FATEC COTIA

Desenvolvimento de Software Multiplataforma

PROJETO INTEGRADOR III – SISTEMA ABOA

Celso Sebastião
Leandro Cardoso
Jhonathan Henrique

COTIA – SP
2025

SISTEMA ABOA – DOCUMENTO A2

Documento apresentado como requisito parcial da disciplina Gestão Ágil de Projetos de Software, referente ao Projeto Integrador III.

Professor(a): Meg Lima Andrade

Autores:

Celso Sebastião

Jhonathan Henrique

Leandro Cardoso

Cotia, 2025

Sumário

1. Introdução

O Sistema Aboa é uma plataforma web destinada a fornecer recomendações personalizadas de bares e restaurantes na cidade de Cotia, considerando fatores como localização, clima, preferências do usuário, avaliações da comunidade e eventos locais. O sistema foi desenvolvido utilizando Node.js, React e MongoDB, garantindo uma arquitetura moderna, flexível e escalável.

Este documento representa a entrega A2 do Projeto Integrador III, apresentando a modelagem, metodologia, requisitos, 5W2H, casos de uso, banco de dados e demais componentes essenciais para o desenvolvimento completo do sistema.

---

2. Objetivo

O objetivo do Sistema Aboa é facilitar o processo de busca de estabelecimentos gastronômicos, oferecendo uma experiência personalizada, inteligente e contextualizada. A plataforma pretende auxiliar tanto usuários quanto restaurantes, ampliando visibilidade, fornecendo estatísticas de consumo e permitindo interação direta entre as partes.

---

3. Justificativa

Aplicativos atuais costumam apresentar informações desatualizadas, baixa personalização, interfaces complexas e ausência de integração com dados reais, como clima e eventos. O Sistema Aboa surge para solucionar tais limitações, oferecendo uma experiência simples, objetiva e inteligente.

O uso exclusivo do MongoDB justifica-se pela necessidade de armazenar informações dinâmicas, como cardápios, fotos, horários, preferências e avaliações, sem rigidez estrutural.

