

ENGENHARIA DE SOFTWARE

**Fernando Melo Fonseca
Levi Pires Soares Dos Santos
Lucas Freire Alves**

ENGENHARIA DE SOFTWARE

**Salvador
2024**

Levi Pires Soares Dos Santos
Fernando Melo Fonseca
Lucas Freire Alves

ENGENHARIA DE SOFTWARE

Trabalho apresentado para a Disciplina
Engenharia de Software, pelo Curso de
Engenharia da Computação do Centro
Universitário Uniruy, ministrada pelo Prof.
Heleno Cardoso.

Salvador
2024

SUMÁRIO

- 1. Introdução**
- 2. Objetivos da Aplicação**
- 3. Requisitos Funcionais (RF)**
- 4. Requisitos Não Funcionais (RNF)**
- 5. Casos de Uso**
- 6. Diagrama de Casos de Uso**
- 7. DFD Nível Zero**
- 8. DFD Por Evento**
- 9. DER**
- 10. Estrutura do Dicionário de Dados (DDL)**
- 11. Modelo Comportamental**
- 12. Layout das Interfaces**
- 13. Conclusão**

1. Introdução

A aplicação de agendamento visa facilitar a gestão de compromissos e eventos, permitindo que usuários possam criar, editar, visualizar e deletar agendamentos em uma interface intuitiva. O sistema busca otimizar o tempo do usuário, evitando conflitos de horários.

Problema que resolve: Muitas pessoas e empresas têm dificuldades em organizar compromissos e eventos, o que pode levar a perda de prazos e falta de organização. A aplicação serve para simplificar esse processo, permitindo uma gestão eficiente de calendários.

Benefícios:

- Gestão centralizada de compromissos.
 - Redução de conflitos de horários.
 - Facilidade de visualização de agenda semanal ou mensal.
-

2. Objetivos da Aplicação

- Prover uma plataforma de fácil uso para agendamentos.
 - Evitar conflitos de horários, garantindo que dois compromissos não sejam marcados no mesmo horário.
 - Permitir a visualização clara da agenda em diferentes formatos (diária, semanal, mensal).
 - Suportar múltiplos usuários com agendas independentes, para permitir uso em contextos individuais ou empresariais.
-

3. Requisitos Funcionais (RF)

Aqui listamos o que o sistema deve fazer:

1. RF01 - Cadastro de Usuário: O sistema deve permitir o cadastro de novos usuários com dados como nome e senha.
 2. RF02 - Autenticação de Usuário: O sistema deve fornecer uma tela de login para autenticação segura dos usuários.
 3. RF03 - Criar Agendamento: O sistema deve permitir que o usuário crie agendamentos com data, hora, descrição.
 4. RF04 - Editar Agendamento: O sistema deve permitir que o usuário edite agendamentos existentes.
 5. RF05 - Deletar Agendamento: O sistema deve permitir que o usuário apague agendamentos previamente criados.
 6. RF06 - Visualizar Agendamentos: O sistema deve permitir que o usuário visualize sua agenda de compromissos em uma interface gráfica com opções de visualização diária, semanal e mensal.
 7. RF07 - Gerenciamento de Conflitos: O sistema deve alertar o usuário se houver conflito de horários ao tentar agendar compromissos no mesmo horário.
-

4. Requisitos Não Funcionais (RNF)

Aqui estão os requisitos relacionados à qualidade, desempenho e outras características:

1. RNF01 - Usabilidade: A aplicação deve ser intuitiva e fácil de usar, com uma interface clara e objetiva.
2. RNF02 - Desempenho: O sistema deve carregar a lista de compromissos em menos de 2 segundos para uma melhor experiência do usuário.
3. RNF03 - Escalabilidade: O sistema deve ser capaz de suportar múltiplos usuários simultâneos sem degradação do desempenho.
4. RNF04 - Segurança: As informações dos usuários e agendamentos devem ser protegidas por criptografia durante o transporte e armazenamento.
5. RNF05 - Compatibilidade: A aplicação deve funcionar em dispositivos móveis, tablets e desktops.

5. Casos de Uso

Os Casos de Uso são descrições de interações entre o usuário e o sistema para realizar uma funcionalidade específica. Aqui, vamos definir os principais casos de uso com uma breve descrição e os passos envolvidos.

