FACULDADE DE TECNOLOGIA DE AMERICANA "MINISTRO RALPH BIASI"
Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Lucas Denadai Belotti
Vítor Hugo Cartone Neves
Luis Felipe Barbosa
Heitor Campanhol da Silva

Sistema de Agendamento - Corta pra mim

Americana, SP
2024

Lucas Denadai Belotti
Vítor Hugo Cartone Neves
Luis Felipe Barbosa
Heitor Campanhol da Silva

Sistema de Agendamento - Corta pra mim

Trabalho de Conclusão de Curso desenvolvido em cumprimento à exigência curricular do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas na área de concentração em

Orientador(a): Prof. Me. Thiago Salhab Alves

Este trabalho corresponde à versão final do Trabalho de Conclusão de Curso apresentado por Lucas Denadai Belotti, Vítor Hugo Cartone Neves, Luis Felipe Barbosa, Heitor Campanhol da Silva e orientado pelo Prof. Me. Thiago Salhab Alves.

Americana, SP
2024

FICHA CATALOGRÁFICA – Biblioteca Fatec Americana Ministro Ralph Biasi- CEETEPS Dados Internacionais de Catalogação-na-fonte

BELOTTI, Lucas Denadai

Sistema de Agendamento - Corta pra Mim. / Lucas Denadai Belotti, Vítor Hugo Cartone Neves, Luis Felipe Barbosa, Heitor Campanhol da Silva – Americana, 2024.

81f.

Monografia (Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) - - Faculdade de Tecnologia de Americana Ministro Ralph Biasi – Centro Estadual de Educação Tecnológica Paula Souza

Orientador: Prof. Ms. Thiago Salhab Alves

1. C # – linguagem de programação 2. Desenvolvimento de software 3. SQL – banco de dados. I. BELOTTI, Lucas Denadai, II. NEVES, Vítor Hugo Cartone, III. BARBOSA, Luis Felipe, IV. SILVA, Heitor Campanhol da V. ALVES, Thiago Salhab VI. Centro Estadual de Educação Tecnológica Paula Souza – Faculdade de Tecnologia de Americana Ministro Ralph Biasi

CDU: 681.3.061C#
681.3.05
681.3.07SQL

Elaborada pelo autor por meio de sistema automático gerador de ficha catalográfica da Fatec de Americana Ministro Ralph Biasi.

Lucas Denadai Belotti
Vítor Hugo Cartone Neves
Heitor Campanhol da Silva
Luis Felipe Barbosa

Sistema de Agendamento - Corta pra mim

Trabalho de graduação apresentado como exigência parcial para obtenção do título de Tecnólogo em Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas pelo Centro Paula Souza – FATEC Faculdade de Tecnologia de Americana Ministro Ralph Biasi.
Área de concentração: Desenvolvimento de Sistemas.

Americana, 20 de junho de 2024.

Banca Examinadora:

Thiago Salhab Alves
Mestre
Fatec Americana Ministro Ralph Biasi

João Emmanuel D Alkmin Neves
Doutor
Fatec Americana Ministro Ralph Biasi

Odilon Delmont Filho
Doutor
Fatec Americana Ministro Ralph Biasi

RESUMO

Este projeto surge como resposta às crescentes demandas de agendamentos, buscando inovar e elevar a qualidade dos serviços oferecidos. Inspirado nas tendências do mercado, o objetivo é criar um sistema de gestão de agenda, destacando-se pela interatividade entre a Aplicação Web, Desktop e Mobile para proporcionar agilidade e praticidade ao gestor. O trabalho é estruturado em objetivos específicos, incluindo a entrega de um protótipo para validação do trabalho e a apresentação de conclusões finais, juntamente com possibilidades para trabalhos futuros.

Palavras-Chave: Agendamento; Gestão; Sistema; Aplicação.

ABSTRACT

This project arises from responses to the growing of the scheduling demands, seeking to innovate and elevate the quality of services offered. Inspired by market trends, the goal is to create an agenda management system, standing out for its interactivity between the web, desktop and mobile application to provide agility and practicality for the manager. The work is structured in specific objectives, including the delivery of a prototype to validation of the work and the presentation of final conclusions, along with possibilities for future works.

Key-Words: Scheduling; Management; System; Application**.**

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

UML	*Unified Modeling Language*

SUMÁRIO

1 INTRODUÇÃO

No mundo em constante evolução tecnológica e empresarial, muitas pessoas estão ficando sem tempo para agendar um horário em algum local, seja por uma semana agitada ou por motivos mais simples como uma gripe ou outra enfermidade. Nesse contexto, o presente trabalho surge como uma resposta às demandas crescentes desse setor em constante mutação.

A inspiração para este trabalho vem das tendências do mercado e do desejo de elevar a qualidade dos serviços oferecidos. Acredita-se que as conclusões e recomendações apresentadas serão valiosas para os gestores de diversas empresas que utilizam uma agenda para atender os clientes.

Este trabalho tem como objetivo criar um sistema de agendamento – Corta pra Mim, tendo como diferencial a interatividade do Sistema Web, com o Sistema Desktop e um Sistema Mobile, trazendo agilidade e praticidade ao gestor. Para melhor entendimento e gestão do trabalho foi escolhido um tema específico e este tema é a barbearia. Quanto aos objetivos específicos do trabalho são:

- Reuniões semanalmente para expor o andamento do projeto;
- Quinzenalmente enviar feedback ao professor orientador;
- Ao final do projeto realizar a entrega de um protótipo para validação do cliente;
- Apresentar as conclusões finais e exibir as possibilidades para trabalhos futuros.

O restante do trabalho está organizado em quatro capítulos conforme descrição a seguir: Capítulo 1 introdução ao sistema, Capítulo 2 apresenta as informações principais do sistema desenvolvido, o Capítulo 3 descreve o desenvolvimento do projeto, o Capítulo 4 apresenta as etapas de entregas feitas pela equipe, e por fim, as considerações finais juntamente com as diversas possibilidades de trabalhos futuros são apresentadas no Capítulo 5.

2 PROJETO DO SISTEMA

Neste capítulo, é apresentado o processo de criação do protótipo do aplicativo em detalhes. Para entender melhor os requisitos essenciais, foram conduzidas pesquisas em sites e softwares semelhantes. Permitindo assim identificar as características cruciais para o desenvolvimento de um programa de assistência na gestão de barbearias, simplificando o levantamento de requisitos e a definição dos recursos e ferramentas desejados no aplicativo.

2.1 Recursos e Ferramentas

Esta seção contempla as ferramentas de programação e os conceitos necessários para o desenvolvimento do sistema:

- **Visual Studio:** O IDE do Visual Studio é uma plataforma de lançamento criativa que onde é usado para editar, depurar e criar código e, em seguida, publicar um aplicativo. Além do editor e depurador padrão fornecidos pela maioria dos IDEs, o Visual Studio inclui compiladores, ferramentas de conclusão de código, designers gráficos e muitos outros recursos para aprimorar o processo de desenvolvimento de software. (VISUAL STUDIO, 2022).
- **Visual Studio Code:** É um editor de código leve, mas poderoso que funciona em seu computador e está disponível para Windows, *macOS* e Linux. Ele vem com um suporte integrado de JavaScript, TypeScript e Node.js e tem um ecossistema de extensões rica em outras linguagens e tempos de execução (tal como C++, C#, Java, Python, PHP, Go, .NET). (Microsoft, 2024).
- **C#**: É uma linguagem de programação moderna, orientada a objetos. Usando C#, os desenvolvedores podem usar o .NET para criar muitos tipos de aplicativos seguros e robustos. Como o C# tem suas raízes na família de linguagens C, os programadores C, C++, Java e JavaScript podem se atualizar rapidamente. (Microsoft, 2022).

- **React Native:** Combina as melhores partes do desenvolvimento nativo com React, A melhor categoria de biblioteca JavaScript para criação de interface de usuários. (Meta, 2024).
- **Blazor:** É uma estrutura para criar uma interface do usuário web interativa do lado do cliente com o .NET. Crie interfaces do usuário interativas e avançadas em C# em vez de JavaScript. Compartilhe a lógica de aplicativo do lado do cliente e do servidor gravada no .NET. Renderize a interface do usuário, como HTML e CSS para suporte amplo de navegadores, incluindo navegadores móveis. Integre-se a plataformas de hospedagem modernas, como o Docker. Crie aplicativos móveis e de área de trabalho híbrida com .NET e Blazor. (Microsoft, 2023).
- **ASP.NET Core:** O ASP.NET Core é uma estrutura de interface do usuário completa. Escolha quais funcionalidades se ajustam às necessidades da interface do usuário da Web do aplicativo. (Microsoft, 2023).
- **SQL Server**: O SQL Server é projetado para armazenar, recuperar, e gerenciar dados de maneira eficiente. Ele oferece suporte a uma linguagem de consulta chamada SQL (*Structured Query Language*) para realizar operações em bancos de dados relacionais. (Microsoft, 2022)
- **HTML: HTML** (Linguagem de Marcação de Hipertexto) é o bloco de construção mais básico da web. Define o significado e a estrutura do conteúdo da web. Outras tecnologias além do HTML geralmente são usadas para descrever a aparência/apresentação (CSS) ou a funcionalidade/comportamento (JavaScript) de uma página da web. (Mozilla, 2022).
- **CSS: CSS** (*Cascading Style Sheets* **ou Folhas de Estilo em Cascata)** é uma linguagem de estilo (en-US) usada para descrever a apresentação de um documento escrito em HTML ou em XML (incluindo várias linguagens em XML como SVG, *MathML* ou XHTML). O CSS descreve como elementos são mostrados na tela, no papel, na fala ou em outras mídias. (Mozilla, 2022).
- **JavaScript:** É uma linguagem de programação de alto nível, dinâmica e orientada a objetos. Originalmente desenvolvida pela Netscape, tornou-

se uma das linguagens de programação mais populares e é amplamente usada para criar páginas web interativas. (Mozilla, 2022).

- **Trello:** Com o Trello, o time pode realizar o trabalho com mais facilidade. Seja qual for o projeto, o fluxo de trabalho ou o tipo de time, o Trello pode ajudar a manter tudo em ordem. É simples: faça a inscrição, crie um quadro e pronto! A produtividade está esperando por você. (Atlassian, 2011).
- **Postman:** Postman é uma plataforma de API para criar e usar APIs. O Postman simplifica cada etapa do ciclo de vida da API e simplifica a colaboração para que você possa criar APIs melhores — mais rapidamente. (Postman, 2023)
- ***Google Maps*:** É um serviço de mapas online desenvolvido pela Google. Ele fornece informações sobre localização geográfica, imagens de satélite, mapas de ruas, rotas para navegação, informações sobre empresas locais e muito mais. É amplamente utilizado para encontrar direções, explorar áreas específicas e visualizar mapas interativos. (Google, 2024)
- **Azure:** É uma plataforma de computação em nuvem oferecida pela Microsoft. Ele fornece uma ampla gama de serviços e recursos para ajudar empresas e desenvolvedores a construir, implantar e gerenciar aplicativos por meio da infraestrutura global da Microsoft. (Microsoft, 2023)
- ***Expo*:** *Expo GO* é um *sandbox* grátis e de código aberto para aprendizagem e experimentação com *React Native* nos dispositivos Android e iOS. Você pode instalar diretamente de loja de aplicativos, e comece a trabalhar em poucos minutos - não há necessidade de instalar um conjunto de ferramentas nativo e compilar um aplicativo. (*Expo*, 2024).
- **SQLite:** O banco de dados no SQLite é um simples documento em disco. Além disso, o formato do arquivo é multiplataforma. Um banco de dados que é criado em uma máquina pode ser copiado e usado em diferentes máquinas com diferentes estruturas. O banco de dados SQLite funciona em máquinas 32-bit e 64-bite entre arquiteturas big-endian e little-endian. (SQLite, 2022).

2.2 Softwares Similares

A fim de melhorar a comunicação entre cabelereiros e seus clientes, existem variados sites e softwares de agendamento. Atualmente, os softwares mais populares que apresentam estas funcionalidades são o Reservio, Booksy, Trinks.

Reservio é um website de agendamento para a empresa, onde seu cliente pode marcar uma data utilizando uma conta ou como convidado.

Booksy é um website de agendamento para salões de beleza, cabelereiros entre outros estabelecimentos do tipo, onde seu cliente pode agendar um corte de cabelo apenas se criar uma conta.

Trinks é um website de agendamento para salões de beleza, cabelereiros, barbearias, spas e clínicas de estética, no qual há um destaque maior para o gerenciamento da empresa e não do agendamento do cliente.

Levando estes aspectos em consideração, foi elaborada a Tabela 1 mostrando as principais diferenças entre Reservio, Booksy, Trinks e o trabalho desenvolvido neste trabalho:

Tabela 1 – Comparativo de funcionalidades entre Reservio, Booksy, Trinks e o aplicativo desenvolvido neste trabalho.

Funcionalidade	Sistema	Reservio	Booksy	Trinks
Registro de Estabelecimento		X	X	X
Cadastrar Cliente	X	X	X	X
Reservar como Convidado	X	X		
Aplicação Desktop	X			
Aplicação WEB	X	X	X	X
Aplicação Mobile	X			
Avisar sobre desistência em Horário	X			
Confirmar antecipadamente	X	X	X	X

Fonte: Autoria própria, 2023.

2.3 Levantamento de Requisitos

A engenharia de requisitos (RE – *Requirements Engineering*) é o processo de descobrir, analisar, documentar e verificar requisitos de um sistema. Um requisito pode ser definido como uma descrição dos serviços fornecidos pelo sistema e as suas restrições operacionais (SOMMERVILLE, 2018). Tradicionalmente, os requisitos são divididos em dois tipos: requisitos funcionais e requisitos não funcionais.

2.3.1 Requisitos Funcionais

Os requisitos funcionais descrevem o que o sistema deve fazer, isto é, definem a funcionalidade desejada do software (SOMMERVILLE, 2018). A Tabela 2 apresenta os requisitos funcionais deste projeto.

Tabela 2 – Requisitos funcionais do projeto.

Identificação	Requisito Funcional	Prioridade
RF001	Agendamento	Essencial
RF002	Cadastro de Cliente	Importante
RF003	Resumo da Agenda	Desejável
RF005	Tabela de Clientes do Dia	Importante
RF006	Visualização dos dados do Cliente	Importante
RF007	Cadastro de Usuários	Importante
RF008	Visualização dos dados do Usuário	Importante
RF009	Visualização dos horários agendados	Essencial
RF010	Notificações automáticas	Desejável

Fonte: Autoria própria, 2023.

2.3.2 Requisitos Não Funcionais

“Os requisitos não funcionais são aqueles não diretamente relacionados às funções específicas fornecidas pelo sistema” (SOMMERVILLE, 2018). A Tabela 3 apresenta os requisitos não funcionais deste projeto.

Tabela 3 – Requisitos não funcionais do projeto.

Identificação	Requisito não funcional	Categoria	Prioridade
RNF001	Confirmação de comparecimento	Usabilidade	Importante
RNF002	Sobre do Sistema Desktop	Usabilidade	Desejável
RNF003	Compatibilidade Desktop	Hardware e Software	Importante
RNF004	Página Sobre	Confiabilidade	Desejável
RNF005	Página Localização	Confiabilidade	Desejável
RNF006	Tabela LOG	Usabilidade	Importante
RNF007	Máscara para Telefone	Desempenho	Essencial
RNF008	Política de Privacidade	Segurança	Essencial
RNF009	Manual do Sistema Desktop	Usabilidade	Importante
RNF010	Sair do Sistema Desktop	Usabilidade	Desejável
RNF014	Confirmação do agendamento cliente	Usabilidade	Importante
RNF016	Confirmação da conta	Segurança	Essencial
RNF017	Criptografia	Segurança	Essencial
RNF018	Campo de login	Usabilidade	Importante
RNF019	Informação de usuário logado	Usabilidade	Desejável
RNF020	Verificar se o cliente já foi autenticado	Confiabilidade	Essencial
RNF021	Fonte para melhor visibilidade	Usabilidade	Desejável

Fonte: Autoria própria, 2023.

Após o levantamento de Requisitos, tanto funcionais quanto não funcionais, foi dado início a modelagem do sistema.

3 MODELAGEM

Na fase da modelagem é feita a documentação do aplicativo, trata-se de diagramas que facilitam na compreensão do projeto de forma padronizada.

A documentação deste trabalho utilizará a linguagem de modelagem *Unified Modeling Language* [1] (UML) para modelar os casos de uso e o diagrama de classe.

3.1 Casos De Uso

Em um livro que discute como redigir casos de uso eficazes, Alistair Cockburn [Coc01b] observa que “um caso de uso captura um contrato... [que] descreve o comportamento do sistema sob várias condições à medida que o sistema responde a uma solicitação de um de seus interessados...”. Essencialmente, um caso de uso conta uma história estilizada sobre como um usuário final (desempenhando um de uma série de papéis possíveis) interage com o sistema sob um conjunto de circunstâncias específicas (SOMMERVILLE, 2018).

Os diagramas de caso de uso descrevem um cenário de funcionalidades do ponto de vista do usuário, catalogando os requisitos funcionais do sistema. Dentro do diagrama são retratados os atores (representado pelos bonecos), as funcionalidades (representadas pelos balões com a ação escrita por dentro) e as relações (representadas pelas linhas).

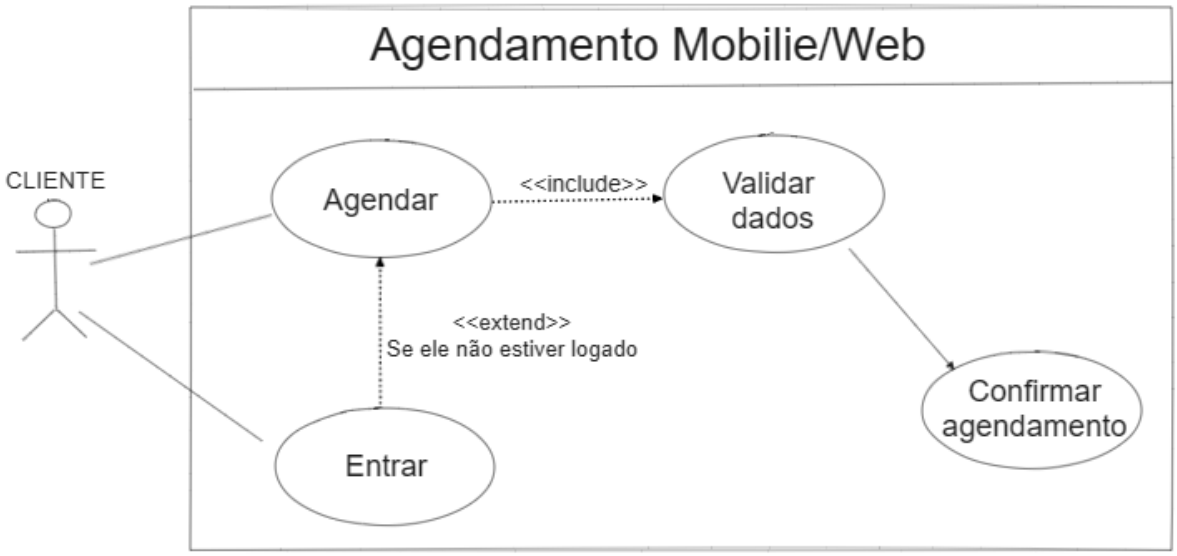
Os atores que interagem com o sistema são: o Usuário e o Cliente. O sistema é um caso de uso explícito e se trata do sistema em si em que os casos de uso acontecem.

- **Usuário** é o ator que representa os cabeleireiros. Um ator pode, por exemplo, cadastrar clientes, agendar horários, visualizar horários agendados, cancelar agendamentos, entre outros.
- **Cliente** é o ator que representa os clientes. Um ator pode, por exemplo, fazer o cadastro e agendar o horário.

[1] *Unified Modeling Language* ou Linguagem Unificada de Modelagem (UML) é uma linguagem padrão para modelagem e documentar os sistemas orientados a objetos.

A Figura 1 apresenta o caso de uso para o agendamento do cliente no sistema web e mobile.

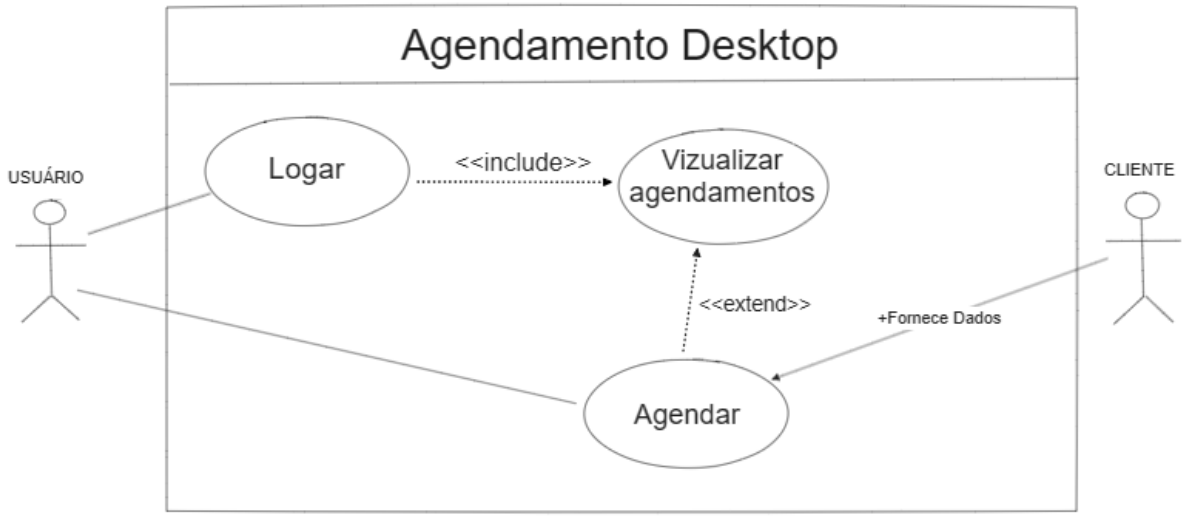
Figura 1 – Diagrama de caso de uso do Agendamento do cliente no sistema WEB/Mobile



Fonte: Autoria própria, 2024.

