

INSTITUTO FEDERAL DO TOCANTINS - CAMPUS COLINAS DO TOCANTINS
TECNOLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
JOSÉ VÍCTOR MODESTO DE OLIVEIRA

3Z: DESENVOLVIMENTO DE UMA PLATAFORMA INTEGRADA DE
INTERAÇÃO SOCIAL E MERCADO DIGITAL

COLINAS DO TOCANTINS - TO
2026

JOSÉ VICTOR MODESTO DE OLIVEIRA

**3Z: DESENVOLVIMENTO DE UMA PLATAFORMA INTEGRADA DE
INTERAÇÃO SOCIAL E MERCADO DIGITAL**

Trabalho de curso apresentado ao curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal do Tocantins – Campus Colinas do Tocantins, como requisito parcial para a obtenção do título de bacharel em Tecnologia em Análise e Desenvolvimento de Sistemas, sob orientação do Prof. Dr. Ronneesley Moura Teles.

COLINAS DO TOCANTINS - TO

2026

Dados Internacionais de Catalogação na Publicação (CIP)
Bibliotecas do Instituto Federal do Tocantins

0483 Oliveira, José Victor Modesto

3Z: Desenvolvimento de uma plataforma integrada de interação social e mercado digital. / José Victor Modesto Oliveira. – Colinas do Tocantins, TO, 2026.
75 f. : il. color.

Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia do Tocantins, Campus Colinas do Tocantins, Colinas do Tocantins, TO, 2026.

Orientador: Dr. Ronneesley Teles

1. Marketing, comunicação e gestão. 2. Mercado digital e local. 3. Django e Next. I. Teles, Ronneesley. II. Título.

CDD 004.2

A reprodução total ou parcial, de qualquer forma ou por qualquer meio, deste documento é autorizada para fins de estudo e pesquisa, desde que citada a fonte.

Elaborado pelo sistema de geração automática de fichas catalográficas do IFTO com os dados fornecidos pelo(a) autor(ã).

JOSÉ VICTOR MODESTO DE OLIVEIRA

**3Z: DESENVOLVIMENTO DE UMA PLATAFORMA INTEGRADA DE
INTERAÇÃO SOCIAL E MERCADO DIGITAL**

Trabalho de curso apresentado ao curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal do Tocantins – Campus Colinas do Tocantins, como requisito parcial para a obtenção do título de bacharel em Tecnologia em Análise e Desenvolvimento de Sistemas, sob orientação do Prof. Dr. Ronneesley Moura Teles.

Banca Examinadora

Documento assinado digitalmente
 **RONNEESLEY MOURA TELES**
Data: 24/06/2026 17:58:34-0300
Verifique em <https://validar.iti.gov.br>

Dr. Ronneesley Moura Teles
Instituto Federal do Tocantins Campus Colinas do Tocantins

Documento assinado digitalmente
 **LAERCIO PONTIN JUNIOR**
Data: 24/06/2026 09:35:06-0300
Verifique em <https://validar.iti.gov.br>

Me. Laercio Pontin Junior
Instituto Federal do Tocantins Campus Colinas do Tocantins

Documento assinado digitalmente
 **GILVAN VIEIRA MOURA**
Data: 25/06/2026 08:11:30-0300
Verifique em <https://validar.iti.gov.br>

Me. Gilvan Vieira Moura
Instituto Federal do Tocantins Campus Araguaína

Aprovada em 02 de junho de 2026

Dedico este trabalho a duas pessoas muito especiais: à minha mãe, uma mulher forte e guerreira, que sempre me apoiou e acreditou em mim nos melhores e piores momentos da minha vida, e à minha doce sobrinha Helena, uma pequena princesa que trouxe alegria e motivação a minha jornada.

AGRADECIMENTOS

Meus agradecimentos a todos os professores do IFTO Campus Colinas do Tocantins, por contribuírem com excelência para a minha formação. Em especial, ao meu orientador Ronneesley Moura Teles, que durante esses últimos anos, não foi apenas um professor, mas tornou-se um mentor em minha jornada acadêmica e profissional.

“Tendemos a esquecer que nenhum computador
jamais fará uma nova pergunta.”

Grace Hopper

RESUMO

A consolidação da tecnologia de software transformou-se em uma ferramenta estratégica essencial para o *marketing*, comunicação e gestão de negócios. No entanto, microempresários locais e profissionais autônomos frequentemente enfrentam barreiras técnicas e financeiras para acessar e manter múltiplas ferramentas digitais de forma contínua, o que gera uma desvantagem competitiva e acentua o cenário de exclusão digital. Diante dessa problemática, este trabalho apresenta o desenvolvimento do sistema 3Z, um protótipo de plataforma web integrada projetada para fortalecer a economia regional e impulsionar o comércio local por meio de ferramentas centralizadas. O objetivo geral consistiu em projetar e implementar um ambiente digital unificado que integra a dinâmica interativa das redes sociais, a visibilidade dos *marketplaces* e a robustez de sistemas de gestão empresarial simplificados, atuando como um facilitador comercial para conectar o comércio e o público local. A metodologia adotada fundamentou-se em um modelo de desenvolvimento incremental, apoiado em revisão da literatura científica e em uma pesquisa de campo estruturada com microempreendedores locais, cujos dados coletados justificaram a concepção de uma alternativa unificada. A arquitetura tecnológica do projeto foi estruturada com a divisão entre *back-end* e *front-end* integrados via *Application Programming Interface* (API) do tipo *Representational State Transfer* (REST), utilizando a linguagem Python com o Django REST *framework* para as regras de negócio e segurança, e JavaScript/TypeScript com React e Next.js na concepção de interfaces fluidas e reativas. Conclui-se que o sistema 3Z cumpriu com êxito os objetivos propostos ao fornecer uma solução tecnologicamente viável e de boa usabilidade diante do cenário de exclusão digital e gargalos financeiros, concebendo um protótipo eficiente que serve como base sólida para um projeto real em ambiente de produção.

Palavras-chave: Marketing. Comunicação. Gestão. Django. Next.

ABSTRACT

The consolidation of software technology has transformed into an essential strategic tool for marketing, communication, and business management. However, local micro-entrepreneurs and self-employed professionals frequently face technical and financial barriers to accessing and maintaining multiple digital tools continuously, which generates a competitive disadvantage and accentuates the digital exclusion scenario. Faced with this problem, this work presents the development of the 3Z system, a prototype of an integrated web platform designed to strengthen the regional economy and boost local commerce through centralized and free tools. The general objective consisted of designing and implementing a unified digital environment that integrates the interactive dynamics of social networks, the visibility of marketplaces, and the robustness of simplified business management systems (ERP), acting as a commercial facilitator to connect businesses and the local public. The adopted methodology was based on an incremental development model, supported by a literature review and a structured field research with local micro-entrepreneurs, whose collected data justified the conception of a unified alternative. The project's technological architecture was structured with the split between back-end and front-end via Representational State Transfer (REST) Application Programming Interface (API), using the Python language with the Django REST Framework for business rules and security, and JavaScript/TypeScript with React and Next.js in the design of fluid and reactive interfaces. It is concluded that the 3Z system successfully fulfilled the proposed objectives by providing a technologically viable and highly usable solution to the digital exclusion scenario and financial bottlenecks, conceiving an efficient prototype that serves as a solid foundation for a real project in a production environment. **Keywords:** Marketing. Communication. Management. Django. Next.

LISTA DE ILUSTRAÇÕES

Figura 1 – Digitalização nas micro e pequenas empresas	16
Figura 2 – Digitalização da população brasileira referente ao mercado digital	17
Figura 3 – Arquitetura MVC	21
Figura 4 – Ranking TIOBE de linguagens de programação 2026	23
Figura 5 – Página inicial do site OLX.	30
Figura 6 – Página de marketplace do Facebook.	31
Figura 7 – Diagrama de componentes do 3Z	39
Figura 8 – Protótipo da tela de cadastro e listagem dos estados	40
Figura 9 – Protótipo da página inicial	41
Figura 10 – Protótipo de perfil	42
Figura 11 – Protótipo da tela de cadastro de postagem empresarial.	42
Figura 12 – Diagrama Entidade Relacionamento do sistema	45
Figura 13 – Diagrama de casos de uso sistema	47
Figura 14 – Tela inicial	49
Figura 15 – Tela de exploração	49
Figura 16 – Tela de login	50
Figura 17 – Tela de cadastro de usuário	50
Figura 18 – Tela de manutenção das áreas de atividade	51
Figura 19 – Tela de gerenciamento dos usuários pessoais)	52
Figura 20 – Tela principal feed	53
Figura 21 – Tela de postagem	54
Figura 22 – Tela de listagem de favoritos	55
Figura 23 – Tela de listagem de usuários que estão sendo seguidos	55
Figura 24 – Perfil Público	56
Figura 25 – Catálogo	57
Figura 26 – Tela de informações do usuário	57
Figura 27 – tela de perfil privado	58
Figura 28 – tela de configurações do usuário	59
Figura 29 – Tela de movimentações de caixa	60
Figura 30 – Tela de cadastro das entradas	61
Figura 31 – Relatório de entrada	61

Figura 32 – Tela de cadastro de novas postagens empresariais	62
Figura 33 – Tela de estoque de produtos	63
Figura 34 – Informações sobre o produto empresarial	63
Figura 35 – Relatórios de caixas	64
Figura 36 – Gráfico de pizza das respostas da pergunta: Qual é o seu principal canal de divulgação online de produtos ou serviços?	66
Figura 37 – Gráfico de pizza das respostas da pergunta: Qual o principal meio de contato utilizado com seus clientes?	66
Figura 38 – Gráfico de pizza das respostas da pergunta: Como é feita a sua sua gestão financeira?	67
Figura 39 – Gráfico de pizza das respostas da pergunta: Considerando o uso de ferramentas digitais citadas acima para Marketing, Comunicação e Gestão, qual a sua maior dificuldade em operar com pelo menos uma solução em cada um desses três pilares?	68

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Objetivo Geral	15
1.2	Objetivos específicos	15
1.3	Justificativa	16
1.4	Contribuições realizadas	17
1.5	Organização do trabalho	18
2	REVISÃO BIBLIOGRÁFICA	19
2.1	Fundamentação teórica	19
2.1.1	Hypertext Transfer Protocol (HTTP)	19
2.1.2	Model View Controller (MVC)	20
2.1.3	Python	21
2.1.4	Django.PY	23
2.1.5	Django Rest Framework (DRF)	25
2.1.6	JavaScript	26
2.1.7	TypeScript	27
2.1.8	React	27
2.1.9	Next.js	28
2.2	Trabalhos correlatos	30
2.2.1	OLX	30
2.2.2	Facebook Marketplace	31
3	MATERIAIS E MÉTODOS	33
3.1	Materiais	33
3.1.1	Linguagens de programação.	33
3.1.2	Frameworks	34
3.1.3	Outras ferramentas	35
3.1.4	Hardware	35
3.2	Métodos	36
4	PROJETO	38
4.1	Arquitetura do projeto	38
4.2	Protótipos	39
4.3	Diagrama Entidade Relacionamento (DER)	43

4.4 Diagrama de Casos de Uso	46
5 RESULTADOS	48
5.1 Interfaces gráficas de visitantes	48
5.2 Interfaces gráficas de administradores	51
5.3 Interfaces gráficas de usuário pessoais e empresariais	52
5.4 Interfaces gráficas de gestão empresarial	59
6 DISCUSSÃO	65
6.1 Discussão das entrevistas	65
6.2 Discussão dos resultados	68
7 CONSIDERAÇÕES FINAIS	70
7.1 Trabalhos Futuros	70
7.2 Conclusão	72
REFERÊNCIAS	73

1 INTRODUÇÃO

Atualmente, a tecnologia de *software* consolidou-se como uma das principais ferramentas estratégicas para empresas de todos os portes e profissionais autônomos, destacando-se em áreas como *marketing*, comunicação e gestão.

No *marketing* tradicional, a prioridade era atingir metas de vendas de produtos ou serviços. Entretanto, as abordagens contemporâneas passaram a direcionar seus esforços para a compreensão e a satisfação das necessidades dos consumidores. Enquanto o conceito de venda estava centrado na obtenção de lucro por meio da comercialização de produtos e da integração entre vendas e promoção, o *marketing* moderno busca gerar valor para o cliente por meio de estratégias integradas, tendo como resultado a satisfação do consumidor e, consequentemente, a obtenção de lucro (KOTLER; KELLER, 2012).

Para alcançar esse objetivo, o uso de tecnologias digitais, como redes sociais, *market-places*, plataformas de *e-commerce* e *websites* próprios, tem se tornado cada vez mais comum. Essas ferramentas proporcionam maior visibilidade e alcance no ambiente digital, contribuindo para a atração de novos clientes e para o fortalecimento de estratégias de fidelização (TORRES, 2018).

Outro ponto importante é o impacto dessas tecnologias na comunicação, afinal, não adianta um possível cliente conhecer o negócio se não houver meios de contato. Nesse contexto, ferramentas de *chat* em tempo real ganham destaque. Um exemplo claro é o WhatsApp, aplicativo desenvolvido pela Meta, que se consolidou como uma das principais formas de comunicação entre empreendedores e clientes.

No Brasil, a plataforma é amplamente utilizada por empresas em estratégias de vendas, atendimento e relacionamento com o público, uma vez que a troca instantânea de mensagens contribui para fortalecer a comunicação entre consumidores e estabelecimentos comerciais. Segundo dados apresentados pelo Sebrae, 76% dos brasileiros já utilizaram o WhatsApp para interagir com marcas, evidenciando a relevância da plataforma para a comunicação entre empresas e clientes (SEBRAE, 2021).

E uma última área que vale a pena ser mencionada neste trabalho é a gestão financeira, sistemas *Enterprise Resource Planning* (ERP), ou sistemas de planejamento dos recursos empresariais, possuem uma abordagem na qual são reunidas, em uma única plataforma, funcionalidades que abrangem finanças, recursos, vendas, análise de dados, entre outras (SOUSA, 2026).

No contexto da gestão empresarial, esse tipo de sistema tem sido cada vez mais utilizado no mercado com o intuito de automatizar tarefas que antes eram exaustivas e feitas em papel e caneta, além de auxiliar na tomada de decisões importantes, que muitas vezes envolvem informações de meses ou até anos armazenadas em seu banco de dados (SOUSA, 2026).

Com todos esses recursos disponíveis e ofertados no mercado, não somente as empresas, como também as pessoas físicas têm cada vez mais adotado os recursos digitais para realizar compras e buscar serviços *online*. Esse número de pessoas aumentou significativamente após a pandemia da COVID-19, quando, durante o *lockdown*, as pessoas se viram obrigadas a permanecer em casa e recorrer a recursos digitais para tomar decisões de compra mais conscientes e assertivas (SOUSA; DANTAS, 2023).

Tudo isso fez com que o comércio local, que antes dependia da proximidade física com os clientes, passasse a competir em um mercado mais amplo, superando as limitações geográficas. Apesar de toda essa evolução ter trazido diversos benefícios, também trouxe pontos negativos, principalmente para micro e pequenas empresas, que, segundo dados do Sebrae, representam 97% do total de empresas do país e são responsáveis por 26,5% do Produto Interno Bruto (PIB) nacional (SEBRAE, 2025).

Muitas dessas empresas enfrentam dificuldades para se adequarem ao mercado digital e competir por espaço com as grandes, afinal, a maioria ainda depende, em grande parte, do lucro que circula localmente ao seu redor. A falta de recursos técnicos e financeiros é um dos principais desafios enfrentados por esses negócios, pois, apesar de existirem diversas ferramentas disponíveis no mercado, onde o custo unitário da maioria pode ser baixo, ao somar o uso de várias delas, o valor total acaba se tornando muito acima do que a maioria consegue pagar ou manter continuamente (NOVAIS, 2024).

Além disso, cada uma dessas ferramentas exige conhecimento técnico para seu uso adequado, de modo que seja possível obter o melhor aproveitamento, evitando que os gastos superem os lucros do investimento digital. Isso faz com que haja uma dificuldade adicional para a inserção e consolidação dessas empresas no ambiente digital, levando muitos empreendedores a adiarem ou abandonarem esse processo (BARROS, 2023).

Diante desse cenário apresentado, e com base em pesquisas realizadas através de conteúdos acadêmicos, conteúdos digitais e empresas reais da cidade de Colinas do Tocantins - TO, foi observado que com o avanço constante da tecnologia e a crescente digitalização dos processos comerciais, tornou-se essencial o desenvolvimento de sistemas capazes de gerenciar

informações de forma eficiente, segura e escalável. Nesse contexto, plataformas que integram usuários comuns e empresariais em um único ambiente têm ganhado destaque, principalmente por possibilitarem a centralização de dados, automação de processos e melhor experiência do usuário.

1.1 Objetivo Geral

Pretende-se, como objetivo geral deste trabalho, apresentar um protótipo de sistema *web* que contribua para o fortalecimento do mercado local, sendo uma plataforma digital com ferramentas de auxílio para *marketing*, comunicação e gestão financeira, voltada principalmente para micro e pequenas empresas, além de recursos para pessoas que buscam produtos e serviços de forma fácil e rápida em suas respectivas proximidades.

Vale a pena também ser ressaltado que o sistema 3Z, nome dado a plataforma em desenvolvimento, não tem como objetivo ser um *e-commerce* onde clientes realizem compras pelo mesmo, mas sim ser uma plataforma que integra funcionalidades essenciais em um único ambiente digital, atuando como um facilitador na conexão entre o comércio local e o público, sem intermediar transações financeiras ou entregas.

1.2 Objetivos específicos

Para chegar nesse objetivo geral foram definidos os seguintes objetivos específicos:

- Levantar os principais desafios enfrentados por micro e pequenas empresas no contexto do mercado digital local, bem como as necessidades de pessoas ao buscar produtos e serviços próximos;
- Investigar e comparar soluções existentes (redes sociais, marketplaces, sistemas de gestão empresarial e plataformas de comunicação) utilizadas por pequenos empreendimentos;
- Projetar a arquitetura, a prototipagem gráfica e o modelo de dados da aplicação *web*;
- Desenvolver funcionalidades voltadas ao *marketing* digital, como cadastro e divulgação de produtos e serviços, implementar uma forma de comunicação via Whatsapp entre usuários, desenvolver funcionalidades básicas de gestão financeira empresarial, além de garantir a comunicação entre as diferentes funcionalidades da plataforma;

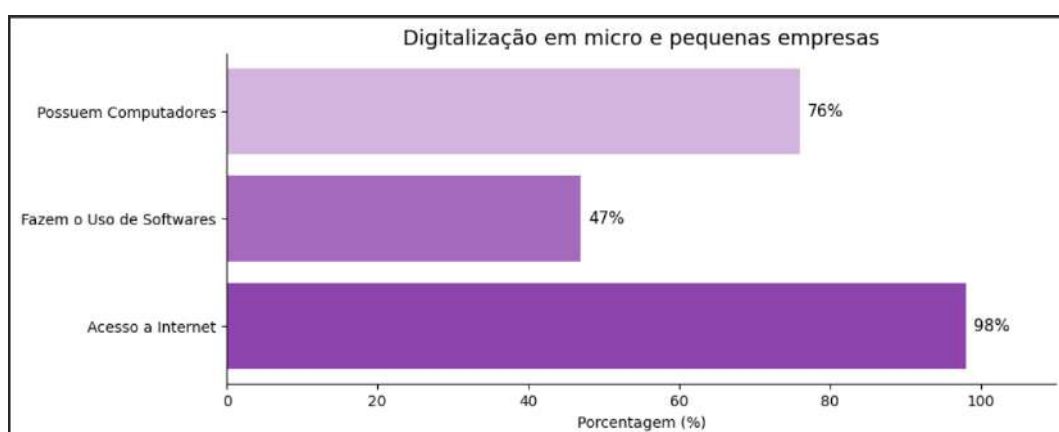
- Avaliar a usabilidade e a eficiência do sistema, propor melhorias e identificar falhas para posterior correção de possíveis erros da aplicação.

