



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

MATHEUS MARQUES DE SOUZA

BARBERWEB: PLATAFORMA ONLINE PARA AGENDAMENTOS

**Assis/SP
2023**



**Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"**

MATHEUS MARQUES DE SOUZA

BARBERWEB: PLATAFORMA ONLINE DE AGENDAMENTOS

Trabalho de Conclusão de Curso apresentado ao curso de Análise e Desenvolvimento de Sistemas do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão

**Orientando: Matheus Marques de Souza.
Orientador: Dr. Almir Rogério Camolesi.**

**Assis/SP
2023**

Souza, Matheus Marques de

S729b BarberWeb – Plataforma de agendamentos / Matheus Marques de Souza. -- Assis, 2023.

58p.

Trabalho de Conclusão de Curso (Análise e Desenvolvimento de Sistemas) -- Fundação Educacional do Município de Assis (FEMA), Instituto Municipal de Ensino Superior de Assis (IMESA), 2023.

Orientador: Prof. Dr. Almir Rogério Camolesi.

1. Plataforma digital. 2. JavaScript. 3. Linguagem de programação de domínio específico. I Camolesi, Almir Rogério. II Título.

CDD 005.2

BARBERWEB: PLATAFORMA ONLINE DE AGENDAMENTO

MATHEUS MARQUES DE SOUZA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Dr. Almir Rogério Camolesi

Examinador: _____
Esp. Domingos de Carvalho Villela Junior

**Assis/SP
2023**

AGRADECIMENTOS

Primeiro devo agradecer a Deus por me proporcionar força, foco e principalmente determinação para chegar até essa etapa de minha vida. Também agradecer a minha mãe Leonor que sempre me apoiou e me incentivou a continuar, meu irmão Gabriel que serve como uma inspiração e exemplo, e meu pai Marco que sempre foi meu super-herói.

Não posso deixar de agradecer a meu grande amor Laiane, por estar comigo nos momentos de estresse, e me orientar para me fazer um homem e aluno melhor. Aos meus companheiros de sala, por me aconselhar e ajudar nesses anos em que estamos juntos, e por último e não menos importante, agradecer meu orientador e amigo, Almir Rogério Camolesi, por ter paciência e firmeza em seus conselhos e por me mostrar caminhos e soluções para todos os problemas que encontrei nesse trajeto.

**Assis/SP
2023**

RESUMO

O trabalho em questão apresenta todo o desenvolvimento de uma plataforma online, desenvolvida em JavaScript, onde empresários donos de estabelecimentos voltados à higiene e cuidados faciais masculinos (Barber-Shops) podem cadastrá-los, para que haja mais visibilidade em suas cidades, além de facilidades para o agendamentos. Já para os clientes, a plataforma se resume a um portal de variedades e facilidades para também realizar o agendamentos e seleção de serviços desejados.

Palavras-chave: *BarberWeb, Plataforma Online, Javascript, Sistema Web.*

ABSTRACT

The paper in question presents the all development of an online platform, developed with JavaScript, where businesspersons that own masculine facial hygiene and care establishments (Barber shops) will be able to register their business, to receive more visibility in their region, besides facilitating scheduling processes. While for customers, the platform summarizes into a portal of choice variety, and facilitations for scheduling and selection of desired services.

Keywords: BarberWeb, Online platform, Javascript, Web system.

LISTA DE ILUSTRAÇÕES

Imagem 1: Mapa Mental:	Erro! Marcador não definido.
Imagem 2: Diagrama de Entidade e Relacionamento:	12
Imagem 3: Diagrama de Caso de Uso Geral.....	13
Imagem 4: UC de acesso ao Sistema	14
Imagem 5: UC de Manter Barbeiro	16
Imagem 6: UC de Manter Serviço.....	18
Imagem 7: UC de Agendar Horários.....	20
Imagem 8: Diagrama de Atividade de Agendamento	22
Imagem 9: Layout Inicial de Login:	25
Imagem 10: Layout de Login para Clientes	26
Imagem 11: Layout do Index	26
Imagem 12: Layout de Serviços	27
Imagem 13: Layout de Horários.....	27
Imagem 14: Layout de Agendamentos para Clientes	28
Imagem 15: Layout de Agendamneto para Barbeiros	28
Imagem 16: Layout Header Mobile	29
Imagem 17: HTML Header	30
Imagem 18: CSS Header usado para Mobile	30
Imagem 19: Layout Header Notebooks	31
Imagem 20: CSS Header usado para Notebooks.....	31
Imagem 21: Layout Header para Monitores.....	32
Imagem 22: CSS Header usado para Monitores	32
Imagem 23: Declaração de Valores de Serviços	33
Imagem 24: Salvamento e Definição de Serviços em tela.....	34

Imagem 25: Serviços ostrado em Tela	35
Imagem 26: Função para gravar agendamentos em banco	36
Imagem 27: Layout deProjeto Node e Express	37
Imagem 28: Index Back-End.....	38
Imagem 29: Definições de Rotas para cada verbo	39
Imagem 30: Services	40
Imagem 31: Models	41
Imagem 32: Database.....	42

LISTA DE TABELAS

Tabela 1: Narrativa de Caso de Uso – Acessar Sistema.....	14
Tabela 2: Narrativa de Caso de Uso – Manter Barbeiro	16
Tabela 3: Narrativa de Caso de Uso – Manter Serviço.....	18
Tabela 4: Narrativa de Caso de Uso – Agendamento de Horário.....	20

LISTA DE ABREVIATURAS E SIGLAS

- DER – Diagrama de Entidade e Relacionamento.
- FEMA – Fundação Educacional do Município de Assis.
- UC – Casos de Uso.
- ORM – Object Relational Mapping.

SUMÁRIO

1. INTRODUÇÃO	1
1.1. OBJETIVO.....	2
1.2. JUSTIFICATIVA	3
1.3. MOTIVAÇÃO.....	3
1.4. PERSPECTIVA DE CONTRIBUIÇÃO.....	4
1.5. METODOLOGIA.....	4
1.6. PÚBLICO ALVO	4
1.7. ESTRUTURA DO TRABALHO.....	5
2. CONCEITOS E TECNOLOGIAS USADAS NO PROJETO	6
2.1. UML.....	6
2.2. ASTAH	6
2.3. JAVASCRIPT	7
2.4. VISUAL STUDIO CODE.....	7
2.5. NODEJS.....	8
2.6 EXPRESS	8
2.7 MONGODB	9
2.8 HTML.....	9
2.9 CSS.....	ERRO! MARCADOR NÃO DEFINIDO.
3. PROPOSTA E MODELAGEM DE CASOS DE USOERRO! MARCADOR NÃO DEFINIDO.	
3.1. MAPA MENTAL.....	ERRO! MARCADOR NÃO DEFINIDO.
3.2. DIAGRAMA DE ENTIDADE E RELACIONAMENTOS (DER)	ERRO! MARCADOR NÃO DEFINIDO.
3.3. DIAGRAMA DE CLASSE	ERRO! MARCADOR NÃO DEFINIDO.
3.4. DIAGRAMA DE CASO DE USO	ERRO! MARCADOR NÃO DEFINIDO.
3.3.1 UC01 – ACESSAR SISTEMA	14
3.3.2 UC02 MANTER BARBEIRO.....	16
3.3.3 UC03 – MANTER SERVIÇO.....	18
3.3.4 UC04 – AGENDAR HORÁRIO.....	20
3.4 DIAGRAMA DE ATIVIDADES.....	22

4. IMPLEMENTAÇÃO	23
4.1. OBJETIVOS E IMPLEMENTAÇÃO	23
4.2. METODOLOGIA ABORDADA	ERRO! MARCADOR NÃO DEFINIDO.
4.3. DIFICULDADES ENCONTRADAS	ERRO! MARCADOR NÃO DEFINIDO.
4.4. PRINCIPAIS FUNCIONALIDADES	ERRO! MARCADOR NÃO DEFINIDO.
4.5. TELAS ILUSTRATIVAS	25
4.5.1 LOGIN	25
4.5.2 INDEX	26
4.5.3 SERVIÇOS E HORÁRIOS	27
4.5.4 DIAGRAMA DE ATIVIDADES	28
4.6 CODIFICAÇÃO	29
4.6.1 FRONT-END MOBILE-FIRST	29
4.6.2 JAVASCRIPT NO FRONT-END	33
4.6.3 BACK END	ERRO! MARCADOR NÃO DEFINIDO.
4.6.3.1 INDEX	ERRO! MARCADOR NÃO DEFINIDO.
4.6.3.2 CONTROLLERS	39
4.6.3.3 SERVICES	40
4.6.3.4 MODELS	41
4.6.3.5 UTILS	42
5. CONCLUSÃO	43
REFERÊNCIAS	44

1. INTRODUÇÃO

Hoje em dia, a visão de que um homem tem que ser bruto e não se preocupar com sua aparência, está quase extinta, pois o homem se torna cada dia mais vaidoso, tendo foco tanto na moda como, roupas, sapatos e acessórios. Mas também um foco enorme em sua higiene pessoal e cuidados corporais, pele, unhas, sobrancelhas e principalmente cabelo e barba. Segundo uma pesquisa realizada pela ABIHPEC - Associação Brasileira de Higiene Pessoal, Perfumaria e Cosméticos (2016):

Cerca de 83% dos homens entrevistados disseram que, já se foi o tempo que em apenas mulheres se preocupavam com a aparência, tendo 54% deles afirmando que frequentam barbearias e salões de beleza com frequência.

Com isso vimos que os homens estão cada vez mais homens seguem para esse caminho e isso nos leva para as BarberShops.

Sendo a principal inovação no ramo, as BarberShops são lugares especializados na figura masculina, e dedicada não só aos cuidados físicos, mas também ao lazer e boas práticas. Contando com diversos estilos e métodos diferentes, as Barbershops possuem jogos, venda de bebidas e espaços dedicados ao lazer e conversas, além é claro do atendimento. Como também dito por (Terra, 2017):

Os novos estabelecimentos oferecem um serviço que atende às demandas do homem sofisticado que busca serviços personalizados e exclusivos e que conta com um ambiente reservado, elegante dentro de um espaço de multisserviços relacionados à saúde, beleza e bem-estar.

Então com as BarberShops, e os novos hábitos masculinos o mercado disparou tendo um lucro exponencial a cada ano, como mencionado por (Estadão, 2017):

De acordo com a Euromonitor Internacional, de 2011 a 2016, o faturamento do setor cresceu 94,4%, passando de R\$ 10,07 bilhões para R\$ 19,6 bilhões. No mercado de perfumaria, por exemplo, as vendas aumentaram de R\$ 5,1 bilhões para R\$ 11,9 bilhões. Em desodorantes, foram de R\$ 2,8 bilhões para R\$ 4,9 bilhões. Em segmentos mais novos, como o de produtos de banho exclusivos para homens, as vendas partiram de R\$ 11,9 milhões em 2011 e alcançaram R\$ 261 milhões no ano passado.

Porém com a pandemia em 2020, as BarberShops foram obrigadas a baixar suas portas e mesmo com a redução de casos e normalização de frequências, um pequeno problema surgiu, a comunicação entre os profissionais e os clientes. Hoje em dia normalmente esse contato é feito através do número pessoal ou profissional do barbeiro. Fazendo com que o trabalho para ele seja bem grande, pois além de gerenciar os clientes ele também precisa sozinho gerenciar sua agenda.

Barbearias com um maior porte, possuem apps próprios para resolver tal problema, ou apps adquiridos como o Fresha, que como relatado por uma cliente em sua página (Pamela, 2023):

Ela oferece aos meus clientes uma experiência profissional de agendamento, com recursos de marcação eficazes, lembretes automáticos por e-mail e SMS, recursos avançados de vendas e as melhores taxas de processamento de pagamento para a minha empresa.

Porém mesmo com esses aplicativos próprios de cada estabelecimento, os clientes ainda não conseguem em apenas um local encontrar sua variedade, e o mesmo vale para empresário, que depende ainda mais do marketing em outros meios para atrair a clientela.

1.1 OBJETIVO

O objetivo do devido trabalho é o desenvolvimento de uma plataforma online onde os profissionais (barbeiros) poderão cadastrar seus estabelecimentos e definir os serviços a serem prestados juntamente dos valores, os clientes através do acesso a mesma, poderão marcar seus horários, fazendo com que tudo se torne mais simples e prático. E para o barbeiro mais organizado e sem preocupações.

Tendo funcionalidades de confirmar e cancelar o agendamento, definir horários de atendimento e serviços a serem prestados, com adicionais de valores dependendo da escolha do cliente.

1.2 JUSTIFICATIVA

Com a vinda da pandemia nos anos de 2020-2021, muitos lugares entraram em estado de alerta e defesa (lockdown), forçando assim os lugares a baixarem as portas por tempo

indeterminado. Com tudo, quando o número de casos começou a baixar, os estabelecimentos tiveram a permissão de retomar seus afazeres, porém com limite de pessoas dentro dele.

Devido a esse fato, a maioria das Barbershops começaram a atender com horário marcado, sendo esses agendamentos realizados por diversos meios, redes sociais, telefone pessoal ou profissional do barbeiro, ou em casos de grandes estabelecimentos aplicativos próprios. Com tudo, o cliente acaba tendo um pequeno trabalho, pois caso ele não tenha acesso ao número de telefone, ou rede social do profissional, isso acaba impossibilitando o cliente de fazer seu agendamento.

Vendo isso, surgiu a ideia da BarberWeb uma plataforma online que será uma forma de facilitar e diminuir a problemática apresentada, tendo suas funcionalidades e praticidades para os barbeiros, a plataforma pode abrigar diversas empresas, deixando assim o acesso para o cliente bem mais rápido e fácil, além de apresentar uma devida variedade para que ele possa ter uma melhor escolha de onde depositar sua confiança e dinheiro.

1.3 MOTIVAÇÃO

A motivação para o desenvolvimento do devido trabalho, surgiu após analisar ambos os lados vivendo algumas dificuldades causadas pelos problemas de comunicação entre clientes e seus barbeiros. Sendo assim enxergando que havia a necessidade de um meio para facilitar esse processo e também o agilizar, tanto para o cliente que precisa do seu horário, quanto para o profissional que necessita de um auxílio para organizar seus dias, e atendimentos.

1.4 PERSPECTIVA E DISTRIBUIÇÃO

Com esse trabalho pretendesse mostrar como a tecnologia e desenvolvimento web pode gerar facilidades em diversas áreas de atuação. Além é claro de realizar a tentativa de mostrar um caminho que poderá ser aperfeiçoado e alimentado, para que se torne um meio totalmente viável de escolhas para profissionais de pequeno, médio e grande porte, além de mostrar para o cliente que existem opções e diferentes formas de ser atendido.

1.5 METODOLOGIA

Para o desenvolvimento deste trabalho, juntei opiniões e desejos de diversos profissionais e de seus clientes, para ter um entendimento mais abrangente do cenário em que pretendo desenvolver, com isso obtive uma base bem elaborada para o que será necessário de implementar e construir. Junto a parte de engenharia de software, os diagramas serão desenvolvidos através do software especializado, Astah, tendo base na linguagem UML. Me permitindo assim realizar o desenvolvimento dos mesmo com mais agilidade e praticidade.

Para a implementação, foi utilizado uma base em Orientação Objetos, fazendo com que o código se torne melhor e mais otimizado. Usando Express como framework com base em Javascript, teremos uma maior liberdade para criar, já para o banco de dados foi usado MongoDB que é um serviço gratuito e que através do conhecimento adquirido na grade escolar, já nos trouxe uma base para o início dos estudos aprofundados. Por fim, para a codificação e escrita, foi usado o Visual Studio Code.

1.6 PÚBLICO-ALVO

A plataforma desenvolvida nesse projeto, tem como público-alvo, homens que se preocupam com sua aparência e cuidados físicos, frequentadores de Barbershops e outros estabelecimentos do tipo. Também homens que se encontram em dúvida dos melhores locais na sua região, homens que enfrentam dificuldades em encontrar um barbeiro de confiança ou até mesmo de agendar seus horários.

Também buscando os profissionais, que querem atrair uma clientela mais variada para seu próprio estabelecimento, ou que deseja uma forma organizada de trabalhar e organizar seus agendamentos.

1.7 ESTRUTURA DO TRABALHO

O devido trabalho está dividido em cinco capítulos que detalham a ideia e o desenvolvimento do projeto, sendo eles: Capítulo 1 - Introdução; Capítulo 2 – Conceitos e Tecnologias Usadas no Projeto; Capítulo 3 – Proposta de Modelagem de Casos de Uso, Capítulo 4 - Implementação e Capítulo 5 - Conclusão.

O Capítulo 1 - Introdução, está dividido em tópicos que mostram um contexto geral da ideia e mercado onde o projeto está localizado, também dando ênfase em público-alvo e citando as tecnologias. Por conta disso o capítulo se encontra estruturado da seguinte forma: Introdução, objetivo, justificativa, motivação, perspectiva de contribuições, metodologia, público-alvo, estrutura do trabalho.

O Capítulo 2 – Conceitos e Tecnologias Usadas no Projeto, também se dividi em tópicos que mostram um pouco de cada tecnologia que foi usada nesse projeto, desde as linguagens até mesmo os softwares. Por conta disso se encontra estruturado da seguinte forma: UML, Astah, Javascript, Visual Studio Code, NodeJs, Express, MongoDB, HTML e CSS.

O Capítulo 3 – Proposta e Modelagem de Casos de Uso, está dividido em tópicos destinados a cada tipo de diagrama essencial no desenvolvimento desse projeto, mostrando desde os caminhos a seguirem, até mesmo diagramas para explicar suas funcionalidades. Por conta disso o capítulo se encontra estruturado da seguinte forma: Mapa mental, DER, diagrama de casos de uso, diagrama de atividades e cronograma.

O Capítulo 4 - Implementação, está dividido em tópicos destinados a descrições sobre a implementação como. objetivos, metodologia usada para o desenvolvimento, dificuldades encontradas no processo, imagens do sistema e codificação.

O Capítulo 5 - Conclusão, traz a conclusão do projeto, com uma visão sobre todo o processo experienciado e planejado para tal.

2. CONCEITOS E TECNOLOGIAS USADOS NO PROJETO

Este capítulo trata de todas as tecnologias que foram usadas no desenvolvimento do projeto, algumas como a linguagem UML e Astah, tivemos a oportunidade de ver com profundidade dentro da grade escolar apresentada no curso de Análise e Desenvolvimento de Sistemas, porém outras como o NextJs e React já não estão integradas a mesma.,

2.1 UML

Para o desenvolvimento e projeção dos diagramas do projeto, foi usada a linguagem UML – Unified Modeling Language, que é uma linguagem usada para desenhar e estruturar os diagramas que fazem parte do projeto. Para Noieto (2020), “Trata-se de uma linguagem de modelagem única, cujo papel é auxiliar a equipe de desenvolvimento a visualizar os diversos aspectos da aplicação, facilitando a compreensão do seu funcionamento.”.

2.2 ASTAH

Com o intuito de encontrar um software que permita que seja aplicar todo o conhecimento adquirido, e que permita gerar os diagramas, foi escolhido o Astah que é um software específico para o desenvolvimento e projeções do mesmo, onde além de termos uma boa ferramenta, também possuímos uma boa vivencia com o software através da grade estudantil. De acordo com Lombardi (2021):

Na área de Engenharia de Software, a UML (Linguagem de Modelagem Unificada) é uma linguagem de modelagem que permite representar um sistema de forma padronizada. Astah utilizada nos diagramas dinâmicos, essa ferramenta já é bastante consolidada, voltada para a modelagem de sistemas utilizando a UML

2.3 JAVASCRIPT

Para linguagem de programação, foi usado Javascript que é uma linguagem de alto nível focada em desenvolvimento web, servindo tanto para o front-end quanto para o back-end. Para Roveda (2021):

Javascript é uma linguagem de programação de alto nível criada, a princípio, para ser executada em navegadores e manipular comportamentos de páginas web.”. Com isso, achamos que foi uma boa escolha para o desenvolvimento do projeto, pois nos trará maiores facilidades e vantagens para o desenvolvimento.

Hoje em dia Javascript se encontra em uma posição muito alta no mercado, sendo bem procurada para o seu foco o desenvolvimento Web. Porém desde popularização o Javascript é um marco e uma referência para todos os Devs que buscam o desenvolvimento Web. Como dito por Flanagan (2013, p18):

JavaScript é a linguagem de programação da Web. A ampla maioria dos sites modernos usa JavaScript e todos os navegadores modernos – em computadores de mesa, consoles de jogos, tablets e smartphones – incluem interpretadores JavaScript, tornando-a a linguagem de programação mais onipresente da história. JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo de páginas Web; CSS, para especificar a apresentação dessas páginas; e JavaScript, para especificar o comportamento delas.

Assim sendo, o uso de uma tecnologia como o Javascript, é vantajoso para o projeto em diversos setores, busca de mão de obra, acesso a conteúdos para estudo, entre outras.

2.4 VISUAL STUDIO CODE

O VS Code, é um leitor de código aberto desenvolvido pela Microsoft, que está disponível em diversas plataformas. Tem algumas vantagens como sua alta compatibilidade e customizações, além é claro dos seus pacotes baixáveis. Para Hanashiro (2021):

O VS Code é capaz de fazer muitas coisas que nos ajudam a trabalhar melhor. A princípio ele é uma ferramenta muito simples, mas ele possui uma loja de extensões imensa, e que continua crescendo. Ou seja, com essa enorme coleção de extensões, podemos adicionar diversas funcionalidades ao VS Code de forma

bem simples. Dessa maneira, ele pode facilmente ser comparado a grandes IDEs pagas. Qualquer um pode criar uma extensão e publicar na loja. Desse modo, sempre há novas ferramentas que podem ser interessantes para alguém. Já fizemos aqui no blog uma série de posts indicando as melhores extensões de VS Code para desenvolvedores Front-End.

2.5 NODEJS

NodeJS é uma tecnologia que permite que Javascript seja utilizada pro back-end, onde o mesmo converte o código para a linguagem do computador então possibilitando o entendimento do mesmo. Como citado por Oliveira (2022, p06):

O Node.js é um ambiente de servidor gratuito e de código aberto, que possibilita a criação de aplicações no lado servidor (server-side) com algumas vantagens em relação aos seus principais concorrentes (como PHP ou Java, por exemplo), no que se refere a leveza, flexibilidade, suporte e produtividade. Além disso, está alinhado com o que há de mais moderno no desenvolvimento de software para Internet, como o uso de um padrão de arquitetura de entrada e saída assíncrona e conexão com banco de dados não relacionais (como no MongoDB).

Com o crescimento da qualidade da ferramenta, diversas empresas têm feito uso da mesma, o que também permite que o conhecimento em tal tecnologia seja uma boa opção para portfólio. Como também citado por (Bassa,2016), “Por fornecer uma boa **performance**, várias plataformas têm adotado o Node.js como uma solução viável e eficaz de tecnologia para Back-end, como o próprio Google, Netflix., etc.”

2.6 EXPRESS

O Express é um framework com base em Javascript e NodeJS que permite uma gama de facilidades para a criação de sistemas, tanto para o back-end, quanto para tomadas full-stack, como dito por (Andrade, 2021):

O Express.js é um Framework rápido e um dos mais utilizados em conjunto com o Node.js, facilitando no desenvolvimento de aplicações back-end e até, em conjunto com sistemas de templates, aplicações full-stack. Escrito em JavaScript, o Express.js é utilizado por diversas empresas ao redor do mundo, dentre elas a Fox Sports, PayPal, IBM, Uber, entre outras. Muito popular tanto em grandes empresas quanto na comunidade, o Express facilita a criação de aplicações utilizando o Node em conjunto com o JavaScript, tornando este ecossistema ainda mais poderoso.

Levando em consideração a compatibilidade com outras ferramentas já citadas para a implementação do projeto, o Express se tornou uma das escolhas mais viáveis para uso, em uma imensa gama de tecnologias disponíveis.

2.7 MONGODB

MongoDB é um sistema de Banco de Dados, que vem crescendo muito ultimamente, por ser gratuito, fácil e de bom desempenho o Mongo se tornou uma ótima opção para diversas empresas e alunos. Uma forte vantagem do Mongo é sua flexibilidade e adaptabilidade, como digo pela MongoDB (2023):

MongoDB armazena dados em documentos do tipo JSON flexíveis, o que significa que os campos podem variar de documento para documento e a estrutura de dados pode ser alterada ao longo do tempo". Com isso podemos ter uma variedade maior no arsenal de tecnologias escolhidas para o desenvolvimento, além das vantagens de ter acesso gratuito e já termos uma base de estudo relacionado a Mongo, em certas matérias da grade estudantil.

2.8 HTML

HTML é uma linguagem de marcação para front-end, onde o computador entende o layout que deve seguir através de tags que são representadas pelos sinais de <>, sempre atrelado a diversas outras ferramentas para estilizações e funções o HTML é uma linguagem muito utilizada. Como dito por (Mozilla, 2023):

HTML (Linguagem de Marcação de HiperTexto) é o bloco de construção mais básico da web. Define o significado e a estrutura do conteúdo da web. Outras tecnologias além do HTML geralmente são usadas para descrever a aparência/apresentação (CSS) ou a funcionalidade/comportamento (JavaScript) de uma página da web.

2.9 CSS

O css é uma outra ferramenta, conhecida como folha de estilo, onde a mesma é acoplada ao HTML para estilizar todas suas tags e função. Utilizando de classes e metodos especificos o CSS permite a criação de diversas estilizações diferentes, gifs, animações, logos, etc. Para (Mozilla, 2023):

O CSS pode ser usado para estilizar um documento muito básico de texto — por exemplo, alterando a cor e tamanho dos títulos e links. Pode ser usado para criar layout — por exemplo, transformando uma simples coluna de texto em um layout (en-US) com uma área de conteúdo principal e um sidebar (barra lateral) para as informações relacionadas. Pode até ser usado para efeitos tais como animação.

Com isso, seu uso seria indispensável para a implementação. Combinando assim para o front-end HTML, CSS e Javascript.

3. PROPOSTA E MODELAGEM DE CASOS DE USO

Com o intuito de facilitar o desenvolvimento do software, essa seção usará de ensinamentos passados através da matéria de Engenharia de Software. Trazendo com clareza, mapa mental, DER, Diagrama de casos de uso, Diagrama de Classe, e Diagrama de sequência. Utilizando a linguagem UML e Astah como software de apoio.

Também para uma melhor compreensão, será explicado de forma mais clara e detalhada cada fase do sistema, além de suas funcionalidades. A BarberWeb, é uma plataforma online, onde empresários do ramo de barbearias podem cadastrar seus estabelecimentos, trazendo assim visibilidade, e facilidade dentro do quesito agendamentos.

3.1 MAPA MENTAL

Para melhorar o entendimento sobre o projeto se suas funções, construímos um mapa mental, que de acordo com a Estácio (2021), “O mapa mental pode ser compreendido como uma ferramenta que ajuda a gerir informações. Ou seja, é uma maneira de simplificar a compreensão de um tema, relacionar conteúdos, revisar dados ou mesmo registrar ideias”. Usar esse método traz diversas vantagens, como melhor visão do geral de um projeto, uma melhor compreensão de cada funcionalidade, e claro uma forma simplificada de apresentá-lo.

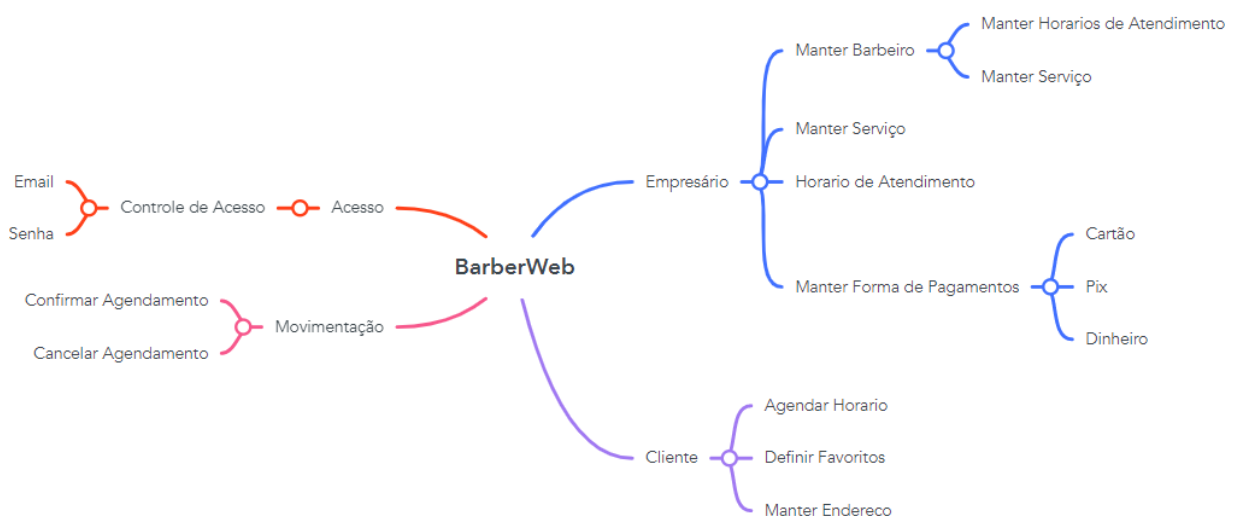


imagem 1 – Mapa Mental.

3.2 Diagrama de Entidade e Relacionamentos (DER)

Um DER ou Diagrama de Entidade e Relacionamento, é a forma visual de representar a composição de um banco de dados. Sendo assim ele serve para que possamos ter uma ideia e uma visão geral do que será o banco, com suas classes ou entidades, e relacionamentos modelados, e testados através da lógica. De acordo com Carvalho (2023), o DER é uma forma mais intuitiva e visual de se representar um banco de dados. Por base nesses fatores, foi desenvolvido o seguinte diagrama.

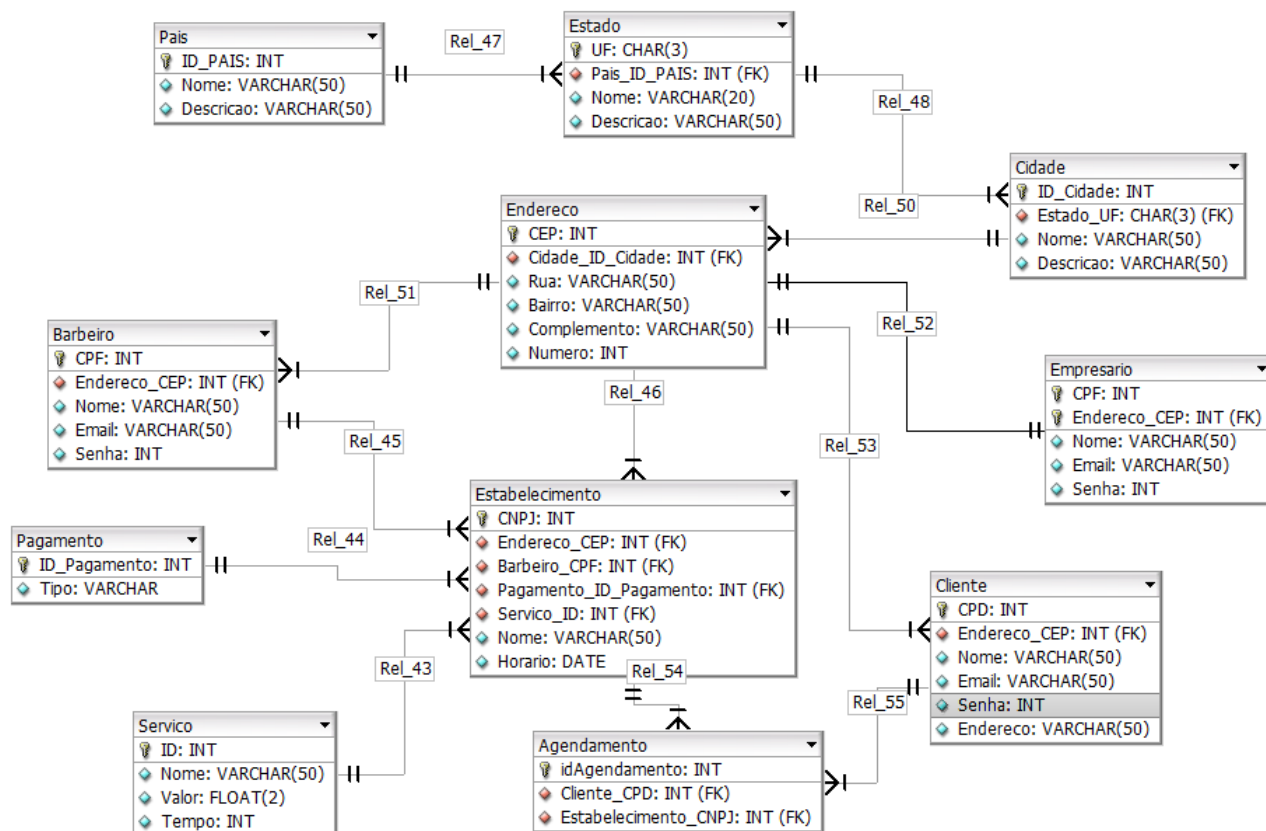


Imagem 2 – Diagrama de Entidade e Relacionamento

3.3 DIAGRAMA DE CASO DE USO

Um diagrama de caso de uso, são diagramas que mostram como é o fluxo de uma função do sistema, como por exemplo um cadastro, ou agendamento. Nesses casos o diagrama de casos de uso mapearia o fluxo da ação, do seu início até o fim. Como dito por (Licidchart, 2023):

Na Linguagem de modelagem unificada (UML), o diagrama de caso de uso resume os detalhes dos usuários do seu sistema (também conhecidos como atores) e as interações deles com o sistema. Para criar um, use um conjunto de símbolos e conectores especializados. Um bom diagrama de caso de uso ajuda sua equipe a representar e discutir:

A figura 3 ilustra o Diagrama de Casos de Uso geral.

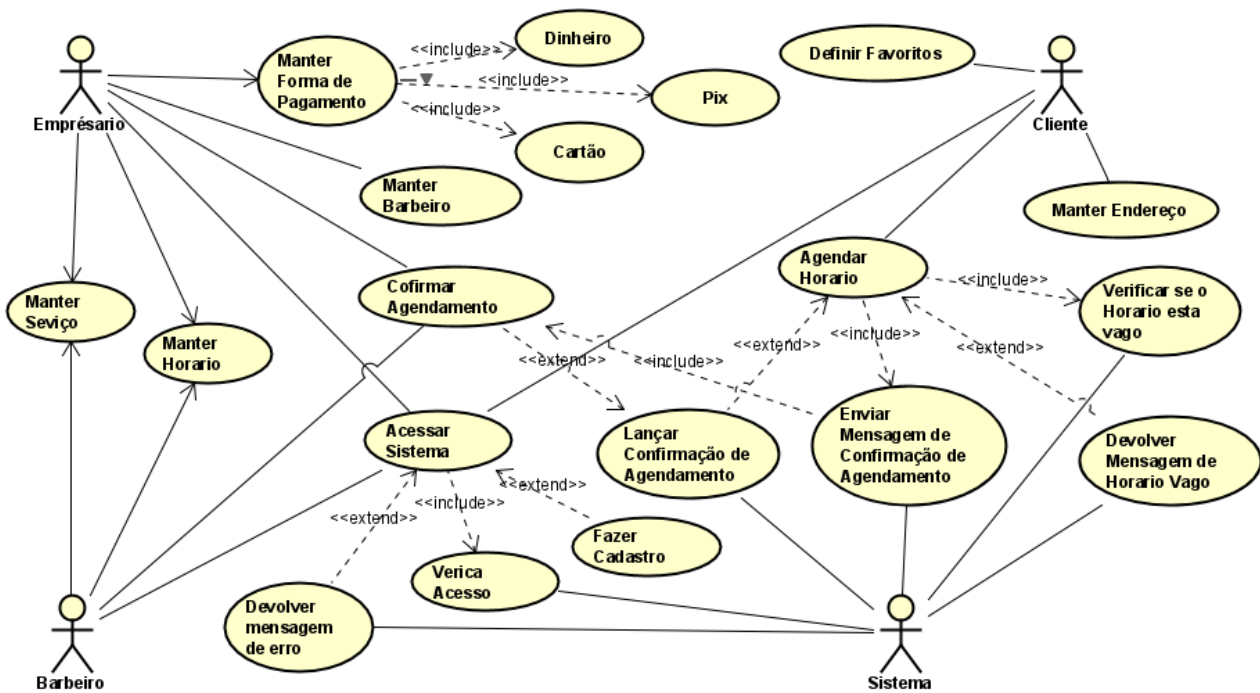


Imagem 3 – Diagrama de Caso de Uso Geral

3.3.1 UC01 – ACESSAR SISTEMA

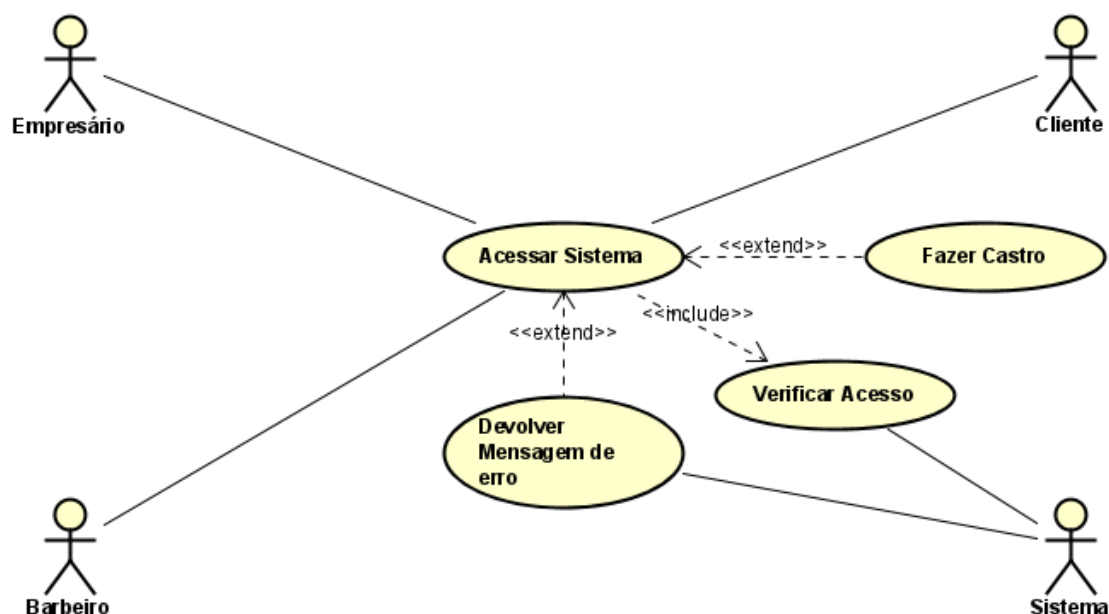


Imagem 4 – UC de acesso ao sistema

FINALIDADE/OBJETIVO	Acessar a plataforma, ter acesso a todas as funcionalidades que o perfil do usuário tem direito.
ATORES	Cliente, Empresário e Barbeiro.
EVENTO INICIAL	O usuário precisa acessar o a URL da plataforma
FLUXO PRINCIPAL	<p>a) O caso de uso é iniciado quando o ator acessa a plataforma BarberWeb através do seu navegador, clicando logo em seguida em “Fazer Login”.</p> <p>b) O ator digita seus dados nos campos obrigatórios, para que possa ser feita a leitura.</p> <p>c) O ator clica no botão “Login”, para indicar que acabou de digitar seus dados.</p> <p>d) O sistema verifica seus dados, e dá acesso ao ator, mostrando a mensagem “Login efetuado com sucesso”.</p> <p>e) O caso de uso é encerrado</p>
FLUXO ALTERNATIVO	<p>a) O caso de uso é iniciado quando o ator acessa a plataforma BarberWeb através do seu navegador, clicando logo em seguida em “Fazer Login”.</p>

	<p>b) O ator digita seus dados nos campos obrigatórios, para que possa ser feita a leitura.</p> <p>c) O ator clica no botão “Login”, para indicar que acabou de digitar seus dados.</p> <p>d) O sistema verifica seus dados, e dá acesso ao ator, mostrando a mensagem “Usuário ou senha errados”.</p> <p>e) O ator é levado a tela onde tem acesso a botões “recuperar senha” e “Cadastre-se”</p> <p>f) O caso de uso é encerrado.</p>
FLUXO DE EXCEÇÃO	<p>a) O caso de uso se inicia com atores que marcaram a opção “Manter-se Conectado”, levando assim que após o acesso ao URL da plataforma, ele já estará conectado.</p> <p>b) O ator clica no botão “Desconectar”.</p> <p>c) O sistema o leva para a tela de login novamente, onde ele pode digitar seus dados.</p> <p>d) O caso de uso se encerra.</p>
TESTES	<p>a) O sistema faz a chamada de verificação do e-mail e senha do ator.</p> <p>b) O banco devolve a chamada com a confirmação ou barramento do usuário.</p> <p>c) O sistema confirma a mensagem com o ator.</p>

Tabela 1 – Narrativa de caso de uso – Acessar Sistema

3.3.2 UC02 – MANTER BARBEIRO

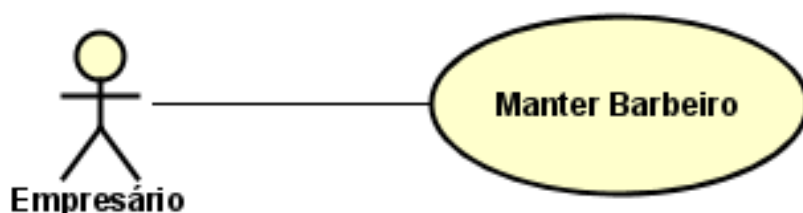


Imagem 5 – UC de Manter Barbeiro

FINALIDADE/OBJETIVO	Cadastrar, excluir e editar, Barbeiros, do estabelecimento.
ATORES	Empresário
EVENTO INICIAL	O ator deve acessar a aba “Barbeiros” dentro do perfil do estabelecimento.
FLUXO PRINCIPAL	<p>a) O caso de uso se inicia quando o ator clica no botão “Cadastrar Barbeiro”, onde ele é levado para a tela de cadastro.</p> <p>b) O ator digita as informações necessárias para o cadastro em seus devidos campos.</p> <p>c) Então clicando em “Confirmar”, o ator encerra o cadastro.</p> <p>d) O caso de uso se encerra.</p>
FLUXO ALTERNATIVO	<p>a) O caso de uso se inicia quando o ator clica no botão “Excluir Barbeiro”, onde ele é levado a uma tela de exclusão.</p> <p>b) O ator digita as informações necessárias para a exclusão em seus devidos campos.</p> <p>c) Então clicando em Excluir, o barbeiro é excluído do estabelecimento.</p> <p>d) O caso de uso se encerra.</p>
FLUXO DE EXCEÇÃO	<p>E1 – CADASTRO DE BARBEIROS</p> <p>a) O caso de uso se inicia quando o ator clica no botão “Cadastrar Barbeiro”, onde ele é levado para a tela de</p>

	<p>cadastro.</p> <p>b) O ator digita as informações necessárias para o cadastro em seus devidos campos.</p> <p>c) Então clicando em “Confirmar”, o ator encerra o cadastro.</p> <p>d) O sistema informa que o barbeiro já está cadastrado em outro estabelecimento.</p> <p>e) O caso de uso se encerra.</p> <p>E2 - EXCLUSÃO DE BARBEIROS</p> <p>a) O caso de uso se inicia quando o ator clica no botão “Excluir Barbeiro”, onde ele é levado a uma tela de exclusão.</p> <p>b) O ator digita as informações necessárias para a exclusão em seus devidos campos.</p> <p>c) Então clicando em Excluir, o barbeiro é excluído do estabelecimento.</p> <p>d) O sistema informa que o barbeiro não está cadastrado no estabelecimento, ou não existe.</p> <p>d) O caso de uso se encerra</p>
--	---

Tabela 2 – Narrativa de caso de uso – Manter Barbeiro

3.3.3 UC03 – MANTER SERVIÇO



Imagem 6 – UC de Manter Serviço

FINALIDADE/OBJETIVO	Cadastrar, Editar e Excluir serviços que podem ser prestados pelo estabelecimento e barbeiro.
ATORES	Empresário e Barbeiro
EVENTO INICIAL	Ator acessa o perfil do estabelecimento ou o perfil do barbeiro, dentro da plataforma.
FLUXO PRINCIPAL	<p>a) O caso de uso se inicia quando o ator clica no botão “Serviços”, assim sendo levado pra tela de serviços.</p> <p>b) Clicando no botão “Cadastrar”, a página de cadastro de serviços é apresentada.</p> <p>c) O ator preenche os dados obrigatórios para um serviço prestado.</p> <p>d) O ator clica no botão “Confirmar Serviço”.</p> <p>e) O sistema analisa os dados, e devolve a mensagem “Serviço Cadastrado”.</p> <p>f) O caso de uso é encerrado.</p>
FLUXO ALTERNATIVO	<p>A1 – EDITAR SERVIÇO</p> <p>a) O caso de uso começa quando o ator clica em editar serviço na tela de serviços.</p> <p>b) Após selecionar o serviço a ser editado, a tela de edição é apresentada.</p> <p>c) O ator faz as modificações necessárias.</p> <p>d) Ao clicar no botão “Encerrar”, a edição é finalizada.</p> <p>e) O caso de uso é encerrado.</p> <p>A2 – EXCLUIR SERVIÇO</p> <p>a) O caso de uso começa quando o ator clica em excluir serviço na tela de serviços.</p> <p>b) Após selecionar o serviço a ser excluído, a tela de exclusão é apresentada.</p> <p>c) O ator preenche os campos necessários para a exclusão.</p>

	<p>d) Ao clicar no botão “Excluir”, a exclusão é feita</p> <p>e) O caso de uso é encerrado.</p>
FLUXO DE EXCEÇÃO	<p>E1 - JÁ EXISTENTE</p> <p>a) O serviço a ser cadastrado, já se encontra no perfil do estabelecimento ou barbeiro.</p> <p>b) Sistema informa que já está cadastrado, e pergunta se o ator quer editar o mesmo.</p> <p>c) Ator é levado a tela de edição [A1].</p> <p>d) O caso de uso se encerra.</p> <p>E2 – CAMPOS OBRIGATÓRIOS</p> <p>a) O ator não preenche de forma correta os campos necessários para a inclusão, edição ou exclusão de um serviço.</p> <p>b) Sistema informa o erro, e pede para que preencha novamente.</p> <p>c) Ator é levado novamente pra tela em que se encontrava.</p> <p>d) O caso de uso se encerra.</p>
TESTES	<p>T1 – CAMPOS OBRIGATORIOS</p> <p>a) O sistema avalia se os campos obrigatórios estão preenchidos corretamente.</p> <p>b) Após avaliação o mesmo devolve uma mensagem de confirmação ou barramento.</p> <p>T2 - SERVIÇO JÁ EXISTENTE</p> <p>a) O sistema faz uma chamada ao banco, para averiguar se o serviço já se encontra existente.</p> <p>b) O banco devolve a chamada, com a confirmação ou barramento.</p> <p>c) O sistema envia ao ator a mensagem, confirmando ou redirecionando ele para a tela de edição [A1].</p>

Tabela 3 – Narrativa de caso de uso – Manter Serviço

3.3.4 UC04 – AGENDAR HORARIO.

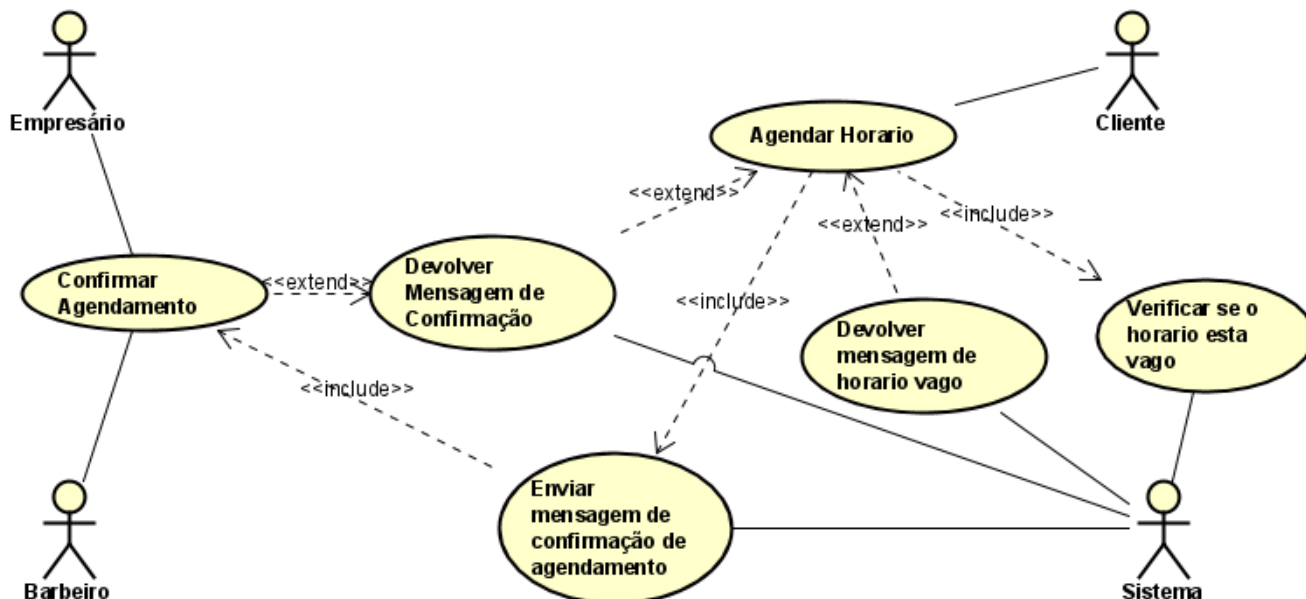


Imagem 7 – UC de agendar Horários.

FINALIDADE/OBJETIVO	Agendamento de Horários
ATORES	Cliente, Empresário e Barbeiro.
EVENTO INICIAL	Cliente encontra a barbearia deseja na plataforma, e escolhe o barbeiro.
FLUXO PRINCIPAL	<p>a) O cliente escolhe um serviço pré-cadastrado pelo barbeiro ou empresário.</p> <p>b) Após a seleção de serviço, a tela de horários é apresentada, já adequada ao serviço escolhido.</p> <p>c) O cliente escolhe um horário.</p> <p>d) O sistema faz a verificação, para saber se o horário já está vago.</p> <p>e) Com a confirmação do horário, o sistema envia uma mensagem é enviada ao empresário ou barbeiro, para que ele confirme o agendamento.</p> <p>d) Após confirmado, a mensagem é exibida “Agendamento Confirmado”.</p> <p>e) O caso de uso se encerra.</p>
FLUXO DE EXCEÇÃO	<p>E1 – SEM HORARIOS VAGOS</p> <p>a) O sistema não encontra nem um horário vago, mandando o cliente para outro dia.</p> <p>b) O caso de uso é encerrado.</p>

	<p>E2 - NÃO CONFIRMAÇÃO</p> <p>a) O barbeiro ou Empresário, não confirmam o agendamento.</p> <p>b) Sistema devolve a mensagem “Agendamento Cancelado, escolha outro horário”.</p> <p>c) Cliente volta para tela de horários.</p> <p>d) O caso de uso é encerrado.</p>
<p>TESTES</p>	<p>T1 - HORÁRIOS.</p> <p>a) O sistema faz a chamada ao banco, para averiguar se o horário está vago.</p> <p>b) Banco devolve a mensagem de confirmação ou barramento.</p> <p>c) Sistema informa ao cliente, sobre a situação do horário.</p> <p>T2 - CONFIRMAÇÕES</p> <p>a) O sistema envia a mensagem de agendamento para o barbeiro ou empresário.</p> <p>b) O sistema devolve a mensagem de confirmação ou barramento de agendamento.</p> <p>c) O sistema redireciona o cliente para tela de reagendamento.</p>

Tabela 4 – Narrativa de caso de uso – Agendamento de Horário

3.4 DIAGRAMAS DE ATIVIDADES

Os diagramas de atividades são um conjunto de formas, setas e linhas, que simbolizam o decorrer de um processo realizado pelo sistema ou ator. Como um mapa mostrando cada passo que é preciso ser realizado do começo ao fim de uma atividade realizada dentro do Software. De acordo com a IBM (2021):

Um diagrama de atividade fornece uma visualização do comportamento de um sistema descrevendo a sequência de ações em um processo. Os diagramas de atividades são semelhantes a fluxogramas porque mostram o fluxo entre as ações em uma atividade; no entanto, os diagramas de atividades também podem mostrar fluxos paralelos ou simultâneos e fluxos alternativos.

Com isso desenvolvemos um diagrama de atividades, para representar o processo de agendamento, realizado pelo cliente, perante uma lista de serviços e horários já pré-definidos pelo barbeiro ou Empresário. O diagrama de atividades está representado pela imagem 8.

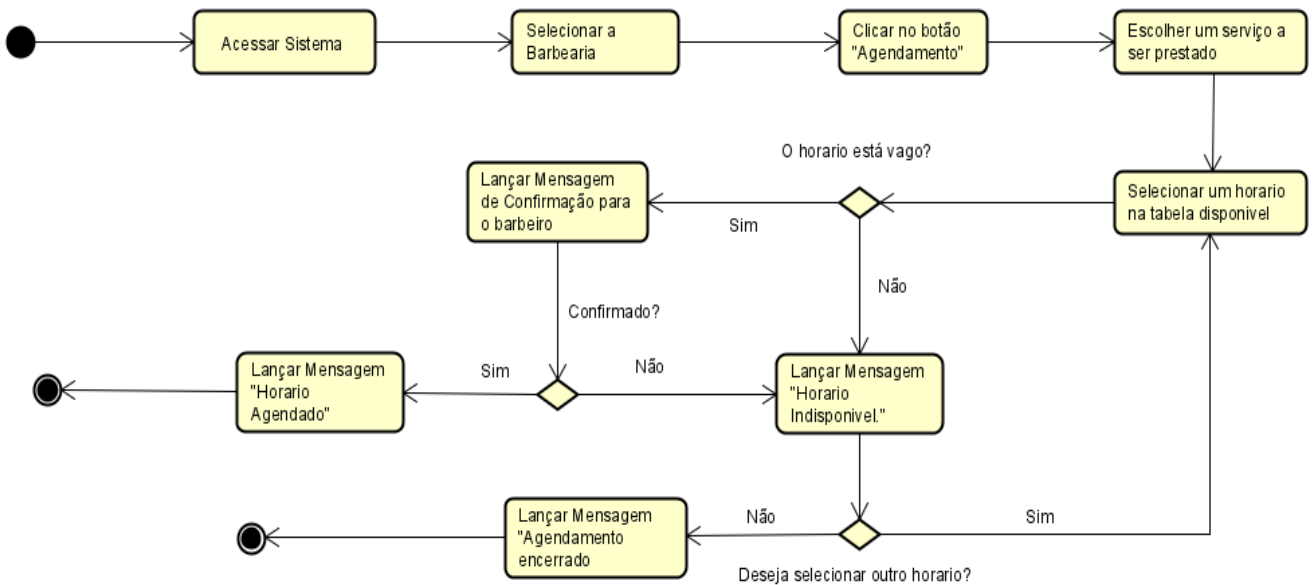


Imagem 8 – Diagrama de Atividades de Agendamento

4. IMPLEMENTAÇÃO

Nesse capítulo vamos destacar todas as fases do desenvolvimento da BarberWeb – Plataforma Online de Agendamentos. Destacando objetivos e metodologias usadas, dificuldades encontradas, suas principais funcionalidades e mostrando telas ilustrativas para maior compreensão do sistema. Vale ressaltar que o mesmo ainda está em fase de implementação, por conta disso o objetivo é passar uma amostra do sistema em si.

4.1 Objetivos da implementação

O objetivo é a implementação de uma plataforma online, para agendamentos em barbearias e barbershops, facilitando assim a vida do cliente que vai ter seu horário marcado com mais eficiência, além de já conhecer exatamente o valor a ser pago e também o serviço a ser prestado. E para o empresário que já saberá antecipadamente qual cliente virá, e o que deverá ser feito, além é claro de trazer maior visibilidade ao estabelecimento.

4.2 Metodologia Abordada

A metodologia abordada para o desenvolvimento e implementação foi seguida da seguinte forma, uma abordagem dinâmica entre estudos e desenvolvimento onde diversos cursos e treinamentos foram realizados e aplicado simultaneamente ao projeto. Então um esboço era feito ao adquirir um seguinte conhecimento e logo em seguida aplicado ao projeto.

Para que tivéssemos 100% de aproveitamento em todas as funcionalidades desenvolvidas, sempre que algo novo era implementado, uma pequena reunião com um barbeiro local era realizada, para que o mesmo explicasse, se aquilo era realmente útil para seu negócio, e se estava implementada de forma com que facilitasse o mesmo.

4.3 Dificuldades Encontradas

Diversas dificuldades foram encontradas durante a fase de implementação, porém a que mais trouxe problemas para o projeto, foi a falta de conhecimento através do desenvolvedor. A plataforma Alura trouxe um auxílio enorme onde diversos cursos estão disponíveis em sua plataforma, porém o aprendizado exige tempo, o que acabou atrasando algumas etapas no desenvolvimento.

Outra dificuldade grande, foi a falta de um hardware descente para o escopo do projeto, onde o disponível acabou sendo inadequado para o projeto.

4.4 Principais Funcionalidades

Durante todo o processo de desenvolvimento e implementação, o foco real foram as principais funcionalidades que uma barbearia precisaria conter para que já fosse possíveis os primeiros agendamentos serem realizado.

- Agendamentos: Foi desenvolvida uma tela para que o cliente tenha acesso aos serviços já definidos através do front-end, conseguindo assim selecioná-los e ter uma previa do valor a ser pago, seguindo para a escolha da hora onde o mesmo pode realizar seu agendamento.
- Telas Intuitivas: Tendo como intuito deixar tudo o mais simples possível para que ambos os lados conseguissem entender exatamente a proposta do sistema e suas telas, buscamos um design simples e focado nas informações necessárias para cada funcionalidade.

4.5 TELAS ILUSTRATIVAS

A seguir podemos ver exemplos de algumas telas que compõem o sistema. Buscamos utilizar uma paleta de cores que se lembra barbearias dos anos 80, onde sempre exibiam seus Poles de Barbeiros nas cores brancas, vermelhas e azuis. Junto das fontes, que também remetem a barbearias.

4.5.1 LOGIN

Os layouts de Login estão selecionados em duas etapas, onde no primeiro acesso a pessoa tem que escolher a forma de se conectar, sendo cliente ou barbeiro. Segue imagem ilustrativa do layout inicial:



Imagem 9 – Layout Inicial de Login

Após o usuário escolher a sua forma de login, o usuário terá acesso a outro Layout para que consiga estar digitando suas credenciais e fazer o acesso ao sistema. Segue imagem ilustrativa do Layout de Login:



Imagem 10 – Layout de Login para Clientes.

4.5.2 INDEX

O layout do menu principal não difere entre o acesso de cliente ou barbeiro, contando com logotipo, menu, e conteúdos sobre a barbearia, o Index traz um estilo despojado e

visando remeter ao estilo de barbearias dos anos 80 como já citado anteriormente. Segue imagem ilustrativa do layout do Index.



Imagem 11 – Layout do Index

4.5.3 SERVIÇOS E HORÁRIOS

O layout de serviços e horários são idênticos, para que o estilo se mantenha conciso perante o restante do sistema. Ambos os layouts visam mostrar ao cliente, os serviços e Horários em que a barbearia trabalha. Os serviços inicialmente foram definidos diretamente no front-end, porém futuramente poderão ser cadastradas pelo barbeiro. Seguem imagens ilustrativas dos layouts de Serviços e Horarios:

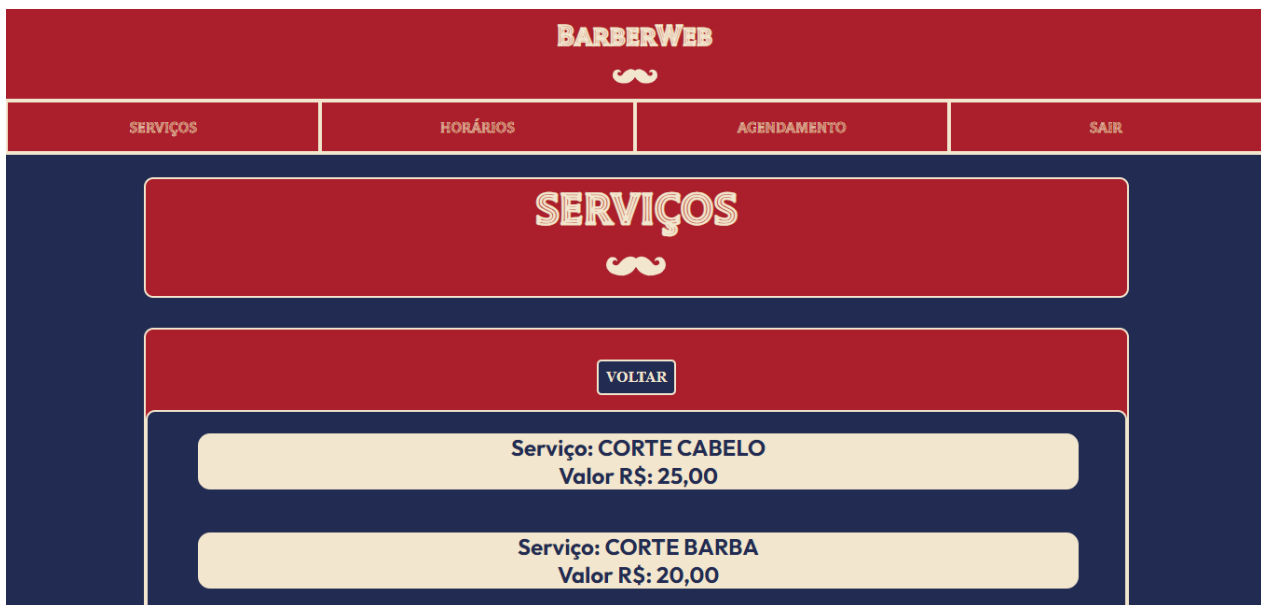


Imagem 12 - Layout de Serviços



Imagem 13 – Layout de Horários

4.5.4 AGENDAMENTOS

O layout de agendamentos conta com uma divergência entre clientes e barbeiros, pois na área de clientes o layout precisa servir como uma forma de realizar tal função, e na de barbeiros a mesma precisa mostrar os agendamentos realizados. Seguem imagens ilustrativas dos layouts de Agendamentos:



Imagem 14 – Layout de Agendamentos para Clientes



Imagem 15 – Layout de Agendamentos para Barbeiros

4.6 CODIFICAÇÃO

Nesse subcapítulo será tratado o assunto de desenvolvimentos, onde as tecnologias anteriormente citadas foram utilizadas no processo, tendo um foco separado para o front-end e back-end.

4.6.1 FRONT-END MOBILE-FIRST

O Front-End do sistema foi a primeira parte a ser desenvolvida. Utilizando na maior parte apenas HTML e CSS, o front foi desenvolvido com a técnica de mobile-first, onde inicialmente os layouts são definidos com as medidas e formatos para Mobile, e depois expandidas para outras plataformas através de responsividade. Como por exemplo na imagem a seguir:



Imagem 16 – Layout Header Mobile

No exemplo acima, podemos ver o layout do Index em sua versão primária (mobile), da forma que foi inicialmente pensado, onde os botões para as funcionalidades se encontram em formato de coluna. O HTML e CSS do Header, foram utilizados em todos os layouts pelo fato de todos possuírem o mesmo header. Com isso economizamos códigos, e

deixamos o sistema mais leve. Podemos ver a seguir o HTML e parte do CSS da versão mobile:

```
<header class="estiloHeader">
  <div class="estiloHeader_logomarca">
    <h1 class="estiloHeader_logomarca_titulo">BarberWeb</h1>
    
  </div>
  <section class="estiloHeader_botoes">
    <a href="../../services/servicesCliente/" class="estiloHeader_botoes_links">SERVIÇOS</a>
    <a href="../../hours/hourscliente/horarioCliente.html" class="estiloHeader_botoes_links">HORÁRIOS</a>
    <a href="../../scheduling/schedulingClient/schedulingCliente.html" class="estiloHeader_botoes_links">AGENDAMENTO</a>
    <a href="../../Index.html" class="estiloHeader_botoes_links">SAIR</a>
  </section>
</header>
```

Imagem 17 – HTML Header

Como visto o header utiliza de classes em suas tags para que o CSS consiga entender qual estilo aplicar em cada uma, segue exemplo das definições de tais classes na folha de estilo:

```
.estiloHeader {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  background-color: var(--cor-secundaria);
  border-bottom: 2px solid var(--cor-terciaria);
}

.estiloHeader_logomarca {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  width: 100%;
  margin-top: 0.5em;
  border-bottom: 2px solid var(--cor-terciaria);
}
```

Imagem 18 – CSS Header usado para Mobile

Como podemos ver, as classes “EstiloHeader” que define o padrão para o header como um todo, e “EstiloHeader_Logomarca” que define os padrões a serem seguidos pela logo respectivamente estão recebendo suas características que estão sendo aplicadas no HTML, como por exemplo “background-color” que define a cor de fundo da tag em si. Então com tudo definido, foi necessário mudar apenas alguns parâmetros de tamanhos, baseando-se no tamanho de telas, seguem exemplos para Notebooks e Monitores de grande escala, juntamente de seus respectivos CSS’s:



Imagem 19 – Layout Header Notebooks

Os tamanhos definidos para fontes e espaçamentos são feitos através de unidades de medidas que quando aplicadas já se tornam responsivas ao tamanho da tela, que no caso em questão são “em” e “rem”, por conta disso para o layout de notebooks, foi necessário apenas trocar a posição dos itens filhos do Header:

```
@media screen and (min-width: 1024px) {  
  
  .estiloHeader_botoes {  
    flex-direction: row;  
  }  
}
```

Imagem 20 – CSS Header usado para Notebooks

Essa mudança por tamanho é feita através da função de Media Querys, que permitem que o CSS e o navegador entendam que o que está ali dentro, é para ser mostrado em telas que atinjam o tamanho pedido, no exemplo acima 1024px. Segue imagens para ilustrar as versões para munitores:



Imagem 21 – Layout Header para Monitores

Como vemos o Header na versão para monitores fica totalmente em linha, no caso algumas outras modificações foram colocadas para que o layout ficasse mais apresentável para os usuários que veriam nesse modo, segue exemplo usado no CSS:

```
@media screen and (min-width: 1720px) {  
  
  .estiloHeader {  
    flex-direction: row;  
    height: 20%;  
  }  
  
  .estiloHeader_botoes {  
    flex-direction: row;  
  }  
  
  .estiloHeader_logomarca {  
    width: 30%;  
    border-bottom: none;  
  }  
  
  .estiloHeader_botoes_links {  
    border: none;  
    font-size: 2rem;  
  }  
}
```

Imagem 22 – CSS Header usado para Monitores

4.6.2 JAVASCRIPT NO FRONT-END

O Javascript foi usado no front, para a definição dos serviços e horário no agendamento, como o sistema não teve tempo de ter a implementação aplicada por back-end, a declaração direta no front foi necessária. E o Javascript nos fornece a capacidade de registrar dados que após escolhidos pelo usuário, ainda conseguiram passar informações ao banco, e realizar a gravação, a seguir podemos ver um exemplo de como os serviços foram declarados no layout de agendamentos:

```
let services = [
  {
    name: "Corte Cabelo",
    value: 25,
  },
  {
    name: "Corte Barba",
    value: 20,
  },
  {
    name: "Acerto Sombrancelha",
    value: 10,
  },
  {
    name: "Corte Cabelo + Barba",
    value: 40,
  },
  {
    name: "Corte Cabelo + Barba + Sombrancelha",
    value: 45,
  }
]
```

Imagem 23 - Declaração de valores de serviços

Como podemos ver, uma variável (let), foi definida com o nome “services”, onde a mesma recebeu diversos objetos com parâmetros “name” e “value”, já definidos. A seguir podemos ver o processo para salvamento em banco assim que selecionado e a apresentação no layout:

```

const addService = (name = '', value = 0) => {
  body['nameService'] = name
  body['valueService'] = value
  body['valueTotal'] = value
}

const servicesList = document.getElementById('section-list-services')

let listServices = ''

services.map(({ name, value }) => listServices = listServices +
  `<div id="${name}" onclick="addService('${name}', ${value})" class="estiloMain_telas_checkbox"><h2>Serviço:
  ${name}<br> Valor R$: ${value},00</h2></div>`)

servicesList.innerHTML = listServices

```

Imagem 24 – Salvamento e Definição de serviços em tela

Como podemos notar, uma const foi declarada como “AddService”, const essa que recebe como parâmetros name e value, que serão respectivamente puxados assim que selecionados pelo usuário. Essa const tem como função mostrar ao back-end quais são os dados que ele precisa gravar e em qual parte da tabela ele deve alocar cada dado.

Seguindo pela declaração da const “servicesList” que tem como a função de mostrar seus elementos na sessão referenciada através do método “getElementById. Por fim podemos ver no “services.map” que tal map é corrido recebendo todos os dados já declarados anteriormente, e criando um objeto dentro da já declarada “listServices”. Após isso podemos notar que cada objeto recebe seu respectivo estilo, e também a função “onclick” para que se tornem clicáveis e selecionáveis.

Por último “serviceList” recebe a variável “listServices” já criando um HTML na section já referenciada anteriormente para que o usuário consiga ter acesso aos dados cadastrados, e também selecionar cada um deles. Segue exemplo de como é visto pelo usuário:



Imagem 25 - Serviços mostrados em tela

O mesmo procedimento foi feito com horários, para que os mesmos possam também ser mostrados ao usuário e selecionados. Após seleção, o usuário pode clicar em “Agendar” que já tem definido através de onclick, que executará a função “HandleSubit”, que é responsável por realizar o método Post, levando os dados até o back-end. A seguir podemos ver a função:


```

const handleSubmit = () => {
  console.log(body)
  createScheduling(body).then(() => {

  })
}

let body = {
  nameService: "",
  hour: 0,
  valueService: 0,
  valueTotal: 0,
}

const showScheduled = () => {
  document.getElementById('scheduled-success').innerHTML = `
  <div class="estiloMain_telas_checkbox"> <h2> AGENDADO <br> Servico: ${body.nameService} <br>
  Horário: ${body.hour}:00 <br> Valor R$: ${body.valueTotal},00 </h2></div>
`
}

const showError = () => {
  console.log('showErrorshowErrorshowError')
  document.getElementById('scheduled-success').innerHTML = `
  <div class="estiloMain_telas_checkbox"><h2>Esse horário já esta reservado!</h2> <br>
  <h2>Por favor escolha outro!</h2></div>
`
}
}

```

Imagem 26 - Função para gravar agendamentos em banco

A função “handleSubmit” realiza o metodo Post, através da função body que é levada até o banco. Assim que realizada, o código verifica o sucesso ou fracasso, onde caso sucesso o agendamento é mostrado em tela para servir como confirmação para o usuário, já no caso de fracasso a mensagem de erro é emitida para que o mesmo consiga ter noção que o agendamento não foi realizado.

4.6.3 BACK-END

O back-end foi a parte mais problemática no desenvolvimento do sistema, e isso ocasionou em diversos cortes para o produto, onde inicialmente teria várias outras funções que não foram possíveis de serem implementadas.

4.6.3.1 INDEX

Para a criação do projeto foram usados diversos comandos, para que o vs-code entendesse com quais ferramentas iremos trabalhar, inicialmente começando com

“npm init -y” que basicamente diz que iremos usar o NodeJs, seguido por “npm i express” onde o comando criará nosso layout padrão de folders e também deixará instalado o express para uso. Segue figura ilustrativa do layout de folders:

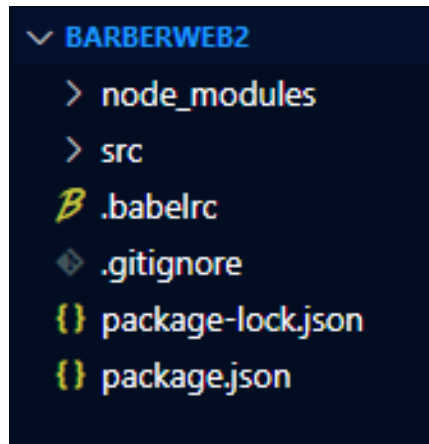


Imagem 27 – Layout de Projeto Node e Express

Seguindo por baixar as dependências que seriam usadas, sempre utilizando comandos com base no “npm l” com isso conseguimos utilizar e realizar todas as funções que queremos. O index do back-end serviu como a porta de entrada e ativação do sistema. A seguir podemos ver a figura ilustrativa do mesmo:

```
import express from 'express'
import bodyParser from 'body-parser'
import schedulingController from './controllers/scheduling'
import cors from 'cors'

const app = express()
const port = 3000

app.use(cors())
app.use(bodyParser.json())

app.use('/scheduling', schedulingController)

app.listen(port, () => {
  console.log(`App rodando em http://localhost:${port}`)
})
```

Imagem 28 – Index Back-End

Podemos ver que iniciamos importando todas as tecnologias que serão usadas, respectivamente de cima para baixo:

- Express: Framework usados na aplicação.
- Body-Parser: Dependência para que possamos receber informações vindas do body.
- SchedulingController: Importação do File schedulingController;
- Cors: Dependência utilizada para a conversa entre front-end e banco de dados.

