**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Projeto Integrado

Relatório Técnico

Agendify

Marcos Martins da Silva

Vagner Freitas de Sousa

Belo Horizonte

Junho 2023

# Projeto Integrado

**Sumário**

Projeto Integrado 3

1. Cronograma de Trabalho 4

2. Introdução 5

3. Definição Conceitual da Solução 6

3.1 Diagrama de Casos de Uso 6

3.2 Requisitos Funcionais 6

3.3 Requisitos Não-funcionais 7

4. Protótipo Navegável do Sistema 7

5. Diagrama de Classes de Domínio 8

6. Arquitetura da Solução 8

6.1 Padrão Arquitetural 8

6.2 C4 model - Diagrama de Contexto 9

7. Frameworks de Trabalho 10

8. Estrutura Base do Front End 10

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

10. Plano de Testes 11

11. Apropriação de Horas no Projeto 11

12. Código da Aplicação 12

13. Avaliação Retrospectiva 12

13.1 Objetivos Estimados 12

13.2 Objetivos Alcançados 12

13.2 Lições aprendidas 12

14. Referências 13

## Cronograma de Trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 15 / 02 / 23 | 28 / 02 / 23 | 1.Formação dos objetivos e propostas do trabalho a ser apresentado. | Documentação de proposta técnica do sistema. |
| 05 / 03 / 23 | 20 / 03 / 23 | 2. Definição conceitual da aplicação | Diagramas de casos de uso |
| 21 / 03 / 23 | 02 / 04 / 23 | 3.Definição conceitual da aplicação | Requisitos funcionais e não funcionais do projeto |
| 05 / 04 / 23 | 05 / 05 / 23 | 4. Protótipo navegável do sistema | Protótipo inicial do projeto |
| 10 / 05 / 23 | 25 / 05 / 23 | 5. Diagrama de classes de domínio | Diagrama de classes de domínio |
| 30 / 05 / 23 | 05 / 06 / 23 | 6. Apropriação de horas do projeto | Documentação do tempo gasto no projeto. |
| 01/07/23 | 14/10/23 | 7. Desenvolvimento das funcionalidades | Código e implementação das funcionalidades planejadas |
| 15/10/23 | 15/10/23 | 8. Implantação do sistema | Deploy do sistema em ambiente cloud |

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

## Introdução

O projeto que a ser desenvolvido será uma plataforma de organização de agendamento de horários, que proverá um meio para que os estabelecimentos ou provedor de serviço divulgar seus horários disponíveis de atendimento individual, o cliente interessado irá acessar a página deste provedor para agendar o horário para a realização do atendimento, assim facilitando para ambos os lados organizar o horário de atendimento do serviço em si.

A plataforma terá uma interface que ficará disponível via web e será responsiva a maioria dos dispositivos que podem acessar a web, facilitando sua utilização seja ela via desktop ou em dispositivos mobile.

Com a popularização da internet e dispositivos móveis, cada vez mais estes vem sendo utilizados para facilitar o dia-a-dia e as tarefas mais simples, como consequência, sendo possível fazer mais no dia-a-dia a rotina da maioria da população se tornou cada vez mais corrida tornando o tempo o bem mais valioso, pois ter que agendar alguma prestação de serviço, consulta ao médico ou uma manicure pode ser trabalhoso ou até desgastante em ter de se comunicar com o estabelecimento e verificar um horário disponível e as vezes ter de aguardar o mesmo verificar a possibilidade, assim como da mesma forma pelo outro lado da moeda, o empreendedor que as vezes por ser pequeno não possui uma secretária ou alguém responsável por organizar a agenda dos seus atendimentos, o que pode ser dificultoso em situações que ao ser abordado por um possível cliente pode perder a agilidade ou pontualidade por não ter o controle dos seus horários, podendo perder uma recomendação do seu trabalho ou até mesmo não conseguir realizar um atendimento e perceber apenas próximo do horário agendado.

Este software tem como objetivo principal fornecer uma plataforma que possa facilitar ao máximo o agendamento de um atendimento com um prestador de serviços, de forma que seja feito o mínimo esforço possível para esta integração acontecer.

O prestador de serviço pode ser qualquer pessoa ou empresa que forneça um atendimento por horário agendado e precise ter este tipo de controle, que poderá criar um cadastro na plataforma e crie um perfil que descreva de forma breve seus serviços oferecidos e disponibilize sua agenda para assim um usuário interessado em um atendimento, poderá verificar diretamente os horários disponíveis de um prestador através de seu perfil, assim efetuando a reserva de horário, agilizando o evento do para ambos os lados, já que não é necessário ligação telefônica, tudo de forma simples e direta.

