**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Projeto Integrado

Relatório Técnico

Gestão Glamour

Wagner Martins Santana Vieira

Belo Horizonte

Agosto de 2023.

# Projeto Integrado

**Sumário**

[Projeto Integrado 2](#_Toc142558243)

[1. Cronograma de Trabalho 3](#_Toc142558244)

[2. Introdução 4](#_Toc142558245)

[3. Definição Conceitual da Solução 7](#_Toc142558246)

[3.1 Diagrama de Casos de Uso 8](#_Toc142558247)

[3.2 Requisitos Funcionais 9](#_Toc142558248)

[3.3 Requisitos Não-funcionais 10](#_Toc142558249)

[4. Protótipo Navegável do Sistema 11](#_Toc142558250)

[5. Diagrama de Classes de Domínio 12](#_Toc142558251)

[6. Apropriação de Horas no Projeto 13](#_Toc142558252)

## Cronograma de Trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 01 / 07 / 23 | 15 / 07 /23 | 1.Definição Conceitual | Criar o diagrama de casos de uso.  Listar os requisitos funcionais.  Identificar os requisitos não-funcionais. |
| 15 / 07 / 23 | 30 / 07 / 23 | 2.Protótipo Navegável do Sistema | Desenvolver um protótipo navegável da aplicação.  Criar telas básicas para a agenda e o financeiro. |
| 30 / 07 / 23 | 10 / 08 / 23 | 3.Diagrama de Classes de Domínio | Modelar o diagrama de classes de domínio para representar os principais objetos da aplicação. |
| 10 / 08 / 23 | 16 / 08 / 23 | 4.Arquitetura da Solução | Definir o padrão arquitetural a ser utilizado.  Criar o diagrama de contexto usando o C4 model. |
| 17 / 08 / 23 | 23 / 08 / 23 | 5.Frameworks de Trabalho | Escolher e configurar os frameworks necessários para o desenvolvimento |
| 24 / 08 / 23 | 30 / 08 / 23 | 6.Estrutura Base do Front End | Criar a estrutura base do front end da aplicação |
| 31 / 08 / 23 | 06 / 09 / 23 | 7.Modelo Relacional ou Projeto de Banco de Dados NoSQL | Definir o modelo de banco de dados relacional ou estrutura para banco de dados |
| 07 / 09 / 23 | 24 / 09 / 23 | 8.Desenvolvimento das Funcionalidades | Implementar a funcionalidade da agenda e financeiro e relatórios.  Conectar o sistema ao banco de dados. |
| 25 / 09 / 23 | 02 / 10 / 23 | 9.Plano de Testes | Elaborar um plano de testes para garantir a qualidade do software.  Realizar testes unitários para as principais funcionalidades. |
| 03 / 10 / 23 | 06 / 10 / 23 | 11.Avaliação Retrospectiva | Revisar o progresso do projeto.  Verificar se todos os objetivos foram alcançados. |
| 06 / 10 / 23 | 08 / 10 / 23 | 12.Ajustes e Finalização | Realizar os ajustes necessários na aplicação.  Preparar o sistema para o lançamento. |
| 09 / 10 / 23 | 10 / 10 / 23 | 13.Entrega | Disponibilizar todo o projeto. |

## Introdução

O segmento de salões de beleza desempenha um papel significativo na indústria de cuidados pessoais, atendendo a uma vasta clientela que busca serviços variados, desde cortes de cabelo e tratamentos estéticos até serviços de manicure e maquiagem. Com a crescente demanda por serviços de beleza e bem-estar, a eficiência operacional e a organização interna tornam-se fatores críticos para o sucesso e a competitividade desses estabelecimentos.

Este trabalho apresenta o desenvolvimento de um sistema de gestão específico para salões de beleza, visando otimizar suas operações e proporcionar uma experiência mais agradável tanto para os clientes como para os profissionais do salão. O objetivo principal do sistema é simplificar e centralizar o gerenciamento das atividades diárias, englobando áreas cruciais como agendamento de serviços e controle financeiro.

Um dos principais desafios enfrentados pelos salões de beleza é a gestão de agendamentos, que muitas vezes pode ser caótica e sujeita a conflitos de horários, levando à insatisfação dos clientes e à perda de receita. Além disso, o controle financeiro muitas vezes é complexo, dificultando a visualização do lucro real e a identificação de oportunidades de crescimento.

