

Pediline: Solução SaaS para Gerenciamento de Catálogo e Delivery

Filipe Brener, Yan Guilherme, Liziane Santos Soares

Instituto de Ciências Exatas e Tecnológicas

Universidade Federal de Viçosa

Rio Paranaíba, Brasil

{filipe.brener, yan.leite, liziane.soares}@ufv.br

Resumo—Wearable technology and easy access to the internet have revolutionized various daily tasks, promoting convenience and comfort for users. This work aims to build and document delivery automation software compatible with different business sectors, specifically targeting small towns that do not yet have access to large brand applications. The proposed system is a multi-tenancy SaaS, developed in Java with the support of the Spring Framework. It will offer two user interfaces: one for managers, who will be able to manage products and sales, and another for end consumers, who will have access to an online catalog to make purchases. This paper also explores the impact of the pandemic on the growth of delivery services in small towns and the gap filled by Pediline.

Index Terms—SaaS, delivery, e-commerce, multi-tenancy, online catalog.

I. INTRODUÇÃO

Nos últimos anos, o acesso à *internet* via *smartphones* é algo comum na vida da maioria dos brasileiros, sendo ele o principal meio de acesso à *internet* em 99,5% dos domicílios com acesso à grande rede. Em segundo lugar como principal dispositivo para conectar-se à *internet*, vem a TV com 44,4% das residências. O computador pessoal fica em terceiro lugar, com 42% das moradias [1]. Devido a facilidade e praticidade de um *smartphone*, uma simples busca já oferece quase todas as informações úteis e necessárias.

Essa ampla conectividade e a conveniência proporcionada pelos dispositivos móveis também têm impulsionado outras áreas, especialmente o comércio eletrônico (*e-commerce*). Devido à evolução tecnológica e à alta conectividade entre as pessoas nos últimos anos, o *e-commerce* teve um avanço significativo, mudando a maneira de vender e de comprar produtos. Com início nos meados de 2000, onde as primeiras lojas *online* no Brasil foram criadas, o *e-commerce* vem ganhando força e faturando cada vez mais com o passar dos anos. Em 2022 o *e-commerce* atingiu a marca de R\$169.59 bilhões de faturamento, um aumento de mais de R\$18 bilhões em relação ao faturamento de 2021. [2]

Durante o período de pandemia causada pela *COVID-19*, um dos ramos que mais se destacou foi o dos serviços de *delivery*. Segundo a ABRASEL em meados de 2021,

89% dos estabelecimentos possuíam o serviço de entrega a domicílio. A difusão de aplicativos de compra de comida *online* e o isolamento social causado pela *COVID-19* criaram um novo costume na vida dos cidadãos de pedir comida somente usando o celular, sem sair de casa. [3]

Devido a essas mudanças culturais e socioeconômicas, o mercado consumidor tornou-se mais exigente quanto à comodidade de realizar compras sem sair de casa, obrigando o comércio produtor a inovar seus métodos de venda. Com a popularização da *internet* cada vez mais veloz e de celulares mais potentes, a venda pela *internet*, independentemente do ramo, torna-se uma estratégia significativa para aumentar o faturamento. Desde de lojas consolidadas a pequenos negócios, o uso da *internet* para venda e exposição dos produtos é essencial para um alcance maior de possíveis compradores.

Muitos comerciantes já utilizam de aplicativos de *delivery* prontos e disponíveis no mercado, como o *iFood*, *Rappi*, e *UberEats*. Contudo há uma limitação quanto ao tipo de produto a ser vendido por esses aplicativos, que são restritos à alimentos. Outro problema é a disponibilidade desses aplicativos em cidades de pequeno porte, onde existe uma comércio interno de produtos como cosméticos, roupas, acessórios e outros. Além disso, o custo para utilização dos serviços não costumam serem baratos, com taxas relativas às vendas na plataforma.

Diante deste contexto surgiu a ideia central deste projeto, desenvolver o Pediline, que será uma aplicação *web* portátil e independente do sistema operacional utilizado. A arquitetura escolhida é a de um *SaaS* (*Software as a Service*) *multi-tenancy*. O objetivo é a construção de um *software* como serviço que atenderá a todos que desejam vender e exibir seus produtos *online* e fazer a entrega do mesmo, aplicável a qualquer município brasileiro. O sistema almeja contribuir com a economia da cidade, possibilitando uma redução de custo de utilização pelo comerciante, aumento do lucro do estabelecimento e mais praticidade e comodidade na vida do cliente final.

