANGO-TERMOS, 2025</a> )

**Fonte:** Elaborado pelo autor com base nos sites oficiais das plataformas (2025).

A tabela de fontes mostra que todas as plataformas disponibilizam informações sobre taxas e funcionalidades em seus sites oficiais, termos de uso e centrais de ajuda. O acesso a essas informações é essencial para que empreendedores tomem decisões conscientes, considerando os custos e benefícios de cada serviço. A padronização e transparência desses canais contribuem para maior confiança por parte dos usuários e parceiros comerciais.

## 3 Desenvolvimento

A modelagem de desenvolvimento de software é um processo estruturado que busca criar soluções eficazes e eficientes para atender a necessidades específicas. Para isso, diversas metodologias podem ser adotadas, como as abordagens ágil, cascata e *DevOps*. Cada uma dessas metodologias possui suas particularidades, permitindo que o processo de desenvolvimento seja adaptado conforme o projeto e os objetivos desejados.

A Engenharia de Software é a área da computação voltada para o desenvolvimento sistemático, disciplinado e quantificável de software, bem como a manutenção e evolução desses sistemas. Este projeto faz uso das boas práticas dessa área para garantir a entrega de um sistema de qualidade, seguro, modular e de fácil manutenção (SOMMERVILLE, 2011). O processo adotado contempla as principais etapas da Engenharia de Software: levantamento de requisitos, análise e especificação, projeto (design), implementação, testes e manutenção.

Para o desenvolvimento deste aplicativo, optou-se pela adoção de práticas baseadas em metodologias ágeis, priorização de requisitos com feedback de stakeholders e desenvolvimento iterativo e incremental. Diagramas de casos de uso, classes, sequência e ER foram utilizados para ilustrar e documentar as decisões arquiteturais e de modelagem. As principais etapas incluem:

### 3.1 Análise de Requisitos

Com base nas entrevistas e formulários aplicados aos perfis de cliente, entregador e empreendedor, foram identificadas necessidades específicas que nortearam a construção do sistema.

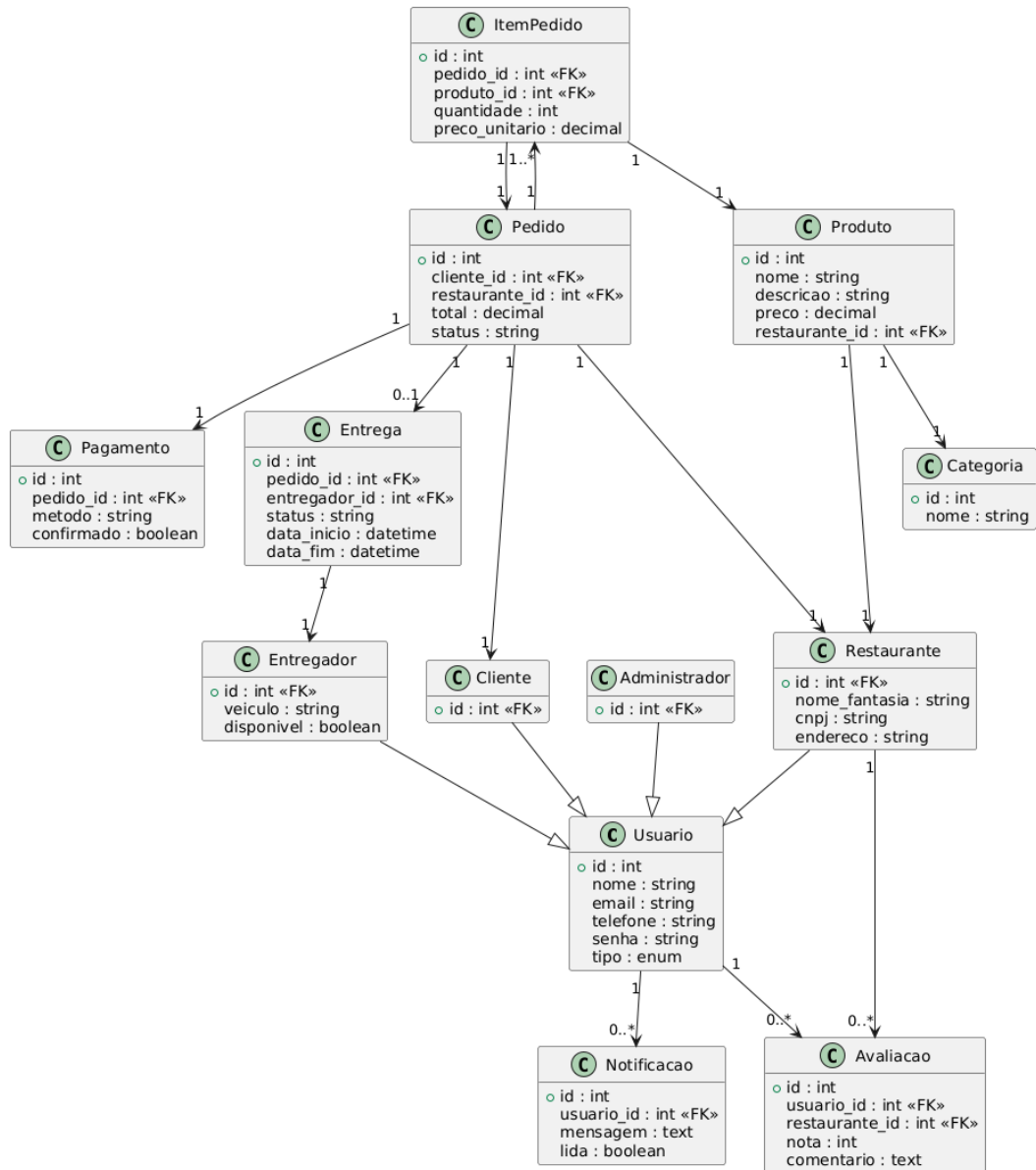
O diagrama a seguir, Figura 21, representa o modelo de domínio do sistema proposto, descrevendo as principais entidades, atributos e os relacionamentos entre elas. A estrutura é centrada na entidade **Usuario**, que está associada a diferentes perfis de atuação — **Cliente**, **Restaurante**, **Entregador** e **Administrador** —, permitindo a existência de múltiplos papéis vinculados a um mesmo usuário, conforme o tipo especificado.

As entidades **Pedido**, **ItemPedido**, **Produto**, **Pagamento**, **Entrega**, **Avaliação** e **Notificação** representam os processos centrais do sistema. O modelo também evidencia as cardinalidades entre as entidades, garantindo clareza nas relações como, por exemplo, o vínculo de um pedido com múltiplos itens ou a associação exclusiva entre um pedido e seu respectivo pagamento.

A modelagem foi elaborada com foco na separação de responsabilidades, integridade

dos dados e flexibilidade para expansão futura, servindo como base para a implementação de um sistema de delivery voltado à economia solidária.

Figura 21 – Diagrama de Modelo de Domínio do sistema proposto



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

**Backend:** Escolha de uma tecnologia de servidor adequada, como *Node.js*, *Python* (*Django/Flask*), ou *Java* (*Spring Boot*), com escalabilidade e segurança para suportar um grande volume de acessos.

**Frontend:** Seleção de *frameworks* adequados para interfaces de usuário, como *React Native* ou *Flutter*, para um desenvolvimento multiplataforma (*iOS* e *Android*).

**Banco de Dados:** Optar por um banco de dados que suporte consultas em tempo

real e que seja escalável, como *PostgreSQL* ou *MongoDB*.

**Integração de Pagamentos:** Escolher provedores de pagamento que ofereçam uma *API* segura e fácil de implementar, como *Mercado Pago*, *Pagar.me* ou *PagSeguro*.

**Geolocalização e Roteamento:** Identificar serviços de mapa e geolocalização, como *Google Maps API* ou *Mapbox*, para permitir o rastreamento de entregas.

### 3.1.1 Especificação Técnica

Com base nos requisitos identificados e nas diretrizes conceituais previamente discutidas, esta subseção apresenta a especificação técnica do sistema proposto. O objetivo é detalhar os componentes funcionais e tecnológicos da solução, abrangendo desde a arquitetura de software até requisitos não funcionais e diretrizes legais. Essa especificação serve como base para orientar a implementação e garantir que o sistema atenda aos princípios de segurança, escalabilidade e conformidade normativa.

**Arquitetura do Sistema:** A plataforma foi concebida com uma arquitetura cliente-servidor, em que o *front-end* (aplicativo) se comunica com o back-end por meio de chamadas *RESTful*. Essa estrutura favorece a escalabilidade e permite a modularização dos serviços, facilitando atualizações e manutenção.

**Requisitos Não Funcionais:** O sistema foi projetado para atender requisitos como escalabilidade horizontal, desempenho adequado sob carga moderada, segurança de dados e conformidade com a Lei Geral de Proteção de Dados (*LGPD*). Esses aspectos são fundamentais para garantir a viabilidade a longo prazo e a confiança por parte dos usuários.

**Interfaces de Programação de Aplicações (*APIs*):** Foram especificadas *APIs* responsáveis por operações críticas como autenticação, geolocalização, criação e gestão de pedidos, integração com *gateways* sistemas de pagamento e envio de notificações. O uso de *APIs* também facilita eventuais integrações com sistemas externos ou entre cooperativas regionais.

**Políticas de *LGPD*:** Toda a estrutura do sistema respeita os princípios da *LGPD*, com mapeamento dos pontos de coleta, armazenamento e tratamento de dados pessoais. São implementadas práticas como consentimento explícito, minimização de dados, registro de finalidade e políticas de exclusão, assegurando conformidade legal e ética em relação à privacidade dos usuários.

Para ilustrar a viabilidade técnica da proposta, foram analisados exemplos de plataformas digitais mantidas por organizações comunitárias ou coletivos com infraestrutura reduzida. Essas iniciativas demonstram que é possível manter sistemas funcionais e confiáveis com equipes enxutas e custos operacionais controlados, desde que haja clareza

técnica, planejamento e divisão colaborativa de responsabilidades.

Diferentemente do custo de desenvolvimento, que pode ser viabilizado por meio de parcerias, financiamento coletivo ou editais públicos, o custo de manutenção representa um compromisso recorrente. Isso inclui despesas como serviços de hospedagem (R\$ 80–200/mês), banco de dados em nuvem (R\$ 0–50/mês), uso de [APIs](#) externas e suporte técnico (prestado por membros da cooperativa ou contratado). Estima-se que um coletivo organizado possa operar a plataforma com um custo mensal inferior a R\$ 500, o que viabiliza um modelo autossustentável e acessível para iniciativas de economia solidária.

### 3.1.2 LGPD

Propósito de Coleta de Dados: Limitar a coleta apenas ao que é necessário para operar o aplicativo.

Consentimento: Garantir que os usuários aceitem as políticas de privacidade e uso de dados, com consentimento explícito para dados sensíveis.

Anonimização e Segurança de Dados: Implementar métodos de anonimização de dados pessoais e políticas de segurança, incluindo criptografia e controle de acesso.

Direitos dos Usuários: Prover aos usuários o direito de acesso, correção e exclusão de seus dados, em conformidade com os direitos garantidos pela [LGPD](#).

Documentação de Incidentes e Relatórios: Manter registros detalhados sobre qualquer incidente relacionado à segurança e acesso aos dados dos usuários.

## 3.2 Levantamento de Requisitos

Entender e documentar o que o sistema precisa fazer para atender de forma eficiente às necessidades dos usuários e do mercado. Esse objetivo será alcançado por meio de entrevistas com *stakeholders*, análise de sistemas semelhantes e criação de documentação detalhada dos requisitos.

### 3.2.1 Engenharia de Requisitos

Ela permite reduzir riscos, alinhar expectativas e garantir que o produto final atenda às necessidades reais do público-alvo. Os requisitos podem ser obtidos por meio de entrevistas, formulários, observações e protótipos interativos. Além disso, é comum classificá-los como funcionais (ações e comportamentos do sistema) e não funcionais (restrições e critérios de qualidade como desempenho, segurança e usabilidade).

Traduzir os requisitos iniciais do *app de delivery* em especificações técnicas e funcionais detalhadas para garantir um desenvolvimento eficaz e conforme as leis vigentes,

como a [LGPD](#).

Definição dos Objetivos do *app*: Entender a finalidade e os objetivos principais do aplicativo, como permitir que usuários possam fazer pedidos de refeições ou produtos para entrega, definir opções de pagamento e monitorar entregas.

Definição de Escopo e Priorização: Estabelecer as principais funcionalidades de lançamento (*MVP*) e planejar melhorias futuras com base no feedback dos usuários.

Análise Legal (*LGPD*): Avaliar os requisitos de privacidade e proteção de dados dos usuários para definir quais dados serão coletados, como serão armazenados e processados e quais políticas de privacidade e segurança serão implementadas.

### 3.2.2 Entrevistas com Stakeholders

Identificar as necessidades e expectativas dos diferentes *stakeholders* (usuários finais, entregadores, donos de restaurantes, gerência, etc.).

Stakeholders:

- Donos de restaurantes e estabelecimentos
- Entregadores
- Clientes finais (usuários do *app*)

Perguntas:

- Quais funcionalidades são necessárias no *app*?
- Quais são as expectativas em termos de usabilidade?
- Como deve ser o fluxo de pedidos?
- Existem sistemas ou processos atuais que precisam ser integrados?
- Quais são os principais desafios para os entregadores e restaurantes hoje?

Os questionários foram elaborados para garantir uma experiência fluida e objetiva aos participantes, evitando respostas longas e focando nas informações mais relevantes. A divisão em seções permite uma análise detalhada de cada aspecto do uso dos aplicativos de *delivery*.

Este documento serve como referência para futuras revisões ou adaptações dos questionários, garantindo que sua estrutura original seja preservada e possa ser aprimorada conforme necessário.

Foram aplicados três formulários distintos, direcionados a consumidores, empreendedores e entregadores, com o objetivo de levantar informações sobre necessidades, expectativas, dificuldades e preferências no uso de plataformas de *delivery*. A amostra foi composta por:

- 41 consumidores;
- 30 empreendedores (donos ou responsáveis por estabelecimentos que utilizam *delivery*);
- 40 entregadores.

Essa diversidade de participantes permitiu uma visão abrangente do ecossistema local de entregas, contribuindo para uma modelagem mais alinhada com os diferentes perfis de usuários envolvidos no sistema proposto. As respostas completas dos formulários encontram-se nos Apêndices [A](#), [B](#) e [C](#).

### 3.2.3 Metodologia de Elaboração dos Formulários

Para levantar os requisitos do sistema e compreender as dores e expectativas dos usuários, foram aplicados formulários online com abordagem exploratória e empírica. Esse tipo de pesquisa é indicado quando há pouco conhecimento consolidado sobre o fenômeno estudado, permitindo a descoberta de padrões e a formulação de hipóteses para projetos aplicados (GIL, 2008).

As perguntas foram elaboradas com o objetivo de coletar informações qualitativas e quantitativas sobre a experiência dos usuários com plataformas de entrega. Três públicos-alvo foram contemplados:

- **Consumidores (Clientes):** questões sobre frequência de uso, satisfação, principais problemas (como atrasos ou taxas altas) e sugestões de melhorias;
- **Entregadores:** perguntas relacionadas à jornada de trabalho, dificuldades (clima, segurança, rotas), remuneração e uso de aplicativos;
- **Empreendedores (Lojistas):** foco na visibilidade dos produtos, taxas cobradas, logística e percepção sobre suporte das plataformas existentes.

