

SISTEMA WEB DE GESTÃO DE ESTOQUE PARA UMA LOJA DE PRODUTOS AROMÁTICOS: UM ESTUDO DE CASO COM FOCO NA DIGITALIZAÇÃO E ORGANIZAÇÃO DE PROCESSOS DE MICROEMPREENDEDORES

Vitor Hugo Morás*; Alexssandro Ferreira Cordeiro**

*Acadêmico de Engenharia de Software - Uniguaçu, vitor.moras@hotmail.com.

**Mestre Tecnologias Computacionais para o Agronegócio – UTFPR Medianeira, alexssandrofc@gmail.com.

INFORMAÇÕES

Histórico de submissão:

Recebido em: 20 maio. 2025

Aceite: 27 jun. 2025

Publicação online: jun. 2025

RESUMO

Este artigo demonstra a relevância dos Microempreendedores Individuais (MEIs) no cenário econômico brasileiro e os desafios referentes a gestão manual de seus negócios, particularmente no controle de estoque, vendas e relacionamento com clientes e fornecedores. Diante das limitações dos métodos manuais, que frequentemente resultam em erros, retrabalho e perda de informações, evidencia-se a necessidade de soluções tecnológicas acessíveis, intuitivas e eficazes. Este trabalho apresenta o desenvolvimento e a análise de um sistema web de gestão de estoque, criado de um estudo de caso focado nas necessidades de uma microempreendedora do ramo de produtos aromáticos em Medianeira-PR. A metodologia empregada seguiu princípios da engenharia de software, utilizando o framework Laravel 10, banco de dados MySQL, e tecnologias frontend como Bootstrap, TailwindCSS, JavaScript e AJAX, sob a arquitetura MVC. O processo incluiu levantamento de requisitos detalhado junto à usuária, modelagem de dados relacional, desenvolvimento de módulos CRUD para gestão de clientes, fornecedores, produtos, compras e vendas, implementação de autenticação segura com Laravel Sanctum e criação de um painel administrativo (*dashboard*) com indicadores-chave. O sistema visa automatizar o controle de estoque, centralizar o cadastro de informações, otimizar o registro de transações e fornecer relatórios gerenciais, substituindo processos manuais propensos a falhas. Visando o potencial impacto positivo da ferramenta na organização, agilidade e segurança da gestão, contribuindo para o negócio e demonstrando o valor da digitalização para o microempreendedorismo. O foco em escalabilidade e a preparação para futuras integrações com plataformas de e-commerce e aplicativos móveis também são destacados como diferenciais da solução proposta.

Palavras-chave: gestão de estoque; sistema web; microempreendedor individual; automação, empreendedorismo.

ABSTRACT

This article highlights the importance of Individual Microentrepreneurs (MEIs) in the Brazilian economic landscape and the challenges they face in manually managing their businesses, particularly in inventory control, sales, and customer/supplier relationships. Given the limitations of manual methods—which often lead to errors, inefficiencies, and data loss—there is a clear need for accessible, intuitive, and effective technological solutions. This study presents the development and analysis of a web-based inventory management system, based on a case study of an aromatic products microentrepreneur in Medianeira-PR. The methodology followed software engineering principles, utilizing the Laravel 10 framework, MySQL database, and frontend technologies such as Bootstrap, TailwindCSS, JavaScript, and AJAX, under an MVC architecture. The process included detailed requirement gathering, relational data modeling, and the development of CRUD modules for managing clients, suppliers, products, purchases, and sales. Secure authentication was implemented using Laravel Sanctum, and an administrative *dashboard* with key performance indicators was created. The system aims to automate inventory control, centralize data registration, optimize transaction records, and provide managerial reports, replacing error-prone manual processes. The study emphasizes the tool's potential to enhance organization, efficiency, and security in business management, demonstrating the value of digitalization for microentrepreneurship. Scalability and future integrations with e-commerce platforms and mobile applications are also highlighted as key advantages of the proposed solution.

Keywords / Palabras clave: inventory management; web system; individual microentrepreneur; automation; entrepreneurship

Copyright © 2025, Vitor Hugo Morás; Alessandro Ferreira Cordeiro. *This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.*

Citação: MORÁS, Vitor Hugo; CORDEIRO, Alessandro Ferreira. Sistema Web de gestão de estoque para uma loja de produtos aromáticos: um estudo de caso com foco na digitalização e organização de processos de microempreendedores. **Iguazu Science**, São Miguel do Iguacu, v. 3, n. 7, p. 08-18, jun. 2025.

INTRODUÇÃO

O avanço tecnológico das últimas décadas tem sido um catalisador para transformações profundas em diversos segmentos da sociedade, impactando diretamente a maneira como os negócios são concebidos, gerenciados e expandidos. No contexto brasileiro, essa revolução digital encontra um terreno particularmente fértil entre os Microempreendedores Individuais (MEIs).

De acordo com dados do Sebrae de 2023, os MEIs representam uma parcela significativa e vital do cenário empresarial do país, constituindo 73,4% das empresas ativas no Brasil. Este regime foi estabelecido com o propósito fundamental de trazer à formalidade trabalhadores autônomos e pequenos empreendimentos, consolidando-se como um pilar crucial para a geração de renda, a inclusão produtiva e a movimentação da economia em todas as regiões. A formalização via MEI simplifica a carga tributária, permite acesso a benefícios previdenciários e facilita a obtenção de crédito, tornando-se um passo importante para a sustentabilidade de muitos pequenos negócios (Sebrae, 2023).

