



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ**  
**CAMPUS PICOS**  
**TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**LIVYA KELLY DA SILVA VELOSO**

**AURAMIST: UMA PLATAFORMA DE GESTÃO DE AGENDAMENTOS PARA  
PROFISSIONAIS E CLIENTES DO SETOR DE BELEZA**

**PICOS, PIAUÍ**  
**2025**

LIVYA KELLY DA SILVA VELOSO

AURAMIST: UMA PLATAFORMA DE GESTÃO DE AGENDAMENTOS PARA  
PROFISSIONAIS E CLIENTES DO SETOR DE BELEZA

Relatório Técnico de Software apresentado  
como exigência para aprovação na disciplina  
Elaboração de Projeto do Curso de Análise e De-  
senvolvimento de Sistemas do Instituto Federal  
de Educação, Ciência e Tecnologia do Piauí -  
*Campus Picos*.

**Orientador:** Prof. Dr. Rogério Figueredo de  
Sousa

PICOS, PIAUÍ  
2025

LIVYA KELLY DA SILVA VELOSO

AURAMIST: UMA PLATAFORMA DE GESTÃO DE AGENDAMENTOS PARA  
PROFISSIONAIS E CLIENTES DO SETOR DE BELEZA

Relatório Técnico de Software apresentado  
como exigência para aprovação na disciplina  
Elaboração de Projeto do Curso de Análise e De-  
senvolvimento de Sistemas do Instituto Federal  
de Educação, Ciência e Tecnologia do Piauí -  
*Campus Picos.*

04 de Setembro de 2025

BANCA EXAMINADORA:

---

**Prof. Dr. Rogério Figueredo de Sousa (Orientador)**  
Instituto Federal do Piauí (IFPI)

---

**Prof. Anthony Irlan Marques Luz (co-orientador)**  
Instituto Federal do Piauí (IFPI)

---

**Prof. M<sup>º</sup>. Jucie Xavier da Silva**  
Instituto Federal do Piauí (IFPI)

---

**Prof. M<sup>º</sup>. Denis Costa Paiva**  
Instituto Federal do Piauí (IFPI)

PICOS, PIAUÍ  
2025

## AGRADECIMENTOS

Agradeço primeiramente a Deus, pela força, coragem e sabedoria concedidas ao longo desta trajetória, sustentando-me nos momentos de dificuldade e iluminando o caminho até a conclusão desta etapa acadêmica. Expresso minha sincera gratidão à minha família, especialmente à minha mãe, que esteve ao meu lado em cada momento. Ela passava comigo finais de semana em claro, até bem tarde da noite, oferecendo sua presença e apoio, com um abraço acolhedor e palavras que me faziam sentir o quanto tem orgulho de mim. Seu incentivo constante, seu amor incondicional e sua força foram fundamentais para que eu pudesse seguir em frente, mesmo nos momentos mais difíceis, sempre me apoiando enquanto estudava, fazia atividades ou me dedicava às tarefas. Sou imensamente grata à minha mãe por tudo o que fez por mim ao longo dessa jornada.

Agradeço também ao meu pai, avós, avô e meus irmãos, que sempre estiveram ao meu lado, com palavras de incentivo e apoio, me dando forças para seguir em frente. A cada gesto, cada palavra, cada olhar de confiança, senti o amor e o apoio de toda minha família, e isso foi essencial para que eu chegasse até aqui. Cada um de vocês teve um papel fundamental nessa trajetória, e sou eternamente grata a todos por me impulsionarem com tanto carinho e dedicação.

Estendo meus agradecimentos aos meus amigos, com quem compartilhei risos, desafios e noites em claro. Maykon, Avelar, Camilla, Carlos e Victor, sem vocês, essa jornada não teria sido a mesma. Cada momento ao lado de vocês foi uma lição de amizade, apoio e companheirismo. Passamos juntos noites estudando, fazendo trabalhos, e até dormindo na casa uns dos outros para conseguir concluir nossas tarefas. Cada um de vocês tem um lugar muito especial no meu coração, e a amizade que construímos durante essa fase da faculdade certamente será levada para a vida toda. Agradeço imensamente por toda ajuda, por me fazerem sentir que a jornada nunca foi difícil demais, porque sempre tive o apoio de vocês. Não teria chegado até aqui sem a amizade de todos vocês. A gratidão que sinto por cada um é imensa.

Gostaria de expressar meu agradecimento ao meu orientador, Prof. Dr. Rogério, pela orientação e apoio ao longo dessa etapa. Seu acompanhamento foi valioso para o desenvolvimento deste trabalho. Agradeço também ao meu co-orientador, Anthony, que, além de ser um grande amigo, teve um papel fundamental no andamento deste projeto. Sua ajuda, conselhos e amizade foram sempre valiosos, e sou grata por sua contribuição e por estar ao meu lado durante essa jornada.

A todos que, de alguma forma, contribuíram direta ou indiretamente para a realização deste trabalho, registro meu sincero agradecimento. Cada palavra de incentivo, gesto de apoio e manifestação de confiança teve importância fundamental nesta trajetória, tornando possível a superação dos desafios encontrados e fortalecendo a determinação necessária para concluir esta etapa acadêmica.

## RESUMO

Este trabalho apresenta o desenvolvimento do AuraMist, uma plataforma destinada a facilitar o agendamento de serviços no setor de beleza. O sistema permite que clientes visualizem os serviços disponíveis e efetuem reservas de forma rápida e prática, enquanto os profissionais podem acompanhar os agendamentos realizados para os serviços que oferecem. O desenvolvimento envolveu a definição de requisitos, a modelagem do sistema com diagramas de caso de uso e entidade-relacionamento, e a implementação utilizando Node.js e Express no back-end, Prisma ORM para gerenciamento de dados, e React com Next.js no front-end. A aplicação foi hospedada em Vercel e Render, garantindo que usuários tenham acesso contínuo e estável. Testes manuais das principais funcionalidades confirmaram o correto funcionamento do sistema. O AuraMist oferece uma solução funcional, de fácil utilização, contribuindo para a organização e eficiência da rotina de agendamentos em salões de beleza e serviços autônomos.

**Palavras-chaves:** Agendamento; Serviços de beleza; Gestão de reservas; Usuário e profissional.

## **ABSTRACT**

This work presents the development of AuraMist, a platform designed to facilitate the scheduling of services in the beauty sector. The system allows clients to view available services and make reservations quickly and conveniently, while professionals can monitor the appointments made for the services they offer. The development process involved requirements definition, system modeling using use case and entity-relationship diagrams, and implementation with Node.js and Express on the back-end, Prisma ORM for data management, and React with Next.js on the front-end. The application was deployed on Vercel and Render, ensuring continuous and stable access for users. Manual tests of the main functionalities confirmed the proper operation of the system. AuraMist provides a functional and easy-to-use solution, contributing to the organization and efficiency of scheduling routines in beauty salons and independent service providers.

**Keywords:** Scheduling; Beauty services; Reservation management; Client and professional.

## LISTA DE ILUSTRAÇÕES

|                                                                                             |    |
|---------------------------------------------------------------------------------------------|----|
| Figura 1 – Diagrama de Caso de Uso do AuraMist. . . . .                                     | 20 |
| Figura 2 – Diagrama de Entidades-Relacionamentos do AuraMist. . . . .                       | 21 |
| Figura 3 – Representação gráfica da arquitetura do AuraMist. . . . .                        | 22 |
| Figura 4 – Tela inicial do painel do cliente, exibindo os serviços disponíveis. . . . .     | 22 |
| Figura 5 – Modal para seleção de data e horário do serviço desejado. . . . .                | 23 |
| Figura 6 – Modal de confirmação com os detalhes do agendamento e profissional. . . . .      | 23 |
| Figura 7 – Mensagem de sucesso exibida após a confirmação do agendamento. . . . .           | 24 |
| Figura 8 – Tela de visualização dos agendamentos realizados pelo cliente. . . . .           | 24 |
| Figura 9 – Modal com detalhes do agendamento e opção de cancelamento. . . . .               | 25 |
| Figura 10 – Tela de perfil do usuário com suas informações de contato. . . . .              | 25 |
| Figura 11 – Modal para edição das informações do perfil do usuário. . . . .                 | 26 |
| Figura 12 – Mensagem de sucesso exibida após a atualização do perfil. . . . .               | 26 |
| Figura 13 – Painel do profissional para cadastrar e gerenciar seus serviços. . . . .        | 27 |
| Figura 14 – Modal para edição dos detalhes de um serviço cadastrado. . . . .                | 27 |
| Figura 15 – Tela de visualização dos agendamentos dos clientes para o profissional. . . . . | 28 |

## LISTA DE TABELAS

|                                                            |    |
|------------------------------------------------------------|----|
| Tabela 1 – Requisitos funcionais do AuraMist . . . . .     | 17 |
| Tabela 2 – Requisitos não funcionais do AuraMist . . . . . | 19 |
| Tabela 3 – Testes do AuraMist . . . . .                    | 30 |