1.3 Justificativa

De acordo com dados obtidos por uma pesquisa realizada pela Agência SEBRAE de Notícias (ASN) em 2025, a digitalização em micro e pequenas empresas tem alcançado níveis históricos. Segundo a pesquisa, 76% dos empreendedores contam com computadores em suas atividades, o que representa um crescimento de seis pontos percentuais desde 2022, igualando o pico registrado há 10 anos. O uso de aplicativos, softwares ou programas “integrativos” também cresceu, sendo adotado por 47% dos empreendimentos. Já o acesso à Internet está praticamente universalizado, com 98% dos empreendedores afirmando utilizar conexão própria ou de terceiros como mostra a Figura 1 (Agência Sebrae de Notícias, 2025).

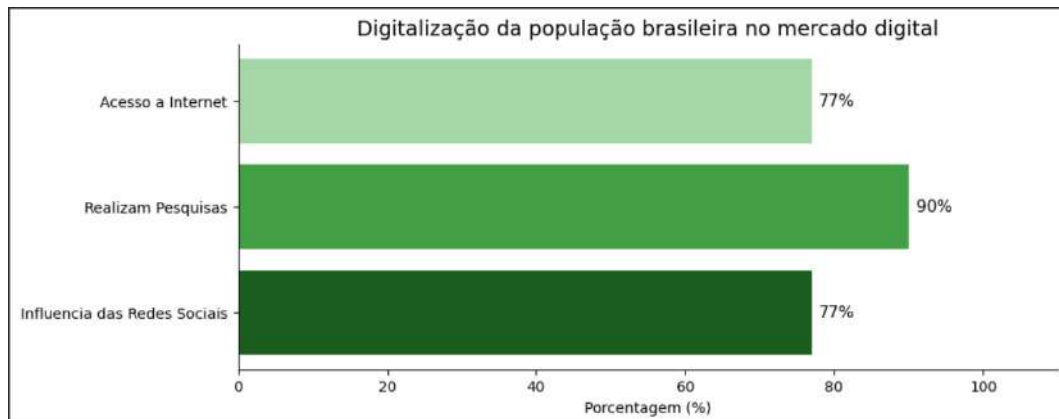
Por outro lado, pesquisas realizadas pelo portal do SEBRAE mostram também que o Brasil possui 165,3 milhões de usuários de Internet própria, o que corresponde a cerca de 77% da população do país. Outros dados coletados mostram que 9 em cada 10 brasileiros realizam pesquisas na Internet antes de fazer uma compra e que as redes sociais influenciam a decisão de compra de 77% das pessoas, a Figura 2 mostra essas informações de forma visual (SEBRAE Santa Catarina, 2022).

Figura 1 – Digitalização nas micro e pequenas empresas



Fonte: Desenvolvido pelo autor com base nos dados do SEBRAE (2026)

Figura 2 – Digitalização da população brasileira referente ao mercado digital



Fonte: Desenvolvido pelo autor com base nos dados do SEBRAE (2026)

Diante disso, este trabalho justifica-se pela crescente importância das tecnologias digitais no contexto empresarial e pela necessidade de adaptação das micro e pequenas empresas às novas dinâmicas do mercado. Apesar da ampla oferta de ferramentas digitais disponíveis, muitos pequenos empreendimentos ainda enfrentam dificuldades para utilizá-las de forma eficiente devido a limitações financeiras e à falta de conhecimento técnico.

Além disso, o aumento do uso de plataformas digitais por consumidores, evidencia a necessidade de soluções que facilitem a conexão entre empresas locais e clientes, permitindo maior visibilidade e acesso a produtos e serviços de forma prática e rápida.

1.4 Contribuições realizadas

O sistema 3Z apresenta diversas características que podem ser destacadas, algumas já encontradas em outras aplicações e outras desenvolvidas especificamente para a plataforma, o que contribui para sua diferenciação. Dentre elas, destacam-se:

- **Estímulo do comércio local através do digital.** Embora existam diversas plataformas voltadas à divulgação de produtos e serviços, a maioria delas busca ampliar o alcance geográfico das publicações, enquanto poucas são direcionadas ao fortalecimento do comércio local. Nesse contexto, o 3Z contribui para a valorização da economia regional ao oferecer maior visibilidade a empreendedores, autônomos e usuários em sua própria região, além de facilitar a busca por produtos e serviços próximos.
- **Custo em relação a ferramentas digitais e tempo.** Existem diversas ferramentas no mercado voltadas à divulgação de conteúdos, comunicação entre usuários, catálogos

online e gestão empresarial. No entanto, muitas delas oferecem apenas parte dessas funcionalidades, exigindo o uso de múltiplas plataformas. O 3Z, por outro lado, busca, como sua essência reunir em um único ambiente, um conjunto abrangente de ferramentas voltadas ao comércio local e regional. Dessa forma, contribui para a redução de custos e tempo ao centralizar recursos essenciais e disponibilizá-los de forma gratuita ou com baixo custo, diminuindo a necessidade de contratação, conhecimento técnico e gerenciamento de várias ferramentas distintas.

- **Valorização da economia regional.** Ao incentivar a conexão entre consumidores e negócios locais, o 3Z contribui para a circulação de renda dentro da própria região, fortalecendo a economia local e incentivando um crescimento sustentável.
- **Redução da dependência de grandes plataformas.** O 3Z oferece uma alternativa mais acessível, simples e direcionada ao contexto local, diminuindo a dependência de *marketplaces* e redes sociais de grande porte, que muitas vezes não priorizam pequenos negócios regionais.

1.5 Organização do trabalho

Este capítulo apresentou o problema de pesquisa, as motivações, os objetivos e as principais contribuições deste trabalho. O Capítulo 2 apresenta os conceitos teóricos, os trabalhos relacionados e as principais tecnologias e *frameworks* utilizados no desenvolvimento da solução proposta. O Capítulo 3 descreve os materiais e métodos adotados durante a pesquisa e no processo de desenvolvimento do sistema.

No Capítulo 4 é apresentado o projeto de *software* do sistema 3Z, incluindo a arquitetura da aplicação, os protótipos de interface, a modelagem do banco de dados e os diagramas responsáveis por representar a estrutura e o funcionamento do sistema. O Capítulo 5 apresenta os resultados obtidos com o desenvolvimento da plataforma, bem como uma análise das funcionalidades implementadas e das contribuições da solução proposta. Por fim, o Capítulo 6 apresenta as considerações finais do trabalho, suas limitações e possíveis propostas para trabalhos futuros relacionados ao tema.

2 REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta a revisão bibliográfica utilizada como base para o desenvolvimento do protótipo final. Inicialmente, são abordados os principais conceitos e ferramentas relacionados ao desenvolvimento de sistemas *web*, arquitetura de *software*, tecnologias e *frameworks* utilizados na construção da aplicação, compondo a fundamentação teórica do trabalho, seção 2.1.

Em seguida na seção 2.2, são apresentados os trabalhos correlatos, por meio da análise de sistemas e plataformas que possuem características semelhantes à proposta do 3Z, especialmente nos contextos de *marketplace*, interação social e gestão empresarial. A análise dessas plataformas permite compreender as abordagens utilizadas por soluções já existentes, identificar funcionalidades relevantes e observar limitações que contribuíram para o desenvolvimento da proposta apresentada neste trabalho.

2.1 Fundamentação teórica

Segundo Wazlawick (2021), a fundamentação teórica consiste nos conceitos obtidos na literatura. Esta seção apresenta os principais conceitos, tecnologias e ferramentas encontrados na literatura técnica que serviram como base para o desenvolvimento do sistema 3Z. Essa etapa é fundamental para embasar as decisões de projeto, arquitetura e implementação adotadas durante o desenvolvimento da aplicação. A seguir, são apresentados os principais conceitos relacionados ao desenvolvimento de sistemas *web*, arquitetura de *software*, além dos *frameworks* e tecnologias utilizados na construção da plataforma.

2.1.1 Hypertext Transfer Protocol (HTTP)

O HTTP se originou na década de 1980, quando o britânico Tim Berners-Lee, trabalhava na Organização Europeia para Pesquisa Nuclear (CERN). Em 1989, ele teve a ideia de um sistema global que compartilhasse informações, pois percebeu a necessidade e buscava maneiras de facilitar a colaboração entre cientistas, pesquisadores e instituições (BERNERS-LEE, 1989).

E assim foi criado o projeto “WorldWideWeb”. Já em 1990, Berners-Lee apresentou sua primeira proposta do protocolo HTTP. Essa primeira versão era bastante simples e permitia apenas solicitações do tipo GET (leitura). Dessa forma, os clientes podiam solicitar páginas armazenadas em um servidor e receber como resposta apenas o conteúdo *HyperText Markup Language* (HTML) correspondente, sem informações adicionais em cabeçalhos.

Posteriormente, foram sendo adicionados diversos outros métodos, como POST (criar), PUT (atualizar) e DELETE (deletar). Esses métodos passaram a ser amplamente utilizados no desenvolvimento de *Application Programming Interface* (API), especialmente seguindo o padrão *Representational State Transfer* (REST), no qual se estabelece uma associação entre as operações do protocolo HTTP e as operações de *Create, Read, Update e Delete* (CRUD).

O HTTP opera em um modelo cliente-servidor, em que um cliente, faz requisições a um servidor e obtém uma resposta, como páginas da web, imagens ou arquivos. E toda essa interação acontece por meio do protocolo *Transmission Control Protocol* (TCP), que garante uma comunicação confiável. Ao longo dos anos o HTTP se tornou o protocolo que clientes e servidores mais utilizam para se comunicar e, inclusive, é o protocolo mais utilizado para aplicações e APIs da *web*.

Posteriormente, houve também a criação do *HyperText Transfer Protocol Secure* (HTTPS), que acrescentou uma camada de segurança ao protocolo, implementando criptografia *Secure Sockets Layer* (SSL) e *Transport Layer Security* (TLS) para proteger dados sensíveis. Compreender o HTTP é de extrema importância para desenvolvedores, principalmente no contexto *web*, pois permite a criação de aplicações eficientes e seguras (FIELDING; NOTTINGHAM; RESCHKE, 2022). Estas mudanças, mostram que o protocolo HTTP continua em constante evolução e continua a desempenhar um papel central na arquitetura da *web*, impulsionando a troca de informações e a conexão global de sistemas digitais.

2.1.2 Model View Controller (MVC)

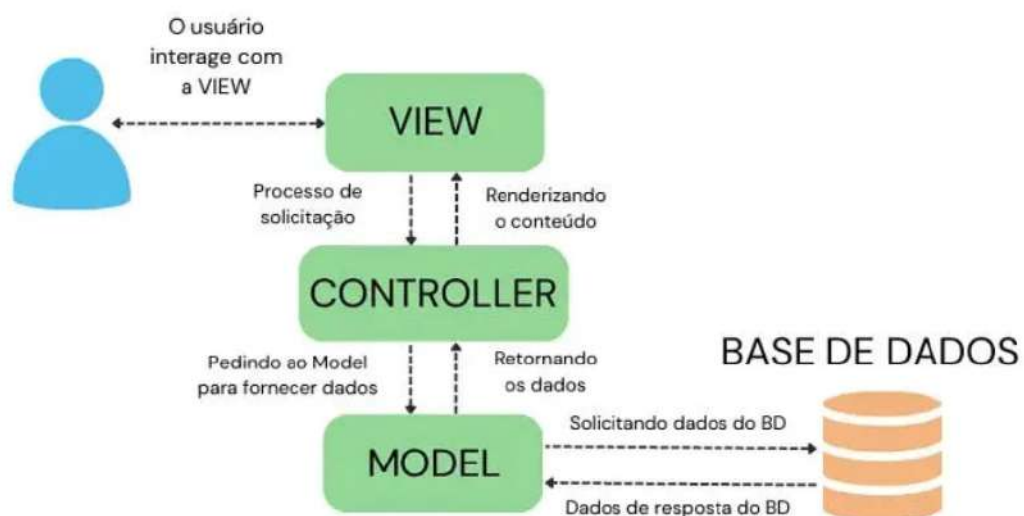
A arquitetura MVC foi criada nos anos 80 na Xerox Parc por Trygve Reenskaug, que, em 1979, iniciou o que viria a ser conhecido como arquitetura MVC, esse acrônimo que em português significa arquitetura Modelo-Visão-Controle é um padrão de arquitetura de software focado no reuso de código e na separação da apresentação e a interação dos dados, onde é estruturado em três componentes lógicos: *model* (modelo), *view* (visualização, visão ou vista) e *controller* (controlador).

Esses componentes interagem entre si da seguinte maneira. A camada de modelo (*model*) é responsável por modelar os dados e o comportamento por trás do processo de negócio. Preocupa-se apenas com o armazenamento, a manipulação e a geração de dados, é um encapsulamento de dados e de comportamento independente da apresentação.

A camada de visualização (*view*) pode ser qualquer forma de representação dos dados,

como uma tabela ou um diagrama, é onde os dados solicitados do modelo são exibidos. Essa camada promove também a interação com o usuário. Por fim, tem-se a camada de controle (*controller*), que faz a mediação entre entrada e saída, comandando a visualização e o modelo para que sejam alterados de forma apropriada conforme a solicitação do usuário. O foco do controle é a ação do usuário, onde são manipulados os dados que ele insere ou atualiza, chamando, em seguida, o modelo (DUARTE, 2011). A Figura 3 representa visualmente o funcionamento e a organização dessa arquitetura.

Figura 3 – Arquitetura MVC



Fonte: Medium (2024)

2.1.3 Python

Linguagem de programação desenvolvida pelo programador holandês Guido van Rossum, no final da década de 1980, teve sua primeira versão lançada em 1991. O objetivo principal era desenvolver uma linguagem de programação que fosse simples, legível, eficiente e fácil de aprender, corrigindo as limitações de linguagens existentes na época. O Python fez grandes avanços para o desenvolvimento de código aberto, utilizando as Propostas de Enriquecimento do Python (PEP), principal ferramenta para sugestões de melhoramento e discussões da comunidade da linguagem.

Suas principais características que podem ser citadas são:

- Linguagem interpretada: o python é uma linguagem interpretada, ou seja, o código não

precisa passar previamente por um processo de compilação tradicional. Sua execução ocorre por meio de uma máquina virtual, responsável por interpretar o código e traduzi-lo para instruções compreensíveis pelo sistema. Essa característica contribui para maior agilidade no desenvolvimento e testes de aplicações (Python Software Foundation, 2026). Por outro lado, esse processo de interpretação adiciona uma camada extra de execução, o que pode tornar o desempenho da linguagem inferior quando comparado a linguagens compiladas;

- **Sintaxe simples e legível:** a sintaxe do Python é simples, intuitiva e próxima da linguagem natural, o que facilita o aprendizado e a leitura do código. Por esse motivo, é considerada uma linguagem de alto nível, amplamente utilizada tanto por iniciantes quanto por desenvolvedores experientes;
- **Multiparadigma:** onde há possibilidade de programar em vários paradigmas, tais como procedural, funcional e orientação a objetos;
- **Tipagem dinâmica:** o Python possui tipagem dinâmica, ou seja, o tipo das variáveis é definido automaticamente em tempo de execução. Dessa forma, não é necessário declarar previamente o tipo dos dados, o que torna o desenvolvimento mais ágil e flexível;
- **Ampla aplicabilidade:** a linguagem é amplamente utilizada em diversas áreas, como desenvolvimento de *software*, análise de dados, inteligência artificial, automação de tarefas e desenvolvimento de aplicações *web*;
- **Bibliotecas e comunidade:** o python conta com uma extensa biblioteca padrão, além de um vasto ecossistema de bibliotecas externas. Sua comunidade, sempre ativa, facilita o aprendizado a resolução de problemas e o acesso a recursos para diferentes tipos de projetos.

Além disto é uma linguagem de programação de uso geral que funciona para criar uma grande variedade de aplicações diferentes e que não é especializada em nenhum problema específico. Essa versatilidade e facilidade de uso para pessoas iniciantes, fez com o Python que se tornasse uma das linguagens mais comuns atualmente.

Segundo informações coletadas no índice TIOBE, o Python é a linguagem de programação mais utilizada nos dias atuais como mostrado na Figura 4 (TIOBE Software, 2026). Isso se deve à sua grande versatilidade, destacando-se em áreas como desenvolvimento *web*, análise

de dados e inteligência artificial. Apesar disso, é uma linguagem que possui peculiaridades e limitações que devem ser respeitadas e reconhecidas durante o desenvolvimento. Assim, o desenvolvedor deve seguir as boas práticas da linguagem e da programação para garantir um resultado de excelência em seu produto final.

Figura 4 – Ranking TIOBE de linguagens de programação 2026

Junho de 2026	Junho de 2025	Mudar	Linguagem de Programação	Classificações	Mudar
1	1		 Python	18,96%	-6,91%
2	3	▲	 C	10,77%	+1,30%
3	2	▼	 C++	8,03%	-2,65%
4	4		 Java	7,90%	-0,94%
5	5		 C#	4,85%	+0,17%
6	6		 JavaScript	3,04%	-0,17%

Fonte: TIOBE (2026)

2.1.4 Django.PY

O Django é um *framework web* Python de código aberto, que se destaca por oferecer um ambiente que simplifica a criação de aplicações *web* escaláveis, e que promove o desenvolvimento rápido e um *design* limpo, proporcionando ferramentas robustas e eficientes para desenvolvedores (Django Software Foundation, 2026a).

A linguagem Python é reconhecida por sua simplicidade e flexibilidade, a decisão de utilizar o Python como base para plataformas de *backend*, está fortemente ligada ao surgimento deste *framework*. O Django surgiu com a ideia de facilitar o desenvolvimento para *web*, dispensando a necessidade de reinventar a roda, ou seja, reescrever códigos comuns a todos os projetos *web* em Python.

O *framework* disponibiliza uma estrutura pré-configurada e diversas bibliotecas com código pronto, automatizando grande parte das tarefas repetitivas. Dessa forma, os desenvolvedores podem concentrar seus esforços nas regras de negócio da aplicação, sem a necessidade de lidar intensivamente com configurações de ambiente ou implementações básicas.

Essa abordagem está alinhada à filosofia conhecida como “*batteries included*”, na qual o *framework* já oferece, de forma nativa, uma ampla gama de funcionalidades essenciais, como autenticação, painel administrativo e mecanismos de segurança, reduzindo a dependência de bibliotecas externas.

Ele foi desenvolvido em 2003 pela empresa Lawrence Journal World e liberado publicamente em 2005, a partir disso, o Python passou a ser mais adotado dentro do desenvolvimento de aplicações *server-side* (o famoso “*Back-end*”), como desenvolvimento *web*, criação de serviços *online* e processamento de dados, devido à sua simplicidade e facilidades trazidas pelo *framework*.

