

**CENTRO PAULA SOUZA**

**ETEC JOÃO MARIA STEVANATTO**

**Ensino Médio Integrado ao Técnico em Desenvolvimento de  
Sistemas em Período Integral**

**Isabella Pereira Silva**

**Isadora Soliani Lima**

**Jhanaina Sardinha**

**João Henrique Saraiva Gardinali**

**GoLand**

**Itapira - SP**

**2025**

**Isabella Pereira Silva**  
**Isadora Soliani Lima**  
**Jhanaina Sardinha**  
**João Henrique Saraiva Gardinali**

**DESENVOLVIMENTO DE UM SITE PROMOCIONAL DE *LANDING*  
*PAGES***

Trabalho de Conclusão de Curso  
apresentado ao Ensino Médio Integrado  
ao Técnico em Desenvolvimento de  
Sistemas em Período Integral, orientado  
pelo Professor Jonas José Oliveira Duzo,  
como requisito parcial para obtenção do  
título de técnico em Desenvolvimento de  
Sistemas.

**Itapira – SP**

**2025**

Gostaríamos de dedicar nossos esforços para o término deste trabalho a nossos professores orientadores que nos apoiaram e apostaram muito na nossa capacidade, além de nossos amigos, colegas e familiares que tiveram um papel muito importante no desenvolvimento deste projeto.

## **AGRADECIMENTOS**

Dedicamos nossos agradecimentos aos nossos professores, por nos auxiliarem e incentivarem, permitindo uma melhora significativa em nossos processos de desempenho profissional.

Também agradecemos aos amigos e familiares que nos apoiaram e estiveram conosco durante o desenvolvimento acadêmico, obrigado pelas palavras de apoio.

Queremos pontuar nosso reconhecimento a Etec João Maria Stevanatto e ao centro Paula Souza por nos fornecer os materiais suficientes para que fosse possível concluirmos nosso aprendizado junto a este projeto.

Agradecemos ao nosso orientador de TCC, Jonas Duzo, que esteve nos orientando e dedicando sua atenção a resolução de nosso problema para que a conclusão deste projeto fosse possível, pelo aprendizado proporcionado e pela oportunidade de vivenciar um ambiente de trabalho e amizade.

"O maior inimigo do conhecimento não é a ignorância, é a ilusão do conhecimento." – Stephen Hawking

## RESUMO

O projeto foi concebido com a proposta de desenvolver um *template* de *site* para *Landing Pages*, que atenda às necessidades de vendedores *online* que encontram dificuldades na criação de *sites* voltados para a divulgação e venda de seus produtos. Criar páginas de vendas que realmente chamem a atenção e ajudem a converter visitantes em compradores pode ser um desafio para muitos empreendedores e profissionais de vendas, especialmente porque isso muitas vezes exige conhecimentos técnicos específicos. Uma pesquisa da *WebJump* em colaboração com o Brasil do comércio eletrônico apresenta uma escassez de desenvolvedores de *back-end* como o principal desafio. Além disso, um problema enfrentado por vendedores da área digital é contratar profissionais capacitados para a criação de *sites* que apresentem seus produtos e informações de forma convincente, estimulando a compra por parte dos clientes (*E-Commerce* Brasil, 2023).

Nesse cenário, a solução proposta é o desenvolvimento de um *site template* específico para *Landing*. Fornecendo uma ferramenta eficiente, prática e personalizável para vendedores que desejam *sites* de conversão com alta qualidade. Este projeto busca oferecer uma experiência de navegação fluida em diversos dispositivos, atendendo às necessidades de usuários e desenvolvedores, com foco em otimizar tempo e recursos para profissionais que desejam adaptar e implementar novas ideias em um mercado digital em crescimento. Adotaram-se linguagens de programação e *frameworks* para promover um desenvolvimento superior. *JavaScript* foi a principal linguagem usada para criar funcionalidades dinâmicas e interativas na parte do cliente. A continuidade do projeto envolve ampliar suas funcionalidades e melhorar a experiência dos usuários, tornando o *site* mais completo e interativo.

**Palavras-chave:** Site; Landing Pages; Comércio eletrônico; Loja virtual.

## ABSTRACT

The project was conceived with the purpose of developing a website template for Landing Pages that meets the needs of online sellers who struggle to create sites focused on promoting and selling their products. Creating sales pages that truly capture attention and help convert visitors into buyers can be challenging for many entrepreneurs and sales professionals, especially because it often requires specific technical knowledge. A study by WebJump in collaboration with E-Commerce Brasil highlights a shortage of back-end developers as a major challenge. In addition, a common issue faced by digital sellers is hiring qualified professionals to create websites that present their products and information convincingly, encouraging customers to make a purchase (E-Commerce Brasil, 2023). In this scenario, the proposed solution is the development of a specific template website for Landing Pages, providing an efficient, practical, and customizable tool for sellers who seek high-quality conversion-focused sites. This project aims to deliver a smooth browsing experience across multiple devices, meeting the needs of both users and developers, with a focus on optimizing time and resources for professionals who want to adapt and implement new ideas in a growing digital market. Programming languages and frameworks were adopted to ensure high-quality development. JavaScript was the main language used to create dynamic and interactive client-side functionalities. The continuation of the project involves expanding its features and improving the user experience, making the website more complete and interactive.

**Tags:** Website; E-commerce; Furniture store; Application.





## SUMÁRIO

1	INTRODUÇÃO	7
1.1	Problema	7
1.2	Justificativas	7
1.3	Objetivo Geral	7
1.4	Objetivos Específicos	7
2	REFERENCIAL TEÓRICO	8
2.1	E-commerce	8
2.2	Lojas de Móveis	8
2.3	Linguagens de Programação WEB	8
3	METODOLOGIA	9
3.1	Objetivo da Pesquisa	9
3.2	Tipo de Pesquisa	9
4	RESULTADOS E DISCUSSÃO	10
4.1	Análise das Necessidades do Cliente	10
4.2	Conceito Gráfico	10
4.3	Desenvolvimento	10
4.4	Publicação	10
4.5	Prototipagem	10
4.6	Aplicativos Semelhantes	11
4.7	Análise de Teste de Software	11
5	CONSIDERAÇÕES FINAIS	12
	REFERÊNCIAS	13
	APÊNDICES	14
	ANEXOS	15

## 1 INTRODUÇÃO

Na era atual, a *internet* é uma ferramenta fundamental para os negócios, permitindo uma conexão rápida e acesso a pessoas de todo o mundo. Com o crescimento do *marketing digital*, as *landing pages*, ou páginas de captura, se destacaram como uma das estratégias mais eficazes para converter visitantes em clientes e atrair *leads* qualificados. No entanto, um problema sobre a quantidade de *sites* fraudulentos e golpes digitais se intensifica. Muitos usuários têm dificuldade em diferenciar ofertas legítimas e páginas criadas para enganar ou explorar (Brunner digital, 2024).

Neste contexto, uma *Landing Page* é definida como páginas criadas para preencher a atenção do visitante em uma ação específica, como comprar um produto, preencher um formulário ou entrar em contato com o proprietário da página (Plotze, 2023). Essencialmente, é uma página de conversão, cujo *design* é focado em eliminar distrações e guiar o usuário para o objetivo único da campanha.

O projeto foi concebido com a proposta de desenvolver um *template* de *site* para *Landing Pages*, que atenda às necessidades de vendedores *online* que encontram dificuldades na criação de *sites* voltados para a divulgação e venda de seus produtos. Criar páginas de vendas que realmente chamem a atenção e ajudem a converter visitantes em compradores pode ser um desafio para muitos empreendedores e profissionais de vendas, especialmente porque isso muitas vezes exige conhecimentos técnicos específicos.

Pensando nisso, foi feita proposta de desenvolver um *template*, que é um *layout* pré-definido e pronto para ser usado e personalizado sem necessidade de criar só zero, que permite que esses profissionais apresentem seus infoprodutos de forma visualmente atraente e otimizada para conversão, seguindo as melhores práticas do mercado, facilitando a personalização. Assim, mesmo quem não tem experiência em *design* ou programação poderá adaptar o *template* às suas necessidades. Dessa maneira, queremos democratizar o acesso à criação de páginas de vendas eficazes e contribuir para o sucesso dos vendedores *online*, oferecendo uma ferramenta poderosa que os ajude a promover seus produtos e serviços de maneira autônoma e eficiente.

*Landing pages* também são conhecidas como páginas de conversão. Nos dias atuais, essas páginas são peças-chave em campanhas de otimização de *marketing online*, uma vez que chamam a atenção de visitantes e despertam sua curiosidade, convertendo-os em clientes em potencial, em sua estrutura de desenvolvimento, e incentivam as ações de um segmento específico, considerando-as como estratégias de *marketing* (Silva, 2023).

## **1.1 Problema**

Pesquisas recentes mostram a dificuldade de empreendedores digitais em encontrar profissionais qualificados para criar *sites*. Uma pesquisa da *WebJump* em colaboração com o Brasil do comércio eletrônico apresenta uma escassez de desenvolvedores de *back-end* como o principal desafio, com 55% de escassez de especialistas. Essa falta de profissionais qualificados não só eleva o custo, mas também prolonga o tempo de desenvolvimento, impedindo que pequenos e médios empreendedores reajam rapidamente às demandas do mercado. A consequência direta é a perda de oportunidades de conversão, pois páginas mal elaboradas ou genéricas de marketplaces falham em construir confiança e autoridade ética, elementos cruciais para o sucesso no comércio eletrônico. Além disso, um problema enfrentado por vendedores da área digital é contratar profissionais capacitados para a criação de *sites* que apresentem seus produtos e informações de forma convincente, estimulando a compra por parte dos clientes (*E-Commerce Brasil*, 2023).

Outro estudo realizado pelo Sebrae em colaboração com a UNCTAD e Cepal, duas importantes organizações internacionais ligadas às Nações Unidas, destacou que 55% dos pequenos empreendedores veem uma escassez de pessoal qualificado como o maior obstáculo às admissões digitais para seus negócios (Sebrae Agency, 2022). Estes dados aprimoram a relevância deste projeto.