## **LISTA DE ABREVIATURAS E SIGLAS**

|       |                                                              |
|-------|--------------------------------------------------------------|
| API   | Application Programming Interface                            |
| BI    | Business Intelligence                                        |
| CI/CD | Continuous Integration / Continuous Delivery                 |
| CRM   | Customer Relationship Management                             |
| CSS   | Cascading Style Sheets                                       |
| DER   | Diagrama de Entidade-Relacionamento                          |
| HTTP  | Hypertext Transfer Protocol                                  |
| IFPI  | Instituto Federal de Educação, Ciência e Tecnologia do Piauí |
| JSON  | JavaScript Object Notation                                   |
| JWT   | JSON Web Token                                               |
| ORM   | Object-Relational Mapping                                    |
| SQL   | Structured Query Language                                    |
| UX    | User Experience                                              |

## LISTA DE SÍMBOLOS

|   |             |
|---|-------------|
| % | Porcentagem |
|---|-------------|

## SUMÁRIO

|          |                                                |           |
|----------|------------------------------------------------|-----------|
|          | <b>Lista de tabelas . . . . .</b>              | <b>7</b>  |
| <b>1</b> | <b>INTRODUÇÃO . . . . .</b>                    | <b>11</b> |
| <b>2</b> | <b>OBJETIVOS . . . . .</b>                     | <b>13</b> |
| 2.1      | OBJETIVO GERAL . . . . .                       | 13        |
| 2.2      | OBJETIVOS ESPECÍFICOS . . . . .                | 13        |
| <b>3</b> | <b>TECNOLOGIAS RELACIONADAS . . . . .</b>      | <b>14</b> |
| 3.1      | BACK-END . . . . .                             | 14        |
| 3.2      | FRONT-END . . . . .                            | 15        |
| <b>4</b> | <b>MODELAGEM . . . . .</b>                     | <b>17</b> |
| 4.1      | Levantamento de Requisitos . . . . .           | 17        |
| 4.2      | Diagrama de Caso de Uso . . . . .              | 20        |
| 4.3      | Diagrama de Entidades-Relacionamento . . . . . | 20        |
| 4.4      | Arquitetura do Sistema . . . . .               | 21        |
| 4.5      | Interface . . . . .                            | 22        |
| <b>5</b> | <b>SOFTWARE . . . . .</b>                      | <b>29</b> |
| 5.1      | IMPLANTAÇÃO . . . . .                          | 29        |
| 5.2      | TESTES . . . . .                               | 29        |
| <b>6</b> | <b>CONSIDERAÇÕES FINAIS . . . . .</b>          | <b>32</b> |
| 6.1      | Resultados alcançados . . . . .                | 32        |
|          | <b>REFERÊNCIAS . . . . .</b>                   | <b>33</b> |

## 1 INTRODUÇÃO

A transformação digital tem redefinido a dinâmica dos mercados, impulsionando empresas a adotarem novas tecnologias para otimizar operações e melhorar a experiência do cliente (SOUZA, 2021). No Brasil, onde o setor de serviços, especialmente o de beleza e bem-estar, representa uma parcela significativa da economia, a digitalização não é mais uma opção, mas uma necessidade para a competitividade. Dados do relatório *Transformação Digital nos Pequenos Negócios* (SEBRAE, 2023) revelam que 89,7% dos pequenos empreendimentos já utilizam a internet para otimizar suas operações, evidenciando uma clara demanda por soluções tecnológicas eficientes.

Apesar dessa tendência, muitos estabelecimentos do setor de beleza ainda dependem de métodos de agendamento tradicionais, como ligações telefônicas e aplicativos de mensagens como o WhatsApp. Embora digitais, essas ferramentas não foram projetadas para a gestão de agendamentos, o que frequentemente resulta em falhas operacionais, como reservas duplicadas, erros de comunicação e sobrecarga administrativa para os profissionais. Essa ineficiência compromete não apenas a produtividade, mas também a satisfação e a fidelização dos clientes (NOVAIS, 2024).

Embora o mercado disponibilize plataformas de agendamento como *Trinks*<sup>1</sup>, *AppBeleza*<sup>2</sup>, *Fresha*<sup>3</sup> e *Gendo*<sup>4</sup>, sua adequação varia conforme o perfil do negócio. Nos planos de entrada, observam-se limitações típicas de personalização (branding, domínio próprio, formulários/fluxos customizados e regras finas de agenda, como buffers e bloqueios), de integrações específicas (gateways de pagamento, WhatsApp oficial/API, CRM/BI) e de relatórios avançados (métricas e exports sob medida). Em cenários de microempreendedores e autônomos, essas restrições podem comprometer a aderência aos processos reais do dia a dia.

Quanto ao custo, há modelos por assinatura e por transação. AppBeleza divulga planos a partir de R\$ 48,90/mês (1 profissional), R\$ 69,90/mês (2–5) e R\$ 108,00/mês (6–15); Trinks apresenta valores “a partir de R\$ 72” para estabelecimentos pequenos; Fresha não cobra mensalidade para autônomos, mas aplica comissão única de 20% para novos clientes vindos do marketplace (mínimo de US\$ 6 por cliente) e taxas de processamento em pagamentos (ex.: 2,29% + US\$ 0,20 no presencial e 2,79% + US\$ 0,20 no online). Gendo prioriza automação via WhatsApp e lembretes, com oferta por assinatura e divulgação de preços menos detalhada. Para profissionais com demanda irregular, um fixo mensal na faixa de R\$ 48,90 a R\$ 72 pode ser desproporcional; e, no modelo por transação, um serviço de R\$ 100 para um cliente novo do marketplace implicaria R\$ 20 de comissão já no primeiro agendamento (além das taxas de pagamento), o que pesa especialmente em tíquetes baixos.

Diante dessa problemática, a tecnologia surge como uma ferramenta estratégica para

<sup>1</sup> <<https://negocios.trinks.com/planos>>

<sup>2</sup> <<https://appbeleza.com.br/>>

<sup>3</sup> <<https://www.fresha.com/pricing>>

<sup>4</sup> <<https://acesso.gendo.app/>>

preencher essa lacuna. É neste contexto que o presente relatório técnico detalha o desenvolvimento do **AuraMist**, uma solução de software projetada para otimizar a gestão de agendamentos no setor de beleza. Diferentemente das opções de mercado analisadas, a plataforma é disponibilizada sem cobrança de mensalidade, licença ou comissão por agendamento. Essa abordagem visa oferecer autonomia para os clientes realizarem seus próprios agendamentos, ao mesmo tempo em que fornece aos profissionais uma ferramenta simplificada para o controle de suas agendas e atendimentos.

O restante deste relatório está organizado da seguinte forma: a Seção 2 apresenta os objetivos do projeto, divididos em objetivo geral e objetivos específicos; a Seção 3 descreve as tecnologias e ferramentas adotadas durante o ciclo de desenvolvimento; a Seção 4 detalha a modelagem do sistema, desde o levantamento de requisitos e casos de uso até o design da interface; na Seção 5, são apresentados o processo de implementação e os testes realizados; e, por fim, a Seção 6 consolida as considerações finais e os resultados alcançados com este trabalho.

## **2 OBJETIVOS**

### **2.1 OBJETIVO GERAL**

Desenvolver uma ferramenta de gestão e atendimento online e avaliar o seu impacto no setor de beleza.

### **2.2 OBJETIVOS ESPECÍFICOS**

- Identificar as principais dificuldades enfrentadas pelos estabelecimentos de beleza na gestão de agendamentos e atendimento ao cliente;
- Investigar como a automação pode reduzir falhas, como sobreposição de horários e ausência de comparecimento;
- Desenvolver uma ferramenta específica para o setor de beleza, visando otimizar a organização dos agendamentos e melhorar a experiência dos clientes;
- Avaliar a funcionalidade e o desempenho da ferramenta desenvolvida, garantindo sua eficácia na organização de agendamentos no setor de beleza.

### 3 TECNOLOGIAS RELACIONADAS

#### 3.1 BACK-END

O JavaScript foi a linguagem escolhida para o desenvolvimento do back-end do AuraMist, executado no ambiente Node.js. A decisão se baseou na ampla adoção dessa tecnologia e na facilidade de integração com aplicações web. Pesquisas recentes confirmam o JavaScript como a linguagem mais utilizada em projetos de software, presente em mais da metade das aplicações modernas (JetBrains, 2024).