A estrutura das perguntas seguiu os seguintes princípios:

- **Clareza e objetividade:** perguntas diretas, com linguagem acessível;
- **Formato misto:** combinação de questões fechadas (múltipla escolha e escalas) e abertas (respostas descritivas);

- **Segmentação:** cada formulário foi específico para o perfil de usuário, respeitando as particularidades de cada grupo;
- **Instrumento de coleta:** Google Forms, permitindo alcance remoto e respostas anônimas.

Essa estrutura permitiu captar dados estatísticos simples e, ao mesmo tempo, insights mais aprofundados por meio de respostas discursivas. Embora não tenha sido adotado um modelo metodológico formal (como entrevistas estruturadas ou aplicação de escalas validadas), a lógica de construção e aplicação dos questionários está alinhada às boas práticas de pesquisas de campo exploratórias, como descrito por Gil (2008) em estudos de natureza aplicada.

### 3.2.3.1 Formulário Consumidores

Este documento apresenta a estrutura e os objetivos do questionário intitulado “*Questionário para Consumidores*“, elaborado com a finalidade de coletar informações diretamente dos usuários de aplicativos de *delivery*. A pesquisa foi aplicada com fins acadêmicos no contexto do Trabalho de Conclusão de Curso, visando compreender preferências, experiências, desafios e sugestões dos consumidores em relação ao uso dessas plataformas.

O formulário foi dividido em sete seções temáticas, organizadas para facilitar o preenchimento e garantir a coleta de dados relevantes e objetivos.

#### Estrutura do Questionário

**Seção 1: Introdução** Apresentação do questionário e seus objetivos. Garantia de anonimato e confidencialidade das respostas.

#### **Seção 2: Sobre a Experiência Atual**

**Pergunta:** "Quais são as maiores dificuldades que você enfrenta ao usar aplicativos de delivery?"

**Pergunta:** "Com qual frequência é seu uso de aplicativos de delivery?" (Escala de 1 a 10 - Pouco a Todo dia)

**Pergunta:** "Qual sua faixa etária?" (Opções: Menor de 18, 18-26, 27-37, 38-49, 50+)

**Pergunta:** "O que você mais valoriza em um aplicativo de delivery?" (Seleção múltipla: Cupons de Desconto, Preço, Promoções, Tempo de Entrega, Variedade de Lojas, Outros)

#### **Seção 3: Sobre Funcionalidades**

**Pergunta:** "Você gostaria de funcionalidades extras, como agendar pedidos ou deixar uma avaliação detalhada para restaurantes?" (Sim, Não, Tanto Faz, Outro)

**Pergunta:** "Que tipos de filtros ou categorias ajudariam você a encontrar opções mais rapidamente?"(Múltiplas opções: Mostrar apenas Farmácias, Mostrar apenas Lanchonetes, Mostrar apenas Mercados, Mostrar apenas Restaurantes, Mostrar lojas com entrega grátis, Mostrar apenas Pet Shop, Todas as opções acima)

#### **Seção 4: Sobre Preços e Promoções**

**Pergunta:** "Promoções ou cupons influenciam sua decisão de compra?"(Sim, Não, Talvez)

#### **Seção 5: Sobre Sustentabilidade e Ética**

**Pergunta:** "Você considera importante saber sobre a origem dos produtos ou práticas sustentáveis dos restaurantes?"(Sim, Não, Tanto Faz)

**Pergunta:** "Estaria disposto a pagar uma taxa adicional para entregas mais ecológicas?"(Sim, Não, Tanto Faz)

#### **Seção 6: Sobre Suporte ao Cliente**

**Pergunta:** "Qual foi sua experiência com suporte ao cliente em outros aplicativos? 1 - Muito ruim e 5 - Muito boa"(Escala de 1 a 5 - Muito ruim a Muito boa)

**Pergunta:** "De acordo com a pergunta anterior, o que poderia ser melhor?"

**Pergunta:** "Como você prefere ser notificado sobre o status dos pedidos?"(E-mail, Notificação no aplicativo, [SMS](#))

#### **Seção 7: Extra**

Pergunta opcional onde o participante pode sugerir pontos não abordados e fornecer contato para futuras pesquisas.

### **3.2.3.2 Formulário Entregador**

Este documento apresenta a estrutura e os objetivos do questionário intitulado “*Questionário para Entregadores*”, elaborado com a finalidade de coletar informações diretamente dos trabalhadores responsáveis pela entrega nos aplicativos de *delivery*. A pesquisa foi aplicada com fins acadêmicos no contexto do Trabalho de Conclusão de Curso, visando compreender as preferências, experiências, desafios e sugestões dos entregadores no uso dessas plataformas.

O formulário foi dividido em seis seções temáticas, organizadas para facilitar o preenchimento e garantir a coleta de dados relevantes e objetivos.

#### **Seção 1: Perfil do Participante**

**Pergunta:** "Qual sua faixa etária? (Opções: 18-26, 27-36, 37-46, 47-56, 56+)

**Pergunta:** "Há quanto tempo trabalha como entregador? (Menos de 1 ano, 1–2

anos, 3 anos ou mais)

**Pergunta:** "Qual veículo você utiliza para fazer entregas? (Bicicleta, Moto, Carro, Outros)

**Pergunta:** "Por qual motivo você decidiu trabalhar como entregador? (Renda extra, Facilidade, Negócio próprio, Autonomia, Desemprego, Outros)

**Pergunta:** "Este trabalho é sua principal fonte de renda? (Sim, Não)

## Seção 2: Operações e Logística

**Pergunta:** "Quais são os desafios enfrentados por ser um entregador? (Baixa remuneração, Falta de direitos trabalhistas, Dependência dos algoritmos, Concorrência, Outros)

**Pergunta:** "Quais são os desafios enfrentados durante a entrega? (Risco de violência, Clima, Exaustão, Falta de respeito, Manutenção do veículo, Tempo de espera, Bloqueios injustos, Outros)

**Pergunta:** "Você prefere definir sua sequência de entregas ou seguir um roteiro definido pelo aplicativo? (Sequência própria, Ordem do [app](#))

**Pergunta:** "Como você prioriza as entregas? (Valor, Rota, Distância, Não prioriza, Outros)

**Pergunta:** "Como você gerencia o tempo de entregas? (Menor distância, Maior rota com várias entregas, Agilidade, Não gerencia, Outros)

**Pergunta:** "Quais funcionalidades ajudariam a gerenciar melhor o tempo e a prioridade das entregas? (Otimização de rotas, Personalização, Tempo de espera remunerado, Integração entre apps, Paradas inteligentes, Outros)

## Seção 3: Remuneração e Incentivos

**Pergunta:** "Qual seria a remuneração mínima justa por entrega? (R\$6,50–R\$10,00; R\$10,00–R\$15,00; Acima de R\$15,00; Outros)

**Pergunta:** "Qual seria a remuneração mínima justa por quilômetro rodado? (R\$1,50–R\$2,50; R\$2,50–R\$5,00; Acima de R\$5,00; Outros)

**Pergunta:** "Quais bônus ou incentivos seriam mais atrativos? (Reajuste por entrega/quilômetro, Pagamento integral por pedido, Bonificação por distância ou tempo de espera, Limitação de rotas de bicicleta, Outros)

**Pergunta:** "Quais métodos de pagamento você prefere receber? (Dinheiro físico, [PIX](#), Transferência bancária)

## Seção 4: Ferramentas no Aplicativo

**Pergunta:** "Que tipo de informações são essenciais em tempo real? (Mapas deta-

lhados, Contato com o cliente, Informações sobre o pedido, Nível de demanda, Tempo de espera, Outros)

**Pergunta:** "O que tornaria o processo de entrega mais simples? (Informações sobre endereço, Contato rápido com o cliente, Sugestões de locais estratégicos, Outros)

#### **Seção 5: Segurança e Suporte**

**Pergunta:** "Você já enfrentou situações de insegurança durante uma entrega? (Sim, Não)

**Pergunta:** "Como o aplicativo pode ajudar a prevenir ou lidar com insegurança? (Botão de emergência, Relatório de áreas perigosas, Alertas sobre acidentes, Avaliação de clientes, Outros)

**Pergunta:** "Que tipo de suporte você esperaria dos aplicativos? (Chat 24h, Suporte para problemas, acidentes ou bloqueios, Auxílio na manutenção do veículo, Proteção contra penalizações, Outros)

#### **Seção 6: Extra**

Pergunta opcional onde o participante pode sugerir pontos não abordados e fornecer contato para futuras pesquisas.

### 3.2.3.3 Formulário Empreendedores

Este documento teve como objetivo registrar a estrutura e o conteúdo do questionário intitulado “*Questionário para Empreendedores*“, elaborado para coletar informações sobre a experiência de empresários com aplicativos de delivery. O questionário foi desenvolvido com fins acadêmicos e busca compreender desafios, preferências e percepções sobre o uso dessas plataformas.

O formulário está dividido em seis seções, abordando desde a identificação do negócio até a avaliação dos serviços de *delivery*.

Estrutura do Questionário

#### **Seção 1: Introdução e Identificação do Negócio**

Apresentação do questionário e seus objetivos. Garantia de anonimato e confidencialidade das respostas.

**Pergunta:** "Qual o ramo de atuação do seu negócio?"(Opções: Lanchonete, Farmácia, Mercado, Restaurante, Supermercado, Outros.)

#### **Seção 2: Sobre Operações e Integração**

**Pergunta:** "Possui serviço de entrega por aplicativo de delivery?"(Sim/Não)

**Pergunta:** "Qual aplicativo de delivery você utiliza?"(Opções: iFood, Uber Eats,

Aiqfome, Pedidos Já, Zé Delivery, AnotaAi, Nenhum aplicativo, Outros.)

### Seção 3: Desafios e Gestão

**Pergunta:** "Qual o maior desafio do seu empreendimento ao trabalhar com delivery atualmente?" (Seleção múltipla: comissões elevadas, dependência de plataformas, concorrência intensa, problemas logísticos, falta de personalização, reputação online, entre outros.)

**Pergunta:** "Você utiliza algum sistema de gestão de pedidos? Se sim, qual?" (Aberta para resposta livre.)

**Pergunta:** "Quais são os principais horários de pico para os pedidos?" (Manhã, Tarde, Entre 18h e 20h, Entre 20h e 22h, Após 22h.)

### Seção 4: Preços e Taxas

**Pergunta:** "Na sua opinião, pagar pelas taxas de marketing cobradas pelos aplicativos de delivery impacta no seu negócio?" (Neutro, Impacta positivamente, Impacta negativamente.)

**Pergunta:** "Você acha que o valor das taxas para o uso do aplicativo de delivery é elevado?" (Sim/Não.)

**Pergunta:** "Qual você consideraria um valor justo para a taxa de comissão cobrada pelos aplicativos?" (Menos de 10%, 10% a 15%, 15% a 20%, 20% a 25%, mais de 25%.)

### Seção 5: Funcionalidades e Melhorias

**Pergunta:** "Quais incentivos fariam seu negócio priorizar o uso de um aplicativo de delivery?" (Taxas menores, maior visibilidade, promoções/descontos, suporte personalizado, programas de fidelidade, melhor controle de entregas, entre outros.)

**Pergunta:** "Que tipo de informações sobre os pedidos você gostaria de acessar facilmente no aplicativo?" (Tempo de entrega, dados do cliente, detalhamento do pedido, status em tempo real, histórico de pedidos, comissões e taxas por pedido, entre outros.)

### Seção 6: Experiência e Feedback

**Pergunta:** "Como você avalia os aplicativos de delivery?" (Escala de 1 a 5 – Muito ruim a Muito bom.)

**Pergunta:** "Cite alguma experiência positiva ou negativa que influenciou na sua avaliação." (Resposta aberta.)

### Seção 7: Extra

Pergunta opcional onde o participante pode sugerir pontos não abordados e fornecer contato para futuras pesquisas.

### 3.2.4 Análise de Sistemas Existentes

Avaliar os principais aplicativos de delivery no mercado para identificar boas práticas e funcionalidades desejáveis, além de limitações que podem ser melhoradas como: Funcionalidades de pesquisa e navegação, Processos de pagamento, Rastreabilidade do pedido, Experiência do usuário (**UI** e **UX**), Avaliações e comentários sobre os estabelecimentos, e Processo de cadastro e verificação de entregadores.

### 3.2.5 Criação de Documentação Detalhada

Documentar todos os requisitos funcionais e não funcionais levantados de maneira clara, para guiar o desenvolvimento.

Conteúdo:

1. **Requisitos Funcionais:** Funcionalidades que o sistema deve ter (ex. cadastro de usuário, pagamento, acompanhamento de pedido, avaliação).
  - Cadastro e *login* de usuários, entregadores e restaurantes
  - Criação e gerenciamento de pedidos
  - Sistema de pagamento integrado (cartão, **PIX**, etc.)
  - Notificações de atualização de status do pedido
  - Rastreamento em tempo real do pedido
  - Avaliação e feedback de restaurantes e entregadores
  - Filtros e categorias de busca para produtos e estabelecimentos
2. **Requisitos Não Funcionais:** Critérios de desempenho, usabilidade, segurança e disponibilidade.
  - **Desempenho:** Respostas rápidas (ex. carregamento de tela em menos de 2 segundos).
  - **Escalabilidade:** Suporte a alta quantidade de usuários e pedidos simultâneos.
  - **Segurança:** Criptografia de dados pessoais e financeiros, conformidade com a **LGPD**.
  - **Usabilidade:** Interface intuitiva e responsiva para diferentes dispositivos (desktop e mobile).
  - **Disponibilidade:** Sistema com alta disponibilidade (99,9% uptime esperado).
  - **Conformidade Legal:** Cumprimento de regulamentações de proteção ao consumidor e segurança de dados.

Técnicas de Documentação:

- Casos de uso
- Diagramas de fluxo de processo (fluxo de pedidos, entrega)
- Protótipos de interface (*wireframes*)

### 3.2.6 Modelagem de Casos de Uso

Figura 22 – Atores Principais



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

- Cliente: Cadastro, busca de produtos/restaurantes, adicionar ao carrinho, efetuar pagamento, acompanhar pedido, avaliar serviço.
- Entregador: Aceitar/Recusar pedido, ver informações de entrega, atualizar status de entrega, registrar entrega finalizada.
- Restaurante: Gerenciar cardápio, visualizar pedidos, atualizar status de pedido (preparação, pronto para entrega).

- Administrador: Gerenciar usuários (clientes e parceiros), monitorar atividade, gerar relatórios, gerenciar políticas de privacidade e [LGPD](#).