## **1.2 Justificativas**

Para o sucesso de um negócio, especialmente no comércio eletrônico, é fundamental estabelecer uma presença online. No entanto, um desafio que os

vendedores *online*, especialmente os de pequeno e médio porte, enfrentam é a construção de *sites* eficazes que apresentem seus produtos de forma atraente e funcional. Além disso, a falta de recursos técnicos e de conhecimento contribui para a formação de ambientes virtuais estratégicos com desempenho abaixo do esperado. Outro aspecto relevante é a escassez de plataformas simples e acessíveis que auxiliam esses empreendedores a mostrar seus produtos. Muitas pessoas utilizam redes sociais ou *marketplaces*, que são plataformas *online*s que funcionam como um ‘*Shopping* virtual’, reunindo diversos vendedores em um único ambiente digital, que nem sempre proporcionam o nível de personalização ou otimização necessário para atender às suas necessidades.

Nesse cenário, a solução proposta neste trabalho é o desenvolvimento de um *site* template específico para *Landing*. Fornecendo uma ferramenta eficiente, prática e personalizável para vendedores que desejam sites de conversão com alta qualidade.

Além de servir como uma ponte entre um vendedor e o desenvolvedor, oferecendo um espaço claro, direto e objetivo; visando reduzir o tempo, o custo e os obstáculos técnicos que dificultam a presença digital desses profissionais.

Este projeto é importante por democratizar o acesso a uma infraestrutura digital de alta performance, atualmente limitada pelo uso de plataformas simples que comprometem a personalização, coleta de dados e construção de marca. A solução proposta é um template de site para Landing Pages, desenvolvido com Django/Python, que oferece uma ferramenta segura, prática e visualmente profissional, permitindo que vendedores tenham um ambiente confiável para aumentar vendas sem precisar de um desenvolvedor especializado.

### **1.3 Objetivo Geral**

Este projeto busca propor e desenvolver uma plataforma/ferramenta digital que ofereça a vendedores e profissionais do mercado digital um ambiente seguro e de fácil utilização para a criação ágil e acessível de páginas de vendas responsivas, otimizando o tempo de desenvolvimento e garantindo uma apresentação visualmente confortável e profissional dos produtos.

### 1.4 Objetivos Específicos

As metas a serem atingidas com nosso projeto são:

Criar *Landing Pages* que se adaptem às necessidades dos usuários, baseado em pesquisas feitas com vendedores digitais, onde foi coletado as maiores dificuldades e necessidades dos mesmos, e adaptadas a este projeto.

Oferecer uma plataforma com o intuito de criar *sites* atrativos, que sejam eficazes na conversão de visitantes em compradores.

Desenvolver *templates* para diversas áreas, como cursos, *e-books* e consultorias.

Agilizar o processo de criação de *sites*, permitindo que os vendedores foquem na promoção e venda de seus produtos.

## 2 REFERENCIAL TEÓRICO

### 2.1 *Landing pages*

*Landing pages* são como vitrines profissionais, páginas criadas para prender a atenção do visitante em uma ação específica, como comprar um produto, preencher um formulário ou entrar em contato com o proprietário da página. Na construção de *landing pages*, é necessário uma série de conhecimentos. No ponto de vista técnico, precisa-se conhecer os fundamentos do desenvolvimento *web*, utilizando linguagens como *HyperText Markup Language* (HTML) e *Cascading Style Sheets* (CSS) para estilizar e estruturar os conteúdos. A escuta ativa e o levantamento de requisitos são partes fundamentais para entender os objetivos da página e transformá-los em elementos visuais e funcionais (Plotze, 2023).

Referindo-se a *landing pages*, o uso da aplicação de ramificação pode ser melhor para a experiência do visitante, deixando-a mais interessante e participativa. Usando elementos de jogos, como desafios, conquistas e recompensas, é possível criar uma conexão mais forte com o visitante, incentivando-o a interagir mais com a página e aumentando as chances de conversão da mesma (Oliveira, 2018).

A experiência do usuário é muito importante nas estratégias de vendas, especialmente em *landing pages*. Reestruturar essas páginas passou a ser uma necessidade estratégica. *Landing pages* otimizadas, organizadas e com um *design* agradável, ajudam a guiar o usuário, impactando diretamente no aumento de engajamento e das vendas. São consideradas algumas métricas como tempo médio na página, taxa de rejeição e taxa de conversão. Páginas lentas, com excesso de elementos ou *layout* confuso podem desestimular a navegação (Bonelli, 2023).

Uma maneira para criar uma *landing page* é adotar o método de desenvolvimento de *software* em cascata (*Waterfall*), que é constituído em etapas sequenciais com objetivos específicos. Inicialmente, tem a fase de comunicação, com uma coleta de informações com potenciais usuários. Em seguida, na fase de planejamento, é definido o cronograma e distribuído às atividades. Na etapa de modelagem, é definido os requisitos funcionais e não funcionais e projetado a arquitetura do sistema. E na última fase, a construção, ocorre a codificação da

aplicação usando as tecnologias selecionadas, seguida pela implantação, que envolve testes, validação e a disponibilização do sistema para uso (Turnip, 2020).

A criação e uso de *Landing Pages* tem muitas vantagens para o vendedor digital, entre elas temos: *Leads*: a *landing page* é a forma mais eficiente de captar *leads*. Começando o processo de qualificação para que, depois, possam ser atendidos pela equipe de vendas;

Coleta de Informações: através de formulários, é possível coletar dados sobre o visitante, permitindo filtrar melhor o público e adequar as abordagens de vendas;

Acompanhamento de Dados: O desempenho das *landing pages* pode ser monitorado para entender o nível de engajamento dos visitantes, possibilitando ajustes e otimizações constantes;

Eliminação de Distrações: Ao contrário de *websites* tradicionais, as *landing pages* eliminam elementos que possam desviar a atenção, concentrando-se em apenas um objetivo;

Testes e Otimização: É possível realizar testes A/B nas *landing pages* para experimentar diferentes versões e identificar quais elementos geram melhores resultados (Meesala, 2022).

## 2.2 Banco de Dados

Os Sistemas Gerenciadores de Banco de Dados (SGBDs) surgiram por volta de 1960 e evoluíram a partir dos primeiros sistemas de arquivos, oferecendo funcionalidades como controle de concorrência, segurança, recuperação de falhas e integridade dos dados. Um ponto importante dos SGBDs é o controle das transações, que precisam seguir as propriedades Atomicidade, Consistência, Isolamento e Durabilidade (ACID) para garantir que as operações sejam confiáveis. O modelo relacional, criado por Edgar Codd em 1970, se tornou a forma mais comum de organizar os dados, usando tabelas conectadas por relações. No entanto, com o crescimento rápido de dados causado pela *Web 2.0* e pelo *Big Data*, esse modelo começou a mostrar suas limitações, principalmente na hora de escalar e se



adaptar às mudanças. Por isso, surgiram os bancos de dados NoSQL, que conseguem lidar com grandes volumes de dados semi estruturados ou não estruturados. Eles oferecem vantagens como escalabilidade horizontal, flexibilidade na estrutura dos dados, replicação nativa e maior disponibilidade para aplicações distribuídas, especialmente em ambientes *web* (Garcia, 2019).

A modelagem de banco de dados é o processo da criação de uma representação visual e conceitual das estruturas e relacionamentos dos dados em um banco de dados. A proposta será definir o que deve ser armazenado sem se preocupar com detalhes técnicos. Propósito: Implementar o modelo lógico em um banco de dados específico, considerando as características do SGBD. Definição: A modelagem de banco de dados é crucial para garantir que o banco de dados seja eficiente e atenda às necessidades do sistema. Este trabalho visa criar uma base sólida através de um modelo lógico que servirá de referência para a implementação técnica no SGBD (Fávero, 2017).

No caso dos bancos de dados, os principais ativos que precisam ser protegidos são as informações armazenadas. Por isso, a segurança está diretamente relacionada à proteção desses dados e ao controle sobre quem pode acessá-los. A segurança, então, visa minimizar essas ameaças e, caso ocorra algum incidente, reduzir os prejuízos e garantir a recuperação rápida do sistema e baseia-se em três princípios essenciais: integridade, disponibilidade e confidencialidade.

Integridade: garante que as informações armazenadas não sejam alteradas indevidamente por alguém sem permissão, evitando modificações não autorizadas, exclusões ou inserções incorretas.

Disponibilidade: assegura que os dados estejam sempre acessíveis aos usuários autorizados, garantindo que os sistemas continuem operando mesmo diante de falhas ou tentativas de ataques.

Confidencialidade: busca proteger os dados contra acessos não autorizados, evitando que informações sensíveis sejam divulgadas para pessoas ou grupos não autorizados (David Luiz, 2021).

Banco de dados são bens de valores imensuráveis às empresas atualmente, pois podem armazenar informações de produtos, serviços, colaboradores,

clientes, dados financeiros, fiscais e muitos outros de diversos tipos e finalidades, que caso perdidos ou acessados indevidamente podem representar prejuízos e quebras legais e de contratos que podem até mesmo levar a empresa à falência. Atualmente os bancos de dados podem ser relacionais e não relacionais. O banco de dados relacional armazena dados em tabelas, o que se tornou uma limitação em grandes quantidades de dados, causando o surgimento dos bancos não relacionais. Podem ser divididos em quatro diferentes tipos: bancos de dados de valor chave, de documentos, armazenamentos de famílias de colunas, e bancos de dados gráficos (Basílio, 2022).

No contexto deste projeto de Pesquisa Aplicada, o Sistema Gerenciador de Banco de Dados (SGBD) escolhido é o *SQLite*. A escolha justifica-se pela sua robustez e por se integrar nativamente ao *framework* Django, que oferece um sistema de mapeamento ORM (Object-Relational Mapping).