Dentre as principais características do Django, segundo a sua documentação oficial (Django Software Foundation, 2026a), podemos citar:

- Arquitetura Model View Template (MVT): sendo uma arquitetura similar a MVC mas com uma abordagem diferente da tradicional, é dividida em *Model*, *View* e *Template*, na qual o modelo (*Model*) assim como no MVC é a camada que representa os dados, a *View* atua como controle intermediando a conexão entre os dados e a visualização, e os *Templates* que são responsáveis por enviar requisições e exibir as respostas;
- Segurança: o Django é um *framework* que se preocupa bastante com segurança e fornece mecanismos integrados para proteção contra diversas vulnerabilidades, como ataques *Cross-Site Request Forgery* (CSRF), *Structured Query Language Injection* (SQL Injection) e *Cross-Site Scripting* (XSS), além de validação automática de formulários e sistema de autenticação;
- *Object-Relational Mapping* (ORM): permite que o desenvolvedor interaja com o banco de dados usando classes e objetos em Python, sem precisar escrever diretamente para o SQL. Isso simplifica o acesso de dados e ajuda a evitar problemas de segurança ao mesmo tempo que melhora a legibilidade do código;
- Interface de administração: o Django disponibiliza um painel administrativo pronto, o que facilita o gerenciamento de dados da aplicação e teste;
- Sistemas de *cache* e *caching*: é possível armazenar temporariamente dados acessados com frequência em uma área de memória de alta velocidade para acelerar o acesso. Isso melhora o desempenho da aplicação, reduzindo a carga sobre o banco de dados e os recursos do servidor;
- Internacionalização: as aplicações criadas podem ser facilmente adaptadas para suportar múltiplos idiomas e regiões;

- *Template*: o Django fornece maneiras poderosas e flexíveis de separar a lógica de apresentação do código *backend*, através de arquivos que definem a estrutura e o visual das páginas *web*. O *Template Engine* é o mecanismo responsável pelo processamento dos arquivos de *template*, permitindo a inserção e manipulação dinâmica de dados, o que possibilita a geração de conteúdo dinâmico na aplicação;
- *Uniform Resource Locator* (URL) amigáveis: o Django permite associar URLs às *views* correspondentes nas aplicações, através do mapeamento de URL. Isso fornece uma maneira mais organizada de direcionar as solicitações do usuário para as partes apropriadas do código.

2.1.5 Django Rest Framework (DRF)

O foco dessa ferramenta é o desenvolvimento de *web* APIs de forma simples e ágil. Embora seja possível criar APIs utilizando apenas o Django, o DRF foi projetado para facilitar a construção de APIs REST, abstraindo grande parte da complexidade envolvida nesse processo. Para isso, ele disponibiliza um conjunto de recursos e abstrações específicas, que simplificam esse processo (Django REST Framework Team, 2026).

O DRF gera APIs navegáveis que auxiliam na usabilidade para os desenvolvedores. Além disso, possui um sistema de autenticação e serialização dos dados. Esse é um dos motivos para usar o DRF, porque ele torna a serialização mais fácil. Considere o cenário de desenvolvimento de uma API e o *JavaScript Object Notation* (JSON) é a linguagem principal dos recursos, o serializador transforma os dados JSON para um formato que o Python entenda, e vice-versa, isso é feito de forma simples com este *framework*.

O Django Rest funciona como um complemento ao Django, transformando-o em uma poderosa ferramenta *back end*, dentre suas principais características encontradas em sua documentação (Django REST Framework Team, 2026), podemos citar:

- *Serialização*: a criação e o consumo de dados da API são muito mais eficientes, devido a um poderoso sistema de serialização que permite converter objetos complexos, como *models* do Django, em formatos como JSON, *Extensible Markup Language* (XML), entre outros;
- *Views e Endpoints* predefinidos: as operações de CRUD apresentam *views* predefinidas, o que facilita a criação de *endpoints* da API;

- Validação: com uma validação de entrada integrada no DRF, há uma camada adicional de segurança para garantir que os dados inseridos sejam válidos e consistentes.

2.1.6 JavaScript

Na década de 90 houve uma grande disputa no mercado dos navegadores. Nesse momento da história, a Empresa Netscape, planejava construir algo que pudesse deixar a navegação mais dinâmica, assim, em 1995, a Netscape contratou o desenvolvedor Brendan Eich para criar uma linguagem de *script* que trouxesse “vida” às páginas *web*. Desse modo, foi criada uma linguagem que foi chamada de LiveScript. Após uma parceria com a Sun Microsystems, empresa responsável pela criação da linguagem Java, seu nome foi alterado para JavaScript (Mozilla Developer Network, 2026).

O JavaScript, é uma linguagem de programação interpretada de alto nível, sendo uma das linguagens de programação mais populares do mundo, devido principalmente ao fato de ser a linguagem padrão que os navegadores interpretam e que em conjunto com o HyperText Markup Languages (HTML) e Cascading Style Sheets (CSS) formam a base de toda a *web* (Mozilla Developer Network, 2026).

Ele também se destaca por deixar as páginas *web* dinâmicas, o que é sua principal função a linguagem atua no lado do cliente (ou *client-side*), portanto consegue executar suas tarefas diretamente na máquina do usuário sem precisar acessar algum servidor. O JavaScript se mostrou tão relevante, que, em 2009, foi criado o Node.js, um ambiente de execução JavaScript no qual é possível que essa linguagem também seja executada no *server-side* (lado do servidor) (OpenJS Foundation, 2026).

O Node.js é capaz de interpretar um código JavaScript, da mesma forma que no navegador. Sendo assim, tornando possível o envio de instruções sem precisar de um navegador ativo, bastando ter o Node.js instalado e utilizar o terminal para executar um programa construído em JavaScript.

Todas as linguagens de programação têm suas desvantagens e o JavaScript não é a exceção. Suas principais desvantagens são, a possibilidade do JavaScript ser usado para executar programas maliciosos sem o conhecimento do usuário, o fato de ser vulnerável a brechas de segurança e por se tratar de uma linguagem interpretada, pode apresentar desempenho inferior quando comparada a linguagens compiladas

2.1.7 TypeScript

O TypeScript é uma super linguagem baseada no JavaScript, ou seja, uma linguagem que amplia outra já existente, adicionando funcionalidades adicionais sem substituir sua base original (Microsoft, 2026a). Foi desenvolvido e é mantido pela Microsoft, onde lançaram a primeira versão em outubro de 2012, pois viram a necessidade, conforme os projetos começaram a ficar maiores e mais complexos, os desenvolvedores começaram a sentir falta de algo que os auxiliasse a gerenciar essa complexidade.

Enquanto o JavaScript é uma linguagem dinamicamente tipada, ou seja, a verificação de tipos ocorre em tempo de execução, o TypeScript é uma linguagem com tipagem estática opcional, permitindo a verificação de tipos durante o desenvolvimento, antes da execução do código. Conseguindo fazer muitas inferências de tipo e ajudar a identificar possíveis falhas (Microsoft, 2026a).

2.1.8 React

O React é uma biblioteca de código aberto, criado pela Meta que é usado para criar interfaces de usuário (UI) em aplicativos *web*. Popularizado por ser fácil de usar, altamente flexível e escalável.

Utiliza JavaScript para construir interfaces por meio de componentes, que representam partes específicas da interface de um aplicativo *web*. Cada componente pode possuir um estado, responsável por armazenar informações dinâmicas que mudam conforme a interação do usuário, como dados de formulários ou alterações visuais, quando é realizada alguma ação, como cliques ou inserção de dados, o estado é atualizado por meio de funções *callback*, refletindo automaticamente essas mudanças na interface.

Além disso, o React utiliza o conceito de Virtual Document Object Model Virtual (DOM), que é uma representação em memória da interface do usuário. Quando ocorre alguma alteração no estado dos componentes, o Virtual DOM é atualizado e comparado com o DOM real, aplicando apenas as mudanças necessárias. Isso é muito mais rápido do que atualizar-lo diretamente, o que torna o React muito rápido e eficiente.

O React é amplamente utilizado no desenvolvimento de aplicações *web* que exigem atualizações em tempo real, como redes sociais e sistemas interativos. Isso é possível graças à eficiência do React em atualizar apenas as partes necessárias da interface, garantindo alto desempenho. Outro uso importante está no desenvolvimento de aplicações escaláveis e de fácil

manutenção, especialmente em projetos com equipes grandes, por ser uma biblioteca baseada em componentes reutilizáveis, ela facilita a organização do código e o trabalho colaborativo.

Vale a pena ser destacado a presença da JavaScriptXML (JSX), uma extensão da linguagem JavaScript usada pelo React para criar interfaces de usuário. Ele permite que você misture código JavaScript com sintaxe de HTML, o que torna mais fácil escrever componentes de interface de usuário em um único arquivo de código, além de ser possível a integração do CSS para estilização dos componentes.

2.1.9 Next.js

A documentação oficial do React recomenda o uso de um *framework* para a manipulação da biblioteca. E um dos que mais tem se destacado no dias atuais é o Next (React Team, 2026). Um *framework* baseado em React que permite desenvolver aplicações *web full-stack* de forma mais simples e eficiente, combinando a criação de interfaces com recursos avançados e otimizações automáticas, ele cuida de configurações complexas, permitindo que o desenvolvedor foque no desenvolvimento do produto. Com isso, é possível criar aplicações rápidas, dinâmicas e interativas, seja trabalhando sozinho ou em equipe.

Inicialmente focado no desenvolvimento de interfaces usando a biblioteca React, que é o seu forte, surgiu como uma evolução desse ecossistema. Diferente do React, que atua apenas como uma biblioteca para criação de interfaces, o Next.js funciona como um *framework* que adiciona uma estrutura mais organizada e recursos avançados ao desenvolvimento, como renderização no lado do servidor (SSR), geração de páginas estáticas (SSG) e roteamento automático baseado nos arquivos (Vercel, 2026).

Dessa forma, o ele não substitui o React, mas o potencializa, oferecendo ferramentas que melhoram significativamente o desempenho, a escalabilidade e a otimização para mecanismos de busca. Além disso, ele simplifica diversos processos que, no React puro, exigiriam configurações manuais, com isso, o desenvolvimento se torna mais ágil e estruturado, permitindo que o desenvolvedor foque mais na lógica da aplicação e na experiência do usuário, enquanto o *framework* cuida de aspectos como organização de rotas, otimização de carregamento e integração entre *front-end* e *back-end*.

O Next.js também aproveita recursos que já existentes, mas, pouco utilizados no React puro, sendo eles a separação entre "*Client Components*" e "*Server Components*", que, como o próprio nome indica, são componentes que podem rodar tanto no lado do cliente (navegador)

quanto no servidor do próprio Next. Esses recursos estão diretamente relacionados ao React, mas foram popularizados e aplicados de forma prática pelo Next.js.

O conceito de *client* está relacionado aos componentes que são executados no navegador do usuário. Esses componentes são responsáveis pela interação da aplicação, como cliques, formulários e atualizações dinâmicas de estado, eles seguem o modelo tradicional do React, utilizando JavaScript no lado do cliente para construir interfaces reativas.

No entanto, por rodarem no navegador, esses componentes estão mais expostos, o que exige maior cuidado com segurança, já que dados sensíveis não devem ser manipulados diretamente no *client*, evitando vulnerabilidades como exposição de informações ou manipulação indevida por usuários. Segundo a documentação (Vercel, 2026) da *framework* esse componentes devem ser utilizados para:

- Manipular o estado da interface e usar manipuladores de eventos, como por exemplo, `onClick`, `onChange`;
- Lógica do ciclo de vida;
- APIs exclusivas para navegadores, por exemplo, `localStorage`, `window`, `Navigator.geolocation`, etc;
- *Hooks* personalizados.

Já o servidor no Next.js é responsável por processar e renderizar partes da aplicação antes mesmo de serem enviadas ao navegador, por meio dos *Server Components* e da renderização no lado do servidor (SSR). Nesse modelo, além de melhorar o desempenho inicial e o SEO, há também um ganho importante em segurança, pois dados sensíveis, regras de negócio e acessos a banco de dados podem ser tratados diretamente no servidor, sem exposição ao usuário final.

O Next.js gerencia esse ambiente com base em Node.js, permitindo uma integração mais eficiente entre *front-end* e *back-end*, tornando a aplicação mais leve, performática e segura. De acordo com a documentação (Vercel, 2026) eles devem ser utilizados para:

- Obter dados de bancos de dados ou APIs próximos à fonte;
- Utilizar chaves de API, *tokens* e outros segredos sem expô-los ao cliente;
- Reduzir a quantidade de JavaScript enviada ao navegador;

- Melhorar a primeira renderização do conteúdo e transmitir conteúdo progressivamente para o cliente.

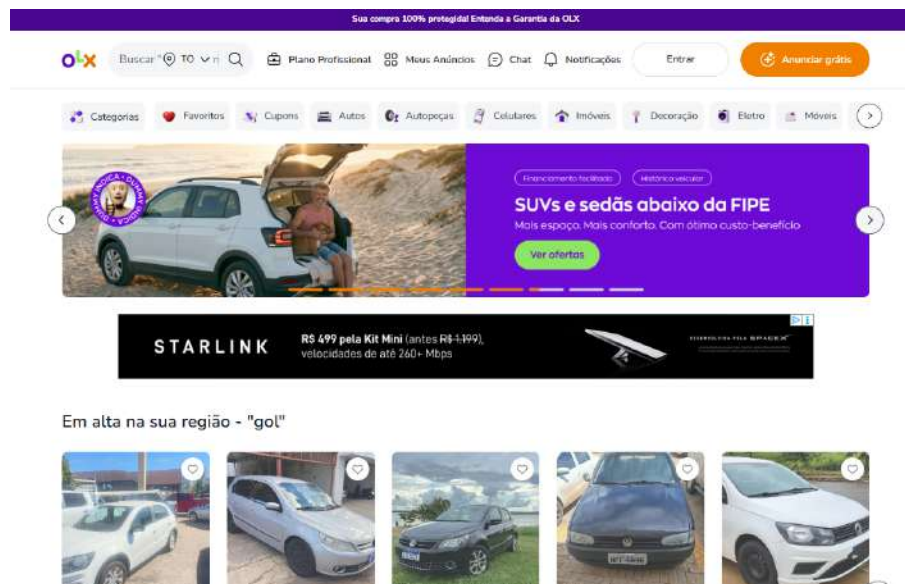
2.2 Trabalhos correlatos

Esta seção apresenta os principais projetos relacionados ao tema abordado neste trabalho, destacando sistemas e plataformas que possuem características semelhantes à proposta do sistema 3Z. Considerando a grande quantidade de soluções existentes na área, foram selecionados os sistemas mais relevantes e populares no contexto de comunicação, gestão e *marketing*.

A análise desses projetos permitiu compreender as abordagens utilizadas por outras plataformas, identificar funcionalidades semelhantes e observar limitações que motivaram o desenvolvimento da solução proposta, o estudo dos sistemas correlatos contribui para fundamentar as decisões de projeto, arquitetura e funcionalidades implementadas na aplicação.

2.2.1 OLX

Figura 5 – Página inicial do site OLX.



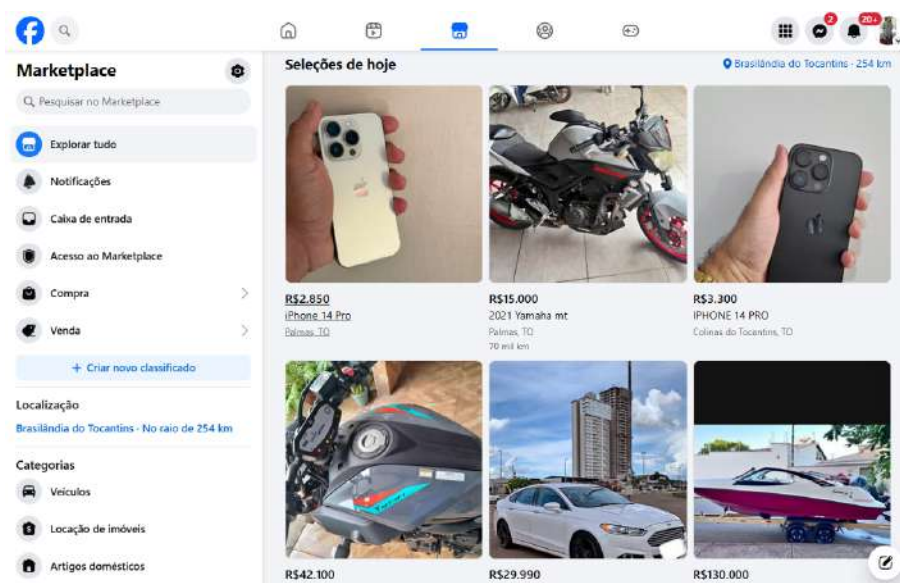
A Figura 5 apresenta a interface inicial da OLX, considerada uma das principais inspirações e referências para o desenvolvimento do sistema 3Z. A plataforma digital é voltada para anúncios e comercialização de produtos e serviços entre usuários. O sistema funciona como um ambiente de classificados *online*, no qual usuários podem publicar anúncios, visualizar produtos por categorias e entrar em contato diretamente com anunciantes por meio de ferramentas de comunicação integradas à plataforma (AMORIM; QUARESMA, 2022).

A OLX destaca-se pela simplicidade na criação de anúncios e pela facilidade de conexão entre compradores e vendedores através de postagens, permitindo negociações rápidas em diferentes segmentos do mercado. Além disso, a plataforma utiliza mecanismos de localização e filtros de pesquisa que auxiliam os usuários na busca por produtos e serviços próximos à sua região (AMORIM; QUARESMA, 2022).

A relação com o sistema 3Z está principalmente na proposta de divulgação de produtos e interação entre usuários dentro de um ambiente digital. Entretanto, o 3Z amplia essa proposta ao incorporar diferentes recursos, permitindo não apenas a publicação de anúncios, mas também o gerenciamento financeiro, acompanhamento de movimentações, utilização de seguidores e favoritos, além da separação entre perfis pessoais e empresariais. Dessa forma, o sistema busca oferecer uma experiência mais integrada entre comunicação, comercialização e administração empresarial.

2.2.2 Facebook Marketplace

Figura 6 – Página de marketplace do Facebook.



A Figura 6 apresenta o Facebook Marketplace, recurso integrado à plataforma Facebook que permite a divulgação e negociação de produtos e serviços entre os próprios usuários da rede social (Meta Platforms, 2026). Diferentemente de *marketplaces* tradicionais, o Facebook Marketplace utiliza a estrutura social já existente da plataforma, permitindo que usuários negociem utilizando seus perfis pessoais, grupos e conexões dentro da rede.

Um dos principais diferenciais da plataforma está na facilidade de divulgação local, já que os anúncios são exibidos de acordo com a localização e interesses dos usuários. Além disso, o sistema aproveita recursos já presentes na rede social, como mensagens instantâneas, compartilhamentos e perfis públicos, tornando a comunicação entre compradores e vendedores mais rápida e acessível (Meta Platforms, 2026).

A proposta do sistema 3Z possui semelhanças com o Facebook Marketplace principalmente pela integração entre interação social e comercialização dentro de um mesmo ambiente digital. Entretanto, o 3Z busca expandir essa ideia ao adicionar recursos voltados à gestão empresarial e organização financeira, permitindo que usuários empresariais realizem gerenciamento mais estruturado de suas atividades comerciais dentro da plataforma.

3 MATERIAIS E MÉTODOS

Este capítulo apresenta os materiais, tecnologias e métodos utilizados no desenvolvimento do sistema 3Z. Inicialmente, são descritas as linguagens de programação, *frameworks*, ferramentas, banco de dados e recursos de *hardware* empregados na construção da aplicação, destacando suas funções e importância dentro da arquitetura do sistema (Seção 3.1). Em seguida, é apresentada a metodologia de desenvolvimento adotada, bem como os processos de modelagem, levantamento de requisitos e validação das funcionalidades implementadas (Seção 3.2).

3.1 Materiais

Este capítulo apresenta os materiais e tecnologias utilizados no desenvolvimento do sistema 3Z. Cada tecnologia é descrita conforme sua função dentro da arquitetura do sistema, destacando sua contribuição para o funcionamento e organização da solução proposta.

3.1.1 Linguagens de programação.

Nesta seção são apresentadas as linguagens de programação utilizadas no desenvolvimento do sistema. Sendo elas o Python e o JavaScript com TypeScript já apresentadas anteriormente no capítulo 2.

- Python: principal linguagem utilizada no sistema, atuando no desenvolvimento do *back-end* da aplicação. É responsável pela implementação das regras de negócio, manipulação e persistência de dados, além de garantir a comunicação entre a interface e o banco de dados. Foi escolhida graças à sua versatilidade, legibilidade e conjunto de bibliotecas e *frameworks* disponíveis.
- JavaScript / TypeScript: utilizadas no desenvolvimento do front-end do sistema, sendo responsáveis pela construção de interfaces dinâmicas e interativas. Em conjunto com o CSS, permitem a criação da camada visual da aplicação e a interação do usuário com o sistema. Além disso, possibilitam a comunicação com o back-end por meio de requisições HTTP e APIs REST, garantindo a integração entre as diferentes partes da aplicação. Suas escolhas se justificam pela ampla adoção no desenvolvimento web, pelo ecossistema de ferramentas e pela facilidade de integração com serviços e APIs externas.

