



Fundação Presidente Antônio Carlos
Conselheiro Lafaiete
Engenharia de Computação



**Desenvolvimento de uma aplicação para o controle de
agendamento de uma barbearia.**

Bruno Viana Cotta

Conselheiro Lafaiete, 20 de novembro de 2023

Bruno Viana Cotta

**Desenvolvimento de uma aplicação para o controle de fluxo de uma
barbearia.**

Trabalho de conclusão de curso apresentado
como parte das atividades para obtenção do título
de bacharel em Engenharia de Computação, da
Fundação Presidente Antônio Carlos de
Conselheiro Lafaiete

Orientador: Jean Carlo Mendes

Conselheiro Lafaiete
20 de novembro de 2023

“Insanidade é fazer sempre a mesma coisa e esperar resultados diferentes.”

Albert Einstein

Resumo

Este artigo descreve o desenvolvimento de um sistema web para uma barbearia. O sistema permite que a barbearia controle o fluxo dos clientes. Foram adotadas algumas das tecnologias mais utilizadas atualmente para o desenvolvimento da aplicação. Será abordado as principais etapas do desenvolvimento do sistema, a implementação de recursos de. O artigo conclui com uma avaliação geral do sistema, destacando os principais recursos e benefícios para a barbearia e seus clientes.

Palavras-chaves: Barbearia, JavaScript, Sistema Web, Tecnologia

Lista de ilustrações

| | |
|--|----|
| Figura 1 - Criptografia do JWT..... | 19 |
| Figura 2 - Diagrama entidade relacionamento..... | 22 |
| Figura 3 - Diagrama de caso de uso..... | 23 |
| Figura 4 - Diagrama de sequência: Logar no sistema..... | 24 |
| Figura 5 - Interface de login. | 25 |
| Figura 6 - Diagrama de sequência: Cadastrar no sistema. | 26 |
| Figura 7 - Interface de cadastro. | 27 |
| Figura 8 - Diagrama de sequência: Alterar cadastro. | 28 |
| Figura 9 - Interface de alteração de dados. | 29 |
| Figura 10 - Diagrama de sequência: Deslogar do sistema..... | 30 |
| Figura 11 - Interface com o botão sair da conta. | 30 |
| Figura 12 - Diagrama de sequência: Cadastrar corte..... | 32 |
| Figura 13 - Interface de cadastro do corte..... | 32 |
| Figura 14 - Diagrama de sequência: Editar corte | 34 |
| Figura 15 - Interface de cadastro do corte..... | 34 |
| Figura 16 - Diagrama de sequência: Agendamento de cliente | 36 |
| Figura 17 - Interface de agendamento do cliente | 36 |
| Figura 18 - Diagrama de sequência: Alterar status do corte. | 38 |
| Figura 19 - Interface do botão de alternar status do corte..... | 38 |
| Figura 20 - Diagrama de sequência: Finalizar serviço. | 40 |
| Figura 21 - Modal de finalização de serviço. | 40 |

Lista de abreviaturas e siglas

API – *Application programming Interface*

CC – *Cloud Computing*

CSS - *Cascading Style Sheets*

HTTP - *Hypertext Transfer Protocol*

HTML - *Hypertext markup language*

IDE - *Integrated Development Environment*

JWT - *Json Web Token*

ORM – *Object Relational Mapper*

SGDBs – *Sistema de gerenciamento de Banco de Dados*

TI – *Tecnologia da Informação*

URL - *Uniform Resource Locator*

VS Code - *Visual Studio Code*

SUMÁRIO

| | |
|----------------------------------|----|
| 1 Introdução | 9 |
| 2 Objetivos..... | 10 |
| 2.1 Objetivo Geral | 10 |
| 2.2 Objetivo Específico | 10 |
| 3 metodologias..... | 10 |
| 4 Revisão de literatura | 11 |
| 4.1 NodeJs | 11 |
| 4.2 JavaScript | 11 |
| 4.3 TypeScript | 12 |
| 4.4 Express | 12 |
| 4.5 PostgreSQL..... | 12 |
| 4.6 Prisma | 13 |
| 4.7 Chakra UI | 13 |
| 4.8 React | 14 |
| 4.9 Visual Studio Code..... | 15 |
| 4.10 DataGrip | 15 |
| 4.11 Json Web Token | 16 |
| 4.12 Insomnia | 18 |
| 5 Desenvolvimento | 19 |
| 5.1 Requisitos | 19 |
| 5.1.1 Requisitos Funcionais..... | 19 |

| | |
|--|----|
| 5.2 Diagrama Entidade Relacionamento | 20 |
| 5.3 Caso de Uso | 21 |
| 5.3.1 Diagrama de caso de uso | 21 |
| 5.3.2 Detalhamento de casos de uso | 21 |
| 5.3.2.1 Identificação: UC1..... | 21 |
| 5.3.2.2 Identificação: UC2..... | 23 |
| 5.3.2.3 Identificação: UC3..... | 25 |
| 5.3.2.4 Identificação: UC4..... | 27 |
| 5.3.2.5 Identificação: UC5..... | 29 |
| 5.3.2.6 Identificação: UC6..... | 31 |
| 5.3.2.7 Identificação: UC7..... | 33 |
| 5.3.2.8 Identificação: UC8..... | 35 |
| 5.3.2.9 Identificação: UC9..... | 37 |
| 6 Considerações Finais | 39 |
| 7 Referencias | 40 |

1 Introdução

Com o contínuo desenvolvimento da tecnologia e a crescente necessidade de otimização de processos em diversos setores, a implementação de sistemas web está se tornando cada vez mais comum. O setor de higiene e cuidados pessoais inclui a indústria de barbearia, que atende a uma ampla gama de clientes com diferentes preferências de estilo. No entanto, as técnicas convencionais de gerenciamento manual das barbearias podem produzir ineficiências e problemas operacionais. Portanto, é necessária a criação de um sistema de ponta baseado na Web que possa agilizar com sucesso as operações de gerenciamento e melhorar a eficiência geral [1], [2], [3].

Nesse contexto, este trabalho apresenta um estudo de caso sobre o desenvolvimento de um sistema web para gerenciamento de uma barbearia. O objetivo da implantação desse sistema é proporcionar mais praticidade e agilidade aos proprietários e clientes do negócio. A arquitetura do sistema é projetada para garantir escalabilidade e manutenção do sistema, bem como garantir bom desempenho e segurança dos dados.

Por fim, este trabalho propõe algumas possibilidades de evolução e melhoria do sistema que podem ser implementadas futuramente para tornar a plataforma mais completa e eficiente. Por meio deste estudo de caso, espera-se promover o avanço do conhecimento sobre a utilização de sistemas em rede na indústria da beleza e incentivar a adoção de soluções tecnológicas que melhorem a experiência do cliente e a gestão do negócio.

Para se atingir ao objetivo proposto, este relatório técnico foi organizado em sete sessões, no qual a II sessão apresenta o objetivo geral e específico, a III sessão apresenta a metodologia utilizada, a IV sessão apresenta uma breve revisão bibliográfica acerca do tema e das tecnologias utilizadas, a V sessão apresenta todo

o desenvolvimento proposto, a VI sessão apresenta as conclusões finais e a VII sessão apresenta as referências utilizadas na confecção deste documento.

2 Objetivos

2.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver e avaliar um sistema web para gerenciamento de uma barbearia, a fim de otimizar os processos de gestão e melhorar a experiência do cliente.

2.2 Objetivo Específico

- Modelar o sistema de acordo com as principais demandas identificadas, utilizando boas práticas de programação.
- Desenvolver e implementar o sistema utilizando tecnologias web modernas e eficientes.
- Avaliar o desempenho do sistema em relação aos objetivos propostos e identificar possíveis oportunidades de melhoria.
- Discutir as possibilidades de evolução e aprimoramento do sistema, a fim de torná-lo ainda mais completo e eficiente para as necessidades do cliente.

3 metodologias

Para atingir os objetivos apresentados neste trabalho, foi empregado um estudo bibliográfico com o objetivo de identificar na literatura as características, conceitos e aplicações das ferramentas utilizadas para desenvolver sistemas Web. Palavras-chave primárias foram definidas para investigação bibliográfica no Google Acadêmico. Além da base de conhecimento científico, as tecnologias utilizadas são pesquisadas no portal e na documentação oficial.

A pesquisa bibliográfica sustenta teoricamente o objeto de estudo de forma a contribuir para subsidiar futuras análises dos dados obtidos. Todo estudo deve ser respaldado e apoiado por pesquisa bibliográfica, podendo ser realizado isoladamente ou baseado em pesquisa preparatória de outro tipo.

4 Revisão de literatura

No decorrer desta sessão serão apresentados pequenos estudos exploratórios não exaustivos sobre o conteúdo proposto, além de informações técnicas obtidas de documentos que tratam das tecnologias e padrões utilizados.

4.1 NodeJs

Node.js é um ambiente de tempo de execução JavaScript multiplataforma e de código aberto. O Node.js é construído sobre o mecanismo JavaScript fora do Google Chrome [4].

Aplicações desenvolvidas em Node.js rodam apenas em um Manipula e não cria um novo thread para cada nova solicitação. Além disso, o Node.js usa um modelo de E/S para impedir o bloqueio do código e, em geral, as bibliotecas criadas no Node.js têm o mesmo comportamento sem bloqueio [4].

Essa linguagem foi utilizada para o desenvolvimento do protótipo por ser uma linguagem simples e de fácil implementação [4].

4.2 JavaScript

É uma linguagem de programação do lado do cliente criada por Brendan Eich que permite funcionalidades mais complexas em páginas da web. Normalmente usados para controlar HTML e CSS e manipular o comportamento em uma página. JavaScript é uma linguagem dinâmica multiparadigma baseada em protótipo que suporta estilos orientados a objetos, imperativos e declarativos [5], [6]

O padrão JavaScript é ECMAScript. Todos os navegadores modernos suportam totalmente o ECMAScript 5.1 desde 2012, e novas especificações são lançadas todos os anos [6].

4.3 TypeScript

TypeScript é uma linguagem de programação de código aberto desenvolvida pela Microsoft. A linguagem adiciona digitação estática opcional à linguagem JavaScript, portanto, o código de validação funciona corretamente. Além disso, o TypeScript transpila o código em JavaScript, para que possa ser executado em qualquer ambiente em que o JavaScript seja executado [7].

Essa linguagem é usada para prototipagem porque fornece digitação estática e recursos de programação orientada a objetos que o JavaScript não fornece por padrão [7].

