

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**  
**PUC Minas Virtual**  
**Pós-graduação *Lato Sensu* em Engenharia de *Software***

Trabalho de Conclusão de Curso

PMaster – Sistema de Gerenciamento de Projetos

GABRIEL HENRIQUE DE OLIVEIRA LEITE

Belo Horizonte  
Abril, 2022.

## Trabalho de Conclusão de Curso

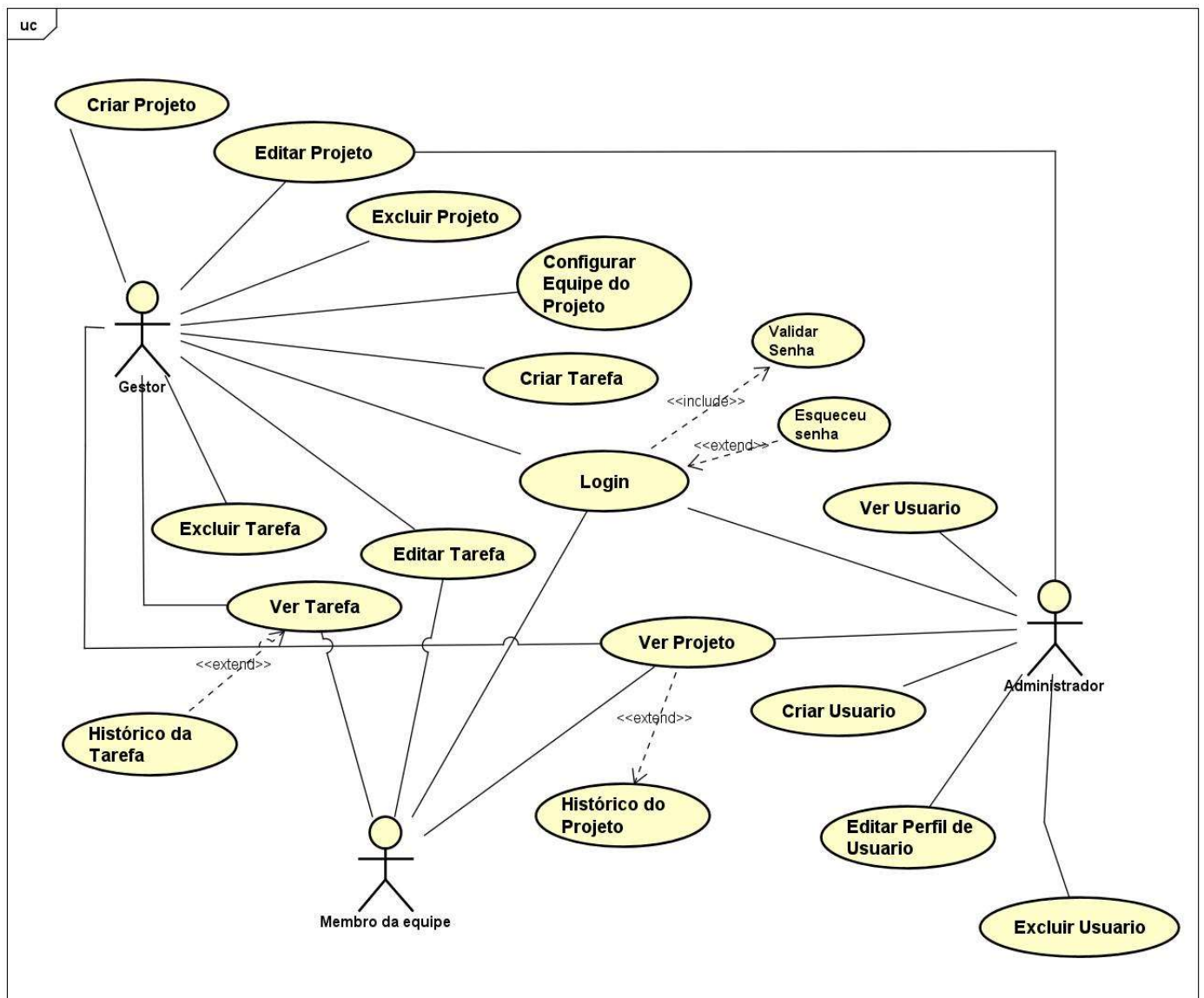
### *Sumário*

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	5
5. Diagrama de classes de domínio	6
6. Modelo de componentes	7
6.1. Padrão arquitetural	7
6.2. Diagrama de componentes	8
6.3. Descrição dos componentes	9
7. Diagrama de implantação	10
8. Plano de Testes	11
9. Estimativa de pontos de função	13
10. Referências	13

## 1. Cronograma de trabalho

Datas		Atividade / Tarefa	Produto / Resultado
De	Até		
29 / 11 / 21	03 / 12 / 21	1. Diagrama de casos de uso	Produzir o diagrama de casos de uso com os atores e casos do sistema.
06 / 12 / 21	10 / 12 / 21	2. Requisitos não-funcionais	Listar e especificar todos os requisitos não-funcionais essencial para o sistema.
13 / 12 / 21	07 / 01 / 22	3. Protótipo navegável do sistema	Fazer um protótipo do sistema mostrando as telas e os fluxos do sistema.
10 / 01 / 22	21 / 01 / 22	4. Modelo de componentes	Diagrama contendo os componentes do sistema e também uma especificação de cada um deles.
24 / 01 / 22	28 / 01 / 22	5. Diagrama de implantação	Fazer o diagrama de como será a implantação do sistema.
31 / 01 / 22	04 / 02 / 22	6. Plano de testes	Listagem e descrição dos testes que deve ser feito no sistema para uma melhor qualidade do software.
