**UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO**

Claudia Regina Adati, RA: 2001561

Eduardo de Domingos Neto, RA: 2008020

Fabiano Gonçalves Kataoka, RA:2010241

José Eugênio Damiati Theossi, RA: 2014566

Josiane Barbosa de Pieri, RA: 2010701

Luciano Alves Moreira, RA: 2000521

Matheus Fonseca de Macêdo, RA: 2002271

Otavio Ferreira Martins, RA: 2004959

***Software* para Agendamento de Serviços Estéticos**

**Tecnologia como suporte a pequenas empreendedoras da área de estética**

|  |
| --- |
| **Vídeo de apresentação do Projeto Integrador**  <link> |

Jaú e Cidade de Areias - SP

2022

**UNIVERSIDADE VIRTUAL DO ESTADO DE SÃO PAULO**

***Software* para Agendamento de Serviços Estéticos**

Relatório Técnico-Científico apresentado na disciplina de Projeto Integrador para o curso de Bacharel em Tecnologia da Informação e Engenharia da Computação da Universidade Virtual do Estado de São Paulo (UNIVESP).

Jaú e Cidade de Areias - SP

2022

ADATI, Claudia Regina; DE PIERI, Josiane Barbosa; DOMINGOS NETO, Eduardo; KATAOKA, Fabiano Gonçalves; MACÊDO, Matheus Fonseca; MARTINS, Otavio Ferreira; MOREIRA, Luciano Alves; THEOSSI, José Eugênio Damiati. **Software para Agendamento de Serviços Estéticos.** Relatório Técnico-Científico. Bacharelado em Tecnologia da Informação e Engenharia da Computação – **Universidade Virtual do Estado de São Paulo**. Tutor: Talita Helena Landiva. Pólos: Jaú e Cidade de Areias, 2022.

**RESUMO**

A utilização de *softwares* para tarefas do cotidiano tornou-se uma ferramenta de grande importância na resolução de problemas e melhorias tecnológicas. O fácil acesso e a utilização de aplicativos apontam que essa evolução tecnológica pode otimizar o tempo, custo e controlar processos com muito mais eficácia e rapidez. O desenvolvimento de *softwares* que atendam às necessidades pontuais de determinados segmentos não só auxilia, mas cria oportunidades de crescimento, que associada a um bom *software* de respostas precisas, eficientes, de fácil usabilidade, devem ser levados em conta, no desenvolvimento e criação. Diante desta oportunidade, o objetivo desse trabalho consiste em desenvolver um *software* que auxilie no controle de agendamento de uma microempreendedora de estética e alongamento de cílios, de modo a facilitar a escolha de data e horário, minimizando falhas, duplos agendamentos e cancelamentos de última hora, colaborando para a utilização total dos horários da agenda disponível e diminuição da insatisfação dos clientes.

**PALAVRAS-CHAVE:** *Software*, agendamento, otimização, controle.

**SUMÁRIO**

[1. INTRODUÇÃO 8](#_Toc59168079)

[2. DESENVOLVIMENTO 9](#_Toc59168080)

[2.1. Objetivos 9](#_Toc59168081)

[2.2. Justificativa e delimitação do problema 9](#_Toc59168082)

[2. 3. Fundamentação teórica 9](#_Toc59168083)

[2.4. Metodologia 10](#_Toc59168084)

[REFERÊNCIAS 12](#_Toc59168085)

[ANEXOS (opcional) 13](#_Toc59168086)

[APÊNDICES (opcional) 14](#_Toc59168087)

[Apêndice A – 14](#_Toc59168088)

# 1. INTRODUÇÃO

A pandemia por *Covid*-19 ocasionou um cenário de incertezas econômicas, mas de intensificação do uso das tecnologias e aceleração dos processos digitais. Essa dinâmica alterou a experiência e a interação dos usuários com as tecnologias e mudou a forma como os consumidores interagem com o mundo dos negócios (SEIN, 2020).

Além do setor de tecnologia, outro mercado em crescente expansão foi o de estética (ABIHPEC, 2022). Ressaltando que o setor de predominância de MEI (Microempreendedor Individual) do sexo feminino é a atividade de tratamento de beleza (SEBRAE, 2019).

Diante do exposto, é bastante promissor oferecer um recurso tecnológico para que microempreendedoras da área da estética possam interagir com os seus clientes, visando proporcionar experiências aprimoradas para o consumo do serviço e gerar dados que possam ser transformados em informações úteis para captação e fidelização dos clientes.

O presente Projeto Integrador tem como objetivo, desenvolver um *software* com *framework web* que utilize conceitos de bancos de dados, inclua *script web* na linguagem *JavaScript*, nuvem, uso de API (*ApplicationProgram Interface)*, acessibilidade, controle de versão e testes.

Em todas as etapas de imersão até a descoberta do problema, foi mantido o foco no usuário e utilizando as etapas da metodologia *Design Thinking* e sessões de *brainstormings* e mapas mentais para discussão e organização das ideias.

# 2. DESENVOLVIMENTO

## **2.1 Objetivos**

O objetivo geral é desenvolver um *software* para agendamento com *framework web* que utilize conceitos de bancos de dados, inclua *script web* na linguagem *JavaScript*, nuvem, uso de API, acessibilidade, controle de versão e testes.

Para alcançar o objetivo exposto, os objetivos específicos são:

* Criar a primeira versão do *software* a partir das necessidades da comunidade externa (solução inicial);
* Coletar *feedbacks* junto à comunidade externa a respeito da solução inicial;
* Implementar as melhorias sugeridas nos *feedbacks* (solução final).

## **2.2. Justificativa e delimitação do problema**

A delimitação do problema foi baseada na escuta e observação de uma profissional do ramo de estética que trabalha com extensão de cílios, como microempreendedora individual e que atende em uma sala locada em um centro de estética na cidade.

Para compreender os desafios enfrentados pela profissional, foram realizadas visitas no local para entrevista e entendimento por observação dos problemas. Alguns dos desafios enfrentados pela profissional entrevistada eram:

· Tempo dispensado para agendar os atendimentos;

· Esquecimento de inclusão na agenda do profissional ou da cliente o serviço agendado através de simples diálogo, o que geralmente acarretava em agendamento duplo (duas clientes agendadas no mesmo horário);

· Atendimentos perdidos por falta de organização da agenda, como por exemplo, as clientes terem até 20 (vinte) dias para realizar o agendamento do serviço de manutenção, mas acabavam agendando em datas posteriores em decorrência da desorganização citada. Isso acarretava em maior uso de material, pois havia maior queda dos cílios, insatisfação pela aparência (comentada pelas clientes) e até mesmo desistência do serviço.

A motivação para a escolha do escopo do trabalho e o tema foi baseada nos desafios enfrentados pela profissional de estética e baseada na questão: como a tecnologia pode otimizar os serviços prestados por essa profissional?

Diante do exposto, é proposto nesse projeto desenvolver um *software* para que profissionais autônomos do ramo de estética possam automatizar os processos manuais de agendamento de serviços, para garantir a otimização do tempo, praticidade para o cliente, resolver conflitos de agendas existentes e possibilitar ações para captação e fidelização de clientes.

Segundo Nassif, Corrêa e Rosseto (2020), após a pandemia por *Covid*-19, houve alterações nos padrões de consumo. Este evento acarretou na necessidade das empresas e microempreendedores repensarem as suas estratégias de vendas de serviços e produtos, buscando meios de atender às novas exigências do mercado. De certo modo, inovar é preciso, principalmente em ambientes de restrições de recursos, o que vêm se tornando uma realidade cada vez mais presente e necessária, muitas vezes, para manter o nível mínimo de operação com vistas à manutenção do negócio.

A tecnologia é uma ferramenta de suporte a pequenos empreendedores, proporcionando a substituição das tarefas manuais por processos automatizados. O *software* de agendamento traz benefícios como maior agilidade e otimização do tempo, redução de erros como agendamento de clientes distintos no mesmo horário entre outros benefícios.

O empreendedorismo e tecnologia são aliados para a sobrevivência de pequenos negócios, diante do cenário competitivo da economia brasileira e mundial (NASSIF, CORRÊA E ROSSETO, 2020).

