

# ***Agenda Simetria - Controle de Agenda de Espaço Estético***

**Rafael Rovani Tognon, Gabriel Paniz Patzer (Orientador)**

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul - Câmpus Sertão - Rodovia RS 135, Km 32,5 - Distrito Eng. Luiz Englert - CEP: 99170-000 - Sertão-RS

rafaelrovanitognon@gmail.com, gabriel.patzer@sertao.ifrs.edu.br

**Abstract.** *This article presents the development of a Progressive Web App software for an aesthetic space that manages the scheduling of appointments. This software was made as a final work for the Systems Analysis and Development Course, at the Federal Institute of Education, Science and Technology of Rio Grande do Sul - Câmpus Sertão. It also aims to help Espaço Estético Simetria to publicize its social networks, facilitate the management of internal processes and offer an option to pay for bookings through Pix.*

**Resumo.** *Este artigo apresenta o desenvolvimento de um software Progressive Web App para um espaço estético que gerencia o agendamento de horários. O referido software foi elaborado como trabalho de conclusão do Curso de Análise e Desenvolvimento de Sistemas, do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul - Câmpus Sertão. Também almeja-se auxiliar o Espaço Estético Simetria a divulgar suas redes sociais, facilitar o gerenciamento dos processos internos e oferecer opção de pagamento dos agendamentos através do Pix.*

## **1. Introdução**

Este artigo relata o desenvolvimento de uma aplicação para o espaço estético Simetria, denominada Agenda Simetria. Percebe-se que o gasto com salões de beleza faz parte do perfil de Orçamentos Familiares do Brasil. Além disso, destaca-se que o cenário durante e pós-pandemia da COVID-19, apresentou uma queda na procura por atendimentos no contexto de salões de beleza, e atualmente, busca-se retomar o crescimento nesse segmento. Assim, alguns estabelecimentos realizaram ações, baseadas na utilização da *internet*, para recuperar o desenvolvimento do setor.

Nesta circunstância, surge a ideia da criação de um aplicativo para o Espaço Estético Simetria, que também não teve um cenário favorável durante e pós-pandemia. O referido estabelecimento está localizado na cidade de Erebangó, dispõe de 2 profissionais e abrange os serviços de mechas, limpeza de pele, corte de cabelo, manicure, pedicure, entre outros. No momento, o estabelecimento utiliza apenas as redes sociais para auxiliar na divulgação e agendamentos, e não dispõe de um recurso informatizado para gerir os agendamentos e processos internos.

Desta forma, é relatado o desenvolvimento de um *software Progressive Web App* (PWA) para o Espaço Estético Simetria que tem por objetivo principal gerenciar o agendamento de horários. O *software* foi elaborado como trabalho de conclusão do curso (TCC) de Análise e Desenvolvimento de Sistemas, do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul - Câmpus Sertão. Além disso, pretende-se oferecer a opção de pagamento dos agendamentos realizados.

A fim de apresentar o processo de desenvolvimento, na seção 2, é destacado como a abordagem PWA atende os acessos através da *web* e *mobile*, bem como a justificativa que levou à construção deste *software*. A seção 3 contém o objetivo geral e os específicos que pretende-se atingir. Logo após, a seção 4 aborda os trabalhos encontrados que são relacionados a este e a seção 5 é composta pela metodologia adotada na concepção da aplicação (requisitos, diagrama de casos de uso, protótipo e tecnologias utilizadas). No final, a seção 6 apresenta os resultados obtidos com o desenvolvimento do trabalho e a seção 7 apresenta as considerações referentes à experiência obtida na elaboração do projeto, assim como trabalhos futuros para o aperfeiçoamento do produto obtido neste TCC.

## **2. Definição do problema e justificativa**

De acordo com o Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (SEBRAE), “salão de beleza abrange desde serviços especializados em estética capilar, maquiagem, unhas, *design* de sobrancelhas, podologia, atendimento especializado para noivas e noivos, podendo ainda abranger serviços de estética facial e corporal” [Sebrae 2022].

Com o objetivo de analisar o impacto e reação do segmento de Salões de Beleza no contexto da crise do Coronavírus, em 2021, foi realizada uma pesquisa pela Federação do Comércio de Bens, Serviços e Turismo do Estado do Rio Grande do Sul (Fecomércio-RS). Dentre os resultados obtidos, observou-se que 36,4% dos estabelecimentos entrevistados realizam agendamentos de horários através de aplicativos próprios. Outro dado observado foi que 97,7% utilizam de redes sociais para promover o seu negócio e se relacionar com seus clientes [Fecomércio-RS 2021].

Além disso, nas pesquisas sobre Orçamentos Familiares realizadas pelo Instituto Brasileiro de Geografia e Estatística (IBGE), as categorias de “Higiene e cuidados pessoais” e “Serviços pessoais” estão entre as 10 observadas para representar o perfil de gastos do brasileiro [Ibge 2018].

Para melhor atender as necessidades desse segmento, surge a possibilidade do desenvolvimento de um *software* aplicativo, com intuito de controlar a agenda dos profissionais, agendamentos e pagamentos de serviços. Essas funções tornam-se apropriadas, uma vez que otimizarão os processos rotineiros e ampliarão as opções do cliente. A necessidade de atender plataformas *desktop* e *mobile* fundamenta o desenvolvimento de um PWA.

Essa proposta foi desenvolvida com base no Espaço Estético Simetria, fundado em 19 de outubro de 2021 e localizado na cidade de Erebango. O estabelecimento conta com duas profissionais e disponibiliza os serviços de progressiva, mechas, tratamento capilar, *design* de sobrancelha, limpeza de pele, manicure e pedicure.

Atualmente, o Espaço Simetria realiza os agendamentos através de contatos pessoais e mensagens por redes sociais, o que dificulta e torna o processo custoso. Também não existe um canal onde possam ser feitas divulgações, promoções e facilitação de pagamento.

Optou-se por desenvolver um PWA, que “são aplicativos da *web* que usam *service workers*, manifestos e outros recursos de plataforma da *web* em combinação com aprimoramento progressivo para oferecer aos usuários uma experiência semelhante aos aplicativos nativos” [Developer Mozilla 2022]. Além disso, “os PWAs oferecem uma série de vantagens aos usuários, incluindo serem instaláveis, progressivamente aprimorados, projetados de forma responsiva, reativados, vinculáveis, detectáveis, independentes de rede e seguros” [Developer Mozilla 2022].

Outro conceito de PWA é apresentado pela *Microsoft*, como sendo “*sites* que são progressivamente aprimorados para funcionar como aplicativos nativos instalados em plataformas de suporte, enquanto funcionam como *sites* regulares em outros navegadores” [Microsoft 2022].

A *Google* começou a substituir aplicativos do *Android* presentes no *Chrome OS* por versões em PWA [TecMundo 2020]. Além disso, outras empresas já estão utilizando essa tecnologia em suas aplicações, como por exemplo: *Instagram*, *Uber*, *Twitter* e *Spotify*.

Na Figura 1, o aplicativo *Spotify* está sendo acessado na versão *site*. Na área destacada em azul existe a opção de fazer o *download* do aplicativo. Após a realização do *download* e instalação, o Aplicativo, no contexto *desktop*, se comporta como um aplicativo baixado da própria *Microsoft Store*.

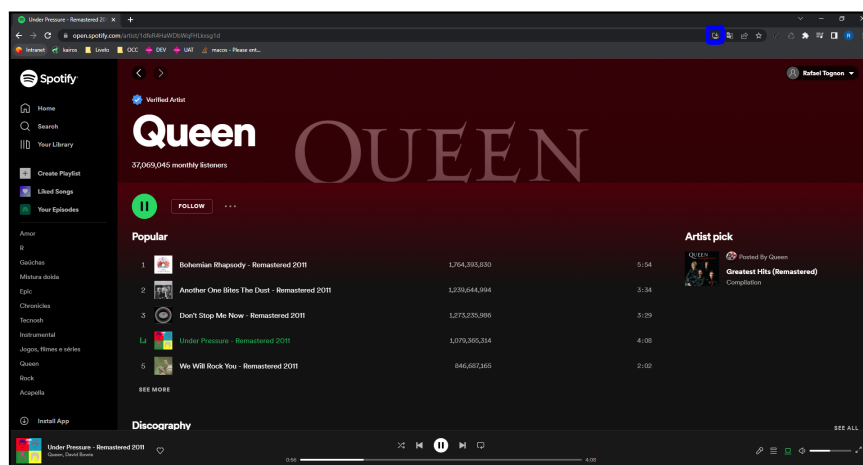


Figura 1. PWA do Spotify acessada pelo browser

Pode-se observar que nas Figura 2 e na Figura 3, o comportamento do PWA é semelhante aos *softwares* usuais.