## Definição Conceitual da Solução

## Diagrama de Casos de Uso

|  |
| --- |
|  |

## Requisitos Funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| RF01 | Um Usuário deve ser capaz de cadastrar o seu perfil como Prestador ou Cliente | M | A |
| RF02 | Um Usuário deve ser capaz de receber e-mails de atualizações | B | B |
| RF03 | Um Cliente deve ser capaz de pesquisar vários prestadores | A | A |
| RF04 | Um Cliente deve ser capaz de acessar perfil do prestador | B | A |
| RF05 | Um Cliente deve ser capaz de ver a agenda de um prestador | M | A |
| RF06 | Um Cliente deve ser capaz de fazer agendamento com um prestador | M | A |
| RF07 | Um Cliente deve ser capaz de cancelar agendamento com um prestador | M | A |
| RF08 | Um Cliente deve ser capaz de alterar agendamento com um prestador | A | A |
| RF09 | Um Cliente deve ser capaz de ver os agendamentos criados. | M | A |
| RF10 | Um Prestador deve ser capaz de ver seus agendamentos | B | A |
| RF11 | Um Prestador deve ser capaz de cancelar um agendamento | M | A |
| RF12 | Um Prestador deve ser capaz de alterar um agendamento | A | A |
| RF13 | Um Prestador deve ser capaz de criar o seu perfil de apresentação de prestador | M | A |
| RF14 | Um prestador deve ser capaz de alterar o seu perfil de apresentação de prestador | M | A |
| RF15 | Um prestador deve ser capaz de criar a agenda de disponibilidade | A | A |
| RF16 | Um prestador deve ser capaz de alterar a agenda de disponibilidade | A | A |
| RF17 | Um Cliente deve ser capaz de confirmar o comparecimento em um agendamento | M | A |
| RF18 | Um Prestador deve ser capaz de ver os seus agendamentos por data | B | A |
| RF19 | Um usuário não pode alterar se é Cliente ou Prestador após ter finalizado o cadastro | B | A |
| RF20 | O Sistema deve apresentar uma criação de cadastro individual para o Prestador e o Cliente | M | A |
| RF21 | Um Agendamento não pode ser removido, apenas cancelado. | M | A |

\* B = Baixa, M = Média, A = Alta.

