SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL

ESCOLA E FACULDADE SENAI “GASPAR RICARDO JUNIOR”

CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS

**JOÃO VITOR LIMA MELLO**

**KEREN STEVAUX ANASTÁCIO**

**LAURA LOPES SOARES DA SILVA**

**MARIA FERNANDA SOARES OLIVEIRA**

SISTEMA DE DELÍCIAS FER LOPES

SOROCABA

2025

**JOÃO VITOR LIMA MELLO**

**KEREN STEVAUX ANASTÁCIO**

**LAURA LOPES SOARES DA SILVA**

**MARIA FERNANDA SOARES OLIVEIRA**

SISTEMA DE DELÍCIAS FER LOPES

Projeto integrador apresentado à Escola e Faculdade SENAI “Gaspar Ricardo Junior”, como como parte do desenvolvimento pedagógico do curso.

Orientador: Prof. Esp. Vedilson Prado

SOROCABA

2025

Dedico este trabalho a nossa empresa parceira;

a todos os professores que acreditaram no meu trabalho;

a todos os que, direta ou indiretamente, contribuíram

para a conclusão de mais esta etapa.

**AGRADECIMENTOS**

Agradeço, primeiramente, a Deus, pela realização deste trabalho.

A todos os professores que contribuíram para o meu enriquecimento cultural ao longo desses dois anos de graduação.

Em especial, ao meu orientador, Prof. Dr. Vedilson Prado, o apoio, as conversas e discussões no processo de elaboração desta monografia, que compartilhou comigo sua sabedoria, conduzindo o trabalho de maneira firme, porém amigável, deixando uma contribuição extremamente importante e positiva nesta fase das nossas vidas acadêmicas.

**RESUMO**

Este trabalho apresenta o desenvolvimento do site **Delícias Fer Lopes**, uma plataforma digital, buscando criar um vínculo afetivo com os clientes. O principal objetivo foi unir funcionalidade e experiência personalizada, transformando o site em uma facilitação de encomendas, onde cada cliente sente que está adquirindo produtos de alguém especial. Entre as funcionalidades implementadas, destacam-se: cardápio interativo com memórias afetivas, montagem personalizada de caixas de doces com pré-visualização e sistema de sugestões automáticas de produtos. O sistema foi desenvolvido utilizando tecnologias como Java Spring no backend, MySQL como banco de dados e HTML, CSS e JavaScript no frontend. A metodologia adotada baseou-se em práticas ágeis com **Scrum**, dividindo o projeto em sprints de entrega incremental. Os testes de usabilidade demonstraram uma aceitação positiva por parte dos usuários, enquanto a validação técnica evidenciou estabilidade e segurança na comunicação entre os módulos. Conclui-se que a solução proposta atende aos requisitos funcionais e não funcionais estabelecidos, sendo adequada para a Delícias Fer Lopes.

**Palavras-Chave:** Sites interativos; Doces artesanais; Experiência do usuário; Delícias Fer Lopes.

**LISTA DE FIGURAS**

[Figura 1- Logomarca do Sistema SAMEO 8](#_Toc205012802)

[Figura 2- Persona do público-alvo 9](#_Toc205012803)

[Figura 3- Diagrama de Arquitetura 13](#_Toc205012804)

[Figura 4- Diagrama UML - Paciente e Agendamento 13](#_Toc205012805)

[Figura 5- Modelagem Conceitual 14](#_Toc205012806)

[Figura 6- Modelagem Lógica 14](#_Toc205012807)

[Figura 7- Modelagem Física 15](#_Toc205012808)

[Figura 8- Página inicial do sistema. 17](#_Toc205012809)

[Figura 9- Página do perfil de usuário, apresentada após o login 18](#_Toc205012810)

**LISTA DE TABELAS**

[Tabela 1- Lista de Tecnologias e Ferramentas 10](#_Toc205012811)

[Tabela 2- Glossário de Termos 10](#_Toc205012812)

[Tabela 3- Cronograma de execução das atividades 11](#_Toc205012813)

**SUMÁRIO**

[1. INTRODUÇÃO 7](#_Toc205012815)

[2. METODOLOGIA 8](#_Toc205012816)

[2.1. Visão do Projeto 8](#_Toc205012817)

[2.1.1. Nome do projeto 8](#_Toc205012818)

[2.1.2. Justificativa 8](#_Toc205012819)

[2.1.3. Objetivo 8](#_Toc205012820)

[2.1.4. Público-alvo 8](#_Toc205012821)

[2.2. Escopo 9](#_Toc205012822)

[2.2.1. Funcionalidades principais 9](#_Toc205012823)

[2.2.2. Limitações e exclusões 9](#_Toc205012824)

[2.3. Stakeholders 9](#_Toc205012825)

[2.3.1. Equipe técnica 9](#_Toc205012826)

[2.3.2. Usuários finais 10](#_Toc205012827)

[2.3.3. Clientes e patrocinadores 10](#_Toc205012828)

[2.4. Metodologias 10](#_Toc205012829)

[2.5. Tecnologias e Ferramentas 10](#_Toc205012830)

[2.6. Glossário de termos 10](#_Toc205012831)

[2.7. Cronograma 11](#_Toc205012832)

[3. REQUISITOS 11](#_Toc205012833)

[3.1. Requisitos Funcionais 11](#_Toc205012834)

[3.2. Requisitos Não Funcionais 12](#_Toc205012835)

[3.3. Casos de Uso / Histórias de Usuário 12](#_Toc205012836)

[4. ARQUITETURA E DESIGN 12](#_Toc205012837)

[4.1. Arquitetura de Software 13](#_Toc205012838)

[4.2. Diagrama UML do sistema 13](#_Toc205012839)

[4.3. Modelagem de dados 14](#_Toc205012840)

[4.3.1. Modelagem Conceitual 14](#_Toc205012841)

[4.3.2. Modelagem Lógica 14](#_Toc205012842)

