**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-graduação *Lato Sensu* em Engenharia de *Software***

Projeto Integrado

Relatório Técnico

GoBarber

André Wiliam da Silva

Belo Horizonte

2023.

# Projeto Integrado

**Sumário**

Projeto Integrado 3

1. Cronograma de Trabalho 4

2. Introdução 5

3. Definição Conceitual da Solução 6

3.1 Diagrama de Casos de Uso 6

3.2 Requisitos Funcionais 6

3.3 Requisitos Não-funcionais 7

4. Protótipo Navegável do Sistema 7

5. Diagrama de Classes de Domínio 8

6. Arquitetura da Solução 8

6.1 Padrão Arquitetural 8

6.2 C4 model - Diagrama de Contexto 9

7. Frameworks de Trabalho 10

8. Estrutura Base do Front End 10

9. Modelo Relacional ou Projeto de Banco de Dados NoSQL 10

10. Plano de Testes 11

11. Relatório de Execução de Testes de Software 11

12. Apropriação de Horas no Projeto 11

13. Código da Aplicação 12

14. Avaliação Retrospectiva 12

14.1 Objetivos Estimados 13

14.2 Objetivos Alcançados 13

14.3 Lições aprendidas 13

15. Referências 13

## Cronograma de Trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 08/09/2023 | 09/10/2022 | 1. Rever toda matéria referente ao Projeto Integrado, textos e vídeos. | Entendimento geral do Projeto Integrado. |
| 10/09/2023 | 11/09/2023 | 2. Relatório Técnico: Objetivos do Trabalho, Apresentação do Problema, Descrição Geral do Software. | Confecção do Relatório Técnico – Descritivos. |
| 12/09/2023 | 13/09/2023 | 3. Relatório Técnico: Identificação de atores, requisitos funcionais e não funcionais. | Confecção do Relatório Técnico – Definição conceitual. |
| 14/09/2023 | 17/09/2023 | 4. Relatório Técnico: Identificação da ferramenta e desenvolvimento do diagrama de Caso de Uso. | Confecção do Relatório Técnico – Diagrama de Caso de Uso. |
| 18/09/2023 | 23/09/2023 | 5. Relatório Técnico: Descrição casos de usos. | Confecção do Relatório Técnico – Descrição dos casos de usos. |
| 24/09/2023 | 26/09/2023 | 6. Relatório Técnico: Criação do Protótipo de Interface, navegável e seus itens relacionados. | Confecção do Relatório Técnico – Inclusão das referências do Protótipo. |
| 27/09/2023 | 03/10/2023 | 7. Relatório Técnico: Criação do Diagrama de Classes de domínio. | Confecção do Relatório Técnico – Diagrama de Classes de Domínio. |
| 04/10/2023 | 05/10/2023 | 8. Relatório Técnico: Definição do Padrão Arquitetural do Projeto e Tecnologias. | Confecção do Relatório Técnico – Descrição da Arquitetura escolhida e tecnologias utilizadas. |
| 06/10/2022 | 07/10/2023 | 9. Relatório Técnico: Criação do Diagrama de Contexto do Projeto e uma breve explicação. | Confecção do Relatório Técnico – Diagrama de Contexto (C4 Model) e explicação. |
| 08/10/2022 | 08/10/2023 | 10. Relatório Técnico: Apresentar Frameworks e Estrutura Base do Front End. | Confecção do Relatório Técnico – Descrição dos Frameworks e visualização do layout do front end. |
| 09/10/2022 | 09/10/2023 | 11. Relatório Técnico: Criação do Modelo Relacional do Banco de Dados. | Confecção do Relatório Técnico – Diagrama do Modelo Relacional. |
| 10/10/2023 | 10/10/2023 | 12. Relatório Técnico: Criação do Plano e Relatório de Execução de Testes e Apropriação de Horas. | Confecção do Relatório Técnico – Descrição dos Testes e Apropriação de Horas. |
| 11/11/2023 | 12/10/2023 | 13.Relatório Técnico: Finalização do Projeto com as Descrições Finais. | Confecção do Relatório Técnico – Descrição Finais como Retrospectiva, Objetivos Estimados e Alcançados e Lições Aprendidas. |

## Introdução

***2.1. Objetivos do trabalho***

O propósito primordial deste projeto consiste na concepção de uma plataforma destinada à elaboração de uma API que viabilize a conexão entre um Prestador de Serviços, especificamente um Barbeiro, e um Usuário, caracterizado como Cliente. Adicionalmente, será projetado um sítio eletrônico que consumirá a mencionada API, simplificando, dessa maneira, tanto o fornecimento de serviços quanto a busca por um profissional qualificado.