4.4 Express

O Express é um framework mais usado com Node.js, facilitando o desenvolvimento de aplicações *back-end* e até aplicações *full-stack* com um sistema de *templates* [8].

Ele é escrito em JavaScript e é usado por várias empresas em todo o mundo, incluindo Fox Sports, PayPal, IBM, Uber. O Express é muito popular entre grandes empresas e comunidades e facilita o uso de Node e JavaScript juntos para criar aplicativos, tornando esse ecossistema ainda mais forte [8].

4.5 PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados objeto-relacional de uso geral e um dos sistemas de banco de dados de código aberto mais avançados. O PostgreSQL é baseado no POSTGRES 4.2 do Departamento de Ciência da Computação da Universidade da Califórnia, Berkeley. [9]

Ele é projetado para rodar em plataformas semelhantes ao UNIX. No entanto, também foi projetado para ser portátil, de modo que possa ser executado em várias plataformas, como Mac OS X, Solaris e Windows. [9]

O PostgreSQL é um software de código aberto. Seu código-fonte está disponível sob a licença PostgreSQL. Devido à sua estabilidade, é fácil de manter.

Portanto, se você desenvolver aplicativos baseados em PostgreSQL, o custo de desenvolvimento será menor em comparação com outros sistemas de gerenciamento de banco de dados. [9]

4.6 Prisma

O Prisma ORM é uma estrutura de desenvolvimento de banco de dados moderna, poderosa e fácil de usar. Ele fornece uma abordagem de programação orientada a objetos para interagir com bancos de dados relacionais [10].

Uma das principais vantagens do Prisma ORM é que ele fornece um alto nível de segurança ao lidar com operações de banco de dados. Isso ocorre porque o Prisma ORM impõe uma camada de abstração de banco de dados para evitar a injeção de SQL e outras falhas de segurança comuns. Além disso, o Prisma ORM oferece suporte nativo à criptografia e autenticação de dados [10].

Outra vantagem do Prisma ORM é sua facilidade de uso. Possui uma sintaxe simples e intuitiva, o que torna sua implementação muito fácil. Além disso, o Prisma ORM oferece extensa documentação e uma comunidade de desenvolvedores ativa que pode ajudar na solução de problemas e dar suporte ao desenvolvimento de aplicativos [10].

O Prisma ORM é uma estrutura poderosa e fácil de usar que fornece uma abordagem segura, flexível e escalável para o desenvolvimento de bancos de dados relacionais. Com sua sintaxe intuitiva, suporte para vários bancos de dados e recursos avançados, o Prisma ORM é uma excelente opção para desenvolvedores que desejam simplificar o processo de interação com bancos de dados em seus aplicativos [10].

4.7 Chakra UI

Chakra UI é uma biblioteca de componentes de interface do usuário baseada em React, projetada para ajudar os desenvolvedores a criar aplicativos da Web responsivos modernos de maneira fácil e eficiente. Ele fornece uma ampla variedade de componentes prontos para uso, como botões, formulários, menus, tabelas e muito mais. Além disso, Chakra UI é altamente personalizável com opções de estilo e tema

abrangentes, permitindo que os desenvolvedores criem interfaces de usuário visualmente atraentes e consistentes [11].

De acordo com a documentação oficial do Chakra UI, a biblioteca é construída sobre princípios de design de sistema, acessibilidade e usabilidade. Isso significa que a IU do Chakra foi cuidadosamente projetada para que cada componente seja fácil de usar, acessível a todos os usuários e facilmente integrado a outros componentes do sistema. Além disso, a biblioteca é altamente extensível, permitindo que os desenvolvedores adicionem facilmente seus próprios componentes personalizados para estender a funcionalidade existente da biblioteca [11].

O Chakra UI também é amplamente considerado como uma das bibliotecas de componentes de UI mais populares e bem documentadas disponíveis para os desenvolvedores do React atualmente. É de código aberto, mantido pela comunidade, e recebe atualizações regulares, para que a biblioteca esteja sempre atualizada e pronta para uso em projetos de produção [11].

4.8 React

React é uma biblioteca JavaScript desenvolvida pelo Facebook para construir interfaces de usuário declarativas e componentizadas para aplicações web. Ele permite que os desenvolvedores criem aplicativos da Web complexos e de alto desempenho, decompondo-os em componentes reutilizáveis e gerenciando com eficiência o estado do aplicativo. Além disso, o React é altamente escalável, permitindo que os desenvolvedores adicionem facilmente novos recursos e funcionalidades à medida que o aplicativo cresce [12].

Um dos principais pontos fortes do React é sua abordagem declarativa para a construção de interfaces de usuário. Em vez de escrever código para manipular diretamente o DOM, os desenvolvedores definem a estrutura da interface do usuário em termos de componentes, propriedades e estado. Isso torna o código mais fácil de entender e manter, reduz a possibilidade de erros e melhora a eficiência do desenvolvimento [12].

Outra vantagem do React é sua capacidade de renderizar aplicativos no lado do servidor. Isso permite que os desenvolvedores criem aplicativos da Web que carregam rapidamente e funcionam bem em dispositivos com recursos limitados. Além

disso, o React possui uma ampla gama de ferramentas e extensões disponíveis para aumentar ainda mais sua funcionalidade e personalização [12].

O React também é amplamente considerado como uma das bibliotecas JavaScript mais populares e bem documentadas atualmente disponíveis para desenvolvedores da web. É de código aberto, mantido pela comunidade e recebe atualizações regulares, garantindo que a biblioteca esteja sempre atualizada e pronta para uso em projetos de produção [12].

4.9 Visual Studio Code

O Visual Studio Code (VS Code) é uma ferramenta de desenvolvimento integrado (IDE) bastante utilizada para criar aplicativos web. Desenvolvido pela Microsoft, é um aplicativo multiplataforma de código aberto, compatível com Windows, macOS e Linux. Possui uma interface de usuário intuitiva e personalizável com um layout de fácil navegação. Além disso, suporta múltiplas linguagens de programação como JavaScript, TypeScript, Python e C++, tornando-o versátil no desenvolvimento web [13].

Um dos recursos mais poderosos do VS Code é sua extensibilidade. Com a extensa biblioteca de extensões disponíveis, você pode adicionar recursos adicionais, como suporte à depuração, controle de versão, integração com estruturas populares e trechos de código [13].

O VS Code possui um terminal integrado que permite executar comandos diretamente da interface do IDE. Isso facilita a execução de tarefas de compilação, teste e gerenciamento de pacotes sem alternar entre o IDE e a linha de comando. Como é uma ferramenta desenvolvida pela Microsoft, o VS Code possui integrações nativas com produtos e serviços de outras empresas, como Azure, Visual Studio e GitHub. Isso facilita a implantação de aplicativos da Web e a colaboração em equipe [13].

4.10 DataGrip

DataGrip é uma IDE para desenvolvimento de banco de dados. Desenvolvido pela JetBrains, o DataGrip é utilizado por desenvolvedores de sistemas web devido aos seus recursos avançados e suporte a diversos tipos de bancos de dados [14].

O software possui uma interface intuitiva e permite aos desenvolvedores trabalhar com diferentes sistemas de gerenciamento de banco de dados (SGDBs) como MySQL, PostgreSQL, Oracle, Microsoft SQL Server e SQLite em um único ambiente. Por meio dessa interface, você pode visualizar e manipular com eficiência estruturas de banco de dados, tabelas, esquemas e relacionamentos [14].

Uma das principais características é seu poderoso editor de consultas. Com recursos como realce de sintaxe, preenchimento automático, formatação de código e suporte a várias guias, é fácil escrever e executar consultas complexas. Além disso, o software também fornece funções de exportação de dados em diferentes formatos, que podem atender com flexibilidade às necessidades dos desenvolvedores [14].

4.11 Json Web Token

JSON Web Token (JWT) é um padrão aberto amplamente utilizado para autenticação e autorização em sistemas web e serviços *RESTful*. Ele oferece suporte à transferência segura de informações entre as partes interessadas em um formato compacto e independente baseado em JSON [15].

Um *JWT* consiste em três partes: cabeçalho, carga útil e assinatura. O cabeçalho contém informações sobre o tipo de token e o algoritmo de assinatura usado. A carga útil armazena informações adicionais, como identificadores de usuário e metadados relacionados. As assinaturas são geradas usando uma chave secreta do servidor para garantir a integridade dos dados [15].

Na autenticação, quando o usuário faz login com suas credenciais, o servidor gera um *JWT* e o envia de volta ao cliente. Esse token é incluído no cabeçalho de autorização em todas as solicitações subsequentes. O servidor verifica a autenticidade do token e concede acesso ao recurso protegido [15].

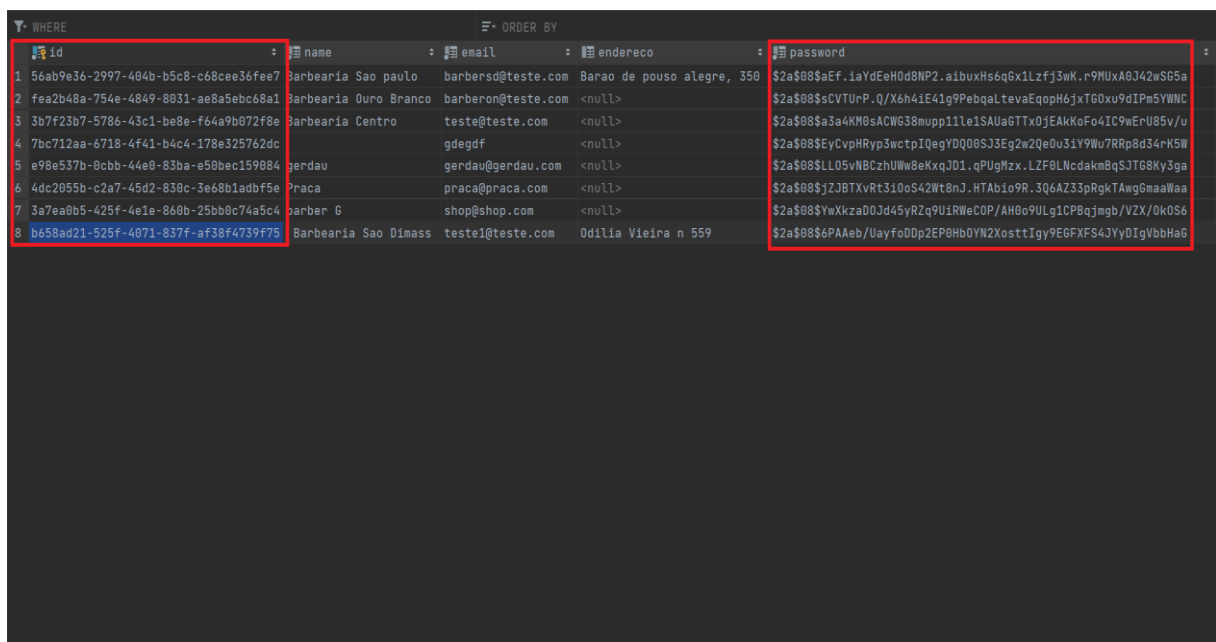
Além da autenticação, os *JWT* também podem conter informações sobre funções e permissões do usuário. Esses dados podem ser adicionados à carga útil do token e usados pelo servidor para autorizar o acesso aos recursos com base em regras definidas [15].

