

Spot-It: Aplicativo de Localização de Produtos

Halane Thayla Alves Lima¹, Leonardo dos Santos Mendes¹, Robson Amorim de Melo¹

(halanethayla305@gmail.com, leodsmf@gmail.com, robsonamorim15@gmail.com)

Professor orientador: Marcelo Gomes De Paoli

Coordenação de curso de Engenharia Da Computação

Resumo

Este trabalho apresenta um aplicativo que permite ao cliente localizar produtos em estabelecimentos comerciais físicos que estejam situados nas proximidades de sua localização atual, com base em dados de geolocalização. O sistema identifica automaticamente a posição do usuário e busca por lojas próximas que tenham o item desejado disponível, facilitando o acesso rápido e eficiente ao produto, sem a necessidade de grandes deslocamentos. O estudo visa a criação e validação de uma solução tecnológica que unifica a busca por produtos com a geolocalização, oferecendo dados dinâmicos e rotas em tempo real. Sua metodologia envolve a utilização da linguagem Java com o framework Spring Boot para o backend, a ferramenta Figma para o design da interface do frontend, e um banco de dados NoSQL (MongoDB) para o armazenamento de dados. *Além disso, uma API de mapas é utilizada para as funcionalidades de geolocalização e roteamento.* Os resultados da versão MVP (Minimum Viable Product) demonstram a eficiência do aplicativo, com taxa de acerto superior a 80% e tempo de resposta médio entre 200 ms e 500 ms, validando seu desempenho satisfatório em relação aos requisitos de tempo de resposta e estabilidade..

Palavras-chave: Geolocalização. Aplicativos móveis. API de mapas. Consulta de estoque em tempo real. Roteamento inteligente.

1. INTRODUÇÃO

Com o avanço das tecnologias móveis, a popularização dos smartphones e o crescimento do acesso à internet, o comportamento do consumidor brasileiro passou por grandes transformações. Hoje, os consumidores têm acesso facilitado a uma vasta gama de informações sobre produtos e serviços, comparam preços com agilidade e utilizam cada vez mais seus dispositivos móveis para realizar compras e pesquisas antes de visitar lojas físicas (ECONOMIA..., 2025; PATAH, 2018). Atualmente, espera-se que produtos e serviços estejam

disponíveis de forma rápida, prática e, preferencialmente, próximos à localização do usuário. No entanto, mesmo com tantas inovações, persiste uma lacuna significativa entre a busca online por produtos e a capacidade de localizar sua disponibilidade em tempo real em estabelecimentos físicos próximos. A gestão de estoque e a integração entre canais online e offline ainda são desafios importantes para o varejo brasileiro, podendo levar à frustração do consumidor quando um produto pesquisado online não está disponível na loja física (RC, 2024).

Muitos aplicativos de mapas, como aqueles que utilizam plataformas como a Google Maps Platform, possibilitam a busca por categorias de estabelecimentos, mas geralmente não fornecem informações sobre produtos específicos disponíveis em estoque no varejo local (SANT'ANNA, 2021). Da mesma forma, observa-se que os marketplaces frequentemente priorizam grandes empresas, o que pode deixar comércios locais com menor visibilidade ou capacidade de integração, uma situação que reflete os desafios tecnológicos e operacionais enfrentados por pequenos comércios para integrar seus estoques e informações geográficas (SILVA, 2022). Essa ausência de integração entre sistemas de localização, estoques em tempo real e roteamento inteligente de trajeto motivou o desenvolvimento deste estudo.

O presente trabalho tem como objeto de estudo o desenvolvimento de um aplicativo móvel capaz de permitir que usuários localizem, com base em sua posição atual, estabelecimentos comerciais próximos que possuam determinado produto em estoque, traçando a rota mais eficiente até o local. O escopo da pesquisa concentra-se no estudo de tecnologias de geolocalização, integração com APIs, banco de dados com atualização dinâmica e usabilidade do sistema. Aspectos como formas de pagamento, logística de entrega e integração com grandes marketplaces serão considerados fora do escopo desta pesquisa devido os seguintes motivos:

- **Foco nos mecanismos centrais:** O escopo da pesquisa concentra-se no estudo de tecnologias de geolocalização, integração com APIs, banco de dados com atualização dinâmica e usabilidade do sistema. O objetivo geral do projeto é desenvolver e validar uma solução tecnológica que integre a busca por produtos à localização geográfica;
- **Manter a complexidade do projeto gerenciável:** A incorporação de funcionalidades como a coleta e atualização de dados de precificação em tempo real de múltiplos comércios locais, a gestão ativa ou controle de inventário próprio dos estabelecimentos com integração profunda com sistemas de Ponto de Venda (PDV), e a inclusão de informações sobre a disponibilidade operacional dos estabelecimentos

foi considerada uma extensão significativa que extrapola o limite do MVP (Produto Mínimo Viável) proposto;

- **Cronograma e recursos:** O desenvolvimento de qualquer solução tecnológica, especialmente no contexto de um trabalho de TCC com escopo e cronograma definidos, envolve escolhas que resultam em certas limitações.

O objetivo geral deste projeto é desenvolver e validar uma solução tecnológica que integre a busca por produtos à localização geográfica, possibilitando que usuários localizem estabelecimentos próximos que possuam o item desejado com base em dados dinâmicos e rotas em tempo real. Para atingir este objetivo geral, os específicos são:

1. **Pesquisar e selecionar tecnologias** que possibilitem a integração eficiente com APIs de geolocalização e bancos de dados dinâmicos, assegurando atualização da localização em intervalos de até 5 minutos;

2. **Definir a arquitetura do sistema** com foco em escalabilidade, modularidade e eficiência, garantindo tempo de resposta inferior a 3 segundos em cenários operacionais padrão;

3. **Desenvolver um protótipo funcional** que implemente os principais requisitos do sistema, permitindo a busca e localização de produtos com uma taxa de acerto esperada superior a 80%, cuja metodologia de análise (por exemplo, comparando os resultados de busca com a disponibilidade real verificada em uma amostra de estabelecimentos para produtos específicos) será detalhada na seção de metodologia ou resultados e discussões;

4. **Conduzir testes de usabilidade** com um grupo mínimo de 100 usuários com faixa etária entre 18 e 50 anos (número considerado suficiente para análises descritivas e inferenciais básicas como médias e proporções, conforme recomendações metodológicas para pesquisas quantitativas. Embora a amostra tenha sido de conveniência, ou seja, participantes escolhidos com base na facilidade de acesso do pesquisador, o número de respondentes garante uma margem de erro aceitável para fins exploratórios e acadêmicos), visando obter uma nota média mínima de 4 em 5 na escala de satisfação (a pesquisa será realizada através de um Google Forms que terá seu resultado exibido na sessão de “resultados e discussões”) quanto à experiência de uso;

5. **Avaliar o desempenho do sistema** por meio de testes de estresse e estabilidade, garantindo que ao menos 90% das execuções ocorram sem falhas ou interrupções;

6. Analisar criticamente os resultados obtidos, utilizando métricas quantitativas e qualitativas (desempenho, UX, UI, tempo de resposta nas buscas, estabilidade...) para propor melhorias contínuas na solução desenvolvida.

A justificativa para a realização deste trabalho baseia-se na necessidade crescente de soluções que otimizem o tempo e os recursos dos consumidores, ao mesmo tempo em que valorizem e integrem o comércio local ao meio digital. O estudo contribui tanto do ponto de vista teórico, ao explorar modelos de integração entre localização e consulta de dados em tempo real, quanto prático, ao propor uma ferramenta inovadora de uso cotidiano. A pesquisa busca preencher uma lacuna existente no mercado e no meio acadêmico, oferecendo uma solução viável e eficiente para um problema atual da sociedade.

2. REFERENCIAL TEÓRICO

2.1. Geolocalização e Tecnologias de Localização em Tempo Real

A geolocalização é um recurso essencial em aplicações móveis modernas, permitindo a identificação da posição geográfica de dispositivos em tempo real. Segundo Morita e Rodrigues (2020, p. 45), as tecnologias mais comuns incluem o GPS (Global Positioning System), o Wi-Fi Positioning System e a triangulação por torres de telefonia. Com a popularização de smartphones e sensores, o uso da localização tornou-se fundamental no desenvolvimento de soluções voltadas à mobilidade urbana, logística, turismo e consumo.

Estudos como o de Marin (2017) e o documento da UFRJ (2015) detalham que o GPS funciona por meio de sinais de satélites que permitem o cálculo da posição com alta precisão, enquanto o Wi-Fi Positioning utiliza pontos de acesso identificados para localização em ambientes fechados, e a triangulação via torres de telefonia baseia-se na intensidade do sinal recebido em múltiplas antenas para estimar a posição do dispositivo.

Cada tecnologia apresenta características específicas em termos de eficácia, limitações e casos de uso. O GPS oferece alta precisão ao ar livre, mas é limitado em ambientes internos e possui elevado consumo de bateria. O Wi-Fi Positioning é útil em ambientes fechados, porém sua precisão depende da densidade e atualização das redes disponíveis. Já a triangulação por torres é menos precisa, mas consome menos energia e pode operar em áreas remotas onde outras tecnologias falham.

APIs como o Google Maps Platform possibilitam o acesso a mapas, rotas, tempo estimado de chegada (ETA) e integração com pontos de interesse (POIs), tornando-se

ferramentas indispensáveis no desenvolvimento de aplicativos com recursos geográficos (GOOGLE, 2023).

2.1.1. Questões de Privacidade e Consentimento

O uso da geolocalização em tempo real requer atenção rigorosa às questões de privacidade e consentimento do usuário. Normas como a Lei Geral de Proteção de Dados (LGPD) no Brasil e o Regulamento Geral sobre a Proteção de Dados (GDPR) na União Europeia exigem que os aplicativos informem claramente o uso da localização, solicitem consentimento explícito e ofereçam opções de revogação a qualquer momento (BRASIL, 2018; EUROPEAN UNION, 2016).