[4.3.3. Modelagem Física 15](#_Toc205012843)

[4.3.4. Dicionário de dados 15](#_Toc205012844)

[5. QUALIDADE E TESTES 15](#_Toc205012845)

[5.1. Plano de Testes 15](#_Toc205012846)

[5.2. Casos de Teste 16](#_Toc205012847)

[5.3. Planejamento de testes 16](#_Toc205012848)

[5.4. Resultados dos Testes 16](#_Toc205012849)

[5.5. Testes Automatizados 16](#_Toc205012850)

[6. MANUAL DO USUÁRIO 17](#_Toc205012851)

[6.1. Guia de Navegação 17](#_Toc205012852)

[6.2. FAQs e Problemas Comuns 18](#_Toc205012853)

[7. MANUAL TÉCNICO 18](#_Toc205012854)

[7.1. Guia do Desenvolvedor 19](#_Toc205012855)

[7.1.1. Guia de Instalação 19](#_Toc205012856)

[7.2. Documentação da API 20](#_Toc205012857)

[7.2.1. Formato de exemplo de request 20](#_Toc205012858)

[7.2.2. Resposta esperada 21](#_Toc205012859)

[7.3. Padrões de Código 21](#_Toc205012860)

[7.3.1. Backend – Java (Spring Boot): 21](#_Toc205012861)

[7.3.2. Frontend – JavaScript: 21](#_Toc205012862)

[7.3.3. Boas práticas gerais: 21](#_Toc205012863)

[8. PROPOSTA COMERCIAL 21](#_Toc205012864)

[8.1. Descrição da Proposta 21](#_Toc205012865)

[8.2. Escopo Comercial 22](#_Toc205012866)

[8.2.1. Itens Inclusos 22](#_Toc205012867)

[8.2.2. Itens Exclusos: 22](#_Toc205012868)

[8.2.3. Preço e Condições de Pagamento 22](#_Toc205012869)

[8.3. Cronograma de Entrega 24](#_Toc205012870)

[8.4. Termos e Condições 24](#_Toc205012871)

[8.5. Validade da Proposta 25](#_Toc205012872)

[9. CONSIDERAÇÕES FINAIS 26](#_Toc205012873)

[REFERÊNCIAS 27](#_Toc205012874)

# INTRODUÇÃO

Com o avanço da transformação digital, diversos setores da sociedade têm buscado soluções tecnológicas com o objetivo de otimizar processos e aprimorar a experiência do usuário. No segmento de confeitaria, um dos principais desafios enfrentados pelos empreendedores é a gestão eficiente das encomendas de doces e bolos, que, em muitos casos, ainda depende de métodos manuais ou de atendimentos via redes sociais e mensagens.

Nesse contexto, este projeto propõe o desenvolvimento de uma plataforma online de encomendas de doces, integrada ao site da **Delícias Fer Lopes**. A proposta visa facilitar o processo de pedidos tanto para os clientes interessados quanto para a equipe responsável pela produção, promovendo maior agilidade, praticidade e organização.

A ideia surgiu a partir da identificação de problemas comuns no modelo atual de atendimento, como a sobrecarga nos canais de comunicação, atrasos na confirmação de pedidos e perda de potenciais clientes devido à demora no retorno. Com a digitalização desse processo, busca-se otimizar o fluxo de trabalho da confeitaria, reduzir falhas operacionais e proporcionar uma experiência mais eficiente, acessível e satisfatória para os usuários.

# METODOLOGIA

Este projeto foi desenvolvido em etapas organizadas para facilitar o planejamento e a execução das tarefas. Primeiro, foi feito um levantamento das necessidades do sistema, com base em observações do funcionamento atual da **Delícias Fer Lopes** e dos problemas enfrentados no processo de recebimento e organização das encomendas de doces.

## Visão do Projeto

### Nome do projeto

Encomendas Online via WhatsApp

Figura 1- Logomarca da Delícias Fer Lopes



Fonte: Feito pelo autor (Canva).

### Justificativa

O desenvolvimento do site Delícias da Fer surgiu da necessidade de tornar o processo de agendamento de visitas a imóveis mais ágil e eficiente, tanto para os clientes quanto para os corretores. Atualmente, esse processo costuma demandar contato direto, troca de mensagens e disponibilidade de ambas as partes, o que muitas vezes gera atrasos, sobrecarga de tarefas e até perda de potenciais clientes.  
Com a plataforma, o cliente pode selecionar o corretor de sua preferência e agendar diretamente a visita ao imóvel desejado, sem burocracia e de forma prática. Isso proporciona uma experiência mais organizada, transparente e acessível, além de otimizar o tempo dos corretores, que passam a ter maior controle sobre seus atendimentos.  
Dessa forma, o site se torna uma ferramenta essencial para modernizar o relacionamento entre a Re/max, seus corretores e os clientes, fortalecendo a credibilidade da empresa e oferecendo um serviço alinhado às expectativas do mercado atual.

### Objetivo

O objetivo deste projeto é desenvolver uma solução web que permita a realização de encomendas de doces de forma simples, rápida e automatizada, integrando o sistema diretamente ao site da **Delícias Fer Lopes**. Espera-se, com isso, melhorar a organização dos pedidos, otimizar o fluxo de produção e reforçar a imagem da confeitaria como uma marca moderna, prática e inovadora.

### Público-alvo

O público-alvo deste projeto é composto por homens e mulheres, com faixa etária entre 15 e 60 anos, pertencentes à classe média e residentes em cidades de médio a grande porte. Trata-se de indivíduos que buscam praticidade em seu cotidiano, muitos dos quais atuam em regime de trabalho remoto (home office). Esse grupo valoriza momentos especiais e demonstra interesse por produtos artesanais e personalizados, especialmente aqueles que possam ser utilizados como presentes criativos e diferenciados. Além disso, apresentam preferência por opções que unam qualidade, exclusividade e praticidade no processo de compra.

Figura 2- Persona do público-alvo



Fonte: Elaborada pelo autor (2025).