### 3.2.7 Diagrama de Sequência

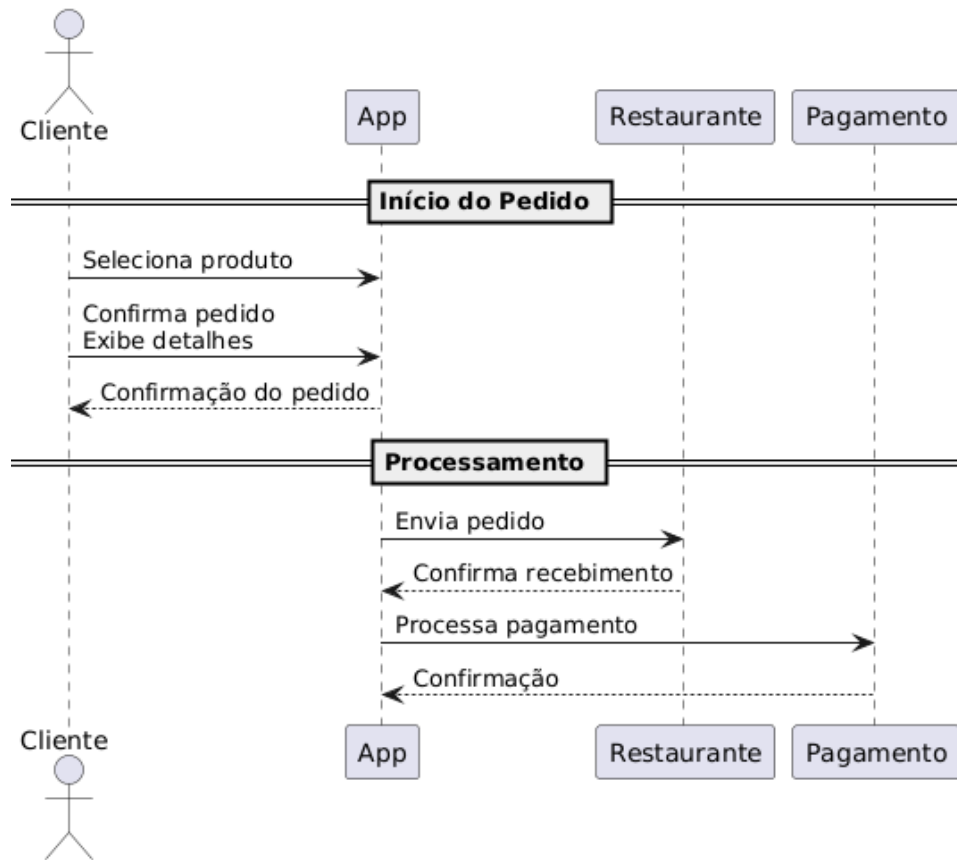
Realização de Pedido: Diagrama ilustrando a sequência de ações para o usuário realizar um pedido (cliente -> restaurante -> entregador).

Atualização de Status: Diagrama para representar como o status de um pedido é atualizado e notificado aos usuários (usuário -> restaurante -> entregador).

Rastreamento em Tempo Real: Representação das interações para que o cliente acompanhe a localização do entregador até a entrega do pedido.

O diagrama de sequência apresentado a seguir, Figura 23, descreve o fluxo de ações realizadas no processo de realização de um pedido por parte do cliente. O cenário envolve interações entre o cliente, o aplicativo, o restaurante e o sistema de pagamento. O cliente seleciona os produtos desejados, confirma o pedido e realiza o pagamento, enquanto o aplicativo se encarrega de repassar as informações ao restaurante e processar a confirmação da transação.

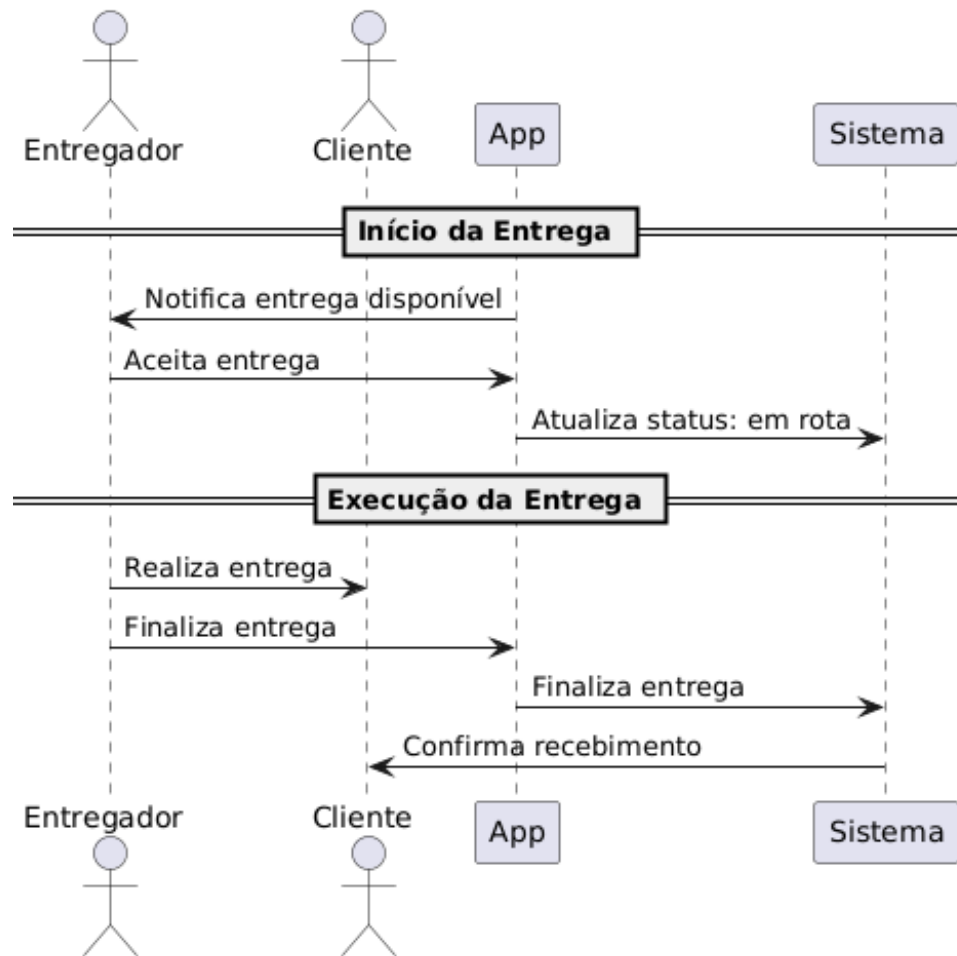
Figura 23 – Diagrama de sequência do processo de realização de pedido pelo cliente



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

O próximo diagrama de sequência, Figura 24, representa o processo de entrega do pedido realizado pelo entregador. A sequência inicia com a notificação de uma nova entrega pelo aplicativo, seguida da aceitação por parte do entregador, execução da entrega ao cliente e confirmação de recebimento. O sistema é responsável por atualizar o status da entrega e registrar a conclusão do serviço.

Figura 24 – Diagrama de sequência do processo de entrega realizada pelo entregador



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

### 3.2.8 Especificação Funcional

Funcionalidades principais: Cadastro, busca, pagamento, acompanhamento de pedidos, geolocalização, avaliações.

Regras de Negócio: Definir políticas de preços, comissões, promoções e políticas de cancelamento.

Fluxo de Operações: Descrever fluxos detalhados de cada funcionalidade.

## 3.3 Design e Implementação

Nesta sessão será definida a arquitetura do software e a estrutura detalhada de suas partes. Criação de diagramas de arquitetura, design de banco de dados, especificação da interface do usuário (*UI/UX*) e definição de componentes modulares. O projeto do sistema foi desenvolvido com base em princípios fundamentais da Engenharia de Software, como

modularidade, separação de responsabilidades e escalabilidade (PRESSMAN; MAXIM, 2016). Modelo de projeto ou protótipo, arquitetura detalhada (*software* e infraestrutura). Será utilizada uma arquitetura de três camadas:

**Frontend (Cliente):** Aplicativo *mobile* e/ou *web* com *UI* intuitiva para clientes, entregadores e restaurantes.

**Backend (Serviços de Negócio):** *APIs RESTful* para comunicação entre o app e os servidores, com lógica de negócios centralizada.

**Banco de Dados:** Sistema de armazenamento para gerenciar dados de usuários, pedidos, localização de entregadores, cardápios etc.

### 3.3.1 Design de Banco de Dados

#### 1. Entidades Principais:

- Usuário (cliente, entregador, restaurante, administrador)
- Pedido (detalhes, status, tempo estimado)
- Produto (cardápio, preço, descrição)
- Endereço (localização do cliente e do restaurante)
- Pagamento (método de pagamento, status)
- Localização (geoposicionamento em tempo real dos entregadores)

#### 2. Relações:

- Usuário - Pedido: Um cliente pode fazer vários pedidos; um pedido pertence a um cliente.
- Restaurante - Produto: Um restaurante pode ter vários produtos, mas um produto pertence a um restaurante.
- Pedido - Entregador: Um pedido pode ser associado a um entregador durante a entrega.

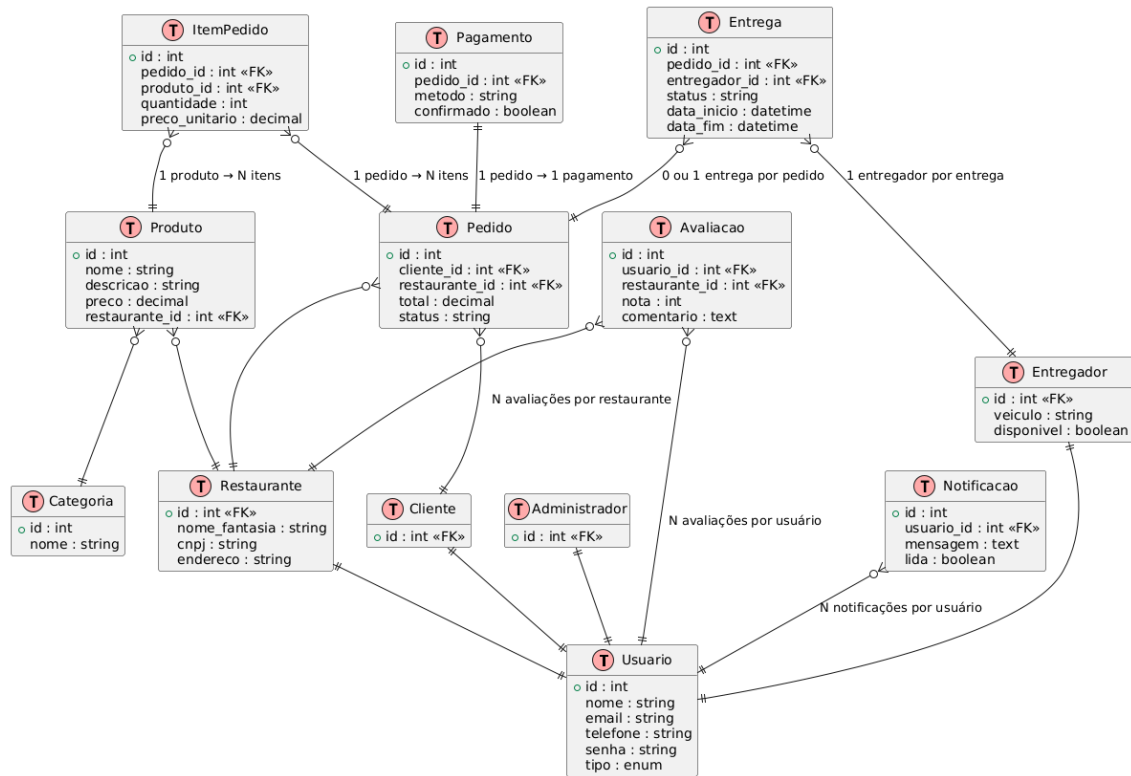
#### 3. Diagrama de Entidade-Relacionamento (ER):

- Visualizará as tabelas e relacionamentos, incluindo atributos como `id_usuario`, `id_pedido`, `status`, `data`, etc.

O Diagrama Entidade-Relacionamento (ER), apresentado na Figura 25, de forma simplificada, mostra os principais elementos que compõem o modelo de dados do protótipo do sistema proposto. As entidades centrais incluem usuários, restaurantes, entregadores, clientes e produtos, bem como os relacionamentos entre eles — como pedidos, pagamentos,

entregas e avaliações. Este modelo visa estruturar a lógica do banco de dados de forma clara e eficiente, permitindo o gerenciamento integrado das operações realizadas na plataforma, desde o cadastro dos usuários até o processamento completo dos pedidos.

Figura 25 – Diagrama Entidade-Relacionamento (ER) simplificado do sistema



Fonte: Elaborado pelo autor (2025).

### 3.3.2 UI/UX e Especificação da Interface

#### 1. Interfaces:

- Cliente: Tela de *login*, cadastro, escolha de restaurante, visualização de cardápio, carrinho, finalização de pedido, acompanhamento em tempo real e histórico de pedidos.
- Entregador: Tela de *login*, pedidos disponíveis, detalhes de pedido, navegação e entrega, finalização de pedido.
- Restaurante: Tela de *login*, cadastro de produtos, visualização de pedidos, confirmação e preparo.

#### 2. Princípios de Design *UI/UX*:

- Simplicidade: Design intuitivo com foco em minimizar o número de passos para realizar um pedido.

- **Acessibilidade:** Fontes legíveis, alto contraste e design responsivo.
- **Feedback Visual:** Mensagens e animações que guiem o usuário nas ações, especialmente em áreas como status do pedido e pagamento.
- **Prototipagem:** Usar *Figma* ou *Adobe XD* para criar protótipos de alta fidelidade para validação de *UI/UX* com usuários.

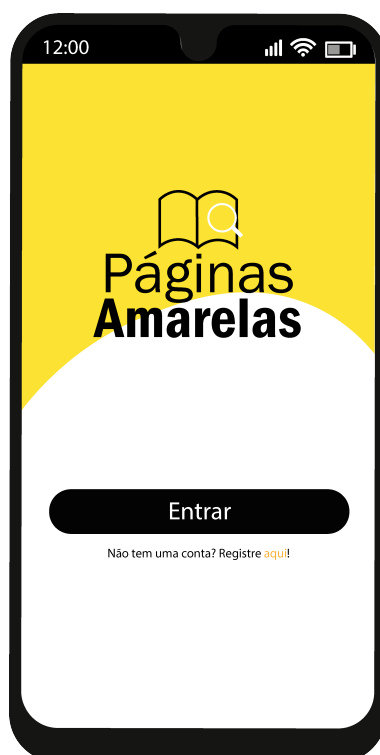
Classe Entrega foi mantida separadamente da entidade Pedido por representar o processo logístico e operacional da entrega, incluindo dados como o entregador responsável, o status da entrega e timestamps. Essa separação permite maior rastreabilidade, controle de concorrência e flexibilidade para futuras expansões da funcionalidade, como múltiplas entregas, rotas otimizadas ou análise de desempenho logístico.

Para validar a proposta visual e a experiência do usuário (*UI/UX*), foram desenvolvidas telas representativas do aplicativo utilizando ferramentas de prototipagem como *Figma*. As interfaces foram planejadas com foco na simplicidade, acessibilidade e coerência com a identidade visual do projeto.

As imagens a seguir representam a navegação por diferentes perfis de usuários e funcionalidades principais do sistema:

- **Figura 26 - Tela Inicial:** Exibe o nome do aplicativo “Páginas Amarelas” e oferece o botão de entrada para usuários cadastrados, além de link para registro de novos usuários.

Figura 26 – Tela Inicial



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

- **Figura 27 – Tela de Login:** Apresenta os campos para inserção de e-mail e senha, bem como a opção de autenticação via conta Google. Mantém o mesmo padrão visual da tela inicial, reforçando a identidade do sistema.

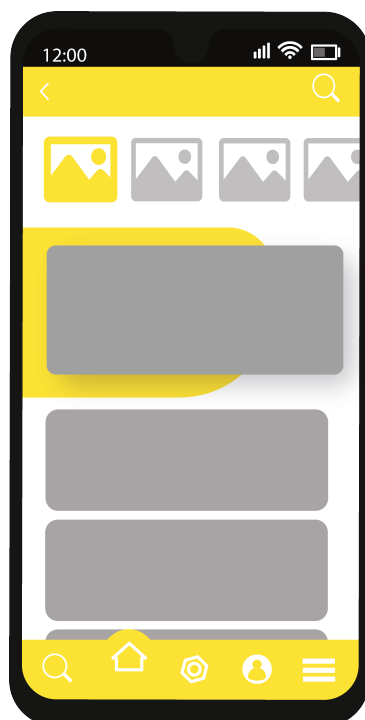
Figura 27 – Tela de Login



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

- **Figura 28 – Tela de Lista de Estabelecimentos:** Após a autenticação, o usuário é direcionado para a tela de visualização dos locais disponíveis. Os estabelecimentos são organizados em cartões com imagem, nome, avaliação e botão de acesso rápido. Há também filtros por categorias como Restaurantes, Farmácias, entre outros.