Sudha Nagesh (2018) faz uma revisão detalhada sobre as técnicas de mineração de dados usadas em redes de sensores sem fio, conhecidas como WSNs. Ele destaca os principais desafios relacionados ao armazenamento, processamento e gerenciamento do grande volume de dados que essas redes geram. As WSNs produzem um fluxo contínuo de informações que são dinâmicas, não estruturadas e bastante volumosas, o que exige soluções específicas de bancos de dados e mineração de dados. O texto mostra que os bancos de dados tradicionais, que trabalham com dados estáticos e centralizados, não são adequados para lidar com a natureza dessas redes, que precisam de arquiteturas distribuídas e processamento em tempo real. Entre as principais técnicas de mineração de dados abordadas estão o *Frequent Pattern Mining*, que identifica padrões frequentes nos fluxos de dados das redes WSNs; *Sequential Pattern Mining*, que busca descobrir sequências de eventos que ocorrem em uma determinada ordem; *Clustering* que é um método que organiza os dados ou os sensores em grupos, levando em conta as semelhanças entre eles, e *Classification*, que usa algoritmos, como árvores de decisão, regras ou máquinas de vetor de suporte (SVM), para categorizar os dados que foram coletados (Sudha; Nagesh, 2018).

## 2.3 Linguagens e Frameworks

*Javascript* é a linguagem da *web*. Ela começou como uma forma de manipular alguns tipos de elementos selecionados em uma página *web* (como imagens ou campos de formulários), mas acabou crescendo de maneira significativa. Além de servir como *script* de navegador no lado do cliente, atualmente você pode usar o *Javascript* para programar em uma variedade cada vez maior de plataformas. É possível escrever códigos no lado do servidor (usando NET ou *Node.js*), aplicações *desktop* e extensões de aplicação, aplicações para dispositivos móveis e *scripts* de linha de comando. O *Javascript* também é uma linguagem incomum. Não possui classes, e as funções são usadas como objetos de primeira classe em várias tarefas. Inicialmente, a linguagem foi considerada deficiente por vários desenvolvedores, mas nos últimos anos esse sentimento tem mudado. Curiosamente, linguagens como *Java* e *PHP* começaram a adicionar funcionalidades como closures e funções anônimas, que os desenvolvedores *Java Script* vêm utilizando corriqueiramente há algum tempo. O *Javascript* é suficientemente dinâmico a ponto de ser possível fazê-lo parecer com qualquer outra linguagem com a qual você se sinta confortável (Stefanov, 2011).

O *Css* é uma ferramenta de desenvolvimento *web* que ajuda a adicionar cores, estilos de *design* e um toque pessoal aos sites. Os métodos de *CSS* mais comuns são os de cor, dimensionamento e posicionamento. Existem também métodos de *CSS* mais avançados que podem ajudar a estilizar cada parte de uma página da *web*. Existem três maneiras de implementar o *CSS* durante a criação de uma página *web*: *inline* (em linha), *internal* (interno) ou *external* (externo). O *CSS* pode ter vários *frameworks*, esses *frameworks* ajudam bastante os desenvolvedores durante o processo de criação, oferecendo uma base pré-definida de estilos e componentes. Isso torna o trabalho mais rápido, especialmente em projetos maiores ou equipes que precisam manter um padrão visual consistente. Entre os *frameworks* mais conhecidos estão o *Bootstrap*, o *Tailwind CSS* e o *Foundation*. Além dos *frameworks*, o *CSS* moderno trouxe muitas novidades, como variáveis, media queries para criar *layouts* responsivos, animações e até recursos como *grid* e *flexbox*. Essas ferramentas permitem controlar o posicionamento dos elementos com

muito mais precisão e flexibilidade, facilitando a criação de interfaces cada vez mais sofisticadas e adaptáveis (Mohd, 2022).

O HTML, que é a linguagem usada para criar páginas na internet, evoluiu bastante desde sua criação. Atualmente, o padrão mais utilizado é suportado por todos os navegadores modernos é o HTML5. Ele continua se desenvolvendo, com a possibilidade de surgir novos elementos e atributos que tornam o desenvolvimento mais fácil e deixam as páginas mais semânticas, ou seja, mais claras sobre o conteúdo que apresentam. No futuro, espera-se que haja ainda mais recursos voltados para melhorar a interação dos usuários com as páginas e tornar tudo mais acessível a todos. Por exemplo, novos formatos de vídeo e áudio poderiam melhorar a qualidade de reprodução e reduzir a carga na rede, o que é especialmente importante para usuários móveis.

A segurança das aplicações *web* está se tornando cada vez mais crítica. Novos padrões HTML incluirão mecanismos aprimorados de proteção contra ataques como *Cross-Site Scripting* (XSS) e *Cross-Site Request Forgery* (CSRF). Isso tornará as aplicações *web* mais confiáveis e seguras (NetCoders, 2024).

*Python* é uma linguagem de programação interpretada, orientada a objetos, de alto nível, além de apresentar tipagem dinâmica, execução eficiente e uma sintaxe simples, o que contribui para sua ampla aceitação entre desenvolvedores de diferentes níveis de experiência. Ao longo dos anos, *Python* tornou-se uma das linguagens mais utilizadas, em grande parte devido à extensa variedade de bibliotecas e *frameworks* disponíveis, os quais ampliam significativamente suas funções de uso. Na área da computação científica, destaca-se pelo suporte que oferece ferramentas avançadas. Essas bibliotecas possibilitam a realização de cálculos numéricos, processamento de sinais, construção de gráficos, manipulação simbólica de expressões matemáticas (*Computer Algebra System*), análise estatística, entre outras funções essenciais em contextos científicos e acadêmicos. Além da computação científica, *Python* é utilizado em diversas outras áreas, como automação de tarefas, desenvolvimento de sistemas, ciência de dados, inteligência artificial, e até mesmo em aplicações multimídia (Oliveira, 2016).

Django é um *framework* de desenvolvimento *web* usando *Python*. Ele é conhecido por ser forte, versátil e por seguir o princípio de evitar repetir código, o

que ajuda a deixar o sistema mais organizado e fácil de manter. Sua estrutura é baseada no modelo *Model-View-Controller* (MVC), o que significa que separa bem as funções, tornando o desenvolvimento mais claro. Uma das maiores vantagens do *Django* é o seu Object-Relational Mapping (ORM), que simplifica o trabalho com bancos de dados, permitindo que você manipule os dados usando classes em *Python*, sem precisar lidar diretamente com comandos complexos de SQL. Esse recurso facilita o acesso e a manutenção de dados, tornando tudo mais fácil de entender e de usar em diferentes sistemas de gerenciamento de banco de dados. Além disso, o *framework* oferece várias ferramentas integradas, como autenticação de usuários, recursos de segurança, opções de escalabilidade e a possibilidade de criar componentes reutilizáveis, chamados de apps. O painel administrativo do *Django* também é um destaque, pois permite gerenciar os dados de forma prática, com operações automáticas de criar, ler, atualizar e excluir informações (conhecidas como CRUD). Essas características fazem do *Django* uma opção consolidada e adequada para aplicações modernas que precisam de segurança, crescimento fácil e desenvolvimento ágil. Por isso, ele é bastante usado em projetos que vão desde pequenos sistemas até plataformas maiores (Chen, 2020).

### 3 METODOLOGIA

A pesquisa bibliográfica é um procedimento metodológico que utiliza materiais já publicados, como livros, artigos, teses e fontes audiovisuais, como base para a investigação. Diferente da revisão de literatura, que é uma etapa preliminar, a pesquisa bibliográfica tem como objetivo principal a análise crítica e a síntese dessas fontes, visando aprofundar o entendimento sobre determinado tema. Ela permite acesso a uma variedade de fontes e contextos que seriam difíceis de obter por pesquisa de campo, facilitando o diálogo com o conhecimento já existente e evitando redundâncias. Além de fornecer uma base sólida para compreender o tema, a pesquisa bibliográfica busca gerar novas abordagens, interpretando e criticando as fontes para ampliar ou redefinir o entendimento do assunto estudado, contribuindo de forma original para o debate acadêmico (Dos Santos, 2021).

Segundo Boccato (2006), uma pesquisa bibliográfica busca a resolução de um problema, por meio de referenciais teóricos publicados, analisando e discutindo contribuições científicas variadas. Esse tipo de pesquisa oferece informações importantes para ampliar o conhecimento sobre o que foi pesquisado, como e qual o enfoque e perspectiva trata o assunto apresentado na literatura científica. Para isso, é muito importante que o pesquisador utilize de um planejamento no processo de pesquisa, assim compreendendo desde sua definição, até a construção lógica do trabalho, e dessa forma também, a definição da forma que realizará a comunicação e divulgação (Boccato, 2006).

As pesquisas de referência foram feitas usando sites, artigos e documentos digitais, essenciais para a elaboração do projeto, como o google acadêmico e a Biblioteca Digital Brasileira de Teses e Dissertações (BDTD). Foram utilizadas ferramentas online para localizar artigos de qualidade e sites de pesquisa que ajudassem a aprofundar o tema. A escolha cuidadosa das fontes garantiu que as informações fossem atuais e relevantes, formando uma base teórica sólida para o desenvolvimento do trabalho. Além disso, foram consideradas publicações acadêmicas, bases de dados especializadas e jornais científicos reconhecidos, que ajudaram a ampliar o entendimento sobre o assunto.

A metodologia deste projeto é de Pesquisa Aplicada, pois busca desenvolver uma solução prática: uma plataforma para criar páginas de vendas. Diferente da pesquisa básica, que visa conhecimento teórico, o objetivo aqui é interferir diretamente na problemática de vendedores e profissionais digitais, oferecendo uma ferramenta que economiza tempo e recursos na criação de seus ambientes digitais.

