



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO TOCANTINS
CAMPUS COLINAS DO TOCANTINS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

MATHEUS DE ABREU SILVA

**StyleHub: Desenvolvimento de um aplicativo para gestão de salões de beleza e
barbearias**

COLINAS DO TOCANTINS

2026

MATHEUS DE ABREU SILVA

**StyleHub: Desenvolvimento de um aplicativo para gestão de salões de beleza e
barbearias**

Trabalho de Conclusão de Curso
apresentado à Coordenação do Curso de
Tecnologia em Análise e Desenvolvimento
de Sistemas, do *Campus* Colinas do
Tocantins do Instituto Federal do
Tocantins, como exigência à obtenção do
título de tecnólogo(a) em Análise e
Desenvolvimento de Sistemas.

Orientador(a): Esp. Mateus Nunes dos
Santos.

COLINAS DO TOCANTINS

2026

Dados Internacionais de Catalogação na Publicação (CIP)
Bibliotecas do Instituto Federal do Tocantins

S586s Silva, Matheus de Abreu Silva
StyleHub: Desenvolvimento de um aplicativo para gestão de
salões de beleza e barbearias / Matheus de Abreu Silva Silva. –
Araguaína, TO, 2026.
76 p. : il. color.

Trabalho de Conclusão de Curso (Curso Superior de Tecnologia
em Análise e Desenvolvimento de Sistemas) – Instituto Federal de
Educação, Ciência e Tecnologia do Tocantins, Campus Colinas do
Tocantins, Colinas do Tocantins, TO, 2026.

Orientador: Esp. Mateus Nunes Dos Santos

1. Aplicativo. 2. Software. 3. Barbearias. I. Nunes Dos Santos,
Mateus. II. Título.

CDD 004.2

A reprodução total ou parcial, de qualquer forma ou por qualquer meio, deste documento é autorizada para fins
de estudo e pesquisa, desde que citada a fonte.

**Elaborado pelo sistema de geração automática de ficha catalográfica do IFTO com os dados fornecidos
pelo(a) autor(a).**

MATHEUS DE ABREU SILVA

**STYLEHUB: DESENVOLVIMENTO DE UM APLICATIVO PARA GESTÃO DE
SALÕES DE BELEZA E BARBEARIAS**

Trabalho de curso apresentado ao curso de Tecnologia do Instituto Federal do Tocantins – Campus Colinas do Tocantins, como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, sob orientação do Prof. Esp. Mateus Nunes dos Santos.

Banca Examinadora

Esp. Mateus Nunes dos Santos
Instituto Federal do Tocantins Campus Colinas do Tocantins

Documento assinado digitalmente
 **MATEUS NUNES DOS SANTOS**
Data: 12/06/2026 14:46:58-0300
Verifique em <https://validar.iti.gov.br>

Me. Fernando Turibio de Moura
Instituto Federal do Tocantins Campus Colinas do Tocantins

Documento assinado digitalmente
 **FERNANDO TURIBIO DE MOURA**
Data: 12/06/2026 16:57:06-0300
Verifique em <https://validar.iti.gov.br>

Grad. Leonardo Xavier Brito
Instituto Federal do Tocantins Campus Colinas do Tocantins

Documento assinado digitalmente
 **LEONARDO XAVIER BRITO**
Data: 12/06/2026 17:57:26-0300
Verifique em <https://validar.iti.gov.br>

Aprovada em 03 de Junho de 2026

Dedico este trabalho a Deus, por me conceder força e sabedoria ao longo dessa jornada, e à minha família, pelo apoio, incentivo e compreensão em todos os momentos.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me conceder sabedoria e força ao longo dessa jornada, à minha família pelo apoio em todas as etapas, incentivo e compreensão nos momentos mais difíceis, por último e não menos importante a todos os profissionais e professores do campus por todos os ensinamentos e conhecimentos compartilhados comigo durante esse período de estudos, e em menção especial ao meu orientador por toda a ajuda, paciente e contribuição essencial para o desenvolvimento desse trabalho, além disso, meu desenvolvimento pessoal como estudante. Aos meus colegas e amigos, que de alguma forma contribuíram para a realização desse projeto, em especial aos colegas, Ana Clara Carrilho Silva e Pedro Henrique Ribeiro Gonçalves, que desde o começo e de perto acompanharam a criação desse projeto, colaborando diretamente para a realização e melhorias dessa aplicação desenvolvida.

“A tecnologia move o mundo.”
(Steve Jobs)

RESUMO

Este trabalho apresenta o desenvolvimento do StyleHub, um aplicativo voltado à gestão e ao agendamento de serviços em barbearias e salões de beleza. A proposta surgiu a partir da necessidade de organizar atendimentos, reduzir conflitos de horários, melhorar o controle financeiro e proporcionar uma melhor experiência aos usuários. O sistema foi desenvolvido com foco em dois perfis principais, cliente e barbeiro, permitindo a realização de agendamentos, gerenciamento de serviços, acompanhamento financeiro e organização das informações do empreendimento. A metodologia adotada foi baseada em práticas ágeis inspiradas no XP, com desenvolvimento incremental e testes ao longo do processo. Os resultados indicaram boa aceitação do sistema, demonstrando que a aplicação atende aos objetivos propostos e pode contribuir para a modernização da gestão de pequenos negócios do setor de beleza.

Palavras-chave: StyleHub. Aplicativo. Agendamento. Barbearias. Financeiro. Empreendimento.

ABSTRACT

This work presents the development of StyleHub, an application focused on management and service scheduling for barbershops and beauty salons. The proposal emerged from the need to organize appointments, reduce scheduling conflicts, improve financial control, and provide a better experience for users. The system was developed with a focus on two main user profiles, clients and barbers, allowing appointment scheduling, service management, financial monitoring, and organization of business information. The methodology adopted was based on agile practices inspired by XP, with incremental development and continuous testing throughout the process. The results demonstrated good user acceptance, showing that the application meets the proposed objectives and can contribute to the modernization of management processes in small businesses within the beauty industry.

Keywords: StyleHub. Application. Scheduling. Barbershops. Financial management. Business management.

SUMÁRIO

1 INTRODUÇÃO.....	10
1.1 Objetivo geral.....	11
1.2 Objetivos específicos.....	11
1.3 Justificativa.....	12
2 REFERENCIAL TEÓRICO.....	14
2.1 Plataforma do sistema.....	15
2.2 Tecnologias utilizadas.....	15
2.2.1 Linguagens.....	15
2.2.2 JavaScript.....	15
2.2.3 PHP.....	16
2.2.4 Frameworks.....	17
2.2.4.1 Laravel.....	17
2.2.4.2 Ionic.....	18
2.2.4.3 Vue.Js.....	19
2.2.5 Bibliotecas.....	19
2.2.5.1 Axios.....	20
2.2.6 API REST.....	20
2.2.7 Banco de dados.....	21
2.2.7.1 MySQL.....	22
2.2.8 Android Studio.....	22
2.3 Metodologias de desenvolvimento ágeis.....	23
2.3.1 Extreme Programming (XP).....	23
3 METODOLOGIA.....	25
3.1 Softwares e Hardwares utilizados.....	25
3.2 Método de desenvolvimento.....	26
4 SOFTWARE.....	28
4.1 Arquitetura.....	28
4.2 Diagramas.....	29
4.3 Funcionalidades do Sistema.....	31
4.3.1 Cliente.....	31
4.3.2 Barbeiro.....	32
4.4 Banco de dados.....	33
4.5 Telas do sistema.....	35
4.6 Testes do Sistema.....	49
4.6.1 Testes técnicos.....	49
4.6.2 Testes com usuários.....	50
4.6.3 Resultados obtidos.....	50
5 CONCLUSÃO.....	52
REFERÊNCIAS.....	55
APÊNDICES.....	58

Apêndice A - Questionário de opinião aplicado durante os testes do sistema.....	58
Apêndice B - Registros fotográficos dos testes do sistema.....	59
Apêndice C - Registros da pesquisa de aplicativos similares.....	65
Apêndice D - Trechos de código do sistema.....	71

1 INTRODUÇÃO

Tecnologia da Informação (TI) tem recebido as mais variadas definições. Em seu início, a computação era tida como um mecanismo que tornava possível automatizar determinadas tarefas em grandes empresas e nos meios governamentais. Com o avanço tecnológico, as “máquinas gigantes” começaram a perder espaço para equipamentos cada vez menores, mais poderosos e mais rápidos. A evolução das telecomunicações permitiu que, aos poucos, os computadores passassem a se comunicar. Como consequência, tais máquinas deixaram de simplesmente automatizar tarefas e passaram com sistemas de informações (LEMOS II, 2011, p.51).

Para Lemos (2011), as aplicações de Tecnologia da Informação são muitas e estão ligadas às mais diversas áreas, e nenhuma consegue determiná-la por completo. O termo TI também é comumente utilizado para designar o conjunto de recursos não humanos dedicados ao armazenamento, processamento e comunicação da informação, bem como, o modo de como esses recursos estão organizados num sistema capaz de executar um conjunto de tarefas ou automatizar ações.

Os anos consolidaram o celular como um dispositivo essencial na vida moderna. Desde seu surgimento, o celular evoluiu de um simples meio de comunicação para um aparelho multifuncional que altera a dinâmica de interação. Hoje em dia utilizamos nossos aparelhos digitais para rodar softwares e aplicativos (WhatsApp, jogos, navegadores) para funções como internet, câmera, GPS, multimídia e não menos importante, acessar plataformas de serviços e trabalho.

A Tecnologia tem desempenhado um papel cada vez mais importante nos tempos contemporâneos, com diversas atividades do dia-a-dia dependendo de sua utilização, tomando como base sua aplicação no desenvolvimento de sistemas de informação voltados à área de serviços. No campo do trabalho, a tecnologia tem automatizado processos e aumentado a eficiência. A inteligência artificial e a automação têm substituído tarefas repetitivas, permitindo que os profissionais se concentrem em atividades mais estratégicas e criativas. Nesse contexto, tais sistemas tornam-se ferramentas fundamentais para a organização e centralização de dados, apoiando a gestão operacional e administrativa dos empreendimentos.

A gestão de serviços, por sua vez, apresenta desafios específicos, como o acompanhamento de processos, o monitoramento de custos e o controle financeiro. Desde os processos mais simples até os que necessitam de mais atenção, o sistema é crucial para centralizar e integrar informações, com o propósito de auxiliar o empreendedor a ter controle dos processos e manter um padrão de qualidade. (SILVESTRE, 2024).

“O mercado de estética masculina está cada vez mais competitivo, saturado e orientado por tecnologia e conveniência. Para se destacar neste cenário e garantir o sucesso a longo prazo, não basta apenas oferecer um bom corte de cabelo ou um serviço de barba impecável; é absolutamente necessário ter previsibilidade de receita, organização diária e foco no aumento real do faturamento. É exatamente diante deste grande desafio operacional que a adoção de um app (aplicativo) para barbeiros se torna o divisor de águas entre um negócio que apenas sobrevive apagando incêndios e uma barbearia que prospera, cresce e enriquece seus donos” (CORTHY, 2026).

Quando a agenda falha ou é negligenciada por falta de um app para barbeiros eficiente, surgem atrasos, faltas de clientes, conflitos internos com a equipe e uma perda silenciosa, mas constante, de dinheiro. Com um app para barbeiros, é totalmente possível visualizar os atendimentos em tempo real, ajustar horários rapidamente e manter tudo sempre atualizado, mesmo quando há mudanças e imprevistos.

Um app para barbearias e salões de beleza completo, permite que o cliente agende o seu serviço online, 24 horas por dia, 7 dias por semana. Essa praticidade significa que a sua barbearia continua aberta e recebendo agendamentos mesmo durante horários adversos ou em dias fechados, capturando clientes que seriam perdidos caso dependessem de um atendimento por celular ou manual.

1.1 Objetivo geral

Desenvolver um sistema para dispositivos móveis voltado à gestão e ao agendamento de serviços em barbearias e salões de beleza.

1.2 Objetivos específicos

- Melhorar a experiência do cliente no agendamento.
- Organizar a gestão do empreendimento.
- Controlar conflitos de horários.

- Melhorar a gestão financeira.
- Implementar mecanismos de segurança para proteger os dados e integridade das informações dos usuários.

1.3 Justificativa

No início do desenvolvimento do projeto foram realizadas pesquisas nas lojas de aplicativos de celular, Play Store (Android) e App Store (IOS) por apps de gestão e agendamentos com foco no mercado de beleza e estética, haviam alguns poucos, entretanto, com propagandas, valores altos e funcionalidades que não atendiam 100% das propostas do StyleHub. Os registros dessas pesquisas podem ser consultados no apêndice C. Além de pesquisas em barbearias, salões e empreendedores de pequeno porte na cidade, Nova Olinda - TO, com questionamentos sobre resolução de desafios para agendamentos, gestão e controle financeiro de pequenos negócios com baixos custos. Assim como proposto por este projeto, configurando-se assim uma excelente opção neste segmento.

Segundo Online *Booking Software*, em 2024, a gestão de empresas de beleza e barbearia apresentam uma complexibilidade que vai muito além de apenas fazer tratamentos e cortes de cabelo. Mais da metade das barbearias e salões não utilizam sistema para gerenciar os agendamentos, finanças e estoque. Tal impasse é um dos principais causadores de desafios para esse estilo de negócio, impedindo que clientes consigam agendar serviços de forma prática e eficiente, proprietários enfrentando dificuldades em controlar fluxo de caixa, e despesas além de outras. O acúmulo de tais erros ocasionam até em fechamentos de alguns estabelecimentos.

“No mundo moderno, a tecnologia tem sido uma grande aliada na transformação de diversos setores e no mercado da beleza não é diferente. Com as ferramentas certas, é possível melhorar a organização de horários, agilizar processos e aplicar o marketing de forma estratégica. Tudo isso impacta diretamente no aumento do faturamento. Para alcançar esses resultados, a implementação de um **sistema de gestão para barbearias** eficiente é o primeiro passo para o sucesso” (XAVIER, 2026).

O uso de agendas de papel é um dos maiores atrasos na produtividade de um barbeiro. Um sistema de gestão para barbearias ameniza falhas no atendimento e promete que nenhum horário será perdido por falta de resposta no WhatsApp. Com o agendamento online, você oferece praticidade total para seus clientes, que podem marcar serviços 24 horas por dia (XAVIER, 2026).

Portanto, foi possível observar uma enorme gama de oportunidades para o desenvolvimento de aplicativos e softwares para a área de beleza em geral, que espera-se que supram as necessidades tanto do lado do empreendedor como do lado do cliente, através da implementação de um sistema de agendamento online e gestão, podendo minimizar problemas financeiros, organizacionais e em relação a conflitos de horários.

2 REFERENCIAL TEÓRICO

A seguir apresento um artigo produzido pela Frizzar utilizado como principal base teórica de estudo para o desenvolvimento desse projeto, com tema “Por que a gestão efetiva é fundamental para Barbearias de Sucesso”.

Segundo Frizzar (2025), ter uma boa gestão é essencial para o sucesso de qualquer barbearia. Um gerenciamento bem estruturado transforma o atendimento ao cliente em uma experiência agradável e eficiente. Mas como isso pode ser feito na prática?

Primeiro, é importante lembrar que a experiência do cliente começa muito antes dele sentar na cadeira do barbeiro. Desde o agendamento até o momento em que ele sai do local, tudo deve ser cuidadosamente planejado (FRIZZAR, 2025).

- Agendamento sem complicações:
 - Ter um sistema de agendamento eficiente evita esperas prolongadas e conflitos de horários, mostrando ao cliente que o tempo dele é valorizado.
- Comunicação clara:
 - Todas as informações sobre serviços e preços devem ser acessíveis e fáceis de entender. Isso garante que o cliente saiba exatamente o que esperar.
 - Como a gestão Financeira pode transformar seu negócio?

A gestão financeira é a base para o sucesso de qualquer barbearia. Muitas vezes, é a diferença entre fechar as portas ou prosperar no mercado. Primeiramente, entender para onde o dinheiro está indo é crucial. Isso significa ter controle sobre todas as despesas e receitas (FRIZZAR, 2025). Usar um software específico para o mercado de beleza pode facilitar muito esse processo. Automatizar a gestão das finanças garante precisão e economia de tempo. Com essa ferramenta, você pode se concentrar no que realmente importa: atender bem seus clientes e fazer a empresa crescer.

- Qual a importância do controle de agendamentos?

Investir em gestão para barbearias é mais do que apenas manter tudo em ordem. Imagine chegar na sua barbearia favorita e descobrir que seu horário de

corpo foi confundido com o de outra pessoa. Isso pode ser frustrante, certo? (FRIZZAR, 2025). É por isso que evitar os conflitos de agendamentos e longas esperas são essenciais. Com a boa gestão de negócio através de um software, tem bons resultados para que cada cliente seja atendido no horário certo, sem surpresas desagradáveis.

2.1 Plataforma do sistema

Conceituado como aplicativo móvel, o projeto foi desenvolvido para usuários de smartphones das plataformas Android e IOS para versões do Android 7.0 e IOS 15.0 ou superiores. Segundo Taysa Bocard (2023), aplicativos móveis, também abreviados de “apps” ou chamados de “app mobile”, são softwares desenvolvidos exclusivamente para dispositivos móveis como celulares e tablets, gratuitos ou pagos. Eles se popularizaram a partir do primeiro smartphone, em 2007, com o sistema operacional IOS. Hoje já estão em outros dispositivos como *smart* TVs e vestíveis. O objetivo deles é disponibilizar funcionalidades de forma fácil e intuitiva.

2.2 Tecnologias utilizadas

2.2.1 Linguagens

No vasto mundo da computação, existe um elemento essencial que permite aos humanos comunicar-se com as máquinas eficazmente: a linguagem de programação. Assim como as linguagens humanas, as linguagens de programação são sistemas estruturados e formais que permitem instruir dispositivos eletrônicos para executar tarefas específicas e automatizar ações (VIDAL, 2023).

Em um nível fundamental, uma linguagem de programação pode ser definida como um conjunto de regras, símbolos e códigos que os desenvolvedores usam para criar programas de software. Porém, por trás desta aparente simplicidade reside um conjunto complexo de estruturas, regras e funções que tornam possível o funcionamento de todo o ecossistema tecnológico em que diariamente nos imergimos (VIDAL, 2023).

2.2.2 JavaScript

Utilizada no frontend, segundo a AWS Amazon ([s.d.]), JavaScript é uma linguagem de programação usada por desenvolvedores para fazer páginas interativas da Internet. As funções do JavaScript podem melhorar a experiência do

usuário durante a navegação em um site, como, por exemplo, desde a atualização do feed na página da mídia social até a exibição de animações e siglas do nome do usuário no perfil de forma personalizada. Como uma linguagem de script do lado do cliente, ele é uma das tecnologias principais da World Wide Web(famoso WWW ou rede mundial de computadores). Por exemplo, ao navegar na Internet, é possível visualizar a qualquer momento um carrossel de imagens, um menu suspenso, “clique para visualizar” ou mesmo mudar dinamicamente as cores dos elementos de uma página da Web. Tudo isso graças ao JavaScript.

No contexto do projeto, o JavaScript é utilizado para gerenciar a interatividade da interface do aplicativo, tornando as telas mais dinâmicas e responsivas. Além disso, é responsável pela lógica de comunicação com a API (Application Programming Interface ou Interface de Programação de Aplicações), permitindo o envio e o recebimento de dados do backend de forma assíncrona ou em tempo real. Dessa forma, possibilita a atualização das informações do usuário, contribuindo para uma melhor experiência de uso da aplicação móvel.

2.2.3 PHP

PHP (*Hypertext Preprocessor* (Pré-Processador de Hipertexto)) é uma linguagem open-source de programação do lado do servidor, voltada para a web. Ela funciona a partir de uma linguagem de scripts, ou seja, instruções em forma de código. Com isso, é possível criar aplicações como websites, lojas virtuais, dentre outras. A maioria do núcleo do WordPress, que representa grande parte dos sites na web, está em PHP.

“WordPress é um sistema de gerenciamento de conteúdo (CMS) gratuito e de código aberto que permite que qualquer pessoa crie e gerencie sites com facilidade. Criado como uma plataforma de blog, o software do WordPress evoluiu ao longo dos anos e, hoje em dia, permite que seus usuários criem vários tipos de sites, desde blogs e portfólios até lojas virtuais completas.” (BRUNO, 2025).

Isso o torna uma das linguagens mais importantes no cenário de desenvolvimento web. Essa linguagem já faz parte da produção web há mais de 20 anos. Além de ser mundialmente conhecida, é fácil de aprender e compatível com praticamente todos os sistemas operacionais, diminuindo os custos no desenvolvimento e implementação (NOGUEIRA, 2025).

O PHP foi a escolha para o desenvolvimento da API (Application Programming Interface ou Interface de Programação de Aplicações) de backend do projeto, sendo responsável pelo processamento das requisições, manipulação e integração com o banco de dados. Apresenta uma grande documentação disponível online. Sua utilização justifica-se pela facilidade de integração com bancos de dados relacionais e pelo suporte à criação de APIs REST utilizando o formato JavaScript *Object Notation* (JSON) para troca de informações. Além disso, segundo Carlos E., 2020, PHP apresenta baixo custo de implementação, grande disponibilidade em micro serviços e uma comunidade atuante mesmo sendo uma linguagem mais antiga, fatores que contribuem para a manutenção e escalabilidade do projeto.

2.2.4 Frameworks

Em leitura de um artigo produzido pela BeTrybe, em meados de 2020, *Frameworks de software* são conceituados como: “uma série de bibliotecas e classes, ou seja, códigos prontos que oferecem alguma funcionalidade específica”. Em outras palavras, é um padrão que pode ser incorporado a sistemas para agilizar a codificação de certas partes.

“Suas funções são adequadas para determinadas tarefas repetitivas e mais simples, que geralmente são comuns em softwares de diferentes naturezas. Assim, o programador pratica o reuso de código. Esse conjunto de funcionalidades é como uma máquina que automatiza parte do seu trabalho para que você consiga focar em outras tarefas.” (BETRYBE,[s.d.]).

O projeto foi desenvolvido utilizando frameworks, divididos em backend e frontend/mobile. O uso desses para acelerar o processo de implementação do app, com a utilização de códigos e bibliotecas prontas para as linguagens PHP e JavaScript, sendo internas, desenvolvidas pelos próprios criadores das linguagens e externas, desenvolvidas por terceiros.

2.2.4.1 Laravel

Segundo a documentação oficial do Laravel ([2026]), Laravel é um framework de aplicação web com sintaxe expressiva e elegante. Um framework web que fornece uma estrutura e um ponto de partida para criar seu aplicativo, permitindo que você se concentre em criar algo incrível enquanto cuidamos dos detalhes.

O Laravel se esforça para fornecer uma experiência incrível ao desenvolvedor, ao mesmo tempo em que fornece recursos poderosos, como injeção

completa de dependência, uma camada expressiva de abstração de banco de dados, filas e trabalhos programados, testes de unidade e integração e muito mais (LARAVEL, [s.d]).

O framework Laravel é utilizado neste projeto como responsável pela implementação do *backend*, sendo encarregado da construção da API de comunicação entre o frontend e o banco de dados. É utilizado principalmente para gerenciar as requisições realizadas pelos usuários por meio do frontend (interface do sistema), atuando como intermediário no processamento e na persistência das informações.

Com seu conjuntos de códigos prontos e sua arquitetura baseada no padrão MVC (*Model-View-Controller*) permite a organização do código em camadas, facilitando a manipulação dos dados por meio dos modelos, o controle das regras de negócio pelos controladores além da definição de rotas responsáveis pela comunicação da aplicação. O laravel acelera a criação de algoritmos para o desenvolvimento pois disponibiliza um conjuntos de funcionalidades prontas, tornando a padronização e manutenção do sistema mais fácil, sendo adequado a sua implementação no projeto.

2.2.4.2 Ionic

Segundo a documentação oficial do Ionic ([2026]), tem-se que o framework é um kit de ferramentas de interface do usuário de código aberto para criar aplicativos móveis de alta qualidade e desempenho, usando tecnologias da web *HyperText Markup Language*, ou Linguagem de Marcação de Hipertexto (HTML), *Cascading Style Sheets* (CSS) e JavaScript, com integrações para estruturas populares de geração de códigos *frontend* prontos como Angular, React, e Vue.

No contexto do projeto, o Ionic é utilizado para a criação das interfaces gráficas da aplicação, estando relacionado à experiência do usuário e sua interação com o sistema. Sua escolha justifica-se pelo reaproveitamento de código, característica que permite o desenvolvimento de aplicações híbridas a partir de uma mesma base de código. Além disso, possibilita a construção de designs responsivos, compatíveis com diferentes plataformas, como dispositivos Android, IOS e também ambientes web, com baixa necessidade de aplicar mudanças ao código principal.

Dessa forma, foi escolhido como uma solução adequada dentro do contexto de desenvolvimento de apps híbridos e de tal projeto, visando uma maior escalabilidade e aceitação de mercado.

2.2.4.3 Vue.Js

Segundo a documentação oficial do framework, Vue ([s.d]) (pronuncia-se /vju:/, como view, em inglês) é um framework progressivo para a construção de interfaces de usuário. Ao contrário de outros *frameworks* monolíticos, Vue foi projetado desde sua concepção para ser adotável incrementalmente. A biblioteca principal é focada exclusivamente na camada visual (*view layer*), sendo fácil adotar e integrar com outras bibliotecas ou projetos existentes. Por outro lado, Vue também é perfeitamente capaz de dar poder a sofisticadas *Single-Page Applications* quando usado em conjunto com ferramentas modernas e bibliotecas de apoio.

No contexto do projeto, enquanto o Ionic é responsável pela construção das interfaces gráficas, o Vue.js atua como complemento ao aplicar dinamismo e interatividade às telas do sistema. Na aplicação, o Vue.js é utilizado para gerenciar a reatividade da interface, respondendo às interações do usuário, como cliques, preenchimento de formulários e navegação entre telas. Como exemplo dentro do sistema proposto, pode-se destacar a atualização automática da lista de agendamentos após a realização de um novo item, a exibição dinâmica de horários disponíveis para o agendamento e a alteração em tempo real de informações na interface. Dessa forma, o Vue.js contribui para a criação de uma experiência mais fluida e interativa aos usuários.

2.2.5 Bibliotecas

Uma biblioteca de programação consiste em um conjunto de funções reutilizáveis, funcionando como uma caixa de ferramentas com recursos prontos para uso no seu código sempre que necessário. Diferentemente de aplicações autônomas, as bibliotecas não são executadas por conta própria; seu código é quem as “chama” para executar tarefas específicas. Por exemplo, ao precisar realizar uma requisição de rede, processar imagens ou executar assinaturas criptográficas, a biblioteca de programação oferece “funções” pré-montadas (recursos executáveis) e “APIs” (interfaces de chamada), evitando que você precise recriar soluções já

existentes. Isso acelera o desenvolvimento e reduz o risco de erros humanos (GATE, 2013-2026).

2.2.5.1 Axios

Segunda a documentação oficial do Axios ([s.d.]), Axios é um cliente *Hypertext Transfer Protocol* (HTTP) poderoso que oferece uma API simples e fácil de usar para fazer requisições HTTP. Ele suporta todos os navegadores modernos e é amplamente utilizado na comunidade JavaScript.

Segundo o artigo publicado na Dev Community, por Matheus Chaves, 2023, ao usar o Axios, pode-se fazer solicitações HTTP para consumir APIs REST, enviar dados para um servidor diferente ou realizar qualquer outra operação de rede. Além disso, o Axios oferece recursos como filtro de solicitação e resposta, que permitem manipular entradas e saídas de maneira fácil e contínua.

O Axios é utilizado no projeto como uma biblioteca responsável pela realização de requisições HTTP, estabelecendo a comunicação entre o cliente (*frontend*) e o servidor (*backend*). Sua utilização contribui para a padronização das requisições e evita a repetição de código ao longo da aplicação. Dentro do sistema, o Axios é utilizado no envio das informações de login para autenticação dos usuários fazendo o uso do método HTTP POST, e também no recebimento dos dados do usuário autenticado. Após o login, a aplicação realiza requisições utilizando o método *GET*, à exemplo: na busca de horários disponíveis, que são posteriormente exibidos na tela para o usuário. Com isso, tem-se que o Axios a integração entre as camadas do sistema.

A configuração da biblioteca no projeto, incluindo a definição da URL base e o interceptor responsável pela injeção automática do token JWT nas requisições, pode ser visualizada no apêndice C1 ao final do projeto.

2.2.6 API REST

Uma interface de programação de aplicação (API) define as regras que você precisa seguir para se comunicar com outros sistemas de software. Os desenvolvedores expõem ou criam APIs para que outras aplicações possam se comunicar com suas aplicações de forma recorrente (AWS AMAZON, [2026]). Por exemplo, a aplicação de planilha de horas expõe uma API que solicita o nome

completo de um funcionário e um intervalo de datas. Ao receber essas informações, processa internamente a planilha de horas do funcionário e retorna o número de horas trabalhadas nesse intervalo de datas.

“APIs REST se comunicam por meio de solicitações HTTP para executar funções padrão de banco de dados, como criar, ler, atualizar e excluir registros (também conhecidos como CRUD) em um recurso. Por exemplo, uma API REST usaria uma solicitação *GET* para recuperar um registro. Uma solicitação *POST* cria um novo registro. Uma solicitação *PUT* atualiza um registro, e uma solicitação *DELETE* exclui um. Todos os métodos HTTP podem ser usados em chamadas de API. Uma API REST bem projetada é semelhante a um site em execução em um navegador web com funcionalidade HTTP integrada” (IBM, [2026]).

O projeto completo é baseado na arquitetura de API REST, na qual as aplicações de frontend e backend são organizadas em camadas independentes, necessitando de comunicação para o correto funcionamento do sistema. A API atua como intermediária na troca de dados entre o cliente e o servidor, sendo responsável pelo processamento das requisições e pela interação com o banco de dados por meio dos modelos e controles definidos na aplicação. A comunicação é realizada seguindo os padrões da arquitetura REST, utilizando requisições HTTP e o formato JSON, “um padrão leve muito utilizado para a troca de informações entre sistemas” (MOZILLA, [s.d]). Como exemplo prático, no cadastro de usuários, os dados são enviados pelo frontend, processados pela API no backend e, em seguida, armazenados no banco de dados. Dessa forma, a utilização de API REST contribui para a organização do sistema, tornando suas camadas independentes e facilitando sua manutenção e escalabilidade.

2.2.7 Banco de dados

Um banco de dados é uma coleção organizada de informações - ou dados - estruturadas, normalmente armazenadas eletronicamente em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (SGBD - Sistema de Gerenciamento de Banco de Dados). Juntos, os dados e o SGBD, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados (ORACLE, 2020).

2.2.7.1 MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional (SGBD) de código aberto usado para armazenar e gerenciar dados. Sua confiabilidade, desempenho, escalabilidade e facilidade de uso fazem do MySQL uma escolha popular entre os desenvolvedores. Na verdade, você o encontrará no centro de aplicações exigentes e de alto tráfego, como Facebook, Netflix, Uber, Airbnb, Shopify e Booking.com (ERICKSON, 2024).

A aplicação proposta assim como quase todas as aplicações web utilizam e necessitam de banco de dados para armazenamento e persistência das informações, sendo elas de origem do cliente ou de origem do próprio sistema. Nesse contexto, como conceituado anteriormente, o banco de dados do projeto foi desenvolvido com a utilização do sistema de gerenciamento de banco de dados relacional, MySQL, baseado como o próprio nome já diz na linguagem SQL (*Structured Query Language*) amplamente utilizada para manipulação de dados.

“SQL (*Structured Query Language*), que significa linguagem de consulta estruturada em inglês, é uma linguagem de programação usada para recuperar, atualizar, excluir e manipular dados em bancos de dados relacionais. Oficialmente, "SQL" deve ser lido como uma sigla, mas é comumente usado como um acrônimo” (ERICKSON, 2024).

Em aplicação prática é usado para guardar, recuperar e analisar diversos tipos de dados, incluindo texto simples, números, datas, horas. Por exemplo, guardar de forma relacional os dados dos usuários cadastrados, agendamentos realizados, os serviços cadastrados pelos barbeiros, além de garantir a persistência dos dados ao longo do tempo. Escolhido devido ao seu “desempenho, a facilidade de uso e o baixo custo, combinados com sua capacidade de dimensionar de forma confiável à medida que uma empresa cresce” (ERICKSON, 2024). Tornando-o adequado ao projeto e totalmente escalável pensando em atualizações futuras do sistema.

2.2.8 Android Studio

Segundo a documentação oficial, o Android Studio é o ambiente de desenvolvimento integrado (IDE) oficial para o desenvolvimento de apps Android. O Android Studio oferece ainda mais recursos para aumentar sua produtividade ao criar apps Android.

“A ferramenta é gratuita e oferece recursos que permitem a desenvolvedores iniciantes ou experientes programar, testar e compartilhar apps compatíveis com celulares, tablets e outros dispositivos que utilizam o sistema operacional do robô verde. O ambiente é compatível com as linguagens Java e C/C++, além de contar com integração a outros serviços, como o Gemini Google. Na matéria a seguir, saiba o que é e para que serve o IDE do Google” (CATALDO, 2025).

O Android Studio foi utilizado neste projeto como ambiente auxiliar para execução, compilação e testes da aplicação mobile em dispositivos Android, possibilitando a validação prática das funcionalidades implementadas no sistema.

2.3 Metodologias de desenvolvimento ágeis

Para Marco Tulio Valente (2019), a ideia de processos ágeis é que um sistema deve ser construído de forma incremental e iterativa. Pequenos incrementos de funcionalidade são produzidos, em intervalos de cerca de um mês e, logo em seguida, validados pelos usuários. Uma vez que o incremento produzido seja aprovado, o ciclo se repete.

“Processos ágeis tiveram um profundo impacto na indústria de software. Hoje, eles são usados pelas mais diferentes organizações que produzem software, desde pequenas empresas até as grandes companhias da Internet. Diversos métodos que concretizam os princípios ágeis foram propostos, tais como **XP**, **Scrum**, **Kanban** e **Lean Development**” (VALENTE, 2019, p.35).

Esses métodos também ajudaram a disseminar diversas práticas de desenvolvimento de software, como testes automatizados, test-driven development (isto é, escrever os testes primeiro, antes do próprio código) e integração contínua (*continuous integration*). Integração contínua recomenda que desenvolvedores integrem o código que produzem imediatamente, se possível todo dia. O objetivo é evitar que desenvolvedores fiquem muito tempo trabalhando localmente, em sua máquina, sem integrar o código que estão produzindo no repositório principal do projeto (VALENTE, 2019, p.35).

2.3.1 *Extreme Programming (XP)*

XP é uma metodologia leve indicada para o desenvolvimento de softwares quando os requisitos ainda não estão totalmente definidos ou podem mudar durante o projeto. Por ser uma metodologia ágil, ela segue ciclos curtos e repetidos de desenvolvimento, dá menos foco a documentações extensas e planejamentos muito detalhados, e permite que o design do sistema seja construído aos poucos, conforme o projeto evolui. Além disso, o XP costuma ser mais adequado para

equipes pequenas, facilitando a comunicação e a adaptação durante o processo de desenvolvimento (VALENTE, 2019, p.64).

“Porém, XP não é um método prescritivo, que define um passo a passo detalhado para construção de software. Em vez disso, XP é definido por meio de um conjunto de **valores, princípios e práticas de desenvolvimento**. Ou seja, XP é inicialmente definido de forma abstrata, usando-se de valores e princípios que devem fazer parte da cultura e dos hábitos de times de desenvolvimento de software. Depois, esses valores e princípios são concretizados em uma lista de práticas de desenvolvimento. Frequentemente, quando decidem adotar XP, desenvolvedores e organizações concentram-se nas práticas. Porém, os valores e princípios são componentes chaves do método, pois são eles que dão sentido às práticas propostas em XP. Sendo mais claro, se uma organização não está preparada para trabalhar no modelo mental de XP — representado pelos seus valores e princípios — recomenda-se também não adotar suas práticas” (VALENTE, 2019, p.64).