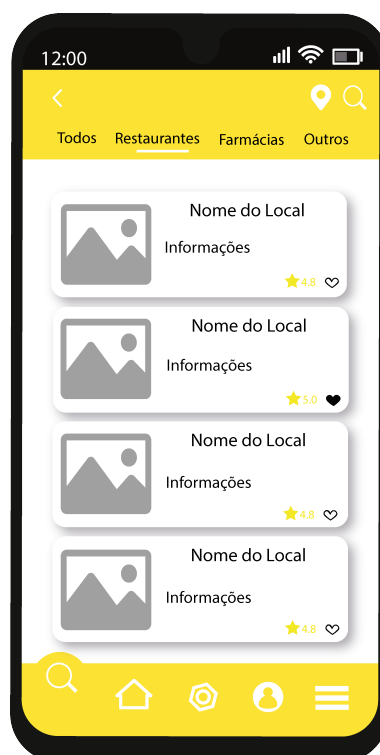
Figura 28 – Tela de Inicio



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

- **Figura 29 – Tela de Detalhes do Estabelecimento:** Nesta tela, o usuário pode acessar os detalhes de um local específico, visualizar os itens disponíveis para compra e realizar pedidos.

Figura 29 – Tela com Lista de Estabelecimentos



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

- **Figura 30 – Tela de Pedidos Recebidos (Restaurante):** Esta interface é destinada ao perfil do restaurante e apresenta a listagem dos pedidos realizados pelos clientes. Cada item da lista exibe informações como número do pedido, itens solicitados, valor total e status atual. A tela permite ao estabelecimento aceitar ou recusar os pedidos, funcionando como central de controle de entregas.

Figura 30 – Tela de Pedidos Recebidos (Restaurante)



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

- **Figura 31 – Tela de Solicitação de Entrega (Entregador):** Essa interface é exibida para o entregador, com a listagem dos pedidos disponíveis, suas respectivas informações (valor, itens e cliente), e botões para aceitar ou recusar o serviço.

Figura 31 – Tela Recebimento de Pedidos ao Entregador



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

Todas as interfaces foram desenhadas com base nas boas práticas de UX, como uso de cores de alto contraste, fontes legíveis, botões grandes para toque em dispositivos móveis e disposição clara dos elementos. A navegação é centrada no usuário e busca garantir uma experiência fluida e intuitiva para todos os perfis.

### 3.3.3 Definição de Componentes Modulares

#### 1. Módulos *Frontend*:

- Tela de *login*/cadastro: Validação de entradas e integração com o *backend*.
- Carrinho de compras: Interação com os produtos, aplicação de cupons, cálculo de total e envio ao *backend*.
- Acompanhamento de entrega: Integração com a *API* de mapas para atualizar a posição do entregador.

#### 2. Módulos *Backend*:

- Serviço de Pedidos: Inclui *APIs* para criar, atualizar e gerenciar status de pedidos.

- Serviço de Notificação: Envia notificações de atualização de status em tempo real.
- Serviço de Geolocalização: Calcula e atualiza a posição dos entregadores, otimiza rotas e faz o *tracking*.
- Para tratar acessos concorrentes: como dois entregadores tentando aceitar a mesma entrega, o backend deve implementar mecanismos de lock transacional e controle de estado via timestamp, garantindo que apenas a primeira requisição válida seja processada, rejeitando as subsequentes com status atualizado.

### 3. Módulo de Pagamento:

- Integração com *gateways* de pagamento, permitindo métodos diversos como cartão e [PIX](#).
- Sistema de segurança para proteger dados financeiros e confirmar transações.

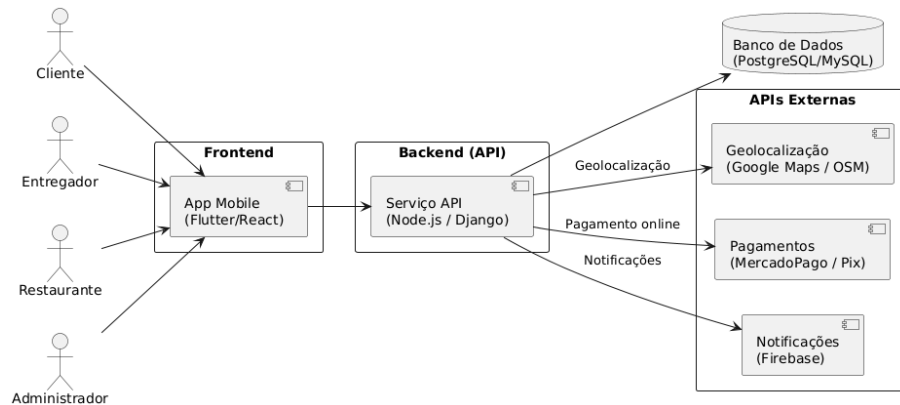
A arquitetura do sistema proposto adota o modelo cliente-servidor, com separação clara entre as camadas de apresentação, lógica de negócio e persistência de dados. A aplicação será composta por:

- **Frontend:** Interface web e/ou mobile desenvolvida em tecnologias como *Flutter* ou *React*, responsável pela interação direta com o usuário (cliente, entregador, restaurante).
- **Backend:** [API RESTful](#) desenvolvida em *Node.js* ou *Python* (com *frameworks* como *Express* ou *Django*), que centraliza as regras de negócio, controle de acesso, autenticação e autorização.
- **Banco de Dados:** *PostgreSQL* ou *MySQL*, com estrutura relacional modelada para garantir integridade, escalabilidade e rastreabilidade.
- **APIs externas:** integração com serviços como MercadoPago (pagamentos), *Google Maps* ou *OpenStreetMap* (geolocalização e cálculo de rotas), *Firebase Cloud Messaging* (notificações *push*).

Essa abordagem modular favorece a escalabilidade, a manutenção isolada de componentes e a adoção por diferentes organizações comunitárias com realidades tecnológicas diversas.

A Figura 32 ilustra a arquitetura geral do sistema.

Figura 32 – Arquitetura geral da solução de delivery solidário



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

### 3.3.4 Escolha da tecnologia

Utilizando informações já descritas no [3.1](#)

*Frontend:* *React Native* ou *Flutter* (para desenvolvimento híbrido, reduzindo custos de desenvolvimento e manutenção).

*Backend:* *Node.js* ou *Django (Python)*, com banco de dados *MongoDB* ou *PostgreSQL*.

Integração: *APIs RESTful* para comunicação entre *frontend* e *backend*. Infraestrutura: *AWS*, *Firebase* ou *Azure* para hospedagem, banco de dados e gerenciamento de usuários.

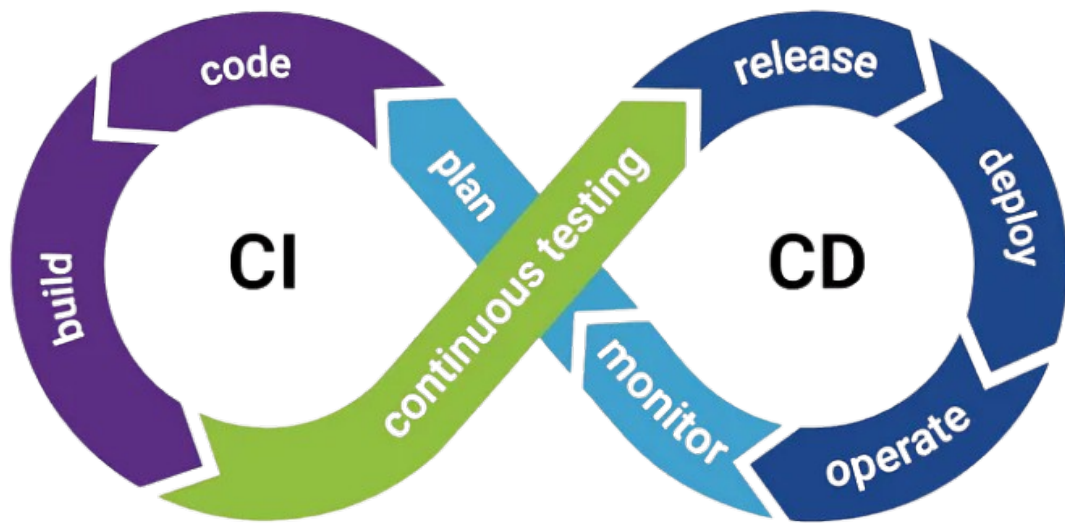
### 3.3.5 Atividades de Desenvolvimento

Especificação do código-fonte: Escrever o código do *backend* e *frontend* seguindo as melhores práticas de desenvolvimento (*clean code*, *SOLID*).

Integração de módulos: Conectar módulos de autenticação, gestão de pedidos, pagamento e notificações.

Versionamento e *CI/CD*: Configurar um *pipeline* de *CI/CD* (ex.: *GitHub Actions*) para testes automatizados e *deploy*.

Figura 33 – Versionamento e CI/CD



Fonte: (HABBEMA, 2021)

### 3.3.6 Estimativa de Custos

Os custos são divididos em:

Desenvolvimento: Contratação de desenvolvedores (*frontend*, *backend* e *mobile*), com valores que podem variar entre R\$50 a R\$150 por hora (GLASSDOOR, 2025).

Infraestrutura e hospedagem:

Servidores: AWS(AWSPRICE, 2025) ou *Firebase*(FIREBASEPRICE, 2025), com custo inicial entre R\$200 a R\$500/mês para projetos pequenos.

Banco de dados e *APIs*: Dependendo do volume de dados e uso, a partir de R\$100/mês.

*APIs* de pagamento e mapa: Cobrança por transação (pagamentos) e por chamada (*Google Maps*).

Manutenção e suporte: Aproximadamente 10% a 20% do custo total anual.

Custo estimado total (para um projeto de pequeno a médio porte): entre R\$ 60.000 e R\$ 90.000, considerando três a seis meses de desenvolvimento e uma equipe reduzida (APPMASER.IO, 2025).

### 3.3.7 Casos de Uso

O Diagrama de Casos de Uso, apresentado na Figura 34, mostra as principais interações entre os diferentes perfis de usuários do sistema e as funcionalidades disponi-

bilizadas pela plataforma. Os atores considerados são: cliente, restaurante, entregador e administrador. Cada um deles possui um conjunto específico de ações que refletem as responsabilidades e permissões associadas ao seu papel. O cliente, por exemplo, pode realizar pedidos e acompanhar entregas; o restaurante pode gerenciar seu cardápio e atualizar o status dos pedidos; o entregador é responsável pela logística de entrega; enquanto o administrador possui funções gerenciais e analíticas dentro do sistema.

Figura 34 – Diagramas de Casos de Uso da Plataforma de Delivery Solidário

Figura 35 – Casos de Uso - Cliente, Restau-  
rante

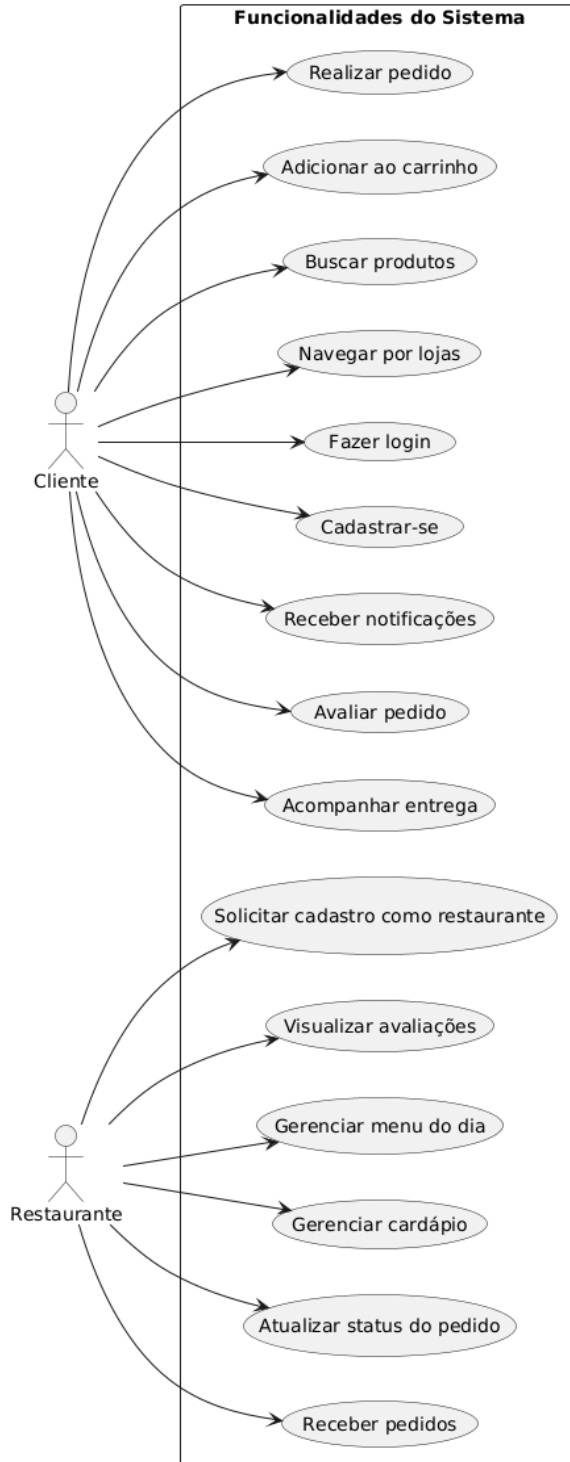
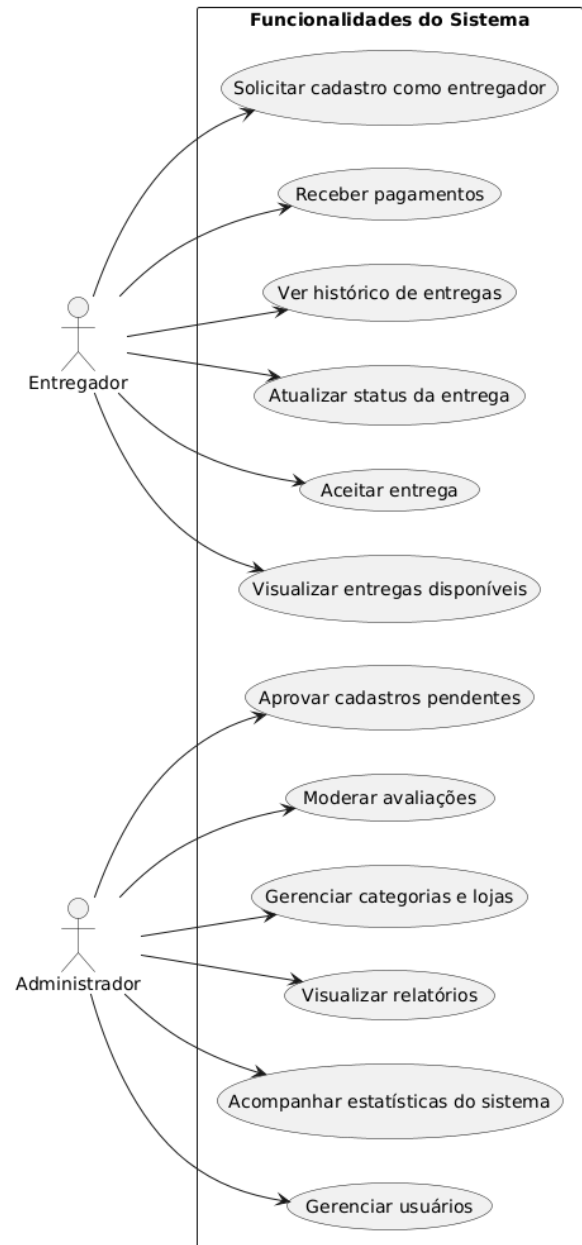


Figura 36 – Casos de Uso - Entregador, Ad-  
ministrador



Fonte: Elaborado pelo autor (2025).