Apesar da importância crescente e das vantagens da formalização, grande parte dos MEIs ainda se depara com desafios consideráveis na administração cotidiana de seus negócios. As dificuldades são multifacetadas e frequentemente envolvem aspectos cruciais como o controle eficiente de estoque, a gestão de vendas, e a organização do cadastro de clientes e fornecedores. A prática comum entre muitos desses empreendedores é a realização dessas atividades de forma predominantemente manual. Isso se manifesta em métodos como anotações em cadernos, o uso de planilhas eletrônicas que, por vezes, carecem de organização estruturada, ou até mesmo a dependência da memorização de dados vitais.

Essa abordagem manual, embora possa parecer simples inicialmente, acarreta uma série de consequências negativas que comprometem a eficiência operacional e a capacidade de crescimento sustentável do negócio. A falta de sistematização leva a retrabalho constante na busca e organização de informações, aumenta drasticamente a probabilidade de erros nos registros de estoque e vendas, e, crucialmente, limita o tempo que o empreendedor poderia dedicar a atividades mais estratégicas ou mesmo ao seu lazer. A ausência de registros claros e

estruturados impacta diretamente na tomada de decisões, podendo resultar em problemas como a perda de mercadorias por falta de controle preciso, falhas no atendimento ao cliente devido a informações incompletas ou incorretas, desorganização financeira pela dificuldade em rastrear receitas e despesas vinculadas às vendas, e, em casos mais graves, pode até mesmo comprometer a viabilidade do negócio a longo prazo.

É nesse contexto de desafios e limitações impostas pela gestão manual que se torna evidente a necessidade urgente de soluções tecnológicas. Tais soluções devem ser acessíveis, simples de usar e, acima de tudo, eficazes para atender à realidade operacional específica dos MEIs. O cenário ideal seria aquele em que os microempreendedores pudessem contar com ferramentas digitais capazes de organizar e centralizar informações, automatizar tarefas repetitivas e fornecer relatórios claros que sirvam de base para o planejamento e a tomada de decisões estratégicas.

Com um sistema de gestão digital eficiente, seria possível transformar a rotina do MEI. A organização dos processos internos seria aprimorada, o controle de estoque seria realizado em tempo real (evitando surpresas com falta ou excesso de produtos), os registros de vendas se tornariam precisos e rastreáveis, e manter um histórico confiável de fornecedores e clientes seria uma tarefa simples. Além disso, o acompanhamento de indicadores-chave de desempenho ofereceria uma visão clara da saúde do negócio, permitindo que o empreendedor identificasse tendências e oportunidades (E-Commerce Brasil, 2023).

Diante dessa realidade e da clara necessidade de ferramentas adaptadas aos MEIs, este artigo se propõe a apresentar o desenvolvimento de um sistema de gestão web. Este sistema foi concebido com base nas necessidades reais identificadas em um estudo de caso focado em uma microempreendedora do ramo de produtos aromáticos, localizada na cidade de Medianeira-PR. O objetivo primordial deste trabalho é demonstrar o impacto positivo que a tecnologia pode exercer na rotina de um MEI. A solução digital desenvolvida visa promover organização, agilidade e segurança na administração das atividades diárias, buscando reduzir as falhas inerentes aos processos manuais, evitar a perda de informações valiosas e contribuir diretamente para a evolução e profissionalização do negócio, reafirmando assim o

papel da inovação como um diferencial competitivo no microempreendedorismo brasileiro.

METODOLOGIA

O processo de desenvolvimento do sistema web de gestão de estoque seguiu uma abordagem estruturada, fundamentada nos princípios da engenharia de software. Foram adotados métodos ágeis no que tange ao levantamento de requisitos e aplicadas boas práticas de programação orientada a objetos, visando construir uma solução que não fosse apenas funcional, mas também escalável e segura (Pressman, 2016).

O ponto de partida foi a necessidade de resolver um problema prático e real. Isso exigiu uma interpretação aprofundada das demandas da usuária final, a microempresendedora. Através de uma abordagem empática e investigativa, foram apresentadas as principais dificuldades enfrentadas em sua rotina e as funcionalidades essenciais que o sistema deveria incorporar para otimizar seu trabalho. O estudo de caso concentrou-se especificamente em uma empresa local de Medianeira-PR, atuante no setor de produtos aromáticos, cuja gestão de estoque e vendas era realizada de maneira inteiramente manual antes da intervenção tecnológica proposta, o que a tornava suscetível a erros operacionais e à perda de informações cruciais por falta de organização estratégica.

A aplicação foi construída utilizando o framework Laravel em sua versão 10, conhecido por sua vasta documentação, comunidade ativa e recursos que promovem segurança e produtividade no desenvolvimento web. A arquitetura adotada seguiu o padrão MVC (Model-View-Controller), que garante uma clara separação de responsabilidades entre as camadas da aplicação. O motor de templates Blade foi integrado para a renderização das páginas (Laravel, 2025).

Um aspecto crucial para garantir a segurança do sistema, tanto para acesso direto quanto para futuras integrações, foi a implementação do Laravel Sanctum. Este pacote oferece uma solução de autenticação leve baseada em *tokens*, ideal para proteger APIs e ações sensíveis dentro do sistema. A escolha do Laravel Sanctum habilita a capacidade de o sistema se conectar de forma segura com clientes externos, como uma aplicação mobile ou plataformas de e-commerce, mantendo a integridade e a segurança das requisições HTTP (Laravel Sanctum, 2025).

A configuração do projeto, gerenciada pelo arquivo `.env`, utilizou o servidor web Apache e um banco de dados MySQL (Paul, 20214). O MySQL foi escolhido por ser um Sistema Gerenciador de Banco de Dados (SGBD) relacional robusto, de código aberto e com integração eficiente ao Laravel. A interface do usuário (frontend) foi desenvolvida com HTML e CSS. Foram

utilizados os frameworks Bootstrap e TailwindCSS, que em conjunto proporcionam flexibilidade visual e agilidade no desenvolvimento da interface, com um baixo custo de aprendizado. Para adicionar funcionalidades dinâmicas e melhorar a experiência do usuário sem a necessidade de recarregar a página inteira, foram empregadas tecnologias como JavaScript e AJAX, com a troca de dados ocorrendo no formato JSON (MDN Web Docs, 2025).