3.1.2 Frameworks

Esta seção fala sobre os frameworks utilizados no desenvolvimento e suas principais aplicações dentro do sistema, sendo eles o Django, DRF e Next.js, já apresentados no capítulo anterior.

- Django: principal *framework* de *back-end* do sistema. É responsável por criar e manipular os dados por meio de sua ORM, permitindo a abstração do banco de dados e facilitando operações de CRUD. É responsável também por implementar as configurações de desenvolvimento do sistema, implementar boa parte das estruturas de segurança da aplicação, além de gerenciar rotas, regras de negócio e a comunicação entre as diferentes camadas do sistema, o que contribui para uma aplicação mais estruturada, segura e de fácil manutenção. Foi selecionada graças ao seu amplo arsenal de ferramentas nativas disponíveis para o desenvolvedor, além de sua agilidade na construção de aplicações *web*;
- Django Rest Framework (DRF): *framework* essa que em conjunto com o Django compõe todo o conjunto de *back-end*, sendo responsável pela criação de *endpoints*, serialização de dados e manipulação de APIs REST. Além disso, atua diretamente nos processos de autenticação e autorização de usuários, limitação de requisições (*throttling*), paginação de dados e implementação de mecanismos de busca e filtros dentro do sistema. Seu uso foi de suma importância no desenvolvimento, pois garante a comunicação eficiente entre a camada cliente e o servidor da aplicação. Escolhido principalmente por ser um complemento ao Django padrão, fornecendo funcionalidades que, sem ele, precisariam ser implementadas manualmente.
- Next.js: Fechando o conjunto de *frameworks* do 3Z, o Next.js atua principalmente, mas não exclusivamente, no lado do cliente, sendo responsável por fornecer a base do *front-end* da aplicação. Ele realiza a renderização das interfaces React, recebe e envia informações via API para o Django, gerencia a navegação entre páginas, contribui para a construção de uma experiência de usuário mais rápida e fluida e oferece a estrutura para a construção dos componentes React. Além disso, dentro do sistema, o servidor do Next também atua como intermediário entre o *front-end* e o *back-end*, garantindo que o cliente não se comunique diretamente com o servidor Django da aplicação. Dessa forma, evita-se a exposição de informações sensíveis e reduz o risco de manipulação indevida de

dados por parte do usuário final. Sua escolha se justifica pela eficiência, desempenho e organização que proporciona no desenvolvimento de aplicações com React.

3.1.3 Outras ferramentas

- PostgreSQL: o banco de dados escolhido e utilizado na aplicação foi o PostgreSQL, sendo responsável por criar uma estrutura relacional entre as tabelas e armazenar os dados do sistema. Seu uso dentro do sistema se deve principalmente ao fato de ser um dos bancos de dados mais utilizados em conjunto com o Django, devido à sua robustez, confiabilidade e suporte a recursos avançados, proporcionando melhor desempenho e escalabilidade ao sistema, além de ser o banco de dados recomendado pela documentação oficial do Django para aplicações em produção (Django Software Foundation, 2026b).
- Docker: Ferramenta de containerização, que desempenha um papel fundamental no desenvolvimento do 3Z. Ele é responsável por criar e gerenciar os contêineres da aplicação, permitindo a separação das diferentes camadas do sistema em ambientes isolados, que se comunicam entre si de forma eficiente (Docker Inc., 2026).
- Visual Studio Code (VS Code): ambiente de desenvolvimento integrado (IDE) utilizado na construção do 3Z, trata-se de uma ferramenta leve e altamente extensível, que oferece suporte a diversas linguagens de programação por meio de extensões. No contexto do projeto, foi utilizado para o desenvolvimento tanto do *back-end* quanto do *front-end*, proporcionando recursos como, destaque de sintaxe, depuração de código, automação de tarefas (Microsoft, 2026b), o que contribuiu para maior produtividade e organização no processo de desenvolvimento.

3.1.4 Hardware

O desenvolvimento dos algoritmos e os testes dos mesmos foram realizados em um notebook Acer Nitro V15 (2024), utilizado como ambiente local do sistema, com as seguintes especificações técnicas:

- Processador 13th Gen Intel(R) Core(TM) i7-13620H (2.40 GHz);
- Memória RAM: DDR5 16 GB;

- Placa gráfica: NVIDIA GeForce RTX 4060 Laptop GPU (8 GB) Intel(R) UHD Graphics (128 MB);
- Armazenamento: Unidade de armazenamento SSD de 477 GB (sendo utilizado 462 GB);
- Sistema Operacional e Arquitetura: Windows 11 Home 64 bits;

Tais configurações de *hardware* foram essenciais para garantir um bom desempenho durante o desenvolvimento e a execução do projeto, possibilitando a utilização de múltiplas ferramentas simultaneamente, servidores locais e contêineres Docker. Contribuíram também para a realização de testes mais eficientes e para a validação das funcionalidades da aplicação em um ambiente controlado.

3.2 Métodos

A metodologia de desenvolvimento escolhida foi baseada em um modelo incremental. Modelo no qual o desenvolvimento foi realizado de forma gradual em pequenas etapas, permitindo a construção contínua das funcionalidades e testes frequentes a cada nova implementação. Esse processo possibilitou maior flexibilidade durante mudanças de requisitos e adaptações, além de facilitar a identificação e correção de erros ao longo do desenvolvimento.

Essa metodologia também facilitou a modelagem da aplicação. Que foi modelado inicialmente usando um Diagrama Entidade Relacionamento que será apresentado na sessão 4.3, Essa modelagem inicial e as adaptações realizadas ao longo do desenvolvimento utilizaram como base requisitos levantados a partir de uma pesquisa sólida, informações confiáveis coletadas em meios *online* e uma entrevista.

Essa entrevista foi realizada através de um questionário criado no Google Formulários, no qual foram feitas 4 perguntas relacionadas a *marketing*, comunicação e gestão, para 12 pessoas, de diferentes áreas de atuação, que se encaixam na categoria de autônomos, micro ou pequenos empresários das cidades de Colinas do Tocantins e Guaraí.

Nessa entrevista foram feitas as seguintes perguntas:

- Pergunta 01: Qual é o seu principal canal de divulgação online de produtos ou serviços?
- Pergunta 02: Qual o principal meio de contato utilizado com seus clientes?
- Pergunta 03: Como é feita a sua sua gestão financeira?

- Pergunta 04: Considerando o uso de ferramentas digitais citadas acima para Marketing, Comunicação e Gestão, qual a sua maior dificuldade em operar com pelo menos uma solução em cada um desses três pilares?

Essas perguntas foram elaboradas com o objetivo de identificar as principais ferramentas utilizadas pelos entrevistados, bem como compreender as dificuldades enfrentadas em relação à divulgação, comunicação e gestão. As respostas obtidas contribuíram para uma melhor compreensão da realidade e das necessidades dos profissionais entrevistados, servindo como uma das bases para orientar as decisões relacionadas ao desenvolvimento e às funcionalidades propostas pelo sistema 3Z.

4 PROJETO

Este capítulo apresenta os principais elementos de projeto do 3Z. São detalhadas a arquitetura do sistema, os protótipos gráficos das interfaces, a modelagem de dados, os diagramas estruturais e comportamentais baseados na *Unified Modeling Language* (UML), bem como representações lógicas dos utilizadas.

Também serão apresentados os principais componentes responsáveis pela comunicação entre as camadas da aplicação, evidenciando a integração entre interface, regras de negócio e persistência de dados. Dessa forma, este capítulo estabelece a base técnica necessária para a implementação e evolução do sistema 3Z.

4.1 Arquitetura do projeto

O sistema foi construído baseando-se na arquitetura *Model View e Controller* (MVC), na qual as partes foram projetadas seguindo o princípio da separação de responsabilidades. Essa abordagem permite organizar a aplicação de forma modular, facilitando o desenvolvimento, a manutenção e a escalabilidade do sistema (DUARTE, 2011).

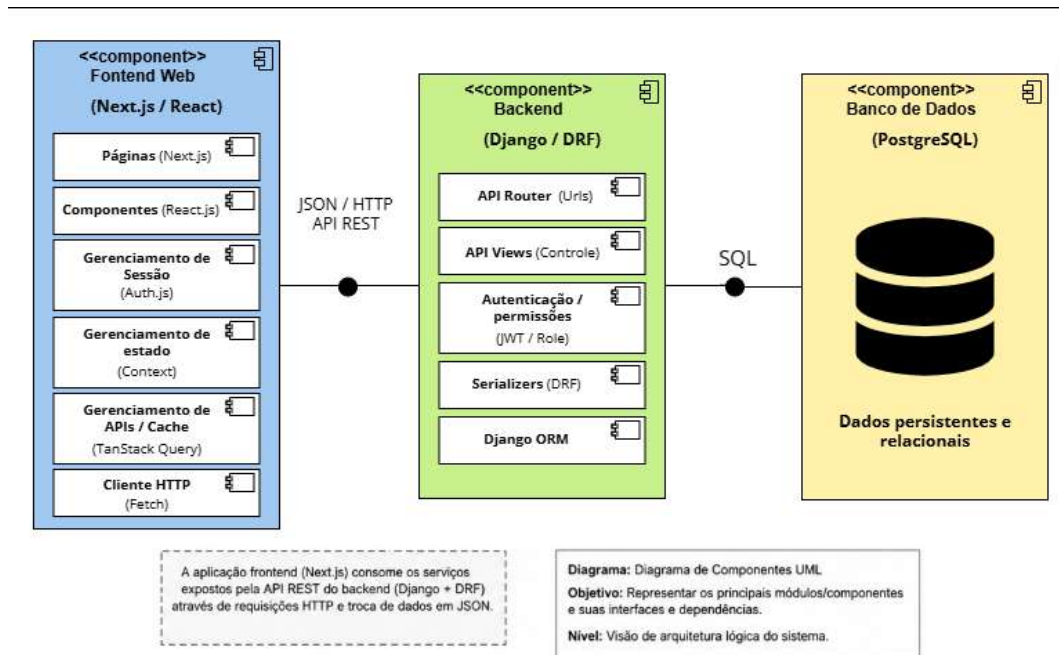
A adoção dessa arquitetura contribui também para uma maior reutilização de código, redução do acoplamento entre componentes e melhor organização estrutural do projeto (DUARTE, 2011). Além disso, o padrão MVC favorece futuras expansões e manutenções, permitindo que alterações em uma camada tenham impacto reduzido sobre as demais partes do sistema.

Na camada de modelo, é utilizado o banco de dados PostgreSQL, responsável pelo armazenamento e pela persistência das informações do sistema. A camada de controle foi desenvolvida com auxílio do framework Django, encarregado de receber as requisições enviadas pelo cliente, processá-las e retornar as respostas adequadas.

Já a camada de visualização foi desenvolvida utilizando o Next, responsável pela construção das interfaces gráficas e pela interação com o usuário. A *framework* também atua como "intermediário" entre o cliente e o Django, de modo que as requisições são inicialmente enviadas ao servidor do Next onde é injetado informações sensíveis da sessão, e somente após esse processo, encaminhadas a camada de controle. O mesmo ocorre com as respostas retornadas ao cliente.

A Figura 7 apresenta um diagrama de componentes que detalha a arquitetura do sistema, evidenciando a interação entre seus módulos e o fluxo de funcionamento da aplicação.

Figura 7 – Diagrama de componentes do 3Z



Fonte: Desenvolvido pelo autor (2026)

4.2 Protótipos

Nesta seção, são apresentadas algumas das principais interfaces gráficas do sistema. Estes protótipos foram desenvolvidos na ferramenta Figma e têm como objetivo estabelecer um esboço visual inicial da aplicação. Além disso, as prototipagens foram elaboradas com foco na Experiência do Usuário (UX), buscando proporcionar interfaces intuitivas, acessíveis e visualmente consistentes.

A interface a seguir, apresentada na Figura 8, mostra um protótipo da tela de cadastro e gerenciamento dos estados na parte administrativa do sistema. Ela foi projetada com uma abordagem minimalista, buscando oferecer ao usuário acesso rápido e intuitivo às funcionalidades disponíveis.

Essa interface inicial serviu como base para a criação do padrão visual das demais prototipagens administrativas, além de ter sido desenvolvida com o objetivo de possibilitar o reaproveitamento de componentes semelhantes em diferentes partes da área administrativa, promovendo maior padronização, organização e eficiência no desenvolvimento.

Figura 8 – Protótipo da tela de cadastro e listagem dos estados

Controle de Estruturas

Gerais ▼ Produtos ▼ Serviços ▼

Gerais: Estado

Nome

Sigla

Cadastrar Limpar

Lista de Estados

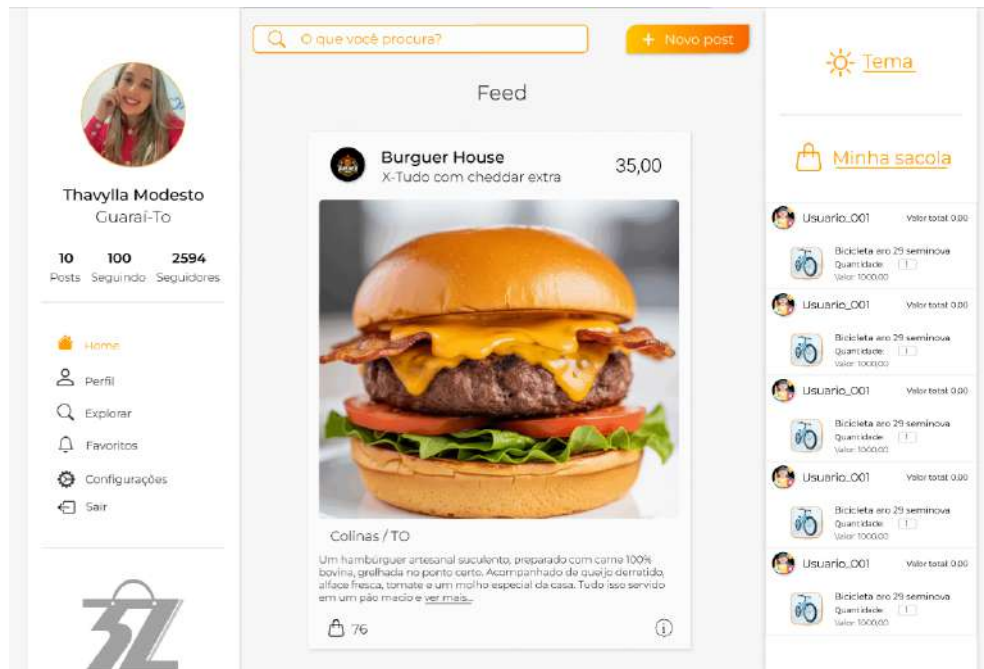
Nome	Editar	Deletar
Tocantins		
Pará		

Fonte: Desenvolvido pelo autor (2026)

A interface principal da plataforma, apresentada na Figura 9, adota uma abordagem minimalista baseada em um *feed* dinâmico, projetado para integrar de forma fluida as interações sociais e o *marketplace*. O *layout* é organizado em três colunas: à esquerda, um menu de navegação com informações do usuário; ao centro, o *feed* de publicações que tem como intuito ser uma espécie de “vitrine”; e, à direita, o módulo “Minha Sacola”, posteriormente substituído pela área de favoritos.

O *design* serviu como base para o padrão visual do sistema, utilizando a cor laranja para reforçar a identidade visual e destacar elementos interativos. De modo geral, o protótipo tenta apresentar uma interface moderna, organizada e intuitiva, priorizando acessibilidade visual, navegação simplificada e integração entre funcionalidades sociais e comerciais do sistema.

Figura 9 – Protótipo da página inicial



Fonte: Desenvolvido pelo autor (2026)

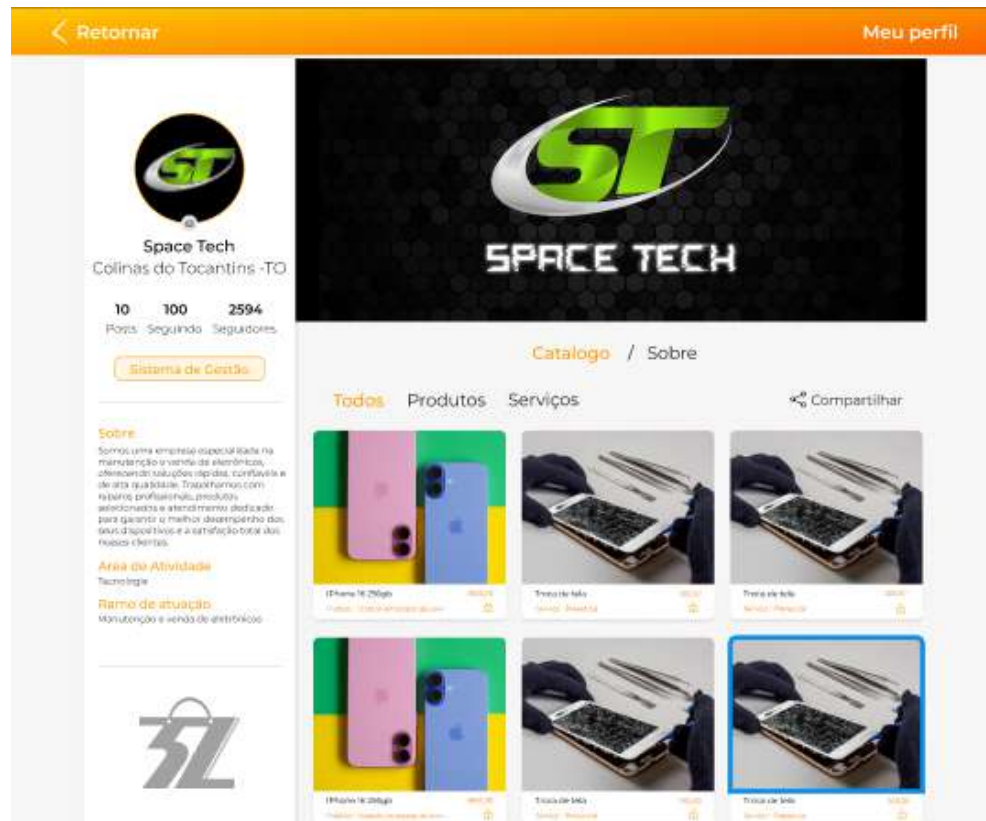
O perfil, apresentado na Figura 10, adota uma estrutura visual que mantém uma abordagem minimalista e direta, preservando a identidade visual e foco na cor laranja apresentada anteriormente.

O *layout* é dividido em duas seções principais: à esquerda, um painel vertical com algumas informações do usuário; à direita, há um banner superior personalizado e um sistema de abas que separa a exibição entre o catálogo e as informações sobre do usuário. Abaixo, filtros rápidos permitem alternar entre produtos e serviços, exibidos em uma grade de cartões padronizados com imagens, preços e especificações.

E por fim é mostrado na Figura 11 a prototipagem da interface de cadastro de novas postagens dos usuários empresariais dentro de sua área de gestão, apesar de manter cores e estilo parecido com as apresentadas anteriormente, essa tenta fugir da estilização de rede social e busca apresentar algo mais próximos do mercado, com um menu de navegação do lado esquerdo e o formulário de cadastro no restante da página.

Esse protótipo inicial serviu como base para a criação das demais interfaces da área de gestão empresarial, servindo como referência para uma padronização visual.

Figura 10 – Protótipo de perfil



Fonte: Desenvolvido pelo autor (2026)

Figura 11 – Protótipo da tela de cadastro de postagem empresarial.