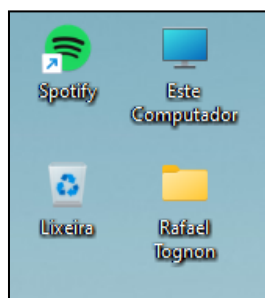
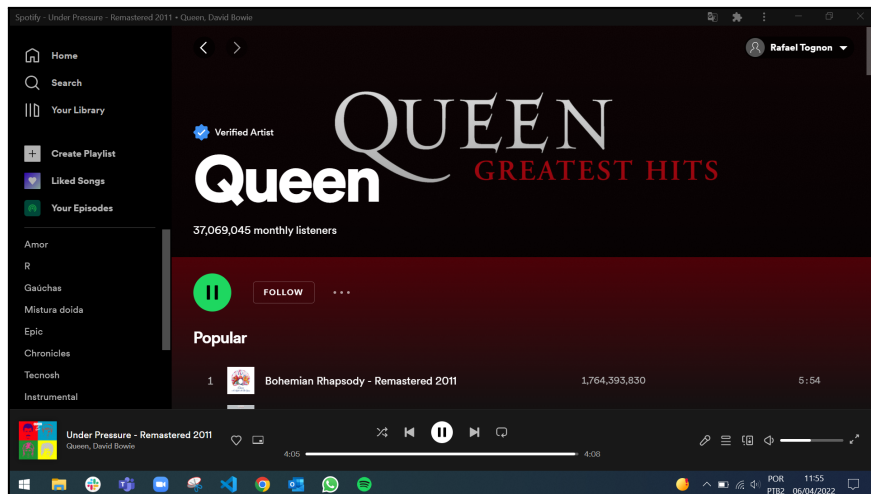


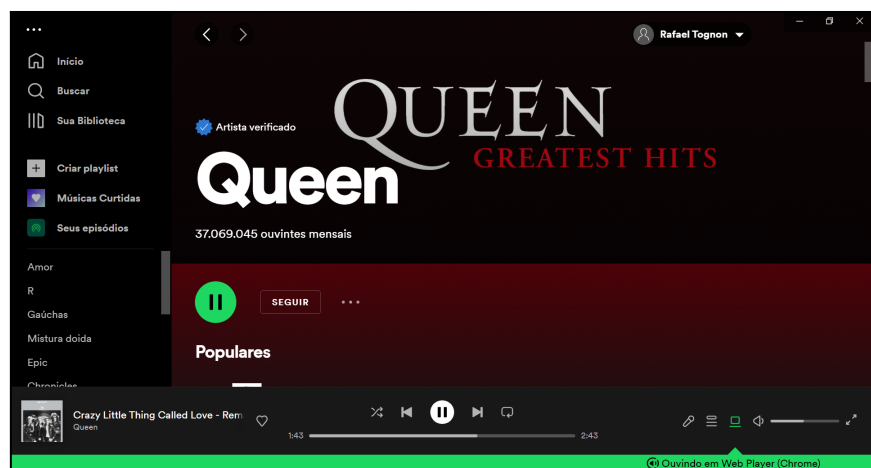
Figura 2. PWA do Spotify depois de instalada no Windows

Logo após o *download*, foi adicionado um atalho do *Spotify* na área de trabalho do *Windows*, da mesma forma que acontece em *softwares* nativos. Na Figura 3, é apresentado o PWA sendo executado no *Windows*.



**Figura 3. PWA do Spotify em execução no Windows**

Na Figura 4, o aplicativo para *Windows* do *Spotify* está sendo executado. Pode-se notar que a aparência, assim como o funcionamento, é muito semelhante nas versões *App* e PWA. É importante destacar que o PWA também dispõe dos recursos de fixar na barra de tarefas e é listado nos aplicativos e recursos instalados no *Windows*.



**Figura 4. App Spotify em execução no Windows**

### 3. Objetivos

Este trabalho tem como objetivo geral desenvolver um aplicativo PWA para um espaço estético que permite agendar horários, viabilizar um espaço na aplicação com *link* para as redes sociais da empresa e gerenciar os horários dos serviços prestados. Além disso, busca-se atingir os seguintes objetivos específicos:

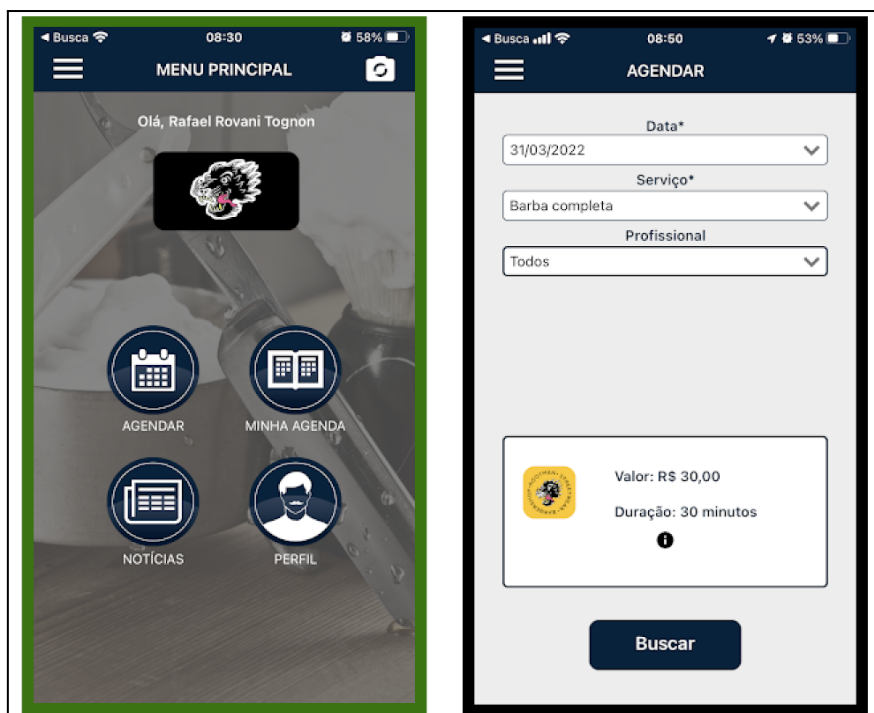
- Disponibilizar o aplicativo para *Android*, *iOS* e *Web* através do PWA.
- Facilitar o controle das atividades e agendamentos por parte da proprietária.

- Oferecer opção de pagamento dos agendamentos pela aplicação (Pix).

#### 4. Trabalhos relacionados

Um trabalho relacionado a este é o aplicativo *AppBarber*. O *AppBarber* é um aplicativo disponível para *iOS* e *Android* focado em controlar agendas de barbearias. Ele é pago para utilização e o custo mensal varia de R\$ 36,00 a R\$ 199,90, dependendo do tipo de plano (mensal, semestral ou anual) e da quantidade de funcionários que o estabelecimento possui. O *software* é dividido em dois módulos, sendo o primeiro um módulo *WebAdmin*, no qual é possível fazer a gestão do estabelecimento, como o controle de cada profissional, histórico de clientes, estoque e relatórios financeiros. O segundo módulo pode ser acessado tanto pelo profissional, para acompanhar sua agenda, quanto pelo cliente, para realizar agendamentos, receber notícias e promoções e lembretes automáticos. O uso é gratuito para o cliente do estabelecimento e se encontra disponível nas lojas *App Store* e *Play Store*.

A Figura 5 com destaque em borda verde apresenta o menu principal do módulo acessível ao cliente. Através dele, o usuário pode agendar horários, consultar a agenda, visualizar notícias do estabelecimento e dados do perfil. A Figura 5 com destaque em borda preta está apresentando a tela de agendamento do *AppBarber*. Nela, é possível escolher a data, tipo de serviço e profissional desejado.