Uma das vantagens do *JWT* é sua estrutura compacta, que permite que seja facilmente transmitido em *URLs*, cabeçalhos *HTTP* ou como parâmetros em requisições *POST*. Além disso, os *JWT* são independentes, o que significa que todas

as informações necessárias para verificar a autenticidade e validade do token estão contidas nele, eliminando a necessidade de consultar o servidor para cada solicitação [15].

A segurança do *JWT* depende do algoritmo de assinatura usado. É fundamental escolher um algoritmo robusto e proteger a chave secreta do servidor. Recomenda-se usar uma conexão segura HTTPS para evitar ataques de espionagem e manipulação de token [15].

Os benefícios fornecidos pelo *JWT* incluem escalabilidade do sistema, dependência reduzida de servidores de autenticação e descentralização da autenticação. No entanto, é importante entender as práticas recomendadas de segurança e considerar os casos de uso específicos ao implementar *JWT* em sistemas de desenvolvimento web. A fig 1 demonstra o token de criptografia do usuário e senha [15].



| id | name | email | endereco | password |
|---|-----------------------|--------------------|----------------------------|---|
| 1 56ab9e36-2997-404b-b5c8-c68cee36fee7 | Barbearia Sao paulo | barbersd@teste.com | Barao de pouso alegre, 350 | \$2a\$08\$aEf.iaYdEeH0d8NP2.aibuxHs6qX1Lzfj3wK.r9MUxA8J42wSG5a |
| 2 fea2b48a-754e-4849-8031-ae8a5ebc68a1 | Barbearia Ouro Branco | barberon@teste.com | <null> | \$2a\$08\$sCVTU+P.Q/X6h41E41g9PebqaLtevaEqopH6jxTGDxu9dIPm5YWNc |
| 3 3b7f23b7-5786-43c1-be8e-f64a9b072f8e | Barbearia Centro | teste@teste.com | <null> | \$2a\$08\$a3a4KM8sACW638mupp11le1SAUa6TTx0jEAKkoFo4IC9wErU85v/u |
| 4 7bc712aa-6718-4f41-b4c4-178e325762dc | | gdegdf | <null> | \$2a\$08\$EyCvpHRyp3wctplQegYDQ08SJ3Eg2w2Qe0u3iY9Wu7RRp8d34rK5W |
| 5 e98e537b-0cbb-44e0-83ba-e50bec159084 | gerdau | gerdau@gerdau.com | <null> | \$2a\$08\$LL05vN8CzhUWw8eKxqJD1.qPUgMzx.LZF0LNdakmBqSJTG8Ky3ga |
| 6 4dc2055b-c2a7-45d2-830c-3e68b1adbff5e | Praca | praca@praca.com | <null> | \$2a\$08\$jZJ8TXvRt310oS42Wt8nJ.HTAbio9R.3Q6AZ33pRgkTAWgGmaaWaa |
| 7 3a7ea8b5-425f-4e1e-860b-25bb8c74a5c4 | barber G | shop@shop.com | <null> | \$2a\$08\$YmXkzaD0Jd45yRzq9UiRWeCOP/AH0o9ULg1CPBqjmgB/VZX/0k0S6 |
| 8 b658ad21-525f-4071-837f-af38f4739f75 | Barbearia Sao Dimass | teste1@teste.com | Odilia Vieira n 559 | \$2a\$08\$6PAAb/UayfoDDp2EPHbDYN2XosttIgy9EGFXS4JYyDIgVbbHa6 |

Fig. 1. Criptografia do JWT.

4.12 Insomnia

Insomnia é uma ferramenta de desenvolvimento que simplifica o processo de teste e depuração de APIs. Ele fornece uma interface de usuário amigável e intuitiva, permitindo que os desenvolvedores enviem facilmente solicitações HTTP e visualizem as respostas. Usando este software, você pode criar e executar solicitações HTTP personalizadas. Você pode definir métodos, adicionar cabeçalhos, parâmetros de consulta e corpo da solicitação. A ferramenta também fornece recursos avançados, como autenticação e suporte a cookies [16].

Um recurso útil é o gerenciamento de ambiente, que permite definir e alternar entre diferentes configurações, como ambientes de desenvolvimento, teste e produção. Isso facilita o teste em diferentes configurações e cenários [16].

A ferramenta suporta extensões e plugins, oferecendo a possibilidade de estender sua funcionalidade e integrar com outras ferramentas. Além disso, permite compartilhar projetos e ambientes com sua equipe, facilitando a colaboração. Insomnia é uma ferramenta gratuita e de código aberto disponível para Windows, macOS e Linux. Sua abordagem focada em testar e depurar APIs o torna uma escolha popular entre desenvolvedores e equipes de desenvolvimento web [16].

4.13 Luxon

A biblioteca Luxon é uma ferramenta muito valiosa no ambiente de programação JavaScript, principalmente quando se trata de datas e horas. Ele foi cuidadosamente projetado para simplificar as operações que envolvem tempo, fornecendo uma alternativa moderna e eficiente à biblioteca de data nativa do JavaScript.

A principal vantagem do Luxon é sua abordagem orientada a objetos, que fornece manipulação de data e hora mais intuitiva e legível. Com classes como *DateTime*, *Duration* e *Interval*, a biblioteca fornece uma compreensão mais clara e estruturada desses elementos temporais [17].

5 Desenvolvimento

Esta seção visa detalhar as estratégias adotadas no desenvolvimento do sistema, como arquitetura, algoritmos, levantamento de requisitos, modelagem de banco de dados, diagramas de casos de uso e seus detalhes.

5.1 Requisitos

5.1.1 Requisitos Funcionais

- O sistema deve permitir que o usuário cadastrado realize *login*.
- O sistema deve permitir o cadastro do usuário.
- O sistema deve permitir que o usuário altere seus dados.
- O sistema deve permitir que o usuário realize *logout*.
- O sistema deve permitir que o usuário troque o plano.
- O sistema deve permitir que o usuário cadastre um novo corte.
- O sistema deve permitir que o usuário edite o corte.
- O sistema deve permitir que o usuário ative o corte.
- O sistema deve permitir que o usuário desative o corte.
- O sistema deve permitir que o usuário marque o cliente.
- O sistema deve permitir que o usuário encerre o serviço.
- O sistema deve impedir o acesso de usuários não cadastrados.
- Somente cortes ativos poderão ser atribuídos no serviço.
- O sistema deve impedir usuários do plano básico cadastrarem mais de 3 cortes.
- O sistema deve impedir o cadastro da mesma barbearia mais de uma vez.
- Somente usuários cadastrados podem alterar os dados.
- O sistema deve impedir o cadastro do mesmo corte mais de uma vez.
- Somente cortes cadastrados poderão ser atribuídos no serviço.