## Escopo

### Funcionalidades principais

* Encomendas de doces

### Limitações e exclusões

* Não haverá forma de pagamento

## Stakeholders

### Equipe técnica

* João Vitor – Scrum Master, Desing e Frontend
* Keren Stevaux- Desenvolvedora Backend e Analista de Erro
* Laura Lopes- Desenvolvedora Frontend
* Maria Fernanda- Desenvolvedora Backend

### Usuários finais

Os usuários finais deste projeto são consumidores urbanos, pertencentes à classe média, que buscam produtos artesanais de qualidade para diferentes finalidades. Entre eles, destacam-se os indivíduos que adquirem doces para consumo próprio, aqueles que desejam presentear amigos, familiares ou colegas com itens exclusivos, bem como clientes interessados em opções personalizadas para confraternizações e datas comemorativas. Além disso, esse público utiliza, de forma recorrente, as redes sociais digitais, como Instagram e WhatsApp, tanto como fonte de inspiração quanto como canal de compra e comunicação direta.

### Clientes e patrocinadores

## Metodologias

A metodologia adotada pelo squad é o Kanban baseada em princípios e características específicas para melhorar a eficiência e a produtividade de uma organização. É gerado um quadro que permite visualizar o fluxo do trabalho dividido em etapas do processo, como “em desenvolvimento”, “testes”, “realizado”.

## Tecnologias e Ferramentas

Tabela 1- Lista de Tecnologias e Ferramentas

Tabela

O conteúdo gerado por IA pode estar incorreto.

Fonte: Produção do autor (2025)

## Glossário de termos

Tabela 2- Glossário de Termos

(tem que acrescentar mais termos)

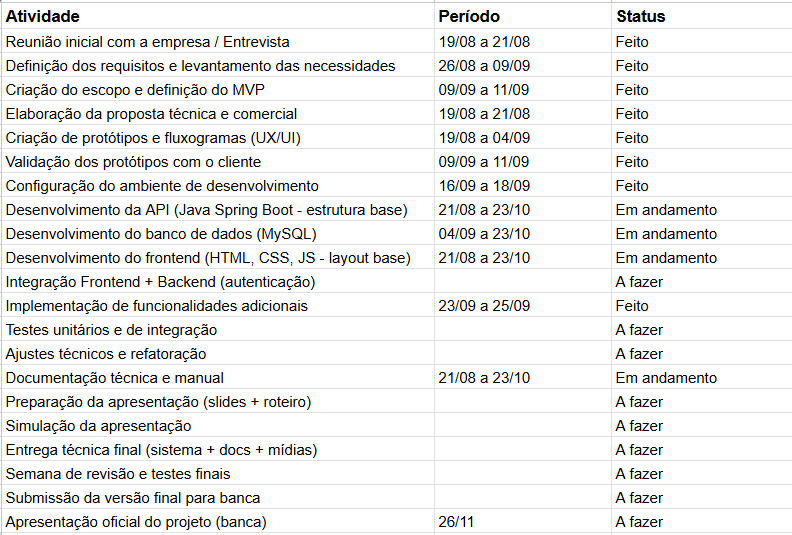
Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Fonte: Produção do autor (2025)

## Cronograma

Tabela 3- Cronograma de execução das atividades



Fonte: Produção de autor (2025)

# REQUISITOS

Esta seção define o que o sistema deve fazer (funcionalidades) e como ele deve se comportar (qualidade, desempenho, restrições). Serve como base para o desenvolvimento, testes e validação do sistema.

## Requisitos Funcionais

* RF01: O sistema deve **solicitar os dados essenciais para a entrega e contato** do cliente (nome, e-mail e telefone) no momento da finalização da compra.
* RF02: O sistema deve permitir que o **cliente visualize o catálogo de doces** com fotos, descrições e preços.
* RF03: O sistema deve enviar um e-mail de **confirmação após o envio do pedido/encomenda**.
* RF04: O sistema deve permitir que o cliente **adicione itens ao carrinho** de compras.

RF05: O sistema deve permitir que o cliente **escolha a data e o método de entrega/retirada** do pedido, informando o endereço completo se for entrega.

## Requisitos Não Funcionais

* RNF01: O sistema deve estar disponível **24 horas por dia, 7 dias por semana** para receber pedidos.
* RNF02: O tempo de resposta para carregamento da **página de catálogo de produtos** não deve ultrapassar 3 segundos.
* RNF03: As informações dos clientes (contato e endereço) devem ser armazenadas com criptografia AES-256. (A exigência de criptografia se mantém para dados pessoais, mesmo sem dados bancários).

## Casos de Uso / Histórias de Usuário

**ID:** HU01

**Como** cliente,

**Quero** fazer uma encomenda de doces e pagar na entrega/retirada,

**Para que** eu possa reservar os doces e combinar o pagamento diretamente.

**Critérios de aceitação:**

* Deve ser possível selecionar os produtos e a quantidade.
* Deve ser possível informar a data de entrega ou retirada e escolher o meio de pagamento offline.
* O sistema deve exibir a confirmação do pedido e o valor total a ser pago.

**Caso de Uso:** Fazer pedido/Encomenda.

**Ator:** Cliente

**Fluxo principal:**

* + - 1. O cliente acessa o site e navega pelo catálogo de doces.
      2. O cliente adiciona os itens desejados ao carrinho.
      3. O cliente acessa o carrinho e escolhe a opção "Finalizar Compra”.
      4. O cliente informa seus dados de contato e endereço de entrega.
      5. O cliente seleciona a data e método de entrega/retirada.
      6. O cliente seleciona a forma de pagamento (Ex: "Dinheiro na entrega", "Cartão na retirada", "Pix por contato").
      7. O cliente finaliza a encomenda.
      8. O sistema salva o pedido e envia um e-mail de confirmação com o resumo da compra e a forma de pagamento escolhida.

**Fluxo alternativo:** 3a. Se não houver horário disponível, o sistema exibe uma mensagem informando indisponibilidade.

# ARQUITETURA E DESIGN

Esta seção descreve como o sistema é estruturado internamente, suas camadas, componentes, tecnologias e como eles se relacionam. Ajuda desenvolvedores e arquitetos a compreenderem o funcionamento técnico e a manterem a consistência do projeto.

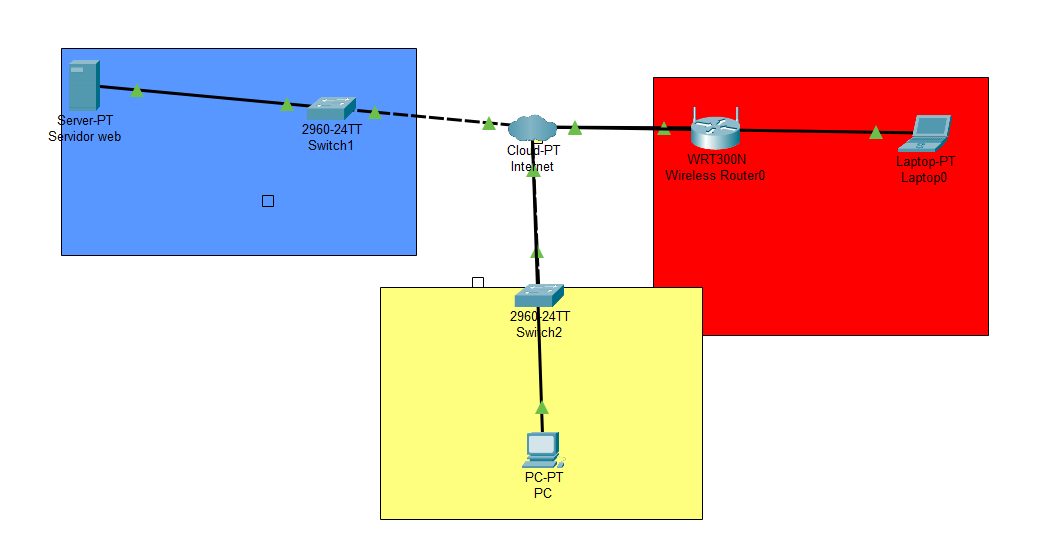
## Arquitetura de Software

**Estilo arquitetural:** MVC (Model-View-Controller)

**Descrição:** O sistema utiliza uma arquitetura MVC dividida em três camadas principais:

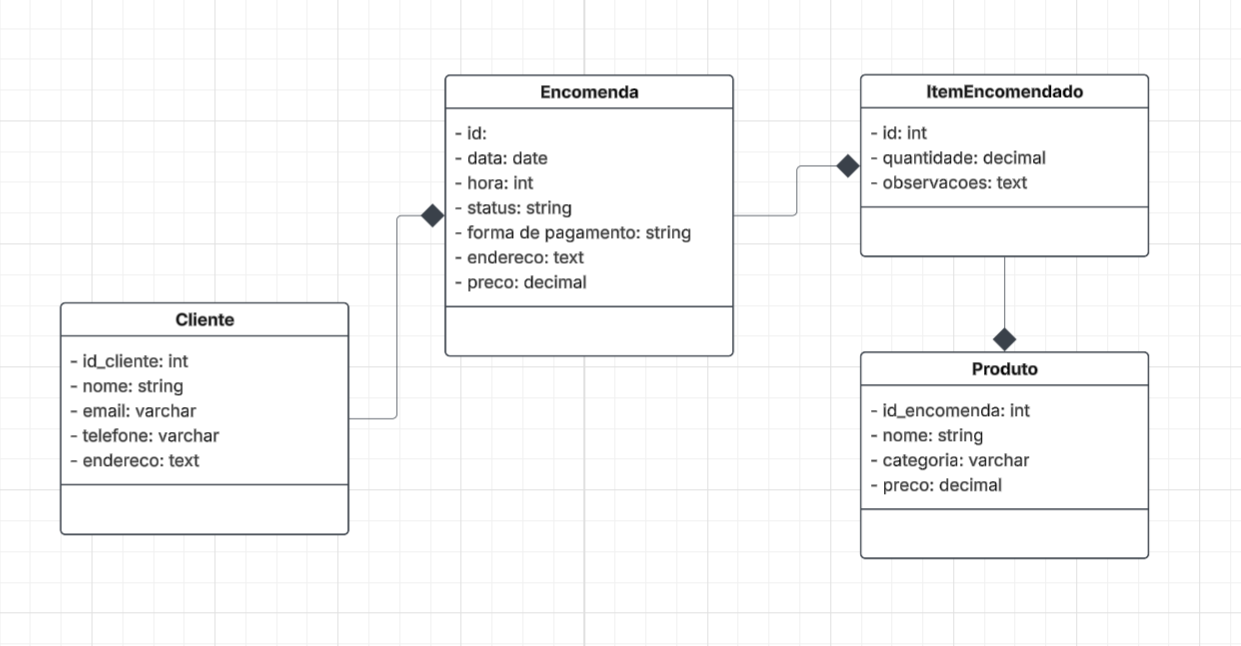
* **Model:** responsável pela lógica de negócio e acesso ao banco de dados.
* **View:** páginas web renderizadas para o usuário.
* **Controller:** gerencia as requisições, coordena os modelos e define as respostas.

Figura 3- Diagrama de Arquitetura



Fonte: Produção do autor (2025)

## Diagrama UML do sistema

Figura 4- Diagrama UML 

Fo

Fonte: Produzido pelo autor.

Descrição:

• Um **Cliente** pode realizar várias **Encomendas**. • Cada **Encomenda** pertence a exatamente um **Cliente**.

• Uma **Encomenda** pode conter vários **ItensEncomendados**.

• Cada **ItemEncomendado** está vinculado a exatamente uma **Encomenda** e a um **Produto** específico.

• Um **Produto** pode estar presente em vários **ItensEncomendados**, ou seja, pode ser encomendado por diferentes clientes em diferentes pedidos.

• Cada **Encomenda** possui os atributos: data, hora, status (como "pendente", "confirmada", "cancelada"), forma de pagamento, endereço de entrega e preço total. • Cada **ItemEncomendado** possui os atributos: quantidade e observações (como instruções especiais ou modificações).

• Cada **Produto** possui os atributos: nome, categoria e preço unitário.

## Modelagem de dados

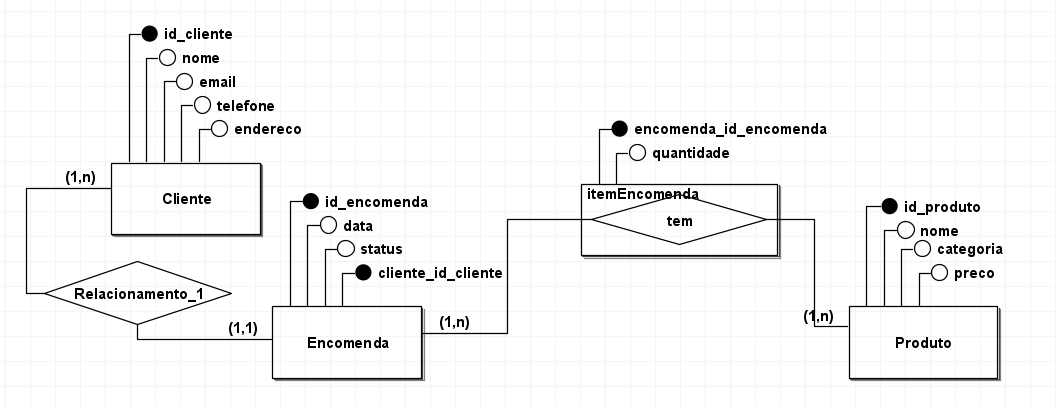
A modelagem de dados foi essencial, neste projeto, para representar de forma estruturada os elementos centrais do processo de encomendas de doces, como clientes, doces (produtos) e pedidos/encomendas.

Esse processo envolveu a criação de modelos conceitual, lógico e físico, permitindo a construção de um banco de dados relacional (no MySQL), com integridade, consistência e clareza nas relações entre as entidades.

A modelagem garantiu que o sistema armazenasse corretamente os dados (por exemplo, detalhes do produto, informações de contato do cliente, e a composição de cada pedido) e facilitasse consultas rápidas e seguras sobre o catálogo, estoque e histórico de vendas em ambiente local/nuvem.

### Modelagem Conceitual

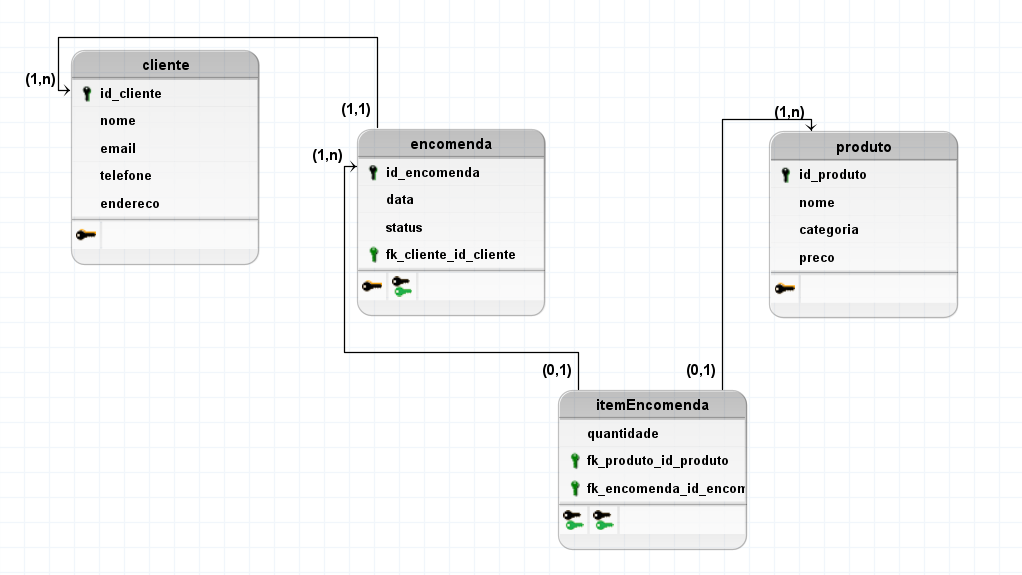
Figura 5- Modelagem Conceitual



Fonte: Produzido pelo autor.

### Modelagem Lógica

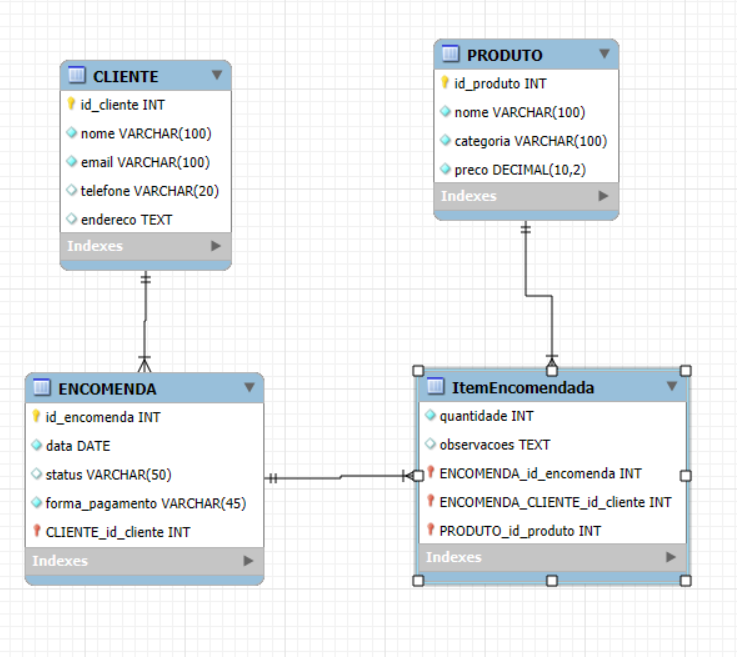
Figura 6- modelagem Lógica



Fonte: Produzido pelo autor.

### Modelagem Física

Figura 7- Modelagem Física



Fonte: Produzido pelo autor.

### Dicionário de dados

Tabela

O conteúdo gerado por IA pode estar incorreto.

(tem que alterar essa tabela)

(**alterar todas as informações que estão abaixo**)

# QUALIDADE E TESTES

Essa seção documenta as estratégias de verificação e validação usadas para garantir que o software funcione corretamente, atenda aos requisitos e tenha qualidade técnica e funcional.

## Plano de Testes

Plano de Testes – Sistema de Agendamento Médico (SAMEO)

* Objetivo: Garantir que as funcionalidades principais operem corretamente e com segurança.
* Escopo: Cadastro, login, agendamento, cancelamento, notificações.
* Responsáveis: Equipe de QA e desenvolvedores
* Critérios de entrada: Funcionalidade implementada e integrada
* Critérios de saída: 100% dos testes automatizados e manuais aprovados sem erros críticos

## Casos de Teste

Caso de Teste CT-002 – Agendar Consulta

* Objetivo: Verificar se o paciente consegue agendar uma consulta com um médico disponível.
* Pré-condição: O paciente está autenticado e o médico possui horários disponíveis.
* Passos:
  1. Acessar o menu "Agendar Consulta"
  2. Selecionar especialidade e médico
  3. Escolher data e horário
  4. Confirmar agendamento

**Resultado Esperado:** O sistema exibe mensagem de sucesso e envia e-mail de confirmação.

## Planejamento de testes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID Teste** | **Tipo de teste** | **Resultado esperado** | **Duração** | **Data de início** | **Data de término** |
| CT-001 | Teste de cadastro | Cadastrar um novo usuario no sistema. | 1h | 14/10/2024 | 14/10/2024 |
| CT-002 | Teste de login | Conectar usuario a plataforma | 1h | 16/10/2024 | 16/10/2024 |
| CT-003 | Agenda | Agendar uma consulta | 2h | 04/11/2024 | 04/11/2024 |

## Resultados dos Testes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID Teste** | **Funcionalidade** | **Status** | **Observações** | **Duração** | **Data de início** | **Data de término** |
| CT-001 | Cadastro de usuário | Aprovado | Sem falhas | 1h | 14/10/2024 | 14/10/2024 |
| CT-002 | Login | Reprovado | Sistema não enviou e-mail de confirmação | 1h | 16/10/2024 | 16/10/2024 |
| CT-003 | Cancelamento | Aprovado | Funcionando conforme esperado | 2h | 04/11/2024 | 04/11/2024 |

## Testes Automatizados

Tecnologia utilizada: Jest (para testes unitários em JavaScript)

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

**Cobertura de Testes:**

* 90% de cobertura de código em módulos críticos
* Executado a cada push no repositório (via GitHub Actions)

# MANUAL DO USUÁRIO

Esta seção orienta o usuário final sobre como navegar e utilizar o sistema, explicando o funcionamento das principais telas, ações disponíveis e o que fazer em situações comuns. Idealmente, deve ser simples, objetiva e acessível até para pessoas não técnicas.

## Guia de Navegação

**Página Inicial (Home):** Apresenta um resumo do sistema e opções para login ou cadastro.

Figura 8- Página inicial do sistema.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Fonte: Elaborada pelo autor com uso da IA DALL·E (OpenAI, 2025).

**Menu Principal:**

* **Agendar Consulta** – Permite escolher especialidade, médico, data e hora.
* **Minhas Consultas** – Lista as consultas futuras e passadas.
* **Perfil** – Permite editar dados pessoais e senha.
* **Sair** – Encerra a sessão do usuário.

Figura 9- Página do perfil de usuário, apresentada após o login

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Fonte: Produção do autor (2025).

**Fluxo Básico para o Paciente:**

* 1. Acessar o sistema pelo navegador: [http://localhost:3000](http://localhost:3000/)
  2. Fazer login com e-mail e senha
  3. Escolher “Agendar Consulta”
  4. Selecionar a especialidade e médico
  5. Escolher data e horário disponíveis
  6. Confirmar
  7. Visualizar confirmação e receber e-mail

## FAQs e Problemas Comuns

|  |  |
| --- | --- |
| **Pergunta** | **Resposta** |
| Esqueci minha senha. E agora? | Clique em “Esqueci minha senha” na tela de login e siga as instruções. |
| Não recebi e-mail de confirmação. | Verifique a caixa de spam ou tente reagendar a consulta. |
| O botão “Agendar” não funciona. | Verifique se você escolheu uma data e horário válidos. |
| Posso cancelar uma consulta? | Sim, até 2 horas antes da consulta, na aba “Minhas Consultas”. |
| Funciona no celular? | Sim, o sistema é responsivo e funciona em navegadores móveis modernos. |

# MANUAL TÉCNICO

Esta seção orienta desenvolvedores e administradores sobre a estrutura do código-fonte, execução local, boas práticas de contribuição, e documentação das APIs disponíveis. É essencial para manutenção, expansão e integração do sistema.

## Guia do Desenvolvedor

**Estrutura de diretórios:**

/sameo  
├── /frontend → Aplicação web (React/HTML/JS/CSS)  
├── /backend → API em Java (Spring Boot)  
├── /database → Scripts SQL (schema e dados iniciais)  
└── README.md → Instruções básicas do projeto

### Guia de Instalação

**Pré-requisitos:**

* Java JDK 17+
* Node.js 18+ e npm
* MySQL 8+
* Git instalado
* IDE recomendadas: Spring (backend), VS Code (frontend)

**Passos de instalação:**

1. Clonar o repositório:

Logotipo

O conteúdo gerado por IA pode estar incorreto.

1. Configurar o banco de dados (MySQL):

* Criar banco: CREATE DATABASE sameo;
* Usuário/senha padrão: root/root
* Importar script: /database/schema.sql

1. Executar o backend (API - Spring Boot):

* Navegar até /backend
* Configurar application.properties:

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

* Rodar o projeto com:

Forma

O conteúdo gerado por IA pode estar incorreto.

1. Executar o frontend (JS/HTML/CSS):

* Navegar até /frontend
* Instalar dependências:

Uma imagem contendo Forma

O conteúdo gerado por IA pode estar incorreto.

* Iniciar servidor local:

Logotipo

O conteúdo gerado por IA pode estar incorreto.

* Acesse: [http://localhost:3000](http://localhost:3000/)

## Documentação da API

|  |  |  |  |
| --- | --- | --- | --- |
| **Método** | **Endpoint** | **Descrição** | **Autenticação** |
| GET | /api/pacientes | Lista todos os pacientes | Sim |
| POST | /api/pacientes | Cadastra um novo paciente | Não |
| POST | /api/login | Realiza autenticação do usuário | Não |
| GET | /api/consultas | Lista consultas do paciente logado | Sim |
| POST | /api/consultas | Agenda uma nova consulta | Sim |
| DELETE | /api/consultas/{id} | Cancela uma consulta | Sim |

### Formato de exemplo de request



### Resposta esperada



## Padrões de Código

### Backend – Java (Spring Boot):

* Nome de classes em PascalCase: PacienteController,
* Nome de variáveis em camelCase: dataConsulta, nomeCompleto
* Organização por camadas:

/controller

/service

/repository

/entity

### Frontend – JavaScript:

* Componentes React em PascalCase: AgendamentoForm.jsx
* Separação de responsabilidades (componentes, páginas, estilos)
* Uso de ESLint + Prettier para padronização automática

### Boas práticas gerais:

* Código comentado sempre que necessário
* Funções curtas, com nomes descritivos
* Commits claros (ex: fix: validação de e-mail no cadastro)

# PROPOSTA COMERCIAL

## Descrição da Proposta

A presente proposta tem como objetivo oferecer uma solução tecnológica completa para a informatização do processo de agendamento médico da contratante, por meio do desenvolvimento do **Sistema de Agendamento Médico Online (SAMEO)**. Trata-se de um sistema personalizado que será desenvolvido sob medida, com foco na usabilidade, segurança e eficiência operacional da instituição de saúde.

O SAMEO será composto por três camadas principais:

* **Frontend (interface do usuário):** Desenvolvido em JavaScript, HTML5 e CSS3, com foco em navegabilidade simples e intuitiva, acessível via navegador em ambiente local.
* **Backend/API:** Desenvolvido utilizando o framework Java Spring Boot, responsável por toda a lógica de negócios, gerenciamento de sessões, validações, regras de agendamento e comunicação com o banco de dados.
* **Banco de Dados:** Utilização do MySQL, responsável pelo armazenamento seguro e estruturado de informações relacionadas aos usuários, médicos, horários, especialidades, agendamentos, entre outros.

O sistema será projetado para **funcionar em ambiente local (intranet)**, sem dependência de internet ou hospedagem em nuvem, sendo ideal para clínicas, consultórios ou centros médicos que desejam maior controle de seus dados e maior segurança interna.

## Escopo Comercial

### Itens Inclusos

* Workshops de levantamento de requisitos (até 3 sessões)
* Desenvolvimento de todas as funcionalidades descritas nos requisitos
* Testes unitários e integrados
* Documentação completa (usuário, técnico, implantação)
* Treinamento remoto de até 8 horas

### Itens Exclusos:

* Hospedagem em nuvem ou servidores externos
* Integrações futuras com sistemas de terceiros (ex: SUS)
* Manutenção após 3 meses de garantia

### Preço e Condições de Pagamento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Descrição** | **Qtd/Horas** | **Valor Unitário (R$)** | **Total (R$)** |
| 1. Mão de Obra | | | | |
| 1.1 | Analista de Requisitos (levantamento, documentação funcional) | 40h | R$ 70,00 | R$ 2.800,00 |
| 1.2 | Desenvolvedor Backend (Java Spring – APIs e lógica de negócios) | 100h | R$ 75,00 | R$ 7.500,00 |
| 1.3 | Desenvolvedor Frontend (JS, HTML, CSS – telas web responsivas) | 60h | R$ 65,00 | R$ 3.900,00 |
| 1.4 | Analista de Testes e QA | 18h | R$ 65,00 | R$ 1.170,00 |
| 1.5 | Gerente de Projeto (Scrum Master, reuniões e acompanhamento) | 18h | R$ 60,00 | R$ 1.080,00 |
|  | Subtotal Mão de Obra |  |  | R$ 16.450,00 |
| 2. Infraestrutura e Equipamentos | | | | |
| 2.1 | Notebook de desenvolvimento/teste (uso interno – rateio) | 1 | R$ 3.000,00 | R$ 3.000,00 |
| 2.2 | Tablet para interface de recepção (Samsung Tab 4) | 2 | R$ 1.200,00 | R$ 2.400,00 |
| 2.3 | Servidor local (mini-PC dedicado – offline) | 1 | R$ 2.800,00 | R$ 2.800,00 |
| 2.4 | Licença IDE (IntelliJ, WebStorm – uso da equipe) | 3 | R$ 400,00 | R$ 1.200,00 |
| 2.5 | Licença Antivírus/Firewall local | 1 | R$ 200,00 | R$ 200,00 |
| 2.6 | Internet dedicada (uso da clínica – estimativa setup) | 1 | R$ 500,00 | R$ 500,00 |
|  | Subtotal Infraestrutura |  |  | R$ 10.100,00 |
| 3. Software | | | | |
| 3.1 | Desenvolvimento da aplicação web (interface paciente e admin) | 1 | R$ 6.000,00 | R$ 6.000,00 |
| 3.2 | Desenvolvimento da API REST em Java Spring | 1 | R$ 3.000,00 | R$  3.000,00 |
| 3.3 | Banco de dados MySQL (modelagem, queries e integração) | 1 | Incluso no item 3.2 | - |
| 3.4 | Configuração e instalação local do sistema | 1 | Incluso no item 3.2 | - |
|  | Subtotal Software |  |  | R$ 9.000,00 |
| 4. Treinamento e Capacitação | | | | |
| 4.1 | Treinamento com a recepção (uso do sistema e dashboard) | 6h | R$ 250,00 | R$ 1.500,00 |
| 4.2 | Treinamento com o médico/gestor (cadastros, relatórios, etc.) | 6h | R$ 250,00 | R$ 1.500,00 |
|  | Subtotal Treinamento |  |  | R$ 3.000,00 |