07 / 02 / 22	25 / 02 / 22	7. Planilha de pontos de função	Preenchimento da planilha da contagem de pontos de função
28 / 02 / 22	04 / 03 / 22	8. Organização dos materiais	Separar os diagramas feitos e o trabalho para compartilhar com a banca
07 / 03 / 22	11 / 03 / 22	9. Gravar o protótipo	Fazer um vídeo com o protótipo navegável e seu funcionamento.
14 / 03 / 22	18 / 03 / 22	10. Revisão do trabalho	Revisão do trabalho, tanto na parte linguística quanto na parte lógica
17 / 04 / 22	17 / 04 / 22	11. Entrega do trabalho	Anexo do trabalho na plataforma da PUC

## 2. Diagrama de casos de uso



## 3. Requisitos não-funcionais

- O sistema deve rodar em qualquer dispositivo móvel.
- O sistema deve ser responsivo nas interfaces gráficas.
- O sistema só poderá gravar a senha dos usuários de forma criptografada no banco de dados.

- O sistema deve manter a sessão de login do usuário por no máximo 10 minutos de inatividade.
- O sistema deverá ter interoperabilidade com a plataforma da Google para login e autenticação via Gmail.
- O sistema deverá processar pelo menos 10 requisições por segundo.
- O sistema deve ter uma disponibilidade de 99.0% ou mais.
- O sistema deve tolerar uma taxa de falha de 1/1.000 segundos.
- O sistema deve utilizar Java na versão 8 ou superior para desenvolvimento.
- O sistema deve ser feito em microsserviços.
- O sistema deverá fazer cópias de backup com pelo menos 24 horas de intervalo entre as cópias.

#### ***4. Protótipo navegável do sistema***

O protótipo para do Sistema de Gerenciamento de Projetos (PMaster) foi feito no figma e da para ser acessado no link abaixo:

-<https://www.figma.com/proto/Sw8bGj7cNGnllsH7BZW4UT/Pmaster-Prototipo?node-id=1%3A2&scaling=contain&page-id=0%3A1&starting-point-node-id=1%3A2>

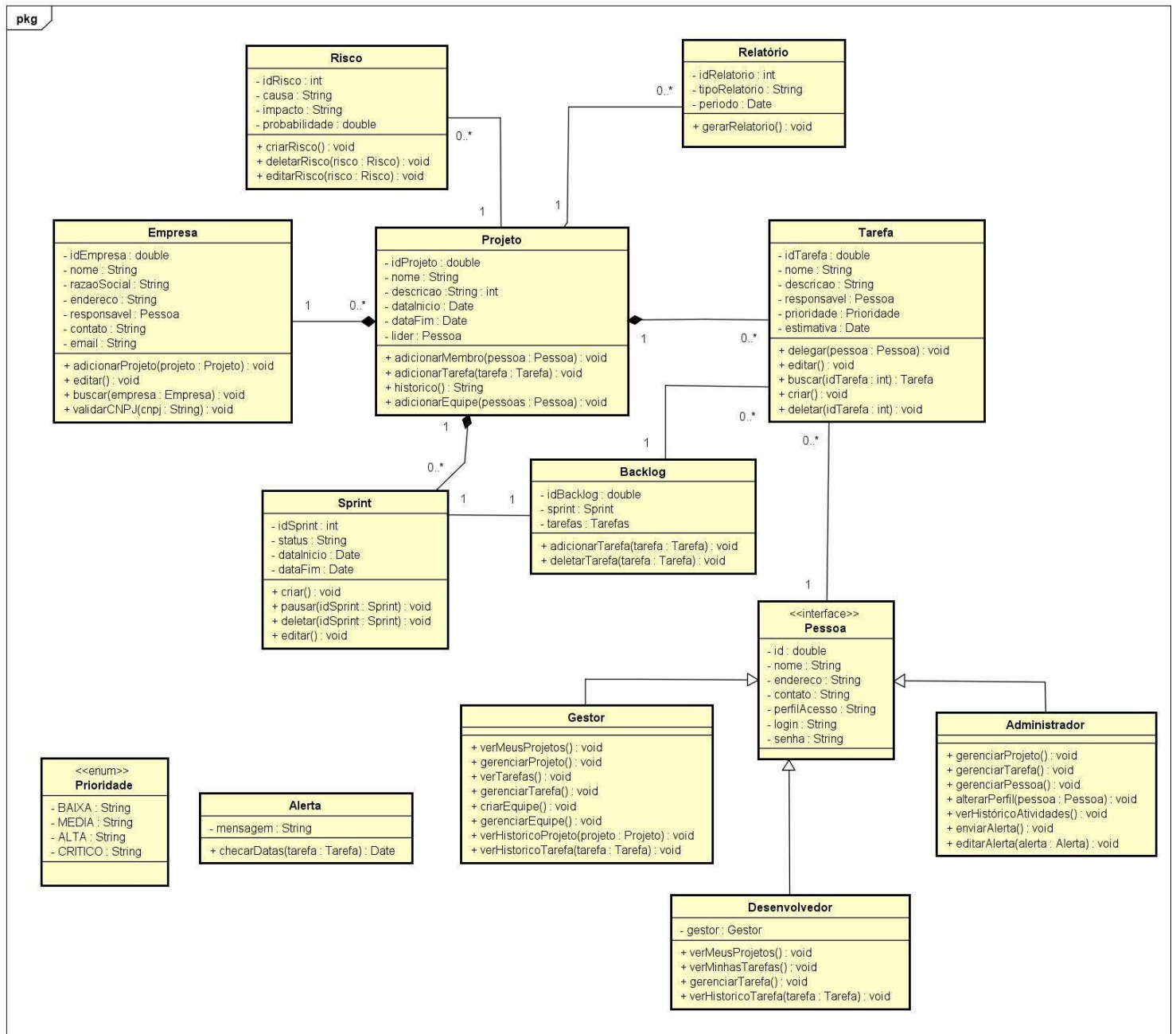
O vídeo de apresentação do protótipo está no meu Github, no link:

- <https://github.com/ghleite/TCC-PUCMinas/tree/main/VideoPrototipo>

Também foi feito um upload no Youtube do mesmo video:

- [https://youtu.be/XZIJ0\\_gfVf8](https://youtu.be/XZIJ0_gfVf8)

## 5. Diagrama de classes de domínio



## **6. Modelo de componentes**

### **1. Padrão arquitetural**

Para o Sistema de Gerenciamento de Projetos **PMaster** foi escolhido uma arquitetura de microsserviços para se ter uma maior independência entre serviços e, com isso, os desenvolvedores terem mais autonomia para tomarem decisões importantes em serviços específicos, essa arquitetura é também utilizada para ter uma maior escalabilidade da aplicação e uma implantação contínua e automatizada.

O Spring Boot vai ser utilizado no sistema por se apresentar um framework completo voltado para a Web e com várias vantagens como: ter servidores embutidos (Tomcat), aumento da produtividade, fácil configuração para ambientes de produção, oferece diversos plugins para melhor desenvolvimento e oferece também um padrão de testes unitários e de integração.

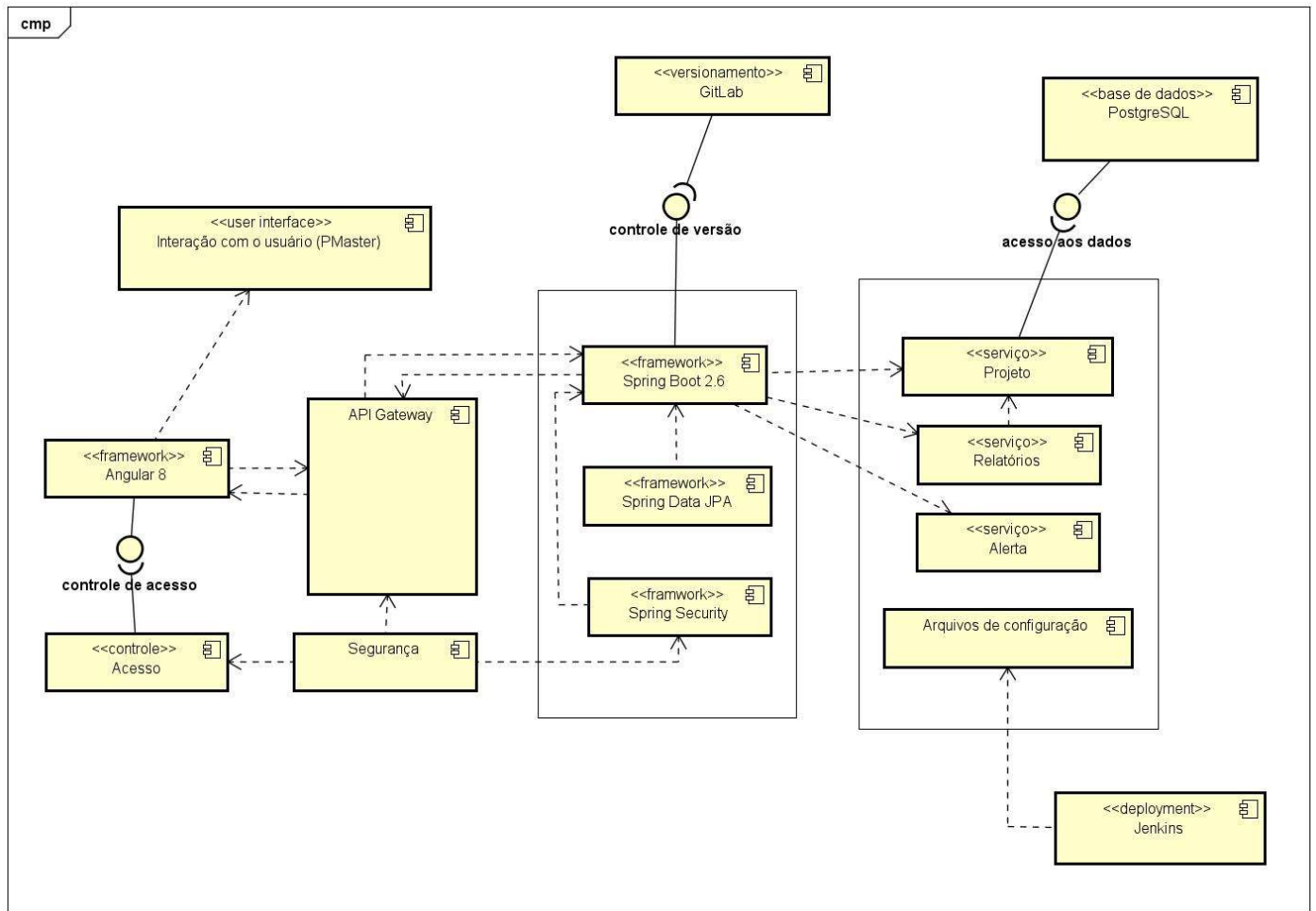
Além do Spring Boot será utilizado o Spring Data para a integração com o banco de dados podendo fazer consultas dinâmicas e ter muitas possibilidades para tratar a persistência de dados. Ainda no universo Spring será utilizado o Spring Security para autenticação e autorização com OAuth 2.0.

