

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA
PAULA SOUZA**

**Faculdade de Tecnologia Baixada Santista
Rubens Lara**

**Curso Superior de Tecnologia em
Sistemas para Internet**

**Mayumi Nascimento Tamashiro
Michelle Jeanne Ferrinho Baraçal**

**BeleSys
Sistema de gestão multiplataforma para salões de beleza**

**Santos, SP
2025**

**Mayumi Nascimento Tamashiro
Michelle Jeanne Ferrinho Baraçal**

BeleSys
Sistema de gestão multiplataforma para salões de beleza

Trabalho de Conclusão de Curso apresentado à Faculdade de Tecnologia Rubens Lara, como exigência para a obtenção do Título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Dr. Joseffe Barroso de Oliveira

**Santos, SP
2025**

RESUMO

Este trabalho de conclusão de curso trata do desenvolvimento do BeleSys, um sistema web de gestão criado para atender às necessidades de salões de beleza, com foco na organização e no controle de agendamentos e atividades internas. Considerando a rotina administrativa desses estabelecimentos, frequentemente marcada por anotações manuais e falta de controle eficiente, observou-se a necessidade de uma ferramenta prática que auxilie na organização e melhore a gestão dos atendimentos e serviços prestados. O objetivo do projeto foi oferecer uma solução digital que tornasse o gerenciamento mais eficiente, contribuindo para a redução de falhas operacionais e para a melhoria da experiência tanto dos administradores quanto dos profissionais do salão. O sistema desenvolvido possui funcionalidades como o cadastro e a gestão de funcionários e serviços, controle de agendamentos com visualização por calendário, atualização de status dos atendimentos, envio automático de lembretes aos clientes e geração de relatórios financeiros por período. Os usuários são segmentados por níveis de acesso: enquanto o administrador tem acesso completo a todos os recursos do sistema, os funcionários visualizam apenas os atendimentos atribuídos a eles. Além disso, o sistema possui interface responsiva, permitindo seu uso tanto em dispositivos móveis quanto em desktops, o que amplia sua acessibilidade. Para verificar a eficiência da solução, foram realizados testes com usuários reais, os quais executaram tarefas simulando o uso cotidiano da aplicação. Os resultados mostraram que os participantes conseguiram navegar com facilidade, localizar informações com rapidez e compreender as funções sem maiores dificuldades. Dessa forma, conclui-se que o BeleSys atende aos objetivos propostos, mostrando-se uma ferramenta útil para otimizar a gestão de salões de beleza, promovendo maior organização, controle e praticidade.

Palavras-chave: sistema, salão de beleza, agendamento, gestão.

ABSTRACT

This Final Project presents the development of BeleSys, a web-based management system designed to meet the needs of beauty salons, with a focus on organizing and controlling appointments and internal activities. Considering the administrative routine of these establishments, often marked by manual notes and lack of efficient control, the need for a practical tool to assist in organization and improve the management of appointments and services was identified. The aim of this project was to offer a digital solution capable of making management more efficient, reducing operational failures, and improving the experience of both administrators and salon professionals. The system offers functionalities such as employee and service registration and management, appointment control with calendar visualization, status updates, automated client reminders, and financial report generation by date range. Users are segmented by access level: administrators have full access to all system features, while employees can view only their assigned appointments. Additionally, the interface is responsive, allowing use on both mobile devices and desktops, which enhances accessibility. To evaluate the system's effectiveness, usability tests were conducted with real users performing simulated tasks. The results showed that participants navigated easily, located information quickly, and understood the system's features without major difficulties. Therefore, it is concluded that BeleSys meets the proposed objectives, proving to be a useful tool for optimizing beauty salon management by promoting greater organization, control, and practicality.

Keywords: system, beauty salon, usability, management.

LISTA DE ABREVIATURAS E SIGLAS

UML – UNIFIED MODELING LANGUAGE	11
API – APPLICATION PROGRAMING INTERFACE	11
ORM – OBJECT-RELATIONAL MAPPING	11
JWT – JSON WEB TOKEN	12
OWASP – OPEN WEB APPLICATION SECURITY PROJECT	12
MER – MODELO ENTIDADE RELACIONAMENTO	23
SGBD – SISTEMA GERENCIADOR DE BANCO DE DADOS	23
SQL – STRUCTURED QUERY LANGUAGE	24
CRUD – CREATE, READ, UPDATE, DELETE	27
WAMMI – WEBSITE ANALYSIS AND MEANSUREMENT INVENTORY	36

LISTA DE ILUSTRAÇÕES

Ilustração 1 - Arquitetura do sistema Belesys	12
Ilustração 2 - Diagrama de caso de uso	15
Ilustração 3 - Modelo Entidade-relacionamento	23
Ilustração 4 - método criar agendamento	27
Ilustração 5 - método atualizar agendamento	28
Ilustração 6 - Configuração do node-cron	29
Ilustração 7 - Implementação da lógica de lembretes	29
Ilustração 8 - Função de criptografia de senhas.....	30
Ilustração 9 - Tela de Agenda	31
Ilustração 10 - Modal de cadastro e edição.....	31
Ilustração 11 - Modal de detalhes	32
Ilustração 12 - Tela de relatórios	33
Ilustração 13 - Gráfico da Pergunta 1.....	36
Ilustração 14 - Gráfico da Pergunta 2.....	36
Ilustração 15 - Gráfico da Pergunta 3.....	37
Ilustração 16 - Gráfico da Pergunta 4.....	37
Ilustração 17 - Tela de funcionários cadastrados.....	42
Ilustração 18 - Tela de visualização de serviços cadastrados.....	42
Ilustração 19 - Modal do usuário.....	43
Ilustração 20 - Tela de comissões	43
Ilustração 21 - Gráfico da Pergunta 5.....	44
Ilustração 22 - Gráfico da Pergunta 6.....	44
Ilustração 23 - Gráfico da Pergunta 7.....	44
Ilustração 24 - Gráfico da Pergunta 8.....	44

LISTA DE TABELAS

TABELA 01 – FLUXO DE EVENTO: LOGIN	16
TABELA 02 – FLUXO DE EVENTO: GERAR RELATÓRIOS	16
TABELA 03 – FLUXO DE EVENTO: CRIAR AGENDAMENTO	17
TABELA 04 – FLUXO DE EVENTO: VISUALIZAR AGENDAMENTO	18
TABELA 05 – FLUXO DE EVENTO: ATUALIZAR AGENDAMENTO	18
TABELA 06 – FLUXO DE EVENTO: ENVIAR LEMBRETE	19
TABELA 07 – FLUXO DE EVENTO: CRIAR FUNCIONÁRIO	19
TABELA 08 – FLUXO DE EVENTO: VISUALIZAR FUNCIONÁRIO	20
TABELA 09 – FLUXO DE EVENTO: ATUALIZAR FUNCIONÁRIO	20
TABELA 10 – FLUXO DE EVENTO: CRIAR SERVIÇO	21
TABELA 11 – FLUXO DE EVENTO: VISUALIZAR SERVIÇO	21
TABELA 12 – FLUXO DE EVENTO: ATUALIZAR SERVIÇO	22
TABELA 13 – FLUXO DE EVENTO: EXCLUIR SERVIÇO	22
TABELA 14 – TABELA DE SERVIÇOS	24
TABELA 15 - TABELA DE FUNCIONÁRIOS	25
TABELA 16 – TABELA DE AGENDAMENTOS	25
TABELA 17 – TABELA DE AGENDAMENTOS_ITENS	26
TABELA 18 – TAREFAS DO TESTE DE USABILIDADE	34

SUMÁRIO

1 INTRODUÇÃO	8
1.1 OBJETIVO	9
1.1.1 OBJETIVO GERAL	9
1.1.2 OBJETIVOS ESPECÍFICOS	9
1.2 ESTADO DA ARTE	9
2 DESENVOLVIMENTO.....	11
2.1 ANÁLISE DO SISTEMA	11
2.1.1 ANÁLISE DE REQUISITOS.....	13
2.1.2 DIAGRAMA DE CASO DE USO.....	15
2.1.3 FLUXO DE EVENTOS.....	16
2.2 BANCO DE DADOS	23
2.3 CAMADA DE NEGÓCIO	27
2.4 CAMADA DE APRESENTAÇÃO	30
3 RESULTADO.....	34
3.1 TESTES DE USABILIDADE	34
3.2 CONCLUSÃO	39
3.3 PRÓXIMOS PASSOS.....	39
REFERÊNCIAS BIBLIOGRÁFICAS	41
APÊNDICE A – CAMADA DE APRESENTAÇÃO.....	42
APÊNDICE B - RESULTADO.....	44

1 INTRODUÇÃO

Tradicionalmente, o agendamento nos salões de beleza era realizado de forma manual, utilizando agendas de papel, o que frequentemente resultava em problemas como linhas telefônicas ocupadas, desencontros de informação e erros de agendamento. Além disso, o modelo dependia fortemente da atenção e organização da equipe de recepção, e estava sujeito a falhas humanas, como anotações incorretas ou perda de registros (SEBRAE, 2025).