* + 1. **Empreendedorismo, inovação e tecnologia**

Analisando a evolução histórica da tecnologia, inicialmente era considerada como uma atividade meio e não como atividade fim. Essa afirmação pode ser verificada quando pensamos em consumidores e empreendedores, pois a forma de usar a tecnologia define em qual perfil nos enquadramos. Os consumidores utilizam a tecnologia como fim em si mesmo (redes sociais, aplicativos de músicas, jogos, entre outros). Mas se utilizamos a tecnologia para satisfazer outras necessidades tanto nossas como as de terceiros, passamos a vê-la como empreendedores.

Os empreendedores visualizam a tecnologia como um facilitador, acelerador ou como um atalho para algum processo de trabalho e isso é primordial no atual cenário de mercado consumidor.

Nos últimos anos, muitas empresas têm procurado se renovar com o objetivo de acompanhar o rápido desenvolvimento tecnológico e a globalização dos mercados, bem como uma maior exigência dos consumidores por produtos e serviços (DORNELAS, 2015).

Segundo Wilians (2021), trazer a inovação social como modelo econômico viável para solucionar problemas inerentes à sociedade nos parece atrativo e urgente. Sendo assim, o pensamento empreendedor aliado com a tecnologia é primordial para o desenvolvimento de organizações e estabelecimentos que visam ter competitividade no mercado.

Com base nos conceitos de empreendedorismo e competitividade, e por meio da utilização de recursos tecnológicos, o presente projeto dará suporte a pequenos empreendedores de clinicas de estéticas através de uma aplicação *web* que realiza o agendamento de clientes.

## **2. 3. Fundamentação teórica**

Neste capítulo será apresentada toda a fundamentação teórica do projeto. São discutidas questões, como empreender com tecnologia, o setor de Estética no mercado atual e as ferramentas que são utilizadas para o desenvolvimento do projeto.

* + 1. ***Design Thinking***

O *Design Thinking* é um método de inovação desenvolvido pela *d. School*, um Instituto da Universidade de *Stanford*, no vale do Silício na Califórnia. É uma abordagem, uma forma de pensar e encarar problemas, focado na empatia, colaboração e experimentação (SIMON, 1969).

Entende-se também, que é uma metodologia criativa e prática para resolução de problemas e concepção de projetos, que tem sido usada por diversas organizações, na busca por inovação em negócios, processos, produtos e serviços (GRANDO, 2011). O autor completa que pode ser entendido como uma mentalidade otimista e experimental centrada no ser humano. É uma mentalidade porque, pensar como um designer é estar ciente do que acontece ao seu redor, ter consciência que pode fazer parte de um processo de mudança, com habilidades e ações criativas e decisões em direção de um futuro mais desejável (GRANDO, 2011). E experimental, pois oferece a oportunidade de tentativa e erro, porque se apresentam ideias, obtém-se *feedback* e tem-se a chance de realizar as mudanças, graças a sua natureza iterativa (GRANDO, 2011). Centrado no ser humano, porque se baseia na relação empática, e entendimento das necessidades e desejos das pessoas. A ideia fundamental e que ganhou espaço em muitas empresas, é fazer as coisas pensando no usuário delas, ou seja, pensar com a cabeça do público, e não com a cabeça da empresa. Mas para que o *design thinking* possa ser compreendido e aplicado, alguns elementos precisam ser melhor compreendidos.

Pode-se dividir o método em quatro etapas:

1. Imersão: análise do cenário e oportunidades atuais, coletando informações necessárias para entendimento do problema da comunidade.
2. Ideação: encontros onde todas as ideias são geradas e sugeridas. Momento em que todos têm a liberdade para colocar seus posicionamentos, *brainstorming.*
3. Prototipação: escolha das melhores ideias que podem ser utilizadas. Essa etapa é de extrema importância, pois possíveis falhas ou ideias incompletas podem ser perceptíveis quando colocadas em um protótipo.
4. Desenvolvimento: Executa-se a implementação das funcionalidades. É colocado em prática tudo que foi testado e aprovado no protótipo.

Resultados significativos podem ser obtidos utilizando este método, sendo que esta abordagem pode ser utilizada na criação de novos produtos e também na solução de problemas encontrados, segundo *Brown* (2010).

O conceito de solução para o problema escolhido foi baseado na metodologia de *design thinking*, com abordagem qualitativa. Ainda no método de *design thinking,* a inovação provida deve ser gerada a partir da perspectiva do usuário. Assim sendo, torna-se importante compreender suas experiências, necessidades e desafios, como se sente e quais suas dificuldades, aborda PINHEIRO (2011).

* + 1. **Controle de Versão**

Controle de versão, segundo DIAS (2016), é a prática de rastrear e gerenciar as alterações realizadas em um código-fonte de *software*. Os sistemas de controle de versão são ferramentas computacionais que ajudam as equipes de *software* a gerenciar as alterações ao código-fonte ao longo do tempo. O *software* de controle de versão mantém registro de todas as modificações no código-fonte em um tipo especial de banco de dados, se um erro for cometido, os desenvolvedores podem utilizar versões anteriores e compará-las com o objetivo de encontrar e corrigir o erro. Além de um *software* local de controle de versão, ou seja, um que esteja instalado na máquina em que se está trabalhando com o desenvolvimento de códigos-fontes, há também serviços de repositório online para o armazenamento desses mesmos controles de versão, o que facilita tanto o desenvolvimento colaborativo de softwares quanto a disseminação de conhecimento através da disponibilização de tais códigos-fontes à comunidade de desenvolvedores. Para o presente projeto foi escolhido utilizar o *Git* como controlador de versionamento local e o *GitHub* como repositório online.

* + 1. ***Software* de Agendamento**

Automação de procedimentos manuais proporciona uma melhora significativa, neste

caso, afetando positivamente a forma de gerenciamento das agendas, horários, disponibilidade, possíveis atrasos, avisos e divulgação dos serviços das microempresárias (profissionais e estabelecimentos), assim como praticidade e comodidade para o cliente.

[...] aplicativos exercem um papel relevante em vários momentos do dia, como a busca de informações, conexão em redes sociais e a procura por produtos. Eles representam espaços privilegiados para que as marcas criem relações efetivas e produtivas. A publicidade nos aplicativos móveis possui a função de complementar a experiência off-line (ofertas especiais em lojas, por exemplo), direcionar o *e-commerce*, ou contribuir com a comunicação de qualidade da marca (BORGES, 2019, p. 105).

Procuramos desenvolver um *software* que auxilie nos problemas que o profissional encontra em sua rotina, como:

- Demora no tempo de espera;

- Faltas ou atrasos dos clientes;

- Tempo gasto desnecessariamente fazendo os agendamentos manualmente;

- Duplo agendamento de clientes;

- Distribuição ineficaz do *Marketing.*

* + 1. **Ferramentas para o Desenvolvimento**

As ferramentas escolhidas para o desenvolvimento do presente projeto são:

* *Visual Studio Code*: é um editor de código-fonte desenvolvido pela *Microsoft*, embora funcione em outros sistemas operacionais como *IOS* e *Linux*. Ela inclui uma conexão com o controle de versionamento *Git*.
* *Node.JS*: pode ser definido como um **ambiente de execução Javascript**server-side, ou seja, ele possibilita a criação de aplicações *Javascript* para rodar como uma aplicação standalone em uma máquina, não dependendo de um *browser* para a execução, como estamos acostumados.
* *MySQL Workbench*: é uma ferramenta multiplataforma, ou seja, disponível para todos os sistemas operacionais. Ela disponibiliza recursos de modelagem de dados, administração e configuração para a construção de banco de dados relacionais utilizando a linguagem *SQL*.
* *Figma*: é uma ferramenta utilizada pelos *designers* para a criação de interfaces (prototipagem de projetos).
* *Wamp Server***:** é um *software* utilizado para instalar no computador os *softwares PHP, MySQL e Apache*, disponibilizando suporte ao uso de *scripts PHP* localmente no *Windows*.
* *Outsystems Service Studio 11:* plataforma e *framework* pra desenvolvimento de *softwares web* com SGBD (Sistema Gerenciador de Banco de Dados) próprio, versionamento integrado, integração com o servidor *web* IIS da Microsoft nativo, entre outros recursos disponíveis.
  + - 1. **Linguagens de Programação**

Uma linguagem de programação é um método padronizado, formado por um conjunto de regras sintáticas e semânticas, para a implementação de um código fonte, que por sua vez poderá ser compilado e transformado em um programa de computador. Por meio do uso da linguagem de programação é possível que um programador especifique precisamente quais os dados em que o computador irá atuar, como serão armazenados ou transmitidos e quais ações devem ser tomadas de acordo com as circunstâncias (FISCHER; GRODZINSKY, 1993).