### 3.3.7.1 Cliente - Cadastrar-se

- **Ator principal:** Cliente
- **Pré-condições:** Nenhuma
- **Pós-condição:** Conta criada e usuário autenticado
- **Fluxo principal:**
  1. Cliente acessa tela de cadastro
  2. Preenche dados pessoais obrigatórios
  3. Envia informações
  4. Sistema valida e cria conta de usuário

### 3.3.7.2 Cliente - Fazer Login

- **Ator principal:** Cliente
- **Pré-condições:** Conta já cadastrada
- **Pós-condição:** Sessão iniciada
- **Fluxo principal:**
  1. Cliente acessa tela de login
  2. Informa e-mail e senha
  3. Sistema valida e redireciona para a interface do usuário

### 3.3.7.3 Cliente - Realizar Pedido

- **Ator principal:** Cliente
- **Pré-condições:** Usuário autenticado, produtos disponíveis
- **Pós-condição:** Pedido criado e registrado no sistema
- **Fluxo principal:**
  1. Cliente seleciona produtos
  2. Cliente confirma pedido
  3. Pedido é enviado para o restaurante
  4. Sistema registra o pedido e notifica restaurante

#### 3.3.7.4 Cliente - Acompanhar Entrega

- **Ator principal:** Cliente
- **Pré-condições:** Pedido já realizado
- **Pós-condição:** Atualizações visíveis para o cliente
- **Fluxo principal:**
  1. Cliente acessa a aba "Meus Pedidos"
  2. Seleciona pedido em andamento
  3. Visualiza localização do entregador e status da entrega

#### 3.3.7.5 Cliente - Avaliar Pedido

- **Ator principal:** Cliente
- **Pré-condições:** Pedido entregue
- **Pós-condição:** Avaliação registrada
- **Fluxo principal:**
  1. Cliente acessa o pedido finalizado
  2. Informa nota e comentário
  3. Envia avaliação ao sistema

#### 3.3.7.6 Entregador ou Restaurante - Solicitar Cadastro

- **Ator principal:** Interessado (Entregador ou Restaurante)
- **Pré-condições:** Acesso ao formulário de cadastro
- **Pós-condição:** Cadastro submetido e aguardando aprovação
- **Fluxo principal:**
  1. Interessado acessa área de cadastro
  2. Preenche formulário com dados pessoais ou empresariais
  3. Submete solicitação
  4. Administrador analisa e aprova ou recusa

### 3.3.7.7 Restaurante - Gerenciar Menu do Dia

- **Ator principal:** Restaurante
- **Pré-condições:** Autenticação realizada
- **Pós-condição:** Menu do dia ajustado
- **Fluxo principal:**
  1. Restaurante acessa aba de gerenciamento do cardápio
  2. Seleciona itens especiais para destaque no dia
  3. Define horários ou restrições de disponibilidade
  4. Salva alterações

### 3.3.7.8 Entregador - Aceitar Entrega

- **Ator principal:** Entregador
- **Pré-condições:** Entregador autenticado e disponível
- **Pós-condição:** Entrega atribuída ao entregador
- **Fluxo principal:**
  1. Entregador acessa lista de entregas disponíveis
  2. Seleciona uma entrega
  3. Confirma aceite
  4. Sistema vincula entrega ao entregador

### 3.3.7.9 Entregador - Atualizar Status da Entrega

- **Ator principal:** Entregador
- **Pré-condições:** Entrega aceita e em andamento
- **Pós-condição:** Status atualizado no sistema
- **Fluxo principal:**
  1. Entregador acessa painel de entregas
  2. Atualiza status conforme a etapa: "em rota", "entregue"
  3. Sistema armazena e propaga atualização ao cliente

### 3.3.7.10 Restaurante - Editar Cardápio em Tempo Real

- **Ator principal:** Restaurante
- **Pré-condições:** Restaurante autenticado
- **Pós-condição:** Cardápio atualizado imediatamente
- **Fluxo principal:**
  1. Restaurante acessa painel de gerenciamento
  2. Altera item de cardápio (quantidade, descrição, preço, disponibilidade)
  3. Mudança é refletida para os usuários ativos

### 3.3.7.11 Administrador - Gerenciar Usuários

- **Ator principal:** Administrador
- **Pré-condições:** Administrador autenticado no sistema
- **Pós-condição:** Usuários gerenciados (bloqueados, removidos ou editados)
- **Fluxo principal:**
  1. Administrador acessa painel de usuários
  2. Localiza o usuário desejado
  3. Aplica ação (editar, bloquear ou excluir)
  4. Sistema confirma a modificação

### 3.3.7.12 Administrador - Acompanhar Estatísticas do Sistema

- **Ator principal:** Administrador
- **Pré-condições:** Dados já processados pelo sistema
- **Pós-condição:** Informações visualizadas no dashboard
- **Fluxo principal:**
  1. Administrador acessa dashboard
  2. Visualiza gráficos e relatórios de uso
  3. Analisa métricas de pedidos, entregas, usuários e avaliações

### 3.3.7.13 Administrador - Visualizar Relatórios

- **Ator principal:** Administrador
- **Pré-condições:** Dados registrados no sistema
- **Pós-condição:** Relatório exportado ou analisado
- **Fluxo principal:**
  1. Administrador acessa área de relatórios
  2. Escolhe período e tipo de relatório
  3. Sistema gera e exibe relatório solicitado
  4. Administrador pode exportar os dados, se desejar

### 3.3.7.14 Administrador - Gerenciar Categorias e Lojas

- **Ator principal:** Administrador
- **Pré-condições:** Acesso administrativo validado
- **Pós-condição:** Categorias ou lojas criadas, atualizadas ou removidas
- **Fluxo principal:**
  1. Acessa painel de gestão de categorias/lojas
  2. Insere, edita ou exclui registros
  3. Confirma a operação
  4. Sistema atualiza as informações para os usuários

### 3.3.7.15 Administrador - Moderar Avaliações

- **Ator principal:** Administrador
- **Pré-condições:** Avaliações publicadas no sistema
- **Pós-condição:** Avaliações mantidas, editadas ou removidas
- **Fluxo principal:**
  1. Administrador acessa área de moderação
  2. Visualiza novas avaliações publicadas
  3. Avalia se o conteúdo viola políticas
  4. Aprova, edita ou remove a avaliação

### 3.3.7.16 Administrador - Aprovar Cadastros Pendentes

- **Ator principal:** Administrador
- **Pré-condições:** Solicitações de cadastro recebidas
- **Pós-condição:** Solicitação aprovada ou recusada
- **Fluxo principal:**
  1. Administrador acessa painel de cadastros pendentes
  2. Visualiza informações enviadas por entregadores ou restaurantes
  3. Analisa validade e conformidade dos dados
  4. Aprova ou recusa a solicitação

### 3.3.8 Papéis e Adaptação no Projeto

Embora o *Scrum* preveja a atuação de um *Product Owner*, um *Scrum Master* e um *Development Team*, as adaptações necessárias para o contexto de um TCC envolveram a centralização desses papéis no próprio autor do trabalho, que atuou tanto como desenvolvedor quanto como gestor das tarefas e do *backlog*.

Mesmo com a equipe reduzida a uma única pessoa, foram utilizados conceitos fundamentais como:

***Backlog do Produto:*** Criado com base nos requisitos levantados via formulários aplicados a consumidores, empreendedores e entregadores.

***Sprints:*** O desenvolvimento foi dividido em ciclos quinzenais, com metas claras, como “implementar o CRUD de pedidos” ou “desenvolver tela de rastreamento.”

***Daily Meetings (adaptadas):*** Foram feitas anotações diárias sobre os avanços e impedimentos, funcionando como retrospectivas pessoais para a tomada de decisões técnicas.

***Sprint Reviews:*** Ao final de cada ciclo, o protótipo era testado e ajustado com base em critérios definidos inicialmente no *backlog*.

#### 3.3.8.1 Papéis e Adaptação no Projeto

Durante o desenvolvimento, os papéis previstos na metodologia foram adaptados à realidade de um projeto individual. Nesta seção, descrevem-se essas adaptações e exemplos práticos que ilustram como os conceitos foram aplicados.

1. Definição de Requisitos pelo Usuário Final: Nesta fase, o papel de *Product Owner* foi simulado por meio de entrevistas informais e questionários aplicados aos stakeholders, principalmente consumidores e comerciantes locais. Suas respostas guiaram a formulação do *backlog* inicial, priorizando funcionalidades como: cardápio dinâmico, notificações automáticas e integração com métodos de pagamento simplificados.
2. Gerenciamento de Iterações e *Sprints*: O papel de *Scrum Master* foi adaptado para um modelo de autogestão. As entregas foram divididas em ciclos semanais, com metas definidas ao final de cada período. Mesmo em um projeto de um único desenvolvedor, foi possível simular *sprints* e retrospectivas, permitindo reavaliações constantes das funcionalidades e priorizações.

### 3.3.8.2 Exemplo de Aplicação Prática

Durante o desenvolvimento da funcionalidade de **rastreamento em tempo real**, por exemplo, o item foi inicialmente adicionado ao *backlog* como:

"Permitir que o cliente visualize a localização atual do entregador em tempo real durante a entrega."

Na *sprint* seguinte, essa tarefa foi dividida em subtarefas como:

- Integração da [API](#) de mapas (*Google Maps*);
- Armazenamento da localização em tempo real no *back-end*;
- Atualização automática da interface do cliente.

Essa abordagem possibilitou que, ao final da *sprint*, o incremento estivesse funcional e testável, ainda que em ambiente de desenvolvimento.

## 4 Conclusão

Este trabalho teve como objetivo propor a modelagem de um sistema de delivery digital voltado à economia solidária, buscando oferecer uma alternativa técnica e organizacional às plataformas tradicionais de entrega. A proposta se baseia em princípios como autogestão, cooperação, soberania tecnológica e valorização da economia local.

Através da análise de plataformas existentes, da coleta de dados com diferentes perfis de usuários (clientes, entregadores e empreendedores) e da aplicação de práticas consolidadas de engenharia de software, foi possível estruturar um modelo funcional e conceitual que serve como base para soluções coletivas de delivery.

A fundamentação teórica abordou temas como proteção de dados, metodologias ágeis, prototipação e modelagem UML, permitindo um alinhamento entre as necessidades sociais e as exigências técnicas de um sistema moderno. O modelo proposto contempla múltiplos perfis de usuário, modularização, integração com APIs externas e controle de acesso concorrente, incluindo diferenciais como a possibilidade de atualização do cardápio em tempo real, sem necessidade de interrupção do serviço.

Acredita-se que esta proposta possa ser adotada por cooperativas, coletivos ou associações, servindo de base para plataformas de delivery mais justas, acessíveis e sustentáveis. Embora não tenha havido implementação do sistema neste trabalho, a modelagem apresentada já oferece subsídios suficientes para que tal desenvolvimento ocorra de forma orientada, colaborativa e replicável.

## 5 Trabalhos futuros

Como continuidade deste trabalho, propõe-se as seguintes frentes de aprofundamento e aplicação:

- **Desenvolvimento funcional do sistema:** implementar o protótipo descrito, validando fluxos com usuários reais em ambiente controlado.
- **Criação de cooperativa digital:** realizar um estudo multidisciplinar para orientar a formação de uma cooperativa de entregadores e estabelecimentos que possa adotar e gerir o sistema de forma autônoma.
- **Aprimoramento da segurança:** implementar criptografia de dados sensíveis, autenticação e mecanismos de auditoria para reforçar a conformidade com a [LGPD](#).
- **Painel de gestão e BI:** desenvolver uma interface administrativa com indicadores em tempo real sobre pedidos, entregas, avaliações e desempenho de usuários.
- **Expansão de funcionalidades:** incluir funcionalidades como cupons de desconto, integração com cooperativas de agricultura familiar e logística reversa.
- **Validação em ambiente real:** realizar testes-piloto em comunidades, bairros ou redes de economia solidária, ajustando o modelo com base em *feedbacks* e métricas de desempenho.

# Referências

- ABRAHAMSSON, P. et al. Agile software development methods: Review and analysis. *arXiv preprint*, 2017. Revisão sobre métodos ágeis e prototipação. Citado na página 34.
- AIQFOME-AJUDA. *aiqfome*. 2025. <<https://aiqfome.zendesk.com/hc/pt-br>>. Acesso em: 02 abr. 2025. Citado na página 49.
- AIQFOME-SOBRE. *aqui é sobre o aiqfome*. 2025. <<https://aiqfome.zendesk.com/hc/pt-br/categories/360003142034-aqui-%C3%A9-sobre-o-aiqfome>>. Acesso em: 02 abr. 2025. Citado na página 45.
- AIQFOME-TERMOS. *Aiqfome - termos de uso*. 2025. <<https://aiqfome.com/termos/BR>>. Acesso em: 02 abr. 2025. Citado na página 49.
- ANALYTICA, C. *Cambridge Analytica*. 2025. <<https://cambridgeanalytica.org/>>. Acesso em: 02 abr. 2025. Citado na página 20.
- ANDERSON, D. J. *Kanban: Successful Evolutionary Change for Your Technology Business*. [S.l.]: Blue Hole Press, 2010. Citado na página 40.
- APPMASTER.IO. *Quanto custa desenvolver um aplicativo?* 2025. <<https://appmaster.io/pt/blog/custo-para-desenvolver-um-aplicativo>>. Acesso em: 02 abr. 2025. Citado na página 78.
- AppsFlyer. *App Annie: Brasil foi o 5º país que mais baixou aplicativos em 2021*. 2021. Acesso em: 02 abr. 2025. Disponível em: <<https://www.appsflyer.com/br/resources/reports/app-annie-mobile-2021/>>. Citado na página 17.
- AWSPRICE. *AWS Pricing Calculator*. 2025. <<https://calculator.aws/#/>>. Acesso em: 02 abr. 2025. Citado na página 78.
- BECK, K.; ANDRES, C. *Extreme Programming Explained: Embrace Change*. 2. ed. [S.l.]: Addison-Wesley, 2004. Citado 2 vezes nas páginas 39 e 43.
- BECK, K. et al. *Manifesto Ágil*. 2001. Acesso em: 20 maio 2025. Disponível em: <<https://agilemanifesto.org/iso/ptbr/manifesto.html>>. Citado na página 39.
- BOEHM, B. W. A spiral model of software development and enhancement. *ACM SIGSOFT Software Engineering Notes*, ACM, v. 11, n. 4, p. 14–24, 1988. Citado 2 vezes nas páginas 37 e 38.
- Brasil de Fato. *Setor de energia lidera reversão de privatizações no mundo com foco no ramo da Enel*. 2024. Acesso em: 02 abr. 2025. Disponível em: <<https://www.brasildefato.com.br/2024/10/25/setor-de-energia-lidera-reversao-de-privatizacoes-no-mundo-com-foco-no-ramo-da-enel>>. Citado na página 21.
- CALIFORNIA, S. of. *TITLE 1.81.5. California Consumer Privacy Act of 2018 [1798.100 - 1798.199.100]*. 2025. <[https://leginfo.legislature.ca.gov/faces/codes\\_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5](https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5)>. Acesso em: 02 abr. 2025. Citado na página 24.

