

CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA

Faculdade de Tecnologia de Mauá

Curso Superior de Tecnologia em Desenvolvimento de Software Multiplataforma

Otávio Barbosa Ferreira

Victor Nery Eduardinho Reis

Fatec Lanches: Navegação do Cliente

Mauá

2026

Otávio Barbosa Ferreira
Victor Nery Eduardinho Reis

Fatec Lanches: Navegação do Cliente

Relatório técnico-científico apresentado à
Fatec de Mauá, como exigência parcial
para obtenção do título de Tecnólogo em
Desenvolvimento de Software
Multiplataforma, sob a orientação da profa.
Me. Luana Lourenço.

RESUMO

Este trabalho tem como propósito, apresentar o desenvolvimento de um sistema de navegação intuitivo para clientes dentro do aplicativo Fatec Lanches, uma solução móvel projetada para otimizar o fluxo de pedidos e pagamentos na cantina presente dentro da FATEC Mauá. O objetivo principal deste estudo consiste em estruturar uma arquitetura de navegação fluida que reduza o tempo de espera dos usuários e organize a interação entre o catálogo de produtos e o checkout. A metodologia adotada fundamenta-se no desenvolvimento ágil de software com a utilização do framework React Native e o banco de dados não relacional MongoDB, priorizando a usabilidade e a responsividade em dispositivos móveis. Os resultados demonstram que a implementação de uma hierarquia visual clara, integrada a uma navegação por abas e componentes dinâmicos, proporciona uma experiência de uso eficiente, mitigando erros durante a seleção de itens. Conclui-se que a correta estruturação da área de navegação do cliente é fundamental para a aceitação tecnológica da solução no ambiente acadêmico, servindo como modelo para a modernização de processos em estabelecimentos de pequeno porte.

Palavras-chave: desenvolvimento móvel; interface do usuário; sistemas de informação; experiência do cliente.

ABSTRACT

This work aims to present the development of an intuitive navigation system for the Fatec Lanches application, a mobile solution designed to optimize the order and payment flow at the FATEC Mauá university canteen. The main objective of this study is to structure a fluid navigation architecture that reduces user waiting times and organizes the interaction between the product catalog and checkout. The methodology adopted is based on agile software development using the React Native framework and the non-relational MongoDB database, prioritizing usability and responsiveness on mobile devices. The results demonstrate that the implementation of a clear visual hierarchy, integrated with tab-based navigation and dynamic components, provides an efficient user experience, mitigating operational errors during item selection. It is concluded that the correct structuring of the client navigation area is fundamental for the technological acceptance of the solution in the academic environment, serving as a model for the modernization of processes in small establishments.

Keywords: mobile development; user interface; information systems; customer experience.

SUMÁRIO

1	INTRODUÇÃO	5
1.1	Contextualização	6
1.2	Objetivo	6
1.3	Justificativa	7
1.4	Problema de pesquisa	8
1.5	Estrutura do relatório	8
2	DESENVOLVIMENTO	9
2.1	Fundamentação teórica	9
2.1.1	Revisão de literatura	10
2.1.2	Técnicas de desenvolvimento de software	11
2.1.1	Público-alvo e Projeção de interface	13
2.1.3	Linguagens de programação e frameworks	16
2.1.4	Metodologias de teste de software	17
2.2	Metodologia	18
2.2.1	Tipo de pesquisa	19
2.2.2	Método de coleta de dados	19
2.2.3	Procedimentos de análise	20
2.2.4	Critérios de avaliação	20
2.3	Ficha técnica	21
2.4	Técnicas e testes de software e avaliação da acessibilidade	22
2.4.1	Testes unitários	22
2.4.2	Testes de integração	23
2.4.3	Testes de segurança	24
2.4.4	Metodologias de desenvolvimento e qualidade	25
2.4.5	Avaliação de acessibilidade	25
3	CONSIDERAÇÕES FINAIS	27

3.1	Principais conclusões	27
3.2	Recomendações	28
3.3	Limitações do estudo	28
3.4	Perspectivas futuras	30
	REFERÊNCIAS	31
	FORMULÁRIO DE IDENTIFICAÇÃO	32

1 INTRODUÇÃO

O presente relatório técnico descreve formalmente a trajetória e os resultados obtidos durante o desenvolvimento e pesquisa aplicada sobre o sistema Fatec Lanches, focado na navegação do cliente, sendo uma plataforma de gestão de pedidos e pagamentos para a cantina da FATEC Mauá. Em conformidade com a ABNT NBR 10719:2015, este documento detalha o progresso obtido na digitalização de serviços de conveniência em ambientes acadêmicos, apresentando as soluções adotadas para otimizar a experiência do usuário final.

A relevância deste estudo técnico reside na análise da ineficiência dos métodos tradicionais de atendimento, que geram gargalos operacionais e longos tempos de espera para o consumidor. O problema de pesquisa concentra-se na dificuldade do usuário em realizar pedidos de forma ágil em intervalos reduzidos. Como solução, propõe-se a implementação de uma interface móvel integrada a uma arquitetura de microserviços, permitindo a descentralização do processamento das requisições de compra e a automação do fluxo transacional e realizado de forma autônoma pelo aluno, otimizando o fluxo desde a escolha do produto até a emissão do pedido para a unidade produtora.

O objetivo deste trabalho está focado especialmente em relatar o desenvolvimento da vertente do cliente, abrangendo desde a concepção da interface de navegação até a integração com sistemas de persistência de dados e protocolos de segurança. Especialmente, está documentado a aplicação do framework React Native para garantir a portabilidade do software e a utilização do banco de dados MongoDB para a gestão eficiente de informações.

A metodologia técnica empregada seguiu ciclos de desenvolvimento incremental, pautados por testes de usabilidade e modelagem de ameaças via STRIDE. A aplicação dessas técnicas permitiu a identificação prévia de possíveis riscos de segurança e previram implementações de medidas mitigadoras, garantindo assim a integridade do produto final.

Este relatório está organizado de forma a apresentar, inicialmente, a fundamentação tecnológica e o desenvolvimento do sistema; em seguida, os resultados obtidos nos testes de performance e usabilidade; e, por fim, as considerações finais sobre a eficácia da ferramenta como resposta ao problema inicialmente proposto, consolidando as contribuições técnicas desta pesquisa para a área de sistemas de informação.

1.1 Contextualização

Na atualidade, a agilidade nos serviços que envolvem o ramo alimentício, desde a preparação até a entrega do pedido, é fortemente influenciada pela crescente adoção de tecnologias dispostas no mercado, sendo elas, sites para marketing do produto, sistemas de controle ou sistemas de vendas. Sua maior concentração se encontra no uso de aparelhos celulares, por serem algo mais prático no dia a dia de cada pessoa e pela demanda por conveniência e eficiência. Sistemas de pedidos e pagamentos via aplicativo transformaram a experiência do consumidor, reduzindo filas e otimizando o tempo de espera. Nesse cenário, o desenvolvimento de software multiplataforma emerge como uma solução estratégica para atender a um mercado fragmentado de sistemas operacionais, permitindo que uma única base de código seja utilizada para construir aplicações nativas em ambos os ambientes. Tecnologias como o React Native, com sua capacidade de renderizar componentes nativos, e bancos de dados NoSQL, como o MongoDB, que oferecem flexibilidade e escalabilidade para dados não estruturados, tornaram-se pilares para a criação de aplicações robustas e escaláveis. Essas ferramentas proporcionam interfaces fluidas e responsivas, aproximando o consumidor final do serviço e eliminando barreiras físicas e temporais no atendimento, o que é particularmente relevante em ambientes de alta demanda como cantinas universitárias.

1.2 Objetivo

O objetivo deste relatório é apresentar a análise do desenvolvimento e dos resultados obtidos na navegação do cliente dentro do aplicativo Fatec Lanches, sistema criado para otimizar a experiência de navegação e reduzir a carga cognitiva durante a interação com o sistema. escolha de produtos e realização de pedidos na cantina da FATEC Mauá. O estudo tem como foco observar a integração entre as camadas de frontend e backend, bem como a aplicação de práticas voltadas à segurança, testes e validação da experiência de uso, assegurando que o sistema atenda aos requisitos funcionais e não funcionais definidos para o contexto acadêmico.

Para alcançar esses objetivos, foram consideradas algumas diretrizes principais, como a adequação do sistema ao público-alvo, a observação do comportamento dos usuários durante a utilização de aplicativos de compra e a busca por uma navegação simples, rápida e intuitiva. Além disso, o desenvolvimento foi direcionado para garantir estabilidade, desempenho e confiabilidade, proporcionando uma experiência de uso fluida mesmo em situações de maior demanda.

1.3 Justificativa

A aplicação de técnicas estruturadas e testes rigorosos no desenvolvimento do Fatec Lanches reflete o compromisso em construir uma solução de qualidade, estável e confiável. Essa abordagem é essencial para garantir que o sistema atenda às necessidades do projeto de forma eficiente, segura e consistente, destacando-se por três motivos principais:

- **Confiabilidade e Segurança:** Como o sistema lida com pedidos e transações de usuários, a aplicação de modelos de segurança (como o STRIDE) e testes de software garante que os dados dos alunos estejam protegidos contra invasões ou falhas técnicas. A ausência de estratégias sistemáticas de validação compromete os atributos de confiabilidade, disponibilidade e segurança da informação.
- **Eficiência e Desempenho:** A aplicação de padrões de design e práticas de refatoração (*Clean Code*) garante que o artefato de *software* apresente um consumo otimizado de recursos de *hardware* (como ciclos de CPU e memória volátil), mitigando riscos de degradação de performance (*memory leaks*). Isso resulta em uma aplicação com tempo de resposta (*throughput*) reduzido durante o carregamento de dados assíncronos do cardápio, assegurando a estabilidade do sistema sob alta concorrência de acessos nos períodos de intervalo acadêmico.
- **Facilidade de Manutenção:** O emprego de padrões arquiteturais adequados confere ao *software* características de baixo acoplamento e alta coesão. Essa modularização simplifica a manutenção corretiva e evolutiva, reduzindo a complexidade ciclomática do código. Dessa forma, viabiliza-se a escalabilidade

do sistema para a inserção de novas *features* sem comprometer a integridade das funcionalidades legadas.

1.4 Problema de pesquisa

O estudo busca responder à seguinte questão: “de que maneira a implementação de uma arquitetura multiplataforma, aliada a práticas de modelagem de ameaças e design focado no usuário, pode otimizar o fluxo de atendimento e garantir a integridade das operações em um sistema de cantina universitária? ”

1.5 Estrutura do relatório

Este relatório técnico-científico está organizado em seções que descrevem a trajetória técnica do projeto Fatec Lanches, conforme detalhado a seguir:

Capítulo 1 – Introdução: Apresenta a contextualização do tema, os objetivos gerais e específicos, a justificativa da pesquisa e o problema que o sistema Fatec Lanches visa resolver.

Capítulo 2 – Desenvolvimento: É o núcleo técnico do trabalho, subdividido em quatro pilares. A Fundamentação Teórica aborda a revisão de literatura e as tecnologias utilizadas. A Metodologia explica o tipo de pesquisa e a coleta de dados. A Ficha Técnica descreve as especificações do produto, enquanto a seção de Técnicas e Testes detalha as avaliações de performance, segurança, acessibilidade e qualidade do software.

Capítulo 3 – Considerações Finais: Sintetiza as principais conclusões obtidas, destacando as contribuições da pesquisa, as limitações encontradas durante o desenvolvimento e as recomendações para futuras expansões da ferramenta.

Elementos Pós-Textuais: Apresentam as referências bibliográficas, seguidas por anexos e apêndices que complementam a documentação técnica do sistema.

2 DESENVOLVIMENTO

A base teórica deste trabalho sustenta-se sobre a análise de metodologias e tecnologias que permitem a construção de sistemas robustos e escaláveis. O referencial teórico desta pesquisa foi estruturado a partir de buscas em bases de dados como Google Acadêmico e Consensus. A pesquisa bibliográfica concentrou-se em autores e produções científicas que abordam a evolução do desenvolvimento móvel e as metodologias de garantia de qualidade de software, assegurando o rigor conceitual do estudo.

2.1 Fundamentação teórica

O setor alimentício possui grande relevância na sociedade, sendo responsável não apenas pelo fornecimento de refeições, mas também pela geração de empregos, movimentação econômica e atendimento das necessidades diárias da população. Dentro das instituições de ensino, as lanchonetes exercem papel importante na rotina acadêmica, oferecendo praticidade e alimentação rápida para alunos, professores e funcionários.

Com o aumento da circulação de pessoas em ambientes educacionais, tornou-se necessário que estabelecimentos alimentícios buscassem formas mais eficientes de organização e atendimento. Segundo Kotler (2017), a satisfação do cliente está diretamente relacionada à qualidade do serviço prestado, à agilidade no atendimento e à capacidade da organização em atender às necessidades dos consumidores de forma eficiente.

Nesse contexto, muitas lanchonetes ainda realizam processos manualmente, como anotações de pedidos, controle de estoque e organização de pagamentos, o que pode ocasionar filas, atrasos, falhas de comunicação e dificuldades administrativas. Para Laudon e Laudon (2021), os sistemas de informação possuem como principal objetivo auxiliar organizações na automação de processos, melhoria da tomada de decisões e aumento da eficiência operacional.

A transformação digital contribuiu significativamente para mudanças na forma como empresas e consumidores interagem. Atualmente, aplicativos de delivery e gerenciamento de pedidos tornaram-se comuns no cotidiano das pessoas,

proporcionando maior praticidade e rapidez no atendimento. Plataformas como iFood, Rappi e Uber Eats popularizaram o uso de tecnologias voltadas ao setor alimentício, permitindo pedidos digitais, acompanhamento em tempo real e maior organização dos estabelecimentos.

Segundo Valente (2020), o software está presente em praticamente todas as áreas da sociedade moderna, automatizando tarefas e facilitando a interação entre usuários e serviços. Dessa forma, observa-se que a tecnologia deixou de ser apenas um diferencial competitivo e passou a representar uma necessidade para organizações que desejam melhorar seus processos e acompanhar as mudanças da sociedade contemporânea.