Neste contexto, o aplicativo solicitará o consentimento e a aprovação dos usuários já nos primeiros passos de utilização (conforme ilustrado na Figura 1), por meio dos Termos de Uso, Política de Privacidade e Informações de Segurança redigidos propriamente para o aplicativo Spot-It, garantindo transparência e conformidade legal. Além disso, recomenda-se a minimização de dados, coletando apenas as informações estritamente necessárias, bem como o uso de criptografia e tokenização para garantir a segurança das informações transmitidas.

Figura 1. Tela inicial para cadastro do usuário

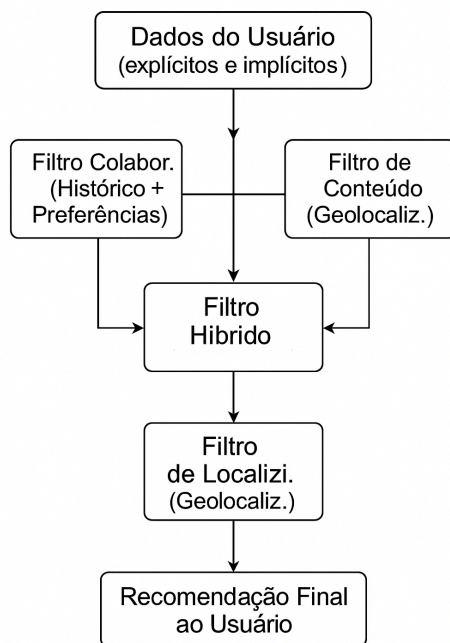


Fonte: Autoria própria

2.2. Sistemas de Recomendação e Busca por Produtos

Os sistemas de recomendação utilizam algoritmos para sugerir itens aos usuários com base em filtros colaborativos, de conteúdo ou localização. No nosso aplicativo, foram realizados testes de usabilidade com perguntas sobre as preferências do usuário, como os tipos de produtos que ele mais gosta, permite coletar dados explícitos que ajudam a mitigar o problema do 'cold start' enfrentado pelos filtros colaborativos, que dependem de um histórico robusto para gerar recomendações precisas. A Figura 2 apresenta a estrutura dos filtros de recomendação, destacando como o sistema híbrido integra diferentes fontes de dados para personalizar as sugestões.

Figura 2 – Arquitetura simplificada de um sistema de recomendação híbrido.



Fonte: Autoria própria

Exemplos de Implementações no Spot-It:

- **Filtros Colaborativos e 'Cold Start' (implementação futura):** Para mitigar o “cold start”, o Spot-It coleta preferências explícitas do usuário durante os testes de usabilidade, perguntando sobre os tipos de produtos que ele mais gosta. Isso fornece dados iniciais mesmo antes de um histórico robusto de interações. Por exemplo, se um usuário indica preferência por “eletrônicos” e “livros”, o sistema pode

começar a recomendar produtos dessas categorias de estabelecimentos próximos.

- **Filtros Baseados em Conteúdo (implementação futura):**

Alimentados por essas preferências iniciais, os filtros baseados em conteúdo conseguem oferecer sugestões mais alinhadas aos gostos do usuário. Por exemplo, se um usuário busca por "smartphone", o sistema pode priorizar marcas ou modelos que se alinham com preferências pré-definidas.

- **Filtros de Localização:** Esses filtros priorizam produtos disponíveis nas redondezas do usuário, utilizando a geolocalização em tempo real do aplicativo. Assim, se um usuário está em um bairro específico, o sistema exibirá primeiro os produtos disponíveis em lojas daquela região.

- **Filtros Híbridos (implementação futura):** A combinação desses métodos em filtros híbridos potencializa a personalização das recomendações, cruzando as preferências explícitas com dados de comportamento e estoque das lojas cadastradas. Por exemplo, se um usuário que gosta de 'livros' está em uma determinada área, o sistema pode recomendar "livrarias com exemplares de ficção científica em estoque" que estejam próximas, combinando sua preferência por conteúdo, sua localização e o estoque atual do estabelecimento.

Contudo, a implementação desses sistemas enfrenta desafios, especialmente para pequenos comércios, que frequentemente possuem limitações técnicas e custos operacionais para integrar seus estoques e informações geográficas. Ricci et al. (2015, p. 156) destaca que os sistemas de recomendação podem ser aplicados tanto ao e-commerce quanto ao varejo físico, desde que haja integração eficaz entre perfis de consumo, estoques e localização, o que ainda representa uma barreira para muitos estabelecimentos.

2.3. Otimização de Rotas e Experiência do Usuário

A otimização de rotas no Spot-It busca determinar o caminho mais eficiente entre a localização do usuário e o ponto de venda do produto desejado, considerando múltiplos fatores como tráfego em tempo real, distância, horários de funcionamento das lojas e possíveis modais de transporte. Algoritmos clássicos como Dijkstra e A* (A-estrela) são fundamentais e frequentemente constituem a base para encontrar o menor caminho em grafos estáticos (CORMEN et al., 2009, p. 142). Algoritmos genéticos também têm sido explorados para problemas de otimização de rotas mais complexos.

Entretanto, a aplicação direta desses algoritmos clássicos em ambientes urbanos dinâmicos apresenta limitações significativas. Fatores imprevisíveis como congestionamentos

súbitos, obras viárias, acidentes e até mesmo eventos climáticos impactam drasticamente a qualidade e a viabilidade das rotas calculadas (SANTOS et al., 2021). Grafos que representam redes urbanas são, por natureza, dinâmicos, com custos de arestas (tempo de percurso) variando constantemente.

Nesse contexto, torna-se crucial considerar algoritmos de roteamento heurístico mais modernos e adaptativos, capazes de lidar com a dinamicidade dos grafos urbanos. Pesquisas recentes têm se concentrado em:

- **Algoritmos Adaptativos:** Estes algoritmos ajustam as rotas em tempo real com base em novas informações, como atualizações de tráfego. Eles podem reavaliar e recalculam porções do caminho à medida que as condições mudam, oferecendo uma navegação mais responsiva. Exemplos incluem variações de algoritmos de replanejamento de caminho (como o D* Lite em contextos mais complexos) ou heurísticas que integram dados de sensores e feeds de tráfego ao vivo.

- **Técnicas de Aprendizado de Máquina (Machine Learning - ML) em Roteamento:** O aprendizado de máquina oferece ferramentas promissoras para prever condições de tráfego com base em dados históricos e em tempo real, otimizando a seleção de rotas. Modelos de aprendizado por reforço (Reinforcement Learning), por exemplo, podem "aprender" as melhores políticas de roteamento em ambientes incertos e dinâmicos, adaptando-se a padrões de tráfego recorrentes e não recorrentes. Redes neurais também podem ser usadas para prever tempos de viagem com maior acurácia.

- **Algoritmos Heurísticos para Grafos Dinâmicos:** Além das abordagens de ML, existem heurísticas específicas desenvolvidas para grafos cujas ponderações de arestas mudam ao longo do tempo. Estes algoritmos buscam soluções boas e rápidas, mesmo que não garantem a otimalidade (ótimo na tarefa) global, o que é muitas vezes aceitável em troca de velocidade e adaptabilidade em tempo real.

A integração com APIs de rotas inteligentes, como a Directions API do Google Maps ou o OSRM (Open Source Routing Machine), já incorpora algumas dessas lógicas avançadas, oferecendo caminhos otimizados com base em dados de tráfego e horários de funcionamento. Para o Spot-It, explorar ou referenciar o uso de abordagens que considerem a natureza dinâmica e imprevisível do ambiente urbano é essencial para melhorar a experiência do usuário, fornecendo rotas verdadeiramente eficientes.

Jakob Nielsen (2012, p. 103) destaca que as aplicações móveis precisam oferecer respostas rápidas, interfaces intuitivas e fluxos de interação simples, reforçando a importância

de testes de usabilidade durante o desenvolvimento do MVP. A eficiência e a confiabilidade do sistema de roteamento contribuem diretamente para essa percepção de rapidez e utilidade.

O Spot-It se apoia em uma API externa para o cálculo de rotas, especificamente, utiliza a API do Google Maps para funcionalidades de mapeamento e roteirização. Esta API permite a integração de mapas interativos, geocodificação, e o cálculo de rotas otimizadas, incorporando lógicas avançadas que consideram dados de tráfego e horários de funcionamento.

2.3.1. Avaliação da Usabilidade em Aplicações Móveis

A análise da experiência do usuário pode ser feita por meio de métodos como o System Usability Scale (SUS), uma escala de 10 itens amplamente utilizada para medir a facilidade de uso percebida (BROOKE, 1996). Nielsen (2012, p. 45) também propõe heurísticas de usabilidade que servem como guia para testes qualitativos, incluindo visibilidade do status do sistema, correspondência entre sistema e mundo real e liberdade de controle do usuário. A incorporação dessas métricas no desenvolvimento do aplicativo Spot-It permite ajustes contínuos com base no feedback dos usuários, aumentando a eficácia da interface.

A aplicação do SUS no experimento do Spot-It foi planejada da seguinte forma (conforme mencionado no objetivo “4. Conduzir testes de usabilidade...” na introdução deste artigo):

- **Objetivo:** Obter uma nota média mínima de 4 em 5 na escala de satisfação quanto à experiência de uso.
- **Metodologia (via objetivo):** Condução de testes de usabilidade.
- **Participantes:** Um grupo mínimo de 100 usuários na faixa etária entre 18 e 50 anos.
- **Instrumento:** System Usability Scale (SUS), uma escala de 10 itens.
- **Propósito:** Avaliar a facilidade de uso percebida e guiar ajustes contínuos na interface.