As linguagens de programação elegidas para o desenvolvimento deste projeto foram:

* *C# (CSharp)*: é uma linguagem de programação moderna, orientada a objeto e fortemente tipada. Permite o desenvolvimento de aplicativos seguros e robustos que são executados na plataforma .NET. O C# tem suas raízes na família de linguagens C e os programadores em C, C++, Java e JavaScript a reconhecerão imediatamente.
* *JavaScript* (abreviado como *JS*): é uma linguagem de programação interpretada estruturada, de *script* em alto nível com tipagem dinâmica fraca e multiparadigma (orientado a objeto, imperativa e funcional). É utilizada em páginas *web* para fazer verificações ou mudar dinamicamente a apresentação visual das páginas mas também do lado do servidor com o uso do *engine* Node.JS.
* *SQL (Standard Query Language)*: é uma linguagem padrão para trabalhar com bancos de dados relacionais. É declarativa e não precisa de profundos conhecimentos de programação. É utilizada na construção do banco de dados e para realizar consultas (*queries)* no mesmo (SILVEIRA, 2019).

As linguagens de marcação utilizadas no projeto foram:

* *HTML* (Linguagem de Marcação de Hipertexto): é uma linguagem utilizada para a construção e estruturação de páginas *web*. Os documentos *HTML* são interpretados por navegadores.
* *CSS* (*Cascading Style Sheets* ou Folhas de Estilo em Cascata): é uma maneira de dar estilo ao código criado por linguagem *HTML*. É usada para a personalização visual e otimização de conteúdo de um site. Ela complementa o desenvolvimento *HTML*.
  + - 1. **Banco de Dados**

Um banco de dados é um sistema que converte uma grande coleção de dados em uma ferramenta abstrata, permitindo que os usuários busquem e extraiam itens de informações pertinentes de forma conveniente (BROOKSHEAR, 2013).

Com dados, queremos dizer fatos conhecidos que podem ser registrados e possuem significado implícito (ELMASRI, 2018). Sendo assim, um banco de dados é item essencial para o desenvolvimento de um aplicativo *web* de agendamento de horários bem-sucedido, pois torna possível o registro e a recuperação de informações tais como horários agendados, nome do cliente, meios de contato etc.

* + - 1. ***Framework Web***

*Frameworks* são plataformas de desenvolvimento que fornecem modelos básicos com diferentes funcionalidades já implementadas, que podem ser utilizadas ou não pelo desenvolvedor. Costumam favorecer o gerenciamento ágil de projetos (MATOS e ZABOT, 2020).

Um *framework web* é destinado a suportar o desenvolvimento de aplicações voltadas para a *web*. Com um *framework*, é possível construir *sites*, *softwares* e aplicativos a partir de um esqueleto pré-definido, alterando apenas algumas particularidades.

Os *frameworks* servem para auxiliar o programador no momento de construir uma aplicação. Algumas de suas vantagens são: aumento de produtividade, códigos padronizados, redução do tempo de desenvolvimento, redução do custo de desenvolvimento, códigos com menor chance de erros, entre outros.

* + - 1. **Acessibilidade na *Web***

A acessibilidade é uma prática de desenvolvimento de aplicações *web* que possam ser acessada por todas as pessoas, inclusive por aquelas que possuam necessidades visuais, motora, auditiva, cognitiva, entre outras.

Segundo Santos e Marques (2016), optar pelo desenvolvimento de um projeto *web* com acessibilidade é mais interessante por eliminar as barreiras e empregar a igualdade de direito a todos, contribuindo para o avanço intelectual do ser humano e aderindo ao conceito de uma *web* para todos.

* + - 1. ***Sites* Responsivos**

O *web design* responsivo trata-se do conceito onde *layouts* de *sites* se adaptam a quaisquer dispositivos/resolução. Com aumento de usuários de *smartphones* era possível analisar que muitos sites ficavam ilegíveis quando abertos neste dispositivo. A adaptação de *sites* e a responsividade permitiu a navegabilidade rápida e de fácil acesso às informações em qualquer dispositivo, seja em computadores *desktops, tablets* ou celulares (*smartphones*) (FRANÇA, 2015).

* + - 1. **API (*Application Program Interface)***

API, acrônimo para *Application Program Interface*, é um conjunto de definições e protocolos para criar e integrar *softwares* de aplicações distintas. Como exemplos temos *Facebook* que é integrado ao *Instagram* por uma API, sua conta no *Google* que pode ser utilizada como *login* em diversos sites, a busca de logradouro pelo CEP para cálculo de frete em *e-commerces* e plataforma de pagamentos nesses mesmos *e-commerces*. Sendo todos estes APIs de desenvolvedores diferentes que conversam entre si.

* + - 1. **Testes de Aplicações *Web***

Os testes feitos em aplicações *web* medem o seu desempenho através

do uso de ferramentas que simulam o acesso de usuários ao *site*/aplicação, como também o seu comportamento esperado e confiabilidade.

Os 3 principais testes de aplicação *web* são:

* Testes unitários: tem interação direta com o código fonte, também conhecido como “caixa branca”.
* *Web Service*/API: validação de APIs, serviços e bibliotecas de aplicação, também conhecido como “caixa-preta.
* *GUI tests*: valida o comportamento das interfaces.
  + - 1. **Computação em Nuvem**

A computação em nuvem é um sistema distribuído que oferece serviços de computação (como servidores para armazenamento e processamento de dados, banco de dados, aplicativos, entre outros) através de uma rede de comunicação geralmente utilizando o protocolo de comunicação TCP/IP. O principal objetivo da computação em nuvem é fazer o melhor uso de recursos e resolver problemas em larga escala (SADIKU, MUSA E O MOMOH, 2014).

Utilizando os serviços de computação em nuvem hoje disponíveis no mercado por empresas como *Amazon Web Services (AWS), Oracle, Microsoft*, entre outras, é possível realizar o pagamento somente pela quantidade de recursos utilizados, possibilitando acessos a dispositivos e informações em qualquer lugar do globo terrestre que esteja conectado à *Internet*.

* + - 1. **Análise de Dados**

A análise de dados é um processo que visa resgatar os dados armazenados no banco de dados (ou em um *Datawarehouse)*, modelando-os com métodos estatísticos que visa a transformá-los em informações com a capacidade de gerar *insights e* dar suporte à tomada de decisão.

## **2.4. Aplicação das disciplinas estudadas no Projeto Integrador**

O desenvolvimento do presente projeto e sistema *web* utilizou conhecimentos de diversas matérias estudadas durante o curso do eixo computação da UNIVESP até o momento, incluindo matérias que estão sendo vistas concomitantemente ao próprio Projeto Integrador, tais como Engenharia de Software - COM390 e Segurança da Informação - COM440, que nos trouxe o conhecimento necessário para o planejamento e desenvolvimento de cada etapa do desenvolvimento de um software segundo as mais modernas diretrizes de desenvolvimento e segurança.

Também se utilizou matérias previamente estudadas, como Desenvolvimento Web – COM320, Interface Humano-Computador – COM330, Banco de Dados – COM300, Sistemas Computacionais COM210, Programação Orientada a Objetos – COM230, Infraestrutura para Sistemas de Softwares – COM310, Estrutura de Dados – COM160, Algoritmos e Programação de Computadores I – COM110 e Algoritmos e Programação de Computadores II – COM120 que foram de suma importância para a completude do objetivo final, entregar um sistema *web* integrado a um banco de dados para a melhoria de processos na comunidade externa selecionada. Dessa forma, consolidando conhecimentos já adquiridos e introduzindo os integrantes do grupo à prática em um contexto real, servindo como uma experiência enriquecedora para a aprendizagem acadêmica.

## **2.5. Metodologia**

Ao longo do desenvolvimento de nossa aplicação *web*, foram realizados um conjunto de atividades aprendidas na disciplina Engenharia de *Software* no intuito de definir, desenvolver, testar e manter o *software*, além de metodologias preconizadas pelo *Design Thinking*.

Algumas das atividades realizadas durante o desenvolvimento foram:

* Levantamento de requisitos (etapa de ouvir);
* Análise de Requisitos (etapa de ouvir e criar);
* Projeto (etapa para criar);
* Implementação (etapa para criar) e
* Testes (etapa para testar a solução).