5.2 Diagrama Entidade Relacionamento

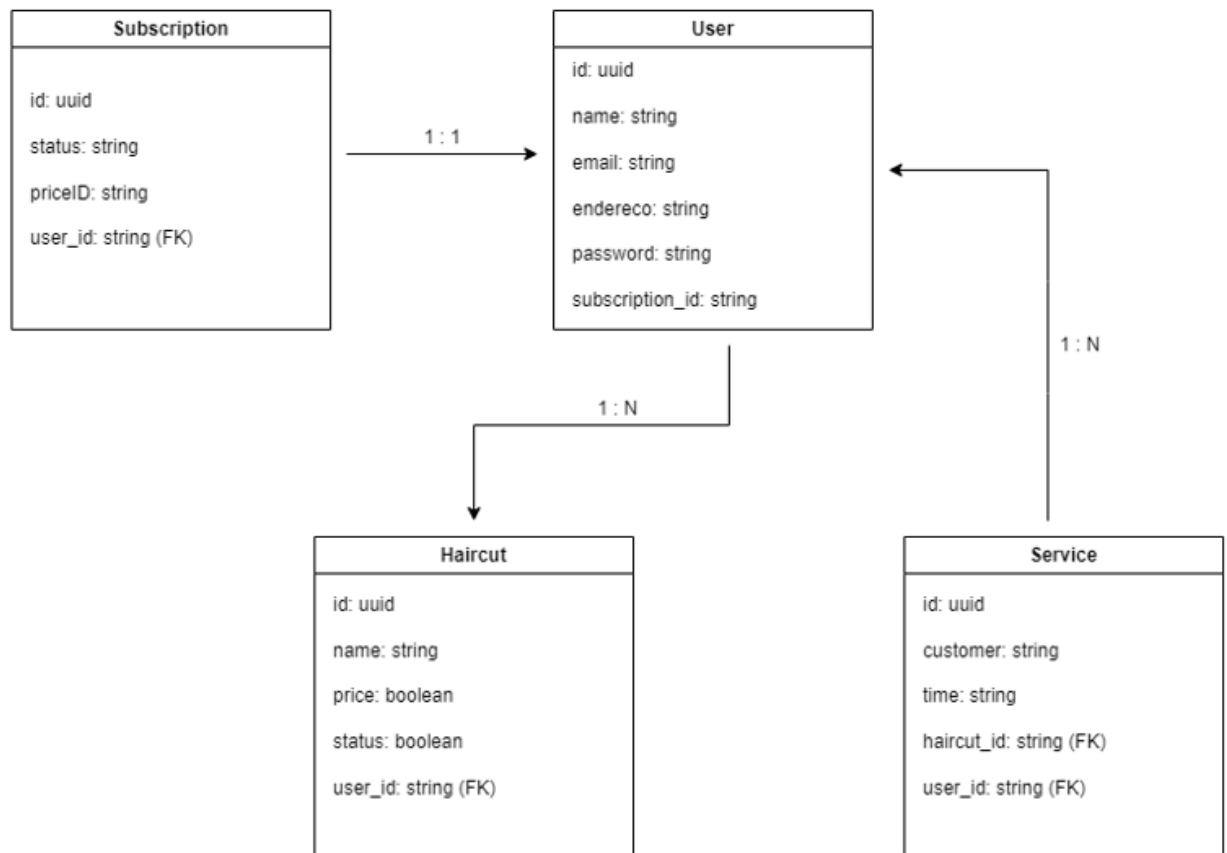


Fig. 2. Diagrama Entidade Relacionamento

5.3 Caso de Uso

5.3.1 Diagrama de caso de uso

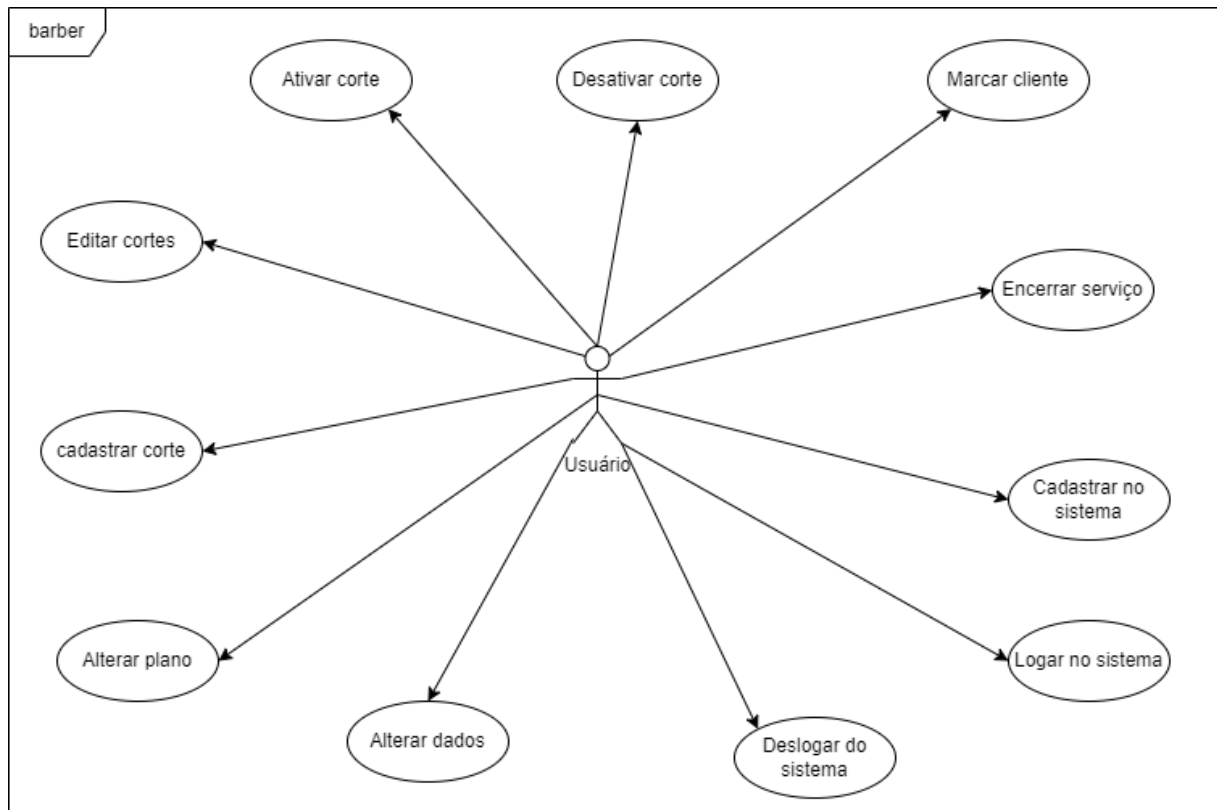


Fig. 3. Diagrama de caso de uso

5.3.2 Detalhamento de casos de uso

5.3.2.1 Identificação: UC1

- **Caso de uso:** Logar no sistema;
- **ATOR:** Usuário;
- **Objetivo:** Realizar a autenticação do usuário no sistema;
- **Descrição:** Caso de uso responsável por receber e validar o e-mail e senha de autenticação do usuário;
- **PRÉ CONDIÇÃO:** O usuário estar cadastrado no sistema;
- **PÓS CONDIÇÃO:** Redirecionar para página principal;
 - **Seção Principal:**

1. O sistema apresenta a interface de login.
2. O usuário preenche o formulário informando e-mail e senha;
3. O sistema confere se o usuário preencheu todos os campos corretamente;
4. O Web Tokens válida o conteúdo do token e confirma autenticidade do usuário;
5. O sistema redireciona o usuário para a página principal;
6. O caso de uso é encerrado.

▪ **Fluxo Alternativo:**

1. A validação do Token encontra algum problema;
2. O sistema envia mensagem de erro;
3. O caso de uso é encerrado.

▪ **Requisitos não Funcionais:**

- Somente deve impedir o acesso de usuários não cadastrados.

▪ **Diagrama de sequência:**

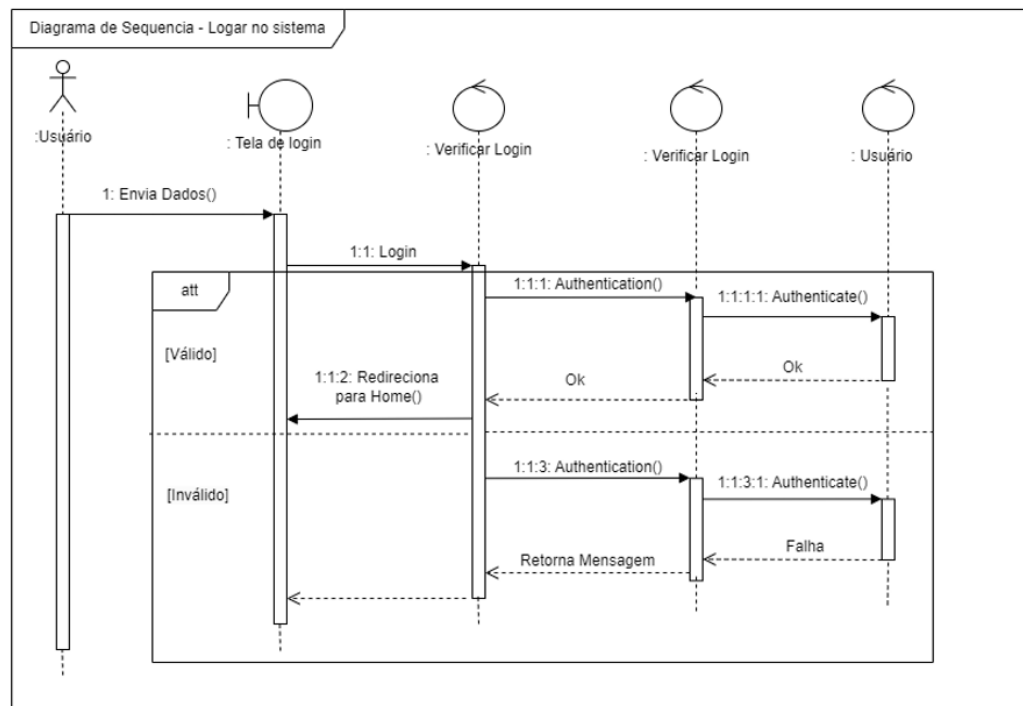


Fig. 4. Diagrama de sequência: Logar no sistema.

- **Requisito da tela:**

A fig. 4 apresenta a interface com o formulário de login.

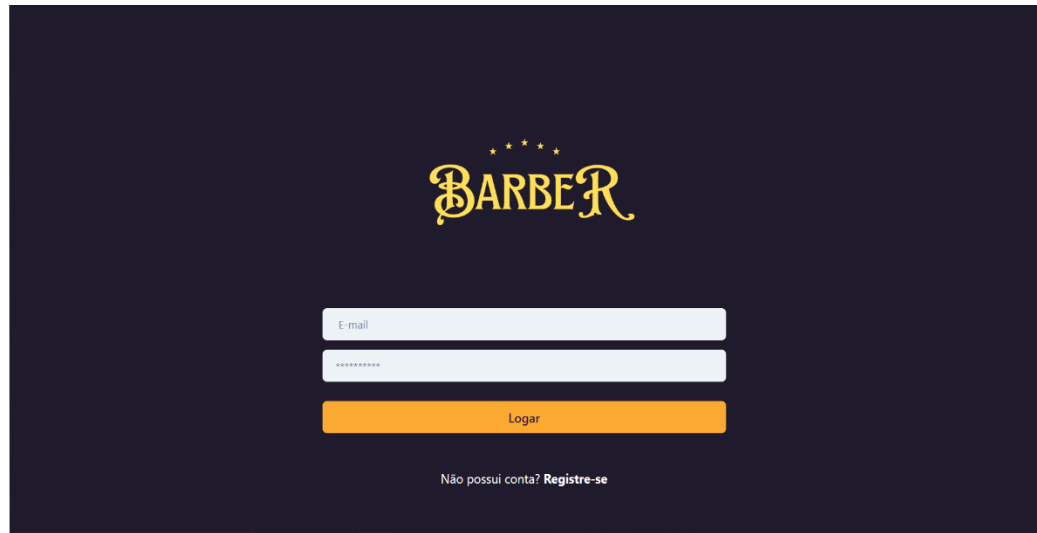


Fig. 5. Interface de login.

5.3.2.2 Identificação: UC2

- **Caso de uso:** Cadastrar no sistema;
- **ATOR:** Usuário;
- **Objetivo:** Cadastrar um novo usuário no sistema;
- **Descrição:** Caso de uso responsável por registrar uma nova conta de usuário;
- **PRÉ CONDIÇÃO:** Nenhuma;
- **PÓS CONDIÇÃO:** Redirecionar para página de login;
 - **Seção Principal:**
 1. O sistema apresenta a interface de cadastro.
 2. O usuário preenche o formulário apresentando nome da barbearia e-mail se senha;
 3. O sistema confere se os dados foram preenchidos corretamente;
 4. O sistema registra a conta de usuário e atribui um token válido de autenticação;

5. O sistema cadastra a entidade User no banco de dados;
6. O sistema exibe mensagem de barbearia cadastrada;
7. O sistema redireciona o usuário para a página de login;
8. O caso de uso é encerrado.

▪ **Fluxo Alternativo:**

1. A validação encontra algum problema;
2. O sistema aborta o cadastro;
3. O sistema apresenta a tela de cadastro novamente;
4. O caso de uso é encerrado.