A metodologia adotada para o desenvolvimento deste projeto foi baseada em práticas ágeis, inspiradas no XP. Embora o desenvolvimento tenha sido realizado por uma única pessoa, a utilização dessa abordagem contribuiu para a organização das etapas do projeto, definição de metas e acompanhamento do progresso. O desenvolvimento foi dividido em pequenas partes, que eram implementadas em intervalos de tempo, geralmente entre uma e duas semanas. As definições das próximas etapas e a avaliação das funcionalidades desenvolvidas eram realizadas em encontros periódicos com o orientador, por meio de reuniões online e exponencialmente reuniões presenciais no próprio campus. Essa forma de organização foi aplicada durante todo o processo, contribuindo para a redução de erros, melhor controle do desenvolvimento e melhoria da qualidade final do sistema.

3 METODOLOGIA

Este capítulo apresenta a metodologia utilizada no desenvolvimento do projeto, mostrando as ferramentas empregadas, os recursos utilizados e o método adotado para a construção da aplicação. Dessa forma, busca-se descrever as etapas e processos envolvidos durante todo o tempo de desenvolvimento do sistema proposto, desde o levantamento de requisitos até a fase de testes reais com usuários.

3.1 Softwares e Hardwares utilizados

Para o desenvolvimento do sistema proposto, foram utilizados diferentes softwares e recursos de hardware, que contribuíram para a construção, testes e validação da aplicação. Essas ferramentas foram escolhidas por sua compatibilidade com as tecnologias adotadas no projeto, bem como pela facilidade de uso e disponibilidade.

No desenvolvimento do sistema, foi utilizado o ambiente de desenvolvimento Visual Studio Code (V 1.83) como principal editor de código, auxiliando na implementação das funcionalidades tanto do *frontend* quanto do *backend*. Para o *backend*, foram utilizados a linguagem PHP (V 8.1) em conjunto com o framework Laravel (V 12.0), responsáveis pela criação da API e processamento das requisições do sistema. Já no *frontend*, foram empregados o Ionic *Framework* (V 4.0) e o Vue.js (V 2.0), possibilitando a construção de interfaces dinâmicas e responsivas, além da utilização do JavaScript (V ES2025) para a lógica da aplicação.

Para a comunicação entre frontend e backend, foi utilizada a biblioteca Axios (V 1.14), responsável pela realização de requisições HTTP. No gerenciamento do banco de dados, foi adotado o MySQL (V 8.4), com auxílio do MySQL Workbench (V 2.3) e do phpMyAdmin (V 5.2) para modelagem, administração e manipulação dos dados. Durante o desenvolvimento local, foi utilizado o XAMPP (V 8.3) como ambiente de servidor.

Para testes das requisições da API, foi utilizado o Postman (V 11.74), permitindo validar as funcionalidades do *backend* de forma isolada. O controle de versão do projeto foi realizado por meio do Git (V 2.54), com armazenamento remoto

no GitHub (V 3.5). Para testes de execução da aplicação web, foi utilizado o navegador Microsoft Edge (V 147.0).

Além disso, foram utilizadas ferramentas de apoio ao desenvolvimento, como o Figma (V 2025) para prototipação das interfaces e o Adobe Photoshop (V 26.11) para criação e edição de elementos visuais do sistema. O Android Studio (V PANDA 4) foi utilizado como ambiente auxiliar para execução, compilação e testes da aplicação mobile em um dispositivos Android, possibilitando a validação prática das funcionalidades implementadas no sistema.

Em relação ao hardware, o desenvolvimento foi realizado em um notebook Asus VivoBook (Windows 11, versão 25H2) Integrado com processador Ryzen 5 series 4000, 20GB de memória RAM (*random Access Memory*, ou Memória de Acesso Aleatório) e 256GB de memória interna SSD (*Solid State Drives* ou unidade de estado sólido) com acesso à internet, utilizado para programação e testes do sistema. Também foi utilizado um dispositivo móvel iPhone 12 com sistema operacional IOS (V 26.4), 6GB de memória RAM e 128GB de memória interna e um dispositivo A05 com sistema operacional Android (V 13.0), 4GB de memória RAM e 128GB de memória interna, para testes da aplicação em ambiente real, permitindo avaliar a usabilidade e o comportamento do sistema em dispositivos móveis.

3.2 Método de desenvolvimento

O desenvolvimento do sistema foi realizado com base em uma abordagem, inspirada em práticas de metodologias ágeis baseadas no XP. Essa forma de desenvolvimento permitiu a organização do projeto em etapas, facilitando o acompanhamento da evolução do sistema e a realização de ajustes ao longo do processo.

Inicialmente, foi realizado o levantamento dos requisitos do sistema através de entrevistas com possíveis usuários do sistema como cliente e alguns barbeiros da cidade de Nova Olinda-TO, considerando as principais necessidades de gestão em salões de beleza e barbearias, como controle de agendamentos, cadastro de usuários e organização financeira. A partir disso, foi possível definir as funcionalidades que seriam implementadas no sistema.

O desenvolvimento foi dividido em pequenas partes, que eram implementadas em intervalos de tempo, geralmente entre uma e duas semanas. A cada etapa concluída, as funcionalidades eram avaliadas e ajustadas conforme necessário. As definições das próximas etapas, revisões do que já estava concluído na aplicação e possíveis novas ideias eram realizadas em encontros com o orientador, por meio de reuniões online através do google *Meet* ou presenciais.

Durante o processo, foi adotada a prática de desenvolvimento por funcionalidades, permitindo que cada parte do sistema fosse construída, testada e validada antes da implementação da próxima, por exemplo, a implementação do frontend só foi iniciada a partir da finalização dos CRUDS (C é de *Create* Criar, R é de *Read* Ler, U é de *Update* Atualizar e D é de *Delete* Apagar) básicos da api. Essa abordagem contribuiu para a redução de alguns erros, melhor organização do código e maior controle sobre o processo de desenvolvimento do projeto.

Além disso, foram realizados testes a cada implementação ao longo do desenvolvimento, tanto no *backend* quanto no *frontend*, com o objetivo de verificar o funcionamento correto das funcionalidades implementadas e garantir a boa integração entre as diferentes partes do sistema. Dessa forma, a metodologia adotada contribuiu para a construção de um sistema mais estável, organizado e alinhado com os objetivos propostos ao início do projeto.

4 SOFTWARE

Este capítulo apresenta o sistema desenvolvido neste trabalho (StyleHub), descrevendo sua estrutura, organização e principais funcionalidades. Com abordagem na base da aplicação, relacionada à arquitetura do sistema, modelagem, funcionamento, implementação e testes, com o objetivo de demonstrar como a aplicação foi construída e como atende às necessidades propostas.

4.1 Arquitetura

A arquitetura do sistema desenvolvido é baseada no modelo cliente-servidor, no qual as funcionalidades da aplicação são divididas em diferentes camadas, permitindo maior organização, manutenção e escalabilidade do sistema. Nesse modelo, o *frontend* é responsável pela interface com o usuário, comunica-se com o *backend*, que realiza o processamento das informações e a interação com o banco de dados.

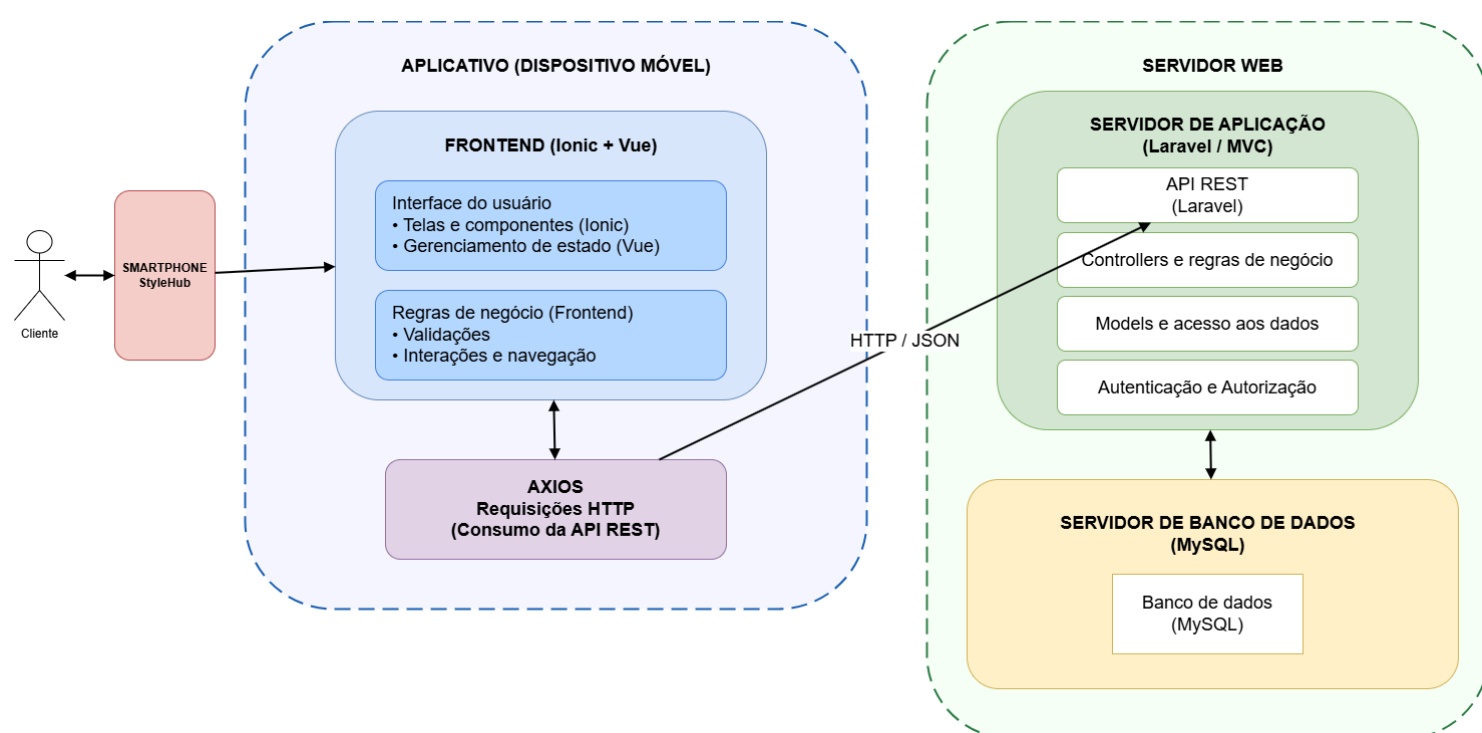
No contexto do projeto, o *frontend* foi desenvolvido utilizando o framework Ionic em conjunto com o Vue.js e a linguagem JavaScript, sendo responsável pela apresentação das interfaces gráficas, reatividade e pela interação com o usuário. Já o *backend* foi implementado com o modelo *Model-View-Controller* (MVC) “padrão de arquitetura de software, separa a aplicação em 3 camadas. A camada de interação do usuário, a camada de manipulação dos dados e a camada de controle” (SCHONS, 2023), usando a linguagem PHP e o framework Laravel, responsável pelo processamento das requisições, controles das regras de negócio e gerenciamento dos dados. Um exemplo do método responsável pelo processamento e persistência de um novo agendamento no backend, incluindo a validação de conflitos de horário, está disponível no apêndice C4.

A comunicação entre o *frontend* e o *backend* é realizada por meio de uma API REST, utilizando requisições HTTP e o formato JSON para troca de informações. Isso por meio da biblioteca Axios no *frontend*, permitindo o envio e recebimento de dados de forma assíncrona. As rotas responsáveis pelo gerenciamento dos agendamentos, definidas no arquivo de rotas da aplicação, podem ser visualizadas no apêndice C3.

O armazenamento das informações é realizado em um banco de dados relacional MySQL, responsável pela persistência dos dados do sistema, como usuários, agendamentos e serviços cadastrados e financeiro.

Dessa forma, a arquitetura adotada permite a separação entre as camadas do sistema, facilitando sua manutenção, organização e já deixando pré pronto para as futuras expansões já idealizadas, além de contribuir para um melhor desempenho e reutilização dos componentes da aplicação.

Diagrama 01 - Arquitetura do sistema



Fonte: Desenvolvido pelo autor (2026).

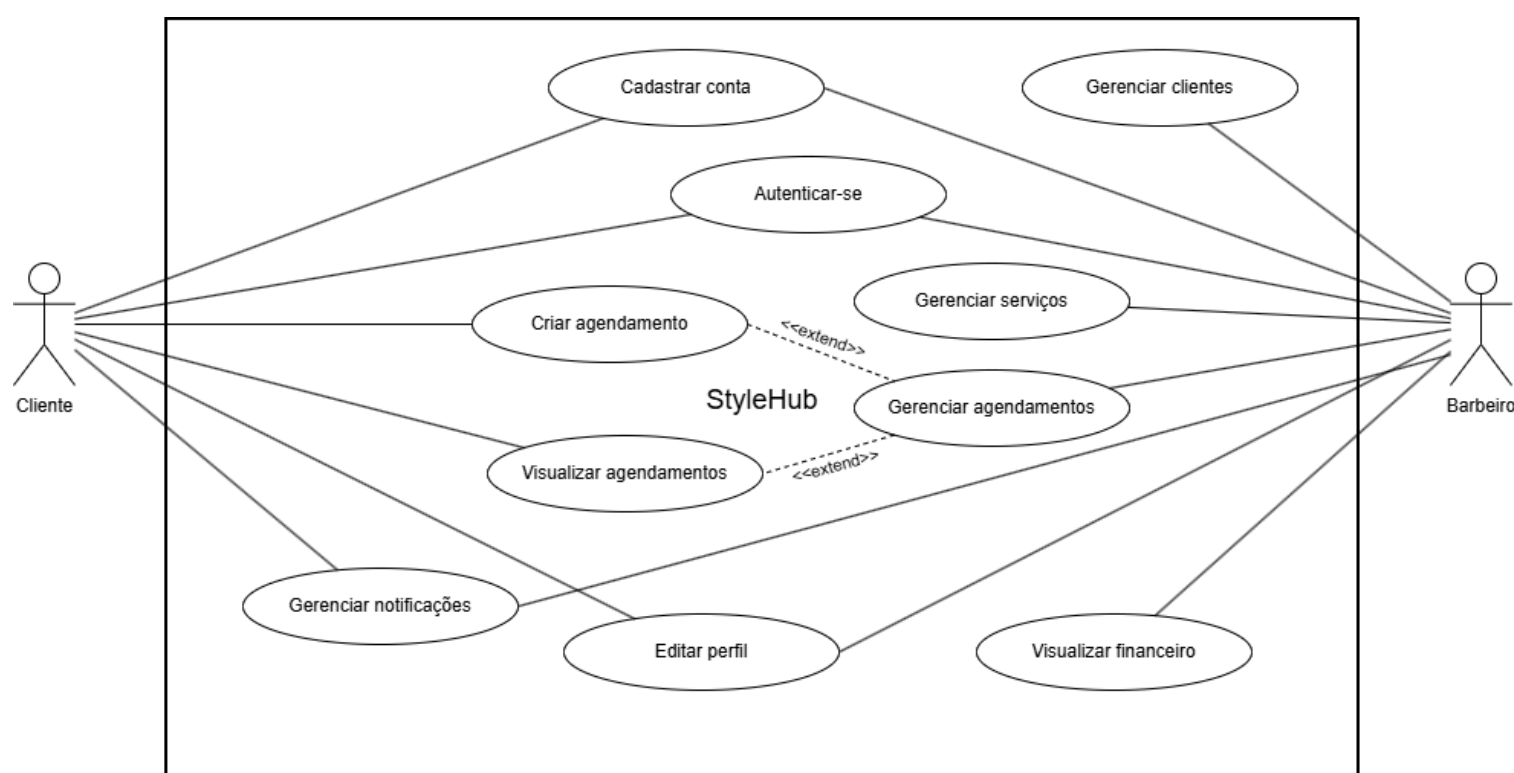
O Diagrama 01 apresenta a arquitetura do sistema, evidenciando a comunicação entre o usuário, a interface da aplicação, a API e o banco de dados.

4.2 Diagramas

Nesta etapa do trabalho, é apresentado os diagramas do projeto, desenvolvidos utilizando o diagrams.net. Sendo um software gratuito para a construção de diagramas *Unified Modeling Language* ou Linguagem de modelagem

unificada (UML). A UML estabelece uma linguagem de modelagem visual comum, semanticamente e sintaticamente rica, para arquitetura, design e implementação de sistemas de software complexos, tanto estruturalmente quanto para comportamentos. Além do desenvolvimento de software, a UML tem aplicações em fluxos do processo de fabricação (LUCIDCHART, [s.d]). Os diagramas desenvolvidos foram os de caso de uso, ligado ao que se observa no diagrama 01, vista anteriormente.

Diagrama 02 - Casos de uso



Fonte: Desenvolvido pelo autor (2026).

O diagrama 02 apresenta os casos de uso do sistema. Principais interações e funcionalidades do sistema Stylehub com os atores demonstrados. Nesse exemplo apresentei dois perfis de uso, Cliente e Barbeiro. O Cliente possui acesso a funcionalidades relacionadas ao cadastro, autenticação, criação e visualização de agendamentos, gerenciamento de notificações e edição de perfil. Já o Barbeiro possui acesso às funcionalidades de gestão, como gerenciamento de clientes, serviços, agendamentos, notificações, perfil e visualização financeira.

“<<Extend>>” indica que uma funcionalidade pode complementar ou ampliar o comportamento de outra.

O objetivo deste diagrama é representar, de forma visual e simplificada, as ações que cada tipo de usuário pode realizar no sistema, facilitando a compreensão de cada perfil dentro da aplicação.

4.3 Funcionalidades do Sistema

Este tópico apresenta as principais funcionalidades do sistema desenvolvido, descrevendo as ações disponíveis para os usuários e como elas contribuem para o funcionamento da aplicação. As funcionalidades foram organizadas de acordo com os perfis de acesso, sendo eles, cliente e barbeiro.

4.3.1 Cliente

O usuário do tipo cliente possui acesso às funcionalidades relacionadas ao agendamento de serviços e interação com a barbearia. Entre as principais funcionalidades, destacam-se:

- Cadastro de conta
- Autenticação no sistema
- Visualização da tela inicial com informações gerais
- Consulta de horários disponíveis
- Realização de agendamentos: **Permite ao usuário selecionar o barbeiro desejado, escolher a data, o serviço e o horário de atendimento, além de possibilitar a verificação das informações antes da confirmação do agendamento. O trecho de código responsável por essa funcionalidade no frontend, com a validação dos dados e o envio das informações para a API, está disponível no apêndice C2.**
- Visualização e cancelamento de agendamentos
- Aplicação de cupons de desconto
- Visualização de barbeiros e galeria de fotos
- Recebimento e gerenciamento de notificações
- Edição de dados pessoais e alteração de senha
- Visualização da localização da barbearia

4.3.2 Barbeiro

O usuário do tipo barbeiro possui acesso às funcionalidades de gestão do sistema, permitindo o controle das informações e operações da barbearia. Entre as principais funcionalidades, destacam-se:

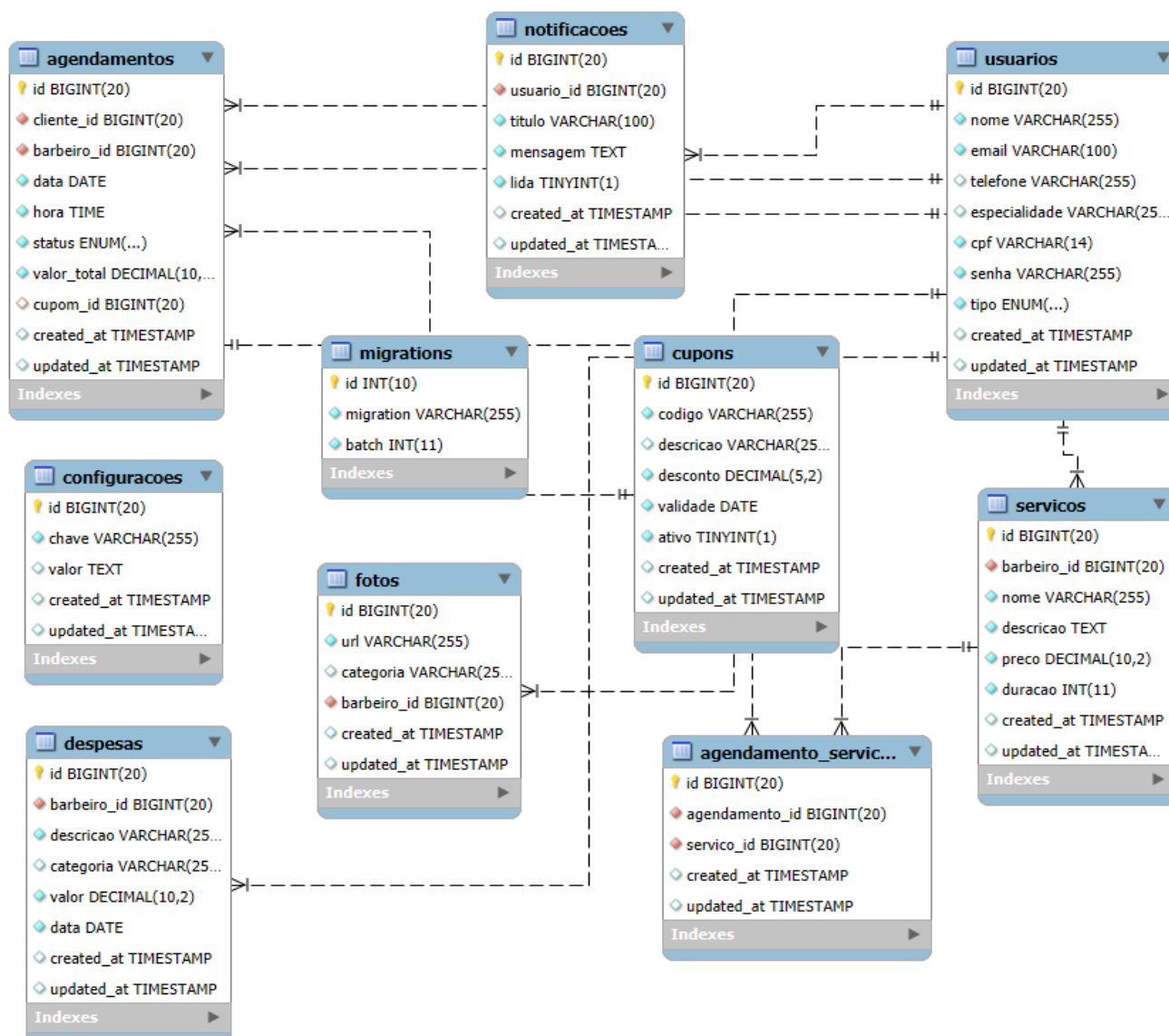
- Autenticação no sistema
- Visualização da tela inicial com resumo de atividades
- Visualização e gerenciamento de agendamentos: **Permite ao usuário consultar seus agendamentos, bem como realizar ações como cancelamento, confirmação ou conclusão, garantindo maior controle sobre os atendimentos.**
- Consulta da agenda diária
- Gerenciamento de serviços (cadastro, edição e exclusão): **Permite ao usuário cadastrar, editar e remover serviços oferecidos, possibilitando a organização das opções disponíveis para agendamento no sistema.**
- Gerenciamento de clientes: **Permite ao usuário visualizar, editar e remover informações dos clientes cadastrados, possibilitando um melhor controle e organização da barbearia.**
- Gerenciamento de cupons
- Gerenciamento da galeria de imagens
- Controle de despesas
- Visualização de informações financeiras: **Permite ao usuário acompanhar dados financeiros do sistema, como faturamento e despesas, possibilitando uma melhor organização e controle das atividades financeiras do estabelecimento.**
- Envio e gerenciamento de notificações: No perfil de barbeiro permite o envio de notificações para os usuários do sistema, bem como o recebimento e gerenciamento dessas mensagens perdidos aos dois tipos de usuários, possibilitando a comunicação eficiente entre o sistema e os usuários.
- Edição de perfil
- Configuração de horários disponíveis: **Permite ao usuário definir os horários de atendimento disponíveis para agendamento, possibilitando maior controle sobre sua agenda e organização dos atendimentos.**
- Configuração de localização

- Definição de metas mensais

4.4 Banco de dados

Este tópico apresenta a estrutura do banco de dados utilizado no sistema, mostrando as principais entidades e seus relacionamentos. O modelo foi desenvolvido com o objetivo de garantir a organização, integridade e persistência das informações e dados manipulados pela aplicação.

Diagrama 03 - Entidade-Relacionamento (DER)



Fonte: Elaborado pelo autor (2026).

O banco de dados do sistema foi modelado de forma relacional, sendo composto por entidades que representam os principais elementos da aplicação, como usuários, agendamentos, serviços e demais informações necessárias para o funcionamento do sistema. A persistência dos dados pode ser comprovada no apêndice C6, que apresenta os registros reais armazenados na tabela de agendamentos durante a fase de testes do sistema.

A entidade de usuários armazena os dados dos clientes e barbeiros, permitindo o controle de acesso e identificação dentro do sistema. Já a entidade de serviços está relacionada aos barbeiros, possibilitando o cadastro e gerenciamento dos serviços oferecidos.

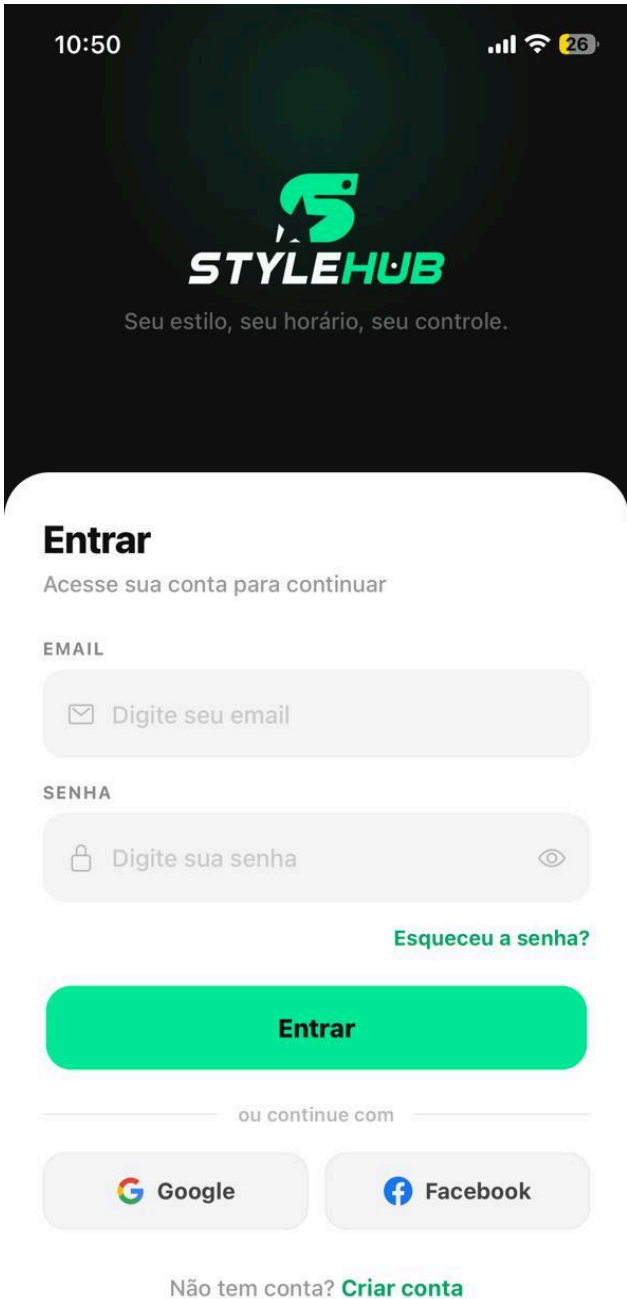
A tabela de agendamentos faz um papel central no sistema, sendo responsável por registrar as informações relacionadas aos atendimentos, como data, horário, serviço e usuário vinculado. Essa entidade se relaciona com usuários e serviços, garantindo a organização e integridade das informações. A definição do modelo de agendamentos, contendo os campos permitidos e seus relacionamentos com as demais entidades, pode ser visualizada no apêndice C5. Além disso, o sistema conta com outras entidades auxiliares, como controle financeiro, notificações e cupons, que contribuem para o funcionamento completo da aplicação.

Dessa forma, a estrutura do banco de dados foi projetada para atender às necessidades do sistema, garantindo consistência, organização e eficiência no armazenamento e recuperação das informações.

4.5 Telas do sistema

Este tópico mostra as interfaces do sistema desenvolvido e descreve de forma sucinta, evidenciando o layout e a organização das telas disponíveis para os usuários. As telas foram projetadas com o objetivo de proporcionar uma navegação simples e intuitiva, facilitando a utilização das funcionalidades do sistema.

Figura 01 - Tela de login (Cliente/Barbeiro)



A imagem mostra a interface de login do aplicativo StyleHub. No topo, há uma barra de status com o horário 10:50, ícones de sinal de rede e bateria (26%). O fundo da tela é preto com o logo 'STYLEHUB' em verde e branco no centro, acompanhado do slogan 'Seu estilo, seu horário, seu controle.' Abaixo, há uma seção branca com o título 'Entrar' e o subtítulo 'Acesse sua conta para continuar'. Seguem campos de entrada para 'EMAIL' e 'SENHA', ambos com ícones de envelope e cadeado, respectivamente. Um link 'Esqueceu a senha?' está à direita do campo de senha. Um botão verde 'Entrar' está abaixo dos campos. Abaixo do botão, há a opção 'ou continue com' seguida por botões para 'Google' e 'Facebook'. No rodapé, há o link 'Não tem conta? Criar conta'.

10:50

STYLEHUB

Seu estilo, seu horário, seu controle.

Entrar

Acesse sua conta para continuar

EMAIL

✉ Digite seu email

SENHA

🔒 Digite sua senha

[Esqueceu a senha?](#)

Entrar

ou continue com

 Google

 Facebook

Não tem conta? [Criar conta](#)

Fonte: Elaborado pelo autor (2026).

Figura 01 apresenta a tela de login do sistema StyleHub. A tela de login permite que o usuário acesse o sistema por meio de suas credenciais, garantindo a autenticação e segurança no acesso às funcionalidades disponíveis.

Figura 02 - Tela de Cadastro (Cliente/Barbeiro)

A imagem mostra a interface de usuário para a criação de uma conta no aplicativo StyleHub. No topo, há uma barra de status com o horário 10:50, ícones de sinal de rede e bateria (26%). Abaixo, o logo do StyleHub em verde e o texto "Crie sua conta e agende seu próximo corte". O formulário principal, intitulado "Criar conta", pede para o usuário preencher seus dados para começar. Os campos de entrada são: "NOME COMPLETO" (com ícone de pessoa), "EMAIL" (com ícone de envelope), "CPF" (com ícone de documento), "TELEFONE" (com ícone de telefone), "SENHA" (com ícone de cadeado e aviso de "Mínimo 8 caracteres") e "CONFIRMAR SENHA" (com ícone de cadeado e aviso de "Repita sua senha"). Cada campo tem um ícone de ajuda à esquerda e um ícone de visibilidade à direita. No final do formulário, há um botão verde com o texto "Criar conta".

Fonte: Elaborado pelo autor (2026).

A figura 02 apresenta a tela de cadastro do sistema StyleHub. A tela de cadastro permite ao usuário realizar seu registro no sistema por meio do preenchimento de informações pessoais. Essa funcionalidade é responsável por criar o acesso do usuário à aplicação, garantindo a identificação e possibilitando o uso.

Figura 03 - Tela Inicial (Cliente)

Fonte: Elaborado pelo autor (2026)

A figura 03 apresenta a tela inicial do sistema StyleHub. A tela inicial apresenta um resumo das informações do sistema, como próximos agendamentos e acesso rápido às principais funcionalidades, barbeiros, notificações e galeria de imagens, facilitando a navegação do usuário.

Figura 04 - Tela de novo agendamento (Cliente)

Novo Agendamento
Preencha os dados abaixo

BARBEIRO

- BT** Barbeiro seco
- PR** preto Barbeiro

SERVIÇOS

- ☐ **Corte Masculino**
R\$ 90,00 · 40 min
- ☐ **Barba**
R\$ 25,00 · 30 min
- ☐ **Corte + Barba**
R\$ 50,00 · 60 min
- ☐ **Sobrancelha**
R\$ 15,00 · 15 min
- ☐ **corte 45**
R\$ 45,00 · 45 min

SELECIONE A DATA

abril de 2026 > < >

DOM. SEG. TER. QUA. QUI. SEX. SÁB.

1 2 3 4

5 6 7 8 9 10 11

HORÁRIO DISPONÍVEL

- 13:00
- 13:30
- 14:00
- 14:30
- 15:00
- 15:30
- 16:00
- 16:30
- 17:00
- 17:30

Cupom de desconto

☒ **STYLE10** **Aplicar**

Cupom inválido.

RESUMO

Data	20/04/2026
Serviços	Corte Masculino
Barbeiro	Barbeiro teste
Horário	13:00
Cupom	style10
Total	R\$ 90,00

Confirmar Agendamento

Fonte: Elaborado pelo autor (2026)

A Figura 04 apresenta a tela de agendamento do sistema StyleHub. A tela de agendamento permite ao usuário selecionar o barbeiro, serviço, data e horário disponíveis, possibilitando a realização de agendamentos de forma prática e organizada.

Figura 05 - Tela de agendamentos ou agenda (Cliente)



Fonte: Elaborado pelo autor (2026)

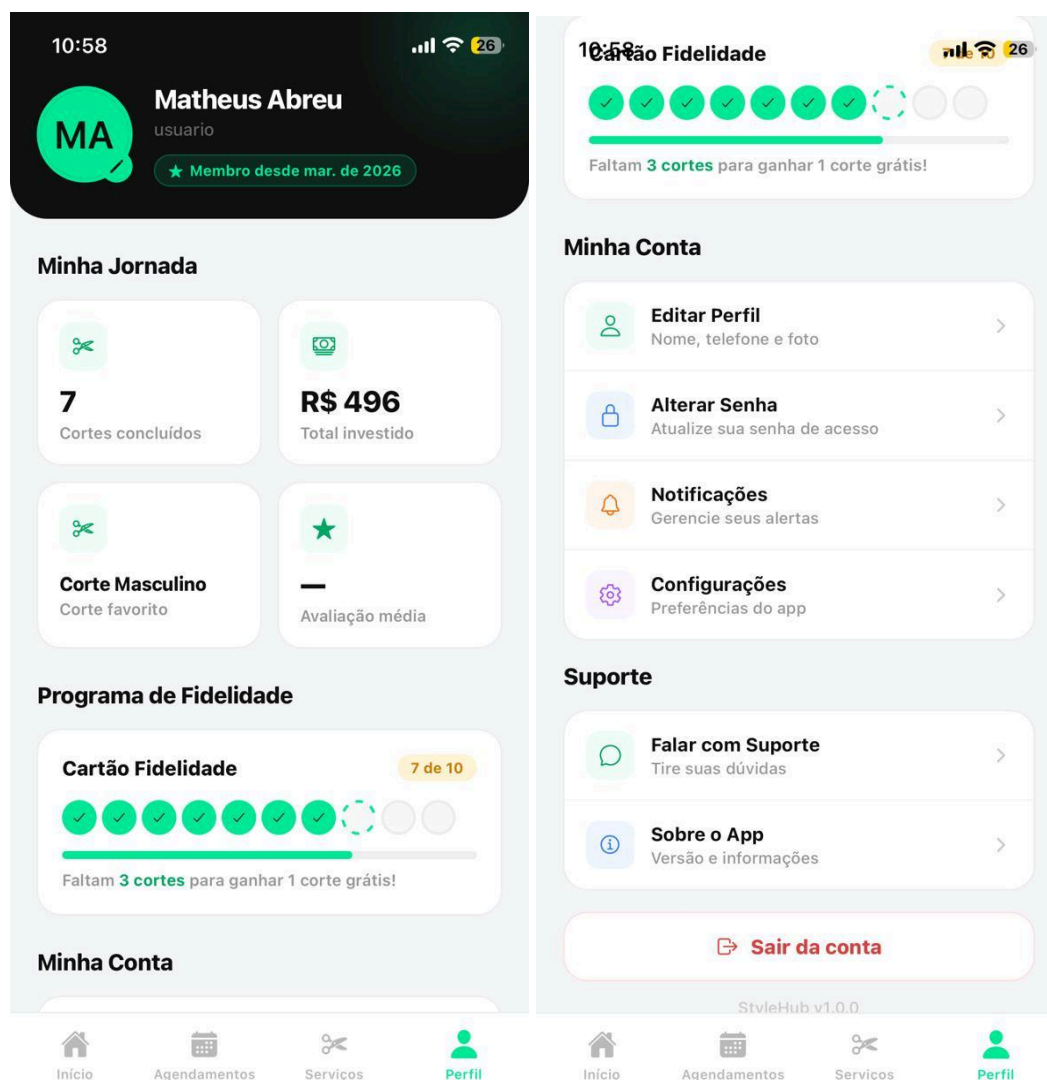
A figura 05 apresenta a tela de agendamentos do sistema StyleHub. A tela de agendamentos permite ao usuário visualizar os agendamentos realizados, apresentando informações como data, horário, serviço e barbeiro responsável. Além disso, possibilita a realização de ações como cancelamento ou acompanhamento do status do agendamento, proporcionando maior controle e organização dos atendimentos.

Figura 06 - Tela de serviços (Cliente)

Fonte: Elaborado pelo autor (2026)

A figura 06 apresenta a tela de serviços do sistema StyleHub. A tela de serviços permite ao usuário do tipo cliente visualizar os serviços disponíveis oferecidos pelos barbeiros, apresentando informações como nome, descrição e valor. Essa funcionalidade possibilita ao usuário conhecer os serviços oferecidos e de forma rápida ir para um novo agendamento estando passos à frente.

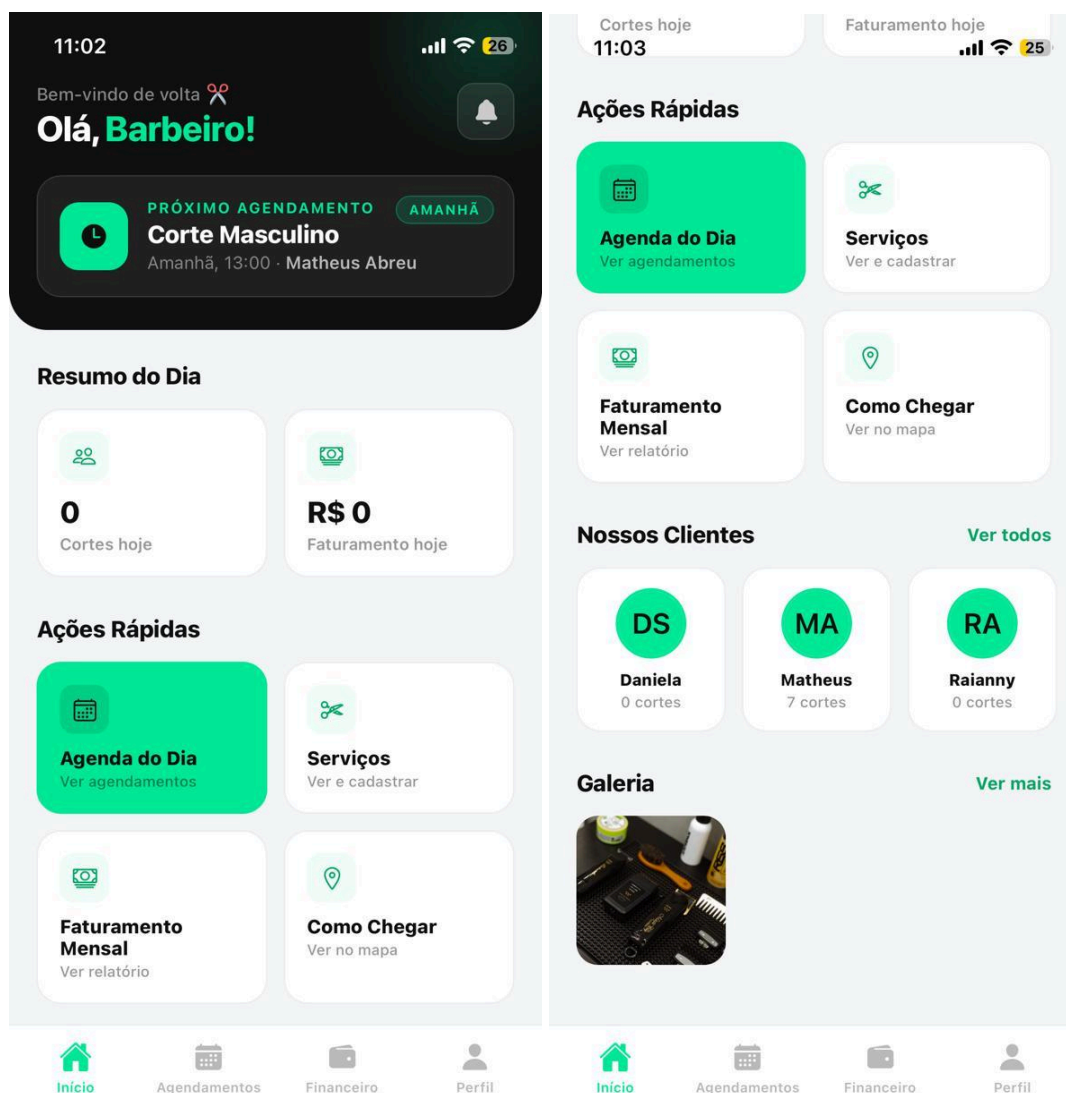
Figuras 07 - Tela de perfil (Cliente)



Fonte: Elaborado pelo autor (2026)

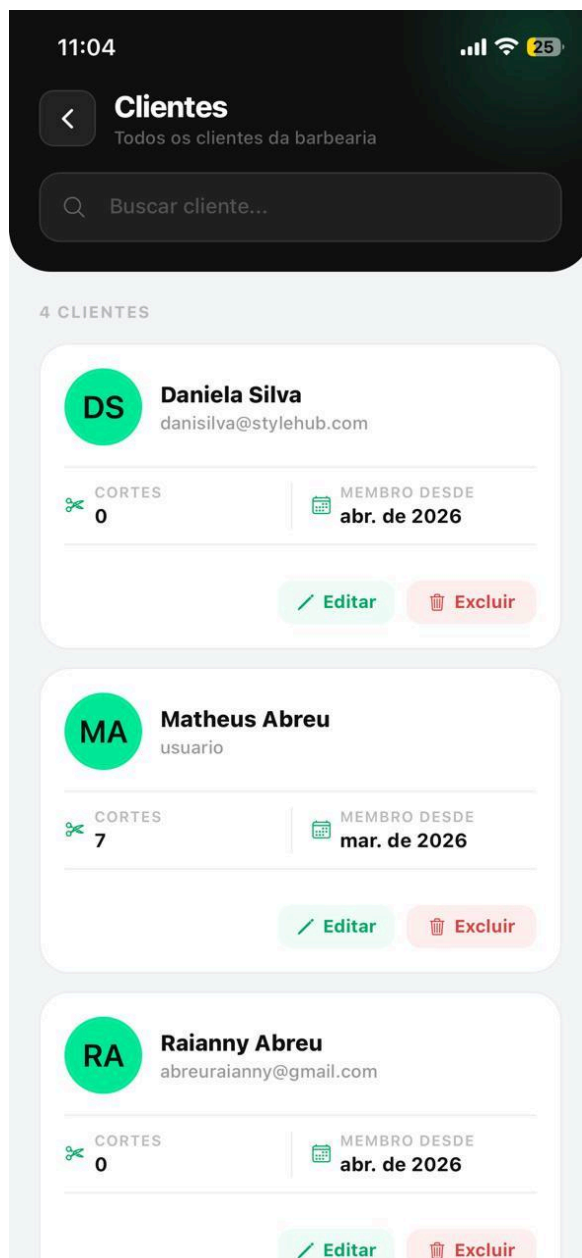
A figura 07 apresenta a tela de perfil do sistema StyleHub. A tela de perfil permite ao usuário visualizar suas informações cadastradas no sistema e alterá-las. Apresenta a jornada do usuário, o programa de fidelidade além de dar acesso às configurações e área de suporte.

Figuras 08 - Tela de início (Barbeiro)



Fonte: Elaborado pelo autor (2026)

A figura 08 apresenta respectivamente a tela de início do sistema StyleHub. A tela inicial do barbeiro apresenta um resumo das principais informações do sistema, como os próximos agendamentos, clientes atendidos e acesso rápido às funcionalidades de gestão, além de clientes cadastrados e galeria de imagens. Essa interface permite uma visão geral das atividades do dia, auxiliando na organização e no controle dos atendimentos.

Figura 09 - Tela de Clientes (Barbeiro)

Fonte: Elaborado pelo autor (2026)

A figura 09 apresenta a tela de clientes do sistema StyleHub. A tela de clientes permite ao usuário do tipo barbeiro visualizar a lista de clientes cadastrados no sistema, assim como acessar e gerenciar suas informações. Nessa tela, é possível realizar ações como edição e exclusão de dados, possibilitando maior controle e organização.

Figura 10 - Tela de agendamentos ou agenda (Barbeiro)



Fonte: Elaborado pelo autor (2026)

A figura 10 apresenta a Tela de agendamentos do sistema StyleHub. A tela de agendamentos permite ao usuário do tipo barbeiro visualizar todos os atendimentos agendados, apresentando informações como cliente, serviço, data e horário. Além disso, possibilita o gerenciamento dos agendamentos, permitindo ações como confirmação, conclusão ou cancelamento, e busca por filtros inteligentes contribuindo para a organização e controle da agenda.

Figura 11 - Tela de financeiro (Barbeiro)

Fonte: Elaborado pelo autor (2026)

A figura 11 apresenta a tela de financeiro do sistema StyleHub. A tela financeira permite ao usuário do tipo barbeiro acompanhar informações relacionadas ao faturamento e às despesas do sistema. Nessa interface, é possível visualizar os valores obtidos com os serviços realizados, e também registrar e acompanhar gastos, contribuindo para uma melhor organização e controle financeiro do estabelecimento.

Figura 12 - Tela de Perfil (Barbeiro)

Fonte: Elaborado pelo autor (2026)

A figura 12 apresenta a Tela de perfil do sistema StyleHub. A tela de perfil permite ao usuário do tipo barbeiro visualizar e gerenciar suas informações cadastradas no sistema, bem como sua jornada de uso do app, apresentação e definição de meta mensal, além de mais funções de acesso.

Figura 13 - Tela de configurações (Barbeiro)

Fonte: Elaborado pelo autor (2026)

A figura 13 apresenta a tela de configurações do sistema StyleHub. A tela de configurações permite ao usuário do tipo barbeiro ajustar preferências do sistema, como definição de horários disponíveis para atendimento e configuração da localização do estabelecimento. Além das funções já mencionadas anteriormente, a interface dá acesso ao gerenciamento de cupons.

4.6.2 Testes com usuários

Após a implementação das funcionalidades principais, foram realizados testes com usuários reais. Inicialmente, os primeiros testes foram feitos com um colega de trabalho, utilizando uma versão inicial do sistema com foco no perfil de cliente.

Posteriormente, com o sistema em estágio mais avançado, “produto mínimo viável” (MVP), os testes foram realizados em ambiente real de barbearia, envolvendo um barbeiro e clientes do estabelecimento. Essa etapa permitiu avaliar o sistema em um lugar real de uso.

Apresentei ao final deste trabalho, em anexo (apêndice B), algumas imagens dos testes realizados com os usuários do sistema, em ambientes controlados e uma barbearia local. As imagens mostram o sistema sendo usado por clientes em dispositivos androids e IOS, além de um barbeiro utilizando a aplicação em um notebook na versão web.

4.6.3 Resultados obtidos

Ao todo, participaram dos testes 8 usuários, sendo 3 barbeiro e 5 clientes. Os resultados foram bastante positivos, indicando uma boa aceitação do sistema. Os usuários relataram facilidade na utilização da aplicação, destacando que o sistema é intuitivo e de fácil entendimento. Nenhum dos participantes relatou grandes dificuldades durante o uso.

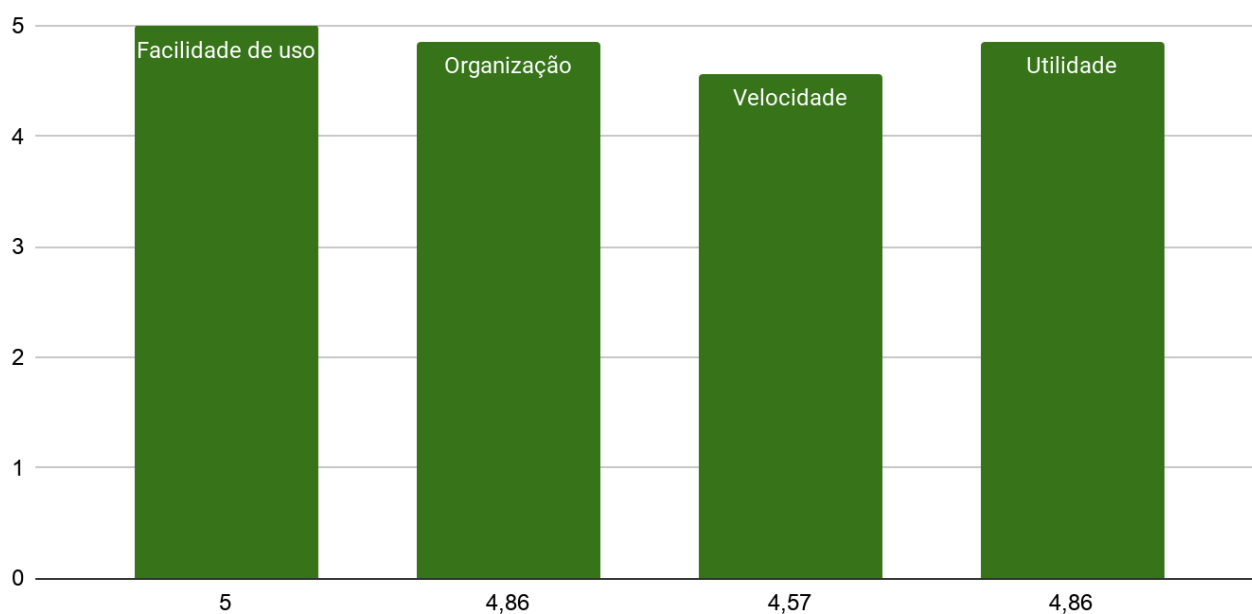
Em relação ao processo de agendamento, todos os usuários afirmaram que a funcionalidade é simples e clara, demonstrando segurança ao realizar as operações. As telas também foram avaliadas como fáceis de entender e usar, e o sistema apresentou bom desempenho. Sobre a utilização no dia a dia, todos os participantes afirmaram que utilizariam o sistema, destacando sua facilidade e utilidade.

Não foram sugeridas melhorias pelos participantes, além das que já tinha mencionado que aconteceria futuramente a eles, sendo destacado que o sistema já apresenta boa organização e funcionamento.

Os resultados observados na pesquisa de opinião podem ser comprovados no gráfico 01, que apresenta a média das notas atribuídas pelos usuários aos principais aspectos do sistema, com valores de uma à cinco estrelas.

Gráfico 01 - Avaliação dos usuários sobre usabilidade do sistema StyleHub.

Avaliações



Fonte: Elaborado pelo autor (2026).

5 CONCLUSÃO

O presente trabalho teve como objetivo o desenvolvimento de um sistema para gestão e agendamento de serviços voltado para barbearias e salões de beleza, denominado StyleHub. A aplicação foi projetada com foco na organização dos atendimentos, controle das informações e melhoria da experiência dos usuários, tanto clientes quanto barbeiros.

Durante o desenvolvimento, foram implementadas funcionalidades que contribuíram diretamente para a melhoria da experiência do cliente no agendamento, tornando o processo mais simples, rápido e intuitivo, evitando estresse desnecessários de filas, discussões e ambientes lotados. Além disso, a interface foi pensada para facilitar o uso, permitindo que os usuários realizem agendamentos de forma clara e eficiente em segundos.

Em relação à organização da gestão do empreendimento, o sistema possibilitou a centralização das informações, permitindo ao barbeiro gerenciar serviços, clientes e atendimentos em um único ambiente além de entrada e saídas de valores. Isso contribui para uma melhor visão geral do negócio, além de facilitar a tomada de decisões.

No que diz respeito ao controle de conflitos de horários, o sistema realiza o gerenciamento automatizado dos agendamentos, permitindo apenas um agendamento por horário diário e evitando sobreposição destes, além do controle manual de horários disponíveis para as datas de trabalho. De forma automática o sistema bloqueia dias que o salão não está aberto e ainda assim permite agendamentos para dias de trabalho da barbearia 24 horas por dia, garantindo maior captação de clientes e organização na agenda do profissional.

Quanto à gestão financeira, foram implementadas funcionalidades que permitem o acompanhamento de receitas e despesas, gerenciando todas as entradas por meio de serviços realizados e possibilitando o cadastro de saídas/despesas, além de mostrar de forma personalizada um balanço com filtros inteligentes auxiliando no controle financeiro e contribuindo para uma administração mais eficiente do estabelecimento.

Em relação à segurança das informações dos usuários, o sistema foi desenvolvido considerando a proteção dos dados durante o uso, por meio de validação de usuários e controle de acessos através de tokens e persistências de dados com um sistema de gerenciamento de banco, garantindo maior confiabilidade no armazenamento e no acesso às informações.

Embora haja diversas opções no mercado, poucos softwares atendem às particularidades das barbearias locais, especialmente em pequenos municípios, onde a simplicidade e a funcionalidade e menor custo são fatores essenciais para a adoção da tecnologia. A apresentação inicial do StyleHub aos usuários teste revelou que o sistema tem potencial para implementação e valor de mercado, destacando-se pela centralização de funções essenciais como controle financeiro, agendamento e comunicação com clientes.

Com base no projeto, fica clara a importância de desenvolver soluções que priorizem a usabilidade, com interfaces de fácil usabilidade e intuitivas, comandos simples e funcionalidades que realmente contribuam para o dia a dia dos profissionais e de seus clientes. Os Resultados obtidos servem como gás para o trabalho de futuras melhorias no StyleHub. Destaca-se a possibilidade de expansão das funcionalidades, aprimoramento da interface e implementação de novos recursos, que possam agregar ainda mais valor à aplicação.

Dessa forma, conclui-se que os objetivos propostos foram alcançados, sendo possível desenvolver um sistema funcional, organizado e bem aceito pelos usuários. Além disso, validou o uso dos conhecimentos obtidos durante todo o tempo de curso, ajudando também a desenvolver áreas ainda não exploradas no desenvolvimento de software.

Como trabalhos futuros, o foco da aplicação é virar uma plataforma completa para qualquer profissional do ramo de beleza e estética, proporcionando não só uma boa gestão, mas de fato uma experiência melhor, além de buscar uma implementação com o aplicativo popular de mensagem WhatsApp. Agindo como uma ponte de conexão entre grandes empresas do ramos de vendas de produtos, até pequenos negócios do ramo de serviços, além disso, fidelizando cada vez mais clientes aos bons profissionais da área. Essas e mais melhorias deverão focar em

oferecer uma experiência mais fluida, eficiente e alinhada às expectativas dos usuários.

REFERÊNCIAS

- AMAZON WEB SERVICES. **O que é uma API RESTful?**. [S. l.: s. n.], [2026]. Disponível em: <https://aws.amazon.com/pt/what-is/restful-api/>. Acesso em: 07 mar. 2026.
- AMAZON WEB SERVICES. **O que é o JavaScript (JS)?**. [S. l.: s. n.], [s.d.]. Disponível em: <https://aws.amazon.com/pt/what-is/javascript/>. Acesso em: 25 fev. 2026.
- ANDROID STUDIO. **Conheça o Android Studio**. [S. l.: s. n.], [s.d.]. Disponível em: <https://developer.android.com/studio/intro?hl=pt-br>. Acesso em: 17 mai. 2026.
- AXIOS. **Características**. [S. l.: s. n.], [s.d.]. Disponível em: <https://axios.rest/pages/getting-started/features.html>. Acesso em: 05 jan. 2026.
- BETRYBE. **Framework: o que é, como ele funciona e para que serve?**. [S. l.: s. n.], 2021. Disponível em: <https://blog.betrybe.com/framework-de-programacao/o-que-e-framework/>. Acesso em: 08 mar. 2026.
- BOCARD, Taysa. **O que são aplicativos? Definição da desenvolvedora Usemobile**. [S. l.: s. n.], 2023. Disponível em: <https://usemobile.com.br/aplicativo-movel/>. Acesso em: 21 abr. 2026.
- BRUNO. **O que é WordPress, como funciona e como usar: guia para iniciantes**. [S. l.: s. n.], 2025. Disponível em: <https://www.hostinger.com/br/tutoriais/o-que-e-wordpress>. Acesso em: 04 mai. 2026.
- CATALDO, Diego. **Quer criar um app? Descubra como funciona o Android Studio**. [S. l.: s. n.], 2025. Disponível em: <https://www.techtudo.com.br/guia/2025/10/quer-criar-um-app-descubra-como-funciona-o-android-studio-edapps.gh.html>. Acesso em: 17 mai. 2026.
- CARLOS. **O que é PHP e para que serve? Guia simples e completo**. [S. l.: s. n.], 2025. Disponível em: <https://www.hostinger.com/br/tutoriais/o-que-e-php-guia-basico>. Acesso em: 25 abr. 2026.
- CHAVES, Matheus. **Como e por que usar o axios?**. [S. l.: s. n.], 2023. Disponível em: <https://dev.to/matheushchaves/como-e-por-que-usar-o-axios-44gh>. Acesso em: 05 jan. 2026.
- CORTHY. **Por que você precisa de um app para barbeiros para profissionalizar sua barbearia**. [S. l.: s. n.], 2026. Disponível em: <https://corthy.com/app-para-barbeiros-profissionalizar-barbearia/>. Acesso em: 15 mar. 2026.
- ERICKSON, Jeffrey. **MySQL: Entendendo o que é e como é usado**. [S. l.: s. n.], 2024. Disponível em: <https://www.oracle.com/br/mysql/what-is-mysql/>. Acesso em: 05 abr. 2026.
- FRIZZAR. **Por Que a Gestão Efetiva é Fundamental para Barbearias de Sucesso**. [S. l.: s. n.], 2025. Disponível em: <https://backup.frizzar.com.br/blog/por-que-investir-em-gestao-para-barbearias-e-essencial/>. Acesso em: 26 fev. 2026.

GATE. **o que significa uma biblioteca em programação de computadores.** [S. l.: s. n.], [2013-2026]. Disponível em: <https://www.gate.com/pt-br/learn/glossary/what-is-a-library-in-computer-programming>. Acesso em: 01 abr. 2026.

IBM. **O que é API de REST?** [S. l.: s. n.], [2026]. Disponível em: <https://www.ibm.com/br-pt/think/topics/rest-apis>. Acesso em: 03 abr. 2026.

IONIC. **Introdução ao Ionic.** [S. l.: s. n.], [2026]. Disponível em: <https://ionicframework.com/docs>. Acesso em: 12 dez. 2025.

LARAVEL. **Instalação.** [S. l.: s. n.], [2026]. Disponível em: <https://laravel.com/docs/13.x>. Acesso em: 10 nov. 2025.

LE MOS II, Dalton. **Tecnologia da Informação.** 2. ed. Florianópolis: Publicação do IF-SC, 2011.

LUCIDCHART. **O que é um diagrama UML?** [S. l.: s. n.], [2026]. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-uml>. Acesso em: 26 abr. 2026.

MOZILLA. **Trabalhando com JSON.** [S. l.: s. n.], [s.d.]. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn_web_development/Core/Scripting/JSON. Acesso em: 25 nov. 2025.

NOGUEIRA, Alexandre. **PHP: guia básico.** [S. l.: s. n.], 2025. Disponível em: <https://www.hostgator.com.br/blog/php-guia-basico/>. Acesso em: 20 jan. 2026.

ONLINE BOOKING SOFTWARE. **Os 5 maiores desafios na gestão de uma barbearia e como superá-los.** [S. l.: s. n.], 2024. Disponível em: <https://www.linkedin.com/pulse/os-5-maiores-desafios-na-gest%C3%A3o-de-uma-barbearia-e-como-super%C3%A1-los-02u5f/>. Acesso em: 06 jan. 2026.

ORACLE. **O que é um banco de dados.** [S. l.: s. n.], 2020. Disponível em: <https://www.oracle.com/br/database/what-is-database/>. Acesso em: 05 abr. 2026.

SCHONS, Luiz. **O que é MVC? — Conceitos Básicos.** [S. l.: s. n.], 2020. Disponível em: <https://medium.com/@sschonss/o-que-%C3%A9-mvc-conceitos-b%C3%A1sicos-6363f9662f8c>. Acesso em: 12 mai. 2026.

SILVESTRE, Roberto. **O impacto da tecnologia na sociedade atual.** [S. l.: s. n.], 2024. Disponível em: <https://www.portaldequalidade.com.br/impacto-da-tecnologia-na-sociedade-atual/>. Acesso em: 25 abr. 2026.

VALENTE, Marco. **Engenharia de software moderna: Princípios e práticas para desenvolvimento de software com produtividade.** 1. ed. Pampulha: Independente, 2019.

VIDAL, Sebastião. **O que é uma linguagem de programação.** [S. l.: s. n.], 2023. Disponível em: <https://tecnobits.com/pt/O-que-%C3%A9-uma-linguagem-de-programa%C3%A7%C3%A3o/>. Acesso em: 18 abr. 2026.

VUE JS. **Introdução.** [S. l.: s. n.], [s.d.]. Disponível em: <https://vuejs.org/guide/introduction.html>. Acesso em: 05 jan. 2026.

XAVIER, Laena. **Sistema de Gestão para Barbearias: O Guia Definitivo para Transformar seu Negócio.** [S. l.: s. n.], 2026. Disponível em: <https://barbercode.com.br/sistema-de-gestao-para-barbearias-guia/>. Acesso em: 26 fev. 2026.

APÊNDICES

Apêndice A - Questionário de opinião aplicado durante os testes do sistema

- **CLIENTES (usuários comuns)**

- Você conseguiu entender como usar o sistema facilmente?
- Teve dificuldade em alguma parte? Qual?
- Foi fácil realizar um agendamento?
- O processo de escolha de horário foi claro?
- Você sentiu segurança ao confirmar o agendamento?
- As telas são fáceis de entender?
- O sistema é rápido e responsivo?
- Você utilizaria esse sistema no dia a dia?

- **BARBEIROS**

- O sistema ajuda na organização dos agendamentos?
- Você acha útil para o dia a dia da barbearia?
- Foi fácil gerenciar serviços e clientes?
- O controle financeiro é útil?
- Teve dificuldade em alguma funcionalidade?
- As informações são claras e bem organizadas?
- Você usaria esse sistema na sua barbearia?
- O que você adicionaria ou melhoraria?

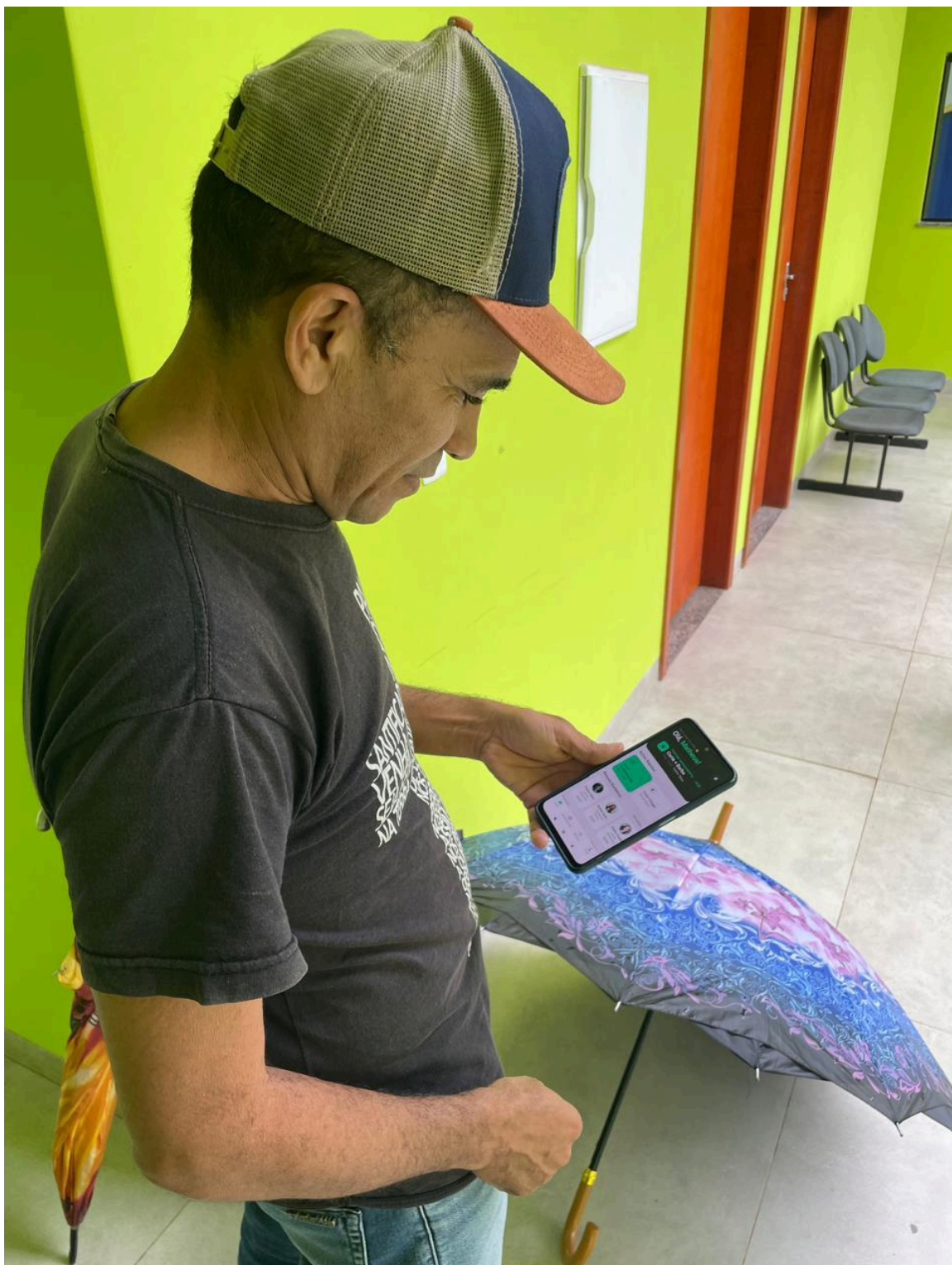
- **BARBEIRO/CLIENTE (Perguntas de notas)**

- Facilidade de uso
- Organização
- Velocidade
- Utilidade

Fonte: Elaborado pelo autor (2026).

Apêndice B - Registros fotográficos dos testes do sistema

Figura B1 - Teste das funcionalidades com usuário do tipo cliente.



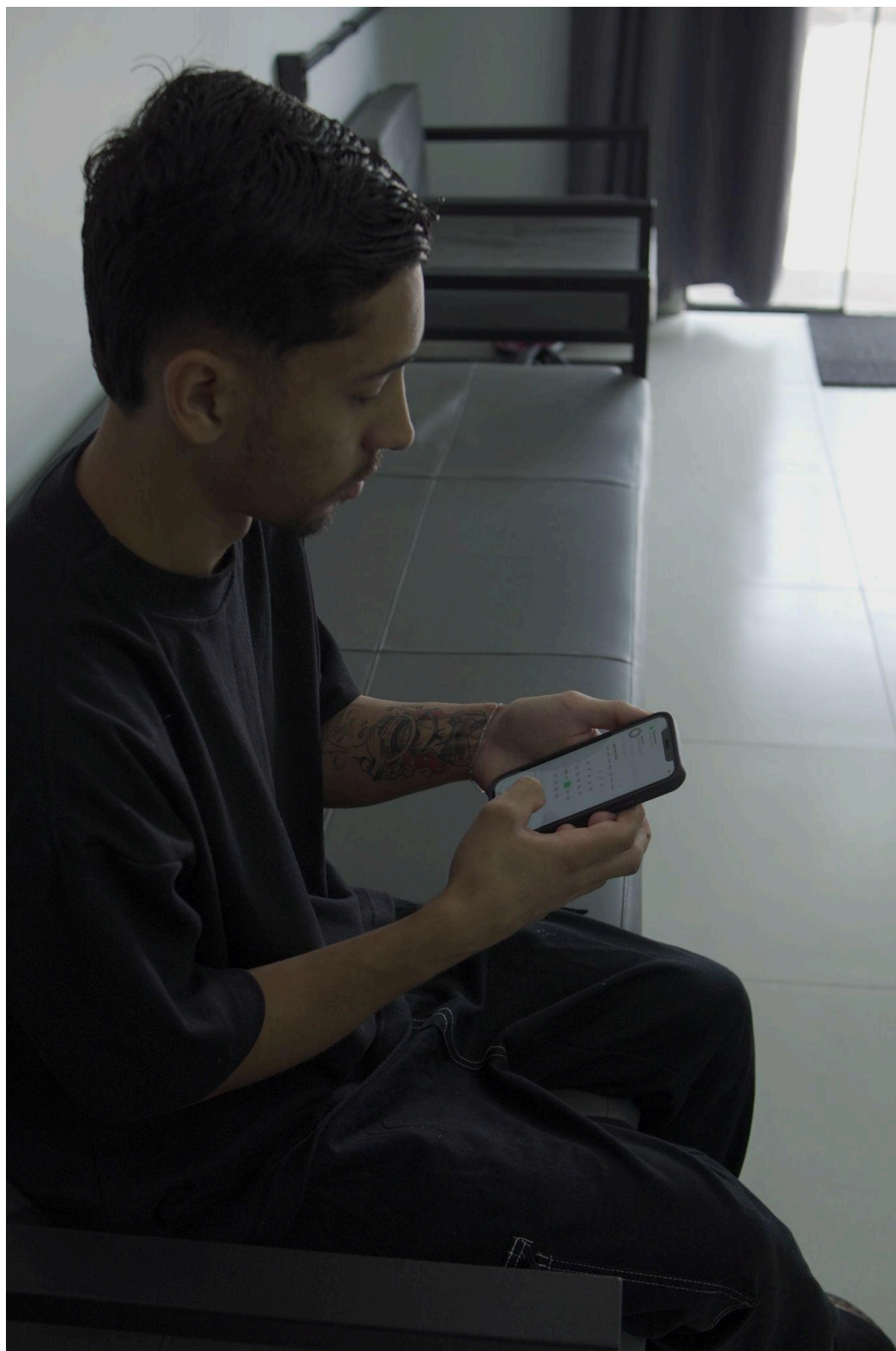
Fonte: Elaborado pelo autor (2026).

Figura B2 - Teste das funcionalidades com usuário do tipo cliente.



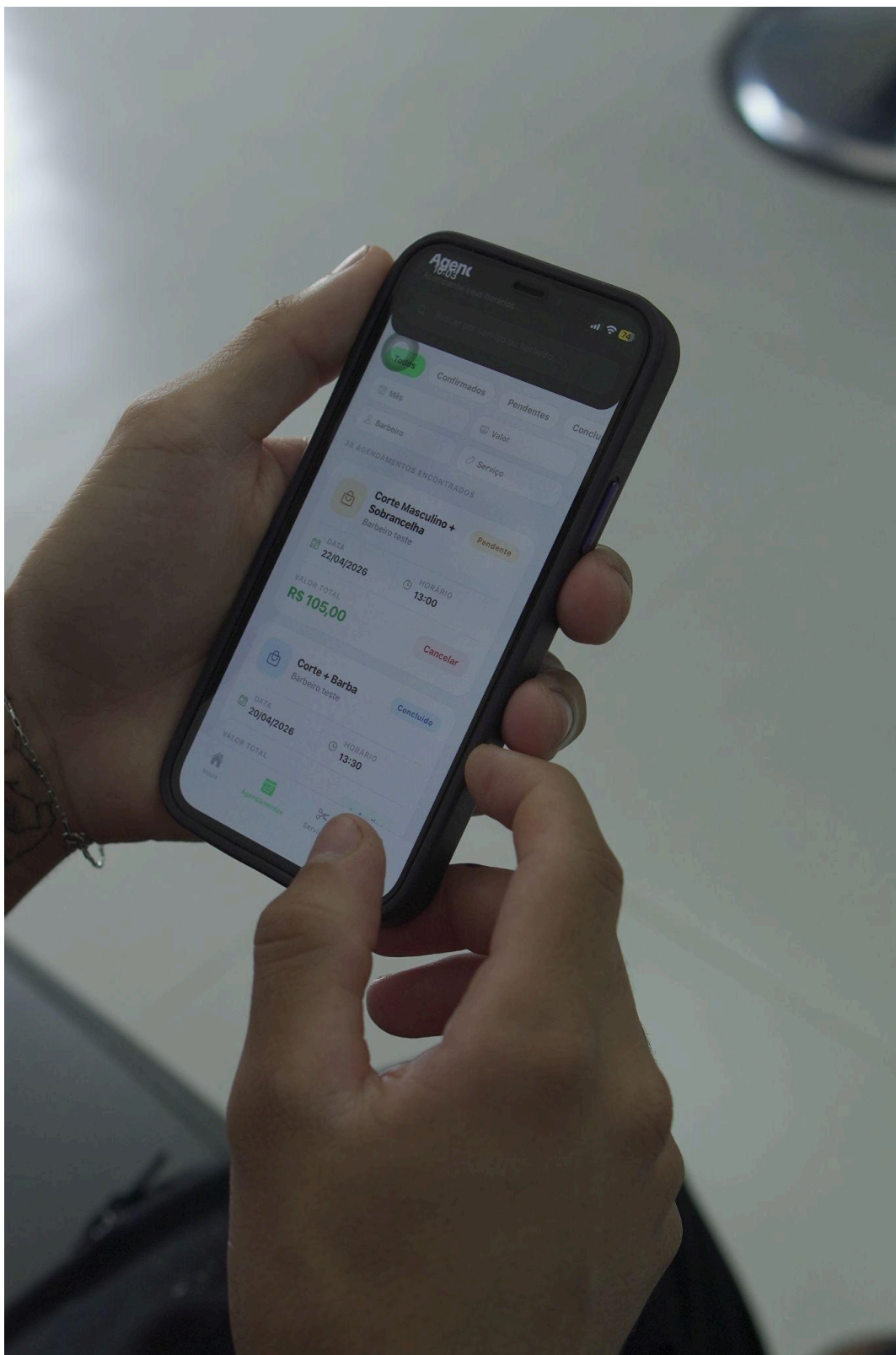
Fonte: Elaborado pelo autor (2026).

Figura B3 - Teste do sistema realizado com usuário em ambiente de barbearia.



Fonte: Elaborado pelo autor (2026).

Figura B4 - Teste do sistema realizado com usuário em ambiente de barbearia.



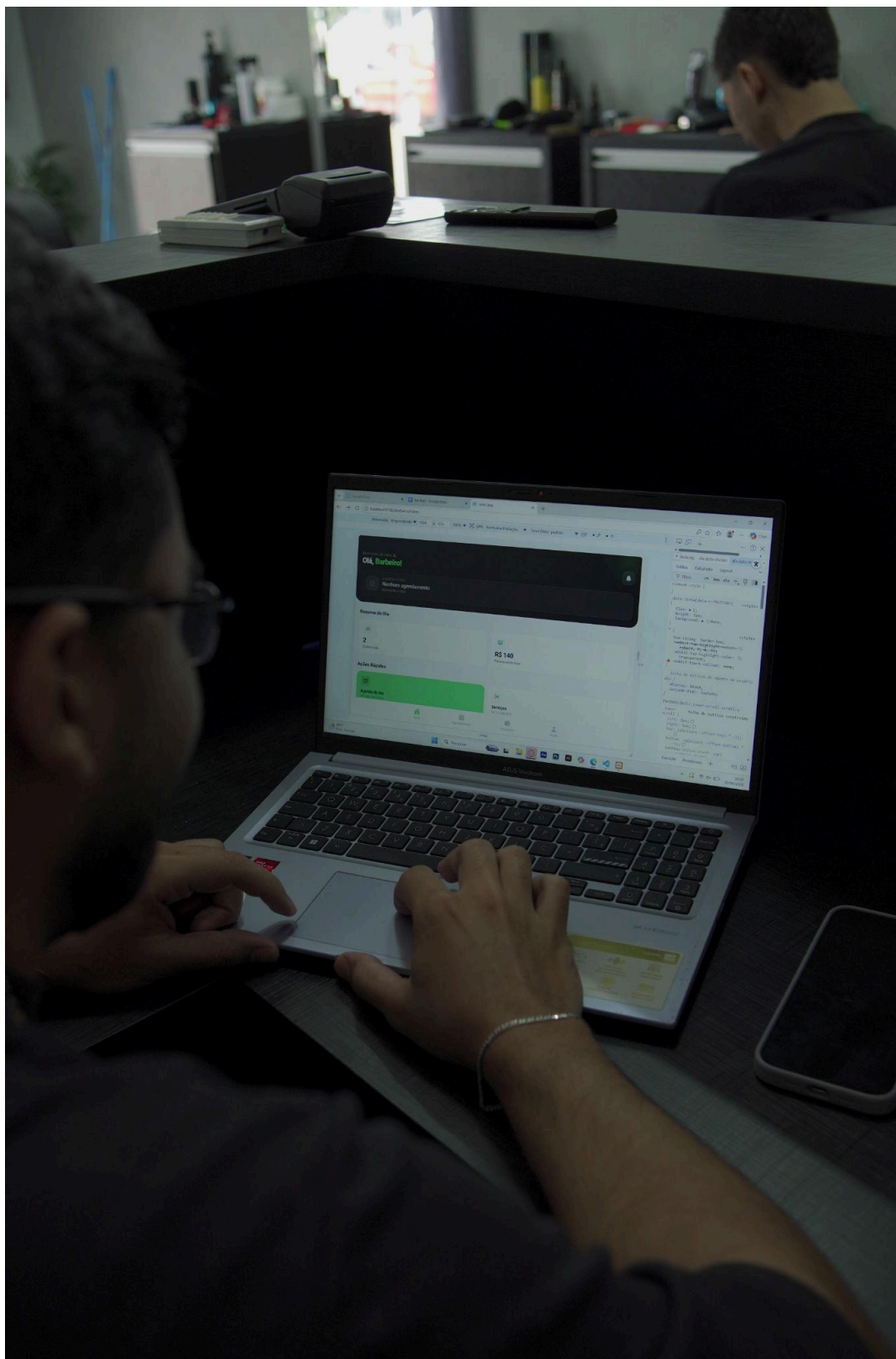
Fonte: Elaborado pelo autor (2026).

Figura B5 - Teste das funcionalidades com usuário do tipo barbeiro.



Fonte: Elaborado pelo autor (2026).

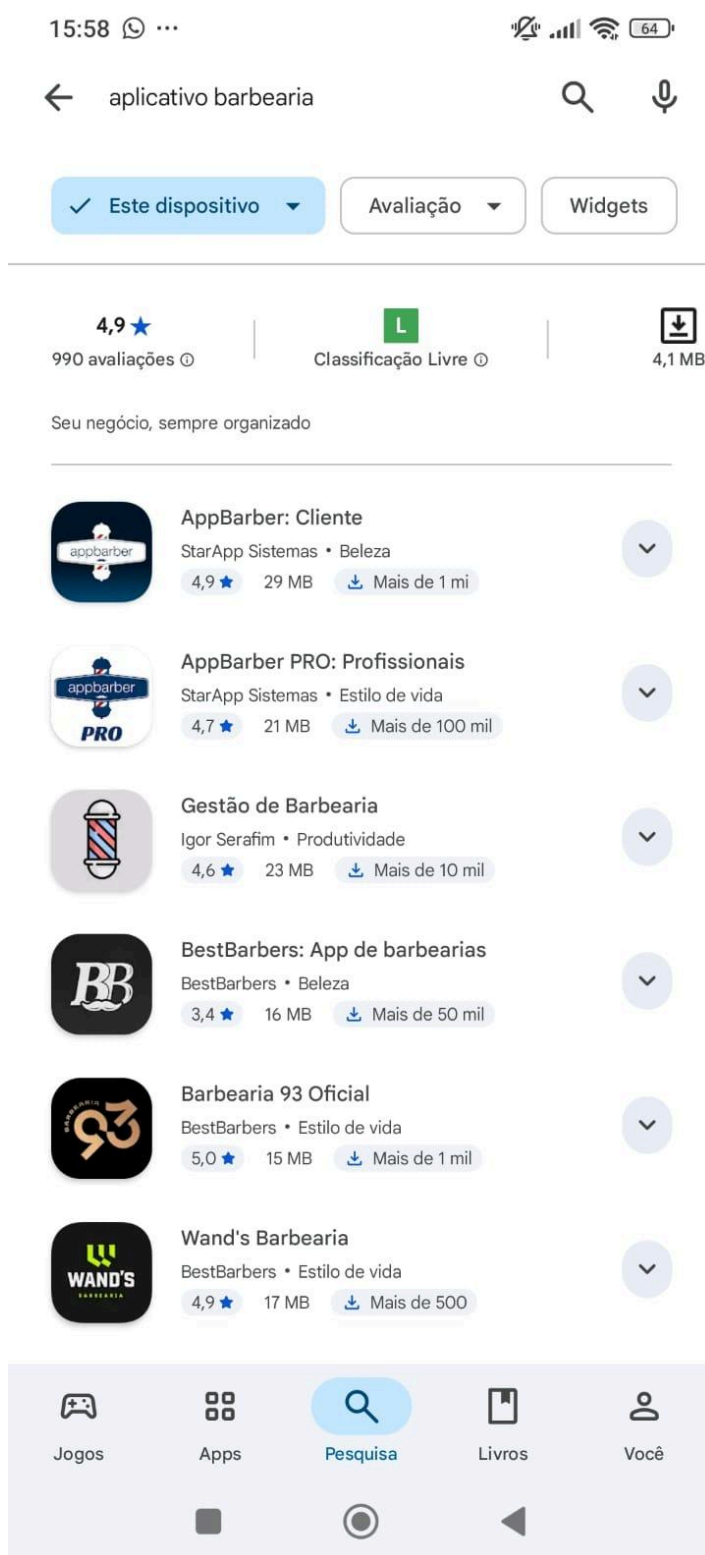
Figura B6 - Teste das funcionalidades com usuário do tipo barbeiro.



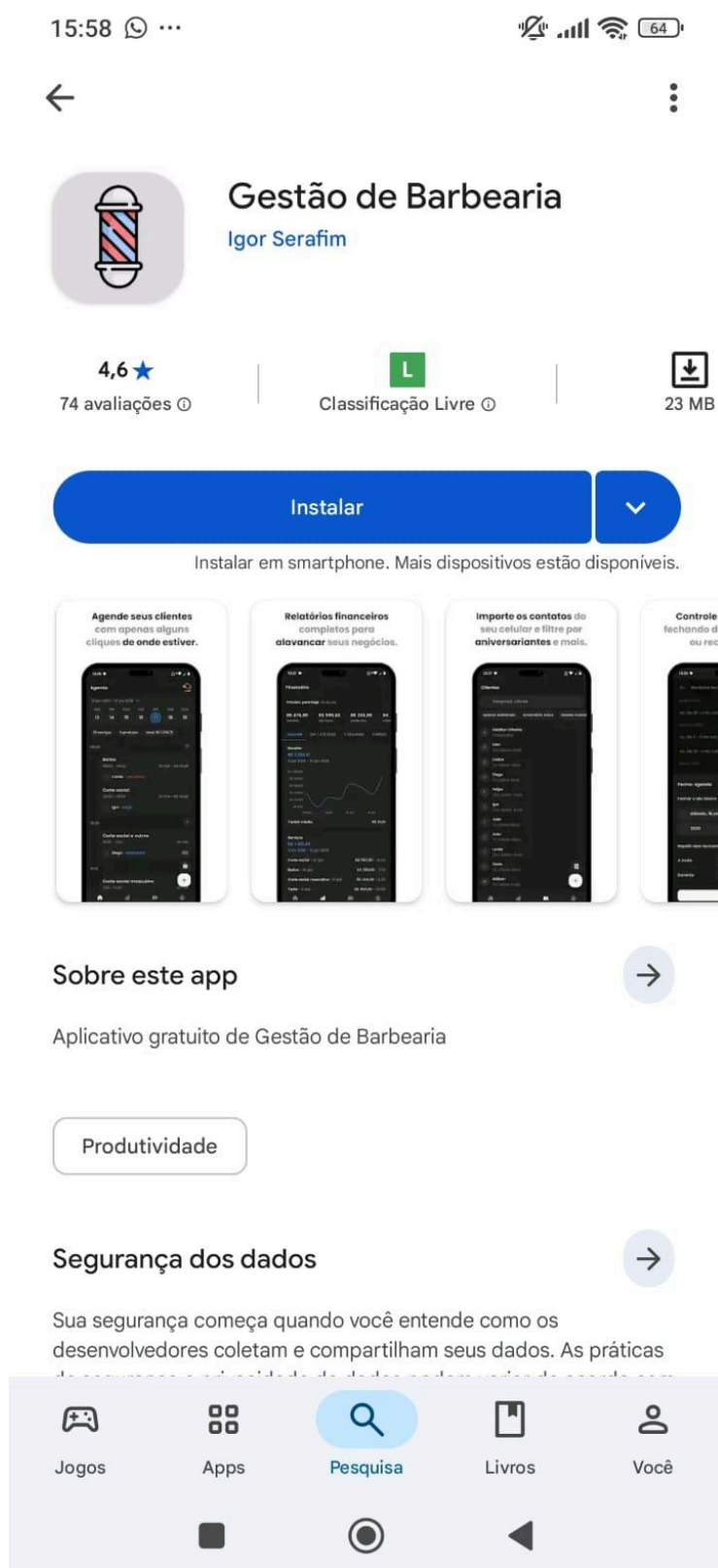
Fonte: Elaborado pelo autor (2026).

Apêndice C - Registros da pesquisa de aplicativos similares

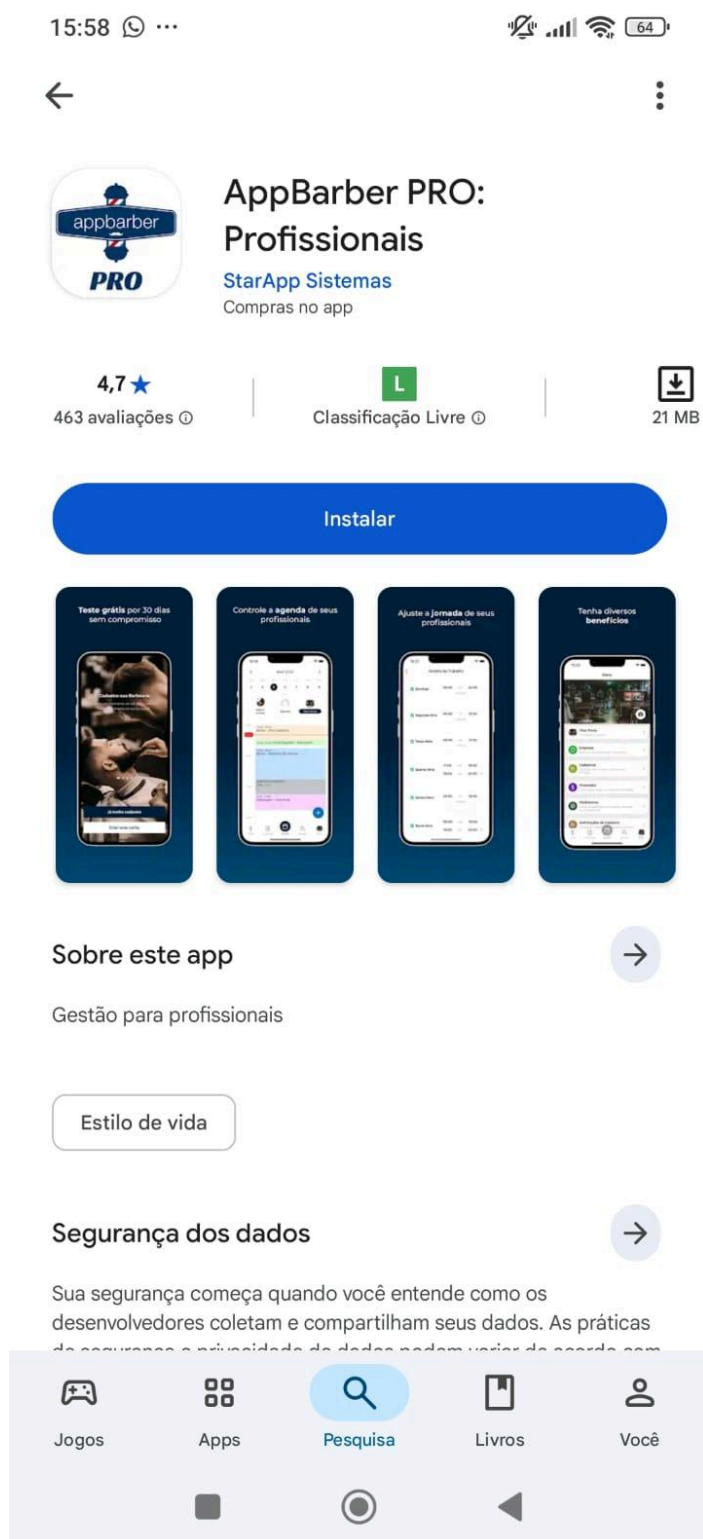
Figura C1 - Registro da loja de aplicativo (Android)



Fonte: Elaborado pelo autor (2026).

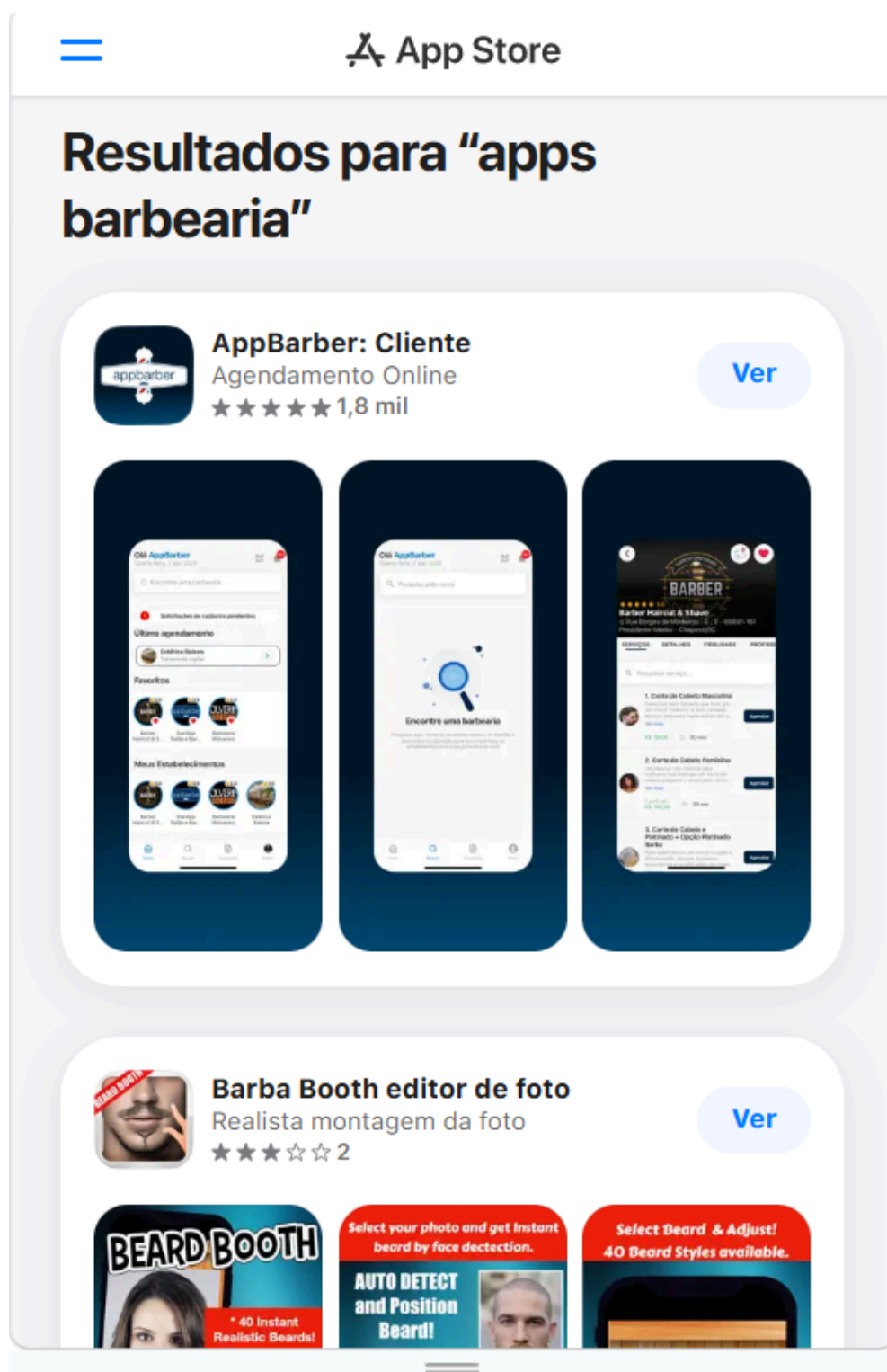
Figura C2 - Registro da loja de aplicativo (Android)

Fonte: Elaborado pelo autor (2026).

Figura C3 - Registro da loja de aplicativo (Android)

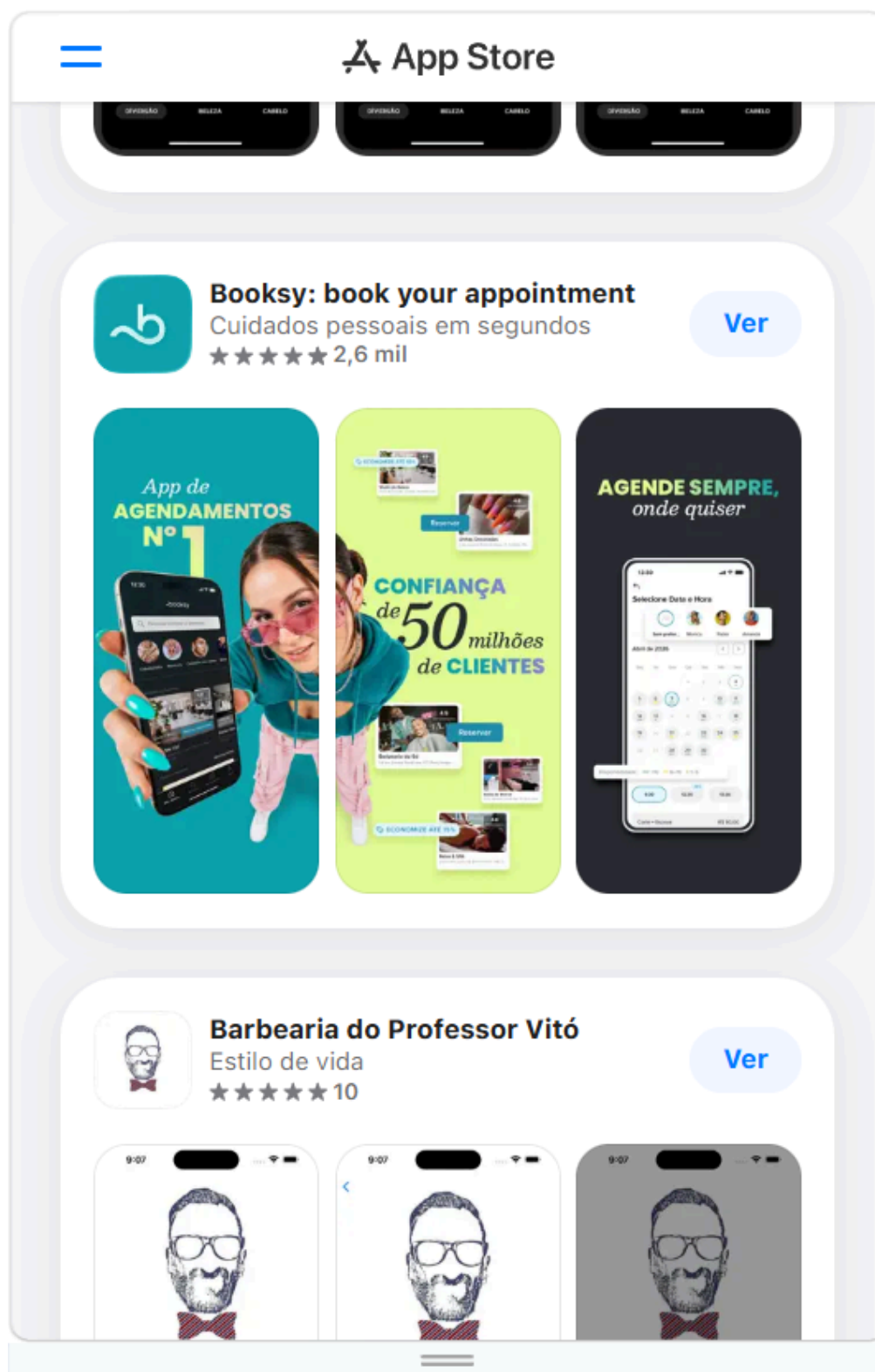
Fonte: Elaborado pelo autor (2026).

Figura C4 - Registro da loja de aplicativo (IOS)



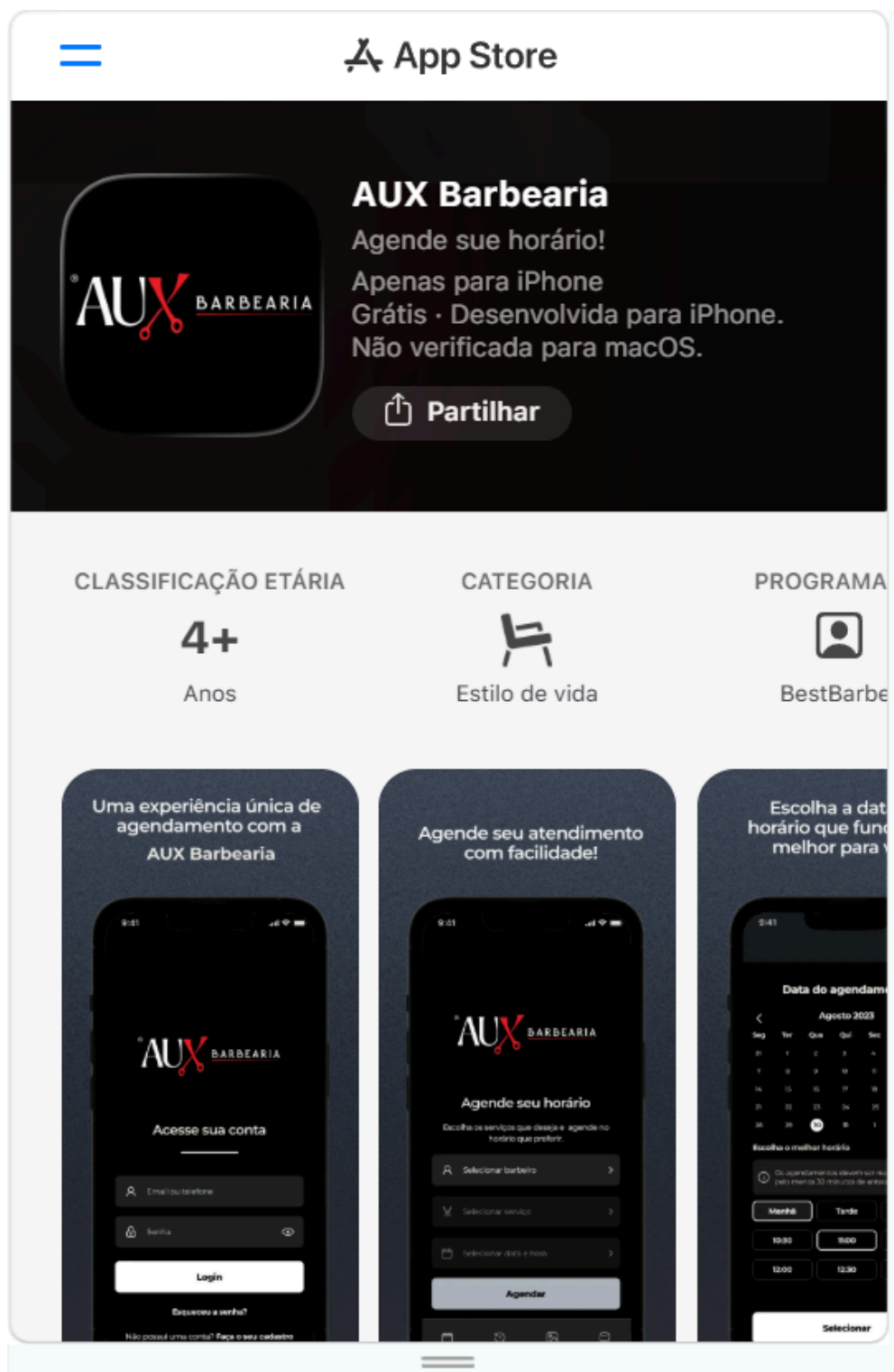
Fonte: Elaborado pelo autor (2026).

Figura C5 - Registro da loja de aplicativo (IOS)



Fonte: Elaborado pelo autor (2026).

Figura C6 - Registro da loja de aplicativo (IOS)



Fonte: Elaborado pelo autor (2026).

Apêndice D - Trechos de código do sistema

A seguir apresento trechos de codificação de uma das partes mais importantes do sistema, o agendamento de um horário dentro da aplicação, dividido em 6 figuras. As figuras vão desde a configuração do axios no frontend para o envio de requisições autorizadas, até a persistência dos dados apresentados no sistema de gerenciamento de banco de dados - MySQL.

Figura D1 - Configuração do Axios (Frontend)

```
import axios from 'axios';

const api = axios.create({
  baseURL: 'http://10.0.0.33:8080/api',
});

api.interceptors.request.use((config) => {
  const token = localStorage.getItem('token');

  if (token) {
    config.headers.Authorization = `Bearer ${token}`;
  }

  return config;
});

export default api;
```

Fonte: Elaborado pelo autor (2026).

Figura D2 - Função “confirmarAgendamento” (Frontend)

```
const confirmarAgendamento = async () => {
  if (!minhaData.value) return alertErro('Selecione uma data.')
  if (form.value.opcoes.length === 0) return alertErro('Selecione pelo menos um serviço.')
  if (!barbeiroSelecionado.value) return alertErro('Selecione um barbeiro.')
  if (!horarioSelecionado.value) return alertErro('Selecione um horário.')

  loading.value = true

  try {
    await api.post('/agendamentos', {
      barbeiro_id: barbeiroSelecionado.value.id,
      data:        minhaData.value.split('T')[0],
      hora:        horarioSelecionado.value,
      status:      'pendente',
      valor_total: total.value,
      cupom_id:    null,
      servicos:    form.value.opcoes,
    })
  }
}
```

Fonte: Elaborado pelo autor (2026).

Figura D3 - Rota da API (Backend)

```
// Rotas de Agendamento
$router->get('/agendamentos', 'AgendamentoControle@index');
$router->get('/agendamentos/{id}', 'AgendamentoControle@show');
$router->post('/agendamentos', 'AgendamentoControle@store');
$router->put('/agendamentos/{id}', 'AgendamentoControle@update');
$router->delete('/agendamentos/{id}', 'AgendamentoControle@destroy');
//
$router->get('/horarios-disponiveis', 'AgendamentoControle@horariosDisponiveis');
$router->get('/meus-agendamentos', 'AgendamentoControle@meusAgendamentos');
$router->get('/meus-agendamentos-barbeiro', 'AgendamentoControle@meusAgendamentosBarbeiro');
```

Fonte: Elaborado pelo autor (2026).

Figura D4 - Controller store (Backend)

```
public function store(Request $request)
{
    // Valida se o horário ainda está disponível antes de salvar
    $conflito = Agendamento::where('data', $request->data)
        ->where('barbeiro_id', $request->barbeiro_id)
        ->where('hora', $request->hora)
        ->whereNotIn('status', ['cancelado'])
        ->first();

    if ($conflito) {
        return $this->error('Este horário já está ocupado. Escolha outro.', '', 409);
    }

    $agendamento = Agendamento::create([
        'cliente_id' => auth()->user()->id,
        'barbeiro_id' => $request->barbeiro_id,
        'data' => $request->data,
        'hora' => $request->hora,
        'status' => $request->status,
        'valor_total' => $request->valor_total,
        'cupom_id' => $request->cupom_id
    ]);

    foreach ($request->servicos as $servicoId) {
        \DB::table('agendamento_servicos')->insert([
            'agendamento_id' => $agendamento->id,
            'servico_id' => $servicoId
        ]);
    }

    return $this->sucesso('Agendamento criado com sucesso!', $agendamento, 200);
}
```

Fonte: Elaborado pelo autor (2026).

Figura D5 - Model Agendamento (Backend)

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Model;
use App\Models\Usuario;
use App\Models\Servico;

class Agendamento extends Model
{
    protected $table = 'agendamentos';

    protected $fillable = [
        'cliente_id',
        'barbeiro_id',
        'data',
        'hora',
        'status',
        'valor_total',
        'cupom_id'
    ];

    public $timestamps = true;
}
```

Fonte: Elaborado pelo autor (2026).

Figura D6 - Banco de dados (MySQL)

1 • `SELECT * FROM stylehub.agendamentos;`

id	cliente_id	barbeiro_id	data	hora	status	valor_total	cupom_id	created_at	updated_at
27	5	2	2026-03-30	11:30:00	concluido	60.00	60.00	2026-03-29 21:57:55	2026-03-30 15:18:56
28	5	2	2026-03-30	13:30:00	concluido	90.00	NULL	2026-03-29 22:52:54	2026-03-29 22:53:07
29	5	2	2026-03-30	16:30:00	cancelado	25.00	NULL	2026-03-30 16:13:39	2026-04-05 22:33:07
30	5	2	2026-04-14	14:00:00	concluido	90.00	NULL	2026-03-31 20:57:22	2026-03-31 20:59:22
31	5	2	2026-04-14	08:30:00	cancelado	65.00	NULL	2026-03-31 21:02:12	2026-04-02 00:23:45
32	5	2	2026-04-01	10:00:00	cancelado	90.00	NULL	2026-03-31 21:31:59	2026-04-05 22:33:00
33	5	2	2026-04-03	17:00:00	concluido	25.00	NULL	2026-04-02 00:36:21	2026-04-20 10:32:05
34	5	2	2026-04-03	13:00:00	cancelado	25.00	NULL	2026-04-02 00:42:02	2026-04-05 22:33:02
35	5	2	2026-04-03	13:30:00	cancelado	50.00	NULL	2026-04-02 00:42:32	2026-04-05 22:32:57
36	5	2	2026-04-03	14:00:00	cancelado	22.50	NULL	2026-04-02 01:33:58	2026-04-05 22:32:53
37	5	2	2026-04-06	13:00:00	concluido	81.00	NULL	2026-04-05 22:32:28	2026-04-05 22:38:09
38	10	2	2026-04-14	13:30:00	concluido	140.00	NULL	2026-04-10 20:33:46	2026-04-20 10:32:04
39	5	2	2026-04-20	13:00:00	concluido	90.00	NULL	2026-04-19 10:55:02	2026-04-20 09:21:31
40	5	2	2026-04-20	13:30:00	concluido	50.00	NULL	2026-04-19 10:55:48	2026-04-20 09:21:28
41	5	2	2026-04-22	13:00:00	concluido	105.00	NULL	2026-04-20 10:03:12	2026-04-20 10:04:26
42	5	2	2026-04-20	14:00:00	concluido	15.00	NULL	2026-04-20 10:28:08	2026-04-20 10:32:01
43	5	2	2026-04-28	13:00:00	cancelado	90.00	NULL	2026-04-27 22:14:35	2026-04-27 22:16:23
44	5	2	2026-04-28	16:30:00	pendente	25.00	NULL	2026-04-28 15:27:38	2026-04-28 15:27:38
45	5	2	2026-04-28	17:00:00	pendente	25.00	NULL	2026-04-28 15:34:48	2026-04-28 15:34:48

Fonte: Elaborado pelo autor (2026).