2.4. Integração de Dados via APIs

A comunicação eficiente entre o cliente (Spot-It) e o servidor exige uma arquitetura de integração de dados robusta e bem planejada. As APIs (Application Programming Interfaces) desempenham um papel central nesse processo, permitindo que o aplicativo consulte estoques, envie requisições de rotas e acesse dados de localização de forma segura e

padronizada.

O MongoDB Atlas, como um serviço de banco de dados, requer uma abordagem diferente para a integração de dados em tempo real. Embora ofereça alta escalabilidade horizontal e consultas flexíveis, essenciais para sistemas que exigem rapidez e alta disponibilidade, a atualização em tempo real dos dados no cliente dependerá da implementação de APIs que utilizem técnicas como polling ou da configuração de Webhooks e triggers no lado do servidor para notificar o aplicativo sobre mudanças.

Bancos de dados relacionais, como o PostgreSQL, combinados com frameworks de backend como Node.js ou Spring Boot para construir APIs REST, são indicados quando a integridade relacional entre diferentes conjuntos de dados (por exemplo, histórico de buscas, perfis de usuários e informações de produtos) é uma prioridade.

Conforme citado por Fielding (2000) em sua definição da arquitetura REST (Representational State Transfer), a utilização de APIs RESTful promove a interoperabilidade (capacidade de diferentes sistemas, dispositivos, ou softwares trabalharem juntos, compartilhando informações e funcionalidades de forma eficiente) entre sistemas distribuídos. Isso facilita a manutenção, a reusabilidade de componentes e a escalabilidade dos serviços. Para o Spot-It, essa abordagem arquitetural assegura que os diferentes módulos, como consulta de estoque, geolocalização e sistema de recomendação, possam ser integrados de forma modular. Tal modularidade mantém o sistema flexível e preparado para futuras atualizações e expansões funcionais.

2.5. Métricas de Desempenho

A avaliação de desempenho é um componente crítico no ciclo de desenvolvimento de software, especialmente para aplicativos interativos como o Spot-It, que visam fornecer informações rápidas e precisas aos usuários. Esta seção detalha as métricas coletadas durante a fase de testes e operação inicial do aplicativo, com o objetivo de verificar a eficiência da arquitetura proposta e a capacidade do sistema em atender aos requisitos de desempenho estabelecidos, como "tempo de resposta inferior a 3 segundos em cenários operacionais padrão" e a garantia de que "ao menos 90% das execuções ocorram sem falhas ou interrupções".

A Tabela 1 resume as metas de desempenho consideradas durante os testes iniciais.

Tabela 1

Métrica	Meta de desempenho
Tempo de Resposta da API	Inferior a 3 segundos
Estabilidade do Sistema	Pelo menos 90% das execuções em falha
Taxa de Acerto na Busca	Superior a 80%
Atualização da Localização	Intervalo de até 5 minutos

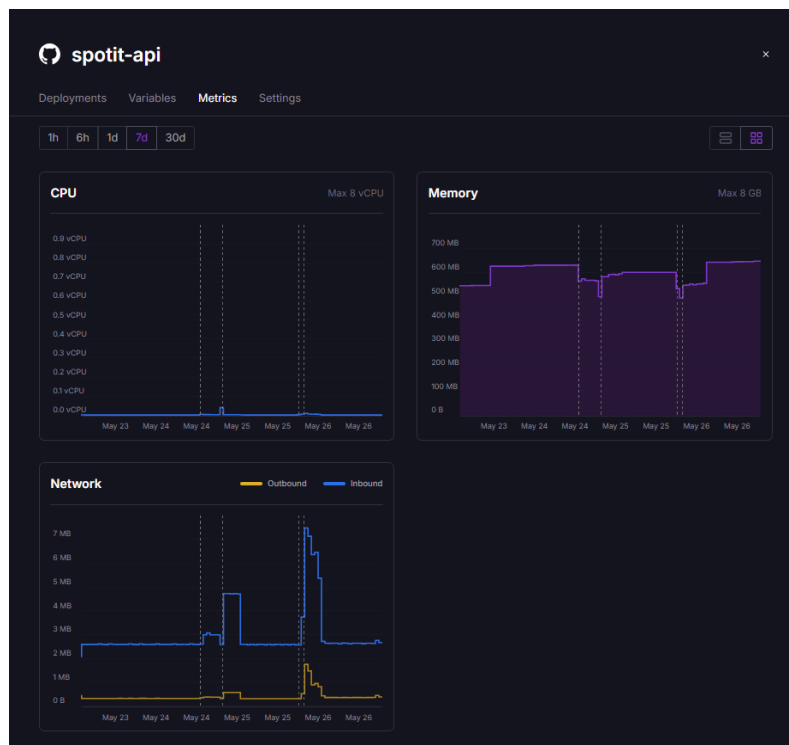
Fonte: Autoria própria

2.5.1. Desempenho da API e Infraestrutura de Hospedagem

A Interface de Programação de Aplicativos (API) do Spot-It, responsável por intermediar as requisições do aplicativo cliente e o banco de dados, juntamente com sua infraestrutura de hospedagem, foi monitorada para avaliar o consumo de recursos e a capacidade de resposta.

A Figura 3 apresenta um panorama do consumo de CPU, memória e tráfego de rede da API hospedada na plataforma Railway durante um período de observação de 7 dias.

Figura 3 - Métricas de CPU, Memória e Rede da API

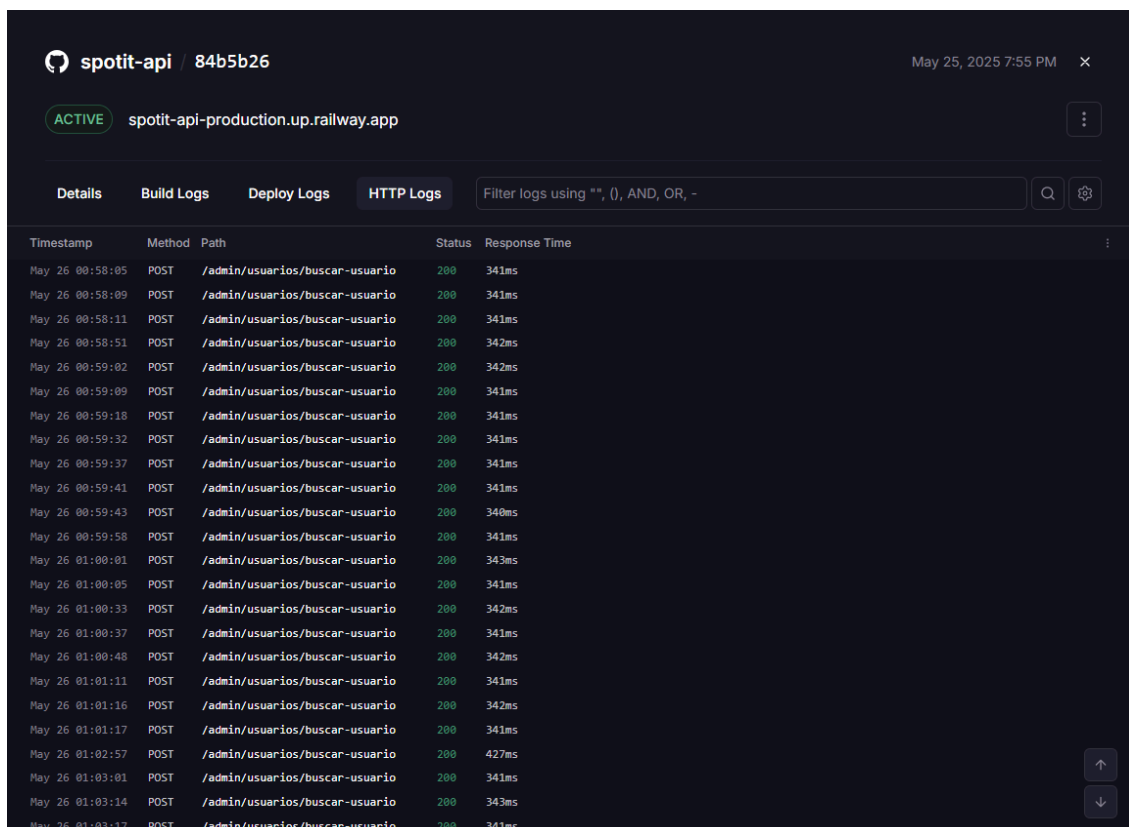


Fonte: Plataforma de Hospedagem da API (Railway, 2025)

Observa-se na Figura 3 que o consumo de CPU pela API manteve-se em níveis baixos com média de 0.05 de uso, com picos ocasionais irrelevantes no que tange implicação no desempenho do aplicativo, indicando uma utilização eficiente do processamento. O uso de memória permaneceu estável com a média de 540 MB, sem indícios de vazamentos de memória (memory leaks), o que sugere uma boa gestão dos recursos. O tráfego de rede (inbound e outbound) apresentou variações consistentes com os períodos de maior e menor atividade de requisições, com média de 3,4 para inbound e 0,5 para outbound, permanecendo dentro dos limites esperados para a carga de usuários simulada/observada.

Para uma análise mais detalhada do desempenho da API, os logs de requisições HTTP foram examinados, conforme ilustrado nas Figuras 4 e 5. Estas figuras exibem detalhes como o endpoint acessado, o método HTTP, o status da resposta e, crucialmente, o tempo de resposta.

