**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

Gustavo Duarte Santos

Belo Horizonte

agosto de 2022

# Trabalho de Conclusão de Curso

**Sumário**

[Trabalho de Conclusão de Curso 2](#_Toc112770073)

[1. Cronograma de trabalho 3](#_Toc112770074)

[2. Diagrama de casos de uso 4](#_Toc112770075)

[3. Requisitos não-funcionais 5](#_Toc112770076)

[4. Protótipo navegável do sistema 6](#_Toc112770077)

[5. Diagrama de classes de domínio 7](#_Toc112770078)

[6. Modelo de componentes 8](#_Toc112770079)

[6.1. Padrão arquitetural 8](#_Toc112770080)

[6.2. Diagrama de componentes 9](#_Toc112770081)

[6.3. Descrição dos componentes 9](#_Toc112770082)

[7. Diagrama de implantação 11](#_Toc112770083)

[8. Plano de Testes 12](#_Toc112770084)

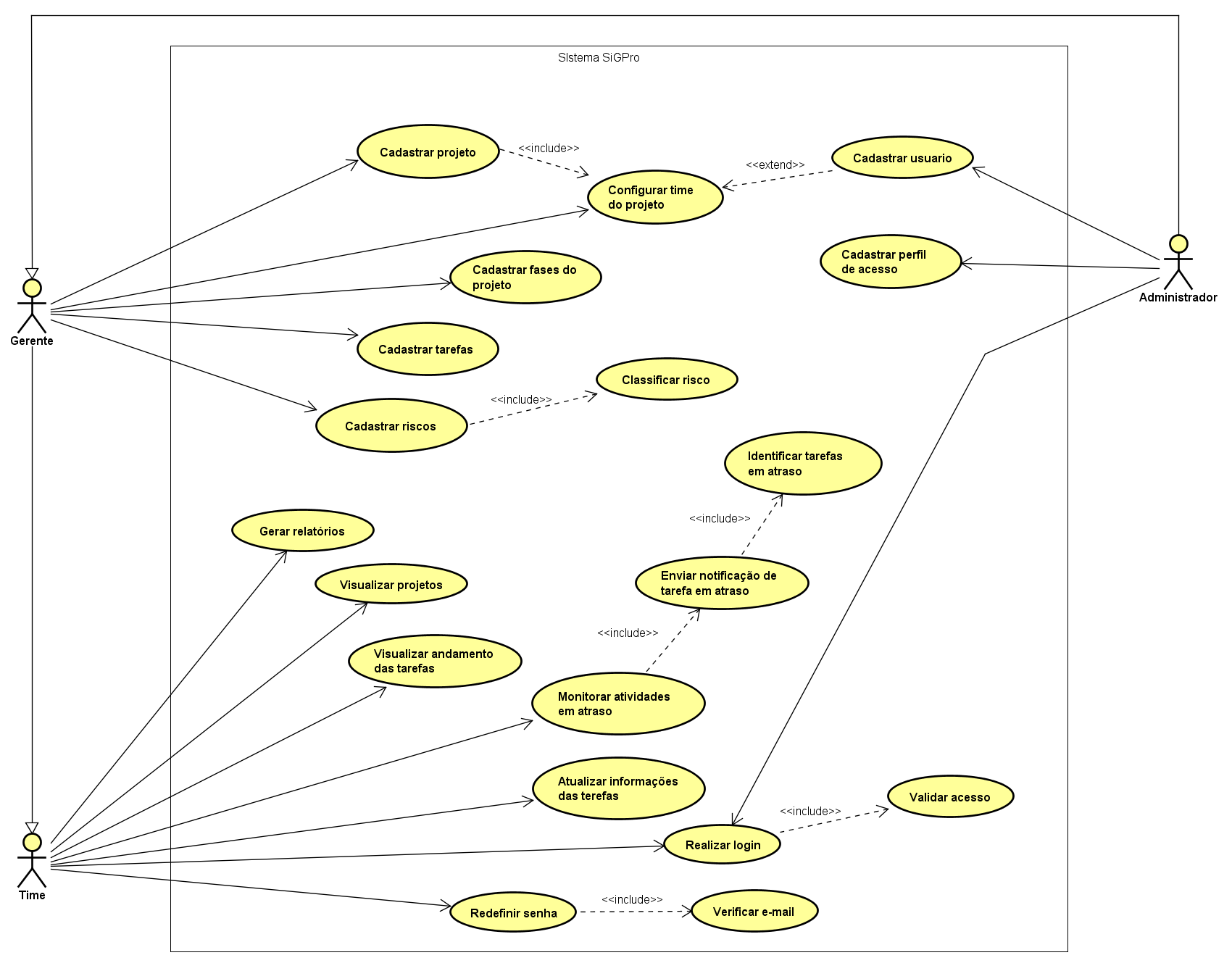
[9. Estimativa de pontos de função 15](#_Toc112770085)