Fonte: Desenvolvido pelo autor (2026)

4.3 Diagrama Entidade Relacionamento (DER)

Esta seção apresenta o Diagrama Entidade Relacionamento Figura 12 do sistema, responsável por representar de forma estrutural a organização dos dados, os atributos das entidades e os relacionamentos existentes entre elas. A modelagem foi desenvolvida utilizando o modelo relacional, adotado devido à sua confiabilidade, integridade e eficiência no gerenciamento de informações estruturadas (CHEN, 1976). Além disso, a utilização de banco de dados relacional possibilita maior consistência dos dados e melhor controle das relações entre as tabelas do sistema.

Entre as principais entidades do sistema, destacam-se de "usuario" e de "postagem" que servem como base para a grande maioria. A entidade "usuario" é responsável por armazenar os dados de autenticação e informações públicas e privadas dos usuários cadastrados na plataforma. Já a entidade "postagem" armazena os dados dos conteúdos publicados e possui um relacionamento de pertencimento com a entidade "usuario", garantindo que cada registro esteja vinculado ao seu respectivo autor.

Também existem as entidades de "perfil" e "perfil_empresa", que estão diretamente ligadas à "usuario" e são responsáveis pela separação entre pessoas físicas e jurídicas, deixando definido o papel e as informações de cada usuário dentro do sistema. Essa divisão permite uma melhor organização dos dados e possibilita que diferentes tipos de usuários possuam características e funcionalidades específicas na plataforma, lembrando que administradores não são ligados a uma entidade de "perfil", onde seu cargo dentro do sistema é definido no atributo "tipo" da entidade "usuario".

O mesmo se aplica às entidades de produtos e serviços, que possuem relacionamento direto com as postagens, garantindo que cada publicação seja classificada especificamente como um produto ou um serviço, de acordo com sua finalidade, dessa forma, evitando-se a criação de postagens que não estejam de acordo com o padrão do sistema.

Outras que se destacam são as estruturas gerais, ligadas às entidades de "usuario" e "postagem", que podem representar: "cidade", "estado", "garantia", "area_atividade" e "ramo_atuacao". Há também as estruturas ligadas à entidade de "produto", que são: "condição", "cor", "material", "tipo_produtos" e "categoria_produtos". E, por fim, as estruturas ligadas à entidade de "servico", que são elas: "recorrencia", "forma_execucao", "tipo_servicos" e "categoria_servico".

Apesar de serem entidades que possuem poucos atributos, elas desempenham um papel

fundamental na organização e padronização das informações do sistema, auxiliando na coleta de dados, na realização de filtros e buscas, além de contribuírem para a simplificação dos processos de cadastro. Essas estruturas também ajudam a reduzir redundâncias e garantem maior consistência nas informações armazenadas pela aplicação.

É possível observar também as entidades relacionadas a área de gestão financeira do sistema, como "caixa", "movimentacao", "entradas" e "saidas". Elas estão diretamente relacionadas as entidades de "usuario" e "postagem", permitindo que usuários jurídicos possam fazer o controle sobre suas operações financeiras realizadas em suas empresas.

Destaca-se também a entidade "movimentacao", responsável pelo registro das operações financeiras, podendo ser classificada como entrada ou saída no respectivo caixa em aberto. As movimentações do tipo entrada podem estar associadas à entidade "postagens", que estabelece um relacionamento de muitos para muitos, no qual uma "entrada" pode estar vinculada a várias postagens, assim como uma mesma postagem pode compor diferentes movimentações. Essa modelagem permite ao usuário relacionar conjuntos de postagens cadastradas na plataforma às respectivas transações financeiras, proporcionando maior controle das operações realizadas.

Além de registrar movimentações financeiras, essas entidades auxiliam no acompanhamento de ganhos, despesas e transações vinculadas às atividades do sistema, proporcionando uma organização financeira mais eficiente e integrada às funcionalidades da aplicação. Vale ressaltar que os dados utilizados nessas movimentações vêm das entidades "produto_empresa" e "servico_empresa", entidades essas que são de uso exclusivo para perfis empresariais.

E, por fim, as entidades "seguindo_seguidores", que está relacionada aos usuários, e "favoritos", que está relacionada às postagens cadastradas na plataforma. Essas entidades permitem uma maior interação entre os usuários e as publicações do 3Z, possibilitando o acompanhamento de perfis, a criação de conexões entre usuários e o armazenamento de postagens de interesse dentro do sistema.

4.4 Diagrama de Casos de Uso

Os diagramas de casos de uso são utilizados para representar os requisitos funcionais de um sistema sob a perspectiva do usuário, descrevendo as funcionalidades oferecidas e as interações entre os atores e o sistema. Dessa forma, eles contribuem para a compreensão do comportamento esperado da aplicação e para a definição de seus requisitos (BOOCH; RUMBAUGH; JACOBSON, 2005).

O diagrama a seguir, apresentado na Figura 13, mostra os principais casos de uso do sistema 3Z, demonstrando as principais funcionalidades disponíveis e a interação entre os diferentes tipos de usuários da plataforma, é composto pelos atores Usuário Anônimo, Pessoal, Empresarial e Administrador, cada um possuindo diferentes níveis de acesso e permissões dentro do sistema.

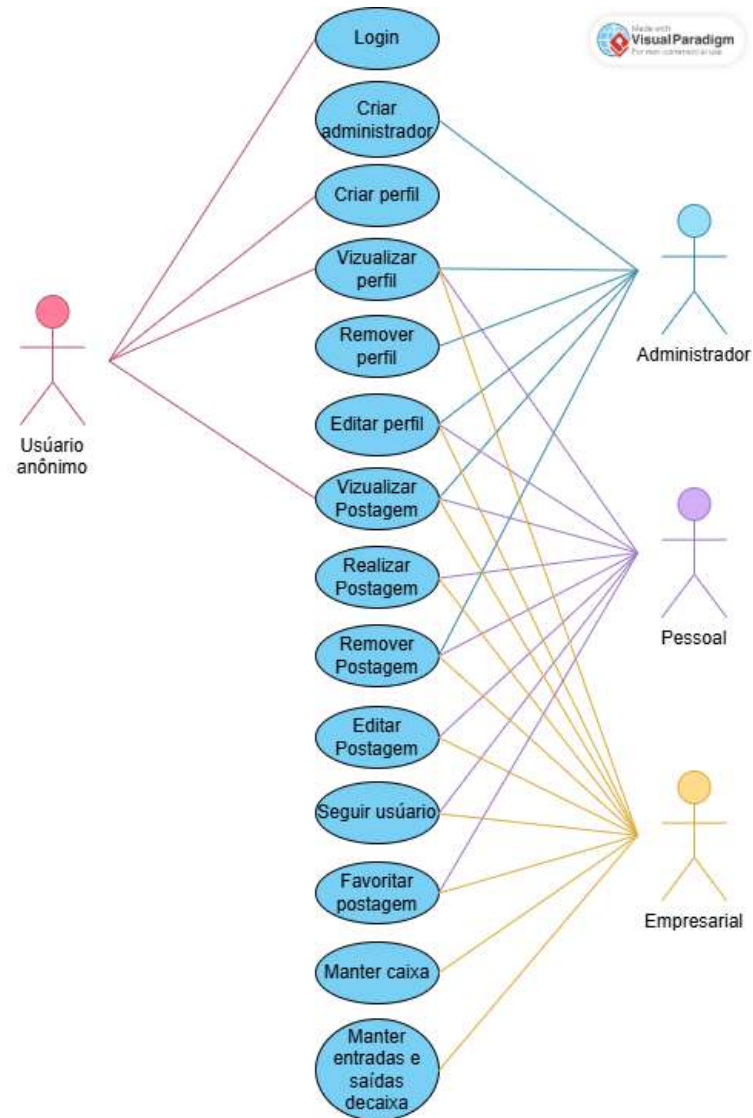
Iniciando pelas funcionalidades dos usuários anônimos, destacam-se o *login*, utilizado para realizar a autenticação na plataforma, caso o usuário não possua uma conta registrada, ele poderá efetuar seu cadastro. Além disso, possui permissão para visualizar perfis de outros usuários, visualizar postagens e entrar em contato com os anunciantes via WhatsApp por meio dessas publicações.

Seguindo para o ator administrador, este possui funcionalidades principalmente voltadas ao gerenciamento de contas e conteúdos cadastrados na plataforma. Entre suas atribuições, estão a visualização e a exclusão de perfis e publicações, mantendo o controle e a moderação do sistema, além de poder editar as próprias informações. Esse tipo de acesso pode ser obtido por meio do *login*, desde que a conta tenha sido previamente criada por outro administrador.

O ator representado pelo usuário pessoal possui permissões próximas às dos usuários anônimos, podendo visualizar conteúdos e outros perfis cadastrados, além de entrar em contato via WhatsApp com outros usuários. Porém, dispõe de funcionalidades adicionais, como o cadastro e o gerenciamento completo de suas próprias postagens, o gerenciamento do próprio perfil, além da possibilidade de seguir outras contas e favoritar publicações de seu interesse.

Por fim, o ator representado pelos usuários empresariais possui, em geral, funcionalidades semelhantes às dos usuários pessoais. No entanto, dispõe de recursos adicionais voltados à gestão empresarial, permitindo a abertura e o gerenciamento de caixas, bem como o controle de movimentações financeiras de entrada e saída. Além disso, esses usuários podem acompanhar dados e registros financeiros relacionados às suas atividades, proporcionando maior controle, organização e acompanhamento das finanças da empresa.

Figura 13 – Diagrama de casos de uso sistema



Fonte: Desenvolvido pelo autor (2026)

5 RESULTADOS

Este capítulo apresenta e discute os resultados obtidos após o desenvolvimento do sistema 3Z. São expostas as principais interfaces gráficas finais da plataforma, considerando que diversas telas do sistema apresentam funcionalidades, estruturas e elementos visuais semelhantes, será exibido apenas aquelas mais relevantes para a compreensão geral da aplicação.

Também são estabelecidas comparações entre os protótipos idealizados durante a etapa de planejamento e as versões implementadas no sistema, juntamente com as principais funcionalidades desenvolvidas, destacando aspectos relacionados à usabilidade, integração entre os módulos sociais e empresariais, organização visual e experiência do usuário.

5.1 Interfaces gráficas de visitantes

As Figuras 14 e 15 representam as interfaces iniciais do sistema. A primeira é exibida logo após o usuário acessar a plataforma, permitindo que ele selecione uma das cidades cadastradas. Após a seleção, o usuário é redirecionado para a área de exploração, responsável por exibir as postagens relacionadas à cidade escolhida. Nessa área, estão disponíveis recursos como filtros, buscas e paginação, possibilitando localizar publicações de acordo com o interesse desejado.

Ao selecionar uma postagem, o usuário poderá acessar a página de postagens públicas, além de poder entrar em contato com o anunciante ou acessar seu perfil público, funcionalidade que também será apresentada posteriormente. Ressalta-se que não é necessário estar autenticado para acessar essas páginas públicas do sistema.

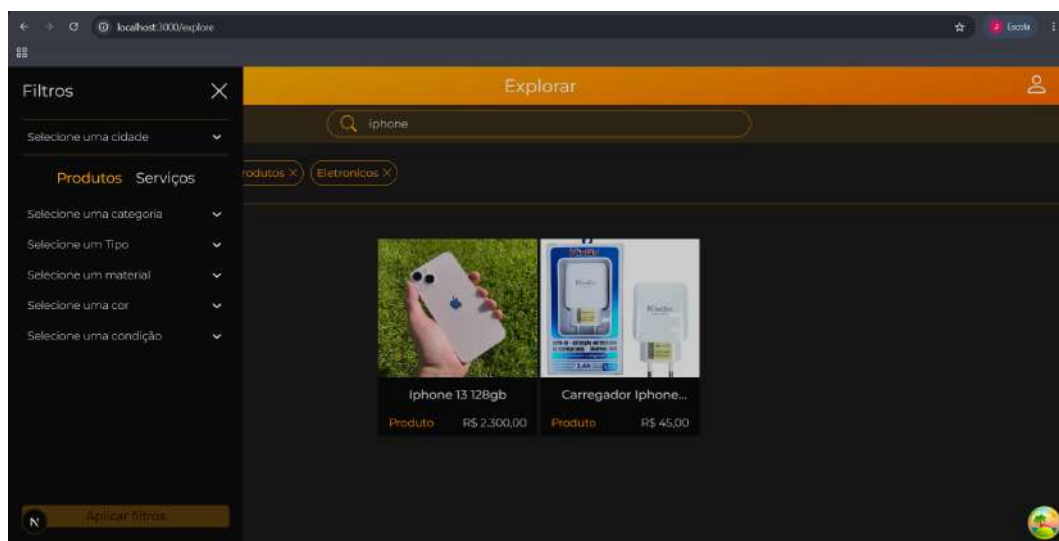
Contudo, vale destacar que os visitantes que, não realizaram autenticação, possuem apenas permissões de leitura, não sendo permitido realizar interações com publicações ou quaisquer ações que envolvam alterações no sistema.

Figura 14 – Tela inicial



Fonte: Desenvolvido pelo autor (2026)

Figura 15 – Tela de exploração



Fonte: Desenvolvido pelo autor (2026)

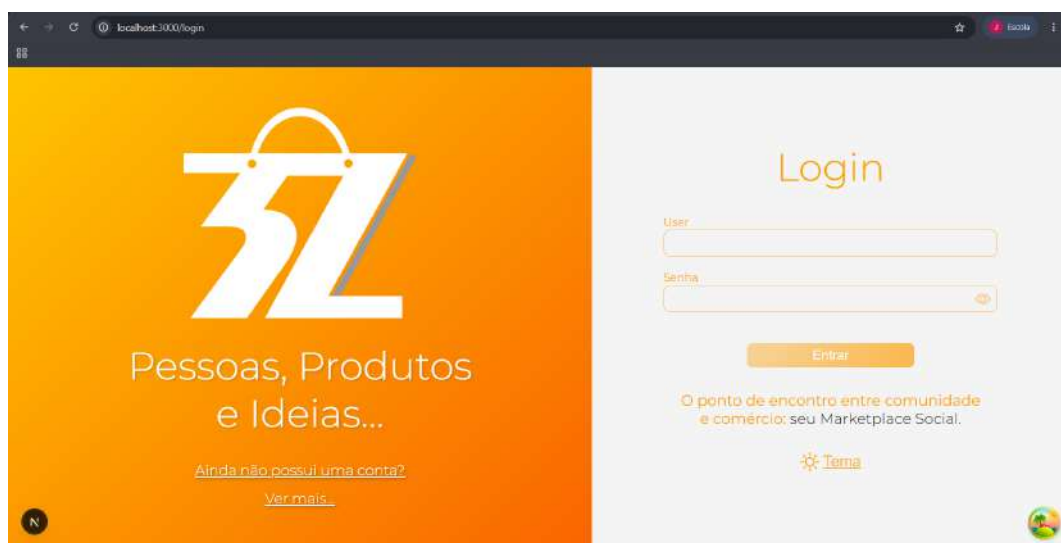
A seguir, são apresentadas as duas telas nas quais usuários não autenticados possuem permissão para interagir diretamente, a página de *login* Figura 16 e a página de cadastro de usuário Figura 17. A interface de *login* permite que usuários previamente cadastrados realizem a autenticação na plataforma por meio de suas credenciais de acesso (*username* e senha), possibilitando o uso das funcionalidades disponíveis no sistema de acordo com suas permissões de usuário.

Como medida de segurança, o sistema limita a quantidade de tentativas de autenticação a 10 requisições por hora. Dessa forma, caso o usuário informe credenciais incorretas mais

de 10 vezes, será necessário aguardar o término do período estabelecido para realizar novas tentativas de autenticação.

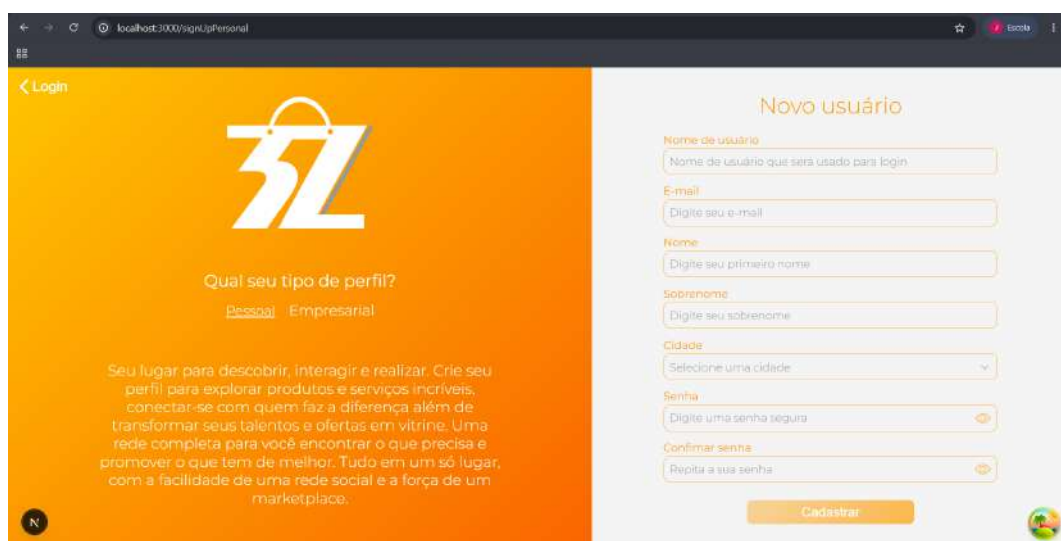
Já a interface de cadastro permite que novos usuários criem contas pessoais ou empresariais de maneira simples e intuitiva, fornecendo as informações básicas necessárias para identificação e acesso à plataforma. Ambas as telas seguem o padrão visual adotado pelo sistema, mantendo consistência estética, organização dos elementos e foco na usabilidade, buscando proporcionar uma experiência de navegação mais clara e acessível.

Figura 16 – Tela de login



Fonte: Desenvolvido pelo autor (2026)

Figura 17 – Tela de cadastro de usuário



Fonte: Desenvolvido pelo autor (2026)

5.2 Interfaces gráficas de administradores

Mesmo não sendo possível realizar o cadastro de administradores de forma convencional, é possível acessar essa área por meio do formulário de *login*, desde que o usuário já possua um cadastro previamente criado por outro administrador.

Nas figuras 18 e 19 são exibidas algumas das telas dessa área administrativa. Na primeira, é apresentada a tela de manutenção das áreas de atividades na qual apenas os administradores possuem permissão de edição. Essas estruturas podem ser visualizadas por quaisquer usuários, autenticados ou não, entretanto, apenas usuários administradores podem criar, editar ou excluir tais informações. Já na segunda interface é apresentada a área de gerenciamento de usuários, onde o administrador possui permissão para visualizar informações e, se necessário, excluir usuários empresariais e pessoais, além de poder criar e gerenciar outros usuários administradores.

No menu lateral de navegação também está disponível a área administrativa de postagens, semelhante à de usuários. Na parte inferior do menu, encontram-se as configurações do sistema, onde o administrador pode editar suas próprias informações de perfil. Vale destacar que, embora o administrador possua permissão para excluir usuários e postagens, ele pode editar apenas as informações referentes à sua própria sessão, não sendo permitido alterar dados pessoais de terceiros.

Algumas alterações foram realizadas em relação ao esboço inicial da interface figura 8, com o objetivo de facilitar o gerenciamento dessas estruturas e melhorar a usabilidade da área administrativa.