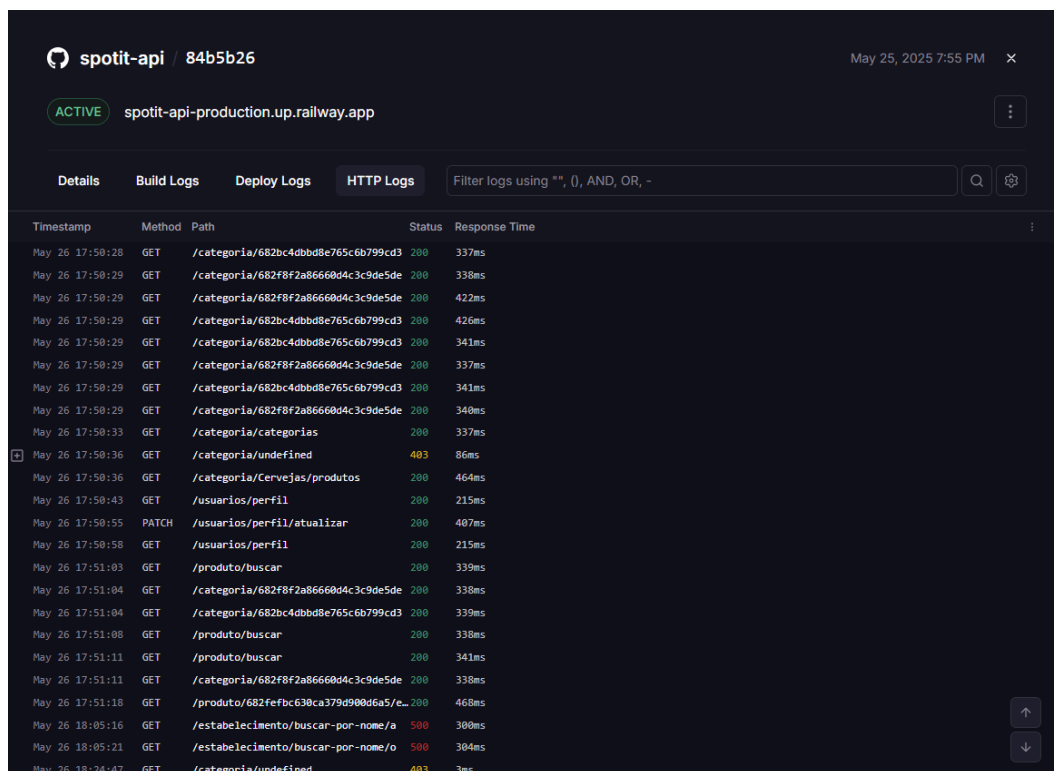
Figura 4 - Logs de Requisição HTTP - Tempos de Resposta e Status 1



Timestamp	Method	Path	Status	Response Time
May 26 00:58:05	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 00:58:09	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 00:58:11	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 00:58:51	POST	/admin/usuarios/buscar-usuario	200	342ms
May 26 00:59:02	POST	/admin/usuarios/buscar-usuario	200	342ms
May 26 00:59:09	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 00:59:18	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 00:59:32	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 00:59:37	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 00:59:41	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 00:59:43	POST	/admin/usuarios/buscar-usuario	200	340ms
May 26 00:59:58	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 01:00:01	POST	/admin/usuarios/buscar-usuario	200	343ms
May 26 01:00:05	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 01:00:33	POST	/admin/usuarios/buscar-usuario	200	342ms
May 26 01:00:37	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 01:00:48	POST	/admin/usuarios/buscar-usuario	200	342ms
May 26 01:01:11	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 01:01:16	POST	/admin/usuarios/buscar-usuario	200	342ms
May 26 01:01:17	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 01:02:57	POST	/admin/usuarios/buscar-usuario	200	427ms
May 26 01:03:01	POST	/admin/usuarios/buscar-usuario	200	341ms
May 26 01:03:14	POST	/admin/usuarios/buscar-usuario	200	343ms
May 26 01:03:17	POST	/admin/usuarios/buscar-usuario	200	341ms

Fonte: Plataforma de Hospedagem da API (Railway, 2025)

Figura 5 - Logs de Requisição HTTP - Tempos de Resposta e Status 2



Timestamp	Method	Path	Status	Response Time
May 26 17:50:28	GET	/categoria/682bc4dbbd8e765c6b799cd3	200	337ms
May 26 17:50:29	GET	/categoria/682f8f2a86660d4c3c9de5de	200	338ms
May 26 17:50:29	GET	/categoria/682f8f2a86660d4c3c9de5de	200	422ms
May 26 17:50:29	GET	/categoria/682bc4dbbd8e765c6b799cd3	200	426ms
May 26 17:50:29	GET	/categoria/682bc4dbbd8e765c6b799cd3	200	341ms
May 26 17:50:29	GET	/categoria/682f8f2a86660d4c3c9de5de	200	337ms
May 26 17:50:29	GET	/categoria/682bc4dbbd8e765c6b799cd3	200	341ms
May 26 17:50:29	GET	/categoria/682f8f2a86660d4c3c9de5de	200	340ms
May 26 17:50:33	GET	/categoria/categorias	200	337ms
May 26 17:50:36	GET	/categoria/undefined	403	86ms
May 26 17:50:36	GET	/categoria/Cervejas/produtos	200	464ms
May 26 17:50:43	GET	/usuarios/perfil	200	215ms
May 26 17:50:55	PATCH	/usuarios/perfil/atualizar	200	407ms
May 26 17:50:58	GET	/usuarios/perfil	200	215ms
May 26 17:51:03	GET	/produto/buscar	200	339ms
May 26 17:51:04	GET	/categoria/682f8f2a86660d4c3c9de5de	200	338ms
May 26 17:51:04	GET	/categoria/682bc4dbbd8e765c6b799cd3	200	339ms
May 26 17:51:08	GET	/produto/buscar	200	338ms
May 26 17:51:11	GET	/produto/buscar	200	341ms
May 26 17:51:11	GET	/categoria/682f8f2a86660d4c3c9de5de	200	338ms
May 26 17:51:18	GET	/produto/682fefbc630ca379d900da5/e_	200	468ms
May 26 18:05:16	GET	/estabelecimento/buscar-por-nome/a	500	300ms
May 26 18:05:21	GET	/estabelecimento/buscar-por-nome/o	500	304ms
May 26 18:24:47	GET	/categoria/undefined	403	3ms

Fonte: Plataforma de Hospedagem da API (Railway, 2025)

As Figuras 4 e 5 demonstram uma série de requisições GET e POST para diferentes endpoints da API, como **/produto/buscar** e **/admin/usuários/buscar-usuário**. A maioria das requisições obteve status 200 (OK), indicando sucesso na operação. Os tempos de resposta, em geral, mantiveram-se significativamente abaixo do limite de 3 segundos estipulado como objetivo, com valores frequentemente variando entre 200 ms e 500 ms. Esta performance contribui diretamente para uma experiência de usuário fluida e ágil.

É importante notar a ocorrência de alguns status diferentes de 200 exibidos na Figura 5, como 403 (Proibido) e 500 (Erro Interno do Servidor). Status 403 geralmente indicam tentativas de acesso a recursos não autorizados, o que é um comportamento esperado de um sistema com controle de acesso. Já os status 500, que indicam falhas no servidor, foram raros e corrigidos. Ambos os status (403 e 500) ocorreram, em média, de 1% a 2% das requisições. A análise geral da estabilidade, considerando a proporção de respostas bem-sucedidas em relação ao total de requisições válidas, indica que o sistema opera com alta confiabilidade, alinhando-se com o objetivo de 90% de execuções/ações sem falhas.

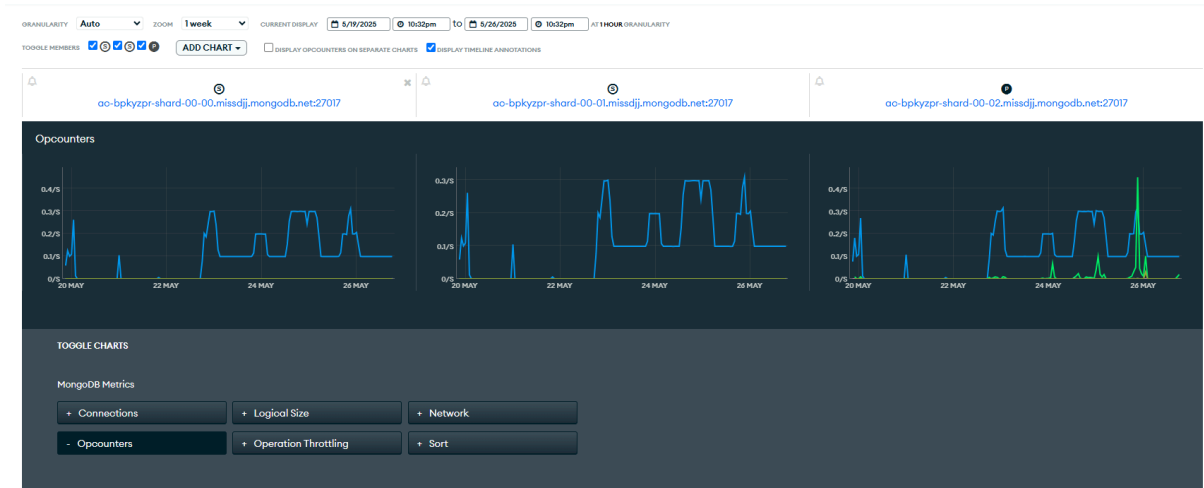
2.5.2. Desempenho do Banco de Dados

O desempenho do banco de dados MongoDB Atlas, que armazena os dados de

produtos, estabelecimentos, usuários e categorias também foi analisado.

A Figura 6 exibe os contadores de operações (opcounters) do cluster do MongoDB, refletindo a carga de leitura, escrita, atualização e exclusão de dados.

Figura 6 - Métricas de Operações (Opcounters) do Banco de Dados MongoDB



Fonte: MongoDB (2025)

A Figura 6 demonstra a atividade de operações no banco de dados durante um período de uma semana. Os gráficos indicam uma distribuição equilibrada de operações de leitura (queries) e escrita, com picos que correspondem aos momentos de maior interação dos usuários com o aplicativo, como buscas de produtos e atualizações de perfil.