II. REFERENCIAL TEÓRICO

Nesta seção será apresentado um panorama sobre o domínio de negócio foco do sistema a ser desenvolvido, que abrange operações de venda e *delivery* local/regional

em segmentos diversos de negócio. E serão descritos os conceitos e tecnologias envolvidas no desenvolvimento do sistema, além da infraestrutura do mesmo.

A. Contexto de delivery em vendas locais

Em pequenas cidades do Brasil o comércio local influencia diretamente no cotidiano dos habitantes, tanto socialmente quanto economicamente. Observa-se maior suporte para segmentos de *delivery* ligados ao setor alimentício, contudo, outros comércios como papelarias, quitandas, material de construção e vários outros ainda não são bem atendidos por *software* de gerência de catálogo de produtos e *delivery*.

Uma das soluções encontrada por esses comerciantes é a venda por meio de redes sociais. Segundo pesquisa da *Nuvem Shop*, vendas por meio de redes sociais representaram 21% do total de todas as transações em 2018, com o *Instagram* liderando com 59% das operações. O segundo lugar fica com o *Facebook* que representou 37% dos atos de compra e venda. [4] Outro aspecto interessante da pesquisa se refere aos segmentos de mercados presente nas redes sociais, que são bastante diversificados. Mesmo já existindo diversos sites e aplicativos focados no gerenciamento de *delivery* de comida e bebida, há outros segmentos que ainda podem ser explorados. Por exemplo, categorias como Presentes, e Artesanato, que representam uma parcela menor do *e-commerce*, indicam oportunidades para inovação e desenvolvimento de novos serviços de *delivery*, atendendo a nichos específicos que ainda não foram completamente explorados.

SEGMENTO	TRANSAÇÕES
Moda e Vestuário	38%
Outros	21%
Saúde e Beleza	12,4%
Acessórios	11,5%
Casa e Jardim	4,9%
Artesanato	3,1%
Comidas e Bebidas	2,6%
Eletrônicos	2,4%
Produtos Eróticos	2,2%
Presentes	1,9%

Tabela I

TOP 10 SEGMENTOS DE E-COMMERCE POR TRANSAÇÕES. FONTE: NUDEM SHOP 2018

Assim como o *Facebook* e o *Instagram*, o *WhatsApp* também permite a criação de catálogos para negócios, oferecendo uma plataforma onde os usuários podem visualizar informações de produtos e entrar em contato diretamente com o estabelecimento. Esses três aplicativos, são atualmente controlados e gerenciados pela *Meta*. Mesmo com todos os sistemas, as redes sociais não possuem como fins específicos a compra e venda de produtos, o que causa algumas limitações nessas transações, como por exemplo a falta de pagamentos integrados, personalização específica de cada estabelecimento, gerenciamento de *delivery*, cupons de desconto exclusivos e outros.

B. SaaS multi-tenancy

Software como serviço, ou *Software as a Service (SaaS)*, é um tipo de serviço em nuvem no qual os aplicativos são disponibilizados como um serviço. Em vez de adquirirem licenças, os clientes pagam apenas pelo acesso e pelo uso das funcionalidades oferecidas por meio da *internet*. [5]

Aplicações *multi-tenancy* são soluções que permitem que diferentes clientes compartilhem os mesmos recursos de *hardware* através de uma única instância da aplicação. No entanto, cada cliente pode configurar a aplicação de acordo com suas necessidades, como se estivesse utilizando um ambiente dedicado. [6]

Ao utilizar tais tecnologias para desenvolvimento de um sistema, alguns devidos cuidados no gerenciamento dos *tenants* devem ser levados em consideração. Por outro lado, a arquitetura apresenta melhoras quanto a escalabilidade e redução de custos. Segundo Neto [7], *SaaS e Cloud Computing* mostra eficiência quanto ao uso de *hardware* e rede. Uma vez que vários consumidores podem usar a mesma instância do *software*, os preços para os assinantes do recurso tendem a cair. Outra vantagem fica no aproveitamento de *datacenters* que não são utilizados todo o tempo. Esse cenário impacta diretamente no *Pediline* e em aplicações comuns, no fato que o aumento no número de clientes não afeta diretamente a necessidade de melhoria no servidor *SaaS*.