Figura 18 – Tela de manutenção das áreas de atividade

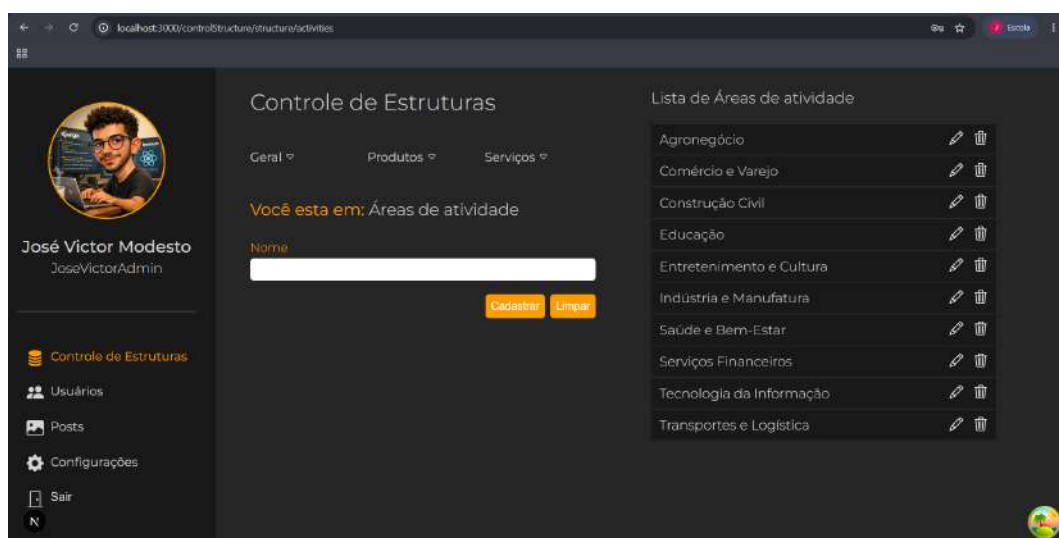


Figura 19 – Tela de gerenciamento dos usuários pessoais)



Fonte: Desenvolvido pelo autor (2026)

5.3 Interfaces gráficas de usuário pessoais e empresariais

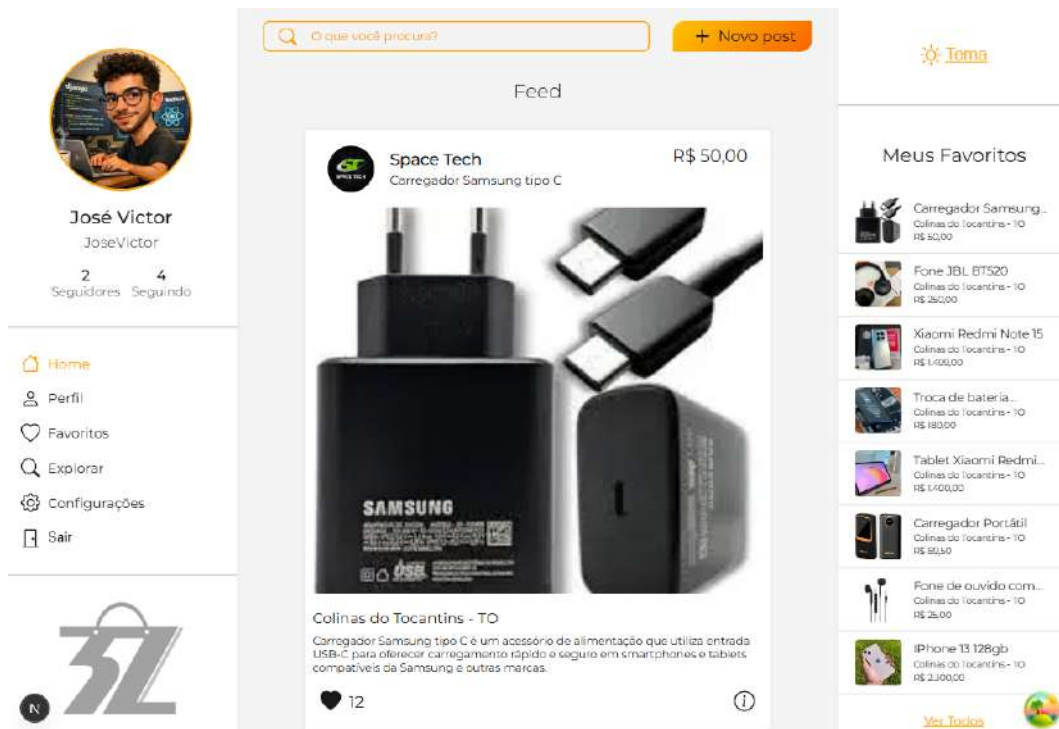
De agora em diante, as interfaces e funcionalidades apresentadas são permitidas apenas para usuários que possuam permissões pessoais ou empresariais. A primeira delas apresenta a página principal ou *home*, Figura 20, desenvolvida com o objetivo de facilitar a navegação e proporcionar acesso rápido às diversas áreas da aplicação. Essa interface permanece bastante próxima da prototipagem apresentada anteriormente figura 14, sofrendo apenas pequenas alterações, como a substituição da funcionalidade de sacola pela área de favoritos.

No lado esquerdo encontra-se o menu de navegação, responsável por simplificar o acesso às principais funcionalidades da plataforma, como perfil do usuário, área de favoritos, exploração já apresentada anteriormente na figura 9, configurações da conta e encerramento da sessão. Na região central está localizado o *feed*, considerado a principal vitrine do sistema, onde novas publicações realizadas por outros usuários na mesma região são exibidas continuamente, facilitando tanto a interação quanto a visualização de produtos e serviços compartilhados na plataforma.

Na parte superior da interface, encontra-se um campo de busca no qual os resultados pesquisados são exibidos na área de exploração. Já no lado direito da tela estão localizadas a lista de favoritos do usuário e a funcionalidade de alteração dinâmica de tema da interface, permitindo personalizar a aparência visual do sistema de acordo com a preferência do usuário.

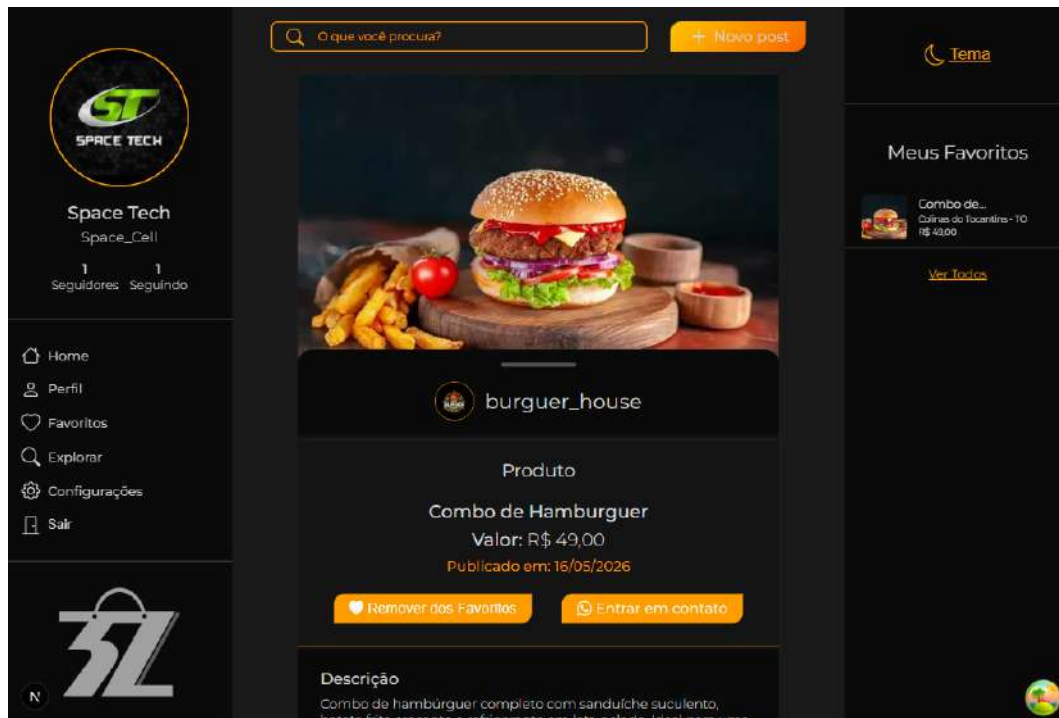
Na Figura 21 é exibido uma das telas de postagens que podem ser acessadas pelos usuários, permitindo a visualização das informações das mesmas, bem como o acesso ao perfil do anunciante (assim como no *feed*). Além disso, o usuário pode adicionar a publicação à sua lista de favoritos ou entrar em contato diretamente com o anunciante por meio do WhatsApp cadastrado, facilitando a comunicação e a negociação entre as partes.

Figura 20 – Tela principal feed



Fonte: Desenvolvido pelo autor (2026)

Figura 21 – Tela de postagem



Fonte: Desenvolvido pelo autor (2026)

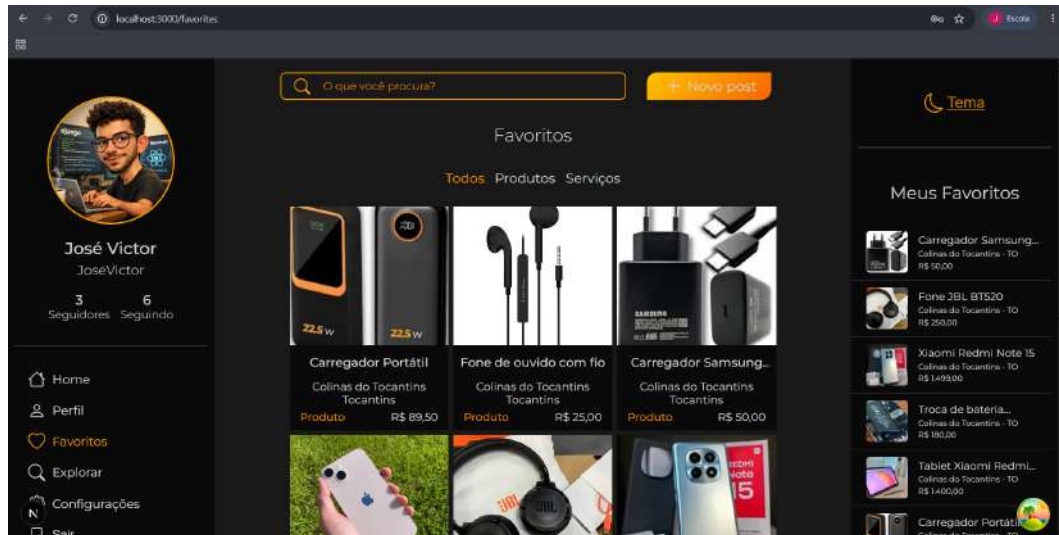
A seguir, na Figura 22, é apresentada a interface de listagem de postagens favoritas pelo usuário. Nessa área, o sistema disponibiliza uma filtragem que permite separar as publicações entre produtos, serviços ou ambos, facilitando a organização e o acesso aos conteúdos salvos pelo usuário dentro da plataforma. Essa funcionalidade foi desenvolvida com o objetivo de tornar mais prática a localização de anúncios de interesse, permitindo que o usuário acompanhe posteriormente publicações relevantes de maneira rápida e organizada.

Já na Figura 23 é exibida a interface de relacionamentos entre usuários da plataforma, onde é possível visualizar a listagem de perfis seguidos pelo usuário, bem como acessar a lista de seguidores, funcionalidade que apresenta estrutura semelhante. Esse recurso busca fortalecer a interação social dentro do sistema, permitindo maior conexão entre usuários pessoais e empresariais, além de facilitar o acompanhamento de novos conteúdos e publicações realizadas pelos perfis seguidos.

Ambas as funcionalidades têm como objetivo estimular a interação entre usuários e publicações, além de permitir a coleta de informações relacionadas às preferências e interesses dos mesmos dentro da plataforma. Vale destacar que tanto usuários com permissões pessoais quanto empresariais possuem acesso a essas funcionalidades, entretanto, apenas o próprio proprietário da sessão pode visualizar as informações listadas em sua conta, garantindo maior

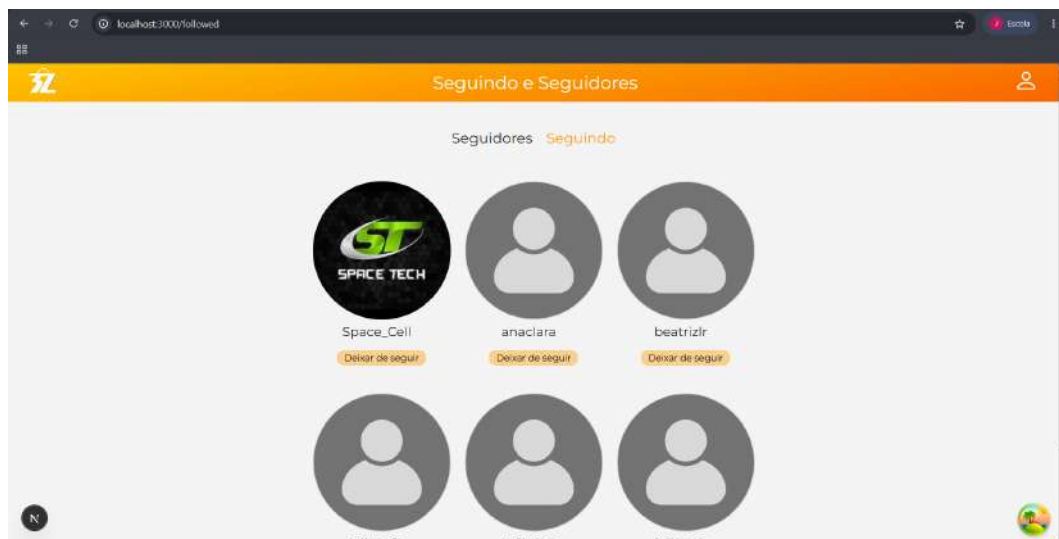
privacidade e segurança dos dados apresentados pelo sistema. É importante ressaltar que os usuários não podem favoritar as próprias postagens nem seguir a si mesmos.

Figura 22 – Tela de listagem de favoritos



Fonte: Desenvolvido pelo autor (2026)

Figura 23 – Tela de listagem de usuários que estão sendo seguidos



Fonte: Desenvolvido pelo autor (2026)

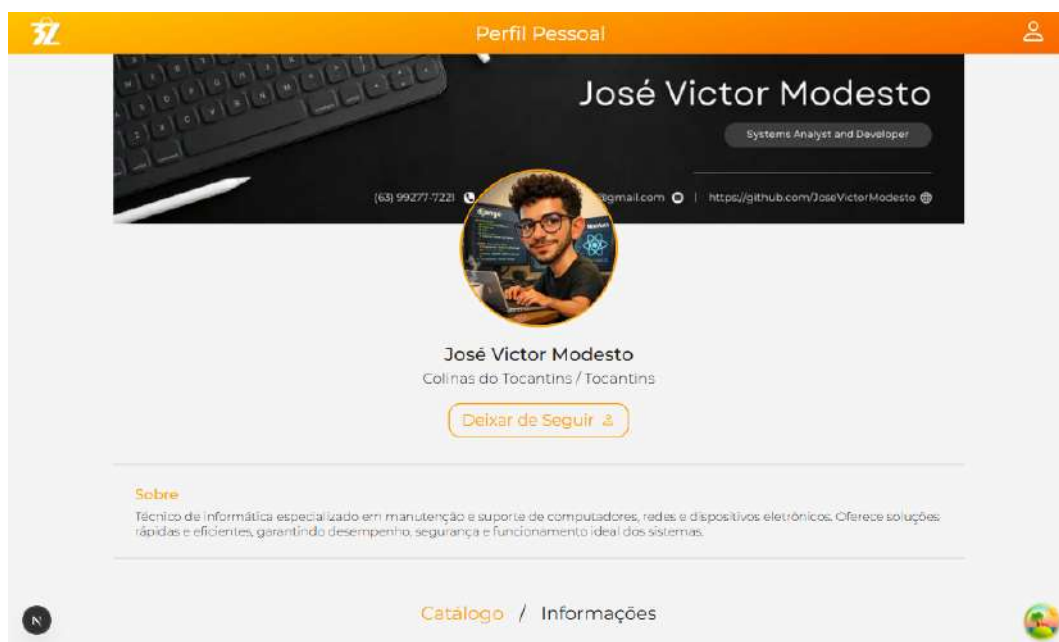
Na Figura 24, é apresentada a página de perfil público, neste caso pertencente a um usuário pessoal. Essa interface pode ser acessada por meio das postagens publicadas na plataforma ou pelas listas de seguidores e seguindo. Nela, além da exibição das principais informações do usuário visitado, disponíveis na opção “Informações” Figura 26, também é possível seguir o perfil do mesmo.

Lembrando que esses perfis públicos podem ser acessados por qualquer um, cadastrado ou não na aplicação. Entretanto, apenas usuários autenticados possuem permissão para realizar interações dinâmicas dentro da plataforma.

Além disso, através do perfil é possível acessar o catálogo público do usuário, Figura 25, onde são exibidas suas publicações, produtos e serviços compartilhados. Essa funcionalidade permite que outros usuários acompanhem as atividades e anúncios do perfil visitado de forma centralizada e organizada, facilitando tanto a navegação quanto a descoberta de novos conteúdos dentro da plataforma.

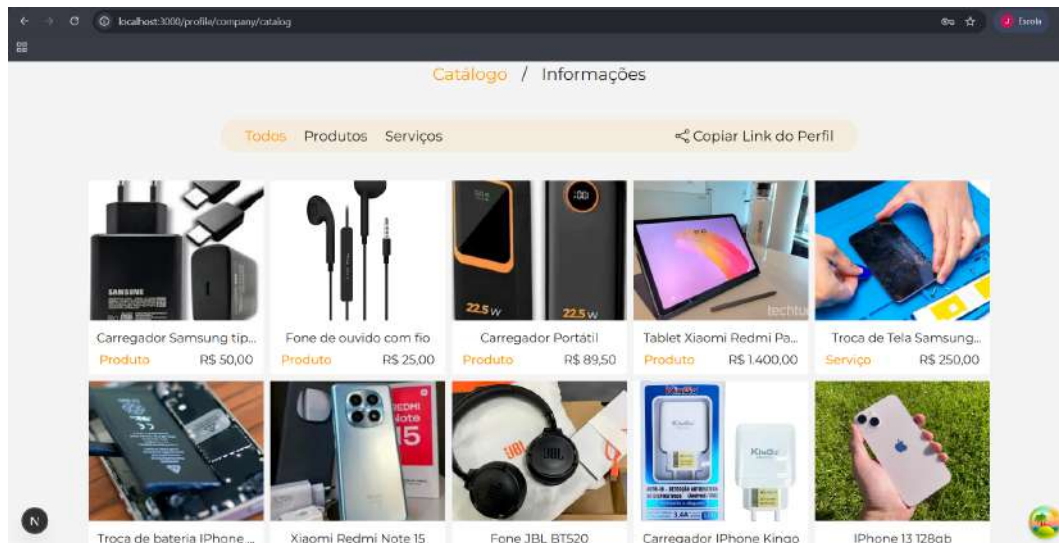
No catálogo também estão disponíveis filtros que auxiliam na diferenciação de produtos e serviços. Outra funcionalidade presente é a opção de copiar o link do perfil público, facilitando o compartilhamento do catálogo em outros meios digitais e ampliando o alcance das publicações através da *web*.