**Observação: acrescente quantas linhas forem necessárias.**

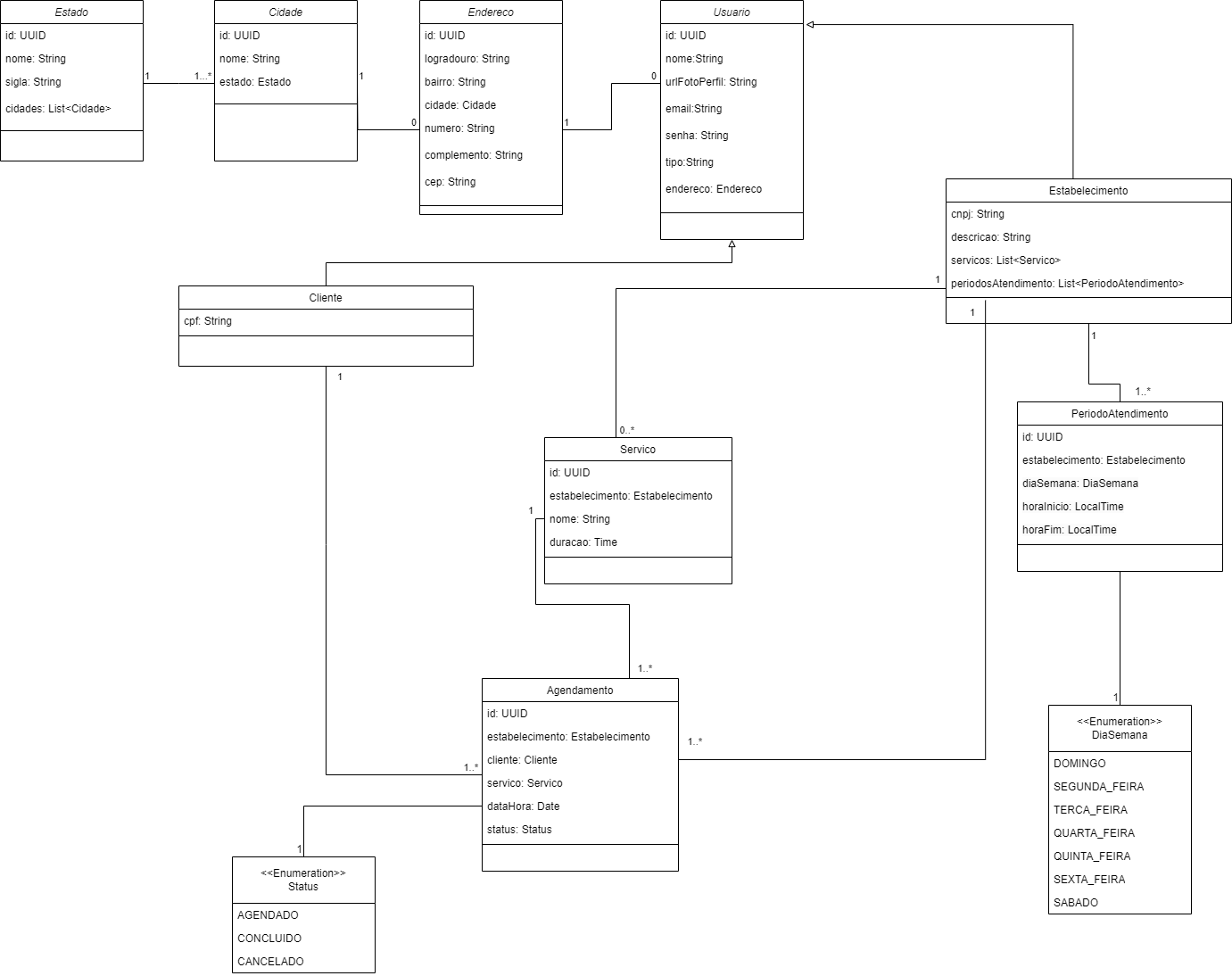
## Requisitos Não-funcionais

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| RNF01 | O sistema deve apresentar responsividade em todas as suas páginas | A |
| RNF02 | As informações do usuário devem ser armazenadas de forma segura e protegidas contra acesso não autorizado. | A |
| RNF03 | O Sistema deve estar disponível em qualquer período, todos os dias. | A |
| RNF04 | O sistema deve ser atualizado regularmente para corrigir erros e bugs, e melhorar sua eficiência e desempenho. | A |
| RNF05 | O sistema deve ser escalável para lidar com o aumento do tráfego e demanda. | A |
| RNF06 | O sistema deve ser capaz de suportar muitos usuários simultâneos. | M |

## Protótipo Navegável do Sistema

* Abaixo seguem os links de apresentação do protótipo navegável:
* Vídeo de apresentação:
* <https://drive.google.com/file/d/1_aU2JDHbsnIKbkOhqM1IJ9L5E9wluHLE/view?usp=drive_link>
* WireFrames no Figma:
* <https://www.figma.com/proto/pCSIVfdd6Me4gML3qkjM9r/Agendify?node-id=13-178&starting-point-node-id=13%3A178>

## Diagrama de Classes de Domínio

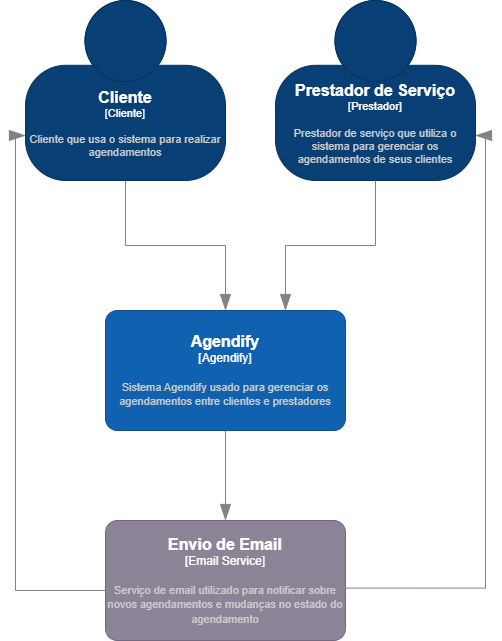


## Arquitetura da Solução

## Padrão Arquitetural

Utilizamos o Padrão Arquitetural MVC organizado em um Monolito Modular, onde a aplicação é dividida em módulos com responsabilidades e testáveis de forma independente, podendo ser facilmente atualizados ou substituídos sem afetar outros módulos, mas com repositório único e com distribuição em um único pacote mesmo usando tecnologias distintas no Backend e Frontend.