- CETIC.br. *TIC Domicílios 2022: Pesquisa sobre o uso das tecnologias de informação e comunicação no Brasil*. 2022. Acesso em: 02 abr. 2025. Disponível em: <<https://cetic.br/pesquisa/domicilios/>>. Citado na página 17.
- COUNCIL, E. *GDPR EU*. 2025. <<https://www.consilium.europa.eu/en/policies/data-protection/data-protection-regulation/>>. Acesso em: 02 abr. 2025. Citado na página 25.
- Energy Transition. *Privatization of public services is failing, says think tank*. 2017. Acesso em: 02 abr. 2025. Disponível em: <<https://energytransition.org/2017/08/privatization-of-public-services-is-failing-says-think-tank/>>. Citado na página 22.
- FARIAS, D. S. d.; BARROS, V. D. *Sistema de Delivery para Restaurantes de Pequeno Porte*. 2014. Trabalho de Conclusão de Curso, UTFPR – Universidade Tecnológica Federal do Paraná. Disponível em: <[https://repositorio.utfpr.edu.br/jspui/bitstream/1/20063/3/FB\\_DESIDM\\_I\\_2014\\_14.pdf](https://repositorio.utfpr.edu.br/jspui/bitstream/1/20063/3/FB_DESIDM_I_2014_14.pdf)>. Citado na página 45.
- FIREBASEPRICE. *Firebase Pricing*. 2025. <<https://firebase.google.com/pricing?hl=pt-br>>. Acesso em: 02 abr. 2025. Citado na página 78.
- GIL, A. C. *Métodos e técnicas de pesquisa social*. 6. ed. São Paulo: Atlas, 2008. Citado na página 56.
- GLASSDOOR. *Salários de empresas*. 2025. <<https://www.glassdoor.com.br/Sal%C3%A1rios/index.htm>>. Acesso em: 02 abr. 2025. Citado na página 78.
- Globo; Kantar IBOPE. *Pesquisa Hábitos Digitais dos Brasileiros*. 2023. Acesso em: 02 abr. 2025. Disponível em: <<https://gente.globo.com/>>. Citado na página 17.
- GOV.BR. *Lei Geral de Proteção de Dados Pessoais (LGPD)*. 2025. <<https://www.gov.br/esporte/pt-br/acesso-a-informacao/lgpd>>. Acesso em: 02 abr. 2025. Citado na página 23.
- GUARDIAN, T. *The Cambridge Analytica Files*. 2025. <<https://www.theguardian.com/news/series/cambridge-analytica-files>>. Acessado em: 25 mar. 2025. Citado na página 20.
- HABBEMA, C. *Desvendando CI/CD*. 2021. Acesso em: 13 maio 2025. Disponível em: <<https://medium.com/@habbema/desvendando-ci-cd-b56f515ddd20>>. Citado na página 78.
- IFOOD-AJUDA. *Como podemos ajudar?* 2025. <<https://www.ifood.com.br/ajuda>>. Acesso em: 02 abr. 2025. Citado na página 49.
- IFOOD-SOBRE. *iFood | Comida é nossa paixão. Tecnologia, o nosso talento*. 2025. <<https://institucional.ifood.com.br/>>. Acesso em: 02 abr. 2025. Citado na página 46.
- IFOOD-TERMOS. *Termos e Condições de Uso da Plataforma iFood para os usuários*. 2025. <<https://webmiddleware.ifood.com.br/termos>>. Acesso em: 02 abr. 2025. Citado na página 49.
- KASHFI, P.; NILSSON, A.; FELDT, R. Integrating user experience practices into software development processes. *arXiv preprint*, 2016. Retrieved from arXiv.org. Citado na página 35.

- KIM, G.; BEHR, K.; SPAFFORD, G. *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win*. [S.l.]: IT Revolution Press, 2016. Citado na página 41.
- KRUCHTEN, P. *The Rational Unified Process: An Introduction*. 3. ed. [S.l.]: Addison-Wesley, 2004. Citado na página 43.
- MORTIER, G. du. *Top 12 Database Design Principles in 2023*. 2023. Disponível em: <<https://vertabelo.com/blog/database-design-principles/>>. Citado 2 vezes nas páginas 28 e 30.
- OMG. *Use case*. 2025. Definição de caso de uso no contexto de sistemas. Disponível em: <[https://en.wikipedia.org/wiki/Use\\_case](https://en.wikipedia.org/wiki/Use_case)>. Citado na página 28.
- PALM, J.; MALM, A.; HÖRNSTRÖM, L. Remunicipalization and the policy cycle: A comparative analysis of european public service provision. *Journal of Economic Policy Reform*, 2023. Acesso em: 02 abr. 2025. Disponível em: <<https://journals.sagepub.com/doi/full/10.1177/10780874241233535>>. Citado na página 22.
- PARADIGM, V. *What is Use Case Diagram in UML?* 2024. Disponível em: <<https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/>>. Citado na página 31.
- PRATAMA, M.; CAHYADI, A. Effect of user interface and user experience on application sales. *IOP Conf. Ser.: Mater. Sci. Eng.*, v. 879, 2020. Citado na página 35.
- PRESSMAN, R. S.; MAXIM, B. R. *Engenharia de Software*. 8. ed. [S.l.]: McGraw Hill Brasil, 2016. Citado 3 vezes nas páginas 36, 42 e 67.
- RICCIARDI, A. *UML Diagrams: A Guide for Software Engineers*. 2024. Disponível em: <<https://www.alexomegapy.com/post/uml-diagrams-a-guide-for-software-engineers>>. Citado na página 33.
- SCHWABER, K.; SUTHERLAND, J. *The Scrum Guide*. [S.l.]: Scrum.org, 2017. Disponível em: <<https://www.scrumguides.org/scrum-guide.html>>. Citado 3 vezes nas páginas 41, 43 e 44.
- SILVA, J. P. S. *Desenvolvimento de Aplicativo de Delivery para Pequenos Comerciantes*. 2023. Trabalho de Conclusão de Curso, UFOP – Universidade Federal de Ouro Preto. Disponível em: <<https://www.monografias.ufop.br/handle/35400000/3661>>. Citado na página 45.
- SOCIETY, I. C. *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. 2014. Definição padrão de Engenharia de Software. Disponível em: <[https://en.wikipedia.org/wiki/Software\\_Engineering\\_Body\\_of\\_Knowledge](https://en.wikipedia.org/wiki/Software_Engineering_Body_of_Knowledge)>. Citado na página 27.
- SOMMERVILLE, I. *Software Engineering*. 9. ed. [S.l.]: Addison-Wesley, 2011. Conceito de Engenharia de Software como disciplina de produção de software. Citado 3 vezes nas páginas 27, 37 e 51.
- SPRINGER. *Database Design Overview*. 2022. Capítulo "Database Design Fundamentals". Disponível em: <[https://link.springer.com/chapter/10.1007/978-981-19-3032-4\\_7](https://link.springer.com/chapter/10.1007/978-981-19-3032-4_7)>. Citado 2 vezes nas páginas 30 e 32.

Transnational Institute. *Reclaiming Public Services: How cities and citizens are turning back privatisation*. 2017. Acesso em: 02 abr. 2025. Disponível em: <[https://www.tni.org/files/publication-downloads/chapter\\_1\\_reclaiming\\_public\\_services.pdf](https://www.tni.org/files/publication-downloads/chapter_1_reclaiming_public_services.pdf)>. Citado na página 21.

UAIRANGO-AJUDA. *Perguntas Frequentes*. 2025. <<https://www.uairango.com/faq>>. Acesso em: 02 abr. 2025. Citado na página 49.

UAIRANGO-SOBRE. *Sobre o UaiRango Delivery*. 2025. <<https://www.uairango.com/sobre>>. Acesso em: 02 abr. 2025. Citado na página 46.

UAIRANGO-TERMOS. *Termos de Uso*. 2025. <<https://www.uairango.com/politica>>. Acesso em: 02 abr. 2025. Citado na página 49.

# Apêndices

# APÊNDICE A – Respostas Consumidores

Carimbo de data/hora	Quais são as maiores dificuldades que você enfrenta ao usar aplicativos de delivery?	Com qual frequência é seu uso de aplicativos de delivery? Qual sua faixa etária?	O que você mais valoriza em um aplicativo de delivery? Você gostaria de funcionalidades extras, como age
17/01/2025 14:38:09	Algumas empresas não oferecem retirada no local e querem que eu pague o delivery	7 27 - 37	Cupons de Desconto, Preço, Promoções Sim
17/01/2025 14:38:25	Dificuldade para localizar pratos específicos	6 27 - 37	Cupons de Desconto, Preço, Promoções, Tempo de Sim
17/01/2025 14:42:23	Uma lista unificada de produtos, sem precisar entrar na página de cada loja.	5 27 - 37	Cupons de Desconto, Preço, Promoções Tanto Faz
17/01/2025 14:44:46	Nenhuma	2 27 - 37	Preço, Promoções Tanto Faz
17/01/2025 14:54:53	Preços mais altos que na compra direta com o restaurante e superexploração dos entregadores	3 27 - 37	Preço Tanto Faz
17/01/2025 15:21:04	Mecanismo de busca de ingredientes específicos	3 27 - 37	Preço Sim
17/01/2025 15:51:17	Encontrar um preço acessível ,tempo de entrega	1 50+	Preço, Tempo de Entrega, Variedade de lojas Tanto Faz
17/01/2025 16:15:16	Não tenho	6 50+	Preço Sim
17/01/2025 16:23:15	Resgate do cupom não funciona.	5 18 - 26	Cupons de Desconto, Preço, Promoções, Variedade Sim
17/01/2025 16:27:15	Entrega	1 50+	Promoções Sim
17/01/2025 16:39:56	Achar o que eu procuro, e o tempo de entrega.	3 50+	Tempo de Entrega Sim
17/01/2025 16:40:04	algo que raramente acontece mas que ainda assim, infelizmente, acontece, é quando o entregador f	5 18 - 26	Tempo de Entrega, Variedade de lojas Tanto Faz
17/01/2025 16:56:09	Uso e entrega	1 50+	Tempo de Entrega, Variedade de lojas Sim
17/01/2025 17:11:11	Nenhuma	4 18 - 26	Cupons de Desconto Tanto Faz
17/01/2025 17:42:55	-	4 27 - 37	Cupons de Desconto, Preço, Promoções Sim
17/01/2025 18:15:09	Nenhuma	7 18 - 26	Cupons de Desconto, Preço, Tempo de Entrega Sim
17/01/2025 18:43:44	Demora para entregar	8 27 - 37	Cupons de Desconto, Promoções Sim
17/01/2025 19:12:02	Informações erradas	8 18 - 26	Cupons de Desconto, Tempo de Entrega Tanto Faz
17/01/2025 19:39:42	Tempo de espera.	6 27 - 37	Cupons de Desconto, Preço, Promoções Sim
17/01/2025 19:57:07	Taxas de entrega alta	8 18 - 26	Cupons de Desconto, Promoções, Tempo de Entrega Sim
17/01/2025 23:08:33	Não avisar quando o pedido sai para entrega	3 18 - 26	Cupons de Desconto, Preço, Tempo de Entrega Sim
17/01/2025 23:35:26	Quando encontro dificuldade entro em contato pra esclarecimentos	4 50+	Preço, Tempo de Entrega Sim
17/01/2025 23:39:39	Saber quando o restaurante está fechado	8 27 - 37	Cupons de Desconto Sim
18/01/2025 08:58:29	Não tenho dificuldades no uso.	9 27 - 37	Preço, Tempo de Entrega, Variedade de lojas Sim
18/01/2025 09:57:24	Nenhum	6 18 - 26	Cupons de Desconto, Preço, Promoções Sim
18/01/2025 11:28:28	Acompanhar a entrega	6 27 - 37	Preço, Promoções Tanto Faz
18/01/2025 11:36:48	suporte	6 27 - 37	Cupons de Desconto, Promoções, Tempo de Entrega Tanto Faz
18/01/2025 11:41:58	Tempo de espera	1 18 - 26	Cupons de Desconto, Preço, Promoções, Tempo de Sim
18/01/2025 12:05:08	A demora na entrega	3 38 - 49	Cupons de Desconto Sim
18/01/2025 12:13:04	Demora	1 50+	Tempo de Entrega Tanto Faz
18/01/2025 16:30:17	As descrições dos produtos, ou são muito pequenas e não específicas, ou geradas por IA, muito gra	6 27 - 37	Preço, Tempo de Entrega Sim
18/01/2025 17:23:01	Nenhuma	6 27 - 37	Cupons de Desconto, Tempo de Entrega Sim
18/01/2025 17:59:17	Valor da entrega	7 18 - 26	Cupons de Desconto Sim
18/01/2025 18:11:16	.	5 18 - 26	Promoções Sim
18/01/2025 18:38:03	Não tenho nenhuma dificuldade	5 27 - 37	Cupons de Desconto, Promoções Sim
19/01/2025 00:59:05	Dicidir diante das possibilidades	6 27 - 37	Cupons de Desconto, Preço, Promoções Tanto Faz
19/01/2025 10:09:45	Encontrar preço e qualidade	10 18 - 26	Preço Sim
19/01/2025 10:31:47	Fazerem entrega no meu bairro a noite	6 18 - 26	Preço, Promoções Tanto Faz
19/01/2025 11:22:11	Geralmente eu vou buscar meus pedidos, pois tenho condições	5 27 - 37	Tempo de Entrega Sim
19/01/2025 21:27:28	Tempo de preparo/entrega	4 27 - 37	Cupons de Desconto, Tempo de Entrega, Variedade Sim
20/01/2025 12:42:16	A demora .	1 50+	Preço, Tempo de Entrega Sim