Nesse sentido o agendamento online está transformando a forma como os salões de beleza atendem seus clientes oferecendo uma experiência mais simples e agradável (Fortes, 2024).

Embora já existam várias ferramentas disponíveis no mercado como o Sistema Beauty, Simples Agenda e Salão 365 que oferecem esse tipo de solução este trabalho apresenta o desenvolvimento do sistema web BeleSys, cujo objetivo principal é proporcionar maior controle administrativo e agendamentos mais eficientes nos salões de beleza.

Para atingir esse objetivo, o projeto contempla os seguintes objetivos específicos:

- Criar uma interface intuitiva e acessível para administradores e funcionários.
- Implementar o agendamento de serviços com visualização interativa de calendário.
- Desenvolver controle de agenda individual por profissional.
- Gerar relatórios financeiros por período.
- Estabelecer níveis de acesso diferenciados para administradores e funcionários.
- Implementar o envio de lembretes via WhatsApp.
- Garantir a compatibilidade do sistema com dispositivos móveis e desktops.

O BeleSys se posiciona como uma solução de gestão multiplataforma, acessível por diversos dispositivos. Ele oferece funcionalidades como gerenciamento de funcionários, serviços, agendamentos e relatórios financeiros detalhados. O sistema se destaca pelo seu mecanismo robusto de agendamento (com visualização interativa, controle individual e validação de horários) e pela comunicação integrada via WhatsApp — o aplicativo de mensagens mais utilizado atualmente.

Além disso, o BeleSys proporciona relatórios de faturamento, compatibilidade multiplataforma e excelente usabilidade. Dessa forma, a transição do agendamento manual para soluções digitais, como o BeleSys, busca superar desafios como a dificuldade de controle de horários, o risco de conflitos de agenda e a ausência de um histórico digitalizado.

1.1 OBJETIVO

Neste tópico serão abordados os objetivos gerais e os objetivos específicos deste trabalho, com o objetivo de apresentar os dados explicitados.

1.1.1 OBJETIVO GERAL

Desenvolver um sistema web multiplataforma de gerenciamento de salão de beleza que proporciona maior controle administrativo e organização de atendimentos.

1.1.2 OBJETIVOS ESPECÍFICOS

1. Criar uma interface intuitiva e acessível para administradores e funcionários.
2. Permitir o agendamento de serviços com visualização interativa de calendário.
3. Implementar controle de agenda individual por profissional.
4. Gerar relatórios financeiros por período.
5. Estabelecer diferentes níveis de acesso conforme o perfil do usuário (administrador ou funcionário).
6. Implementar uma ferramenta de envio de lembretes via WhatsApp.
7. Garantir a compatibilidade do sistema com dispositivos móveis e desktops.

1.2 ESTADO DA ARTE

A rotina em salões de beleza costuma ser agitada, exigindo organização para evitar atrasos e garantir um bom atendimento. Durante muito tempo, o agendamento de horários foi feito de forma manual, usando agendas de papel ou anotações em cadernos. No entanto, esse método acabou se tornando um desafio. Problemas como conflitos de horário, informações perdidas e dificuldade

de comunicação com os clientes passaram a ser comuns, além de consumirem tempo que poderia ser melhor aproveitado (SEBRAE, 2025).

Com a vida cada vez mais acelerada, tanto para os profissionais quanto para os clientes, contar com um sistema digital de agendamento deixou de ser um luxo e passou a ser uma necessidade. Essas ferramentas ajudam a otimizar o tempo, evitar filas e garantir mais conforto para quem agenda e para quem atende.

Atualmente, existem diversas opções no mercado, como o Reservio, Sistema Beauty, Simple Agenda e Salão 365. Cada um oferece suas funcionalidades. Foi pensando nessa realidade que surgiu o BeleSys, um sistema web multiplataforma desenvolvido para facilitar a gestão do salão como um todo, desde o controle da equipe até a organização dos agendamentos.

O BeleSys permite que tudo fique mais claro e organizado: profissionais conseguem visualizar facilmente sua agenda, os serviços ficam bem definidos e os relatórios financeiros são gerados de forma simples. O sistema conta com um calendário interativo, que mostra os horários disponíveis, evita conflitos de marcação e permite um controle individual para cada profissional.

Um diferencial importante do BeleSys é a comunicação com os clientes. O sistema envia lembretes automáticos pelo WhatsApp, ajudando a reduzir faltas e reforçando o compromisso com o agendamento. Como esse é um dos meios de comunicação mais usados atualmente, a funcionalidade se mostra bastante eficiente.

Além disso, o sistema foi pensado para funcionar bem tanto em computadores quanto em celulares, adaptando-se a diferentes tamanhos de tela. Testes realizados durante o desenvolvimento mostraram que a interface é amigável e de fácil uso, mesmo para quem não tem muita familiaridade com tecnologia.

Em resumo, o BeleSys se destaca pela proposta de oferecer uma gestão prática, com agendamentos organizados e comunicação direta com os clientes. Enquanto outras ferramentas focam em marketing ou pagamentos, o BeleSys concentra seus esforços na organização interna e na experiência do usuário dentro do salão.

2 DESENVOLVIMENTO

Neste capítulo serão abordadas as fases do desenvolvimento do sistema proposto. Para tanto, esse capítulo será subdividido nos seguintes itens: análise do sistema, a estrutura do Banco de Dados, as ferramentas utilizadas para o desenvolvimento do *front-end* e as principais rotinas implementadas no *back-end*.

2.1 ANÁLISE DO SISTEMA

Entre vários mecanismos de análise disponíveis para o desenvolvimento do sistema *Belesys*, foi feita a escolha pela *Unified Modeling Language (UML)* como linguagem de modelagem, por sua capacidade de representação visual de sistemas de software. Segundo Fowler (2004) “*UML (Unified Modeling Language)* é uma família de notações gráficas, apoiada por um metamodelo único, que ajuda na descrição e no projeto de sistemas de software”.

O sistema *web Belesys* possui 4 principais páginas, sendo elas:

1. Funcionários: parte exclusiva da gerência, destinada à gestão dos funcionários.
2. Serviços: também exclusiva da gerência para criação e gerenciamento dos serviços oferecidos pelo estabelecimento.
3. Agenda: página destinada para que os funcionários acompanhem cada serviço agendado ao qual o administrador criará com os devidos dados do serviço a ser realizado, podendo ser alterados e finalizados.
4. Relatórios: utilizado pela gestão para acompanhar o andamento de seu negócio e obter dados importantes para a tomada de decisão.

A implementação do *Belesys* utilizou um conjunto de tecnologias modernas, alinhadas aos requisitos de desempenho, segurança e usabilidade.

No *backend*, utilizou-se *Node.js* para a criação da *Application Programming Interface (API)* juntamente com o *Sequelize* como *ORM (Object Relational Mapping)* para interação com o banco de dados.

Para proteção de dados sensíveis, implementou-se criptografia de senhas com *bcrypt* e autenticação via *tokens JWT (json web token)*, seguindo as diretrizes de segurança da *Open Web Application Security Project (OWASP)*.

No *frontend*, o sistema foi construído com *React.js*, uma biblioteca *JavaScript* mantida pelo *Facebook/Meta* para construção de interfaces declarativas baseadas em componentes, aliado ao *TypeScript*, uma linguagem fortemente tipada desenvolvida pela *Microsoft* que, segundo sua documentação oficial, "oferece melhores ferramentas em qualquer escala" (MICROSOFT, 2024). Além disso, a biblioteca *Bootstrap* foi incorporada para assegurar a responsividade do sistema em diferentes dispositivos, enquanto o *SweetAlert2* foi utilizado para a exibição de modais interativos que, de acordo com Nielsen (1994), atende aos princípios de *feedback* imediato e controle do usuário, elementos fundamentais para a experiência do usuário.