C. Tecnologias utilizadas

1) *Angular*: O *Angular* é um *framework* para *front-end* criado e gerenciado pela *Google* que utiliza a abordagem orientada a componentes para criar aplicações *web* dinâmicas e de página única. O *Angular* é um projeto *open-source*, escrito em *TypeScript*. Nesta pesquisa, este *framework* é utilizado na implementação do painel do gestor [8].

2) *Docker*: *Docker* é uma plataforma que facilita a criação, o desenvolvimento e a execução de aplicações em *containers*, que são pacotes padronizados que incluem todas as dependências e bibliotecas necessárias para rodar uma aplicação em qualquer ambiente. Isso elimina problemas de compatibilidade entre ambientes de desenvolvimento, testes e produção [9]. No desenvolvimento do *Pediline*, o *Docker* foi utilizado para criar e gerenciar os *containers* que hospedam os serviços da aplicação, facilitando a automação de *deploy* e gerenciamento do ambiente.

3) *GitHub*: O *GitHub* é uma plataforma de hospedagem de código-fonte amplamente utilizada, que facilita a colaboração entre desenvolvedores e o versionamento de código [10]. Juntamente com o *GitHub Actions*, é possível implementar *pipelines* de Integração Contínua (*CI*) e Entrega Contínua (*CD*), automatizando testes, *builds* e *deploys* [11].

4) *React*: O *React* é uma biblioteca *JavaScript* desenvolvida e mantida pelo *Facebook*, amplamente utilizada para construir interfaces de usuário (*UI*) dinâmicas e interativas em aplicações *web*. Neste projeto, utilizamos o *Next.js* como *framework* principal para o desenvolvimento

do *front-end*, com *React* como a biblioteca para a construção dos componentes da interface [14].

5) *Spring*: O *Spring Framework* é um *framework open-source* para *Java* que fornece suporte para a criação de aplicações robustas, seguras e escaláveis. O foco do *framework* é na velocidade, simplicidade e produtividade, o que o torna um dos *frameworks* mais populares entre os desenvolvedores *Java*. Além disso, ele oferece o *Spring Boot*, que facilita o desenvolvimento e a configuração de projetos *Spring* através de convenções e um arquivo de configuração simplificado. O *Spring* será utilizado para criar o módulo *core* do sistema, que será a API que todo o sistema utilizará [13].

D. Infraestrutura

Para a implementação deste projeto, a *Amazon Web Services (AWS)* foi escolhida como a plataforma de infraestrutura devido à sua ampla gama de serviços, escalabilidade, confiabilidade e suporte a diversas soluções tecnológicas, oferecendo a flexibilidade necessária para atender aos requisitos da aplicação [12].

Em seguida, será apresentada uma explicação detalhada da função de cada componente da infraestrutura.

- **Instância EC2:** Hospeda o *back-end Spring* (*core* do sistema) e o *front-end Next.js*. Essa arquitetura permite que o *front-end* obtenha os dados do *back-end* com baixa latência, aproveitando a comunicação local na mesma máquina.
- **Bucket S3:** Armazena e distribui o módulo do gestor (*dashboard* em *Angular*) e outros conteúdos estáticos, garantindo alta disponibilidade e escalabilidade.
- **Amazon CloudFront:** Utilizado para distribuir o conteúdo estático globalmente e atuar como proxy *HTTPS*. Ele serve as páginas estáticas diretamente, e, caso o conteúdo não esteja presente no cache distribuído, faz o *fetch* do S3.
- **Elastic Load Balancer (ELB):** Implementado para gerenciar o tráfego de entrada e garantir a segurança no acesso *HTTPS*. Ele direciona as requisições para a instância *EC2*, onde o *front-end* dinâmico e o *core* da aplicação estão hospedados.
- **Route 53:** Gerencia o *DNS* do sistema, direcionando o tráfego para o *Amazon CloudFront*, que serve o conteúdo estático, e para o *ELB*, que por sua vez direciona as requisições dinâmicas para o *EC2*. Essa configuração garante que cada tipo de conteúdo seja servido da forma mais eficiente possível.
- **AWS Certificate Manager:** Utilizado para fornecer certificados *SSL/TLS* de forma gratuita. Esses certificados são essenciais para habilitar o protocolo *HTTPS*, que é uma versão segura do *HTTP*. O *HTTPS* criptografa os dados trocados entre o cliente e o servidor, garantindo confidencialidade e proteção contra ataques de interceptação (*man-in-the-middle*). Os certificados emitidos pelo *AWS Certificate Manager* são usados tanto no *CloudFront* quanto no *ELB*,