Para o banco de dados foi decidido pelo PostgreSQL por ter uma melhor forma de escalabilidade e também uma ótima contribuição da comunidade, tornando-o mais confiável e seguro.

Na questão do versionamento será utilizado o Gitlab como forma de trabalho em equipe e colaboração com o projeto, no lado da implantação será utilizado o Jenkins com todo o seu poder de entrega contínua e configuração de scripts.

Por último o Angular 8 vai fazer todo o processo de interação com usuário ao criar as páginas e tratar os dados. É uma ótima ferramenta por trabalhar com Typescript que tem maior escalabilidade e manutenibilidade, além de ser mantido pela Google e ter uma vasta comunidade por trás do projeto.

## 2. Diagrama de componentes

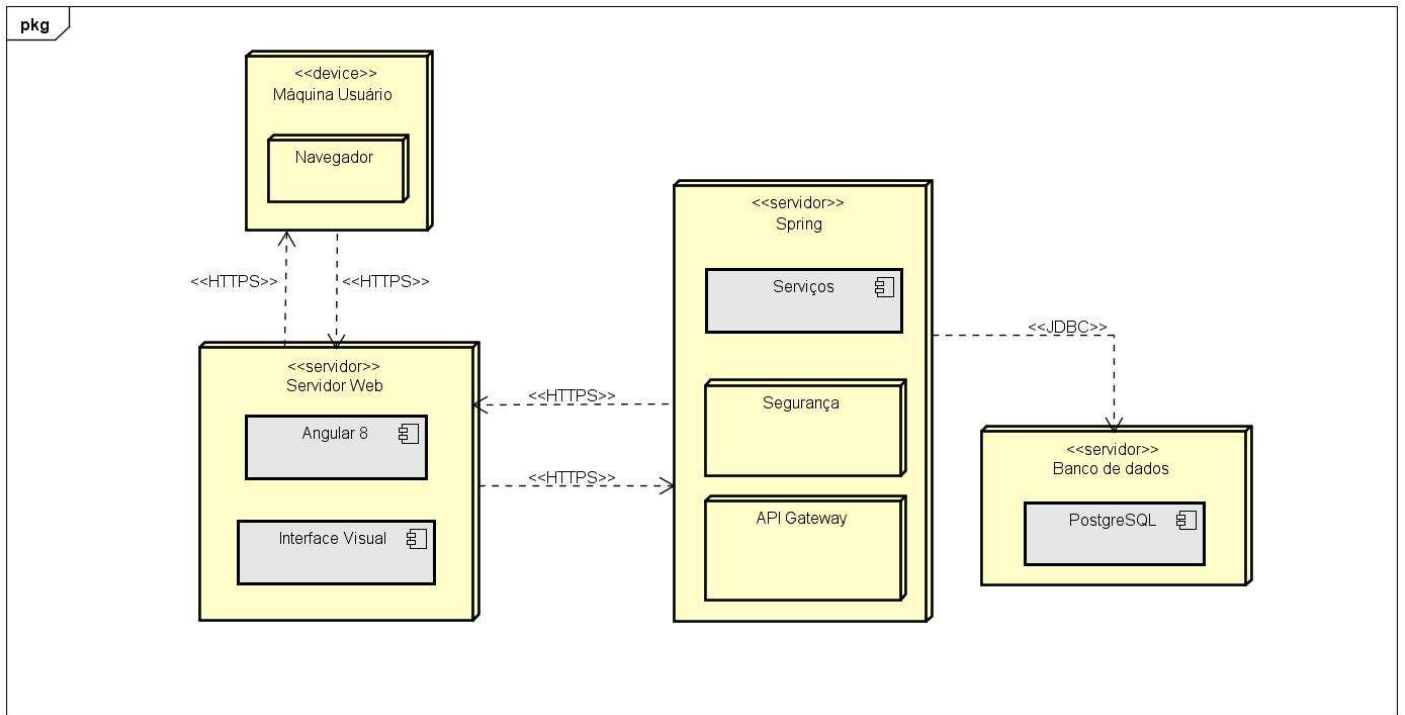




**3. Descrição dos componentes**

<b>Número</b>	<b>Componente</b>	<b>Descrição</b>
1	Spring Boot 2.6	Framework voltado para Java que permite maior produtividade dentro de sistemas Web.
2	Spring Data	Parte do framework Spring que faz todo o gerenciamento e comunicação com o banco de dados
3	Spring Security	Parte do framework Spring responsável pela segurança da aplicação como as funções de autenticação e autorização
4	GitLab	Gerenciador de repositório baseado em Git para versionamento do projeto
5	Jenkins	Ferramenta para automatizar a implantação do sistema
6	PostgreSQL	Banco de dados de código aberto para persistência de dados
7	Angular 8	Framework para construção de aplicações web
8	Projeto	Componente para gerenciar as funcionalidades principais do projeto
9	Relatórios	Componente focado em gerar relatórios e disponibilizá-los diariamente para os usuários
10	Alerta	Componente responsável por fazer leitura de logs do sistema e gerar alertas se necessário
11	Arquivos de configuração	Componente para toda configuração embutida no sistema
12	API Gateway	Componente de integração do sistema para gerenciamento de APIs e entradas/saídas do sistema
13	Segurança	Componente de gerenciamento de todos os pontos de segurança do sistema, seja para o mundo externo seja entre perfis de usuário
14	Acesso	Componente que gerencia acessos dos usuários e seus respectivos perfis
15	Interação com o Usuário	Componente do projeto que possibilita a interação do usuário com o sistema.