Para a construção da API, responsável por conectar o front-end ao banco de dados, utilizou-se o Express.js<sup>1</sup>, um *framework* leve que simplifica a criação de rotas e o tratamento de requisições HTTP. No contexto do AuraMist, o Express foi essencial para organizar os *endpoints* relacionados a autenticação, serviços e agendamentos, além de permitir o uso de *middlewares* personalizados para validação de dados, tratamento de erros e proteção de rotas. Em conjunto, foi implementado o mecanismo de autenticação por meio de JSON Web Tokens (JWT), garantindo controle de acesso às funcionalidades da aplicação de acordo com o perfil do usuário e reforçando a segurança no uso cotidiano do sistema.

O banco de dados selecionado foi o PostgreSQL<sup>2</sup>, reconhecido por sua robustez, conformidade com padrões SQL e forte comunidade de suporte. No AuraMist, optou-se pelo PostgreSQL em vez de alternativas como MySQL ou MongoDB, devido ao seu suporte a operações transacionais e à facilidade de modelagem relacional entre entidades críticas, como usuários, serviços e agendamentos. O banco foi hospedado na plataforma Neon<sup>3</sup>, que oferece escalabilidade automática, interface web para administração remota e criação simplificada de instâncias, eliminando a necessidade de configuração manual de servidores locais.

Na camada de persistência, foi utilizado o Prisma ORM<sup>4</sup>, que gera consultas tipadas para o Node.js e facilita a manipulação de dados. Entre suas vantagens estão a geração automática de clientes tipados, suporte a *migrations* e integração nativa com PostgreSQL. No AuraMist, o Prisma foi utilizado para estruturar entidades como usuários, serviços, agendamentos e imagens, trazendo ganhos de produtividade no desenvolvimento e maior confiabilidade nas operações de escrita e leitura de dados.

As imagens cadastradas pelos profissionais foram armazenadas diretamente no banco de dados PostgreSQL, por meio do Prisma ORM, que gerencia as referências vinculadas às entidades correspondentes. Essa abordagem centralizada simplifica a arquitetura da aplicação e garante consistência no gerenciamento das mídias, sem necessidade de serviços externos de hospedagem.

---

<sup>1</sup> <<https://expressjs.com/>>

<sup>2</sup> <<https://www.postgresql.org/>>

<sup>3</sup> <<https://neon.tech/>>

<sup>4</sup> <<https://www.prisma.io/>>

O serviço de hospedagem escolhido para o back-end foi o Render<sup>5</sup>, que oferece suporte integrado a aplicações Node.js, automação de deploy e escalabilidade sob demanda. Para o AuraMist, o Render trouxe praticidade na configuração e confiabilidade no processo de publicação, permitindo implantações rápidas e contínuas sem complexidade adicional de infraestrutura. Além disso, o Render facilitou a integração de variáveis de ambiente e o monitoramento de logs em produção, recursos importantes para assegurar a estabilidade e a evolução contínua da aplicação.

### 3.2 FRONT-END

O front-end do AuraMist foi desenvolvido com a biblioteca React<sup>6</sup>, utilizada para construção de componentes dinâmicos e reutilizáveis. Essa abordagem possibilitou a criação de elementos como os cartões de serviços, a listagem de agendamentos e os modais de reserva, garantindo maior organização e manutenção do código. A ampla comunidade do React e sua documentação consolidada foram fatores decisivos na escolha, pois facilitaram a resolução de problemas e aceleraram o processo de desenvolvimento.

Para estruturar as páginas e gerenciar o roteamento da aplicação, adotou-se o Next.js<sup>7</sup>, *framework* que expande as funcionalidades do React e simplifica a criação de rotas. No AuraMist, o Next.js permitiu organizar áreas distintas para clientes e profissionais, além de fornecer suporte a páginas privadas protegidas por autenticação, garantindo que cada perfil tivesse acesso apenas às funcionalidades correspondentes. Essa divisão contribuiu para uma navegação mais clara e alinhada às regras de negócio do sistema.

A estilização foi implementada com CSS, organizada em arquivos modulares que possibilitam separação de responsabilidades e maior controle sobre a aparência da aplicação. Foram criadas classes reutilizáveis para padronizar cores, espaçamentos e tipografia, de modo a manter consistência em todas as páginas. Essa abordagem também garantiu responsividade, possibilitando adaptação da interface a diferentes tamanhos de tela, aspecto essencial considerando que grande parte dos usuários acessa o sistema por dispositivos móveis. Optou-se pelo CSS puro em vez de frameworks como *Bootstrap* ou *Tailwind*, justamente para manter a aplicação mais leve, com menor acoplamento a bibliotecas externas e maior liberdade de personalização.

Complementarmente, foram integradas bibliotecas como `react-icons`<sup>8</sup>, utilizada para ícones vetoriais; `react-toastify`<sup>9</sup>, responsável por exibir notificações e mensagens de feedback em ações críticas; e `axios`<sup>10</sup>, utilizada para realizar requisições HTTP entre front-end e back-end de forma eficiente, facilitando a comunicação com a API.

---

<sup>5</sup> <<https://render.com>>

<sup>6</sup> <<https://reactjs.org/>>

<sup>7</sup> <<https://nextjs.org/>>

<sup>8</sup> <<https://react-icons.github.io/react-icons/>>

<sup>9</sup> <<https://fkhadra.github.io/react-toastify/>>

<sup>10</sup> <<https://axios-http.com/>>



O deploy do front-end foi realizado na plataforma Vercel<sup>11</sup>, que oferece integração nativa com projetos em Next.js. Essa escolha simplificou o processo de publicação e proporcionou suporte a variáveis de ambiente, pré-visualização de versões e escalabilidade automática, garantindo estabilidade e praticidade na manutenção da aplicação em produção. Além disso, a Vercel oferece métricas de desempenho e atualizações rápidas, permitindo acompanhar a evolução do sistema e aplicar melhorias contínuas de forma eficiente.

---

<sup>11</sup> <<https://vercel.com>>

## 4 MODELAGEM

### 4.1 LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é uma fase essencial do processo de desenvolvimento de sistemas, pois define as funcionalidades esperadas e estabelece os limites do que o software deverá contemplar. Trata-se de compreender de maneira clara quais problemas serão resolvidos, quem são os usuários envolvidos e quais restrições precisam ser consideradas.

No caso do *AuraMist*, essa etapa foi conduzida a partir da análise das dificuldades enfrentadas por profissionais da área de beleza no gerenciamento de suas agendas e pela observação das demandas recorrentes dos clientes na marcação de serviços. A definição dos requisitos buscou traduzir esses cenários em funcionalidades objetivas, garantindo que a plataforma fosse prática, intuitiva e capaz de oferecer autonomia tanto para profissionais quanto para clientes.

A partir dessa análise, foram definidos os requisitos funcionais e não funcionais do *AuraMist*, apresentados, respectivamente, nas **Tabelas 1 e 2**.

Tabela 1 – Requisitos funcionais do AuraMist

| Código | Requisito                | Descrição                                                                                     |
|--------|--------------------------|-----------------------------------------------------------------------------------------------|
| RF01   | Login do usuário         | O sistema deve permitir que o usuário entre utilizando suas credenciais (e-mail e senha).     |
| RF02   | Registro de usuário      | O sistema deve possibilitar o cadastro de novos usuários com nome, telefone, e-mail e senha.  |
| RF03   | Definição de perfil      | O usuário deve indicar se atuará como Cliente ou como Profissional durante o cadastro.        |
| RF04   | Edição de perfil         | O usuário pode visualizar e atualizar suas informações pessoais no sistema.                   |
| RF05   | Cadastro de serviço      | O profissional deve cadastrar serviços com nome, descrição, preço, duração e imagem opcional. |
| RF06   | Gerenciamento de serviço | O profissional pode editar ou remover serviços cadastrados.                                   |
| RF07   | Consulta de serviços     | O cliente pode visualizar serviços cadastrados, sem filtros por profissional.                 |

Continua na próxima página

**Continuação da Tabela 1 — Requisitos funcionais do AuraMist**

| <b>Código</b> | <b>Requisito</b>                             | <b>Descrição</b>                                                                                                                       |
|---------------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| RF08          | Detalhamento do serviço                      | O sistema exibe informações completas de um serviço, com a possibilidade de selecionar horário para agendamento e confirmar a reserva. |
| RF09          | Disponibilidade do serviço                   | O profissional deve informar, na descrição do serviço, os horários em que está disponível.                                             |
| RF10          | Visualização de reservas (prof.)             | O profissional pode visualizar os agendamentos feitos para os seus serviços.                                                           |
| RF11          | Agendamento de serviço                       | O cliente pode selecionar uma data e horário para reservar um serviço, sendo notificado em caso de conflito.                           |
| RF12          | Evitar conflitos de agendamento              | O sistema exibe uma mensagem de conflito caso haja sobreposição de horário no agendamento.                                             |
| RF13          | Alteração de agendamento                     | O cliente deve cancelar um agendamento para realizar uma nova reserva.                                                                 |
| RF14          | Cancelamento de agendamento                  | O cliente ou profissional pode cancelar agendamentos, sem a necessidade de justificar a ação.                                          |
| RF15          | Mensagens de confirmação                     | O sistema exibe uma mensagem de confirmação para agendamentos realizados com sucesso.                                                  |
| RF16          | Visualização de agendamentos do cliente      | O cliente pode visualizar os agendamentos realizados, passados e futuros, desde que não tenham sido cancelados.                        |
| RF17          | Visualização de agendamentos do profissional | O profissional pode visualizar os agendamentos realizados, passados e futuros, dos serviços cadastrados, exceto os cancelados.         |
| RF18          | Imagens de serviços                          | O sistema permite que o profissional associe uma imagem ao serviço, visível para o cliente durante o agendamento.                      |
| RF19          | Controle de acesso por perfil                | O sistema restringe funcionalidades com base no perfil do usuário (Cliente ou Profissional).                                           |