A fim de complementar a visão funcional apresentada até aqui, a Ilustração 1 apresenta um panorama da arquitetura em camadas do sistema BeleSys, representando sua estrutura interna e as tecnologias utilizadas na construção da solução.

Ilustração 1 - Arquitetura do sistema Belesys



Fonte: Autoras, 2025

2.1.1 ANÁLISE DE REQUISITOS

Nesta fase ocorre a captura das intenções e as necessidades dos usuários a ser desenvolvido através dos requisitos funcionais (RF), assim como cita Sommerville (2011) “Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferecem e as restrições a seu funcionamento”.

Os requisitos funcionais foram classificados conforme os módulos principais da aplicação, contemplando as interações realizadas por administradores, funcionários e o próprio sistema.

1. Autenticação e autorização

- RF01: O sistema deve permitir que funcionários e administradores façam login validando seus respectivos e-mail e senha.
- RF02: O sistema deve permitir que funcionários e administradores redefinam sua senha.

2. Gestão de funcionários

- RF03: O sistema deve permitir que administradores cadastrem novos funcionários
- RF04: O sistema deve permitir que administradores editem e excluam funcionários.
- RF05: O sistema deve permitir que administradores listem todos os funcionários, com opção de visualização detalhada.

3. Gestão de serviços

- RF06: O sistema deve permitir que administradores cadastrem novos serviços.
- RF07: O sistema deve permitir que administradores editem e excluam serviços.
- RF08: O sistema deve permitir que administradores listem todos os serviços oferecidos.

4. Gestão de agendamentos

- RF09: O sistema permite que administradores realizem novos agendamentos.
- RF10: O sistema permite que o administrador confirme ou cancele agendamentos.
- RF11: O sistema deve apresentar a agenda em diferentes visualizações: diária, semanal e mensal.
- RF12: O sistema permite que um usuário do tipo funcionário visualize os agendamentos vinculados a ele.
- RF13: O sistema permite que administradores e funcionários visualizem detalhes de um agendamento específico.

5. Relatórios e anfilises

- RF14: O sistema deve permitir aos administradores gerarem relatórios financeiros com faturamento por período.

6. Lembretes

- RF15: O sistema deve identificar agendamentos próximos e enviar lembretes via WhatsApp para os clientes automaticamente.

A definição de requisitos não funcionais é essencial para a construção de um sistema, pois estabelece atributos como desempenho, segurança e confiabilidade. Segundo Sommerville (2011), esses requisitos não estão diretamente ligados aos serviços do sistema, mas a sua não conformidade pode comprometer a usabilidade ou até mesmo inviabilizar o sistema. A seguir são apresentados os requisitos não funcionais identificados no sistema:

- RNF-01: O sistema deve ter seu design responsivo para funcionar em qualquer dispositivo.
- RNF-02: O sistema deve criptografar as senhas antes de armazenar no banco de dados.
- RNF-03: O sistema deverá ser protegido para acesso apenas a usuários autenticados.

2.1.3 FLUXO DE EVENTOS

O fluxo de eventos descreve detalhadamente o comportamento dos casos de uso, com início, execução e término da ação esperada, evidenciando tanto o fluxo principal quanto alternativas em situações específicas. A seguir, apresentam-se os principais fluxos organizados por funcionalidade, começando pela Tabela 1 representando o fluxo de login de um usuário até a conclusão da ação.

Tabela 1: Fluxo de eventos Login

Caso de uso: Realizar Login
Atores: Administrador, Funcionário

Pré-condições: O usuário deve estar cadastrado no sistema.

Fluxo Principal:

1. O usuário acessa a página de login.
 2. O sistema solicita credenciais (e-mail e senha).
 3. O usuário informa as credenciais.
 4. O sistema valida os dados informados.
 5. Se os dados estiverem corretos, o usuário é redirecionado para a página correspondente ao seu perfil.
-

Pós-condições: O usuário tem acesso ao sistema conforme seu nível de permissão.

Fluxo Alternativo:

Se as credenciais forem inválidas, o sistema exibe uma mensagem de erro e permit nova tentativa.

Fonte: Autoras, 2025

A partir do login, o sistema oferece múltiplas funcionalidades de acordo com o perfil do usuário. Seguindo com a Tabela 2, a representação do fluxo de geração de relatórios, funcionalidade exclusiva da administração.

Tabela 2 - Fluxo de eventos Gerar relatórios

caso de uso: Gerar Relatórios

Ator: Administrador

Pré-condições: O administrador deve estar autenticado.

Fluxo Principal:

1. O administrador acessa o sistema.
 2. Seleciona a opção "Gerar relatórios".
 3. O administrador informa os parâmetros desejados.
 4. O sistema gera e exibe o relatório solicitado.
 5. O administrador pode visualizar ou exportar o relatório.
-

Pós-condições: O relatório gerado fica disponível para consulta e exportação.

Fonte: Autoras, 2025

A criação de novos agendamentos também está entre as ações essenciais ao funcionamento da plataforma, conforme indicado na Tabela 3.

Tabela 3 - Fluxo de eventos Criar Agendamento

caso de uso: Criar Agendamento

Ator: Administrador

Pré-condições: O administrador deve estar autenticado.

Fluxo Principal:

1. O sistema exibe um formulário de cadastro de agendamento.
 2. O administrador preenche as informações.
 3. O sistema valida as informações preenchidas
 4. O sistema salva o agendamento e o exibe na agenda.
-

Pós-condições: O agendamento é registrado e pode ser visualizado no sistema.

Fonte: Autoras, 2025

A Tabela 4 evidencia o fluxo de visualizar agendamentos, onde o sistema recupera e exibe os agendamentos registrados no banco de dados, permitindo a visualização de detalhes específicos, conforme o nível de permissão do usuário.

Tabela 4 - Fluxo de eventos Visualizar Agendamento

caso de uso: Visualizar Agendamento
Ator: Administrador, funcionário

Pré-condições:

- O usuário deve estar autenticado.
 - O agendamento precisa existir no banco de dados.
-

Fluxo Principal:

1. O usuário acessa o painel do sistema.
 2. Seleciona a opção "Visualizar Agendamento".
 3. O sistema recupera os agendamentos e os exibe.
-

Pós-condições: O usuário pode consultar os agendamentos cadastrados.

Fonte: Autoras, 2025

Em casos em que ajustes são necessários, o administrador pode atualizar os dados dos agendamentos já registrados. Segundo a Tabela 5, após acessar os detalhes do agendamento, o administrador pode editar e salvar as informações desejadas, mantendo os registros sempre atualizados.

Tabela 5 - Fluxo de eventos Atualizar Agendamento

caso de uso: Atualizar Agendamento

Ator: Administrador

Pré-condições: O agendamento deve estar cadastrado no sistema.

Fluxo Principal:

1. O administrador acessa o sistema.
 2. Seleciona um agendamento na agenda.
 3. O sistema exibe os detalhes do agendamento.
 4. O administrador realiza as alterações necessárias.
 5. O sistema salva as mudanças.
-

Pós-condições: O agendamento é atualizado com as novas informações.

Fonte: Autoras, 2025

Uma funcionalidade automatizada foi desenvolvida para envio de lembretes via WhatsApp, como forma de reduzir faltas e aumentar a pontualidade dos clientes. Na Tabela 6, observa-se que o sistema verifica os

agendamentos próximos e dispara lembretes para os respectivos clientes, promovendo maior organização no atendimento.

Tabela 6 - Fluxo de eventos Enviar Lembretes

caso de uso: Enviar Lembretes

Ator: Sistema

Pré-condições: O sistema deve ter agendamentos registrados.

Fluxo Principal:

1. O sistema verifica os agendamentos próximos.
 2. Para cada agendamento, o sistema envia um lembrete via WhatsApp para o cliente.
 3. O sistema registra o envio do lembrete.
-

Pós-condições: O cliente recebe um lembrete do seu agendamento.

Fonte: Autoras, 2025

O processo de gestão de funcionários também conta com fluxos específicos, iniciando pela criação de novos registros. De acordo com a Tabela 7, o administrador, após preencher e validar as informações, consegue registrar novos funcionários que passam a ter acesso ao sistema.

Tabela 7 - Fluxo de eventos Criar Funcionário

Caso de uso: Criar Funcionário

Ator: Administrador

pré-condição: O administrador deve estar autenticado.