Que tipos de filtros ou categorias ajudariam você a		Promoções ou cupons influenciam sua decisão de	Você considera importante saber sobre a origem d	Estaria disposto a pagar uma taxa adicional para e	Qual foi sua experiência com suporte ao cliente em	De acordo com a pergunta anterior, o que poderia s
Mostrar apenas Farmácias, Mostrar apenas Lanch	Sim	Não	Não	5	O tempo de resposta	
Todas opções acima	Sim	Sim	Não	3	Maior rapidez e facilidade da funcionalidade	
Mostrar apenas Lanchonetes, Mostrar apenas Rest	Sim	Tanto Faz	Não	3	Nunca usei.	
Todas opções acima	Sim	Sim	Não	3	Nunca utilizei o suporte	
Todas opções acima	Sim	Não	Não	3	Maior agilidade na resposta	
Mostrar apenas Lanchonetes, Mostrar lojas com er	Sim	Sim	Sim	3	Celeridade	
Mostrar lojas com entrega grátis, Mostrar apenas F	Sim	Sim	Não	2	Tempo de entrega	
Mostrar lojas com entrega grátis, Todas opções ao	Sim	Sim	Não	3	Tempo de resposta	
Todas opções acima	Sim	Sim	Sim	5	Até o momento não tive nenhum problema com o s	
Todas opções acima	Sim	Sim	Sim	2	Mais breve	
Todas opções acima	Talvez	Sim	Sim	2	Tempo de resposta.	
Todas opções acima	Talvez	Sim	Tanto Faz	2	muitas das vezes o atendimento é feito por bot e re	
Todas opções acima	Talvez	Sim	Sim	2	Tempo de resposta e atendimento	
Mostrar lojas com entrega grátis	Sim	Tanto Faz	Não	5	Nada	
Todas opções acima	Sim	Não	Não	3	-	
Todas opções acima	Sim	Tanto Faz	Não	5	Não tenho nada a reclamar	
Todas opções acima	Sim	Sim	Sim	3	Atendimento mais detalhado	
Todas opções acima	Sim	Tanto Faz	Tanto Faz	4	Informações sobre tempo de entrega correto	
Mostrar apenas Lanchonetes	Sim	Sim	Sim	5	Nada, sempre fui bem atendido.	
Mostrar apenas Lanchonetes, Mostrar lojas com er	Sim	Tanto Faz	Não	3	Tempo de resposta e agilidade para resolver	
Todas opções acima	Sim	Sim	Sim	3	Nunca usei o suporte ao cliente	
Todas opções acima	Sim	Sim	Sim	5	Deu tudo certo	
Todas opções acima	Não	Tanto Faz	Tanto Faz	1	Tempo de atendimento	
Mostrar apenas Lanchonetes, Mostrar apenas Rest	Sim	Sim	Não	4	Velocidade de atendimento pelo aplicativo e pelas	
Todas opções acima	Sim	Sim	Sim	3	Suporte	
Mostrar apenas Lanchonetes, Mostrar apenas Rest	Sim	Não	Não	3	Nunca tive contato com o suporte. Na resposta aci	
Todas opções acima	Sim	Sim	Sim	1	respostas mais rápidas	
Todas opções acima	Sim	Sim	Sim	4	Um suporte mais eficiente	
Todas opções acima	Sim	Sim	Sim	2	Atendimento	
Todas opções acima	Talvez	Tanto Faz	Não	5	Está bom assim	
Todas opções acima	Sim	Sim	Não	5	Foi boa	
Todas opções acima	Sim	Sim	Não	4	Comunicar com o cliente quando a entrega será m	
Mostrar apenas Lanchonetes, Mostrar apenas Rest	Sim	Sim	Não	3	Valor de entrega	
Mostrar lojas com entrega grátis	Sim	Sim	Tanto Faz	4	.	
Todas opções acima	Sim	Tanto Faz	Não	5	Para mim está ok	
Mostrar apenas Lanchonetes, Mostrar apenas Men	Sim	Tanto Faz	Não	3	Agilidade	
Mostrar apenas Lanchonetes, Mostrar apenas Rest	Sim	Sim	Não	3	Tempo de resposta	
Mostrar apenas Farmácias, Mostrar apenas Lanch	Não	Tanto Faz	Sim	3	Tempo e valor da entrada	
Todas opções acima	Sim	Sim	Sim	4	Um pouco mais de rapidez no retorno	
Todas opções acima	Sim	Tanto Faz	Sim	5	Eu nunca tive problema com o suporte.	
Mostrar lojas com entrega grátis	Sim	Sim	Sim	2	Educação e rapidez	

Como você prefere ser notificado sobre o status de
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
SMS
Notificação no aplicativo
E-mail
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo, SMS
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo, SMS
Notificação no aplicativo
E-mail, Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
E-mail, Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo
Notificação no aplicativo, SMS
Notificação no aplicativo

## APÊNDICE B – Respostas Empreendedores

Carimbo de data/hora	Qual o ramo de atuação do seu negócio?	Possui serviço de entrega por aplicativo de delivery	Qual aplicativo de delivery você utiliza?	Qual o maior desafio do seu empreendimento ao tr
21/01/2025 03:22:03	Outros	Sim	Zé Delivery	Falta de personalização
24/01/2025 21:48:59	Farmácia	Sim	iFood	Concorrência intensa
25/01/2025 21:14:57	Farmácia	Sim	Nenhum aplicativo	Concorrência intensa
26/01/2025 17:44:14	Lanchonete	Sim	Zé Delivery	Problemas logísticos
26/01/2025 19:27:44	Restaurante	Sim	Pedidos Já	Comissões elevadas
29/01/2025 00:30:20	Restaurante	Sim	Aiqfome	Concorrência intensa
30/01/2025 08:28:22	Farmácia	Sim	iFood	Falta de personalização
02/02/2025 10:44:37	Supermercado	Não	Aiqfome	Concorrência intensa
02/02/2025 12:11:52	Restaurante	Não	Zé Delivery	Falta de personalização
03/02/2025 00:09:53	Lanchonete	Sim	Outros	Falta de personalização
04/02/2025 01:40:03	Outros	Não	Nenhum aplicativo	Concorrência intensa
05/02/2025 08:31:42	Mercado	Sim	Uber Eats	Comissões elevadas
08/02/2025 06:29:09	Outros	Sim	Aiqfome	Concorrência intensa
12/02/2025 19:06:41	Lanchonete	Sim	iFood	Comissões elevadas
18/02/2025 17:29:48	Outros	Sim	Pedidos Já	Comissões elevadas
22/02/2025 17:13:58	Lanchonete	Não	Zé Delivery	Problemas logísticos
23/02/2025 02:17:43	Farmácia	Sim	Zé Delivery	Concorrência intensa
23/02/2025 16:45:14	Supermercado	Sim	Zé Delivery	Problemas logísticos
23/02/2025 23:02:29	Restaurante	Sim	Zé Delivery	Reputação online
26/02/2025 12:21:39	Outros	Sim	Uber Eats	Dependência de plataformas
27/02/2025 08:23:53	Mercado	Não	AnotaAi	Falta de personalização
01/03/2025 22:48:38	Outros	Não	AnotaAi	Falta de personalização
05/03/2025 05:38:20	Lanchonete	Não	iFood	Concorrência intensa
06/03/2025 22:43:31	Restaurante	Sim	iFood	Comissões elevadas
07/03/2025 00:46:02	Supermercado	Não	Pedidos Já	Concorrência intensa
08/03/2025 01:33:46	Supermercado	Não	Uber Eats	Comissões elevadas
10/03/2025 18:01:17	Restaurante	Sim	AnotaAi	Dependência de plataformas
12/03/2025 02:41:44	Supermercado	Sim	Aiqfome	Dependência de plataformas
13/03/2025 17:19:44	Outros	Sim	Zé Delivery	Dependência de plataformas
15/03/2025 02:52:17	Outros	Não	Uber Eats	Dependência de plataformas

Você utiliza algum sistema de gestão de pedidos? Se sim, qual?	Quais são os principais horários de pico para os pe
AnotaAi	Tarde
Não utiliza	Tarde
AnotaAi	Entre 20h e 22h
Não utiliza	Manhã
Sistema próprio	Manhã
Sistema próprio	Tarde
ERP externo	Entre 20h e 22h
Planilha Excel	Manhã
ERP externo	Entre 20h e 22h
Planilha Excel	Entre 20h e 22h
IFood Gestão	Após 22h
Planilha Excel	Entre 20h e 22h
AnotaAi	Entre 18h e 20h
Planilha Excel	Entre 20h e 22h
ERP externo	Após 22h
ERP externo	Tarde
Sistema próprio	Manhã
Planilha Excel	Após 22h
Sistema próprio	Entre 20h e 22h
ERP externo	Manhã
IFood Gestão	Entre 18h e 20h
Não utiliza	Tarde
AnotaAi	Tarde
Não utiliza	Após 22h
IFood Gestão	Entre 18h e 20h
IFood Gestão	Entre 18h e 20h
Planilha Excel	Manhã
Sistema próprio	Após 22h
IFood Gestão	Entre 18h e 20h
AnotaAi	Entre 18h e 20h

Na sua opinião, pagar pelas taxas de marketing cobradas pelos aplicativos de delivery impactam no seu negócio?	Você acha que o valor das taxas para o uso do aplicativo de delivery é elevado?
Neutro	Sim
Impacta negativamente	Sim
Neutro	Sim
Impacta negativamente	Sim
Neutro	Sim
Impacta positivamente	Sim
Impacta negativamente	Sim
Impacta positivamente	Sim
Impacta negativamente	Sim
Neutro	Não
Impacta positivamente	Sim
Neutro	Sim
Neutro	Sim
Impacta positivamente	Sim
Impacta negativamente	Não
Impacta negativamente	Sim
Impacta positivamente	Sim
Neutro	Sim
Neutro	Sim
Impacta negativamente	Sim
Impacta negativamente	Não
Neutro	Não
Impacta positivamente	Sim
Impacta negativamente	Sim
Impacta positivamente	Não
Impacta negativamente	Sim
Impacta positivamente	Sim
Neutro	Sim
Neutro	Sim
Impacta positivamente	Sim

Qual você consideraria um valor justo para a taxa de comissão cobrada pelos aplicativos de delivery sobre o valor do pedido?

10% a 15%
20% a 25%
15% a 20%
20% a 25%
10% a 15%
20% a 25%
20% a 25%
10% a 15%
15% a 20%
Mais de 25%
Menos de 10%
10% a 15%
Menos de 10%
10% a 15%
Menos de 10%
Menos de 10%
10% a 15%
10% a 15%
15% a 20%
20% a 25%
Menos de 10%
10% a 15%
10% a 15%
10% a 15%
15% a 20%
10% a 15%
20% a 25%
Menos de 10%
Menos de 10%
10% a 15%

Quais incentivos fariam seu negócio priorizar o uso de aplicativo de delivery? (Selecione uma ou mais opções)
Taxas menores
Taxas menores
Programas de fidelidade
Taxas menores
Maior visibilidade
Suporte personalizado
Suporte personalizado
Promoções/descontos
Suporte personalizado
Promoções/descontos
Promoções/descontos
Melhor controle de entregas
Suporte personalizado
Maior visibilidade
Maior visibilidade
Programas de fidelidade
Suporte personalizado
Promoções/descontos
Taxas menores
Melhor controle de entregas
Taxas menores
Programas de fidelidade
Melhor controle de entregas
Suporte personalizado
Promoções/descontos
Promoções/descontos
Suporte personalizado
Melhor controle de entregas
Promoções/descontos
Taxas menores

Que tipo de informações sobre os pedidos você gostaria de acessar facilmente no aplicativo de delivery? (Selecione uma ou mais opções)	Como você avalia os aplicativos de delivery? 1 - Muito ruim e 5 - Muito bom
Status em tempo real	3
Comissões e taxas por pedido	5
Histórico de pedidos	5
Dados do cliente	4
Detalhamento do pedido	5
Detalhamento do pedido	4
Dados do cliente	5
Status em tempo real	1
Tempo de entrega	1
Comissões e taxas por pedido	4
Histórico de pedidos	2
Histórico de pedidos	5
Status em tempo real	5
Comissões e taxas por pedido	4
Comissões e taxas por pedido	5
Detalhamento do pedido	3
Comissões e taxas por pedido	5
Status em tempo real	4
Comissões e taxas por pedido	1
Status em tempo real	3
Detalhamento do pedido	2
Histórico de pedidos	4
Comissões e taxas por pedido	3
Histórico de pedidos	5
Tempo de entrega	3
Comissões e taxas por pedido	5
Detalhamento do pedido	1
Comissões e taxas por pedido	2
Dados do cliente	3
Dados do cliente	3

Cite alguma experiência positiva ou negativa que influenciou na sua avaliação:
Melhorar taxas
Cientes reclamam da taxa de entrega
Melhorar taxas
Plataforma muito estável
Plataforma muito estável
Suporte poderia ser mais rápido
Melhorar taxas
Plataforma muito estável
Melhorar taxas
Plataforma muito estável
Suporte poderia ser mais rápido
Cientes reclamam da taxa de entrega
Melhorar taxas
Suporte poderia ser mais rápido
Melhorar taxas
Melhorar taxas
Cientes reclamam da taxa de entrega
Plataforma muito estável
Cientes reclamam da taxa de entrega
Plataforma muito estável
Cientes reclamam da taxa de entrega
Suporte poderia ser mais rápido
Suporte poderia ser mais rápido
Melhorar taxas
Plataforma muito estável
Cientes reclamam da taxa de entrega
Suporte poderia ser mais rápido

## APÊNDICE C – Respostas Entregadores

Carimbo de data/hora	Qual sua faixa etária?	Há quanto tempo trabalha como entregador?	Qual veículo você utiliza para fazer entregas?	Por qual motivo você decidiu trabalhar como entregador?	Este trabalho é sua principal fonte de renda?	Quais são os desafios enfrentados por ser um entregador?
17/01/2025 11:57:51	37-46	1-2 anos	Moto	Autonomia	Sim	Dependência dos algoritmos
17/01/2025 19:31:18	18-26	Menos de 1 ano	Bicicleta	Desemprego	Sim	Dependência dos algoritmos
19/01/2025 22:30:32	27-36	3 anos ou mais	Moto	Autonomia	Não	Baixa remuneração
21/01/2025 01:25:34	27-36	3 anos ou mais	Moto	Facilidade	Sim	Concorrência
24/01/2025 08:09:32	27-36	1-2 anos	Moto	Autonomia	Sim	Concorrência
26/01/2025 10:33:53	27-36	3 anos ou mais	Moto	Facilidade	Sim	Falta de direitos trabalhistas
27/01/2025 12:22:49	18-26	Menos de 1 ano	Moto	Facilidade	Sim	Dependência dos algoritmos
27/01/2025 20:54:09	27-36	Menos de 1 ano	Moto	Facilidade	Sim	Dependência dos algoritmos
28/01/2025 13:51:31	37-46	3 anos ou mais	Bicicleta	Autonomia	Sim	Baixa remuneração
29/01/2025 03:10:31	18-26	1-2 anos	Moto	Facilidade	Não	Dependência dos algoritmos
29/01/2025 16:01:17	37-46	1-2 anos	Bicicleta	Renda extra	Sim	Dependência dos algoritmos
29/01/2025 17:10:57	27-36	1-2 anos	Bicicleta	Facilidade	Sim	Dependência dos algoritmos
03/02/2025 19:06:22	27-36	3 anos ou mais	Moto	Negócio próprio	Sim	Dependência dos algoritmos
04/02/2025 15:35:40	27-36	Menos de 1 ano	Moto	Autonomia	Não	Falta de direitos trabalhistas
05/02/2025 18:41:13	18-26	Menos de 1 ano	Moto	Desemprego	Sim	Concorrência
06/02/2025 17:44:56	37-46	1-2 anos	Bicicleta	Facilidade	Sim	Dependência dos algoritmos
06/02/2025 19:42:41	27-36	3 anos ou mais	Bicicleta	Renda extra	Sim	Baixa remuneração
08/02/2025 21:46:21	18-26	Menos de 1 ano	Bicicleta	Facilidade	Sim	Concorrência
09/02/2025 21:51:19	37-46	Menos de 1 ano	Moto	Renda extra	Sim	Baixa remuneração
13/02/2025 06:40:40	27-36	1-2 anos	Moto	Autonomia	Não	Baixa remuneração
15/02/2025 02:24:17	37-46	1-2 anos	Moto	Desemprego	Não	Baixa remuneração
21/02/2025 03:59:34	18-26	1-2 anos	Bicicleta	Facilidade	Sim	Falta de direitos trabalhistas
22/02/2025 19:37:59	18-26	Menos de 1 ano	Bicicleta	Autonomia	Sim	Concorrência
23/02/2025 16:06:30	27-36	Menos de 1 ano	Moto	Autonomia	Sim	Concorrência
23/02/2025 21:47:47	18-26	3 anos ou mais	Moto	Negócio próprio	Sim	Concorrência
24/02/2025 15:59:15	37-46	3 anos ou mais	Moto	Negócio próprio	Sim	Dependência dos algoritmos
26/02/2025 21:41:43	18-26	3 anos ou mais	Moto	Negócio próprio	Sim	Dependência dos algoritmos
27/02/2025 06:36:58	27-36	Menos de 1 ano	Bicicleta	Renda extra	Não	Baixa remuneração
27/02/2025 10:20:30	18-26	1-2 anos	Moto	Autonomia	Sim	Baixa remuneração
01/03/2025 06:22:17	37-46	Menos de 1 ano	Moto	Autonomia	Não	Falta de direitos trabalhistas
01/03/2025 16:25:47	18-26	1-2 anos	Bicicleta	Renda extra	Sim	Concorrência
02/03/2025 04:59:01	37-46	3 anos ou mais	Moto	Autonomia	Não	Baixa remuneração
02/03/2025 14:46:22	27-36	3 anos ou mais	Moto	Renda extra	Não	Falta de direitos trabalhistas
04/03/2025 00:44:46	37-46	1-2 anos	Moto	Facilidade	Não	Dependência dos algoritmos
05/03/2025 02:20:32	18-26	1-2 anos	Bicicleta	Facilidade	Não	Dependência dos algoritmos
05/03/2025 09:24:08	27-36	3 anos ou mais	Bicicleta	Renda extra	Não	Falta de direitos trabalhistas
06/03/2025 09:46:49	37-46	1-2 anos	Moto	Autonomia	Não	Baixa remuneração
09/03/2025 14:16:48	18-26	Menos de 1 ano	Moto	Facilidade	Sim	Baixa remuneração
15/03/2025 12:21:11	27-36	Menos de 1 ano	Moto	Autonomia	Não	Baixa remuneração
17/03/2025 04:43:21	18-26	Menos de 1 ano	Moto	Negócio próprio	Não	Falta de direitos trabalhistas