**2.5.1. Levantamento de Requisitos**

Essa atividade tem como objetivo compreender o problema junto à comunidade externa a fim de levantar e priorizar as necessidades (requisitos) dos futuros usuários do *sistema*. Para realizá-la, ouvimos a comunidade com o objetivo de direcionar a criação e a implementação da solução.

**2.5.2. Análise (ou Especificação) de Requisitos**

Nessa etapa fizemos um estudo detalhado dos dados levantados na atividade anterior. Construímos modelos (protótipos) para representar o sistema a ser desenvolvido para resolver o problema solicitado pela comunidade externa. Então, validamos se o protótipo escolhido poderia resolver o problema e também satisfazer as solicitações do Projeto Integrador. Posteriormente, verificamos com a comunidade se os modelos construídos na análise estão em conformidade com os requisitos solicitados.

## **2.5..3. Projeto**

Na etapa da construção do projeto, utilizamos conhecimentos adquiridos na disciplina Banco de Dados para criar o funcionamento interno do sistema de modo que os requisitos especificados pelo cliente pudessem ser atendidos. Inicialmente pensamos na modelagem dos dados, no Sistema Gerenciador de Banco de Dados (SGBD), no banco de dados e nas linguagens de programação utilizadas.

A linguagem SQL é um grande padrão de banco de dados. Isto decorre da sua simplicidade e facilidade de uso. Ela se diferencia de outras linguagens de consulta a banco de dados no sentido em que uma consulta SQL especifica a forma do resultado e não o caminho para chegar a ele. Ela é uma linguagem declarativa em oposição a outras linguagens procedurais. Isto reduz o ciclo de aprendizado daqueles que se iniciam na linguagem.

Essa ferramenta é altamente versátil, permitindo que se lide diretamente com os dados e interaja com softwares e diversos sistemas digitais.

Utilizamos em nosso projeto integrador o MySQL Workbench, que é um sistema gerenciador de banco de dados relacional de código aberto usado na maioria das aplicações gratuitas para gerir suas bases de dados, com um modelo de cliente-servidor e os dados armazenados estão organizados em tabelas.

O sistema de agendamento de serviços para a profissional que trabalha com extensão de cílios deve possibilitar ao usuário (cliente) escolher um horário e data para receber o atendimento e as informações devem ser armazenadas em um banco de dados.

Para a modelagem dos dados, utilizamos o Modelo Entidade-Relacionamento (MER) para representar graficamente os objetos do mundo real e a forma como eles se relacionam (Figura A). Os principais componentes do MER são as entidades (objetos ou tabelas), seus atributos, e as relações entre elas, que definem o que será armazenado no banco de dados físico.

O conceito de chaves em um Banco de Dados trata-se de uma ou mais colunas que distinguem uma linha (registro) das demais dentro de uma entidade (tabela), tornando cada registro único, sendo esta chamada de chave primária (PK – *Primary Key*). Ou para relacionar uma entidade à outra, sendo este tipo de chave conhecida como chave estrangeira (FK – *Foreign Key*).

A partir das escolhas, construímos três tabelas: *Users*, *Appointments* e *Hour* que estão relacionadas conforme demonstrada na figura (Figura A):

Diagrama

Descrição gerada automaticamente

(Figura A)

A tabela *Users* possui 5 atributos (*id\_user, name, email, phone\_number, creation\_date*), sendo o atributo *id\_user* sua chave primária. A tabela *Users* está relacionada com a tabela *Appointments* em uma relação 1 – n (um para muitos).

A tabela *Hours* possui 4 atributos (*id\_hour, label, order, is\_active*), sendo o atributo *id\_hour* a chave primária. A tabela *Hours* está relacionada com a tabela *Appointments* em uma relação 1 - n.

A tabela *Appointments* possui 6 atributos (*id\_appointment, date, id\_user, id\_hour, users\_id\_user, hours\_id\_hour*), sendo *id\_appointment* chave primária, *id\_user* e *id\_hour* chaves estrangeiras, *users\_id\_user* e *hours\_id\_hour* as restrições que garantem a unicidade de horários nos agendamentos.

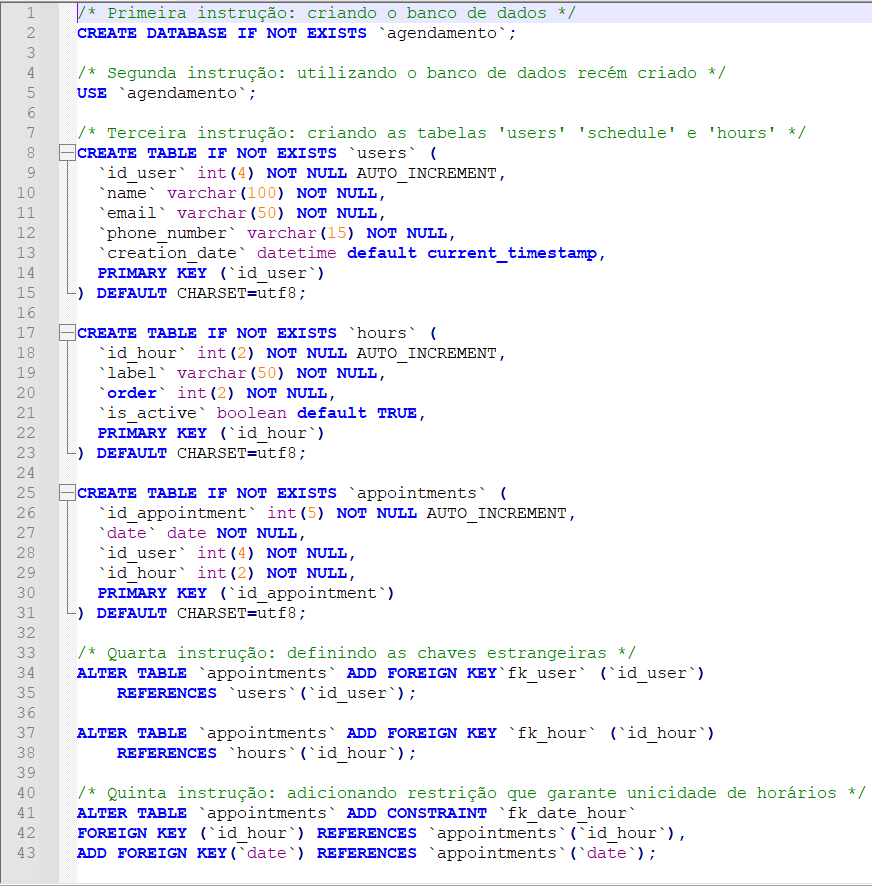
A tabela *Appointments* está relacionada com a tabela *Hour* (entidade estática para conter os registros dos horários permitidos na tabela *Appointments*) e com a tabela *Users* (entidade que conter os registros dos usuários cadastrados no sistema).

Para realizar a modelagem do banco de dados utilizamos o SGBD MySQL Workbench, posteriormente também foi utilizado o SGBD integrado ao *Framework Web* *OutSystem Service Studio* na sua versão 11.14.16 (Figura B), que disponibiliza o servidor na nuvem IIS da *Microsoft* para a hospedagem dos dados.