Figura 24 – Perfil Público



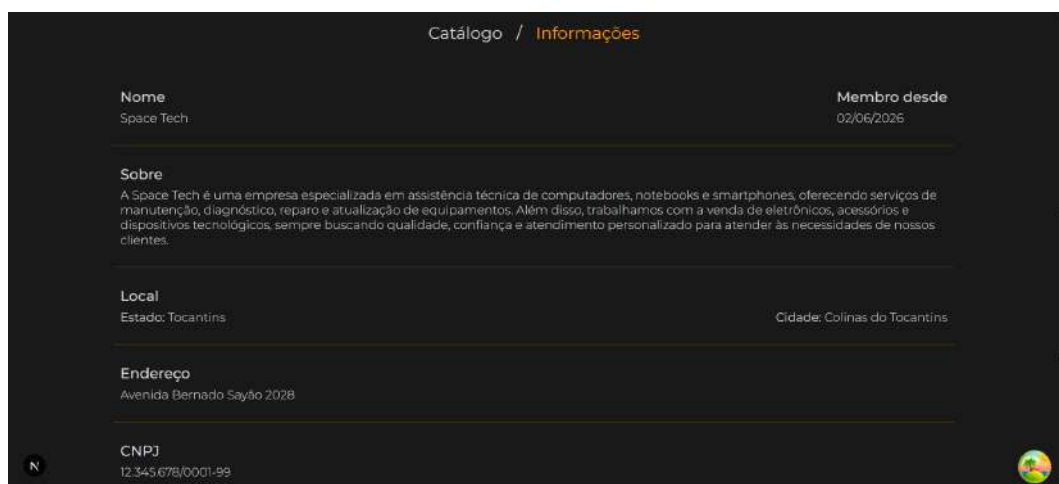
Fonte: Desenvolvido pelo autor (2026)

Figura 25 – Catálogo



Fonte: Desenvolvido pelo autor (2026)

Figura 26 – Tela de informações do usuário



Fonte: Desenvolvido pelo autor (2026)

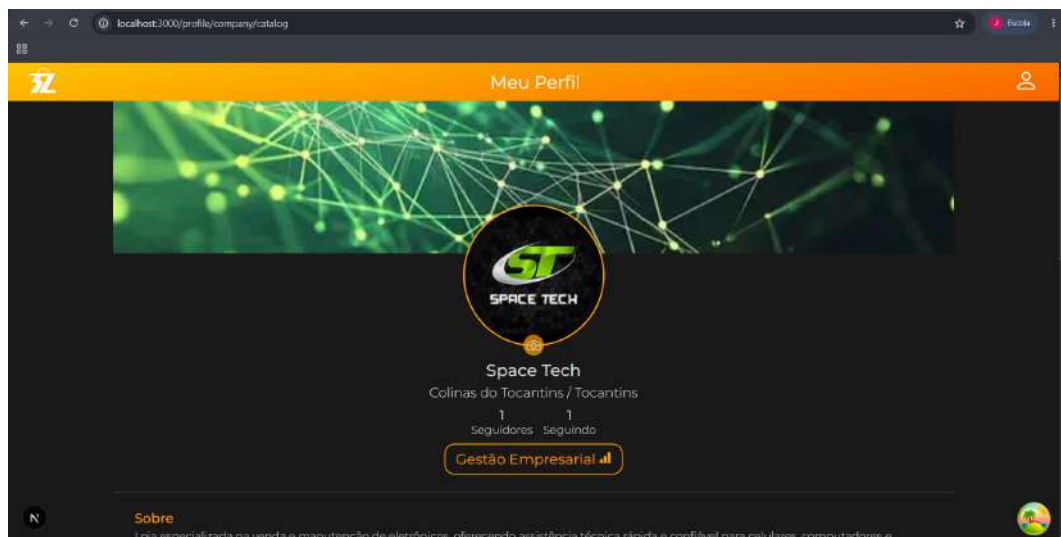
Na Figura 27 é mostrado a tela do perfil privado, diferentemente dos perfis públicos, somente o usuário autenticado na sessão possui permissão para acessá-lo, além de contar com algumas funcionalidades adicionais, como edição de avatar e banner, acesso à listagem de seguindo e seguidores e, no caso de usuários empresariais, acesso ao sistema de gestão financeira.

O perfil privado também possui o catálogo do usuário da sessão, contendo as mesmas funcionalidades presentes nos catálogos públicos (figura 25). Entretanto, diferencia-se pelo fato de que, ao acessar uma postagem, o usuário não é redirecionado para o a tela de postagens públicas, mas sim para o privado, que apresenta estrutura semelhante à anterior (figura 24). A

principal diferença está nas opções disponíveis, pois, ao invés de funcionalidades de interação e contato, o usuário possui permissões para editar ou excluir suas próprias publicações. Assim como ocorre no perfil privado, esse tipo de postagem pode ser acessado apenas pelo usuário autenticado na sessão correspondente.

E encerrando a área de perfis, temos a tela de edição das próprias informações do usuário Figura 28, onde é possível editar informações públicas e privadas da conta, adicionar ou alterar o contato de WhatsApp e modificar a senha de acesso. A interface foi desenvolvida de forma simples e organizada, buscando facilitar o gerenciamento das informações pessoais pelo usuário. Vale ressaltar que cada usuário possui permissão apenas para editar os próprios dados da sessão autenticada, não sendo permitido alterar informações pertencentes a terceiros.

Figura 27 – tela de perfil privado



Fonte: Desenvolvido pelo autor (2026)

Figura 28 – tela de configurações do usuário

The screenshot shows a web application interface for user settings. The browser address bar indicates the URL is localhost:3000/profile/settings/contacts. The page has an orange header with the title 'Configurações' and a user profile icon. The main content area is divided into two columns. The left column contains two sections: 'Meus Contatos Públicos' with input fields for 'E-mail Público' (containing 'space@xmail.com') and 'Rede Social' (containing 'Instagram: @spacetech'), followed by a yellow 'Salvar' button; and 'Whatsapp' with a text description, a prompt to enter the number, and an input field for 'Número de Whatsapp' (containing '63992777221'), also followed by a yellow 'Salvar' button. The right column contains a sidebar titled 'Minhas informações' with links for 'CNPJ', 'Contatos' (highlighted in orange), and 'Alterar senha'. A small globe icon is visible in the bottom right corner of the sidebar area.

Fonte: Desenvolvido pelo autor (2026)

5.4 Interfaces gráficas de gestão empresarial

Entrando agora na área de gestão, à qual somente usuários empresariais autenticados possuem permissão de acesso, temos as telas de gerenciamento de caixas e movimentações Figura 29, onde há *cards* que exibem de forma breve os valores de saldo inicial de abertura do caixa, o valor total de entradas e saídas, além de mostrar o lucro referente ao caixa em aberto. Na parte inferior, são listadas as movimentações de forma individual, permitindo que a empresa possa acessá-las de maneira independente e visualizar os dados de cada movimentação em formato PDF, como será mostrado mais adiante na Figura 31.

A Figura 30 mostra como é realizada a ação de cadastrar novas entradas relacionadas a postagens. Nessa interface, está disponível um formulário no qual a empresa pode preencher com as informações necessárias para o registro da movimentação. Ao cadastrar uma nova entrada vinculada a uma postagem, a quantidade correspondente é abatida das informações de estoque do produto e acrescentada ao total de saídas registrado individualmente na publicação.

Os formulários de entradas simples e saídas possuem funcionamento semelhante, porém de maneira mais simplificada, exigindo apenas o preenchimento dos atributos necessários, já que não demandam informações relacionadas às postagens. Os valores referentes a essas entradas e saídas são automaticamente adicionados às movimentações do caixa em aberto. Importante ser ressaltado que essas ações só podem ser realizadas caso exista algum caixa ativo do usuário no sistema.

As interfaces destinadas ao controle de caixa possuem um menu de gerenciamento que permite o registro de entradas e saídas financeiras, bem como a realização do fechamento de caixa. Como medida de controle operacional, o sistema permite que o usuário empresarial mantenha apenas um caixa aberto por vez. Dessa forma, caso exista uma tentativa de abertura simultânea de múltiplos caixas, o servidor bloqueia automaticamente a ação, garantindo maior integridade e organização das movimentações financeiras realizadas na plataforma.

Encerrando a área de caixas e movimentações, a Figura 31 apresenta um modelo de relatório de entrada gerado em formato PDF. Esse relatório contém informações essenciais sobre a movimentação realizada, permitindo que o usuário consulte e verifique os dados registrados de forma organizada. Além disso, o documento também pode ser utilizado como recibo para o cliente referente à venda de produtos ou à prestação de serviços realizada pela empresa. Os relatórios de saída seguem uma estrutura semelhante, contendo as informações relacionadas às despesas financeiras registradas.

Figura 29 – Tela de movimentações de caixa

Movimentação	Valor	Horário
Saída Outros	R\$ 300,00	19:40
Entrada Troca de Tela	R\$ 250,00	19:40
Saída Aluguel	R\$ 1.200,00	19:39
Entrada Venda iPhone	R\$ 2.500,00	19:39
Entrada Entrada em dinheiro	R\$ 100,00	19:34

Fonte: Desenvolvido pelo autor (2026)

Figura 30 – Tela de cadastro das entradas

Fonte: Desenvolvido pelo autor (2026)

Figura 31 – Relatório de entrada

Fonte: Desenvolvido pelo autor (2026)

Entrando na área de cadastro, a Figura 32 mostra como funciona o formulário de cadastro de novas postagens, onde o usuário pode cadastrar novos produtos ou serviços empresariais. O formulário reaproveita dados já cadastrados no sistema, além de coletar informações públicas e privadas sobre a postagem.

Essas informações serão utilizadas tanto no sistema de gestão quanto nas postagens públicas. Esse tipo de cadastro é mais completo, pois utiliza também a tabela de posts empresariais, apresentada nos diagramas acima do capítulo 4. Os usuários pessoais também possuem uma área de cadastro de postagens, porém de forma simplificada e rápida, já que não

possuem acesso à gestão financeira e não necessitam de informações tão detalhadas quanto as empresas.

As postagens cadastradas são enviadas para a área de estoque, onde cada listagem é adaptada para atender às necessidades específicas de produtos ou serviços, conforme mostrado na Figura 33. No estoque, é possível consultar informações comerciais tanto de forma geral quanto individualizada para cada postagem. Figura 34, é possível também gerenciar cada uma dessas postagens.

Outro ponto relevante a ser mencionado é que, nas regras de negócio implementadas, a realização de uma postagem está condicionada ao cadastro prévio do número de WhatsApp do usuário. Caso essa informação não esteja devidamente registrada, o usuário é direcionado a tela de edição de seus dados de contato (Figura 28) para que possa preenchê-lo, garantindo a integridade das informações e o correto funcionamento das funcionalidades que dependem desse contato.

Figura 32 – Tela de cadastro de novas postagens empresariais

A imagem mostra a interface de usuário para o cadastro de produtos no sistema Space_Cell. A interface é dividida em duas partes principais: uma barra lateral esquerda com o menu de navegação e uma área principal de formulário.

Barra Lateral Esquerda:

- Logo: Space_Cell
- Menu: Dashboard, Caixa, Cadastros (destacado), Estoque, Relatórios.
- Logo de fundo: 3L

Área Principal: Cadastro de Produtos

Abas: Produtos (selecionada), Serviços

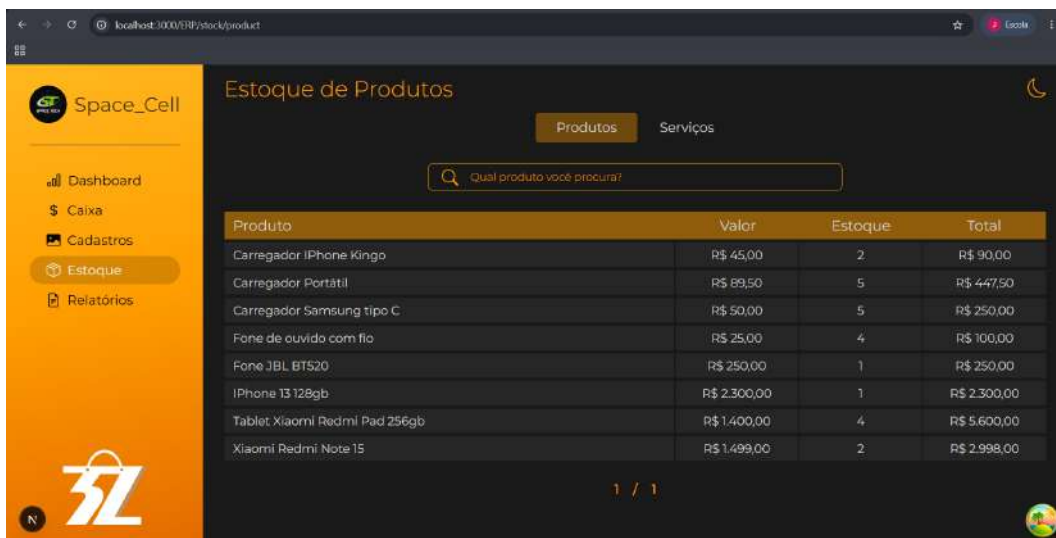
Formulário de Cadastro:

- Nome ***: Campo de texto com placeholder "Digite o nome do produto".
- Descrição ***: Campo de texto com placeholder "Digite uma descrição sobre o produto".
- Tipo de Produto ***: Menu suspenso com opção "Selecione".
- Condição de Produto ***: Menu suspenso com opção "Selecione".
- Valor de custo ***: Campo de texto com placeholder "Digite o valor de custo do produto".
- Preço ***: Campo de texto com placeholder "Digite o valor do produto".
- Quantidade em estoque ***: Campo de texto com placeholder "Digite a quantidade em estoque".
- Garantia**: Menu suspenso com opção "Selecione".
- Modelo**: Campo de texto com placeholder "Digite o modelo do produto".
- Cor**: Menu suspenso com opção "Selecione".
- Material**: Menu suspenso com opção "Selecione".

Existem também ícones para upload de imagem e uma barra de progresso no topo da área principal.

Fonte: Desenvolvido pelo autor (2026)

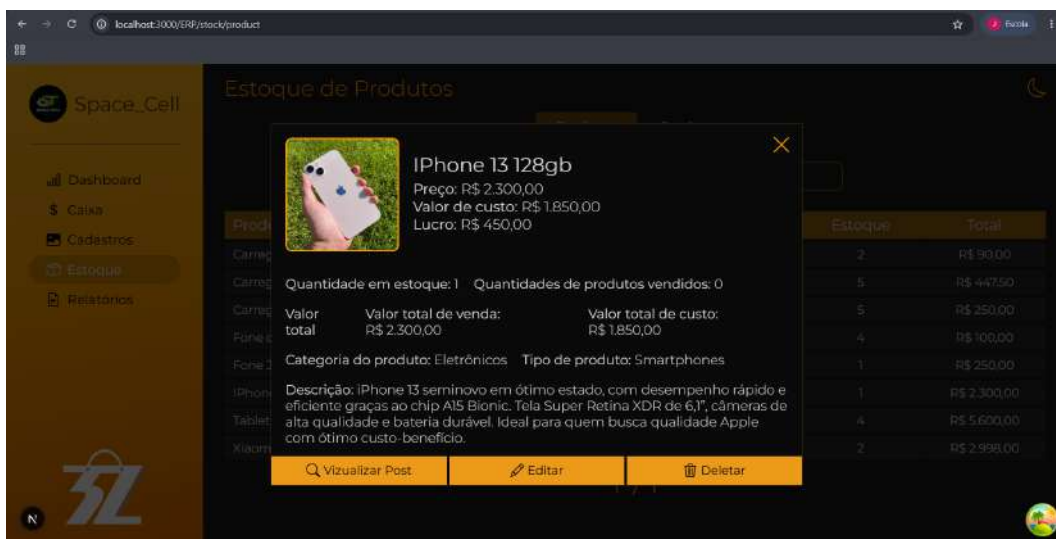
Figura 33 – Tela de estoque de produtos



Produto	Valor	Estoque	Total
Carregador iPhone Kingo	R\$ 45,00	2	R\$ 90,00
Carregador Portátil	R\$ 89,50	5	R\$ 447,50
Carregador Samsung tipo C	R\$ 50,00	5	R\$ 250,00
Fone de ouvido com fio	R\$ 25,00	4	R\$ 100,00
Fone JBL BT520	R\$ 250,00	1	R\$ 250,00
iPhone 13 128gb	R\$ 2.300,00	1	R\$ 2.300,00
Tablet Xiaomi Redmi Pad 256gb	R\$ 1.400,00	4	R\$ 5.600,00
Xiaomi Redmi Note 15	R\$ 1.499,00	2	R\$ 2.998,00

Fonte: Desenvolvido pelo autor (2026)

Figura 34 – Informações sobre o produto empresarial



Produto	Valor	Estoque	Total
Carregador iPhone Kingo	R\$ 45,00	2	R\$ 90,00
Carregador Portátil	R\$ 89,50	5	R\$ 447,50
Carregador Samsung tipo C	R\$ 50,00	5	R\$ 250,00
Fone de ouvido com fio	R\$ 25,00	4	R\$ 100,00
Fone JBL BT520	R\$ 250,00	1	R\$ 250,00
iPhone 13 128gb	R\$ 2.300,00	1	R\$ 2.300,00
Tablet Xiaomi Redmi Pad 256gb	R\$ 1.400,00	4	R\$ 5.600,00
Xiaomi Redmi Note 15	R\$ 1.499,00	2	R\$ 2.998,00

Fonte: Desenvolvido pelo autor (2026)

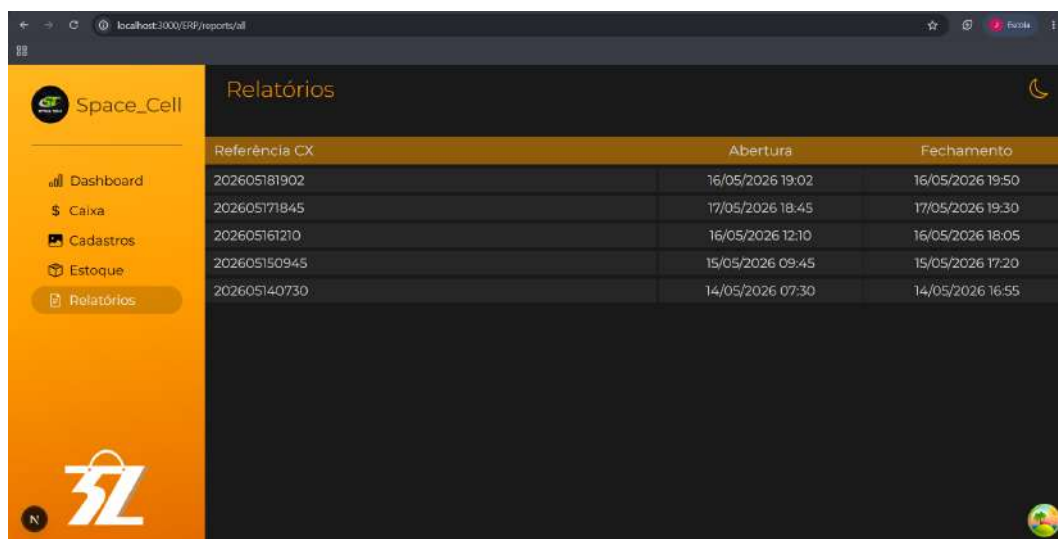
E, para finalizar as principais funcionalidades do sistema de gestão financeira, e mostrado na Figura 35 a área de relatórios gerais, na qual são listados todos os caixas já fechados pelo usuário. Nessa seção, é possível consultá-los de forma geral ou individual, garantindo um melhor controle sobre os faturamentos.

Ao clicar em um caixa específico, é exibido um relatório detalhado em formato PDF referente ao caixa selecionado. Esse relatório segue um modelo semelhante ao apresentado anteriormente na Figura 31, porém adaptado às informações específicas do caixa em questão,

incluindo dados como entradas, saídas, totalizações e demais informações financeiras registradas durante o período de abertura e fechamento do caixa.

Dessa forma, o sistema proporciona maior transparência e confiabilidade na análise dos resultados, além de facilitar o acompanhamento histórico das operações realizadas pelo usuário.

Figura 35 – Relatórios de caixas



Referência CX	Abertura	Fechamento
202605181902	16/05/2026 19:02	16/05/2026 19:50
202605171845	17/05/2026 18:45	17/05/2026 19:30
202605161210	16/05/2026 12:10	16/05/2026 18:05
202605150945	15/05/2026 09:45	15/05/2026 17:20
202605140730	14/05/2026 07:30	14/05/2026 16:55

Fonte: Desenvolvido pelo autor (2026)

6 DISCUSSÃO

Este capítulo apresenta a discussão dos resultados obtidos ao longo do desenvolvimento do sistema 3Z. Inicialmente na Seção 6.1, são analisadas discutidas as informações coletadas por meio das entrevistas realizadas com profissionais autônomos, microempreendedores e pequenos empresários, destacando sua relação com as funcionalidades propostas pelo sistema.