Para abordar essas questões, este sistema propõe uma solução abrangente que permitirá aos salões de beleza gerenciar sua agenda de forma eficiente, agendando e reagendando compromissos com facilidade, e possibilitando o controle detalhado das finanças, incluindo registros de receitas e despesas.

Ao fornecer ferramentas intuitivas e eficazes, o sistema visa liberar os proprietários e funcionários de tarefas administrativas repetitivas, permitindo que se concentrem no atendimento ao cliente e na oferta de serviços de qualidade, elevando, assim, a satisfação dos clientes e a reputação do salão.

Ao longo deste trabalho, serão apresentados os requisitos funcionais e não-funcionais do sistema, a arquitetura proposta, bem como o desenvolvimento do protótipo navegável do sistema, que permitirá visualizar como as funcionalidades serão implementadas na interface do usuário.

Dessa forma, o sistema de gestão para salão de beleza busca alavancar a tecnologia e a automação para aprimorar a eficiência operacional desses estabelecimentos, promovendo um ambiente organizado e profissional, capaz de atender às demandas do mercado competitivo de beleza e bem-estar.

Para contextualizar o presente trabalho, é essencial considerar o cenário em ascensão da indústria de beleza e cuidados pessoais, que tem sido impulsionada pela crescente demanda por serviços de beleza e estética. Os salões de beleza desempenham um papel crucial nesse contexto, oferecendo uma ampla gama de tratamentos e serviços para atender às necessidades dos clientes em busca de cuidados pessoais e bem-estar. No entanto, muitos desses estabelecimentos enfrentam desafios na organização interna, especialmente no gerenciamento de agendamentos e controle financeiro. A falta de uma solução tecnológica adequada pode resultar em conflitos de horários e perda de receita.

O problema que o projeto se propõe a resolver é a ineficiência na gestão dos salões de beleza, que muitas vezes enfrentam desafios na organização de agendamentos e no controle financeiro. A ausência de um sistema especializado pode levar a conflitos de horários, cancelamentos de última hora e dificuldades na visualização do fluxo de caixa, afetando diretamente a satisfação dos clientes e a rentabilidade do negócio. Além disso, a falta de uma visão clara do desempenho financeiro pode prejudicar a tomada de decisões estratégicas e dificultar o crescimento sustentável do salão.

A motivação para resolver o problema da gestão ineficiente dos salões de beleza é abrangente e justifica-se em diversas frentes. Sob o aspecto acadêmico, este projeto proporciona a oportunidade de desenvolver uma solução tecnológica inovadora, aplicando conceitos avançados de desenvolvimento de software ao setor em crescimento. Mercadologicamente, o sistema de gestão busca atender à demanda real do mercado, diferenciando os salões por meio de uma administração eficiente que pode levar à fidelização de clientes e aumento da rentabilidade. Socialmente, almeja-se elevar a qualidade dos serviços prestados, melhorando a experiência do cliente e proporcionando maior satisfação aos profissionais do salão. Com a expectativa de trazer economia de recursos, agilidade nas decisões e impactos positivos na qualidade de vida, a relevância deste trabalho é evidente, promovendo a excelência dos salões de beleza e contribuindo para o seu crescimento e sucesso no mercado competitivo de beleza e bem-estar.

O objetivo geral deste trabalho é desenvolver um sistema de gestão para salões de beleza que proporcione uma administração eficiente e organizada, visando otimizar o agendamento de serviços e o controle financeiro, a fim de aprimorar a experiência do cliente e fortalecer a competitividade desses estabelecimentos no mercado de beleza e bem-estar.

Os objetivos específicos são:

* Identificar os requisitos funcionais do sistema: Esta etapa envolve uma análise aprofundada das necessidades dos salões de beleza, com o objetivo de mapear e documentar os requisitos funcionais essenciais. Isso inclui entender as operações diárias, as interações com clientes, o fluxo de agendamentos, a gestão financeira e outros aspectos cruciais do negócio.
* Projetar a arquitetura do sistema: A arquitetura do sistema será elaborada de forma cuidadosa, definindo a estrutura geral, a disposição dos componentes e a interação entre eles. Serão estabelecidas as interfaces com o usuário, os mecanismos de comunicação e a lógica de negócios, garantindo uma base sólida para o desenvolvimento futuro.
* Desenvolver o protótipo navegável: A criação de um protótipo navegável permitirá visualizar antecipadamente a aparência e a usabilidade da aplicação. Isso auxiliará na validação das ideias, na coleta de feedback dos usuários e na identificação de melhorias necessárias, garantindo que a solução atenda às expectativas.
* Efetuar o desenvolvimento do sistema de gestão: Nesta fase, ocorrerá a implementação prática das funcionalidades principais, como o agendamento de serviços, o controle financeiro e a geração de relatórios. Serão aplicadas as melhores práticas de desenvolvimento, visando à eficiência, ao desempenho e à facilidade de manutenção do sistema.
* Realizar testes e validação do sistema: Uma bateria completa de testes será conduzida para assegurar a qualidade do sistema. Isso envolverá testes de funcionalidade, desempenho e segurança. Além disso, a validação será realizada em conjunto com os usuários reais, garantindo que o sistema atenda às suas necessidades e expectativas de maneira eficaz.

## Definição Conceitual da Solução

A abordagem proposta para esta solução busca uma transformação profunda na maneira como os salões de beleza operam e interagem com seus clientes. O desenvolvimento de uma aplicação web altamente sofisticada, construída com tecnologias de última geração, representa um avanço significativo na modernização desse setor.

A escolha deliberada do framework Spring Boot e da linguagem de programação Java para a implementação do sistema oferece inúmeras vantagens, desde a robustez até a facilidade de manutenção. Essa decisão estratégica visa garantir uma base sólida para a construção de uma aplicação que atenda às necessidades exclusivas dos salões de beleza, permitindo uma gestão eficiente e uma experiência excepcional para os clientes.

A arquitetura de micro serviços, amplamente reconhecida por sua flexibilidade e escalabilidade, desempenhará um papel crucial no desdobramento da solução. A modularidade inerente aos micros serviços permite a divisão dos componentes em unidades independentes, simplificando a complexidade e facilitando a manutenção contínua. Além disso, essa abordagem proporciona um alto grau de adaptabilidade, permitindo que a aplicação cresça e se adapte às mudanças dinâmicas do mercado.

A elaboração cuidadosa de requisitos funcionais e não-funcionais reflete o compromisso em entregar um sistema de alta qualidade e desempenho. O diagrama de casos de uso fornecerá uma visão clara das interações entre os usuários e o sistema, garantindo uma compreensão abrangente dos fluxos de trabalho essenciais. Ao definir requisitos funcionais específicos, o sistema será capaz de lidar de forma eficaz com funções essenciais, como agendamento de serviços, cadastro de clientes, controle financeiro e geração de relatórios.

Os requisitos não-funcionais, por sua vez, assegurarão que a aplicação seja executada de maneira otimizada, mantendo altos padrões de desempenho, segurança e escalabilidade. Toda essa estrutura conceitual serve como alicerce para o desenvolvimento do sistema de gestão para salões de beleza, que tem como objetivo primordial elevar a eficiência, competitividade e satisfação geral do cliente nesse mercado dinâmico e em constante evolução.

## Diagrama de Casos de Uso

Diagrama do caso de uso realizado em UML disponibilizado no github:   
[ProjetoIntegrado-WagnerMartins-DigramaCasoUso](https://github.com/wagnermartinssantana/projetoIntegrado/blob/main/docs/C4model/diagrama-caso-uso.puml).