A Figura 2 apresenta o caso de uso para o agendamento do cliente no sistema desktop.

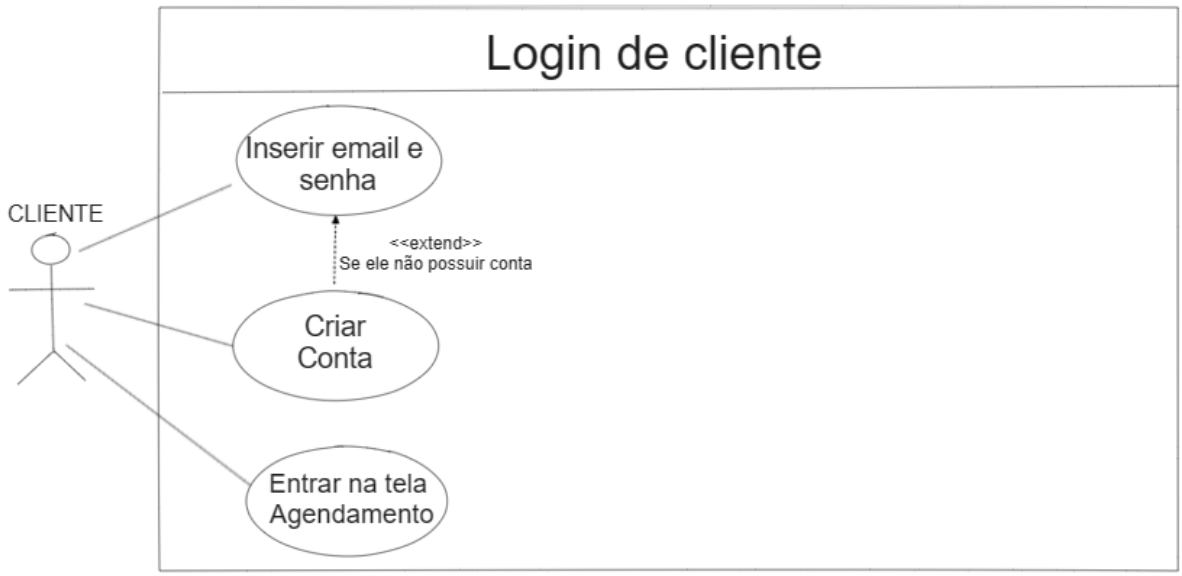
Figura 2 – Diagrama de caso de uso do Agendamento do cliente no sistema desktop



Fonte: Autoria própria, 2024.

A Figura 3 apresenta o caso de uso para o login do cliente no sistema web e mobile.

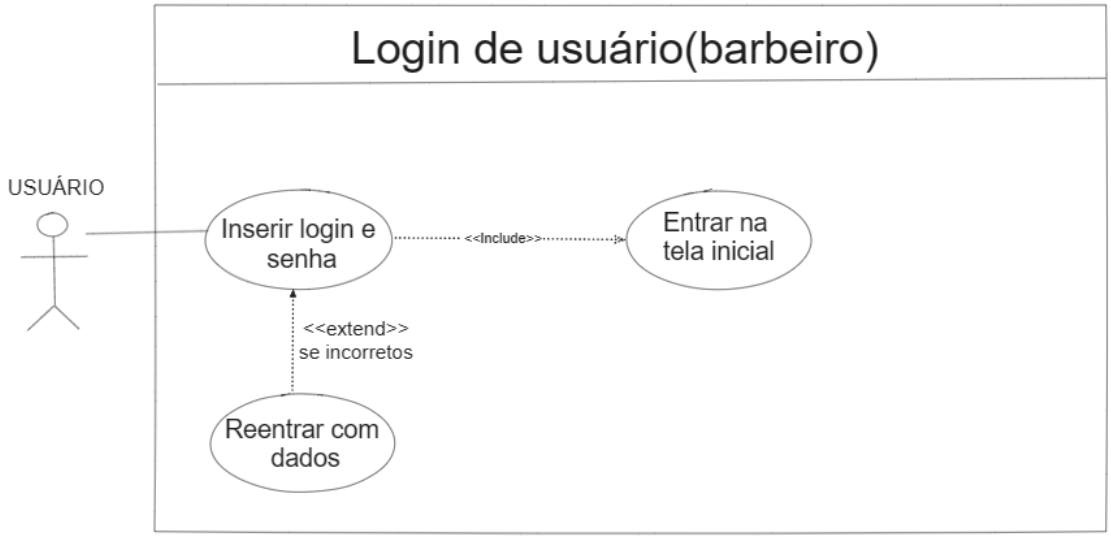
Figura 3 – Diagrama de caso de uso para o login do cliente no sistema WEB/Mobile



Fonte: Autoria própria, 2024.

A Figura 4 apresenta o caso de uso para o login do cliente no sistema desktop.

Figura 4 – Diagrama de casa de uso para o login do usuário no sistema desktop



Fonte: Autoria própria, **2024.**

No subcapítulo 3.1.1 será apresentado à documentação dos casos de uso do projeto deste trabalho.

3.1.1 Documentação dos Casos de Uso

Cada funcionalidade dos diagramas de casos de uso será descrita da Tabela 4 à Tabela 7.

Tabela 4 – Caso de uso “Agendamento WEB/Mobile”.

Nome do caso de uso	Agendamento WEB/Mobile
Atores envolvidos	Cliente
Objetivo	Este caso de uso descreve os passos de agendamento no sistema WEB e Mobile
Ações do ator	**Ações do Sistema**
1. O cliente clica em Agendar ou Entrar.	
	2. O sistema verifica se o cliente está logado.

3. Se o cliente não estiver logado ele terá que entrar na conta.	
4. Se o cliente já estiver logado, vai para o próximo passo.	
	5. O sistema leva até a página para escolher o dia e a hora do corte.
6. O cliente escolhe data e hora do corte e clica em continuar.	
	7. O sistema valida os dados do cliente no banco.
Validações	Para o agendamento ser bem-sucedido, os dados do cliente precisa estar nos conformes, e a data e hora precisam estar disponíveis.

Fonte: Autoria própria, 2024.

Tabela 5 – Caso de uso "Agendamento Desktop".

Nome do caso de uso	Agendamento Desktop
Atores envolvidos	Cliente, Usuário
Objetivo	Este caso de uso descreve os passos de agendamento no sistema Desktop.
Ações do ator	**Ações do Sistema**
1. O usuário faz o login.	
2. O usuário visualiza agendamentos.	
3. O usuário verifica data e hora disponíveis para o cliente	
4. Cliente fornece dados para usuário agendar o corte.	
	5. O sistema, salva o agendamento no banco.
Validações	Para o agendamento ser bem-sucedido, os dados do cliente precisam estar nos conformes, e a data e hora precisam estar disponíveis.

Fonte: Autoria própria, 2024.

Tabela 6 – Caso de uso "Login no Sistema WEB/Mobile".

Nome do caso de uso	Login do cliente no sistema WEB/Mobile
Atores envolvidos	Cliente
Objetivo	Este caso de uso descreve os passos do login do cliente no sistema web
Ações do ator	**Ações do Sistema**
1. O cliente clica no botão de entrar.	
	2. O sistema leva o cliente para a tela de login.
3. Cliente fornece e-mail e senha.	
	4. O sistema valida os dados para efetuar o login.
	5. Se cliente não existir, o sistema avisa para criar uma conta.
6. Cliente cria a conta caso não exista no banco.	
	7. O sistema valido os dados.
Validações	Para o login ser bem-sucedido, os dados do cliente precisa estar nos conformes e existir no banco.

Fonte: Autoria própria, 2024.

Tabela 7 – Caso de uso "Login no Sistema Desktop".

Nome do caso de uso	Login do cliente no sistema Desktop
Atores envolvidos	Cliente
Objetivo	Este caso de uso descreve os passos do login do usuário no sistema Desktop.
Ações do ator	**Ações do Sistema**
1. O usuário insere login e senha	
	2. O sistema valida os dados fornecidos.

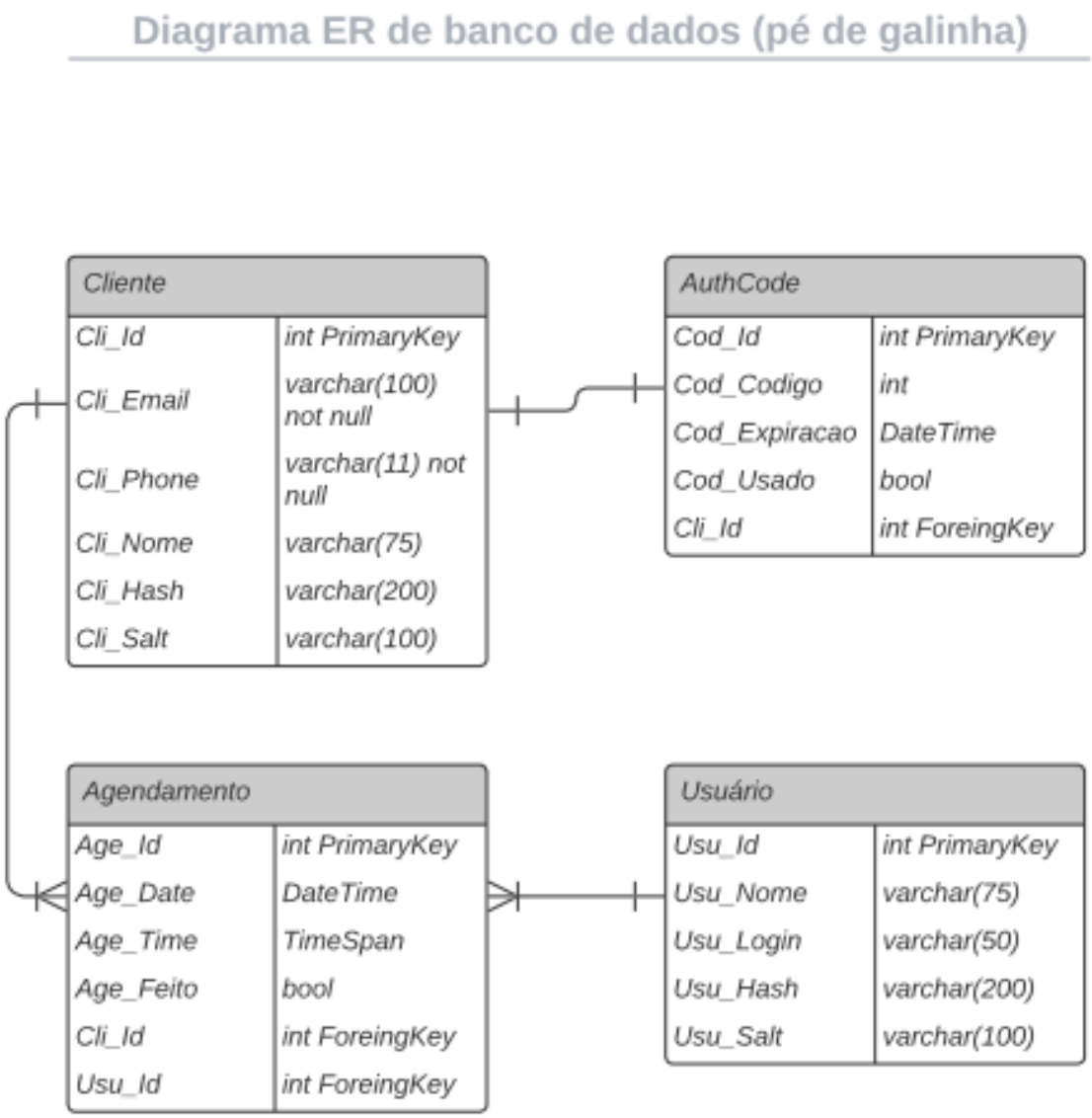
	3. Se os dados estiverem incorretos, o sistema avisa para reentrar com os dados.
4. Cliente redigita os dados.	
	5. O sistema valida os dados fornecidos
Validações	Para o login ser bem-sucedido, os dados do usuário precisam existir no banco.

Fonte: Autoria própria, 2024.

3.2 Diagramas entidade relacionamento

O Modelo Entidade Relacionamento (também chamado Modelo ER, ou simplesmente MER), como o nome sugere, é um modelo conceitual utilizado na Engenharia de Software para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).(DEVMEDIA, 2024)

Figura 5 – Diagrama Entidade Relacionamento



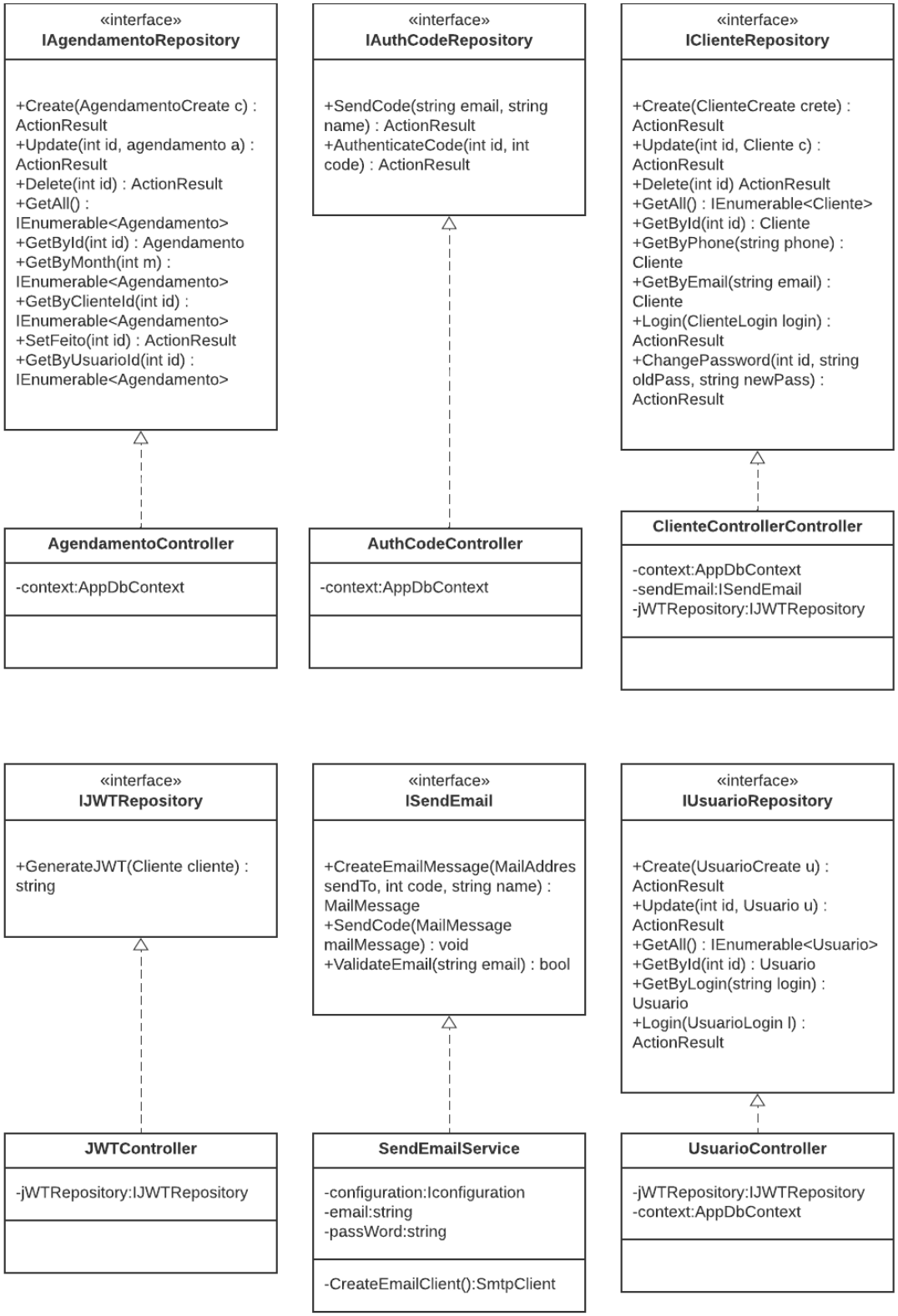
Fonte: Autoria própria, **2024.**

3.3 Diagramas de Classe

Diagrama de classe surge a partir dos cenários, cada cenário de uso implica um conjunto de objetos manipulados à medida que um ator interage com o sistema. Esses objetos são categorizados em classes (SOMMERVILLE, 2018).

Abaixo na figura 6, pode-se observar o diagrama de classe referente a API, nela é possível verificar as principais classes utilizadas e suas interfaces de atuação.

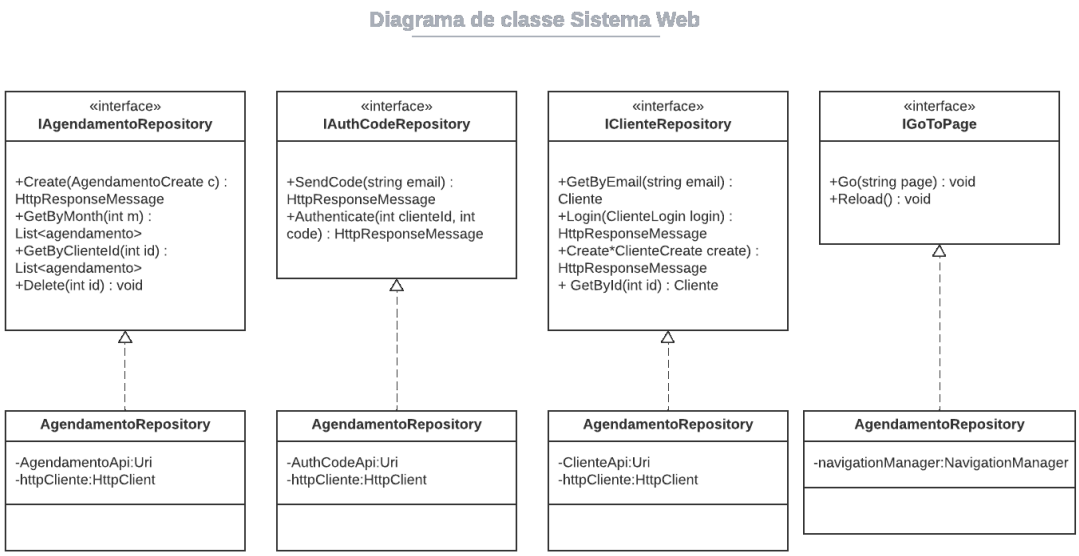
Figura 6 - Diagrama de Classe API



Fonte: Autoria própria, 2024.

Abaixo na figura 7, pode-se observar o diagrama de classe referente a Web, nele é possível ver as classes e suas respectivas telas de atuação.

Figura 7 - Diagrama de Classe Web

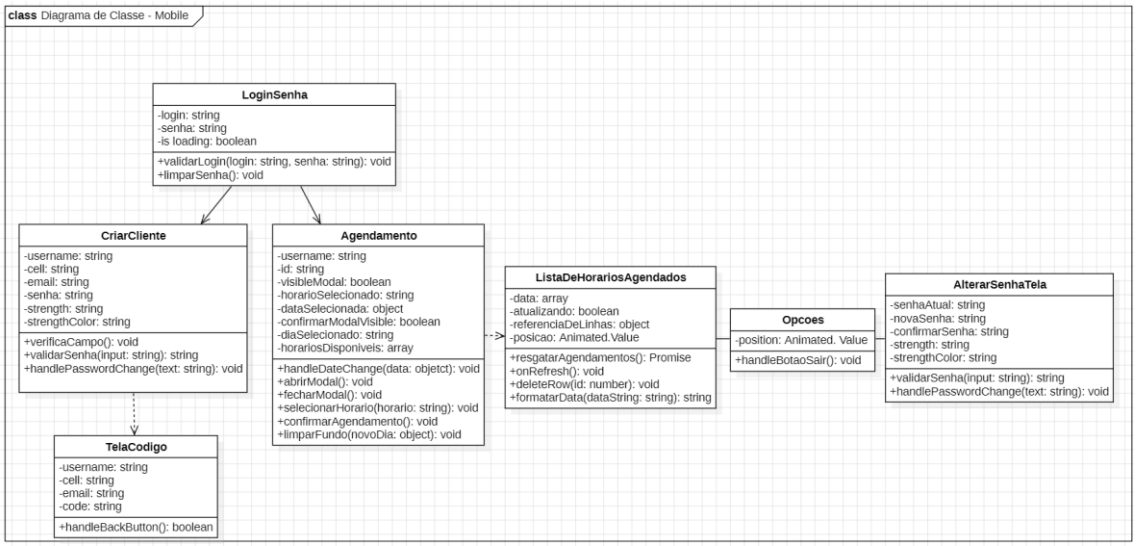


Fonte: Autoria própria, 2024.