**Figura 5. Tela inicial do *AppBarber* e tela de agendamentos do *AppBarber***

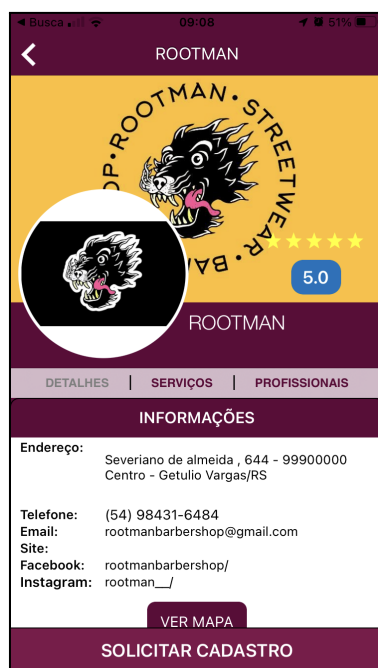
Através do *AppBarber*, o proprietário pode cadastrar o estabelecimento informando os serviços disponíveis, contato e profissionais. Após realizar uma solicitação de cadastro, o cliente pode fazer agendamentos no estabelecimento, desde que a solicitação seja aprovada pelo responsável. Também é possível cadastrar programas de fidelidade, de acordo com a escolha do proprietário. Porém, ele não

possibilita realizar o pagamento dos agendamentos, é focado no segmento de barbearia e é necessário pagar um valor mensal para acessar o módulo de gerenciamento.

Outro trabalho relacionado seria o *AppBeleza*. É um sistema para gestão *online* de salões de beleza, clínicas de estética ou similares que permite agendamentos de horários de diversos serviços previamente cadastrados. Ele conta com 2 módulos com características semelhantes aos do *AppBarber* e tem o custo mensal de R\$ 36,00 a R\$ 199,90 ao proprietário, de acordo com o tipo de plano escolhido (mensal, semestral ou anual) e da quantidade de funcionários.

Na Figura 6, é exibida a tela de detalhes do estabelecimento no *AppBeleza*. Nela é possível visualizar a localização, redes sociais, serviços oferecidos e profissionais. Também é possível solicitar o cadastro no estabelecimento, caso o usuário ainda não tenha feito.

O *AppBeleza* é focado no segmento de salões de beleza e fornece ao proprietário a possibilidade de cadastrar o estabelecimento informando os serviços disponíveis, contato e profissionais. Após realizar uma solicitação de cadastro, o cliente pode fazer agendamentos, desde que a solicitação seja aprovada pelo responsável. Também é possível cadastrar programas de fidelidade, de acordo com a escolha do proprietário. Entretanto, no decorrer dos estudos não foi identificada a possibilidade de realizar o pagamento dos agendamentos pelo aplicativo e o proprietário precisa pagar um valor mensal para acessar o módulo de gerenciamento.



**Figura 6. Tela de informações do estabelecimento do *AppBeleza***

Com base nos estudos realizados, um *software* PWA foi desenvolvido para atender às necessidades do Espaço Estético Simetria. Como pontos diferenciais, a aplicação disponibiliza a opção de realizar o pagamento do agendamento e é gratuita aos usuários e de baixo custo à proprietária.



## 5. Projeto do Software

Nesta seção serão descritos de forma mais detalhada a metodologia abordada, requisitos funcionais e não funcionais, casos de uso e tecnologias utilizadas na execução do desenvolvimento do sistema Agenda Simetria.

### 5.1 Descrição do Software

O *software* Agenda Simetria é dividido em duas partes com objetivos distintos. A primeira delas é um módulo de gerência disponível na *web*, que tem a proprietária cadastrada como usuário administrador. Quando acessado, é solicitado, através de uma tela de *login*, as credenciais de acesso da proprietária. Depois da autenticação realizada, uma interface possibilita realizar as funcionalidades de CRUD (*Create, Read, Update e Delete*) dos agendamentos.

A segunda parte se trata de um módulo do cliente, acessível por qualquer usuário na *web*. Quando acessado, é apresentado o endereço, horário de funcionamento e serviços oferecidos no estabelecimento junto de uma tela de *login*, que solicita ao usuário se autenticar na aplicação. Caso o cliente ainda não tenha acessado a aplicação, é possível criar uma conta informando *login* e senha ou entrando com uma conta *Google*. Após ter realizado o *login*, o cliente pode visualizar os serviços oferecidos, horário de atendimento, profissionais disponíveis e acessar as redes sociais do estabelecimento através dos seus respectivos *links*.

Além disso, pode agendar horários para os serviços disponíveis. O agendamento se dá através de uma interface que apresenta ao cliente as opções de serviços por profissional disponíveis e possibilita escolher entre essas opções. Após ter escolhido, é apresentado um botão de busca, que quando acionado verifica e apresenta os horários disponíveis na agenda dos profissionais que oferecem o serviço escolhido. O cliente pode selecionar uma opção de horário apresentada, e para confirmar o agendamento, é apresentada uma mensagem de confirmação, que se for selecionada a opção “Confirmar”, agenda o horário escolhido. Além disso, esse módulo apresenta as notícias cadastradas pela proprietária.

Depois que o agendamento é realizado, a aplicação possibilita ao usuário pagá-lo através do procedimento “Pix copia e cola” ou escaneando o *QR Code* apresentado na tela. Esta possibilidade gera uma cobrança com uma chave identificadora do pagamento que, quando informada no aplicativo da instituição financeira, cria um pagamento com os dados do estabelecimento já definidos (valor e conta bancária), sendo necessário somente confirmá-los para efetuar o pagamento.

### 5.2 Definição de Requisitos

A definição dos requisitos é uma atividade fundamental no desenvolvimento de um *software*. Os requisitos para um sistema de *software* “estabelecem o que o sistema deve fazer e definem as restrições sobre seu funcionamento e implementação” [Sommerville 2011].

Nesta seção serão abordados os requisitos funcionais e não funcionais que pretende-se alcançar na produção desta aplicação.

### 5.2.1 Requisitos funcionais

De acordo com Sommerville (2011) “requisitos funcionais são declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer”. A seguir apresenta-se os requisitos funcionais analisados para a aplicação Agenda Simetria:

- Manter agendamentos: é possível agendar e cancelar horários dos serviços oferecidos pelo estabelecimento através do módulo do cliente. Também é possível editar a agenda através do módulo de administrador, a fim de adicionar horários que foram agendados por outros meios (tais como *WhatsApp*, *Instagram*, *Messenger*, contato pessoal) para sempre manter a agenda do aplicativo atualizada.
- Manter pagamentos: caso o cliente optar, é possível realizar o pagamento do agendamento realizado através da aplicação. Um agendamento não deve ser bloqueado ou invalidado por não ter sido realizado o pagamento, o pagamento pela aplicação deve ser feito conforme opção do cliente.
- Gerar pagamentos: o pagamento será feito utilizando a estratégia de Pix Cópia e Cola ou *QR Code*. A aplicação deve gerar a chave para a utilização desse tipo de pagamento.
- Manter clientes: deve ser possível gerenciar os clientes existentes na aplicação.

### 5.2.2 Requisitos não funcionais

Para Sommerville (2011, p. 59), “requisitos não funcionais são restrições aos serviços ou funções oferecidos pelo sistema. Incluem restrições de *timing*, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo.” Desta forma, os tópicos abaixo apresentam os requisitos não funcionais do sistema da aplicação a ser desenvolvida:

- Realizar login: a aplicação deve solicitar login do usuário para realizar o agendamento.
- Para controle e versionamento de código fonte será utilizada a ferramenta *Git* hospedada na plataforma *GitHub*.
- Para o desenvolvimento do *front-end*, será utilizada a biblioteca *React*.
- Na construção do *back-end* será utilizada a tecnologia *Node.js*.
- O Sistema Gerenciador de Banco de Dados (SGBD) utilizado será o *MySQL*.
- Será utilizado o software *Figma* para criar o protótipo das telas da aplicação.
- O software *Visual Paradigm* será utilizado para construir os diagramas da *Unified Modeling Language* (UML) relacionados à aplicação.

A escolha da biblioteca *React* se justifica pelo fato de facilitar a criação de UIs interativas. Ele permite criar *views* simples para cada estado da aplicação, e o *React* atualiza e renderiza apenas os componentes necessários na medida em que os dados são alterados [React 2022].