Caso de Uso 01: Cadastro de Usuário

- Descrição: O usuário se cadastra no sistema criando uma conta.
- Atores: Usuário
- Pré-condição: O usuário ainda não possui uma conta.
- Fluxo Principal:
 1. O usuário acessa a página de cadastro.
 2. O sistema solicita dados (nome e senha).
 3. O usuário insere as informações e confirma o cadastro.
 4. O sistema valida os dados e cria a conta.
- Pós-condição: O usuário tem sua conta criada e pode acessar o sistema.

Caso de Uso 02: Login do Usuário

- Descrição: O usuário faz login no sistema.
- Atores: Usuário
- Pré-condição: O usuário já tem uma conta cadastrada.
- Fluxo Principal:
 1. O usuário acessa a página de login.
 2. O sistema solicita usuário e senha.
 3. O usuário insere os dados e clica em "Login".
 4. O sistema autentica o usuário e redireciona para a tela inicial.
- Pós-condição: O usuário tem acesso às funcionalidades do sistema.

Caso de Uso 03: Criar Agendamento

- Descrição: O usuário cria um novo agendamento no sistema.
- Atores: Usuário
- Pré-condição: O usuário está autenticado.
- Fluxo Principal:
 1. O usuário acessa a tela de criação de agendamento.
 2. O sistema solicita data, hora e descrição.
 3. O usuário preenche as informações e confirma.
 4. O sistema valida os dados e salva o agendamento.
 5. O usuário recebe uma confirmação visual de que o agendamento foi criado.
- Pós-condição: O agendamento é salvo no sistema.

Caso de Uso 04: Editar Agendamento

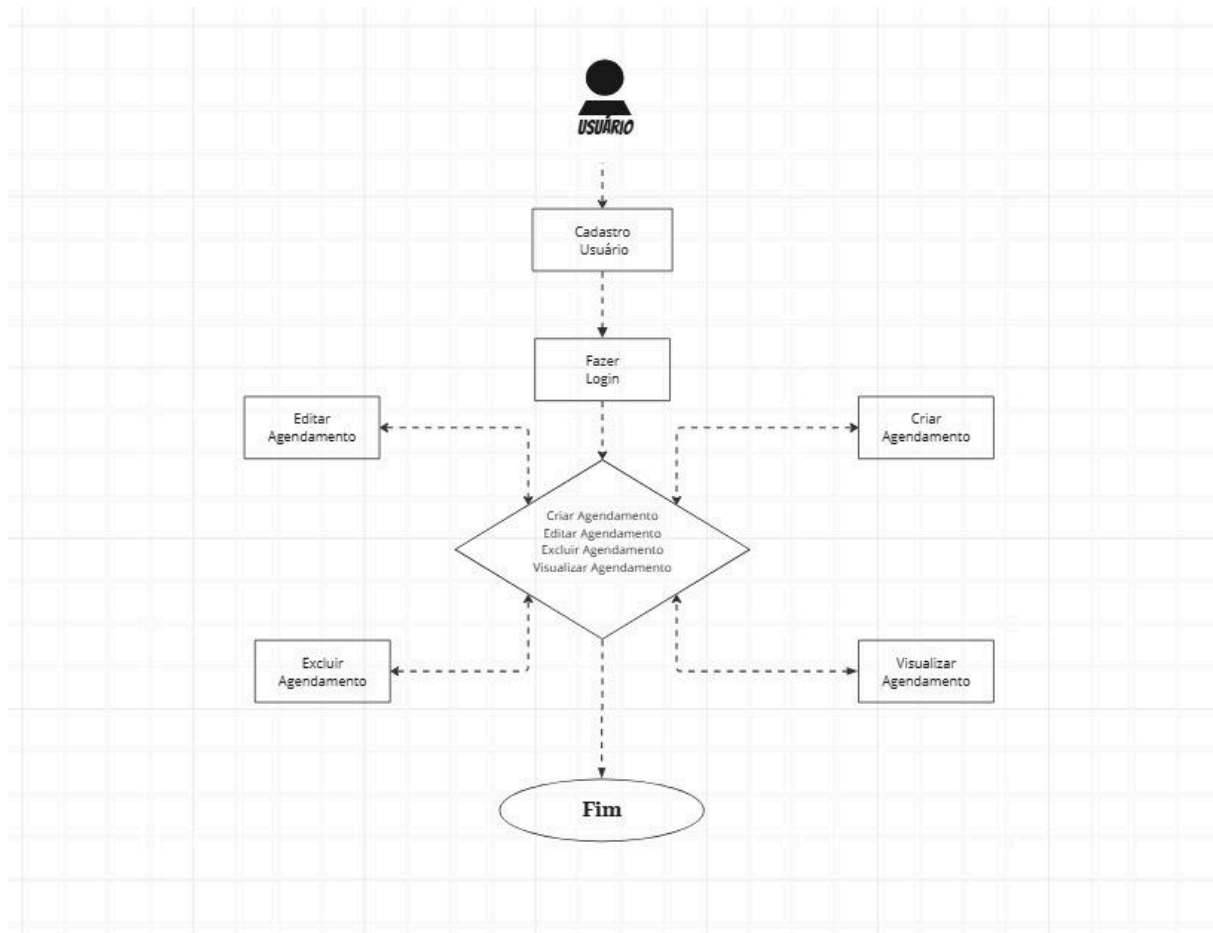
- Descrição: O usuário edita um agendamento existente.