As etapas seguidas na construção do sistema de gerenciamento de estoque estão detalhadas a seguir:

1. Levantamento de Requisitos: Esta fase foi crucial e realizada através de entrevistas informais com a microempresendedora. A abordagem adotada foi empática e investigativa, buscando compreender profundamente a rotina e as frustrações da usuária com o método manual. Os principais desafios identificados incluíam a dificuldade na organização das compras, a perda de contato e a dificuldade em localizar fornecedores, a ausência de um controle automatizado e preciso de estoque, e a falta de um histórico organizado de vendas e clientes. Um relato que ilustra bem essa situação foi: *"Nunca acho novamente os meus fornecedores para ter um controle, sempre é necessário revirar por todos os aplicativos que tenho, perco muito tempo"*. Este depoimento ressaltou a necessidade premente de uma solução centralizada que consolidasse essas informações. Com base nas dificuldades e no fluxo de trabalho atual da microempresa, foi realizado o mapeamento dos processos para definir precisamente os requisitos funcionais (o que o sistema deve fazer) e não funcionais (qualidades do sistema, como segurança e usabilidade) (Pressman, 2016).

- 1.1 O processo de entrevista explorou tópicos como o ciclo completo de uma venda (desde o atendimento ao cliente até a entrega), o processo de compra de produtos (como fornecedores são encontrados, pedidos feitos, recebimento), como o estoque é monitorado (ou a falta dele), como os clientes são registrados e contatados, e como os registros de vendas são mantidos. Os requisitos funcionais detalhados incluíram, por exemplo: RF1: Registrar e gerenciar informações de clientes (nome, contato, endereço, histórico de compras); RF2: Registrar e gerenciar informações de fornecedores (contato, histórico de compras); RF3: Cadastrar e gerenciar produtos (nome, descrição, preço, quantidade, validade, fornecedor); RF4: Registrar vendas (vincular cliente, produtos, calcular total automaticamente); RF5: Registrar compras de fornecedores (vincular

fornecedor, produtos, valor pago); RF6: Atualizar estoque automaticamente após vendas e compras; RF7: Gerar relatórios de vendas por período, produtos, etc.; RF8: Painel administrativo com resumo de indicadores. Requisitos não funcionais abrangeram a segurança (autenticação e autorização), a usabilidade (interface intuitiva e simples), a performance (rapidez nas operações) e a escalabilidade (capacidade de futuras expansões)

2. Modelagem de Dados: Com os requisitos funcionais em mãos, foi possível projetar a estrutura do banco de dados. A modelagem de dados foi elaborada utilizando diagramas de entidade-relacionamento (MER) (HEUSER, 2009). A estrutura do banco de dados foi implementada e versionada utilizando as migrations do Laravel, garantindo que as alterações na estrutura do banco de dados fossem controladas e rastreáveis. As tabelas principais criadas para dar suporte às funcionalidades incluíram:

- users: para o gerenciamento de usuários do sistema e autenticação.
- clients: para armazenar as informações dos clientes.
- products: para o cadastro e controle dos produtos.
- suppliers: para o registro e histórico dos fornecedores.
- purchases: para registrar as compras feitas dos fornecedores.
- sales: para registrar as vendas realizadas aos clientes.
- stocks: para gerenciar os níveis de estoque dos produtos, possivelmente com movimentações de entrada e saída.

Cada tabela foi cuidadosamente projetada, incluindo chaves primárias para identificação única e chaves estrangeiras para estabelecer os relacionamentos entre elas (por exemplo, uma venda está relacionada a um cliente e a múltiplos produtos, uma compra está relacionada a um fornecedor e múltiplos produtos). Campos adicionais como status, timestamps (para registro de criação e atualização) e campos específicos para a lógica de controle de estoque também foram incorporados. A normalização foi aplicada durante o processo de modelagem para eliminar redundâncias, garantir a integridade dos dados e facilitar a manutenção futura do banco.

3. Criação de Módulos Funcionais (CRUD): O desenvolvimento dos módulos do sistema seguiu o padrão CRUD (Create, Read, Update, Delete) para as entidades principais. Foi

estabelecido um padrão claro de organização, utilizando controladores (Controllers) para a lógica de negócio, rotas (Routes) para mapear URLs às ações dos controladores, e views (Views) para a apresentação da interface ao usuário, seguindo o padrão MVC. Foram criados módulos específicos para o cadastro, edição, visualização detalhada e exclusão de:

- Clientes: Permitindo registrar todas as informações relevantes do cliente, essencial para histórico de vendas e relatórios.
 - Fornecedores: Focando em resolver a dificuldade da usuária, com dados de contato e histórico de compras associado.
 - Usuários: Para gerenciar quem tem acesso ao sistema.
 - Produtos: Incluindo detalhes como nome, descrição, quantidade em estoque, preço e validade.
 - Compra de fornecedores: Para registrar as entradas de produtos no estoque.
 - Vendas: Um módulo central que permite registrar cada transação de venda, com funcionalidades como o cálculo automático do total com base nos produtos selecionados e suas quantidades, e a vinculação direta do(s) produto(s) vendido(s). Cada módulo utilizou os recursos do Laravel, com rotas nomeadas para facilitar a manutenção e o referenciamento interno dentro da aplicação. A implementação seguiu as boas práticas de desenvolvimento, incluindo validações server-side para garantir que os dados inseridos atendam aos requisitos de integridade antes de serem persistidos no banco de dados (Santos, 2020).
4. Autenticação, Controle de Acesso e Segurança de API: A segurança é um pilar fundamental em qualquer sistema, especialmente um que lida com dados sensíveis de negócios. O sistema de segurança foi construído aproveitando os recursos nativos de autenticação e autorização do Laravel. A verificação de credenciais (*login*) e a proteção de rotas e funcionalidades contra acesso não autorizado foram implementadas utilizando middlewares. Para adicionar uma camada extra de segurança, especialmente pensando em futuras integrações via API, foi implementado o pacote Laravel Sanctum. O Sanctum permite a autenticação baseada em *tokens* pessoais. Após um usuário se autenticar com sucesso (*login*), um token é gerado e associado a essa sessão/usuário. Este token é então enviado com cada requisição subsequente para a API ou para rotas