Continua na próxima página

**Continuação da Tabela 1 — Requisitos funcionais do AuraMist**

| <b>Código</b> | <b>Requisito</b>  | <b>Descrição</b>                                                           |
|---------------|-------------------|----------------------------------------------------------------------------|
| RF20          | Logout do usuário | O usuário deve ser capaz de sair da aplicação, retornando à tela de login. |

**Tabela 2 – Requisitos não funcionais do AuraMist**

| <b>Código</b> | <b>Requisito</b>         | <b>Descrição</b>                                                                                                                        |
|---------------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| RNF01         | Usabilidade              | A plataforma deve ser simples e intuitiva, garantindo navegação fácil tanto para profissionais quanto para clientes.                    |
| RNF02         | Acessibilidade           | O sistema deve ser acessível em diferentes dispositivos e tamanhos de tela, garantindo fácil navegação.                                 |
| RNF03         | Escalabilidade           | A arquitetura deve suportar o aumento de usuários e dados, mantendo a eficiência com a demanda crescente.                               |
| RNF04         | Confiabilidade           | O sistema deve assegurar a integridade dos dados e a continuidade das operações, mesmo em falhas inesperadas.                           |
| RNF05         | Proteção de dados        | O sistema deve proteger as informações sensíveis, garantindo que apenas usuários autorizados tenham acesso a funcionalidades restritas. |
| RNF06         | Estabilidade             | O sistema deve manter desempenho constante e disponibilidade para os usuários.                                                          |
| RNF07         | Flexibilidade de acesso  | O sistema deve ser acessível em qualquer dispositivo conectado à internet, garantindo experiência consistente.                          |
| RNF08         | Manutenibilidade         | O código do sistema deve ser modular e bem documentado, facilitando a manutenção e a evolução de funcionalidades.                       |
| RNF09         | Compatibilidade com APIs | O sistema deve integrar-se de forma eficiente com APIs externas, como serviços de pagamento e notificações.                             |

Continua na próxima página

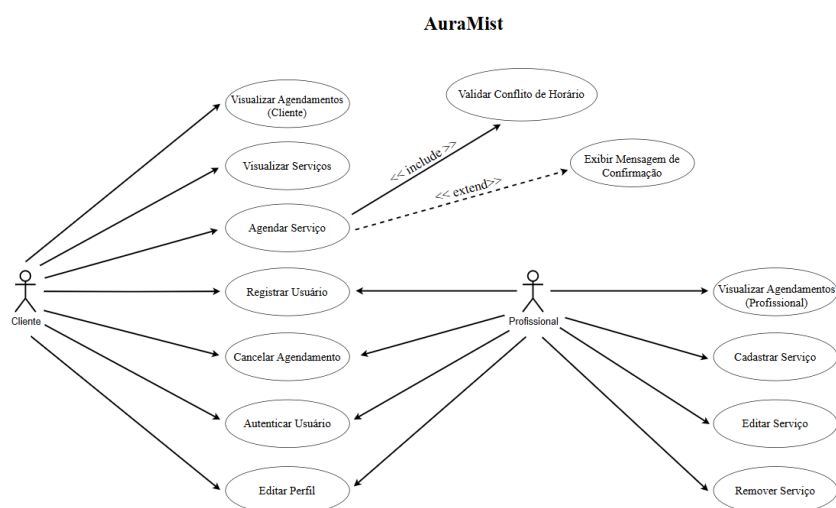
**Continuação da Tabela 2 — Requisitos não funcionais do AuraMist**

| Código | Requisito  | Descrição                                                                                                                               |
|--------|------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| RNF10  | Desempenho | O sistema deve garantir respostas rápidas para operações como consultas e agendamentos, mesmo com grande número de acessos simultâneos. |

## 4.2 DIAGRAMA DE CASO DE USO

O Diagrama de Caso de Uso é uma ferramenta fundamental para representar as interações entre os usuários (atores) e o sistema, identificando as funcionalidades principais do software. Ele descreve, de forma clara e objetiva, como os diferentes perfis de usuários interagem com o sistema e o que é esperado de cada ação realizada. Esse diagrama é peça essencial na modelagem dos requisitos, pois permite um entendimento claro sobre o comportamento visível do sistema e os serviços que ele deve fornecer no contexto de seu ambiente (SILVA; MARTINS; DINIZ, 2017).

No contexto do *AuraMist*, o diagrama da Figura 1 mapeia as interações entre *Clientes*, *Profissionais* e o próprio *Sistema*. O *Cliente* possui interações como: *Agendar Serviço*, *Visualizar Agendamentos* e *Cancelar Agendamentos*. O *Profissional*, por sua vez, pode: *Cadastrar Serviço*, *Editar Serviços* e *Visualizar Agendamentos*. Já o *Sistema* atua na validação e organização dos agendamentos, garantindo que o processo ocorra sem conflitos de horários.



**Figura 1 – Diagrama de Caso de Uso do AuraMist.**

## 4.3 DIAGRAMA DE ENTIDADES-RELACIONAMENTO

O Diagrama de Entidade-Relacionamento (DER) é uma ferramenta visual essencial no processo de modelagem de banco de dados, pois organiza e explicita as entidades do sistema e

os relacionamentos entre elas, sendo vital para garantir integridade, consistência e desempenho eficiente.

No contexto do AuraMist, o diagrama da Figura 2 apresenta as entidades centrais: *Usuários*, *Serviços* e *Agendamentos*. *Usuários* abrange tanto clientes quanto profissionais (via atributo `role`); *Serviços* pertence a um profissional; e *Agendamentos* conecta cliente, serviço e profissional. As imagens de serviço e de perfil não constituem entidade separada: são armazenadas diretamente nos registros do banco, em formato Base64, nos campos `Service.urlImage` e `User.profilePictureUrl`.

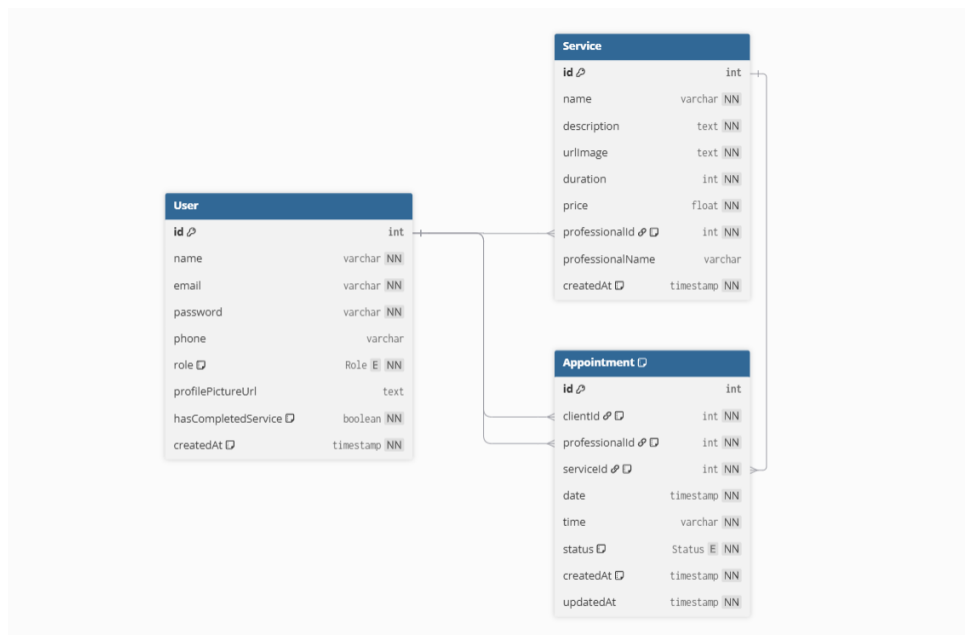


Figura 2 – Diagrama de Entidades-Relacionamentos do AuraMist.