Na figura 8, pode-se verificar o diagrama de classe referente ao Sistema Mobile, nela é possível verificar as principais classes da aplicação, sendo a de

Agendamento mais extensa por conter as informações mais importantes da aplicação.

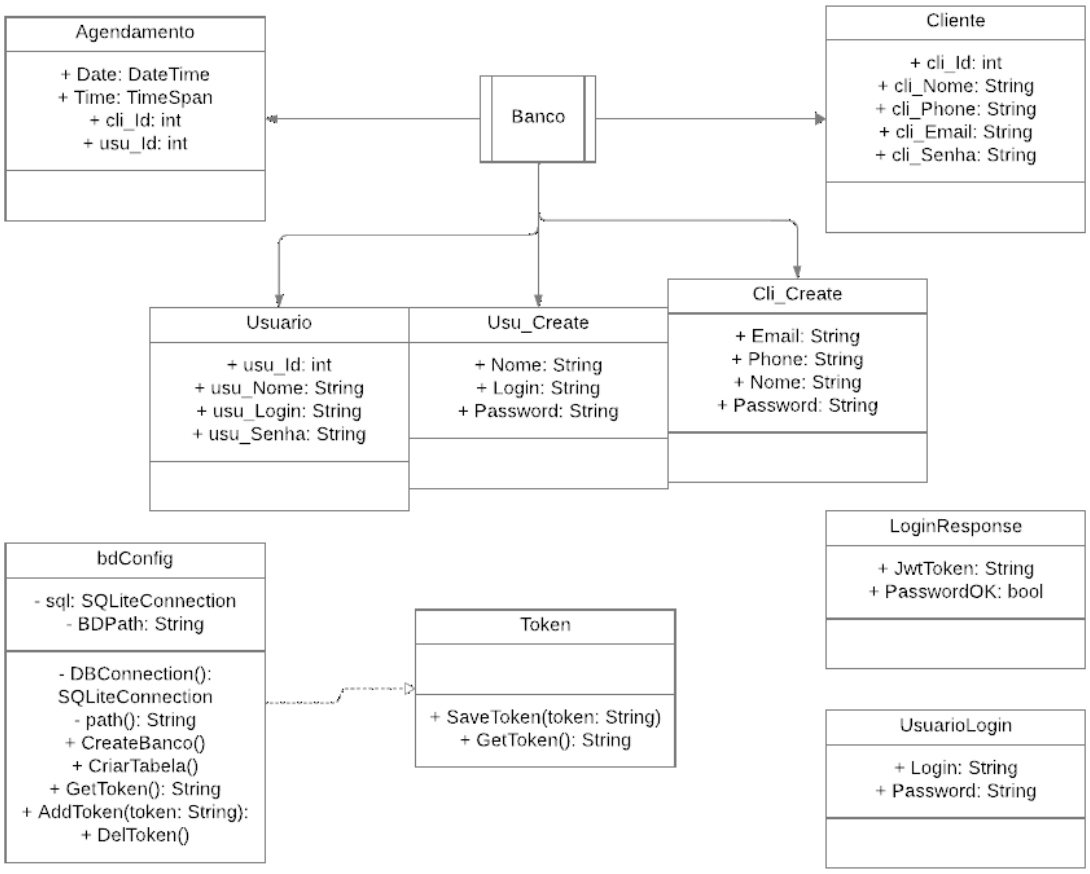
Figura 8 - Diagrama de Classe Mobile



Fonte: Autoria própria, 2024)

Na figura seguinte, é possível constatar o diagrama de classe referente ao Desktop. A diferença desse diagrama para os outros é a classe “bdConfig” que é onde é feita a criação e conexão do Banco de Dados SQLite.

Figura 9 - Diagrama de classe desktop



Fonte: Autoria própria, 2024.

3.4 Diagramas de Sequência

O diagrama de sequência indica como os eventos provocam transições de objeto para objeto. Uma vez que os eventos tenham sido identificados pelo exame de um caso de uso, o modelador cria um diagrama de sequência (SOMMERVILLE, 2018).

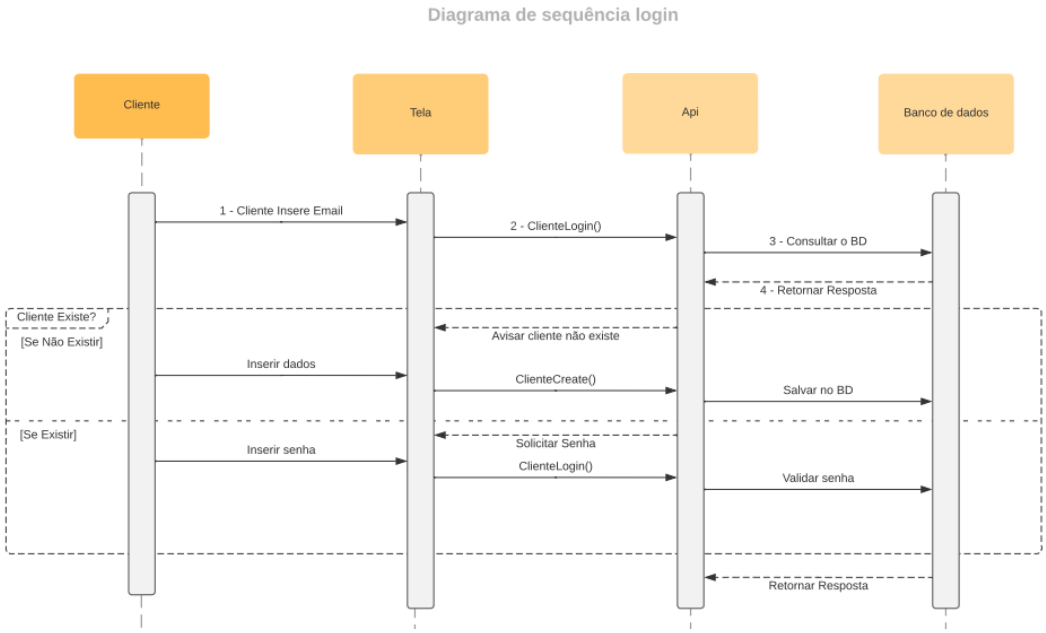
Estes diagramas estarão mais bem exemplificados a seguir:

3.4.1 Login

Na figura 10, é possível visualizar o diagrama de sequência referente ao login, sendo que o cliente/usuário irá informar seu respectivo login na tela e a API pesquisará a correspondência no banco de dados, caso o cliente não exista, a API irá avisar ao cliente/usuário por uma mensagem que aparecerá na tela,

assim dando a opção de efetuar um cadastro, caso o cliente/usuário seja encontrado a API irá solicitar a senha, após digitar a senha a API verificará se é correspondente ao login digitado anteriormente, concluindo assim o processo de entrada no sistema.

Figura 10 – Diagrama de Sequência de Login

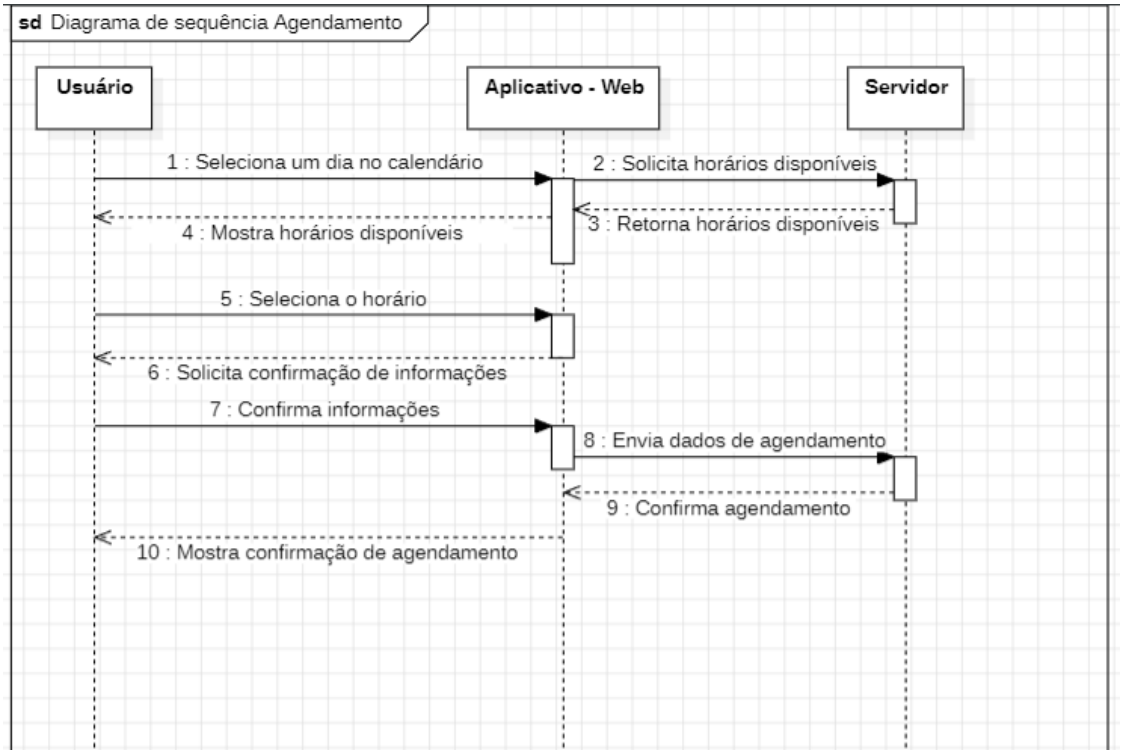


Fonte: Autoria própria, 2024.

3.4.2 Agendamento

Na figura 11, é possível visualizar o diagrama de sequência do agendamento, sendo que o usuário/cliente, seleciona um dia na aplicação e a aplicação faz uma requisição na API para que os horários disponíveis sejam visualizados, após o horário ser selecionado e confirmado, os dados do agendamento serão enviados ao servidor concretizando a ação.

Figura 11 – Diagrama de Sequência Agendamento



Fonte: Autoria própria, 2024.

4. DESENVOLVIMENTO

O método utilizado para entrega deste Projeto foi o Scrum, que consiste em uma estrutura de gestão ágil que auxilia a equipe a controlar situações problemáticas e adversas que esteja acontecendo no projeto, esse controle ocorre com a utilização de reuniões semanais entre os membros:

Lucas: CEO e Desenvolvedor Desktop.

Vítor: Desenvolvedor.

Luis: Desenvolvedor Mobile.

Heitor: Desenvolvedor Mobile.

4.1 Etapas de Desenvolvimento

4.1.1 API

A API é utilizada para que as aplicações “conversem” com o banco de dados presente na nuvem. Feita em C# com ASP.NET Core, ela é dividida em funções relacionadas ao CRUD, além de métodos exclusivos para login, autenticação de e-mail e para trabalhar com os tokens JWT (Json Web Token) dentro das aplicações. Todas as funções retornam alguns dos códigos HTTP, tais como:

- 200 – Sucesso
- 400 – Requisição incorreta
- 401 – Não autorizado
- 404 – Não encontrado
- 500 – Erro no servidor

Além disso, o JWT foi adotado para a utilização de alguns métodos da API, para garantir a maior segurança das aplicações.

4.1.1.1 Cliente

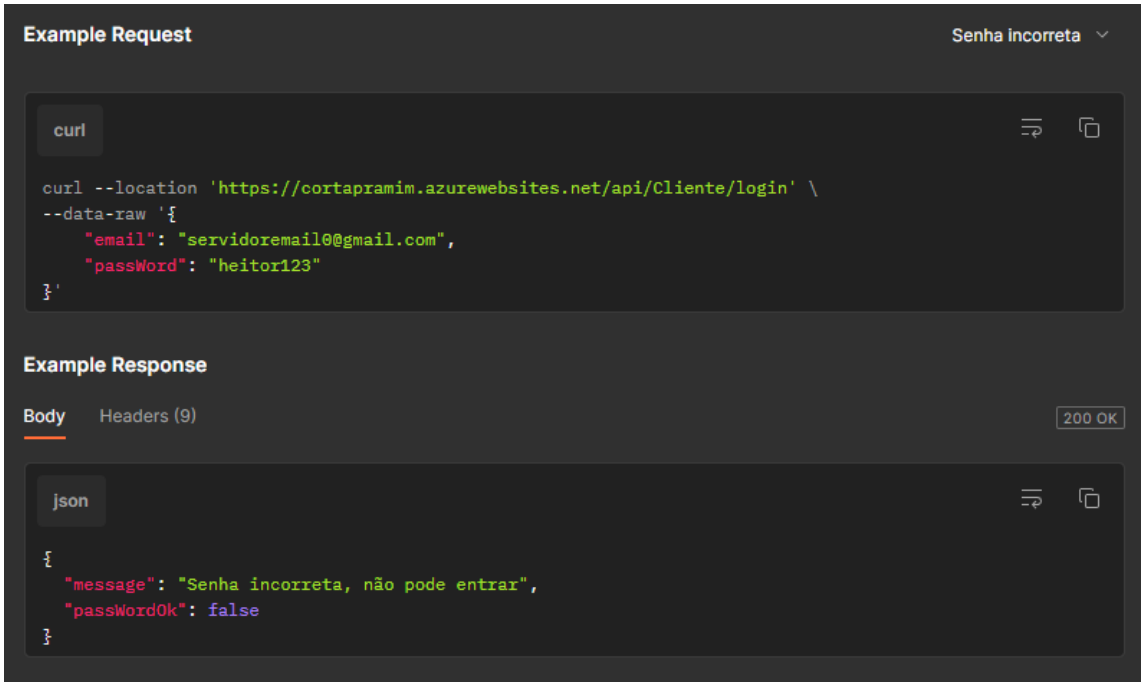
Create (figura 12): Esta é a função essencial para o funcionamento do sistema, sem a criação de uma conta, o cliente não consegue fazer o uso da aplicação. que recebe como parâmetros: e-mail, número de telefone, nome e senha. Realizado a validação dos dados na API, é criado um cliente.

Figura 12 - Função cliente create

Fonte: Autoria própria, 2024.

Na figura 13 é apresentado a função utilizada para o cliente entrar nos sistemas. Todas as aplicações utilizam esse método, tanto para clientes, quanto para usuários. Para clientes os parâmetros necessários são: e-mail e senha e para usuários *login* e senha. Estando certos os dados fornecidos, o cliente ou usuário recebe um JWT e acesso à aplicação. Lembrando que esse *Token JWT* é responsável por manter o ele *"logado"* na aplicação, dessa forma, o tempo do usuário é poupado quando for utilizar a o sistema de novo.

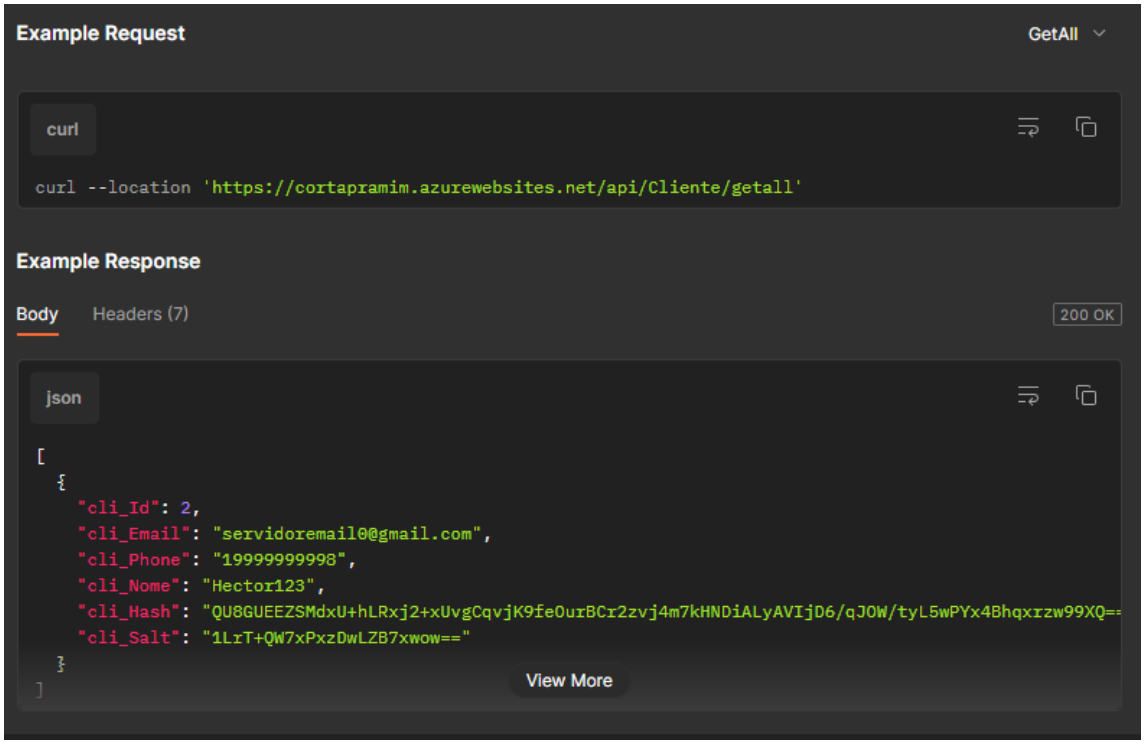
Figura 13 - Função cliente login



Fonte: Autoria própria, 2024.

Get All (figura 14): Utilizada para pesquisar todos os clientes, esta função, ao fazer a requisição, retornará uma lista com todos os clientes registrados no banco de dados.

Figura 14 - Função cliente get all



Fonte: Autoria própria, 2024.

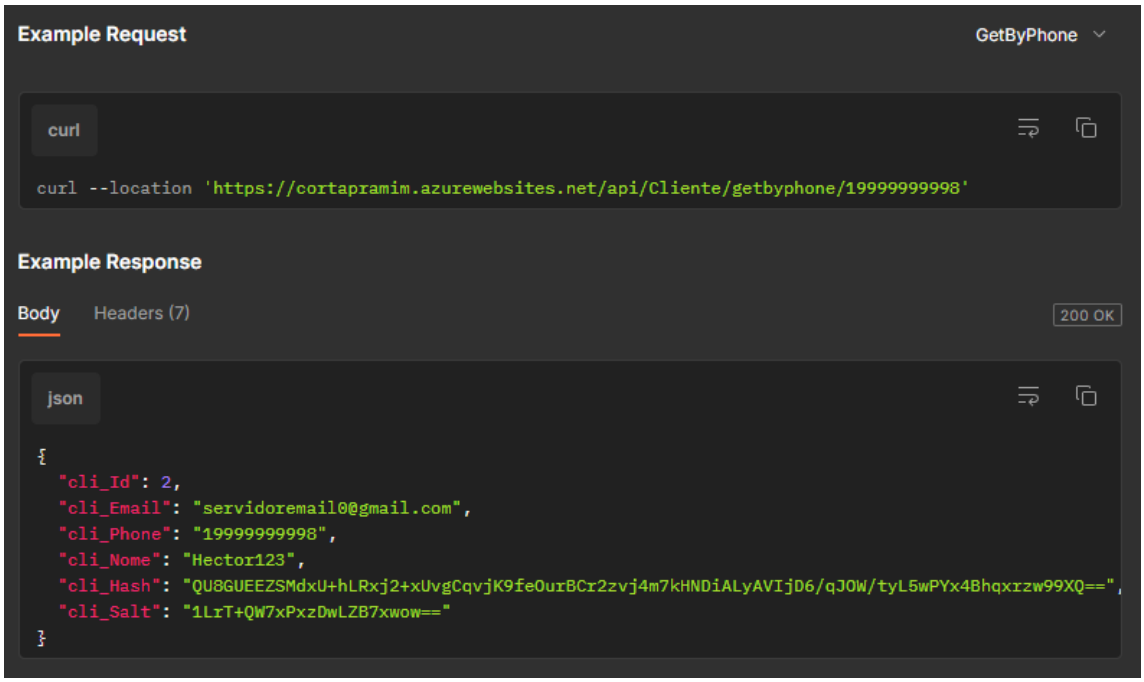
Get by Id: Esta é a função utilizada para pesquisar os clientes por id. Como pode ser visto na figura 15, ao fazer a requisição é retornado os dados do cliente com o id especificado.

Figura 15 - Função cliente get by id

Fonte: Autoria própria, 2024.

Get by Phone (figura 16): Este método recebe como parâmetro um número de telefone, que é usado como meio para pesquisar e então retorna os dados do cliente encontrado.