protegidas. O middleware do Sanctum intercepta essas requisições, valida o token e garante que o usuário autenticado tenha as permissões necessárias para acessar o recurso solicitado. Isso garante que apenas usuários devidamente autenticados e autorizados possam acessar os dados e funcionalidades sensíveis do sistema, protegendo informações cruciais de estoque, vendas, clientes e fornecedores (Laravel Sanctum, 2025).

5. Painel administrativo (*Dashboard*): O *dashboard* ou painel administrativo serve como a porta de entrada principal para o usuário após o *login*. Ele foi projetado para fornecer uma visão geral rápida da operação do negócio. O *dashboard* apresenta indicadores-chave e atalhos visuais para os módulos mais acessados, como cadastro/gerenciamento de produtos, clientes, fornecedores e o acesso aos relatórios. A área de relatórios é dinâmica, permitindo filtros (como por período) para visualizar dados de vendas, movimentações de estoque e cadastros. A forma como os dados são apresentados é clara e concisa, com o objetivo de facilitar a tomada de decisões rápidas e estratégicas e auxiliar em previsões futuras para a empreendedora (Gouvêa, 2017).

5.1 indicadores no *dashboard* incluem, por exemplo: Volume de Vendas Total (soma da quantidade de itens vendidos); Faturamento Total (soma do valor total das vendas registradas); Ticket Médio (Faturamento Total dividido pelo número de vendas); e Vendas Pendentes (se houver um status para vendas não finalizadas ou em processo de entrega). Estes indicadores são calculados dinamicamente a partir dos dados registrados nas tabelas sales e stocks. A capacidade de filtrar relatórios por período, por exemplo, permite analisar o desempenho mensal ou trimestral, identificar sazonalidades e comparar resultados ao longo do tempo, transformando dados brutos em insights acionáveis para o negócio.

6. Testes: Embora o sistema ainda não estivesse em ambiente de produção, uma etapa crucial foi a realização de testes manuais. Os testes foram baseados em cenários reais que a usuária forneceu, simulando as operações cotidianas do negócio. Foram exaustivamente testadas as operações CRUD (cadastro, edição, exclusão e busca) em cada um dos módulos implementados (Clientes, Fornecedores, Produtos, Compras, Vendas, Usuários). A precisão dos dados nos relatórios de vendas e compras também foi validada para garantir

que os cálculos automáticos e a agregação de dados estivessem corretos. Um diferencial importante foi a participação ativa da usuária final no processo de validação. O feedback direto dela sobre a usabilidade, o fluxo das funcionalidades e a clareza das informações foi fundamental para identificar pontos de melhoria e possibilitou a realização de ajustes importantes, como a adaptação de nomenclaturas, melhorias na experiência visual e refinamentos na experiência geral de uso do sistema (Preece; Rogers; Sharp, 2013).

6.1 Exemplos de cenários de teste incluíram: cadastrar um novo cliente com todos os dados válidos; tentar cadastrar um cliente sem preencher campos obrigatórios (validando as mensagens de erro da validação server-side); editar as informações de contato de um fornecedor; excluir um produto que ainda possui estoque (testando possíveis restrições ou avisos); registrar uma venda selecionando múltiplos produtos e verificando se o total é calculado corretamente; registrar uma compra e verificar se o estoque do produto correspondente é atualizado; gerar um relatório de vendas para um mês específico e conferir os totais.

7. Foco em Escalabilidade e Trabalhos Futuros: O sistema foi concebido desde o início com um forte foco em escalabilidade. A escolha da arquitetura REST e a implementação do Laravel Sanctum pavimentam o caminho para integrações futuras com diversas plataformas. Isso inclui potencial conexão com plataformas de e-commerce populares no Brasil, como Nuvemshop e Loja Integrada, e o desenvolvimento de aplicativos móveis. Tais integrações são cruciais para ampliar a presença digital da microempreendedora e permitir a gestão centralizada de múltiplos canais de venda. Funcionalidades como a exportação de dados, a sincronização de pedidos realizados em outras plataformas e a atualização de estoque em tempo real entre o sistema de gestão e os canais de venda online estão previstas para futuras releases do projeto. O objetivo é que o sistema se torne uma plataforma unificada, profissionalizando ainda mais a gestão e permitindo que a microempreendedora escale seu negócio (Mdn Web Docs, 2025).

8. Escolha das Tecnologias: As decisões sobre a pilha tecnológica foram tomadas considerando a eficiência, a robustez, a segurança e o custo-benefício para o contexto de um sistema para MEI.

- O framework Laravel foi a escolha principal para a *backend*. Suas vantagens incluem uma

ampla e detalhada documentação, uma comunidade de desenvolvedores grande e ativa que facilita a resolução de problemas e a busca por conhecimento, e recursos nativos robustos voltados para a segurança (como o próprio Sanctum, autenticação e proteção CSRF) e para a produtividade (como o Eloquent ORM, o Artisan console para tarefas de linha de comando, e o sistema de routing eficiente) (Laravel, 2025).