#### 4.4 ARQUITETURA DO SISTEMA

A arquitetura de software organiza os componentes e define como as partes do sistema interagem entre si e com o ambiente. No *AuraMist*, adotou-se uma arquitetura em camadas com foco em desempenho, escalabilidade e facilidade de manutenção. No nível de apresentação, o *front-end* em *React* é responsável pela interação com o usuário e está hospedado na Vercel; a comunicação com a aplicação ocorre por meio de uma *API REST*, construída em *Node.js* e *Express* e hospedada no Render, com troca de mensagens via *HTTPS/JSON*.

Na camada de dados, a persistência utiliza *PostgreSQL* (Neon), acessado por meio do *Prisma ORM*, que provê tipagem, *migrations* e acesso seguro aos dados. O armazenamento de imagens segue o modelo definido no DER (cf. Figura 2). A Figura 3 ilustra a organização geral.

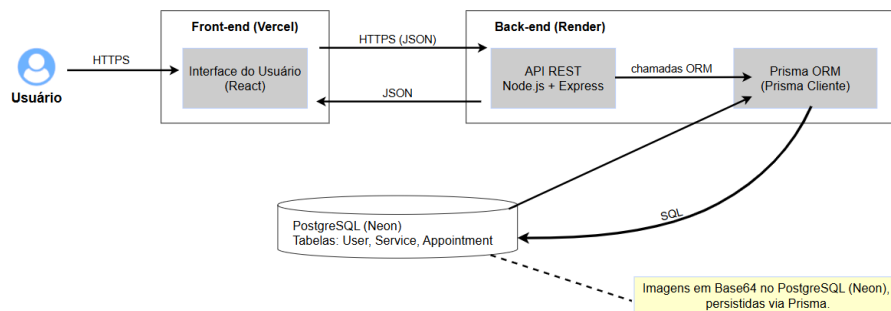


Figura 3 – Representação gráfica da arquitetura do AuraMist.

## 4.5 INTERFACE

A interface do *AuraMist* foi projetada para ser simples, intuitiva e eficiente, com foco em experiência do usuário (UX) e redução da curva de aprendizado. O *front-end* em *React* adota componentes dinâmicos e modulares, garantindo consistência e alta performance. A interface é responsiva (smartphones, tablets e desktops), e a comunicação com o *back-end* foi planejada para manter interações fluidas. Optou-se por CSS puro, priorizando controle fino da apresentação e menor dependência externa.

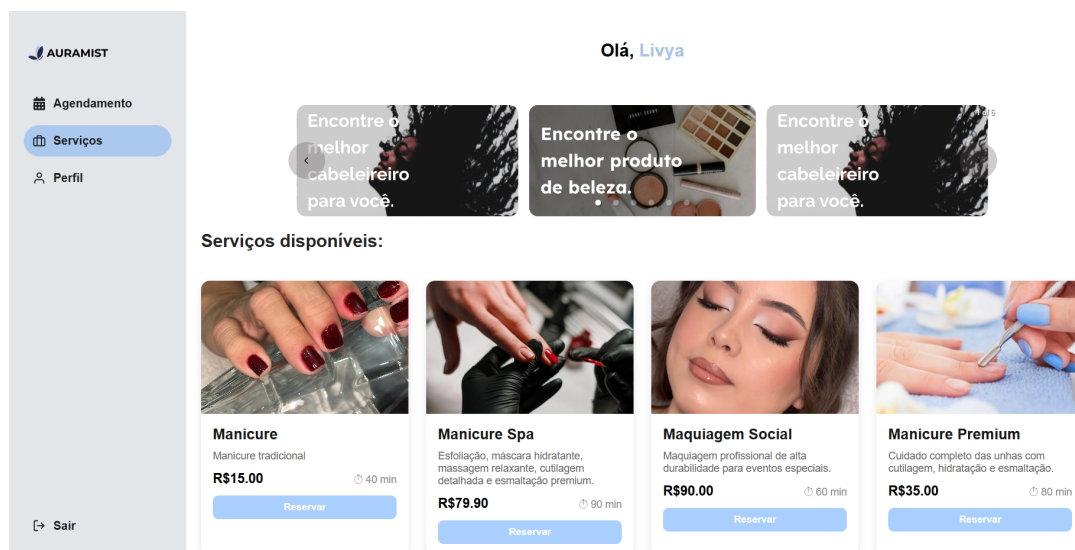


Figura 4 – Tela inicial do painel do cliente, exibindo os serviços disponíveis.

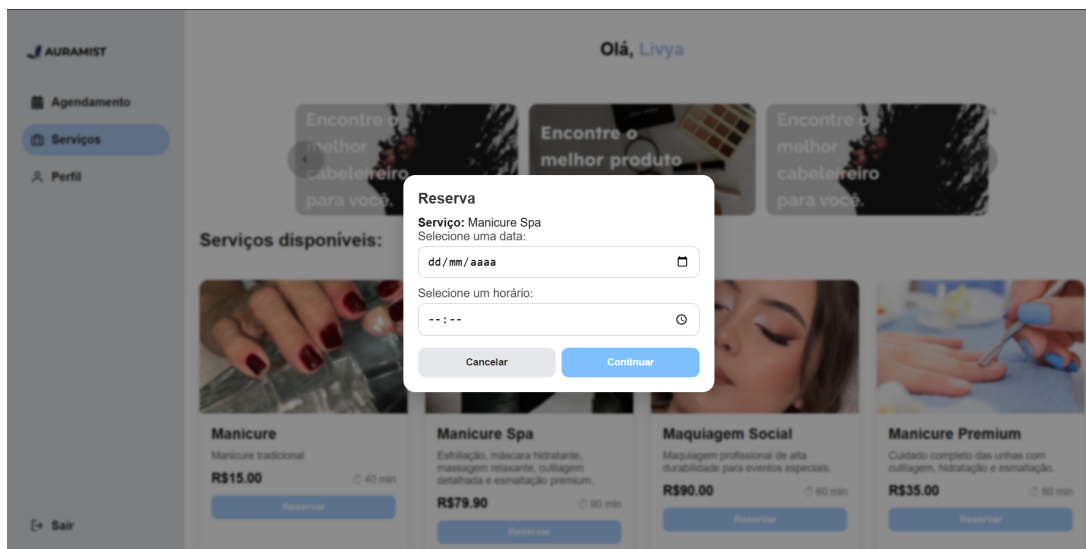


Figura 5 – Modal para seleção de data e horário do serviço desejado.

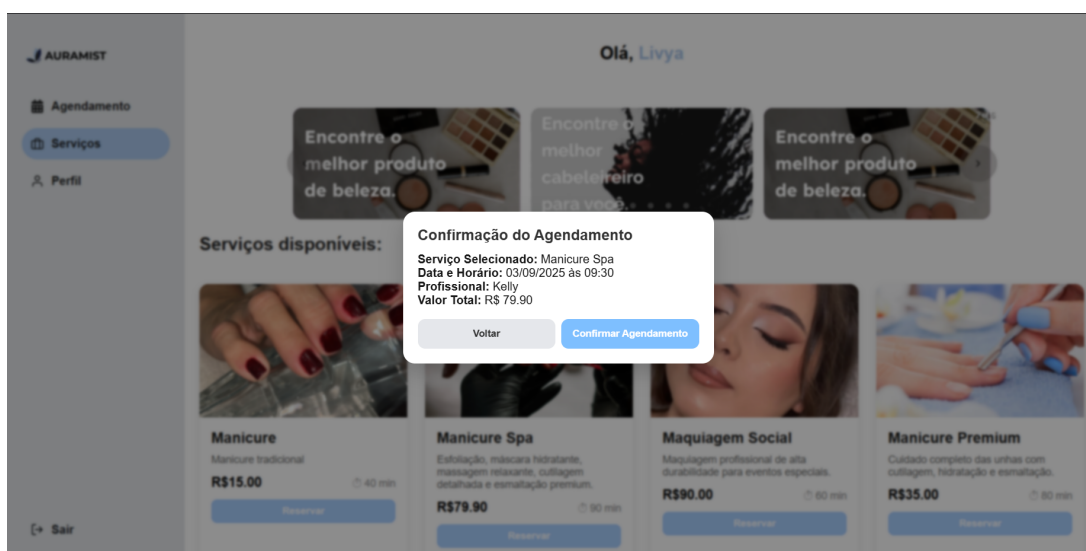


Figura 6 – Modal de confirmação com os detalhes do agendamento e profissional.



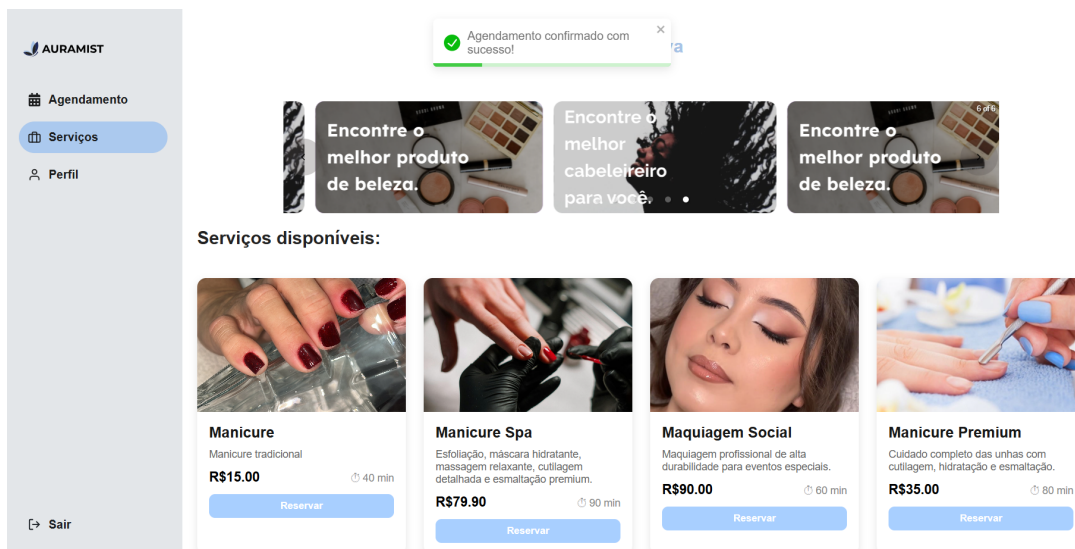


Figura 7 – Mensagem de sucesso exibida após a confirmação do agendamento.

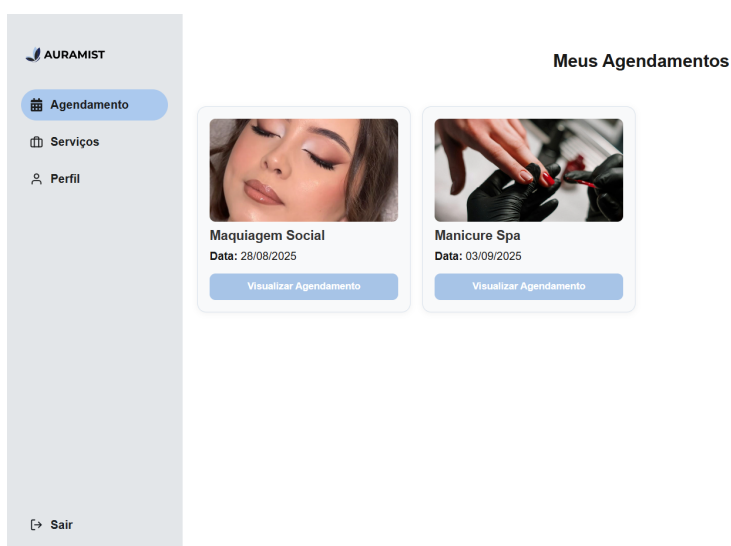


Figura 8 – Tela de visualização dos agendamentos realizados pelo cliente.

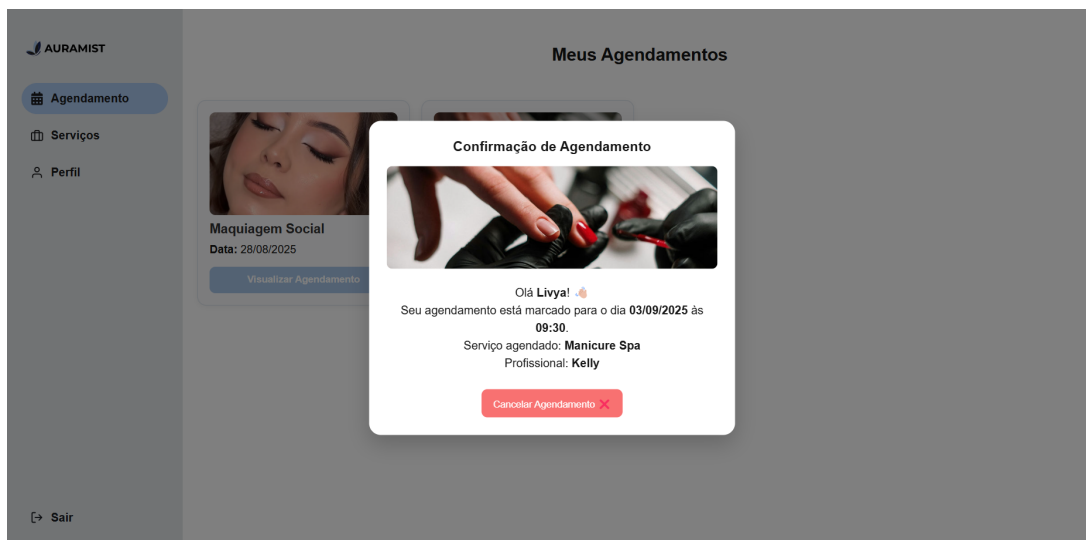


Figura 9 – Modal com detalhes do agendamento e opção de cancelamento.

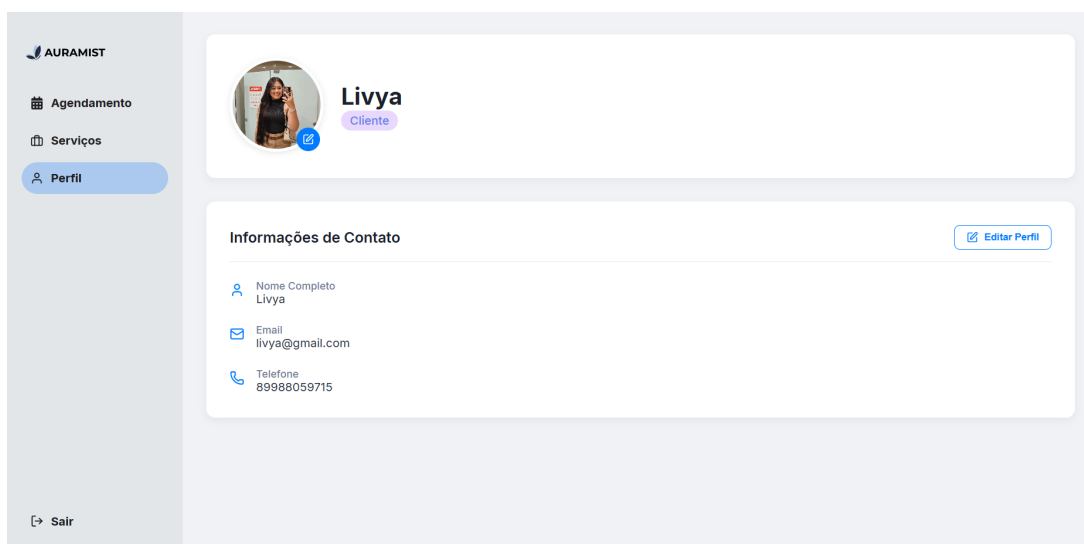


Figura 10 – Tela de perfil do usuário com suas informações de contato.

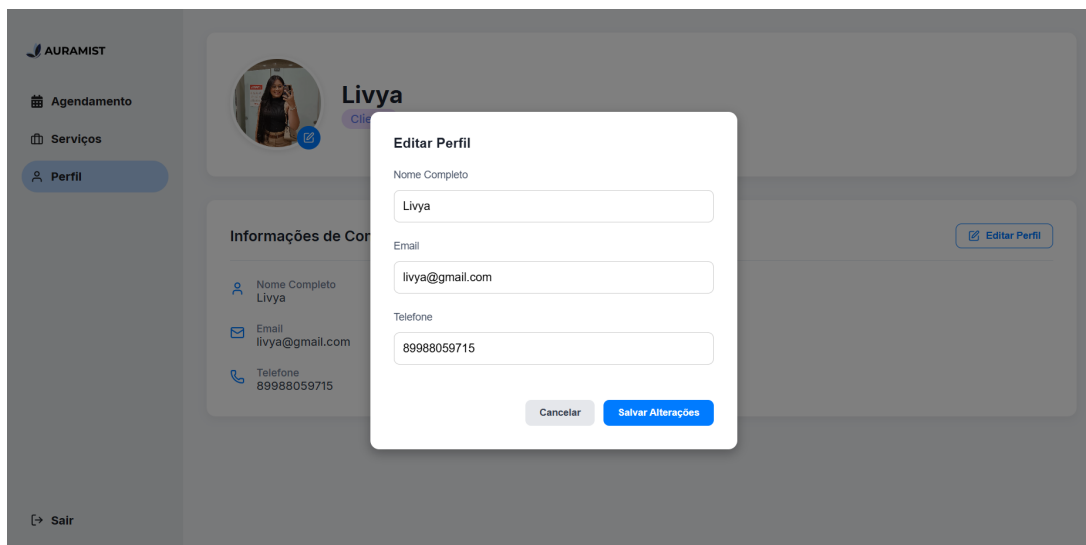


Figura 11 – Modal para edição das informações do perfil do usuário.

