

BS Beauty Academy

Projeto Integrado de Extensão I

São Paulo, SP - Brasil

2025

BS Beauty Academy

Projeto Integrado de Extensão I

Este projeto integrado de extensão, desenvolvido como parte do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, tem como objetivo desenvolver uma aplicação web para otimizar a gestão e o agendamento de serviços de estética em um ambiente coworking. A área de concentração do projeto é a inovação tecnológica aplicada ao setor de serviços de beleza.

Instituto Federal de Educação, Ciência e
Tecnologia de São Paulo – IFSP

Tecnologia em Análise e Desenvolvimento de Sistemas

Programa de graduação

Orientador: Marcelo Tavares

São Paulo, SP - Brasil

2025

BS Beauty Academy

Projeto Integrado de Extensão I/ BS Beauty Academy. – São Paulo, SP - Brasil, 2025

68p. : il. color; 30cm

Orientador: Marcelo Tavares

Projeto Integrado de Extensão – Instituto Federal de Educação, Ciência e Tecnologia de São Paulo – IFSP

Tecnologia em Análise e Desenvolvimento de Sistemas

Programa de graduação, 2025.

1. Graduação 2. Extensão 3. Integrado

I. Marcelo Tavares. II. IFSP. III. Análise e Desenvolvimento de Sistemas. IV. PIE1

Projeto Integrado de Extensão I

Este projeto integrado de extensão, desenvolvido como parte do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, tem como objetivo desenvolver uma aplicação web para otimizar a gestão e o agendamento de serviços de estética em um ambiente coworking. A área de concentração do projeto é a inovação tecnológica aplicada ao setor de serviços de beleza.

Trabalho aprovado. São Paulo, SP - Brasil, _____ de _____ de 2025:

Marcelo Tavares
Orientador

Professor
Convidado 1

Professor
Convidado 2

São Paulo, SP - Brasil
2025

Resumo

Este projeto integrado de extensão apresenta o desenvolvimento da aplicação *web* BS *Beauty*, com o objetivo de otimizar a gestão e o agendamento de serviços de beleza no ambiente *coworking*, sob responsabilidade da gestora, nossa parceira de extensão.

Como parte de uma iniciativa competitiva, a solução digital desenvolvida tem como propósito melhorar o atendimento e fidelizar clientes, reduzindo o tempo gasto em tarefas administrativas e garantindo uma interface de usuário intuitiva. Para isso, o sistema centraliza agendas, previne conflitos de horário e fornece notificações automáticas, relatórios financeiros e *dashboards* de desempenho.

A fim de atender as demandas da nossa parceira, a comunicação constante foi essencial para o levantamento de requisitos, análise de concorrentes e definição de regras de negócio. Para a gestão das etapas do projeto, foi adotado o framework ágil *Scrum*, formalizando o planejamento e controle de tarefas no *ProjectLibre*, um software de gestão de projetos que permite gerenciar cronogramas e alocar recursos para as tarefas definidas.

A arquitetura da aplicação foi idealizada em camadas, sendo detalhada em diagramas de componentes e de implantação. Paralelamente, elaborou-se o plano de testes, padronizou-se a documentação e avaliou-se a viabilidade financeira em cenários realistas, otimistas e pessimistas.

Como resultado, a aplicação desenvolvida fortalece conhecimentos teóricos do curso de graduação, aproxima-os das demandas de mercado e promove inovação tecnológica no setor de beleza apoiando o modelo de *coworking*.

Palavras-chave: aplicação *web*. agendamento online. *coworking* de beleza. gestão de serviços. *Scrum*.

abstract

This integrated extension project presents the development of the BS Beauty web application, aimed at optimizing the management and scheduling of beauty services in a coworking environment under the responsibility of the manager, our extension partner.

As part of a competitive initiative, the developed digital solution aims to improve service and foster customer loyalty by reducing the time spent on administrative tasks and ensuring an intuitive user interface. Therefore, the system centralizes schedules, prevents scheduling conflicts and provides automatic notifications, financial reports and performance dashboards.

In order to meet our partner's demands, constant communication was essential for requirements gathering, competitor analysis and definition of business rules. For project phase management, the agile framework Scrum was adopted, however the planning and task control was formalized in ProjectLibre, a project management software that allows managing schedules and allocating resources to tasks.

The application architecture was designed in layers, detailed in component and deployment diagrams. In parallel, the test plan was developed, documentation was standardized and financial viability was assessed in realistic, optimistic and pessimistic scenarios.

As a result, the developed application strengthens the theoretical knowledge of the undergraduate course, aligns it with market demands and promotes technological innovation in the beauty sector by supporting the coworking model.

Keywords: web application. online scheduling. beauty coworking. service management. Scrum.

Lista de ilustrações

Figura 1 – Profissionais da área da beleza no Brasil 2018–2022	17
Figura 2 – Distribuição dos profissionais da área da beleza 2018-2021	17
Figura 3 – Distribuição dos profissionais da área da beleza por local de trabalho 2018-2022	18
Figura 4 – Logo plataforma Trink's	19
Figura 5 – Logo plataforma Gendo	20
Figura 6 – Logo plataforma Avec	21
Figura 7 – módulos básicos em sistemas para salão de beleza	25
Figura 8 – QR Code do repositório da aplicação	30
Figura 9 – Página inicial do repositório	31
Figura 10 – Diagrama de componente da aplicação	44
Figura 11 – Diagrama de implantação da aplicação	46
Figura 12 – Diagrama Geral da Arquitetura	49
Figura 13 – Logo do React	51
Figura 14 – Logo do React	52
Figura 15 – Crescimento anual de custos com cibercrime	58

Lista de quadros

Quadro 1 – Comparação entre as plataformas concorrentes e a aplicação proposta	23
Quadro 2 – Membros e seus respectivos papéis	28
Quadro 3 – Membros e suas atividades	28
Quadro 4 – Papéis dos integrantes com base no Scrum	29

Lista de tabelas

Lista de abreviaturas e siglas

PIE1	Projeto Integrado de Extensão I
B2C	Business to Consumer
ECF	Emissor de Cupom Fiscal
ERP	Enterprise Resource Planning (Sistema Integrado de Gestão Empresarial)
iOS	iPhone Operating System
MEI	Microempreendedor Individual
NFC- e	Nota Fiscal de Consumidor Eletrônica
PDV	Ponto de Venda
Pix	Pagamento Instantâneo
SAT	Sistema Autenticador e Transmissor de Cupons Fiscais
SLA	Service Level Agreement (Acordo de Nível de Serviço)
SMS	Short Message Service
TEF	Transferência Eletrônica de Fundos
TADS	Tecnologia em Análise e Desenvolvimento de Sistemas
LGPD	Lei Geral de Proteção de Dados (Lei nº 13.709/2018)

Lista de símbolos

R\$	Real (moeda brasileira)
-----	-------------------------

%	Porcentagem
---	-------------

Sumário

1	INTRODUÇÃO	14
1.1	Objetivos	15
1.2	Problema e Solução Proposta	15
1.3	Justificativa	16
1.4	Análise da Concorrência	19
1.4.1	Trinks	19
1.4.2	Gendo	20
1.4.3	Avec	21
1.4.4	Quadro comparativo	22
2	REVISÃO DA LITERATURA	24
2.1	Histórico de sistemas gerenciadores de agendamento	24
2.2	Atualidade de sistemas gerenciadores de agendamento	25
3	GESTÃO DO PROJETO	27
3.1	Organização da Equipe	27
3.1.1	Responsabilidades / Papéis / Atividades	27
3.2	Metodologias de Gestão e Desenvolvimento	28
3.2.1	Scrum	28
3.2.1.1	Sprints	29
3.3	Repositório da Aplicação	29
3.3.1	Definição do repositório da aplicação	30
3.3.1.1	Link do repositório e especificações para acesso	30
4	DESENVOLVIMENTO DO PROJETO	32
4.1	Escopo do Projeto	32
4.1.1	Regras de Negócio	32
4.1.2	Requisitos Funcionais	34
4.1.3	Requisitos Não Funcionais	41
4.2	Histórias de Usuário	42
4.2.1	Descrição das Histórias de Usuário	42
4.3	Arquitetura	42
4.3.1	Definições da arquitetura	42
4.3.2	Diagrama da arquitetura	43
4.3.2.1	Diagrama de componentes	43
4.3.2.2	Diagrama de Implantação	45

4.3.2.3	Diagrama de Referência na AWS	48
4.4	Tecnologias e Ferramentas	50
4.4.1	Front-End	50
4.4.1.1	React	50
4.4.1.2	TailwindCSS	50
4.4.1.3	Redux e RTK Query	50
4.4.2	Back-End	50
4.4.2.1	NodeJS	50
4.4.2.2	Express	51
4.4.3	Infraestrutura	51
4.4.3.1	Docker	51
4.4.3.2	Amazon Web Services (AWS)	52
4.4.3.3	Banco de Dados MariaDB	52
4.4.4	Qualidade de Software e Testes	52
4.4.4.1	SonarQube	53
4.4.4.2	Vitest	53
4.5	Testes e Manutenibilidade	53
4.5.1	Plano de Testes	53
4.5.2	Análise Estáticas	54
4.5.3	Testes funcionais	54
4.5.3.1	Testes Unitários	54
4.5.3.2	Testes de Componente	55
4.5.3.3	Testes de Integração	55
4.5.3.4	Testes <i>end-to-end</i>	55
4.5.4	Testes não funcionais	55
4.5.4.1	Testes de performance	55
4.5.4.2	Testes de carga	56
4.5.4.3	Testes de configuração	56
4.5.5	Testes automatizados	56
4.5.6	<i>Logs</i>	56
4.5.7	<i>Code Convention</i>	56
4.6	Segurança, Privacidade e Legislação	57
4.6.1	Critérios de segurança e privacidade	58
4.6.1.1	Cadastro e login com conta Google	58
4.6.1.2	Infraestrutura de Rede	59
4.6.2	Observância à Legislação	59
4.7	Modelo de Banco de Dados	59
4.7.1	Modelo Entidade Relacionamento - MER	59
4.7.2	Diagrama Entidade Relacionamento - DER	59

4.7.3	Dicionário de Dados	59
4.8	Cronograma	59
4.8.1	Análise da Duração do Projeto	59
5	VIABILIDADE FINANCEIRA	60
5.1	Custos	60
5.2	Receitas	60
5.3	Cenário realista	60
5.4	Cenário Otimista	60
5.5	Cenário Pessimista	60
6	CONSIDERAÇÕES FINAIS	61
	REFERÊNCIAS BIBLIOGRÁFICAS	62
	APÊNDICES	65
	APÊNDICE A – EXEMPLO 1	66
	ANEXOS	67
	ANEXO A – EXEMPLO 1	68

1 Introdução

Segundo levantamento do Sebrae ([SEBRAE RS, 2024](#)), em 2024 o setor de beleza no Brasil movimentou aproximadamente US\$ 27 bilhões, colocando o país entre os cinco maiores mercados do mundo nesse ramo. Esse volume financeiro trouxe uma série de novidades e gerou, conseqüentemente, novas demandas. Diante de tantas mudanças e inovações, tornou-se indispensável que os empreendedores se adaptem rapidamente às tendências.

Conforme publicação do hub *Beauty Fair* (Maior evento no setor da beleza no Brasil), até pouco tempo os profissionais autônomos precisavam deslocar-se até a residência de seus clientes para atendê-los ou firmar parcerias prestando serviços dentro de estabelecimentos de terceiros. Com o surgimento dos coworkings de beleza, esse cenário vem se transformando. A própria *Beauty Fair* esclarece que um coworking de beleza é um espaço compartilhado que oferece infraestrutura para que profissionais da área possam trabalhar e colaborar. Trata-se de um local no qual cabeleireiros, maquiadores, esteticistas, manicures, massoterapeutas e demais especialistas podem alugar um posto de trabalho, dividir recursos e alcançar potenciais clientes ([Beauty Fair, 2024](#)).

Reportagem online na Gazeta do Povo destaca que o ambiente coworking vem se consolidando como um dos modelos de negócio que mais crescem no Brasil, oferecendo ao profissional autônomo flexibilidade, troca de experiências e uma infraestrutura completa sem burocracia nem custos inesperados ([Gazeta do Povo, 2023](#)). Nesse ambiente, o prestador de serviços tem o benefício de não precisar arcar com despesas de instalação ou manutenção de um salão próprio; basta utilizar o espaço, atender seus clientes e agendar a próxima sessão, preservando o controle sobre seus horários e ganhos.

À medida que esse formato de trabalho se expande, aumenta também a necessidade de maximizar a autonomia e a rentabilidade de cada profissional. Portanto, surge o desafio de gerir agendas, espaços e custos de forma ágil e intuitiva, evitando conflitos de reserva ou falhas de cobrança. Este projeto propõe-se a desenvolver uma aplicação web que atenda exatamente a essa demanda.

A aplicação web BS Beauty foi desenvolvida especialmente para gerenciar um salão de beleza que opera em modelo coworking, sob a gestão de nossa parceira de extensão Bruna. Seu objetivo principal é otimizar os processos internos e centralizar o agendamento de serviços, atendendo tanto às demandas da gestora quanto às necessidades dos profissionais autônomos.

1.1 Objetivos

A aplicação web BS Beauty foi desenvolvida especialmente para gerenciar um salão de beleza que opera em modelo coworking, sob a gestão de nossa parceira de extensão Bruna. Seu objetivo principal é otimizar os processos internos e centralizar o agendamento de serviços, atendendo tanto às demandas da gestora quanto às necessidades dos profissionais autônomos.

Para os Clientes Finais: A plataforma possibilita o agendamento de serviços de forma intuitiva e flexível. Os clientes poderão escolher profissionais específicos ou optar pelo melhor horário disponível, visualizando facilmente a lista de prestadores, seus serviços, preços, tempo de execução e agendas atualizadas.

Para os Profissionais Autônomos: O sistema BS Beauty tem como propósito reforçar a autonomia dos profissionais sobre sua agenda e finanças. A aplicação permite bloquear horários, editar preços e a duração dos serviços, além de acompanhar os agendamentos realizados (sejam eles do dia, futuros ou passados) e visualizar relatórios detalhados com a receita gerada pelos serviços prestados.

Para a Gestora: Nossa parceira, Bruna, terá acesso a funcionalidades exclusivas que incluem análise de métrica de desempenho (a partir de dashboards), gerenciamento do aluguel ou comissão de cada profissional, visualização do fluxo de agendamentos em períodos específicos, envio de mensagens de marketing e promoções aos clientes, e acesso a relatórios financeiros detalhados. Ademais, a gestora poderá incluir ou remover profissionais da plataforma conforme a necessidade.

1.2 Problema e Solução Proposta

A gestão de um salão por pequenos empreendedores é frequentemente desafiadora. Ademais, demandas surgem e muitas vezes são realizadas manualmente. Portanto, quando alguma etapa falha, evidencia-se a necessidade de uma solução digital capaz de reduzir erros e diminuir o esforço administrativo.

Por isso, o objetivo geral do projeto é suprir as necessidades de um salão de beleza em modelo coworking de forma ágil. Como explicado anteriormente, esse modelo de trabalho é recente (popularizado após a pandemia de COVID-19 em 2020) e atende diferentes profissionais autônomos (relacionados à gerente por locação ou comissão), não uma equipe com objetivo comum. Desta forma, o problema central é gerenciar a ocupação de cada profissional no espaço de trabalho, além de controlar as finanças e a agenda de clientes.

Nossa parceira Bruna já utilizava um sistema digital para gerenciamento do salão. Contudo, apesar dos benefícios trazidos pela solução, o sistema apresentava pontos insatis-

fatórios, sendo o principal deles a instabilidade da plataforma, que gerava insatisfação e perda de clientes.

Nossa solução consiste em criar uma aplicação web que mantenha todas as funcionalidades que já atendem bem a Bruna como o agendamento online e pesquisa de satisfação. Além disso, a plataforma incluirá funções ainda ausentes e ajustará requisitos funcionais e não funcionais cuja concepção é adequada, mas apresenta falhas, como o login instável, senhas excessivamente complexas e erros recorrentes na troca de senha.