**Total Geral do Projeto SAMEO**

* Mão de obra: R$ 38.550,00
* Infraestrutura: R$ 10.100,00
* Software: R$ 9.000,00
* Treinamento: R$ 3.000,00

**Valor Total:** R$ 60.650,00

## Cronograma de Entrega

|  |  |  |  |
| --- | --- | --- | --- |
| **Fase** | **Duração estimada** | **Data de Início** | **Data de Término** |
| Planejamento e Kick-off | 1 semana | 05/08/2025 | 12/08/2025 |
| Desenvolvimento (3 Sprints) | 9 semanas | 13/08/2025 | 14/10/2025 |
| Testes e Ajustes Finais | 2 semanas | 15/10/2025 | 29/10/2025 |
| Treinamento e Entrega | 1 semana | 30/10/2025 | 06/11/2025 |

## Termos e Condições

* O cliente terá garantia de 3 meses para correções de bugs sem custo adicional.
* Qualquer customização além do escopo será cobrada à parte, em contrato suplementar.
* Todo o código-fonte desenvolvido será de propriedade do cliente, mediante quitação integral.
* As partes assinam acordo de confidencialidade (NDA) para proteger informações sensíveis.

## Validade da Proposta

Esta proposta é válida por 30 dias a partir da data de envio (até 30 de agosto de 2025). Após esse prazo, valores e prazos poderão ser revistos.

# CONSIDERAÇÕES FINAIS

O desenvolvimento do Sistema de Agendamento Médico Online (SAMEO) permitiu a aplicação prática dos conhecimentos adquiridos ao longo do curso técnico em desenvolvimento de sistemas, integrando conceitos de análise de requisitos, modelagem, programação full stack e gestão de projetos com metodologias ágeis.

A proposta surgiu da identificação de uma necessidade real: modernizar e automatizar o processo de agendamento de consultas em instituições de saúde que ainda utilizam métodos manuais ou ineficientes. A partir desse problema, foram definidas soluções tecnológicas viáveis, seguras e de fácil utilização, contemplando desde a elaboração da documentação técnica até a entrega de um sistema funcional em ambiente local.

O SAMEO foi construído com base em uma arquitetura multicamadas, utilizando frontend em JavaScript, HTML e CSS, API em Java Spring Boot e banco de dados MySQL, garantindo separação de responsabilidades, escalabilidade e facilidade de manutenção. Durante o desenvolvimento, também foram aplicadas boas práticas de versionamento, testes, documentação e segurança, reforçando o compromisso com a qualidade do software.

Além dos ganhos técnicos e operacionais para o cliente simulado, o projeto proporcionou aos desenvolvedores uma experiência próxima da realidade profissional, envolvendo contato com clientes, trabalho em equipe, prazos, documentação e entrega final.

Assim, conclui-se que os objetivos propostos foram plenamente alcançados, com a entrega de um sistema funcional, documentado e testado. O SAMEO representa uma contribuição significativa para a organização simulada e reforça a importância da tecnologia no aprimoramento dos serviços de saúde.

# REFERÊNCIAS

Livro impresso (obra técnica):

ALMEIDA, Maria C. de. Engenharia de Software Moderna. 2. ed. São Paulo: Atlas, 2020.

Livro impresso (obra técnica/didática):

BEZERRA, Eduardo P. Princípios de Análise e Projeto de Sistemas com UML. 3. ed. Rio de Janeiro: LTC, 2015.

Site institucional / Documentação técnica online:

GITHUB. Documentação oficial do Spring Boot. Disponível em: https://spring.io/projects/spring-boot. Acesso em: 01 ago. 2025.

Site institucional / Página web com conteúdo informativo de ferramenta digital:

OPENAI. Guia de uso da ferramenta DALL·E para criação de imagens com IA. 2025. Disponível em: https://openai.com/dall-e. Acesso em: 01 ago. 2025.

Documento governamental / Cartilha oficial (documento online):

BRASIL. Ministério da Saúde. Cartilha de Boas Práticas para Atendimento em Clínicas Médicas. Brasília, DF: MS, 2018. Disponível em: https://www.gov.br/saude. Acesso em: 01 ago. 2025.

Livro impresso (obra de referência internacional):

PRESSMAN, Roger S. Engenharia de Software. 8. ed. Porto Alegre: McGraw-Hill, 2016.

Norma técnica (NBR):

NBR ISO/IEC 25010:2011. Sistemas e software de engenharia – Requisitos e avaliação da qualidade de produto de software – Modelo de qualidade de produto. Rio de Janeiro: ABNT, 2011.

Documento interno / Projeto acadêmico (não publicado):

SOUZA, Laura B. M. de et al. Sistema de Agendamento Médico Online (SAMEO). Projeto integrador – Curso Técnico em Desenvolvimento de Sistemas. Sorocaba: SENAI Gaspar Ricardo Junior, 2024. Documento interno.