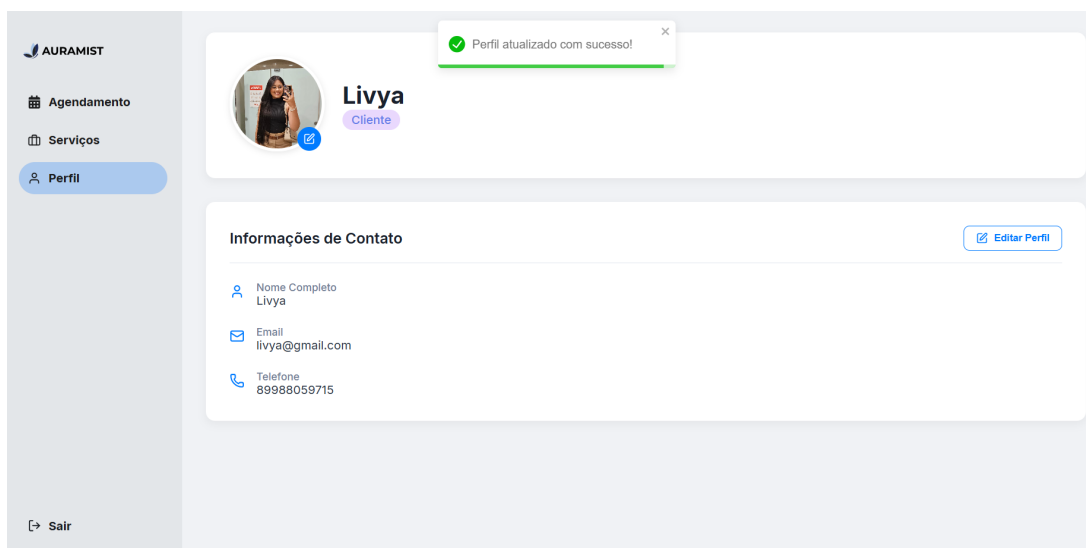


Figura 12 – Mensagem de sucesso exibida após a atualização do perfil.

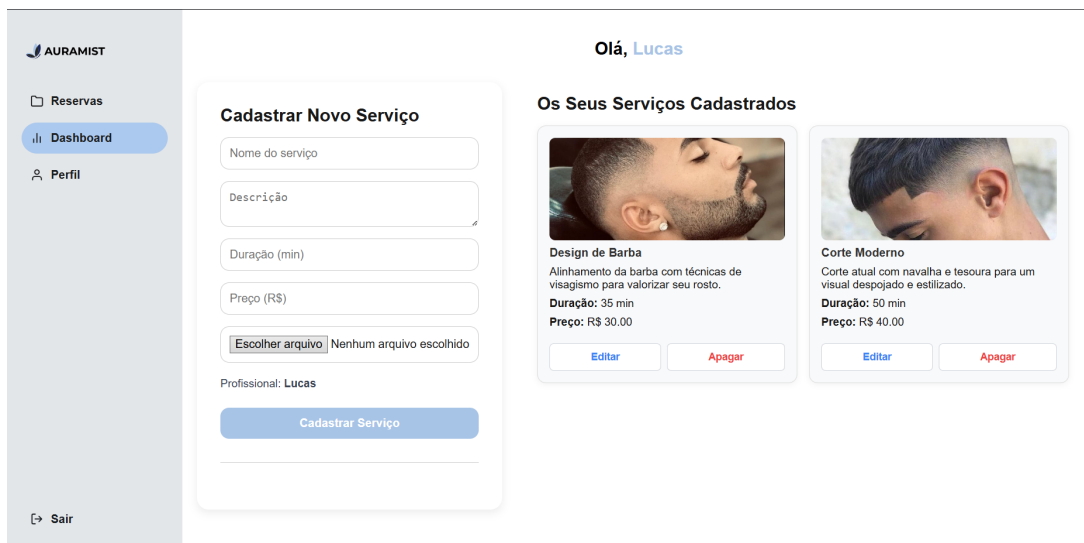


Figura 13 – Painel do profissional para cadastrar e gerenciar seus serviços.

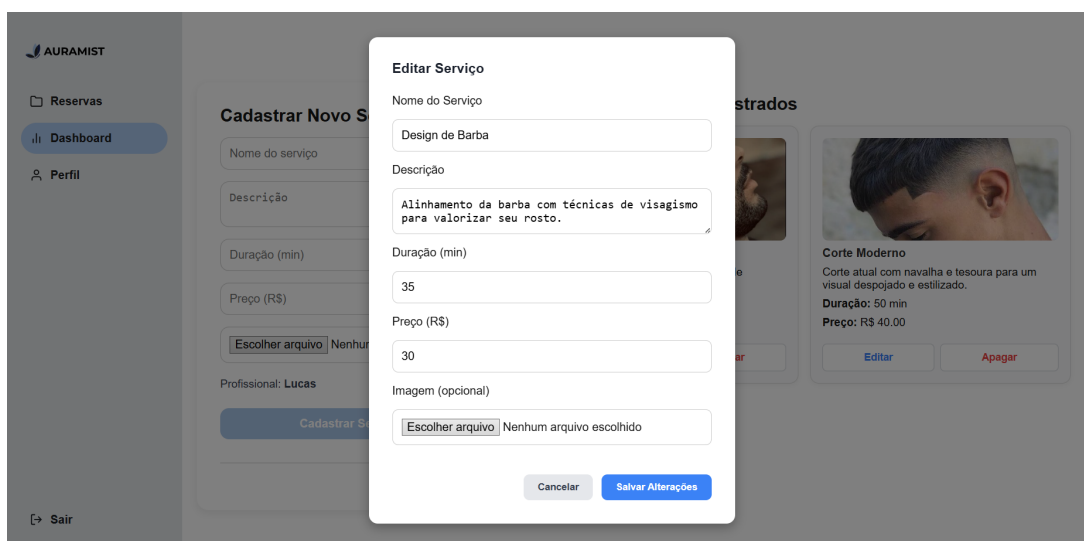


Figura 14 – Modal para edição dos detalhes de um serviço cadastrado.

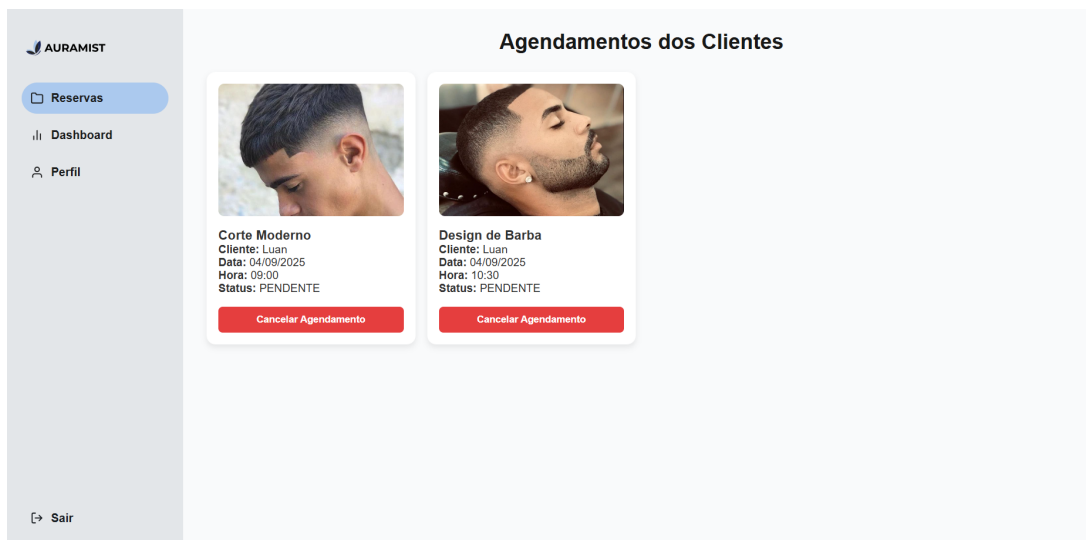


Figura 15 – Tela de visualização dos agendamentos dos clientes para o profissional.

## 5 SOFTWARE

### 5.1 IMPLANTAÇÃO

A implantação do front-end do AuraMist foi realizada na plataforma Vercel<sup>1</sup>, que permite integração contínua com o repositório do projeto. Cada atualização enviada à branch principal do repositório do front-end dispara automaticamente o processo de build, compilando o código React e distribuindo a aplicação para a web. Esse fluxo garante que todas as alterações recentes estejam imediatamente disponíveis para os usuários, mantendo a aplicação atualizada e funcional. A Vercel também oferece escalabilidade automática, garantindo que a plataforma funcione corretamente mesmo quando há aumento no número de acessos simultâneos.