▪ **Requisitos não Funcionais:**

- O sistema deve impedir o cadastro da mesma barbearia mais de uma vez.

▪ **Diagrama de sequência:**

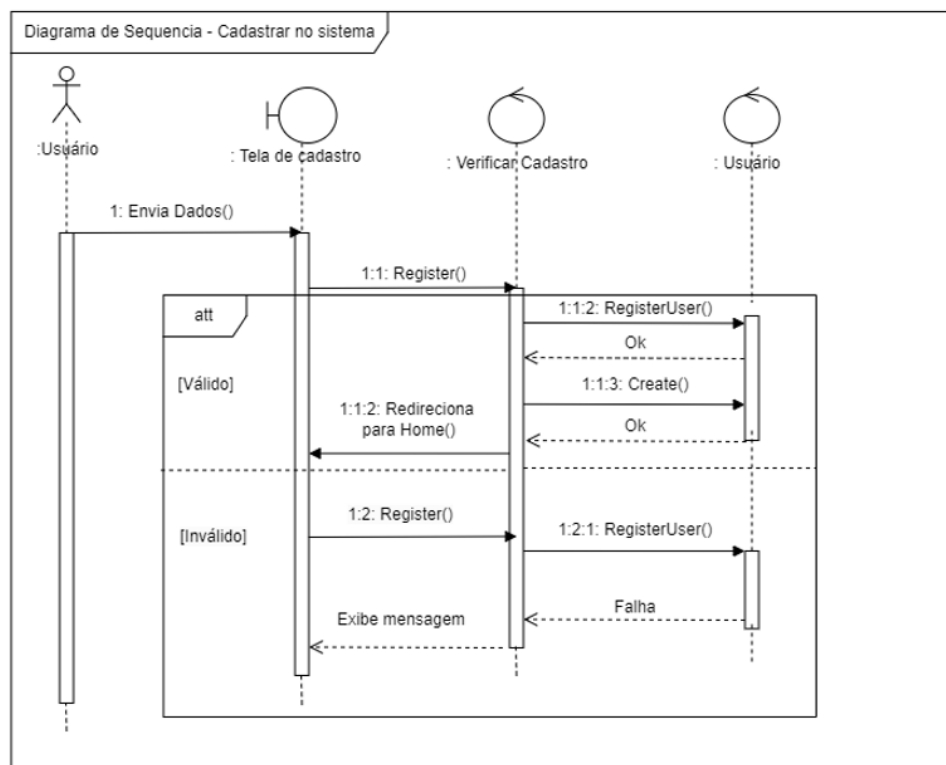


Fig. 6. Diagrama de sequência: Cadastrar no sistema.

▪ **Requisito da tela:**

A fig. 6 apresenta a interface com o formulário de cadastro.

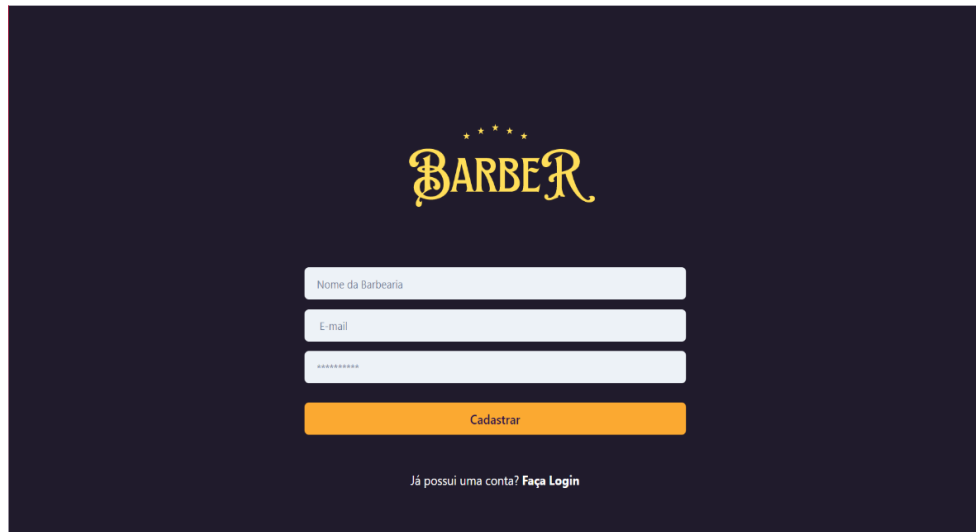


Fig. 7. Interface de cadastro.

5.3.2.3 Identificação: UC3

- **Caso de uso:** Alterar cadastro;
- **ATOR:** Usuário;
- **Objetivo:** Alterar as informações da conta do usuário;
- **Descrição:** Caso de uso responsável por alterar nome e endereço da barbearia;
- **PRÉ CONDIÇÃO:** Estar autenticado no sistema;
- **PÓS CONDIÇÃO:** Exibir mensagem de dados alterados;
 - **Seção Principal:**
 1. O sistema busca as informações do usuário autenticado.
 2. O sistema preenche e apresenta a interface de alterar dados;
 3. O usuário altera as informações;
 4. O sistema verifica se o usuário preencheu todas as informações corretamente;
 5. O sistema altera as informações do usuário no banco de dados;
 6. O sistema exibe a mensagem de dados alterados;

7. O caso de uso é encerrado.

▪ **Fluxo Alternativo:**

8. A validação encontra algum problema;
9. O sistema aborta a alteração de dados;
10. O caso de uso é encerrado.

▪ **Requisitos não Funcionais:**

- Somente usuários cadastrados podem alterar os dados.

▪ **Diagrama de sequência:**

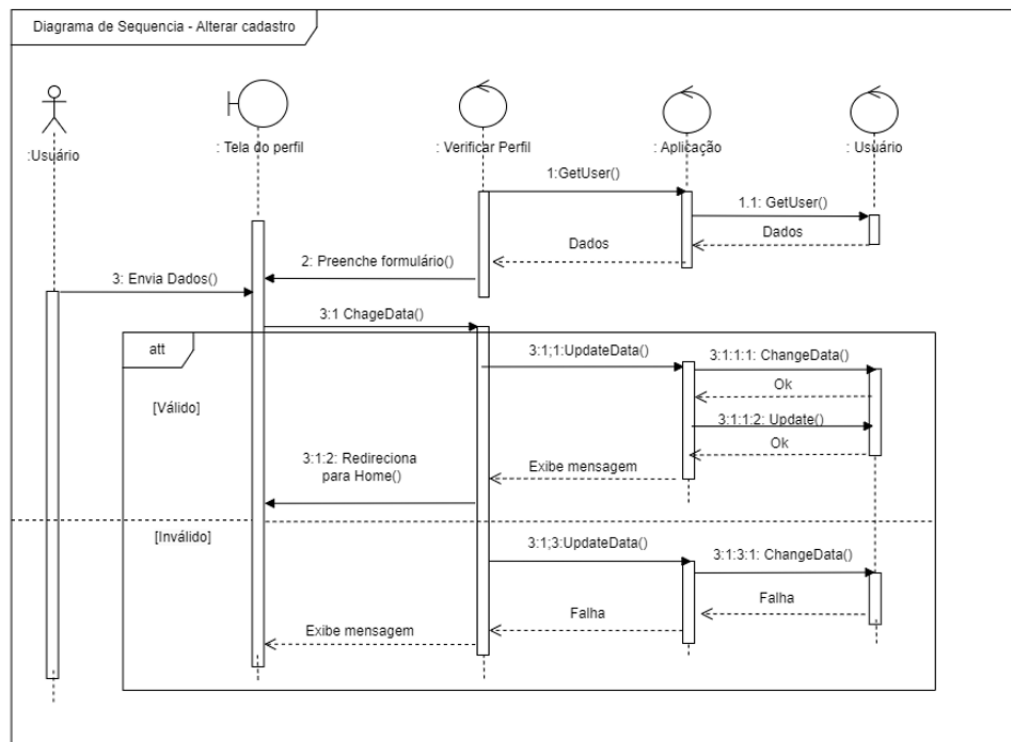


Fig. 8. Diagrama de sequência: Alterar cadastro.

▪ **Requisito da tela:**

A fig. 8 apresenta a interface com o formulário de alteração de dados.

Fig. 9. Interface de alteração de dados.

5.3.2.4 Identificação: UC4

- **Caso de uso:** Desconectar do sistema;
- **ATOR:** Usuário;
- **Objetivo:** Desconectar usuário atual do sistema;
- **Descrição:** Caso de uso responsável por desconectar o usuário do sistema;
- **PRÉ CONDIÇÃO:** Estar autenticado no sistema;
- **PÓS CONDIÇÃO:** Redirecionar para tela de *login*;
- **Seção Principal:**
 1. O usuário acessa a opção meu perfil na *sidebar*.
 2. O usuário clica no botão sair;
 3. O sistema desconecta o usuário da sessão atual.
 4. O sistema redireciona para tela de *login*;
 5. O caso de uso é encerrado.
- **Fluxo Alternativo:** Não se aplica.
- **Diagrama de sequência:**

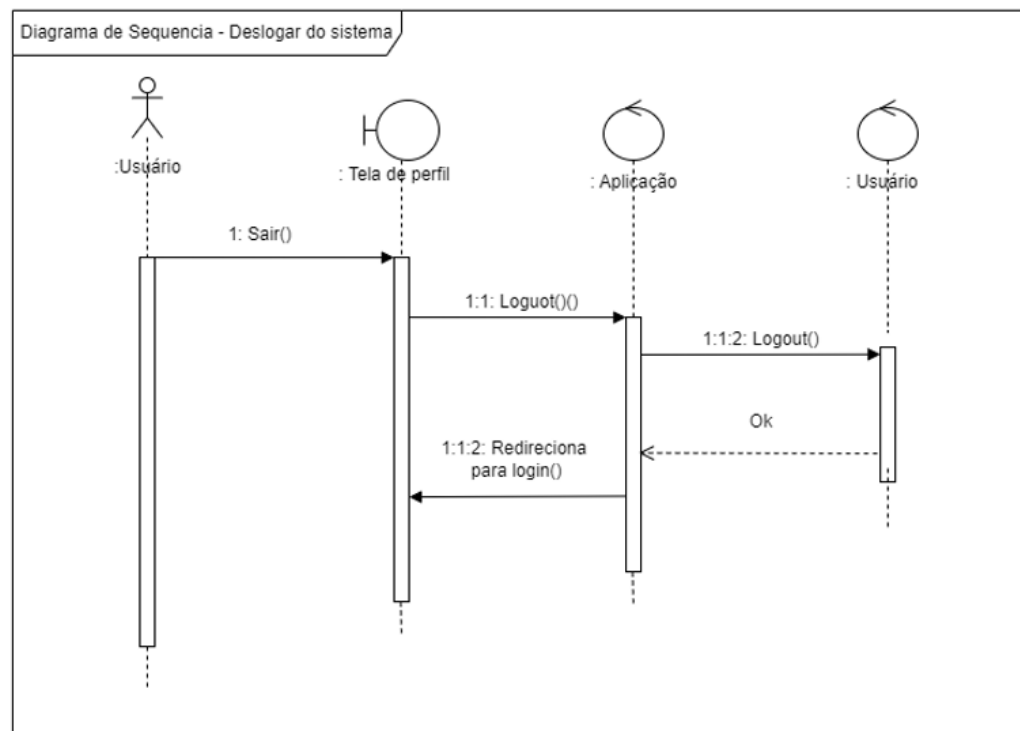


Fig. 10. Diagrama de sequência: Deslogar do sistema.

▪ **Requisito da tela:**

A fig. 10 apresenta a interface com o botão de sair da conta.

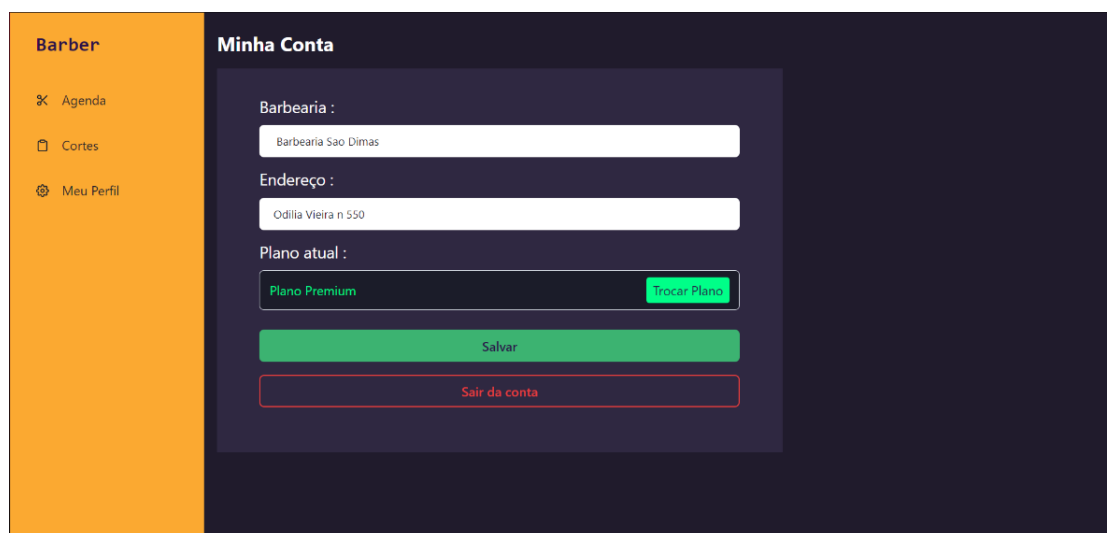


Fig. 11. Interface com o botão sair da conta.

5.3.2.5 Identificação: UC5

- **Caso de uso:** Cadastrar corte;
- **ATOR:** Usuário;
- **Objetivo:** Cadastrar um novo corte no sistema;
- **Descrição:** Caso de uso responsável por registrar um novo corte para utilização do cliente;
- **PRÉ CONDIÇÃO:** Estar autenticado no sistema;
- **PÓS CONDIÇÃO:** Redirecionar para página modelo de cortes;
 - **Seção Principal:**
 1. O usuário seleciona a aba cortes na *sidebar*.
 2. O usuário clica no botão cadastrar;
 3. O sistema redirecionar o usuário para aba cadastrar modelo;
 4. O usuário preenche o formulário com nome e valor do corte;
 5. O sistema verifica se o usuário preencheu todos os campos corretamente;
 6. O sistema cadastra o corte;
 7. O sistema redireciona o usuário para pagina de modelos.
 8. O caso de uso é encerrado.
 - **Fluxo Alternativo:**
 1. A validação encontra algum problema;
 2. O sistema aborta o cadastro;
 3. O sistema apresenta a tela de cadastro novamente;
 4. O caso de uso é encerrado.
 - **Requisitos não Funcionais:**
 - O sistema deve impedir o cadastro do mesmo corte mais de uma vez.
 - **Diagrama de sequência:**

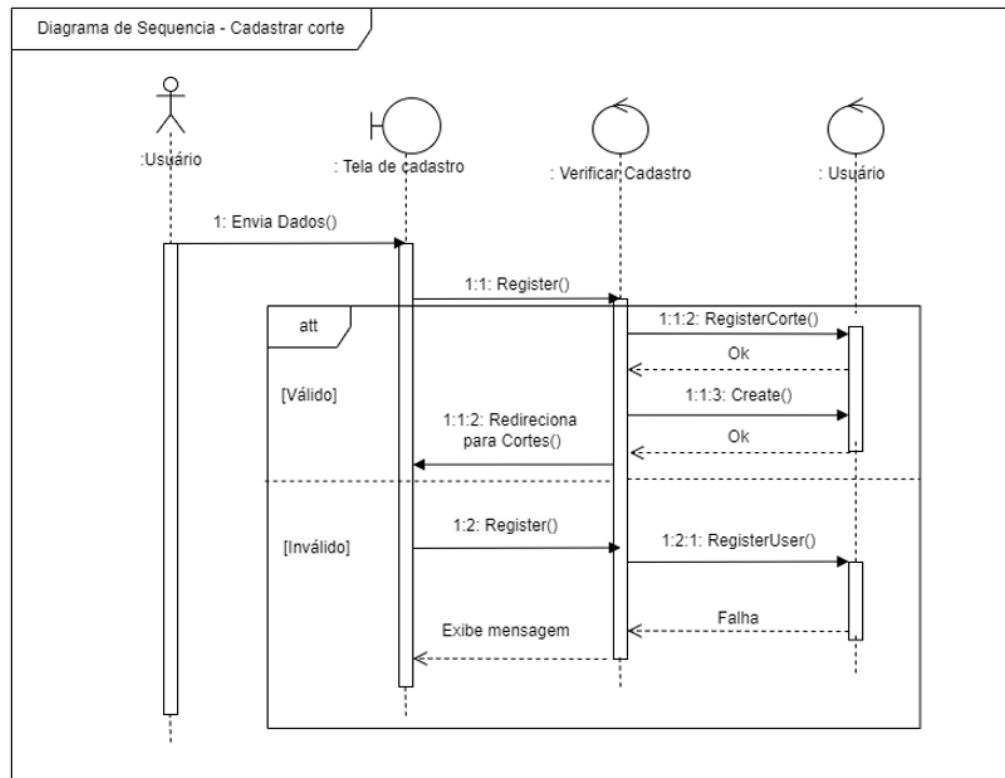


Fig. 12. Diagrama de sequência: Cadastrar corte.

▪ **Requisito da tela:**

A fig. 12 apresenta a interface com o formulário de cadastro do corte.

The screenshot shows the 'Cadastrar modelo' form in the Barber application. The form has two input fields: 'Nome do corte' (Haircut name) and 'Valor do corte R\$ 30.00' (Haircut value). A green 'Salvar' (Save) button is at the bottom. The form is titled 'Cadastrar modelo' and has a back button labeled '<< Voltar'. The sidebar on the left shows the 'Barber' logo and navigation links for 'Agenda', 'Cortes', and 'Meu Perfil'.

Fig. 13. Interface de cadastro do corte.

5.3.2.6 Identificação: UC6

- **Caso de uso:** Editar corte;
- **ATOR:** Usuário;
- **Objetivo:** Alterar as informações do corte cadastrado;
- **Descrição:** Caso de uso responsável por alterar nome e valor do corte cadastrado na barbearia;
- **PRÉ CONDIÇÃO:** Estar autenticado no sistema;
- **PÓS CONDIÇÃO:** Exibir mensagem de dados alterados;
 - **Seção Principal:**
 1. O sistema busca as informações do usuário autenticado.
 2. O sistema preenche e apresenta a interface de alterar dados;
 3. O usuário altera as informações;
 4. O sistema verifica se o usuário preencheu todas as informações corretamente;
 5. O sistema altera as informações do usuário no banco de dados;
 6. O sistema exibe a mensagem de dados alterados;
 7. O sistema redireciona o usuário para pagina modelos de cortes;
 8. O caso de uso é encerrado.
 - **Fluxo Alternativo:**
 1. A validação encontra algum problema;
 2. O sistema aborta a alteração de dados;
 3. O caso de uso é encerrado.
 - **Requisitos não Funcionais:**
 - Somente usuários cadastrados podem alterar os dados.
 - **Diagrama de sequência:**

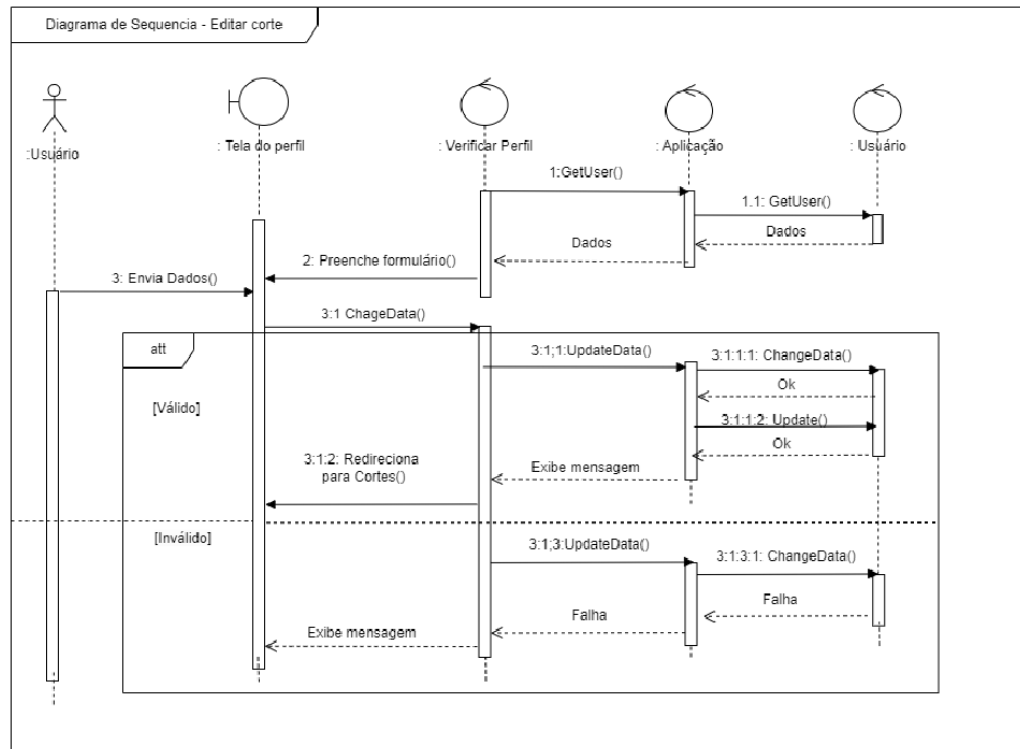


Fig. 14. Diagrama de sequência: Editar corte.