|  |
| --- |
|  |

## Requisitos Funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| RF01 | O sistema deve permitir o cadastro e autenticação de administradores e clientes. | M | A |
| RF02 | Os administradores devem poder cadastrar, visualizar, editar e excluir informações de clientes, serviços e funcionários. | B | B |
| RF03 | Os clientes devem poder visualizar a disponibilidade de horários e agendar serviços. | A | M |
| RF04 | O sistema deve permitir o controle financeiro, registrando receitas e despesas e gerando relatórios. | B | A |
| RF05 | Deve haver um calendário interativo que permita ao administrador gerenciar os agendamentos. | A | M |
| RF06 | O sistema deve permitir que os clientes visualizem a lista de serviços oferecidos, incluindo descrição e preço. | B | A |
| RF07 | O sistema deve permitir que os administradores visualizem um calendário mensal com a agenda de todos os funcionários. | M | M |
| RF08 | O sistema deve calcular automaticamente o valor total a ser pago com base nos serviços agendados. | B | M |  |  |  |
| RF09 | Os administradores devem poder cadastrar novos serviços, incluindo nome, descrição e preço. | B | A |  |  |  |
| RF10 | O sistema deve permitir que os funcionários visualizem os detalhes dos serviços a serem realizados em cada agendamento. | B | A |  |  |  |
| RF11 | Os administradores devem poder gerar relatórios de desempenho de funcionários, incluindo número de atendimentos e serviços realizados. | M | M |  |  |  |
| RF12 | O sistema deve permitir que os clientes avaliem os serviços após o atendimento. | M | M |  |  |  |
| RF13 | Os administradores devem poder bloquear horários específicos no calendário, por exemplo, para feriados ou manutenções. | B | B |  |  |  |
| RF14 | Os clientes devem poder reagendar ou cancelar agendamentos com antecedência. | B | M |  |  |  |
| RF15 | Os administradores devem poder cadastrar informações detalhadas sobre cada funcionário como experiência e especializações. | B | B |  |  |  |
| RF16 | Os administradores devem poder cadastrar promoções especiais e descontos nos serviços. | B | M |  |  |  |
| RF17 | Os clientes devem poder visualizar as promoções ativas e aplicá-las ao agendar um serviço. | A | M |  |  |  |
| RF18 | O sistema deve permitir que os administradores visualizem um histórico completo de transações financeiras. | A | A |  |  |  |
| RF19 | O sistema deve gerar relatórios financeiros mensais com resumo de receitas e despesas. | B | M |  |  |  |
| RF20 | O sistema deve permitir que os administradores exportem relatórios financeiros em formato PDF ou planilha Excel. | B | M |  |  |  |
| RF21 | Os clientes devem poder visualizar os serviços oferecidos, incluindo descrição e preço. | M | A |  |  |  |
| RF22 | O sistema deve fornecer uma interface de busca para os clientes encontrarem facilmente os serviços desejados. | M | M |  |  |  |
| RF23 | O sistema deve permitir que os clientes visualizem as avaliações e comentários de outros clientes sobre os serviços e funcionários do salão. | A | M |  |  |  |
| RF24 | O sistema deve disponibilizar um histórico completo de transações financeiras, incluindo detalhes como data, valor, tipo de transação e agendamento associado. | A | M |  |  |  |
| RF25 | Os administradores devem ter a capacidade de atualizar as informações de contato do salão, como telefone e endereço. | B | A |  |  |  |

\* B = Baixa, M = Média, A = Alta.