A abordagem metodológica adotada para o desenvolvimento do projeto baseia-se na utilização do *framework Django*, que oferece uma estrutura integrada para o gerenciamento de banco de dados em aplicações *web*. O *Django* proporciona um sistema de mapeamento ORM, facilitando a manipulação e a interação com o banco de dados por meio de modelos que representam as entidades da aplicação. Os procedimentos metodológicos envolvem a escolha dos modelos, a configuração das migrações para criar e atualizar as tabelas, além de implementar as operações de consulta, inserção, atualização e exclusão de dados. Essa abordagem tem como objetivo garantir que o banco de dados seja seguro, eficiente e capaz de crescer conforme a necessidade, tornando o sistema mais sólido e confiável. Além disso, esses procedimentos incluem uma seleção cuidadosa das bases de dados, a definição dos critérios para buscas e filtros, e o uso de técnicas específicas para tratar e analisar os dados coletados.

Adotaram-se linguagens de programação e *frameworks* para promover um desenvolvimento superior. *JavaScript* foi a principal linguagem usada para criar funcionalidades dinâmicas e interativas na parte do cliente. Sua flexibilidade, além de poder ser aplicada em diferentes ambientes, sites, aplicativos de *desktop* e até servidores, fazem dela uma ferramenta fundamental para projetos que precisam de agilidade e adaptação. O CSS foi utilizado para definir o visual da interface, ajudando a criar uma aparência consistente e que se adapte bem a diferentes tamanhos de tela. O HTML serviu como a estrutura básica da aplicação *web*, organizando e dando sentido ao conteúdo que o usuário vê. Na utilização da linguagem de programação *Python* foi incluído no projeto para ajudar no processamento de dados e na automação de tarefas. Com uma sintaxe simples, ele permitiu criar soluções eficientes, especialmente na manipulação de dados, análise estatística e suporte ao funcionamento do sistema. Complementando, foi utilizado o *Django*, um *framework* em *Python* conhecido pela sua estabilidade e pelo princípio *Don't Repeat Yourself*

(DRY), que ajuda a evitar repetições e torna a manutenção mais fácil. Ele simplifica o trabalho com bancos de dados. Também oferece recursos integrados de segurança, autenticação e um painel administrativo com funções de criar, ler, atualizar e excluir, sendo uma ótima escolha para sistemas de diferentes tamanhos.

## 4 RESULTADOS E DISCUSSÃO

Este capítulo refere-se à apresentação dos resultados do trabalho e à comparação entre o que já foi alcançado na pesquisa, com os estudos já mencionados no referencial teórico, promovendo argumentações dos resultados obtidos na etapa de desenvolvimento do projeto.

### 4.1 Conceito Gráfico

Neste tópico, apresentamos os primeiros protótipos e o conceito gráfico que deram norte ao desenvolvimento deste projeto.

A seguir, imagem referente ao nome e logo do projeto.

**Figura 1 – Design da marca**



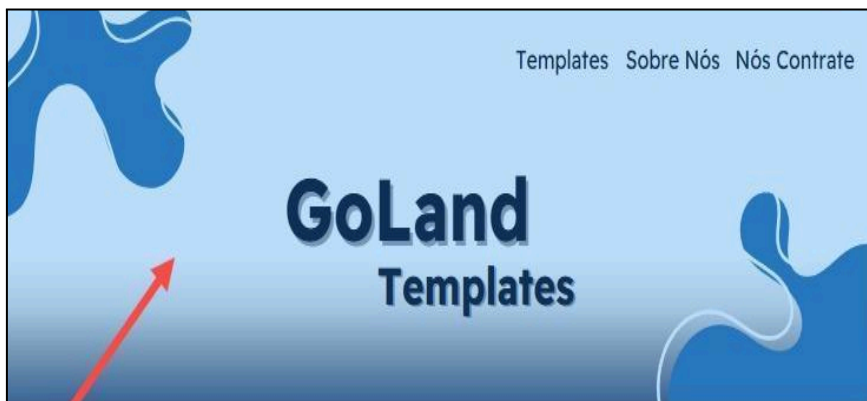
Fonte: Autoria própria

O título e seu *design* foi pensado para criar uma identidade visual única que se destacasse, andando lado a lado com a ideia oferecida do projeto, se diferenciando dos demais sites com a mesma proposta, algo curto, simples mas impactante. Mantendo uma paleta de cores que remetesse à tecnologia e ao *marketing*.



A seguir imagem do primeiro design da tela de menu do projeto.

**Figura 2 – Tela de menu**

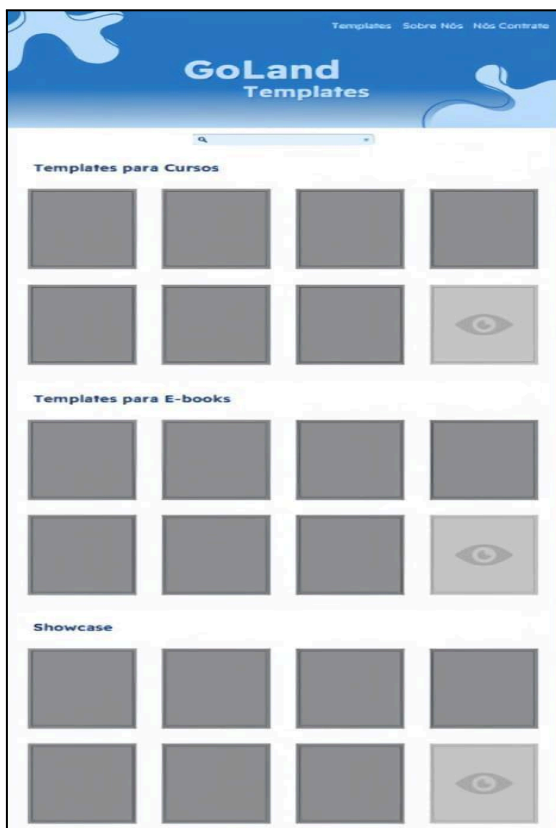


Fonte: Autoria própria

A tela de menu é essencial para transmitir ao usuário a identidade visual do site e a proposta da aplicação, inicialmente buscamos um equilíbrio entre funcionalidade e uma estética tecnológica. O menu do site acabou por ser substituído por outro modelo que passasse melhor um ar de organização e respeitabilidade, utilizando uma paleta de cores harmônica que reforçasse a identidade visual e contribuísse para a experiência do usuário; com elementos projetados para guiar e ajudar o usuário durante a navegação.

A seguir imagem relacionada ao design inicial da tela de templates do site.

**Figura 3 – Tela de templates**

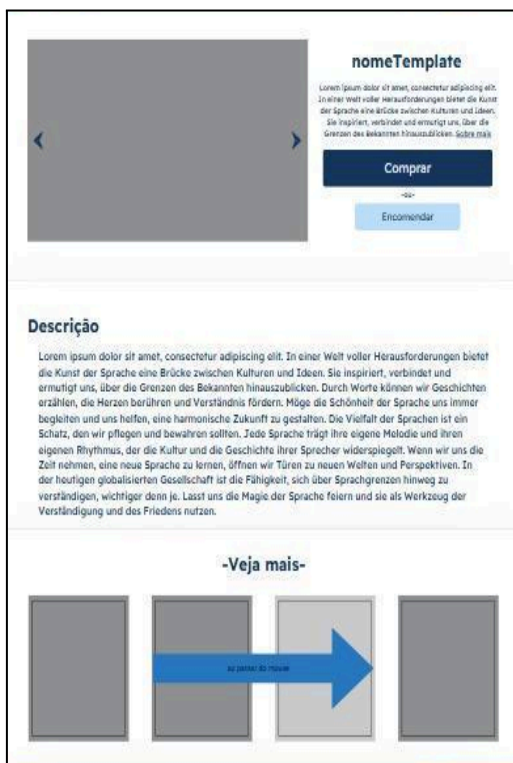


Fonte: Autoria própria.

A tela de templates é o primeiro contato do cliente com os produtos oferecidos no site, é onde o usuário tem acesso às opções de ebooks e cursos, dos quais ele pode obter ou visualizar para realizar a compra e buscar por um modelo específico.

A figura abaixo apresenta o primeiro design da tela de visualização e compra do template.

**Figura 4 – Tela de acesso/visualização ao template**



Fonte: Autoria própria.

A tela onde o usuário/cliente tem acesso ao template escolhido por ele, de acordo com o que deseja buscar. Além disso, pode ter uma visualização da descrição do template. E logo abaixo, é exibido mais modelos de templates como o que o usuário está visualizando, para que ele possa ter mais opções.

A imagem á seguir, mostra o design inicial da tela referente aos desenvolvedores do projeto.

**Figura 5 – Página sobre nós**



Fonte: Autoria própria.

Esta página contém informações “sobre nós”, para que o cliente possa conhecer melhor os profissionais responsáveis pelo site, e também contém o email e o telefone, dando ao usuário a opção de entrar em contato conosco. Esta funciona como um canal direto de comunicação entre donos do site e seus visitantes, gerando credibilidade, facilitando o contato para dúvidas, e criando uma ponte de confiança e transparência.

A seguir imagem referente ao primeiro protótipo da tela de contato com os desenvolvedores do projeto.

**Figura 6 – Página de Contato**

Templates Sobre Nós Nós Contrate

# GoLand

Nós Contrate

## Para que nós contratar?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. In einer Welt voller Herausforderungen bietet die Kunst der Sprache eine Brücke zwischen Kulturen und Ideen. Sie inspiriert, verbindet und ermutigt uns, über die Grenzen des Bekannten hinauszublicken. Durch Worte können wir Geschichten erzählen, die Herzen berühren und Verständnis fördern. Möge die Schönheit der Sprache uns immer begleiten und uns helfen, eine harmonische Zukunft zu gestalten.