Em síntese, a solução proposta é uma plataforma com login simplificado (integrado ao *Single Sign-On*¹ do Google) e agendamento fácil e transparente para os clientes (incluindo todos os serviços e atributos necessários para uma melhor decisão). Também contará com política de não comparecimento, agenda totalmente controlada pelos profissionais, notificações de agendamento e cancelamento para clientes e profissionais, lista de aniversariantes, desconto por frequência e retenção de dados em conformidade com a LGPD. Além disso, a gerente terá acesso à relatórios financeiros e dashboards com métricas de produtividade.

1.3 Justificativa

Uma pesquisa de 2023 do SEBRAE indica mais de 1,3 milhão de atividades econômicas ligadas a negócios de beleza no Brasil, abrangendo serviços, indústria e comércio, e gerando aproximadamente R\$ 75 bilhões em faturamento anual (SEBRAE, 2023b). O gráfico abaixo (figura 1) demonstra o crescimento da quantidade de profissionais no setor da beleza nos últimos anos:

¹ Single Sign-On é um sistema que permite usar um único nome de usuário e senha para acessar vários serviços diferentes, sem precisar criar contas ou lembrar várias senhas.

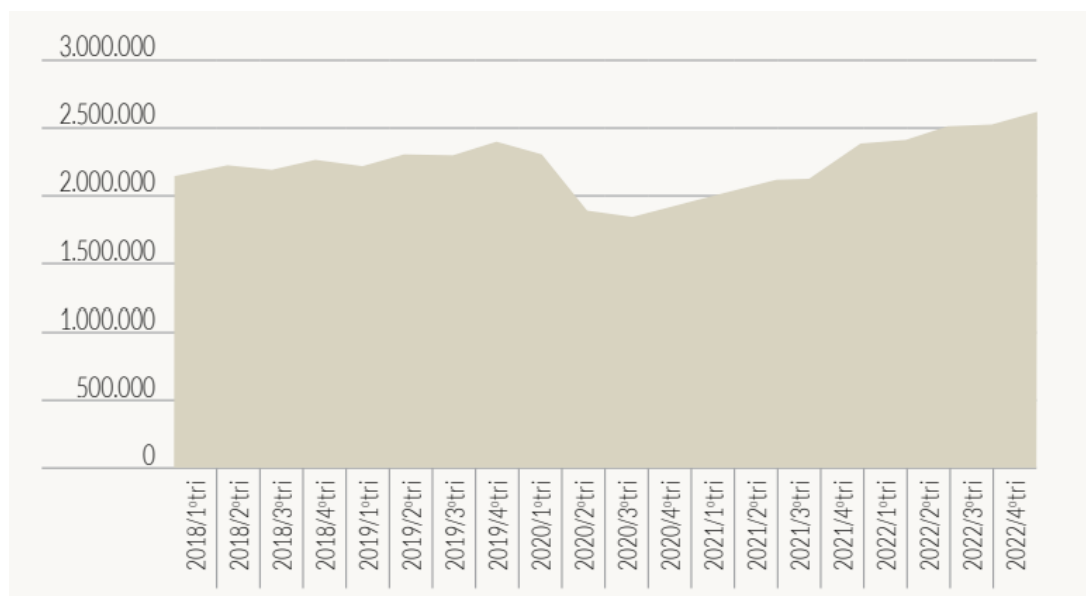


Figura 1 – Profissionais da área da beleza no Brasil 2018–2022

Fonte: (SENAC, 2023)

Embutido neste crescimento, a maior parte dos profissionais são cabeleireiros, como mostra o gráfico abaixo (2)

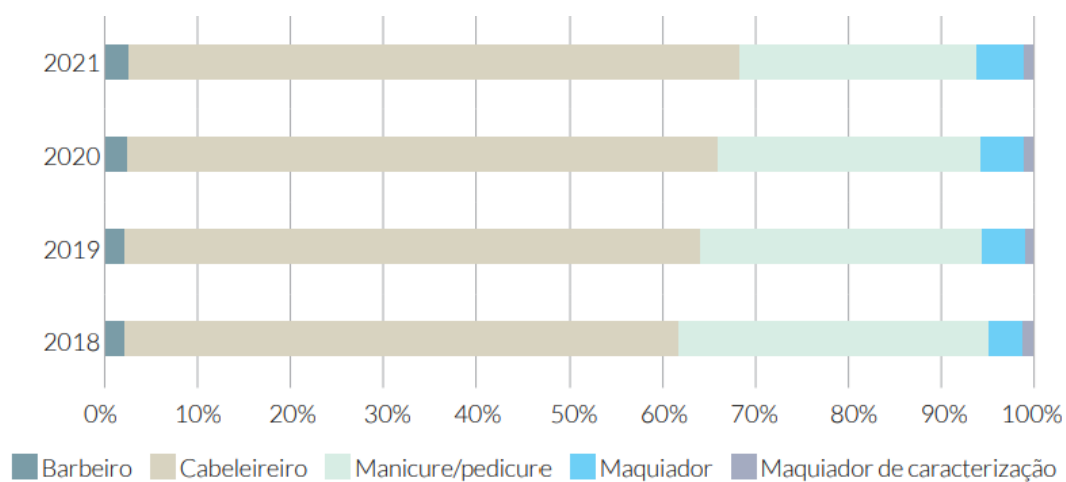


Figura 2 – Distribuição dos profissionais da área da beleza 2018-2021

Fonte: (SENAC, 2023)

Neste cenário robusto, que movimentou cerca de 27 bilhões de dólares em 2024 (ECOMMERCE NA PRÁTICA, 2025), os desafios operacionais crescem cada vez mais:

- Até 30% do tempo de um pequeno empreendedor é consumido por tarefas administrativas (SENAC-SP, 2022);
- Taxa média de não comparecimento de clientes atinge 25% (BOOKSY, 2022);

- Perda de 20% da receita por não comparecimento ([ABIHPEC, 2021](#));
- Média de 15 horas semanais dedicadas ao controle manual de agenda e finanças ([FGV, 2020](#));
- Insatisfação de 40% dos clientes devido a falhas de comunicação e alterações de última hora ([MINDMINERS, 2022](#)).

Paralelamente ao crescimento do setor de beleza, o modelo de *coworking*, originado em ambientes de escritório, expandiu-se para salões, permitindo o compartilhamento de espaços e recursos e a redução de custos ([SEBRAE, 2023a](#); [SEBRAE SC, 2025](#)). Anteriormente à população dos *coworkings* de beleza, os profissionais se distribuíam em diversos locais para economizar recursos, como mostra a figura 3

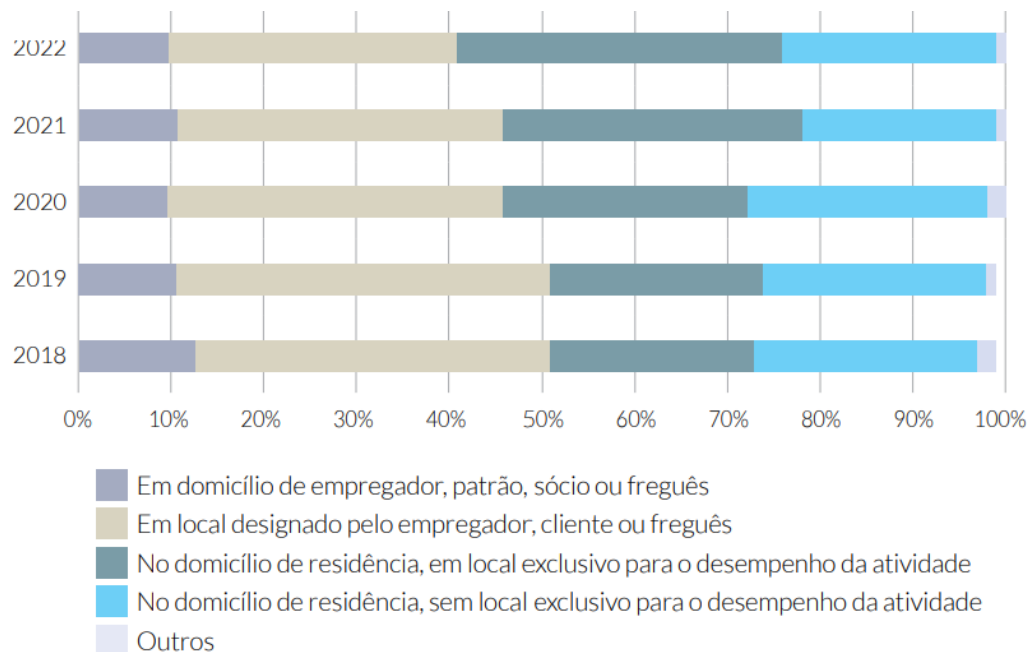


Figura 3 – Distribuição dos profissionais da área da beleza por local de trabalho 2018-2022

Fonte: ([SENAC, 2023](#))

Nesse contexto promissor, justifica-se o projeto de extensão *BS Beauty*, destinado a desenvolver uma aplicação web customizada para o gerenciamento de salões em modelo *coworking*, sob a coordenação de nossa parceira de extensão Bruna. Ao digitalizar e centralizar processos principais, a *BS Beauty* empodera pequenos empreendedores reduzindo custos operacionais e minimizando erros humanos, melhora a experiência do cliente, eleva a receita dos profissionais por meio do controle preciso de comissões e frequências, e oferece a oportunidade de *insights* estratégicos através de dashboards e relatórios financeiros detalhados.

Dessa forma, a solução não só supera os problemas de instabilidade e excesso de esforço administrativo, mas também gera valor para todos os envolvidos no salão de beleza. Além disso, como iniciativa de extensão, o projeto permite que os alunos-desenvolvedores coloquem em prática e melhorem os conhecimentos técnicos e de gestão, aprendendo com desafios reais de requisitos, usabilidade e performance. Assim, é possível aproximar a graduação das demandas do mercado.

1.4 Análise da Concorrência

Foi conduzida uma pesquisa de mercado centrada em plataformas brasileiras que combinam agendamento on-line e gestão financeira para espaços de beleza no modelo *coworking*. Deste levantamento emergiram três empresas que servirão de referência nesta análise: uma já amplamente consolidada no mercado nacional — embora atue além do universo *coworking* — e outras duas que, apesar de conhecidas, ainda estão em expansão, mas com foco mais relacionado ao da nossa proposta, o que as torna concorrentes que merecem maior atenção estratégica.

1.4.1 Trinks



Figura 4 – Logo plataforma Trinks

Trinks é uma plataforma já bem consolidada no mercado de gestão de negócios de beleza, com soluções personalizadas para barbearias, salões de beleza e clínicas de estética. Criada em 2012, é hoje a plataforma de gestão para beleza com a maior base instalada do país, englobando aproximadamente 2,8 milhões de usuários e mais de 40 mil estabelecimentos, sediada no Rio de Janeiro. A plataforma começou como um empreendimento de consultoria em software personalizado, mas logo identificou uma oportunidade no mercado da beleza e mudou de nicho. Em 2024, foi adquirida pelo grupo Stone, o que alavancou ainda mais funcionalidades do aplicativo, como o autoatendimento. Atualmente, a Trinks oferece software de back-office (conjunto de módulos internos que controlam o funcionamento do negócio como finanças, estoque, comissões e relatórios), marketplace B2C e meios de pagamento próprios (Trinks Pay), funcionando praticamente como um “ERP + iFood” para salões e barbearias. Existe um plano grátis que engloba apenas 150 agendamentos por mês, e os planos pagos variam de R\$ 59 a R\$ 249/mês ([Trinks Tecnologia da Informação, 2025](#)).

Além dos serviços comuns, seus principais diferenciais são:

- Ponto de venda (PDV) completo: integração TEF, Pix e split de comissão, atendendo desde MEIs até redes com exigência de NFC-e e SAT/ECF;
- Marketplace *Trinks.com*, que gera maior fluxo de clientes, expõe o salão ao público final e permite pagamento antecipado;
- Estrutura em nuvem madura, com SLA de 99,9 % e aplicativos nativos para iOS/Android.

Apesar dos grandes benefícios, identificamos algumas brechas do ponto de vista do negócio da nossa parceira de extensão, Bruna:

- A interface pode ser considerada “poluída” para clientes iniciantes, devido ao grande número de funcionalidades;
- Há pouco foco no aluguel de estações típico do coworking, exigindo ajustes manuais de comissão;
- Maior parte das funcionalidades estão presentes apenas nos planos superiores.

1.4.2 Gendo



Figura 5 – Logo plataforma Gendo

Lançado em 2017 e sediado em Curitiba-PR, o Gendo se posiciona como um *hub*² de gestão 100 % em nuvem para negócios além do setor da beleza, como estética, saúde, bem-estar, pet-shop e mais recentemente, espaços em formato coworking. Atualmente mantém mais de 10 mil assinantes, com maior penetração nas regiões Sul e Sudeste do Brasil. Foi criado no modelo SaaS com o intuito de oferecer prontamente agenda on-line, automação de lembretes (e-mail/WhatsApp), módulo financeiro completo e integrações com gateways de pagamento (Stone, Cielo e Mercado Pago). Atualmente, os planos são somente pagos e variam de R\$ 32 a R\$ 293/mês, após 14 dias de teste gratuito ([Gendo Sistemas, 2025](#)).

Seus principais diferenciais são:

² Hub: plataforma centralizada que integra agenda, PDV, finanças e pagamentos em um único ambiente, funcionando como “nó” que organiza os fluxos de dados do negócio.

- Caixa do profissional: Módulo pensado para coworking, possibilitando débito automático de aluguel de estação e visualização dos ganhos de cada profissional;
- Aplicativo Gendo Pro (iOS e Android): permite ao profissional ver a agenda, acompanhar comissões, pedir saques e registrar fotos de antes e depois dos serviços;
- Relatórios instantâneos: exibem ticket médio, previsão de faturamento e dados de cancelamentos, com opção de exportar para Excel.

Já os maiores pontos de melhoria identificados são:

- Base instalada ainda pequena, limitando o efeito de rede junto a grandes franquias;
- Dependência de gateways externos, o que adiciona custo extra ao *split* ³;
- Relatórios fiscais avançados disponíveis apenas no plano Premium.

1.4.3 Avec



Figura 6 – Logo plataforma Avec

A Avec é, hoje, a principal concorrente do nosso projeto, pois a entidade parceira que motivou este trabalho utiliza essa plataforma para gerenciar seu salão de beleza em modelo coworking. Por esse motivo, ela foi adotada como referência: buscamos manter as funcionalidades que já funcionam bem na Avec e, ao mesmo tempo, acrescentar ou aprimorar recursos que ainda fazem falta para a nossa parceira.

Lançada em 2014 e sediada em São Paulo-SP, a Avec se apresenta como solução “360°” para salões, barbearias, esmaltarias, spas e estúdios de tatuagem. A plataforma integra software de gestão, um sistema próprio de pagamentos (*Avec Pay*) e um marketplace B2C que encaminha novos clientes aos estabelecimentos. Segundo a empresa, mais de 40 mil negócios utilizam o serviço no Brasil. Também desenvolvida no modelo SaaS, a ferramenta oferece agenda on/line multiprofissional com confirmações via WhatsApp ou SMS, ponto de venda completo com TEF, Pix e *split* interno de comissões, além de módulo financeiro integrado. Dispõe ainda de uma carteira digital empregada em pacotes pré/pagos, gift-cards e cashback, e de dois aplicativos: o *Avec*, voltado ao cliente final, e o

³ Split é a divisão automática do pagamento entre salão e profissional que, se feita por um gateway externo, gera uma taxa extra.

Avec Pro, destinado aos profissionais. Há um plano gratuito “*Avec Go*” que inclui funções básicas e cobra apenas a taxa transacional, enquanto os planos pagos variam de R\$ 77 a R\$249 por mês ([Avec Company, 2025](#)).

Com base no feedback da nossa entidade parceira, destacam-se quatro funcionalidades que a plataforma *Avec* executa bem:

- *Split* instantâneo de comissões, dispensando gateways externos;
- Marketplace B2C e aplicativo do cliente, que ampliam a visibilidade do salão e aumentam os agendamentos on-line;
- App *Avec Pro* (iOS/Android), no qual o profissional acompanha agenda, comissões, saques e registra fotos de “antes e depois” dos serviços.

As principais brechas identificadas são:

- módulos fiscais avançados (NF-e e SAT) disponíveis apenas nos planos superiores;
- dependência do hardware e das tarifas do próprio *Avec Pay* para uso pleno do sistema;
- custos adicionais para envios em massa de SMS/WhatsApp em campanhas de marketing;
- instabilidade recorrente: o domínio eventualmente fica fora do ar.

