**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Projeto Integrado

Relatório Técnico

goBarber

Alcir Cid Boni Junior

Belo Horizonte

2022.

# 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** |
| 08/01/2022 | 09/01/2022 | 1. Rever toda matéria referente ao Projeto Integrado, textos e vídeos. | Entendimento geral do Projeto Integrado. |
| 15/01/2022 | 16/01/2022 | 2. Relatório Técnico: Objetivos do Trabalho, Apresentação do Problema, Descrição Geral do Software. | Confecção do Relatório Técnico – Descritivos. |
| 22/01/2022 | 22/01/2022 | 3. Relatório Técnico: Identificação de atores, requisitos funcionais e não funcionais. | Confecção do Relatório Técnico – Definição conceitual. |
| 29/01/2022 | 30/01/2022 | 4. Relatório Técnico: Identificação da ferramenta e desenvolvimento do diagrama de Caso de Uso. | Confecção do Relatório Técnico – Diagrama de Caso de Uso. |
| 12/02/2022 | 13/02/2022 | 5. Relatório Técnico: Descrição casos de usos. | Confecção do Relatório Técnico – Descrição dos casos de usos. |
| 05/03/2022 | 06/03/2022 | 6. Relatório Técnico: Criação do Protótipo de Interface, navegável e seus itens relacionados. | Confecção do Relatório Técnico – Inclusão das referências do Protótipo. |
| 07/05/2022 | 08/05/2022 | 7. Relatório Técnico: Criação do Diagrama de Classes de domínio. | Confecção do Relatório Técnico – Diagrama de Classes de Domínio. |
| 14/05/2022 | 15/05/2022 | 8. Relatório Técnico: Definição do Padrão Arquitetural do Projeto e Tecnologias. | Confecção do Relatório Técnico – Descrição da Arquitetura escolhida e tecnologias utilizadas. |
| 28/05/2022 | 29/05/2022 | 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. |
| 04/06/2022 | 05/06/2022 | 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. |
| 11/05/2022 | 12/05/2022 | 11. Relatório Técnico: Criação do Modelo Relacional do Banco de Dados. | Confecção do Relatório Técnico – Diagrama do Modelo Relacional. |
| 18/05/2022 | 19/05/2022 | 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. |
| 25/05/2022 | 26/05/2022 | 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. |

***Observação: acrescente ou retire linhas, caso seja necessário.***

## Introdução

***2.1. Objetivos do trabalho***

O objetivo geral deste projeto é desenvolver uma plataforma onde será desenvolvido uma API para fornecer um serviço que faça a ligação entre um Prestador de Serviço (Barbeiro) a um Usuário (Cliente), e um Site para consumir essa API facilitando assim tanto a prestação de serviço quanto a busca de um profissional.

Inicialmente a plataforma será desenvolvida apenas para Web, onde iremos ter uma única interface, que poderá ser acessada pelo Prestador de Serviço (Barbeiro) e Usuário (Cliente), essa interface fornecerá todos recursos para ambos, e tendo a possibilidade de estender toda funcionalidade para um APP Mobile.

***2.2. Apresentação do problema***

Com o avanço das tecnologias e a popularização do acesso à internet, aplicativos e softwares, vem ganhando espaço para facilitar o dia a dia de empresas e usuários.

Com isso várias soluções vêm sendo disponibilizadas para facilitar o dia a dia, pensando nisso identificamos a necessidade da criação de uma solução para Barbeiros , para facilitar o dia a dia desses profissionais, no sentido de ter suas agendas preenchidas de forma mais ágil. E pelo lado do cliente, que também tem pouco tempo livre disponível para perder, a disponibilização de uma plataforma onde é possível buscar um profissional que possa atendê-lo, e já poder verificar sua agenda é de extrema importância.

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

A elaboração desse software tem por objetivo fornecer uma plataforma que possa conectar o Prestador de Serviço (Barbeiro) a um Usuário (Cliente), facilitando assim todo o processo.

O Prestador de Serviço (Barbeiro) poderá fazer o cadastro na plataforma, criando seu perfil e disponibilizando toda sua agenda.

