

Campus Vilhena

Coordenação do Curso Superior em Tecnologia em Análise e  
Desenvolvimento de Sistemas

VANESSA OLIVEIRA DA SILVA

Desenvolvimento de um Web App para cardápio  
digital e gestão de pedidos em Lanchonetes

VILHENA - RO

2025

VANESSA OLIVEIRA DA SILVA

## Desenvolvimento de um Web App para cardápio digital e gestão de pedidos em Lanchonetes

Artigo entregue como Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), Campus Vilhena, como requisito parcial para obtenção do grau de Tecnólogo, junto ao Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do professor Marco Antônio Augusto de Andrade.

VILHENA - RO

2025

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO.

Silva, Vanessa Oliveira da.

Desenvolvimento de um web app para Cardápio digital e gestão de pedidos em lanchonetes / Vanessa Oliveira da Silva. - Vilhena, 2025.

23 f. : il.

Orientador(a): Prof. Me. Marco Antonio Augusto de Andrade.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Vilhena, 2025.

1. Cardápio digital. 2. Aplicação web. 3. Automação de pedidos. 4. Otimização do atendimento. I. Andrade, Marco Antonio Augusto de (orient.). II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

**Bibliotecário(a) Responsável:** Rosilene Maria do Couto Marques, CRB-11/321

VANESSA OLIVEIRA DA SILVA

## Desenvolvimento de um Web App para cardápio digital e gestão de pedidos em Lanchonetes

Artigo entregue como Trabalho de Conclusão de Curso apresentado ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), Campus Vilhena, como requisito parcial para obtenção do grau de Tecnólogo, junto ao Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas, sob a orientação do professor Marco Antônio Augusto de Andrade.

Aprovado em 18/11/2025 pela banca examinadora.

---

Prof. Gilberto Pereira da Silva

Examinador interno

---

Prof. Mestre Wesley Jhonnes Ramos Rolim

Examinador interno

---

Prof. Mestre Marco Antonio Augusto de Andrade

Orientador

## Resumo

Esse trabalho tem como foco desenvolver um web app de cardápio digital, que tem como principal objetivo modernizar o atendimento em lanchonetes automatizando o processo de anotação de pedidos. A solução consiste na realização de pedidos através dos tablets nas mesas, enviando-os automaticamente à cozinha, aumentando a agilidade e eficiência do atendimento e reduzindo possíveis falhas. O sistema foi implementado com o uso de várias tecnologias, entre elas node.js, express, react.js e next.js e também metodologias ágeis como kanban e Git Flow, além de ferramentas como figma, swagger, jest e cypress. Os resultados nos mostram que a aplicação de fato otimiza o fluxo de atendimento, reduz os erros que ocorrem no dia a dia e melhora a organização, sendo uma opção acessível para pequenos e médios estabelecimentos. O sistema mostrou-se viável, podendo incluir futuramente a opção de impressão e integração com pagamentos digitais.

**Palavras-chave:** Cardápio digital, Aplicação web, Automação de pedidos, Otimização do atendimento.

## **Abstract**

This work focuses on the development of a digital menu web application aimed at modernizing customer service in snack bars by automating the order-taking process. The proposed solution allows customers to place their orders directly through tablets located at the tables, which are then automatically sent to the kitchen, increasing efficiency and reducing potential human errors. The system was implemented using several technologies, including Node.js, Express, React.js, and Next.js, as well as agile methodologies such as Kanban and Git Flow. Additional tools like Figma, Swagger, Jest, and Cypress were also employed to support design, documentation, and testing processes. The results demonstrate that the application effectively optimizes service flow, minimizes common operational errors, and improves overall organization, making it a viable and accessible option for small and medium-sized establishments. Future enhancements may include the implementation of printing features and integration with digital payment systems.

**Keywords:** Digital menu, Web application, Order automation, Service optimization.

## **1. Introdução**

Atualmente, observa-se o aumento da influência da tecnologia em diversos setores do ramo alimentício. A transformação digital tem impactado positivamente a forma como os serviços são prestados e consumidos, promovendo maior eficiência, agilidade e qualidade no atendimento ao público. No entanto, muitos estabelecimentos, especialmente os de pequeno e médio porte, ainda apresentam limitações no uso de ferramentas tecnológicas. De acordo com a Pesquisa de Inovação Semestral (PINTEC) do IBGE (2023), embora mais de 90% das empresas médias e grandes já apresentem algum nível de digitalização, a adoção de tecnologias digitais avançadas ainda ocorre de forma lenta entre os negócios de menor porte. Entre os principais fatores que dificultam essa adoção estão os altos custos das soluções tecnológicas (70,3%), a falta de pessoal qualificado (54,6%) e o conhecimento limitado sobre as tecnologias disponíveis. Esses dados reforçam que, apesar dos avanços tecnológicos, muitas lanchonetes e estabelecimentos locais ainda encontram barreiras financeiras e técnicas para implementar sistemas automatizados, como cardápios digitais e plataformas integradas de pedidos.

Nas lanchonetes que ainda fazem o uso de métodos tradicionais, o modelo mais comum se baseia em cardápios físicos, onde os clientes fazem seus pedidos comunicando-se com o garçom e o mesmo registra manualmente em cadernetas ou bloco de anotações. Esse processo além de exigir a constante presença do garçom na mesa, não oferece praticidade e nem automação, comprometendo a eficiência tanto para os clientes quanto para a equipe do estabelecimento. O registro manual também dificulta na organização de pedidos, controle das informações e a comunicação entre as equipes de atendimento e cozinha, podendo ocasionar erros e retrabalhos.

O processo manual de anotação de pedidos é lento e suscetível a erros, podendo causar atrasos nas entregas dos pedidos a mesas, especialmente em períodos de alta demanda. Esse formato torna o acompanhamento dos pedidos mais complexo, dificultando a identificação dos que já foram preparados ou entregues. Como consequência disso, pode ocorrer esquecimentos, troca de pedidos e confusões no atendimento, afetando a experiência do cliente e produtividade da equipe. Além disso, a falta de integração de dados impede o gestor de tomar decisões estratégicas baseadas em informações reais sobre o desempenho do negócio.

Diante desse cenário, surge a necessidade de evoluir e modernizar o processo de atendimento e gerenciamento de pedidos por meio de automação. O desenvolvimento de um cardápio digital, que realize também a gestão possibilita maior controle, agilidade e confiabilidade nas operações. Essa solução digital busca otimizar o fluxo de trabalho entre clientes e funcionários, tornando a experiência mais interativa e eficiente. Com a evolução constante das tecnologias web e móveis, torna-se cada vez mais real a implementação de sistemas de fácil acesso e intuitivos, capazes de atender às demandas de estabelecimentos de diferentes portes na cidade local.

### **1.1 Objetivo geral**

Desenvolver um aplicativo web de cardápio digital para uso em tablets instalados nas mesas, permitindo que os clientes selecionem itens e enviem os pedidos diretamente

à cozinha, onde poderão ser visualizados e gerenciados pelos funcionários, automatizando o fluxo de atendimento.

## **1.2 Objetivos específicos**

- Identificar e documentar os requisitos funcionais e não funcionais necessários para o desenvolvimento do sistema;
- Elaborar as interfaces e protótipos do sistema com base nos requisitos levantados, garantindo a coerência entre o design e as funcionalidades previstas;
- Implementar a codificação das funcionalidades do sistema, integrando os módulos de front-end e back-end;
- Desenvolver e executar testes automatizados para verificar o funcionamento e a estabilidade da aplicação;
- Validar o sistema junto aos usuários finais, incluindo clientes e funcionários da lanchonete, a fim de avaliar sua usabilidade, eficiência e adequação aos objetivos propostos.