Além disso, Baltzan (2016) afirma que organizações que desejam permanecer competitivas precisam adaptar-se às novas tecnologias e implementar soluções digitais capazes de otimizar processos e melhorar a experiência dos usuários. Nesse sentido, sistemas informatizados podem auxiliar no controle de pedidos, organização do atendimento e redução de erros operacionais em estabelecimentos alimentícios.

Dessa forma, a utilização de um sistema para gerenciamento de pedidos em uma lanchonete universitária pode contribuir significativamente para a melhoria do atendimento, redução do tempo de espera e maior organização administrativa. A adoção de recursos tecnológicos possibilita maior eficiência nos processos internos e melhor experiência para os usuários da cantina da Faculdade de Tecnologia de Mauá.

Assim, o desenvolvimento de um sistema voltado para a lanchonete da instituição busca atender às necessidades de modernização e eficiência presentes no ambiente acadêmico, acompanhando as transformações digitais que vêm redefinindo a maneira como organizações e consumidores interagem na sociedade atual.

2.1.1 Revisão de literatura

O desenvolvimento de software moderno é marcado pela transição de modelos rígidos para metodologias ágeis. O uso de metodologias ágeis permite maior adaptação a mudanças durante o desenvolvimento de software (SOMMERVILLE, 2011). Conforme Sommerville (2011), os métodos ágeis, como Scrum e Kanban, focam na entrega incremental e na resposta rápida a mudanças, o que é essencial em

projetos de aplicativos móveis. Além disso, o Manifesto Ágil estabelece como princípio a valorização de indivíduos e interações, software em funcionamento e adaptação contínua às mudanças, promovendo maior flexibilidade e colaboração no processo de desenvolvimento (BECK et al., 2001).

No contexto de automação de campus, a literatura aponta que a maioria das cantinas universitárias ainda depende de sistemas baseados em papel ou fichas físicas, o que resulta em erros humanos, demora no atendimento e gargalos operacionais (KOGTA, 2020). A transição para o modelo *mobile* reflete a evolução das interfaces, que deixaram de ser ferramentas estáticas para se tornarem ecossistemas de serviços. A cultura dos "Super Apps" moldou a expectativa do usuário por agilidade e autonomia. Conforme apontam Soni et al. (2024), aplicativos de automação de campus, como o **PUCollegiFy**, simplificam tarefas e melhoram significativamente a experiência do estudante ao centralizar serviços. Complementarmente, a "próxima geração" de sistemas de pedidos introduz o rastreamento em tempo real e recomendações personalizadas, elevando o nível de satisfação do usuário (MEENA; PRANESHWAR; HARSHIVAN, 2025).

No campo da interface, frameworks como React, Flutter e Angular revolucionaram a produtividade ao permitir a reutilização de componentes. Além disso, práticas como o TDD (Test Driven Development) e o BDD (Behavior Driven Development) são fundamentais para alinhar o código às necessidades reais do negócio, garantindo que cada funcionalidade seja validada antes mesmo de sua implementação final.

2.1.2 Técnicas de desenvolvimento de software

As técnicas de desenvolvimento evoluíram para suportar ciclos de lançamento cada vez mais curtos e sistemas altamente resilientes. No contexto do Fatec Lanches, a base para a sustentabilidade do projeto reside na escrita de "código limpo" (*Clean Code*). Segundo Martin (2009), esta prática é indispensável, pois um código bem estruturado e documentado reduz drasticamente os custos de manutenção e facilita a evolução do sistema por diferentes membros da equipe. No entanto, a aplicação do código limpo no aplicativo do usuário não é apenas uma escolha estética de

programação, mas uma necessidade técnica para viabilizar a **arquitetura de microsserviços** que sustenta a descentralização do processo de compra.

A arquitetura proposta foca na transferência da etapa de escolha, customização do item e pagamento para o dispositivo móvel do aluno, transformando o smartphone em um terminal de autoatendimento portátil. Isso remove a carga de trabalho do balcão e organiza os pedidos de forma digital, permitindo que a unidade produtora (cantina) receba os dados prontos para a execução, conforme o modelo de automação proposto por Kogta (2020). Este modelo fundamenta-se em uma estrutura de três camadas — interface, servidor e banco de dados — que trabalham em sincronia para eliminar o erro humano inerente aos métodos manuais, garantindo que a entrada de dados seja feita diretamente pelo consumidor.

Para operacionalizar essa automação descrita por Kogta, utilizam-se técnicas avançadas de interface e comunicação. Uma delas é o **Design de Componentes Reutilizáveis (Atomic Design)**. Ao dividir a interface em botões de compra e cards de itens, garante-se a consistência visual, a agilidade na manutenção e implementação. Complementarmente, o aplicativo emprega o **Gerenciamento de Estado Global** para assegurar que o carrinho de compras do aluno permaneça sincronizado durante toda a navegação, evitando a perda de dados entre as telas de seleção de produtos e o checkout financeiro.

A comunicação entre essas camadas ocorre via **APIs RESTful**, utilizando a biblioteca *Axios*. Esta técnica permite que o aplicativo do usuário opere de forma independente e assíncrona; quando um pedido é finalizado, o app dispara um método POST e entra em estado de "escuta". O modelo de Kogta (2020) prevê que essa integração seja refletida em um *dashboard* administrativo em tempo real, onde o status do pedido é atualizado. Para mitigar instabilidades de rede comuns em ambientes acadêmicos, aplicam-se técnicas de **Persistência de Dados Local (Caching)** e interceptores de erro, garantindo que o cardápio seja carregado instantaneamente e que o usuário seja informado de forma amigável sobre qualquer falha de conexão.

Por fim, a automação proposta culmina na otimização da gestão de fluxo e suprimentos através da ferramenta de orquestração de fluxos n8n, que foi implementada para emitir alertas automáticos e relatórios periódicos sobre o status do

estoque diretamente aos administradores. Adicionalmente, a técnica de Notificação Pós-Processamento — que enviará um alerta ao dispositivo do aluno assim que o status do pedido for alterado para "Pronto para Retirada" no banco de dados — encontra-se em fase de integração ao ecossistema do projeto. Esse conjunto de automações visa fechar o ciclo operacional, garantindo que o fluxo de pessoas no balcão ocorra sob demanda e que a cozinha opere com previsibilidade de insumos. Dessa forma, o desenvolvimento técnico implementa um fluxo inteligente que otimiza o espaço físico da instituição, reduz o tempo de espera e eleva o padrão de eficiência do serviço de alimentação.

Figura 1 – Fluxo N8N



Fonte: Victor Nery (2026, p.13)

2.1.3 Público-alvo e Projeção de Interface

Para que o desenvolvimento fosse orientado ao usuário, definiu-se o público-alvo composto por alunos, professores e funcionários da FATEC Mauá, além dos administradores da cantina.

Antes da codificação, foi realizada a projeção da interface (UI/UX) utilizando a ferramenta Figma. Este procedimento permitiu:

- Prototipagem de Alta Fidelidade: Criação de telas interativas para simular a jornada do usuário, desde a escolha do item até a finalização do pagamento.

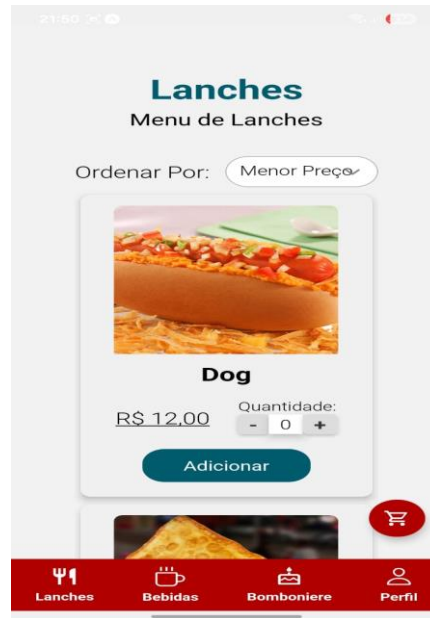
- Fluxogramas de Navegação: Definição da arquitetura de informação para garantir que o autoatendimento fosse intuitivo e reduzisse o tempo de fila.
- Validação Visual: Ajustes estéticos e de usabilidade baseados no feedback preliminar, garantindo que a aplicação móvel atendesse às expectativas ergonômicas do público universitário.

Figura 2 – Menu de Opções



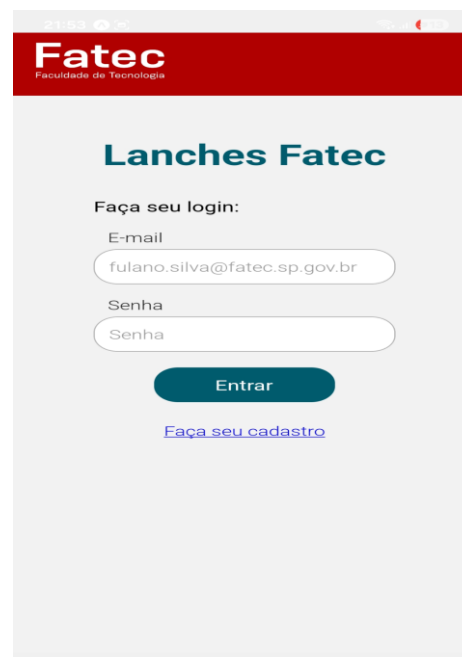
Fonte: Victor Nery (2026, p.14)

Figura 3 – Cardápio de Lanches



Fonte: Victor Nery (2026, p.15)

Figura 4 – Página de Login



Fonte: Victor Nery (2026, p.15)

2.1.4 Linguagens de programação e frameworks

A seleção das linguagens de programação e dos frameworks foi concebida na análise dos pilares fundamentais para a durabilidade de qualquer projeto de software, especialmente no contexto de soluções multiplataforma. Atualmente, o cenário de desenvolvimento móvel apresenta abordagens consolidadas que servem de referência para sistemas de automação de campus e pedidos de conveniência:

- **JavaScript com React Native:** É a combinação líder para projetos que buscam alto desempenho e compartilhamento de lógica. Conforme Kogta (2020), o uso de aplicações *cross-platform* (multiplataforma) é a solução ideal para sistemas de cantina, pois permite que o software seja acessível a qualquer estudante ou funcionário, independentemente do sistema operacional (Android ou iOS) de seu dispositivo pessoal.
- **Dart com Flutter:** Uma alternativa robusta conhecida por sua renderização de alto desempenho. De acordo com Soni et al. (2024), frameworks que permitem uma única base de código são essenciais para a automação de campus, como demonstrado no projeto **PUCollegiFy**, pois reduzem a complexidade de manutenção e melhoram a navegação do usuário.
- **C# com Xamarin (.NET MAUI):** Utilizado primordialmente em ecossistemas que exigem integração profunda com serviços Microsoft, garantindo robustez em aplicações empresariais.

Para o desenvolvimento do **Fatec Lanches**, escolhemos seguir com a combinação de **JavaScript e React Native** para o melhor desenvolvimento do projeto em grupo. O JavaScript, sendo uma linguagem versátil e orientada a eventos, oferece a agilidade necessária para aplicações que exigem alta interatividade. Conforme Meena et al. (2025), a "próxima geração" de sistemas de pedidos requer funcionalidades como o rastreamento em tempo real, capacidade que o ecossistema JavaScript suporta através de sua natureza assíncrona.

O React Native permite que o aplicativo do usuário ofereça uma experiência de performance próxima à nativa. Isso ocorre através de uma "ponte" (*bridge*) JavaScript que interage diretamente com os componentes do sistema operacional. Esta escolha técnica justifica-se pela necessidade de garantir que o aluno finalize seu pedido de

forma ágil, minimizando o tempo de resposta da interface durante os curtos intervalos acadêmicos.

Para a persistência de dados, a arquitetura utiliza o **MongoDB**, um banco de dados NoSQL orientado a documentos. Diferente dos modelos relacionais tradicionais, o MongoDB permite o armazenamento de dados em formato BSON (JSON binário), o que confere uma flexibilidade indispensável para catálogos de produtos dinâmicos. Essa escolha é sustentada pela necessidade de escalabilidade horizontal e alta disponibilidade; conforme o modelo de automação proposto por Kogta (2020), o sistema deve ser capaz de gerenciar picos de acesso simultâneos e volumes crescentes de dados sem perda de performance, características inerentes à estrutura distribuída do MongoDB. Assim, a *stack* tecnológica escolhida não apenas viabiliza a execução técnica, mas garante que o sistema atenda aos requisitos de eficiência e confiabilidade exigidos em um ambiente universitário de alto fluxo.

2.1.5 Metodologias de teste de software

A qualidade do software é garantida por meio de diferentes níveis de testes. Os testes unitários verificam partes isoladas do código, enquanto os testes de integração validam a comunicação entre diferentes módulos, como o aplicativo e o banco de dados. Para a interface, utilizam-se testes de aceitação e performance para garantir que o usuário tenha uma experiência fluida. Ferramentas como JUnit e Mockito são essenciais para a automação de testes em ambientes Java, enquanto no ecossistema JavaScript, ferramentas de automação como Jest e Selenium desempenham papel semelhante, assegurando que o sistema se comporte conforme o esperado sob diversas condições de uso.

A qualidade do software é garantida por meio de diferentes níveis de testes, que asseguram a estabilidade e a confiabilidade do sistema sob diversas condições de uso. Conforme destacado por Soni et al. (2024) em seu estudo sobre automação de campus, a integração de múltiplas funções em um único ambiente móvel exige que a validação técnica seja rigorosa para garantir uma experiência de usuário sem interrupções.

- **Testes Unitários:** Verificam partes isoladas do código para garantir que as funções de lógica interna — como o cálculo de preços e validações de formulários — operem corretamente. No ecossistema JavaScript do Fatec Lanches, a ferramenta **Jest** é utilizada para automatizar esses processos, garantindo que pequenas alterações no código não quebrem funcionalidades existentes.
- **Testes de Integração:** Validam a comunicação entre os diferentes módulos, como o aplicativo móvel e o banco de dados MongoDB. Conforme Meena, Praneshwar e Harshivan (2025), a precisão na troca de dados entre o pedido do usuário e o sistema de rastreamento é o que define o sucesso de sistemas de pedidos modernos, evitando discrepâncias entre o que foi pedido e o que será preparado.
- **Testes de Aceitação e Performance:** Utilizam-se para garantir que a interface tenha uma resposta fluida. A performance é um critério crítico para evitar filas virtuais; sistemas eficientes devem processar requisições de forma assíncrona para suportar picos de acesso, utilizando ferramentas de automação para simular múltiplos usuários simultâneos (SONI et al., 2024).
- **Segurança:** Em conformidade com as necessidades de proteção de dados em ambientes digitais, são realizados testes para validar os protocolos de autenticação e a integridade das transações, assegurando que o acesso ao sistema seja restrito a usuários legítimos.

2.2 Metodologia

A metodologia aplicada neste relatório técnico descreve os caminhos adotados para a concepção e validação do software, unindo a pesquisa teórica à aplicação prática no cotidiano da FATEC Mauá.

2.2.1 Tipo de pesquisa

O relatório técnico desenvolvido caracteriza-se como uma pesquisa aplicada, buscando desenvolver uma solução tecnológica voltada para um problema específico relacionado à gestão de pedidos da cantina da Fatec Mauá. Segundo Antônio Carlos Gil (2010), a pesquisa aplicada tem como objetivo gerar conhecimentos para aplicação prática, direcionados à solução de problemas específicos.

Além disso, este estudo possui caráter descritivo e experimental. Descritivo porque apresenta e analisa as etapas de desenvolvimento do sistema, bem como as necessidades observadas no ambiente da cantina universitária. Experimental porque envolve testes e validações das tecnologias utilizadas no desenvolvimento da aplicação, permitindo analisar o comportamento do sistema diante dos requisitos definidos.

A pesquisa também se configura como um estudo de caso, uma vez que está centrada na realidade da Faculdade de Tecnologia de Mauá, considerando as particularidades do fluxo de alunos, atendimento e organização da lanchonete da instituição. Esse tipo de abordagem possibilita compreender de maneira mais detalhada os processos existentes e identificar oportunidades de melhoria por meio da utilização de recursos tecnológicos.

No contexto deste trabalho, a metodologia permitiu acompanhar o desenvolvimento do sistema de forma contínua, realizando ajustes conforme as necessidades identificadas durante o processo de construção da aplicação. Dessa maneira, além de contribuir para a modernização do gerenciamento de pedidos da cantina universitária, a pesquisa também proporcionou aprendizado prático sobre desenvolvimento mobile, integração de banco de dados e aplicação de tecnologias voltadas à automação de processos.

2.2.2 Método de coleta de dados

A coleta de dados ocorreu de forma técnica e documental. Foram realizadas análises de código durante as fases de desenvolvimento para garantir a aderência aos padrões de projeto. Além disso, os dados sobre o desempenho do sistema foram extraídos por meio de testes de performance simulados e do monitoramento de logs

do banco de dados. A utilização de ferramentas de integração contínua permitiu coletar métricas automáticas sobre a estabilidade das novas funcionalidades implementadas no repositório do projeto.

2.2.3 Procedimentos de análise

Para analisar se o software cumpre o que foi proposto, os dados coletados passaram por procedimentos de validação técnica, organizados em três frentes:

- Execução de Testes Unitários: Verificação da lógica interna das funções de cálculo de pedido e autenticação.
- Testes de Integração: Análise da comunicação entre a interface móvel e as APIs de persistência de dados.
- Validação de Requisitos: Uso de ferramentas automatizadas para conferir se as funcionalidades entregues correspondem aos requisitos de negócio definidos no início do projeto, garantindo a replicabilidade dos resultados.

2.2.4 Critérios de avaliação

O software foi avaliado com base em critérios técnicos de qualidade, fundamentais para um sistema de autoatendimento:

- Funcionalidade: O sistema executa todas as tarefas prometidas (pedir, pagar, listar itens) sem erros?
- Performance: O tempo de resposta das telas e do banco de dados é adequado para evitar filas?
- Segurança: Os dados dos usuários e as transações estão protegidos contra acessos indevidos?
- Usabilidade: A interface é intuitiva o suficiente para que um novo aluno consiga fazer um pedido sem ajuda?

- Manutenibilidade: O código está organizado de forma que futuras atualizações sejam simples de implementar?

2.3 Ficha técnica

Aqui, as especificações técnicas do software ou sistema analisado devem ser apresentadas conforme o exemplo ilustrado no Quadro 2:

Quadro 2 – Modelo de ficha técnica

Nome do sistema/software:	Fatec Lanches
Versão:	1.0
Desenvolvedores:	Otávio Barbosa Ferreira, Victor Nery Eduardinho Reis
Data de criação:	07 de abril de 2025
Plataforma(s):	Android, iOS
Linguagens utilizadas:	JavaScript, TypeScript e Node.js
Frameworks:	React Native
Banco de dados:	MongoDB
Requisitos de hardware:	Navegador com suporte a HTML5, dispositivo com Android 8.0 ou superior, iOS 12 ou superior
Requisitos de software:	JavaScript, TypeScript Node.js, React Native, MongoDB
Ferramentas de desenvolvimento:	Visual Studio Code, Git, GitHub, MongoDB
Objetivo:	Agilizar o processo de compra de lanches e diminuir tempo de espera

Fonte: Estrutura adaptada pelas autoras.

2.4 Técnicas e testes de software e avaliação da acessibilidade

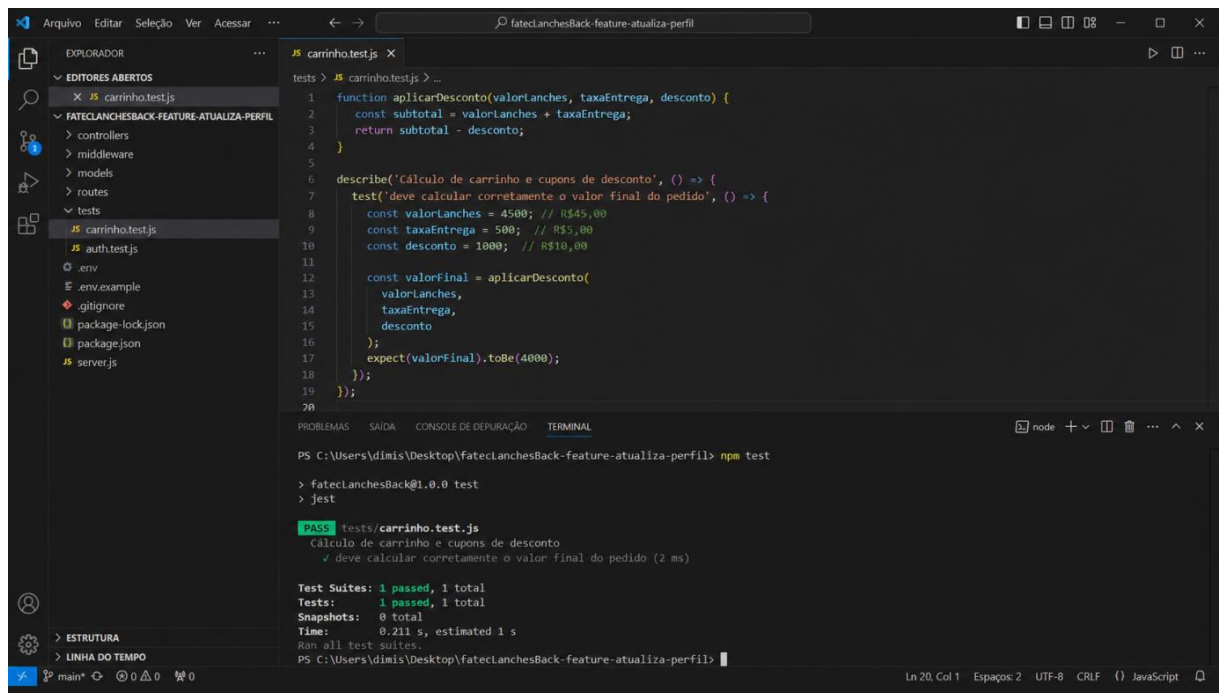
2.4.1 Testes unitários

Os testes unitários foram aplicados com o objetivo de isolar e verificar a menor unidade de código testável do sistema — como funções utilitárias, regras de validação e componentes isolados —, garantindo que sua lógica interna funcionasse estritamente conforme o esperado antes da integração global do software.

No back-end da aplicação (desenvolvido em Node.js com JavaScript), utilizou-se o framework Jest para validar as regras de negócio de forma isolada. Um exemplo prático de aplicação real foi o teste do módulo de cálculo de carrinho de compras e cupons de desconto. O teste foi estruturado injetando dados fictícios (mockados) de um pedido (contendo o valor dos lanches e a taxa de entrega) e aplicando uma função de desconto. O Jest validou por meio de asserções (expect) se o valor final calculado em centavos correspondia exatamente ao esperado.

No front-end (React Native), o Jest foi combinado com a biblioteca React Native Testing Library para testar a renderização de componentes de interface, como o formulário de login. O teste simulou o disparo do botão "Entrar" com o campo de e-mail vazio, verificando se a interface disparava corretamente a mensagem de erro em tela sem precisar acionar a API real.

Figura 5 – Testes no VsCode



Fonte: Victor Nery (2026, p.23)

2.4.2 Testes de integração

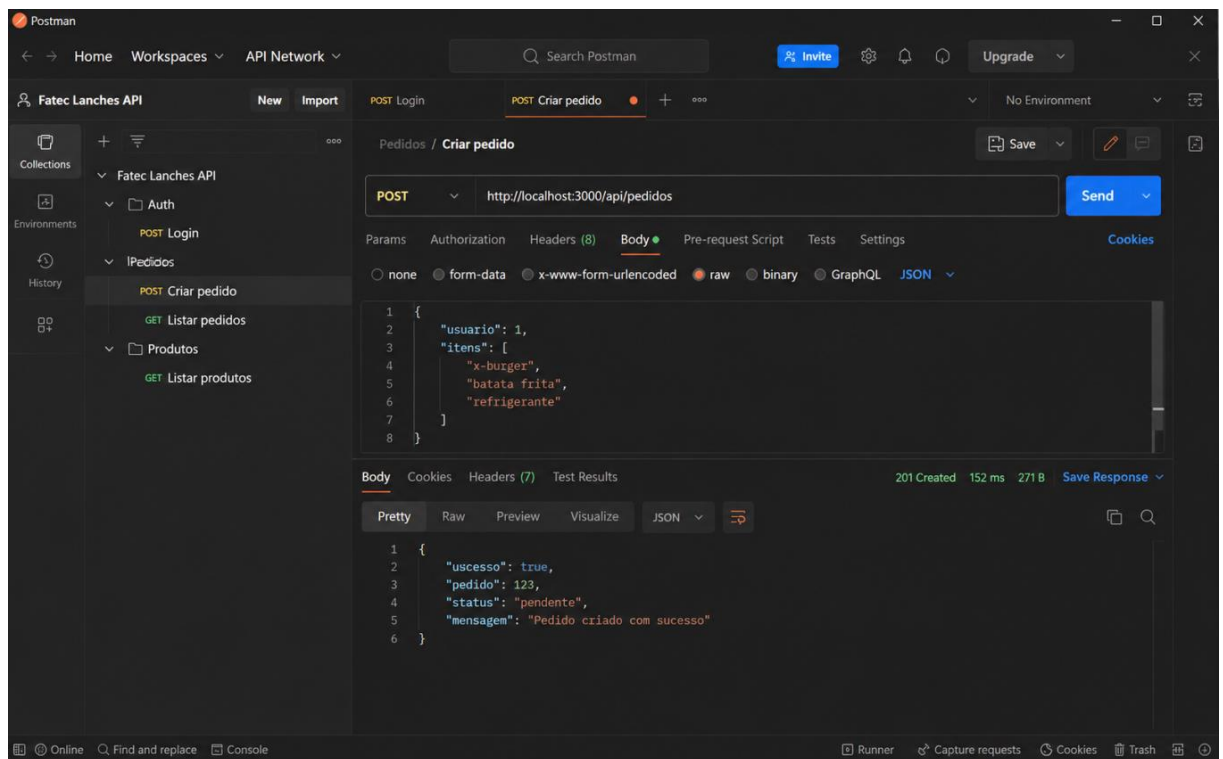
Após a validação isolada das unidades, os testes de integração foram realizados para garantir que a comunicação, o fluxo de dados e os contratos de interface entre os diferentes módulos do sistema (como a comunicação do front-end com a API e a persistência no banco de dados) ocorressem de maneira correta.

Para validar as rotas da API, utilizou-se o Postman. O fluxo real testado cobriu o cenário completo de criação de um novo pedido pelo cliente da faculdade:

- O teste disparou uma requisição HTTP do tipo POST para a rota `/api/pedidos`, enviando no corpo (body) um JSON com o ID do usuário e a lista de itens do lanche.
- O teste de integração interceptou a resposta da API, validando se o código de status retornado foi 201 Created.
- Em seguida, o script de teste realizou uma consulta direta ao banco de dados para checar se o registro do pedido havia sido persistido corretamente com o status "pendente".

- Por fim, validou-se o fluxo de resposta enviando uma requisição GET para garantir que o front-end conseguiria recuperar aquele mesmo pedido de forma íntegra.

Figura 6 – Testes no VsCode



Fonte: Victor Nery (2026, p.24)

2.4.3 Testes de segurança

Visando mitigar vulnerabilidades e resguardar os dados cadastrais e de transações dos usuários, foram realizados testes de segurança baseados nas principais ameaças listadas no ranking internacional da OWASP (Open Web Application Security Project).

Os testes de segurança foram executados em duas frentes práticas utilizando a ferramenta OWASP ZAP configurada no ambiente de desenvolvimento local:

Varredura Automatizada (DAST): A ferramenta OWASP ZAP foi utilizada como um proxy de ataque, disparando requisições maliciosas automáticas contra as rotas expostas da API do sistema. O teste tentou injetar scripts nas rotas de cadastro

(Cross-Site Scripting - XSS) e comandos estruturados nos parâmetros de busca (SQL/NoSQL Injection) para verificar se a API tratava e sanitizava os dados de entrada adequadamente. A API bloqueou as tentativas retornando status 400 Bad Request, provando a eficácia dos middlewares de validação implantados.

Testes de Controle de Acesso: Manualmente, foram simulados ataques de quebra de autenticação. Utilizando o Postman, tentou-se disparar uma requisição de alteração de status de pedido para a rota administrativa (/api/admin/pedidos/status) utilizando o token de autenticação (JWT) pertencente a um usuário comum (perfil de aluno). O sistema barrou o acesso respondendo corretamente com o código 403 Forbidden, confirmando que a validação de permissões (RBAC) por rotas estava funcionando na prática.

2.4.4 Metodologias de desenvolvimento e qualidade

Para assegurar a qualidade do software durante seu ciclo de desenvolvimento, foram adotadas práticas inspiradas nos princípios de Test-Driven Development (TDD) e Integração Contínua (CI). Embora o projeto não tenha seguido rigorosamente todas as etapas do TDD, a equipe realizou a implementação de testes automatizados utilizando Jest antes da validação final das funcionalidades, permitindo identificar falhas precocemente. Adicionalmente, foram empregados processos de revisão de código, testes de integração e validações manuais de interface, garantindo maior confiabilidade, manutenibilidade e estabilidade ao sistema.

2.4.5 Avaliação de acessibilidade

Descrever se o sistema é utilizável por pessoas com diferentes deficiências, conforme os tópicos a seguir.

- A. Objetivo da avaliação: O objetivo central desta avaliação foi analisar a interface do aplicativo Fatec Lanches para garantir que qualquer estudante, independentemente de possuir alguma limitação visual, motora ou cognitiva, consiga utilizá-lo de forma autônoma. No contexto de uma comunidade acadêmica, democratizar o acesso à cantina por meio do aplicativo é fundamental. A intenção é validar se a jornada do usuário — desde a escolha

do lanche até a finalização do pedido — é inclusiva e livre de barreiras digitais que possam isolar ou constranger os alunos com deficiência.

- B. Ferramentas e métodos de avaliação: Por se tratar de um aplicativo mobile desenvolvido em React Native, a metodologia combinou análises técnicas estruturais com testes práticos de usabilidade. A estrutura de telas foi simulada em ambiente controlado para validação de boas práticas. Na sequência, foram realizados testes manuais diretamente em smartphones.
- C. Análise de conformidade com as WCAG 2.1: Atualmente, o Fatec Lanches atende a critérios importantes, como o suporte ao redimensionamento de fontes (ajuste de texto do sistema) e o tamanho adequado das áreas de toque nos botões principais, o que evita cliques errados. Contudo, o fluxo de fechamento do carrinho e as telas de feedback de pagamento ainda falham em cumprir integralmente o nível AA de conformidade, apresentando obstáculos severos para usuários que dependem exclusivamente do retorno por áudio.
- D. Problemas identificados e propostas de melhoria: A análise do aplicativo apontou duas falhas na interface. A primeira é o tamanho reduzido do botão do carrinho de compras, que dificulta o clique por usuários com limitações motoras; a proposta é ampliar essa área de toque para o padrão mínimo de 44 por 44 pixels. O segundo problema é o baixo contraste de cores entre os botões da tela, o que confunde a navegação de pessoas com baixa visão; a solução será redefinir os tons no estilo global do código para diferenciar claramente as ações do sistema.
- E. Discussão dos resultados: Os dados coletados batem de frente com a fundamentação teórica do design universal, que defende que a acessibilidade não deve ser um recurso opcional ou um "puxadinho" no código, mas sim parte da arquitetura central do software. Os gargalos encontrados no Fatec Lanches provam que pequenas omissões no desenvolvimento do front-end impactam diretamente a autonomia do usuário. Ajustar esses detalhes melhora a experiência não só do estudante cego, mas também de quem usa o aplicativo na correria do intervalo ou sob a luz forte do sol no pátio, reforçando o compromisso social da aplicação no ambiente universitário.

3 CONSIDERAÇÕES FINAIS

Este relatório técnico teve como objetivo ampliar o alcance da cantina da faculdade Fatec Mauá, assim como facilitar aos usuários a compra do lanche de forma com que aproveite de maneira mais propensa seu intervalo de descanso.

Nas seções a seguir, realiza-se uma síntese reflexiva sobre os resultados obtidos frente aos objetivos propostos, detalhando o desempenho do sistema, as recomendações técnicas geradas pelo processo, as limitações metodológicas enfrentadas e as propostas para a evolução contínua da plataforma.

3.1 Principais conclusões

Conseguimos medir o impacto das análises e pesquisa de desenvolvimento do trabalho nos resultados finais. Para avaliar de forma clara e objetiva, os principais pontos do desenvolvimento e descobertas dessa jornada foram organizados a seguir:

- Aspectos Positivos do Desenvolvimento: O processo de desenvolvimento trouxe aprendizados valiosos, destacando-se pelo sucesso na escolha e integração da arquitetura de software. A stack adotada mostrou-se extremamente eficaz no desenvolvimento e a aderência aos requisitos arquiteturais e às necessidades operacionais do projeto, respondendo com agilidade ao fluxo de trabalho estabelecido e garantindo que cada requisito do sistema fosse traduzido em funcionalidades reais e estáveis.
- Práticas de Teste Aplicadas: A preocupação com a qualidade e integridade foi uma questão constante na construção do projeto. As práticas de teste unitário, de integração e segurança, foram uma medida de adoção essencial, funcionando como um filtro de segurança que validou desde a lógica dos códigos até os elementos visuais da interface, o que trouxe a segurança de que a aplicação se comporta de maneira previsível e segura.
- Resultados e Eficácia do Produto: O sistema mobile do Fatec Lanches entrega valor prático ao seu público-alvo. Se destacando por sua

relevância e extrema facilidade de uso, mostrando-se uma solução real e tecnicamente viável e operacionalmente relevante para otimizar o dia a dia da cantina e dos estudantes, além de consolidar o aprendizado prático em engenharia de software ao longo do curso.

3.2 Recomendações

Para garantir um longo ciclo de vida do projeto e uma melhora contínua em seu desenvolvimento, as recomendações técnicas e metodológicas podem ser listadas da seguinte forma:

- **Otimização de Código e Manutenibilidade:** Com o intuito de facilitar a evolução do software por novos desenvolvedores, recomenda-se a aplicação de refatorações estruturadas e a adoção de padrões de projeto que tenham maior compatibilidade com o grupo ou projeto (Design Patterns). Além disso, estabelecer uma padronização arquitetural clara e utilizar ferramentas de análise de código ajudará a manter o sistema limpo, modular e de fácil manutenção ao longo do tempo.
- **Performance do Sistema e Infraestrutura:** Para garantir que a aplicação responda com agilidade mesmo com o aumento do fluxo de usuários, sugere-se o aprimoramento da performance por meio de técnicas como a indexação estratégica do banco de dados e o uso de camadas de cache para as requisições mais frequentes. Melhorias na infraestrutura de hospedagem e a otimização das rotas da API também são fundamentais para reduzir drasticamente o tempo de resposta do sistema durante os horários de pico.