(Figura B)

Código SQL utilizado para a construção do Banco de Dados (Figura C):

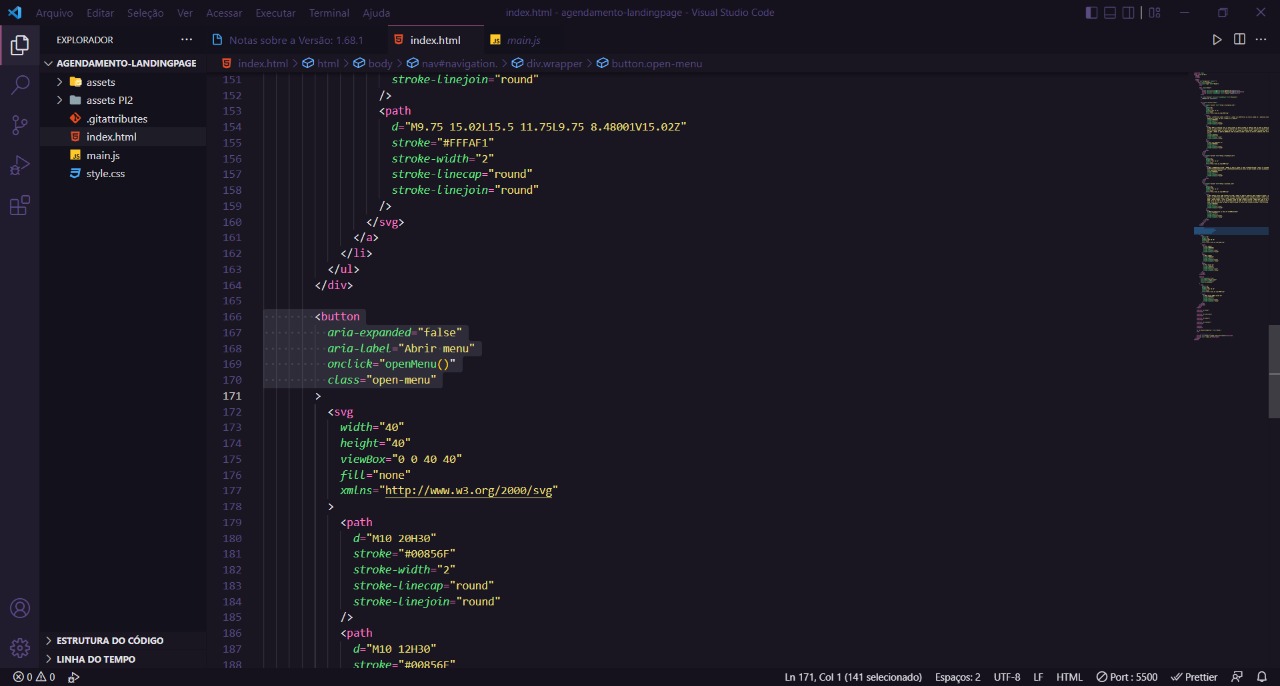


(Figura C)

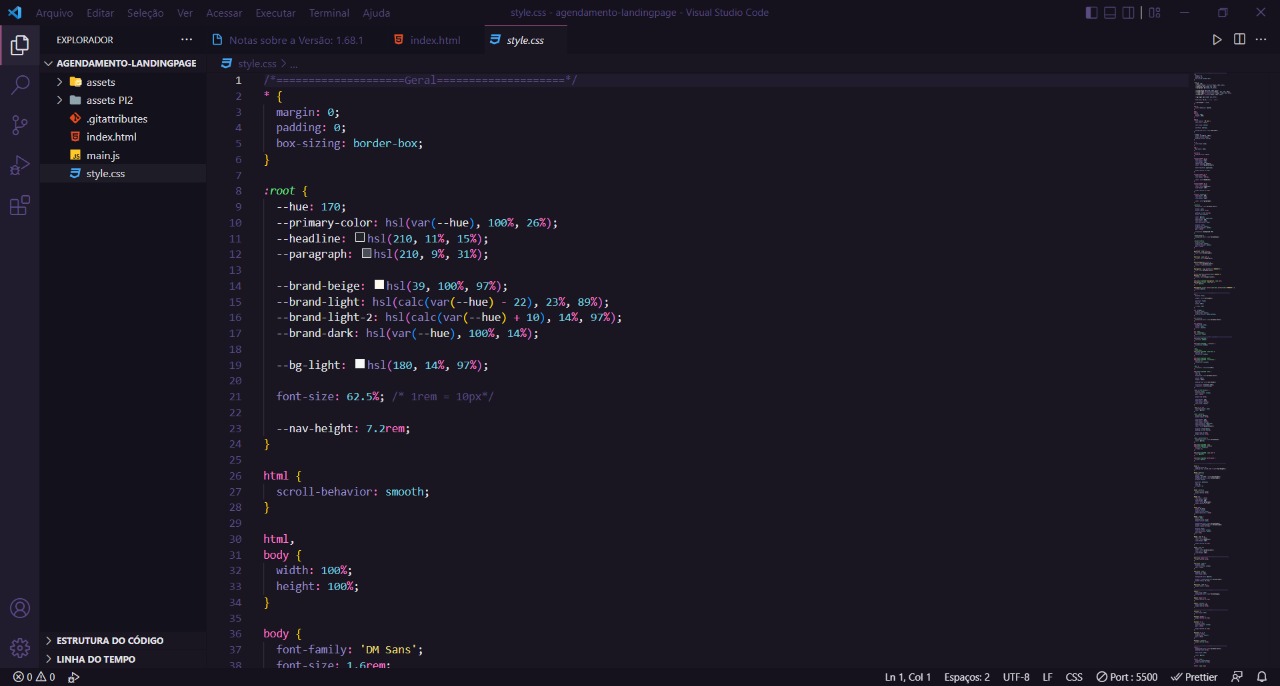
## **2.5.4. Desenvolvimento da Solução inicial**

Na etapa de desenvolvimento do sistema propriamente dito, inicia-se codificando à partir da descrição computacional realizada na fase de planejamento do projeto. O paradigma de desenvolvimento utilizado foi o orientado a objetos, utilizando as linguagens de programação JavaScript e C#, e as linguagens de marcação HTML e CSS.

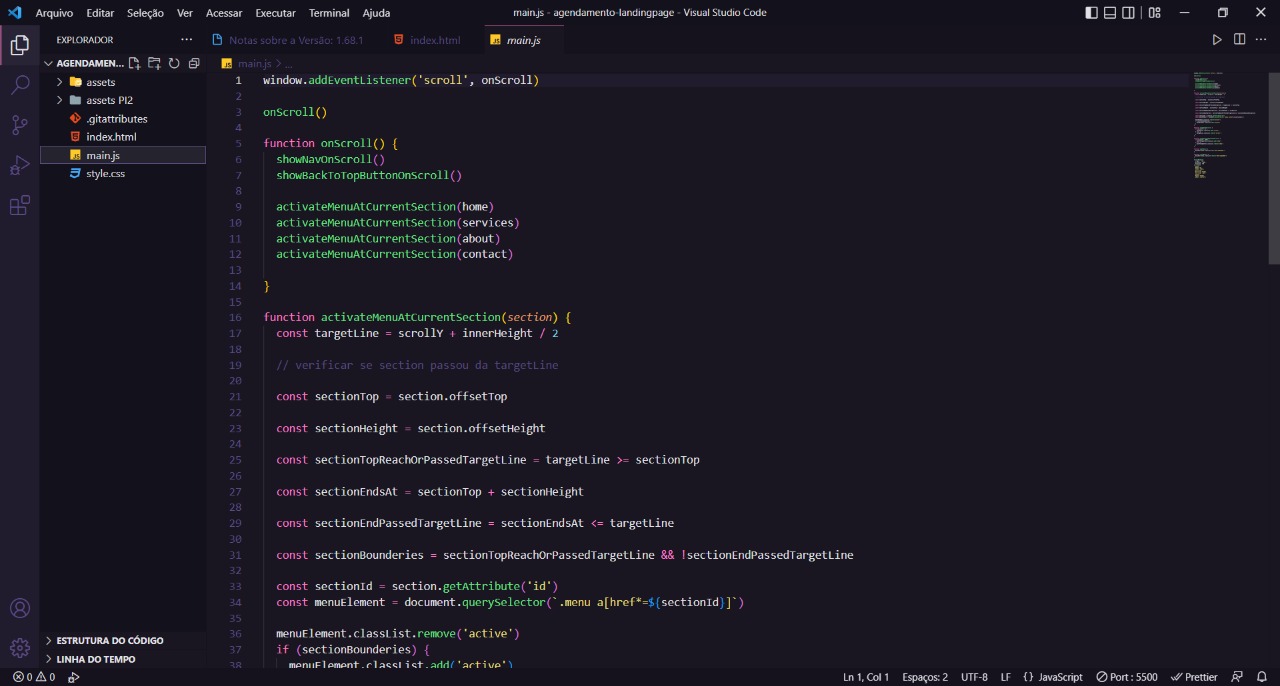
As tecnologias utilizadas para o desenvolvimento *front-end­* (a *interface*, parte da solução que o usuário final consegue ver e interagir, como os menus, imagens, cores, fontes, etc.) foram as linguagens de marcação HTML e CSS e a linguagem de programação JavaScript, dentro da IDE (*Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado) Visual Studio Code. Como visto nas figuras abaixo (Figuras D, E e F).



(Figura D: código HTML no IDE VSCode)



(Figura E: código CSS no IDE VSCode)



(Figura D: código JavaScript no IDE VSCode)

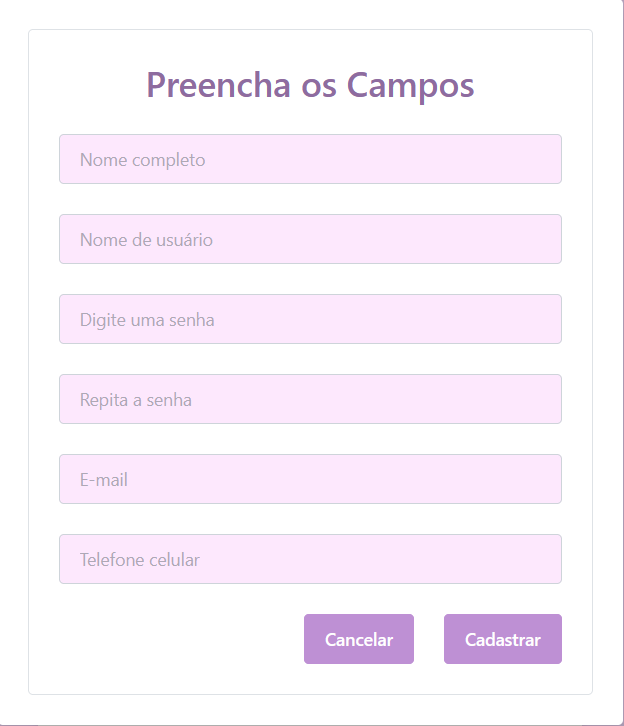
As tecnologias utilizadas para o desenvolvimento *back-end* (tudo que está por trás do site ou do *software*, que o usuário não tem acesso direto e nem consegue interagir) são as linguagens de programação JavaScript e C#.

Para garantir acessibilidade no desenvolvimento da aplicação, utilizamos o recurso ARIA (*Acessible Rich Internet Application*). O ARIA é um conjunto de atributos especiais que garantem acessibilidade e pode ser adicionado à linguagem de marcação HTML, sendo posteriormente utilizado por *softwares* de leitura de páginas, comumente usados por deficientes visuais (Figura G).



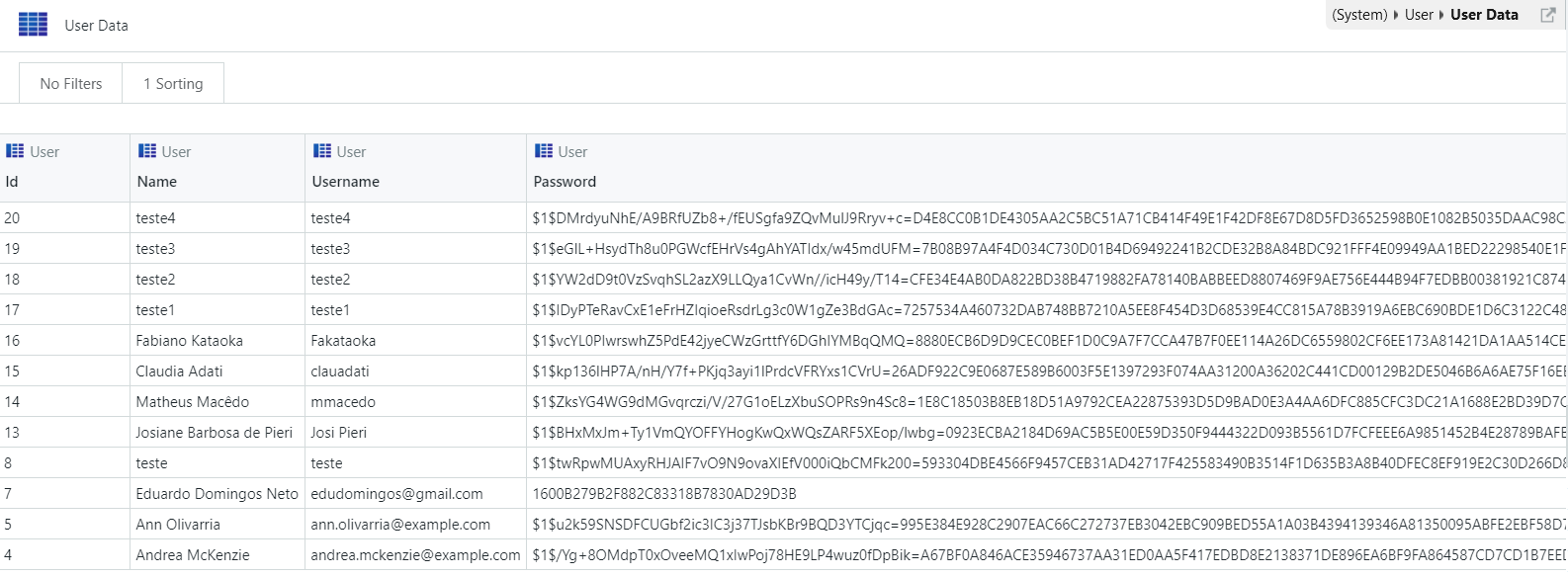
(Figura G: implementação de acessibilidade no código HTML)

Para desenvolver a parte da aplicação que integra as informações com o banco de dados, ou seja, a tela de *Login*, o usuário poderia realizar um cadastro e informar o nome completo, nome de usuário, senha, confirmação de senha, *e-mail* e número do celular, como visto na imagem abaixo (Figura H).



(Figura H: *PopUp* cadastramento)

As informações fornecidas pelo usuário são inseridas no banco de dados respeitando à Lei Geral de Proteção aos Dados (LGPD), Lei nº 13.709/2018, que estabelece regras para a coleta, armazenamento, tratamento e compartilhamento de dados pessoais. As senhas são informadas pelo usuário e inseridas no banco de dados de forma criptografada, não possibilitando nem mesmo aos gestores do banco de dados o acesso a essa informação, conforme figura abaixo (Figura I).

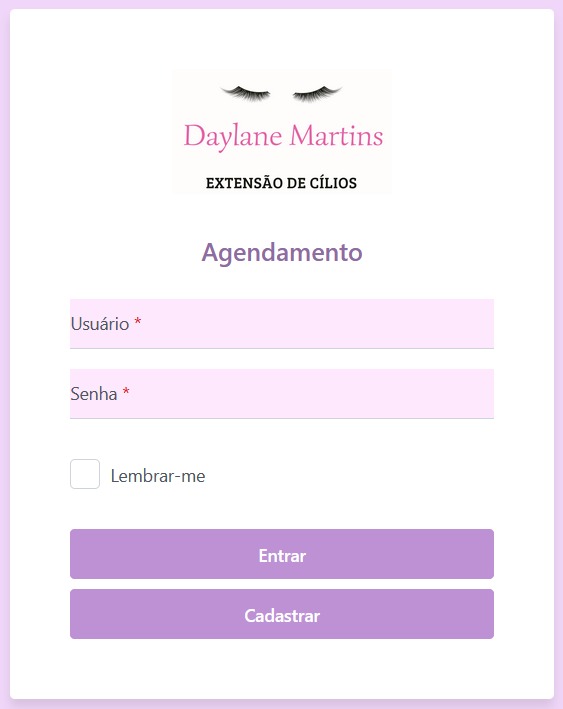


(Figura I: Tabela *Users* com atributo *password* criptografado)

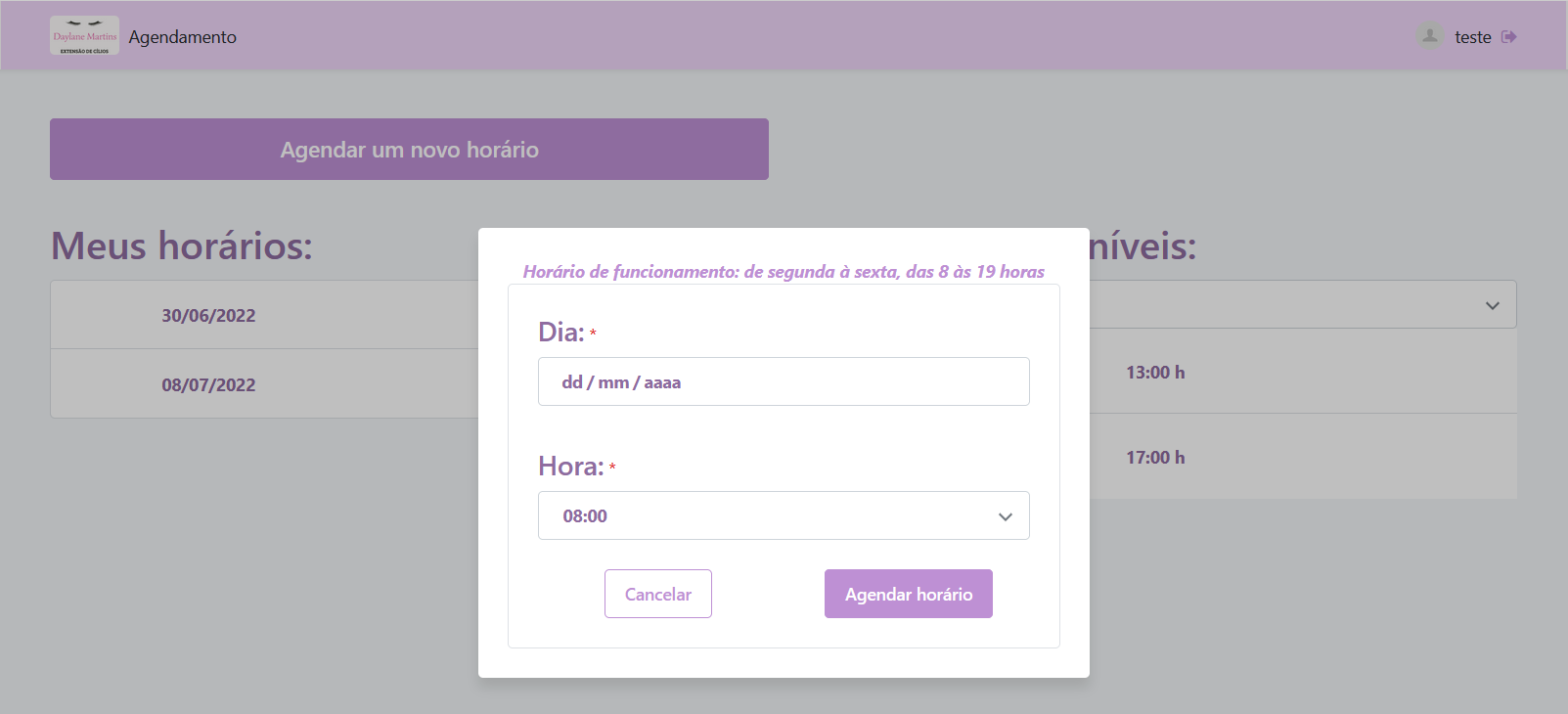
Após o cadastro do novo usuário, é permitido o agendamento através do acesso à tela de *Login* novamente. O usuário deverá informar o nome de usuário e a senha, e a tela de agendamento será disponibilizada (Figura J).

A tela de agendamento contém (Figura K):

* O botão Agendar Atendimento, que deve ser clicado para a escolha do dia e horário do atendimento;
* Meus horários: que disponibiliza as datas e horários agendados pelo usuário e
* Horários indisponíveis: são aqueles que a profissional não realizará atendimento, ou por já ter um agendamento ou por não trabalhar nesse horário. Essa informação é útil para informar um horário válido no momento de agendamento.



(Figura J: Tela de *Login*)



(Figura K: Tela de agendamento)

**2.5.5. Desenvolvimento da Solução Final**

Durante o desenvolvimento do projeto, foram utilizadas as Metodologias Ágeis *Scrum* e *Kanban* e atendendo aos pilares das Metodologias Ágeis, focamos em indivíduos e interações, na colaboração com o cliente e na resposta à mudança. Portanto, visando atender a necessidade da comunidade externa, foi desenvolvida uma *Landing Page* que funciona de modo integrado com *o sistema* desenvolvido para o serviço de agendamento.

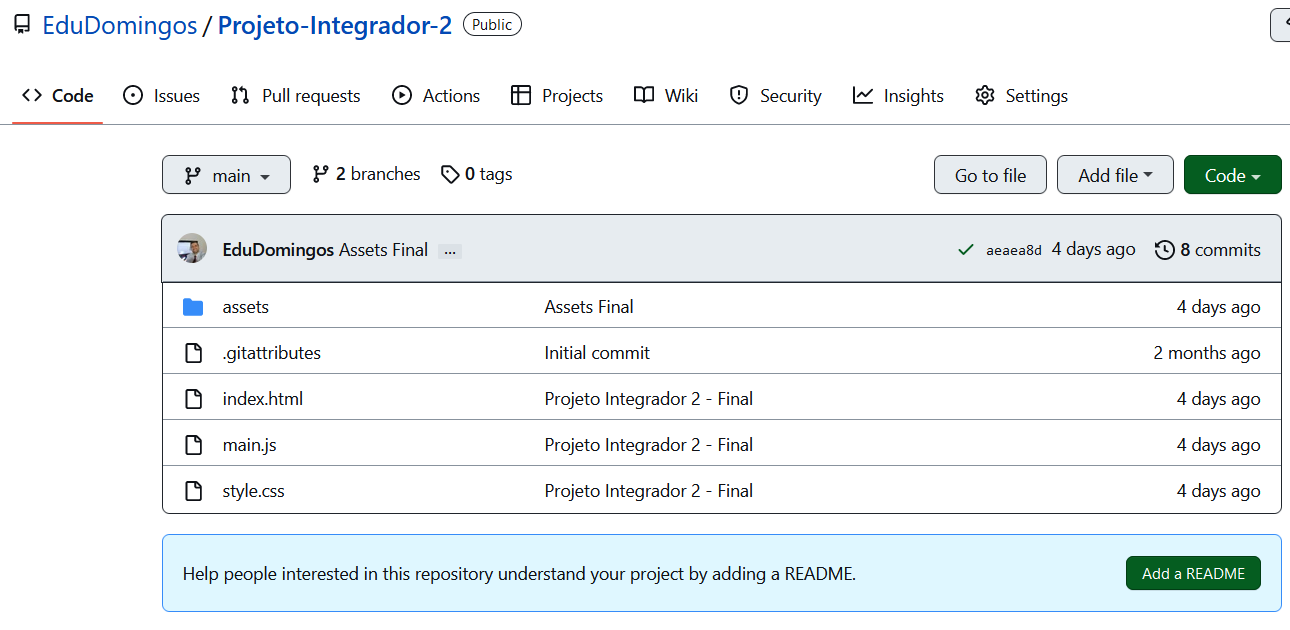
Ao invés da esteticista enviar o *link* diretamente para acesso ao serviço de agendamento, ela envia o *link* para a *Landing Page* à suas clientes, contendo informações sobre a profissional, os serviços prestados e contém um botão que direciona a cliente para o sistema de agendamento ou para o número de *WhatsApp* da profissional. Este botão, permite que a *Landing Page* consuma a API do *WhatsApp*.

INSERIR VÁRIOS PRINTS DA LANDING PAGE E LINK PARA ACESSO

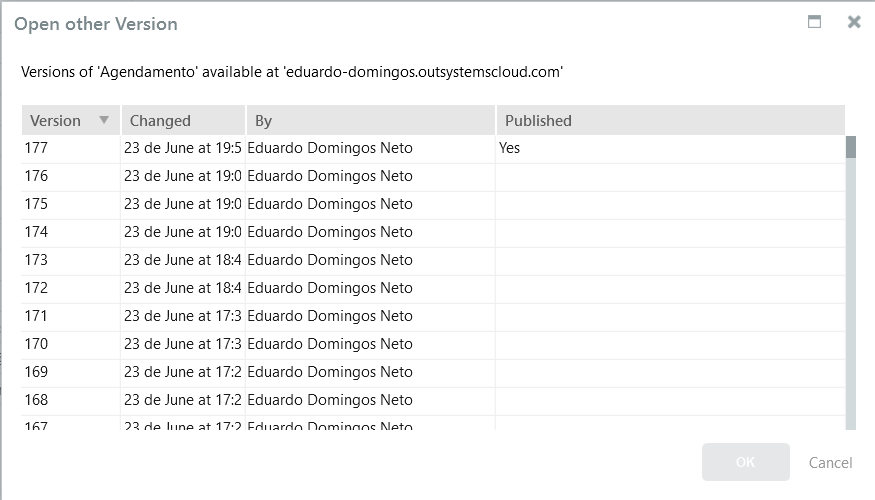
**2.5.6. Controle de Versão**

Controle de versão é a prática de rastrear e gerenciar alterações em um código-fonte que ajudam as equipes de software a gerenciar as alterações ao código-fonte ao longo do tempo.