- Para o banco de dados, optou-se pelo MySQL. Sendo um SGBD relacional amplamente utilizado e de código aberto, o MySQL oferece robustez e confiabilidade para o armazenamento e gerenciamento dos dados estruturados do sistema. Sua integração nativa e eficiente com o framework Laravel através do Eloquent ORM simplificou o desenvolvimento das operações com o banco (Paul, 20214).
 - No frontend, a combinação de Bootstrap e TailwindCSS trouxe o melhor de dois mundos. O Bootstrap fornece um conjunto de componentes prontos e um sistema de grid *layout* que agiliza a construção de interfaces responsivas. O TailwindCSS, por sua vez, é um framework focado em utility classes, permitindo um alto grau de customização visual com baixo custo de aprendizado para estilização granular e específica dos elementos (Tailwind Community, 2024; Bootstrap Doc, 2025).
 - JavaScript e AJAX foram essenciais para introduzir dinamismo e responsividade na aplicação web. O uso de AJAX (Asynchronous JavaScript and XML, embora hoje trabalhe mais com JSON) permite que partes da página sejam atualizadas ou que dados sejam enviados/recebidos do servidor sem a necessidade de recarregar a página inteira⁹.... Isso melhora significativamente a experiência do usuário, tornando a interface mais ágil e interativa, por exemplo, ao salvar um formulário ou filtrar uma lista (Mdn Web Docs, 2025).
9. Boas Práticas de Desenvolvimento: A qualidade do código e a manutenibilidade do sistema foram priorizadas através da aplicação de diversas boas práticas de desenvolvimento de software (Pressman, 2016).
- Aderência rigorosa ao padrão de arquitetura MVC (Model-View-Controller). Isso garantiu uma clara separação de responsabilidades: os Models cuidam da lógica de negócios e interação com o banco de dados, os Controllers gerenciam a lógica da aplicação e a resposta às requisições, e as Views são responsáveis

apenas pela apresentação dos dados ao usuário¹⁰. Essa separação torna o código mais organizado, fácil de entender, testar e manter.

- Implementação de validações server-side em todos os formulários e pontos de entrada de dados. Isso é crucial para a segurança e integridade dos dados, garantindo que apenas informações válidas e no formato correto sejam processadas e armazenadas no banco de dados, mesmo que haja validações no lado do cliente (frontend).
 - Tratamento adequado de exceções e erros. A aplicação foi desenvolvida para lidar com situações inesperadas de forma graciosa, evitando quebras abruptas e fornecendo feedback útil (para o desenvolvedor no log e, quando apropriado, para o usuário).
 - Reutilização de código. Foram criados componentes (especialmente nas views com Blade) e funções auxiliares para evitar a duplicação de código, tornando o desenvolvimento mais eficiente e a manutenção mais simples – uma mudança em um componente é refletida em todos os lugares onde ele é utilizado.
 - Respeito aos princípios SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) sempre que aplicável. A aplicação desses princípios, embora possa parecer um custo inicial, resulta em um código mais legível, flexível, escalável e resiliente a mudanças futuras.
10. Controle de Versão: Para gerenciar as alterações no código-fonte ao longo do desenvolvimento, foi utilizado o sistema de controle de versão distribuído Git (Chacon; Straub, 2014). Foram realizados commits frequentes, pequenas unidades de alteração que registram o progresso e facilitam o rastreamento do histórico de modificações. A organização do projeto em branches específicas foi adotada para simular um ambiente de desenvolvimento colaborativo e manter o código organizado:
- Main ou Master Branch: Representa a versão estável ou pronta para produção.
 - Develop Branch: Onde o desenvolvimento principal ocorre, integrando funcionalidades completas.
 - Feature Branches: Criadas para o desenvolvimento de novas funcionalidades de forma isolada.
 - Bugfix Branches: Utilizadas para corrigir erros identificados.
 - Release Branches: Preparação para novas versões do sistema. Essa estratégia de

branches, combinada com commits descritivos, não apenas auxiliou na organização durante o desenvolvimento pelos próprios autores, mas também preparou o projeto para a integração de futuros desenvolvedores.

RESULTADOS E DISCUSSÃO

É possível definir o Estudo de Impacto Ambiental – EIA como um documento técnico-científico baseado no levantamento ambiental dos meios socioeconômico, biótico e físico, avaliação dos possíveis impactos ambientais de projeto, aceção dos impactos positivos e negativos, sugestão de programas de monitoramento e acompanhamento ambiental. Já Relatório de Impacto Ambiental – RIMA, é um documento elaborado com base nas informações de conclusão contidas no EIA, sendo exibido de formas simples e objetiva a inclusão de toda a população. Serão apresentados nesta mesma etapa algumas Audiências Públicas com intuito e participação de toda a população afetada ou interessada no assunto seja examinada (Ibama, 2014).

Os resultados deste trabalho são apresentados e discutidos com base na implementação das funcionalidades que foram diretamente alinhadas às necessidades reais identificadas durante a fase de levantamento de requisitos com a microempresendedora. Embora o sistema ainda esteja no estado de protótipo funcional e não tenha sido implantado em um ambiente de produção em larga escala no momento da redação deste artigo, as decisões técnicas e de design foram cuidadosamente fundamentadas tanto nas observações práticas da rotina da usuária quanto nas boas práticas estabelecidas pela engenharia de software.

As funcionalidades desenvolvidas foram analisadas quanto à sua aderência aos objetivos do projeto, sua capacidade efetiva de resolver os problemas previamente diagnosticados na gestão manual e seu potencial para gerar melhorias significativas nos processos de controle, organização e, fundamentalmente, na capacidade de tomada de decisões estratégicas pela empresenedora.