E na Seção 6.2, são discutidos os resultados alcançados com o desenvolvimento do protótipo, avaliando sua capacidade de atender aos objetivos estabelecidos, bem como suas contribuições, limitações e possibilidades de aprimoramento.

6.1 Discussão das entrevistas

Nessa seção são discutidos os resultados obtidos por meio da entrevista com profissionais de diversas áreas. Os dados obtidos são apresentados e analisados a seguir, de acordo com cada pergunta do questionário apresentado na Seção 3.2 e como eles contribuíram para o desenvolvimento do protótipo.

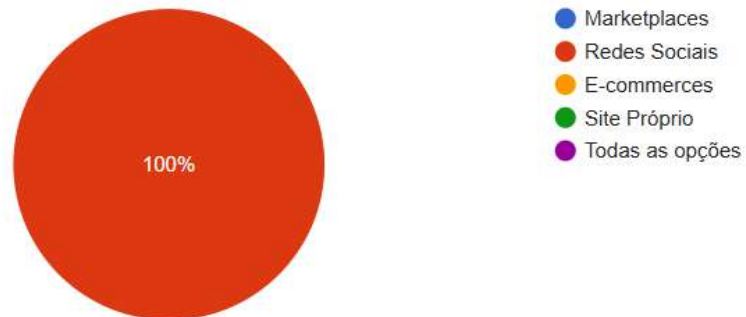
Foram feitos gráficos de pizza para representar visualmente os dados obtidos em cada pergunta do questionário, facilitando a interpretação dos resultados e a identificação das principais tendências observadas entre os entrevistados.

Os dados da primeira Figura 36, mostra que todos os entrevistados utilizam as redes sociais como principal canal de divulgação, evidenciando a relevância da incorporação de elementos desse tipo de aplicação no sistema. Dessa forma, busca-se proporcionar um ambiente digital mais familiar e intuitivo para os usuários.

A incorporação de elementos inspirados nas redes sociais dentro do sistema 3Z contribui para reduzir a curva de aprendizado e facilitar a adaptação dos usuários à plataforma, uma vez que estes já estão habituados a esse tipo de interface em seu cotidiano. Além disso, recursos como publicações, perfis, seguidores e interação entre usuários favorecem a divulgação de produtos e serviços, ampliando a visibilidade dos empreendedores e fortalecendo sua presença no ambiente digital (BARROS, 2023).

Figura 36 – Gráfico de pizza das respostas da pergunta: Qual é o seu principal canal de divulgação online de produtos ou serviços?

12 respostas



Fonte: Desenvolvido pelo autor (2026)

A Figura 37 mostra que os entrevistados utilizam o aplicativo WhatsApp como principal meio de comunicação com seus clientes, evidenciando a relevância dessa ferramenta na interação entre consumidores e empreendedores. Esse resultado reforça a importância da integração do WhatsApp ao sistema 3Z, justificando sua adoção como principal canal de comunicação da plataforma.

Por meio dessa integração, os usuários podem estabelecer contato de forma rápida e direta, facilitando o esclarecimento de dúvidas, negociação e o fortalecimento do relacionamento entre clientes e empreendedores. Além disso, a utilização de uma ferramenta já amplamente difundida reduz barreiras de adaptação e contribui para uma experiência mais prática e acessível aos usuários (SEBRAE, 2021).

Figura 37 – Gráfico de pizza das respostas da pergunta: Qual o principal meio de contato utilizado com seus clientes?

12 respostas



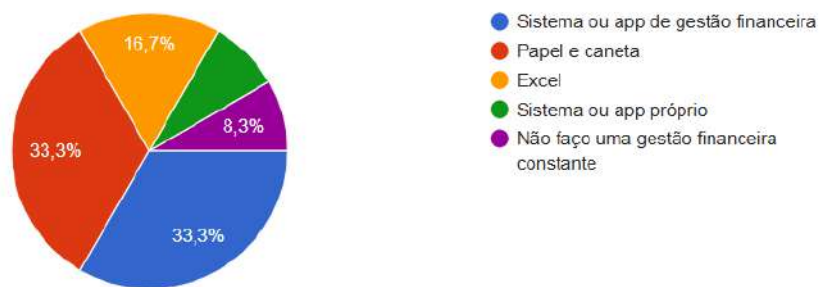
Fonte: Desenvolvido pelo autor (2026)

Nesse ponto na Figura 38, os dados já começam a se mostrar um pouco mais divididos, porém ainda evidenciam aspectos importantes discutidos neste trabalho, como a gestão financeira para esses profissionais. Observa-se também que muitos ainda utilizam papel e caneta para realizar essa gestão, tornando o processo menos eficiente, mais exaustivo e mais propenso a erros.

Nesse contexto, o 3Z pode contribuir ao disponibilizar diversas ferramentas de controle financeiro integradas à plataforma apresentadas na sessão 5.4. Buscando proporcionar uma gestão mais organizada e confiável, reduzindo a necessidade de controles manuais e auxiliando os usuários no acompanhamento de suas atividades financeiras.

Figura 38 – Gráfico de pizza das respostas da pergunta: Como é feita a sua sua gestão financeira?

12 respostas



Fonte: Desenvolvido pelo autor (2026)

Por fim, a última pergunta Figura 39 evidencia um ponto fortemente enfatizado neste trabalho: cerca de 75% dos entrevistados enfrentam dificuldades técnicas e financeiras para utilizar e manter o uso constante dessas ferramentas digitais, o que reforça, mais uma vez, a importância de um sistema com funcionalidades centralizadas nas necessidades desses profissionais.

Figura 39 – Gráfico de pizza das respostas da pergunta: Considerando o uso de ferramentas digitais citadas acima para Marketing, Comunicação e Gestão, qual a sua maior dificuldade em operar com pelo menos uma solução em cada um desses três pilares?

12 respostas



Fonte: Desenvolvido pelo autor (2026)

6.2 Discussão dos resultados

Com base nos resultados apresentados, observa-se que o sistema 3Z, enquanto protótipo funcional, atingiu satisfatoriamente os objetivos propostos no início do trabalho, especialmente no que diz respeito à integração entre funcionalidades sociais e empresariais em uma única plataforma.

Os resultados obtidos por meio da entrevista contribuíram diretamente para a definição das funcionalidades implementadas no sistema 3Z. A utilização unânime das redes sociais como principal meio de divulgação reforçou a ideia de trazer aspectos relacionados para o protótipo, permitindo que usuários possam divulgar produtos e serviços em um ambiente familiar. Da mesma forma, o uso predominante do WhatsApp como ferramenta de comunicação justificou sua integração ao sistema, possibilitando o contato rápido e direto entre clientes e anunciantes por meio de uma plataforma já amplamente utilizada.

Além disso, os resultados relacionados à gestão financeira e às dificuldades enfrentadas no uso de ferramentas digitais influenciaram a criação dos módulos empresariais do sistema. A presença de usuários que ainda realizam controles manuais evidenciou a necessidade de funcionalidades como controle de caixa, movimentações financeiras e geração de relatórios. Já as dificuldades técnicas e financeiras relatadas pelos entrevistados reforçaram a proposta central do 3Z de reunir, em uma única plataforma, recursos de divulgação, comunicação e gestão.

A organização das interfaces mostra preocupação com a usabilidade, mantendo padrões

visuais consistentes e fluxos de navegação intuitivos, o que contribui para uma melhor experiência do usuário. Algumas alterações ocorreram principalmente para melhorar a performance do sistema, simplificar interações e adequar o modelo proposto às limitações técnicas encontradas ao longo do desenvolvimento.

No contexto das funcionalidades sociais, como *feed*, perfis, favoritos, seguidores e comunicação, percebe-se que o protótipo conseguiu promover a interação entre usuários e postagens de forma eficiente, permitindo tanto a visualização quanto o compartilhamento de conteúdos. Já no contexto empresarial, os módulos de gestão financeira, estoque e relatórios mostra uma estrutura mais robusta, capaz de atender às necessidades de controle e organização de dados comerciais.

Outro ponto relevante é a integração entre os módulos, especialmente na relação entre postagens e o sistema de estoque e vendas. Essa conexão permite que informações inseridas em uma parte do sistema sejam refletidas em outras áreas, aumentando a consistência dos dados e reduzindo redundâncias.

Apesar dos resultados positivos, algumas limitações podem ser observadas, como a dependência de bibliotecas externas e a complexidade de determinadas operações durante o desenvolvimento, entre outros aspectos que serão discutidos de forma mais detalhada na Seção 7.1. Tais pontos representam oportunidades de aprimoramento e deverão ser tratados em futuras versões do sistema, contribuindo para sua evolução de um protótipo funcional para uma solução mais completa, robusta e adequada ao uso em cenários reais.

De forma geral, os resultados obtidos indicam que o protótipo do sistema desenvolvido cumpre sua proposta de unir funcionalidades sociais e empresariais em uma única plataforma, oferecendo uma solução integrada, funcional e alinhada aos objetivos estabelecidos ao longo deste trabalho. Contudo, por se tratar de um protótipo, ainda existem possibilidades de aprimoramento de segurança, performance e expansão das funcionalidades em trabalhos futuros.

7 CONSIDERAÇÕES FINAIS

Este capítulo apresenta as considerações finais do trabalho, estando dividido em duas seções. Na Seção 7.1, são apresentados possíveis pontos de melhoria e oportunidades de aprimoramento do sistema, visando contribuir com futuras pesquisas e evoluções do protótipo. Já na Seção 7.2, é apresentada a conclusão final do trabalho, destacando os resultados alcançados e sua relação com os objetivos propostos.

7.1 Trabalhos Futuros

Apesar de o desenvolvimento do protótipo ter ocorrido de forma coerente com o objetivo proposto, diversas melhorias e aprimoramentos ainda podem ser implementados em trabalhos futuros, visando o aumento da robustez, escalabilidade e usabilidade do sistema. Entre essas possíveis evoluções, destacam-se:

- **Segurança.** Apesar do sistema desenvolvido ter sido construído com diversas precauções de segurança, como controle de autenticação via JWT, permissões de acesso tanto no cliente quanto no servidor, intermediação da comunicação entre as diferentes partes do sistema, hash e verificação de senhas, proteção de rotas e páginas privadas, limitação de requisições e preocupação com a exposição de dados sensíveis, *Cross-Origin Resource Sharing* (CORS), entre outras medidas. Em trabalhos futuros, seria interessante a implementação de camadas adicionais de proteção, reduzindo a dependência exclusiva de mecanismos fornecidos por frameworks e bibliotecas externas. Além disso, em um ambiente de produção, seria necessária a utilização de *HyperText Transfer Protocol Secure* (HTTPS) para garantir maior segurança na comunicação entre as diferentes camadas do sistema;
- **Interface de usuário.** O protótipo desenvolvido levou bastante em consideração a experiência de usabilidade do usuário, porém ainda apresenta alguns pequenos *bugs* pontuais que, apesar de não impedir a utilização ou afetar a segurança da aplicação, devem ser corrigidos para proporcionar uma melhor experiência ao usuário final. Além disso, nem todas as telas possuem responsividade, o que deve ser implementado para melhorar a acessibilidade do sistema final.

- **Performance.** Mesmo tendo sido utilizadas medidas de otimização de performance, ainda podem ser integradas diversas outras melhorias, como a implementação de mecanismos de *cache* no *back-end* para redução de consultas repetitivas ao banco de dados, otimização de consultas e relacionamentos para melhorar o desempenho geral do sistema em cenários com maior volume de dados e usuários simultâneos. Adicionalmente, pode-se considerar a separação de algumas tarefas em microserviços independentes para operações de alto custo computacional, o que contribui para a redução da carga no sistema principal e melhora a escalabilidade da aplicação;
- **Funcionalidades.** Funções como avaliações de usuários e postagens, recuperação de senha e notificações para algumas tarefas também seriam interessantes, podendo agregar maior valor ao sistema e ampliar suas funcionalidades. A implementação dessas melhorias permitiria não apenas aumentar a interação entre os usuários, mas também melhorar a experiência geral de uso da plataforma. Além disso, a recuperação de senha contribuiria para maior autonomia do usuário, enquanto o sistema de notificações poderia auxiliar na comunicação de eventos importantes, essas funcionalidades adicionais contribuiriam para tornar o sistema mais dinâmico, interativo e completo, aproximando-o de soluções mais robustas encontradas em aplicações reais no mercado;
- **Controle de usuários e postagens.** Implementação de verificação de usuários por e-mail no momento do cadastro e validação de número de telefone por meio de serviços utilizando mensageria com RabbitMQ. Além disso, pode-se incluir um sistema de denúncias de postagens, permitindo maior controle da comunidade e moderação de conteúdo. Outra possível melhoria seria a utilização de mecanismos automatizados, como inteligência artificial, para análise prévia das postagens antes de sua publicação, contribuindo para a identificação de conteúdos inadequados e aumentando a segurança e confiabilidade da plataforma;
- **Monetização.** Serviços de assinatura mensal com funcionalidades extras aos usuários também podem ser implementados, permitindo a oferta de recursos adicionais mediante planos pagos. Essa abordagem pode contribuir para a sustentabilidade do sistema, viabilizando sua manutenção e evolução contínua por meio de receita recorrente.

7.2 Conclusão

Em suma, o desenvolvimento do sistema 3Z cumpriu com êxito o objetivo de projetar e implementar uma plataforma integrada que unifica de forma eficiente o ecossistema das redes sociais, a dinâmica dos *marketplaces* e a robustez dos sistemas de gestão empresarial. A escolha e articulação das tecnologias modernas utilizando Django no *back-end* para a consolidação de regras de negócio e segurança de dados, e Next no *front-end* para a concepção de interfaces fluidas e reativas provaram-se altamente eficientes para a entrega de um protótipo estável e de uma boa usabilidade.

Pois diante do cenário atual de exclusão digital, dos gargalos financeiros e técnicos que acometem microempresários locais e profissionais autônomos, a plataforma se pode ser uma alternativa centralizada, para impulsionar o comércio regional. Porém mesmo sendo uma alternativa funcional o sistema 3Z atualmente ainda possui algumas melhorias a serem feitas que o impedem de ser considerado um sistema apto a produção, mas pode ser considerado um prototipo eficiente e funcional que pode servir como base para um sistema real.

REFERÊNCIAS

- Agência Sebrae de Notícias. **Digitalização recorde: pequenos negócios no Brasil atingem nível histórico em 2025**. 2025. Disponível em: <https://agenciasebrae.com.br/inovacao-e-tecnologia/digitalizacao-recorde-pequenos-negocios-no-brasil-atingem-nivel-historico-em-2025/>. Acesso em: 04 jun. 2026.
- AMORIM, V.; QUARESMA, M. A criação de confiança através de interfaces digitais: estudo comparativo entre olx e mercado livre. **Ergodesign HCI**, 2022. Disponível em: <https://periodicos.puc-rio.br/index.php/revistaergodesign-hci/article/view/1779>. Acesso em: 13 jun. 2026.
- BARROS, H. M. Transformação digital para mei, micro e pequenas empresas: o papel das ferramentas digitais para a inovação e o desenvolvimento do negócio. **Revista Ibero-Americana de Humanidades, Ciências e Educação**, v. 9, n. 9, p. 4529–4545, 2023.
- BERNERS-LEE, T. **Information Management: A Proposal**. 1989. Disponível em: <https://www.w3.org/History/1989/proposal.html>. Acesso em: 06 jun. 2026.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The Unified Modeling Language User Guide**. 2. ed. Boston: Addison-Wesley, 2005.
- CHEN, P. P.-S. The entity-relationship model: Toward a unified view of data. **ACM Transactions on Database Systems**, v. 1, n. 1, p. 9–36, 1976.
- Django REST Framework Team. **Django REST Framework Documentation**. 2026. Disponível em: <https://www.django-rest-framework.org/>. Acesso em: 11 jun. 2026.
- Django Software Foundation. **Django Documentation**. 2026. Disponível em: <https://docs.djangoproject.com/en/stable/>. Acesso em: 11 jun. 2026.
- Django Software Foundation. **PostgreSQL notes**. 2026. Disponível em: <https://docs.djangoproject.com/en/stable/ref/databases/#postgresql-notes>. Acesso em: 15 jun. 2026.
- Docker Inc. **What is a container?** 2026. Disponível em: <https://docs.docker.com/get-started/docker-concepts/the-basics/what-is-a-container/>. Acesso em: 16 jun. 2026.
- DUARTE, A. R. Metodologia rails: análise da arquitetura model view controller aplicada. Belo Horizonte, 2011.
- FIELDING, R.; NOTTINGHAM, M.; RESCHKE, J. **RFC 9110: HTTP Semantics**. IETF, 2022. Disponível em: <https://www.rfc-editor.org/rfc/rfc9110>. Acesso em: 07 jun. 2026.

KOTLER, P.; KELLER, K. L. **Administração de Marketing**. 14. ed. São Paulo: Pearson Education do Brasil, 2012.

Meta Platforms. **Facebook Marketplace - Help Center**. 2026. Disponível em: https://www.facebook.com/help/1713241952104830/?helpref=popular_topics. Acesso em: 13 jun. 2026.

Microsoft. **TypeScript Documentation**. 2026. Disponível em: <https://www.typescriptlang.org/docs/>. Acesso em: 12 jun. 2026.

Microsoft. **Visual Studio Code Documentation**. 2026. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 16 jun. 2026.

Mozilla Developer Network. **JavaScript Guide**. 2026. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>. Acesso em: 12 jun. 2026.

NOVAIS, R. Investigando os impactos do uso de soluções de gestão de recursos empresariais por micro e pequenos empresas. **Research, Society and Development**, v. 13, n. 8, 2024.

OpenJS Foundation. **Node.js Documentation**. 2026. Disponível em: <https://nodejs.org/en/docs>. Acesso em: 12 jun. 2026.

Python Software Foundation. **General Python FAQ**. 2026. Disponível em: <https://docs.python.org/3/faq/general.html>. Acesso em: 11 jun. 2026.

React Team. **Start a New React Project**. 2026. Disponível em: <https://react.dev/learn/start-a-new-react-project>. Acesso em: 12 jun. 2026.

SEBRAE. **Utilize o WhatsApp Business em seu negócio**. 2021. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/ufs/ms/sebraeaz/utilize-o-whatsapp-business-em-seu-negocio>. Acesso em: 03 jun. 2026.

SEBRAE. **Confira os grandes números dos pequenos negócios no Brasil**. 2025. Disponível em: <https://agenciasebrae.com.br/dados/confira-os-grandes-numeros-dos-pequenos-negocios-no-brasil/>. Acesso em: 03 jun. 2026.

SEBRAE Santa Catarina. **Pesquisa e avaliação de produtos: opinião nas redes sociais**. 2022. Disponível em: <https://www.sebrae-sc.com.br/observatorio/pesquisa-em-redes-sociais/comportamento-do-consumidor>. Acesso em: 04 jun. 2026.

SOUSA, N. A. d.; DANTAS, Y. R. O crescimento do e-commerce durante a pandemia do covid-19. **Revista ADMPG**, v. 13, n. 1, 2023.

SOUSA, V. S. d. Sistemas erp e tomada de decisão: Um estudo sobre a integração de processos nas organizações contemporâneas. **Revista Educação Contemporânea**, v. 3, n. 1, 2026.

TIOBE Software. **TIOBE Index for June 2026**. 2026. Disponível em: <https://www.tiobe.com/tiobe-index/>. Acesso em: 10 jun. 2026.

TORRES, C. **A Bíblia do Marketing Digital**. São Paulo: Novatec, 2018.

Vercel. **Next.js Documentation**. 2026. Disponível em: <https://nextjs.org/docs>. Acesso em: 12 jun. 2026.