A escolha das tecnologias *Node.js*, para a construção do *back-end*, e do *GitHub*, para versionar código, se justificam devido aos conhecimentos prévios obtidos através de experiências no desenvolvimento de trabalhos anteriores.

Será utilizado o software *Visual Paradigm* para a criação dos diagramas da UML por apresentar um conjunto de ferramentas livres para o gerenciamento de projetos [Visual Paradigm 2022].

O *Figma* foi escolhido por possibilitar projetar, prototipar, compartilhar, coletar *feedback* e transformar “o *design* estático em uma experiência interativa, sem necessidade de codificação” [Figma 2022].

Optou-se pelo SGBD *MySQL* por ser um banco de dados relacional utilizado por muitas das maiores organizações do mundo, incluindo *Facebook* e *Twitter*, para alimentar seus sites, que possuem alto volume de dados [MySQL 2022].

### 5.3 Diagrama de Casos de Uso

Neste item é apresentado e detalhado o diagrama de casos de uso. Os diagramas de casos de uso “são focados em interações e mais detalhados do que os requisitos de usuário. Pode-se pensar a seu respeito como uma ponte entre os requisitos e o projeto. Em um modelo de caso de uso, você descreve os passos de cada interação do usuário e, assim, começa a identificar e definir as classes do sistema” [Sommerville 2011]. O diagrama de casos de uso faz parte da UML, uma linguagem de modelagem de *software* que auxilia na definição das características do sistema conforme seus requisitos. Na Figura 7 é apresentado o diagrama de casos de uso da aplicação desenvolvida.

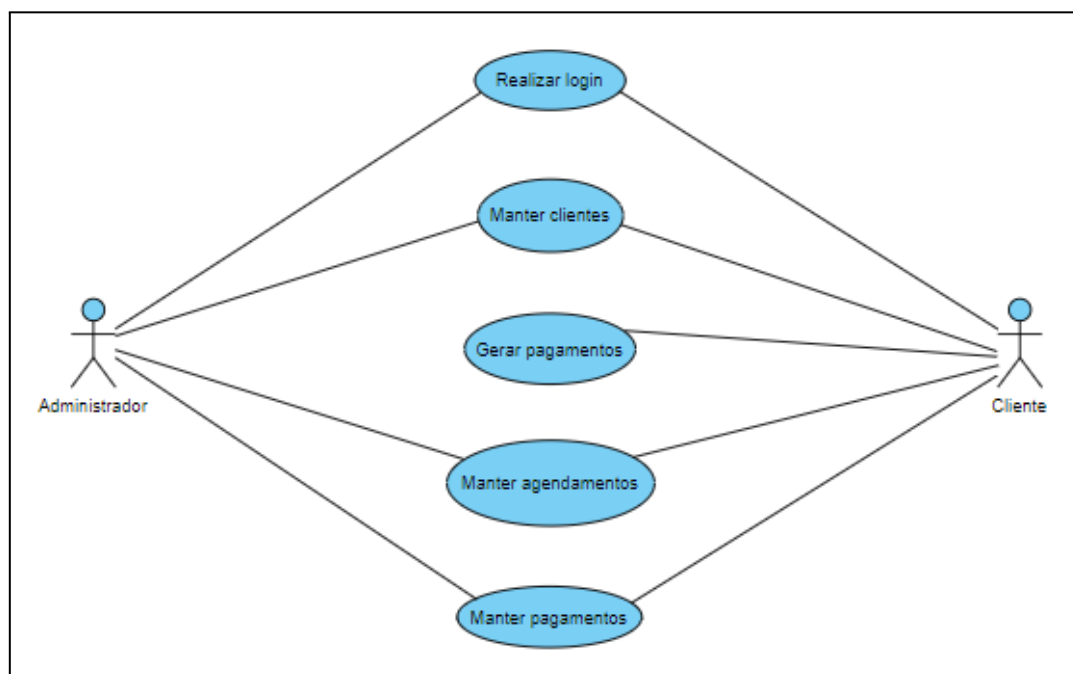
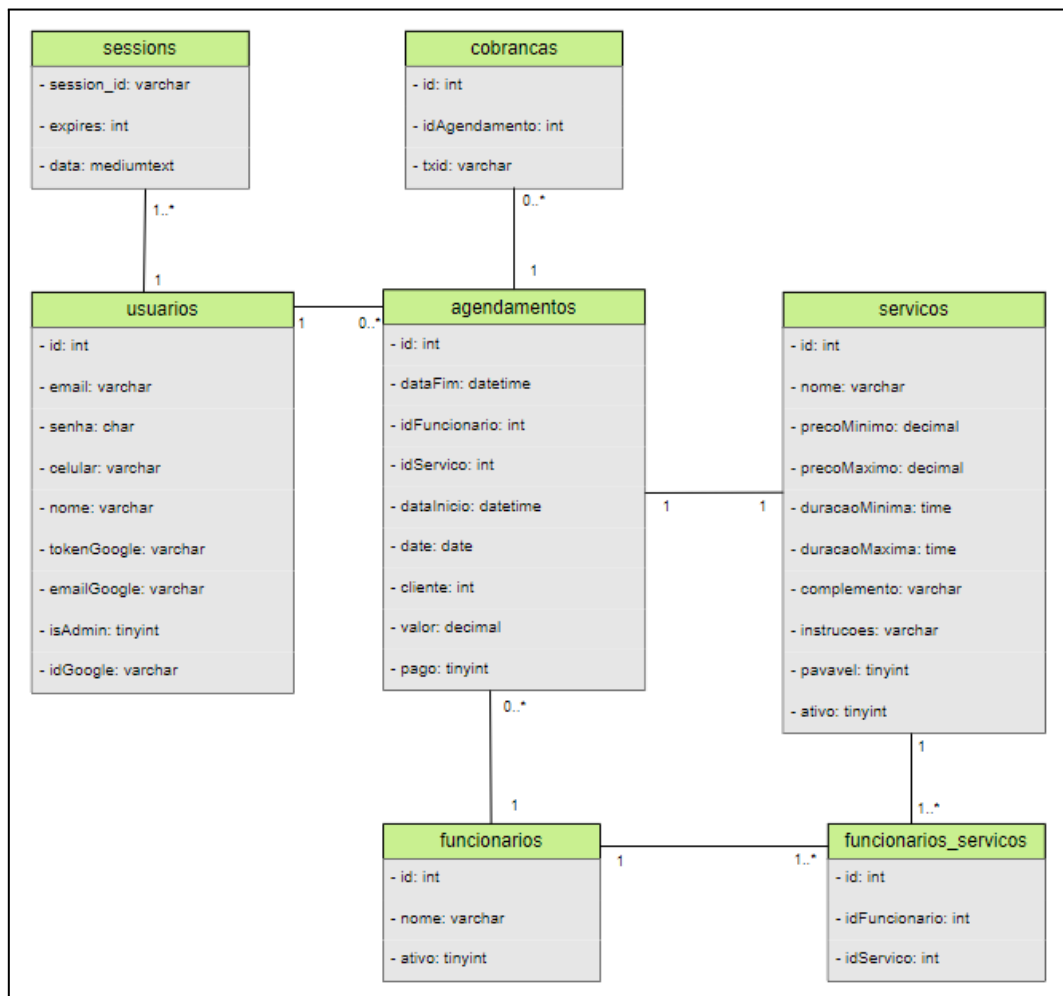


Figura 7. Apresentação do diagrama de casos de uso do Agenda Simetria

### 5.4 Modelo Entidade-Relacionamento

O modelo entidade-relacionamento (MER) é uma representação gráfica que tem por objetivo modelar de forma abstrata um banco de dados, permitindo a especificação das propriedades dos objetos a serem armazenados [Heuser 1998].

As entidades, que são um conjunto de objetos sobre os quais deseja-se guardar informações, são representadas por retângulos. Já os relacionamentos, representados por linhas, interligam as entidades [Heuser 1998]. O MER do sistema proposto é representado na Figura 8.