Inicialmente, a plataforma será desenvolvida com foco exclusivo na esfera virtual, providenciando uma única interface acessível tanto aos barbeiros quanto aos clientes. Esta interface disponibilizará a totalidade dos recursos necessários para ambas as partes.

***2.2. Apresentação do problema***

Com a evolução das tecnologias e a disseminação generalizada do acesso à internet, aplicativos e softwares têm conquistado um espaço significativo na simplificação das operações diárias tanto para empresas quanto para indivíduos.

Nesse contexto, diversas soluções estão sendo desenvolvidas para tornar a vida cotidiana mais conveniente. Diante dessa realidade, identificamos a necessidade premente de criar uma solução direcionada aos Barbeiros, visando otimizar as suas rotinas profissionais. Isso se traduz na agilização do preenchimento das suas agendas, permitindo um gerenciamento mais eficaz do tempo. Da mesma forma, do ponto de vista dos clientes, que também dispõem de pouco tempo livre para desperdiçar, a disponibilidade de uma plataforma que lhes permita encontrar rapidamente um profissional capaz de atendê-los e consultar a sua agenda é de importância indiscutível.

***2.3. Descrição geral do software***

O propósito subjacente à concepção deste software consiste em disponibilizar uma plataforma que efetivamente conecte o barbeiro ao cliente, simplificando de maneira significativa todo o processo envolvido.

O barbeiro terá a capacidade de efetuar o seu registro na plataforma, confeccionando, assim, o seu perfil e disponibilizando integralmente a sua agenda.

Por sua vez, o cliente terá a oportunidade de efetuar o seu registro na plataforma, criando o seu próprio perfil. Além disso, terá a prerrogativa de efetuar consultas em relação aos profissionais que estejam cadastrados na plataforma, com o intuito de encontrar um que possa atendê-lo, tendo plena visibilidade sobre a agenda do profissional escolhido.

## Definição Conceitual da Solução

## Diagrama de Casos de Uso

|  |
| --- |
|  |

## Requisitos Funcionais

1. Um usuário deve ser capaz de cadastrar seu perfil como **barbeiro** ou **cliente**;
2. Um usuário deve ser capaz de receber e-mails de atualizações tanto como **barbeiro** ou **cliente**;
3. Um cliente deve ser capaz de pesquisar um **barbeiro**;
4. Um cliente deve ser capaz de listar todos **barbeiro**;
5. Um cliente deve ser capaz de acessar o perfil do **barbeiro**;
6. Um cliente deve ser capaz de verificar a agenda do **barbeiro**;
7. Um cliente deve ser capaz de escolher uma data na agenda do **barbeiro**;
8. Um cliente deve ser capaz de escolher um horário nada data escolhida na agenda do **barbeiro**;
9. Um cliente deve ser capaz de confirmar o agendamento no **barbeiro** escolhido;
10. Um cliente deve ser capaz de cancelar seus horários agendados;
11. Um cliente deve ser capaz de alterar sua data agendada;
12. Um cliente deve ser capaz de alterar seus horários agendados;
13. Um cliente deve ser capaz de ver seus horários agendados;
14. Um cliente deve ser capaz de ver os últimos **barbeiro** que fez o agendamento;
15. Um usuário **barbeiro** deve ser capaz de ver os últimos **cliente** agendados;
16. Um usuário **barbeiro** deve ser capaz de ver os últimos **cliente** atendidos;
17. Um usuário **barbeiro** deve ser capaz de cancelar um agendamento;
18. Um usuário **barbeiro** deve ser capaz de ver os **cliente** cancelados;
19. Um usuário **barbeiro** deve ser capaz de ver seus agendamentos por data específica;
20. Um usuário **barbeiro** deve ser capaz de ver seus agendamentos por intervalos de datas;

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| RF01 | O usuário pode definir se é **cliente** ou **barbeiro**. | M | A |
| RF02 | O cliente tem de acessar a agenda do usuário **barbeiro**. | M | M |
| RF03 | O cliente pode agendar ou cancelar um horário na agenda do usuário **barbeiro** | A | A |
| RF04 | O usuário **barbeiro** pode cancelar um horário em sua agenda, agendado pelo cliente. | A | A |
| RF05 | O usuário **barbeiro** pode ver os usuários **cliente** agendado no dia. | A | A |

\* B = Baixa, M = Média, A = Alta.

