**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Projeto Integrado

Relatório Técnico

Sistema de agendamento médico

Daniel Augusto Bastos

Rio de Janeiro

Agosto de 2024.

# Projeto Integrado

**Sumário**

Projeto Integrado 3

1. Cronograma de Trabalho 4

2. Introdução 5

3. Definição Conceitual da Solução 6

3.1 Diagrama de Casos de Uso 6

3.2 Requisitos Funcionais 6

3.3 Requisitos Não-funcionais 7

4. Protótipo Navegável do Sistema 7

5. Diagrama de Classes de Domínio 8

6. Arquitetura da Solução 8

6.1 Padrão Arquitetural 8

6.2 C4 model - Diagrama de Contexto 9

7. Frameworks de Trabalho 10

8. Estrutura Base do Front End 10

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

10. Plano de Testes 11

11. Relatório de Execução de Testes de Software 11

12. Apropriação de Horas no Projeto 11

13. Código da Aplicação 12

14. Avaliação Retrospectiva 12

14.1 Objetivos Estimados 13

14.2 Objetivos Alcançados 13

14.3 Lições aprendidas 13

15. Referências 13

## Cronograma de Trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 19 / 08 / 24 | 19 / 08 / 24 | 1. Assistindo apresentação | Entendimento do projeto |
| 20 / 08 / 24 | 21 / 08 / 24 | 1. Planejamento do projeto | Plano de ação |
| 22 / 08 / 24 | 24 / 08 / 24 | 1. Escrevendo a introdução | Texto de introdução |
| 24 / 08 / 24 | 24 / 08 / 24 | 1. Análise das ferramentas de uso | Uso do Astah e PlantUML |
| 25 / 08 / 24 | 25 / 08 / 24 | 1. Requisitos funcionais/não-funcionais | requisitos |
| 26 / 08 / 24 | 27 / 08 / 24 | 1. Diagrama de caso de uso | Diagrama de caso de uso |
| 29 / 08 / 24 | 29 / 08 / 24 | 1. Diagrama de contexto | Diagrama de contexto |
| 31 / 08 / 24 | 31 / 08 / 24 | 1. Definição da arquitetura da solução | A arquitetura da solução |
| 01 / 09 / 24 | 03 / 09 / 24 | 1. Protótipo navegável da solução | Protótipo navegável |
| 05 / 09 / 24 | 06 / 09 / 24 | 1. Plano de testes | Plano de testes |
| 08 / 09 / 24 | 02 / 10 / 24 | 1. Implementação da solução | Versão inicial da aplicação |
| 05 / 10 / 24 | 05 / 10 / 24 | 1. Gravação do vídeo de apresentação | O vídeo |
| 06 / 10 / 24 | 09 / 10 / 24 | 1. Pequenas correções na aplicação | Correção de bugs |
| 12 / 10 / 24 | 13 / 10 / 24 | 1. Textos sobre retrospectiva do projeto | Os textos sobre avaliação retrospectiva, Objetivos alcançados |
| 14 / 10 / 24 | 16 / 10 / 24 | 1. Deploy da aplicação e carga de massa de dados | Aplicação no ar |
| 17 / 10 / 24 | 17 / 10 / 24 | 1. Entrega | Entrega da tarefa |

## Introdução

2.2 – Os objetivos

Projeto de análise de um sistema de agendamento, focado em atendimentos médicos, que ajude tanto ao paciente quanto o profissional de saúde na gerência dos horários e marcações.

2.2 – O problema

Frustração em atendimentos é bastante comum. Atendimentos, mesmo com horas marcadas, é comum existir falhas. Ninguém gosta de esperar, seja o atendente, seja o atendido. Em atendimentos médicos, onde o agendamento é essencial, é uma frustração maior.

Pensando do lado dos profissionais de saúde e seus atendentes, é bastante comum que seus atendidos tenham problemas de esquecimento, levando a faltas que poderiam ser evitadas e vacâncias no atendimento.

Pensando pelo lado dos atendidos, ocorrem imprevistos nos atendimentos. Um atendimento de emergência, ou mesmo um problema pessoal do profissional de saúde pode ocorrer, e isso vai um atraso ou cancelamento do atendimento. Isso gera frustração.

E em ambos os casos, a falta de visibilidade dos atendimento será um problema. Um atendido não sabe se seu atendimento pode atrasar bem como um profissional de saúde não sabe como esta a próxima fila de atendimentos.

Para ambos os lados, dada as suas diferentes particularidades, problemas de atendimento são frustrantes. Muitos destes casos podem ser mitigados com um sistema que automatize e

Se por um lado, é comum que pacientes esqueçam seus compromissos, levando a faltas que poderiam ser evitadas, por outro, profissionais de saúde, com um problema pessoal, pode gerar frustração em uma série de atendimentos.

Do lado do atendido, existe alguns problemas como a verificação de disponibilidade, agendamento, reagendamento e cancelamento e confirmações.

Do lado do atendente, existe o problema de gerência da agenda, gerência de vacância, agendamento, reagendamento e confirmações.

2.3 – Os Objetivos

Pensando nestes problemas, a automação do agendamento melhora a experiência dos envolvidos.

Permitir que o atendente gerencie o quadro de atendimento.

Permitir que o atendido marque, remarque ou cancele seus agendamentos.

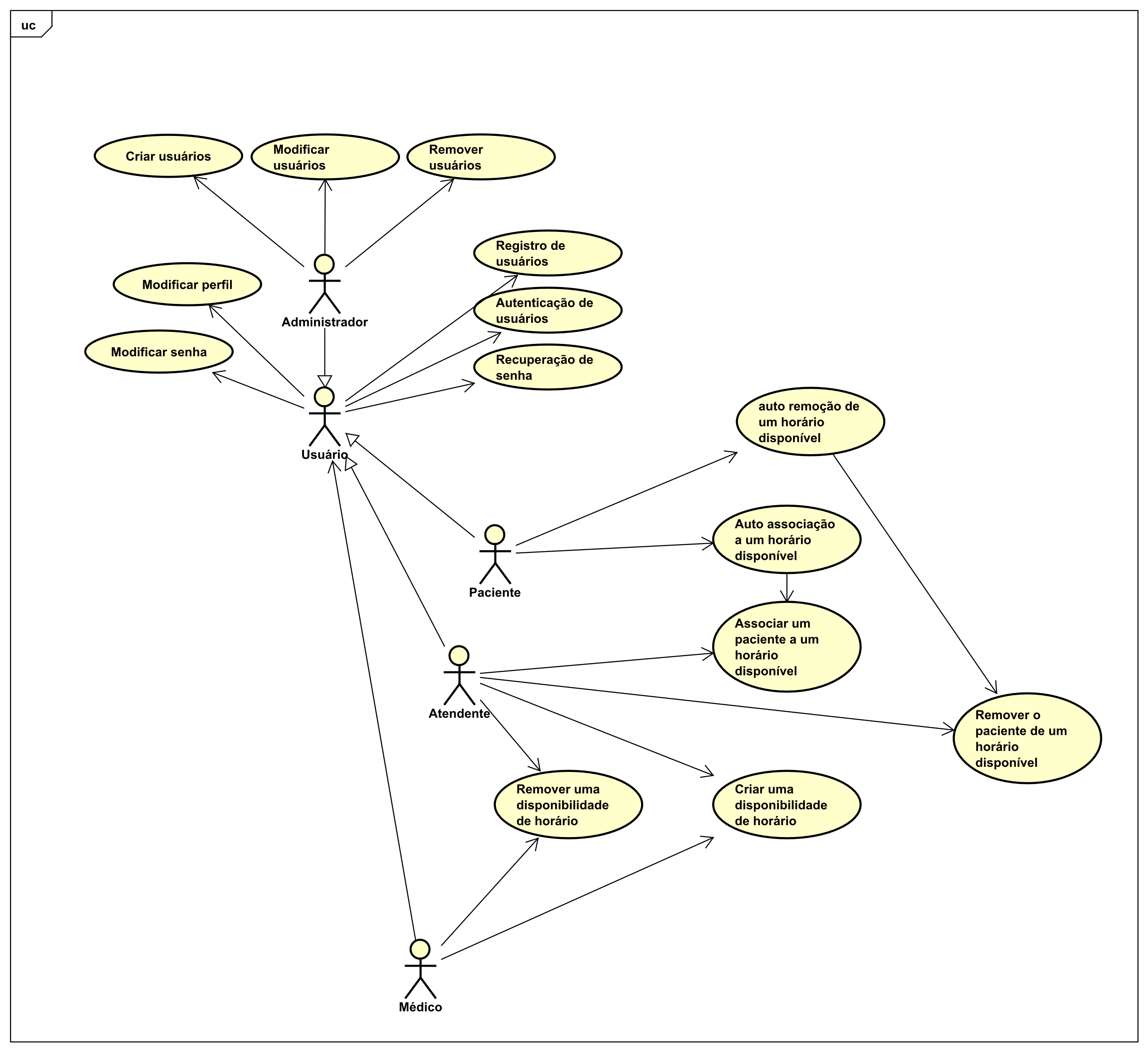
Permitir que o atendente marque, remarque ou cancele atendimentos em nome do atendido.