## Requisitos Não-funcionais

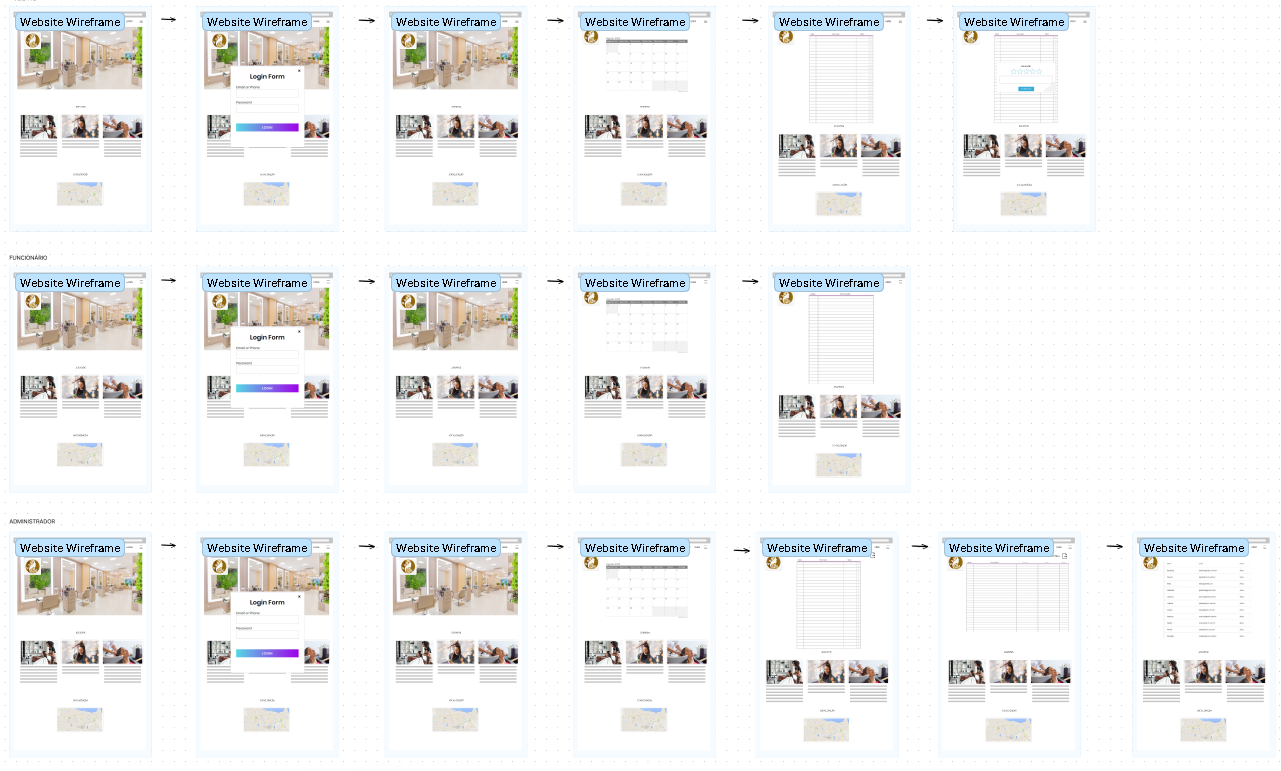
|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| RNF01 | O sistema deve ser seguro, protegendo os dados dos clientes e transações financeiras. | A |
| RNF02 | Deve ser de fácil utilização e possuir uma interface intuitiva. | M |
| RNF03 | Deve ser responsivo, adaptando-se a diferentes dispositivos (computador, tablet, smartphone). | M |
| RNF04 | O sistema deve ter alta disponibilidade, minimizando o tempo de inatividade. | A |
| RNF05 | Deve ser escalável, permitindo o aumento de usuários e funcionalidades sem perda de desempenho. | M |
| RNF06 | O sistema deve ter tempos de resposta rápidos, garantindo uma experiência ágil para os usuários durante a interação com a aplicação. | A |
| RNF07 | A aplicação deve ser desenvolvida seguindo boas práticas de programação e padronização de código. | M |
| RNF08 | A aplicação deve ser compatível com os principais navegadores web, como Chrome, Firefox, Safari e Edge. | M |

\* B = Baixa, M = Média, A = Alta.

## Protótipo Navegável do Sistema

Utilizado a ferramenta Figma, para a realização do protótipo que permitiu a criação de uma experiência interativa e a elaboração de wireframes precisos. O protótipo abrange a tela inicial da aplicação e demonstra três usuários distintos como cliente, funcionário e administrador e as suas principais funcionalidades.

Imagem utilizando o Figma:



O documento em PDF contendo o protótipo navegável e os wireframes está disponível no seguinte link: [ProjetoIntegrado-WagnerMartins.pdf](https://github.com/wagnermartinssantana/projetoIntegrado/blob/main/docs/prototipo/ProjetoIntegrado-WagnerMartins.pdf)