Para oferecer uma compreensão mais concreta das soluções desenvolvidas, apresentamos a seguir algumas imagens (capturas de tela) das principais interfaces do sistema, acompanhadas de textos explicativos que descrevem os elementos visuais (templates), as funcionalidades específicas de cada módulo e os benefícios esperados que cada tela e funcionalidade trarão para a rotina da MEI. Estas ilustrações servem como evidência visual do alinhamento entre os requisitos levantados na fase

inicial e as soluções técnicas que foram efetivamente implementadas¹⁵.

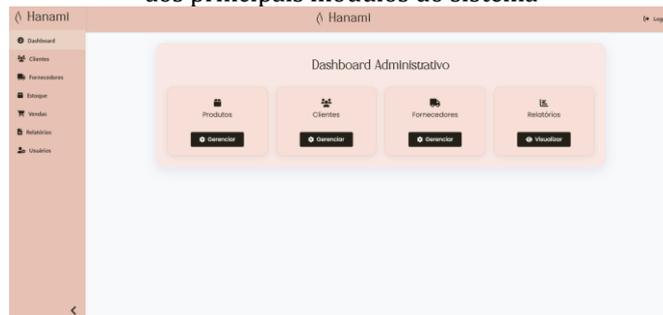
Figura 1. Página de *Login* do sistema



Fonte: Elaborada pelo Autor

Conforme a Figura 1, a tela de *login* foi projetada com foco na simplicidade estética e na segurança. Utilizando o framework Bootstrap, garantiu-se que a interface fosse responsiva, adaptando-se a diferentes tamanhos de tela, desde dispositivos móveis até desktops. O processo de autenticação dos usuários é gerenciado pelo Laravel Sanctum, proporcionando uma camada robusta de segurança através do uso de *tokens* pessoais e um mecanismo de validação eficiente das credenciais de acesso. Este ponto de entrada seguro é fundamental para garantir que apenas usuários autorizados tenham acesso aos dados sensíveis armazenados no sistema, como informações de estoque, registros de vendas, dados de clientes e fornecedores. A segurança na autenticação protege o negócio contra acessos indevidos e a perda ou alteração não autorizada de informações vitais.

Figura 2. Página de *Dashboard* administrativo com acesso aos principais módulos do sistema



Fonte: Elaborada pelo Autor

Conforme a Figura 2, o *dashboard*, ou painel administrativo, funciona como o centro de comando do sistema, oferecendo uma visão geral imediata e concisa da operação da empresa. Nesta tela, estão dispostos atalhos visuais em destaque que permitem acesso rápido aos módulos essenciais: cadastro/gerenciamento de produtos, vendas, clientes e fornecedores. Mais importante, o *dashboard* apresenta indicadores-chave de desempenho (KPIs), como o volume de vendas e o status do estoque. A visualização rápida desses indicadores auxilia a microempresenedora na tomada de decisões rápidas

e mais estratégicas. Por exemplo, um alerta de baixo estoque no *dashboard* pode indicar a necessidade imediata de realizar um pedido de compra a um fornecedor, enquanto a análise do volume de vendas pode ajudar a identificar produtos com alta saída e planejar promoções ou compras futuras. A capacidade de ter essas informações agregadas e visíveis em um único local reduz a necessidade de consultar múltiplos registros manuais e fornece um panorama claro da saúde do negócio em tempo real.

Figura 3. Página de formulário de cadastro de cliente



Fonte: Elaborada pelo Autor

Conforme a Figura 3, esta interface foi desenvolvida para permitir o registro fácil e organizado de novos clientes. O formulário contém campos essenciais e bem definidos, como nome, email, CPF/CNPJ, telefone, endereço completo e um campo para o total consumido pelo cliente. A interface foi cuidadosamente desenhada com um *layout* simples e intuitivo, buscando reduzir a probabilidade de erros de preenchimento por parte do usuário e garantir a integridade dos dados. O cadastro completo e preciso dos clientes é um componente essencial para diversas funcionalidades do sistema. Ele permite a vinculação das vendas a clientes específicos, o que, por sua vez, habilita a geração de relatórios segmentados por cliente, a criação de histórico de compras individual e a possibilidade de realizar ações de marketing direcionadas.

Figura 4. Página de Detalhes de fornecedor com histórico de compras.

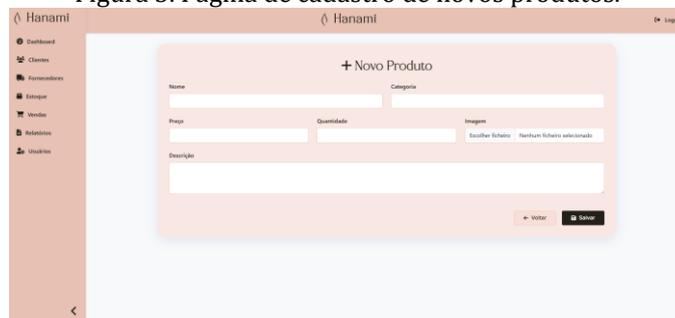


Fonte: Elaborada pelo Autor

Conforme a Figura 4, esta funcionalidade aborda diretamente uma das maiores dificuldades relatadas pela microempresendedora na gestão manual: a perda ou dificuldade em localizar o contato e o histórico de

compras dos fornecedores. A tela de detalhes do fornecedor exibe de forma centralizada todas as informações de contato do fornecedor (nome, contato, telefone, email, endereço), e, crucialmente, apresenta uma lista organizada das compras já realizadas com aquele fornecedor específico, incluindo data, produto comprado, valor pago e descrição. Um indicador útil do Total em Compras com aquele fornecedor também é exibido. Ter este histórico facilmente acessível dentro do sistema facilita enormemente o processo de reposição de estoque, a comparação de preços entre fornecedores e a negociação de futuras compras. O tempo que antes era gasto "revirando aplicativos" ou anotações para encontrar um fornecedor é drasticamente reduzido. A capacidade de adicionar uma "Nova Compra" diretamente desta tela agiliza o registro das entradas de estoque e mantém o histórico atualizado.

Figura 5. Página de cadastro de novos produtos.



Fonte: Elaborada pelo Autor

Conforme a Figura 5, a tela de cadastro de produtos é fundamental para estabelecer um controle preciso e organizado do estoque. O formulário foi otimizado para a entrada das informações essenciais de cada item. Inclui campos como nome do produto, categoria, preço de venda, quantidade disponível em estoque e, pertinentemente para produtos aromáticos, a validade. Há também um campo para descrição detalhada do produto e a possibilidade de associar uma imagem. Esta organização no cadastro evita duplicações de produtos, facilita futuras auditorias de estoque e é a base para manter o estoque sempre atualizado em tempo real conforme as vendas e compras são registradas no sistema. Um cadastro de produto bem estruturado com informações completas é vital para a precisão do controle de estoque exibido no *dashboard* e nos relatórios.

Conforme a Figura 6, a tela de relatórios é uma das funcionalidades mais estratégicas do sistema, transformando os dados operacionais em informações acionáveis. Ela apresenta visualizações agregadas das movimentações do negócio, como o total de vendas realizadas, o faturamento total arrecadado, o ticket médio por venda e o status das vendas pendentes (se aplicável). Os relatórios são dinâmicos, permitindo o uso de filtros por período e, potencialmente, por tipo de operação, produto ou cliente. Essa capacidade de

filtrar e visualizar dados históricos permite à empreendedora acompanhar a performance do negócio ao longo do tempo, identificar produtos mais vendidos, entender o comportamento de compra dos clientes e basear suas decisões de compra, precificação e marketing em dados concretos, e não apenas em intuição. A opção de "Imprimir Relatório" facilita a análise offline ou o compartilhamento das informações.

Figura 6. Página de relatórios com indicadores de venda.



Fonte: Elaborada pelo Autor

Em suma, esta seção, ilustrada pelas figuras apresentadas, demonstra como cada funcionalidade implementada foi diretamente pensada para resolver os problemas identificados no levantamento de requisitos. A transição de métodos manuais para um sistema digital, mesmo em ambiente de teste, evidencia o potencial para melhorar significativamente a organização e a eficiência da microempresa. As decisões técnicas, embasadas nas boas práticas da engenharia de software, resultaram em uma solução que, embora necessite de implantação prática para validar seu impacto total em produção, já demonstra em sua estrutura a capacidade de oferecer à microempreendedora as ferramentas necessárias para uma gestão mais profissional e informada.

TRABALHOS FUTUROS

Como continuidade deste projeto e para validar plenamente o impacto e a eficácia do sistema desenvolvido, sugerem-se os seguintes trabalhos futuros:

1. **Implantação Prática do Sistema:** O passo mais imediato e crucial é a implantação efetiva do sistema em ambiente de produção na microempresa do estudo de caso21.... Esta etapa envolverá o deploy técnico da aplicação (possivelmente em um serviço de hospedagem web acessível), a configuração final do banco de dados em ambiente real, e o treinamento da usuária final e de potenciais colaboradores no uso cotidiano do sistema.
 - 1.1 A implantação prática pode envolver desafios como a escolha do ambiente de hospedagem adequado e com custo compatível para um MEI, a migração de dados históricos (se

existirem em formatos digitais) para a base de dados do sistema, e a adaptação da rotina da empreendedora para incorporar o uso diário da ferramenta.

2. **Coleta de Feedbacks Pós-Uso:** Uma vez que o sistema esteja em operação, será fundamental realizar a coleta sistemática de feedbacks da usuária após um período de uso real. Este feedback será valioso para identificar pontos fortes, funcionalidades que podem ser otimizadas, eventuais bugs não detectados em testes manuais, e novas necessidades que possam surgir com a mudança da rotina. A coleta pode ser feita através de entrevistas periódicas, formulários de avaliação ou canais diretos de comunicação. O feedback real de uso em produção é a fonte mais rica para direcionar a evolução futura da plataforma.
3. **Evolução da Plataforma com Novas Integrações:** Com base nos feedbacks coletados e na visão de crescimento do negócio, o sistema pode ser evolucionado através da implementação de integrações estratégicas. As integrações sugeridas incluem:
 - **Integração com Soluções de Pagamento:** Conectar o sistema a gateways de pagamento online (como Mercado Pago, PagSeguro APIs) ou a sistemas de TEF (Transferência Eletrônica de Fundos) poderia simplificar e agilizar o processo de venda, permitindo que os pagamentos fossem registrados diretamente no sistema de gestão, conciliando vendas e recebimentos. O benefício é a redução de erros manuais na conciliação financeira e uma visão mais precisa do fluxo de caixa. O desafio técnico envolve a comunicação segura via API, o tratamento de diferentes métodos de pagamento e a gestão de status de transação (pago, pendente, cancelado).
 - **Desenvolvimento de Aplicativos Móveis:** Criar uma versão mobile nativa ou PWA (Progressive Web App) permitiria que a microempreendedora e seus colaboradores acessassem as funcionalidades principais do sistema (como registro de vendas e consulta de estoque) de qualquer lugar, aumentando a agilidade e a flexibilidade da operação. O sistema já está preparado para isso graças à arquitetura REST e ao uso do Laravel Sanctum para autenticação via API. O desafio técnico reside no desenvolvimento da interface e lógica na plataforma mobile e na garantia de comunicação eficiente com o *backend*.
 - **Integração com Marketplaces e Plataformas de E-commerce:** Conectar o sistema de gestão a plataformas de venda online como Nuvemshop, Loja Integrada ou marketplaces

(Mercado Livre, Shopee via APIs) permitiria centralizar a gestão de estoque e pedidos. Isso resolveria o problema de manter o estoque sincronizado em múltiplos canais de venda, evitando vendas de produtos indisponíveis. O benefício é a expansão do alcance do negócio sem perder o controle operacional. O desafio técnico envolve o consumo das APIs dos marketplaces/plataformas, a lógica de sincronização bidirecional (vendas no marketplace atualizam o estoque no sistema e vice-versa) e o tratamento de possíveis inconsistências de dados.