Fluxo Principal:

1. O sistema exibe o formulário de cadastro de funcionários.
 2. O administrador preenche os dados.
 3. O sistema valida as informações preenchidas
 4. O sistema cadastra o funcionário.
-

pós-condições: O funcionário é registrado e consegue acessar o sistema.

Fonte: Autoras, 2025

A seguir, é possível consultar os funcionários já cadastrados, conforme demonstrado na Tabela 8.

Tabela 8 - Fluxo de eventos Visualizar Funcionário

Caso de uso: Visualizar Funcionário

Ator: Administrador

Pré-condições: O administrador deve estar autenticado no sistema.

Fluxo Principal:

1. O administrador acessa a página Funcionários cadastrados.
 2. O sistema exibe a lista de funcionários cadastrados.
-

Pós-condições: A lista de funcionários é exibida

Fonte: Autoras, 2025

Caso haja necessidade de ajustes nas informações, o sistema permite a edição dos registros. Na Tabela 9, destaca-se o fluxo para atualização dos dados de um funcionário, processo que exige a existência prévia do colaborador no banco e autenticação do administrador.

Tabela 9 - Fluxo de eventos Atualizar Funcionário

Caso de uso: Atualizar Funcionário

Ator: Administrador

Pré-condições:

- O administrador deve estar autenticado no sistema.
 - O funcionário a ser atualizado deve existir no banco de dados.
-

Fluxo Principal:

1. O administrador acessa a página Funcionários cadastrados.
 2. Seleciona um funcionário na lista.
 3. O sistema exibe os detalhes do funcionário.
 4. O administrador edita as informações necessárias e confirma.
 5. O sistema salva as alterações.
-

Pós-condições: As informações do funcionário são atualizadas

Fonte: Autoras, 2025

Conforme a Tabela 10, o administrador preenche um formulário específico para registrar novos serviços no sistema, agilizando a gestão dos serviços oferecidos, permitindo sua criação conforme necessidade da administração.

Tabela 10 - Fluxo de eventos Criar Serviço

Caso de uso: Criar Serviço

Ator: Administrador

Pré-condições: O administrador deve estar autenticado no sistema.

Fluxo Principal:

1. O administrador acessa a página Serviços cadastrados.
 2. Seleciona a opção "Criar Serviço".
 3. O sistema exibe um formulário.
 4. O administrador preenche as informações e confirma.
 5. O sistema salva o novo serviço.
-

Pós-condições: serviço é registrado no sistema.

Fonte: Autoras, 2025

Também é possível consultar os serviços cadastrados no sistema, conforme a Tabela 11, demonstrando o fluxo de visualização de serviços, essencial para o controle e organização do catálogo do salão.

Tabela 11 - Fluxo de eventos Visualizar Serviço

Caso de uso: Visualizar Serviço

Ator: Administrador

Pré-condições:

- O administrador deve estar autenticado no sistema.
 - Deve haver pelo menos um serviço cadastrado no sistema.
-

Fluxo Principal:

1. O administrador acessa o painel do sistema.
 2. Seleciona a opção "Visualizar Serviços".
 3. O sistema exibe a lista de serviços cadastrados.
-

Pós-condições: A lista de serviços é exibida.

Fonte: Autoras, 2025

Alterações nas informações de serviços já existentes são permitidas por meio do fluxo apresentado a seguir na Tabela 12, onde o administrador pode editar os dados de um serviço cadastrado, realizando as alterações necessárias.

Tabela 12 - Fluxo de eventos Atualizar Serviço

Caso de uso: Atualizar Serviço
Ator: Administrador

Pré-condições:

- O administrador deve estar autenticado no sistema.
 - O serviço a ser atualizado deve existir no sistema.
-

Fluxo Principal:

1. O administrador acessa o painel do sistema.
 2. Seleciona um serviço na lista.
 3. O sistema exibe os detalhes do serviço.
 4. O administrador edita as informações necessárias e confirma.
 5. O sistema salva as alterações.
-

Pós-condições: As informações do serviço são atualizadas no banco de dados.

Fonte: Autoras, 2025

Por fim, apresenta-se o fluxo responsável pela exclusão de serviços do sistema. A Tabela 13 descreve o processo de exclusão definitiva de um serviço, que é realizada mediante confirmação por parte do administrador. Essa funcionalidade é importante para manter apenas os serviços ativos.

Tabela 13 - Fluxo de eventos excluir Serviço

Caso de uso: Excluir Serviço
Ator: Administrador

Pré-condições:

- O administrador deve estar autenticado no sistema.
 - O serviço a ser excluído deve existir no banco de dados.
-

Fluxo Principal:

1. O administrador acessa o painel do sistema.
 2. Seleciona um serviço na lista.
 3. O sistema exibe os detalhes do serviço.
 4. O administrador confirma a exclusão.
 5. O sistema remove o serviço do banco de dados.
 6. O fluxo é encerrado.
-

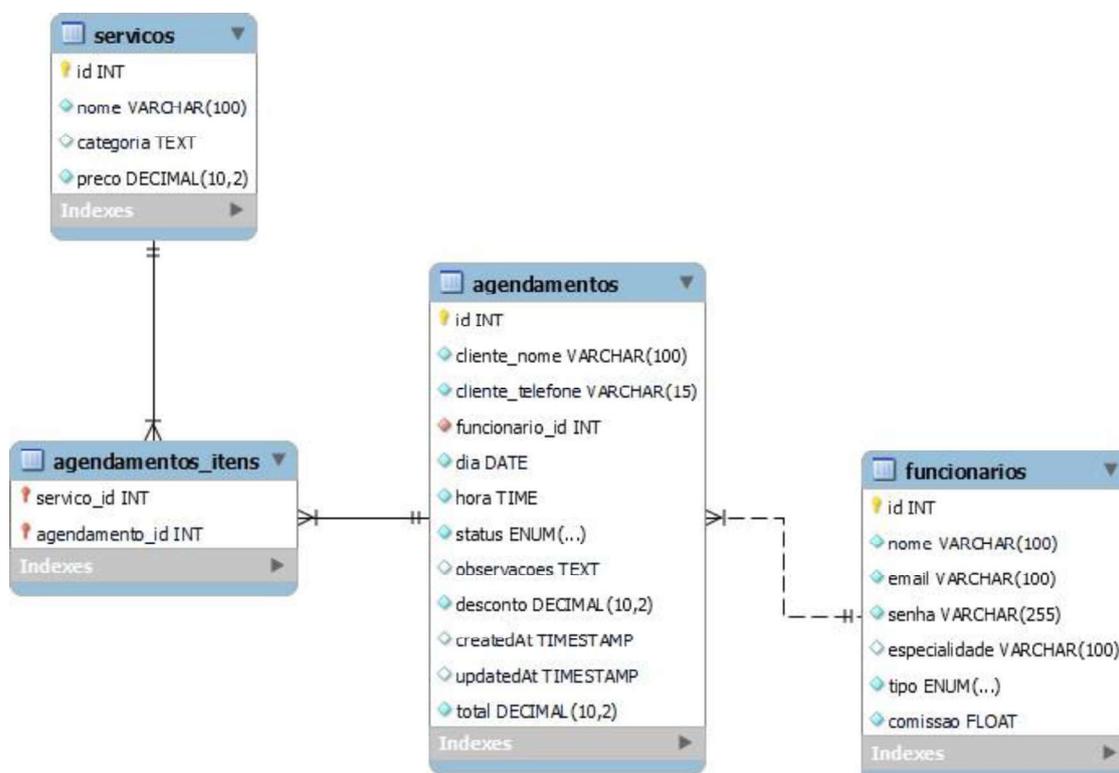
Pós-condições: O serviço é removido do banco de dados.

Fonte: Autoras, 2025

2.2 BANCO DE DADOS

Para realizar a modelagem de banco de dados, optou-se pela criação de um diagrama utilizando o Modelo Entidade-Relacionamento (MER), conforme Ilustração 3, que é uma técnica de modelagem de dados amplamente utilizada no desenvolvimento das estruturas e relacionamentos dentro do banco de dados, fazendo uma representação visual de como irá funcionar o banco de dados, sendo essencial para o planejamento da arquitetura, oferecendo aos desenvolvedores uma ideia clara e precisa sobre toda a estrutura e interações dos dados.

Ilustração 3 - Modelo Entidade-relacionamento



Fonte: Autoras, 2025

Para o armazenamento dos dados foi utilizado Sistema gerenciador de banco de dados (SGBD) MySQL, que segundo Erickson

Embora o MySQL tenha ganhado destaque há quase três décadas, ele não mostra sinais de declínio e está classificado como o segundo banco de dados mais popular, perdendo apenas para o Oracle Database, de acordo com a DB-Engines. O MySQL é versátil o suficiente para sustentar uma ampla variedade de aplicações, desde

pequenos projetos pessoais até sistemas críticos de negócios de nível empresarial, e tem o apoio de uma grande e entusiasta comunidade de código aberto. (ERICKSON, 2024).

Além de sua ampla utilização e popularidade entre a comunidade, a gratuidade deste SGDB reduz os custos de implementação e manutenção do sistema.

Outro fator a ser considerado na escolha do *MySQL* é por se tratar de um banco de dados relacional, pois organiza os dados em tabelas estruturadas, permitindo a criação de relacionamentos entre diferentes entidades, assim como feito na parte da criação do diagrama MER. Os bancos de dados relacionais utilizam da linguagem *SQL (Structured Query Language)* para a manipulação das informações, que garante uma consistência, integridade e eficiência na execução de consultas feitas ao banco de dados.

A seguir, apresenta-se o dicionário de dados das principais tabelas do sistema BeleSys. O dicionário descreve os campos, seus tipos, restrições e finalidade, com o objetivo de detalhar a estrutura lógica implementada no banco de dados relacional.

A Tabela 14 representa os serviços relacionados aos procedimentos oferecidos pelo salão de beleza, como corte ou hidratação. Essa estrutura permite que os serviços sejam associados posteriormente aos agendamentos realizados.

Tabela 14 - Tabela de serviços

Campo	Tipo de Dado	Descrição	Restrições/Observações
id	INT	Identificador único	Chave Primária, Auto-Incremento
nome	VARCHAR(100)	Nome do serviço	Não nulo
categoria	TEXT	Categoria do serviço (e.g., 'corte', 'pintura')	Texto livre
preço	DECIMAL(10,2)	Preço do serviço	Valor monetário com 2 casas decimais

Fonte: Autoras, 2025.

A seguir a Tabela 15, que possui os profissionais vinculados ao salão, seus dados de acesso, especialidades e comissões. As informações

armazenadas aqui são fundamentais para a definição de permissões de uso do sistema e para vincular aos agendamentos.

Tabela 15 - Tabela de Funcionarios

Campo	Tipo de Dado	Descrição	Restrições/Ob-servações
id	INT	Identificador único do funcionário.	Chave Primária, Auto-Incremento
nome	VARCHAR(100)	Nome completo do funcionário.	Não nulo
email	VARCHAR(100)	Endereço de e-mail do funcionário.	Único, utilizado para login/contato
senha	VARCHAR(255)	Senha criptografada do funcionário.	Armazenada como hash
especialidade	VARCHAR(100)	Especialidade do funcionário (e.g., cabeleireiro, manicure).	Pode ser uma lista pré-definida
tipo	ENUM(...)	Tipo de funcionário (e.g., 'admin', 'funcionario').	Define o nível de acesso e a função
comissao	FLOAT	Percentual de comissão do funcionário sobre os serviços.	Valor decimal.

Fonte: Autoras, 2025

A tabela de agendamentos, representada na tabela 16, armazena todos os atendimentos marcados pelos clientes. Contendo informações sobre o cliente, data e hora do atendimento, status e outros dados relacionados ao serviço agendado.

Tabela 16 - Tabela de Agendamentos

Campo	Tipo de Dado	Descrição	Restrições/Obser-vações
id	INT	Identificador único do agendamento.	Chave Primária, Auto-Incremento (geralmente)
cliente_nome	VARCHAR(100)	Nome do cliente.	Não nulo
cliente_telefone	VARCHAR(15)	Telefone de contato do cliente.	Formato variável para números de telefone.
funcionario_id	INT	ID do funcionário responsável pelo agendamento.	Chave Estrangeira referenciando funcionarios.id.
dia	DATE	Data do agendamento.	Formato 'AAAA-MM-DD'.

Campo	Tipo de Dado	Descrição	Restrições/Observações
hora	TIME	Hora do agendamento.	Formato 'HH:MM:SS'.
status	ENUM(...)	Status atual do agendamento.	e.g., 'agendado', 'cancelado', 'concluído'.
observacoes	TEXT	Observações adicionais sobre o agendamento.	Campo opcional.
desconto	DECIMAL(10,2)	Valor de desconto aplicado ao agendamento.	Valor monetário com 2 casas decimais.
createdAt	TIMESTAMP	Data e hora de criação do registro do agendamento.	Preenchido automaticamente na criação.
updatedAt	TIMESTAMP	Última data e hora de atualização do agendamento.	Atualizado automaticamente a cada modificação.
total	DECIMAL(10,2)	Valor total final do agendamento (após descontos).	Calculado a partir da soma dos serviços e descontos.

Fonte: Autoras, 2025

A tabela `agendamentos_itens`, representado pela Tabela 17, estabelece o relacionamento entre os agendamentos e os serviços selecionados. Como um agendamento pode envolver múltiplos serviços, essa tabela associativa garante a correta vinculação e integridade dos dados.

Tabela 17 - Tabela `Agendamentos_itens`

Campo	Tipo de Dado	Descrição	Restrições/Observações
servico_id	INT	ID do serviço incluído no agendamento.	Chave Estrangeira referenciando <code>servicos.id</code> .
agendamento_id	INT	ID do agendamento ao qual o item pertence.	Chave Estrangeira referenciando <code>agendamentos.id</code> .
		Chave Composta	(<code>servico_id</code> , <code>agendamento_id</code>) - Garante unicidade de um serviço em um agendamento.

Fonte: Autoras, 2025

2.3 CAMADA DE NEGÓCIO

A principal funcionalidade do sistema *Belesys* é o gerenciamento de agendamentos, feito através de um calendário interativo permitindo que o administrador gerencie e controle os atendimentos e o funcionário se organize com base nos serviços que lhe foram atribuídos.

No processo de criação do agendamento há uma validação para impedir a sobreposição de horários, verificando se um funcionário já possui um atendimento para aquele mesmo dia e horário, conforme o método *CriarAgendamento* presente na *API* e representado na Ilustração 4, evitando conflitos e protegendo um dos processos mais importantes de todo o sistema.

Ilustração 4 - método criar agendamento

```

1  const criarAgendamento = async (req, res) => {
2    const { cliente_nome, cliente_telefone, funcionario_id, dia, hora, status, observacoes, desconto = 0, Servicos } = req.body;
3
4    try {
5      // Verifica se já existe um agendamento para o mesmo funcionário, dia e hora
6      const agendamentoExistente = await Agendamento.findOne({
7        where: { funcionario_id, dia, hora }
8      });
9      if (agendamentoExistente) {
10       return res.status(400).json({ message: 'Já existe um agendamento para este funcionário neste horário.' });
11     }
12
13     // Verifica se serviços foram enviados
14     if (!Servicos) {
15       return res.status(400).json({ message: 'É necessário informar ao menos um serviço.' });
16     }
17     const servicoIds = Servicos.map(s => s.id);
18     // Busca os serviços e soma os preços
19     const servicosSelecionados = await Servico.findAll({
20       where: { id: servicoIds }
21     });
22
23     const totalServicos = servicosSelecionados.reduce((sum, s) => sum + parseFloat(s.preco), 0);
24     const valorDesconto = parseFloat(desconto) || 0;
25
26     const novoAgendamento = await Agendamento.create({
27       cliente_nome, cliente_telefone, funcionario_id, dia, hora, status, observacoes, desconto: valorDesconto, total: totalServicos
28     });
29
30     await novoAgendamento.setServicos(servicoIds);
31
32     res.status(201).json({ message: 'Agendamento criado com sucesso', agendamento: novoAgendamento });
33   } catch (error) {
34     res.status(500).json({ message: 'Erro ao criar agendamento', error });
35   }

```

Fonte: Autoras, 2025

Outro importante método das funções *CRUD* (*Create, Read, Update e Delete*) é o *atualizarAgendamento*. O método representado na Ilustração 5 possibilita a modificação dos dados relacionados ao cliente e ao serviço prestado, além de um elemento essencial: o *status* do agendamento.

Ilustração 5 - método atualizar agendamento

```

1  const atualizarAgendamento = async (req, res) => {
2    const { id } = req.params;
3    const { cliente_nome, cliente_telefone, funcionario_id, dia, hora, status, observacoes, desconto, servicos } = req.body;
4
5    try {
6      const agendamento = await Agendamento.findByPk(id);
7      if (!agendamento) {
8        return res.status(404).json({ message: 'Agendamento não encontrado' });
9      }
10
11     // Atualiza os campos
12     agendamento.cliente_nome = cliente_nome || agendamento.cliente_nome;
13     agendamento.cliente_telefone = cliente_telefone || agendamento.cliente_telefone;
14     agendamento.funcionario_id = funcionario_id || agendamento.funcionario_id;
15     agendamento.dia = dia || agendamento.dia;
16     agendamento.hora = hora || agendamento.hora;
17     agendamento.status = status || agendamento.status;
18     agendamento.observacoes = observacoes || agendamento.observacoes;
19     agendamento.desconto = desconto || agendamento.desconto;
20
21     // Atualiza os serviços associados ao agendamento
22     if (servicos) {
23       await agendamento.setServicos(servicos);
24     }
25
26     await agendamento.save();
27     res.status(200).json({ message: 'Agendamento atualizado com sucesso', agendamento });
28   } catch (error) {
29     res.status(500).json({ message: 'Erro ao atualizar agendamento', error });
30   }
31 };

```

Fonte: Autoras, 2025

O *status* pode assumir três valores: Agendado, que é o padrão para novos registros; Finalizado, quando o atendimento é concluído; Cancelado, quando não será realizado. Essa funcionalidade possibilita ao administrador monitorar a execução dos serviços, identificar padrões de cancelamento e manter um histórico atualizado dos atendimentos. Além disso, a atualização do status impacta diretamente nos relatórios gerenciais, auxiliando na análise do desempenho do salão e contribuindo para uma gestão mais estratégica.

Complementando a gestão dos agendamentos e visando reduzir a taxa de faltas, o sistema *Belesys* conta com um módulo automatizado de envio de lembretes via *WhatsApp*, garantindo que os clientes sejam lembrados sobre seus atendimentos, contribuindo para a diminuição de ausências e a otimização dos serviços prestados.

Essa funcionalidade é implementada por meio de uma combinação da *API Venom-Bot*, responsável pela comunicação via *WhatsApp* com a biblioteca *node-cron* para a automatização das notificações. A combinação dessas tecnologias assegura que os lembretes sejam precisos sem a necessidade de intervenção manual.

Na Ilustração 6 é exibido a configuração do *node-cron*, onde a expressão '0 8 * * *' define o disparo automático da função responsável por enviar os lembretes às 8 horas, todos os dias e seguindo uma lógica condicional, a mesma lógica é utilizada para os lembretes das 20 horas.

Ilustração 6 - Configuração do node-cron

```

2  cron.schedule('0 8 * * *', lembretesTarde, {
3    timezone: 'America/Sao_Paulo'
4  }); // 8h
5
6  cron.schedule('0 20 * * *', lembretesManha, {
7    timezone: 'America/Sao_Paulo'
8  }); // 20h

```

Fonte: Autoras, 2025

A implementação mostrada na Ilustração 7 demonstra a lógica condicional para o envio dos lembretes. Quando o sistema identifica que são 8h da manhã é feita uma busca no banco de dados por todos os agendamentos marcados para a partir das 12h do mesmo dia, filtrando apenas os que possuem status "agendado".

Para cada um desses agendamentos, o sistema compõe uma mensagem personalizada com o nome do cliente, detalhes do serviço e horário agendado, enviando-a via *WhatsApp*.

Ilustração 7 - Implementação da lógica de lembretes

```

1 // Lembretes da tarde (enviados às 8h do mesmo dia)
2 const lembretesTarde = async () => {
3   const hoje = new Date().toISOString().split('T')[0];
4
5   const agendamentos = await Agendamento.findAll({
6     where: {
7       dia: hoje,
8       hora: { [Op.gte]: '12:00' },
9       status: 'agendado',
10    },
11    include: [{ model: Servico, as: 'Servicos', attributes: ['nome'] }],
12  });
13
14  agendamentos.forEach((agendamento) => {
15    const nomesServicos = agendamento.Servicos.map(s => s.nome).join(', ');
16    const msg = `Olá ${agendamento.cliente_nome}, você tem um agendamento para ${nomesServicos} às ${agendamento.hora}. \nEsperamos você!`;
17    enviarLembreteWhatsApp(agendamento.cliente_telefone, msg);
18  });
19 };

```

Fonte: Autoras, 2025

A mesma lógica se aplica para o envio às 20h, onde o sistema busca os agendamentos do dia seguinte com horários antes das 12h, mantendo a estrutura de código, apenas ajustando os parâmetros de consulta.

Para garantir a segurança dos dados dos usuários, o sistema implementa a criptografia de senhas no banco de dados utilizando a biblioteca *bcrypt*.

Sempre que um novo funcionário é cadastrado ou sua senha é alterada, o sistema aplica um *hash* antes de armazená-la, tornando-a irreversível e protegendo contra acessos indevidos.

Essa funcionalidade é implementada através dos *hooks* do *Sequelize*, que conforme descrito na documentação oficial, se trata de funções que são chamadas antes e depois que as chamadas no *sequelize* são executadas, também são conhecidas como *callbacks* ou eventos de ciclo de vida.

Ilustração 8 - Função de criptografia de senhas

```

1 hooks: {
2   //função para criptografar a senha
3   beforeCreate: async (funcionario) => {
4     if (funcionario.senha) {
5       const salt = await bcrypt.genSalt(10); // Gera um salt
6       funcionario.senha = await bcrypt.hash(funcionario.senha, salt); // Faz o hash da senha
7     }
8   },
9   //função para criptografar novamente a senha pós alteração
10  beforeUpdate: async (funcionario) => {
11    if (funcionario.changed('senha')) { // Verifica se a senha foi alterada
12      const salt = await bcrypt.genSalt(10);
13      funcionario.senha = await bcrypt.hash(funcionario.senha, salt);
14    }
15  },
16 },

```

fonte: Autoras, 2025

Os *hooks* *beforeCreate* e *beforeUpdate*, apresentados da Ilustração 8, garantem que toda senha cadastrada ou alterada passe pelo processo de criptografia.

2.4 CAMADA DE APRESENTAÇÃO

A seguir serão apresentadas as principais telas do sistema, juntamente com suas características técnicas, seguido de algumas observações sobre seu desenvolvimento.

Na primeira interface, representada na Ilustração 9, é a tela de agenda, que permite visualizar os agendamentos registrados no sistema. Cada usuário visualizará apenas os serviços atribuídos a ele.

A primeira tela exibe por padrão, os agendamentos do mês atual, permitindo ao usuário escolher entre 3 diferentes modos de visualização: mensal, semanal e diária. Além disso, a navegação possibilita que sejam visualizados os meses anteriores, como também os agendamentos futuros, proporcionando uma maior flexibilidade no gerenciamento das marcações.

Ilustração 9 - Tela de Agenda



Fonte: Autoras, 2025

Pela visualização mensal que é a exibição padrão desta tela, o usuário poderá ver como anotações as informações iniciais de horário e cliente a ser atendido. Nos demais modos segue o mesmo objetivo, podendo os agendamentos serem visualizados de hora em hora.

Ilustração 10 - Modal de cadastro e edição

Novo Agendamento

(*) Campos obrigatórios

*Cliente: *Telefone (WhatsApp):

*Data: *Hora:

*Serviço:

*Funcionário:

Valor Total: Desconto (opcional):

Valor com Desconto:

Observações:

Salvar Evento

Fonte: Autoras, 2025

Já na tela apresentada pela Ilustração 10, há o modal que é responsável pelo cadastro e edição de agendamentos, sendo acessado através da tela de agenda, clicando em um dia ou horário, a depender do modo de exibição, ou pelo botão com o sinal de + (mais). Neste modal, ao realizar um novo agendamento, deve ser inserido todos os dados necessários para o agendamento, como o cliente, dados de contato do cliente, data e hora, serviço a ser feito, funcionário atribuído, valores e observações.

Quando este modal é utilizado para a edição de um agendamento, os dados já inseridos anteriormente serão exibidos, sendo necessário mudar apenas o necessário para o agendamento.

A Ilustração 11 é a apresentação do modal responsável por mostrar os detalhes de um agendamento, podendo acessá-lo pela tela de agenda também ao clicar em um agendamento existente. Aqui é possível ver todos os detalhes do agendamento que foi inserido no sistema, podendo atualizar a marcação do status, sendo por padrão um agendamento ficar marcado como “Agendado”.

Ilustração 11 - Modal de detalhes

Detalhes do Agendamento

Informações do Cliente
 Nome: toninha
 Telefone: 12345

Profissional
 Nome: Mayumi

Data e Hora
 Data: 24/04/2025
 Hora: 10:00:00

Status
 AGENDADO

Serviços Contratados

Serviço	Preço
corte de cabelo feminino curto	30,00
progressiva em cabelo curto	110,00
Desconto	-5,00
Total	135,00

Observações
 Nenhuma observação registrada

Alterar Status

Agendado Concluído Cancelado

Editar Fechar Salvar Status

Fonte: Autoras, 2025

Pela Ilustração 12, há a apresentação da tela de Relatórios, onde a gerência poderá gerar relatórios sobre os serviços concluídos dentro de um intervalo de datas selecionadas pelo usuário. Antes dos campos para a inserção das datas há uma breve explicação ao usuário das etapas para que ele esteja atento ao gerar o relatório.

Ilustração 12 - Tela de relatórios

Gerar relatório

Para gerar um relatório de atendimentos, siga os passos abaixo:

1. Selecione a **Data Inicial** no campo correspondente.
2. Selecione a **Data Final** no campo correspondente.
Escolha até qual data deseja incluir no relatório. A data final deve ser igual ou posterior à data inicial.
3. Clique no botão "**Gerar Relatório**".

O relatório exibirá todos os atendimentos concluídos dentro do intervalo de datas selecionado.

Data Início: 30/05/2025

Data Fim: 30/05/2025 Incluir cancelados

Gerar Relatório

Fonte: Autoras, 2025

Todas as interfaces apresentadas contam com o design responsivo para se adaptarem a diferentes dispositivos. As demais telas desse sistema se encontram no Apêndice A.

3 RESULTADO

Este capítulo tem como objetivo apresentar os resultados obtidos com o aplicativo desenvolvido, além de relatar a experiência prática de uso e os testes de usabilidade realizados.

A finalidade da aplicação dos testes e da coleta de *feedback* era identificar oportunidades para aprimoramento no sistema, além de confirmar se os objetivos propostos nas fases anteriores de desenvolvimento foram efetivamente alcançados na prática.

3.1 TESTES DE USABILIDADE

Os testes de usabilidade são um passo crucial na avaliação da interação dos usuários com o sistema, possibilitando a identificação de aspectos positivos e áreas que necessitam de aprimoramento. De acordo com Nielsen.

O teste de usuário com usuários reais é o método de usabilidade mais fundamental e, de certa forma, insubstituível, pois fornece informações diretas sobre como as pessoas usam os computadores e quais são seus problemas exatos com a interface concreta que está sendo testada. (Nielsen, 1994)

Para alcançar esse objetivo, foram criadas tarefas específicas, conforme representadas na Tabela 14, para o cenário de usuário com a função de Administrador, onde os participantes foram convidados a utilizar o sistema de acordo com situações reais de uso, permitindo uma avaliação mais precisa da experiência do usuário.

Tabela 18 – tarefas do teste de usabilidade

Nº da tarefa	Tarefa
1	Fazer login com as credenciais de teste: e-mail: admin@e.com senha:123
2	Cadastrar um funcionário: Nome: Charlote

N° da tarefa	Tarefa
	Email: charlie@e.com, cargo: cabeleireiro, Tipo: Funcionário, Gerar uma senha aleatória, Comissão: 30%
3	Adicionar 2 serviços: corte de cabelo feminino, categoria: cabeleireiro, valor: 40 progressiva, categoria: cabeleireiro, valor: 120
4	Criar um atendimento: dia 05/03/2025 às 10 horas, cliente Larissa, telefone: 13991201320, serviços corte e progressiva, inclua a funcionária Charlotte, desconto R\$10 e observação: corte somente com tesoura
5	Marcar o atendimento como concluído
6	Gerar um relatório: 01/03/2025 a 01/04/2025
7	Fazer logout (sair do sistema)

Fonte: Autoras, 2025.

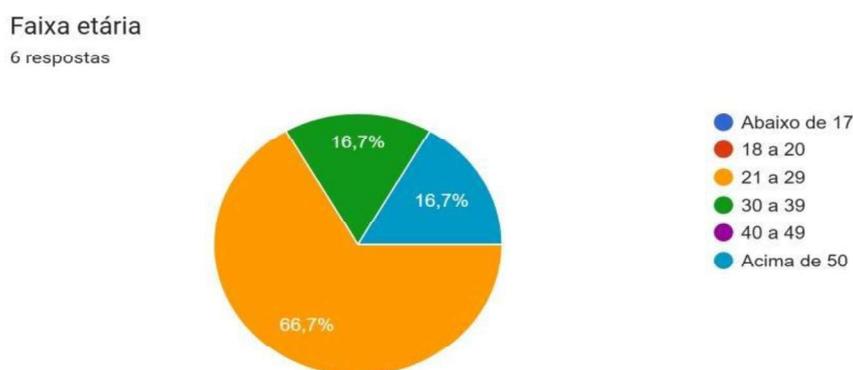
Feita a definição das tarefas, foram selecionadas 6 pessoas de diferentes idades e experiências para participar da avaliação. Dentre os selecionados foram escolhidas pessoas com familiaridade no uso de computador no seu dia a dia e pessoas com pouco acesso.

Os testes de usabilidade foram realizados no período de 4 a 8 de março de 2025, de forma remota e presencial, conforme a disponibilidade de cada convidado. Todos os participantes são residentes da Baixada Santista, em sua maioria na cidade de São Vicente, no estado de São Paulo.

Após a finalização dos testes, os participantes responderam de forma anônima a um questionário, desenvolvido com base no *WAMMI (Website Analysis and Measurement Inventory)*, permitindo aos participantes compartilharem como foi a experiência com a aplicação.

Conforme a Ilustração 13, os testes contaram com participantes de diferentes faixas etárias. A maioria (66,7%) tinha entre 21 a 29 anos, enquanto 16,7% estavam entre 30 a 39 anos e 16,7% tinham mais de 50 anos. Essa diversidade ajuda a avaliar a experiência de uso do sistema de forma mais ampla.

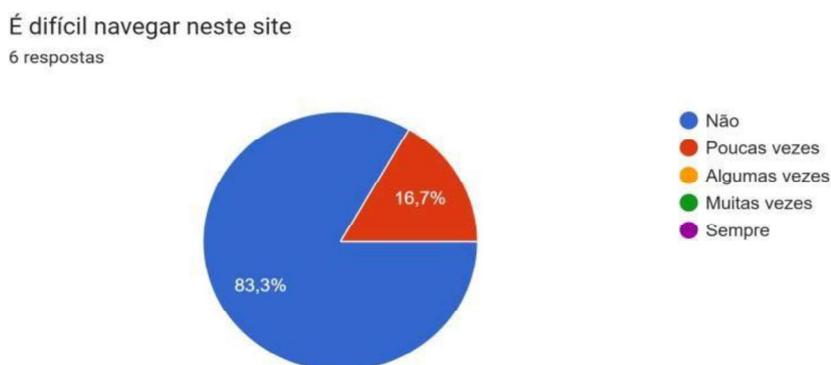
Ilustração 13 - Gráfico da pergunta 1



Fonte: Autoras, 2025

A Ilustração 14 apresenta o gráfico relacionado às dificuldades encontradas pelos participantes durante a navegação no sistema.

Ilustração 14: Gráfico da resposta 2



Fonte: Autoras, 2025

Observa-se que a maioria (83,3%) dos usuários declarou não ter enfrentado dificuldades, enquanto 16,7% relataram ter encontrado obstáculos em poucas ocasiões. Esses resultados indicam que, de modo geral, a navegação foi considerada intuitiva pelos participantes, evidenciando a boa usabilidade da aplicação.

Conforme apresentado na Ilustração 15, os participantes avaliaram positivamente a facilidade de localizar informações no sistema. As respostas obtidas indicam um consenso entre os usuários, sendo que 50% afirmaram "com certeza" encontrar as informações com facilidade, enquanto os 50% restantes responderam "sim". Esses dados demonstram que a aplicação atende de forma satisfatória às expectativas dos usuários quanto à clareza e à organização das informações disponibilizadas na interface.