## **1.3 Justificativa**

O processo de anotar pedidos manualmente em cadernos nas lanchonetes, como o uso de cardápios físicos, aumenta o risco de erros, atrasos e perda de informações. Esse fluxo impacta negativamente a experiência do cliente e a eficiência operacional. A implementação de um cardápio digital busca modernizar esse processo, tornando o atendimento mais ágil, preciso e intuitivo, além de proporcionar maior controle à equipe do estabelecimento.

## **2. Fundamentação teórica**

Atualmente no ambiente comercial, vemos a constante evolução da tecnologia, sendo promovida em diferentes tipos de negócios. Em mercados e lojas podemos ver o uso de sistemas de vendas que já é bastante comum. Entretanto em lanchonetes, essa adoção tecnológica ainda ocorre de forma gradual. Vemos atualmente depois de um tempo, os estabelecimentos fazendo o uso da tecnologia a seu favor, como o uso de cardápios digitais acessíveis através de celulares e tablets o que contribui para melhorar a experiência do cliente e otimizar o atendimento. Conforme Rossetti e Morales (2007, p. 125), contribui para que essas empresas se mantenham competitivas no mercado.

De acordo com Villerá et al. (2024), o crescente domínio dos sistemas e aplicativos têm mudado significativamente os hábitos de consumo, principalmente no setor de alimentação. Essa mudança nos mostra o papel principal da tecnologia nas empresas do setor alimentício e na forma como os serviços são prestados. No setor alimentício, o uso de ferramentas digitais inova a forma como os clientes se relacionam com as empresas, além de incentivar a constante inovação e competitividade entre as empresas por buscarem sempre melhorar os modelos de gestão e atendimento.

Apesar dos avanços tecnológicos, muitos estabelecimentos locais ainda utilizam os processos manuais, que podem reduzir a eficiência e afetar diretamente a qualidade do atendimento. Segundo Lima et al. (2018), esse tipo de método pode causar atrasos, erros nos pedidos e demora no fechamento das contas, problemas comuns em sistemas

que ainda utilizam comandas em papel. Esses cenários nos mostram como a falta de digitalização pode prejudicar tanto a experiência do cliente quanto o desempenho do negócio, destacando a importância do uso de tecnologias que automatizam tarefas e diminuem as falhas recorrentes.

No contexto brasileiro, estudos como os de Costa (2022) demonstram que cardápios digitais acessados via QR Code influenciam positivamente a percepção dos consumidores sobre eficiência e autonomia no processo de compra. Já pesquisas de Curvello et al. (2023) avaliam sistemas de cardápio digital com envio automático de pedidos à cozinha, confirmando melhorias no fluxo de atendimento e redução de erros operacionais. Esses resultados reforçam que soluções digitais no setor alimentício podem gerar impactos diretos na produtividade e na experiência do usuário.

De acordo com Silva (2024), o uso de softwares é essencial para melhorar os processos e aumentar a eficiência em diferentes áreas, principalmente em serviços que precisam de agilidade e precisão para garantir a satisfação dos clientes. Desse modo, a adoção de sistemas digitais em estabelecimentos de setor alimentícios aparece como uma forma de mostrar as novas demandas do mercado, ajudando melhor no processo, reduzindo os erros e melhorando a qualidade do atendimento. Sendo assim a tecnologia deixa de ser apenas um diferencial e passa a ser um fator necessário para a competitividade e sucesso de vários tipos de negócios no ramo alimentício.

Partindo para parte de metodologias utilizadas para desenvolvimento desta aplicação, foram utilizadas metodologias que auxiliam na organização de todo o processo de implementação. A metodologia kanban foi adotada para fazer o gerenciamento de tarefas, permitindo o acompanhamento visual do progresso de cada tarefa das etapas do processo. Essa metodologia contribui para uma melhor divisão de atividades, garantindo maior produtividade e clareza no desenvolvimento. Outra metodologia utilizada na área de Devops, foi o Git Flow, que possibilitou a criação e o controle de *branches* específicas a partir das tarefas definidas, favorecendo uma contribuição contínua no código. O uso dessas metodologias foi essencial para manter a estrutura do projeto, ágil e alinhado às boas práticas de desenvolvimento de software.

### **3. Metodologia**

Esse produto proposto foi conduzido inicialmente através do processo de boas práticas de metodologias ágeis, para maior controle e organização das atividades no decorrer da implementação. Para esse projeto foi utilizada a metodologia ágil Kanban, que tem como princípio a gestão visual do fluxo do trabalho. Segundo Anderson (2010), o Kanban é um método de gestão que busca promover mudanças evolutivas no processo de trabalho, por meio da visualização das tarefas, da limitação do trabalho em progresso e da melhoria contínua do fluxo de entrega de valor. Sendo possível através dela, organizar tarefas de forma clara e acompanhar o andamento do projeto. No gitlab as tarefas foram distribuídas em colunas, representando os estágios do processo a fazer, fazendo e concluído, possibilitando o monitoramento do progresso da implementação do produto e identificar possíveis problemas durante essa implementação. Essa

metodologia proporcionou maior controle sobre as entregas e facilitou a gestão de tempo, principalmente em um projeto desenvolvido individualmente.

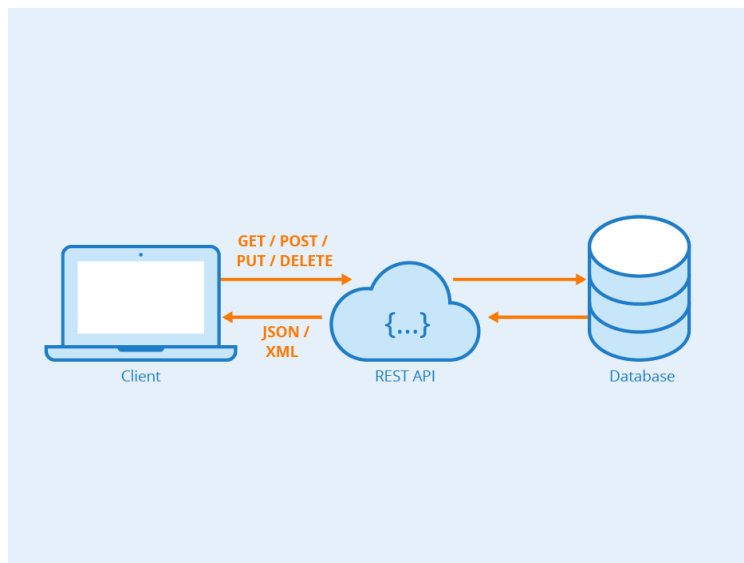
Para planejamento dos protótipos das telas foi utilizado a ferramenta online de design de interface e prototipagem, o Figma. Com ele, é possível criar protótipos interativos, definir fluxos de navegação pelo usuário, padronizar componentes que podem se tornar reutilizáveis e aplicar sistemas de design, o que ajuda a manter a consistência visual e agiliza o processo de criação das interfaces. Além disso, o Figma facilita a na implementação do código front-end, não só para servir como base, mas para exportar informações técnicas entre design e código, como cores, ícones e tipografias, garantindo que a interface final fique fiel ao protótipo inicialmente desenvolvido. E na Figura 1 é possível visualizar o protótipo criado no Figma, que serviu como base para o desenvolvimento do front-end e demonstra o layout de login e cardápio.