## Entre em contato conosco!

Nome:  Telefone:

Email:

Mensagem:

Enviar

### Fale Conosco

(99) 9-9999-9999  
nomesite@gmail.com

- Templates
- Sobre Nós
- Nós Contrate

### Veja resultados reais

Cursos E-books Todos

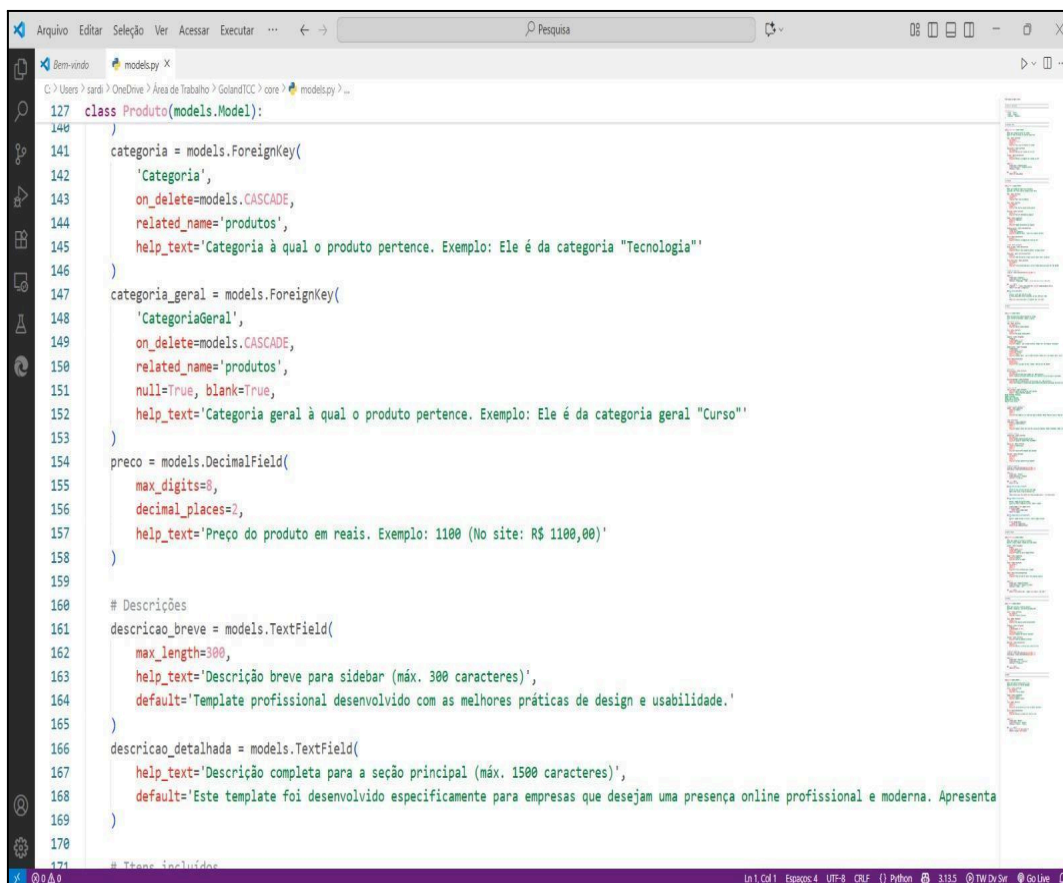
Fonte: Autoria própria.

Nesta página o usuário, se caso desejar, pode entrar em contato com os criadores do projeto. E além disso, é uma página que informa o cliente a importância de contratar os serviços que estão presentes no projeto diretamente conosco, e os pontos positivos de realizar essas contratações.

## 4.2 Desenvolvimento

Abaixo segue uma imagem do código utilizado para desenvolver uma vitrine de modelos de templates.

**Figura 7 – Tela de edição de código Python.**



```

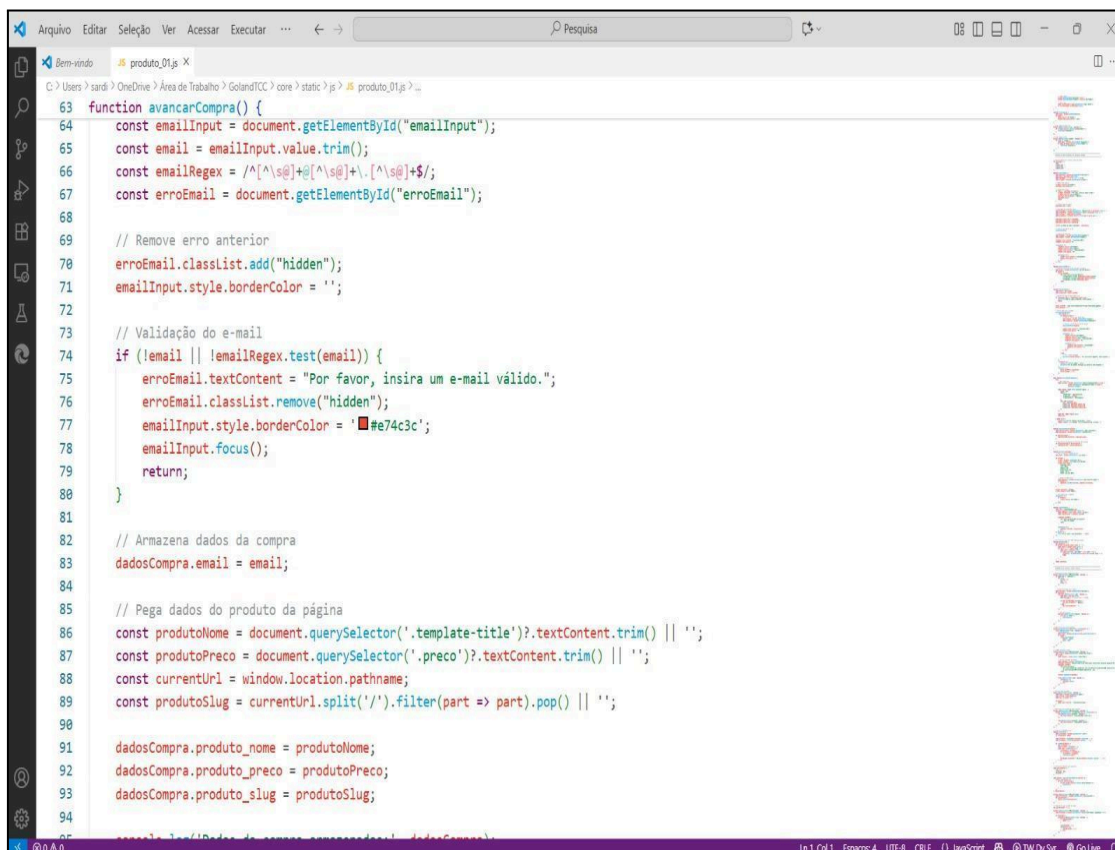
127 class Produto(models.Model):
140
141     categoria = models.ForeignKey(
142         'Categoria',
143         on_delete=models.CASCADE,
144         related_name='produtos',
145         help_text='Categoria à qual o produto pertence. Exemplo: Ele é da categoria "Tecnologia"'
146     )
147     categoria_geral = models.ForeignKey(
148         'CategoriaGeral',
149         on_delete=models.CASCADE,
150         related_name='produtos',
151         null=True, blank=True,
152         help_text='Categoria geral à qual o produto pertence. Exemplo: Ele é da categoria geral "Curso"'
153     )
154     preco = models.DecimalField(
155         max_digits=8,
156         decimal_places=2,
157         help_text='Preço do produto em reais. Exemplo: 1100 (No site: R$ 1100,00)'
158     )
159
160     # Descrições
161     descricao_breve = models.TextField(
162         max_length=300,
163         help_text='Descrição breve para sidebar (máx. 300 caracteres)',
164         default='Template profissional desenvolvido com as melhores práticas de design e usabilidade.'
165     )
166     descricao_detalhada = models.TextField(
167         help_text='Descrição completa para a seção principal (máx. 1500 caracteres)',
168         default='Este template foi desenvolvido especificamente para empresas que desejam uma presença online profissional e moderna. Apresenta
169     )
170
171     # Tags incluídas
  
```

Fonte: Autoria própria.

A imagem mostra um modelo de dados chamado Produto, usado para representar produtos em um banco de dados (no caso do projeto em desenvolvimento, é uma vitrine de modelos de templates para compra). Esse arquivo é usado para criar e manipular dados de forma estruturada dentro da aplicação web.

A imagem abaixo é referente ao código utilizado para a coleta de informações no site.