▪ **Requisito da tela:**

A fig. 14 apresenta a interface com o formulário de alteração de dados do corte.

The screenshot shows the 'Barber' app interface. On the left is an orange sidebar with navigation options: 'Agenda', 'Cortes', and 'Meu Perfil'. The main area has a dark blue background. At the top, there's a header with '< Voltar' and 'Corte'. Below this is the title 'Editar corte'. The form contains two input fields: 'Hidratação' (with a value of 120.99) and a toggle switch for 'Desativar corte'. At the bottom is a green 'Salvar' button.

Fig. 15. Interface de cadastro do corte.

5.3.2.7 Identificação: UC7

- **Caso de uso:** Agendar cliente;
- **ATOR:** Usuário;
- **Objetivo:** Marcar o cliente na fila da barbearia;
- **Descrição:** Caso de uso responsável agendar o cliente na barbearia;
- **PRÉ CONDIÇÃO:** Estar autenticado no sistema;
- **PÓS CONDIÇÃO:** Exibir mensagem de cliente agendado;
 1. O usuário seleciona a aba agenda na *sidebar*.
 2. O usuário clica no botão marcar;
 3. O sistema redireciona o usuário para aba marcar horário;
 4. O usuário preenche o formulário com nome seleciona o serviço desejado pelo cliente;
 5. O sistema verifica se o usuário preencheu todos os campos corretamente;
 6. O sistema agenda o cliente;
 7. O sistema exibe o alerta de cliente agendado.
 8. O Sistema redireciona para pagina da agenda.
 9. O caso de uso é encerrado.
- **Fluxo Alternativo:**
 5. A validação encontra algum problema;
 6. O sistema aborta o cadastro;
 7. O sistema apresenta a tela de cadastro novamente;
 8. O caso de uso é encerrado.
- **Requisitos não Funcionais:**
 - Somente cortes ativos poderão ser atribuídos no serviço.
 - Somente cortes cadastrados poderão ser atribuídos no serviço.
- **Diagrama de sequência:**

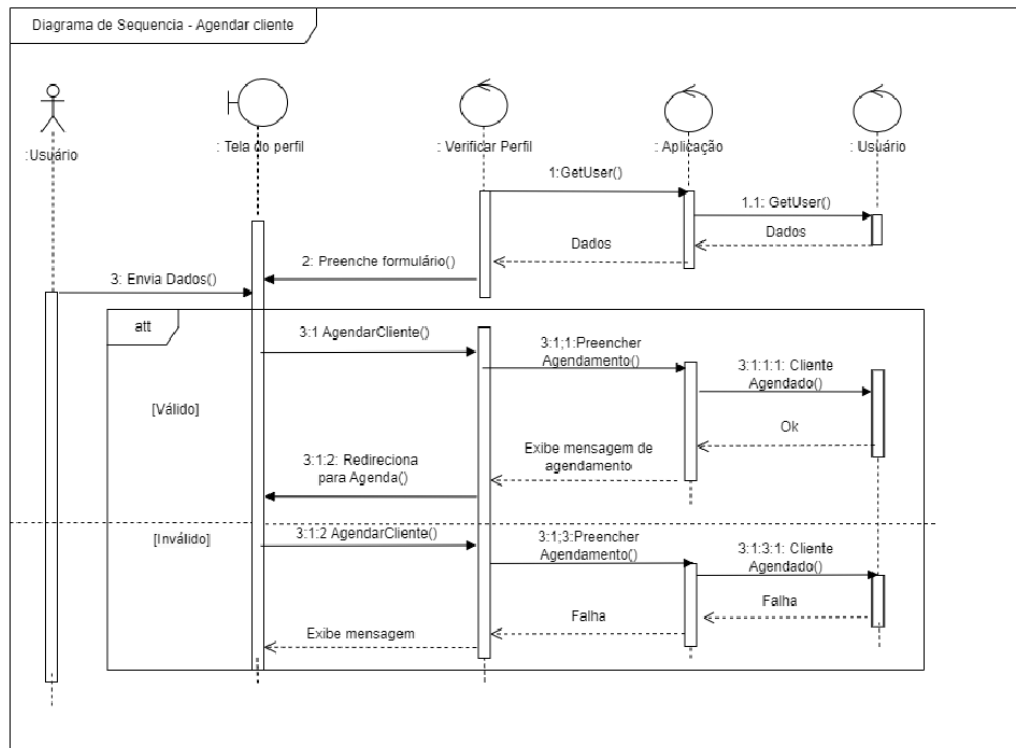


Fig. 16. Diagrama de sequência: Agendamento de cliente.

▪ **Requisito da tela:**

A fig. 16 apresenta a interface com o formulário de agendamento do cliente.

Barber

- Agenda
- Cortes
- Meu Perfil

Marcar Horário

Nome do cliente

Corte Simples

8:00

Salvar

Fig. 17. Interface de agendamento do cliente.

5.3.2.8 Identificação: UC8

- **Caso de uso:** Alterar *status* do corte;
- **ATOR:** Usuário;
- **Objetivo:** Desativar ou ativar o corte cadastrado;
- **Descrição:** Caso de uso responsável por alternar o *status* do corte para utilização;
- **PRÉ CONDIÇÃO:** Estar autenticado no sistema;
- **PÓS CONDIÇÃO:** Exibir mensagem de dados alterados;
 - **Seção Principal:**
 1. O sistema busca as informações do usuário autenticado.
 2. O sistema preenche e apresenta a interface de alterar dados;
 3. O usuário altera os *status* do corte;
 4. O sistema exibe a mensagem de dados alterados;
 5. O sistema altera as informações do corte no banco de dados;
 6. O usuário confirma as alterações;
 7. O sistema redireciona o usuário para pagina modelos de cortes;
 8. O caso de uso é encerrado.
 - **Fluxo Alternativo:**
 1. A validação encontra algum problema;
 2. O sistema aborta a alteração de dados;
 3. O caso de uso é encerrado.
 - **Requisitos não Funcionais:**
 - Somente usuários cadastrados podem alterar os dados.
 - **Diagrama de sequência:**

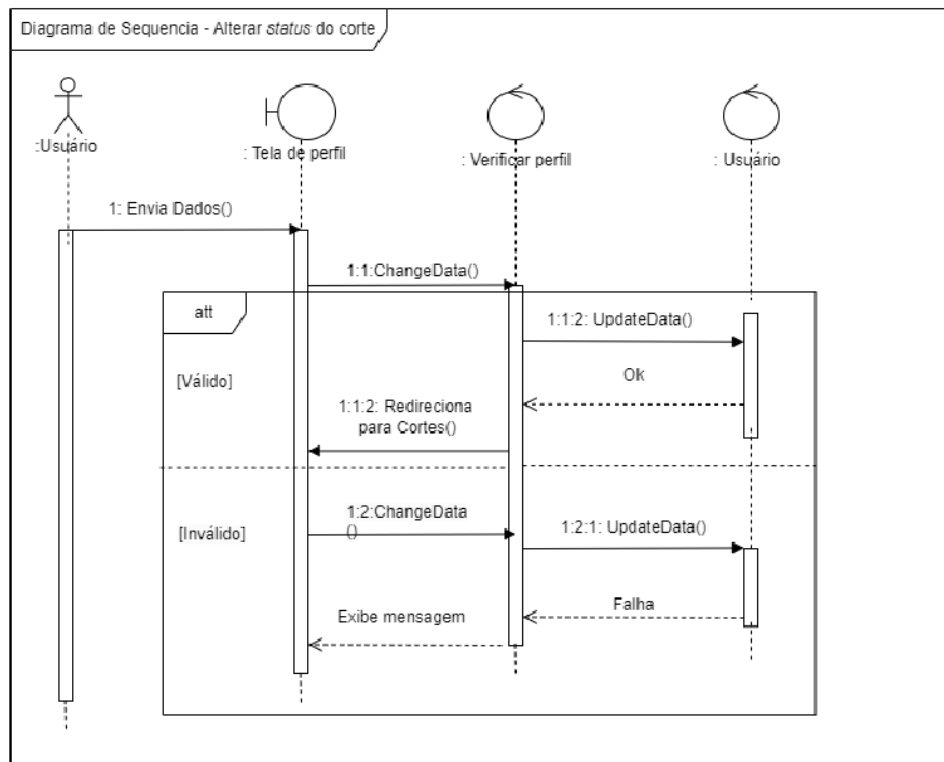


Fig. 18. Diagrama de sequência: Alterar status do corte.

▪ **Requisito da tela:**

A fig. 18 apresenta a interface com o botão switch do status corte.

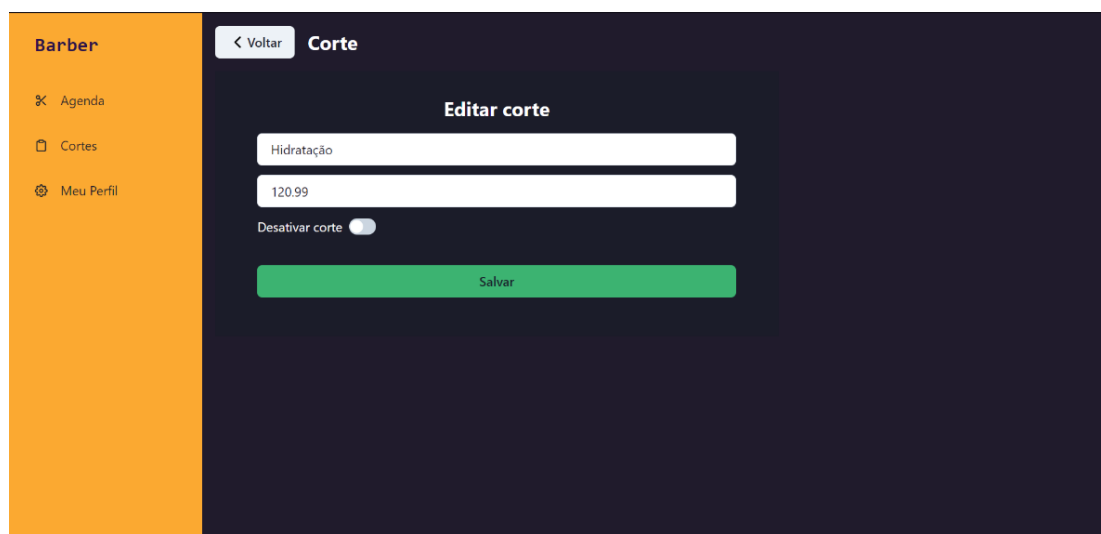


Fig. 19. Interface do botão de alternar status do corte.