## C4 model - Diagrama de Contexto



## Frameworks de Trabalho

* Backend
  + Java 17
  + Spring Boot
  + Spring Data
  + Spring Security
  + Spring MVC
* Frontend
  + Angular
  + TypeScript
  + Bootstrap
  + Angular Material
* Banco de Dados
  + MySQL
* Infra
  + Azure
  + Azure Aplication Services
  + Docker

## Estrutura Base do Front End

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamenteInterface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamenteInterface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamenteInterface gráfica do usuário, Site

Descrição gerada automaticamente

## Modelo Relacional ou Projeto de Banco de Dados NoSQL

Diagrama

Descrição gerada automaticamente

## Plano de Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  | Alterar Perfil de Prestador | Alterar informações do perfil do Prestador de Serviços | - Realizar O processo de login no sistema com uma conta do tipo prestador de serviços - Acessar as configurações da conta/perfil clicando na foto do avatar no canto direto acima na barra de navegação - alterar descrição e serviços - Clicar em Confirmar Mudanças | - O sistema emite uma mensagem de sucesso informando que as informações foram salvas com sucesso. - O Sistema retorna para a tela inicial de agendamentos - Ao retornar a tela de alteração de dados os dados alterados estarão presentes conformes as alterações feitas |
|  | Alterar Perfil de Prestador | Alterar email existente do Perfil do Prestador de Serviços para um email inválido | - Realizar O processo de login no sistema com uma conta do tipo prestador de serviços - Acessar as configurações da conta/perfil clicando na foto do avatar no canto direto acima na barra de navegação - alterar email existente e inserir o email fulano@host - Clicar em Confirmar Mudanças | - O sistema emite uma mensagem de erro informando que as informações que o email informado é inválido. - O Sistema permanece na tela de configurações da conta/perfil |
|  | Fazer Agendamento | Criar agendamento a partir do Perfil Cliente | - Realizar o processo de login no sistema com uma conta do tipo Cliente - Acessar a tela principal - Na barra de busca inserir “Dentista” - Clicar no primeiro perfil retornado e clicar em “Agendar Atendimento” - Selecionar o serviço de consulta, a data e horário disponível e clicar em “Confirmar Agendamento” | - O Sistema deverá mostrar uma mensagem de sucesso informando que foi possível concluir o agendamento - O Sistema deverá redirecionar o usuário a tela principal, mostrando os agendamentos criados por este usuário. |
|  | Fazer Agendamento | Verificar que não é possível agendar atendimento com um Prestador que não possui agenda disponível. | - Realizar o processo de login no sistema com uma conta do tipo Cliente - Acessar a tela principal - Na barra de busca inserir “Dentista” - Clicar no primeiro perfil retornado | - O sistema deverá exibir as informações de perfil do prestador - O botão “Agendar Atendimento” deverá estar apagado e não realizar nenhuma ação. - Deverá ser exibida no perfil uma mensagem indicando que este prestador não possui horários disponíveis - O sistema permanecerá nesta tela - O usuário deverá manualmente voltar a tela anterior clicar fora do modal exibido |
|  | Cancelar agendamento (Cliente) | Cancelar agendamento criado pelo Perfil Cliente | - Realizar o processo de criar agendamento a partir do perfil Cliente - Na tela principal localizar o agendamento criado e clicar no ícone de X localizado no lado direito do agendamento. - Será apresentada uma mensagem de confirmação, clicar em Sim | - O sistema deverá exibir uma mensagem que o cancelamento foi efetuado com sucesso - O Sistema deverá redirecionar o usuário a tela principal, mostrando os agendamentos criados por este usuário. - O Agendamento cancelado deverá exibir o status Cancelado. |
|  | Cancelar agendamento (Cliente) | Cancelar agendamento já confirmado | - Realizar o processo de criar agendamento a partir do perfil Cliente - No perila de Prestador de serviço, localizar o agendamento criado e clicar no ícone de “V” e em seguida clicar em confirmar na tela seguinte. - No perfil Cliente na tela principal, localizar o agendamento criado e clicar no ícone de X localizado no lado direito do agendamento. - Será apresentada uma mensagem de confirmação, clicar em Sim | - O sistema deverá exibir uma mensagem que o cancelamento não foi efetuado. - O Sistema deverá redirecionar o usuário a tela principal, mostrando os agendamentos criados por este usuário. - O Agendamento deverá permanecer presente na tela inicial. |