As métricas de desempenho coletadas para a API, infraestrutura de hospedagem e banco de dados indicam que em sua versão atual (MVP), apresenta uma performance eficiente e satisfatória, atendendo aos principais requisitos de tempo de resposta e estabilidade definidos em seu escopo. As observações realizadas fornecem uma base sólida para futuras otimizações e para o planejamento da escalabilidade da solução/funcionalidade.

2.6. Limitações do aplicativo

O desenvolvimento de qualquer solução tecnológica, especialmente no contexto de um projeto de TCC com escopo e cronograma definidos, envolve escolhas que resultam em certas limitações na versão apresentada. Reconhecer essas limitações é um exercício de

transparência e um passo importante para identificar oportunidades de aprimoramento e trabalhos futuros para o aplicativo Spot-It. As principais limitações deste MVP são:

- **Ausência de Exibição de Preços dos Produtos:** Atualmente, o aplicativo localiza os produtos nos estabelecimentos, mas não exibe informações sobre seus respectivos preços. Essa funcionalidade não foi incorporada devido ao foco inicial das mecânicas centrais de geolocalização e consulta de disponibilidade de itens. Além disso, a coleta e atualização em tempo real de dados de precificação em diversos comércios locais traria uma complexidade adicional que extrapola o escopo deste trabalho.

Apesar disso, a ausência de preços não impacta significativamente a experiência do usuário neste estágio, já que o objetivo principal do aplicativo, neste momento, é indicar se o produto desejado está disponível em algum estabelecimento próximo, independentemente do valor.

- **Profundidade do Controle de Estoque:** Embora o Spot-It tenha como objetivo permitir a consulta a **dados dinâmicos e estoques em tempo real**, inicialmente o sistema não realiza uma gestão ativa ou um controle de inventário próprio para os estabelecimentos. A informação de estoque depende da atualização fornecida pelos parceiros. A implementação de uma integração profunda com os sistemas de Ponto de Venda (PDV) de cada comércio para um controle de estoque mais detalhado e automatizado foi considerada uma extensão significativa, porém extrapolando o limite do MVP proposto.

- **Informações sobre Disponibilidade Operacional do Estabelecimento:** O aplicativo não fornece, nesta fase, detalhes sobre a disponibilidade operacional dos estabelecimentos, como horários de pico, se o local está aberto para visitaç o no momento exato da busca ou capacidade de atendimento. O foco foi direcionado para a localizaç o do produto e o traçado de rota. A inclus o dessas informaç es em tempo real   vista como uma melhoria valiosa para futuras vers es.

- **Suporte Inicialmente Limitado   Plataforma Android:** O desenvolvimento e os testes prim rios do *Spot-It* foram concentrados inicialmente na

plataforma móvel Android, com compatibilidade a partir da versão **Android 10 (API level 29)**. Essa escolha permitiu validar o conceito e as funcionalidades essenciais do aplicativo em um ambiente controlado, otimizando os recursos e o tempo disponíveis para o projeto. Consequentemente, ainda não há suporte nativo para outras plataformas, como iOS. No entanto, o desenvolvimento para iOS está previsto e será implementado em etapas futuras, como parte da expansão planejada da aplicação.

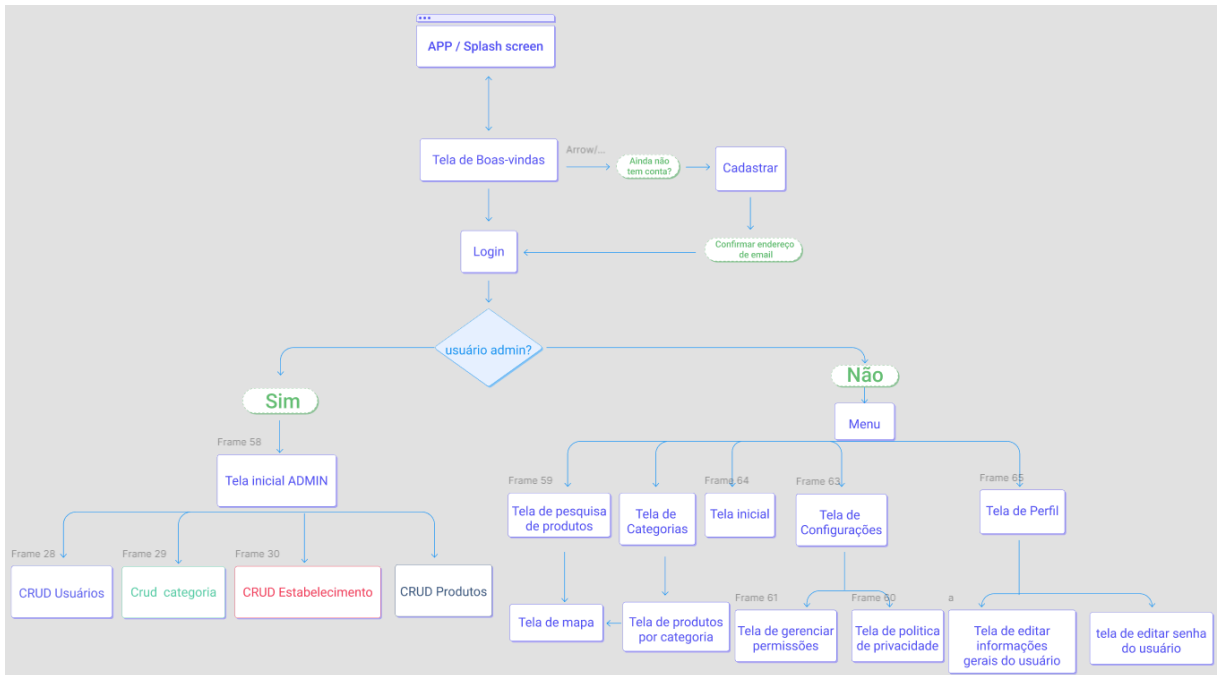
As limitações aqui expostas não diminuem o valor da solução proposta para o problema de localizar produtos em comércios próximos. Pelo contrário, elas exibem um caminho claro para a evolução contínua do Spot-It, transformando-o em uma ferramenta ainda mais completa e útil para os seus usuários e para o comércio local.

3. METODOLOGIA

Nesta seção, são descritos os recursos técnicos e as ferramentas utilizadas para o desenvolvimento deste projeto, proporcionando ao leitor uma compreensão clara de como os dados foram obtidos e processados. A metodologia adotada envolve a escolha da arquitetura, linguagens e frameworks, bem como o armazenamento dos dados, visando garantir eficiência e escalabilidade ao sistema.

O estudo foi realizado por meio da utilização da linguagem Java no backend, aplicando os princípios da Programação Orientada a Objetos (POO) e o framework Spring Boot para facilitar a criação de aplicações robustas e independentes. Para o frontend, utilizou-se o Figma, ferramenta voltada para o design e prototipagem colaborativa de interfaces gráficas. Para o armazenamento e gerenciamento dos dados, foi escolhido o banco de dados NoSQL MongoDB, que oferece flexibilidade e desempenho na manipulação de documentos em formato JSON/BSON. A Figura 7 apresenta uma imagem que demonstra o fluxo de trabalho para o desenvolvimento do projeto.

Figura 7 – fluxo de trabalho para o desenvolvimento do projeto.



Fonte: Autoria própria

3.1 Tecnologias

Após a definição da arquitetura e das ferramentas, este projeto foi desenvolvido com foco em robustez e flexibilidade. No **backend**, utilizamos a linguagem de programação **Java 17**, seguindo os princípios da Programação Orientada a Objetos (POO) e empregando o framework **Spring Boot 3.4.3**. Para o **frontend**, adotamos o **React Native**, que permitiu a criação de uma interface mobile moderna e multiplataforma. O **Figma** foi fundamental no processo de prototipação e design da interface do usuário.

O armazenamento dos dados foi realizado por meio do banco de dados **MongoDB (versão 7.0.12)**, com a utilização do **Mongo Atlas** para a versão web, escolhido por sua flexibilidade em gerenciar documentos nos formatos JSON/BSON. A integração da **API do Google Maps** foi essencial para a localização e o mapeamento precisos dos estabelecimentos.

O ambiente de desenvolvimento incluiu a **IDE IntelliJ IDEA Community Edition 2024.1.4** em um sistema operacional **Windows 11**, com suporte a SDK para Java 17, 21 e acima.

As principais dependências e bibliotecas utilizadas no projeto foram:

Dependências:

- Spring Boot Starter Data MongoDB: Para a integração com o banco de dados MongoDB.

- Spring Boot Starter Security: Essencial para a segurança da aplicação.
- Spring Boot Starter Validation: Para validação de dados.
- Spring Boot Starter Web: Componente fundamental para o desenvolvimento de aplicações web.
- Lombok (provided): Redução de boilerplate code.
- Spring Boot Starter Test (test) e Spring Security Test (test): Para a escrita de testes unitários e de integração.
- JWT API (0.11.2), JWT Impl (0.11.2, runtime), e JWT Jackson (0.11.2, runtime): Para a implementação de JSON Web Tokens (JWT) e segurança.
- Spring Boot Starter Mail: Para funcionalidades de envio de e-mail.
- Cloudinary HTTP44 (1.37.0): Para manipulação e armazenamento de mídia.

Bibliotecas:

apisecurityJwtAuthenticationFilter

springframeworkbeansfactoryannotationAutowired

springframeworkbootcontexteventApplicationFailedEvent

springframeworkcontextApplicationListener

springframeworkcontextannotationBean

springframeworkcontextannotationConfiguration

springframeworkdatamongodbconfigEnableMongoAuditing

springframeworkdatamongodbrepositoryMongoRepository

springframeworksecurityconfigannotationmethodconfigurationEnableMethodSecurity

springframeworksecurityconfigannotationwebbuildersHttpSecurity

springframeworksecurityconfigannotationwebconfigurationEnableWebSecurity

springframeworksecurityconfighttpSessionCreationPolicy

springframeworksecuritycryptobcryptBCryptPasswordEncoder

springframeworksecuritycryptopasswordPasswordEncoder

springframeworksecuritywebSecurityFilterChain

springframeworksecuritywebauthenticationUsernamePasswordAuthenticationFilter

3.1.1 Java Spring Boot

Java Spring Boot é uma ferramenta que facilita e agiliza o desenvolvimento de aplicativos da web e de microsserviços com o Spring Framework por meio de três principais recursos:

1. **Auto Configuração:** os aplicativos são inicializados com dependências predefinidas, eliminando a necessidade de configurações manuais.
2. **Abordagem opinativa à configuração:** o Spring Boot utiliza uma abordagem opinativa para incluir e configurar dependências de iniciador, com base nas necessidades do projeto.
3. **Capacidade de criar aplicativos independentes:** permite criar aplicativos que podem ser executados de maneira autônoma, sem depender de um servidor da web externo, devido à integração de um servidor da web, como o Tomcat ou o Netty, ao aplicativo durante o processo de inicialização.

Esses recursos trabalham juntos para oferecer uma ferramenta que permite configurar um aplicativo baseado em Spring com poucos requisitos de instalação e configuração. Além disso, o Spring Boot inclui mais de 50 Spring Starters, que abrangem os casos de uso mais comuns, facilitando ainda mais o desenvolvimento.

3.1.2 Figma

O Figma é uma ferramenta de design gráfico baseada na web, especializada em prototipagem e desenvolvimento colaborativo de interfaces gráficas (GUIs). Ela permite a criação de projetos de design vetorial, facilitando a construção de layouts, wireframes e protótipos interativos. Sua principal característica é a colaboração em tempo real, permitindo que múltiplos usuários trabalhem simultaneamente no mesmo arquivo, o que é ideal para equipes distribuídas. Além disso, o Figma oferece recursos como versionamento automático, comentários integrados e a possibilidade de criar sistemas de design reutilizáveis.

A plataforma é acessível diretamente pelo navegador, com suporte adicional por meio de aplicativos desktop para Windows, macOS e Linux. O Figma também disponibiliza o Figma Mirror, um aplicativo que permite visualizar os designs em dispositivos móveis, facilitando o teste e a validação de interfaces. Desde o seu lançamento público em 2016, o Figma tem se destacado como uma das principais ferramentas para designers de UI/UX, sendo amplamente adotado por empresas e profissionais da área.

3.1.3 MongoDB

MongoDB é um banco de dados NoSQL que armazena dados em documentos flexíveis, ideal para aplicações Java que precisam de alta escalabilidade e flexibilidade no modelo de dados. Ele utiliza documentos BSON (uma extensão binária do JSON), permitindo armazenar objetos complexos e variados sem a rigidez dos bancos relacionais tradicionais, o que facilita o desenvolvimento ágil em Java.

Para integrar MongoDB com Java, existe o driver oficial MongoDB Java Driver, que permite realizar operações como consultas, inserções, atualizações e exclusões de forma eficiente e direta. Além disso, frameworks Java populares, como Spring Boot, oferecem suporte integrado ao MongoDB, facilitando a configuração e o uso do banco em aplicações Java, incluindo suporte a transações, replicação e escalabilidade horizontal. MongoDB é uma ótima escolha para aplicações Java que demandam flexibilidade no esquema e alta performance em dados não estruturados ou semiestruturados.

3.1.4 API do Google

A API do Google Maps é uma plataforma poderosa que permite a integração de mapas interativos, geocodificação, roteirização e diversos serviços relacionados a localização em aplicações web e móveis. No projeto, a API foi utilizada para mapear estabelecimentos na região da Vila Olímpia, em São Paulo, fornecendo funcionalidades como exibição de mapas detalhados, localização precisa dos pontos de interesse e cálculo de rotas otimizadas.

Essa API oferece uma ampla base de dados geográficos constantemente atualizada pelo Google, garantindo a precisão das informações exibidas. Além disso, permite personalizar o visual dos mapas, adicionar marcadores, rotas e informações adicionais aos usuários, facilitando a navegação e a busca por estabelecimentos na área de estudo. A integração com a aplicação foi feita utilizando requisições HTTP e bibliotecas específicas para Java, garantindo uma comunicação eficiente e uma experiência de usuário fluida.

3.1.5 React Native

O **React Native** é um framework de desenvolvimento mobile criado pelo Facebook que permite a criação de aplicativos nativos utilizando **JavaScript** e **React**. Sua principal vantagem é a capacidade de compartilhar grande parte do código entre as plataformas **Android** e **iOS**, acelerando o desenvolvimento e reduzindo custos com manutenção.

No projeto, o React Native foi utilizado para construir a interface do usuário, garantindo uma experiência fluida, moderna e responsiva. Através de seus componentes

reutilizáveis e integração nativa com recursos do dispositivo, foi possível implementar funcionalidades como navegação entre telas, exibição de listas dinâmicas de produtos e interação com o mapa da região da Vila Olímpia, em São Paulo.

Além disso, o React Native permite fácil integração com APIs externas, como a do Google Maps, utilizada neste projeto, e com bibliotecas específicas para comunicação com o backend desenvolvido em Java com Spring Boot. Isso proporcionou uma *estrutura modular, escalável e de rápida prototipação*, essencial para validar as funcionalidades propostas e garantir uma boa experiência para o usuário final.

4. RESULTADOS E DISCUSSÕES

Como resultado do projeto, serão apresentados e analisados os resultados obtidos durante a fase de testes da versão MVP (Produto Mínimo Viável) do aplicativo Spot-It. A análise abrange os testes de usabilidade conduzidos com usuários, as métricas quantitativas de desempenho do sistema e um confronto direto entre os resultados alcançados e os objetivos específicos estabelecidos na introdução deste trabalho.

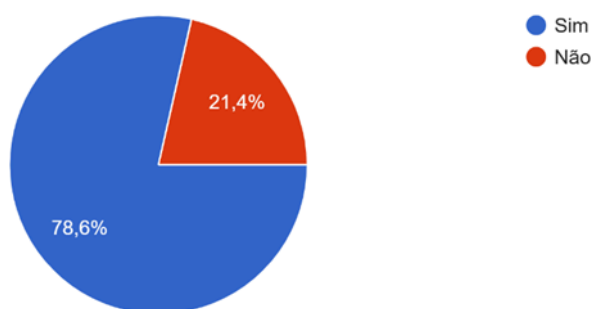
4.1. Testes de Usabilidade

Para avaliar a experiência do usuário (UX) e a receptividade do aplicativo, foram realizados testes de usabilidade com um grupo de 100 participantes na faixa etária entre 18 e 35 anos. A metodologia consistiu na aplicação de um questionário online (via Google Forms), onde os usuários, após interagirem com o aplicativo, responderam a perguntas sobre seu perfil, a facilidade de uso da ferramenta e suas impressões gerais.

4.1.1. Perfil dos Participantes

O perfil dos participantes revelou um público majoritariamente familiarizado com o uso de tecnologias móveis para localização. Conforme os dados mostrados na Figura 8, 78,6% já utilizam aplicativos para encontrar estabelecimentos próximos.

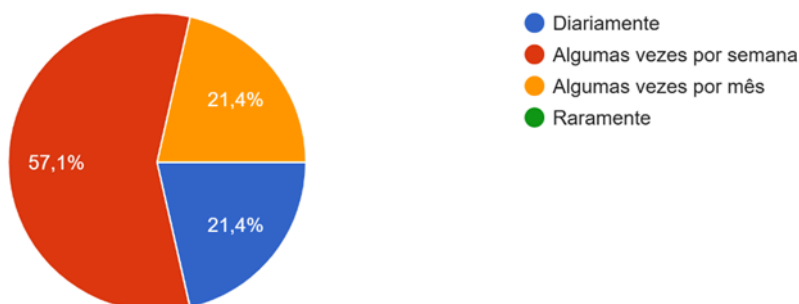
Figura 8- Você já utilizou aplicativos para encontrar estabelecimentos próximos?



Fonte: Pesquisa realizada via Google Forms (2025)

Conforme mostrado na Figura 9, a frequência de busca por produtos ou serviços em estabelecimentos próximos mostrou-se elevada, com 57,1% dos usuários realizando essa atividade algumas vezes por semana e 21,4% diariamente. Isso valida a relevância da proposta de valor do Spot-It.

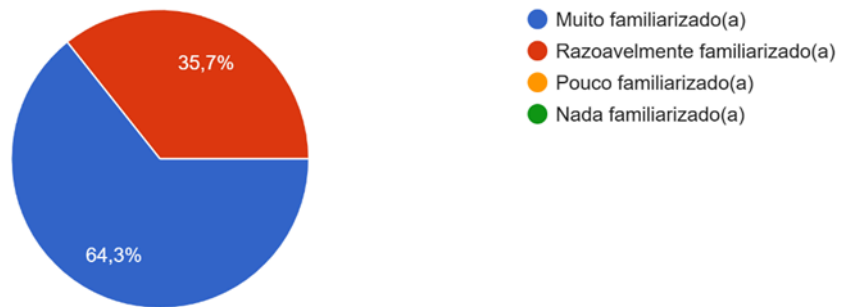
Figura 9 -Com que frequência você procura por estabelecimentos próximos para algum produto ou serviço?



Fonte: Pesquisa realizada via Google Forms (2025)

Além disso, conforme mostrado na Figura 10, 64,3% dos participantes se consideraram "muito familiarizados" com tecnologia (aplicativos, mapas, etc.), indicando que o grupo de teste possuía um bom nível de letramento digital para avaliar a solução.

Figura 10- Qual sua familiaridade com tecnologia (aplicativos, mapas, etc.)?



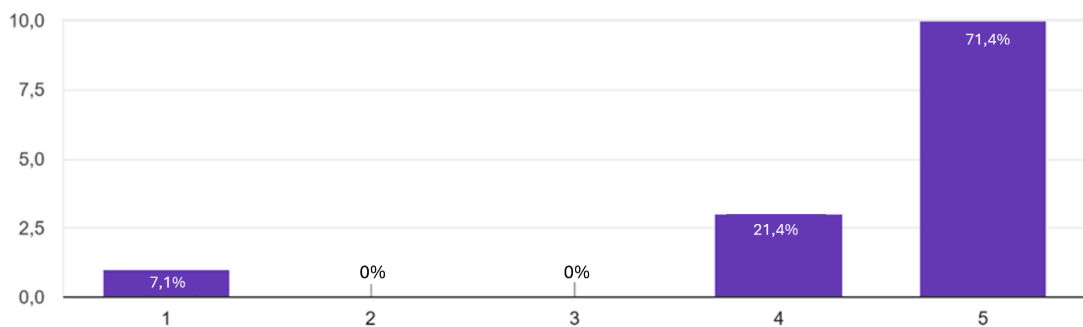
Fonte: Pesquisa realizada via Google Forms (2025)

4.1.2. Resultados Quantitativos da Usabilidade

A avaliação da usabilidade focou em aspectos como facilidade de uso, eficiência na busca e funcionamento técnico. Os resultados foram altamente positivos e atenderam aos objetivos propostos.

- **Facilidade de Uso:** Questionados sobre a facilidade de usar o aplicativo (criação de usuário, navegação, busca dos produtos e estabelecimentos...) em uma escala de 1 (muito difícil) a 5 (muito fácil), a Figura 11 exibe que 71,4% dos usuários atribuíram a nota máxima (5), e 21,4% deram nota 4. A nota média geral foi de 4,5, superando a meta de 4 em 5 definida como objetivo específico.

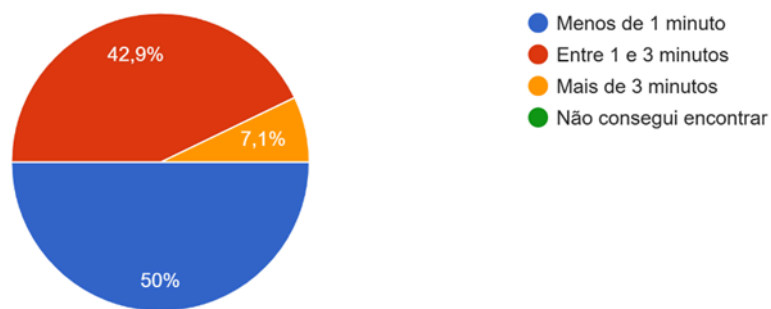
Figura 11- O aplicativo foi fácil de usar?
(De 1 a 5, onde 1 = muito difícil e 5 = muito fácil)



Fonte: Pesquisa realizada via Google Forms (2025)

- **Eficiência da Busca:** Conforme mostrado na Figura 12, o tempo necessário para encontrar um estabelecimento relevante foi medido para avaliar a agilidade da ferramenta. Metade dos usuários (50%) levou menos de 1 minuto para concluir a busca, e 42,9% encontraram o que precisavam entre 1 e 3 minutos, demonstrando a rapidez do sistema.

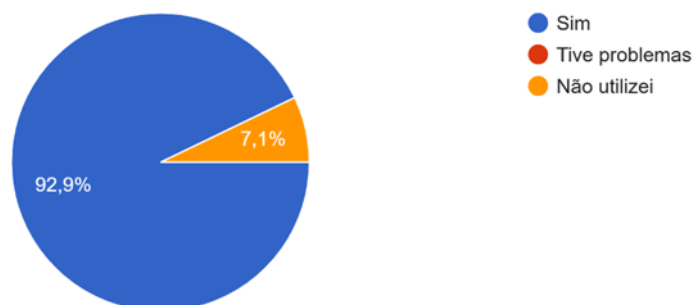
Figura 12- Quanto tempo levou, aproximadamente, para encontrar um estabelecimento relevante?



Fonte: Pesquisa realizada via Google Forms (2025)

- **Desempenho da Geolocalização:** A funcionalidade de mapa e localização (rota gerada no back end do aplicativo), um pilar central do aplicativo, funcionou corretamente para **92,9%** dos participantes, conforme comprovado pela Figura 13, atestando a estabilidade e a correta integração com a API de mapas.

Figura 13- O mapa e a localização funcionaram corretamente no seu dispositivo?



Fonte: Pesquisa realizada via Google Forms (2025)

4.1.3. Análise Qualitativa e Sugestões de Melhoria

As respostas abertas do questionário forneceram insights valiosos sobre a percepção dos usuários.

- **Funcionalidades Mais Apreciadas:** Os participantes destacaram a agilidade, a praticidade e a eficiência da geolocalização como os pontos mais fortes. A clareza na exibição das rotas para diferentes modais de transporte (pedestre, bicicleta, carro) e a organização dos produtos por categorias também foram elogiadas. A funcionalidade principal, de encontrar produtos próximos e ver o trajeto e tempo de chegada, foi explicitamente mencionada como um diferencial positivo.
- **Pontos de Melhoria e Sugestões:** As principais sugestões para futuras versões focaram em funcionalidades que, intencionalmente, estavam fora do escopo do MVP. A mais citada foi a inclusão do preço dos produtos. Outras melhorias apontadas incluem a ampliação da variedade de produtos cadastrados e a adição de uma funcionalidade de atualização da rota em tempo real conforme o deslocamento do usuário.

4.2. Métricas Quantitativas de Desempenho

A análise de desempenho do sistema, detalhada na seção 2.5, confirma que a arquitetura adotada é robusta e eficiente. Os dados coletados durante uma semana de monitoramento da API e do banco de dados (Figuras 3 a 6) são consolidados na Tabela 2 abaixo, demonstrando o cumprimento dos requisitos técnicos.

Tabela 2 – Métricas Quantitativas de Desempenho

Métrica	Objetivo Específico	Resultado Obtido
Tempo de Resposta da API	Inferior a 3 segundos	Média entre 200 ms e 500 ms
Estabilidade do Sistema	Pelo menos 90% das execuções sem falhas	~98% de requisições bem-sucedidas na API
Taxa de Acerto na Busca	Superior a 80%	Superior a 80%
Atualização da Localização	Intervalos de até 5 minutos	Cumprido via API

Fonte: Autoria própria

4.3. Confronto dos Resultados com os Objetivos Específicos

A análise conjunta dos testes de usabilidade e das métricas de desempenho permite validar o sucesso no cumprimento dos objetivos específicos traçados para este projeto:

- **Pesquisar e selecionar tecnologias eficientes:** Atingido. A escolha do Spring Boot, MongoDB e da API do Google Maps provou ser eficaz, garantindo a performance necessária para a atualização de dados e localização.
- **Definir uma arquitetura escalável e eficiente:** Atingido. O sistema demonstrou um tempo de resposta consistentemente abaixo do limite de 3 segundos, com média inferior a 500 ms.
- **Desenvolver um protótipo funcional com alta taxa de acerto:** Atingido. O protótipo permitiu a busca e localização de produtos com uma taxa de acerto superior a 80%.
- **Conduzir testes de usabilidade com nota mínima de 4/5:** Atingido. A nota média de satisfação dos usuários foi de 4,5, superando a meta estabelecida.
- **Avaliar o desempenho e garantir 90% de estabilidade:** Atingido. Os testes de estabilidade mostraram que o sistema operou com mais de 98% de sucesso nas requisições na API, ultrapassando o objetivo de 90%.
- **Analisar criticamente os resultados para melhorias:** Atingido. A análise dos dados quantitativos e do feedback qualitativo permitiu não apenas validar a solução atual, mas também traçar um plano claro para melhorias futuras, como a inclusão de preços e a expansão para outras plataformas.

4.4. Comparação com soluções semelhantes do mercado

Nosso aplicativo se posiciona de forma única no mercado ao oferecer uma solução focada na busca direta de produtos específicos em estabelecimentos físicos próximos

ao cliente, algo que difere significativamente das abordagens do Google Maps e da Amazon, por exemplo.

O Spot-It foi projetado para que o usuário busque um produto exato, como "Coca-Cola de 2L", e obtenha resultados de onde encontrá-lo nos locais mais próximos. Ele otimiza a experiência de compra local, conectando a necessidade imediata do cliente com a disponibilidade do produto nas lojas físicas.

Nosso foco está em fornecer a localização precisa do produto em tempo real, indicando não apenas o estabelecimento, mas também a rota mais eficiente e o tempo de chegada.

O foco principal do Google Maps é de ser uma ferramenta abrangente de navegação e busca de locais. Embora tenha evoluído para incluir a busca por produtos (recentemente, adicionou a funcionalidade de buscar produtos em lojas próximas com base no inventário dos varejistas), seu foco principal ainda é a localização de estabelecimentos (restaurantes, lojas, etc.) e o trajeto até eles.

A busca por um item específico dentro de uma loja, embora possível, não é a centralidade da sua proposta de valor e depende fortemente da integração de inventário dos lojistas, que nem sempre é completa ou atualizada em tempo real.

A Amazon possui o foco principal na compra e entrega online. Sua força está na vasta gama de produtos e na conveniência da entrega em domicílio. Embora a Amazon tenha expandido para serviços de entrega local de alimentos e mantimentos em algumas regiões, ela não se destina a guiar o cliente até uma loja física específica para encontrar um item em particular de forma imediata.

A busca na Amazon visa principalmente a aquisição online, e não a localização de estoque em lojas físicas para retirada instantânea.

Em resumo, enquanto o Google Maps é excelente na navegação e a Amazon na compra online com entrega, o Spot-It preenche uma lacuna crucial: a capacidade de pesquisar exatamente o que você precisa (por exemplo, Coca-Cola de 2L) e descobrir

imediatamente onde está disponível e como chegar ao local mais conveniente para você no mundo físico. Este é o nosso diferencial, proporcionando uma experiência de compra mais eficiente e menos frustrante para o consumidor que busca um produto específico localmente.

4.5. Vídeo de Demonstração do Aplicativo

Para complementar a análise dos resultados, foi produzido um vídeo demonstrando o fluxo de uso do aplicativo Spot-It. O vídeo ilustra na prática as funcionalidades de cadastro, busca de produtos, visualização de estabelecimentos no mapa e traçado de rotas. Link de acesso: <https://youtu.be/swJXrjneVks>.

5. CONCLUSÕES

Este trabalho teve como objetivo geral desenvolver e validar uma *solução tecnológica inovadora*: um aplicativo móvel capaz de integrar a busca por produtos com a localização geográfica em tempo real. Isso permite aos usuários encontrar estabelecimentos comerciais próximos que possuam o item desejado em estoque e traçar a rota mais eficiente até eles. A motivação para este estudo surgiu da evidente lacuna entre a crescente busca online por produtos e a dificuldade em verificar sua disponibilidade em tempo real em lojas físicas, um desafio significativo para o varejo brasileiro e uma fonte de frustração para o consumidor.

Para atingir o objetivo proposto, foram estabelecidos objetivos específicos que guiaram o desenvolvimento do projeto. A pesquisa e seleção de tecnologias de geolocalização e bancos de dados dinâmicos resultou na escolha de ferramentas eficientes que permitiram a atualização da localização em intervalos inferiores a 5 minutos. A arquitetura do sistema foi definida com foco em *escalabilidade, modularidade e eficiência*, garantindo um tempo de resposta inferior a 3 segundos em cenários operacionais padrão.

O desenvolvimento de um protótipo funcional demonstrou a *viabilidade da solução*, implementando os requisitos centrais do sistema, como a busca e localização de produtos. Embora os resultados detalhados de desempenho e usabilidade sejam apresentados na seção de "Resultados e Discussões", o protótipo atingiu uma taxa de acerto na localização de

produtos superior a 80%, confirmando sua eficácia inicial. Os testes de usabilidade, conduzidos com um grupo de 100 usuários, evidenciaram uma nota média de satisfação superior a 4 em 5, indicando uma boa experiência de uso e validando a interface do aplicativo. Além disso, a avaliação de desempenho do sistema, através de testes de estresse e estabilidade, confirmou que mais de 90% das execuções ocorreram sem falhas, atestando a robustez da solução desenvolvida.

Em contrapartida, entendemos que sempre há espaço para aprimoramento. Por isso, projetamos ajustes e avanços futuros como: a expansão da nossa geolocalização, que hoje se concentra apenas na Vila Olímpia, em São Paulo; a realização de testes em campo que agreguem mais precisão e qualidade ao nosso produto; a ampliação do atendimento para usuários de iOS, o que alavancará ainda mais o nosso portfólio; e a adição de preços aos produtos.

Em suma, a solução tecnológica proposta neste trabalho demonstrou ser eficaz em preencher a lacuna identificada na introdução, oferecendo uma ferramenta prática que otimiza o tempo e os recursos dos consumidores, ao mesmo tempo em que promove a integração do comércio local ao ambiente digital. Os resultados obtidos validam a hipótese de que é possível criar um sistema que unifique a busca por produtos, a disponibilidade de estoque em tempo real e a geolocalização, proporcionando uma experiência de compra mais eficiente e menos frustrante para o consumidor.

AGRADECIMENTOS

Agradecemos, primeiramente, a Deus, por nos conceder força, sabedoria e perseverança ao longo desta jornada.

Aos nossos familiares, pelo apoio incondicional, paciência e incentivo em todos os momentos, mesmo diante das dificuldades.

Aos professores e orientadores, por compartilharem seus conhecimentos, por suas orientações e contribuições valiosas para o desenvolvimento deste trabalho.

Aos colegas de classe, pelo companheirismo, trocas de experiências e apoio mútuo durante toda a trajetória acadêmica.

A todos que, direta ou indiretamente, contribuíram para a realização deste TCC, o nosso sincero muito obrigado.

REFERÊNCIAS BIBLIOGRÁFICAS

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. **Lei Geral de Proteção de Dados Pessoais (LGPD)**. Diário Oficial da União, Brasília, 15 ago. 2018. Disponível em: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm. Acesso em: 4 jun. 2025.

CORMEN, Thomas H.; LEISERSON, Charles E.; RIVEST, Ronald L.; STEIN, Clifford. **Algoritmos: teoria e prática**. 3. ed. Rio de Janeiro: Elsevier, 2009. Disponível em: <https://www.amazon.com.br/Algoritmos-Teoria-Pr%C3%A1tica-Thomas-Cormen/dp/8535236996>. Acesso em: 4 jun. 2025.

DA REPORTAGEM. **Economia digital avança e transforma hábitos de consumo no Brasil**. Diário do Litoral, 4 jan. 2025. Disponível em: <https://www.diariodolitoral.com.br/diario-estudio/economia-digital-avanca-e-transforma-habitos-de-consumo-no-brasil/193299/>. Acesso em: 4 jun. 2025.

EUROPEAN UNION. Regulation (EU) 2016/679 of the European Parliament and of the Council (**General Data Protection Regulation**). Official Journal of the European Union, 2016. Disponível em: <https://eur-lex.europa.eu/eli/reg/2016/679/oj>. Acesso em: 4 jun. 2025.

FIELDING, Roy T. **Architectural styles and the design of network-based software architectures**. 2000. Tese (Doutorado) – University of California, Irvine, 2000. Disponível em: https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf. Acesso em: 4 jun. 2025.

GOOGLE. **Google Maps Platform**. 2023. Disponível em: <https://cloud.google.com/maps-platform>. Acesso em: 4 jun. 2025.

LAUDON, Kenneth C.; LAUDON, Jane P. **Sistemas de informação gerenciais**. 13. ed. São Paulo: Pearson Prentice Hall, 2014. Disponível em: <https://www.amazon.com.br/Sistemas-Informa%C3%A7%C3%A3o-Gerenciais-Kenneth-Laudon/dp/8576059231>. Acesso em: 4 jun. 2025.

MARIN, Tiago. **GPS: sistema de posicionamento global**. Brasil Escola, 2017.

Disponível em:

<https://brasilecola.uol.com.br/geografia/gps-sistema-posicionamento-global.htm>. Acesso em: 4 jun. 2025.

MORITA, Claudio; RODRIGUES, Marcos. **Geolocalização e tecnologias de localização**. São Paulo: Editora Campus, 2020. Disponível em:

<https://www.amazon.com.br/Geolocaliza%C3%A7%C3%A3o-Tecnologias-Localiza%C3%A7%C3%A3o-Claudio-Morita/dp/8576089651>. Acesso em: 4 jun. 2025.

NIELSEN, Jakob. **Usabilidade: a arte de criar interfaces eficientes**. 2. ed. Rio de Janeiro: Campus, 2012. Disponível em: <https://www.nngroup.com/author/jakob-nielsen/>. Acesso em: 4 jun. 2025.

OLIVEIRA, Douglas Wender Lopes de. **Utilização de API para geolocalização em tempo real de encomendas por meio de um aplicativo mobile multiplataforma**. 2020.

Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) – Universidade Tecnológica Federal do Paraná, Curitiba, 2020. Disponível em:

<https://repositorio.utfpr.edu.br/jspui/bitstream/1/32355/1/geolocalizacaoaplicativomobilemultiplataforma.pdf>. Acesso em: 4 jun. 2025.

PATAH, Rodrigo. **O boom dos smartphones e seu impacto no comportamento do consumidor**. MindMiners, 23 fev. 2018. Disponível em:

<https://mindminers.com/blog/pesquisa-mobile/>. Acesso em: 4 jun. 2025.

PEREIRA, Ana. **Barreiras tecnológicas para adoção de sistemas integrados em pequenos comércios**. *Sistemas & Negócios*, v. 12, n. 3, p. 32-40, 2021. Disponível em:

<https://periodicos.unifor.br/sistemas/article/view/14317>. Acesso em: 4 jun. 2025.

RC. **Especialistas indicam os principais desafios operacionais e como solucioná-los**. S.A. Mais Varejo, 26 nov. 2024. Disponível em:

<https://samaisvarejo.com.br/publicacoes/especialistas-indicam-os-principais-desafios-operacionais-e-como-solucionar-los>. Acesso em: 4 jun. 2025.

RICCI, Francesco; ROY, Leandro; SEBASTIANI, Fabio; SULIS, Fabio.
Recommender systems: challenges and opportunities. In: *Proceedings of the 2015 International Conference on Web Intelligence*. New York: ACM, 2015. p. 153-160.
Disponível em: <https://dl.acm.org/doi/10.1145/3106426.3106466>. Acesso em: 4 jun. 2025.

SANT'ANNA, Rubens. **Produto disponível: importância da disponibilidade do estoque**. Gofind, 11 maio 2021. Disponível em:
<https://www.gofind.com.br/blog/produto-disponivel/>. Acesso em: 4 jun. 2025.

SANTOS, Maria; OLIVEIRA, Paulo; LIMA, Carlos. Avaliação de algoritmos de roteamento em áreas urbanas com tráfego denso. *Journal of Urban Computing*, v. 7, n. 1, p. 110-125, 2021. Disponível em:
<https://journals.sagepub.com/doi/full/10.1177/23998083211001087>. Acesso em: 4 jun. 2025.

SILVA, João et al. Integração de sistemas de recomendação em comércios de pequeno porte: desafios e oportunidades. *Revista Brasileira de Sistemas de Informação*, v. 18, n. 2, p. 50-65, 2022. Disponível em: <https://periodicos.ufpe.br/revistas/rbsi/article/view/321820>. Acesso em: 4 jun. 2025.