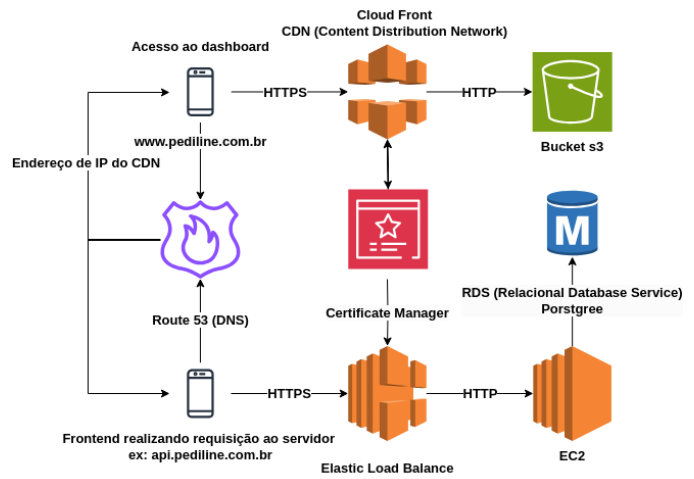


Figura 1. Infraestrutura do sistema

permitindo a comunicação segura com os usuários. Dentro da rede *AWS*, a comunicação entre o *CloudFront* e o *S3*, assim como entre o *ELB* e a instância *EC2*, ocorre via *HTTP*. Isso simplifica a configuração dos servidores, uma vez que o gerenciamento de certificados na instância não é necessário, reduzindo a complexidade operacional.

- **RDS com PostgreSQL:** Um banco de dados relacional gerenciado pela *AWS*, utilizado para armazenar os dados da aplicação. O *PostgreSQL* foi escolhido por sua robustez, flexibilidade e suporte a operações complexas, além de ser altamente escalável. O *RDS* facilita a administração, com *backups* automáticos, atualizações de *software* e alta disponibilidade configurável, garantindo que o banco de dados seja confiável e eficiente para suportar a aplicação *multi-tenancy*.

III. METODOLOGIA

O desenvolvimento do Pediline seguiu uma abordagem modular e orientada a serviços, com uma arquitetura desenhada para atender as necessidades de escalabilidade e segurança exigidas por um sistema *SaaS multi-tenancy*. A seguir, detalhamos os principais componentes do sistema:

A. Backend

O *backend* está sendo desenvolvido utilizando o *Spring Framework*, uma solução robusta para criação de APIs REST. O *Spring Boot* facilitou a configuração e o gerenciamento das dependências, proporcionando agilidade no desenvolvimento.

B. Frontend

O catálogo online está sendo desenvolvido utilizando *Next.js*, um framework *React* com suporte a *Server-Side Rendering* (SSR), garantindo performance e SEO aprimorados. Além disso, o painel de gestão, destinado aos

lojistas, proporcionando uma interface rica e responsiva para o gerenciamento de pedidos e produtos.

C. Infraestrutura em Nuvem

A infraestrutura do sistema foi projetada utilizando serviços da *Amazon Web Services* (AWS), como o *Elastic Load Balancer* (ELB), *EC2* para hospedagem, e *S3* para armazenamento de conteúdo estático. A escalabilidade é garantida pela utilização do *CloudFront* e *RDS* para o gerenciamento de dados.

IV. TRABALHOS RELACIONADOS

Devido à modernização e às mudanças tanto nos hábitos humanos quanto no mercado, o ato de comprar sem sair de casa aumentou consideravelmente. Esse fenômeno é impulsionado por uma série de fatores, incluindo a crescente adoção de tecnologias digitais e a evolução das plataformas de *e-commerce*. As pessoas estão cada vez mais em busca de conveniência e eficiência, o que impulsiona ainda mais o crescimento das compras *online*. Vários trabalhos acadêmicos, estudos de caso e pesquisas sobre o tema têm abordado a evolução das plataformas de *e-commerce* e seus impactos no mercado. Nesta seção, alguns desses trabalhos serão destacados.