- Atores: Usuário
- Pré-condição: O usuário está autenticado e possui agendamentos criados.
- Fluxo Principal:
 1. O usuário acessa a lista de agendamentos.
 2. O sistema exibe os agendamentos do usuário.
 3. O usuário seleciona o agendamento que deseja editar.
 4. O sistema permite alterar data, hora e descrição.
 5. O usuário confirma as alterações.
 6. O sistema salva as alterações.
- Pós-condição: O agendamento é atualizado no sistema.

Caso de Uso 05: Excluir Agendamento

- Descrição: O usuário remove um agendamento existente.
- Atores: Usuário
- Pré-condição: O usuário está autenticado e possui agendamentos criados.
- Fluxo Principal:
 1. O usuário acessa a lista de agendamentos.
 2. O sistema exibe os agendamentos do usuário.
 3. O usuário seleciona o agendamento que deseja excluir.
 4. O sistema solicita uma confirmação de exclusão.
 5. O usuário confirma a exclusão.
 6. O sistema remove o agendamento.
- Pós-condição: O agendamento é excluído do sistema.

6. Diagrama de Casos de Uso



7. DFD Nível Zero

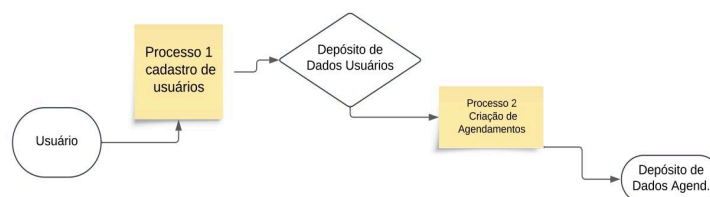
O Diagrama de Fluxo de Dados (DFD) Nível Zero mostra os principais processos e o fluxo de dados entre eles. Ele oferece uma visão geral do sistema, identificando as principais interações de dados entre os componentes.

Descrição do DFD Nível Zero:

- Processos Principais:
 1. Gerenciamento de Usuários: Cadastro, login e autenticação.
 2. Gerenciamento de Agendamentos: Criação, edição, exclusão e visualização de agendamentos.
- Fluxo de Dados:
 1. Usuário → Sistema: Envio de dados para cadastro e login.
 2. Sistema → Banco de Dados: Salvamento e recuperação de dados de usuários e agendamentos.
 3. Usuário → Sistema: Criação, edição e exclusão de agendamentos.

Elementos do DFD:

- Entidades Externas:
 - Usuário
- Processos Internos:
 - Gerenciamento de Usuários
 - Gerenciamento de Agendamentos
- Armazenamentos de Dados:
 - Banco de Dados de Usuários
 - Banco de Dados de Agendamentos



8. DFD POR EVENTO

1. Entidade Externa: Usuário

- O **Usuário** é a entidade que interage com o sistema. Ele pode solicitar várias operações relacionadas aos agendamentos.

2. Processos Detalhados: Gerenciamento de Agendamentos

- O **Gerenciamento de Agendamentos** é decomposto em quatro subprocessos principais:
 - **Criar Agendamento:** O usuário fornece dados como data, hora e descrição de um evento.
 - **Editar Agendamento:** O usuário pode alterar os dados de um agendamento já existente.
 - **Excluir Agendamento:** O usuário solicita a remoção de um agendamento.
 - **Visualizar Agendamentos:** O usuário solicita a visualização de todos os seus agendamentos.