**Figura 8 – Código de validação e coleta de informações do usuário.**



```

63 function avancarCompra() {
64     const emailInput = document.getElementById("emailInput");
65     const email = emailInput.value.trim();
66     const emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
67     const erroEmail = document.getElementById("erroEmail");
68
69     // Remove erro anterior
70     erroEmail.classList.add("hidden");
71     emailInput.style.borderColor = '';
72
73     // Validação do e-mail
74     if (!email || !emailRegex.test(email)) {
75         erroEmail.textContent = "Por favor, insira um e-mail válido.";
76         erroEmail.classList.remove("hidden");
77         emailInput.style.borderColor = '#e74c3c';
78         emailInput.focus();
79         return;
80     }
81
82     // Armazena dados da compra
83     dadosCompra.email = email;
84
85     // Pega dados do produto da página
86     const produtoNome = document.querySelector('.template-title')?.textContent.trim() || '';
87     const produtoPreco = document.querySelector('.preco')?.textContent.trim() || '';
88     const currentUrl = window.location.pathname;
89     const produtoSlug = currentUrl.split('/').filter(part => part).pop() || '';
90
91     dadosCompra.produto_nome = produtoNome;
92     dadosCompra.produto_preco = produtoPreco;
93     dadosCompra.produto_slug = produtoSlug;
94
95     // Envia dados para o servidor (exemplo simplificado)
96     fetch('/api/comprar', {
97         method: 'POST',
98         headers: {
99             'Content-Type': 'application/json',
100         },
101         body: JSON.stringify(dadosCompra),
102     })
103     .then(response => response.json())
104     .then(data => {
105         // Lógica de sucesso ou erro
106     });
107 }

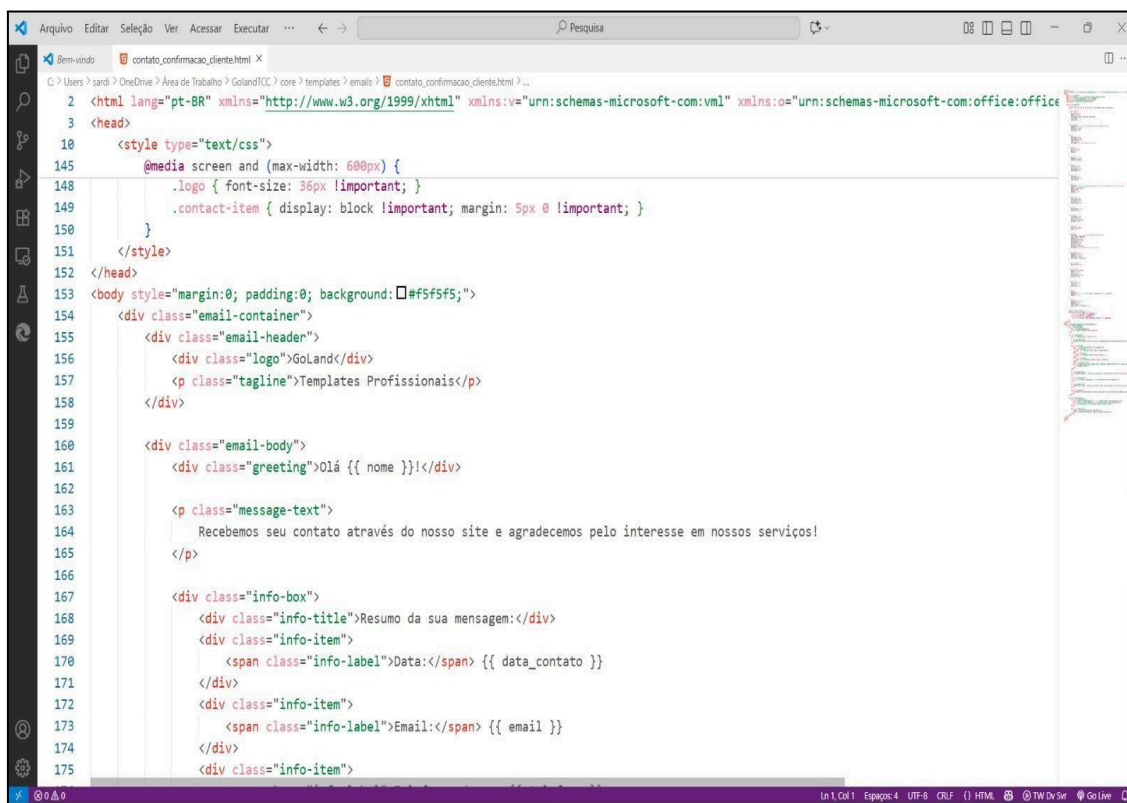
```

Fonte: Autoria própria.

Essa parte do código valida o e-mail do usuário e coleta informações do produto (nome, preço e slug) para armazená-las antes de continuar com o processo de compra. Além disso, obtém o campo de email, do formulário e o texto digitado pelo usuário, também define uma expressão regular para verificar se o email tem um formato válido, e seleciona o elemento de erro para exibir mensagens, se necessário.

A imagem a seguir é referente ao código no qual são gerados e-mails personalizados de contato ou mensagem.

**Figura 9 – Código que gera e-mails personalizados.**



```

1  <html lang="pt-BR" xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vm1" xmlns:o="urn:schemas-microsoft-com:office:office"
2
3  <head>
4
5
6
7
8
9
10 <style type="text/css">
11
12 @media screen and (max-width: 600px) {
13
14   .logo { font-size: 36px !important; }
15
16   .contact-item { display: block !important; margin: 5px 0 !important; }
17
18 }
19 </style>
20 </head>
21
22 <body style="margin:0; padding:0; background-color:#f5f5f5;">
23   <div class="email-container">
24     <div class="email-header">
25       <div class="logo">GoLand</div>
26       <p class="tagline">Templates Profissionais</p>
27     </div>
28
29     <div class="email-body">
30       <div class="greeting">Olá {{ nome }}!</div>
31
32       <p class="message-text">
33         Recebemos seu contato através do nosso site e agradecemos pelo interesse em nossos serviços!
34       </p>
35
36       <div class="info-box">
37         <div class="info-title">Resumo da sua mensagem:</div>
38         <div class="info-item">
39           <span class="info-label">Data:</span> {{ data_contato }}
40         </div>
41         <div class="info-item">
42           <span class="info-label">Email:</span> {{ email }}
43         </div>
44         <div class="info-item">
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

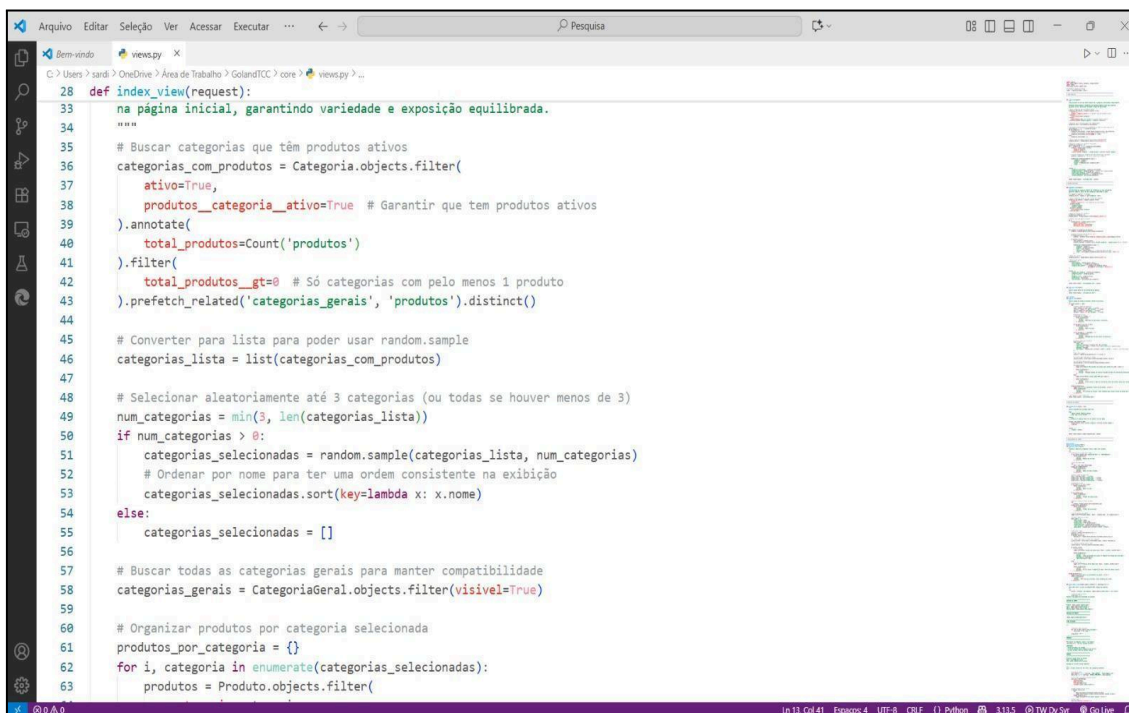
Fonte: Autoria própria.

A imagem mostra um código-fonte de um arquivo HTML, que faz parte de um modelo de e-mail. A estrutura do código foi desenvolvida com o intuito de realizar um e-mail de confirmação de contato ou mensagem. Foi utilizado o css para estilizar a aparência do conteúdo, com destaque para a adaptação para dispositivos móveis. Esse código é parte de um sistema que gera e-mails personalizados, com um design adaptado para diferentes tamanhos de tela.



Na imagem abaixo é possível visualizar a página de código na qual prepara os dados de categorias e produtos.

**Figura 10 – Código que organiza os produtos de cada categoria.**



```
28 def index_view(request):
33     na página inicial, garantindo variedade e exposição equilibrada.
34     """
35     # Buscar categorias que têm produtos ativos
36     categorias_com_produtos = Categoria.objects.filter(
37         ativo=True,
38         produtos_categoria_ativo=True # Garantir que tem produtos ativos
39     ).annotate(
40         total_produtos=Count('produtos')
41     ).filter(
42         total_produtos__gt=0 # Só categorias com pelo menos 1 produto
43     ).prefetch_related('categorias_gerais', 'produtos').distinct()
44
45     # Converter para lista para poder usar random.sample
46     categorias_lista = list(categorias_com_produtos)
47
48     # Selecionar aleatoriamente até 3 categorias (ou todas se houver menos de 3)
49     num_categorias = min(3, len(categorias_lista))
50     if num_categorias > 0:
51         categorias_selecionadas = random.sample(categorias_lista, num_categorias)
52         # Ordenar por nome para ter uma ordem consistente na exibição
53         categorias_selecionadas.sort(key=lambda x: x.nome)
54     else:
55         categorias_selecionadas = []
56
57     # Buscar todas as categorias gerais para manter compatibilidade
58     categorias_gerais = CategoriaGerais.objects.filter(visivel=True)
59
60     # Organizar produtos por categoria selecionada
61     produtos_por_categoria = {}
62     for i, categoria in enumerate(categorias_selecionadas):
63         produtos = Produto.objects.filter(
```

Fonte: Autoria própria.

Essa parte do código busca categorias que tenham produtos ativos, além disso organiza os produtos de cada categoria para exibir na página. Ela prepara os dados de categorias e produtos que vão aparecer na home do site, o que garante uma variedade e organização.

Na imagem a seguir é possível visualizar o código no qual realiza a interação sobre as categorias de templates e seus respectivos produtos.

**Figura 11 – Código que gera interação sobre as categorias e seus respectivos produtos.**