O back-end do AuraMist foi hospedado na plataforma Render<sup>2</sup>, que automatiza o processo de deploy da API de forma integrada ao repositório do GitHub. Sempre que uma nova versão do código é enviada para a branch principal do back-end, o Render realiza a compilação da aplicação Node.js, atualiza a API e reinicia os serviços de forma automática. Essa automação garante que a comunicação entre front-end e back-end permaneça estável e que a aplicação continue disponível para os usuários sem a necessidade de intervenção manual. Além disso, o Render fornece monitoramento de performance e escalabilidade sob demanda, permitindo que o sistema suporte múltiplos usuários e solicitações simultâneas sem comprometer a execução dos agendamentos.

Os repositórios e a aplicação estão disponíveis nos seguintes links:

- Link da aplicação AuraMist: [auramist-client.vercel.app](https://auramist-client.vercel.app)
- Repositório do front-end: <https://github.com/LivyaKelly/auramist-client>
- Repositório do back-end: <https://github.com/LivyaKelly/auramist-server>

### 5.2 TESTES

Durante a fase de testes, todas as funcionalidades implementadas no AuraMist foram avaliadas para garantir o correto funcionamento da plataforma. Foram realizados testes manuais detalhados, abrangendo autenticação de usuários, criação e cancelamento de agendamentos, gerenciamento de serviços e controle de acesso conforme o perfil.

Cada funcionalidade foi verificada quanto ao comportamento esperado, garantindo que os clientes pudessem reservar serviços corretamente e que os profissionais visualizassem os agendamentos associados aos seus serviços. Testes de interface foram realizados em diferentes dispositivos, como desktop, tablet e smartphone, assegurando responsividade e consistência visual.

---

<sup>1</sup> <<https://vercel.com>>

<sup>2</sup> <<https://render.com>>

Além disso, a integração entre front-end e back-end foi validada, conferindo que todas as operações de cadastro, edição e exclusão de serviços, bem como a criação de agendamentos, fossem registradas corretamente no banco de dados e refletidas nas interfaces correspondentes. A Tabela 3 apresenta um resumo detalhado dos testes realizados, incluindo entradas utilizadas, resultados esperados e status obtido.

Tabela 3 – Testes do AuraMist

| <b>Funcionalidade</b>    | <b>Entrada</b>                                  | <b>Resultado Esperado</b>                                             | <b>Status</b> |
|--------------------------|-------------------------------------------------|-----------------------------------------------------------------------|---------------|
| Login                    | Usuário e senha válidos                         | Acesso concedido e redirecionamento à dashboard                       | OK            |
| Login                    | Usuário ou senha inválidos                      | Mensagem de erro e acesso negado                                      | OK            |
| Cadastro de usuário      | Dados completos (nome, e-mail, telefone, senha) | Conta criada com sucesso                                              | OK            |
| Definição de perfil      | Seleção de Cliente ou Profissional              | Perfil definido corretamente no sistema                               | OK            |
| Edição de perfil         | Alteração de dados pessoais                     | Atualização refletida na dashboard                                    | OK            |
| Cadastro de serviço      | Nome, descrição, preço, duração e imagem        | Serviço cadastrado e listado na dashboard do profissional             | OK            |
| Gerenciamento de serviço | Editar serviço existente                        | Alterações refletidas na listagem                                     | OK            |
| Gerenciamento de serviço | Remover serviço existente                       | Serviço removido da listagem                                          | OK            |
| Consulta de serviços     | Acesso à dashboard do cliente                   | Visualização dos serviços cadastrados                                 | OK            |
| Detalhamento de serviço  | Seleção de serviço                              | Exibição completa dos dados do serviço e possibilidade de agendamento | OK            |
| Agendamento de serviço   | Escolha de serviço, data e horário              | Agendamento confirmado, mensagem exibida, registro na dashboard       | OK            |

Continua na próxima página

**Continuação da Tabela 3 – Testes do AuraMist**

| <b>Funcionalidade</b>                       | <b>Entrada</b>                | <b>Resultado Esperado</b>                                             | <b>Status</b> |
|---------------------------------------------|-------------------------------|-----------------------------------------------------------------------|---------------|
| Evitar conflitos de agendamento             | Seleção de horário já ocupado | Mensagem de conflito exibida, cliente deve escolher outro horário     | OK            |
| Cancelamento de agendamento                 | Seleção de agendamento        | Agendamento cancelado, removido da lista do cliente e do profissional | OK            |
| Visualização de agendamentos (cliente)      | Acesso à seção "Agendamentos" | Todos os agendamentos não cancelados são exibidos                     | OK            |
| Visualização de agendamentos (profissional) | Acesso à seção de reservas    | Todos os agendamentos dos serviços cadastrados são exibidos           | OK            |
| Controle de acesso por perfil               | Login com perfil diferente    | Funcionalidades restritas conforme o perfil                           | OK            |
| Logout                                      | Seleção de “Sair”             | Usuário deslogado, redirecionado à tela de login                      | OK            |

A execução desses testes possibilitou a validação do funcionamento das principais funcionalidades do AuraMist, assegurando que os processos de cadastro, agendamento e gerenciamento de serviços operem de forma correta e consistente. Além disso, permitiu identificar e corrigir possíveis inconsistências na comunicação entre front-end e back-end, garantindo que a experiência do usuário seja fluida e responsiva. Com isso, a plataforma encontra-se apta para atender às demandas de clientes e profissionais, proporcionando operações eficientes e contínuas dentro do sistema.



## 6 CONSIDERAÇÕES FINAIS

O presente trabalho partiu da constatação de que, embora existam soluções de agendamento consolidadas, seus planos básicos nem sempre se ajustam à rotina de microempreendedores do setor de beleza. O *AuraMist* foi concebido como uma alternativa leve e objetiva, priorizando clareza de fluxo, autonomia do usuário e implantação simples. A solução adota arquitetura em camadas (front-end em *React/Vercel*; API *Node.js/Express* no *Render*; dados em *PostgreSQL-Neon* com *Prisma ORM*), o que favorece manutenção e evolução.

### 6.1 RESULTADOS ALCANÇADOS

A aplicação foi implantada e os cenários de teste sumarizados na Tabela 3 foram executados com êxito (status *OK*), cobrindo login, cadastro/edição/remoção de serviços, criação e cancelamento de agendamentos, prevenção de conflitos de horário e visualização de reservas por cliente e por profissional. Em produção, os componentes (*Vercel/Render/Neon*) operaram de forma estável; imagens de perfil e de serviços foram persistidas no banco em *Base64* via *Prisma*; e o controle de acesso por perfil atendeu às restrições previstas, demonstrando a viabilidade técnica da proposta.

Como limitações do escopo, destacam-se a ausência de testes automatizados (unitários e de integração) e de ensaios de carga; integrações externas restritas (pagamentos, mensageria oficial via *WhatsApp API*, relatórios analíticos avançados); regras de agenda em nível básico (buffers e políticas ainda pouco parametrizáveis); e a decisão de armazenar imagens em *Base64* no banco, adequada ao porte atual, mas que pode exigir revisão diante de alto volume de dados.

Como desdobramentos futuros, prevê-se a inclusão de notificações de agendamento e lembretes, visão consolidada de histórico com filtros, integrações opcionais (pagamentos, mensageria, relatórios/Bi), testes automatizados com *CI/CD*, ampliação das regras de agenda (buffers por serviço e políticas de cancelamento/remarcação) e avaliação de *object storage* para arquivos. Tais avanços mantêm a solução enxuta e acessível, ao mesmo tempo em que ampliam sua utilidade prática para clientes e profissionais.

## REFERÊNCIAS

JetBrains. **The State of Developer Ecosystem 2024**. 2024. Relatório online. Disponível em: <<https://www.jetbrains.com/lp/devecosystem-2024/>>. Acesso em: 31 ago. 2025.

NOVAIS, R. Análise do impacto da implementação de ferramentas de marketing digital em micro e pequenas empresas. **Research, Society and Development**, v. 13, n. 8, p. e11313846665–e11313846665, 2024.

SEBRAE. **Transformação Digital nos Pequenos Negócios 2023: resumo por segmentos econômicos**. Brasília, 2023. 6. ed. Disponível em: <[https://agenciasebrae.com.br/wp-content/uploads/2023/12/Pesquisa\\_TIC\\_2023-RESUMO\\_segmentos\\_economicos-1.pdf](https://agenciasebrae.com.br/wp-content/uploads/2023/12/Pesquisa_TIC_2023-RESUMO_segmentos_economicos-1.pdf)>. Acesso em: 31 ago. 2025.

SILVA, R. O. da; MARTINS, B. R.; DINIZ, W. G. A complexibilidade da uml e seus diagramas. **Tecnologias em Projeção**, v. 8, n. 1, p. 86–99, 2017.

SOUZA, A. A. de. **O Impacto da Digitalização no Mercado de Trabalho Regional**. Dissertação (Mestrado) — Universidade de Aveiro (Portugal), 2021.