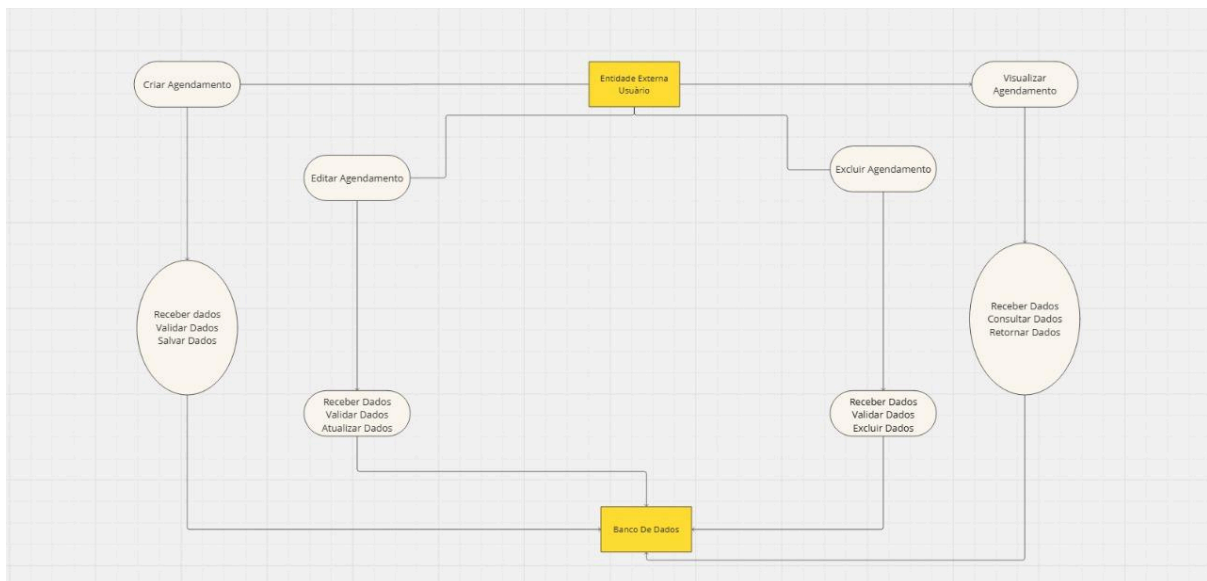
3. Depósito de Dados: Banco de Dados de Agendamentos

- Todos os subprocessos interagem com o **Banco de Dados de Agendamentos**, onde as informações sobre cada compromisso são armazenadas e manipuladas.
 - Quando o usuário cria um agendamento, os dados são enviados e armazenados no banco.
 - Quando o usuário edita ou exclui um agendamento, o banco de dados é atualizado.
 - Ao visualizar, o sistema recupera as informações dos agendamentos do banco de dados e as envia ao usuário.

Fluxos de Dados:

- **Fluxo 1: Criar Agendamento:**
 - O **Usuário** envia os detalhes do agendamento.
 - O sistema valida os dados e armazena o novo agendamento no **Banco de Dados de Agendamentos**.
 - O sistema retorna uma confirmação para o **Usuário**.
- **Fluxo 2: Editar Agendamento:**
 - O **Usuário** seleciona um agendamento existente e envia as novas informações.
 - O sistema atualiza os dados no **Banco de Dados de Agendamentos**.
 - O sistema retorna uma confirmação da edição para o **Usuário**.
- **Fluxo 3: Excluir Agendamento:**
 - O **Usuário** solicita a exclusão de um agendamento.

- O sistema remove o agendamento do **Banco de Dados de Agendamentos**.
- O sistema confirma a exclusão para o **Usuário**.
- **Fluxo 4: Visualizar Agendamentos:**
 - O **Usuário** solicita a visualização de todos os seus agendamentos.
 - O sistema recupera os dados do **Banco de Dados de Agendamentos**.
 - O sistema apresenta a lista de agendamentos ao **Usuário**.