## Apropriação de Horas no Projeto

|  |  |  |
| --- | --- | --- |
| **Histórico de apropriação de horas** | | |
| **Data do registro** | **Atividade** | **Quantidade de horas** |
| 20/02/23 - 05/03/23 | Formação dos objetivos e propostas do trabalho a ser apresentado. | 15 horas |
| 10/03/23 - 25/03/23 | Definição conceitual da aplicação - Diagramas de casos de uso. | 25 horas |
| 03/04/23 - 15/04/23 | Definição conceitual da aplicação - Requisitos funcionais e não funcionais do projeto. | 35 horas |
| 20/04/23 - 10/05/23 | Protótipo navegável do sistema - Protótipo de alta fidelidade inicial do projeto. (Figma) | 80 horas |
| 15/05/23 - 05/06/23 | Diagrama de classes de domínio. | 30 horas |
| 01/08/23 – 10/08/23 | Plano de Teste e Arquitetura da solução | 30 horas |
| 05/07/23- 01/08/23 | Desenvolvimento dos módulos base e classes de domínio | 80 horas |
| 01/08/23-06/08/23 | Desenvolvimento da camada de segurança da aplicação | 30 horas |
| 08/08/23-23/08/23 | Desenvolvimento das telas de criação de cadastro | 40 horas |
| 23/08/23- 12/09/23 | Desenvolvimento das telas e fluxo de login | 50 horas |
| 12/09/23 – 14/10/23 | Desenvolvimento das telas e fluxo de agendamento | 200 horas |
| 14/10/23- 15/10/23 | Implantação em serviço de cloud | 5 horas |
|  |  |  |

## Código da Aplicação

Repositório da Aplicação:

<https://github.com/MarcosMart1ns/Agendify>

Link de acesso a aplicação:

[agendify.azurewebsites.net/](https://agendify.azurewebsites.net/)

Usuários prontos para testar as funcionalidades:

|  |  |  |
| --- | --- | --- |
| Tipo | Login | senha |
| Cliente | [joao@email.com](mailto:joao@email.com) | Senha1234 |
| Estabelecimento | [barber@email.com](mailto:barber@email.com) | Senha1234 |

Vídeo de apresentação:  
<https://sgapucminasbr-my.sharepoint.com/personal/1410472_sga_pucminas_br/_layouts/15/stream.aspx?id=%2Fpersonal%2F1410472%5Fsga%5Fpucminas%5Fbr%2FDocuments%2FTcc%2FAgendify%2Ddemonstra%C3%A7%C3%A3o%2Emp4&ga=1>

Ou

<https://drive.google.com/file/d/16OyR0iZqnkLvbpXfXg_cy975PzSZfqbJ/view>

## Avaliação Retrospectiva

Ao longo da elaboração e desenvolvimento do projeto houve diversas oportunidades de aprendizado, desafios superados e lições aprendidas, mesmo com poucas funcionalidades, mas que nos fizeram crescer muito em tão pouco tempo.

## Objetivos Estimados

O objetivo era entregar as funcionalidades da forma mais objetiva pensando em aplicar os conhecimentos de arquitetura e de outras diversas áreas que contribuem para o desenvolvimento de software, assim como trazer tecnologias e implementações que nos fizessem aprender com a implementação, sem deixar de lado entregar o projeto funcionando em um fluxo completo de criação e interação do agendamento dos dois tipos de usuário.

## 13.2 Objetivos Alcançados

O objetivo de entregar um fluxo de criação de agendamento para os dois tipos de usuários planejados para este sistema foi atendido, assim como conseguimos adicionar funcionalidades extras como autenticação, atualização de status de forma agendada e alteração dos dados do usuário.

## Lições aprendidas

|  |  |  |
| --- | --- | --- |
|  | **Retrospectiva (Lições Aprendidas)** | |
|  | **Descrição da Lição** | **Classificação** |
| 1 | Planejamento e Execução dos cronogramas | Positiva |
| 2 | Implementação dos conceitos do curso ao projeto | Positiva |
| 3 | Implementação de frameworks e Bibliotecas que facilitem o desenvolvimento | Positiva |
| 4 | Decisões de mudança de arquitetura ou tecnologias | Positiva |
| 5 | Aprendizado em novas abordagens por tecnologias já usadas pelos integrantes do grupo | Positiva |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |