**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Trabalho de Conclusão de Curso

Sistema de Gerenciamento de Projetos

Danilo Germiniani Virginio

Belo Horizonte

<Fevereiro - 2022>.

# Trabalho de Conclusão de Curso

**Sumário**

Trabalho de Conclusão de Curso 3

1. Cronograma de trabalho 4

2. Diagrama de casos de uso 5

3. Requisitos não-funcionais 5

4. Protótipo navegável do sistema 6

5. Diagrama de classes de domínio 6

6. Modelo de componentes 8

6.1. Padrão arquitetural 8

6.2. Diagrama de componentes 8

6.3. Descrição dos componentes 9

7. Diagrama de implantação 11

8. Plano de Testes 12

9. Estimativa de pontos de função 19

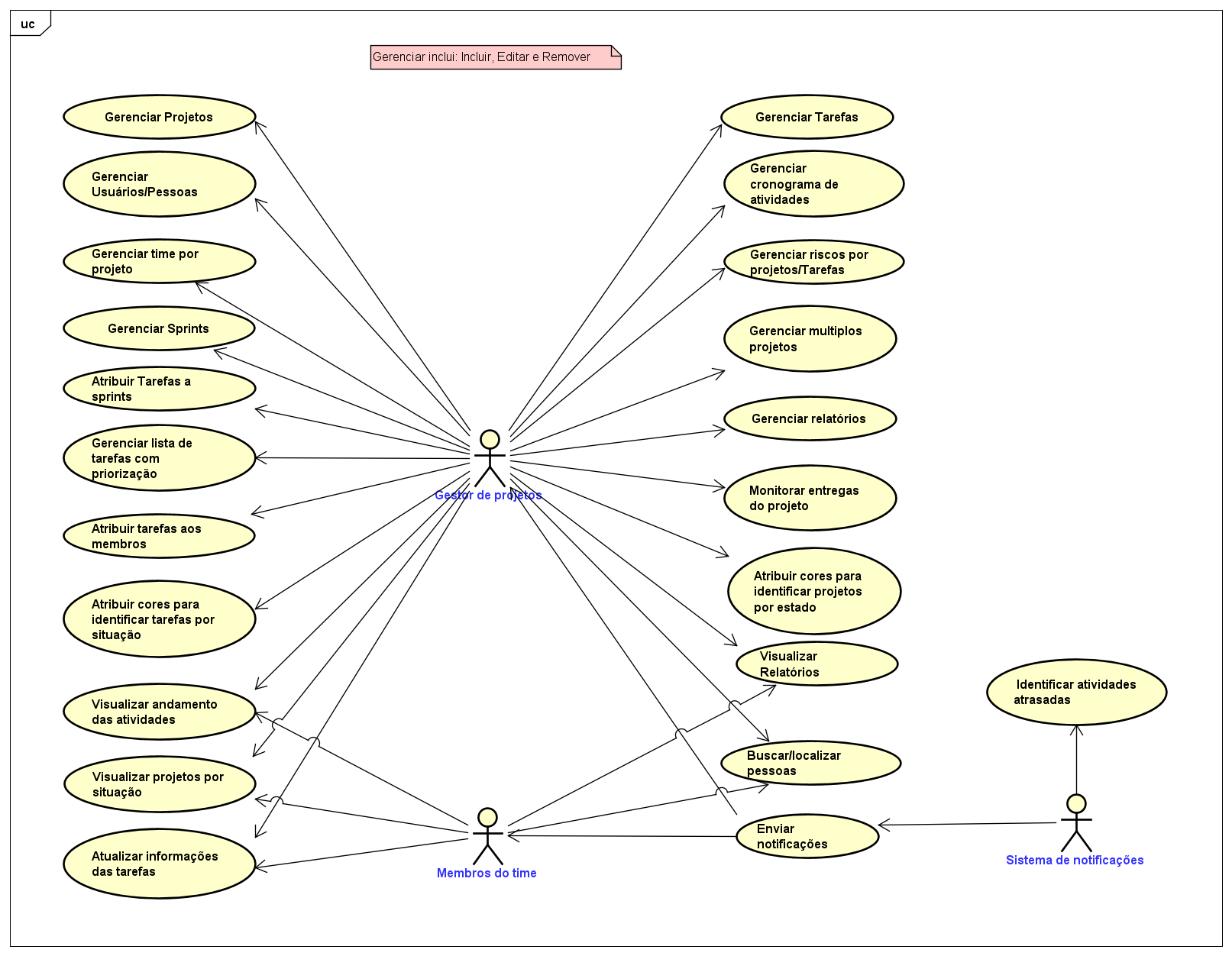
10. Informações da implementação 20

11. Referências 21

## Cronograma de trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 06/11/2021 | 07/11/2021 | 1. Análise do roteiro, organização das atividades e preenchimento do cronograma. | Atividades planejadas e cronograma preenchido. |
| 13/11/2021 | 14/11/2021 | 1. Definição e estudo da ferramenta de prototipação. | Ferramenta de prototipação definida. |
| 20/11/2021 | 21/11/2021 | 1. Análise e refinamento dos requisitos. | Requisitos analisados, refinados e organizados. |
| 27/11/2021 | 04/02/2022 | 1. Criação, atualização e manutenção do protótipo | Versões iniciais do protótipo de acordo com os requisitos. |
| 04/12/2021 | 04/12/2021 | 1. Levantamento dos requisitos não funcionais. | Lista dos requisitos não funcionais do sistema. |
| 05/12/2021 | 05/12/2021 | 1. Estudo e desenvolvimento do diagrama de casos de uso. | Diagrama de casos de uso. |
| 11/12/2021 | 12/12/2021 | 1. Estudo e desenvolvimento do diagrama de classes de domínio. | Diagrama de classes de domínio. |
| 18/12/2021 | 19/12/2021 | 1. Estudo e desenvolvimento do padrão arquitetural. | Padrão arquitetural definido. |
| 08/01/2022 | 08/01/2022 | 1. Estudo e desenvolvimento do diagrama de componentes. | Diagrama de componentes. |
| 15/01/2022 | 16/01/2022 | 1. Estudo e desenvolvimento das descrições dos componentes | Descrição dos componentes. |
| 22/01/2022 | 23/01/2022 | 1. Estudo e desenvolvimento do diagrama de implantação. | Diagrama de implantação. |
| 22/01/2022 | 23/01/2022 | 1. Estudo e desenvolvimento do plano de testes. | Plano de testes. |
| 01/01/2022 | 30/01/2022 | 1. Criação da estimativa de pontos de função. | Planilha de estimativa de pontos de função definidas. |
| 29/01/2022 | 30/01/2022 | 1. Atualização das referências. | Lista das referências utilizadas. |
| 31/01/2022 | 05/02/2022 | 1. Revisão e disponibilidade dos arquivos gerados nesse trabalho | Este documento finalizado e os demais arquivos disponibilizados para consulta |
| 06/02/2022 | 06/02/2022 | 1. Envio do trabalho de conclusão de curso | Trabalho de conclusão de curso enviado. |

## Diagrama de casos de uso



## Requisitos não-funcionais

Ao realizar uma análise sobre o documento de requisitos proposto e de aplicar conceitos importantes sobre confiabilidade, usabilidade, segurança, entre outros, a seguinte lista de requisitos não funcionais foi gerada:

1. O sistema deve ficar totalmente disponível. 7 dias por semana e 24 horas por dia.
2. O sistema deverá ser acessado via Web e não possuíra restrições em relação ao sistema operacional.
3. O sistema deverá ser acessado por navegadores web de dispositivos desktop e mobile; As validações deverão ocorrer nos navegadores mais utilizados no mercado, como Google Chrome, Safari, Microsoft Edge e Mozilla Firefox.
4. O sistema não restringirá o acesso em outro navegador, porém caso o sistema encontre alguma circunstância inesperada, o usuário deverá ser notificado para utilizar alguma das opções do item 3.
5. O sistema deve apresentar confiabilidade contínua, mantendo seu funcionamento em circunstância de rotina e apresentar tratativas corretas em circunstâncias inesperadas.
6. O sistema não deverá apresentar mais do que 15 defeitos conhecidos; Os defeitos não poderão apresentar severidade alta, caso contrário, a correção deve ser iniciada e a solução apresentada o mais rápido possível;
7. O sistema deverá possuir coleta e armazenamento de logs em caso de falhas ocorridas;
8. O sistema deverá apresentar execução de 100% de seu caderno de testes e nenhum defeito de severidade alta antes de ser disponibilizado.
9. O sistema deverá apresentar um plano de contingência, com a restauração de versões e base de dados anteriores, caso ocorra algumas circunstâncias hostis e inesperadas;
10. O sistema deverá ser responsivo, devendo apresentar suas funcionalidades independente da resolução ou configurações de acessibilidade do sistema operacional.
11. O sistema deverá suportar ao menos 500 usuários simultaneamente.
12. O sistema deverá consumir e apresentar as informações ao usuário em menos de 3 segundos, e quando necessário, notificar o usuário em caso de lentidão ou inacessibilidade.
13. O sistema deverá autenticar todos os usuários antes de apresentar qualquer informação;
14. O sistema deverá criptografar dados sensíveis do usuário.
15. O sistema deverá apresentar diretrizes gerais aplicáveis na aparência do software, buscando a melhor usabilidade para o usuário;

## Protótipo navegável do sistema

O documento abaixo detalha os requisitos e casos de uso que foram prototipados. Além disso, também apresenta as imagens da interface geradas pelo protótipo.

A ferramenta utilizada para a prototipação foi a [JustInMind](https://www.justinmind.com/).

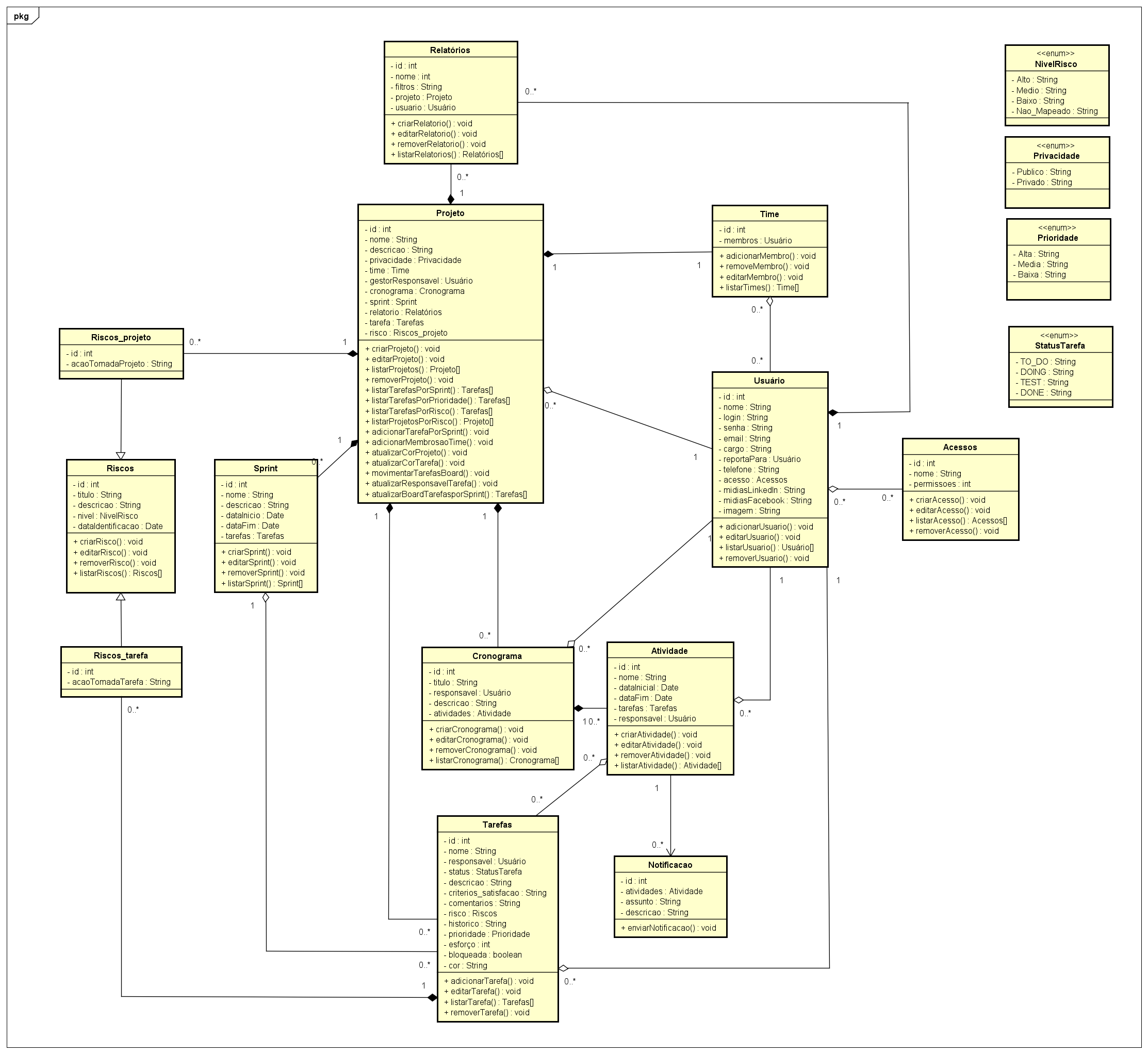
Link para o documento:

1. Github: link
2. GoogleDrive: Link

O vídeo de apresentação do protótipo pode ser acessado em:

1. Github: link
2. GoogleDrive: Link

## Diagrama de classes de domínio



## Modelo de componentes

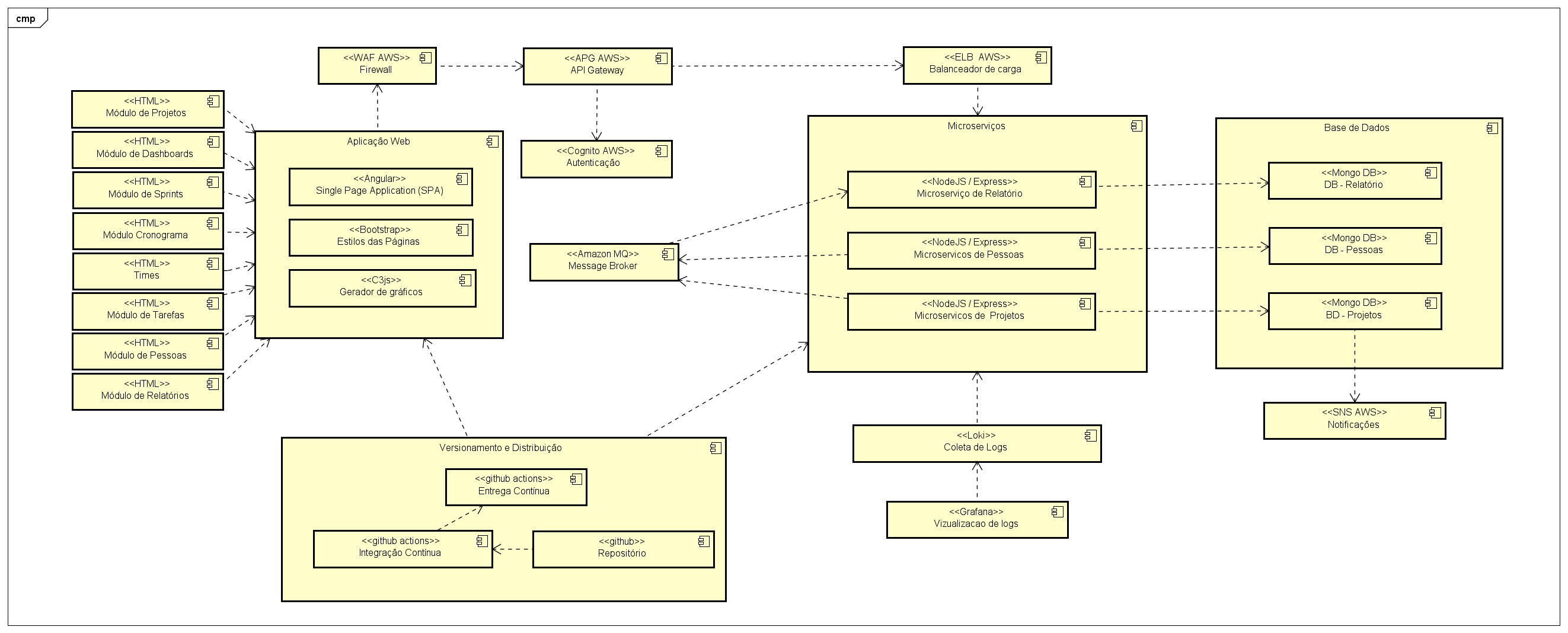
### Padrão arquitetural

Analisando a proposta do trabalho através dos requisitos fornecidos e considerando alguns aspectos como: escalabilidade, flexibilidade, resiliência, entre outros, optou-se pelo desenvolvimento de uma arquitetura híbrida, no qual o backend será baseado em microsserviços, enquanto o frontend será único, baseado em uma arquitetura monolítica.

As seguintes tecnologias foram utilizadas no planejamento e diagramação do projeto:

* [Angular v13](https://angular.io/)
* [C3js v0.7](https://c3js.org/)
* [Bootstrap v5](https://getbootstrap.com/)
* [NodeJs v17](https://nodejs.org/download/release/latest-v17.x/)
* [Express v4](https://expressjs.com/en/changelog/4x.html)
* [Grafana Loki v2](https://grafana.com/docs/loki/latest/maintaining/release/)
* [Grafana v8.3](https://grafana.com/docs/grafana/latest/release-notes/)
* [Mongo DB v5](https://docs.mongodb.com/manual/release-notes/)
* [GitHub Actions](https://github.com/features/actions)
* AWS
  + [Web Application Firewall](https://aws.amazon.com/pt/waf/)
  + [Api Gateway](https://aws.amazon.com/pt/api-gateway/)
  + [Cognito](https://aws.amazon.com/pt/cognito/)
  + [Elastic Load Balancing](https://aws.amazon.com/pt/elasticloadbalancing/)
  + [Amazon MQ](https://aws.amazon.com/pt/amazon-mq/?amazon-mq.sort-by=item.additionalFields.postDateTime&amazon-mq.sort-order=desc)
  + [Single Notification Service](https://aws.amazon.com/pt/sns/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc)
  + [Elastic Kubernetes Service](https://aws.amazon.com/pt/eks/)

### Diagrama de componentes



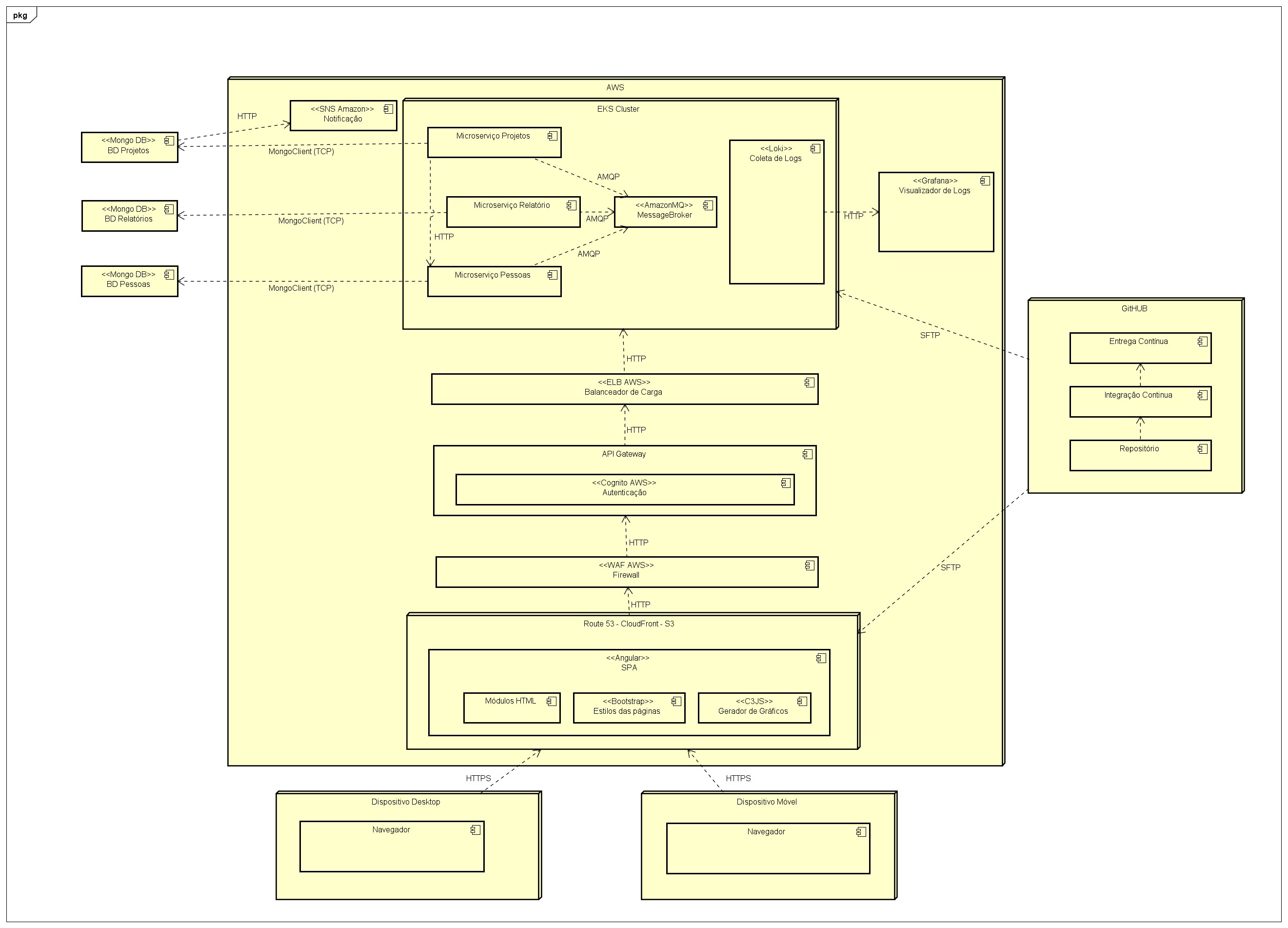
### Descrição dos componentes

|  |  |  |
| --- | --- | --- |
| **Número** | **Componente** | **Descrição** |
|  | Módulos HTML - Projetos, Dashboards, Sprints, Cronograma, Times, Tarefas, Pessoas e Relatórios | Componentes responsáveis pela geração de conteúdo do sistema e que será consumido pela SPA. |
|  | Aplicação Web (SPA, Estilos das páginas e gerador de gráficos e relatórios) | Componente responsável por gerar a interface para o usuário, de maneira responsiva e gerenciar gráficos e relatórios. |
|  | Firewall (WAF AWS) | Responsável por proteger a aplicação contra *bots* e *exploits* comuns na Web que podem afetar a disponibilidade, comprometer a segurança ou consumir recursos em excesso. |
|  | API Gateway | Fornece um ponto de entrada única para todas as rotas expostas, roteando solicitações de clientes para serviços. |
|  | Autenticação (Cognito) | Componente responsável por realizar a autenticação dos usuários e a autorização externa de aplicações. |
|  | Balanceador de Carga (ELB AWS) | Responsável por distribuir automaticamente o tráfego de aplicações, garantindo alta disponibilidade e distribuição de cargas nos serviços. |

|  |  |  |
| --- | --- | --- |
|  | Microsserviços (Projetos, Pessoas e Relatórios) | Componentes independentes responsáveis por determinadas ações do projeto.  Microserviço de Projeto é responsável pelas operações relacionadas ao projeto, sprint, cronograma, time, riscos e tarefas.  Microserviço de Pessoas é responsável pelas operações relacionadas aos usuários (Pessoas).  Microserviço de Relatório é responsável pelas operações relacionadas na geração de relatórios sobre projetos e pessoas. |
|  | Message Broker (Amazon MQ) | Componente responsável por garantir o armazenamento e enfileiramento de mensagens. As mensagens geradas pelos microsserviços de projetos e pessoas são consumidas pelo microserviço de relatório. |
|  | Coleta de logs (Grafana Loki) | Componente responsável pela coleta e agregação de logs da aplicação. |
|  | Visualizador de logs (Grafana) | Componente responsável por coletar os logs gerados pelo componente acima e apresenta-los de forma organizada e visual. |
|  | Banco de Dados (Mongo DB) | Componente responsável pela base de dados da aplicação. O projeto contará com 3 bases distintas: Projetos, Pessoas e Relatórios. |
|  | Versionamento e distribuição – Repositório, Integração Contínua e Entrega Contínua (Github actions) | Componente responsável pelo armazenamento, construção, gerenciamento de dependências, testes e entrega das versões. |
|  | Notificação (SNS AWS) | Componente responsável pelo envio da notificação ao usuário. |

## 

## Diagrama de implantação



## Plano de Testes

O plano de teste abaixo foi planejado afim de validar alguns cenários, envolvendo os casos de uso: Gerenciar Projetos, Gerenciar Usuários/Pessoas e Gerenciar Sprints.

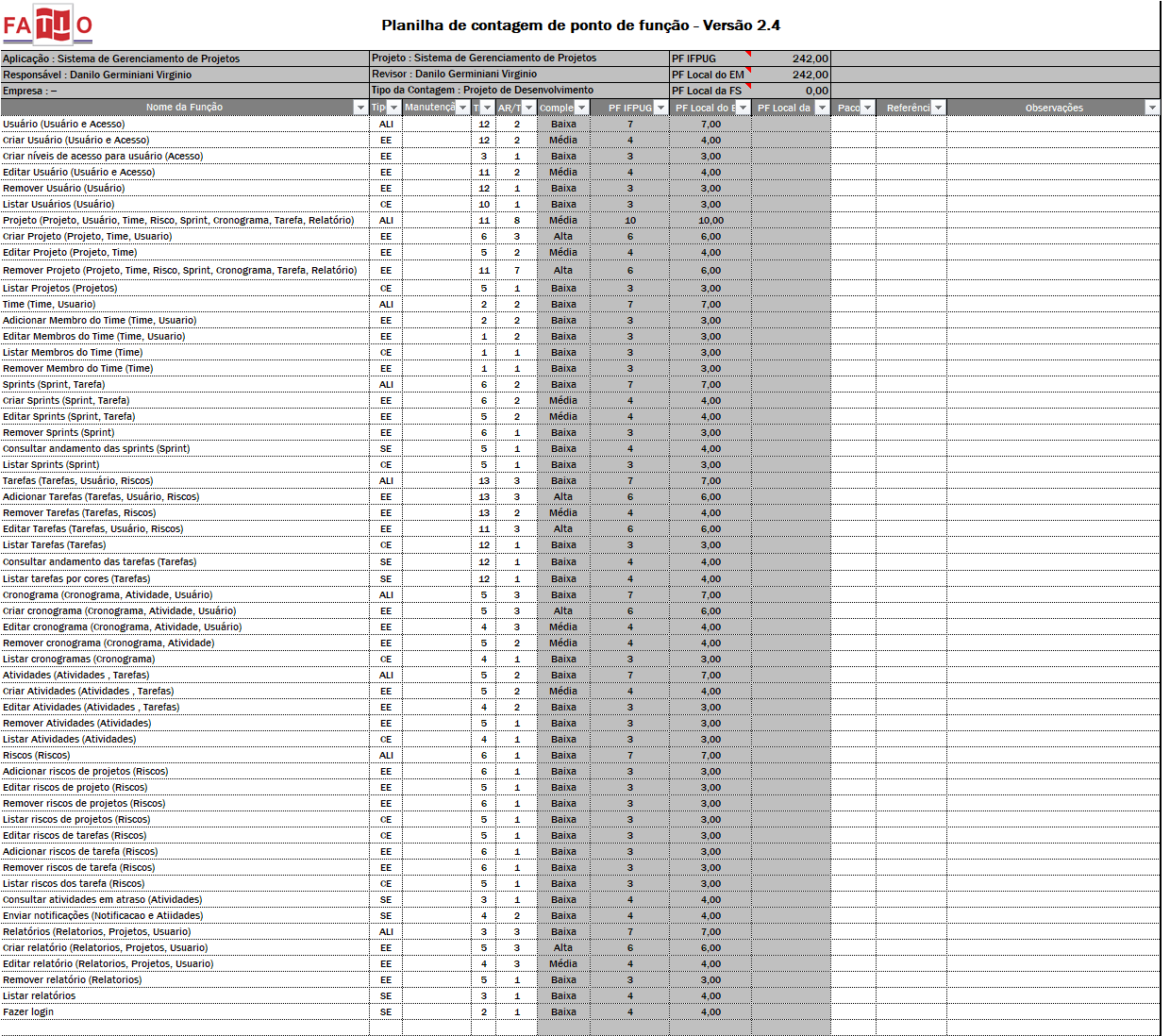
Como complemento, os testes foram executados no protótipo e as evidências podem ser encontradas nos links abaixo:

1. Github: link
2. GoogleDrive: Link

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  | Gerenciar projetos | Validar a inserção de um novo projeto. | ***Pré-condições:***   1. Possuir acesso a plataforma. 2. Possuir nível de acesso para cadastrar novo projeto.   Passos:   1. Acessar a plataforma e realizar o login; 2. Clicar em menu ‘Projetos’ e então em ‘Gerenciar Projetos’; Selecionar a opção: “Criar Projeto”; 3. Preencher os campos com os respectivos dados:   ***Nome do Projeto:***  Projeto TCC Teste  ***Descrição*:**  Testes criação de projeto com sucesso.  ***Privacidade:***  Público  ***Time:***  Selecionar ao menos 1 pessoa da lista.  ***Gestor Responsável:***  Selecionar ao menos 1 pessoa da lista.   1. Clicar no botão “Criar” | Uma janela com a mensagem: “Projeto criado com sucesso” deve ser mostrada.  O projeto criado deve ser listado na lista de projetos. |
|  | Gerenciar projetos | Validar a tentativa de inserção de um novo projeto com um nome já existente. | ***Pré-condições:***   1. Possuir acesso a plataforma. 2. Possuir nível de acesso para cadastrar novo projeto. 3. Possuir um projeto cadastrado com o nome “Projeto TCC Teste”   Passos:   1. Acessar a plataforma e realizar o login; 2. Clicar em menu ‘Projetos’ e então em ‘Gerenciar Projetos’; Selecionar a opção: “Criar Projeto”; 3. Preencher os campos com os respectivos dados:   ***Nome do Projeto:*** Projeto TCC Teste  ***Descrição*:**  Testes criação de projeto com sucesso.  ***Privacidade:***  Público  ***Time:***  Selecionar ao menos 1 pessoa da lista.  ***Gestor Responsável:***  Não selecionar nenhuma pessoa da lista.   1. Clicar no botão “Criar” | Uma janela com a mensagem: “Já existe um projeto com esse nome. Por favor, verifique as informações fornecidas e tente novamente.” deve ser mostrada.  Um botão “Ok” deve ser mostrado e ao clicar nele o usuário é redirecionado para a tela de criação de projeto. |
|  | Gerenciar sprints | Validar a inserção de uma nova sprint. | ***Pré-condições:***   1. Possuir acesso a plataforma. 2. Possuir nível de acesso para criar uma nova sprint. 3. Possuir um projeto cadastrado com o nome “Projeto TCC Teste”. 4. Possuir ao menos uma tarefa cadastrada.   Passos:   1. Acessar a plataforma e realizar o login; 2. Clicar em menu ‘Projetos’ e então selecione o projeto “Projeto TCC Teste”. 3. Selecionar a opção sprint no menu lateral esquerdo, clique em ‘Gerenciar Sprints’ e então em ‘Criar Sprint’. 4. Preencher os campos com os respectivos dados:   ***Nome da Sprint:***  Sprint TCC Teste  ***Descrição*:**  Teste de criação de sprint com sucesso.  ***Data Início:***  03/01/2022  ***Data Fim:***  17/01/2022  ***Tarefas do Backlog:***  Selecionar ao menos 1 tarefa da lista.   1. Clicar no botão “Criar” | Uma janela com a mensagem: “Sprint criada com sucesso” deve ser mostrada.  A sprint criada deve ser listada na lista de sprints. |
|  | Gerenciar Sprints | Validar a tentativa de inserção de uma nova sprint com a data final anterior à data inicial. | ***Pré-condições:***   1. Possuir acesso a plataforma. 2. Possuir nível de acesso para criar uma nova sprint. 3. Possuir um projeto cadastrado com o nome “Projeto TCC Teste”. 4. Possuir ao menos uma tarefa cadastrada.   Passos:   1. Acessar a plataforma e realizar o login; 2. Clicar em menu ‘Projetos’ e então selecione o projeto “Projeto TCC Teste”. 3. Selecionar a opção sprint no menu lateral esquerdo, clique em ‘Gerenciar Sprints’ e então em ‘Criar Sprint’. 4. Preencher os campos com os respectivos dados:   ***Nome da Sprint:***  Sprint TCC Teste  ***Descrição*:**  Teste de criação de sprint com sucesso.  ***Data Início:***  17/01/2022  ***Data Fim:***  03/01/2022  ***Tarefas do Backlog:***  Selecionar ao menos 1 tarefa da lista.   1. Clicar no botão “Criar” | Uma janela com a mensagem: “A data inicial não pode ser posterior a data final. Por favor, verifique as informações fornecidas e tente novamente.” deve ser mostrada.  Um botão “Ok” deve ser mostrado e ao clicar nele o usuário é redirecionado a tela de criação de sprint. |
|  | Gerenciar Usuários/Pessoas | Validar a inserção de uma nova pessoa. | ***Pré-condições:***   1. Possuir acesso a plataforma. 2. Possuir nível de acesso para cadastrar uma nova pessoa.   Passos:   1. Acessar a plataforma e realizar o login; 2. Selecionar a opção ‘Pessoas’ no menu lateral esquerdo, clique em ‘Gerenciar Pessoas’ e então em ‘Adicionar Pessoa’. 3. Preencher os campos com os respectivos dados:   ***Nome da Pessoa:***  Pessoa Teste  ***Email*:**  teste@tcc.com  ***Cargo:***  Tester  ***Reporta para:***  Diretor A  ***Telefone:***  (35)99211897  ***Mídias Sociais:***  Linkedln - <https://www.linkedin.com/in/pessoa-teste>  Facebook - <https://www.facebook.com/pessoa-teste>  ***Acessos:***  Visualizar pessoas;  Visualizar projetos;   1. Clicar no botão “Criar” | Uma janela com a mensagem: “Pessoa cadastrada com sucesso” deve ser mostrada.  A pessoas deve ser listada na lista de pessoas; |
|  | Gerenciar Usuários/Pessoas | Validar a tentativa de inserção de uma nova pessoa sem preencher o campo obrigatório nome. | ***Pré-condições:***   1. Possuir acesso a plataforma. 2. Possuir nível de acesso para cadastrar uma nova pessoa.   Passos:   1. Acessar a plataforma e realizar o login; 2. Selecionar a opção ‘Pessoas’ no menu lateral esquerdo, clique em ‘Gerenciar Pessoas’ e então em ‘Adicionar Pessoa’. 3. Preencher os campos com os respectivos dados:   ***Nome da Pessoa:***  [Não preencher este campo.]  ***Email*:**  pessoa\_tst@tcc.com  ***Cargo:***  Tester Sr  ***Reporta para:***  Diretor A  ***Telefone:***  (35)99211897  ***Mídias Sociais:***  Linkedln - <https://www.linkedin.com/in/pessoa-teste>  Facebook - <https://www.facebook.com/pessoa-teste>  ***Acessos:***  Visualizar pessoas;  Visualizar projetos;   1. Clicar no botão “Criar” | Uma janela com a mensagem: “Não foi possível cadastrar a pessoa. Preencha o campo nome.” deve ser mostrada.  Um botão “Ok” deve ser mostrado e ao clicar nele o usuário é redirecionado a tela de cadastro da pessoa. |

## Estimativa de pontos de função

A seguinte planilha de estimativa de pontos de função foi gerada:



A planilha pode ser encontrada em:

1. Github: link
2. GoogleDrive: Link

## Informações da implementação

Item não implementado.

## Referências

FAKHROUTDINOV, Kirill. UML 2.5 Diagrams Overview. uml-diagrams, 2016. Disponível em: <https://www.uml-diagrams.org/uml-25-diagrams.html>. Acesso em: 05 de dez. de 2021.

ÖZKAYA, Mehmet. Microservices Architecture. Medium, 2021. Disponível em: <https://medium.com/design-microservices-architecture-with-patterns/microservices-architecture-2bec9da7d42a>. Acesso em: 18 de jan. de 2022.

MORGAN, Andrew. The Modern Application Stack – Part 1: Introducing The MEAN Stack. MongoDB, 2017. Disponível em: <https://www.mongodb.com/blog/post/the-modern-application-stack-part-1-introducing-the-mean-stack>. Acesso em: 18 de jan. de 2022.

Frameworks for Developing Single Page Applications in 2022. Toobler, 2022. Disponível em: <https://www.toobler.com/blog/frameworks-for-developing-single-page-applications>. Acesso em: 18 de jan. de 2022.

RESELMAN, Bob. Synchronous vs. asynchronous microservices communication patterns. The Server Side, 2021. Disponível em: <https://www.theserverside.com/answer/Synchronous-vs-asynchronous-microservices-communication-patterns#:~:text=A%20synchronous%20microservice%20is%20one,seen%20in%20Figure%201%20below.>. Acesso em: 22 de jan. de 2022.

LOPES, Jhoney da Silva. Guia Prático em Análise de Ponto de Função. Fattocs, 2011. Disponível em: <https://www.fattocs.com/wp-content/uploads/2020/04/JhoneySLopes-JoseLBraga-2011.pdf>. Acesso em: 30 de jan. de 2022.