- Estas etapas futuras visam não apenas aprimorar as funcionalidades existentes, mas também ampliar significativamente o alcance e a aplicabilidade da solução desenvolvida²², transformando-a em uma ferramenta ainda mais poderosa para o crescimento e a profissionalização de microempreendedores.

CONCLUSÕES

Este estudo descreveu o processo de desenvolvimento de um sistema web de gestão de estoque, direcionado especificamente para atender às necessidades de uma microempreendedora individual atuante no setor de produtos aromáticos, localizada na cidade de Medianeira-PR. O ponto de partida foi a identificação de dificuldades recorrentes enfrentadas pela usuária em sua gestão manual, que incluíam a perda de informações de contato e histórico de fornecedores, a ausência de um controle estruturado e confiável de produtos e vendas, e a carência de indicadores claros para auxiliar na tomada de decisões estratégicas para o negócio. A partir dessa análise aprofundada, foi possível propor e desenvolver uma solução digital sob medida, adaptada à realidade operacional e às limitações de recursos comuns a microempreendedores.

A aplicação foi desenvolvida seguindo rigorosamente os princípios da engenharia de software, com um foco claro em garantir escalabilidade, organização e segurança. As funcionalidades implementadas abordam diretamente os problemas identificados, oferecendo controle de estoque com atualizações em tempo real, um sistema completo para cadastro e gerenciamento de clientes e fornecedores (incluindo histórico de compras), um painel administrativo que centraliza indicadores estratégicos e a capacidade de gerar relatórios detalhados. Cada módulo e interface foi desenhado com particular atenção à usabilidade e à simplicidade, tendo como base as reais necessidades e o feedback levantado junto à usuária final durante o processo de desenvolvimento.

Embora o sistema ainda se encontre em fase de protótipo funcional e não tenha sido formalmente implantado em ambiente de produção, sua estrutura e as funcionalidades desenvolvidas demonstram um potencial significativo. Espera-se que a adoção desta ferramenta possa reduzir falhas inerentes aos processos manuais, aumentar a eficiência operacional ao centralizar e automatizar tarefas, e fornecer insights baseados em dados que auxiliem no crescimento sustentável de pequenos negócios, especialmente no contexto vital dos microempreendedores individuais no Brasil.

A proposta deste trabalho reforça a importância da digitalização como uma ferramenta estratégica indispensável para a profissionalização da gestão de negócios que operam, mesmo que parcialmente, na informalidade ou com poucos recursos tecnológicos. Além disso, o estudo destaca o papel transformador do desenvolvedor de software não apenas como um técnico, mas como um agente capaz de aplicar seus conhecimentos para resolver problemas práticos e contribuir diretamente para a realidade social e econômica de microempreendedores, impulsionando seus negócios através da inovação.

REFERÊNCIAS

- CHACON, Scott; STRAUB, Ben. **Pro Git**. 2. ed. Berkeley: Apress, 2014. Disponível em: <https://git-scm.com/book/en/v2>. Acesso em: 22 maio 2025.
- E-COMMERCE BRASIL. **98% das MPMEs que digitalizaram operações viram impactos positivos**, 2023 Disponível em: <https://www.ecommercebrasil.com.br/noticias/mpmes-transformacao-digital-reconhecem-impacto-positivo-tecnologia>. Acesso em: 22 maio 2025.
- GOUVÊA, Marcos. **Omnichannel: a estratégia do varejo sem fronteiras**. São Paulo: GS&MD, 2017.
- LARAVEL. **Laravel documentation**. Disponível em: <https://laravel.com/docs>. Acesso em: 14 maio 2025.
- LARAVEL. **Laravel Sanctum – documentation**. Disponível em: <https://laravel.com/docs/sanctum>. Acesso em: 14 maio 2025.
- MDN WEB DOCS. **AJAX and JSON – Using asynchronous calls**. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>. Acesso em: 14 maio 2025.

PAUL, Dubois. **MySQL: guia do programador**. Rio de Janeiro: Alta Books, 2014.

PREECE, Jenny; ROGERS, Yvonne; SHARP, Helen.
Design de interação: além da interação homem-computador. 3. ed. Porto Alegre: Bookman, 2013. Acesso em: 22 maio 2025

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

SANTOS, Vagner da Silva. **Desenvolvimento web com Laravel: teoria e prática com aplicações PHP**. São Paulo: Novatec, 2020.

SEBRAE. **Boletim estatístico de MEI – dados nacionais 2023**. Brasília: Sebrae Nacional, 2023. Disponível em: <https://datasebrae.com.br>. Acesso em: 14 maio 2025.

SEBRAE. **Perfil do microempreendedor individual (MEI)**. Brasília: Sebrae Nacional, 2023. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/artigos/brasil-tem-quase-15-milhoes-de-microempreendedores-individuais,e538151eea156810VgnVCM1000001b00320aRCRD> Acesso em: 14 maio 2025.

WILLIAMS, Robin. **Não me faça pensar: uma abordagem de bom senso à usabilidade na web**. 2. ed. São Paulo: Alta Books, 2013.