5.3.2.9 Identificação: UC9

- **Caso de uso:** Finalizar serviço;
- **ATOR:** Usuário;
- **Objetivo:** Concluir atendimento do cliente;
- **Descrição:** Caso de uso responsável por finalizar o serviço do cliente;
- **PRÉ CONDIÇÃO:** Estar autenticado no sistema;
- **PÓS CONDIÇÃO:** Remover o cliente da fila;
 - **Seção Principal:**
 1. O sistema busca as informações do usuário autenticado.
 2. O usuário seleciona o serviço que deseja finalizar;
 3. O sistema exibe o modal de finalização de atendimento com as informações do atendimento;
 4. O usuário encerra o serviço;
 5. O modal é fechado e o serviço retirado da fila de espera;
 6. O sistema redireciona o usuário para página de clientes;
 7. O caso de uso é encerrado.
 - **Fluxo Alternativo:**
 1. A validação encontra algum problema;
 2. O caso de uso é encerrado.
 - **Diagrama de sequência:**

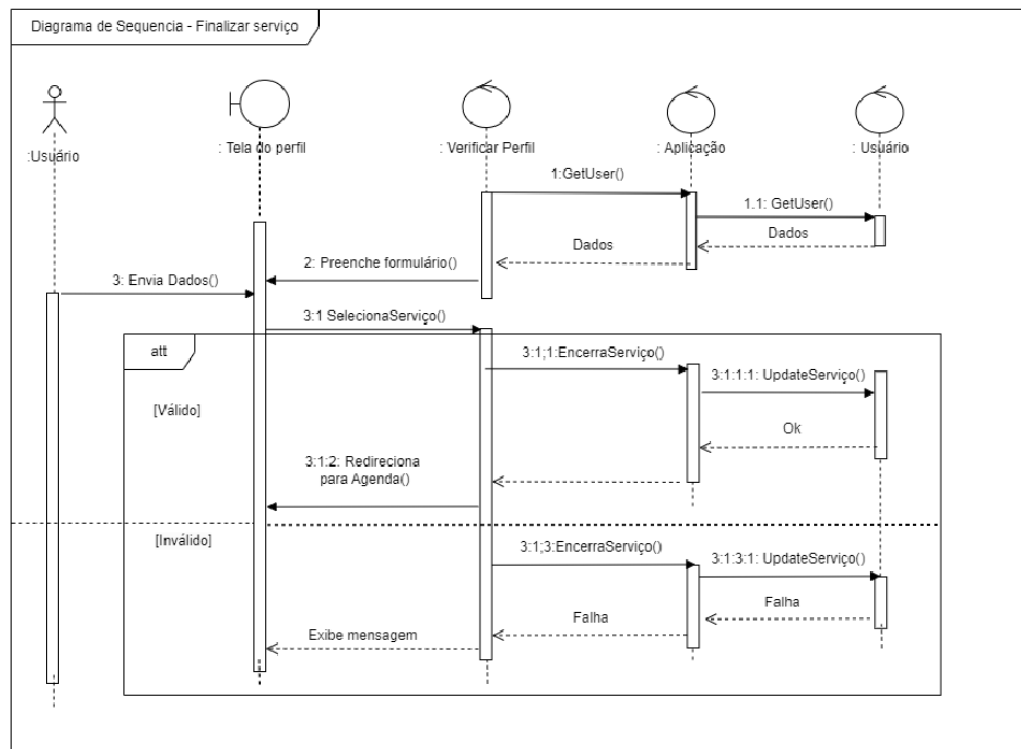


Fig. 20. Diagrama de sequência: Finalizar serviço.

▪ **Requisito da tela:**

A fig. 20 apresenta a interface do modal com as informações do serviço.

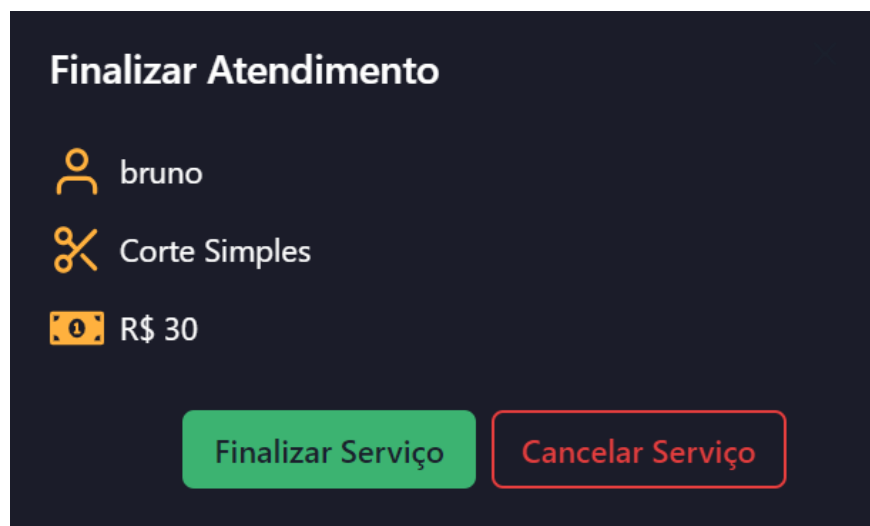


Fig. 21. Modal de finalização de serviço.

6 Considerações Finais

Através deste trabalho, foi possível desenvolver um sistema de controle de fluxo para barbearias, utilizando as tecnologias e ferramentas adequadas para o desenvolvimento web. Os recursos e funcionalidades disponibilizados pelo sistema proposto visam otimizar a gestão e organização do fluxo de clientes na barbearia, proporcionando uma experiência mais eficiente tanto para profissionais quanto para clientes.

Ao longo do trabalho são discutidos conceitos importantes relacionados ao desenvolvimento web, linguagens de programação, bancos de dados e frameworks. Especial ênfase é dada ao uso de tecnologias modernas e de última geração, visando garantir a segurança, desempenho e escalabilidade do sistema.

Durante a implementação do sistema de controle de processos da barbearia, foram identificados e resolvidos desafios relacionados à interface com o usuário, integração de serviços, segurança de dados e sincronização de informações. Essas dificuldades foram superadas por meio de planejamento adequado, análise dos requisitos e consulta de documentações.

Os resultados obtidos com a implementação do sistema são promissores. Com o sistema, as barbearias poderão melhorar o atendimento, reduzir o tempo de espera e agilizar processos.

No que se refere aos trabalhos futuros, os seguintes aspectos devem ser considerados para a implementação de novas versões:

- Desenvolver um aplicativo móvel, aproveitando a base dos serviços web;
- Finalização do serviço *Premium* do sistema;
- Implementação do agendamento autônomo do cliente via aplicativo móvel;
- Implementação do pagamento via sistema e ferramenta de consulta de histórico dos atendimentos;

7 Referencias

- [1] T. L. M. Gontijo; F. Rodrigues; J. F. Lima "Sistema de gerenciamento de salão de beleza Acessado:08/10/2022". [Online]. Disponível <https://www.aedb.br/seget/arquivos/artigos14/47220558.pdf>.
- [2] E. V. Nascimento. "Desenvolvimento de um sistema agregador de barbearias Acessado: 08/10/2022". [Online]. Disponível https://repositorio.ifes.edu.br/bitstream/handle/123456789/1839/TCC_Desenvolvimento_de_um_sistema_agregador_de_barbearias.pdf?sequence=1&isAllowed=y.
- [3] J. F. L. Cordeiro; P.E. L. Chaves; "LEAN BARBER: Aplicação Web para agendamentos de serviços em uma barbearia. Acessado:08/10/2022". [Online]. Disponível: https://www.sje.ifmg.edu.br/portal/images/artigos/biblioteca/TCCs/Sistemas_de_informacao/2021/JEFERSON_FELIPE_LEAL_CORDEIRO_PEDRO_ELIFAS_LEAO_CHAVES.pdf.
- [4] "About Node,JS". [Online] Disponível em: <https://nodejs.org/en/about>.
- [5] "JavaScript — Dynamic client-side scripting" MDN Web Docs, [Online] Disponível em: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript>.
- [6] "JavaScript" MDN Web Docs, [Online] Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- [7] "The TypeScript handbook", [Online] Disponível em: <https://www.typescriptlang.org/docs/handbook/intro.html>.
- [8] "Introdução Express/Node", [Online] Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introduction
- [9] "What is PostgreSQL", [Online] Disponível em: <https://www.postgresql.org/about/>
- [10] "Prisma: uma das melhores coisas que já aconteceu no ecossistema?" [Online] Disponível em: <https://blog.rocketseat.com.br/prisma-uma-das-melhores-coisa-que-ja-aconteceu-no-ecossistema/>
- [11] "Desenvolvendo interfaces intuitivas com o ChakraUI". [Online]. Disponível em <https://coodesh.com/blog/candidates/desenvolvendo-interfaces-intuitivas-com-o-chakra-ui/>
- [12] "React: o que é, como funciona e um guia dessa popular ferramenta JS". [Online]. Disponível em <https://www.alura.com.br/artigos/react-js>
- [13] "Introdução ao visual studio code". [Online]. Disponível em <https://www.devmedia.com.br/introducao-ao-visual-studio-code/34418>
- [14] "DataGrip Recursos". [Online]. Disponível em <https://www.jetbrains.com/pt-br/datagrip/features/>
- [15] "Introduction to JSON Web Tokens". [Online]. Disponível em <https://jwt.io/introduction>
- [16] "Insomnia" [Online]. Disponível em <https://www.sleepfoundation.org/insomnia>

[17] "Luxon" [Online]. Disponível em <https://github.com/moment/luxon>