O Usuário (Cliente) poderá fazer o cadastro na plataforma, criando seu perfil e poderá fazer consultas de profissionais cadastrados na plataforma que possa atendê-lo, verificando sua agenda.

## Definição Conceitual da Solução

## Diagrama de Casos de Uso

|  |
| --- |

## Requisitos Funcionais

1. Um usuário deve ser capaz de cadastrar seu perfil como **(Prestador de Serviço)** ou **(Cliente)**;
2. Um usuário deve ser capaz de receber e-mails de atualizações tanto como **(Prestador de Serviço)** ou **(Cliente)**;
3. Um usuário **(Cliente)** deve ser capaz de pesquisar um **(Prestador de Serviço)**;
4. Um usuário **(Cliente)** deve ser capaz de listar todos **(Prestadores de Serviços)**;
5. Um usuário **(Cliente)** deve ser capaz de acessar o perfil do **(Prestador de Serviço)**;
6. Um usuário **(Cliente)** deve ser capaz de verificar a agenda do **(Prestador de Serviço)**;
7. Um usuário **(Cliente)** deve ser capaz de escolher uma data na agenda do **(Prestador de Serviço)**;
8. Um usuário **(Cliente)** deve ser capaz de escolher um horário nada data escolhida na agenda do **(Prestador de Serviço)**;
9. Um usuário **(Cliente)** deve ser capaz de confirmar o agendamento no **(Prestador de Serviço)** escolhido;
10. Um usuário **(Cliente)** deve ser capaz de cancelar seus horários agendados;
11. Um usuário **(Cliente)** deve ser capaz de alterar sua data agendada;
12. Um usuário **(Cliente)** deve ser capaz de alterar seus horários agendados;
13. Um usuário **(Cliente)** deve ser capaz de ver seus horários agendados;
14. Um usuário **(Cliente)** deve ser capaz de ver os últimos **(Prestadores de Serviços)** que fez o agendamento;
15. Um usuário **(Prestador de Serviço)** deve ser capaz de ver os últimos **(Clientes)** agendados;
16. Um usuário **(Prestador de Serviço)** deve ser capaz de ver os últimos **(Clientes)** atendidos;
17. Um usuário **(Prestador de Serviço)** deve ser capaz de cancelar um agendamento;
18. Um usuário **(Prestador de Serviço)** deve ser capaz de ver os **(Clientes)** cancelados;
19. Um usuário **(Prestador de Serviço)** deve ser capaz de ver seus agendamentos por data específica;
20. Um usuário **(Prestador de Serviço)** 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 **(Prestador de Serviço)**. | M | A |
| RF02 | O usuário **(Cliente)** tem de acessar a agenda do usuário **(Prestador de Serviço)**. | M | M |
| RF03 | O usuário **(Cliente)** pode agendar ou cancelar um horário na agenda do usuário **(Prestador de Serviço)** | A | A |
| RF04 | O usuário **(Prestador de Serviço)** pode cancelar um horário em sua agenda, agendado pelo usuário **(Cliente)**. | A | A |
| RF05 | O usuário **(Prestador de Serviço)** 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 **(Prestador de Serviço)** deve ser capaz de definir datas que não irá fornecer agendamento disponível;
6. Um usuário **(Prestador de Serviço)** 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/community/file/1102323700794746019>

Caso o link não abra acesse o perfil que lista o protótipo:

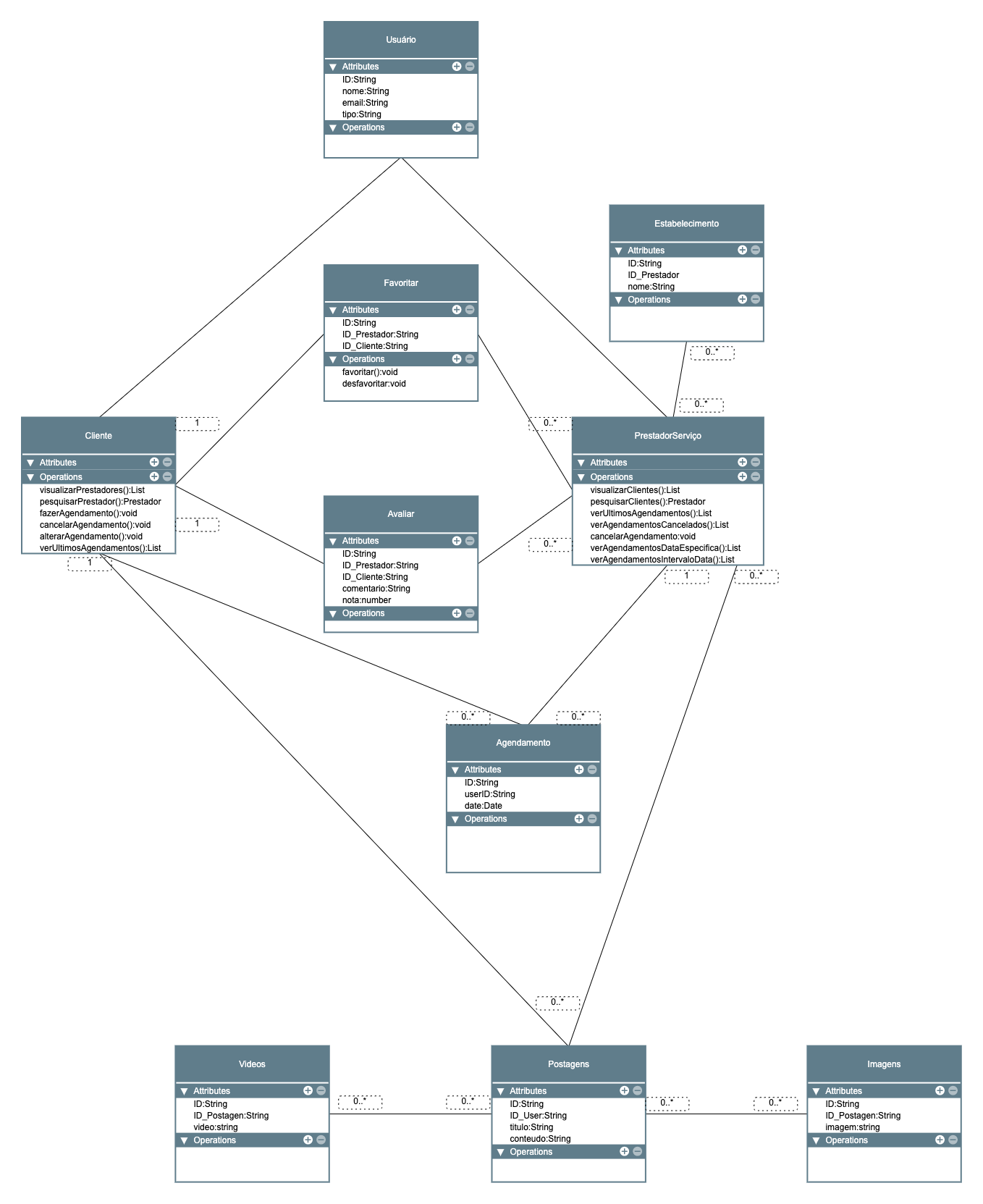
<https://www.figma.com/@cajucomunica>

Vídeo do Protótipo Navegável:

<https://github.com/alcir-junior-caju/study-tcc-puc-minas/blob/main/docs/video-prototipo-interface.mp4>

Obs.: O vídeo ficou parecendo uma troca de imagens, recomendo acessar o link do figma, lá parece mais um site clicando nos botões de login, etc…