"A modernização da tecnologia portátil tem sido um catalisador para o desenvolvimento de novas aplicações móveis, proporcionando experiências de usuário cada vez mais sofisticadas" [21, p. 123].

A crescente popularidade de aplicativos de *delivery* tem transformado a forma como as pessoas compram alimentos e outros produtos. De acordo com Melo [20], enquanto grandes cidades se beneficiam de uma ampla gama de serviços de *delivery*, as cidades menores ainda enfrentam limitações significativas. A solução do autor foi o desenvolvimento de um aplicativo *mobile* para atender o *delivery* em cidades do interior. A diferença prática entre o trabalho dele e este é o fato que como é um aplicativo, será necessário o *download* do aplicativo antes do uso. O Pediline é uma solução totalmente *web*, sem necessidade de *download*.

A *24SevenCommerce* [16] é uma empresa que oferece soluções de tecnologia para integração de sistemas de ponto de venda (PDV) com plataformas de comércio eletrônico e ressalta como o gerenciamento de catálogo é peça fundamental para o sucesso de negócios de *e-commerce*, pois influencia diretamente a experiência do cliente e as vendas. Um catálogo bem gerenciado facilita a navegação do cliente, oferece uma busca mais efetiva, e assegura a precisão das informações de cada produto. Um dos produtos da *24Seven Commerce* é o *Octopus*, que é uma plataforma de integração que conecta sistemas PDV com *marketplaces*, plataformas de *e-commerce* e outros canais de venda. O público alvo do *Octopus* são empresas de varejo que operam tanto fisicamente quanto *online* que desejam integrar todo o sistema. Já o público alvo do Pediline são comerciantes que desejam facilitar a exposição

e venda seus produtos na *internet* e realizar a entrega em domicílio para o comprador.

A escolha por usar um sistema de gerenciamento de catálogo eficaz proporciona diversas vantagens, como a melhoria na organização interna da empresa, maior visibilidade dos produtos no mercado, manutenção do inventário e informações de produtos em tempo real, e a possibilidade de oferecer recomendações personalizadas aos clientes. Essas informações são fornecidas pela Anchanto [18], uma empresa de tecnologia que fornece soluções de *software* voltadas para o gerenciamento de operações de *e-commerce* e logística, fundada em 2011. A empresa ainda ressalta que todos esses fatores favorecem para uma experiência do cliente mais satisfatória e podem levar a um aumento nas vendas. No entanto, a falta de um gerenciamento adequado de catálogo pode ter consequências negativas significativas, como experiências ruins para os clientes, perda de vendas e danos à reputação da marca. As soluções da Anchanto estão em segmentos diferentes em relação ao Pediline; seus produtos são voltados para gerenciamento de diferentes *marketplaces* em um único lugar, enquanto o Pediline deseja facilitar o gerenciamento de vendas e produtos de um estabelecimento que deseja vender *online*.

Um dos serviços mais famosos em atividade deste ramo é o *Shopify*. O *Shopify* é uma solução *SaaS* (*Software as a Service*) projetada para permitir que indivíduos e empresas criem e gerenciem suas próprias lojas virtuais com facilidade. Nas funcionalidades do *Shopify* está inclusa a personalização de lojas *online*, gestão de inventário, processamento de pagamentos e integração com várias ferramentas de *marketing*. Além disso, o *Shopify* oferece como benefícios a simplicidade de uso, a escalabilidade e o suporte técnico oferecido pela plataforma *Shopify* [17]. Assim como a Anchanto, o *Shopify* oferece soluções para gerência de diferentes *marketplaces*.

V. CONCLUSÃO E TRABALHOS FUTUROS

O Pediline está em desenvolvimento e se apresenta como uma solução promissora para o gerenciamento de *delivery* e vendas *online* em pequenas cidades.

Um dos focos atuais do desenvolvimento é a criação de um painel para o gestor, que será enriquecido com funcionalidades essenciais para facilitar o gerenciamento de produtos e pedidos. Este painel terá como objetivo otimizar a experiência do usuário, permitindo um controle mais eficiente das operações.