1.4.4 Quadro comparativo

Quadro 1 – Comparação entre as plataformas concorrentes e a aplicação proposta

Recurso	Trinks	Gendo	Avec	BS Beauty
Aplicação <i>web</i>	—	✓	✓	✓
Foco no coworking	—	✓	—	✓
Controle de acesso para gestão	—	✓	✓	✓
Agendamento de serviços 100% on-line	✓	✓	✓	✓
Controle de conflitos de agenda	✓	✓	✓	✓
Avaliação pós-serviço	✓	✓	✓	✓
Plataforma do <i>cliente</i>	✓	✓	✓	✓
Plataforma do <i>profissional</i>	✓	✓	✓	✓
Confirmação automática (WhatsApp / SMS / e-mail)	✓	✓	✓	✓
Cálculo de <i>Split</i> de comissão	✓	✓	✓	—
Pagamento on-line	✓	—	✓	—
Marketplace B2C	✓	—	✓	—
Marketing integrado (envio em massa de SMS/WhatsApp/e-mail)	✓	—	✓	✓
Gift-card	✓	—	✓	✓
Programa de indicação	✓	—	✓	✓
Relatório financeiro em tempo real	✓	✓	✓	✓
Login simplificado com integração Google	✓	✓	—	✓
Plano gratuito disponível	✓	—	✓	✓
Lista de aniversariantes para promoções	✓	✓	—	✓

Fonte: Produzido pelos autores

2 Revisão da Literatura

Devido ao crescimento contínuo dos coworkings de beleza, gerenciar as informações, ter processos ágeis e oferecer um atendimento de qualidade são atitudes cada vez mais essenciais para se destacar em um mercado tão competitivo. Por isso, a necessidade de utilizar sistemas para gerenciar os dados gerados é crucial para o sucesso de empreendedores, considerando a demanda crescente e sofisticada das tecnologias da informação para registrar dados de forma confiável, melhorar o fluxo de trabalho interno e automatizar processos que antes eram manuais.

2.1 Histórico de sistemas gerenciadores de agendamento

O ato de agendar serviços é uma prática antiga, e o gerenciamento desses agendamentos sempre foi um processo trabalhoso e passível de erros. No entanto, à medida que a tecnologia avança, surgem também ferramentas que facilitam esse processo, diminuindo a chance de falhas e perdas.

Nos primórdios, o agendamento de qualquer serviço era feito apenas presencialmente, dada a falta de tecnologias de comunicação à distância. Consequentemente, o controle financeiro e de clientes era realizado manualmente, o que gerava a necessidade de contratar outras pessoas para auxiliar neste processo, causando mais gastos. Com o advento do telefone, o acesso dos clientes tornou-se mais fácil. Porém, o trabalho de gestão ainda permanecia manual, exceto nos casos em que se adquiriam soluções de gestão em DVDs, que eram pouco personalizadas para o negócio específico e não integradas aos agendamentos. Além disso, mantinha-se a necessidade de alguém disponível para atender às chamadas ou para controlar a correlação das agendas com as informações do DVD.

Posteriormente, com o surgimento da internet e das redes sociais, a maioria dos empresários prestadores de serviços aproveitou a oportunidade para concentrar seus agendamentos em mensagens de texto. Essa abordagem eliminava a necessidade de alguém estar sempre disponível para responder e confirmar, além de permitir a comunicação paralela com clientes. Ademais, a gestão já podia ser mais integrada a calendários virtuais (como o *Google Calendar*), aos do próprio *smartphone*, ou mesmo a planilhas digitais. Contudo, o processo de agendamento ainda dependia de uma ferramenta de comunicação que exigia intervenção humana: uma pessoa precisava estar envolvida na conversa para anotar o serviço agendado em outra ferramenta e controlar as finanças do negócio – tarefas ainda manuais ou realizadas com um sistema à parte, o que poderia levar à perda de informações. Esse excesso de ferramentas e a comunicação fragmentada consumiam tempo e podiam resultar na desistência de clientes.

Diante desse cenário, surgiu a necessidade de um sistema de agendamento automático, que dispensasse a comunicação direta e já integrasse o processo de gestão de clientes e finanças na mesma plataforma. Portanto, surgiram os sistemas de gestão de agendamento online, com a promessa de reduzir gastos, minimizar processos manuais e aumentar a produtividade. Além disso, a possibilidade de fazer reservas a qualquer hora ajuda a atrair mais clientes e a mantê-los (Reservio, 2024).

Por fim, os sistemas de gestão e agendamento de serviços evoluíram rapidamente, tornando-se cada vez mais personalizados para diferentes setores ou até mesmo para empresas diferentes dentro do mesmo setor. Atualmente, é possível contratar facilmente uma instância de aplicação específica para salões de beleza, adaptada ao próprio negócio, dada a vasta quantidade de soluções já existentes no mercado.

2.2 Atualidade de sistemas gerenciadores de agendamento

Atualmente, soluções digitais para gerenciamento e agendamento de serviços já vêm personalizadas para o setor da beleza, por meio de assinaturas. Algumas focam até mesmo na gestão de *coworking*. Consequentemente, consolidaram-se funcionalidades padrão que são fundamentais para manter a competitividade no mercado. A Figura 7 ilustra os principais módulos embutidos em softwares e aplicativos disponíveis para espaços de beleza:



Figura 7 – módulos básicos em sistemas para salão de beleza

Fonte: (SEBRAE, 2023c)

Pensando especificamente em *coworkings* de beleza, surgem necessidades particulares:

- Evitar sobreposições de horários em cadeiras, cabines ou equipamentos;
- Gerenciar a divisão de recursos (produtos ou aparelhos);

- Calcular automaticamente alugueis de estação ou comissões, com *split* de pagamento entre o espaço e o profissional;
- Controlar acessos e permissões de cada perfil (administrador, profissional, cliente).

Embora algumas plataformas já ofereçam módulos de *coworking*, muitas ainda exigem configurações e ajustes manuais para o cálculo de comissões e a alocação de recursos, o que gera retrabalho e eleva o risco de erros. Para enfrentar esses desafios, destacam-se hoje as seguintes tendências tecnológicas:

- Aplicativos dedicados ao relacionamento independente entre profissionais e clientes para gestão de agenda;
- Pagamentos em tempo real;
- *Dashboards* analíticos com indicadores de ocupação, performance individual e projeção de demanda.

No entanto, escolher uma solução digital para um *coworking* de beleza vai além de automatizar processos e melhorar a gestão de serviços. Dada a rapidez do avanço tecnológico e a influência das informações na internet, o *marketing* integrado e uma interface de usuário intuitiva passam a ser diferenciais decisivos. Um design previsível melhora a experiência, fideliza clientes e reduz o tempo de treinamento dos profissionais.

3 GESTÃO DO PROJETO

Neste capítulo de Gestão do Projeto são abordados tópicos como a organização da equipe (definindo as responsabilidades, papéis e atividades de cada integrante do grupo), a metodologia adotada para a gestão e desenvolvimento do projeto, bem como o repositório da aplicação.

3.1 Organização da Equipe

A equipe do presente projeto é composta por seis docentes do curso de graduação Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Campus São Paulo do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP-SPO), a saber:

- Alyson César Fumagalli dos Santos Júnior
- Bruno de Almeida Fischer
- Eliel da Silva
- Giovanna Camille Silva Carvalho
- Henrique Santiago Pires
- Henriky Jhonny de Oliveira Bastos

O grupo de trabalho foi formado logo no início da disciplina de Projeto Integrado de Extensão 1 (SPOPIE1) por estudantes que já haviam realizado diversos outros trabalhos em conjunto e, portanto, estavam acostumados a trabalhar em equipe.

Visando uma transmissão de informações clara e centralizada, utilizou-se o Discord ([DISCORD, 2025](#)) como ferramenta de comunicação para realizar reuniões assíncronas; e o Notion ([NOTION, 2025](#)) para organizar documentos, atribuir tarefas e monitorar o andamento do projeto (juntamente com o ProjectLibre ([PROJECTLIBRE, 2025](#))).

3.1.1 Responsabilidades / Papéis / Atividades

Para cada integrante da equipe foi definido um papel contendo responsabilidades e atividades definidas com base em suas respectivas proeficiências em diferentes áreas a fim de distribuir as tarefas do projeto de maneira eficiente.

O Quadro 2 descreve a distribuição dos membros do grupo com base em seus respectivos papéis de uma forma mais geral.

Quadro 2 – Membros e seus respectivos papéis

Membro	Gestor	<i>Tech Lead</i>	Desenvolvedor <i>Full Stack</i>	Redatora Técnica
Alyson	✓		✓	
Bruno			✓	
Eliel			✓	
Giovanna				✓
Henrique			✓	
Henriky		✓	✓	

Fonte: Produzido pelos autores

Assim, constata-se que a equipe conta com 1 gestor, responsável pelo gerenciamento de todo o projeto; 1 *tech lead*, encarregado de guiar a equipe de desenvolvimento; 5 desenvolvedores *full stack* (com 2 deles desempenhando outros papéis paralelos) incumbidos por desenvolver o sistema em todas as suas etapas e 1 redatora técnica para supervisionar as documentações do projeto.

O Quadro 3 apresenta as atividades desempenhadas pelos membros da equipe nas diversas áreas que contemplaram o desenvolvimento do projeto.

Quadro 3 – Membros e suas atividades

Membro	Notion	<i>Front-End</i> <i>Back-End</i>	Banco de Dados	Documentação	Diário de Bordo	ProjectLibre
Alyson	✓	✓	✓	✓	✓	✓
Bruno	✓	✓	✓	✓		✓
Eliel	✓	✓	✓	✓		✓
Giovanna	✓		✓	✓	✓	✓
Henrique	✓	✓	✓	✓		
Henriky	✓	✓	✓	✓		

Fonte: Produzido pelos autores

3.2 Metodologias de Gestão e Desenvolvimento

Para o projeto, foi adotada a metodologia Scrum ([AWS, 2024](#)) tanto voltada para a gestão quanto para o desenvolvimento do sistema.

3.2.1 Scrum

O *framework* Scrum foi escolhido por ter sido bastante estudado em disciplinas anteriores e também devido à equipe já ter uma certa familiaridade em trabalhar com ele.

A metodologia foi adaptada de forma a incorporar alguns elementos do Kanban (EQUIPE TOTVS, 2023) a fim de ter um sistema visual para monitorar as atividades em andamento.

Tendo em mente as responsabilidades provenientes do Scrum, o Quadro 4 descreve como os membros do grupo foram distribuídos seguindo os papéis da metodologia.

Quadro 4 – Papéis dos integrantes com base no Scrum

Membro	<i>Product Owner</i>	<i>Scrum Master</i>	Equipe de Desenvolvimento
Alyson		✓	✓
Bruno			✓
Eliel			✓
Giovanna			✓
Henrique			✓
Henriky	✓		✓

Fonte: Produzido pelos autores

A plataforma Notion (NOTION, 2025) teve um papel significativo na implementação do Scrum: por meio dela, criou-se um espaço de trabalho para registrar o *product backlog*, os *sprints* e suas respectivas tarefas.

Com essa ferramenta, ainda foi possível atribuir informações — como prioridade, *status*, prazos e responsáveis — para cada atividade, além de criar visualizações para simular as colunas e cartões do Kanban, criando uma estrutura organizada e informativa que facilitou o gerenciamento do projeto.

3.2.1.1 Sprints

Os *sprints* do projeto possuem duração de 2 semanas para garantir que a equipe tenha tempo suficiente para implementar os requisitos definidos para um determinado incremento.

Dessa forma, estabeleceu-se os *sprints* com base nos itens do *product backlog*, que por sua vez foram definidos tendo em mentes as histórias de usuário levantadas junto à entidade parceira do projeto.

Ademais, as reuniões características da metodologia Scrum foram marcadas para ocorrer presencialmente nos dias referentes às disciplinas de Projeto Integrado de Extensão, nas quais os membros poderiam se juntar e organizar para definir quais itens seriam trabalhados no próximo *sprint*, bem como os prazos, tarefas, prioridades e responsáveis.

3.3 Repositório da Aplicação

Nessa seção do capítulo, apresenta-se o repositório da aplicação, que contém todos os arquivos relevantes ao projeto (como código-fonte, documentos e diagramas).

3.3.1 Definição do repositório da aplicação

Baseado na familiaridade dos integrantes em utilizar o sistema de controle de versão Git (GIT, 2025), escolheu-se o GitHub (GITHUB, 2025c) para hospedar o repositório remoto da aplicação e tornar o desenvolvimento mais colaborativo.

Visando garantir um repositório organizado e eficiente, estabeleceu-se uma estrutura de diretórios separando código-fonte da documentação do projeto e adotou-se o modelo de fluxo de trabalho *Git Flow* (OBJECTIVE, 2023) para organizar o versionamento de ramificações. Além disso, foi utilizado *Conventional Commits* (CONVENTIONAL COMMITS, 2025) para padronizar as mensagens de *commit*, conferindo ainda mais organização.

3.3.1.1 Link do repositório e especificações para acesso

O QR Code abaixo (Figura 8) contém o link que leva diretamente ao repositório remoto do projeto hospedado no GitHub. É possível tanto escanear quanto clicar no código abaixo para acessar o repositório.

Figura 8 – QR Code do repositório da aplicação



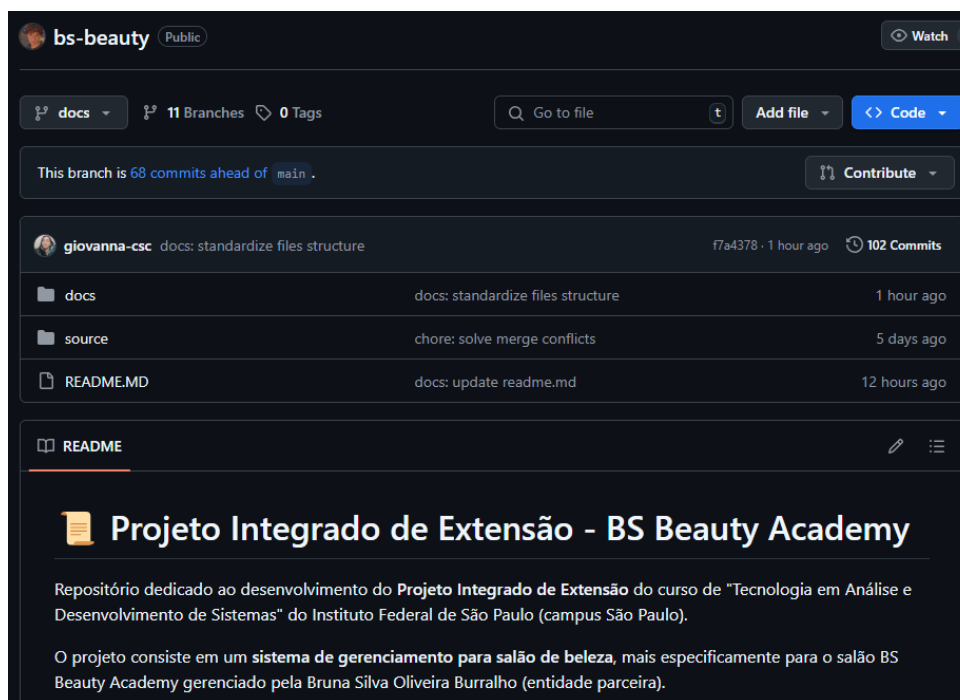
Fonte: Produzido pelos autores

Acessado o repositório, será aberta uma guia no navegador contendo uma página semelhante à Figura 9. Todos os arquivos do projeto podem ser visualizados diretamente pelo navegador.

Como o repositório é público, qualquer pessoa pode acessá-lo e cloná-lo localmente usando HTTPS. Para tanto, é necessário seguir as seguintes etapas:

1. Instalar o Git na máquina (caso não esteja instalado).
2. Escolher o diretório no qual o repositório será clonado.
3. Abrir o terminal e alterar o diretório atual para o diretório escolhido.
4. Usar o comando `git clone https://github.com/Henrriky/bs-beauty`.
5. Usar o comando `cd bs-beauty` para acessar o repositório clonado.

Figura 9 – Página inicial do repositório



Fonte: Produzido pelos autores

Assim, o repositório estará devidamente clonado na máquina. Para execução local, utilize a *branch* principal `main` e consulte o arquivo `README.MD` do repositório, que contém instruções de instalação de dependências e execução do projeto.