---

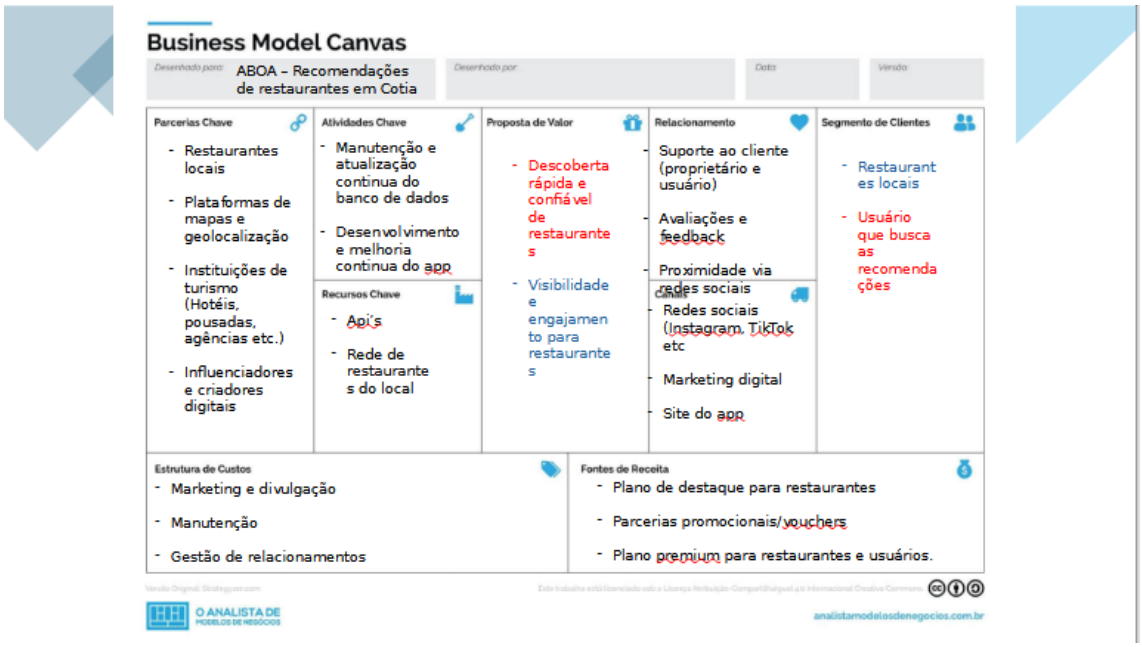
4. Business Model Canvas

O Business Model Canvas do Sistema Aboa foi elaborado com o propósito de descrever os elementos essenciais do modelo de negócios da plataforma, estruturando de forma clara sua proposta e funcionamento. As parcerias-chave englobam o uso da API Here Geocoding, provedores de serviços de mapas, fontes externas de dados e restaurantes parceiros, que possibilitam a oferta de informações precisas e atualizadas aos usuários. As atividades-chave concentram-se no desenvolvimento e na manutenção contínua da plataforma web, bem como na coleta, processamento e análise de dados relacionados à localização, preferências individuais e características dos estabelecimentos cadastrados, garantindo a geração de recomendações contextualizadas e relevantes.

Os recursos principais necessários para a operação do sistema incluem servidores de hospedagem, bancos de dados relacionais e não relacionais, além da equipe técnica responsável pela implementação das funcionalidades e monitoramento do ambiente tecnológico. A proposta de valor da plataforma consiste em fornecer recomendações personalizadas de bares e restaurantes com base em preferências do usuário, localização geográfica e horário de funcionamento dos estabelecimentos, promovendo uma experiência prática, eficiente e contextualizada. O relacionamento com os clientes é estabelecido por meio de avaliações, comentários, suporte integrado ao sistema e interação com redes sociais, fortalecendo o vínculo entre usuários e plataforma. Os segmentos de clientes atendidos abrangem usuários em busca de experiências gastronômicas personalizadas e proprietários de restaurantes interessados em ampliar sua visibilidade no mercado digital. Os canais de entrega correspondem à plataforma web responsiva, acessível em dispositivos móveis e desktops, possibilitando acesso rápido e eficiente às funcionalidades do sistema. A estrutura de custos está associada às despesas envolvendo servidores, manutenção da aplicação e utilização de APIs externas, enquanto as fontes de receita compreendem assinaturas premium direcionadas a estabelecimentos e

parcerias publicitárias que contribuem para a sustentabilidade financeira da plataforma.

A divisão da equipe responsável pelo desenvolvimento do Sistema Aboa foi organizada com base nas competências técnicas e na familiaridade de cada integrante com as ferramentas e tecnologias utilizadas no projeto. A camada de front-end ficou sob responsabilidade de Jhonathan Henrique, que desenvolveu a interface do usuário utilizando o framework React, com foco na criação de uma experiência fluida, intuitiva e responsiva. O back-end foi implementado por Celso Sebastião e Jhonathan Henrique, por meio da construção da API em Node.js, assegurando a integração adequada entre as funcionalidades da aplicação e os dados armazenados. O gerenciamento do banco de dados ficou a cargo de Leandro Cardoso, que realizou a modelagem e a integração entre bancos relacionais utilizados nas fases anteriores e o MongoDB, adotado como banco principal devido à sua flexibilidade estrutural e capacidade de lidar com dados dinâmicos. Essa organização possibilitou maior eficiência no desenvolvimento das sprints, favoreceu o trabalho colaborativo e permitiu o acompanhamento contínuo das atividades em consonância com os princípios da metodologia ágil utilizada pela equipe.



Legenda: Figura 1 – *Business Model Canvas do Sistema Aboa*

5. Gestão de Custos do Sistema Aboa

A gestão de custos do Sistema Aboa contempla os elementos essenciais para o funcionamento de uma plataforma digital escalável, com ênfase em infraestrutura tecnológica, marketing, manutenção operacional e regularização jurídica. Esses componentes garantem sustentabilidade financeira e viabilidade técnica ao longo das etapas de desenvolvimento, lançamento e expansão do sistema.

A infraestrutura de nuvem constitui o núcleo dos custos tecnológicos, sendo adotado um modelo de cobrança baseado no uso, comum a provedores como AWS, Azure e Google Cloud. Em fase inicial, os valores situam-se entre R$ 3.000 e R$ 20.000 mensais, abrangendo servidores, banco de dados e tráfego de rede. A adoção de práticas de otimização, como ajuste de capacidade e utilização de planos de desconto, reduz o impacto financeiro e garante maior eficiência operacional.

Os investimentos em marketing representam uma dimensão central para aquisição de usuários e restaurantes parceiros. As campanhas de mídia digital, incluindo Google Ads e Meta Ads, variam entre R$ 2.500 e R$ 30.000 mensais, ajustadas conforme o retorno sobre investimento. A gestão de redes sociais complementa essa estratégia, com custos entre R$ 1.000 e R$ 15.000 ao mês, contribuindo para o engajamento e a visibilidade da plataforma.

Os custos operacionais envolvem manutenção contínua do software, atualizações e correções, estimadas entre R$ 5.000 e R$ 30.000 mensais. O suporte técnico dos provedores de nuvem apresenta valores reduzidos, enquanto o atendimento ao usuário, quando terceirizado, varia de R$ 5.000 a R$ 30.000 por mês, assegurando a qualidade do serviço e o relacionamento com clientes e parceiros.

Por fim, os custos jurídicos concentram-se em ações iniciais, como a formalização empresarial, o registro da marca no INPI e a elaboração dos documentos legais da plataforma. Esses procedimentos conferem segurança jurídica, proteção da propriedade intelectual e conformidade normativa ao projeto.

---

6. Metodologia (Scrum)

O projeto foi conduzido com o uso da metodologia Scrum, dividindo o desenvolvimento em sprints semanais com reuniões de planejamento, acompanhamento e retrospectivas. O Jira foi utilizado como ferramenta de gestão, registrando backlog, burndown, velocity e entregas incrementais.

Jira

Jira

Displaying **27** issues at **30/Nov/25 9:54 PM**.

Issue Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
Task	PROJ-35	Fazer verificação geral de performance do MongoDB	*Unassigned*	Jhonathan Henrique	Medium	Done	Done	25/Nov/25 12:59 PM	30/Nov/25 3:41 PM
Task	PROJ-33	Criar documentação básica de setup do ambiente	*Unassigned*	Jhonathan Henrique	Medium	Done	Done	25/Nov/25 12:58 PM	30/Nov/25 9:24 PM
Task	PROJ-32	Revisar chamadas do front-end para a API	*Unassigned*	Jhonathan Henrique	Medium	Done	Done	25/Nov/25 12:57 PM	30/Nov/25 9:24 PM
Task	PROJ-31	Revisar endpoints existentes no backend	*Unassigned*	Jhonathan Henrique	Medium	Done	Done	25/Nov/25 12:57 PM	30/Nov/25 3:41 PM
Task	PROJ-30	Revisar estrutura das coleções no MongoDB	*Unassigned*	Jhonathan Henrique	Medium	Done	Done	25/Nov/25 12:56 PM	30/Nov/25 3:41 PM
Story	PROJ-28	Interface	*Unassigned*	Jhonathan Henrique	Medium	Done	Done	15/Sep/25 9:48 PM	29/Sep/25 10:10 PM
Task	PROJ-27	Sistema de Cadastramento de Usuario	*Unassigned*	Leandro Cardoso	Medium	Done	Done	15/Sep/25 8:28 PM	29/Sep/25 9:58 PM
Story	PROJ-26	Cadastro do Usuário	*Unassigned*	Leandro Cardoso	Medium	Done	Done	15/Sep/25 8:26 PM	29/Sep/25 9:58 PM
Task	PROJ-25	Sistema de busca e recomendações	*Unassigned*	Celso Sebastiao de Borba Junior	Medium	Done	Done	15/Sep/25 8:13 PM	25/Nov/25 12:25 PM
Story	PROJ-24	Busca e Recomendações	*Unassigned*	Celso Sebastiao de Borba Junior	Medium	Done	Done	15/Sep/25 8:09 PM	25/Nov/25 12:25 PM
Task	PROJ-23	Sistema de Localização	*Unassigned*	Leandro Cardoso	Medium	To Do	*Unresolved*	15/Sep/25 8:02 PM	15/Sep/25 8:15 PM
Story	PROJ-22	Cadastro de itens no cardápio	Leandro Cardoso	Jhonathan Henrique	Medium	Done	Done	15/Sep/25 8:01 PM	25/Nov/25 12:26 PM

---

7. 5W2H

O projeto consiste no desenvolvimento de uma plataforma multiplataforma de recomendação gastronômica inteligente. O objetivo é oferecer sugestões personalizadas com base em localização, clima e perfil de uso. A necessidade do projeto surge da falta de personalização, informações desatualizadas e baixa inteligência nos aplicativos atuais.

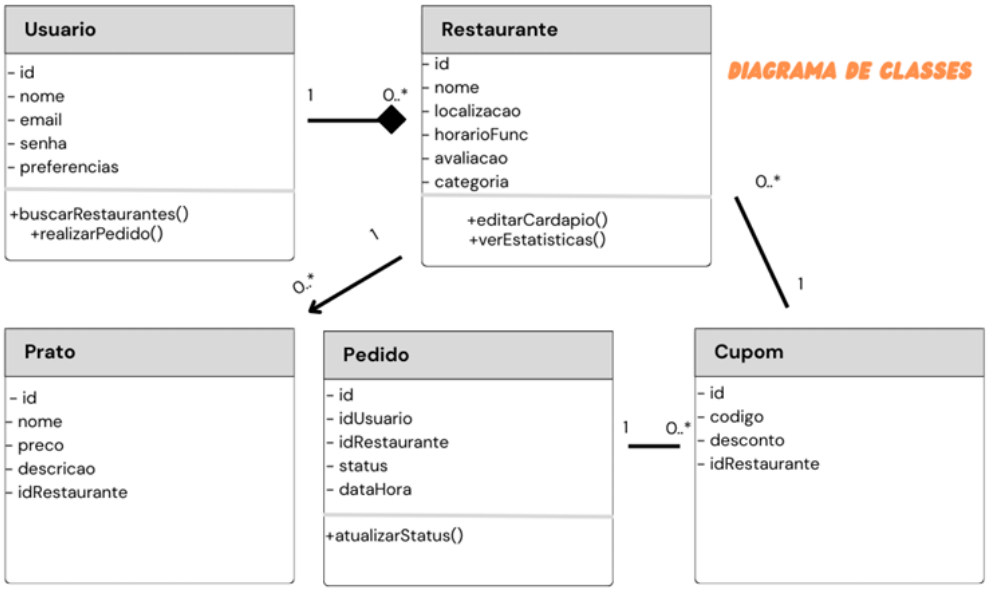
A execução será realizada pela equipe Aboa, composta por Celso Sebastião, Jhonathan Henrique e Leandro Cardoso, sob orientação do professor Braz Izaias da Silva Junior. O desenvolvimento será conduzido em ambientes de programação, simulação e validação.

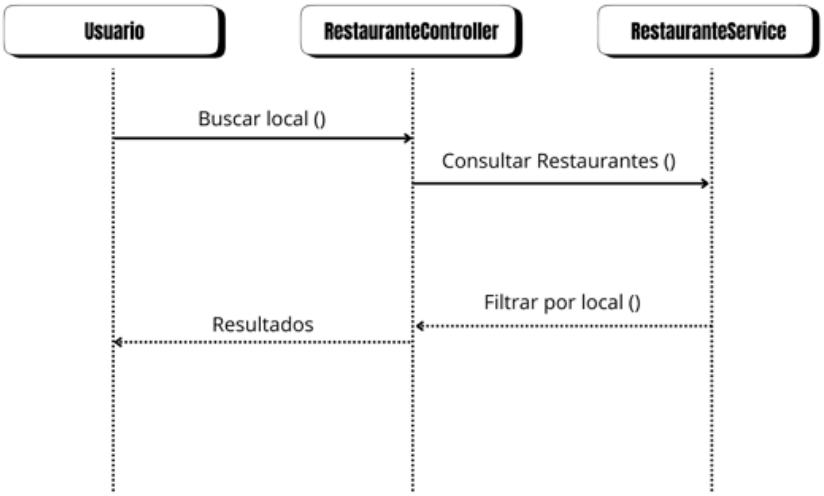
O cronograma está distribuído ao longo de dez semanas, com cenários ideal, médio e crítico definidos. A execução seguirá a metodologia Scrum, com tarefas no Jira, testes contínuos e entregas semanais.

Quanto à sustentabilidade financeira, futuros modelos de receita incluem publicidade segmentada, parcerias de cupons e relatórios analíticos para restaurantes.

---

8. Modelagem UML

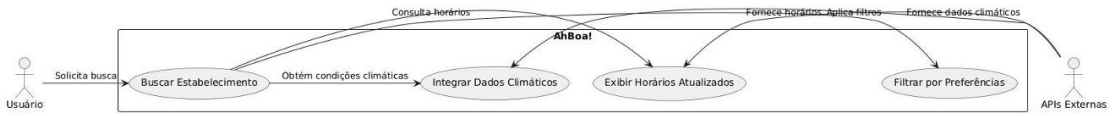




---

9. Casos de Uso

Os casos de uso incluem cadastro de usuários, autenticação, busca de restaurantes, recomendações personalizadas, visualização de cardápios, avaliações, uso de cupons, painel de estatísticas, atualização de cardápios e reservas ou pedidos antecipados. Cada caso de uso considera fluxos principais e exceções, garantindo cobertura das interações possíveis entre usuários, restaurantes e o sistema.



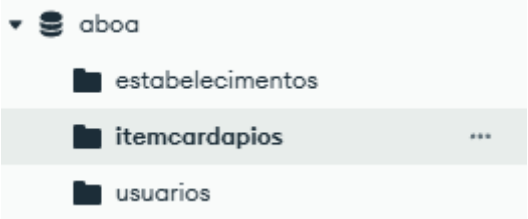
---

10. Banco de Dados – MongoDB

O sistema Aboa utiliza o MongoDB como banco de dados não relacional, escolha que se mostrou adequada ao tipo de informações manipuladas pela aplicação. Por se tratar de uma plataforma dinâmica, que lida com dados que mudam com frequência, como cardápios, fotos, descrições, estabelecimentos e informações de usuários, o uso de um banco orientado a documentos oferece a flexibilidade necessária para acompanhar a evolução contínua do projeto sem a rigidez dos relacionamentos tradicionais encontrados em modelos relacionais. A estrutura do banco foi inicialmente planejada a partir de um Modelo Entidade-Relacionamento (MER), utilizado apenas como referência conceitual para organizar as entidades existentes, seus atributos principais e a forma como elas se conectam dentro do contexto lógico da aplicação. Embora o MongoDB não trabalhe com relacionamentos formais ou obrigatórios entre entidades, essa etapa de modelagem conceitual foi fundamental para definir com clareza a estrutura dos documentos, os campos essenciais e as informações que cada parte do sistema deveria armazenar.

A partir do MER, foi construído o Diagrama Entidade-Relacionamento (DER), que serviu como guia visual para a organização dos documentos no banco. Mesmo que o DER não seja aplicado de forma rígida, ele orientou a divisão do banco em coleções e ajudou a estabelecer o fluxo de informações entre usuários, estabelecimentos e itens de cardápio. No estado atual do projeto, o banco é composto principalmente pelas coleções “usuarios” e “estabelecimentos”. A coleção “usuarios” armazena dados cadastrais, como nome, e-mail, senha e tipo de usuário. Um ponto importante é que todos os usuários começam como usuários comuns e, somente após realizar o cadastro de um estabelecimento, passam a ter o tipo “restaurante”, o que reflete diretamente as regras do sistema e o fluxo real de uso.

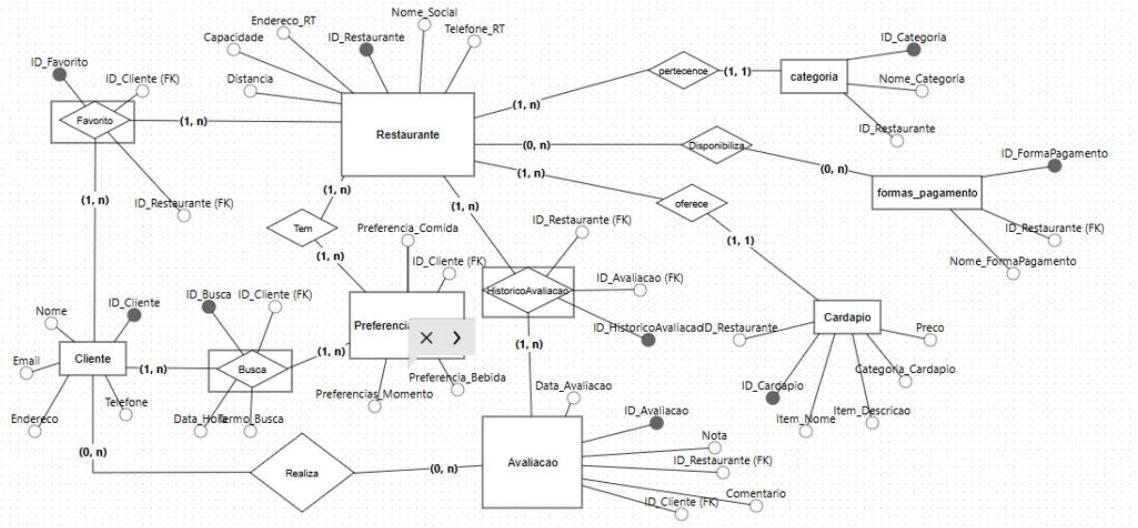
```
use aboa;

db.createCollection("estabelecimentos");
db.createCollection("itemcardapios");
db.createCollection("usuarios");
```



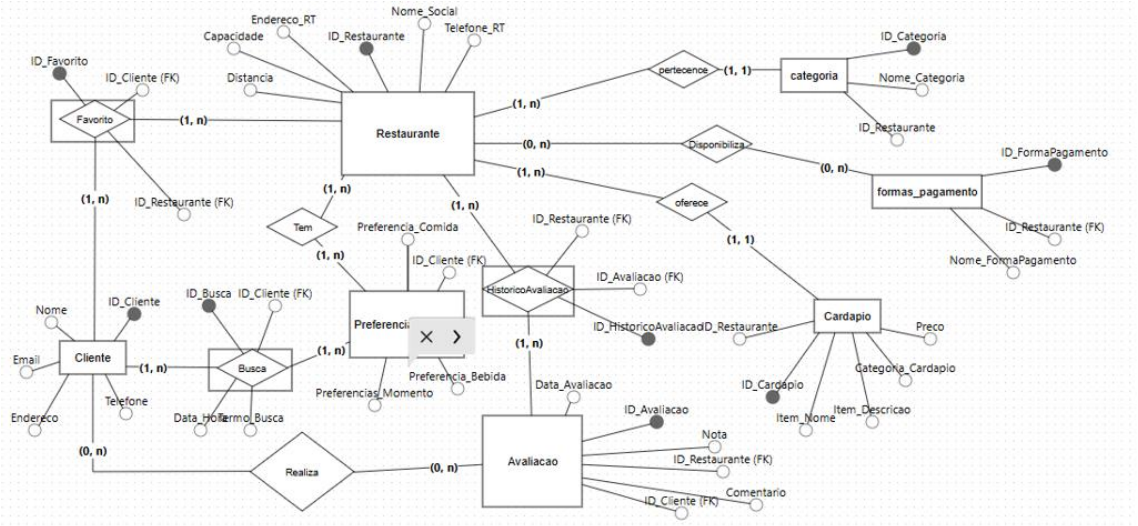
A coleção “estabelecimentos” armazena todas as informações referentes aos restaurantes cadastrados, incluindo nome, endereço, telefone, descrição, foto, além do vínculo com o usuário responsável. O cardápio de cada estabelecimento é estruturado de maneira integrada ao próprio documento do restaurante, aproveitando a flexibilidade de armazenamento do MongoDB. Cada item inclui nome, descrição, categoria, preço, disponibilidade, status de promoção e foto, permitindo que o estabelecimento tenha controle completo sobre sua oferta sem depender de tabelas auxiliares ou relacionamentos rígidos. Essa abordagem é coerente com a arquitetura do MongoDB, pois facilita a leitura dos dados e reduz a necessidade de múltiplas consultas em diferentes coleções.

A adoção de um banco não relacional permitiu que todo o sistema pudesse evoluir sem a necessidade de migrações complexas ou mudanças estruturais profundas. Novos campos podem ser adicionados de forma incremental conforme o projeto cresce, e coleções podem receber informações adicionais sem comprometer as funcionalidades existentes. Essa característica é essencial para um aplicativo que está em desenvolvimento contínuo e que precisa se adaptar a funcionalidades futuras, como avaliações, filtros personalizados, histórico de navegação ou métricas adicionais utilizadas pelo sistema de recomendação. Dessa forma, a utilização do MongoDB não apenas atendeu às necessidades atuais do projeto, mas também garantiu espaço para expansões e melhorias sem impacto negativo na arquitetura geral.

DER



MER



---

11. Avaliação de Usabilidade

A usabilidade será avaliada com base nas heurísticas de Jakob Nielsen, incluindo visibilidade do sistema, prevenção de erros, consistência, controle pelo usuário e feedback. A inspeção semiótica será utilizada para analisar a comunicação entre sistema e usuário.

---

12. Desenvolvimento Web

O front-end do Aboa foi desenvolvido utilizando React, estruturado em componentes reutilizáveis e organizado por páginas, seguindo o padrão de navegação do React Router DOM. O projeto foi configurado com Vite para otimizar o ambiente de desenvolvimento e garantir maior desempenho na renderização.

A interface se comunica diretamente com a API construída em Node.js e Express, responsável por toda a lógica de negócio, persistência de dados e autenticação. As requisições são feitas utilizando `fetch` nativo do navegador (sem Axios), permitindo operações como cadastro, login, gestão de estabelecimentos e manipulação do cardápio.

No back-end, foram implementadas operações completas de CRUD para estabelecimentos e itens do cardápio, além de autenticação baseada em JWT, filtros por categoria, upload de imagens via Multer e integração com MongoDB usando Mongoose. Todo acesso a rotas protegidas é gerenciado por middleware de verificação de token.

A seção de “Recomendações” utiliza dados vindos do servidor para exibir a “Boa do Dia” e demais opções filtradas conforme as categorias cadastradas pelos estabelecimentos. A página de cardápio, por sua vez, apresenta dinamicamente fotos, preços, disponibilidade, promoções e todas as informações cadastradas no painel do restaurante.

---

13. Conclusão

O Sistema Aboa atende de forma satisfatória aos requisitos da A2, apresentando arquitetura moderna, usabilidade aprimorada, métodos ágeis de desenvolvimento e integração entre múltiplas camadas tecnológicas. A solução demonstra potencial para crescimento e aplicabilidade real.

---

14. Referências (ABNT)

NIELSEN, Jakob. *Usability Engineering*. San Francisco: Morgan Kaufmann, 1994. Obra clássica que fundamenta heurísticas e princípios de usabilidade aplicados no design das interfaces do projeto.

PRESSMAN, Roger S. *Engenharia de Software: Uma Abordagem Profissional*. Porto Alegre: McGraw-Hill, 2011.

Referência utilizada para embasar práticas de engenharia de software, organização modular e arquitetura do sistema.

SHNEIDERMAN, Ben; Plaisant, Catherine. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Boston: Addison-Wesley, 2010.
Livro que forneceu diretrizes importantes para decisões de layout, affordances e interação no front-end.

MONGODB. *MongoDB CRUD Operations — Official Manual*. Disponível em: https://www.mongodb.com/docs/manual/crud/.
Documentação utilizada especificamente para implementação e revisão das operações de criação, leitura, atualização e remoção na base de dados.

MONGODB. *Schema Design Best Practices*. Disponível em: https://www.mongodb.com/docs/manual/core/data-modeling-introduction/.
Recurso consultado para definição de modelos e organização dos dados do projeto.

REACT. *React Router – Getting Started Guide*. Disponível em: https://reactrouter.com/en/main/start/overview.
Material utilizado para a implementação e estruturação das rotas do front-end.

VITE. *Vite – Project Setup Guide*. Disponível em: https://vitejs.dev/guide/.
Consultado para configuração, build e otimização do ambiente front-end.

NODE.JS. *Express Middleware Guide*. Disponível em: https://expressjs.com/en/guide/using-middleware.html.
Utilizado como base para criação dos middlewares de autenticação e segurança do back-end.