Quais são os desafios enfrentados durante a entrega?	Você prefere definir sua sequência de entregas ou?	Como você prioriza as entregas?	Como você gerencia o tempo de entregas?	Quais funcionalidades ajudariam a gerenciar melhor?	Na sua opinião, qual seria a remuneração mínima?
Tempo de espera	Ordem do app	Não prioriza	Maior rota com várias entregas	Integração entre apps	Acima de R\$15,00
Falta de respeito	Sequência própria	Distância	Agilidade	Integração entre apps	R\$10,00–R\$15,00
Clima	Sequência própria	Distância	Agilidade	Personalização	R\$10,00–R\$15,00
Tempo de espera	Sequência própria	Distância	Menor distância	Personalização	Acima de R\$15,00
Tempo de espera	Sequência própria	Distância	Não gerencia	Integração entre apps	R\$6,50–R\$10,00
Risco de violência	Sequência própria	Não prioriza	Agilidade	Personalização	Acima de R\$15,00
Tempo de espera	Sequência própria	Rota	Agilidade	Otimização de rotas	R\$10,00–R\$15,00
Tempo de espera	Sequência própria	Rota	Maior rota com várias entregas	Paradas inteligentes	Acima de R\$15,00
Manutenção do veículo	Ordem do app	Distância	Maior rota com várias entregas	Integração entre apps	R\$10,00–R\$15,00
Tempo de espera	Sequência própria	Rota	Menor distância	Paradas inteligentes	Acima de R\$15,00
Exaustão	Ordem do app	Valor	Agilidade	Tempo de espera remunerado	R\$10,00–R\$15,00
Tempo de espera	Sequência própria	Valor	Agilidade	Otimização de rotas	Acima de R\$15,00
Exaustão	Ordem do app	Distância	Menor distância	Tempo de espera remunerado	R\$6,50–R\$10,00
Exaustão	Ordem do app	Rota	Não gerencia	Otimização de rotas	R\$10,00–R\$15,00
Manutenção do veículo	Sequência própria	Rota	Não gerencia	Integração entre apps	R\$6,50–R\$10,00
Bloqueios injustos	Sequência própria	Distância	Agilidade	Tempo de espera remunerado	R\$10,00–R\$15,00
Risco de violência	Sequência própria	Rota	Menor distância	Personalização	R\$10,00–R\$15,00
Manutenção do veículo	Sequência própria	Rota	Maior rota com várias entregas	Paradas inteligentes	R\$6,50–R\$10,00
Clima	Sequência própria	Não prioriza	Menor distância	Tempo de espera remunerado	R\$6,50–R\$10,00
Exaustão	Ordem do app	Não prioriza	Maior rota com várias entregas	Integração entre apps	R\$10,00–R\$15,00
Exaustão	Sequência própria	Distância	Maior rota com várias entregas	Integração entre apps	Acima de R\$15,00
Tempo de espera	Ordem do app	Distância	Não gerencia	Integração entre apps	Acima de R\$15,00
Risco de violência	Ordem do app	Valor	Não gerencia	Personalização	R\$10,00–R\$15,00
Exaustão	Ordem do app	Não prioriza	Agilidade	Integração entre apps	Acima de R\$15,00
Risco de violência	Ordem do app	Distância	Maior rota com várias entregas	Personalização	Acima de R\$15,00
Bloqueios injustos	Sequência própria	Valor	Não gerencia	Paradas inteligentes	Acima de R\$15,00
Manutenção do veículo	Ordem do app	Não prioriza	Não gerencia	Personalização	Acima de R\$15,00
Clima	Ordem do app	Rota	Maior rota com várias entregas	Personalização	R\$6,50–R\$10,00
Tempo de espera	Sequência própria	Distância	Agilidade	Tempo de espera remunerado	R\$10,00–R\$15,00
Manutenção do veículo	Sequência própria	Não prioriza	Agilidade	Tempo de espera remunerado	R\$10,00–R\$15,00
Tempo de espera	Ordem do app	Distância	Menor distância	Integração entre apps	R\$10,00–R\$15,00
Exaustão	Ordem do app	Valor	Não gerencia	Paradas inteligentes	R\$6,50–R\$10,00
Exaustão	Ordem do app	Distância	Menor distância	Integração entre apps	R\$10,00–R\$15,00
Exaustão	Ordem do app	Valor	Não gerencia	Integração entre apps	Acima de R\$15,00
Clima	Sequência própria	Distância	Não gerencia	Integração entre apps	Acima de R\$15,00
Bloqueios injustos	Sequência própria	Não prioriza	Maior rota com várias entregas	Tempo de espera remunerado	Acima de R\$15,00
Bloqueios injustos	Sequência própria	Distância	Agilidade	Personalização	R\$6,50–R\$10,00
Clima	Sequência própria	Não prioriza	Menor distância	Personalização	Acima de R\$15,00
Clima	Ordem do app	Distância	Menor distância	Tempo de espera remunerado	R\$10,00–R\$15,00
Tempo de espera	Ordem do app	Valor	Não gerencia	Paradas inteligentes	R\$10,00–R\$15,00

Na sua opinião, qual seria a remuneração mínima j	Na sua opinião, quais bônus ou incentivos seriam r	Quais métodos de pagamento você prefere receber	Que tipo de informações são essenciais em tempo	O que tornaria o processo de recebimento e finaliza	Você já enfrentou situações de insegurança duran
R\$2,50-R\$5,00	Bonificação por distância ou tempo de espera	Dinheiro físico	Mapas detalhados	Informações sobre endereço	Não
R\$1,50-R\$2,50	Pagamento integral por pedido	Dinheiro físico	Informações sobre o pedido	Informações sobre endereço	Sim
R\$1,50-R\$2,50	Limitação de rotas de bicicleta	Dinheiro físico	Tempo de espera	Sugestões de locais estratégicos	Não
R\$1,50-R\$2,50	Reajuste por entrega/quilômetro	PIX	Mapas detalhados	Informações sobre endereço	Não
R\$2,50-R\$5,00	Limitação de rotas de bicicleta	PIX	Contato com o cliente	Contato rápido com o cliente	Não
R\$2,50-R\$5,00	Bonificação por distância ou tempo de espera	PIX	Informações sobre o pedido	Sugestões de locais estratégicos	Sim
Acima de R\$5,00	Limitação de rotas de bicicleta	Transferência bancária	Nível de demanda	Contato rápido com o cliente	Não
R\$2,50-R\$5,00	Bonificação por distância ou tempo de espera	PIX	Tempo de espera	Contato rápido com o cliente	Sim
R\$2,50-R\$5,00	Bonificação por distância ou tempo de espera	Transferência bancária	Mapas detalhados	Informações sobre endereço	Sim
R\$2,50-R\$5,00	Reajuste por entrega/quilômetro	PIX	Tempo de espera	Contato rápido com o cliente	Não
R\$2,50-R\$5,00	Limitação de rotas de bicicleta	Dinheiro físico	Tempo de espera	Informações sobre endereço	Sim
Acima de R\$5,00	Pagamento integral por pedido	PIX	Contato com o cliente	Contato rápido com o cliente	Sim
R\$2,50-R\$5,00	Bonificação por distância ou tempo de espera	Dinheiro físico	Informações sobre o pedido	Sugestões de locais estratégicos	Não
R\$2,50-R\$5,00	Bonificação por distância ou tempo de espera	Transferência bancária	Nível de demanda	Informações sobre endereço	Sim
R\$2,50-R\$5,00	Pagamento integral por pedido	Transferência bancária	Informações sobre o pedido	Contato rápido com o cliente	Sim
Acima de R\$5,00	Limitação de rotas de bicicleta	Dinheiro físico	Contato com o cliente	Informações sobre endereço	Sim
Acima de R\$5,00	Bonificação por distância ou tempo de espera	PIX	Mapas detalhados	Contato rápido com o cliente	Não
Acima de R\$5,00	Bonificação por distância ou tempo de espera	Transferência bancária	Informações sobre o pedido	Contato rápido com o cliente	Não
R\$2,50-R\$5,00	Pagamento integral por pedido	Dinheiro físico	Mapas detalhados	Informações sobre endereço	Não
R\$2,50-R\$5,00	Limitação de rotas de bicicleta	PIX	Tempo de espera	Sugestões de locais estratégicos	Sim
R\$1,50-R\$2,50	Limitação de rotas de bicicleta	PIX	Contato com o cliente	Contato rápido com o cliente	Não
Acima de R\$5,00	Pagamento integral por pedido	Transferência bancária	Mapas detalhados	Informações sobre endereço	Não
R\$1,50-R\$2,50	Bonificação por distância ou tempo de espera	PIX	Mapas detalhados	Contato rápido com o cliente	Sim
R\$2,50-R\$5,00	Bonificação por distância ou tempo de espera	Dinheiro físico	Mapas detalhados	Sugestões de locais estratégicos	Não
Acima de R\$5,00	Limitação de rotas de bicicleta	PIX	Informações sobre o pedido	Contato rápido com o cliente	Sim
R\$1,50-R\$2,50	Reajuste por entrega/quilômetro	Dinheiro físico	Mapas detalhados	Informações sobre endereço	Sim
Acima de R\$5,00	Pagamento integral por pedido	Transferência bancária	Mapas detalhados	Contato rápido com o cliente	Sim
R\$2,50-R\$5,00	Limitação de rotas de bicicleta	Transferência bancária	Mapas detalhados	Informações sobre endereço	Sim
Acima de R\$5,00	Limitação de rotas de bicicleta	Dinheiro físico	Contato com o cliente	Contato rápido com o cliente	Sim
R\$1,50-R\$2,50	Bonificação por distância ou tempo de espera	Dinheiro físico	Nível de demanda	Contato rápido com o cliente	Não
R\$1,50-R\$2,50	Bonificação por distância ou tempo de espera	Dinheiro físico	Informações sobre o pedido	Informações sobre endereço	Não
R\$2,50-R\$5,00	Pagamento integral por pedido	Dinheiro físico	Mapas detalhados	Informações sobre endereço	Não
R\$1,50-R\$2,50	Reajuste por entrega/quilômetro	Transferência bancária	Informações sobre o pedido	Sugestões de locais estratégicos	Sim
Acima de R\$5,00	Pagamento integral por pedido	Transferência bancária	Contato com o cliente	Sugestões de locais estratégicos	Não
Acima de R\$5,00	Limitação de rotas de bicicleta	PIX	Contato com o cliente	Sugestões de locais estratégicos	Sim
Acima de R\$5,00	Limitação de rotas de bicicleta	Transferência bancária	Nível de demanda	Informações sobre endereço	Sim
R\$2,50-R\$5,00	Pagamento integral por pedido	Dinheiro físico	Mapas detalhados	Contato rápido com o cliente	Não
Acima de R\$5,00	Pagamento integral por pedido	Transferência bancária	Informações sobre o pedido	Sugestões de locais estratégicos	Sim
Acima de R\$5,00	Bonificação por distância ou tempo de espera	Transferência bancária	Contato com o cliente	Contato rápido com o cliente	Não
R\$2,50-R\$5,00	Pagamento integral por pedido	PIX	Nível de demanda	Contato rápido com o cliente	Não

Como o aplicativo pode ajudar a prevenir ou lidar c	Que tipo de suporte você esperaria dos aplicativos? Por se tratar de um formulário para fins acadêmicos
Relatório de áreas perigosas	Chat 24h
Botão de emergência	Chat 24h
Alertas sobre acidentes	Auxílio na manutenção do veículo
Botão de emergência	Chat 24h
Relatório de áreas perigosas	Suporte para problemas, acidentes ou bloqueios
Alertas sobre acidentes	Auxílio na manutenção do veículo
Alertas sobre acidentes	Auxílio na manutenção do veículo
Alertas sobre acidentes	Suporte para problemas, acidentes ou bloqueios
Alertas sobre acidentes	Suporte para problemas, acidentes ou bloqueios
Relatório de áreas perigosas	Chat 24h
Avaliação de clientes	Chat 24h
Alertas sobre acidentes	Proteção contra penalizações
Botão de emergência	Auxílio na manutenção do veículo
Avaliação de clientes	Chat 24h
Alertas sobre acidentes	Proteção contra penalizações
Relatório de áreas perigosas	Suporte para problemas, acidentes ou bloqueios
Botão de emergência	Chat 24h
Avaliação de clientes	Suporte para problemas, acidentes ou bloqueios
Avaliação de clientes	Proteção contra penalizações
Avaliação de clientes	Auxílio na manutenção do veículo
Avaliação de clientes	Proteção contra penalizações
Relatório de áreas perigosas	Proteção contra penalizações
Avaliação de clientes	Auxílio na manutenção do veículo
Avaliação de clientes	Chat 24h
Alertas sobre acidentes	Proteção contra penalizações
Botão de emergência	Proteção contra penalizações
Alertas sobre acidentes	Chat 24h
Relatório de áreas perigosas	Auxílio na manutenção do veículo
Alertas sobre acidentes	Proteção contra penalizações
Botão de emergência	Suporte para problemas, acidentes ou bloqueios
Botão de emergência	Chat 24h
Relatório de áreas perigosas	Auxílio na manutenção do veículo
Avaliação de clientes	Suporte para problemas, acidentes ou bloqueios
Botão de emergência	Chat 24h
Botão de emergência	Proteção contra penalizações
Relatório de áreas perigosas	Chat 24h
Alertas sobre acidentes	Auxílio na manutenção do veículo
Botão de emergência	Proteção contra penalizações
Botão de emergência	Suporte para problemas, acidentes ou bloqueios
Avaliação de clientes	Chat 24h