A Interface de Programação de Aplicações (API, do inglês Application Programming Interface) foi desenvolvida em JavaScript utilizando a tecnologia Node.js, em conjunto com o framework Express, que oferece uma estrutura leve, modular e flexível para a criação de servidores e rotas HTTP (Hypertext Transfer Protocol). Essa combinação possibilitou o desenvolvimento de um backend escalável e de fácil manutenção, com rotas organizadas e suporte eficiente para o tratamento de requisições e respostas. O uso do Express também facilitou a implementação de middlewares, responsáveis por funções como autenticação, tratamento de erros e controle de acesso.

Para o armazenamento e gerenciamento dos dados da aplicação, foi utilizado o banco de dados MongoDB, um banco de dados NoSQL (Not Only Structured Query Language) amplamente adotado em sistemas modernos pela sua flexibilidade e escalabilidade. O MongoDB armazena as informações em documentos JSON (JavaScript Object Notation), o que facilita a integração com aplicações desenvolvidas em JavaScript, como a API deste projeto. Essa estrutura orientada a documentos permite uma modelagem de dados mais dinâmica, ideal para aplicações web que necessitam de constantes atualizações e expansão de funcionalidades. A Figura 1 apresenta de forma ilustrativa o fluxo de comunicação entre o cliente, a API e o banco de dados MongoDB, demonstrando como as requisições são realizadas e os dados trafegam entre as camadas do sistema.

Além disso, foi utilizada a biblioteca Mongoose como ferramenta de mapeamento objeto-documento (ODM), que atua como intermediária entre o código JavaScript e o banco de dados. Com o Mongoose, foi possível definir esquemas para as coleções, estabelecendo regras de estrutura e validação dos dados, o que contribuiu para maior consistência e segurança nas operações realizadas. Essa integração garantiu uma comunicação fluida entre o back-end e o banco de dados, além de simplificar tarefas como criação, leitura, atualização e exclusão de registros, otimizando a eficiência geral da aplicação.

Figura 1 - Estrutura de comunicação entre cliente, API e banco de dados MongoDB



Fonte: DIO (2023)

No processo de desenvolvimento da aplicação, foi utilizada a biblioteca Zod para a validação dos dados de entrada na API. Essa ferramenta foi essencial para garantir que todas as informações enviadas pelos usuários atendessem aos padrões definidos pelo sistema, evitando inconsistências e falhas durante a execução das funcionalidades. Com o Zod, foi possível criar esquemas de validação que verificam automaticamente o tipo, formato e obrigatoriedade dos campos, assegurando maior segurança, confiabilidade e integridade dos dados processados pela aplicação. Essa prática contribuiu diretamente para a robustez do sistema e para a manutenção de uma estrutura de código mais organizada e padronizada.

Para a documentação da API, foi utilizada a ferramenta Swagger UI, que permite a criação de uma documentação interativa e dinâmica das rotas e endpoints do sistema. Essa ferramenta facilita o entendimento e a utilização da API, possibilitando que desenvolvedores e testadores visualizem, testem e validem as requisições HTTP diretamente pelo navegador, sem a necessidade de ferramentas externas. Além disso, o Swagger contribuiu para a padronização e clareza do código, tornando o processo de manutenção e integração mais eficiente ao longo do ciclo de desenvolvimento do projeto.

E com objetivo de assegurar um bom funcionamento do sistema, foram realizados testes automatizados na API, foi utilizado a biblioteca jest, framework amplamente usado em JavaScript para testes unitários e integração. O jest possibilita a criação de testes unitários, responsáveis por validar o funcionamento de funções específicas, e testes de integração que verificam a comunicação entre as diferentes camadas do sistema. Essa prática permitiu detectar e corrigir possíveis falhas durante o processo de desenvolvimento, garantindo estabilidade e segurança no código final.

O front-end foi desenvolvido utilizando a biblioteca JavaScript React.js, que é voltada para a criação de interfaces dinâmicas e interativas para o usuário, em conjunto com o Next.js, um framework que oferece recursos adicionais ao React como

renderização do lado do servidor e melhor gerenciamento de rotas. Possibilitando fácil manutenção de código e seguindo boas práticas de desenvolvimento para uma experiência agradável e intuitiva ao usuário. O design foi implementado tendo como base o protótipo desenvolvido no Figma, mantendo-se fiel à primeira vista, mas tendo mudanças de forma positiva por conta de componentes já prontos que as bibliotecas oferecem. Entre elas temos o shadcn, que oferece componentes com designs mais limpo e elegante.

Para validar o fluxo das telas do front-end, foram implementados os testes automatizados de front-end, foi utilizado a tecnologia para desenvolvimento de testes cypress. O Cypress é um framework de código aberto que possibilita escrever testes automatizados de maneira fácil, desenvolvidos em JavaScript, permitindo verificar o funcionamento da aplicação de forma prática e eficiente. Essa tecnologia facilita identificar erros e garante que o fluxo de operação das funcionalidades funcionem como o esperado.

Durante o desenvolvimento, foi adotado o Git para o controle de versão do código, seguindo o fluxo de trabalho GitFlow. Esse modelo define a criação de *branches* específicas para cada tarefa ou funcionalidade, o que permitiu manter o código organizado e facilitar o reconhecimento de alterações. Essa prática garante maior segurança nas atualizações e contribui para um desenvolvimento mais estruturado.

Além do controle de versão, o GitLab foi utilizado como principal plataforma de hospedagem e gerenciamento dos repositórios do projeto, tanto do front-end quanto do *back-end*. A ferramenta possibilitou o armazenamento seguro do código-fonte e facilitou o versionamento das alterações realizadas ao longo do desenvolvimento. O GitLab também foi fundamental no acompanhamento das tarefas e no controle de *branches* criadas a partir das demandas definidas no Kanban, permitindo uma organização mais clara entre as etapas de implementação. Além dessas funcionalidades, a plataforma permitiu a configuração de um pipeline de integração contínua (CI) e entrega contínua (CD), responsável por automatizar etapas como build, execução de testes e deploy das aplicações. Essa abordagem contribuiu para um fluxo de desenvolvimento mais padronizado, eficiente e alinhado às práticas modernas de engenharia de software.

Por fim, o deploy das aplicações tanto do front-end como da API foi realizado de forma separada. O front-end foi hospedado em um serviço de nuvem com suporte a aplicações Next.js, possibilitando o acesso via navegador de forma estável e responsiva. Já o back-end foi implantado em um ambiente Node.js configurado em servidor online, garantindo a disponibilidade da API para consumo pelas interfaces do sistema. Em ambos os casos, foram configuradas variáveis de ambiente e rotas seguras, assegurando o funcionamento contínuo e a comunicação correta entre as partes da aplicação.

#### **4. Desenvolvimento**

Para o desenvolvimento da aplicação, foram identificados e documentados os requisitos funcionais e não funcionais, os quais descrevem as funcionalidades que o sistema deverá oferecer. No fluxo do sistema ao realizar login como cliente ele terá acesso a visualização do cardápio digital, adicionar itens ao carrinho de compras, enviar pedidos à cozinha e acompanhar o status do pedido em tempo real. Ao logar como

administrador, que é um acesso aos funcionários do estabelecimento, além do acesso às funcionalidades dos clientes, o sistema possibilita o gerenciamento de produtos, pedidos, usuários e de visualização de relatórios.