## 7. Diagrama de implantação



**8. Plano de Testes**

<b>Número</b>	<b>Caso de uso</b>	<b>Objetivo do caso de teste</b>	<b>Entradas</b>	<b>Resultados esperados</b>
1	Login: cadastrar novo usuário	Validar o cadastro no sistema de um novo usuário.	<ul style="list-style-type: none"> <li>- Nome completo</li> <li>- Endereço</li> <li>- Telefone ou e-mail para contato</li> <li>- Nome de usuário (login)</li> <li>- Senha</li> </ul>	- O sistema mostra uma mensagem na tela indicando que o cadastro foi realizado com sucesso e redireciona para a tela de login inicial.
2	Login: cadastrar novo usuário	Indicar possíveis erros no formulário de cadastro.	<ul style="list-style-type: none"> <li>- Nome completo com algum carácter especial</li> <li>- Endereço</li> <li>- Telefone ou e-mail para contato</li> <li>- Nome de usuário (login)</li> <li>- Senha</li> </ul>	- O sistema indica que há um carácter especial no campo Nome e mostra uma mensagem na tela para que seja feita a correção deste campo.
3	Login: entrar no sistema	Validar se o usuário e senha está de acordo com o cadastrado no sistema.	<ul style="list-style-type: none"> <li>- Nome do usuário</li> <li>- Senha</li> </ul>	- O usuário acessa a tela principal do sistema.
4	Login: entrar no sistema	Negar o login do usuário em caso de senha ou nome incorreto.	<ul style="list-style-type: none"> <li>- Nome do usuário</li> <li>- Senha</li> </ul>	- O usuário recebe uma mensagem de erro na tela de Login com a mensagem de usuário e/ou senha incorreta.
5	Projeto: cadastrar	Validar a criação correta de um projeto dentro do sistema.	<ul style="list-style-type: none"> <li>- Nome</li> <li>- Descrição</li> <li>- Data Inicio (DD/MM/YYYY)</li> <li>- Data Fim (DD/MM/YYYY)</li> <li>- Lider</li> <li>- Empresa relacionada</li> </ul>	- O sistema registra o projeto com sucesso e emite a mensagem de que o projeto foi criado com sucesso.
6	Projeto: cadastrar	Validar se a empresa cadastrada já existe na base de dados do sistema.	<ul style="list-style-type: none"> <li>- Nome</li> <li>- Descrição</li> <li>- Data Inicio (DD/MM/YYYY)</li> <li>- Data Fim</li> </ul>	- O sistema emite uma mensagem com a informação que não existe o Lider informado no cadastro do sistema e que é necessário alterar o

			(DD/MM/YYYY) - Lider que não está cadastrado no sistema - Empresa relacionada	campo.
7	Projeto: editar	Validar a edição feita pelo o usuário e confirmar edição.	- Nova descrição para o projeto previamente cadastrado no sistema	- É validado o campo novo alterado e o sistema envia uma mensagem de que o campo descrição foi alterado com sucesso.
8	Projeto: editar	Verificar edição de data e negar a edição devido a erro no campo data.	- É alterada a data de início para uma data posterior a data fim	- O sistema emite uma mensagem de erro na tela informando que não é possível ter uma Data Início maior que Data Fim.
9	Tarefa: cadastrar	Validar o cadastro de uma nova tarefa em projeto já existente.	- Nome da tarefa - Descrição - Responsável - Prioridade - Estimativa de tempo	O sistema emite uma mensagem de tarefa cadastrada com sucesso.
10	Tarefa: cadastrar	Verificar o cadastro de uma tarefa e não cadastrá-la devido a erro no preenchimento.	- Nome da tarefa - Descrição - Responsável - Estimativa de tempo	O sistema emite uma mensagem de erro de que falta ser preenchido o campo prioridade para que a tarefa seja criada.

## ***9. Estimativa de pontos de função***

A contagem de pontos de função foi compartilhada no meu Github como público pelo link:

[-https://github.com/ghleite/TCC-PUCMinas/tree/main/Analise%20de%20PF](https://github.com/ghleite/TCC-PUCMinas/tree/main/Analise%20de%20PF)

Para visualizarem mais sobre os itens deste trabalho como os diagramas basta acessarem a raiz da pasta:

[-https://github.com/ghleite/TCC-PUCMinas](https://github.com/ghleite/TCC-PUCMinas)

## ***10.Referências***

PRESSMAN, Roger. **Engenharia de Software: Uma abordagem Profissional**. 8. ed. McGraw-Hill/Bookman. 2014.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. Pearson Universidades. 2011.

MEDEIROS, Ernani. **Desenvolvendo Software com UML Definitivo 2.0**. Pearson Makron Books, 2004