Notificar o atendimento mudanças feitas pelos atendidos.

Notificar o atendido sobre mudanças feitas pelo atendente.

## Definição Conceitual da Solução

## Diagrama de Casos de Uso



## Requisitos Funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descrição Resumida** | **Dificuldade (B/M/A)\*** | **Prioridade**  **(B/M/A)\*** |
| RF01 | O usuário deve auto cadastrar-se no sistema. O usuário cadastrado deve ser sempre do tipo “paciente” | B | A |
| RF02 | O administrador deve ser capaz de criar usuários dos tipos: “médico” e “atendente”. | B | A |
| RF03 | “médico” deve ser capaz de criar horários para as próprias consultas. | B | A |
| RF04 | “atendente” deve ser capaz de criar horários para consultas de qualquer médico. |  |  |
| RF05 | “médico” deve ser capaz de remover horários para as próprias consultas. | B | B |
| RF05 | “atendente” deve ser capaz de remover horário para qualquer médico. | B | M |
| RF06 | “atendente” deve ser capaz de associar horário disponível de qualquer médico a qualquer paciente. | B | M |
| RF07 | “médico” deve ser capaz de associar seu próprio horário a qualquer paciente. |  |  |
| RF07 | “paciente” deve ser capaz de associar ele mesmo a qualquer horário disponível a qualquer médico. | B | M |
| RF08 | “atendente” deve ser capaz de remover uma consulta de um médico qualquer a um paciente qualquer. | B | M |
| RF09 | “médico” deve ser capaz de remover uma consulta nos seus horários a qualquer paciente. | B | M |
| RF10 | “paciente” deve ser capaz de remover as consulta a um médico qualquer. | B | M |

\* B = Baixa, M = Média, A = Alta.

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

## Requisitos Não-funcionais

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Prioridade**  **B/M/A** |
| RNF01 | O software deve permitir acesso de qualquer local a qualquer hora. | A |
| RNF02 | O software deve ser fácil de manter e atualizar, permitindo a  adição de novas funcionalidades sem interromper o serviço. |  |
| RNF03 | O software não registrará senhas em texto puro. Toda a senha deve ser salva hasheada e com salt. |  |
| RNF04 | O sistema deve ser protegido contra acesso não autorizado e garantir a  confidencialidade dos dados dos pacientes. |  |
| RNF05 | O sistema deve manter registros detalhados de todas as operações realizadas,  para facilitar auditorias e rastreamento de problemas. |  |

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

## Protótipo Navegável do Sistema

A ferramenta de de wireframes escolhida foi o Pencil. O arquivo original do protótipo se encontra no repositório. Existe uma exportação do wireframes, em html, para apresentação. É essa exportação que foi usada no vídeo.

As informações de acesso estão abaixo.

Repositório de código: [Repositório](https://github.com/danielbastos/puc-agendamento)

Arquivo original do Pencil: [Arquivo do Pencil](https://github.com/danielbastos/puc-agendamento/blob/main/docs/Prot%C3%B3tipo%20naveg%C3%A1vel%20do%20sistema.epgz)

Exportação do protótipo: [Protótipo em HTML](https://github.com/danielbastos/puc-agendamento/tree/main/docs/Prot%C3%B3tipo%20naveg%C3%A1vel%20do%20sistema%20-%20export)

Vídeo de apresentação do protótipo: [Video](https://github.com/danielbastos/puc-agendamento/blob/main/docs/Prot%C3%B3tipo%20naveg%C3%A1vel%20do%20sistema%20%E2%80%90%20PUC.mp4)

## Diagrama de Classes de Domínio

Diagrama

Descrição gerada automaticamente

## Arquitetura da Solução

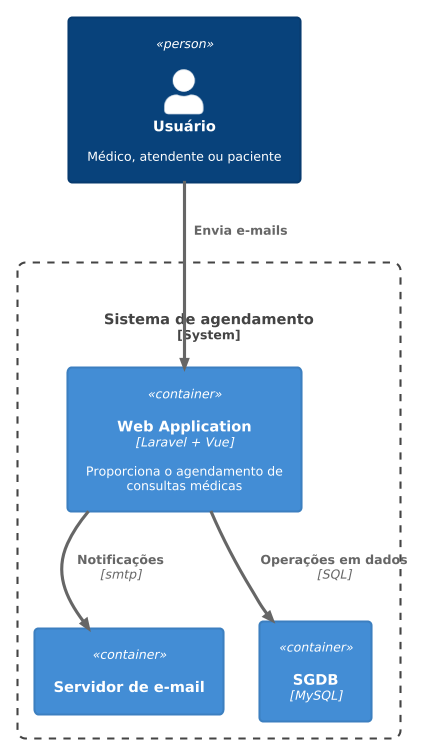
## Padrão Arquitetural

O padrão MVC foi utilizando. Seja pela simplicidade da aplicação.

As tecnologias utilizadas são:

* Backend
  + Laravel como base da aplicação
  + Node como auxiliar para builds das views
  + MySQL como SGDB
* Frontend
  + Vue para interatividade na aplicação.
  + TailwildCss para design
  + Flowbite como suíte de componentes para Vue.
* Infraestrutura
  + Docker para desenvolvimento local
  + Heroku para produção

## C4 model - Diagrama de Contexto

O diagrama apresenta a divisão do servidor e dos containers utilizados pela aplicação.

Aqui podemos ver que o usuário acessa a aplicação responsável pela visão.

Consequentemente a aplicação se comunica com uma base de dados, onde é persistido a aplicação.

Também podemos ver que ela se comunica com um servidor SMTP, responsável pelo envio de e-mails.

Apesar do servidor de SMTP não estar necessariamente no escopo da aplicação, o ambiente de desenvolvimento contem um SMTP fake para fins de testes, por isso optei por inclui-lo.

## Frameworks de Trabalho

Os frameworks utilizados são o Laravel no backend, fazendo o roteamento e a estrutura de controllers, o Elloquent, já integrado ao laravel, fazendo a persistência.

Na fase de desenvolvimento é utilizado o node + vite, responsável por fazer o Build do frontend e disponibilizar a visão ao Laravel, além do Sail, para auxiliar no desenvolvimento dos containers Docker.

## Estrutura Base do Front End

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

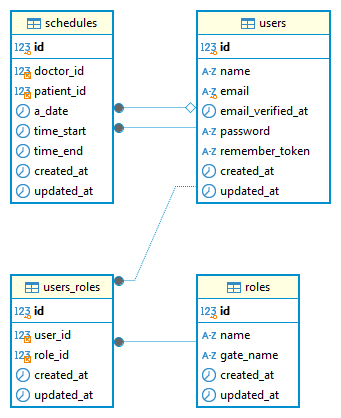
Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

## Modelo Relacional ou Projeto de Banco de Dados NoSQL



## Plano de Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  | [Médico/Atendente] criar disponibilidade de horário | Adicionar horários de atendimento disponíveis em uma data específica | Data, horário de início e de fim | Uma lista de horários de atendimento, divididos em intervalos de 30 minutos adicionados a disponibilidade de horários. |
|  | [Médico/Atendente] remover disponibilidade de horários | Remover um horário específico de atendimento | Clicar no botão remover da lista de horários disponíveis | A remoção do horário disponível |
|  | [Médico/Atendente] tentar remover um horário disponível que esteja associado a um paciente | Tentar remover um horário que já tenha sido associado a um paciente | Tentar clicar no botão de remover o horário na lista de horários disponíveis | O botão não deve aparecer |
|  | [Paciente] criar disponibilidade de horário | Adicionar horários de atendimento disponíveis em uma data específica | Tentar horários de atendimento | O botão adicionar não deve aparecer |
|  | [Paciente] remover disponibilidade de horários | Remover um horário específico de atendimento de outro paciente | Clicar no botão remover da lista de horários disponíveis | O botão remover não deve aparecer |
|  | [Paciente] tentar remover um horário disponível que esteja associado a um paciente | Tentar remover um horário que já tenha sido associado a um paciente | Tentar clicar no botão de remover o horário na lista de horários disponíveis | O botão não deve aparecer |
|  | [Paciente] remover disponibilidade de horários | Remover um horário específico de atendimento próprio | Clicar no botão remover da lista de horários disponíveis | Remover o agendamento |

## Relatório de Execução de Testes de Software

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de teste** | **Saída esperada** | **Resultados encontrados** | **Aprovado?** |
|  | Como médico, registrar novos horários de atendimento | Voltar para a pesquisa com os horários cadastrados. | Retorno da pesquisa com os horários cadastrados | SIM |
|  | Como atendente, registrar novos horários | Voltar para a pesquisa com os horários cadastrados. | Retorno da pesquisa com os horários cadastrados | SIM |
|  | Como médico, registrar novos horários de atendimento para outro médico | Não ser possível mudar o médico no cadastro de horários | A impossibilidade de mudar de médico no cadastro de horários | SIM |
|  | Como médico, registrar novos horários de atendimento para outro médico | Não ser possível cadastrar horários | Impossibilidade de cadastrar horários | SIM |
|  | Como paciente, associar uma consulta | Só é possível e o horário esteja liberado e não ocupado | Impossível cadastrar em um horário que esteja ocupado | SIM |
|  | Como paciente, cancelar uma consulta | Aparecer um botão de cancelamento para uma consulta que já foi previamente marcada | Só é possível cancelar seu próprio agendamento | SIM |
|  | Como médico, remover uma disponibilidade de horário que esteja ocupado | Não aparecer o botão de remoção, sendo obrigado o cancelamento prévio | O botão de remoção não aparece | SIM |
|  | Como médico, remover uma disponibilidade de horário livre | Aparecer o botão para remoção e ter a possibilidade de remover o horário disponível | O botão aparece e o horário é removido | SIM |
|  | Como atendente, remover uma disponibilidade de horário que esteja ocupado | Não aparecer o botão de remoção, sendo obrigado o cancelamento prévio | O botão de remoção não aparece | SIM |
|  | Como atendente, remover uma disponibilidade de horário livre | Aparecer o botão para remoção e ter a possibilidade de remover o horário disponível | O botão aparece e o horário é removido | SIM |

## Apropriação de Horas no Projeto

|  |  |  |
| --- | --- | --- |
| **Histórico de apropriação de horas** | | |
| **Data do registro** | **Atividade** | **Quantidade de horas** |
| 19/08/2024 | Leitura da apresentação do projeto | 3 |
| 20/08/2024 | Levantamento do trabalho e planejamento | 3 |
| 20/08/2024 | Levantamento do trabalho e planejamento | 3 |
| 22/08/2024 | Definindo o escopo do projeto | 2 |
| 23/08/2024 | Escrevendo a introdução | 2 |
| 24/08/2024 | Escrevendo a introdução | 2 |
| 24/08/2024 | Definindo ferramentas de trabalho | 8 |
| 25/08/2024 | Escrevendo requisitos | 8 |
| 26/08/2024 | Diagramas de caso de uso | 3 |
| 27/08/2024 | Diagramas de caso de uso | 3 |
| 29/08/2024 | Diagrama de contexto da aplicação | 3 |
| 31/08/2024 | Definição da arquitetura da solução | 8 |
| 01/09/2024 | Construção do protótipo | 8 |
| 02/09/2024 | Construção do protótipo | 2 |
| 03/09/2024 | Construção do protótipo | 2 |
| 05/09/2024 | Versão inicial do plano de testes | 4 |
| 08/09/2024 | Implementação da solução | 8 |
| 10/09/2024 | Implementação da solução | 3 |
| 12/09/2024 | Implementação da solução | 3 |
| 15/09/2024 | Implementação da solução | 10 |
| 16/09/2024 | Implementação da solução | 2 |
| 19/09/2024 | Implementação da solução | 2 |
| 23/09/2024 | Implementação da solução | 3 |
| 24/09/2024 | Implementação da solução | 3 |
| 28/09/2024 | Implementação da solução | 10 |
| 29/09/2024 | Implementação da solução | 9 |
| 30/09/2024 | Implementação da solução | 2 |
| 31/09/2024 | Implementação da solução | 2 |
| 02/10/2024 | Implementação da solução | 2 |
| 05/10/2024 | Gravação apresentação protótipo | 6 |
| 06/10/2024 | Bugs na aplicação | 6 |
| 09/10/2024 | Bugs na aplicação | 2 |
| 12/10/2024 | Textos sobre retrospectiva | 6 |
| 13/10/2024 | Textos sobre avaliação e objetivos alcançados | 2 |
| 15/10/2024 | Deploy e correção de bugs | 2 |
| 16/10/2024 | Deploy e correção de bugs | 2 |
| 17/10/2024 | Entrega | 1 |

## Código da Aplicação

Repositório de código: [Repositório](https://github.com/danielbastos/puc-agendamento)

Link da aplicação no ar: [Aplicação](http://puc-agendamento-9b590589bb63.herokuapp.com/login)

**Tabela de credenciais da aplicação:**

|  |  |  |
| --- | --- | --- |
| Tipo | Usuário | Senha |
| Médico | joao@medico.dev | medico.dev |
| Médico | maria@medico.dev | medico.dev |
| Médico | jose@medico.dev | medico.dev |
| Atendente | atendente01@atendimento.dev | atendimento.dev |
| Atendente | atendente02@atendimento.dev | atendimento.dev |
| Paciente | eduardo.azevedo@paciente.dev | paciente.dev |
| Paciente | danilo.costa@paciente.dev | paciente.dev |
| Paciente | mateus.alves@paciente.dev | paciente.dev |
| Admin | admin@admin.dev | admin.dev |

**Vídeo de apresentação do projeto:** [**Apresentação do projeto**](https://github.com/danielbastos/puc-agendamento/blob/main/docs/Apresenta%C3%A7%C3%A3o%20do%20projeto.mp4)

## Avaliação Retrospectiva

Este projeto foi desafiante. Mesmo pequeno, tive que utilizar conhecimentos que não são comuns ao meu trabalho diário.

Os modelos C4 são bons pra descrever estruturas de containers e servidores. Fatalmente incluirei no meu cotidiano.

O cronograma foi uma das maiores dificuldades. Atividades concorrentes e o trabalho foram difíceis de conciliar.

Mas acho que consegui um bom resultado. No caminho aprendi e descobri muitas ferramentas que usarei no meu dia a dia.

## Objetivos Estimados

Entregar um sistema de agendamentos que possa ser utilizado para auxiliar a organização de horários entre médicos e pacientes.

Notificar todos os usuários sobre mudanças em suas agendas, facilitando a comunicação entre clinicas e pacientes na questão de horários.

## Objetivos Alcançados

Acredito que consegui fazer um sistema que ajude a organizar os horários. Mas ficou devendo, muito por tempo, um sistema de notificação melhor.

## Lições aprendidas

|  |  |  |
| --- | --- | --- |
|  | **Retrospectiva (Lições Aprendidas)** | |
|  | **Descrição da Lição** | **Classificação** |
| 1 | Familiaridade com diferentes ferramentas de desenvolvimento | Positiva |
| 2 | Usar deploy integrado com controle de versão (Heroku) | Positiva |
| 3 | Gerencia de tempo e cronograma | Negativa |
| 4 | Experiência na resolução de problemas | Positiva |

## Referências

Astah Professional Versão: 9.2.0/0248cd, 2024  
Evolus Pencil Versão 3.1.1, 2024

Laravel Versão 11 (https://laravel.com/docs/11.x)

Heroku Versão continua, (https://devcenter.heroku.com/categories/reference)

Bastos, Daniel Augusto  
Sistema de agendamento médico  
Rio de Janeiro, 2024