Para o presente projeto foi escolhido utilizar o Git como controlador de versionamento local e o GitHub como repositório online, assim como o sistema de gerenciamento de versões intrínseco ao *framework* Outsystems, como vistos nas figuras L e M respectivamente.



(Figura L: Versionamento utilizando *Git/GitHub*)

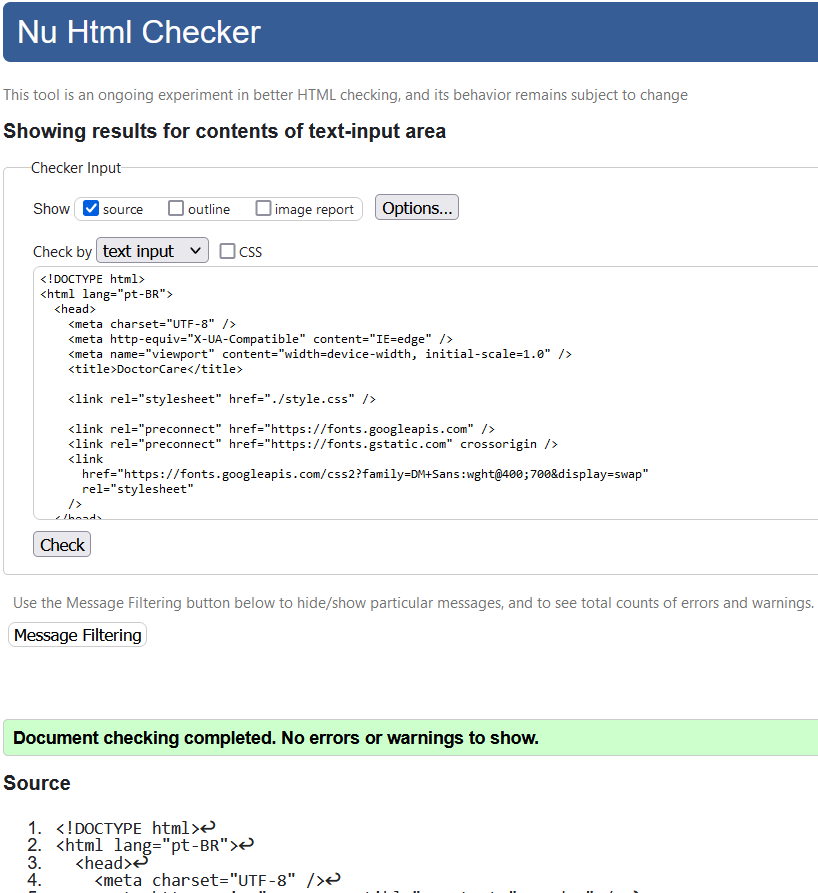
****

(Figura M: Versionamento intrínseco ao *framework* Outsystems)

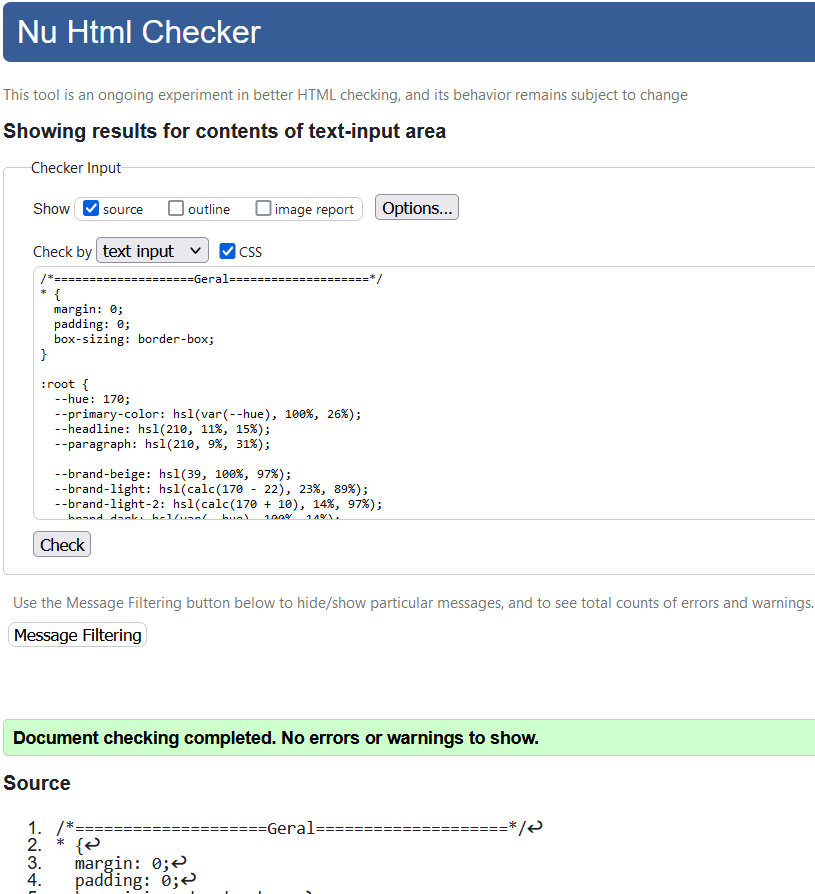
**2.5.7. Testes de Validação**

Para testar o *software*, ou seja, para encontrar eventuais problemas, investigar, analisar a garantir a qualidade da entrega, utilizamos o teste caixa preta com objetivo de validar as entradas e saídas do sistema. Nessa técnica, o testador busca garantir os requisitos funcionais do sistema interagindo com a aplicação e verificando o seu comportamento.

Para a verificação do código HTML e CSS utilizamos a ferramenta *W3CMarkup Validation Service*, assim como visto nas figuras N e O.



(Figura N: Validação do código HTML)



(Figura O: Validação do código CSS)

# 4. CONSIDERAÇÕES FINAIS

O Projeto Integrador tem como objetivo o desenvolvimento de um *software* com *Framework Web* que utilize banco de dados com *script web* utilizando a linguagem *JavaScript*, nuvem, uso de APi, acessibilidade, controle de versão e testes.

INFORMAR QUE CUMPRIMOS TODOS OS REQUISITOS...

Deve-se retomar os objetivos e o contexto em que o Projeto Integrador foi desenvolvido e apresentar:

* Retomada dos resultados à luz das referências estudadas;
* Contribuições e limitações do trabalho realizado;
* Impacto da solução na comunidade externa.

# REFERÊNCIAS

ABNT – Associação Brasileira de Normas Técnicas. **NBR 14724**: Informação e documentação. Trabalhos Acadêmicos - Apresentação**.** Rio de Janeiro: ABNT, 2002.

BOYER, C. B.; UTA, C. M. **História da Matemática** [Trad. Helena Castro]. 3 ed. São Paulo: Blucher, 2012.

D’AMBRÓSIO, U. **Educação Matemática:** da teoria à prática. 23. ed. Campinas: Papirus, 2012.

KUBO, O.; BOTOMÉ, S. **Ensino e aprendizagem:** uma interação entre dois processos comportamentais. Interação, v.5, p.123-32, 2001.

HART-DAVIS, A. **O Livro da Ciência.** 2. ed. São Paulo: Globo, 2016.

PILETTI, C. **Didática geral.** São Paulo: Ática, 1995.

RIBEIRO, J. L. P. Áreas e Proporções nas Superquadras de Brasília Usando o Google Maps. **Revista do Professor de Matemática**. Rio de Janeiro, n. 92, p. 12-15, jan-abr. 2017.

SEVERINO, A. J. **Metodologia do trabalho científico.** 22. ed. rev. e ampl. São Paulo: Cortez, 2002.

O trabalho deverá ser redigido conforme recomendações das Diretrizes para confecção de teses e dissertações da Universidade de São Paulo (USP), disponíveis em: <<http://www.teses.usp.br/index.php?option=com_content&view=article&id=52&Itemid=67>>. Acesso em 24 jun.2021.

# ANEXOS (opcional)

Materiais coletados por meio de pesquisas em diversas fontes.

O grupo pode anexar qualquer tipo de material ilustrativo, tais como tabelas, lista de abreviações, documentos ou parte de documentos, resultados de pesquisas etc.

# APÊNDICES (opcional)

Apêndices são criações do autor ou grupo de autores.

## Apêndice A –