

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

58p. : 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

Errata

pagina opcional para descrever algum correção que deve ser feita no documento.

Folha	Linha	Onde se lê	Leia-se
1	10	auto-conclavo	autoconclavo

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, xx de xxx de 2025:

Marcelo Tavares
Orientador

Professor
Convidado 1

Professor
Convidado 2

São Paulo, SP - Brasil
2025

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

Agradecimentos

escrever agradecimentos (um paragrafo para cada?)

*“Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)*

Resumo

o resumo deve ressaltar o objetivo, o método, os resultados e as conclusões do documento. A ordem e a extensão destes itens dependem do tipo de resumo (informativo ou indicativo) e do tratamento que cada item recebe no documento original. O resumo deve ser precedido da referência do documento, com exceção do resumo inserido no próprio documento. As palavras-chave devem figurar logo abaixo do resumo, antecidas da expressão Palavras-chave:, separadas entre si por ponto e finalizadas também por ponto.

Palavras-chave: latex. abntex. editoração de texto.

abstract

This is the english abstract.

Keywords: latex. abntex. text editoration.

Lista de ilustrações

Figura 1 – Logo plataforma Trinks	21
Figura 2 – Logo plataforma Gendo	22
Figura 3 – Logo plataforma Avec	23
Figura 4 – QR Code do repositório da aplicação	31
Figura 5 – Página inicial do repositório	32
Figura 6 – Diagrama de componente da aplicação	37
Figura 7 – Diagrama de implantação da aplicação	39
Figura 8 – Diagrama Geral da Arquitetura	42
Figura 9 – Logo do React	44
Figura 10 – Logo do React	45

Lista de quadros

Quadro 1 – Comparação entre as plataformas concorrentes e a aplicação proposta	25
Quadro 2 – Membros e seus respectivos papéis	29
Quadro 3 – Membros e suas atividades	29
Quadro 4 – Papéis dos integrantes com base no Scrum	30

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	18
1.1	Objetivos	19
1.2	Problema e Solução Proposta	19
1.3	Justificativa	20
1.4	Análise da Concorrência	21
1.4.1	Trinks	21
1.4.2	Gendo	22
1.4.3	Avec	23
1.4.4	Quadro comparativo	24
2	REVISÃO DA LITERATURA	26
2.1	Histórico de sistemas gerenciadores de agendamento	26
2.2	Atualidade do assunto	27
2.3	Outros contextos do assunto (opcional)	27
3	GESTÃO DO PROJETO	28
3.1	Organização da Equipe	28
3.1.1	Responsabilidades / Papéis / Atividades	28
3.2	Metodologias de Gestão e Desenvolvimento	29
3.2.1	Scrum	29
3.2.1.1	Sprints	30
3.3	Repositório da Aplicação	30
3.3.1	Definição do repositório da aplicação	31
3.3.1.1	Link do repositório e especificações para acesso	31
4	DESENVOLVIMENTO DO PROJETO	33
4.1	Escopo do Projeto	33
4.1.1	Regras de Negócio	33
4.1.2	Requisitos Funcionais	35
4.1.3	Requisitos Não Funcionais	35
4.2	Histórias de Usuário	35
4.2.1	Descrição das Histórias de Usuário	35
4.3	Arquitetura	35
4.3.1	Definições da arquitetura	35
4.3.2	Diagrama da arquitetura	36
4.3.2.1	Diagrama de componentes	36

4.3.2.2	Diagrama de Implantação	38
4.3.2.3	Diagrama de Referência na AWS	41
4.4	Tecnologias e Ferramentas	43
4.4.1	Front-End	43
4.4.1.1	React	43
4.4.1.2	TailwindCSS	43
4.4.1.3	Redux e RTK Query	44
4.4.2	Back-End	44
4.4.2.1	NodeJS	44
4.4.2.2	Express	45
4.4.3	Infraestrutura	45
4.4.3.1	Docker	45
4.4.3.2	Amazon Web Services (AWS)	46
4.4.3.3	Banco de Dados MariaDB	46
4.4.4	Qualidade de Software e Testes	46
4.4.4.1	SonarQube	46
4.4.4.2	Vitest	46
4.5	Testes e Manutenibilidade	46
4.5.1	Plano de Testes	46
4.5.2	Infraestrutura de Testes	47
4.5.3	Análise Estáticas	47
4.5.4	Testes Automatizados	48
4.5.5	Logs	48
4.5.6	Code Convention	48
4.6	Segurança, Privacidade e Legislação	49
4.6.1	Critérios de segurança e privacidade	49
4.6.2	Observância à Legislação	49
4.7	Modelo de Banco de Dados	49
4.7.1	Modelo Entidade Relacionamento - MER	49
4.7.2	Diagrama Entidade Relacionamento - DER	49
4.7.3	Dicionário de Dados	49
4.8	Cronograma	49
4.8.1	Análise da Duração do Projeto	49
5	VIABILIDADE FINANCEIRA	50
5.1	Custos	50
5.2	Receitas	50
5.3	Cenário realista	50
5.4	Cenário Otimista	50
5.5	Cenário Pessimista	50

6	CONSIDERAÇÕES FINAIS	51
6.1	Dificuldades, escolhas e Descartes	51
	REFERÊNCIAS BIBLIOGRÁFICAS	52
	APÊNDICES	55
	APÊNDICE A – EXEMPLO 1	56
	ANEXOS	57
	ANEXO A – EXEMPLO 1	58

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). 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).

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.

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).

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 1 – 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 2 – 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:

- 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 3 – 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

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.

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

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^o” 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 *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

explicar os tópicos principais do projeto

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, então, 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 (??).

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 do assunto

2.3 Outros contextos do assunto (opcional)

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, 2025b) 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 4) 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 4 – QR Code do repositório da aplicação



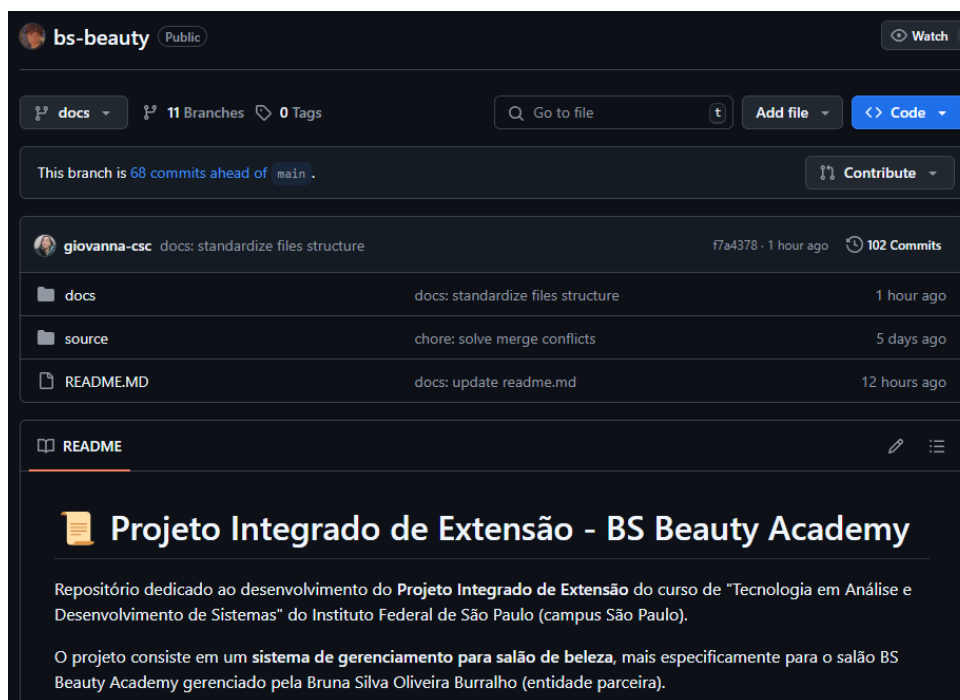
Fonte: Produzido pelos autores

Acessado o repositório, será aberta uma guia no navegador contendo uma página semelhante à Figura 5. 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 5 – 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

4.1.2 Requisitos Funcionais

4.1.3 Requisitos Não Funcionais

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

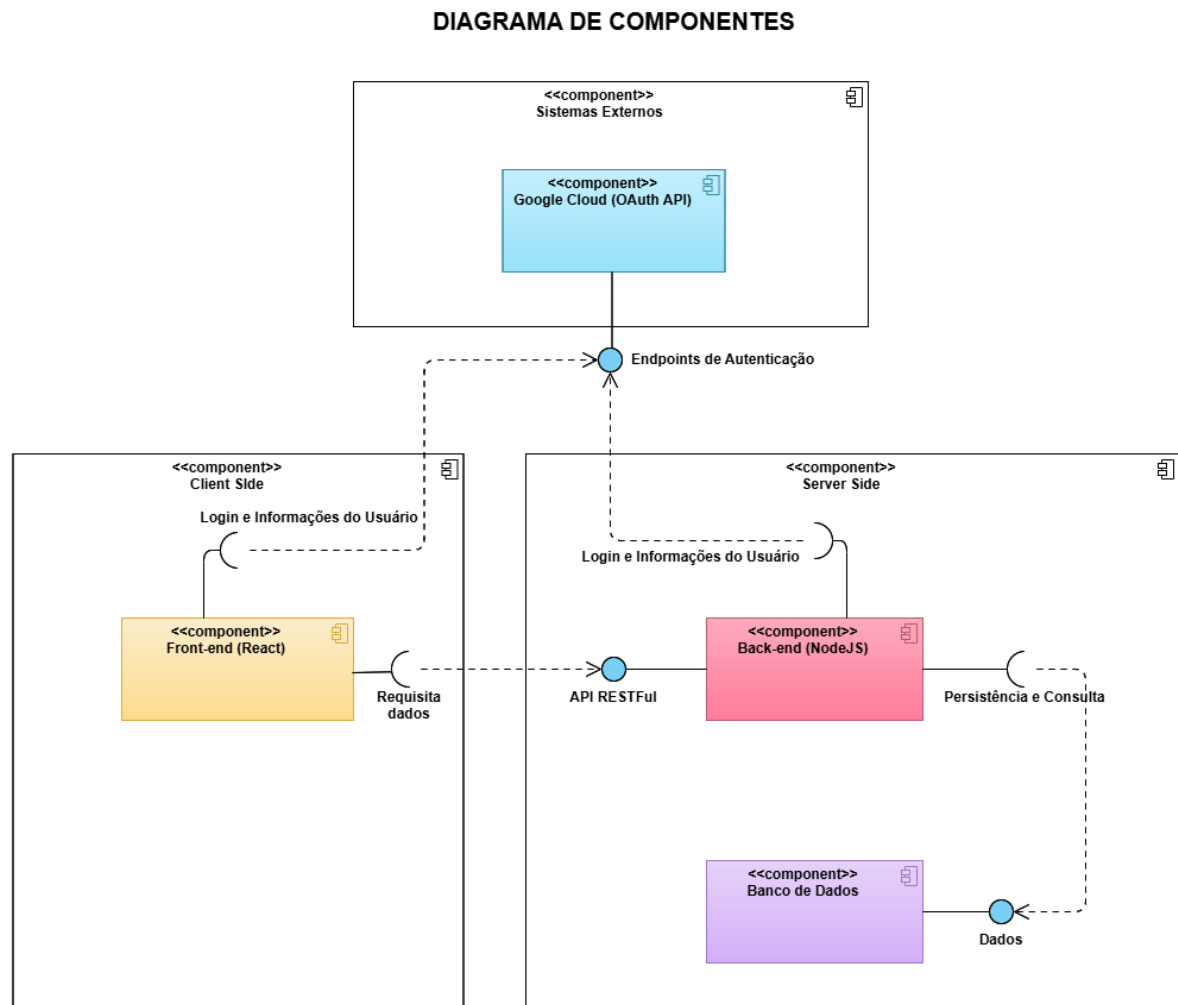


Figura 6 – Diagrama de componente da aplicação

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.
- **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.

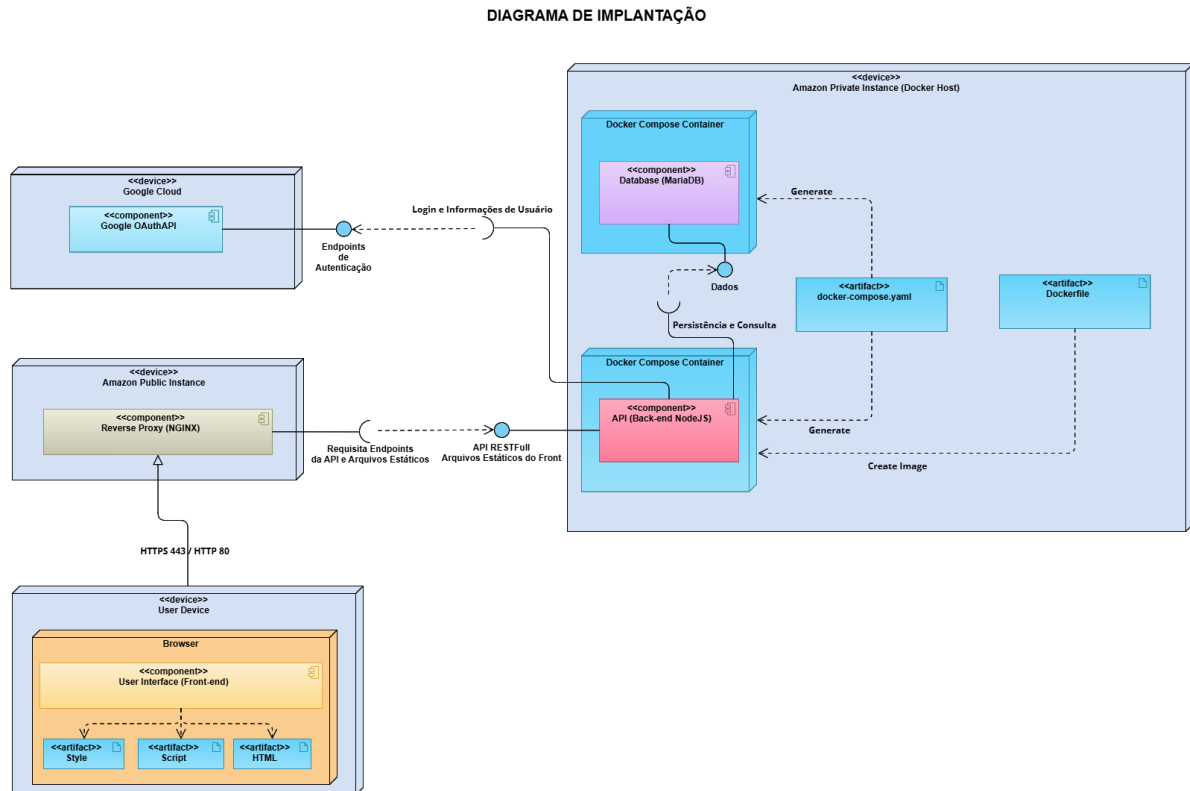


Figura 7 – Diagrama de implantação da aplicação

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 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 (FOUN-DATION, 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.yaml:* 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.

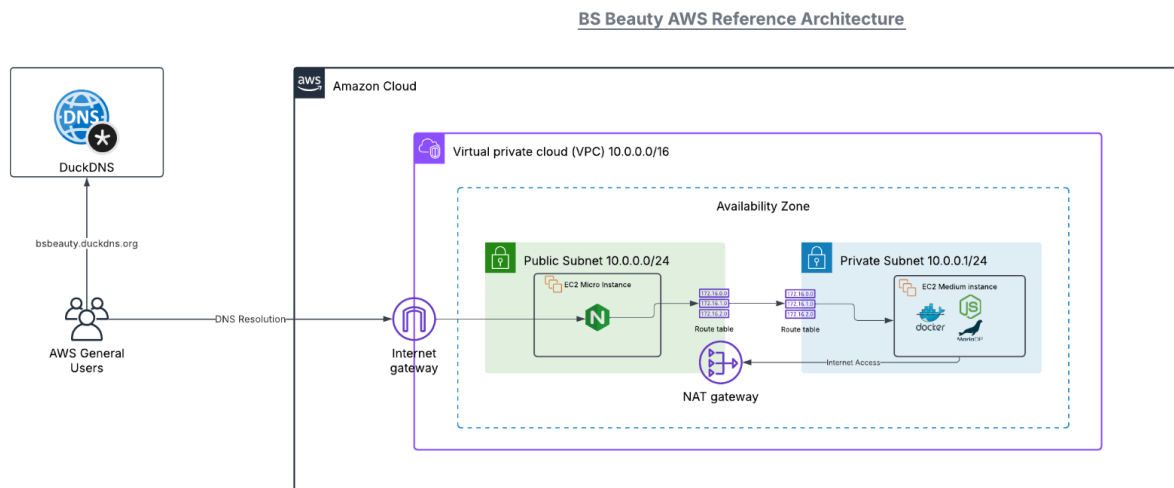


Figura 8 – Diagrama Geral da Arquitetura

- **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 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.

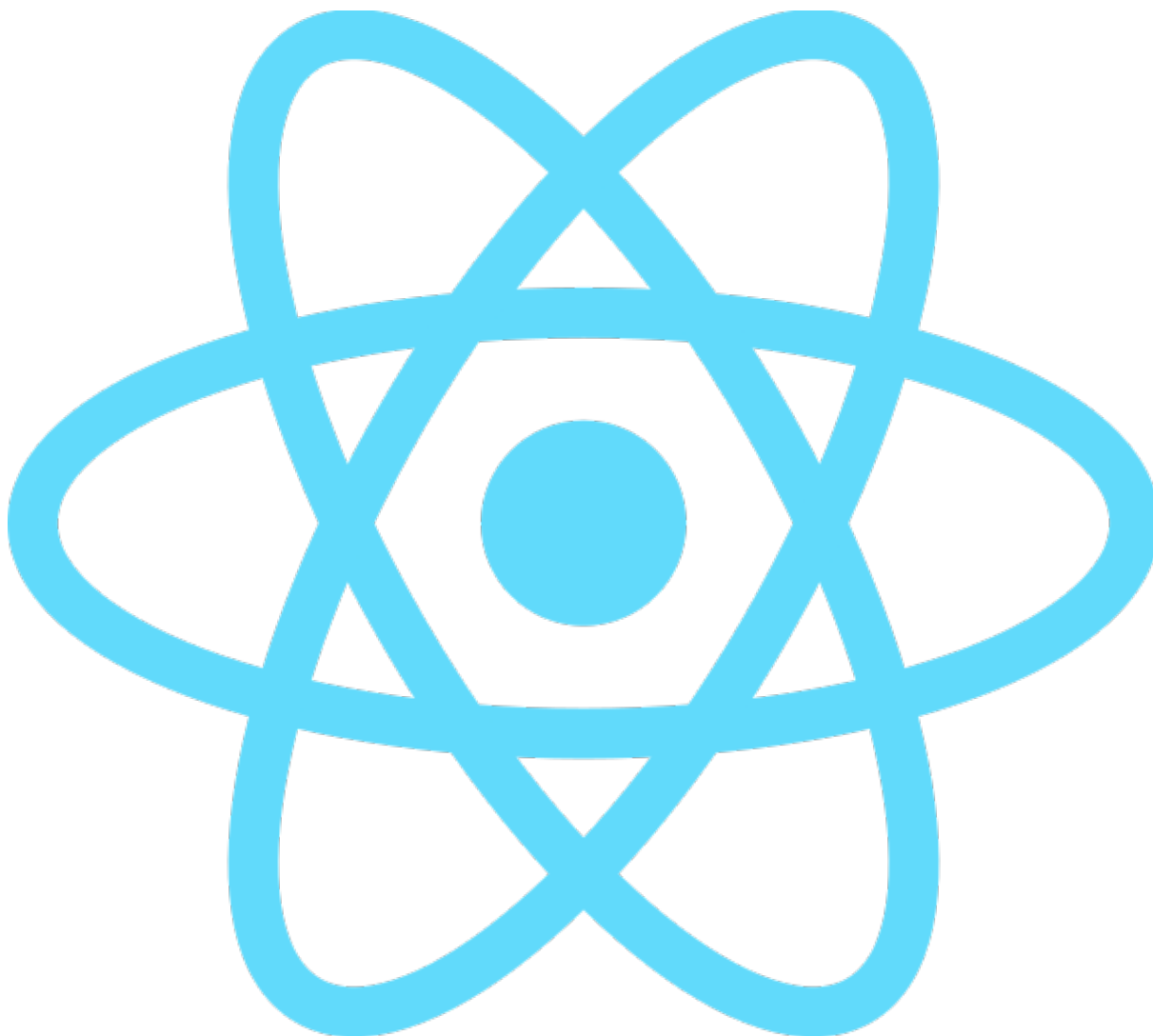


Figura 9 – Logo do React

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.

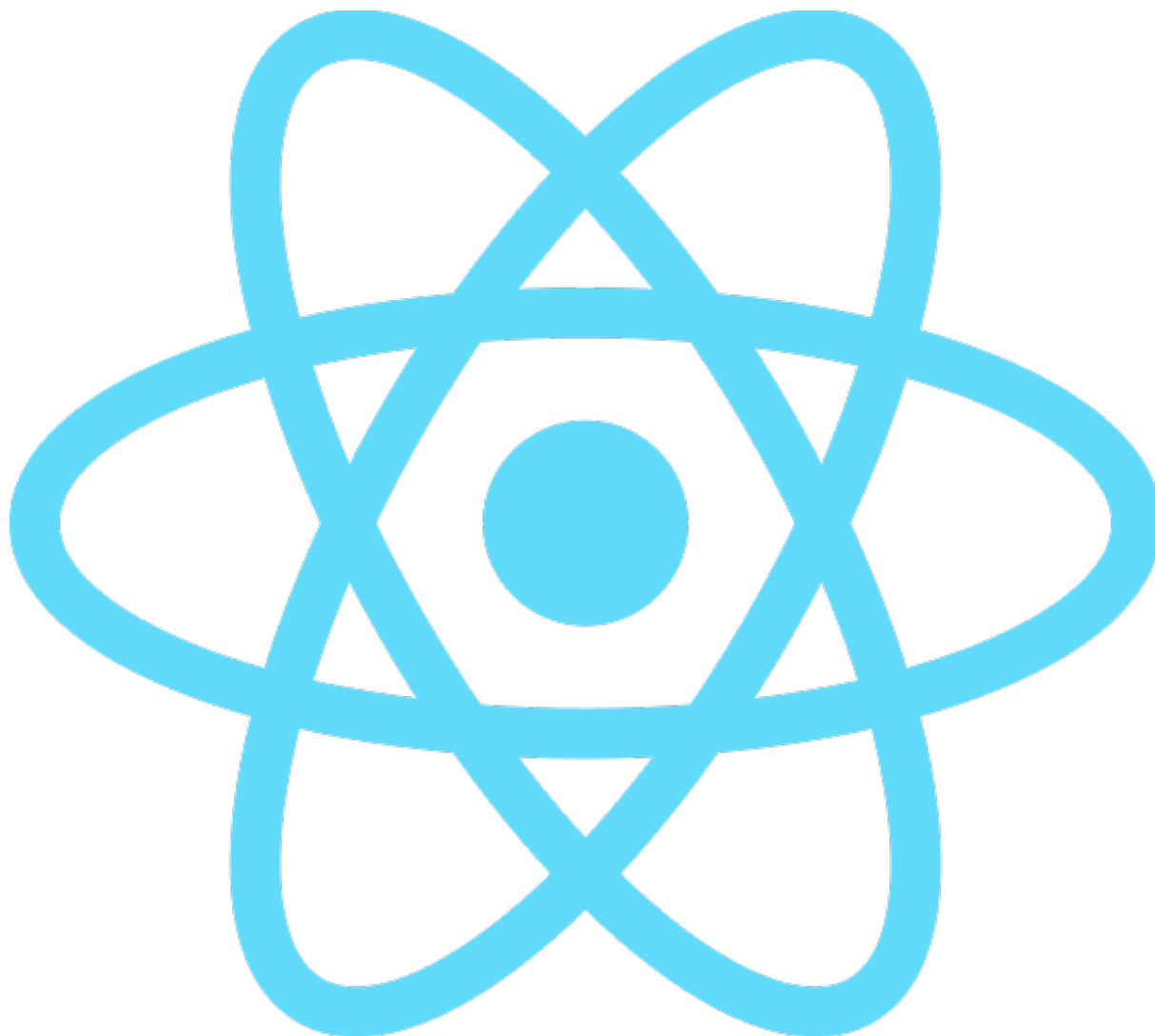


Figura 10 – Logo do React

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.

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 cenários 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 **Vitest** com suporte a geração de relatórios de cobertura. 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 automatizados que envolva todos os processos que foram mapeados no escopo do projeto.

Os arquivos contendo as classes/funções de testes devem estar localizados em diretórios específicos de testes, 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 Infraestrutura de Testes

A infraestrutura de testes do projeto está fortemente integrada ao processo de integração contínua (CI) e entrega contínua (CD), utilizando a ferramenta **GitHub Actions**. 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.

4.5.3 Análise Estáticas

A análise estática de código é realizada utilizando a ferramenta **SonarQube**, que permite 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.

4.5.4 Testes Automatizados

A automação de testes tem como objetivo aumentar a confiabilidade do software e permitir validações rápidas e constantes. O projeto conta com:

- **Testes unitários:** Validam o comportamento de funções e componentes isolados. **Vitest**.
- **Testes integrados:** Verificam a interação entre módulos e componentes da aplicação, também utilizando o **Vitest**.

4.5.5 Logs

4.5.6 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** para garantir padrões de estilo e detectar possíveis erros ou práticas inadequadas de codificação.
 - Uso do **Prettier** 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.
- Aplicação do princípio DRY (Don't Repeat Yourself).

- **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

4.6.1 Critérios de segurança e privacidade

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

6.1 Dificuldades, escolhas e Descartes

([IBGE, 1993](#))

ver documento "abntex2-modelo-include-comandos" para dicas e tutoriais de como fazer tabelas, graficos, quadros, inserir imagens e documentos externos etc aqui.

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 20.
- 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 42 e 43.
- 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 24.
- AWS. *O que é Scrum?* 2024. Disponível em: <<https://aws.amazon.com/pt/what-is/scrum/>>. Acesso em: 1 jun. 2025. Citado na página 29.
- 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 18.
- 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 20.
- CONVENTIONAL COMMITS. *Conventional Commits*. 2025. Disponível em: <<https://www.conventionalcommits.org/pt-br/v1.0.0/>>. Acesso em: 1 jun. 2025. Citado na página 31.
- DISCORD. *Discord*. 2025. Disponível em: <<https://discord.com>>. Acesso em: 31 mai. 2025. Citado na página 28.
- 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 20.
- 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 30.
- 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 20.
- FOUNDATION, E. F. *Certbot - Get HTTPS for free*. 2025. <<https://certbot.eff.org/>>. Acesso em: 05 junho 2025. Citado na página 40.
- 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 18.

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 23.

GIT. *Git*. 2025. Disponível em: <<https://git-scm.com>>. Acesso em: 1 jun. 2025. Citado na página 31.

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 32.

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 31.

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 41.

IBGE. *Normas de apresentação tabular*. 3. ed. Rio de Janeiro: Centro de Documentação e Disseminação de Informações. Fundação Instituto Brasileiro de Geografia e Estatística, 1993. Acesso em: 21 ago 2013. Citado na página 51.

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 20.

NOTION. *Notion*. 2025. Disponível em: <<https://www.notion.com/pt>>. Acesso em: 31 mai. 2025. Citado 2 vezes nas páginas 28 e 30.

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 31.

PROJECTLIBRE. *ProjectLibre Desktop*. 2025. Disponível em: <<https://www.projectlibre.com/projectlibre-desktop/>>. Acesso em: 31 mai. 2025. Citado na página 28.

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 21.

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 20.

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 18.

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 21.

SENAC-SP. *Perfil do Empreendedor de Beleza*. 2022. Relatório SENAC-SP. Disponível em: <<https://www.sp.senac.br/perfil-empresendedor-beleza-2022>>. Acesso em: 01 jun 2025. Citado na página 20.

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 22.

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