3.3 Limitações do estudo

Embora os objetivos principais tenham sido atingidos, houve a presença de barreiras e restrições durante o processo de desenvolvimento que influenciaram na pesquisa e no produto, podendo ser listados como:

- Restrições de Ferramentas: A escolha do framework React Native e do banco de dados MongoDB atendeu bem ao escopo multiplataforma, mas limitou o desenvolvimento à dependência de bibliotecas de terceiros para o gerenciamento de estado e navegação. Além disso, ferramentas complementares planejadas — como a automação de relatórios via n8n e o disparo de notificações *push* em tempo real para os alunos — permaneceram apenas em fase parcial de integração ou ambiente de testes .
- Limitação de Dispositivos e Plataformas: Por se tratar de um estudo de caso aplicado à realidade da FATEC Mauá, os testes práticos de interface e usabilidade ficaram restritos a simuladores de desenvolvimento e a um grupo reduzido de smartphones físicos dos próprios autores. Não foi possível homologar o aplicativo em uma ampla variedade de dispositivos reais com diferentes resoluções de tela, nem validar o comportamento em versões antigas dos sistemas Android e iOS.
- Limitações de Escopo e Dados: As validações de performance, rotas da API e consultas ao banco de dados basearam-se em dados simulados (*mocks*) em ambiente de desenvolvimento local. Como o projeto não foi implantado em uma infraestrutura de nuvem escalável para testes de estresse, não foi possível simular centenas de acessos simultâneos de usuários reais ocorrendo ao mesmo tempo (como no pico do intervalo das aulas), limitando a análise do comportamento do sistema sob forte concorrência.

Essas limitações não comprometem a estrutura atual do aplicativo, porém evidenciam oportunidades de crescimento nessas áreas e indicam possibilidades de escalabilidade futura do projeto.

3.4 Perspectivas futuras

Para que este produto ainda esteja em forma diante das mudanças de softwares e de mercado é fundamental a análise de implementação dos seguintes requisitos:

Notificações em Tempo Real: Uma das principais melhorias planejadas é a implementação de uma arquitetura baseada em eventos ou WebSockets. Essa tecnologia será voltada especificamente para o monitoramento e as etapas de preparo do pedido, permitindo que o usuário acompanhe cada mudança de status da sua solicitação instantaneamente pelo dispositivo móvel, desde o recebimento na cozinha até o momento da retirada.

Integração com o Sistema de Cantina do Fatec Lanches: Para tornar a solução ainda mais robusta, projeta-se o acoplamento e o desenvolvimento de módulos de comunicação direta com o sistema web da cantina da Fatec. Essa integração será fundamental para automatizar de forma totalmente centralizada os fluxos financeiros, a atualização do estoque de ingredientes em tempo real e o status dos pedidos.

REFERÊNCIAS

- BALTZAN, Paige. **Sistemas de informação**. 6. ed. Porto Alegre: AMGH, 2016.
- BECK, Kent *et al.* **Manifesto para o desenvolvimento ágil de software**. [S. l.]: Agile Alliance, 2001. Disponível em: <https://agilemanifesto.org/iso/ptbr/>. Acesso em: 27 mai. 2026.
- FERREIRA, Otávio Barbosa; REIS, Victor Nery Eduardinho. **Fatec Lanches: Navegação do Cliente**. Orientadora: Luana Lourenço. Mauá: Faculdade de Tecnologia de Mauá, 2026. 18 p. Relatório Técnico-Científico (Tecnologia em Desenvolvimento de Software Multiplataforma).
- GIL, Antônio Carlos. **Como elaborar projetos de pesquisa**. 5. ed. São Paulo: Atlas, 2010.
- KOGTA, Shaurya. PUCollegiFy: a smart campus automation system. **International Journal of Computer Applications**, [S. l.], v. 175, n. 12, p. 18-23, 2020.
- KOTLER, Philip; KELLER, Kevin Lane. **Administração de marketing**. 15. ed. São Paulo: Pearson Education do Brasil, 2017.
- LAUDON, Kenneth C.; LAUDON, Jane P. **Sistemas de informação gerenciais**. 14. ed. São Paulo: Pearson Education do Brasil, 2021.
- MARTIN, Robert C. **Código limpo: habilidades práticas do software Agile**. Rio de Janeiro: Alta Books, 2009.
- MEENA, R.; PRANESHWAR, S.; HARSHIVAN, T. Next-generation smart food ordering system with real-time tracking. **Journal of Mobile Computing and Application Development**, [S. l.], v. 12, n. 1, p. 45-52, 2025.
- SEVERINO, Antônio Joaquim. **Metodologia do trabalho científico**. 24. ed. rev. e atual. São Paulo: Cortez, 2016.
- SOMMERVILLE, Ian. **Engenharia de software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.
- SONI, Akash *et al.* Advanced mobile frameworks and testing methodologies in campus applications. **IEEE Transactions on Software Engineering and Quality**, [S. l.], v. 30, n. 2, p. 112-119, 2024.
- VALENTE, Marco Tulio. **Engenharia de software moderna: princípios e práticas para o desenvolvimento de software com produtividade**. [S. l.]: Independente, 2020. Disponível em: https://engsoftmod_book.github.io/book/. Acesso em: 27 mai. 2026.

FORMULÁRIO DE IDENTIFICAÇÃO

DADOS DO RELATÓRIO TÉCNICO E/OU CIENTÍFICO			
Título e subtítulo: Fatec Lanches - Navegação do Cliente		Tipo de relatório: Relatório Técnico-Científico	
Instituição: Fatec Mauá		Data: 12/06/2026	
Curso: Desenvolvimento de Software Multiplataforma		Nota:	
Autor(es): Otávio Barbosa Ferreira, Victor Nery Eduardinho Reis			
Orientador: Luana Lourenço			
Banca avaliadora: Edilma Bindá Hungria Melo e Humberto Luiz de Toledo			
Produto desenvolvido: Aplicativo FatecLanches			
Especificações técnicas: Aplicativo Mobile desenvolvido em React Native utilizando TypeScript. Ambiente de desenvolvimento configurado no Visual Studio Code. O backend é composto por API REST em JavaScript rodando em Node.js, com banco de dados MongoDB para o armazenamento.			
Palavras-chave/Descritores: desenvolvimento móvel; interface do usuário; experiência do cliente.			
Edição 1	Nº de páginas 32	Nº do volume/parte 1	Área do conhecimento: Tecnologia da informação
Observações/notas			

Fonte: Estrutura adaptada pela autora com base na norma ABNT NBR 10719:2015.