Para utilizar outros métodos de clonagem (como SSH, GitHub CLI ou baixar o arquivo `.zip` do projeto), consulte a documentação oficial do GitHub ([GITHUB, 2025a](#)) referente às formas de clonar um repositório.

4 Desenvolvimento do projeto

Este capítulo descreve o processo de desenvolvimento do sistema proposto, abordando desde a definição de seu escopo até os testes realizados para garantir sua qualidade. Para isso, são apresentados os elementos que nortearam a construção da solução, como as histórias de usuário, as decisões arquiteturais, as tecnologias adotadas e as práticas de verificação aplicadas.

4.1 Escopo do Projeto

Esta seção apresenta o escopo do sistema desenvolvido, com o objetivo de delimitar suas funcionalidades, comportamentos esperados e restrições de operação. As definições aqui descritas foram elaboradas com base em reuniões realizadas com a gestora do salão de beleza, nas quais foram discutidas as necessidades do negócio e as principais demandas da operação cotidiana.

Com base nessas interações, foram estabelecidas as regras de negócio que orientam o funcionamento do sistema, os requisitos funcionais que especificam os serviços a serem oferecidos aos usuários e os requisitos não funcionais que tratam de aspectos como desempenho, segurança e usabilidade.

A definição clara desses elementos é essencial para garantir a coerência do sistema com as necessidades dos usuários e para orientar a equipe de desenvolvimento ao longo das etapas do projeto.

4.1.1 Regras de Negócio

As regras de negócio definem os comportamentos, restrições e condições que regem o funcionamento do sistema, conforme as particularidades da BS Beauty Academy, e representam diretrizes que devem ser respeitadas tanto no uso da aplicação quanto no desenvolvimento das funcionalidades.

A seguir, são listadas as principais regras de negócio identificadas:

ID	Descrição da Regra de Negócio	Área Impactada
RN01	Não há políticas ou penalidades referentes a cancelamentos ou atrasos.	Agenda
RN02	Clientes podem agendar múltiplos serviços em um mesmo agendamento.	Agenda, Clientes

ID	Descrição da Regra de Negócio	Área Impactada
RN03	Pagamentos são realizados só presencialmente.	Atendimento, Financeiro, Marketing e Vendas
RN04	Clientes devem avaliar os serviços concluídos.	Atendimento, Clientes
RN05	Não deve haver sobreposição de horários em agendamentos.	Agenda, Atendimento
RN06	Agendamentos devem ser feitos com, no mínimo, 3 horas de antecedência.	Agenda
RN07	O cadastro e edição de funcionárias é feito exclusivamente pela gerente.	RH/Profissionais
RN08	Profissionais podem definir seus próprios horários de trabalho.	Agenda, RH/Profissionais
RN09	Profissionais podem bloquear horários específicos na agenda.	Agenda, RH/Profissionais
RN10	Preços e duração de serviços podem ser definidos pela profissional (autônoma) ou pela gerente (modelo comissionado).	Financeiro, Marketing e Vendas, RH/Profissionais
RN11	O salão possui horário fixo de funcionamento (definido pela gerente).	Atendimento, RH/Profissionais
RN12	Clientes devem informar como conheceram o salão.	Atendimento, Clientes
RN13	Programa de indicação para indicar 3 pessoas e ganhar um procedimento grátis.	Clientes, Financeiro, Marketing e Vendas
RN14	Clientes aniversariantes devem receber mensagens de aniversário e descontos.	Clientes, Marketing e Vendas
RN15	Clientes, profissionais e a gerente devem receber notificações referentes a agendamentos.	Agenda, Atendimento, Clientes, RH/Profissionais
RN16	E-mails e telefones cadastrados por clientes e profissionais devem ser únicos.	Clientes, RH/Profissionais
RN17	Cada serviço possui um tempo de duração definido ao cadastrá-lo.	Agenda, Atendimento
RN18	Não é possível agendar serviços para datas passadas.	Agenda, Atendimento
RN19	Serviços devem estar vinculados a uma ou mais profissionais especializadas.	Atendimento, RH/Profissionais
RN20	Gerente e profissionais devem receber relatórios diversos.	RH/Profissionais
RN21	O histórico de avaliações e agendamentos deve ser mantido.	Atendimento

ID	Descrição da Regra de Negócio	Área Impactada
RN22	Clientes devem agendar com base no serviço ou profissional.	Agenda, Clientes
RN23	A quantidade de clientes atendidos por uma profissional depende do tempo de cada procedimento.	Agenda, Atendimento, RH/Profissionais
RN24	Atualmente, o salão oferece serviços de depilação, unhas, estética corporal e facial, sobrancelhas, remoção e cílios.	Agenda, Atendimento
RN25	Profissionais podem criar seus próprios serviços (com limitações de acesso).	Agenda, RH/Profissionais
RN26	A gerente deve aprovar os novos serviços que forem criados.	Agenda, RH/Profissionais.

4.1.2 Requisitos Funcionais

Os requisitos funcionais representam as funcionalidades que o sistema deve implementar a fim de garantir o cumprimento das regras de negócio definidas durante as reuniões entre a equipe de desenvolvimento e a gestora do estabelecimento. Esses requisitos foram organizados de forma a manter uma correspondência clara com as regras de negócio, permitindo maior rastreabilidade e coerência entre as decisões do domínio e a implementação técnica do sistema.

Código	Descrição do Requisito Funcional
RN01	
RF01-01	O sistema não deve bloquear clientes que se atrasaram em agendamentos anteriores.
RF01-02	O sistema deve permitir cancelamentos de agendamentos sem aplicação de multas ou penalidades.
RN02	
RF02-01	O sistema deve permitir a seleção de múltiplos serviços em uma única operação de agendamento.
RF02-02	O sistema deve calcular o tempo total necessário para todos os serviços selecionados.
RF02-03	O sistema deve calcular o valor total de todos os serviços agendados.
RF02-04	O sistema deve permitir a visualização do resumo dos múltiplos serviços agendados.
RF02-05	O sistema deve verificar a disponibilidade de profissionais para realizar os serviços selecionados no período necessário.
RN03	

Código	Descrição do Requisito Funcional
RF03-01	O sistema não deve solicitar informações de pagamento durante o agendamento online.
RF03-02	O sistema deve gerar um registro de pagamento pendente para cada agendamento.
RF03-03	O sistema deve disponibilizar interface para registro de pagamentos apenas em terminais do salão.
RF03-04	O sistema deve permitir o registro de diferentes formas de pagamento presencial.
RN04	
RF04-01	O sistema deve solicitar avaliação após cada serviço concluído.
RF04-02	O sistema deve enviar notificação de avaliação pendente após o atendimento.
RF04-03	O sistema deve permitir a inclusão de comentários textuais nas avaliações.
RF04-04	O sistema deve compilar e exibir estatísticas de avaliações para gerentes e profissionais.
RN05	
RF05-01	O sistema deve verificar a disponibilidade do profissional antes de confirmar um agendamento.
RF05-02	O sistema deve impedir o agendamento quando houver conflito de horários para o mesmo profissional.
RF05-03	O sistema deve exibir alertas visuais quando tentativas de sobreposição forem detectadas.
RF05-04	O sistema deve considerar o tempo de duração de cada serviço para evitar sobreposições parciais.
RN06	
RF06-01	O sistema deve permitir à gerente definir o tempo mínimo de antecedência.
RF06-02	O sistema deve validar a antecedência mínima para todos os agendamentos.
RF06-03	O sistema deve bloquear tentativas de agendamento inferiores ao tempo de antecedência definido.
RF06-04	O sistema deve exibir mensagem informativa sobre a política de antecedência mínima.
RF06-05	O sistema deve calcular automaticamente os horários disponíveis considerando a antecedência mínima.
RN07	
RF07-01	O sistema deve restringir as operações de cadastro de funcionárias ao perfil de gerente.
RF07-02	O sistema deve restringir as operações de edição de dados de funcionárias ao perfil de gerente.

Código	Descrição do Requisito Funcional
RF07-03	O sistema deve disponibilizar interface de gerenciamento de funcionárias apenas para usuários com perfil de gerente.
RF07-04	O sistema deve registrar log de todas as operações de cadastro e edição de funcionárias.
RF07-05	O sistema deve notificar funcionárias quando seus dados forem alterados pela gerente.
RN08	
RF08-01	O sistema deve disponibilizar interface para profissionais definirem seus horários de trabalho.
RF08-02	O sistema deve permitir a configuração de horários diferentes para cada dia da semana.
RF08-03	O sistema deve validar os horários definidos conforme o horário de funcionamento do salão.
RF08-04	O sistema deve atualizar automaticamente a agenda de disponibilidade.
RN09	
RF09-01	O sistema deve permitir que profissionais bloqueiem intervalos específicos em sua agenda.
RF09-02	O sistema deve possibilitar a definição de motivo para o bloqueio (opcional).
RF09-03	O sistema deve impedir novos agendamentos nos horários bloqueados.
RF09-04	O sistema deve permitir o desbloqueio de horários previamente bloqueados.
RF09-05	O sistema deve exibir visualmente os horários bloqueados na interface de agendamento.
RN10	
RF10-01	O sistema deve permitir configuração de preços e duração de serviços conforme o modelo de trabalho da profissional.
RF10-02	O sistema deve restringir edição de preços e duração às profissionais autônomas ou à gerente, conforme o caso.
RF10-03	O sistema deve calcular automaticamente comissões para profissionais do modelo comissionado.
RF10-04	O sistema deve guardar histórico de alterações de preços e duração de serviços.
RN11	
RF11-01	O sistema deve permitir à gerente configurar os horários de funcionamento do salão para cada dia da semana.
RF11-02	O sistema deve impedir agendamentos fora do horário de funcionamento.
RF11-03	O sistema deve exibir os horários de funcionamento para clientes durante o processo de agendamento.

Código	Descrição do Requisito Funcional
RF11-04	O sistema deve permitir configuração de horários especiais para datas comemorativas ou feriados.
RF11-05	O sistema deve adaptar a visualização da agenda conforme os horários de funcionamento.
RN12	
RF12-01	O sistema deve solicitar informação sobre como o cliente conheceu o salão durante o cadastro inicial.
RF12-02	O sistema deve oferecer opções predefinidas (redes sociais, indicação, busca online, etc.).
RF12-03	O sistema deve permitir especificar o nome do indicador quando a opção for "indicação".
RF12-04	O sistema deve gerar relatórios estatísticos sobre as fontes de captação de clientes.
RF12-05	O sistema deve permitir atualização dessa informação em caso de erro ou mudança.
RN13	
RF13-01	O sistema deve registrar e contabilizar indicações feitas por cada cliente.
RF13-02	O sistema deve confirmar que novos clientes indicados realizaram ao menos um serviço pago.
RF13-03	O sistema deve notificar automaticamente quando um cliente atingir 3 indicações confirmadas.
RF13-04	O sistema deve gerar um voucher/crédito para procedimento gratuito.
RF13-05	O sistema deve permitir a seleção e aplicação do procedimento gratuito durante o agendamento.
RF13-06	O sistema deve reiniciar a contagem de indicações após o uso do benefício.
RN14	
RF14-01	O sistema deve identificar automaticamente clientes aniversariantes do dia/semana/mês.
RF14-02	O sistema deve enviar automaticamente mensagem de aniversário na data correspondente.
RF14-03	O sistema deve gerar cupom de desconto para uso durante o mês de aniversário.
RF14-04	O sistema deve permitir a aplicação do desconto de aniversário durante o agendamento no mês correspondente.
RF14-05	O sistema deve registrar o uso do benefício para evitar duplicidades.
RN15	

Código	Descrição do Requisito Funcional
RF15-01	O sistema deve enviar confirmação de agendamento para o cliente e profissional envolvidos.
RF15-02	O sistema deve enviar lembretes 24h antes do horário agendado.
RF15-03	O sistema deve notificar sobre cancelamentos ou alterações nos agendamentos.
RF15-04	O sistema deve permitir configuração dos tipos de notificação desejados (email, SMS, push).
RN16	
RF16-01	O sistema deve validar a unicidade de e-mails durante o cadastro de usuários.
RF16-02	O sistema deve validar a unicidade de telefones durante o cadastro de usuários.
RF16-03	O sistema deve exibir alerta quando houver tentativa de cadastro com email/telefone já existente.
RF16-04	O sistema deve permitir a atualização de e-mail/telefone com validação de unicidade.
RN17	
RF17-01	O sistema deve solicitar o tempo de duração durante o cadastro de cada serviço.
RF17-02	O sistema deve utilizar a duração para cálculo de disponibilidade na agenda.
RF17-03	O sistema deve permitir duração personalizada por profissional para o mesmo tipo de serviço.
RF17-04	O sistema deve considerar a duração para evitar sobreposições de agendamentos.
RF17-05	O sistema deve exibir o tempo de duração estimado durante o processo de agendamento.
RN18	
RF18-01	O sistema deve validar se a data e hora de agendamento são futuras em relação ao momento da operação.
RF18-02	O sistema deve bloquear e exibir mensagem de erro para tentativas de agendamento em datas/horários passados.
RF18-03	O sistema deve atualizar automaticamente os horários disponíveis conforme o avanço do tempo.
RF18-04	O sistema deve considerar o fuso horário do salão para validação das datas e horários.
RF18-05	O sistema deve desabilitar visualmente na interface os dias e horários que já passaram.
RN19	
RF19-01	O sistema deve permitir a vinculação de serviços a profissionais específicas durante o cadastro do serviço.

Código	Descrição do Requisito Funcional
RF19-02	O sistema deve exibir apenas profissionais habilitadas para cada serviço durante o agendamento.
RF19-03	O sistema deve permitir que uma profissional seja vinculada a múltiplos serviços.
RF19-04	O sistema deve permitir à gerente atualizar as vinculações entre serviços e profissionais.
RF19-05	O sistema deve notificar profissionais quando forem vinculadas a novos serviços.
RN20	
RF20-01	O sistema deve gerar relatórios de produtividade por profissional (serviços realizados, valores, avaliações).
RF20-02	O sistema deve gerar relatórios financeiros (receita diária, semanal, mensal) para a gerência.
RF20-03	O sistema deve gerar relatórios de ocupação (horários mais procurados, serviços mais solicitados).
RF20-04	O sistema deve permitir a exportação dos relatórios em formatos diversos (PDF, Excel, CSV).
RF20-05	O sistema deve gerar dashboard com indicadores-chave de desempenho para gerência e profissionais.
RN21	
RF21-01	O sistema deve armazenar permanentemente todos os agendamentos realizados e suas informações.
RF21-02	O sistema deve manter registro completo de todas as avaliações de serviços feitas pelos clientes.
RF21-03	O sistema deve permitir consulta ao histórico de serviços e avaliações por cliente.
RF21-04	O sistema deve possibilitar consulta ao histórico de serviços realizados por profissional.
RF21-05	O sistema deve implementar política de retenção de dados conforme legislação vigente (LGPD).
RN22	
RF22-01	O sistema deve oferecer duas vias principais de agendamento: por serviço ou por profissional.
RF22-02	O sistema deve exibir lista de profissionais disponíveis após a seleção do serviço.
RF22-03	O sistema deve exibir lista de serviços disponíveis após a seleção da profissional.

Código	Descrição do Requisito Funcional
RF22-04	O sistema deve permitir filtragem por data/horário em ambos os fluxos de agendamento.
RF22-05	O sistema deve exibir informações sobre a profissional (experiência, avaliações) durante o agendamento por serviço.
RF22-06	O sistema deve exibir informações sobre o serviço (duração, preço, descrição) durante o agendamento por profissional.
RN23	
RF23-01	O sistema deve calcular automaticamente a disponibilidade da profissional com base na duração dos serviços agendados.
RF23-02	O sistema deve bloquear horários subsequentes conforme a duração do serviço agendado.
RF23-03	O sistema deve considerar intervalos entre atendimentos quando configurados.
RF23-04	O sistema deve atualizar a agenda em tempo real conforme novos agendamentos são realizados.
RF23-05	O sistema deve permitir visualização da taxa de ocupação diária da profissional.
RN24	
RF24-01	O sistema deve permitir cadastro e categorização dos serviços por tipo (depilação, unhas, estética corporal, etc.).
RF24-02	O sistema deve possibilitar a organização hierárquica dos serviços (categorias e subcategorias).
RF24-03	O sistema deve exibir serviços agrupados por categoria na interface de agendamento.
RF24-04	O sistema deve permitir busca de serviços por nome ou categoria.
RF24-05	O sistema deve exibir informações detalhadas sobre cada serviço (descrição, duração, valor).
RF24-06	O sistema deve permitir adição de imagens ilustrativas para cada categoria de serviço.
RN25	
RF25-01	O sistema deve disponibilizar interface para profissionais cadastrarem novos serviços.
RF25-02	O sistema deve implementar fluxo de aprovação pela gerência para novos serviços criados.
RF25-03	O sistema deve permitir definição de parâmetros específicos para serviços personalizados (preço, duração, descrição).
RF25-04	O sistema deve restringir a visualização e agendamento dos serviços personalizados conforme configuração da profissional.

Código	Descrição do Requisito Funcional
RF25-05	O sistema deve notificar clientes elegíveis sobre novos serviços personalizados quando configurado.
RF25-06	O sistema deve fazer com que novos serviços sejam criados com o status de "Pendente de Aprovação".
RF25-07	O sistema deve impedir que serviços com status "Pendente" sejam disponibilizados para agendamento.
RF25-08	O sistema deve gerar automaticamente uma solicitação de aprovação quando um novo serviço for criado.
RN26	
RF26-01	O sistema deve notificar a gerente imediatamente quando um novo serviço for submetido para aprovação.
RF26-02	O sistema deve notificar a profissional sobre o status da avaliação (aprovado, rejeitado, ou solicitação de alterações).
RF26-03	O sistema deve permitir que a gerente inclua comentários justificando aprovação ou rejeição.
RF26-04	O sistema deve disponibilizar uma área exclusiva para a gerente visualizar todos os serviços pendentes de aprovação.
RF26-05	O sistema deve exibir todas as informações do serviço proposto de forma clara e organizada.
RF26-06	O sistema deve permitir que a gerente aprove, rejeite ou solicite modificações nos serviços.
RF26-07	O sistema deve exibir informações sobre a profissional que criou o serviço.
RF26-08	O sistema deve impedir que profissionais modifiquem serviços que estão em processo de aprovação.
RF26-09	O sistema deve manter os seguintes status para serviços: "Rascunho", "Pendente de Aprovação", "Aprovado", "Rejeitado", "Requer Modificações".
RF26-10	O sistema deve permitir que serviços rejeitados sejam editados e resubmetidos para nova avaliação.
RF26-11	O sistema deve automaticamente disponibilizar serviços aprovados na lista de agendamento.

4.1.3 Requisitos Não Funcionais

A seguir, são apresentados os requisitos não funcionais, relativos a qualidade e a experiência do sistema, garantindo aspectos essenciais como usabilidade, segurança e design.

Código	Descrição do Requisito Não Funcional
RNF01	O sistema deve ser intuitivo e fácil de usar.
RNF02	O sistema deve estar constantemente disponível.
RNF03	O sistema deve proporcionar segurança aos usuários.
RNF04	O sistema deve ser responsivo, adaptando-se a diferentes dispositivos.
RNF05	O sistema deve ter um design simples, minimalista e chique.
RNF06	O sistema deve ter processo de cadastro simplificado.

4.2 Histórias de Usuário

Esta seção apresenta as histórias de usuário elaboradas com o objetivo de descrever os requisitos funcionais do sistema sob a perspectiva dos usuários finais. As histórias foram definidas com base nas necessidades identificadas, e visam representar, de forma simples e objetiva, as funcionalidades esperadas por cada tipo de usuário do sistema: *customer* (cliente), *employee* (funcionário) e *manager* (gerente).

Cada história é redigida utilizando a estrutura “Como [tipo de usuário], eu quero [ação] para [objetivo]”, permitindo evidenciar o papel do usuário, a funcionalidade desejada e o benefício esperado.

4.2.1 Descrição das Histórias de Usuário

4.3 Arquitetura

Nessa seção do capítulo, apresenta-se a arquitetura da aplicação, que define como os componentes do sistema interagem entre si e com o usuário.

4.3.1 Definições da arquitetura

O sistema foi estruturado com base no modelo cliente-servidor, no qual o front-end e o back-end operam como aplicações independentes que se comunicam por meio de requisições HTTP, seguindo a arquitetura RESTful.

O back-end é responsável pelo processamento da lógica de negócios e pela persistência dos dados, enquanto o front-end realiza a apresentação e interação com o usuário. Essa separação garante maior modularidade e facilita a manutenção da aplicação, cujo domínio é um ambiente de *coworking* para salões de beleza.

A arquitetura do sistema adota um padrão baseado no Model-View-Controller (MVC) de forma adaptada, aproximando-se de uma arquitetura em camadas. Cada camada possui uma responsabilidade bem definida, conforme descrito a seguir:

- **Controller:** Responsável por receber e tratar as requisições provenientes do cliente (front-end), encaminhando-as para a camada de serviço correspondente. Esta camada lida diretamente com aspectos de infraestrutura externa, como servidores de borda ou gateways de entrada da aplicação, servindo como ponto inicial de processamento das solicitações.
- **Service:** Centraliza a lógica de negócio da aplicação. É responsável por processar as regras do domínio e pode tanto consumir outras funções de serviço quanto interagir com a camada de persistência.
- **Repository:** Trata das operações relacionadas à persistência de dados. Atua como uma interface entre os serviços e os mecanismos de armazenamento, como bancos de dados relacionais ou caches, promovendo o desacoplamento da lógica de negócio em relação à camada de dados.

4.3.2 Diagrama da arquitetura

Esta subseção apresenta dois diagramas que representam a arquitetura do sistema desenvolvido: o diagrama de componentes e o diagrama de implantação. Esses diagramas auxiliam na visualização do relacionamento entre as partes da aplicação, bem como a sua distribuição nos ambientes computacionais.

4.3.2.1 Diagrama de componentes

O diagrama de componentes representa a organização lógica dos principais módulos da aplicação, evidenciando as dependências e as formas de comunicação entre eles. Ele demonstra como os componentes interagem por meio de interfaces — que definem contratos de uso — e implementações — que oferecem as funcionalidades esperadas.

Esse tipo de diagrama é útil para visualizar a estrutura modular da aplicação, facilitando o entendimento da separação de responsabilidades e da reutilização de código, além de apoiar decisões relacionadas à manutenção e evolução do sistema.

No diagrama de componentes proposto, a aplicação é composta por diversos módulos que representam funcionalidades distintas e organizadas de forma modular. Esses componentes são responsáveis por encapsular regras de negócio e oferecer interfaces bem definidas para comunicação entre si. A seguir, são descritos os principais módulos do sistema:

- **Auth:** Componente responsável pela autenticação de usuários e integração com serviços externos, como o Google OAuth. Garante um processo facilitado de login para os usuários e autenticação no acesso das funcionalidades protegidas da aplicação.

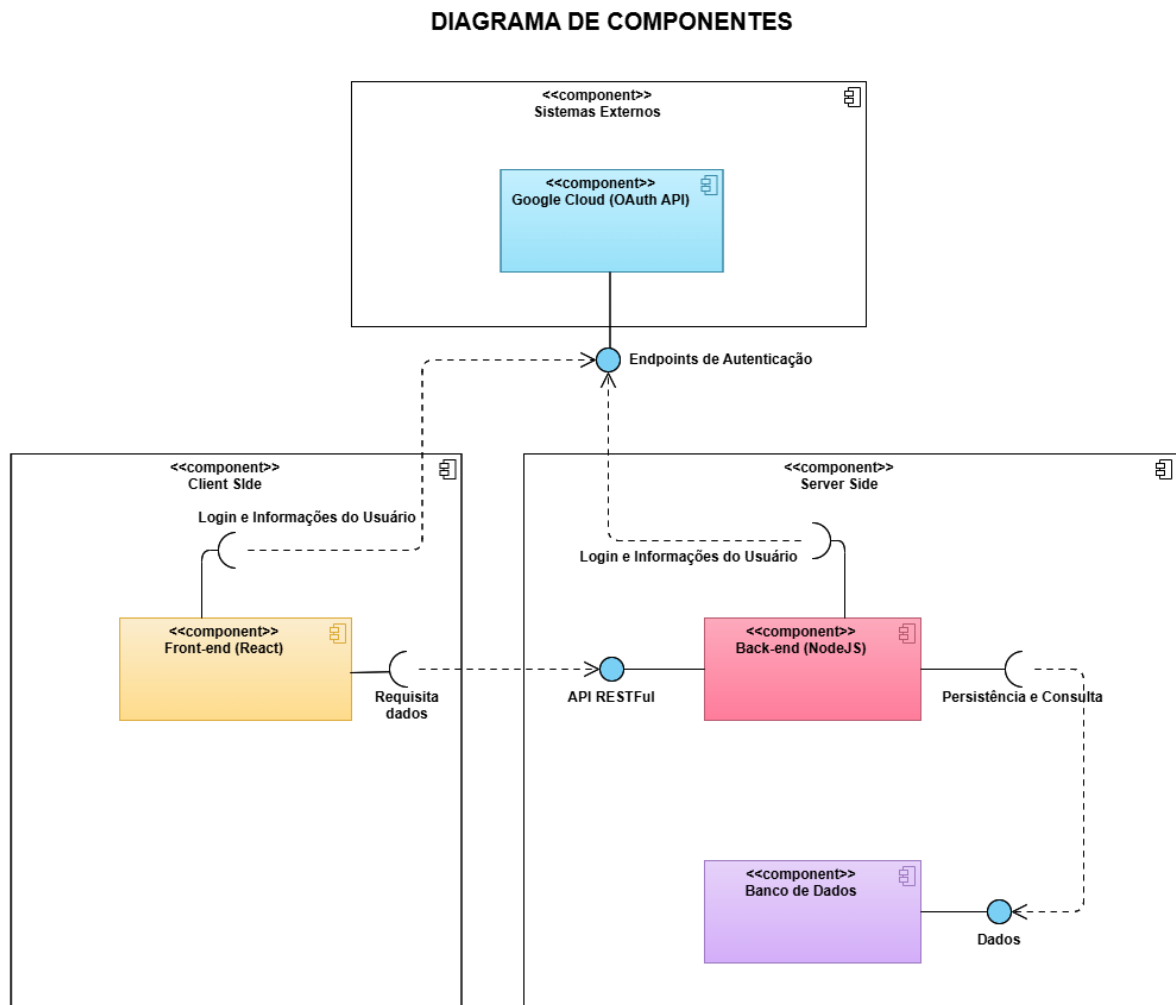


Figura 10 – Diagrama de componente da aplicação

- **Analytics:** Responsável por gerar relatórios e fornecer estatísticas baseadas nas informações do sistema. Auxilia principalmente no acompanhamento de desempenho da plataforma por parte dos gerentes do salão.
- **Appointments:** Gerencia os agendamentos realizados pelos clientes. Engloba tanto a criação, listagem e atualização dos agendamentos quanto a associação com os serviços ofertados.
- **Customers:** Controla os dados relacionados aos clientes da plataforma. Permite o cadastro, consulta e edição de informações do perfil dos usuários.
- **Employees:** Administra os dados dos profissionais que prestam serviços na aplicação, incluindo informações cadastrais, disponibilidade e associação a serviços específicos.
- **Notifications:** Responsável por enviar notificações aos usuários, como lembretes, confirmações de agendamento e atualizações importantes. Pode incluir o envio por e-mail ou outros canais.

- **Offers:** Define a relação entre profissionais e os serviços que eles oferecem. Cada oferta especifica o tempo estimado e o valor cobrado por um funcionário para realizar determinado serviço. Esse componente é fundamental para a composição de um agendamento, pois determina quais combinações de profissional e serviço estão disponíveis.
- **Services:** Representa os serviços oferecidos pela empresa, armazenando informações descritivas como nome e descrição. Este módulo não define valores ou tempos de execução, pois esses dados são especificados nas ofertas individuais de cada profissional (por meio do módulo *Offers*).
- **Shift:** Trata do controle de turnos de trabalho dos funcionários, possibilitando a definição de horários disponíveis para realização dos agendamentos.
- **Rating:** Permite que os clientes avaliem os serviços e os profissionais após os atendimentos, promovendo um sistema de feedback contínuo.
- **Role e Permissions:** Implementa o controle de acesso baseado em papéis, estabelecendo diferentes níveis de permissão conforme o tipo de usuário (cliente, funcionário, administrador, etc.).
- **Static Files (Front-end):** Componente responsável por servir os arquivos estáticos da interface do usuário, gerados após o processo de *build* do projeto front-end (em React). Inclui arquivos HTML, CSS e JavaScript que são entregues ao navegador do usuário final via servidor NGINX.

4.3.2.2 Diagrama de Implantação

O diagrama de implantação mostra como os componentes do sistema estão distribuídos em termos de infraestrutura, seja em servidores físicos ou ambientes virtuais. Ele ajuda a entender onde cada parte da aplicação está rodando, como os serviços se conectam entre si e quais recursos são necessários para que tudo funcione bem em produção.

Esse tipo de representação é especialmente útil para quem for implantar ou manter o sistema, pois facilita a visualização de elementos como servidores, banco de dados, gateways de rede, e outras dependências da aplicação. Além disso, o diagrama contribui para o planejamento de permissões, acessos e políticas de segurança que precisam ser configuradas na infraestrutura.

O diagrama acima é composto pelos seguintes componentes:

- **User Device:** No diagrama proposto, por se tratar de uma aplicação web, o dispositivo do usuário será responsável por executar a aplicação *client-side*, que interpreta através do navegador os arquivos CSS, JavaScript e HTML gerados no

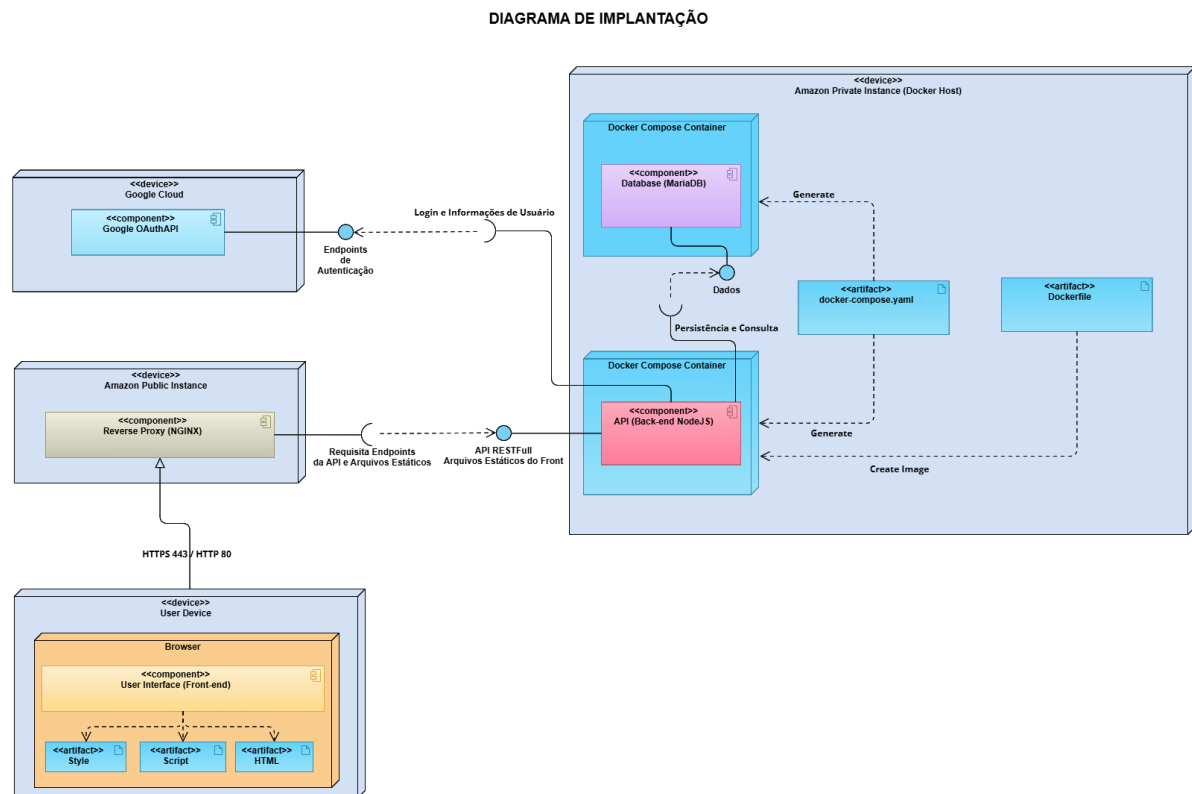


Figura 11 – Diagrama de implantação da aplicação

empacotamento ou *build* do projeto feito com a biblioteca React. Ademais, esse *device* é composto por alguns artefatos importantes que são obtidos pelo navegador por meio de uma requisição ao *Proxy Reverso*: CSS (style), JavaScript (script) e HTML.

- **Amazon Public Instance (Device):** A *Amazon Public Instance* é uma instância EC2 que possui um IP público, o que permite que ela seja acessada diretamente pelo usuário ou resolvida por DNS. Por esse motivo, nela são executados apenas os componentes que devem estar disponíveis publicamente ao cliente final:
 - **Componente NGINX:** Serviço que atua como *Proxy Reverso* para a aplicação que está sendo executada em uma instância privada na arquitetura proposta. Para viabilizar conexões HTTPS, o NGINX utiliza certificados digitais emitidos gratuitamente pelo serviço Let's Encrypt, por meio da ferramenta Certbot, que automatiza todo o processo de emissão e renovação dos certificados (FOUNDATION, 2025). Este serviço é instalado como um **artefato** adicional no ambiente da instância pública, sendo integrado ao próprio ciclo de configuração e inicialização do NGINX.
- **Amazon Private Instance (Device):** Por outro lado, a *Amazon Private Instance* é composta por uma instância EC2 com restrições de rede, o que significa que seu

acesso é limitado à rede interna e não possui IP público. Nessa instância, componentes da arquitetura que não precisam estar disponíveis de forma pública são o caso de uso perfeito, uma vez que garante maior segurança e isolamento dos aspectos internos da aplicação. Em seu interior, ela é composta pelos seguintes componentes:

- *API (Back-end em NodeJS)*: Principal serviço da aplicação, sendo o responsável por se prover os "endpoints" que fornecem os dados e arquivos estáticos para o *front-end* através de uma *API RestFull*, se comunicando com o banco de dados, um componente que é executado no mesmo device.
- *Banco de Dados (SGBD MariaDB)*: Serviço de *SGBD* que provê os dados para o *back-end* da aplicação, o que possibilita a persistência e consulta de forma eficiente. Para garantir a persistência dos dados gerenciados pelo MariaDB, o contêiner utiliza volumes Docker montados na instância EC2 privada. Isso assegura que os dados não sejam perdidos em reinicializações do contêiner.

Além dos serviços em execução, a instância privada também contém os seguintes artefatos essenciais para o empacotamento e execução dos serviços via contêineres Docker:

- *Dockerfile*: Esse artefato descreve as instruções necessárias para criar a imagem Docker da aplicação *back-end*, especificando o ambiente base (como a imagem do Node.js), os arquivos a serem copiados, dependências a serem instaladas e os comandos de inicialização da aplicação.
- *docker-compose.yml*: Esse arquivo é utilizado como ferramenta de orquestração para os serviços Docker da aplicação. Ele define a configuração dos contêineres da aplicação, como o contêiner da API e o do banco de dados, bem como as variáveis de ambiente, volumes, redes e dependências entre os serviços. É a partir deste artefato que os contêineres são gerados e executados de forma integrada.
- **Google Cloud (Device)**: Localizada na nuvem pública da Google (*Google Cloud*), essa API é utilizada pelo back-end da aplicação para realizar a autenticação de usuários através do protocolo OAuth 2.0 ([Google Developers, 2025](#)). Esse processo ocorre quando o usuário opta por fazer login com sua conta Google. Nesse cenário, a aplicação redireciona o usuário para a tela de autenticação da Google, e após a confirmação, a API recebe um *token* de acesso que é utilizado para obter as informações do usuário autenticado. Esse fluxo garante uma autenticação segura, delegando a responsabilidade da validação de identidade à Google.

O fluxo de execução típico da aplicação baseado no diagrama de implantação acima segue os seguintes passos:

1. O usuário acessa a aplicação pelo navegador, requisitando os arquivos HTML/CSS/JS ao servidor NGINX.
2. O NGINX, atuando como proxy reverso, redireciona essas requisições para o serviço de back-end na instância privada.
3. A API processa a requisição, acessa o banco de dados quando necessário e retorna os dados.
4. Em caso de autenticação via Google, a API redireciona o usuário para o serviço Google OAuth, que retorna um token de acesso após o login.
5. Esse token é utilizado pela API para obter os dados do usuário autenticado e estabelecer uma sessão.

Além da organização dos componentes do diagrama, a arquitetura também prioriza a segurança da comunicação e do acesso. O tráfego entre o navegador do usuário e a instância pública é realizado por meio do protocolo HTTPS, o que garante a confidencialidade e a integridade dos dados transmitidos. Para isso, foi utilizado o serviço gratuito de certificação digital Let's Encrypt em conjunto com a ferramenta Certbot, que automatiza a emissão, renovação e instalação dos certificados TLS no servidor NGINX.

Internamente, a comunicação entre o NGINX e os serviços da instância privada ocorre por meio de regras específicas de segurança definidas na VPC, utilizando mecanismos como *Security Groups* e *Route Tables*. Isso reduz significativamente a superfície de ataque da aplicação e assegura uma camada adicional de proteção para os aspectos internos.

4.3.2.3 Diagrama de Referência na AWS

Com o objetivo de fornecer uma visão mais aprofundada da infraestrutura da aplicação na nuvem, o diagrama apresenta a disposição dos principais componentes implantados na arquitetura da Amazon Web Services (AWS).

Este diagrama ilustra elementos de infraestrutura fundamentais como sub-redes públicas e privadas, resolução de DNS, Virtual Private Cloud (VPC), Bastion Server, NAT Gateway, Internet Gateway, banco de dados, entre outros recursos. A representação facilita a compreensão técnica da topologia de rede e da distribuição dos serviços, evidenciando como a aplicação foi projetada para atender requisitos de segurança, escalabilidade e disponibilidade no ambiente da AWS. A seguir, descreve-se brevemente cada elemento presente no diagrama.

- **VPC (Virtual Private Cloud):** A aplicação opera dentro de uma VPC personalizada com o bloco CIDR 10.0.0.0/16, que abriga duas sub-redes: uma pública

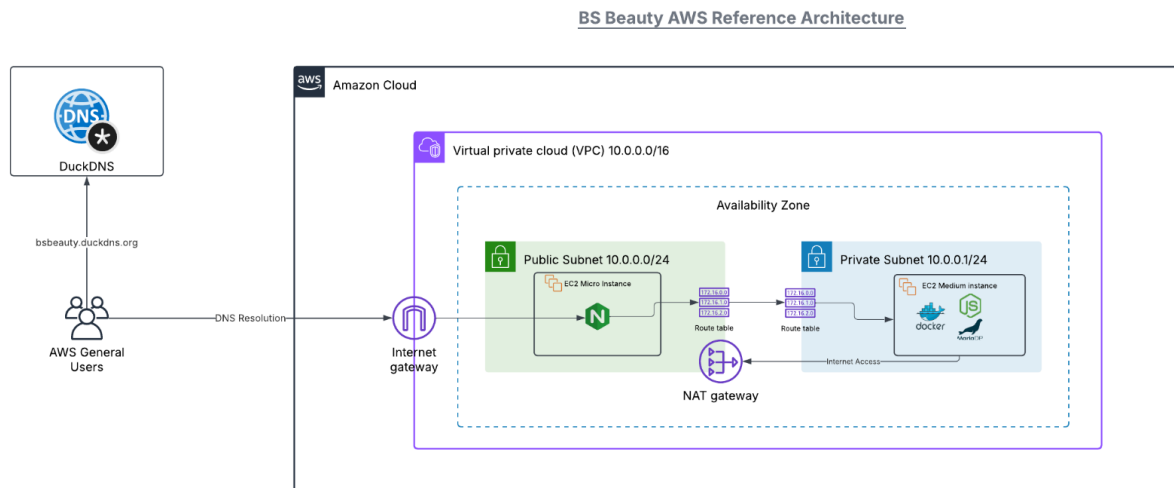


Figura 12 – Diagrama Geral da Arquitetura

e outra privada, seguindo o princípio de segmentação de rede recomendado pela própria AWS ([Amazon Web Services, 2025](#)).

- **Sub-rede pública (10.0.0.0/24):** Contém uma instância EC2 de pequeno porte (`t2.micro`) que executa o serviço NGINX. Esse servidor atua como proxy reverso, roteando as requisições provenientes da internet para os serviços internos hospedados em uma sub-rede privada.
- **Sub-rede privada (10.0.0.1/24):** Hospeda uma instância EC2 de médio porte (`t2.medium`), na qual são executados os contêineres da aplicação via Docker Compose, incluindo o serviço de back-end (Node.js) e o banco de dados relacional MariaDB.
- **NAT Gateway:** Permite que os recursos da sub-rede privada (como a instância EC2 que executa os contêineres) realizem atualizações e acessos à internet de forma segura, sem que sejam diretamente acessíveis externamente.
- **Internet Gateway:** Responsável por permitir o tráfego de entrada e saída entre a VPC e a internet pública. Está associado à sub-rede pública e NAT Gateway, permitindo que o NGINX receba requisições externas e o NAT receba um IP.
- **DuckDNS:** Utilizado como serviço de DNS dinâmico gratuito, permitindo que a aplicação seja acessada por um domínio estável (`bsbeauty.duckdns.org`), mesmo que o endereço IP público da instância EC2 varie. A resolução de nome é feita de forma transparente para o usuário final, facilitando o acesso à aplicação.
- **Usuários externos (AWS General Users):** Representam os clientes que acessam a aplicação via navegador. O tráfego HTTP/HTTPS chega inicialmente ao serviço NGINX na sub-rede pública, que encaminha as requisições para a instância privada onde os serviços da aplicação estão efetivamente em execução.

Essa separação entre sub-rede pública e privada visa seguir boas práticas de segurança e isolamento de ambiente, recomendadas tanto pela documentação oficial da AWS quanto por autores renomados da área de arquitetura de software em nuvem, como em ([Amazon Web Services, 2025](#)). Ao manter os serviços internos em uma sub-rede privada, reduz-se a superfície de ataque da aplicação e melhora a resistência contra acessos não autorizados.

Além disso, a utilização do DuckDNS simplifica a exposição da aplicação para o ambiente externo sem a necessidade de configurar manualmente um serviço de DNS ou pagar por domínios personalizados, o que se alinha aos objetivos de custo deste projeto.

4.4 Tecnologias e Ferramentas

-> 4.4

4.4.1 Front-End

-> 4.4.1

4.4.1.1 React

-> 4.4.1.1 Biblioteca JavaScript para construção de interfaces de usuário com foco em componentes reutilizáveis e reatividade.

4.4.1.2 TailwindCSS

-> 4.4.1.2 Framework de CSS utilitário que permite a criação rápida de layouts e estilizações diretamente nas classes HTML.

4.4.1.3 Redux e RTK Query

-> 4.4.1.3 Biblioteca para gerenciamento de estado global no React, com RTK Query oferecendo funcionalidades de cache e requisições assíncronas de forma otimizada.

4.4.2 Back-End

-> 4.4.2

4.4.2.1 NodeJS

-> 4.4.2.1 Ambiente de execução JavaScript no lado do servidor, baseado no motor V8 do Chrome, ideal para aplicações escaláveis e em tempo real.

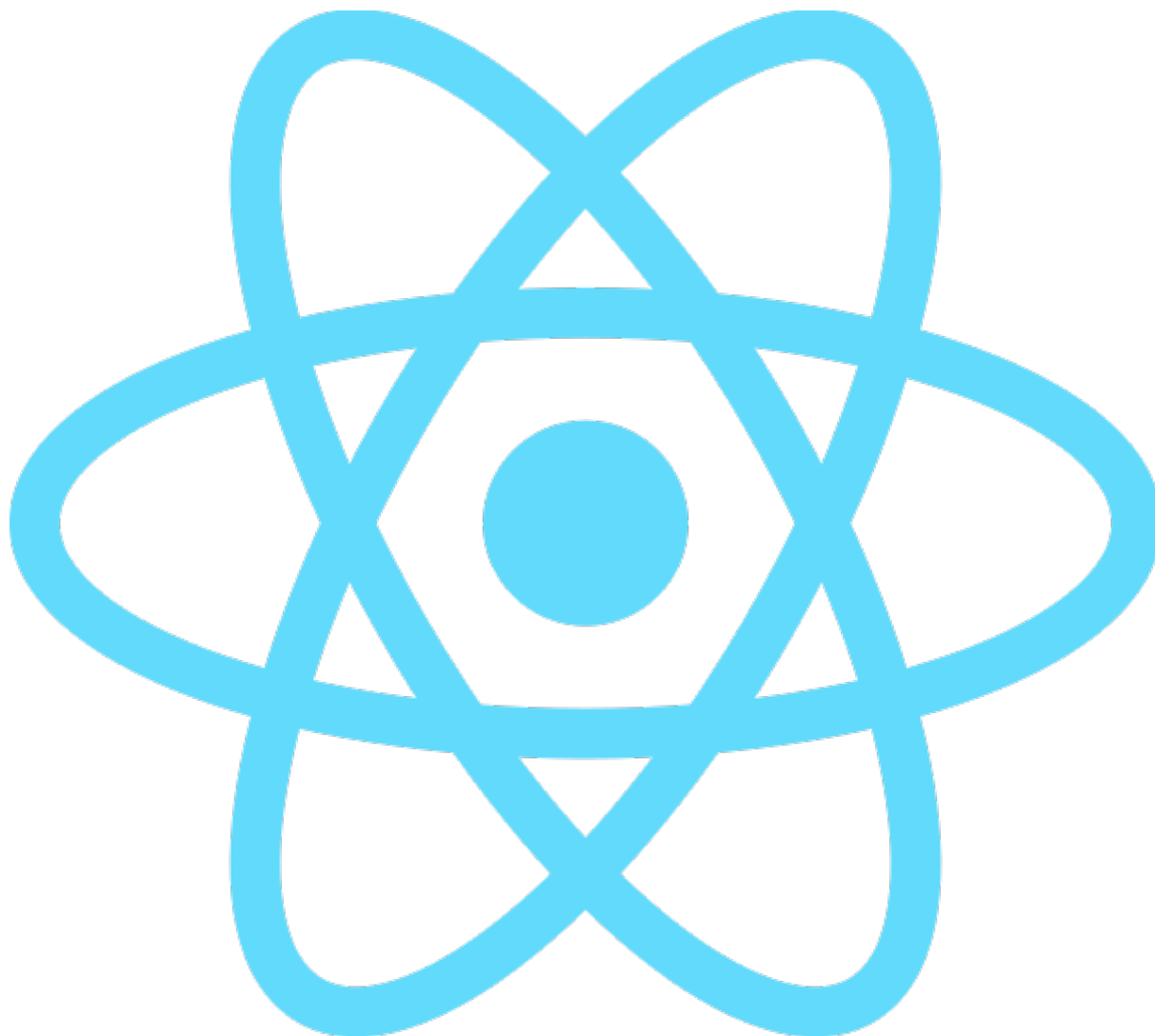


Figura 13 – Logo do React

4.4.2.2 Express

-> 4.4.2.2 Framework minimalista para Node.js que facilita a criação de APIs e rotas HTTP de maneira simples e eficiente.

4.4.3 Infraestrutura

-> 4.4.3

4.4.3.1 Docker

-> 4.4.3.1 Plataforma de containers que permite empacotar aplicações e suas dependências de forma isolada e reproduzível.

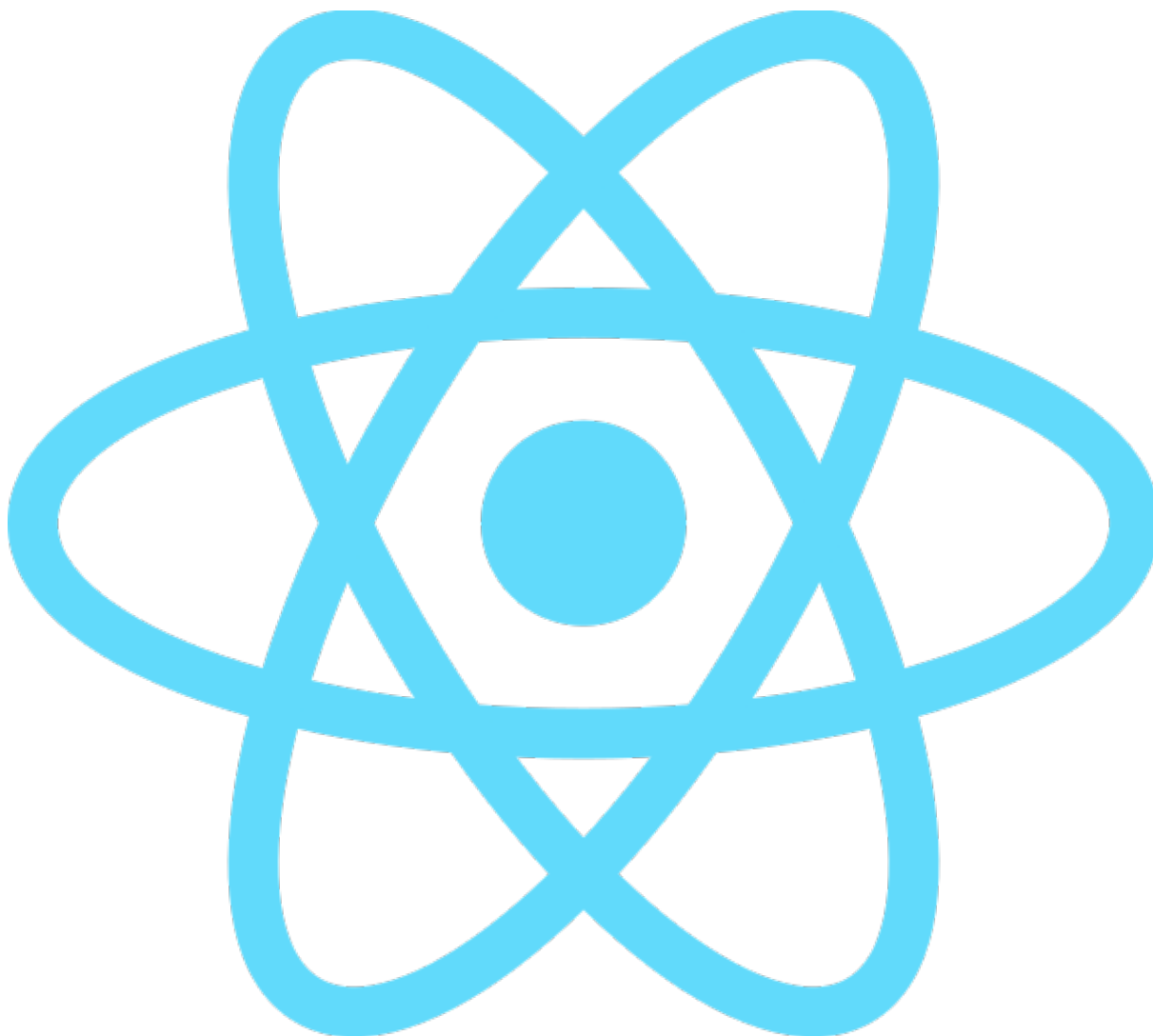


Figura 14 – Logo do React

4.4.3.2 Amazon Web Services (AWS)

-> 4.4.3.2 Conjunto de serviços de computação em nuvem utilizados para hospedagem, armazenamento, e orquestração da infraestrutura.

4.4.3.3 Banco de Dados MariaDB

-> 4.4.3.3 Sistema de gerenciamento de banco de dados relacional, compatível com MySQL, usado para armazenamento persistente e estruturado de dados.

4.4.4 Qualidade de Software e Testes

-> 4.4.4

4.4.4.1 SonarQube

-> 4.4.4.1 Ferramenta para análise contínua da qualidade do código, identificando problemas como bugs, vulnerabilidades e code smells.

4.4.4.2 Vitest

4.4.4.2 Framework moderno de testes para aplicações JavaScript e TypeScript, com integração nativa ao ecossistema do Vite.

4.5 Testes e Manutenibilidade

Nessa seção do capítulo, apresenta-se os mecanismos e ferramentas adotados para garantir a qualidade de software do projeto ao longo do desenvolvimento. Será abordado o plano de testes, assim como cada teste que está incluso. Além disso, assuntos como infraestrutura de testes e convenções de código (*coding convention*) serão detalhados, evidenciando práticas que promovem a manutenibilidade, padronização e confiabilidade do sistema ao longo de sua evolução.

4.5.1 Plano de Testes

O plano de testes define a estratégia adotada para garantir a qualidade e confiabilidade da aplicação. Ele inclui os tipos de testes aplicados, as ferramentas utilizadas, o escopo das validações, e os critérios de sucesso e falha.

Tendo em vista que a arquitetura do *back-end* é constituída por *controllers*, *services* e *repositories* usando o framework *Express*, é necessário garantir um ótimo funcionamento da comunicação entre essas camadas. Portanto, conforme os recursos da *RESTful API* são desenvolvidos (agendamentos, serviços de beleza, clientes, funcionários, etc) com as respectivas camadas que foram comentadas, urge-se a demanda de serem testadas em paralelo, cobrindo os cenários possíveis de sucesso/falha.

Pretende-se atingir, no mínimo, 80% de cobertura nos testes unitários e integrados aplicados sobre as camadas de *controllers* e *services* da *API*. Essa métrica será obtida por meio de ferramentas integradas ao ambiente de testes, como o framework **Vitest** (VITEST, 2025) com suporte a geração de relatórios. Embora a cobertura de testes não garanta por si só a ausência de falhas, ela serve como um forte indicador de que a maior parte da lógica de negócio está sendo exercitada e validada durante a execução dos testes. Essa prática contribui diretamente para a robustez do sistema e facilita a detecção precoce de regressões ao longo do desenvolvimento.

O mesmo propósito de cobertura de testes é válido para o *front-end* desenvolvido em *React*. Como essa tecnologia adota um princípio de componentização de elementos da

interface, é interessante que as páginas com cenários lógicos mais críticos (como as telas de agendamento) sejam validadas de forma precisa. Garantindo que os elementos da interface estejam atendendo o comportamento esperado.

Quanto à cobertura de testes do *front-end*, é tido como propósito, realizar uma cobertura de testes que envolva todos os processos que foram mapeados no escopo do projeto.

Ademais, os testes desenvolvidos nas duas divisões do sistema terão foco, inicialmente, nas funcionalidades essenciais para o funcionamento do sistema, tais como autenticação, criação e gerenciamento de agendamentos, relatórios, serviços e ofertas. Essa abordagem inicial permite validar as principais regras de negócio desde as primeiras entregas, contribuindo para uma base de código mais robusta e confiável à medida que novas funcionalidades forem sendo integradas.

Os arquivos contendo as classes/funções de testes devem estar localizados em diretórios específicos, adotando uma nomenclatura compreensível como `analytics.controller.spec.unit.ts` (referenciando um teste unitário) e `analytics.controller.spec.integration.ts` referenciando um teste integrado).

4.5.2 Análise Estáticas

A análise estática de código é realizada utilizando as ferramentas **SonarQube** (SONARQUBE, 2025) e **ESLint** (ESLINT, 2025), que permitem detectar problemas de qualidade, como vulnerabilidades, *bugs* e código duplicado, sem a necessidade de executar a aplicação. Essa etapa ajuda a manter o código limpo, seguro e sustentável ao longo do tempo.

4.5.3 Testes funcionais

Os testes funcionais baseiam-se nas especificações dos requisitos do sistema. O principal propósito, portanto, é validar se uma determinada funcionalidade solicitada foi desenvolvida com êxito, e está atendendo as expectativas esperadas dentro do sistema. Dessa forma, serão abordados nessa subseção os testes unitários, de integração e *end-to-end*.

4.5.3.1 Testes Unitários

Os testes unitários servem para validar o comportamento de funções e componentes isolados utilizando *mocks* de outras partes que interagem com o que está sendo testado.

Os arquivos de testes unitários são organizados em diretórios específicos com uma nomenclatura padronizada. Um exemplo de nome de arquivo seria `appointments.controller.spec.unit.ts`, indicando que se trata de um teste unitário do módulo de

agendamentos. Essa convenção visa facilitar a identificação, localização e manutenção dos testes ao longo do tempo.

4.5.3.2 Testes de Componente

Os testes de componente têm como objetivo validar o comportamento isolado de componentes funcionais da aplicação, garantindo que cada unidade seja capaz de cumprir sua função específica conforme o esperado. No contexto do projeto, esse tipo de teste está especialmente presente no *front-end*, desenvolvido em *React*, onde a arquitetura baseada em componentes torna natural a aplicação dessa abordagem.

Assim como os demais testes, os arquivos que contêm testes de componentes seguem uma convenção de nomenclatura voltada à organização e fácil localização no projeto. Um exemplo seria `bookingForm.spec.component.tsx`, indicando que se trata de um teste de componente da interface de agendamento. Essa padronização é importante para manter a coesão do código de testes e permitir sua escalabilidade conforme o sistema cresce.

4.5.3.3 Testes de Integração

Os testes de integração verificam a interação entre módulos e componentes da aplicação de acordo com uma determinada funcionalidade que precisa ser desenvolvida.

Assim como os testes unitários, os testes integrados seguem uma convenção de nomenclatura para manter a organização do projeto. Um exemplo de nome de arquivo seria `services.controller.spec.integration.ts` que sinaliza tratar-se de um arquivo de testes vinculado ao módulo de serviços. Essa padronização contribui para a clareza estrutural do projeto de testes.

4.5.3.4 Testes *end-to-end*

Os testes *end-to-end* representam provações feitas através de cenários que validam o sistema completamente desde a interface gráfica até a lógica interna da API.

No atual momento do desenvolvimento, esse tipo de teste será usado para validar, especialmente, os processos levantados nas premissas do projeto, devido à alta complexidade de configuração e lentidão de execução que possui.

4.5.4 Testes não funcionais

Testes não funcionais estão fora do escopo do projeto no atual desenvolvimento.

4.5.4.1 Testes de performance

Testes de performance estão fora do escopo do projeto no atual desenvolvimento.

4.5.4.2 Testes de carga

Testes de carga estão fora do escopo do projeto no atual desenvolvimento.

4.5.4.3 Testes de configuração

Testes de configuração estão fora do escopo do projeto no atual desenvolvimento.

4.5.5 Testes automatizados

Os testes automatizados desempenham um papel fundamental na garantia da qualidade do sistema ao longo de seu ciclo de vida. Eles são responsáveis por verificar continuamente o correto funcionamento das funcionalidades implementadas, promovendo confiança no código e permitindo a identificação precoce de regressões.

A execução dos testes automatizados está ligada ao fluxo de integração contínua por meio da ferramenta **GitHub Actions** (GITHUB, 2025b). Essa integração visa garantir que o código entregue atenda a padrões mínimos de qualidade e estabilidade em todas as etapas do desenvolvimento. A cada *push* ou *pull request*, fluxos automatizados são executados para validar o código por meio de testes automatizados, análise estática, e verificação de cobertura. Esse processo assegura que apenas alterações estáveis e em conformidade com os padrões de qualidade sejam incorporadas à base de código principal, promovendo entregas seguras e contínuas ao longo do ciclo de desenvolvimento.

4.5.6 Logs

4.5.7 Code Convention

Para garantir a legibilidade e padronização do código, são adotadas convenções definidas com base em boas práticas da comunidade *JavaScript/TypeScript*. Essas diretrizes ajudam a manter o código uniforme entre os diferentes desenvolvedores do time, reduzindo ambiguidades e facilitando o entendimento do sistema como um todo.

As principais práticas adotadas incluem:

- **Ferramentas de *Linting* e Formatação:**

- Utilização do **ESLint** (ESLINT, 2025) para garantir padrões de estilo e detectar possíveis erros ou práticas inadequadas de codificação.
- Uso do **Prettier** (PRETTIER, 2025) para formatação automática do código, assegurando que todos os arquivos mantenham a mesma estrutura visual (espaçamento, quebras de linha, indentação, etc).

- **Padrões de Nomenclatura:**

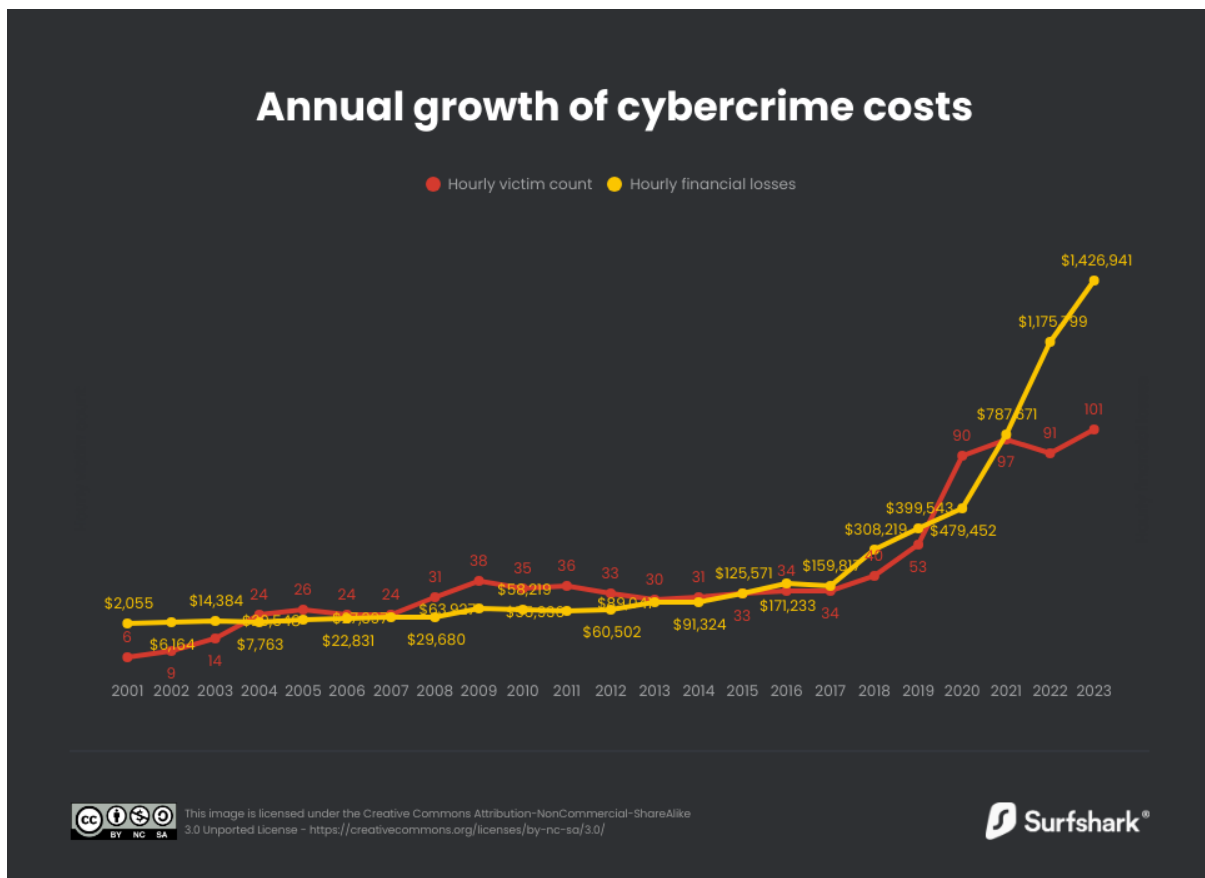
- Uso de `camelCase` para variáveis e funções.
- Uso de `PascalCase` para componentes e classes.
- Uso de `UPPER_SNAKE_CASE` para constantes globais.
- **Organização do Código:**
 - Estrutura modular com separação clara entre camadas (*controllers*, *services*, *repositories*).
 - Agrupamento de arquivos por domínio funcional.
- **Boas Práticas:**
 - Escrita de código limpo e legível, evitando duplicações.
 - Utilização de comentários apenas quando necessário, priorizando nomes autoexplicativos.
- **Revisões e Padronização em Equipe:**
 - Adoção de *pull requests* com as devidas descrições das funcionalidades desenvolvidas.
 - Documentação e comunicação clara de decisões técnicas relevantes.

4.6 Segurança, Privacidade e Legislação

Atualmente, há uma crescente no número de usuários utilizando dispositivos conectados a *Internet*, o que, por um lado mostra que essa tecnologia tão importante está sendo difundida a todas as camadas sociais, mas por outro, gera preocupação quanto as implicações do uso inadvertido das redes. Nesse contexto, tornou-se comum pessoas mal intencionadas que usam da ignorância de alguns para cometer ataques digitais, prejudicando ou tomando vantagem de pessoas, grupos ou organizações. A afirmativa anterior é confirmada por um gráfico (Figura 15) elaborado pela Surfshark (SURFSHARK, 2023), empresa provedora de serviços de VPN, que analisa o crescimento anual de custos em cibersegurança.

Por isso surgiram leis e normas que regularizam como os dados devem ser tratados e também tecnologias que auxiliam a proteger os dois lados da comunicação: os clientes, que desejam ter seus dados protegidos, e das organizações, que precisam se certificar da identidade do usuário. Isto posto, serão abordadas as técnicas e metodologias adotadas a fim de garantir que o software desenvolvido atenda as demandas da legislação e promova segurança e privacidade a seus utilizadores.

Figura 15 – Crescimento anual de custos com cibercrime



Fonte: Surfshark

4.6.1 Critérios de segurança e privacidade

Na aplicação desenvolvida, foram definidas formas de manter a segurança de todas as partes envolvidas. Foi implementado um método de cadastro e *login* facilitado com geração de *tokens* e também uma infraestrutura de rede que protege o dispositivo que armazena o banco de dados da aplicação.

4.6.1.1 Cadastro e login com conta Google

Um dos requisitos da entidade parceira, era o desenvolvimento de um sistema de cadastro e login facilitados, haja vista que muitos dos clientes tinham dificuldade em acessar a aplicação anterior por constantemente esquecerem suas credenciais (e-mail e senha). Considerando essa dificuldade, foi implementado um sistema de registro e login utilizando a API de autenticação OAuth da Google, pelo fato da maioria dos dispositivos no Brasil apresentarem sistema operacional Android, conforme destaca uma pesquisa (APPMYSITE, 2025), que geralmente requerem uma conta Google para seu funcionamento. E mesmo indivíduos com aparelhos de outro sistema operacional, comumente possuem contas Google para usufruir de seus serviços. Assim sendo, a responsabilidade de identificar os usuários

da aplicação foi terceirizada para a Google, e quando estes se cadastram, devem aceitar sua políticas e termos de usuário que definem extensamente como os dados são processados, tratados e protegidos.

4.6.1.2 Infraestrutura de Rede

Os dispositivos (máquinas virtuais) que sustentam a aplicação estão hospedados na AWS, logo sendo de sua inteira responsabilidade protegê-los fisicamente, como diz seu Modelo de Responsabilidade Compartilhada ([AWS, 2025](#)), porém no que tange a software e redes, cabe ao time de desenvolvimento proteger. Para evitar acessos indevidos aos sistema interno, criaram-se duas instâncias EC2, uma delas sendo pública e outra privada. Como já explicado, a instância privada não possui IP público, portanto só é possível acessá-la pela rede interna dentro da infraestrutura de rede criada, delimitando uma camada a mais de segurança. O acesso a essa instância é feito pela instância pública por meio do protocolo SSH, que possui seus próprios métodos de segurança com esquema de chaves de acesso.

4.6.2 Observância à Legislação

4.7 Modelo de Banco de Dados

4.7.1 Modelo Entidade Relacionamento - MER

4.7.2 Diagrama Entidade Relacionamento - DER

tabelas

4.7.3 Dicionário de Dados

4.8 Cronograma

pensar no projeto todo, não só MVP

4.8.1 Análise da Duração do Projeto

considerar o gerenciamento ágil

5 Viabilidade Financeira

mesmo usando uma hospedagem gratis (AWS), precisamos pesquisar uma paga para colocar na tabela de custos

5.1 Custos

5.2 Receitas

5.3 Cenário realista

5.4 Cenário Otimista

5.5 Cenário Pessimista

6 Considerações Finais

A aplicação web *BS Beauty* representa uma contribuição relevante para o mercado de serviços de beleza, sobretudo no modelo de *coworking*. Ao centralizar e digitalizar o agendamento e a gestão do fluxo de trabalho de diferentes profissionais autônomos, a aplicação reduz erros administrativos e melhora a experiência do cliente. Considerando que, segundo estudo do (SENAC-SP, 2022), até 30% do tempo dos pequenos empreendedores é consumido em tarefas manuais, a *BS Beauty* oferece um diferencial competitivo, permitindo que gestores e profissionais dediquem mais tempo ao atendimento do que às operações administrativas.

Durante o desenvolvimento da aplicação, a escolha da metodologia Scrum foi fundamental para o planejamento e a entrega do projeto. Os ciclos de *sprints* permitiram validar rapidamente cada funcionalidade junto à nossa parceira de extensão, garantindo flexibilidade na evolução de requisitos. Durante discussões sobre os requisitos, o módulo de pagamento online foi estrategicamente descartado do sistema, uma vez que a gestora optou por manter os pagamentos apenas de forma presencial. Além disso, o fluxo de agendamento foi inicialmente pensado como horário→profissional, porém, após sugestões do orientador, foi dividido em três caminhos distintos: agendamento apenas por horário, apenas por profissional ou de forma combinada. Essas mudanças só foram possíveis graças à flexibilidade do Scrum.

A comunicação com a gestora Bruna e a coordenação interna da equipe, apesar de bem-sucedidas, representaram desafios significativos. A necessidade de validações constantes das regras de negócio, aliada a conflitos de agenda, impossibilitou reuniões presenciais com todos os *stakeholders*. Por isso, grande parte das interações foi conduzida por mensagens de texto. Ferramentas como *Discord* e *Notion* foram essenciais para alinhar demandas, formalizar decisões e manter a coesão no código e na documentação.

Espera-se que, após a implantação, o sistema elimine conflitos de agenda, reduza a taxa de não-comparecimento por meio de lembretes automáticos e aumente a receita mensal em virtude da otimização da ocupação das estações, da clareza nos relatórios e da fidelização de clientes. Além disso, ao disponibilizar dashboards de performance e indicadores de satisfação, a *BS Beauty* criará uma base de dados estratégica para decisões futuras de *marketing* e expansão.

Em síntese, este projeto de extensão uniu teoria e prática, contribuindo não apenas para a formação de profissionais capacitados, mas também para a criação de um produto de alto impacto para o mercado de beleza, apoiando a consolidação e o crescimento dos *coworkings* de beleza por meio de uma solução digital eficiente.

Referências Bibliográficas

ABIHPEC. *Relatório de Gestão de Salões de Beleza*. 2021. Relatório ABIHPEC. Disponível em: <<https://abihpec.com.br/relatorio-gestao-saloes-2021>>. Acesso em: 01 jun 2025. Citado na página 18.

Amazon Web Services. *VPC Best Practices*. 2025. <https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Scenario2.html>. Acesso em: 05 junho 2025. Citado 2 vezes nas páginas 49 e 50.

APPMYSITE. *Android vs iOS: Mobile Operating System market share statistics (Updated 2025)*. 2025. Disponível em: <<https://www.appmysite.com/blog/android-vs-ios-mobile-operating-system-market-share-statistics-you-must-know/>>. Acesso em: 7 jun. 2025. Citado na página 58.

Avec Company. *Avec — plataforma de gestão e marketplace para negócios de beleza*. 2025. Site institucional da Avec. Disponível em: <<https://negocios.avec.app/>>. Acesso em: 25 maio 2025. Citado na página 22.

AWS. *O que é Scrum?* 2024. Disponível em: <<https://aws.amazon.com/pt/what-is/scrum/>>. Acesso em: 1 jun. 2025. Citado na página 28.

AWS. *Modelo de responsabilidade compartilhada*. 2025. Disponível em: <<https://aws.amazon.com/pt/compliance/shared-responsibility-model/>>. Acesso em: 7 jun. 2025. Citado na página 59.

Beauty Fair. *Coworkings de beleza: entenda o modelo e suas vantagens*. 2024. Blog “Negócios de Beleza” – Beauty Fair. Disponível em: <<https://negociosdebeleza.beautyfair.com.br/coworkings-de-beleza/>>. Acesso em: 27 maio 2025. Citado na página 14.

BOOKSY. *Global Consumer Trends Report*. 2022. Relatório Booksy. Disponível em: <<https://www.booksy.com/global-consumer-trends-2022>>. Acesso em: 01 jun 2025. Citado na página 17.

CONVENTIONAL COMMITS. *Conventional Commits*. 2025. Disponível em: <<https://www.conventionalcommits.org/pt-br/v1.0.0/>>. Acesso em: 1 jun. 2025. Citado na página 30.

DISCORD. *Discord*. 2025. Disponível em: <<https://discord.com>>. Acesso em: 31 mai. 2025. Citado na página 27.

ECOMMERCE NA PRÁTICA. *Mercado da beleza no Brasil 2025: tendências e números*. 2025. Disponível em: <<https://ecommercenapratica.com/blog/mercado-da-beleza/>>. Acesso em: 2025-06-01. Citado na página 17.

EQUIPE TOTVS. *Kanban: conceito, como funciona, vantagens e implementação*. 2023. Blog. Disponível em: <<https://www.totvs.com/blog/negocios/kanban/>>. Acesso em: 1 jun. 2025. Citado na página 29.

ESLINT. *ESLint · Find and fix problems in your JavaScript code*. 2025. Disponível em: <<https://eslint.org/>>. Acesso em: 6 jun. 2025. Citado 2 vezes nas páginas 54 e 56.

FGV. *Estudo de Produtividade em Pequenos Negócios*. 2020. Relatório FGV. Disponível em: <<https://fgv.br/estudo-produtividade-pequenos-negocios-2020>>. Acesso em: 01 jun 2025. Citado na página 18.

FOUNDATION, E. F. *Certbot - Get HTTPS for free*. 2025. <<https://certbot.eff.org/>>. Acesso em: 05 junho 2025. Citado na página 46.

Gazeta do Povo. *O que é um coworking de beleza?* 2023. Gazeta do Povo. Disponível em: <<https://www.gazetadopovo.com.br/conteudo-publicitario/spazio-bellezza-coworking/o-que-e-um-coworking-de-beleza/>>. Acesso em: 18 maio 2025. Citado na página 14.

Gendo Sistemas. *Gendo — Sistema de agendamento completo*. 2025. Site institucional da Gendo. Disponível em: <<https://www.gendo.com.br/>>. Acesso em: 25 maio 2025. Citado na página 20.

GIT. *Git*. 2025. Disponível em: <<https://git-scm.com>>. Acesso em: 1 jun. 2025. Citado na página 30.

GITHUB. *Clonar um repositório*. 2025. Disponível em: <<https://docs.github.com/pt/repositories/creating-and-managing-repositories/cloning-a-repository>>. Acesso em: 1 jun. 2025. Citado na página 31.

GITHUB. *Documentação do GitHub Actions*. 2025. Disponível em: <<https://docs.github.com/pt/actions>>. Acesso em: 6 jun. 2025. Citado na página 56.

GITHUB. *GitHub · Build and ship software on a single, collaborative platform*. 2025. Disponível em: <<https://github.com>>. Acesso em: 1 jun. 2025. Citado na página 30.

Google Developers. *Using OAuth 2.0 to Access Google APIs*. 2025. Documentação oficial – Google Developers. Disponível em: <<https://developers.google.com/identity/protocols/oauth2>>. Acesso em: 05 junho 2025. Citado na página 47.

MINDMINERS. *Pesquisa de Satisfação de Serviços Pessoais*. 2022. Relatório MindMiners. Disponível em: <<https://mindminers.com.br/pesquisa-satisfacao-servicos-2022>>. Acesso em: 01 jun 2025. Citado na página 18.

NOTION. *Notion*. 2025. Disponível em: <<https://www.notion.com/pt>>. Acesso em: 31 mai. 2025. Citado 2 vezes nas páginas 27 e 29.

OBJECTIVE. *Git Flow: como funciona e quais as vantagens desse fluxo de trabalho*. 2023. Disponível em: <<https://www.objective.com.br/insights/git-flow/>>. Acesso em: 1 jun. 2025. Citado na página 30.

PRETTIER. *What is Prettier?* 2025. Disponível em: <<https://prettier.io/docs/>>. Acesso em: 6 jun. 2025. Citado na página 56.

PROJECTLIBRE. *ProjectLibre Desktop*. 2025. Disponível em: <<https://www.projectlibre.com/projectlibre-desktop/>>. Acesso em: 31 mai. 2025. Citado na página 27.

Reservio. *Ferramenta de agendamento on-line vs. agendamento tradicional: Qual é o melhor para o seu negócio?* 2024. Acesso em 26 out. 2023. Disponível em: <<https://www.reservio.com/pt-br/blog/dicas-de-negocios/ferramenta-de-agendamento-on-line-vs-agendamento-tradicional>>. Acesso em: 2025-06-06. Citado na página 25.

SEBRAE. *Coworking de Beleza*. 2023. Disponível em: <<https://sebrae.com.br/sites/PortalSebrae/conteudos/posts/coworking-de-beleza,d409d5dc8e166810VgnVCM1000001b00320aRCRD>>. Acesso em: 2025-06-01. Citado na página 18.

SEBRAE. *Números mostram a pujança dos negócios de beleza*. 2023. Publicado em 25 ago. 2023. Disponível em: <<https://sebrae.com.br/sites/PortalSebrae/artigos/numeros-mostram-a-pujanca-dos-negocios-de-beleza,dc88327896a76810VgnVCM1000001b00320aRCRD>>. Acesso em: 2025-06-01. Citado na página 16.

SEBRAE. *Softwares e aplicativos facilitam e melhoram a gestão do salão*. 2023. Disponível em: <<https://sebrae.com.br/sites/PortalSebrae/artigos/softwares-e-aplicativos-facilitam-e-melhoram-a-gestao-do-salao,f06cac941b896810VgnVCM1000001b00320aRCRD>>. Acesso em: 2025-06-07. Citado na página 25.

SEBRAE RS. *Beleza em 2025: confira as tendências para o setor*. 2024. Blog Digital SEBRAE RS. Disponível em: <<https://digital.sebraers.com.br/blog/mercado/beleza-em-2025-confira-as-tendencias-para-o-setor/>>. Acesso em: 27 maio 2025. Citado na página 14.

SEBRAE SC. *Coworking de beleza*. 2025. Publicado em 07 mai. 2025. Disponível em: <<https://www.sebrae-sc.com.br/observatorio/relatorio-de-inteligencia/coworking-de-beleza>>. Acesso em: 2025-06-01. Citado na página 18.

SENAC. *Panorama do mercado da beleza*. 2023. PDF. Disponível em: <https://forumsetorial.senac.br/assets/images/panorama_mercado_beleza.pdf>. Acesso em: 2025-06-07. Citado 2 vezes nas páginas 17 e 18.

SENAC-SP. *Perfil do Empreendedor de Beleza*. 2022. Relatório SENAC-SP. Disponível em: <<https://www.sp.senac.br/perfil-empreendedor-beleza-2022>>. Acesso em: 01 jun 2025. Citado 2 vezes nas páginas 17 e 61.

SONARQUBE. *SonarQube Server 2025.3 Documentation*. 2025. Disponível em: <<https://docs.sonarsource.com/sonarqube-server/latest/>>. Acesso em: 7 jun. 2025. Citado na página 54.

SURFSHARK. *Cybercrime statistics*. 2023. Disponível em: <<https://surfshark.com/research/cybercrime-risks/statistics>>. Acesso em: 7 jun. 2025. Citado na página 57.

Trinks Tecnologia da Informação. *Plataforma de gestão para salões, barbearias e clínicas de estética*. 2025. Trinks Negócios. Disponível em: <<https://negocios.trinks.com/>>. Acesso em: 25 maio 2025. Citado na página 19.

VITEST. *Getting Started*. 2025. Disponível em: <<https://vitest.dev/guide/>>. Acesso em: 7 jun. 2025. Citado na página 53.

Apêndices

APÊNDICE A – exemplo 1

materiais desenvolvidos pelo próprio autor do trabalho

Anexos

ANEXO A – exemplo 1

matérias de outras fontes que não o próprio autor do trabalho