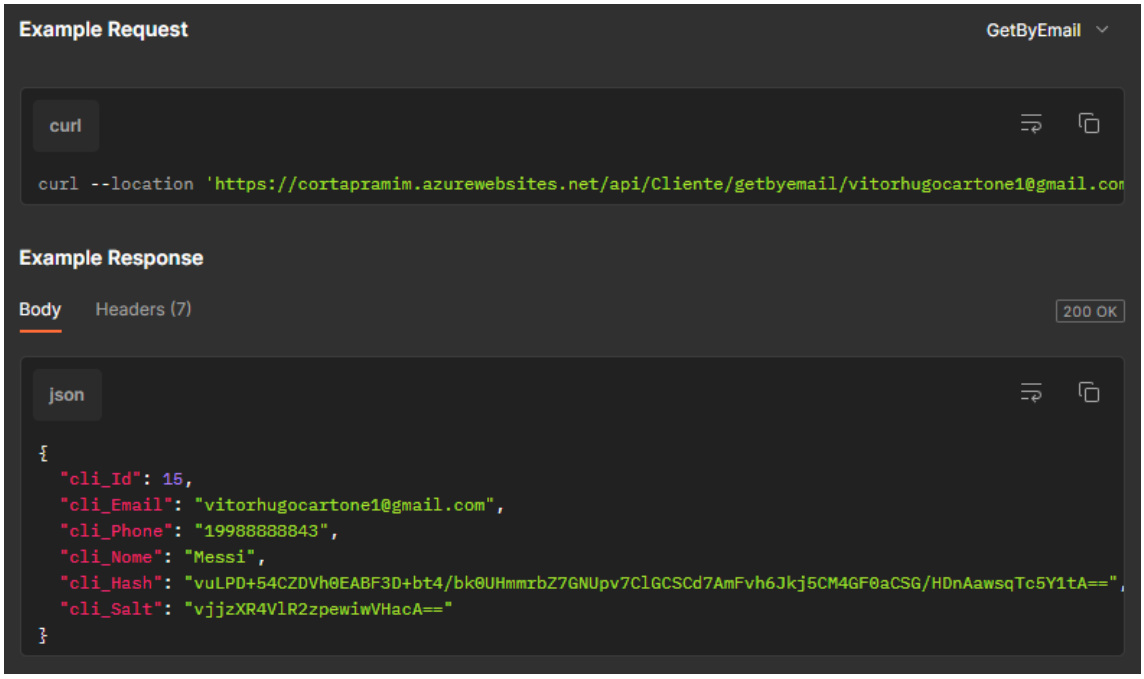
Figura 16 - Função cliente get by phone



Fonte: Autoria própria, **2024.**

Get by Email (figura 17): Este método recebe como parâmetro um e-mail, que é usado como meio para pesquisar e então retorna os dados do cliente encontrado.

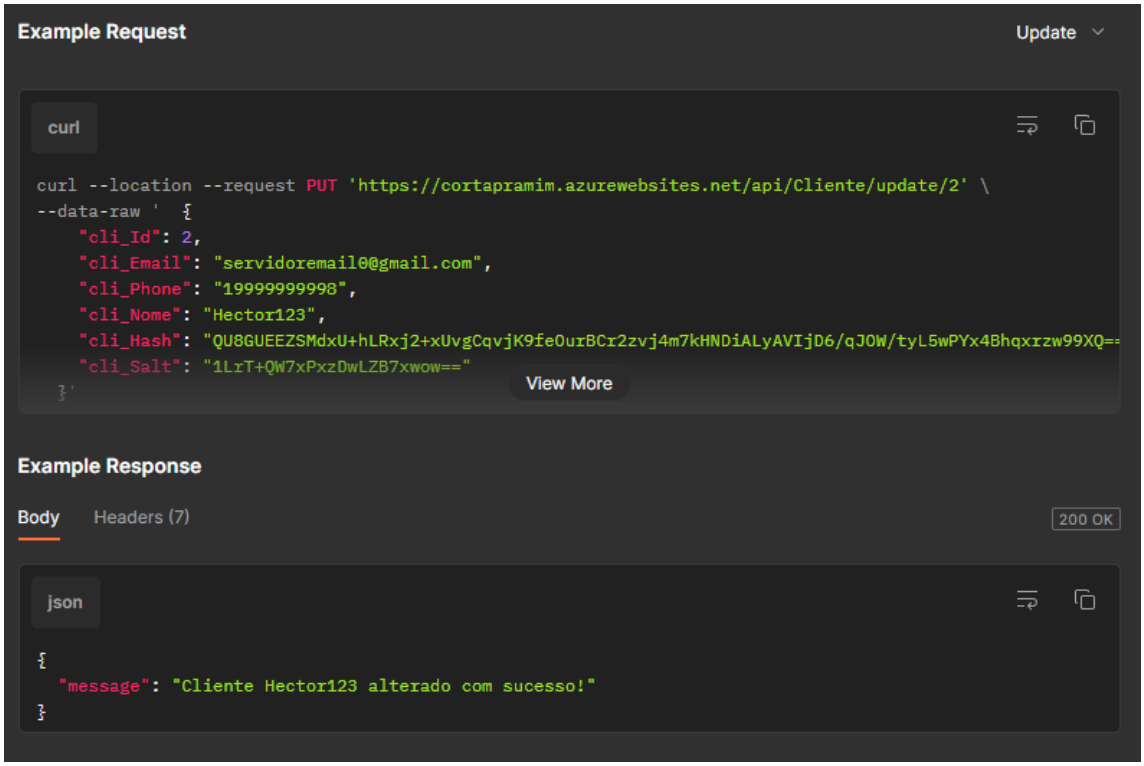
Figura 17 - Função cliente get by email



Fonte: Autoria própria, 2024.

A figura 18 ilustra o resultado de uma atualização nos dados do cliente. Todos os dados podem ser alterados com essa função. Muito útil para mudar informações relevantes que talvez tenha sido digitado erradas ou somente para modificar algum dado.

Figura 18 - Função cliente update



Fonte: Autoria própria, 2024.

Delete (figura 19): Esta é a função utilizada para deletar um cliente. Para excluir um cliente da base de dados é necessário informar o id na requisição. No contexto atual, essa função não foi amplamente utilizada.

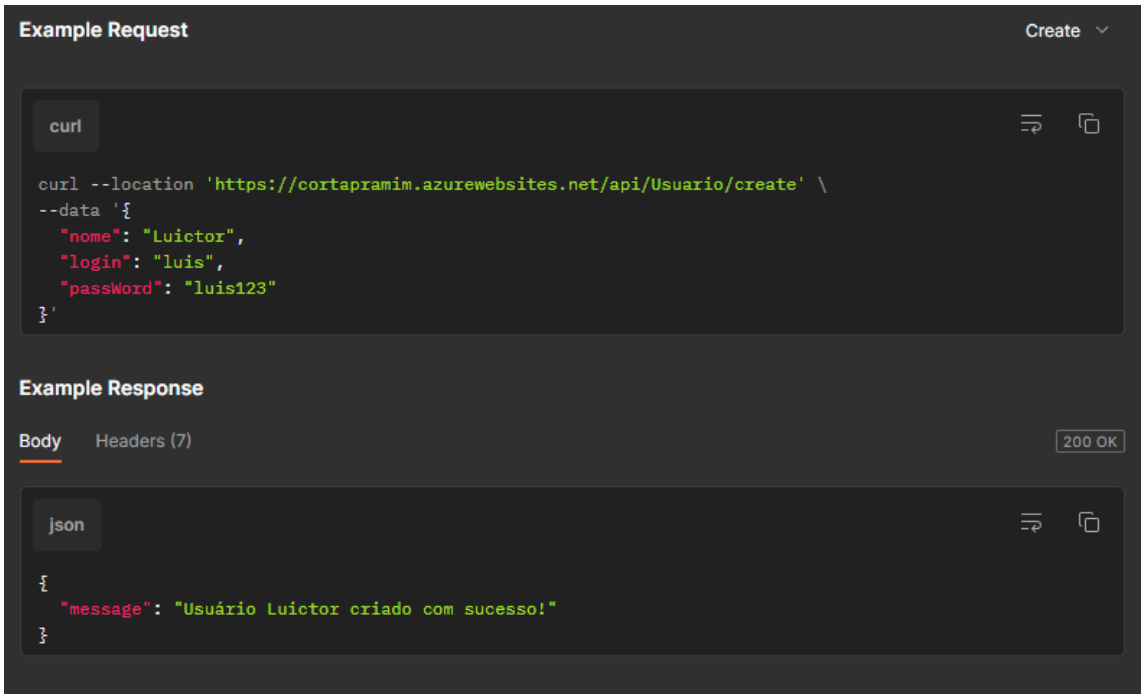
Figura 19 - Função cliente delete

Fonte: Autoria própria, **2024.**

4.1.1.2 Usuário

Create (figura 20): Esta é a função para criar um usuário, que recebe nome, login e senha. Realizando a validação dos dados na API, é criado um usuário. Possui grande utilidade quando um novo barbeiro começar a trabalhar no salão.

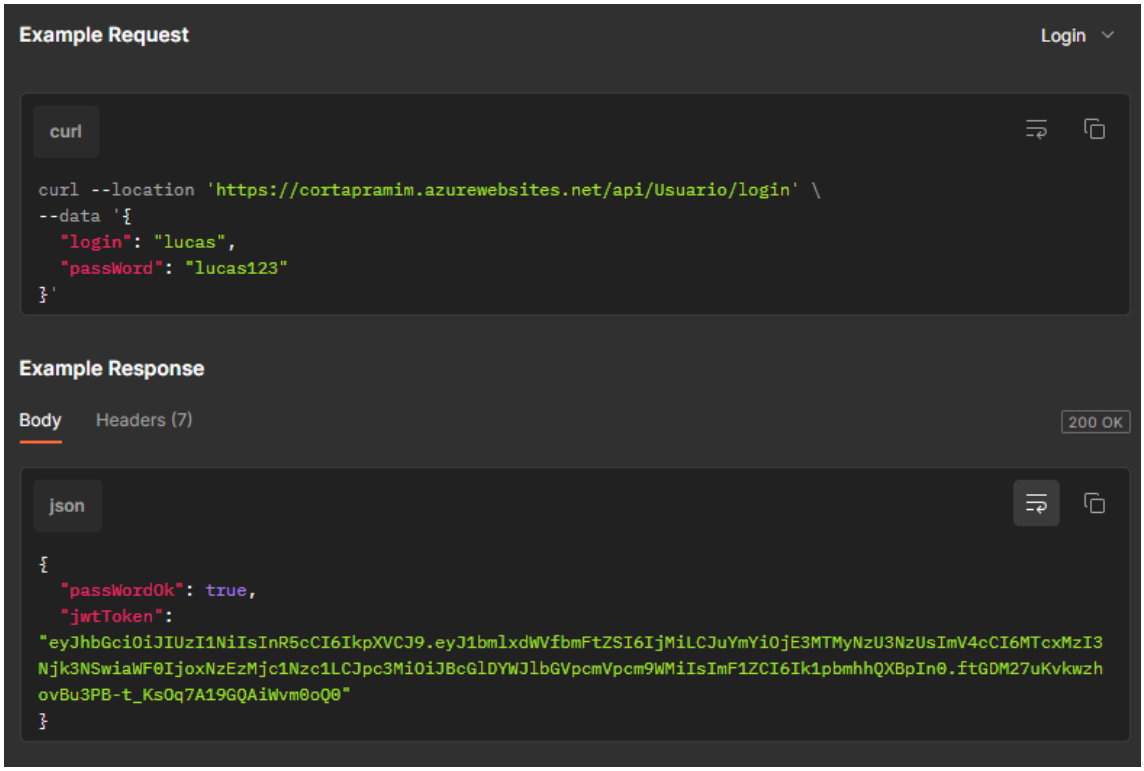
Figura 20 - Função usuário create



Fonte: Autoria própria, 2024.

Na figura 21 é apresentado a função utilizada para o usuário "*logar*" no sistema. Como foi mencionado anteriormente, é necessário o login e senha do usuário para que ele seja autenticado, dessa forma, ele recebe um *JWT Token* para realizar as operações dentro do sistema.

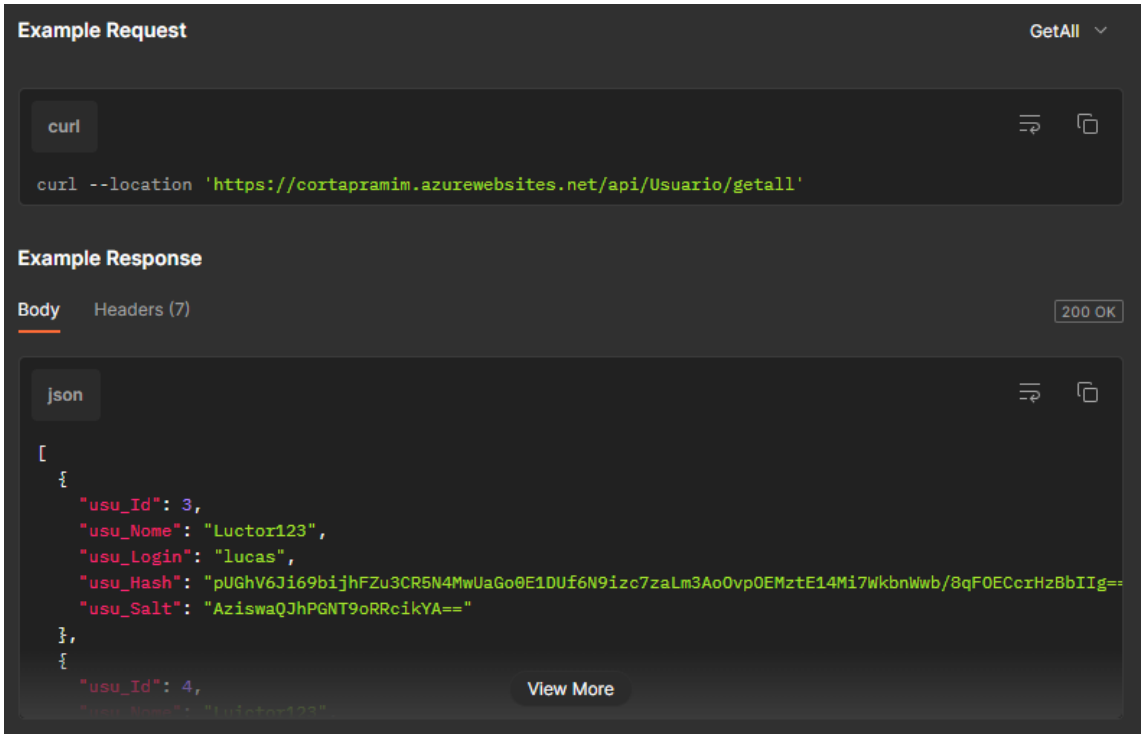
Figura 21 - Função usuário login



Fonte: Autoria própria, 2024.

Get All (figura 22): Esta é a função utilizada para pesquisar todos os usuários. Ao fazer a requisição, o método retorna uma lista com todos os usuários cadastrados no banco de dados. Útil para barbearias muito grandes com uma grande base de colaboradores.

Figura 22 - Função usuário get all



Fonte: Autoria própria, 2024.

Get by Id: Esta é a função utilizada para pesquisar os usuários por id. Como pode ser visto na figura 23, ao fazer a requisição é retornado os dados do usuário com o id especificado. Mais utilizado para verificar o usuário que está logado no sistema no momento.

Figura 23 - Função usuário get by id

Fonte: Autoria própria, 2024.

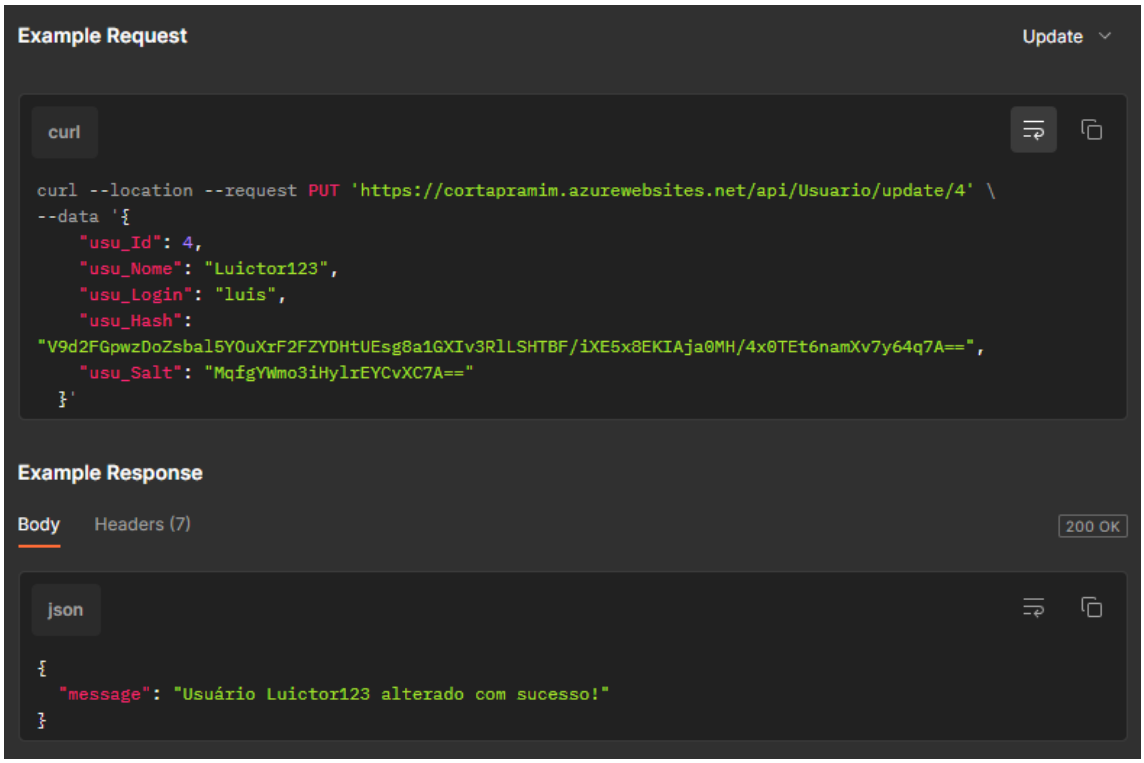
Get by Login (figura 24): Esta é a função utilizada para pesquisar os usuários por login. Informado o login na requisição, a API pesquisa pelo usuário correspondente e envia seus dados como resposta à requisição.

Figura 24 - Função usuário get by login

Fonte: Autoria própria, 2024.

Update (figura 25): Esta é a função utilizada para atualizar um usuário. Todos os dados do usuário podem ser alterados, desde que o dado alterado esteja no corpo da requisição. De grande ajuda caso algum barbeiro esquecer a senha e precisar alterar.

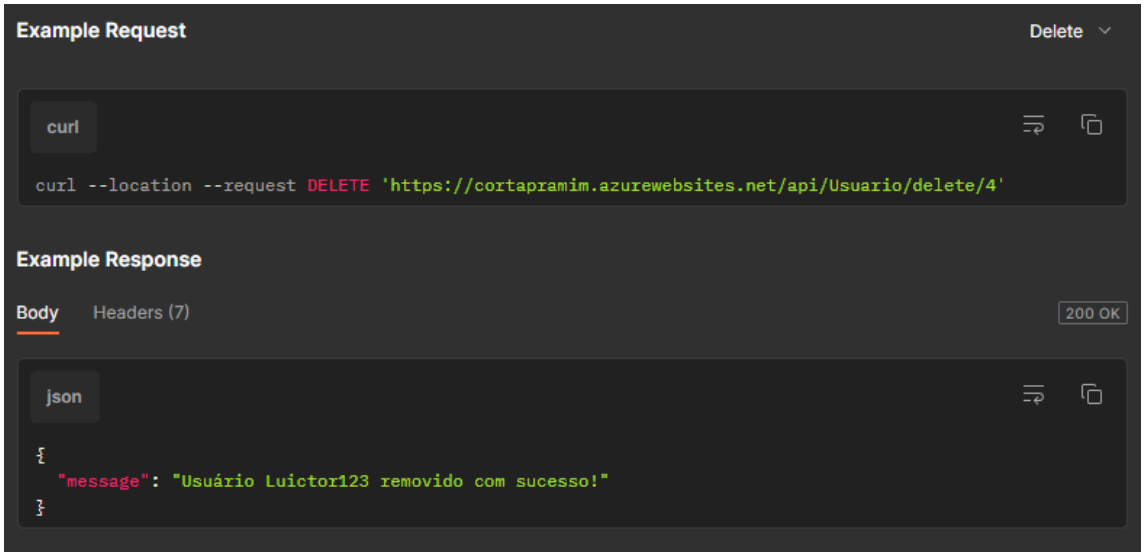
Figura 25 - Função usuário update



Fonte: Autoria própria, 2024.

A função de deletar um usuário do banco de dados pode ser vista na figura 26. Basta fornecer o id do usuário e fazer a requisição que o usuário será deletado. Quando algum colaborador for demitido, essa é a forma mais rápida de excluir os dados pertinentes a ele.

Figura 26 - Função usuário delete

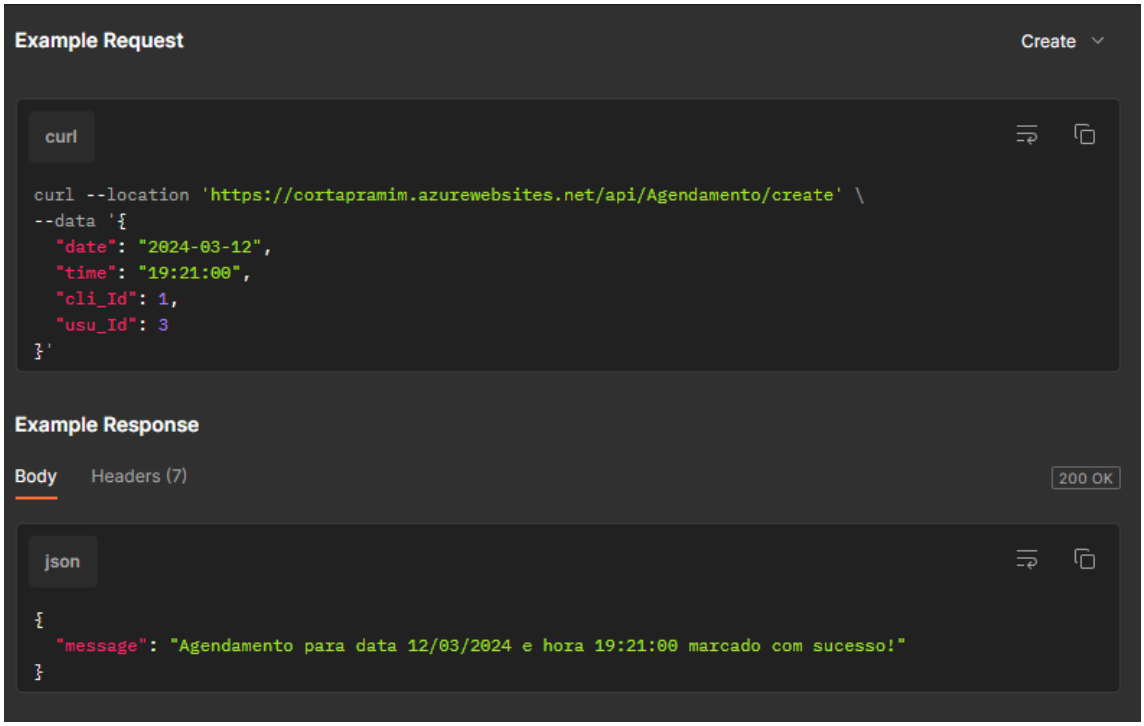


Fonte: Autoria própria, **2024.**

4.1.1.3 Agendamento

Create (figura 27): Esta é a função utilizada para criar um agendamento. Para criar um agendamento é necessário data, horário, id do cliente e id do usuário. Extremamente utilizada por todas as entidades e crucial para o funcionamento da aplicação.

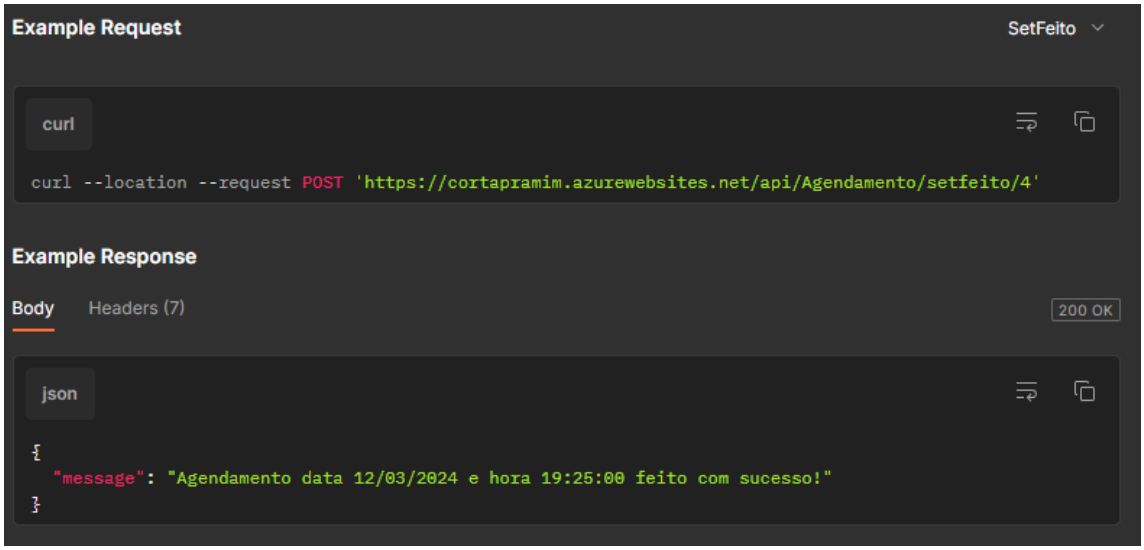
Figura 27 - Função agendamento create



Fonte: Autoria própria, **2024.**

setFeito: Quando criado um agendamento, há um campo “feito” indica que o agendamento foi realizado, sendo por padrão *false.* O método mostrado na figura 28 é utilizado para alterar esse campo para *true* ou *false*.

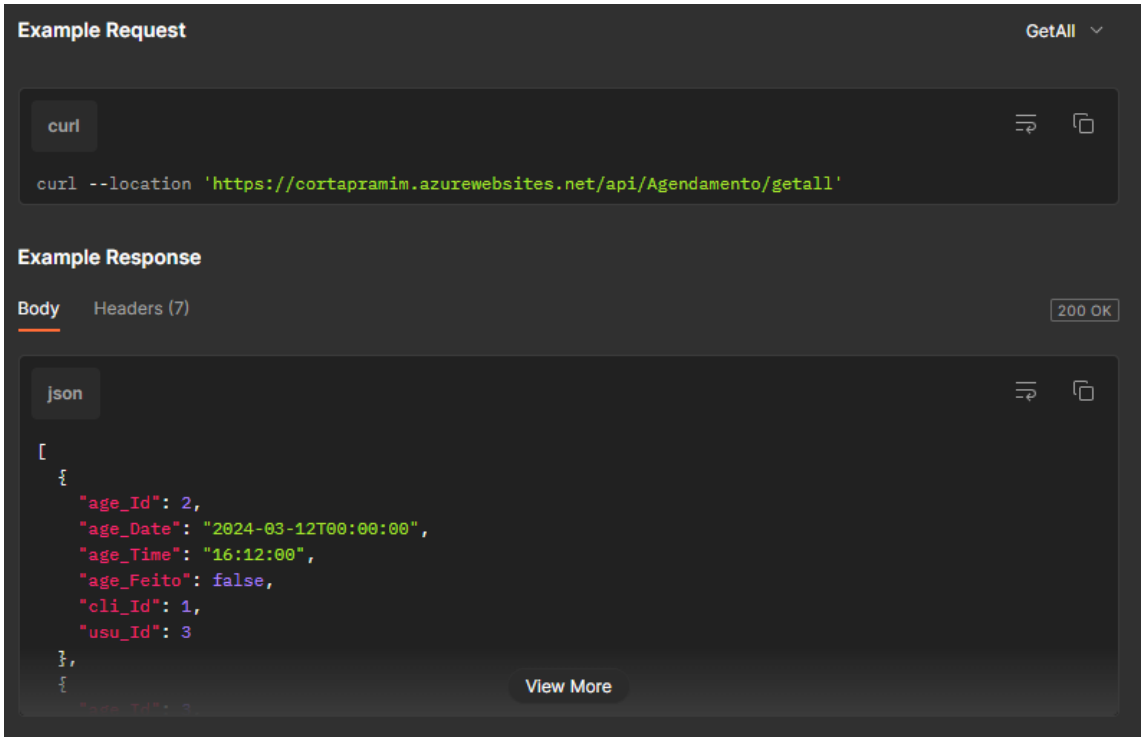
Figura 28 - Função agendamento set feito



Fonte: Autoria própria, **2024.**

Get All (figura 29): Esta é a função utilizada para consultar todos os agendamentos registrados no banco de dados. A finalidade principal é mostrar ao barbeiro todos os horários marcados do dia ou da semana, além de mostrar o cliente que agendou e o barbeiro que confirmou o agendamento

Figura 29 - Função agendamento get all



Fonte: Autoria própria, 2024.

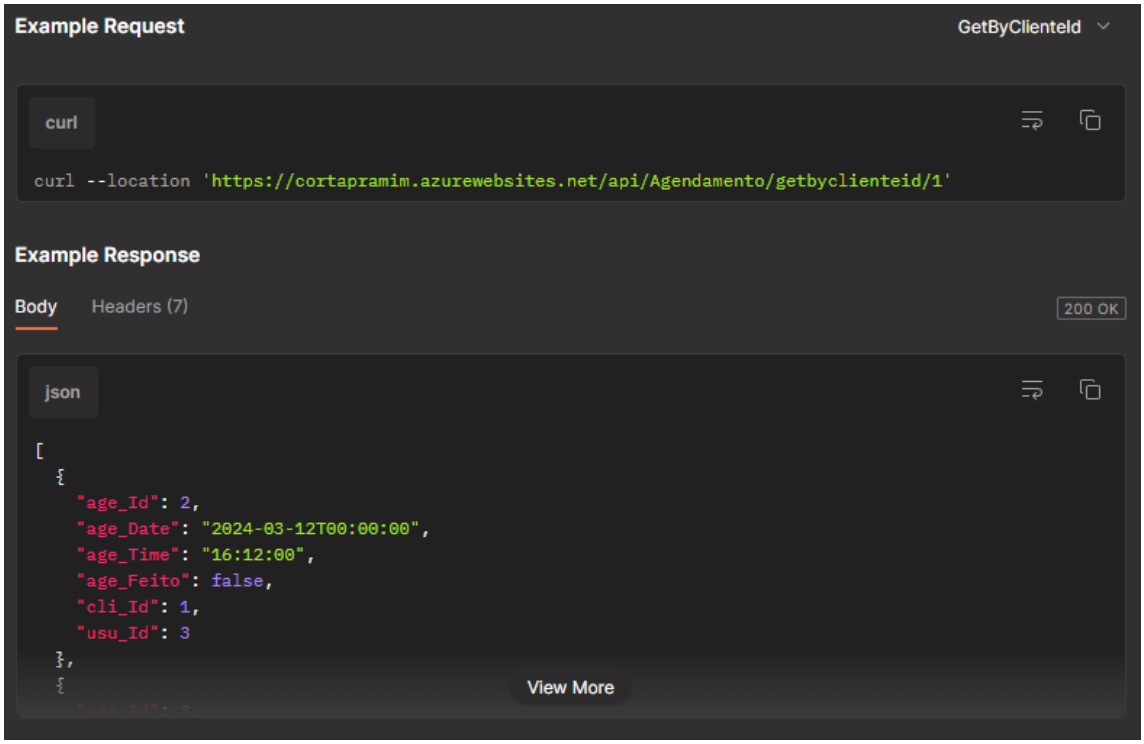
Get by Id: Esta é a função utilizada para pesquisar os agendamentos por id. Fornecendo o id na requisição, é retornados os dados do agendamento correspondente, como pode ser visto na figura 30. Também muito útil para o barbeiro realizar uma pesquisa mais minuciosa.

Figura 30 - Função agendamento get by id

Fonte: Autoria própria, 2024.

Get by Cliente Id (figura 31): Esta é a função utilizada para pesquisar os agendamentos pelo id do cliente. Fornecendo o id do cliente, é retornado todos os agendamentos por este criado. Essa função auxilia o barbeiro a encontrar um cliente específico, também pode se utilizado para outras finalidades no futuro como: detectar padrões de horário que o cliente frequenta o local.

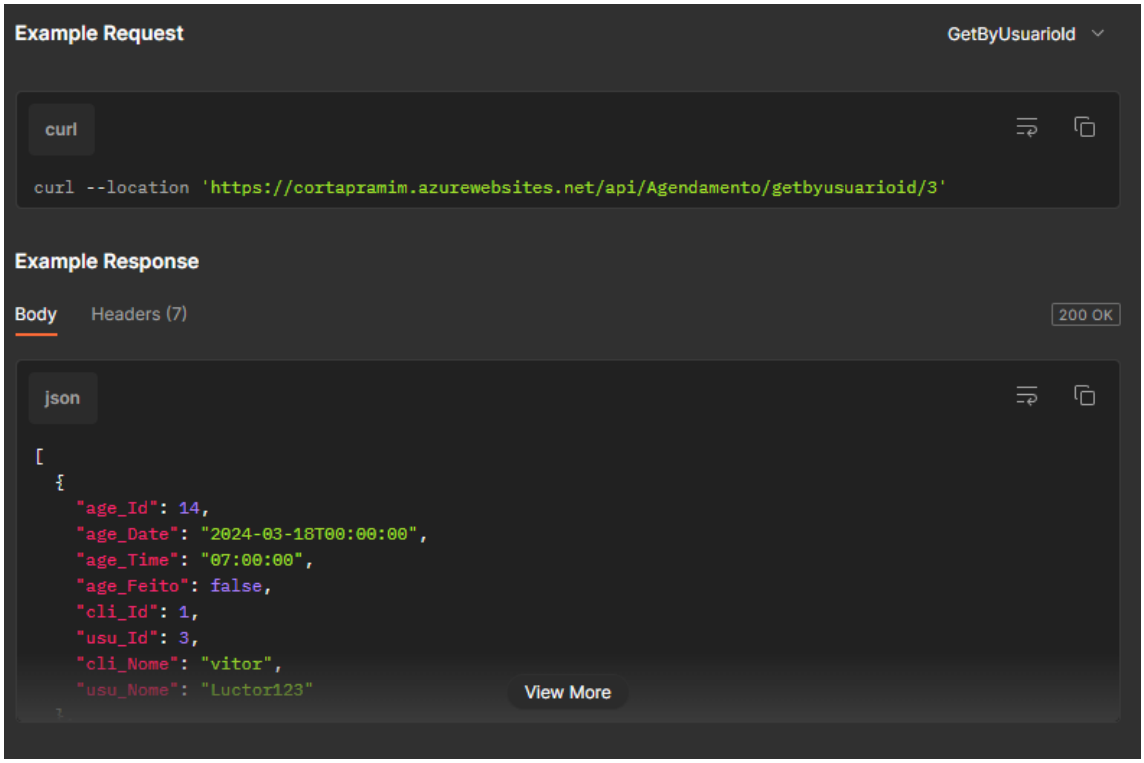
Figura 31 - Função agendamento get by cliente id



Fonte: Autoria própria, 2024.

Get By Usuario Id: Esta é a função utilizada para pesquisar os agendamentos pelo id do usuário. Fornecendo o id do usuário, é retornado todos os agendamentos por este criado, como pode ser visto na figura 32. Útil para os colaboradores verificarem os agendamentos que os mesmo confirmaram.

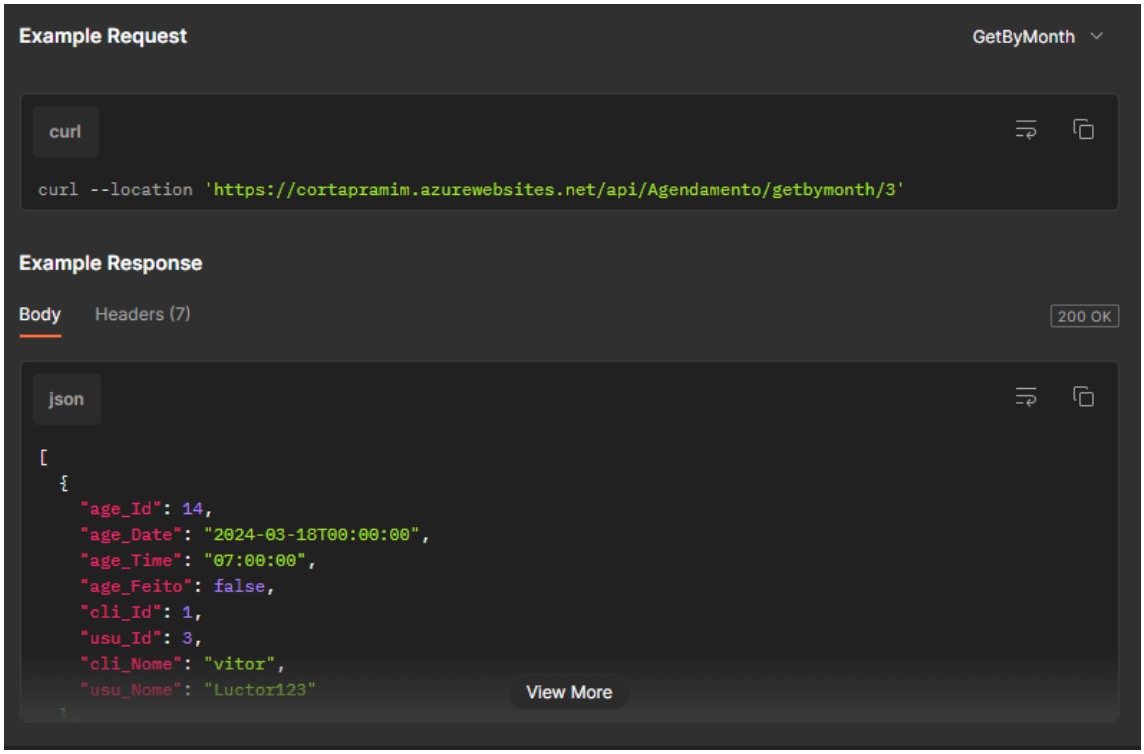
Figura 32 - Função agendamento get by usuario id



Fonte: Autoria própria, 2024.

Get By Month: É possível também filtrar os agendamentos por mês. Fornecendo o mês desejado na requisição, retornado assim todos os agendamentos do mês correspondente. O resultado pode ser visto na figura 33. Útil para tomar decisões de negócio como: quais meses a barbearia tem mais movimento, quais horários são os mais requisitados etc.

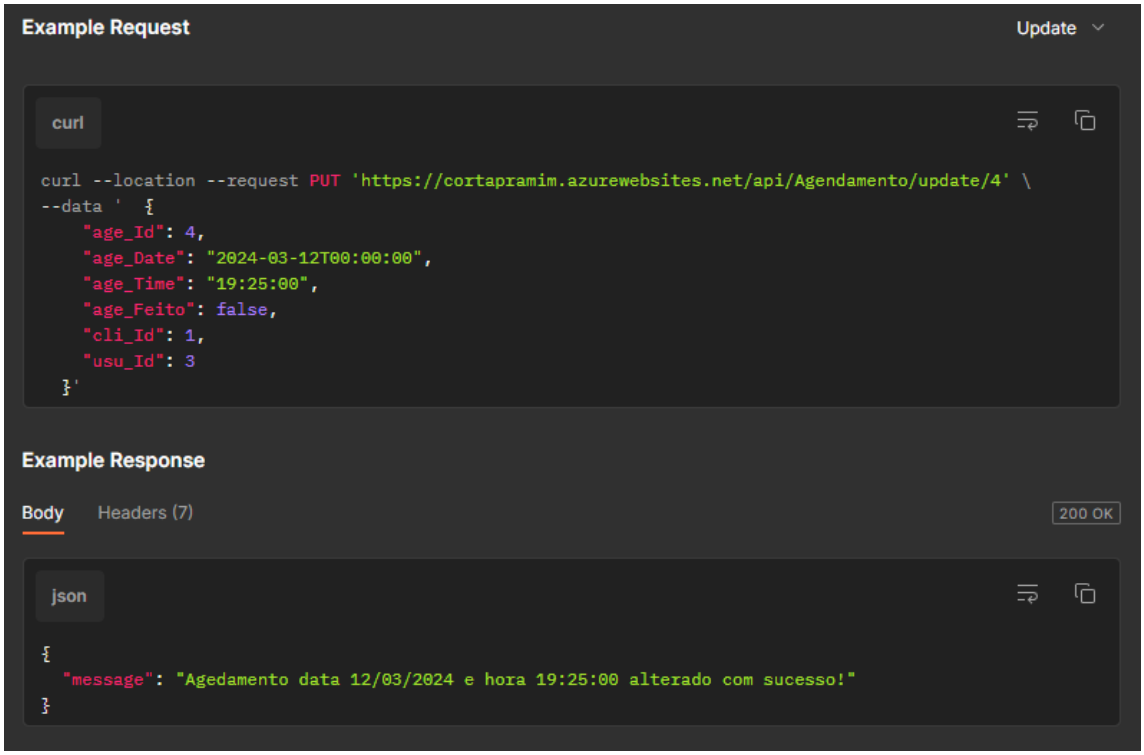
Figura 33 - Função agendamento get by month



Fonte: Autoria própria, 2024.

Update (figura 34): Esta é a função utilizada para atualizar um agendamento. Nela é possível alterar qualquer dado do agendamento. Caso o cliente decida mudar seu horário ou dia, essa função torna-se útil.