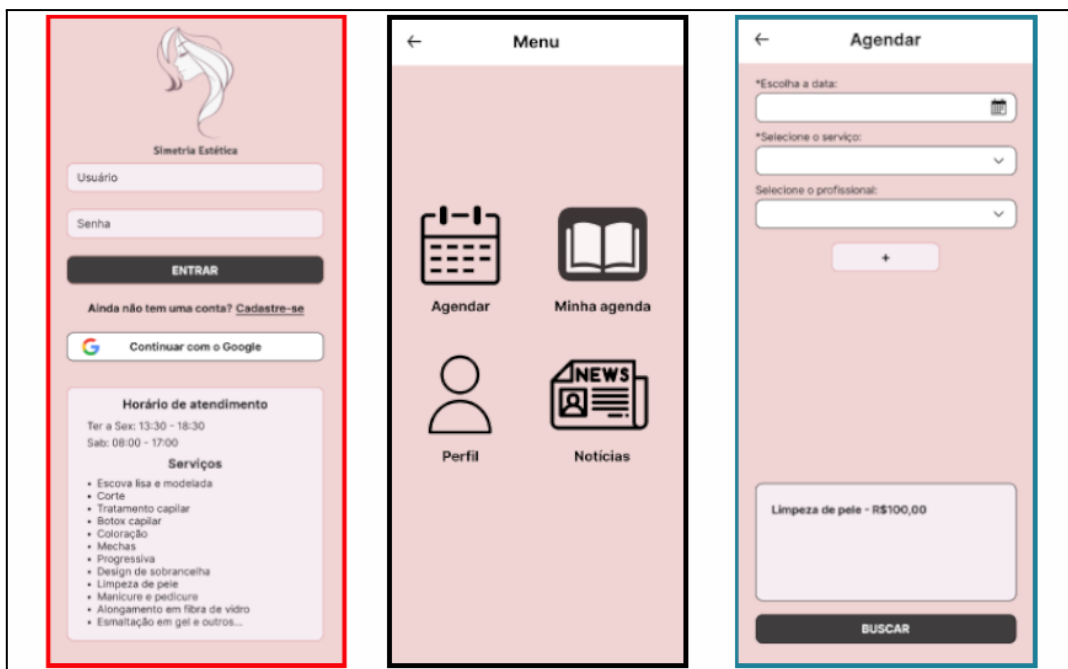
**Figura 8. Apresentação do modelo entidade-relacionamento do Agenda Simetria**

## 5.5 Interface do Sistema

O projeto de interface do sistema, ou seja, o protótipo da aplicação, é uma etapa importante no processo de desenvolvimento de *software*. Sommerville (2011) define projeto de interface de usuário como “o processo de projetar a maneira como os usuários podem acessar a funcionalidade do sistema e a maneira como as informações produzidas por ele são exibidas”. Ainda, Pressman (2011) reforça que “uma interface bem elaborada facilita a integração e auxilia o responsável pelos testes quanto à validação das funções dos componentes” e complementa que “não importa quão sofisticadas sejam as funções internas, quão amplas sejam as estruturas de dados, quão bem projetada seja a arquitetura; um projeto de interface pobre leva à percepção de que

um *software* é ruim”.

Desse modo, buscando exemplificar e tornar a utilização do *software* intuitiva, foi construído um projeto de interface do módulo do cliente da aplicação. Na Figura 9, com destaque em borda vermelha, é apresentada a tela de *login* da aplicação. A seguir, o destaque com borda preta ilustra o menu principal da aplicação. Por fim, a borda azul indica a continuidade do fluxo principal do *software*. Nela, a tela para escolher os serviços, data e profissionais é exemplificada.



**Figura 9. Telas de login, menu principal e agendamento do Agenda Simetria**

Em seguida, a Figura 10 de borda vermelha apresenta a tela de escolha do horário dos serviços buscados. Nela, o cliente define qual horário deseja para realizar o serviço buscado. Por fim, a Figura 10 de borda preta demonstra a tela de confirmação do agendamento. Nela, o valor, profissional, data, hora e serviço escolhidos são apresentados, juntamente com um botão para confirmar ou não o agendamento.

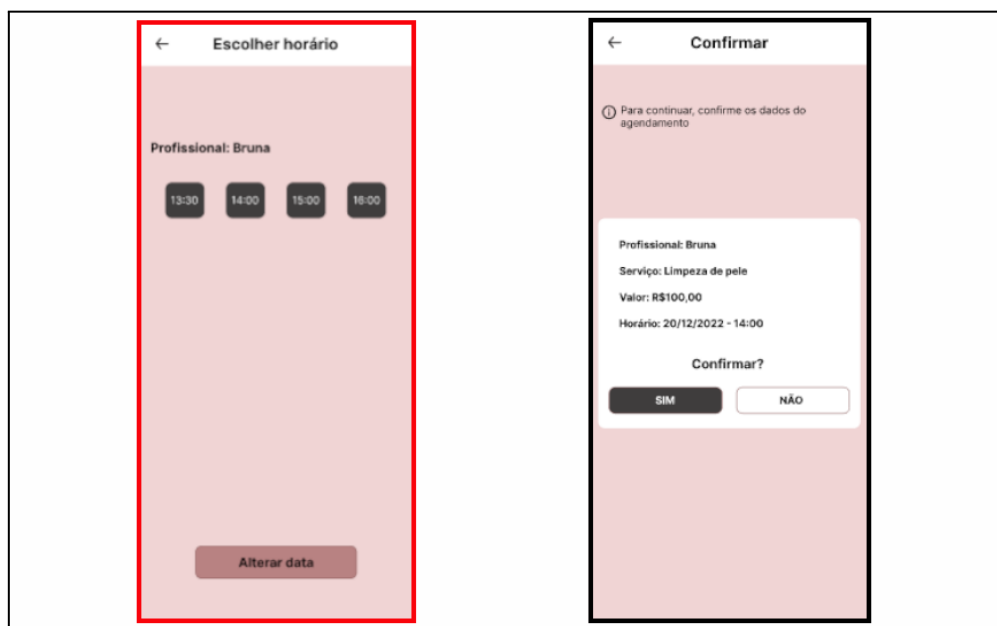


Figura 10. Tela de escolha do horário e tela de confirmação do agendamento

## 6. Resultados

Nesta seção, o produto final do software é apresentado, além de detalhes sobre a implementação, conhecimentos reforçados e adquiridos e custos da aplicação. Por fim, também foi preciso comprar o domínio [esteticasimetria.com.br](http://esteticasimetria.com.br) no Registro BR para que a aplicação tenha a *Uniform Resource Locator* (URL) personalizada.

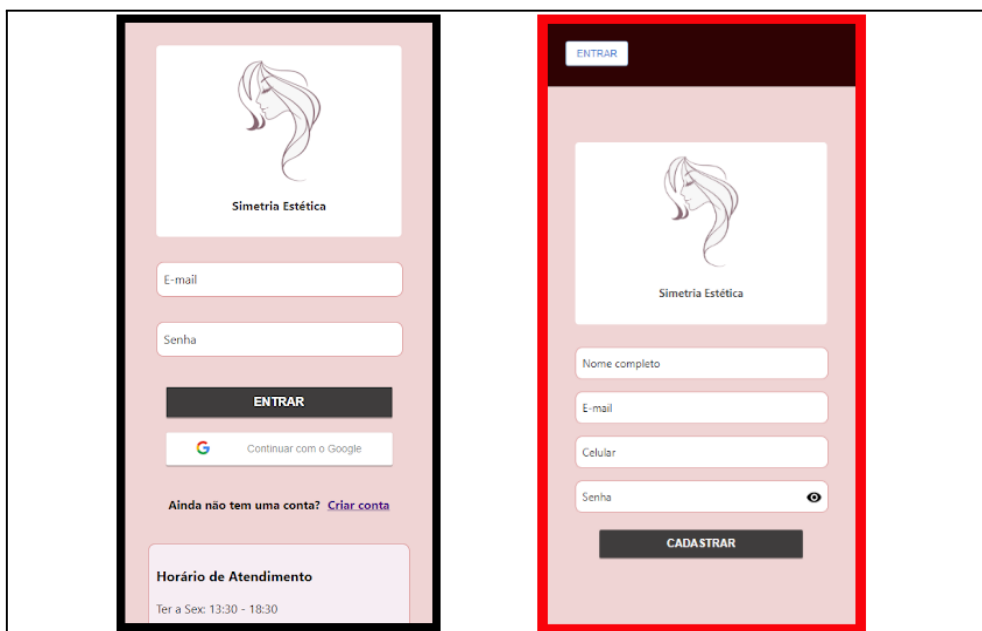
Dentre as funcionalidades implementadas, a que mais exigiu estudos e demorou foi o pagamento através do Pix. Isso porque para utilizar as APIs da Gerencianet, é exigido a autenticação através de certificados para tornar o processo o mais seguro possível.