```

4  <html lang="pt-BR">
22 <body>
41
42     CONTEÚDO PRINCIPAL
43     =====>
44     <!-- Substitua o conteúdo principal por este: -->
45     <main class="container">
46         <!-- Informação sobre seleção aleatória (opcional, para debug) -->
47         {% comment %}
48         {% if user.is_staff %}
49             <div class="admin-info">
50                 <small>Exibindo {{ categorias_selecionadas|length }} de {{ total_categorias_disponiveis }} categorias disponíveis</small>
51             </div>
52         {% endif %}
53         {% endcomment %}
54
55         <!-- Loop das Categorias Selecionadas Aleatoriamente -->
56         {% for categoria_slug, dados in produtos_por_categoria.items %}
57         <section class="section">...
115 </section>
116 {% empty %}
117 <section class="section">...
122 </section>
123 {% endfor %}
124
125 <!-- Botão para Ver Todas as Categorias -->
126 <section class="section">
127     <div class="view-all-section">
128         <h3>Quer ver mais templates?</h3>
129         <a href="{% url 'core:templates' %}" class="btn-view-all">
130             <i class="fas fa-th-large"></i>
131             Ver Todas as Categorias

```

Fonte: Autoria própria.

Essa imagem de código mostra informações administrativas exclusivamente para usuários com permissões de equipe. Em seguida, realiza uma interação sobre as categorias e seus respectivos produtos, a fim de gerar as seções correspondentes na página. Por fim, é apresentado um botão ou link que direciona o usuário à página que contém a listagem completa de todas as categorias.

### 4.3 Publicação

Após a finalização, o projeto será publicado para que a comunidade possa acessá-lo. Para isso, o site que irá hospedar este projeto será um site de hospedagem. O domínio utilizado será [www.goland.com.br](http://www.goland.com.br).

Um site de hospedagem é responsável por guardar os arquivos do site e garantir que ele fique acessível na internet. Ele funciona como a "casa" do site, podendo ser um servidor na Hostinger, Netlify, ou outros serviços similares. Já o domínio é o endereço que as pessoas digitam para chegar até essa "casa", como por exemplo, [www.meusite.com](http://www.meusite.com).

### 4.4 Prototipagem

Na imagem abaixo é possível visualizar a página de contato com os desenvolvedores de templates, atualizada.

**Figura 12 – Tela de contato atualizada.**



Fonte: Autoria própria.

Oferecer serviços de desenvolvimento personalizado. Justificativa para contratação (resultados mensuráveis e levar o negócio ao próximo nível) e chamada para contato.

A seguir imagem da página referente aos desenvolvedores do site, na qual foi atualizada.

**Figura 13 – Tela sobre nós atualizada.**

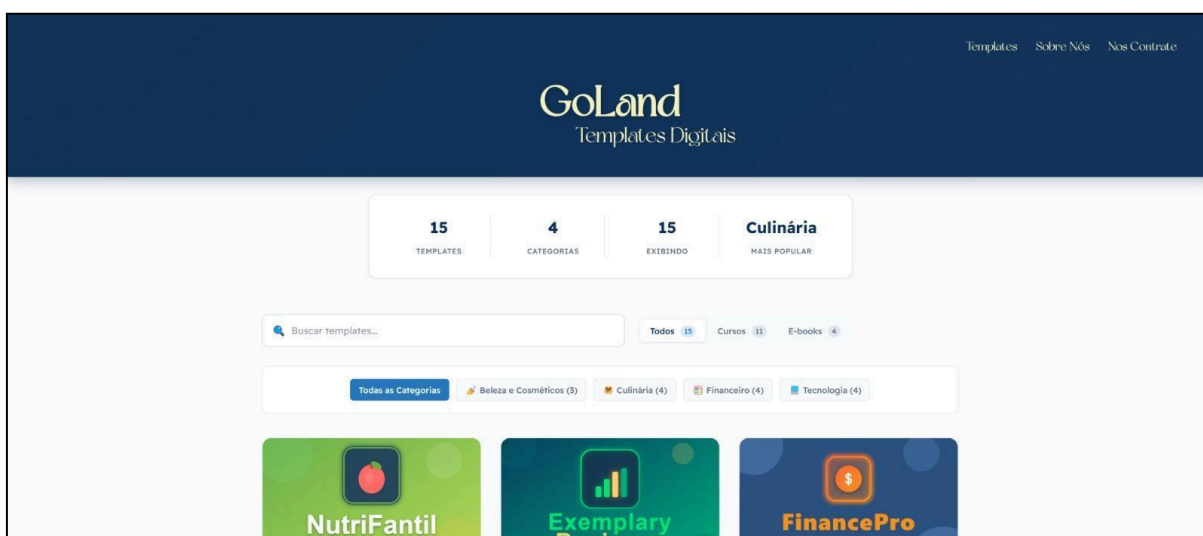


Fonte: Autoria própria.

Informar a missão, visão e valores da empresa. Com um texto destacando a criação de soluções digitais inovadoras, tecnologia de ponta e design centrado no usuário.

Abaixo é possível visualizar a página de busca de templates atualizada.

**Figura 14 – Tela de templates.**



Fonte: Autoria própria.

Exploração e filtragem de todos os templates disponíveis. Contagem de produtos, filtros por categorias e barra de busca.

Abaixo é possível visualizar a tela das coleções de templates atualizada.

**Figura 15 – Tela coleções em destaque.**

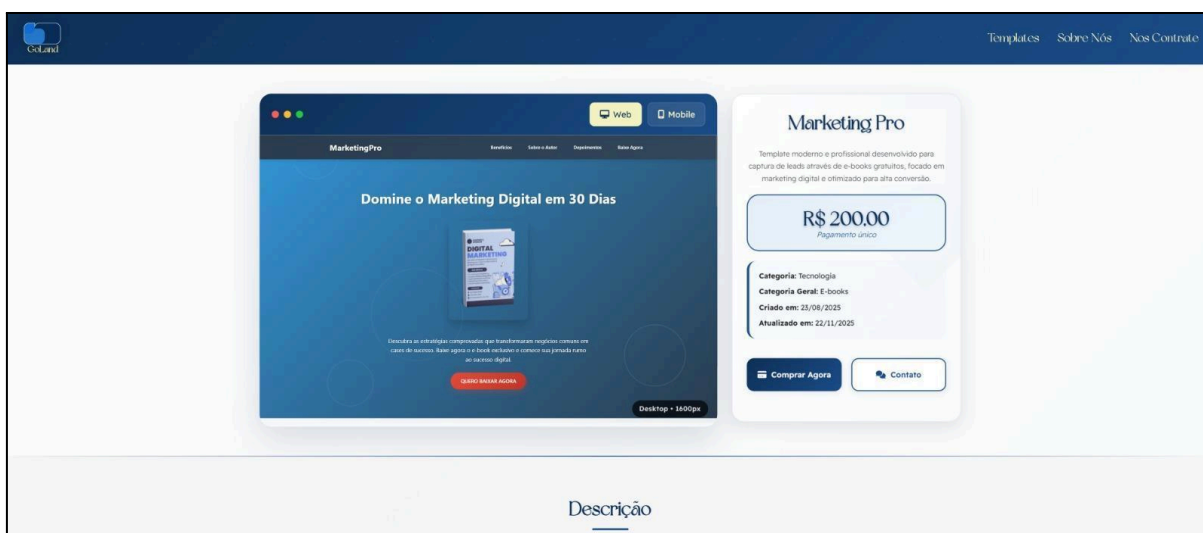


Fonte: Autoria própria.

Apresentação e destaque de produtos Premium. Sessão “coleções em destaque” e exibição de templates por categorias, exemplo tecnologia.

A seguir versão atualizada da página de compra e acesso aos templates.

**Figura 16 – Tela de pré-visualização dos templates, antes da compra.**



Fonte: Autoria própria.

Exibir informações e concretizar a venda de um template. Pré-visualização web/mobile, preço, descrição e botões comprar agora e contato.

#### 4.5 Aplicativos Semelhantes

A seguir é possível visualizar imagens da tela de sites de templates semelhantes ao GOland.

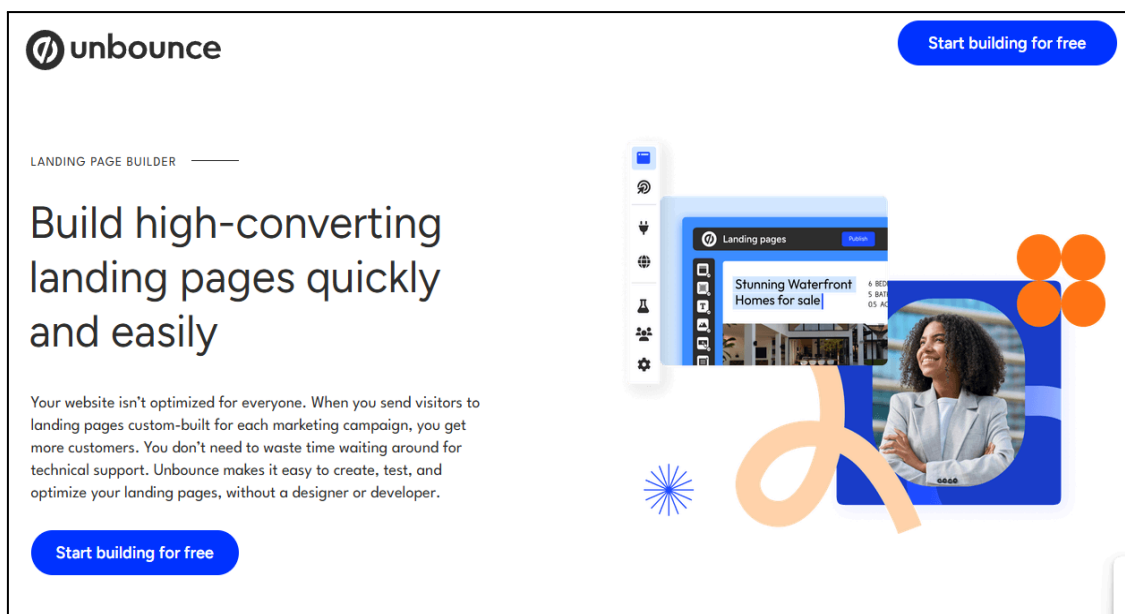
**Figura 17 – Tela do site Rally Fan.**



Fonte: Rally Fan, 2025.

Rally Fan é uma plataforma que tem como objetivo posicionar “tudo-em-um”, é uma plataforma que foi desenvolvida para criadores/vendedores de produtos digitais - templates, downloads, cursos, etc. Além disso, ele permite que sejam criadas lojas personalizadas para desenvolver projetos e vender seus próprios arquivos diretamente. Em comparação ao projeto, o site Rally Fan não possui taxa de transação para pagamentos, possui muitas ferramentas de marketing digital integradas (fluxos) preparadas para criadores.

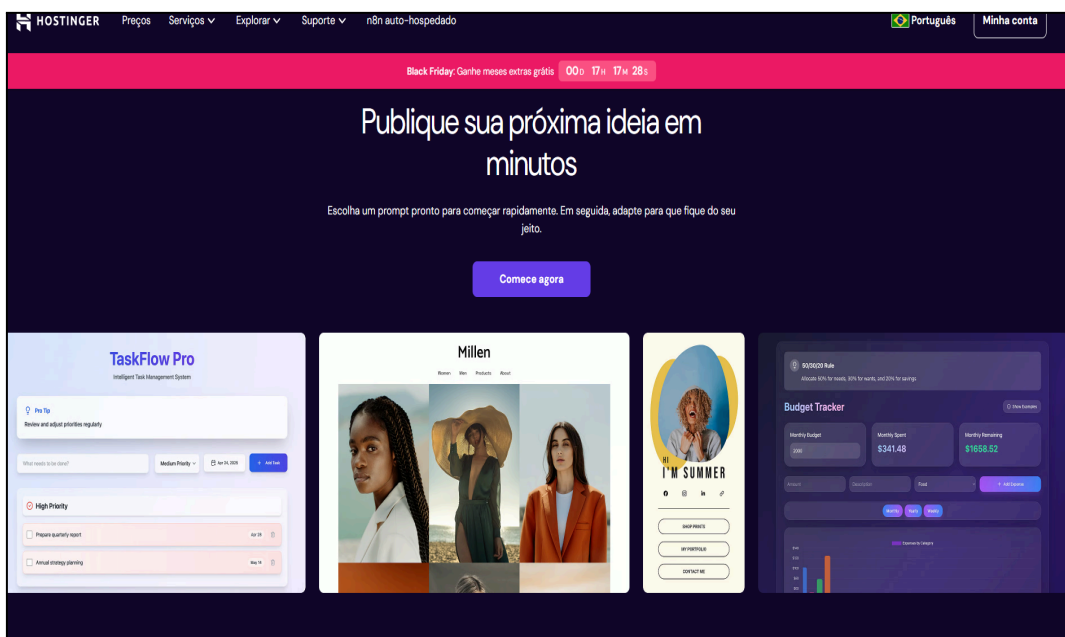
**Figura 18 – Tela do site Unbounce.**



Fonte: Unbounce, 2025.

O Unbounce é um construtor de páginas de destino que permite a profissionais de marketing, empreendedores e empresas criarem, publicarem e testarem Landing Pages otimizadas para conversão, sem a necessidade de conhecimentos em programação (código), assim como o projeto Go land, que permite a criação de landing pages sem um conhecimento prévio sobre programação .

**Figura 19 – Tela do site Hostinger.**



Fonte: Hostinger, 2025.

O site Hostinger, possui uma Hospedagem Compartilhada, que é ideal para sites pequenos e médios, blogs e landing pages. A hospedagem WordPress é otimizada especificamente para sites construídos com WordPress, oferecendo recursos de IA para gerenciamento de conteúdo e solução de problemas. É possível escolher um plano e um período de contratação. Quanto maior o período, mais barato ele fica.



## 5 CONSIDERAÇÕES FINAIS

O resultado esperado foi atingido, baseado nas ideias iniciais, este projeto oferece uma estrutura responsiva e adaptável. Ocorreram apenas mudanças no design da página, e adaptações que se adequassem às alterações necessárias. Além disso, a escolha do framework Django e das linguagens HTML, CSS, JavaScript e Python se apresentaram eficiente, certificando que houvesse um bom desempenho, segurança e flexibilidade na implementação das funcionalidades planejadas.

Quanto à eficiência do projeto, foram realizados testes que revelaram que o site apresenta uma navegação fluida e bem estruturada, com um carregamento rápido e um layout simples de compreender. No entanto, algumas funcionalidades que haviam sido planejadas, como a integração com diferentes formas de pagamento e um painel de administração mais completo, não foram totalmente implementadas devido à limitação de tempo e à complexidade técnica dessas tarefas.

A continuidade do projeto envolve ampliar suas funcionalidades e melhorar a experiência dos usuários, tornando o site mais completo e interativo. Uma das principais metas é criar um painel administrativo que facilite o gerenciamento de templates, usuários e vendas de forma automática. Além disso, se pretende integrar sistemas de pagamento online, permitindo que as transações sejam feitas de forma segura direto na página. Outra prioridade é melhorar o desempenho e a segurança do banco de dados, usando ferramentas mais avançadas e mantendo o framework sempre atualizado.

## REFERÊNCIAS

- BATISTA, Leonardo dos Santos; KUMADA, Kate Mamhy Oliveira. Análise metodológica sobre as diferentes configurações da pesquisa bibliográfica. **Revista brasileira de iniciação científica**, p. e021029-e021029, 2021.
- BOCCATO, Vera Regina Casari. Metodologia da pesquisa bibliográfica na área odontológica e o artigo científico como forma de comunicação. **Rev. Odontol. Univ. Cidade de São Paulo, São Paulo**, v. 18, n. 3, p. 265-274, 2006.
- BONELLI, Ana Carolina Bonani. **Estudo sobre a reformulação de landing pages e o impacto sobre o marketing digital**. – Universidade Federal de São Paulo, São José dos Campos, 2023.
- Basilio, Guilherme Manfrim, and Wdson de Oliveira. **Ferramentas de segurança para banco de dados: focando em SQL Injection**. Revista Brasileira em Tecnologia da Informação 4.2 (2022): 10-19.
- BRUNNER, Thais et al. Implementação de programas de integridade na administração pública: uma revisão sistemática de literatura. **Revista da CGU**, v. 16, n. 30, 2024.
- CHEN, Songtao et al. **Django web development framework: Powering the modern web**. American Journal of Trade and Policy, v. 7, n. 3, p. 99-106, 2020.
- E-commerce sente dificuldade em contratar profissionais de TI, diz pesquisa. E-COMMERCE BRASIL. 2023. Disponível em: <<https://www.ecommercebrasil.com.br/noticias/e-commerce-sente-dificuldade-em-contratar-profissionais-de-ti-diz-pesquisa>>. Acesso em: 10 de abril de 2025.
- Falta de pessoal qualificado é o maior obstáculo para a inclusão digital dos pequenos negócios. Sebrae Agency 2022. Disponível em: <<https://agenciasebrae.com.br/arquivo/falta-de-pessoal-qualificado-e-o-maior-obstaculo-para-inclusao-digital-dos-pequenos-negocios>>. Acesso em: 7 de abril de 2025
- FÁVERO, Luiz Paulo; BELFIORE, Patrícia. **Manual de análise de dados: estatística e modelagem multivariada com Excel®, SPSS® e Stata®**. Elsevier Brasil, 2017.
- GARCIA, Vinícius Salles; SOTTO, Eder Carlos Salazar. **Comparativo entre os modelos de banco de dados relacional e não-relacional**. Revista Interface Tecnológica, v. 16, n. 2, p. 12-24, 2019.
- HTML: o futuro do desenvolvimento web – o que nos aguarda para HTML e CSS. NETCODERS. Disponível em: <<https://netcoders.com.br/html/html-o-futuro-do-desenvolvimento-web-o-que-nos-aguarda-para-html-e-css>>. Acesso em: 17 abr. 2025.
- MEESALA, S. The Role of a Landing Page in Digital Marketing. **Journal Of Emerging Technologies and Innovative Research (JETIR)**, p. 1-6, 2022.

MOHD, Tauheed Khan et al. **Comparative Analysis on Various CSS and JavaScript Frameworks**. J. Softw., v. 17, n. 6, p. 282-291, 2022.

OLIVEIRA, Paulo Henrique Pinho, Isabela Peçanha Gois, e Maria Clara Bravo Goldstein. **Aplicação do funil de vendas para melhoria de engajamento em um SVA**. Revista da Universidade Vale do Rio Verde 16.2 (2018).

OLIVEIRA, rosevaldo; Campos,Victor. **Computação Científica usando Python: uma introdução a linguagem de programação**. (2016)

RODRIGUES, David Luiz. **Segurança em banco de dados**. 2021. 30 f. Trabalho de Conclusão de Curso (Bacharelado em Sistema de Informação) – Faculdade de Negócio e Tecnologia da Informação, Taguatinga, DF, 2021.

SILVA, Camila Cristina Rodrigues da; PLOTZE, Rodrigo de Oliveira; CHINA, Anna Patricia Zakem. **Tecnologias de desenvolvimento web na construção de landing pages para marketing pessoal**. VIII Workshop de Tecnologia da Fatec Ribeirão Preto, v. 1, n. 8, dez. 2023.

STEFANOV, Stoyan. **Padrões JavaScript**. São Paulo: Novatec, 2011.

SUDHA, C.; Nagesh, A. **A comprehensive survey on data mining techniques in wireless sensor networks**. International Journal of Computer Sciences and Engineering, v. 6, n. 6, p. 1523-1527, 2018.

The simplest way to find customers and grow online. RALLY FAN. Disponível em: <<https://rally.fan>>. Acesso em: 30 de outubro de 2025.

TURNIP, F. Firdaus et al. Development of Web Landing Page for Small and Medium Enterprise Promotion Business. In: **Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies**. 2020. p. 622-629.

VAZ, Jacqueline et al. **Criação e implementação de um banco de dados prospectivo e multicêntrico de pacientes com infarto agudo do miocárdio**: RIAM. Arquivos Brasileiros de Cardiologia, v. 114, n. 3, p. 446-455, 2020.