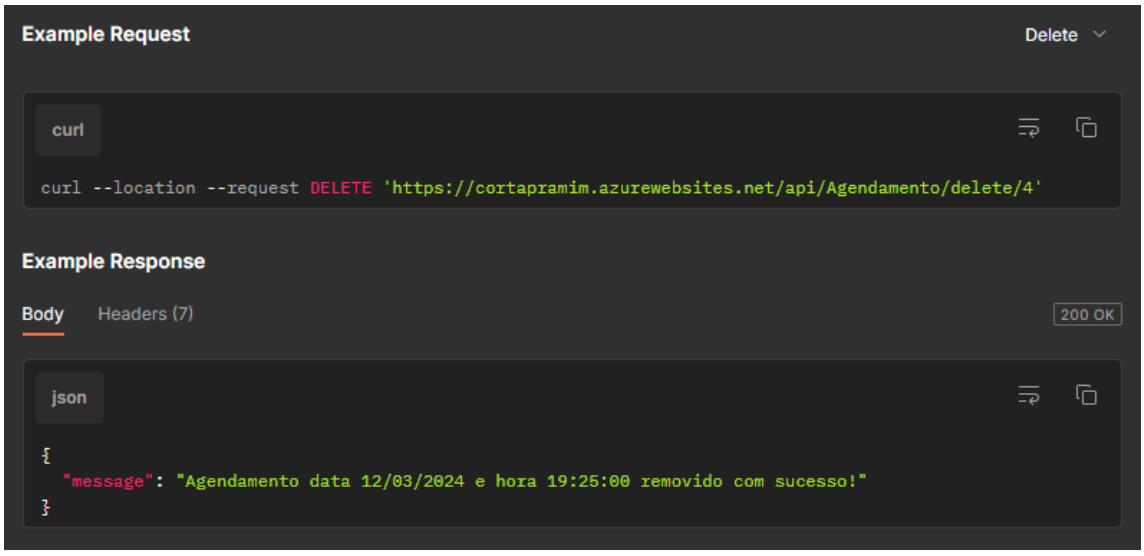
Figura 34 - Função agendamento update



Fonte: Autoria própria, 2024.

Delete (figura 35): Esta é a função utilizada para deletar um agendamento. Como parâmetro é necessário seu *id*. Função que proporciona mais controle ao cliente e facilita o trabalho do barbeiro. Caso algo dê errado na agenda, basta excluir o agendamento, dessa forma, é liberado um novo horário para o barbeiro.

Figura 35 - Função agendamento delete

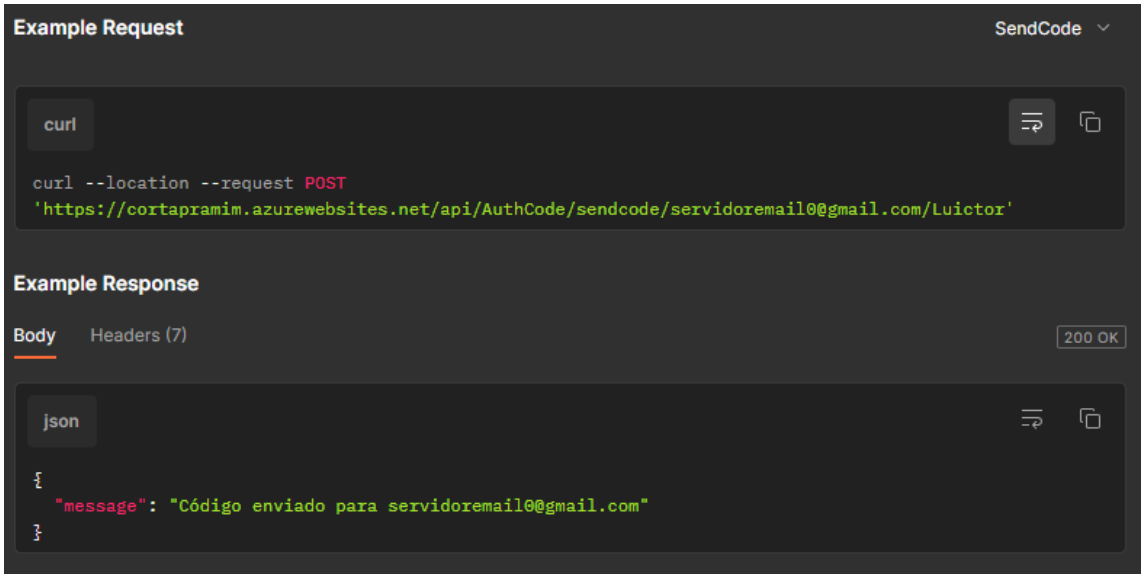


Fonte: Autoria própria, 2024.

4.1.1.4 AuthCode

Send Code (figura 36): Esta é a função utilizada para enviar o código de autenticação via e-mail para o cliente. O código é usado para realizar a autenticação de um novo cliente. Função que trabalha em conjunto com a próxima.

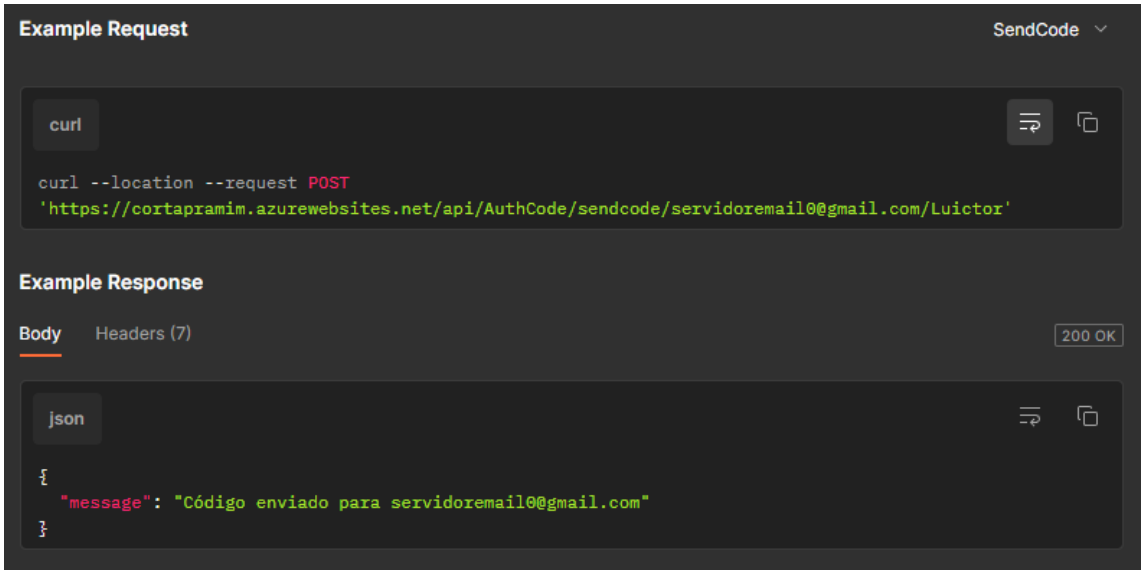
Figura 36 - Função authcode send code



Fonte: Autoria própria, 2024.

Authenticate (figura 37): Esta é a função utilizada para verificar o código de autenticação. O código gerado pela API é validado através deste método. O código estando correto o cliente está autenticado. Extremamente importante para a segurança da aplicação e clientes como um todo, dessa forma, a identidade é confirmada antes de iniciar uma nova sessão.

Figura 37 - Função authcode authenticate



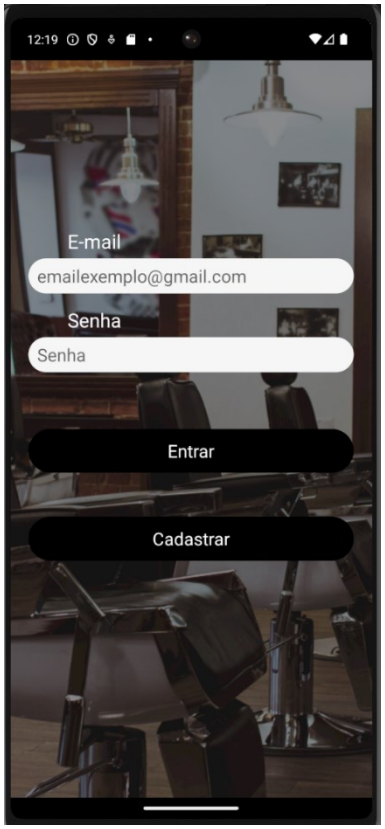
Fonte: Autoria própria, 2024.

4.1.2 Aplicativo Mobile

Esta aplicação feita no mobile é de uso exclusivo do cliente, ou seja, é utilizado apenas por quem frequenta a barbearia. O Mobile é dividido em telas.

Tela de login (figura 38): Essa é a tela inicial do sistema, onde o cliente para entrar em sua conta insere seu e-mail e senha previamente criados, caso o cliente não possua uma conta ele poderá criar uma conta ao clicar no botão cadastrar.

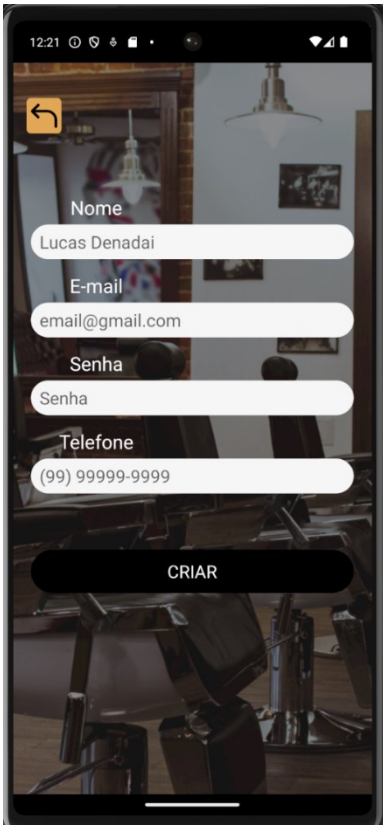
Figura 38 – Tela de Login



Fonte: Autoria própria, 2024.

Tela de criação de conta (figura 39): Essa é a tela onde possibilita o cliente criar sua conta ao clicar no botão “Cadastrar” na tela de Login. Ela possibilita a criação de uma nova conta. Para a realização do cadastro o cliente deverá digitar o nome, e-mail, senha e telefone.

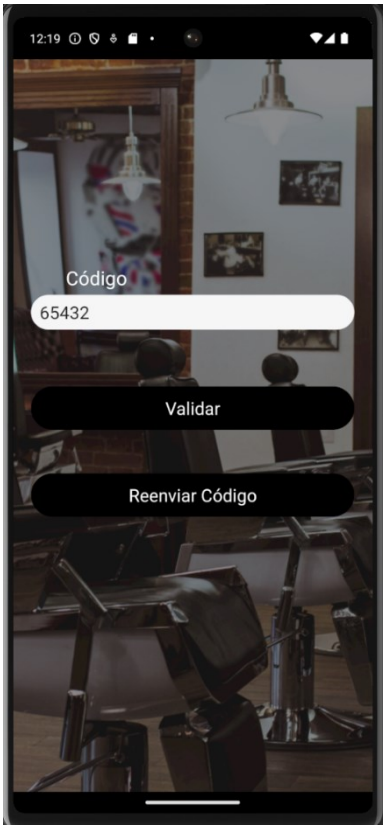
Figura 39 – Tela de criação de conta



Fonte: Autoria própria, 2024.

Tela de autenticação de código (figura 40): Essa tela é onde o cliente vai fazer a Autenticação de Dois Fatores. Nela o cliente pode autenticar sua conta ou pedir para reenviar o e-mail de confirmação caso houver algum problema.

Figura 40 – Tela de autenticação de dois fatores



Fonte: Autoria própria, 2024.

Tela do calendário (figura 41) – Essa é a tela “principal” do sistema, onde possibilita o cliente selecionar a data desejada para marcar o corte. Nela ele já pode ver as outras opções presentes na aba inferior: ver o calendário, consultar os agendamentos já feitos e acessar as opções adicionais do sistema. No calendário é impossível o usuário agendar para dias anteriores (dias transparentes) e é fornecido a opção de agendar até 3 meses seguintes

Figura 41 – Tela de calendário

Fonte: Autoria própria, 2024.

Tela de seleção de horário (figura 42) – Essa é uma “caixa” que aparece quando o cliente seleciona o dia desejado, nela é possível consultar os horários disponíveis para aquele dia. Durante a abertura é feita uma consulta no Banco de Dados os horários já agendados por outros clientes, dessa forma permitindo que não haja conflito de interesses.

Figura 42 – Tela de seleção de horas

Fonte: Autoria própria, 2024.

Tela de confirmação de informações (figura 43): Nessa tela, é fornecido um *feedback* ao cliente para que ele tenha certeza de suas informações. É dado a opção de confirmar o agendamento ou editar os dados do agendamento, se ele optar por alterar algo, a tela retorna para a seleção de dias do calendário.

Figura 43 – Tela de confirmação de informações

Fonte: Autoria própria, 2024.

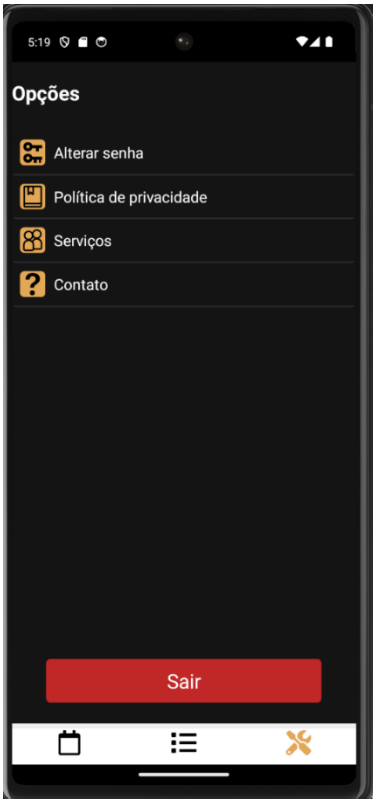
Tela de consulta de agendamentos (figura 44) – Essa tela é acessada via menu inferior clicando no ícone de lista ao meio da tela, nela é possível o cliente consultar seus agendamentos feitos e os excluir caso seja de sua necessidade.

Figura 44 – Tela de consulta de agendamentos

Fonte: Autoria própria, 2024.

Tela de opções (figura 45) – Essa tela também é acessada via barra inferior clicando no ícone da ferramenta ao lado direito. Nela é possível o cliente alterar sua senha, consultar a política de privacidade, consultar os serviços que a barbearia oferece, ver formas de contato. Além disso há o botão “Sair” que faz o cliente voltar à tela de login.

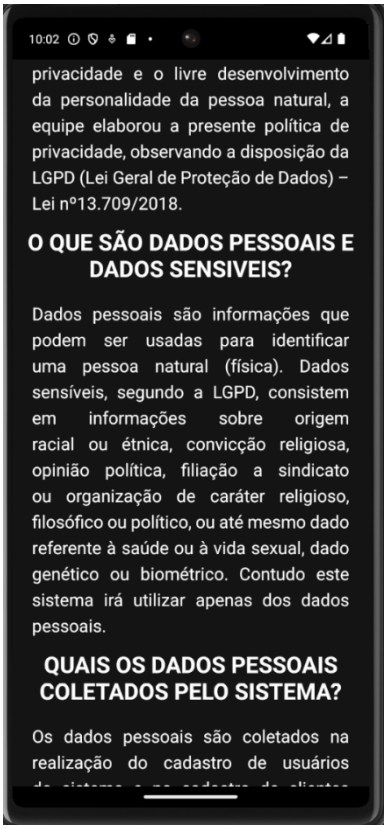
Figura 45 – Tela de opções



Fonte: Autoria própria, 2024.

Tela de política de privacidade (figura 46) – Nessa tela é possível o cliente consultar como suas informações estão sendo tratadas pelo sistema, deixando tudo transparente.

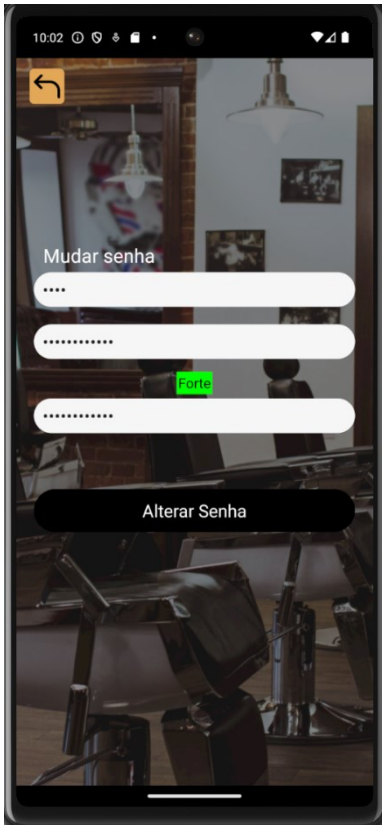
Figura 46 – Tela de Política de Privacidade



Fonte: Autoria própria, 2024.

Tela de alterar a senha (figura 47) – Nessa tela é fornecido ao cliente uma forma de alterar a senha de sua conta. Para isso, basta ele entrar com sua senha atual como forma de segurança, inserir a nova senha e confirmá-la. É possível o cliente ter ciência da força de sua nova senha com o *feedback* visual que é fornecido abaixo do campo da senha.

Figura 47 – Tela de alteração de senha



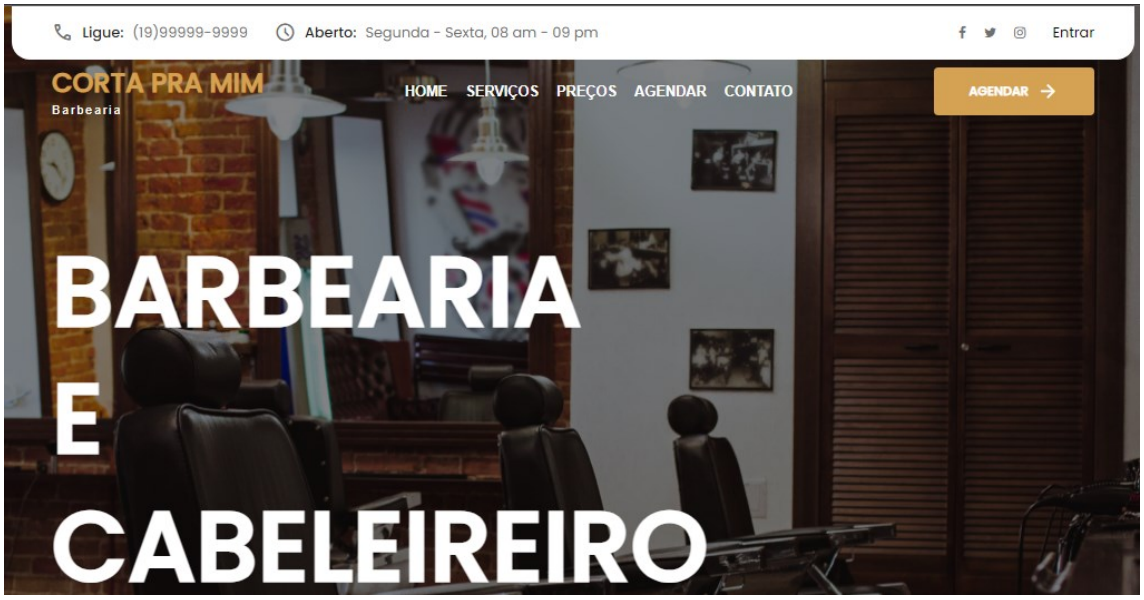
Fonte: Autoria própria, 2024.

4.1.3 Aplicativo Web

Esta aplicação feita no mobile é de uso exclusivo do cliente, ou seja, é utilizado apenas por quem fará o uso dos serviços da barbearia. O Web é dividido em páginas.

Página de início (figura 48): Essa é a página inicial do sistema, onde o cliente pode entrar em sua conta inserindo seu login e senha ou apenas navegar pelo site. Nessa parte inicial é possível ver o *banner,* serviços prestados, preço e informações de contato.

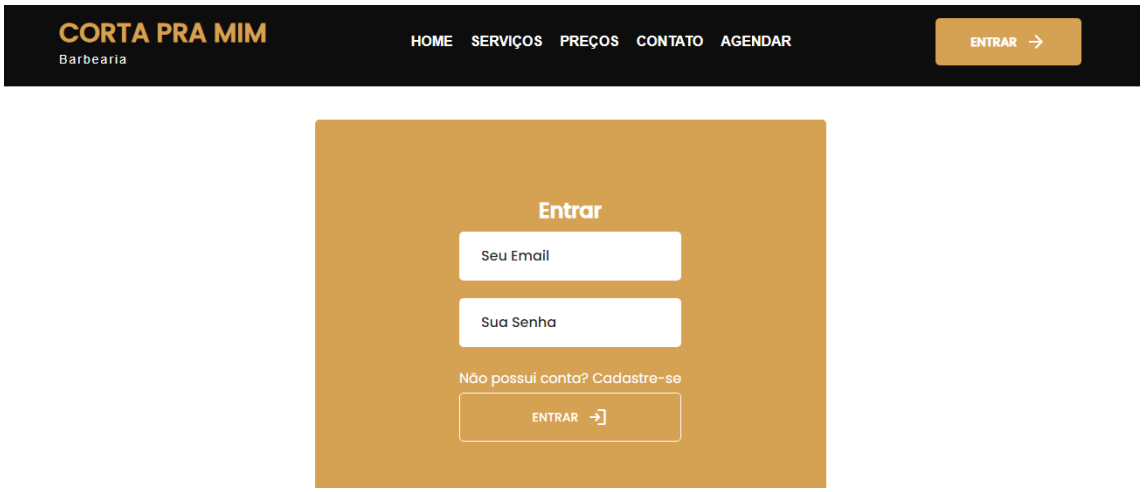
Figura 48 – Página de início



Fonte: Autoria própria, 2024.

Página de login (figura 49): Nessa página o cliente pode entrar em sua conta inserindo seu e-mail e senha previamente criados, caso o cliente precise criar a conta ele poderá fazer isso interagindo com o botão de cadastro.

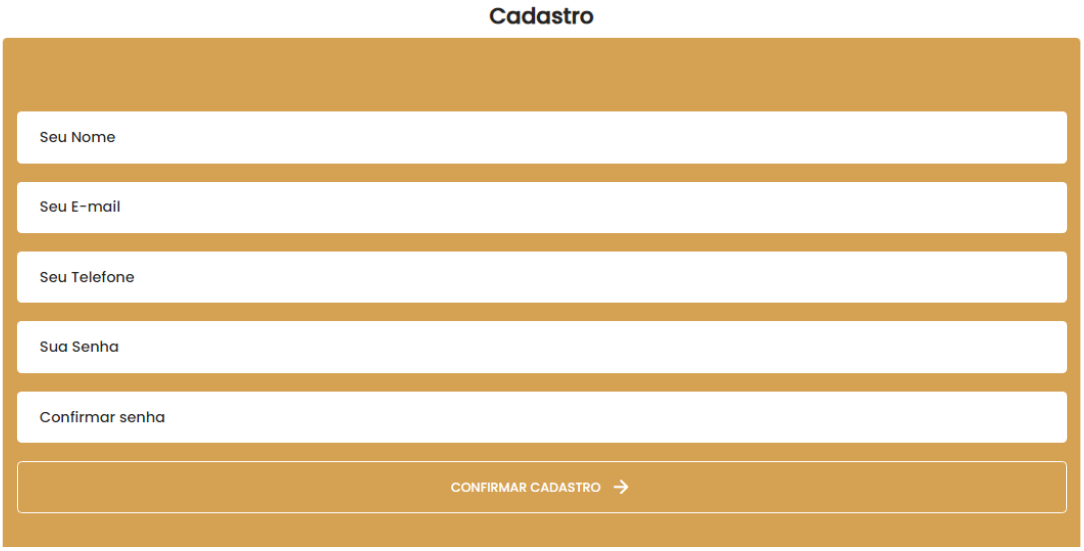
Figura 49 – Página de login



Fonte: Autoria própria, 2024.

Página de cadastro (figura 50): Nessa página é possível realizar o cadastro de um novo cliente. Sendo preciso digitar o nome, o e-mail, o telefone e uma senha, esta última sendo preciso confirmá-la

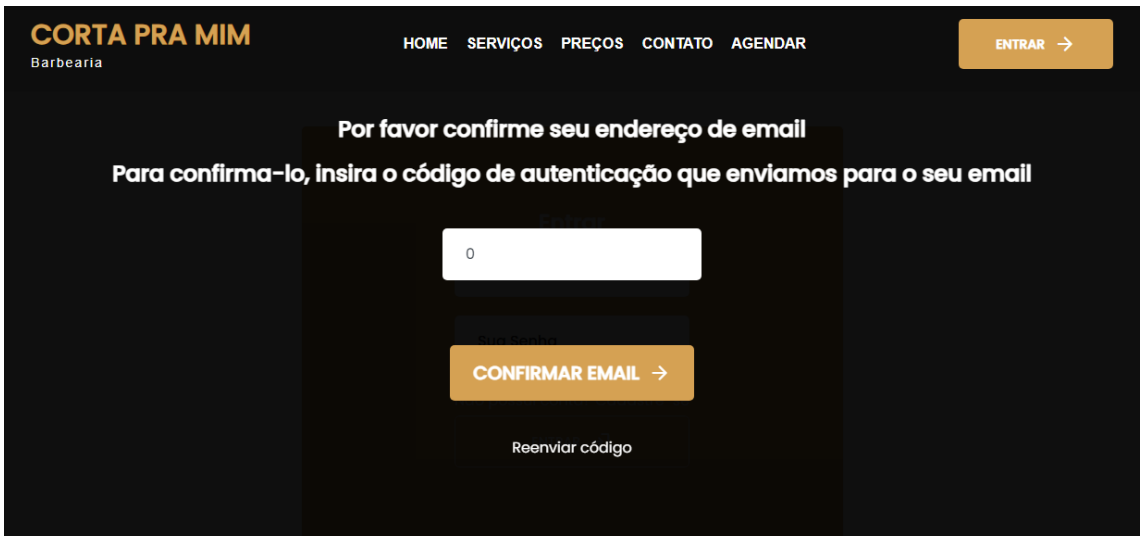
Figura 50 – Página de cadastro



Fonte: Autoria própria, 2024.

Página do código de autenticação (figura 51): Nessa página o cliente insere o código enviado para o *e-mail* para poder validar a conta. Essa página é muito importante pois ajudam a confirmar a identidade do cliente ao começar fazer o uso da aplicação em um novo dispositivo. Com a validação realizada, o cliente pode iniciar sua sessão no site.

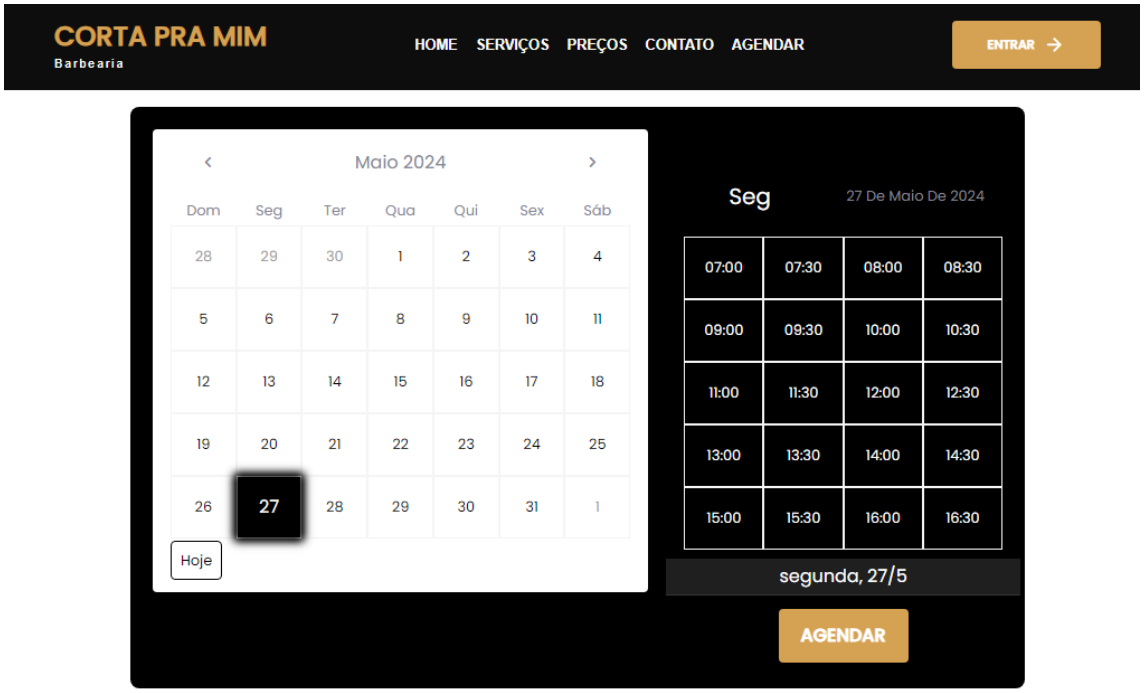
Figura 51 – Página de código de autenticação



Fonte: Autoria própria, 2024.

A página onde o cliente pode ver os dias e horário para realizar um agendamento, além disso a data e horários selecionados poderão ser visualizados acima do botão agendar, tudo isso pode ser visto na figura 52. Os horários que já estiverem ocupados, ficarão destacados em branco, impossibilitando a realização de um agendamento duplicado.

Figura 52 – Página da agenda



Fonte: Autoria própria, 2024.

Página de confirmar agendamento (figura 53): Essa página é onde o cliente pode confirmar as informações que acabou de interagir. Se entrou com algum dado indesejado, basta voltar a página. Isso é útil para evitar dados errôneos para o barbeiro.

Figura 53 – Página de confirmar agendamento

Fonte: Autoria própria, 2024.

Página de lista de agendamentos (figura 54): Nessa página é possível, caso o cliente esteja logado, visualizar os agendamentos realizados. Caso ele queira, pode desmarcar o agendamento interagindo com o botão. Isso agiliza bastante os serviços da barbearia. Dessa forma, ela não precisa desmarcar manualmente cada cliente que deseja desistir de seu horário.

Figura 54 – Página da lista de agendamentos

Fonte: Autoria própria, 2024.

Página de alterar senha (figura 55): Nessa página é possível, caso o cliente esteja logado, atualizar a senha, tendo que digitar sua senha atual e uma nova.

Confirmando a alteração, essa nova informação já entrará em vigor e o cliente pode iniciar sua sessão com a nova senha.

Figura 55 - Página de alterar senha

Fonte: Autoria própria, 2024.

4.1.4 Aplicativo Desktop

Esta aplicação feita para desktop é de uso exclusivo do usuário, ou seja, é utilizado apenas pelo barbeiro. Ele é dividido em telas.

Tela de login (figura 56): Nesta tela o barbeiro, deve colocar seu nome de usuário (*login*) e sua senha, após isso clicando no botão entrar, uma requisição será enviada à API para que assim o barbeiro seja “logado”.

Figura 56 - Tela de *login* Desktop

Fonte: Autoria própria, 2024.

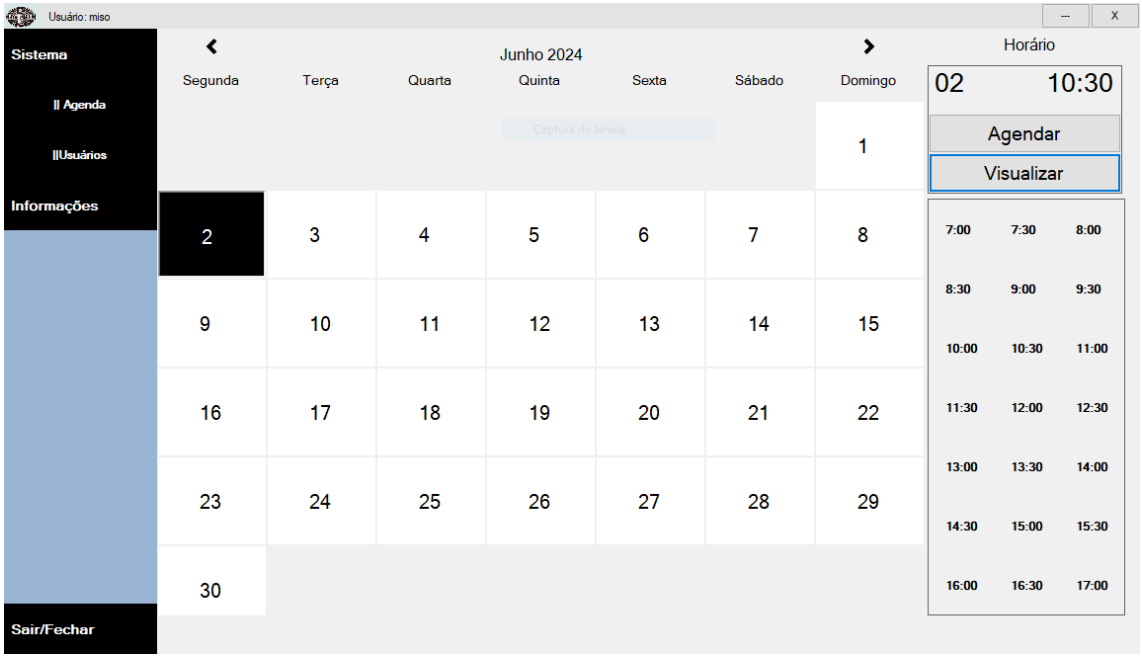
Tela inicial (figura 57): Esta é a principal tela do sistema, a partir desta tela que o usuário poderá acessar as funcionalidades do sistema, nela há 2 botões principais, o botão sistema e o botão informação, após clicar neles novos botões irão aparecer, sendo eles o botão da agenda, usuário, política de privacidade, manual e sobre. Sendo que, o botão usuário, possui dois outros botões, sendo eles o de cadastro e o de visualizar.

Figura 57 - Tela inicial Desktop

Fonte: Autoria própria, 2024.

Tela da agenda (figura 58): Nesta tela o usuário deve selecionar o dia e o horário para que o agendamento ocorra, vale ressaltar que o dia e o horário selecionados serão exibidos em uma *label*. Útil para agendar cliente que ainda não fazem o uso das aplicações *Mobile* ou *Web.*

Figura 58 - Tela da agenda Desktop



Fonte: Autoria própria, 2024.

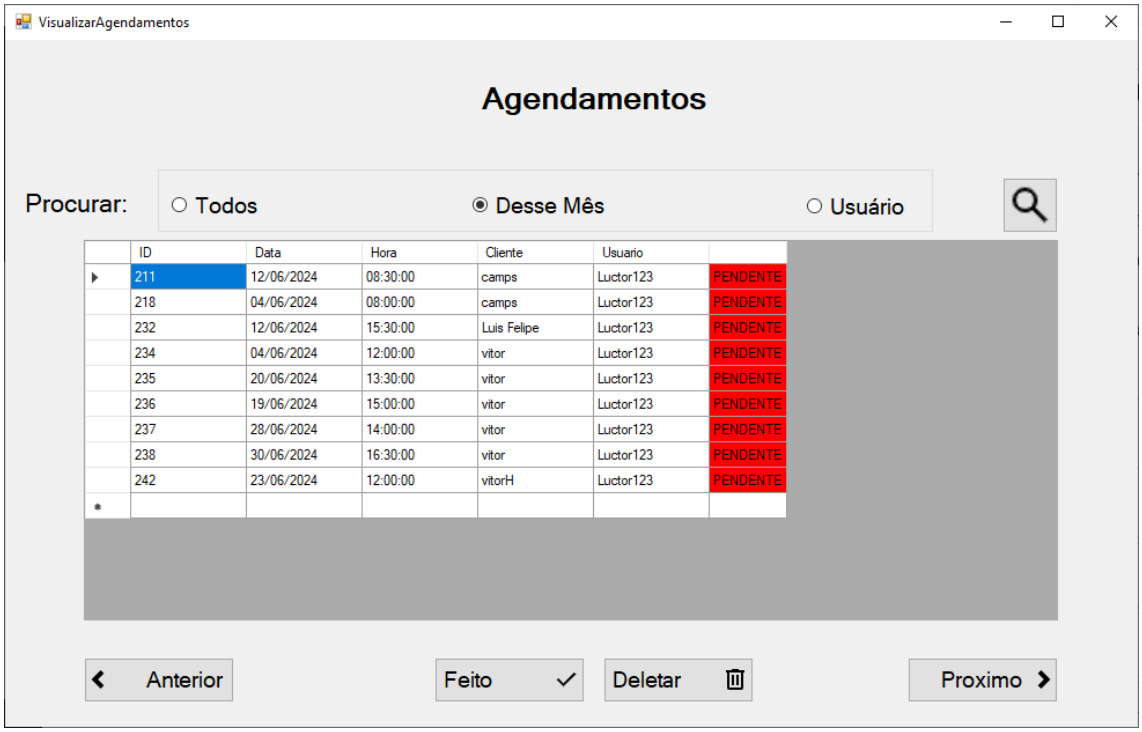
Tela de agendamento (figura 59): Após clicar no botão agendar da tela anterior esta tela será aberta e nela o usuário fará o agendamento, a partir da data e horário selecionados, previamente, na tela da agenda, para o agendamento ser efetuado o usuário deverá informar o nome, o e-mail e o telefone do cliente, se for preciso cadastrar um novo cliente, o *check box* deverá ser selecionado, caso isso aconteça uma senha aleatória será gerada para o primeiro *login* do cliente no Mobile ou Web. Após isso é só clicar em concluir e o agendamento será efetuado, o usuário poderá cancelar a operação a qualquer momento.

Figura 59 - Tela de agendamento Desktop

Fonte: Autoria própria, 2024.

Tela de visualizar o agendamento (figura 60): O usuário, ao invés de efetuar um agendamento, pode optar por clicar no botão visualizar para verificar os agendamentos, sejam eles do mês, feitos pelo próprio usuário ou todos os agendamentos. Além disso, o usuário pode navegar entre os agendamentos, marcar eles como feitos ou deletá-los, esses dois últimos tendo que selecionar um agendamento antes de efetuar a operação.

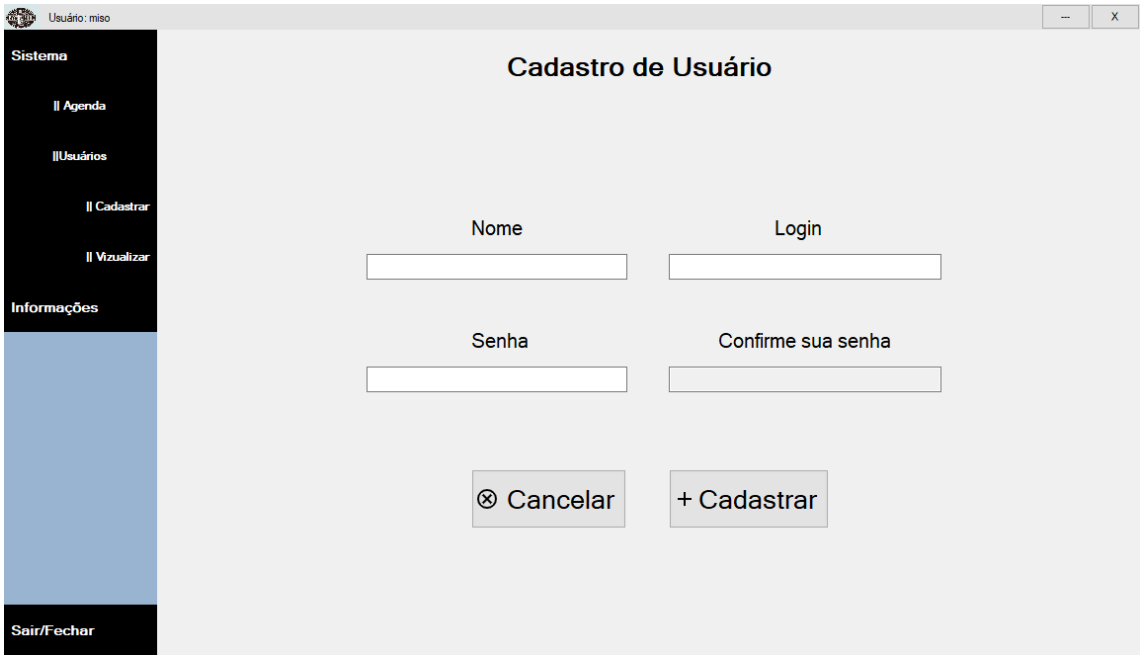
Figura 60 - Tela de visualizar agendamento Desktop



Fonte: Autoria própria, 2024.

Tela de cadastro de usuário (figura 61): Nesta tela podem ser cadastrados outros usuários para o sistema, tendo que digitar nome, *login* e senha, essa última tendo que ser confirmada, o usuário tem a opção de cancelar a operação a qualquer momento.

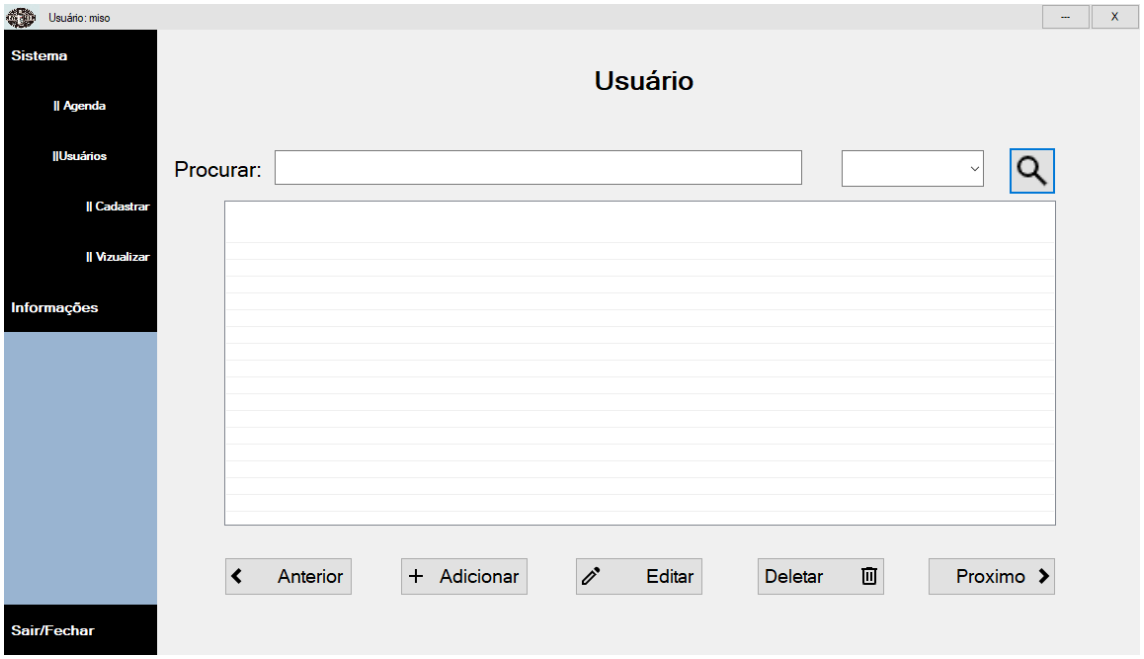
Figura 61 - Tela de cadastro de usuário Desktop



Fonte: Autoria própria, 2024.