Cada camada da aplicação foi hospedada em uma plataforma distinta, visando reduzir os custos. O *front-end* está hospedado utilizando o plano grátis da *Vercel*. Já o *back-end* está na AWS, que por 1 ano oferece o serviço gratuitamente e cobra somente pela utilização de disco, que no primeiro mês desta aplicação foi de \$ 0,50 (cinquenta centavos de dólar americano). Por fim, o banco de dados está na *Heroku*, que também está utilizando o plano grátis.

O maior custo até o momento foi relacionado ao domínio no Registro BR, que foi comprado por 2 anos pelo valor de R\$ 76,00. Com isso, é possível estimar um custo aproximado de R\$ 5,91 por mês para toda a hospedagem do sistema no primeiro ano.

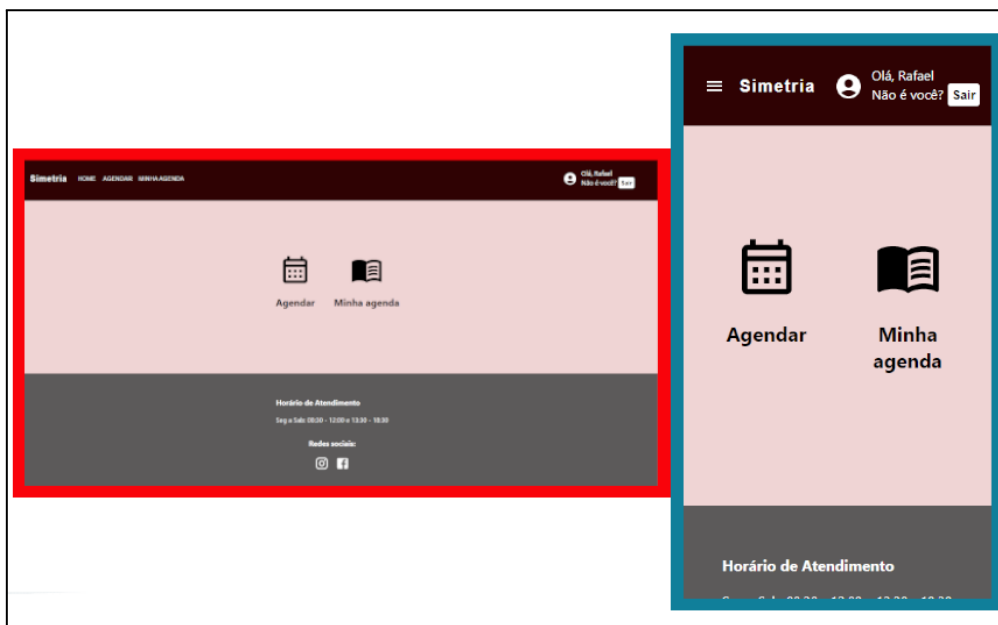
### 6.1 Módulo de usuário

Neste tópico, as telas que foram desenvolvidas no módulo de cliente serão apresentadas considerando, conforme necessário, a visão *desktop* e *mobile*, sendo que a visualização *mobile* será priorizada por acreditar-se que a maior parte do público utilizará a aplicação através de um *smartphone*. A Figura 11, com destaque em borda preta e vermelha, apresenta a tela de *login* e a tela de cadastro na versão *mobile*, respectivamente.



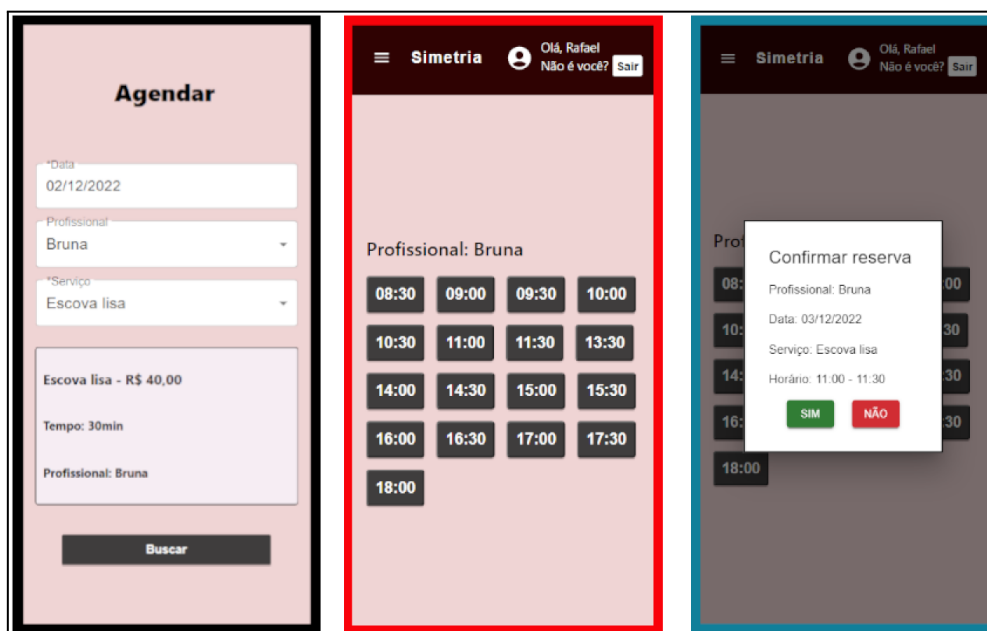
**Figura 11. Tela de *login* e tela de cadastro mobile**

Em seguida, a Figura 12 de borda vermelha apresenta o menu principal da aplicação na versão *desktop* e a Figura 12 de borda azul apresenta o menu principal na versão *mobile*. Na versão *mobile* é possível visualizar o cabeçalho do *site* sendo apresentado de uma forma reduzida.



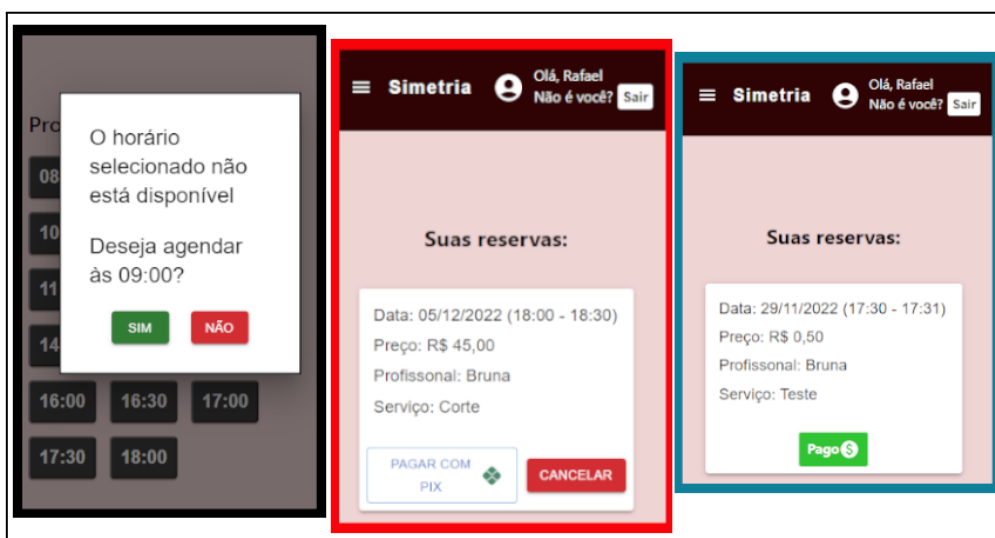
**Figura 12. Menu principal para *desktop* e *mobile***

A seguir, a Figura 13 com destaque em borda preta apresenta a tela em que o cliente escolhe o serviço, data e profissional desejado e faz a busca dos horários disponíveis. A tela com destaque em borda vermelha apresenta o resultado da busca de horários, no qual o cliente escolhe um de preferência. Por fim, o destaque em borda azul apresenta o modal para confirmação do agendamento.



**Figura 13. Tela de busca, tela de escolha de horário e modal de confirmação**

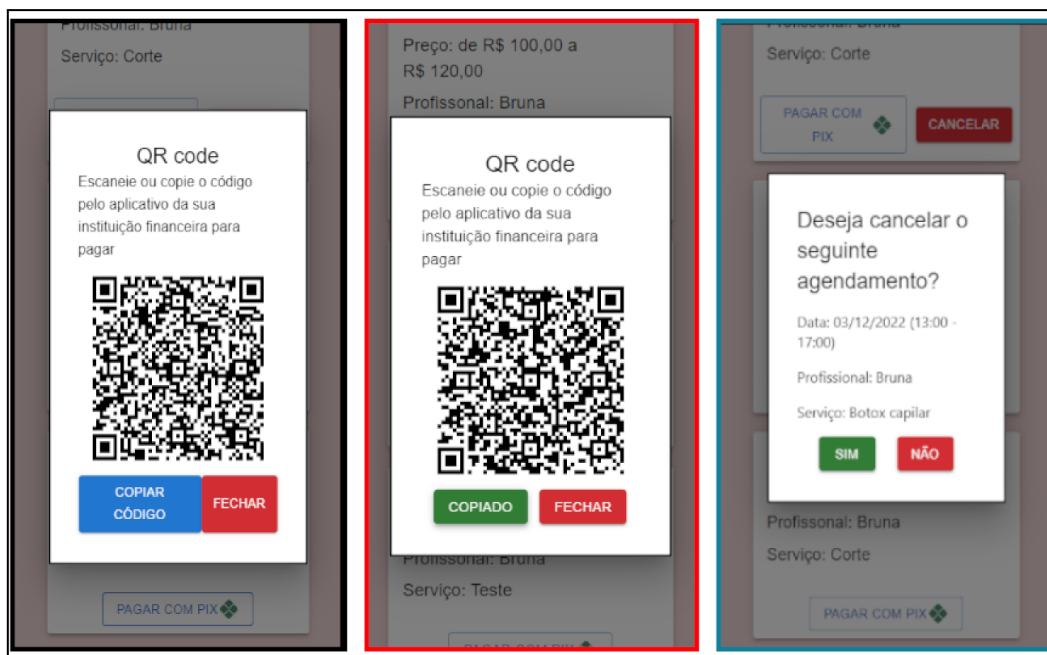
Logo após, a Figura 14 com borda preta apresenta o modal de tratamento de concorrência para a reserva de horário, que é apresentado quando a reserva do mesmo horário já foi feita pouco antes por outro usuário. Sendo assim, a aplicação oferece ao cliente a opção do primeiro horário disponível no dia, que pode aceitar ou recusar. Caso a sugestão seja recusada, a aplicação fechará o modal e fará uma nova busca de horários para o serviço. Por fim, a imagem com destaque em borda vermelha apresenta a tela de reservas do cliente e uma reserva ainda não paga e a imagem com destaque em borda azul apresenta o caso de uma reserva que já foi paga. É importante ressaltar que quando um agendamento tiver seu horário de término passado, a opção de “Cancelar” não estará mais disponível.



**Figura 14. Modal de tratamento de concorrência, tela de reservas com uma reserva não paga e tela de reservas com uma reserva paga**



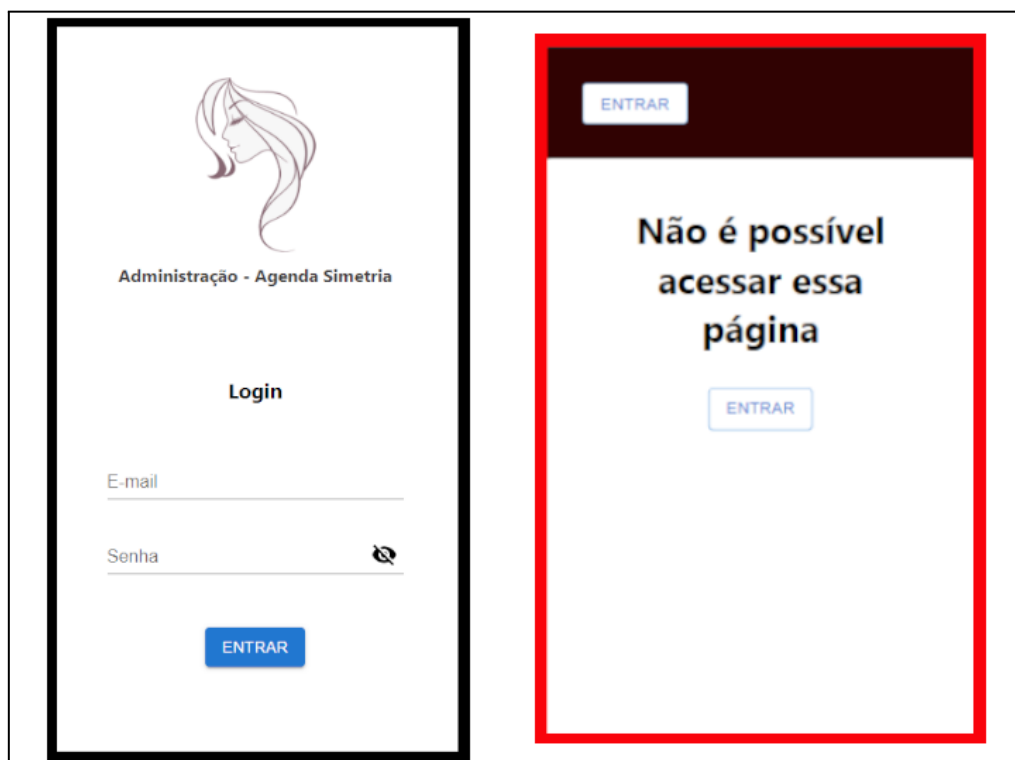
Posteriormente, a Figura 15 em borda preta exibe o modal de *QR Code* que é apresentado quando o botão “Pagar com Pix” é acionado. Após, a Figura 15 de borda vermelha demonstra o comportamento do modal quando o código é copiado através do botão “Copiar Código”. Por fim, a imagem com destaque em borda azul da Figura 15 apresenta o modal que é exibido ao cliente para confirmar que ele realmente deseja cancelar o agendamento.



**Figura 15. Modal de QR Code, modal com código Pix copiado e modal de cancelar agendamento**

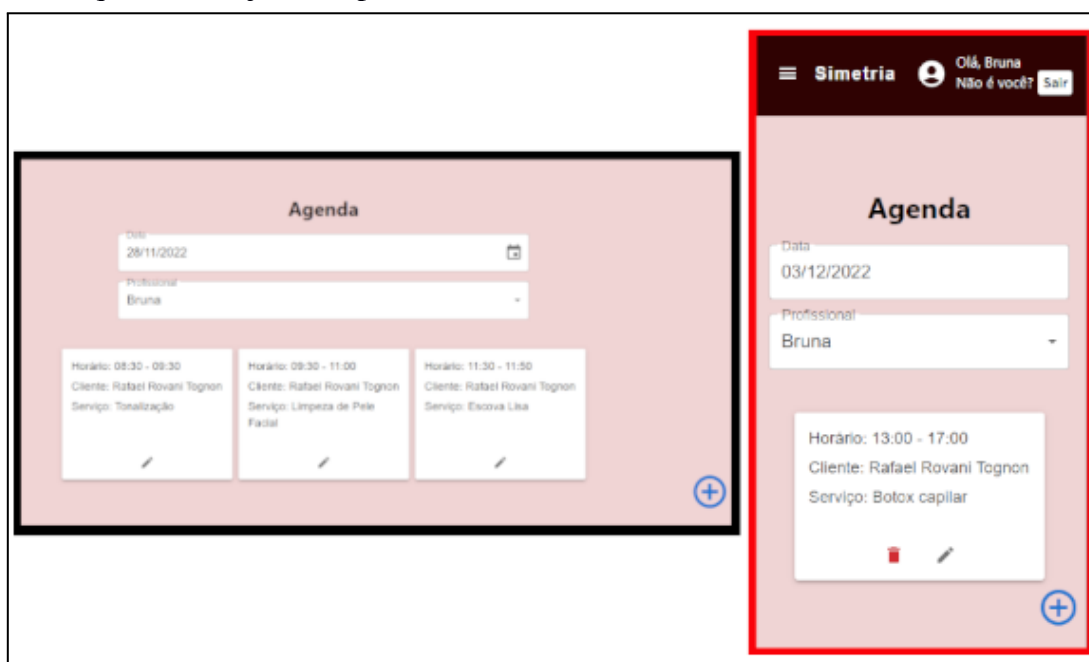
## **6.2 Módulo de administração**

Neste tópico, os resultados referentes ao módulo de administrador obtidos serão apresentados. Inicialmente, a Figura 16 em borda preta apresenta a tela de login do módulo de administração, e a Figura 16 de borda vermelha apresenta a tela inicial apresentada quando o login não foi realizado, que oferece a opção de autenticar-se através dos botões “Entrar”.



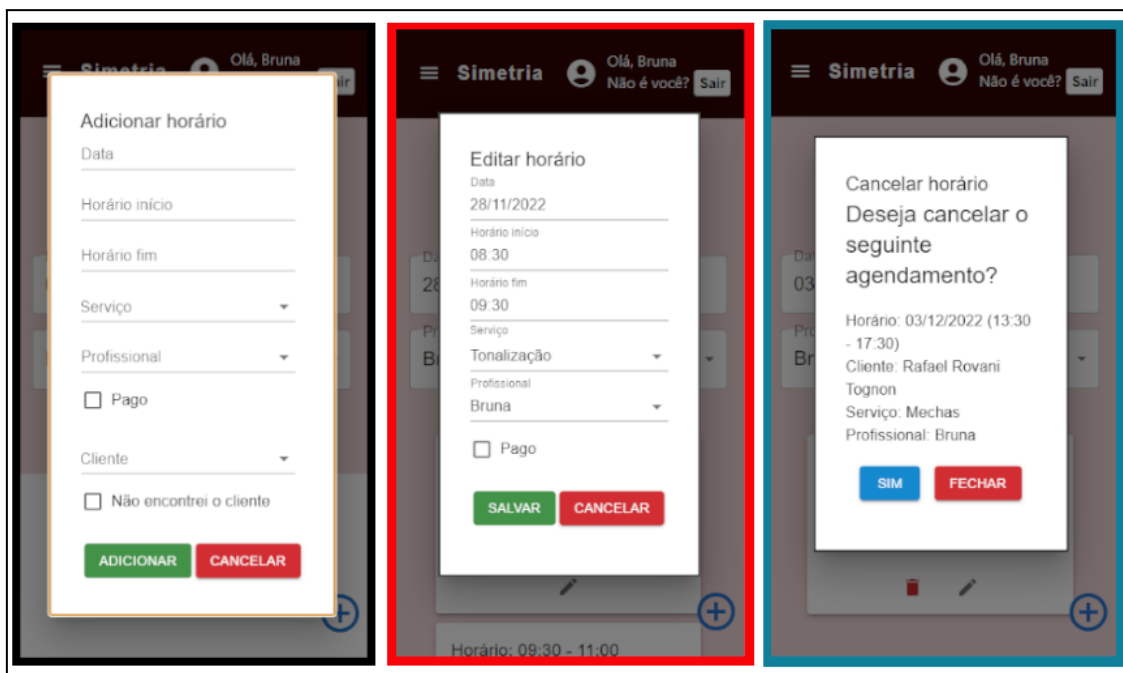
**Figura 16. Tela de login e tela inicial com acesso não autorizado**

Em seguida, a Figura 17 apresenta a tela para visualização dos agendamentos na forma *desktop* (borda preta) e na forma *mobile* (borda vermelha). Nela, a administradora pode adicionar novos agendamentos, bem como visualizar, editar e cancelar agendamentos existentes (desde que eles já não tenham sido realizados, ou seja, o horário que termina já tenha passado) .



**Figura 17. Tela de agendamentos *desktop* e *mobile***

Em seguida, a Figura 18 de borda preta apresenta o modal que é aberto quando o botão “+”, que serve para adicionar um agendamento, é acionado. Já a Figura 18 com borda vermelha apresenta o modal aberto para editar as propriedades de um agendamento existente, que é aberto pela ação de clique no botão de Editar, representado por um ícone de lápis. Por fim, a Figura 18 de borda azul apresenta o modal que é aberto para confirmar a ação de cancelar o agendamento quando o botão de cancelar, representado pelo ícone de lixeira, é acionado.



**Figura 18. Modal de adicionar agendamento, modal de edição de agendamento e modal de confirmação de para cancelar agendamento**

## 7. Considerações finais

O desenvolvimento dessa aplicação reforçou os conhecimentos adquiridos nas disciplinas de Programação, Banco de Dados e Engenharia de Software, além de várias outras que foram ofertadas ao longo do curso. Ademais, foi necessário adquirir conhecimentos sobre o desenvolvimento com a abordagem PWA, autenticação através da conta *Google* e sobre a biblioteca *React*.

Além disso, foi necessário adquirir alguns conhecimentos adicionais, como a integração com a *Application Programming Interface* (API) da Gerencianet (que é o intermediário para o pagamento através do Pix), a hospedagem do banco de dados no *Heroku*, o *front-end* na *Vercel* e o *back-end* na *Amazon Web Services* (AWS).

Com o *software*, é esperado que o Espaço Estético Simetria organize sua agenda de forma mais prática e gere os processos internos do estabelecimento.

No que diz respeito às futuras implementações no módulo de cliente, podem ser adicionadas outras formas de pagamento, como o cartão de crédito. Também podem ser

adicionadas mais opções de *login* por meio de contas externas, como através do *Facebook* e apresentar notícias cadastradas pelo administrador.

Em relação a futuras melhorias no módulo de administrador, pode-se considerar a edição de horário de funcionamento para um dia específico e a opção de agendar serviços em que mais de uma profissional atende (como o dia de princesa). Além disso, pode-se oferecer as opções de editar os profissionais e serviços, cadastrar notícias e registrar a fórmula de coloração utilizada por um cliente pela interface do módulo.

## Referências

- Developer Mozilla (2022) “Progressive web apps (PWAs)”, [https://developer.mozilla.org/en-US/docs/web/progressive\\_web\\_apps](https://developer.mozilla.org/en-US/docs/web/progressive_web_apps), Março.
- Fecomércio-RS (2021) “Sondagens de Segmentos - Salões de Beleza”, <https://fecomercio-rs.org.br/wp-content/uploads/2021/09/Sondagem-de-Sal%C3%B5es-de-Beleza.pdf>, Março.
- Figma (2022) “Prototype while you design, and vice versa”, <https://www.figma.com/prototyping/>, Junho.
- Heuser, Carlos A. (1998) “Projeto de Banco de Dados”, Instituto de Informática da UFRGS, 4ª edição.
- IBGE (2020) “Pesquisa de Orçamentos Familiares”, <https://biblioteca.ibge.gov.br/visualizacao/livros/liv101761.pdf>, Março.
- Microsoft (2022) “Visão geral dos Aplicativos Web Progressivos (PWAs)”, <https://docs.microsoft.com/pt-br/microsoft-edge/progressive-web-apps-chromium/#:~:text=Os%20PWAs%20s%C3%A3o%20sites%20que,executados%20em%20navegadores%2C%20como%20sites>, Março.
- My SQL (2022) “Why MySQL?”, <https://www.mysql.com/why-mysql/>, Junho.
- Pressman, Roger S. (2011), “Engenharia de Software Uma Abordagem Profissional”, AMGH Editora Ltda, 7ª edição.
- React (2022) “Uma biblioteca JavaScript para criar interfaces de usuário”, <https://pt-br.reactjs.org/>, Novembro.
- Sebrae (2022) “Como montar um salão de beleza”, <https://www.sebrae.com.br/sites/PortalSebrae/ideias/como-montar-um-salao-de-beleza,a,42287a51b9105410VgnVCM1000003b74010aRCRD#apresentacao-de-negocio>, Março.
- Sommerville, I (2011), “Engenharia de Software”, Pearson Prentice Hall, 9a edição.
- Tecmundo (2020) “Google substitui apps do Android por PWAs no Chrome OS“, <https://www.tecmundo.com.br/software/152045-google-substitui-apps-android-pwa-chrome.htm>, Abril.