Além disso, a integração com o WhatsApp, embora planejada, será considerada em etapas posteriores do projeto. É importante ressaltar que, para consumir a API do WhatsApp, o Facebook exige que o sistema passe por um processo de verificação, conhecido como "App Review", no qual será necessário apresentar provas da funcionalidade e conformidade do sistema. Portanto, a implementação dessa integração será cuidadosamente planejada e abordada ao final do ciclo de desenvolvimento do Pediline.

Futuramente, o sistema também será expandido para incluir funcionalidades avançadas, como o rastreamento em tempo real de pedidos e a integração com outras plataformas de pagamento.

REFERÊNCIAS

- [1] IBGE. Internet já é acessível em 90,0% dos domicílios do país em 2021. 2023.
- [2] ABCOMM. Previsão de vendas *online*. 2024.
- [3] ABRASEL. *Delivery e pandemia: como o mercado se adaptou e cresceu*. 2024.
- [4] AWAKE, B. Redes sociais foram responsáveis por 21% das vendas em 2018, revela pesquisa. 2018.
- [5] VERAS, Manoel. *Cloud Computing: nova arquitetura da TI*. Brasport, 2012.
- [6] BEZEMER, Cor-Paul; ZAIDMAN, Andy. *Multi-tenant SaaS applications: maintenance dream or nightmare?*. In: PROCEEDINGS of the joint ercim workshop on software evolution (evol) and international workshop on principles of software evolution (iwpsel). 2010. p. 88–92.
- [7] NETO, Josino Rodrigues; GARCIA, Vinicius Cardoso; DE ALENCAR, Andréza Leite; DAMASCENO, Júlio César; ASSAD, Rodrigo Elia; TRINTA, Fernando. *Software as a Service: Desenvolvendo Aplicações Multi-tenancy com Alto Grau de Reuso*. Sociedade Brasileira de Computação, 2012.
- [8] ALURA. *O que é Angular.js e como começar a utilizar*. 2024. Disponível em: <https://www.alura.com.br/artigos/angular-js>. Acesso em: 9 ago. 2024.
- [9] DOCKER, INC. *Docker Documentation*. 2024. Disponível em: <https://docs.docker.com/>. Acesso em: 24 ago. 2024.
- [10] GITHUB, INC. *GitHub Documentation*. 2024. Disponível em: <https://docs.github.com/>. Acesso em: 24 ago. 2024.
- [11] GITHUB, INC. *GitHub Actions Documentation*. 2024. Disponível em: <https://docs.github.com/actions>. Acesso em: 24 ago. 2024.
- [12] AMAZON WEB SERVICES. *AWS Documentation*. 2024. Disponível em: <https://docs.aws.amazon.com/>. Acesso em: 20 ago. 2024.
- [13] SPRING. *Spring Framework Overview*. 2024. Disponível em: <https://spring.io/projects/spring-framework>. Acesso em: 5 ago. 2024.
- [14] ALURA. *O que é React.js e como começar a utilizar*. 2024. Disponível em: <https://www.alura.com.br/artigos/react-js>. Acesso em: 9 ago. 2024.
- [15] BEZEMER, C.-P.; ZAIDMAN, A. *Multi-tenant SaaS applications: maintenance dream or nightmare?* 2010.
- [16] 24SEVEN COMMERCE. *E-commerce Product Catalog Management*. 2024. Disponível em: <https://www.24sevencommerce.com/ecommerce-product-catalog-management/>. Acesso em: 18 ago. 2024.
- [17] SHOPIFY. *O que é Shopify?*. 2024. Disponível em: <https://www.shopify.com/br/blog/o-que-e-shopify>. Acesso em: 18 ago. 2024.
- [18] ANCHANTO. Catalogue Management 101: A Complete Guide for Your E-commerce Business. 2024.
- [19] MELO, C. F. d. A. et al. Bom Delivery: um aplicativo de delivery genérico para cidades do interior. 2019.
- [20] MELO, Caio Felipe de Alcântara et al. *Bom Delivery: um aplicativo de delivery genérico para cidades do interior*. Universidade Federal de Campina Grande, 2019.
- [21] LEE, Valentino; SCHNEIDER, Heather; SCHELL, Robbie. *Aplicações móveis: arquitetura, projetos e desenvolvimento*. Cidade, País: Editora Exemplo, 2004. p. 123.