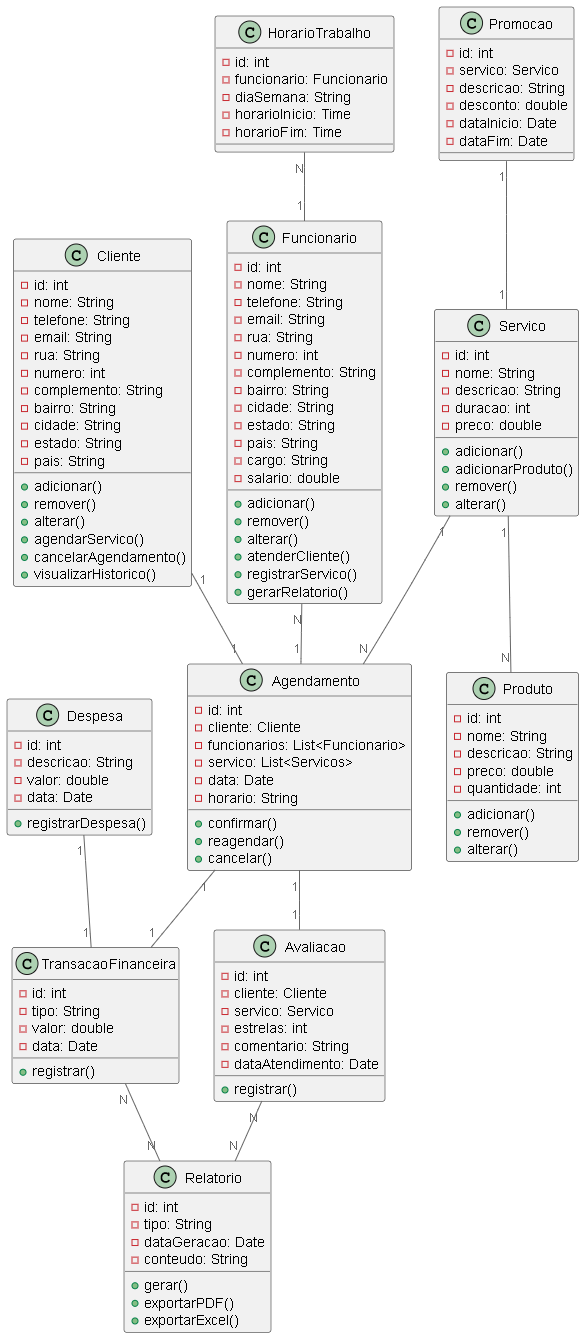
Foi realizado um vídeo de apresentação do protótipo navegável desenvolvido. O vídeo destaca os aspectos-chave do protótipo, demonstrando a experiência de navegação entre as telas e ressaltando os elementos interativos. O vídeo está disponível no formato MP4 e pode ser visto através do seguinte link no Google Drive: [ProjetoIntegrado-WagnerMartins.mp4](https://drive.google.com/file/d/1aePfFP--kkptOlBDxjcz_xpslPRryJO9/view?usp=sharing) ou no GitHub : [ProjetoIntegrado-WagnerMartins.mp4](https://github.com/wagnermartinssantana/projetoIntegrado/blob/main/docs/prototipo/ProjetoIntegrado-WagnerMartins.mp4)

O meu repositório contendo o protótipo navegável na extensão do Figma para ser visualizado está disponível no GitHub: [ProjetoIntegrado-WagnerMartins.fig.](https://github.com/wagnermartinssantana/projetoIntegrado/blob/main/docs/prototipo/ProjetoIntegrado-WagnerMartins.fig)

## Diagrama de Classes de Domínio

Diagrama de classes de domínio em UML registrado no github:

[ProjetoIntegrado-WagnerMartins-ClassesDominio](https://github.com/wagnermartinssantana/projetoIntegrado/blob/main/docs/C4model/diagrama-classe-dominio.puml)



## Apropriação de Horas no Projeto

|  |  |  |
| --- | --- | --- |
| **Histórico de apropriação de horas** | | |
| **Data do registro** | **Atividade** | **Quantidade de horas** |
| 01/07/2023 | Iniciado leitura do manual. | 3 horas |
| 02/07/2023 | Formulando o projeto. | 4horas |
| 08/07/2023 | Pesquisa e Coleta de Informações | 7 horas |
| 09/07/2023 | Finalização da Pesquisa | 6 horas |
| 15/07/2023 | Início da Introdução | 3 horas |
| 16/07/2023 | Continuação da Introdução | 6 horas |
| 17/07/2023 | Finalização Introdução | 2 horas |
| 22/07/2023 | Definição Conceitual da Solução | 5 horas |
| 23/07/2023 | Continuação da Definição Conceitual da Solução | 6 horas |
| 24/07/2023 | Continuação da Definição Conceitual da Solução | 3 horas |
| 25/07/2023 | Finalização da Definição Conceitual da Solução | 2 horas |
| 29/07/2023 | Diagrama de Classes de Domínio | 5 horas |
| 30/07/2023 | Continuação do Diagrama de Classes de Domínio | 4 horas |
| 05/08/2023 | Protótipo Navegável do Sistema | 6 horas |
| 06/08/2023 | Continuação do Protótipo Navegável do Sistema | 3 horas |
| 07/08/2023 | Continuação do Protótipo Navegável do Sistema | 5 horas |
| 11/08/2023 | Realizado o vídeo de apresentação e atualizado o projeto no GitHub | 5 horas |