Após isso podemos ver a declaração de consts para o código entenda o que são quando forem usadas, uma recebendo as funções do Express e a outra apenas a numeração da porta em que a aplicação funcionará.

Por fim vemos uma sequência de “app.use” onde basicamente cada uma delas traz uma função, para utilizar as tecnologias já importadas anteriormente, declaração de uso de rota para o controller, e por fim um aviso que o sistema está sendo inicializado.

4.6.3.2 CONTROLLERS

O file schedulingController é o responsável por definir as rotas para que cada verbo possa ser realizado, em outras palavras ele é como um guia para definir o que cada método realizará no banco de dados, usando os dados informados. Podemos ver a seguir uma imagem ilustrando as rotas para cada método utilizado na aplicação:

```
router.get('/', async (req, res) => {
  const schedulingList = await listSchedulings()
  res.status(201).send(schedulingList)
})

router.post('/', async (req, res) => {
  try {
    const scheduling = await createScheduling(req.body)
    res.status(201).send(scheduling)
  } catch (err) {
    res.status(400).send(err)
  }
})

router.delete('/:schedulingId', async (req, res) => {
  await deleteScheduling(req.params.schedulingId)
  res.send()
})
```

Imagem 29 - Definições de rotas para cada verbo

Como podemos ver apenas três verbos foram utilizados na aplicação, (get, post e delete), como o sistema de Login não estava implementado, acreditamos que o update era desnecessário por momento.

4.6.3.3 SERVICES

O file schedulingServices, serve para dizer ao sistema como ele deve agir quando um verbo é solicitado pelo controller, sendo assim é no services que cada função é declarada para receber cada verbo. Podemos ver essas definições na imagem exemplo a seguir:

```

export const listSchedulings = async () => {
  await databaseConnection()
  const schedulings = await Scheduling.find()
  return schedulings
}

export const createScheduling = async (scheduling) => {
  await databaseConnection()
  const createdScheduling = await Scheduling.create(scheduling)
  return createdScheduling
}

export const deleteScheduling = async (id) => {
  await databaseConnection()
  await Scheduling.findByIdAndDelete(id)
}

```

Imagem 30 – Services

4.6.3.4 MODELS

O file schedulingModels, tem como função mostrar ao banco de dados os tipos de dados que uma coleção deve receber, por conta disso uma const é criada para que a ORM (Object Relational Mapping, ou Mapeamento Objeto Relacional), consiga coletar os dados jogados nela e levar para o banco de dados. Conseguimos ver um exemplo do modelo utilizado na aplicação a seguir:

```

import mongoose from "mongoose";

const SchedulingSchema = new mongoose.Schema({
  nameService: {type: String, required: true},
  valueService: {type: Number, required: true},
  hour: {type: Number, required: true, unique: true},
  valueTotal: {type: Number, required: true}
})

export default mongoose.models.Scheduling || mongoose.model('Scheduling', SchedulingSchema)

```

Imagem 31 – Models

4.6.3.5 UTILS

Por fim, o file database foi utilizado na aplicação para realizar a conversa entre o back-end e o banco. Basicamente dizer ao código onde os dados devem ser armazenados e consultados.

```
import mongoose from "mongoose";

const URI = "mongodb+srv://matheusmarquesdesouza26:mat260599@cluster0.uvvr2oz.mongodb.net/"

const databaseConnection = async () => {
  if (!global.mongoose) {
    mongoose.set('strictQuery', false)
    global.mongoose = await mongoose.connect(URI)
  }
}

export default databaseConnection
```

Imagem 32 – Database

Como podemos ver, também realizamos a importação da ORM Mongoose, que é uma dependência de alta compatibilidade com o banco de dados MongoDB que foi o usado na aplicação. Seguida então pela const URI, que nos traça uma rota direta ao banco de dados criado gratuitamente pelo site do Mongo, e por fim a const de conexão com o banco denominada de “databaseConnection”.

5. Conclusão

Com base no que vimos anteriormente, concluíse que o devido projeto se concentra no desenvolvimento de uma plataforma para barbearias e barbershops terem maior visibilidade e para que seus clientes tenham maiores facilidades na hora de escolher um bafeiro e realizar seus agendamentos. O trabalho em questão se encontra no estado de desenvolvimento, entretanto já oferece uma boa visibilidade do que poderá ser no futuro.

O sistema está sendo desenvolvido utilizando diversas tecnologias como Javascript, NodeJs, Express, MongoDB, HTML5, CSS3, VSCode, Onde a metodologia de estudos e implementação seguiu de forma concisa, desde os diagramas até o fim do desenvolvimento.

Entretanto temos que ressaltar que diversas funcionalidades do projeto tiveram que ser retiradas até o momento, pois o tempo limite não nos permitiria implementar tudo que foi proposto, sendo assim o projeto final é apenas um MVP do que a BarberWeb será no futuro.

Com isso concluímos que a presente etapa do projeto, apresenta uma estrutura onde podemos sim ter uma base do que o projeto será, trazendo plena visão de como seguir os próximos passos e também através de mercado, demonstrando potencial para entrar no ramo de barbearias e barbershops.

Referencias

ABIHPEC. Mercado masculino de HPPC segue em crescimento, 2016, disponível em <<https://abihpec.org.br/release/mercado-masculino-de-hppc-segue-em-crescimento/>> Acessado em 25 de fevereiro de 2023.

ANDRADE, Ana Paula. O que é o Express JS?. 2021, disponível em <<https://www.treinaweb.com.br/blog/o-que-e-o-express-js>> Acessado em 02 de setembro de 2023.

BESSA, Adre. Node.JS: o que é, como funciona esse ambiente de execução JavaScript e um Guia para iniciar, 2023, disponível em <<https://www.alura.com.br/artigos/node-js>> Acessado em 01/09/2023.

CARVALHO, Adreyna. Entenda o que é diagrama de entidade e relacionamento (DER), 2023, disponível em <[https://coodesh.com/blog/candidates/entenda-o-que-e-diagrama-de-entidade-e-relacionamento-der/#:~:text=Diagrama%20de%20Entidade%20e%20Relacionamento%20\(DER\)%20%C3%A9%20uma%20representa%C3%A7%C3%A3o%20gr%C3%A1fica,e%20consequentemente%20mais%20intuitiva](https://coodesh.com/blog/candidates/entenda-o-que-e-diagrama-de-entidade-e-relacionamento-der/#:~:text=Diagrama%20de%20Entidade%20e%20Relacionamento%20(DER)%20%C3%A9%20uma%20representa%C3%A7%C3%A3o%20gr%C3%A1fica,e%20consequentemente%20mais%20intuitiva)> Acessado em 07 de março de 2023.

ESTÁCIO. O que é mapa mental e como utiliza-lo para estudar? Descubra!, 2021, disponível em <https://blog.estacio.br/calouros/o-que-e-mapa-mental/?gclid=Cj0KCQjwk7ugBhDIARIsAGuvGPavNJcASpR7ysMGLm2Ev-tSm-j93WrSsmqyTBWXvS3yFKPRfraOvDcaAiR4EALw_wcB&gclsrc=aw.ds> Acessado em 09 de março de 2023.

FLANAGAN, David. Javascript – O guia definitivo. 6º Ed. Tradução João Eduardo Nóbrega Tortello. Editora Bookman, 2013

FRESHA. O melhor software para salões e spah, 2023, disponível em <https://www.fresha.com/pt/for-business?utm_source=google&utm_medium=paid&utm_campaign=AVE_BR_Search_Salon_Translated&utm_term=Search_Barbershop&utm_content=software%20para%20barbearia&cq_src=google_ads&cq_cmp=18311187088&cq_term=software%20para%20barbearia&cq_plac=&cq_net=g&cq_plt=gp&gclid=Cj0KCQiAutyfBhCMARIsAMgcRJRTPLzi3iJkHvNLe1kqkzDw6i-36VWRYxSqn7mVgunOL6QSLIcd33MaAh__EALw_wcB> Acessado em 25 de fevereiro de 2023.

HANASHIRA, Akira. VS Code – o que é e por que você deve usar?, 2021, disponível em <<https://www.treinaweb.com.br/blog/vs-code-o-que-e-e-por-que-voce-deve-usar>> Acessado em 01 de março de 2023.

IMB. Diagramas de atividades, 2023, disponível em < <https://www.ibm.com/docs/pt-br/rational-soft-arch/9.7.0?topic=diagrams-activity>> Acessado em 09 de março de 2023.

IMB. Diagramas de caso de uso, 2023, disponível em <<https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case>> Acessado em 09 de março de 2023.

LOMBARDI, Talita. O que é Astah? #PostTécnico | Por Bruno Seabra, 2021, disponível em < <https://workstars.com.br/tech-news/o-que-e-o-astah-posttecnico-por-bruno-seabra/>> Acessado em 01 de março de 2023.

MONGODB. O que é MongoDB?, 2023, disponível em < <https://www.mongodb.com/pt-br/what-is-mongodb> > acessado em 09 de março de 2023.

MOZILLA. HTML: Linguagem de Marcação de Hipertexto, 2023, disponível em < <https://developer.mozilla.org/pt-BR/docs/Web/HTML> > acessado em 02 de setembro de 2023.

MOZILLA. O que é CSS?, 2023, disponível em < https://developer.mozilla.org/pt-BR/docs/Learn/CSS/First_steps/What_is_CSS > acessado em 02 de setembro de 2023.

NOLETO, Cairo. UML: O que é, para que serve e quando usar a linguagem de notação?, 2020, disponível em <<https://blog.betrybe.com/tecnologia/uml/>> Acessado em 01 de março de 2023.

OLIVEIRA, Cláudio Luiz V. ZANETTI, Humberto Augusto P. Node.js – Programe de Forma rápida e prática. São Paulo, Expressa, 2022.

RODEVA, ugo. Javascript: O que é, para que serve e como funciona JS?, 2021, disponível em <<https://kenzie.com.br/blog/javascript/>> Acessado em 01 de março de 2023.

TERRA. Novo conceito de Barbearias conquista espaço no disputado mercado de beleza brasileiro, 2017, disponível em <<https://www.terra.com.br/noticias/novo-conceito-de->

barbearia-conquista-espaco-no-disputado-mercado-de-beleza-brasileiro,5a5dfefdf4fd36aa81199f2d166a8tvkvp07p.html> Acessado em 25 de fevereiro de 2023.