Tela de visualizar usuário (figura 62): Nesta tela o usuário poderá visualizar todos os usuários cadastrados (*ID*, NOME, *LOGIN*), ou poderá pesquisar um usuário a partir do login. Além disso, o usuário é capaz de navegar entre os dados, adicionar mais um usuário, editar ou deletar um usuário selecionado.

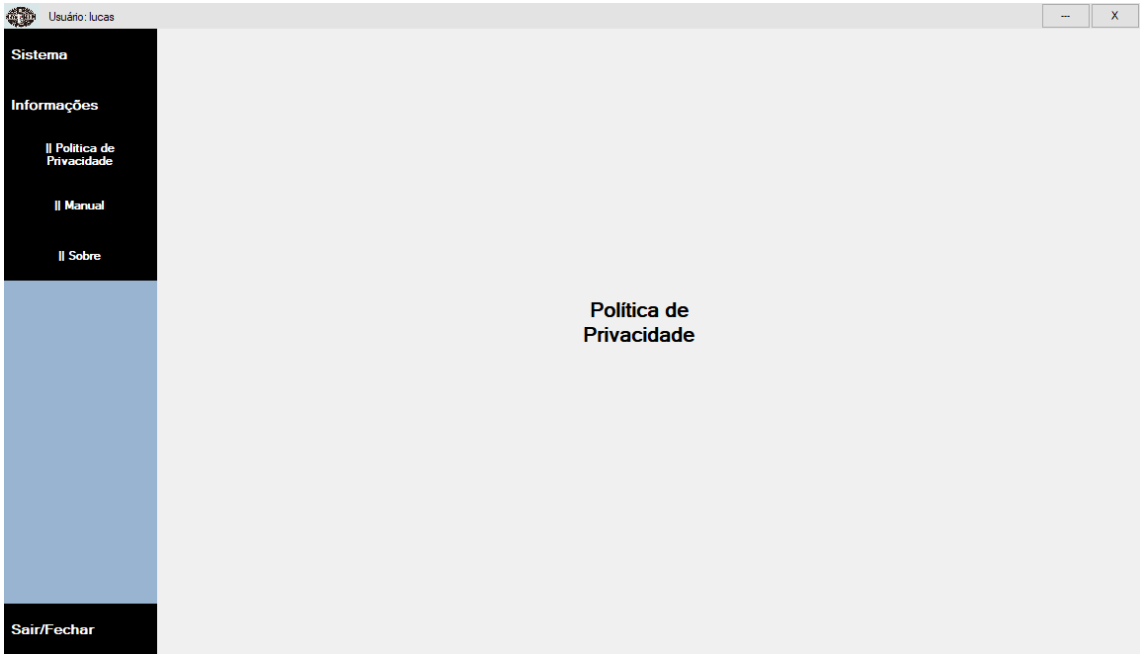
Figura 62 - Tela de visualizar usuário Desktop



Fonte: Autoria própria, 2024.

Tela de política de privacidade (figura 63): Ao clicar no botão que abre esta tela o navegador será aberta, podendo assim, ser visualizado o arquivo, referente à política de privacidade.

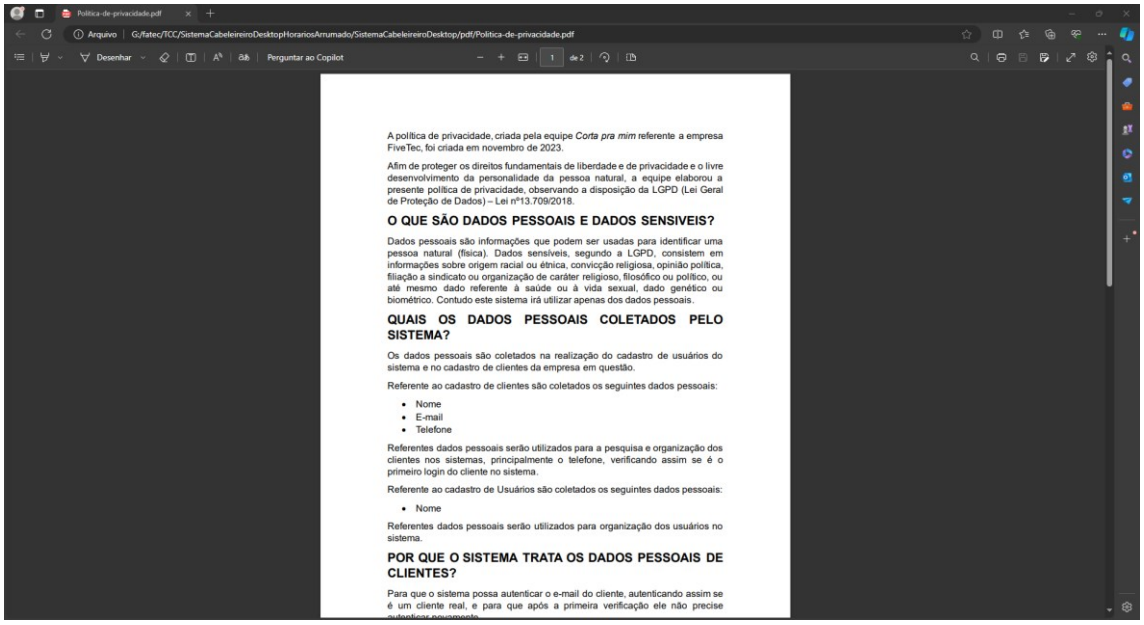
Figura 63 - Tela de política de privacidade Desktop



Fonte: Autoria própria, 2024.

A figura 64 mostra a política de privacidade no navegador, nela é possível determinar de que maneira a empresa trata os dados do cliente. Visa esclarecer para o barbeiro como suas informações estão sendo utilizadas.

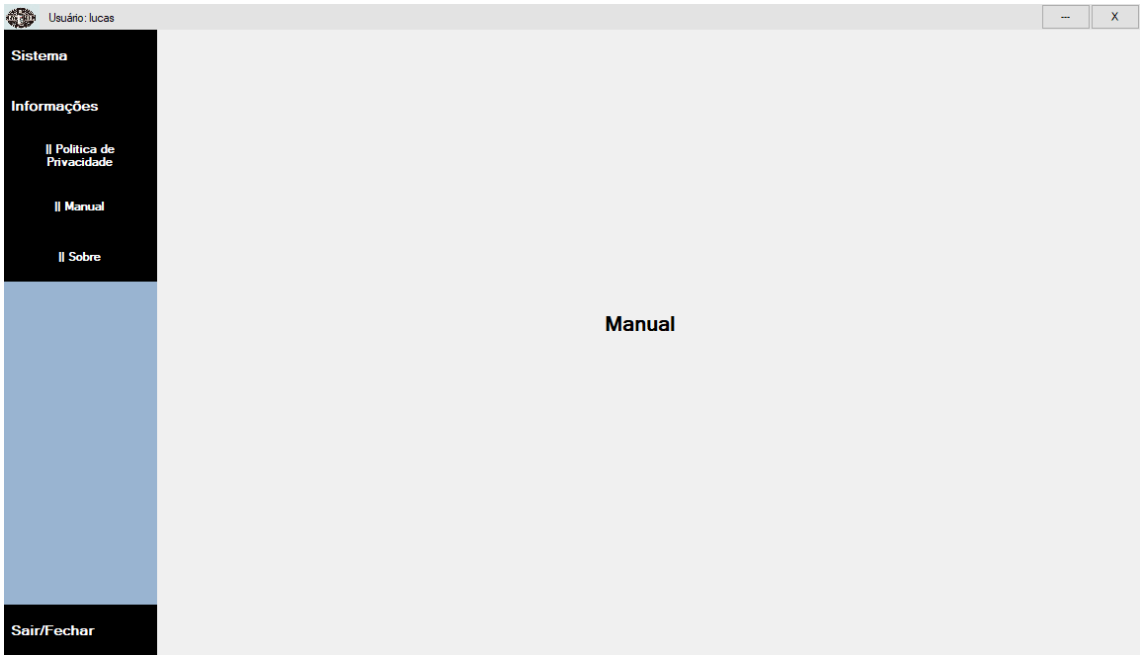
Figura 64 - Arquivo PDF política de privacidade



Fonte: Autoria própria, 2024.

Tela do manual (figura 65): Ao clicar no botão que abre esta tela o navegador será aberta, podendo assim, visualizar o arquivo referente ao manual do sistema.

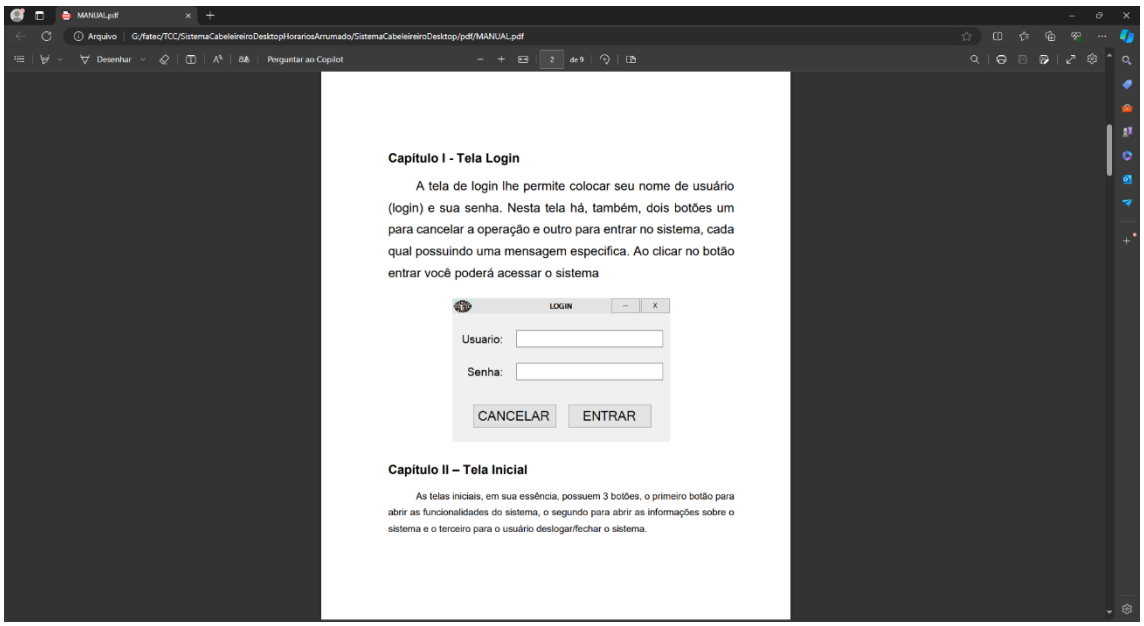
Figura 65 - Tela do manual Desktop



Fonte: Autoria própria, 2024.

A figura 66 mostra o arquivo aberto no navegador, neste arquivo será possível compreender as funcionalidades do sistema. Extremamente útil na fase de adaptação do usuário com a aplicação, pois neste documento é explicado passo a passo como utilizar o programa.

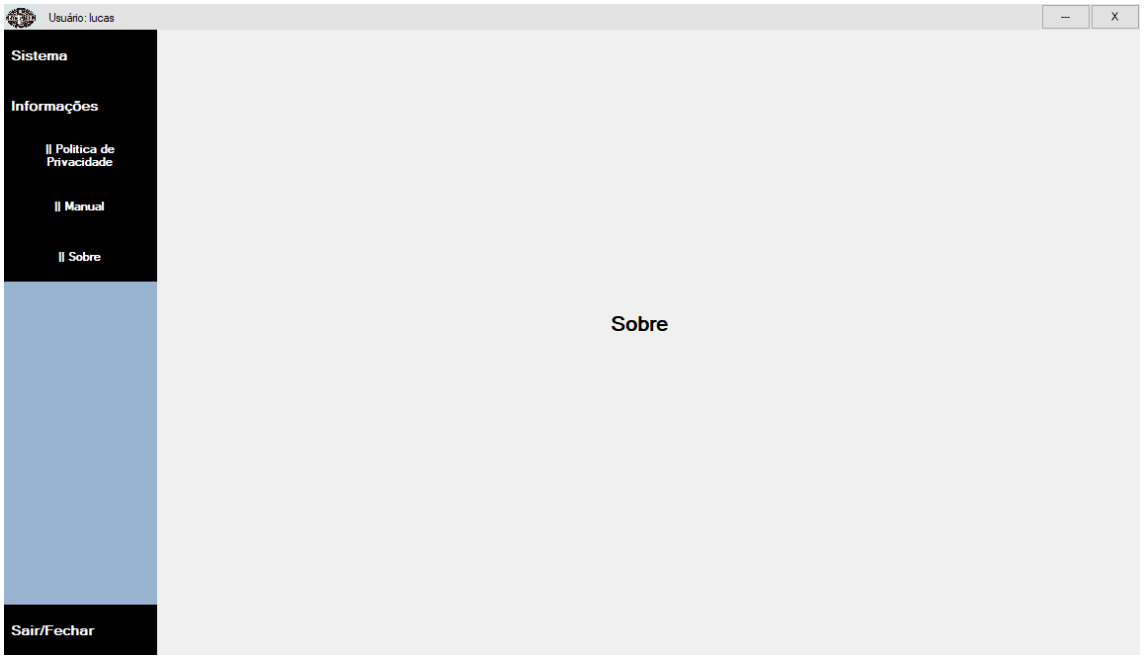
Figura 66 - Arquivo PDF do manual Desktop



Fonte: Autoria própria, 2024.

Tela sobre (figura 67): Ao clicar no botão que abre esta tela o navegador será aberto e um arquivo em PDF poderá ser visualizado.

Figura 67 - Tela sobre o sistema Desktop

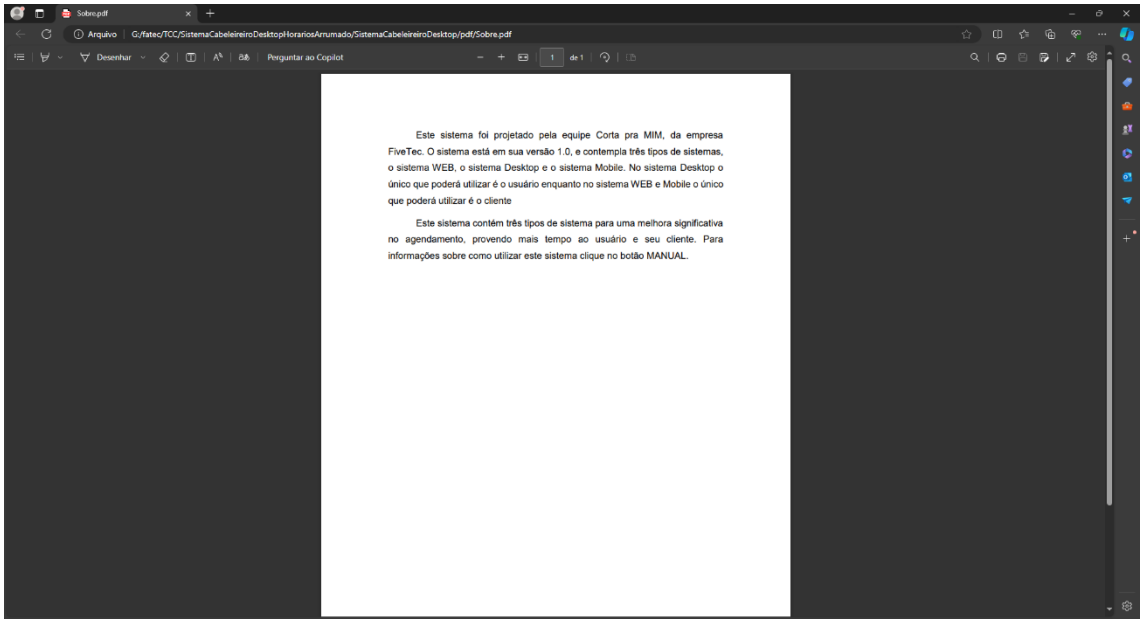


Fonte: Autoria própria, 2024.

Na figura 68 é possível visualizar o arquivo aberto no navegador do usuário, neste arquivo estão algumas informações adicionais, como por exemplo

a versão que ele está utilizando, que aplicações compõe todo o conjunto do sistema e etc.

Figura 68 - Arquivo PDF sobre o sistema Desktop



Fonte: Autoria própria, 2024.

5 Conclusão

Com base no problema identificado no capítulo 1 foi notado uma falta de organização no tempo para agendamentos, onde frequentemente o barbeiro se encontrava incapaz de responder mensagens ou atender seu telefone devido à sua linha de trabalho, onde é necessário foco para a execução das tarefas. Assim, para abordar essa questão de otimização e gestão de tempo, o grupo decidiu iniciar o desenvolvimento de sistemas que pudessem melhorar isso.

O sistema possibilitou poupar mais o tempo tanto dos barbeiros quanto de seus clientes, afinal ao permitir que os clientes façam seus próprios agendamentos, eles não precisam esperar o barbeiro acabar um corte para os responder via mensagem, do mesmo jeito que o barbeiro não precisa parar seu serviço para efetuar um agendamento. Portanto, o sistema aprimorou a dinâmica de agendamento de uma forma substancial.

Para futuras melhorias do projeto, algumas sugestões foram identificadas: a implementação de uma funcionalidade “Esqueci a senha”, a inclusão de um método para escolher o barbeiro, usuário, na hora do agendamento, a introdução de serviços de SMS e Notificação para lembrar o cliente de seus cortes, alteração de fonte para deixar o sistema com uma interface mais personalizada e por fim adicionar mais animações para tornar a experiência de navegação mais dinâmica e moderna.

Em um mundo onde a tecnologia está em constante evolução, é essencial que o sistema proposto neste trabalho acompanhe esse ritmo acelerado. Cada nova tendência que surge deve ser cuidadosamente pesquisada e avaliada para sua possível implementação em futuras versões do nosso sistema. Isso permitirá que o sistema otimize ainda mais a eficiência dos agendamentos, economizando tempo valioso de futuros clientes.

REFERÊNCIAS

ASP.NET core. ASP.NET Core Blazor: Disponível em: <https://learn.microsoft.com/pt-br/aspnet/core/blazor/?view=aspnetcore-7.0&WT.mc_id=dotnet-35129-website>. Acesso em: 14 fev. 2024.

Azure. Possibilidade ilimitadas, da borda para nuvem: Disponível em: <https://azure.microsoft.com/pt-br/>. Acesso em: 22 fev. 2024.

Blazor. Build beautiful web apps with Blazor: Disponível em: <https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor>. Acesso em: 22 maio 2024.

Codewithsadee. barber [repositório online]. 2022. Disponível em <https://github.com/codewithsadee/barber>. Acesso em: 26 jan. 2024.

CSS. CSS: Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em: 13 mar. 2024.

C#. Disponível em: <https://learn.microsoft.com/ja-jp/dotnet/csharp/tour-of-csharp/>. Acesso em: 27 maio 2023.

DEVMEDIA. **MER e DER: Modelagem de Bancos de Dados:** Disponível em: <https://www.devmedia.com.br/mer-e-der-modelagem-de-bancos-de-dados/14332>. Acesso em: 03 jun. 2024.

Expo. Disponível em: <https://docs.expo.dev/get-started/expo-go/>. Acesso em: 22 jan. 2024.

Google MAPS: Disponível em: <https://developers.google.com/maps/documentation/javascript/adding-a-google-map?hl=pt-br>. Acesso em: 13 mar. 2024.

HTML. HTML Linguagem de Formatação de Hipertexto: Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 13 mar. 2024.

JavaScript. Disponível em: <https://developer.mozilla.org/pt-BR/docs/web/javascript/guide/introduction>. Acesso em: 13 mar. 2024.

Opensource-coding. Responsive-Calendar-with-Events [repositório online]. 2023. Disponível em: <https://github.com/opensource-coding/Responsive-Calendar-with-Events>. Acesso em: 26 jan. 2024.

Postman. About Postman: Disponível em: <https://www.postman.com/company/about-postman/>. Acesso em: 23 fev. 2024.

PRESSMAN, R.S. **Engenharia de Software:** Uma abordagem Profissional. 7ª Edição, São Paulo, Editora: Makron Books, 2011.

React Native. Disponível em: <https://reactnative.dev/>. Acesso em: 27 maio 2024.

SCRUM. **Guia do SCRUM**: Um guia definitivo para o Scrum: As regras do jogo. 2013. 19p. Disponível em: <https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-Portuguese-BR.pdf>. Acesso em: 13 maio 2024.

SOMMERVILLE, Ian. Engenharia de Software. Tradução: Luiz Cláudio Queiroz. Revisão técnica: Fábio Levy Siqueira. 10ª ed. São Paulo: Pearson Education do Brasil, 2018.

SQLite. Disponível em: <https://www.sqlite.org/onefile.html>. Acesso em: 20 maio 2024

SQL Server. SQL Server 2022: Disponível em: <https://www.microsoft.com/pt-br/sql-server/sql-server-2022>. Acesso em: 14 fev. 2024.

Trello. Disponível em: <https://trello.com/home>. Acesso em: 18 jan. 2024.

VISUAL STUDIO. Disponível em: <https://visualstudio.microsoft.com/pt-br/>. Acesso em: 22 jan. 2024.

Visual Studio Code. Disponível em: <https://code.visualstudio.com/docs>. Acesso em: 22 jan. 2024.