Ilustração 15: Gráfico da pergunta 3

Este site me ajuda a encontrar o que eu estou procurando.
6 respostas

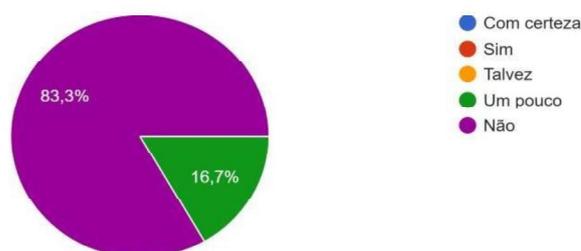


Fonte: Autoras, 2025

A Ilustração 16 apresenta o gráfico em que a maioria dos participantes (83,3%) relatou não ter enfrentado dificuldades para se orientar no sistema, enquanto 16,7% apontaram alguma dificuldade leve.

Ilustração 16: Gráfico da pergunta 4

Aprender a encontrar meu caminho de volta neste site é um problema
6 respostas



Fonte: Autoras, 2025

Esses dados reforçam que, de modo geral, a navegação foi considerada clara pelos usuários, contribuindo para uma experiência positiva durante o uso da aplicação.

Os demais gráficos de respostas estão presentes no Apêndice B.

3.2 CONCLUSÃO

Os testes de usabilidade aplicados demonstraram que o sistema atendeu às expectativas da maioria dos usuários em aspectos como facilidade de navegação, localização de informações e clareza na transição entre as telas. A diversidade etária dos participantes contribuiu para uma avaliação mais ampla, destacando que o sistema é acessível tanto para usuários mais jovens quanto para as demais idades. As avaliações indicaram que a navegação foi considerada intuitiva, com poucos relatos de dificuldades.

Durante os testes, os usuários desempenharam tarefas específicas para a função de Administrador, como registrar um funcionário, adicionar novos serviços, criar e concluir atendimentos, e gerar relatórios. Todos os participantes, de forma unânime, relataram que o sistema os ajudou a encontrar facilmente o que buscavam. Esses resultados reforçam a eficiência e a usabilidade do sistema, validando as escolhas feitas durante seu desenvolvimento.

3.3 PRÓXIMOS PASSOS

Tendo todos os objetivos do projeto alcançados, os próximos passos envolvem a melhoria contínua do que já foi desenvolvido, com o foco principal nos pontos que apresentam margem para aperfeiçoamento, conforme identificado nos testes de usabilidade.

Além disso, estão previstas novas funcionalidades que visam ampliar a eficiência e personalização do sistema, tais como:

- **Parametrização das notificações:** permitir que o administrador defina com maior liberdade quando e como deseja notificar seus clientes, personalizando horários e tipos de lembretes;
- **Desenvolvimento de aplicativo nativo:** criar uma versão 100% mobile do sistema, com aplicativo disponível para download nas lojas *Play Store* e *App Store*, oferecendo mais mobilidade para os usuários no dia a dia;

- **Novos relatórios com foco financeiro:** incluir análises detalhadas sobre receitas, serviços mais lucrativos, comissões e fluxo de caixa, atendendo à demanda por uma gestão financeira mais precisa.

Essas futuras melhorias buscam acompanhar as necessidades práticas dos salões de beleza e garantir que o sistema continue evoluindo, de acordo com as demandas que os salões de beleza venham a apresentar com o uso do sistema, bem como a adaptação a eventuais novas regras de negócio.

REFERÊNCIAS BIBLIOGRÁFICAS

ERICKSON, Jeffrey. **MySQL: Entendendo o que é e como é usado**. Oracle, 29 ago. 2024. Disponível em: <https://www.oracle.com/br/mysql/what-is-mysql/>. Acesso em: 14 mar. 2025.

FORTES, Caio. **Agendamento online: revolução na gestão de salões**. Frizzar, 2024. Disponível em: <https://frizzar.com.br/blog/agendamento-online-revolucao-gestao-saloes/>. Acesso em: 19 maio 2025.

FOWLER, Martin. **UML essencial: um breve guia para a linguagem-padrão de modelagem de dados**. 3. ed. Porto Alegre: Bookman, 2005.

MICROSOFT. **TypeScript: JavaScript with syntax for types**. 2024. Disponível em: <https://www.typescriptlang.org/>. Acesso em: 24 mar. 2025.

NIELSEN, Jakob. **Usability Engineering**. San Francisco: Morgan Kaufmann, 1994.

OWASP Foundation. **OWASP Top 10 Security Risks**. 2021. Disponível em: <https://owasp.org/www-project-top-ten/>. Acesso em 24 mar. 2025.

SEBRAE. **Como o agendamento on-line pode potencializar o atendimento**. Disponível em: <https://sebraeplay.com.br/content/como-o-agendamento-on-line-pode-potencializar-o-atendimento>. Acesso em: 19 maio 2025.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson, 2011.

APÊNDICE A – CAMADA DE APRESENTAÇÃO

Ilustração 17 - Tela de funcionários cadastrados

BeleSys

Relatórios

Agenda

Serviços

Funcionários

Administrador

Funcionários Cadastrados

Buscar por nome, email ou especialidade

Novo Funcionário

Nome	Email	Especialidade
Administrador	admin@gmail.com	Administrador

Fonte: Autoras, 2025

Ilustração 18 - Tela de visualização de serviços cadastrados

BeleSys

Relatórios

Agenda

Serviços

Funcionários

Administrador

Serviços Cadastrados

Filtro de busca:

Novo Serviço

Serviço	Categoria	Valor
corte masculino	cabelereiro	R\$ 25.00

Fonte: Autoras, 2025

Ilustração 19 - Modal do usuário

Perfil do Usuário

Informações do Administrador

Nome: Administrador

Email: admin@gmail.com

Comissão: 10%

Especialidade: Administrador

Trocar Senha

Senha Atual

Nova Senha

Confirmar Nova Senha

Fechar Salvar Senha

Fonte: Autoras, 2025

Ilustração 20 - Tela de comissões

BeleSys

- Minha agenda
- Minhas comissões

Minhas Comissões

Filtrar por período:

Últimos 7 dias

Total: R\$ 72.00

Data	Comissão (R\$)
20/05/2025	R\$ 9.00
18/05/2025	R\$ 9.00
16/05/2025	R\$ 54.00

Mayumi

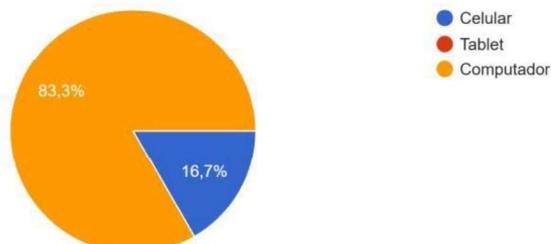
Fonte: Autoras, 2025

APÊNDICE B - RESULTADO

Ilustração 21: Gráfico da pergunta 5

Qual a plataforma utilizada para realizar o teste?

6 respostas

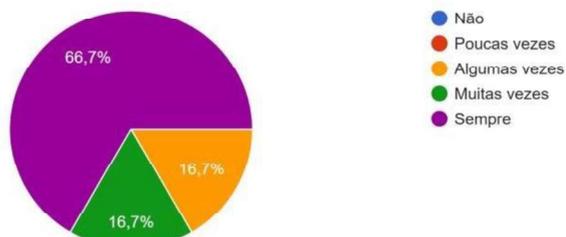


Fonte: Autoras, 2025

Ilustração 22: Gráfico da pergunta 6

Eu pude encontrar facilmente o que eu queria neste site

6 respostas



Fonte: Autoras, 2025

Ilustração 23: Gráfico da pergunta 7

Eu me sinto no controle quando utilizo este web site.

6 respostas

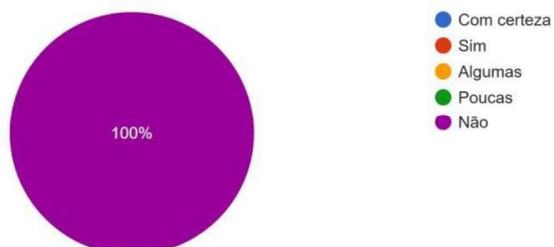


Fonte: Autoras, 2025

Ilustração 24: Gráfico da pergunta 8

Este aplicativo é muito lento

6 respostas



Fonte: Autoras, 2025