## Requisitos Não-funcionais

1. O sistema Web deve ser responsivo de forma a proporcionar a utilização de qualquer uma de suas funcionalidades em resoluções de 576 pixels até 1080 pixels;
2. O sistema deve estar disponível em qualquer período, do dia e da noite em regime 24/7;
3. A aplicação deve estar disponível nos modos claro e escuro, possibilitando assim maior conforto para o usuário;
4. O sistema deve garantir a segurança das senhas dos usuários, criptografando-as ao serem inseridas no banco de dados;
5. Um usuário **barbeiro** deve ser capaz de definir datas que não irá fornecer agendamento disponível;
6. Um usuário **barbeiro** deve ser capaz de definir horários em que não irá fornecer agendamento disponível;

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| RNF01 | O sistema Web deve ser responsivo de forma a proporcionar a utilização de qualquer uma de suas funcionalidades em qualquer resolução. | A |
| RNF02 | O sistema deve estar disponível em qualquer período, do dia e da noite em regime 24/7. | A |
| RNF03 | O sistema deve garantir a segurança das senhas dos usuários, criptografando-as ao serem inseridas no banco de dados. | A |
| RNF04 | O sistema deve garantir que as datas agendadas sempre estejam corretas. | A |

\* B = Baixa, M = Média, A = Alta.

## Protótipo Navegável do Sistema

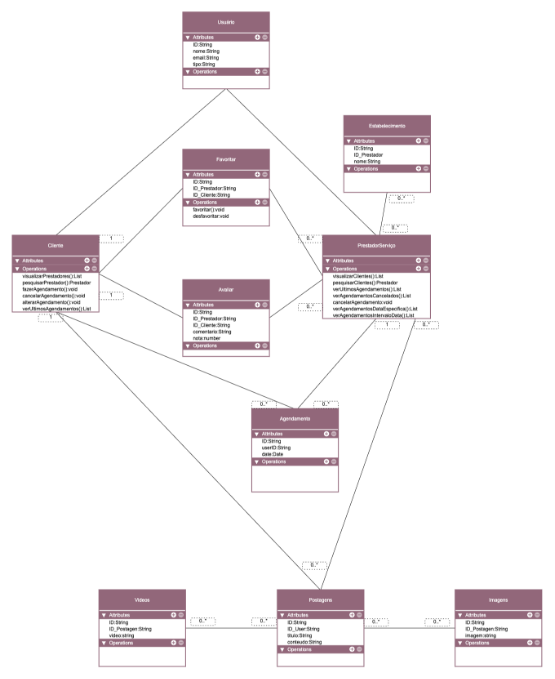
Link protótipo navegável:

<https://www.figma.com/proto/q4d9fHk9PbUfr5GXJHRNBr/PUCMinas-GoBarber?type=design&node-id=34-1181&t=OQSc5FUr7FzbBXgL-1&scaling=min-zoom&page-id=34%3A1180&starting-point-node-id=34%3A1181&show-proto-sidebar=1&mode=design>

Vídeo do Protótipo Navegável:

<https://github.com/andre-aw6/puc-minas/blob/master/Documentation/puc-minas.mp4>

## Diagrama de Classes de Domínio



## Arquitetura da Solução

## Padrão Arquitetural

O Padrão arquitetural escolhido foi o MVC, onde a arquitetura foi modularizada utilizando DDD e TDD, em alguns Domínios não foi necessária a criação da camada de View, e a camada Model foi renomeada para Entities e foi utilizado Repositories para auxiliar os models.

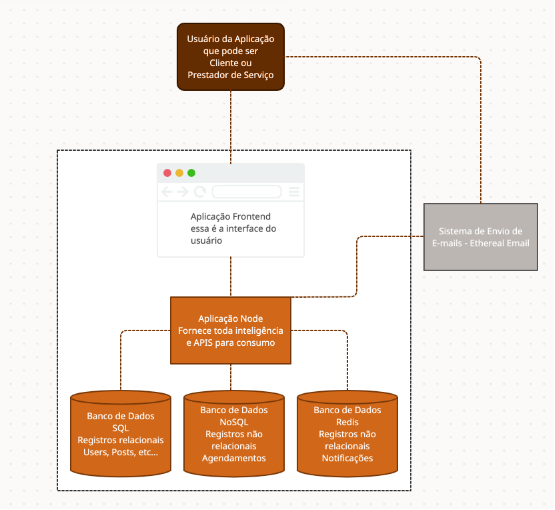
Tecnologias utilizadas:

* Backend
  + Typescript;
  + Node;
  + Express;
  + Jest;
  + TypeORM;
  + Postgres;
  + Redis;
  + MongoDB;
* Frontend
  + Typescript;
  + React;
  + Styled Components;
  + Axios;
* Infra
  + AWS;
  + Docker;
  + Nginx;

## C4 model - Diagrama de Contexto

Abaixo temos um simples Diagrama de Contexto da aplicação, onde é apresentado a aplicação.