## 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.

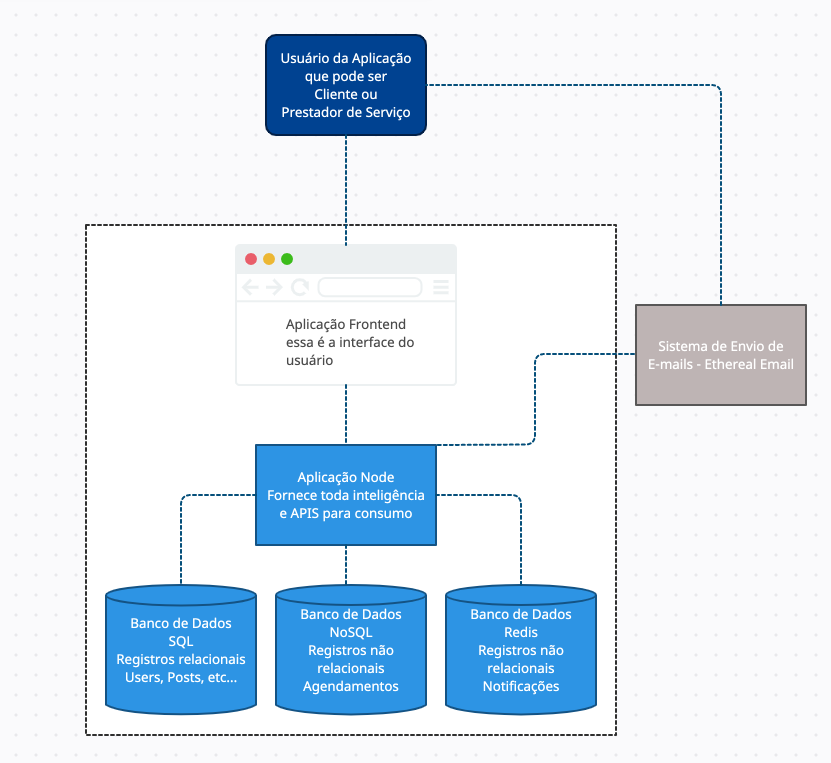
Tecnologias utilizadas:

* 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.

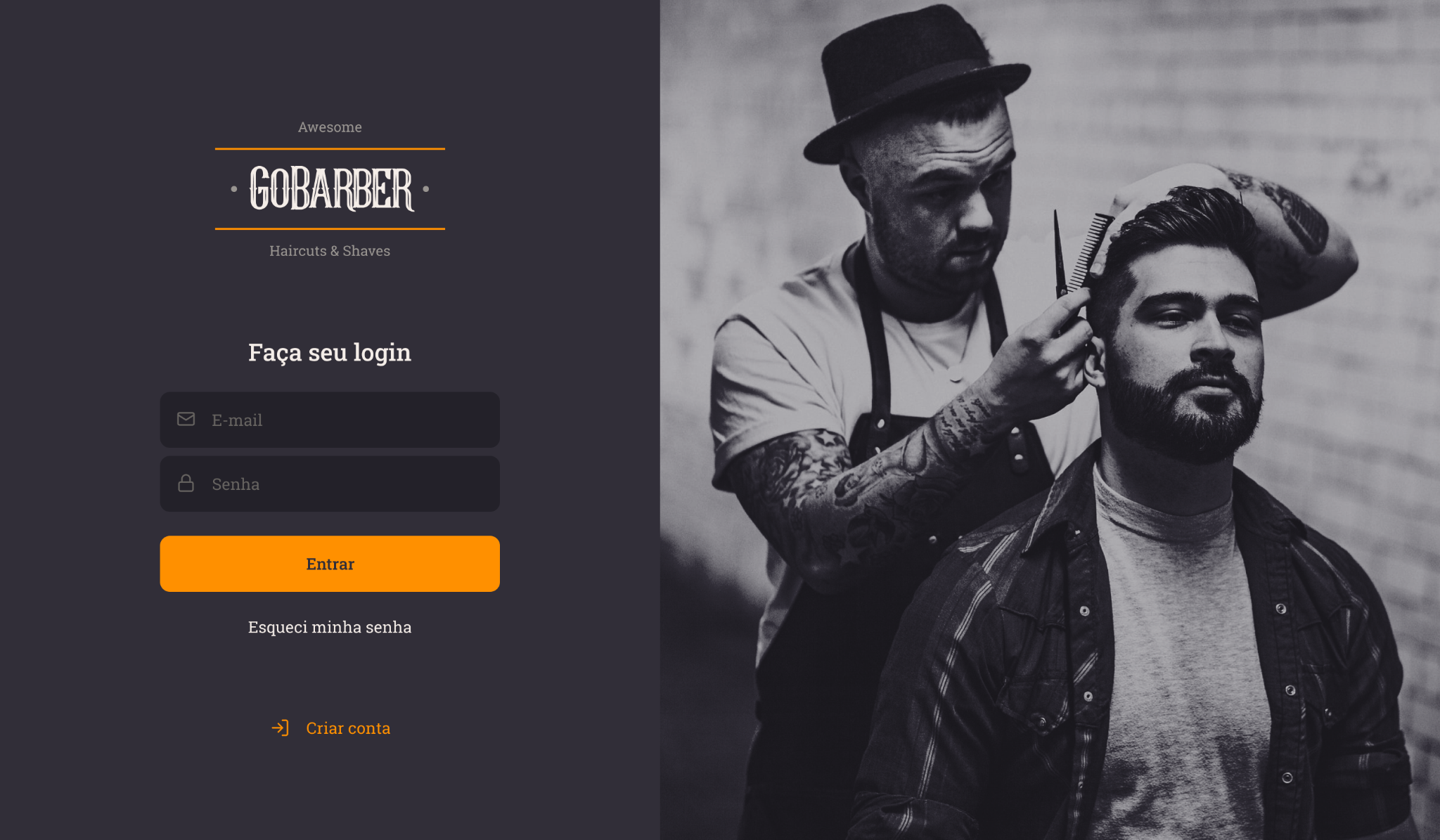
**ORM**:

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.

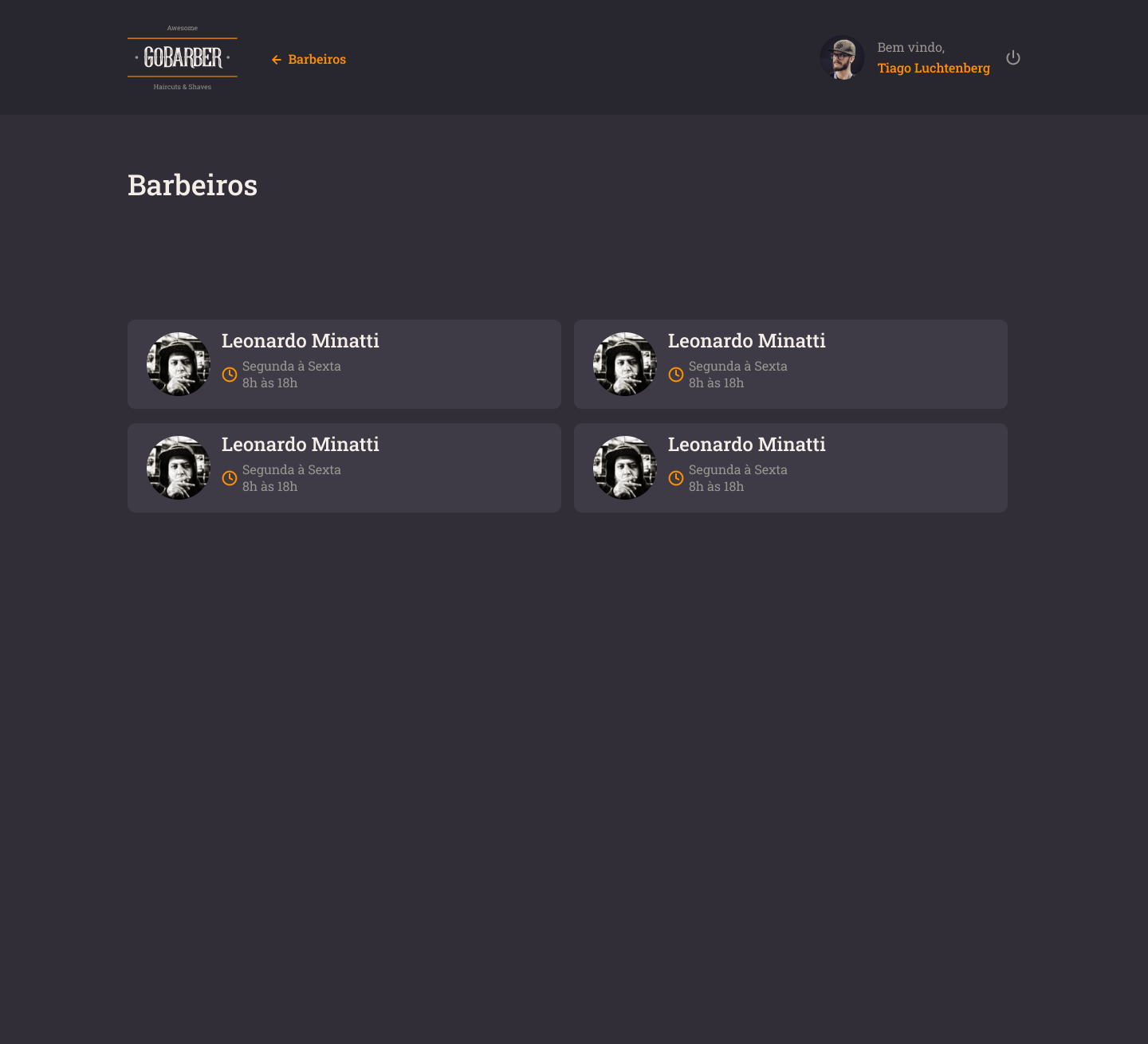
**Frontend**:

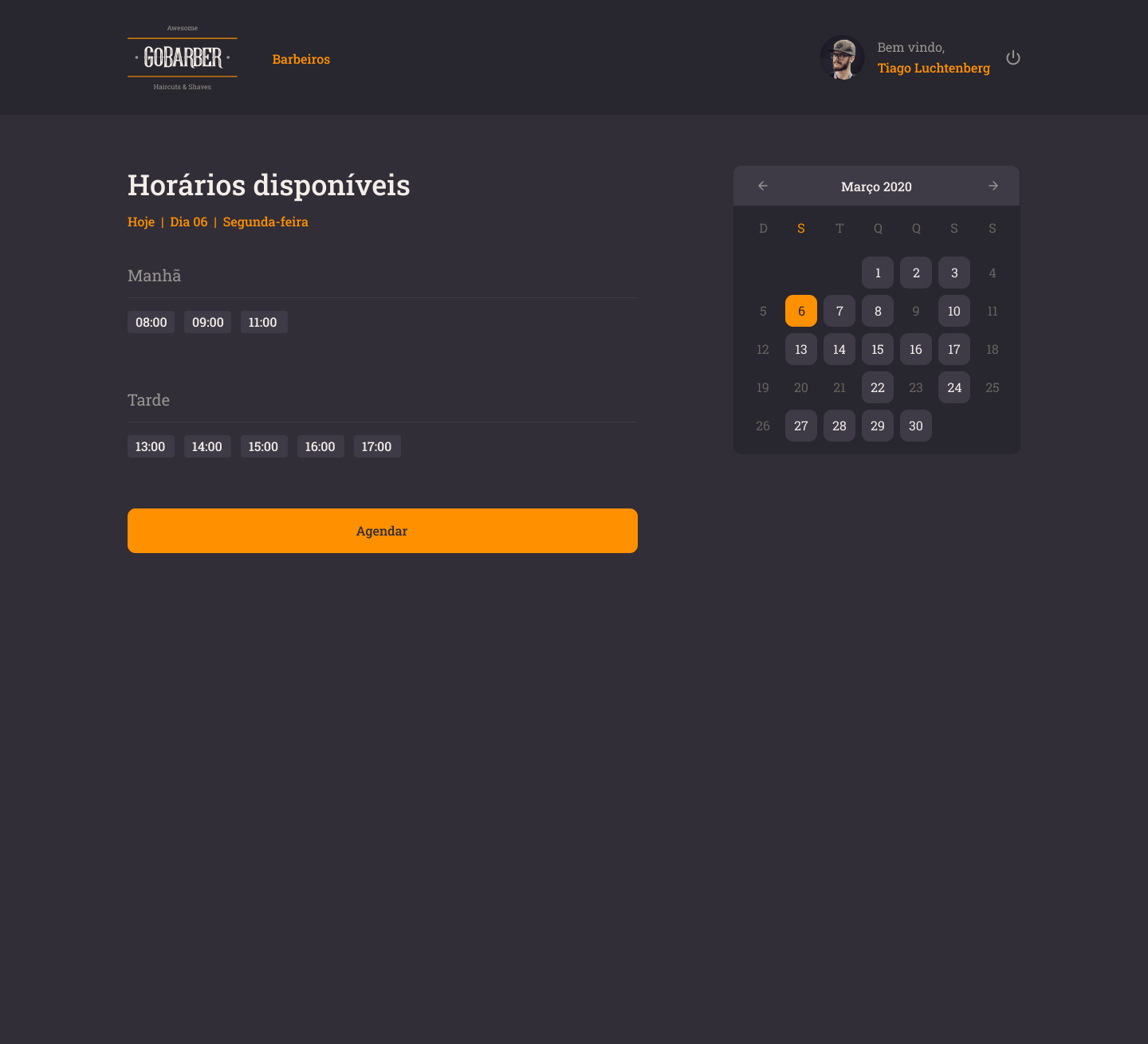
Foi utilizado o React, e boa parte de seu Ecossistema, para a construção de dota interface, amplamente utilizado no mercado, e mantemos toda Stack baseada em Typescript.

## Estrutura Base do Front End

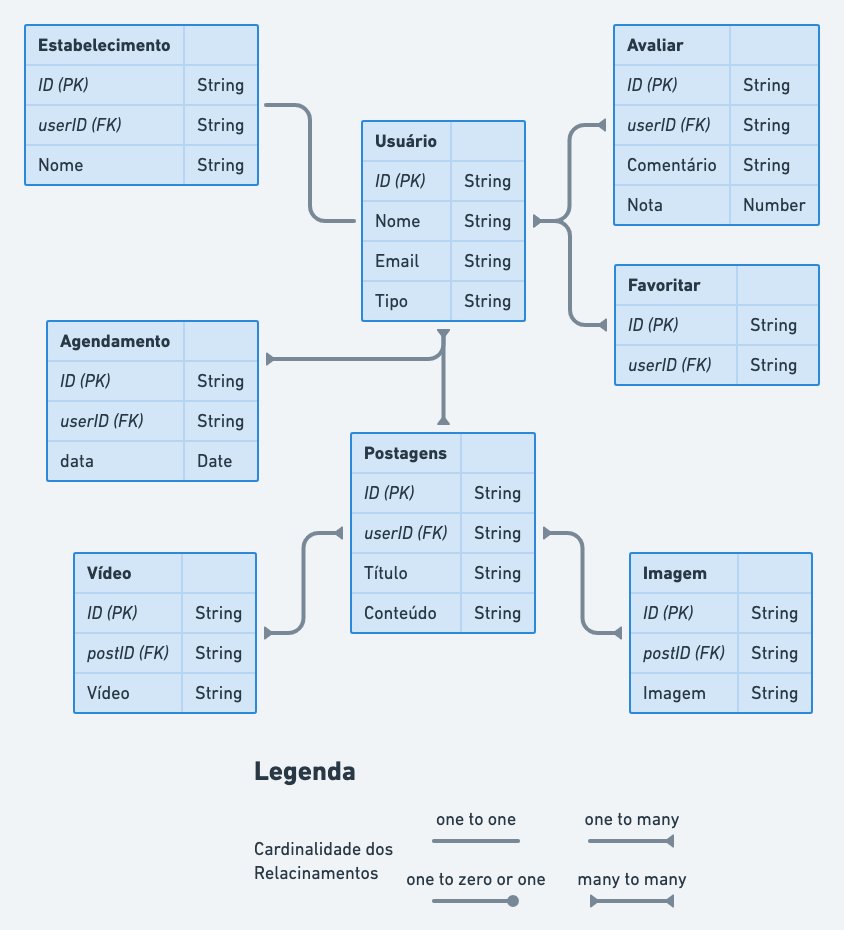








## Modelo Relacional



## 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** |
| 09/01/2022 | Rever toda matéria referente ao Projeto Integrado, textos e vídeos. | 16 |
| 16/01/2022 | Relatório Técnico: Objetivos do Trabalho, Apresentação do Problema, Descrição Geral do Software. | 12 |
| 22/01/2022 | Relatório Técnico: Identificação de atores, requisitos funcionais e não funcionais. | 20 |
| 30/01/2022 | Relatório Técnico: Identificação da ferramenta e desenvolvimento do diagrama de Caso de Uso. | 18 |
| 13/02/2022 | Relatório Técnico: Descrição casos de usos. | 10 |
| 06/03/2022 | Relatório Técnico: Criação do Protótipo de Interface, navegável e seus itens relacionados. | 20 |
| 08/05/2022 | Relatório Técnico: Criação do Diagrama de Classes de domínio. | 20 |
| 15/05/2022 | Relatório Técnico: Definição do Padrão Arquitetural do Projeto e Tecnologias. | 18 |
| 21/05/2022 | Relatório Técnico: Criação do Diagrama de Contexto do Projeto e uma breve explicação. | 12 |
| 22/05/2022 | Relatório Técnico: Apresentar Frameworks e Estrutura Base do Frontend. | 12 |
| 04/06/2022 | Relatório Técnico: Criação do Modelo Relacional do Banco de Dados. | 6 |
| 05/06/2022 | Relatório Técnico: Criação do Plano e Relatório de Execução de Testes. | 10 |
| 11/06/2022 | Implementação do sistema e ajustes. | 16 |
| 12/06/2022 | Relatório Técnico: Finalização do Projeto com as Descrições Finais e vídeo de apresentação. | 16 |

## Código da Aplicação

**Github da Aplicação**

Obs.: Ainda estou trabalhando no desenvolvimento da Aplicação, então possíveis mudanças podem acontecer.

<https://github.com/alcir-junior-caju/study-tcc-puc-minas>

**Link da Aplicação:**

​​<https://cajugobarber.netlify.app/>

[client@gobarber.com.br](mailto:client@gobarber.com.br) / 123456

[provider@gobarber.com.br](mailto:provider@gobarber.com.br) / 123456

**API da Aplicação:**

<https://cajugobarber.netlify.app/api/>

<https://insomnia.rest/run/?label=GoBarberAPI&uri=https%3A%2F%2Fraw.githubusercontent.com%2Falcir-junior-caju%2Fstudy-js-node-gostack-gobarber%2Fmaster%2Fgobarber-api-insomnia.json>

Indique o ***link*** para acesso ao **vídeo de apresentação de seu projeto**. Espera-se a produção de um vídeo sintético de, **no máximo**, **5 minutos**, no formato **MP4**, apresentando o projeto e a solução desenvolvida. Sugere-se que o aluno apresente um ciclo completo do que pode ser realizado pelos usuários principais da solução.>

## Avaliação Retrospectiva

Nesse tempo de aplicação dos conhecimentos adquiridos ao longo do curso, neste projeto, muitas lições aprendidas e desafios vencidos, um dos maiores desafios foi o cronograma, manter um cronograma foi muito complicado, devido às outras atividades, o resultado foi satisfatório, mesmo porque o pequeno projeto foi adaptado e modelado conforme alguns conhecimentos adquiridos no curso.

## Objetivos Estimados

Os objetivos sempre foram focados no cronograma e como distribuir o tempo, mas aos poucos foi possível ir alcançando cada etapa estabelecida, os objetivos estimados no fim foram alcançados.

## Objetivos Alcançados

Os objetivos propostos em relação a tempo e cronograma foram alcançados, sempre com dificuldades, mas sempre alcançados, de forma que aos poucos o projeto foi entregue com algumas funcionalidades, que atendia os critérios.

## Lições aprendidas

|  | **Retrospectiva (Lições Aprendidas)** | |
| --- | --- | --- |
|  | **Descrição da Lição** | **Classificação** |
| 1 | Aplicação de tempos corretos aos cronogramas. | Positiva |
| 2 | Aplicação de conceitos teóricos ao projeto. | Positiva |
| 3 | Mais conhecimento adquirido ao longo do processo. | Positiva |
| 4 | Ideias para aplicar em projetos reais. | Positiva |

## Referências

Esse projeto foi desenvolvido a partir de um treinamento que fiz a algum tempo e o resultado foi esse projeto, onde adicionei novas funcionalidades, e com isso foi elaborado todo relatório técnico e aplicado padrões ao software.

Rocketseat. **Treinamento Ignite.** Rio do Sul: 2020.