De início foi desenvolvido o protótipo das telas do sistema no Figma, o que serviu como base visual e referencial para o desenvolvimento do front-end, seguindo os requisitos definidos anteriormente. O que ajudou muito na construção do front-end foi ter um modelo do que fazer em cada tela. Sem contar com o suporte de informações técnicas, como cores, ícones e tipografias que auxiliou muito na hora de codificar as do front-end.

Foi desenvolvida uma API responsável pelo processamento das funcionalidades do sistema, estruturada segundo uma arquitetura em camadas, composta por controller, service e repository. Os controllers recebem as requisições HTTP, interpretam os parâmetros de entrada e encaminham os dados aos services por meio de DTOs (Data Transfer Objects), que padronizam a entrada e a saída das informações. Nos services são aplicadas as regras de negócio, bem como validações adicionais, utilização de mappers para conversão entre modelos de domínio e DTOs, além da chamada aos *repositories*, responsáveis pela interação direta com o banco de dados.

A API também utiliza middlewares para tratamento de autenticação e autorização, validação de dados, registro de logs e gestão de erros. O padrão de respostas segue contratos REST, incluindo códigos HTTP adequados (200, 201, 400, 404, 409, 500), mensagens padronizadas de erro e sucesso, e implementação de práticas como idempotência em rotas sensíveis e paginação em consultas que retornam múltiplos registros. Após o processamento, o retorno é enviado ao cliente no formato JSON, mantendo consistência e clareza na comunicação entre as camadas do sistema.

Inicialmente foi feita a estrutura inicial do projeto, criando pastas das camadas e os arquivos das rotas necessárias. Conforme as rotas foram implementadas, foi feita também a criação do seed para ajudar nos testes manuais das rotas ao buscar, editar ou excluir dados. O desenvolvimento da API, teve a implementação dos seguintes requisitos funcionais citados na tabela 1 abaixo:

Tabela 1 - Requisitos funcionais do sistema

<b>Administrador</b>	<b>Cliente</b>
RF1 - Gerenciamento de produto/cardápio	RF5 - Visualizar cardápio
RF2 - Gerenciamento de pedidos	RF6 - Adicionar pedido ao carrinho
RF3 - Gerenciamento de usuários	RF7 - Enviar pedido do carrinho
RF4 - Relatório de vendas	RF8 - Excluir pedido do carrinho

**Fonte:** Elaborado pelo autor (2025).

Para o armazenamento e gerenciamento de dados, foi utilizada a tecnologia MongoDB, um banco de dados NoSQL orientado a documentos. A escolha por essa ferramenta se deu por sua flexibilidade, escalabilidade e fácil integração com aplicações JavaScript, o que a torna ideal para o desenvolvimento de APIs modernas. Foi feita a conexão logo após a criação da estrutura inicial do projeto para facilitar no teste de rotas com um banco já populado com o seed.

Foi implementada a validação dos dados enviados pelos usuários à aplicação, utilizando a biblioteca Zod. No processo de autenticação e cadastro de produtos, o Zod verifica se os campos obrigatórios foram preenchidos corretamente, se os tipos de dados estão adequados e se os valores seguem as restrições impostas pela aplicação. Essa abordagem contribui para evitar erros de entrada e garantir a integridade das informações armazenadas no banco de dados, tornando o sistema mais confiável e robusto.

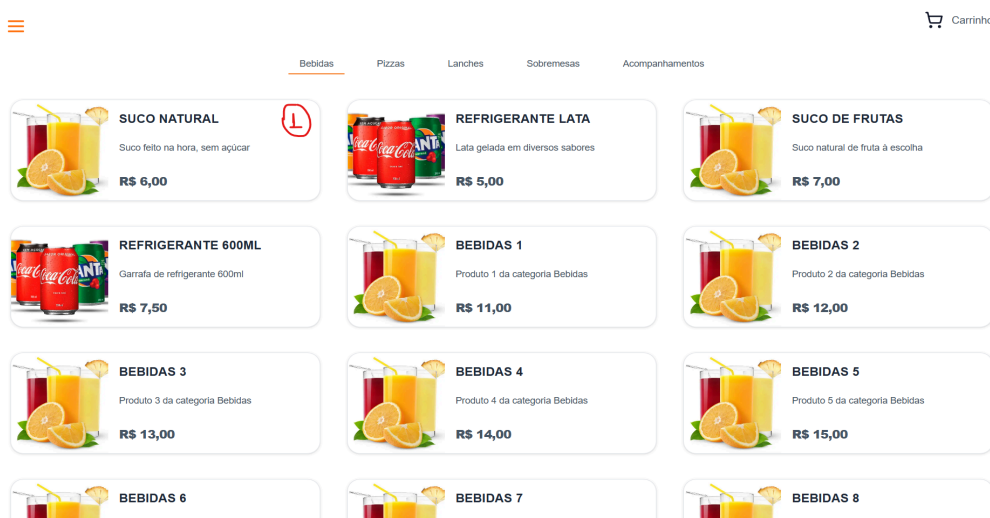
Após a implementação da API, logo em seguida foi desenvolvida a documentação utilizando a ferramenta Swagger, ferramenta muito utilizada para documentar e consumir APIs. Essa documentação foi desenvolvida manualmente conforme as rotas foram sendo implementadas e ela tem como objetivo facilitar no desenvolvimento do front-end e manter sempre registrado as rotas disponíveis para a aplicação. E por esse motivo a documentação swagger é fundamental para a construção das interfaces front-end. Pois é nessa documentação que é possível visualizar de forma mais clara campos de entrada e saída de cada rota. Assim como também mostra as regras de negócio e as respostas que podem ser retornadas para situação de sucesso ou falha. Essa documentação tornou mais eficiente o desenvolvimento do front-end e também a integração com o back-end.

Em seguida foram realizados os testes na API, sendo feitos os testes de unidades e integração para garantir que toda a API do sistema estava funcionando de acordo com o esperado. De início foram implementados os testes de unidades que são responsáveis por testar pequenas partes do código, sendo eles funções, métodos e etc. E depois foram desenvolvidos também, os testes de integração que tem como função testar a comunicação entre as camadas, para saber se a requisição chega no controller, passa pela validação de regra de negócio no service e faz a comunicação com o banco de dados através do repository. E para esse fim, foi utilizado a biblioteca de testes Jest, um framework de teste em JavaScript que pode ser desenvolvido em aplicações Node.js.

Por fim foi desenvolvido a parte final dos sistema, o front-end. Responsável pela interface visual que o usuário irá visualizar quando tiver acesso ao sistema. Essa parte de desenvolver as interfaces do sistemas, o objetivo era proporcionar uma experiência intuitiva e funcional para garantir a usabilidade por parte do usuário e a eficiência de navegação entre as telas, assim como começamos a ver a partir das seguintes figuras que serão mostradas e explicadas a partir delas o fluxo principal do sistema..

Na figura 2 vemos a tela inicial do sistema que é a visualização do cardápio digital, onde o cliente pode clicar em um card do produto como é mostrado no item 1 e selecioná-lo para ser enviado para o carrinho como será mostrado na figura 3.

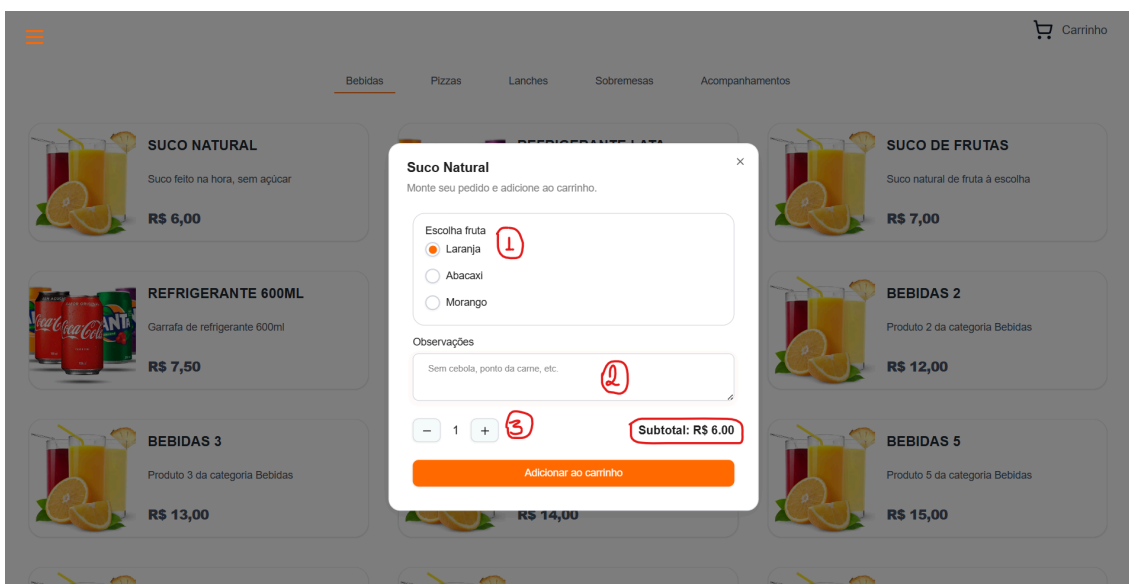
Figura 2 - Tela de cardápio



Fonte: Elaborado pelo autor (2025).

Na figura 3 mostra o formulário aberto ao clicar em um card do produto, então o cliente dependendo do produto terá que escolher uma opção obrigatória conforme indicado no item 1 e poderá adicionar também uma observação ou alterar quantidade conforme mostrado no item 2 e 3. O cliente poderá visualizar o subtotal do pedido e se estiver de acordo, poderá clicar no botão de adicionar pedido ao carrinho.

Figura 3 - Tela com formulário para enviar pedido ao carrinho

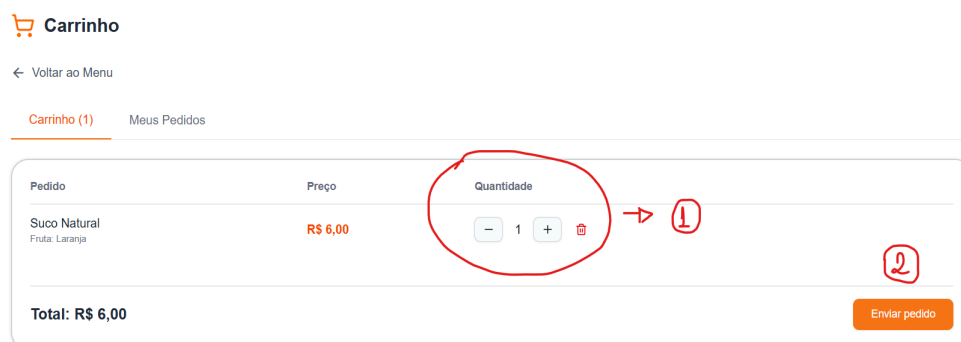


Fonte: Elaborado pelo autor (2025).

Já na Figura 4, observa-se a tela do carrinho, onde o cliente pode visualizar os produtos adicionados, podendo alterar a quantidade de cada item ou removê-los, conforme indicado no item 1. Caso todas as escolhas estejam corretas, o cliente poderá

clicar no botão “Enviar pedido”, como mostrado no item 2, para que as informações sejam encaminhadas à cozinha e registradas como um pedido no sistema.

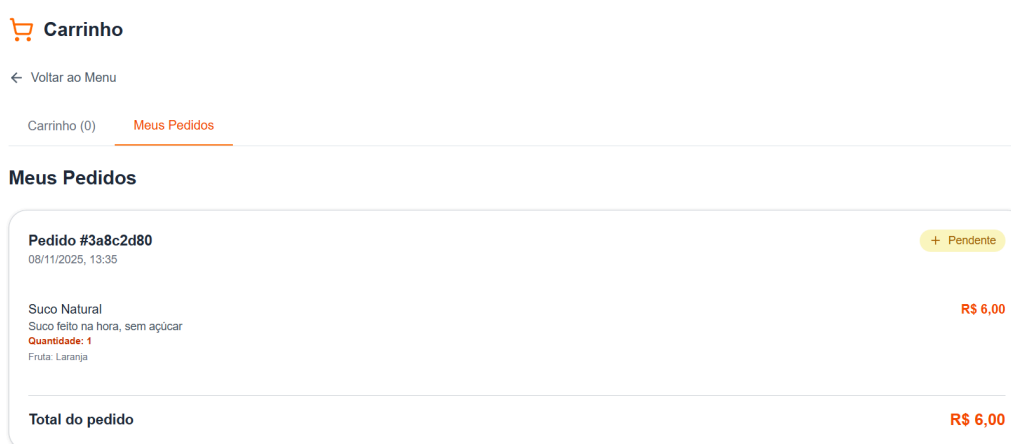
Figura 4 - Tela de carrinho



**Fonte:** Elaborado pelo autor (2025).

Depois que o pedido é enviado, o cliente pode visualizar as informações detalhadas de seus pedidos como produto, descrição, observação (caso tenha inserido), valor total e status na aba “Meus pedidos”, localizada na tela do carrinho, como é mostrado na Figura 5. Inicialmente, o pedido aparece com o status “pendente”, indicando que está em preparo. Assim que a cozinha finaliza o preparo, o status é atualizado para “pronto” por um funcionário, informando que o pedido está aguardando a entrega na mesa pelo garçom. Dessa forma, encerra-se o fluxo de uso do sistema pelo cliente.

Figura 5 - Tela de carrinho(Meus pedidos)



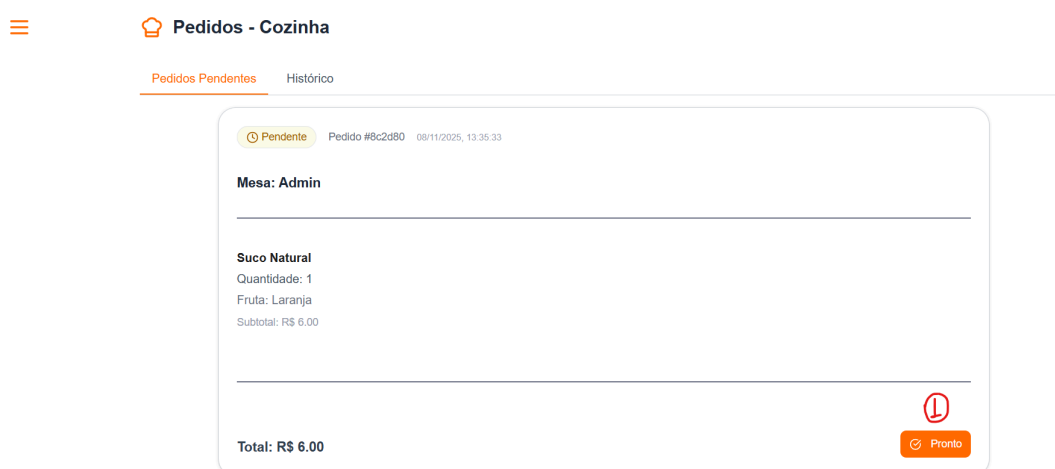
**Fonte:** Elaborado pelo autor (2025).

No fluxo do usuário administrador, é possível realizar o gerenciamento dos pedidos tanto pela equipe da cozinha quanto pelos garçons. A Figura 6 mostra como os funcionários podem acompanhar e controlar esses pedidos de forma mais organizada.

Nessa tela, são exibidos os cards com informações dos pedidos área da cozinha, onde os funcionários podem visualizar, atualizar o status e administrar cada solicitação, garantindo um maior controle e eficiência no processo de preparo e entrega.

Após o envio do pedido pelo cliente a partir do carrinho, ele é exibido na tela de pedidos da cozinha, onde será preparado pelos funcionários. Quando o preparo é finalizado, o funcionário deve clicar no botão “Pronto” e confirmará a alteração dos status desse pedido através de pop-up de confirmação, como mostrado no item 1 da Figura 6, para que o pedido seja direcionado à tela dos garçons e o cliente seja informado de que o pedido está pronto para ser entregue à mesa.

Figura 6 - Tela de gerenciamento de pedidos da cozinha

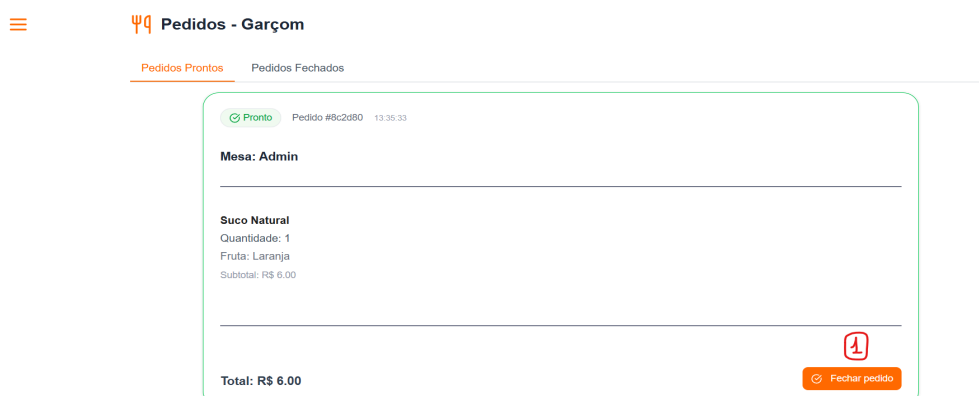


**Fonte:** Elaborado pelo autor (2025).

Após o pedido ser marcado como “pronto”, ele passa a ser exibido na tela do garçom, como mostrado na Figura 7. Nesse momento, o garçom realiza a entrega do pedido à mesa e, após o cliente finalizar o consumo e efetuar o pagamento, o garçom deve fechar o pedido, clicando no botão “Fechar pedido”, conforme indicado no item 1 da Figura 7. Ao realizar essa ação, o pedido é removido da aba “Meus pedidos” na tela de carrinho, garantindo que os próximos clientes da mesma mesa não visualizem os pedidos anteriores. Dessa forma, encerra-se o fluxo de uso do sistema pelo funcionário.

O sistema conta também com funcionalidades para mostrar relatório de vendas por dia através de gráficos e uma tabela de pedidos já fechados contabilizando o total já vendido de todos esses pedidos. Assim como também gerenciamento de produtos caso haja uma nova receita e o gerenciamento de usuários sendo eles mesas do estabelecimento, caso o estabelecimento mude de localidade e aumente a quantidade de mesas para suportar maior número de clientes. Porém tem que ter em mente que teria que comprar novos tablets, pois se tem 10 mesas terá que ter 10 tablets, 1 tablet para cada mesa.

Figura 7 - Tela de gerenciamento de pedidos dos garçons



**Fonte:** Elaborado pelo autor (2025).

Na etapa final do desenvolvimento, foi realizado o deploy de ambas as partes do sistema back-end e front-end. O back-end foi configurado para rodar em um servidor compatível com aplicações JavaScript. Onde foi necessário a definição das variáveis de ambiente, as conexões com o banco de dados e o ajuste das rotas da API para garantir o acesso público de forma segura e estável.

Já o front-end, passou pelo processo de build para gerar os arquivos otimizados de produção, sendo hospedado no mesmo servidor que o back-end. Antes de concluir o deploy, foram realizados testes de integração entre os dois lados do sistema, assegurando que as requisições do front-end e as respostas do back-end se comunicassem corretamente. Esse conjunto de etapas garantiu que a aplicação operasse de forma consistente no ambiente online.

## 5. Resultados e discussões

O desenvolvimento do sistema teve como objetivo principal a criação de um cardápio digital interativo. O projeto previu a integração com uma API responsável pelo gerenciamento de pedidos, com a finalidade de otimizar o atendimento em lanchonetes. Durante o processo de implementação, praticamente todas as funcionalidades especificadas nos requisitos iniciais foram concluídas, exceto o módulo de impressão de pedidos, cuja implementação não foi possível, principal limitação sendo o tempo. A aplicação foi concebida para atender a dois perfis de usuários: clientes e administradores, garantindo uma experiência eficiente, organizada e, sobretudo, intuitiva. Todo o desenvolvimento foi guiado pelos princípios de usabilidade e desempenho, assegurando que o sistema atendesse às demandas práticas do ambiente comercial.

### 5.1 Funcionalidade entregues

No ponto de vista do usuário como cliente, o sistema foi desenvolvido para oferecer uma experiência mais simples, intuitiva e automatizada também. O cardápio digital exibe os produtos disponíveis, contendo nome, descrição e preço, permitindo que o cliente navegue entre categorias, selecionando os itens desejados e adicionando-os ao

carrinho. E dependendo do item desejado, ele tem que selecionar uma opção obrigatória e poderá ter também itens opcionais. Após todos os itens desejados estiverem no carrinho, ele pode ser enviado diretamente para a cozinha, e o usuário poderá acompanhar o status do pedido em tempo real, desde quando estiver pendente até estiver pronto.

Esse fluxo elimina o processo tradicional de atendimento com anotações de pedidos manuais e diminui o tempo de espera. O uso dos tablets nas mesas torna a interação mais moderna e prática, além de otimizar o trabalho dos funcionários. A automação desse novo processo, traz eficiência e melhora a experiência do cliente, demonstrando o potencial da tecnologia para mudar significativamente o setor alimentício.

Ao logar como usuário administrador, ele terá acesso às mesmas funcionalidades que o cliente, sendo eles visualizar o cardápio digital e poder fazer um pedido. Mas também terá permissões adicionais para o gerenciamento do sistema. Dentre as funcionalidades estão: gerenciamento de produtos, usuários, pedidos e visualização de relatório. No que envolve gerenciamento de produtos, o administrador pode cadastrar novos produtos, atualizar informações como preços e remover produtos indisponíveis, mantendo sempre o cardápio atualizado. Em pedidos, é possível alterar o status dos pedidos “pendente” para “pronto” e depois “fechado”, facilitando o controle desses pedidos e o fluxo de comunicação entre a cozinha e o atendimento.

E no módulo de relatórios oferece uma visão geral sobre as vendas realizadas, possibilitando o acompanhamento do desempenho do negócio e auxiliando na tomada de decisões estratégicas. Essas funcionalidades foram desenvolvidas na API e, posteriormente, integradas às telas do front-end. A única funcionalidade não implementada foi a impressão de pedidos, devido à ausência de infraestrutura física para testes com impressoras térmicas. No entanto, o sistema foi projetado para comportar essa funcionalidade futuramente.

## **5.2 Teste e validação**

A etapa de testes foi fundamental para garantir o funcionamento correto do sistema e a confiabilidade das funcionalidades implementadas. Foram realizados testes de unidade e testes de integração na API, além de testes automatizados de interface no front-end.

Os testes de unidade tiveram como objetivo validar o comportamento de funções isoladas, como rotas de cadastro, autenticação, atualização e listagem. Essa abordagem permitiu identificar falhas específicas e garantir que cada parte do código executasse sua função de forma adequada. Com isso, foi possível evitar erros de lógica e assegurar maior estabilidade ao sistema.

Os testes de integração verificaram a comunicação entre os módulos e o banco de dados, confirmando que o fluxo de dados estava sendo executado corretamente entre as camadas do sistema. Essa fase foi essencial para validar a interação entre o front-end e a API, especialmente nos processos de envio e atualização de pedidos. Dessa forma, garantiu-se que as informações trafegam corretamente entre cliente, servidor e banco de dados, mantendo a coerência das operações.

No front-end, os testes automatizados foram implementados utilizando o framework Cypress, amplamente adotado no desenvolvimento web. O Cypress permitiu simular ações de usuários reais, como a navegação pelo cardápio, adição de produtos ao carrinho e acompanhamento de pedidos. Essa automação facilitou a detecção de falhas e assegurou que os elementos de interface respondessem adequadamente às interações do usuário. Além disso, a reexecução rápida dos testes após modificações no código garantiu agilidade e confiabilidade durante o processo de validação.

### **5.3 Deploy e integração**

Durante o desenvolvimento do sistema, foi configurado um processo de CI/CD utilizando o GitLab, com o objetivo de automatizar as etapas de teste, construção e publicação da API. Essa configuração permitiu que cada atualização do código fosse testada e implantada automaticamente, diminuindo erros manuais e tornando o processo de desenvolvimento mais ágil e seguro

Na primeira etapa do pipeline, foram executados os testes automatizados, responsáveis por verificar se todas as funcionalidades da API estavam funcionando corretamente. Essa prática ajudou a identificar e corrigir possíveis erros antes que novas versões fossem publicadas, garantindo mais estabilidade para o sistema. Em seguida, a aplicação passou pela etapa de build, na qual foi criada uma imagem Docker e enviada para o GitLab Registry, servindo como base para futuras implantações. Esse procedimento facilitou a organização das versões do sistema e garantiu que o ambiente de produção fosse igual ao ambiente de desenvolvimento.

Por fim, o deploy foi feito de forma automática por meio do Kubernetes, que executou a publicação da API no servidor. O processo incluiu a substituição de versões antigas, a configuração de variáveis de ambiente e a integração com o banco de dados MongoDB Atlas, hospedado em nuvem. Com essa automação, o projeto obteve resultados positivos, como mais rapidez nas entregas, maior estabilidade do sistema e menos falhas durante as atualizações. O uso do pipeline de CI/CD também tornou o deploy mais seguro e garantiu que a aplicação permanecesse disponível online de forma contínua.

### **5.4 Comparação com Expectativas e Outras Soluções**

Os resultados obtidos demonstraram que o sistema atendeu às expectativas estabelecidas no início do projeto. O cardápio digital mostrou-se eficiente em reduzir etapas manuais, agilizar o atendimento e oferecer uma experiência mais interativa ao cliente. Mesmo em ambiente de testes, foi possível observar uma diminuição significativa no tempo de atendimento e na ocorrência de erros de registro de pedidos.

Comparado a soluções comerciais existentes no mercado, o sistema desenvolvido apresenta simplicidade, baixo custo de implantação e maior adaptabilidade às necessidades de pequenos estabelecimentos. Enquanto muitas plataformas similares exigem assinaturas mensais, infraestrutura complexa ou dependência de equipamentos específicos, esta proposta oferece uma solução mais leve e acessível, podendo ser executada em dispositivos comuns, como tablets e computadores.

Outro diferencial é a autonomia concedida ao cliente, que pode realizar o pedido sem depender da intermediação de atendentes. Essa característica reflete uma tendência crescente de digitalização no setor alimentício, promovendo praticidade, transparência e economia de recursos humanos. O acompanhamento em tempo real também reforça a sensação de controle e confiança, aumentando a satisfação do usuário final.

No entanto, ainda existem pontos a serem aprimorados. A ausência da funcionalidade de impressão limita a utilização do sistema em locais com grande volume de pedidos, onde o controle físico ainda é necessário. Além disso, futuras melhorias podem incluir a integração com métodos de pagamento digital, autenticação mais robusta e personalização de relatórios.

## **5.5 Discussão Geral dos Resultados**

De modo geral, o sistema alcançou os objetivos propostos e demonstrou viabilidade para pequenas lanchonetes que buscam modernizar o processo de atendimento. A implementação das funcionalidades principais, aliada ao uso de testes automatizados, resultou em maior confiabilidade e estabilidade do sistema. Os testes executados pelo framework Jest atingiram 90.17% de cobertura de linhas e 81.76% de cobertura de branches, validando os fluxos críticos da aplicação e reduzindo falhas durante o processo de integração e deploy. A Figura 8 apresenta os indicadores de cobertura obtidos.

O projeto também mostrou a importância da automação no ambiente comercial, especialmente em setores que demandam agilidade e eficiência. A substituição de processos manuais por soluções digitais reduz erros, otimiza o tempo de atendimento e melhora a experiência do cliente, deixando-o mais modernizado. Além disso, o uso de boas práticas de desenvolvimento e integração contribuiu para um sistema mais organizado e de fácil manutenção.

Com os resultados obtidos, pode-se afirmar que o sistema está preparado para futuras expansões, como a inclusão de novas funcionalidades e melhorias na interface. Essas evoluções permitiram que a aplicação acompanhe o avanço tecnológico e se torne uma alternativa competitiva no mercado de sistemas de autoatendimento.

Figura 8 - Cobertura de testes da API

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	90.17	81.76	87.5	90.29	
src	85.71	50	0	85.71	
app.js	85.71	50	0	85.71	15-16
src/controllers	83.62	72.72	86.36	84.11	
AutenticacaoController.js	100	100	100	100	
PedidoController.js	83.33	83.33	83.33	83.33	10,82-93
ProdutoController.js	75.32	71.42	71.42	76.31	24,31,111,115,118,121-124,128-131,163-179
RelatorioController.js	100	100	100	100	
UsuarioController.js	94.11	70	100	94.11	32,87
src/docs/config	87.5	83.33	100	100	
head.js	87.5	83.33	100	100	16
src/docs/routes	100	100	100	100	
autenticacaoRoutesDoc.js	100	100	100	100	
pedidoRoutesDoc.js	100	100	100	100	
produtoRoutesDoc.js	100	100	100	100	
relatorioRoutesDoc.js	100	100	100	100	
usuarioRoutesDoc.js	100	100	100	100	
src/docs/schemas	100	100	100	100	
pedidoSchemaDoc.js	100	100	100	100	
produtoSchemaDoc.js	100	100	100	100	
relatorioSchemaDoc.js	100	100	100	100	
usuarioSchemaDoc.js	100	100	100	100	
src/middleware	88.67	73.33	100	88.23	
AuthMiddleware.js	82.35	62.5	100	82.35	14,23,38
ErrorHandler.js	95.45	83.33	100	95	3
UploadMiddleware.js	85.71	50	100	85.71	15,26
src/respositories	94.2	80.95	92	94.2	
PedidoRepository.js	100	0	100	100	14
ProdutoRepository.js	100	100	100	100	
RelatorioRepository.js	90.9	90	50	90.9	85-86
UsuarioRepository.js	93.1	50	100	93.1	12,54
src/routes	95	100	50	95	
AutenticacaoRoutes.js	100	100	100	100	
PedidoRoutes.js	100	100	100	100	

Fonte: Elaborado pelo autor (2025).

## 6. Conclusão

O propósito essencial deste projeto de cardápio digital foi renovar o atendimento em lanchonetes, trocando as comandas de papel por um sistema totalmente automático e unificado. Ao implantar o Web App, tornou-se possível oferecer uma experiência mais ágil e eficaz tanto para os clientes, que agora fazem seus pedidos sozinhos, quanto para os gestores, que ganharam mais controle e rapidez na administração dos pedidos, itens e usuários.

No decorrer da criação, aplicamos ideias técnicas valiosas, como organizar o back-end em camadas, usar API no front-end e fazer testes automáticos para assegurar a solidez do sistema. A junção das tecnologias usadas se mostrou eficaz para cumprir as exigências definidas no começo, permitindo o funcionamento suave das funções principais, como o envio e gestão de pedidos e a atualização imediata do estado.

Entre os pontos altos do projeto, notam-se a otimização no andamento dos pedidos, a diminuição de erros humanos e o ganho de produtividade no atendimento. Além disso, o sistema se ajusta facilmente a outros negócios que precisem de uma solução fácil e clara para a gestão de pedidos e produtos.

Entretanto, algumas limitações foram encontradas durante o desenvolvimento. Um exemplo foi a impossibilidade de testar, na prática, a impressão física de pedidos, já que não houve integração com impressoras reais. Também há espaço para melhorias

visuais e para aprimorar a responsividade em dispositivos diferentes, garantindo uma experiência fluida para todos os usuários.

Futuramente há muitas ideias para evoluir o sistema: impressão automática de pedidos, interfaces mais otimizadas, relatórios mais completos, autenticação avançada com login via Google e Facebook e um controle mais detalhado de permissões. Além disso, há o desejo de levar o sistema para aplicativos móveis nativos, tanto no Android quanto no iOS, ampliando ainda mais sua praticidade e alcance.

Em resumo, este projeto vai muito além de um cardápio digital: ele representa o uso da tecnologia e da automação como ferramentas que transformam e modernizam a experiência no setor alimentício, mostrando que inovação e eficiência podem caminhar juntas.

## 7. Referências

ANDERSON, David J. Kanban: mudança evolucionária de sucesso para seu negócio de tecnologia. Tradução: Rafael Helm. Blue Hole Press, 2010. ISBN 978-0984521463.

DIO. O que diabos é uma API, e pior ainda, uma API RESTful. [S. l.]: Digital Innovation One, 2023. Disponível em: <https://www.dio.me/articles/o-que-diabos-e-uma-api-e-pior-ainda-uma-api-restful>.

Acesso em: 8 nov. 2025.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA (IBGE). Pesquisa de Inovação Semestral – PINTEC Semestral 2022: Uso de tecnologias digitais avançadas nas empresas industriais. Rio de Janeiro: IBGE, 2023. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/livros/liv102071.pdf>. Acesso em: 6 nov. 2025.

ROSSETTI, Adroaldo Guimarães; MORALES, Aran Bey Tcholakian. O papel da tecnologia da informação na gestão do conhecimento. *Ciência da Informação*, Brasília, v. 36, n. 1, p. 124–135, jan./abr. 2007. Disponível em: <https://www.scielo.br/j/ci/a/FzcdzsLpNJ43cXj5RcRWg5v/?lang=pt>. Acesso em: 1 nov. 2025.

VILLERÁ, Elcivander A.; RODRIGUES, Rodrigo O.; CRUZ, Eduardo L. Bodokó Delícia's: aplicativo para gestão de pedidos e cardápio digital. In: SIMPÓSIO DE TECNOLOGIA DA FATEC JALES, 8., 2024, Jales. Anais... Jales: Faculdade de Tecnologia Professor José Camargo – FATEC Jales, 2024. ISSN 2595-2323. Disponível

em:

[https://ric.cps.sp.gov.br/bitstream/123456789/30787/1/2024\\_2\\_SI\\_elcivander\\_alves\\_villera\\_bodoko\\_delicias\\_aplicativo\\_para\\_gestao\\_de\\_pedidos.pdf](https://ric.cps.sp.gov.br/bitstream/123456789/30787/1/2024_2_SI_elcivander_alves_villera_bodoko_delicias_aplicativo_para_gestao_de_pedidos.pdf). Acesso em: 1 nov. 2025.

Costa, A. M. de M. Gagliardi, É. C. V. (2022). Comportamento do consumidor: um estudo sobre a percepção do consumidor sobre seu comportamento em restaurantes que utilizam o sistema de QR Code. Trabalho de Conclusão de Curso, UniCEUB. Disponível em: <https://repositorio.uniceub.br/jspui/bitstream/prefix/16767/1/21800932.pdf>. Acesso em: 19 nov. 2025.

Curvello, R., Custódio Maciel, M., & Cidade, P. M. (2023). Cardápio Digital. Anais da Feira do Conhecimento Tecnológico e Científico, IFC. Disponível em: <https://publicacoes.ifc.edu.br/index.php/fetec/article/view/4539/3754>. Acesso em: 19 nov. 2025.

SILVA, Bruno Cason da. Sushitech: sistema web para gerenciamento de pedidos. 2024. Trabalho de Graduação (Tecnologia em Análise e Desenvolvimento de Sistemas) – Faculdade de Tecnologia de Franca – Dr. Thomaz Novelino, Centro Paula Souza, Franca/SP. Orientador: Prof. Me. Carlos Alberto Lucas. Disponível em: [https://ric.cps.sp.gov.br/bitstream/123456789/30300/1/ads\\_2024\\_2\\_brunocasondasilva\\_sushitechsystemawebparagerenciamentodepedidos\\_CARLOS\\_LUCAS.pdf](https://ric.cps.sp.gov.br/bitstream/123456789/30300/1/ads_2024_2_brunocasondasilva_sushitechsystemawebparagerenciamentodepedidos_CARLOS_LUCAS.pdf). Acesso em: 1 nov. 2025.

LIMA, Tiago Martins de; PEDROSO, André Luís Alves; ANTUNES NETO, Joaquim M. F.; BARBIERI, Paulo Fernando. Sistema de cardápio digital para bares, restaurantes e similares. In: Programa Educativo e Social JC na Escola: Ciência Alimentando o Brasil, 2. ed. Bauru: AGB Bauru, 2018. p. 455-466. Disponível em: <https://www.agbbauru.org.br/publicacoes/Alimentando2ed/pdf/Alimentando2ed-37.pdf>. Acesso em: 1 nov. 2025.