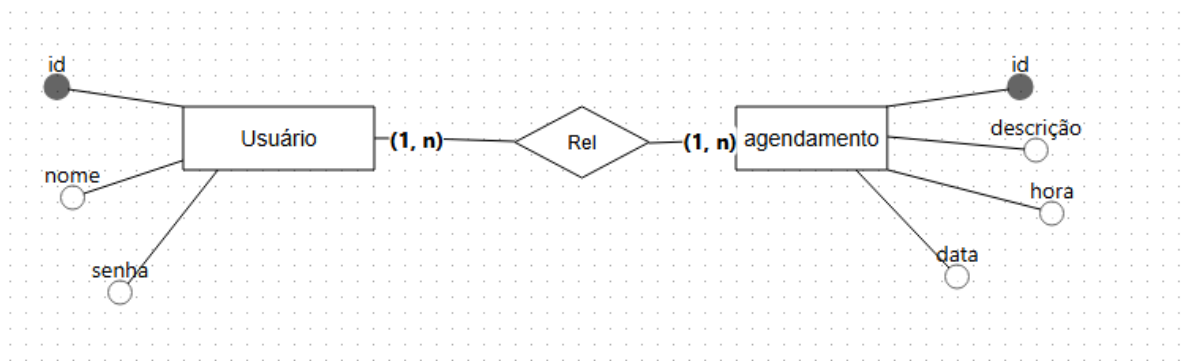
9. DER

Entidades:

- Usuário: Representa os usuários que utilizam o sistema.
 - Atributos: **id_usuario** (chave primária), **nome**, **senha**.
- Agendamento: Representa os compromissos criados pelos usuários.
 - Atributos: **id_agendamento** (chave primária), **data**, **hora**, **descricao**, **id_usuario** (chave estrangeira para associar ao usuário).

Relacionamentos:

- Usuário e Agendamento: Relacionamento 1 (um usuário pode ter vários agendamentos).



10. Estrutura do Dicionário de Dados (DDL)

Tabela: Usuário

- **id_usuario**: INT, PK, NOT NULL — Identificador único do usuário.
- **nome**: VARCHAR(100), NOT NULL — Nome do usuário.
- **senha**: VARCHAR(100), NOT NULL — Senha de acesso do usuário.

Tabela: Agendamento

- **id_agendamento**: INT, PK, NOT NULL — Identificador único do agendamento.
- **data**: DATE, NOT NULL — Data do agendamento.
- **hora**: TIME, NOT NULL — Hora do agendamento.
- **descricao**: TEXT, NULL — Descrição opcional do agendamento.
- **id_usuario**: INT, FK, NOT NULL — Referência ao **id_usuario** da tabela Usuário.

Relacionamento

- **Chave Estrangeira**: **id_usuario** na tabela **Agendamento** faz referência a **id_usuario** na tabela **Usuário** para representar a relação de 1 entre **Usuário** e **Agendamento**.

Passo a Passo no DDL

1. **Definição das Tabelas** com as colunas e tipos de dados.
2. **Definição das Chaves Primárias (PK)** para identificar cada registro de forma única.
3. **Definição das Chaves Estrangeiras (FK)** para estabelecer os relacionamentos.
4. **Restrições de Unicidade e Not Null** onde aplicável.

11. Modelo Comportamental

1. Criar Agendamento

- **Ator:** Usuário
- **Objetivo:** Permitir que o usuário crie um novo agendamento.
- **Fluxo de ações:**
 1. O usuário acessa a opção “Novo Agendamento” na interface.
 2. A interface exibe o formulário de criação.
 3. O usuário preenche os dados (data, hora, descrição) e confirma.
 4. A interface envia os dados ao controlador.
 5. O controlador valida as informações e envia uma solicitação de criação para o banco de dados.
 6. O banco de dados salva o agendamento e retorna uma confirmação de sucesso.
 7. A interface exibe uma mensagem confirmando a criação do agendamento.

2. Editar Agendamento

- **Ator:** Usuário
- **Objetivo:** Permitir ao usuário modificar um agendamento existente.
- **Fluxo de ações:**
 1. O usuário seleciona um agendamento na lista para editar.
 2. A interface exibe o formulário de edição com os dados atuais do agendamento.
 3. O usuário altera os campos desejados e confirma a alteração.
 4. A interface envia os dados atualizados ao controlador.
 5. O controlador valida as novas informações e faz uma atualização no banco de dados.
 6. O banco de dados confirma a atualização e envia a resposta ao controlador.
 7. A interface exibe uma mensagem confirmando a alteração.

3. Excluir Agendamento

- **Ator:** Usuário
- **Objetivo:** Permitir que o usuário remova um agendamento.
- **Fluxo de ações:**
 1. O usuário seleciona um agendamento na lista para excluir.
 2. A interface exibe uma mensagem de confirmação para evitar exclusão acidental.
 3. O usuário confirma a exclusão.
 4. A interface envia a solicitação de exclusão ao controlador.
 5. O controlador encaminha a solicitação para o banco de dados.

6. O banco de dados remove o agendamento e retorna uma confirmação de exclusão.
7. A interface atualiza a lista de agendamentos e exibe uma mensagem de sucesso.

4. Visualizar Agendamentos

- **Ator:** Usuário
- **Objetivo:** Permitir ao usuário ver a lista de agendamentos ou detalhes de um agendamento específico.
- **Fluxo de ações:**
 1. O usuário acessa a tela de listagem de agendamentos.
 2. A interface solicita ao controlador a lista de agendamentos do usuário.
 3. O controlador consulta o banco de dados e retorna os agendamentos.
 4. A interface exibe a lista de agendamentos para o usuário.
 5. O usuário pode clicar em um agendamento específico para visualizar os detalhes.
 6. A interface exibe as informações detalhadas do agendamento selecionado.

12. Layout das Interfaces

Tela inicial

Agendamentos

[Ver Agendamentos](#)

[Criar Agendamento](#)

© 2024 Agenda de Agendamentos

Lista de Agendamentos

Lista de Agendamentos

Descrição: aaaaaa, Data: 2024-11-22, Hora: 15:03	Editar	Excluir
Descrição: 1212, Data: 2024-12-12, Hora: 12:12	Editar	Excluir
Descrição: 1313, Data: 2024-12-13, Hora: 13:13	Editar	Excluir
Descrição: 1414, Data: 2024-12-14, Hora: 14:14	Editar	Excluir
Descrição: 1616, Data: 2024-12-16, Hora: 16:16	Editar	Excluir
Descrição: 1717, Data: 2024-12-13, Hora: 17:17	Editar	Excluir
Descrição: 1818, Data: 2025-12-18, Hora: 18:18	Editar	Excluir
Descrição: 17717, Data: 2025-12-17, Hora: 17:05	Editar	Excluir
Descrição: 1212, Data: 1212-12-12, Hora: 12:12	Editar	Excluir
Descrição: 1313, Data: 1313-12-12, Hora: 13:13	Editar	Excluir
Descrição: 1313, Data: 2020-12-12, Hora: 13:13	Editar	Excluir
Descrição: 1441, Data: 1414-12-14, Hora: 14:14	Editar	Excluir
Descrição: 3131, Data: 3131-12-12, Hora: 03:31	Editar	Excluir

- Criar Novo Agendamento
- Voltar para a Home

Tela de Novo Agendamento

Criar Agendamento

Data:

dd / mm / aaaa

Hora:

-- : --

Descrição:

Salvar Agendamento

[Página Inicial](#) [Listar Agendamentos](#)

Tela de Confirmação de Exclusão do Agendamento

levi13.github.io diz

Tem certeza que deseja excluir este agendamento?

OK

Cancelar

13. Conclusão

A construção deste protótipo de aplicação de agendamento proporcionou uma visão prática do processo inicial de desenvolvimento de software, passando pelas etapas de levantamento de requisitos, modelagem de dados, criação de interfaces e implementação de funcionalidades básicas. Utilizando HTML, CSS e JavaScript, conseguimos ilustrar o começo da construção de um aplicativo, com uma interface amigável e uma estrutura funcional que permite o gerenciamento simples de agendamentos. A experiência serviu como uma introdução ao desenvolvimento de aplicações, fornecendo uma base inicial para futuras expansões e melhorias no projeto.

Além disso, a documentação de cada artefato, incluindo diagramas de casos de uso, DFDs, DER, e especificações de layout, permitiu uma melhor compreensão dos processos de negócio e das relações entre os componentes do sistema. A experiência de criar um protótipo funcional e bem documentado reforçou o entendimento das práticas recomendadas em engenharia de software, destacando a importância de uma abordagem estruturada no desenvolvimento de sistemas.

Como resultado, o protótipo está pronto para evoluir conforme as necessidades do projeto e serve como uma base sólida para futuras expansões, como a implementação de novos recursos ou integração com outras aplicações. Em suma, o projeto cumpriu seus objetivos e representou um passo importante no desenvolvimento de habilidades para projetos reais e escaláveis.