Temos o Usuário que pode ser tanto um Cliente como um Prestador de Serviço, que acessa um site que é a Interface do Usuário, que se comunica com a Aplicação desenvolvida em Node, que fornece toda inteligência e uma API de comunicação, que por sua vez se comunica com algumas Bases de Dados, onde é persistido todas informações referente a Aplicação e por fim temos um serviço externo de envio de E-mails utilizado por toda a Aplicação.



## Frameworks de Trabalho

Os Frameworks utilizados nessa aplicação foram os seguintes:

**Backend**:

Express, que é um Framework Web bem flexível, podendo nos permitir utilizar ele conforme for preciso, essa flexibilidade, nos ajuda a definir a Arquitetura mais adequada, e esse Framework é usado muito pelo mercado.

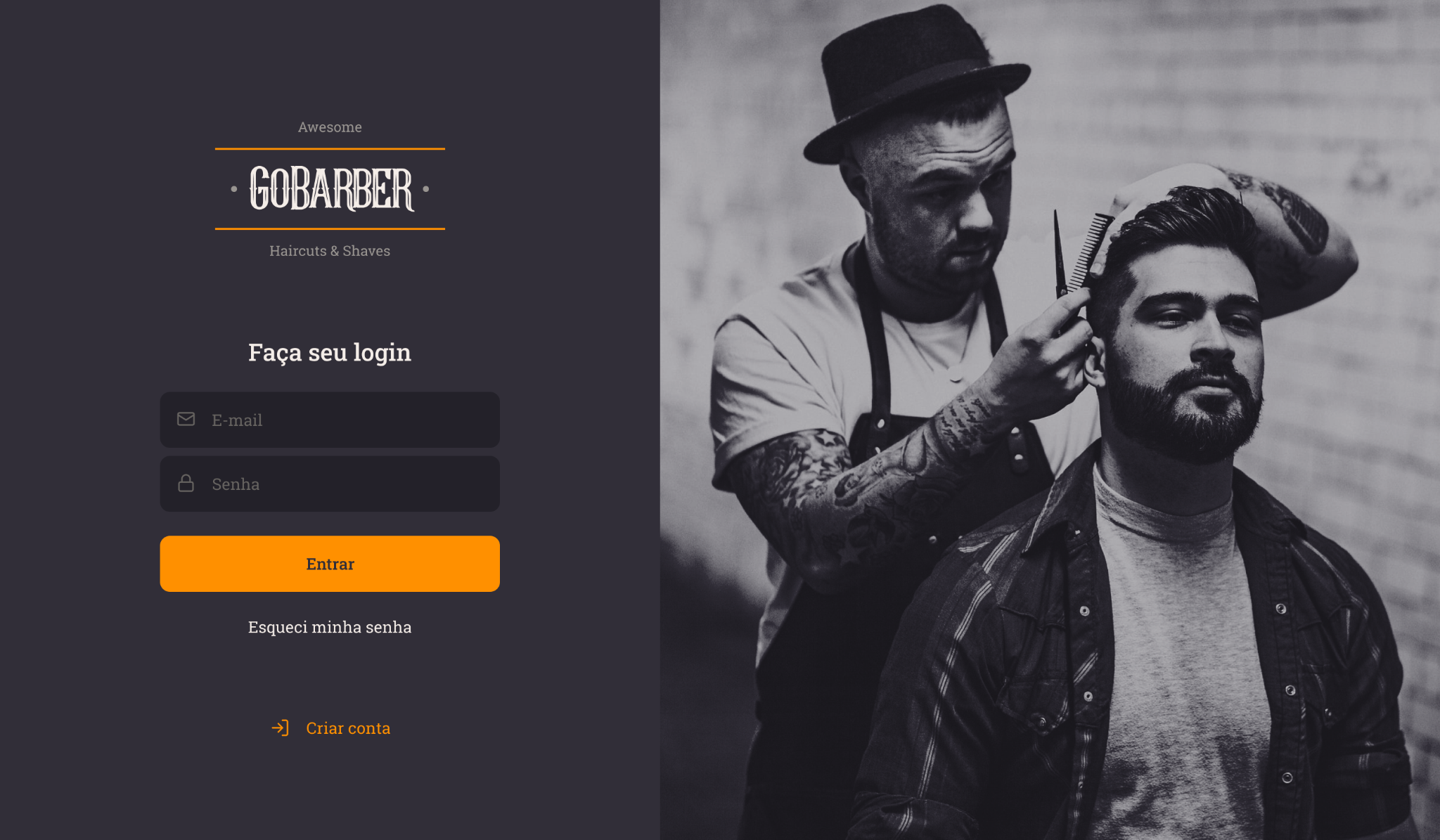
**ORM**:

Para a persistência de Dados foi escolhido o TypeORM, pelo fato dele poder se comunicar tanto com bancos SQL e NoSQL, e nos dá suporte a typescript.

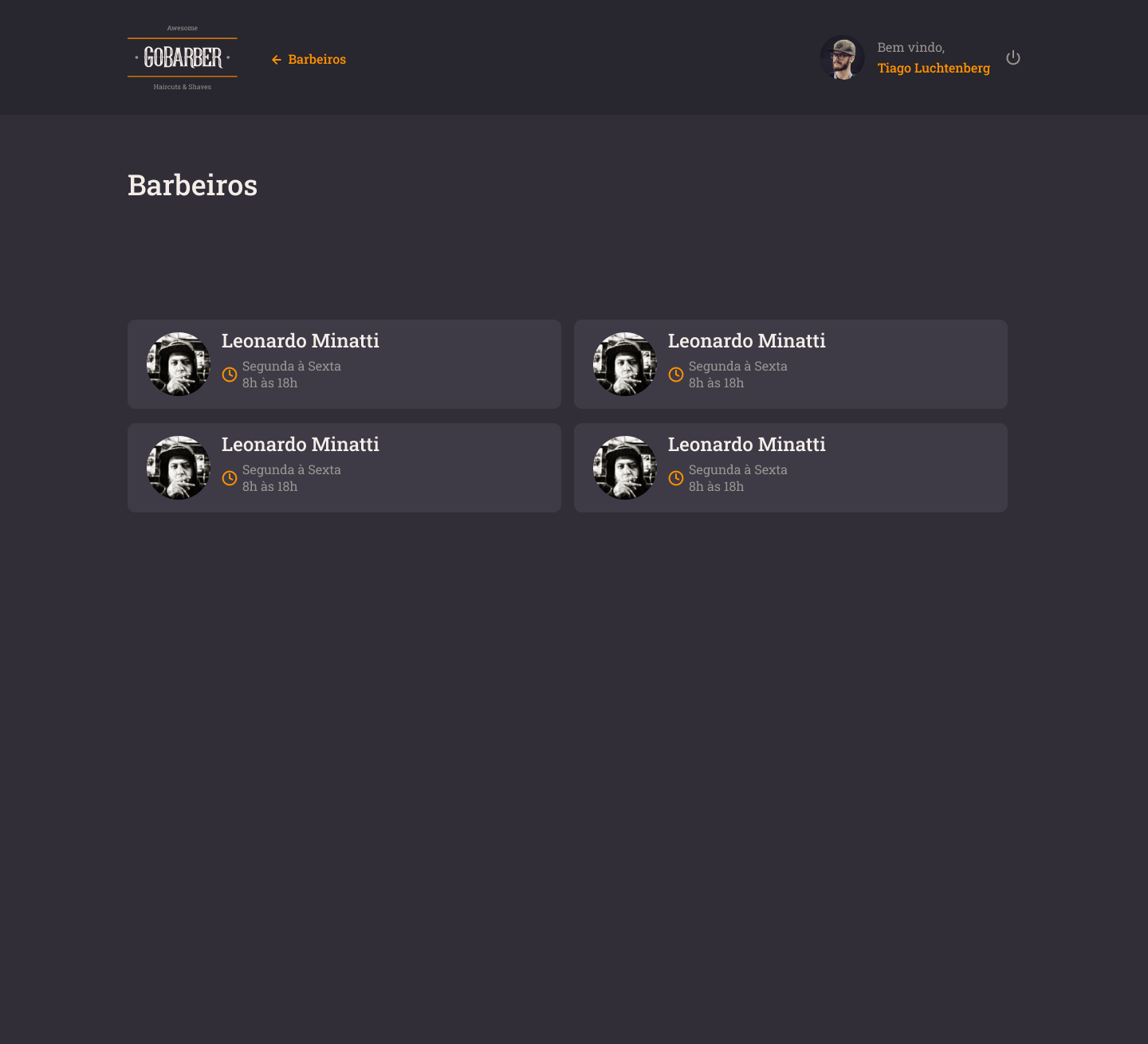
**Frontend**:

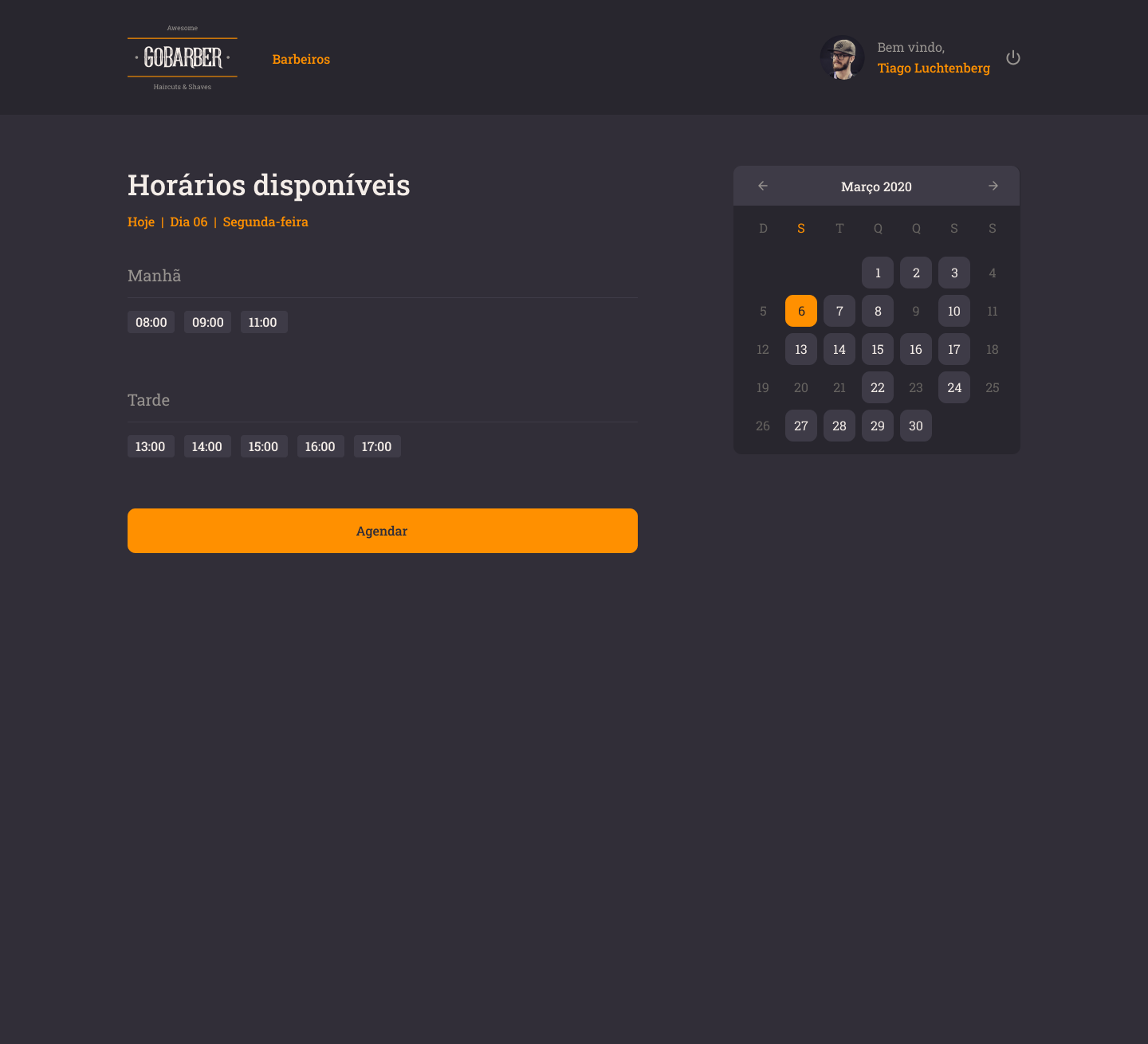
Foi utilizado o React, e boa parte de seu Ecossistema, para a construção de toda interface, amplamente utilizado no mercado, e mantemos toda Stack baseada em Typescript.

## Estrutura Base do Front End

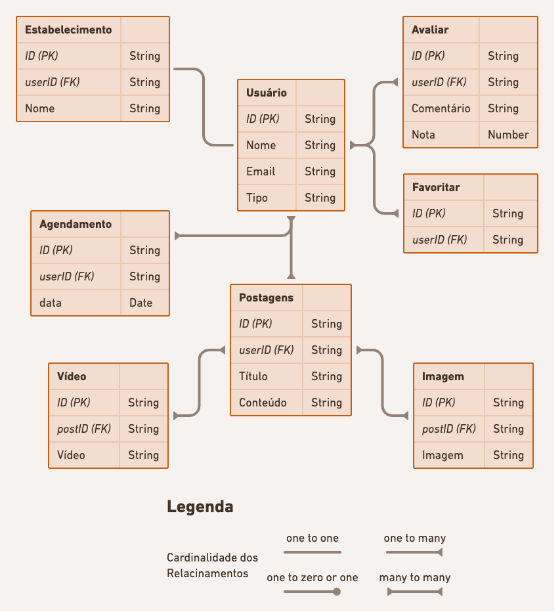








## Modelo Relacional



## Plano de Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  | Fazer um agendamento com o Prestador de Serviço. | Conseguir fazer um agendamento em uma data escolhida. | Clicar na data desejada no calendário, e logo em seguida clicar no botão Agendar. | O sistema apresenta uma mensagem de sucesso do agendamento e continua na tela de agendamento. |
|  | Fazer um agendamento com o Prestador de Serviço. | Não permitir o agendamento em uma data que já exista agendamento. | Clicar na data que já exista um agendamento, e logo em seguida clicar no botão Agendar. | O sistema apresenta uma mensagem de erro do agendamento e continua na tela de agendamento. |
|  | Cancelar um agendamento | Conseguir cancelar um agendamento. | Clicar na data que já exista um agendamento, e logo em seguida clicar no botão Cancelar. | O sistema apresenta uma mensagem de sucesso do agendamento cancelado e continua na tela de agendamento. |
|  | Cancelar um agendamento | Não permitir o cancelamento de uma data que esteja livre. | Clicar na data em que não exista um agendamento, e logo em seguida clicar no botão Cancelar. | O sistema apresenta uma mensagem de erro ao cancelar o agendamento e continua na tela de agendamento. |

## 

## Relatório de Execução de Testes de Software

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de teste** | **Saída esperada** | **Resultados encontrados** | **Aprovado?** |
|  | Fazer uma postagem no Blog. | Exibir mensagem de sucesso e salvar a postagem. | Exibir mensagem de sucesso. | Sim |
|  | Deletar uma postagem no Blog. | Exibir mensagem de sucesso e excluir a postagem. | Exibir mensagem de sucesso. | Sim |

## 

## Apropriação de Horas no Projeto

|  |  |  |
| --- | --- | --- |
| **Histórico de apropriação de horas** | | |
| **Data do registro** | **Atividade** | **Quantidade de horas** |
| 08/09/2023 | Rever toda matéria referente ao Projeto Integrado, textos e vídeos. | 8 |
| 10/09/2023 | Relatório Técnico: Objetivos do Trabalho, Apresentação do Problema, Descrição Geral do Software. | 10 |
| 12/09/2023 | Relatório Técnico: Identificação de atores, requisitos funcionais e não funcionais. | 12 |
| 14/09/2023 | Relatório Técnico: Identificação da ferramenta e desenvolvimento do diagrama de Caso de Uso. | 8 |
| 18/09/2023 | Relatório Técnico: Descrição casos de usos. | 2 |
| 24/09/2023 | Relatório Técnico: Criação do Protótipo de Interface, navegável e seus itens relacionados. | 10 |
| 27/09/2023 | Relatório Técnico: Criação do Diagrama de Classes de domínio. | 14 |
| 04/10/2023 | Relatório Técnico: Definição do Padrão Arquitetural do Projeto e Tecnologias. | 13 |
| 06/10/2022 | Relatório Técnico: Criação do Diagrama de Contexto do Projeto e uma breve explicação. | 8 |
| 08/10/2022 | Relatório Técnico: Apresentar Frameworks e Estrutura Base do Frontend. | 7 |
| 09/10/2022 | Relatório Técnico: Criação do Modelo Relacional do Banco de Dados. | 9 |
| 10/10/2023 | Relatório Técnico: Criação do Plano e Relatório de Execução de Testes. | 3 |
| 11/11/2023 | Implementação do sistema e ajustes. | 10 |
| 08/09/2023 | Relatório Técnico: Finalização do Projeto com as Descrições Finais e vídeo de apresentação. | 11 |

## 

## Código da Aplicação

**Github da Aplicação**

<https://github.com/andre-aw6/puc-minas.git>

**Link da Aplicação:**

​​<https://www.orusbarber.com.br/>

**API da Aplicação:**

<https://api.orusbarber.com.br/sessions>

## Avaliação Retrospectiva

A jornada foi permeada por uma série de lições valiosas. Uma das mais notáveis foi a importância da gestão de tempo e do planejamento. Em projetos de engenharia de software, os desafios técnicos são apenas uma faceta; a capacidade de gerir prazos, integrar tecnologias e adaptar-se às mudanças é igualmente crucial.

## Objetivos Estimados

Manter um cronograma rigoroso provou ser um desafio, especialmente quando equilibrado com outras atividades acadêmicas e compromissos. No entanto, com persistência, adaptação e reorganização contínua, fui capaz de superar esse obstáculo. A cada etapa, o projeto foi sendo modelado e adaptado com base nos conhecimentos adquiridos ao longo do curso, permitindo que ele evoluísse de maneira coesa e eficaz.

## Objetivos Alcançados

Desde o início, havia uma ênfase clara em manter um olhar atento ao cronograma e distribuir o tempo de maneira eficiente. Embora tenham surgido desafios ao longo do caminho, estou satisfeito em dizer que os objetivos propostos, em termos de tempo e cronograma, foram alcançados. A aplicação foi entregue com funcionalidades que não só cumpriram os critérios estabelecidos mas também refletiram a aplicação prática das competências desenvolvidas ao longo do programa de pós-graduação.

## Lições aprendidas

Este projeto não foi apenas uma demonstração de habilidades técnicas, mas também uma prova de resiliência, gestão de tempo e adaptação. A experiência, embora desafiadora, foi extremamente gratificante. Agora, equipado com o conhecimento e a experiência prática, estou ansioso para enfrentar desafios maiores e mais complexos no campo da Engenharia de Software.

|  |  |  |
| --- | --- | --- |
|  | **Retrospectiva (Lições Aprendidas)** | |
|  | **Descrição da Lição** | **Classificação** |
| 1 | Flexibilidade | Positiva |
| 2 | Comunicação | Positiva |
| 3 | Testes Robustos | Positiva |
| 4 | Valorização de Feedback | Positiva |

## 

## Referências

A elaboração deste trabalho e seu consequente sucesso não teriam sido possíveis sem a contribuição, direta ou indireta, de várias fontes. Abaixo estão as referências que deram suporte a este projeto:

1. Documentação Oficial do React: A documentação detalhada e exemplos práticos fornecidos pelo time do React foram fundamentais para o desenvolvimento das interfaces do aplicativo.

2. Node.js e Express.js: As respectivas documentações dessas tecnologias foram consultadas extensivamente, fornecendo orientações sobre como configurar, otimizar e manter o servidor do aplicativo.

3. Smith, J. (2018). Gestão de Projetos em Engenharia de Software. São Paulo: Editora Tecnologia. Este livro proporcionou insights valiosos sobre como abordar o planejamento e a execução de projetos de software.

4. Rodriguez, M. (2020). Desenvolvimento Ágil com React e Node.js. Rio de Janeiro: Editora Moderna. A metodologia e as práticas recomendadas neste livro guiaram vários aspectos do desenvolvimento do aplicativo.

5. Aulas e Seminários da Pós-Graduação: Os conhecimentos transmitidos pelos professores e os debates em sala de aula foram cruciais para moldar a abordagem e as decisões tomadas ao longo do projeto.

6. Forums e Comunidades Online: Plataformas como Stack Overflow, Reddit e comunidades específicas do React e Node.js foram frequentemente consultadas para resolver dúvidas específicas e otimizar soluções.

Agradeço a todos os autores, contribuidores e comunidades mencionados acima por seu valioso trabalho e contribuição ao campo da Engenharia de Software. Seus esforços não só facilitaram o desenvolvimento deste projeto, mas também enriqueceram o conhecimento e as habilidades do autor.