[10. Referências 16](#_Toc112770086)

## Cronograma de trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 04/ 07 / 22 | 09 / 07 / 22 | 1. Entender atividades relacionadas ao projeto, analisar opções de escopo, revisar conteúdo e estudar material de apoio disponibilizado no AVA para realização do trabalho. | Definição do escopo do trabalho. |
| 10 / 01/ 22 | 11 / 01 /22 | 2. Elaborar cronograma de atividades do TCC. | Cronograma com as atividades e datas definidas. |
| 12 / 07 / 22 | 15 / 07 / 22 | 3. Elaborar diagrama de casos de uso. | Diagrama de casos de uso do sistema criado. |
| 16 /07/ 22 | 17 / 07 /22 | 4. Definição dos requisitos não funcionais. | Lista com a definição dos requisitos não funcionais importantes para a implementação do sistema. |
| 18 / 07 /22 | 23/ 07 / 22 | 5. Elaborar protótipo navegável do sistema e vídeo de apresentação do mesmo. | Protótipo navegável elaborado, mostrando a tela inicial do sistema e os três casos de uso principais e vídeo de apresentação do protótipo. |
| 24 /07 / 22 | 26/ 07 / 22 | 6. Elaborar o diagrama de classes de domínio de todo o sistema. | Diagrama de classes de domínio documentado. |
| 27 / 07 / 22 | 01 / 08 / 22 | 7. Criar modelo de componentes do sistema. | Documentação do modelo de componentes de todo o sistema, contendo: a indicação do padrão arquitetural escolhido, o diagrama de componentes bem como a descrição desses componentes. |
| 02 / 08 / 22 | 05 / 08 / 22 | 8. Criar diagrama de implantação do sistema. | Diagrama de implantação criado. |
| 06/ 08 / 22 | 10 / 08 / 22 | 9. Definir plano de testes para os três casos de uso contemplados no protótipo navegável do sistema. | Plano de testes documentado. |
| 11/ 08 / 22 | 23 / 08 / 22 | 10. Estimar pontos de função. | Planilha de pontos de função do sistema documentada. |
| 24 / 08 / 22 | 29 / 08 / 22 | 12. Revisar documentação elaborada. | Documentação revisada. |
| 30 / 08 / 22 | 30 / 08 / 22 | 13. Finalizar entrega do TCC. | Entrega do TCC na plataforma AVA. |

## Diagrama de casos de uso



## Requisitos não-funcionais

RNF1 - O sistema deve utilizar cores para facilitar a identificação de tarefas atrasadas, bem como a visualização de projetos que estão em pior situação.

RNF2 - O sistema deve suportar, pelo menos, 50 usuários conectados simultaneamente.

RNF3 - O sistema deve realizar o controle de segurança baseado em perfis de acesso.

RNF4 - O sistema deve garantir a segurança das senhas dos usuários, criptografando-as ao serem inseridas no banco de dados.

RNF5 - O sistema deve poder ser acessado por meio da web e dispositivos móveis tais como tablets e smartphones. Portanto deve possuir design responsivo que garanta o comportamento adequado da interface, independentemente de onde será acessado.

RNF6- O sistema deverá ter alta disponibilidade, estando disponível em pelo menos 99% do tempo.

RNF7 - O sistema deve possuir compatibilidade com os sistemas operacionais Windows, Linux e Mac OS.

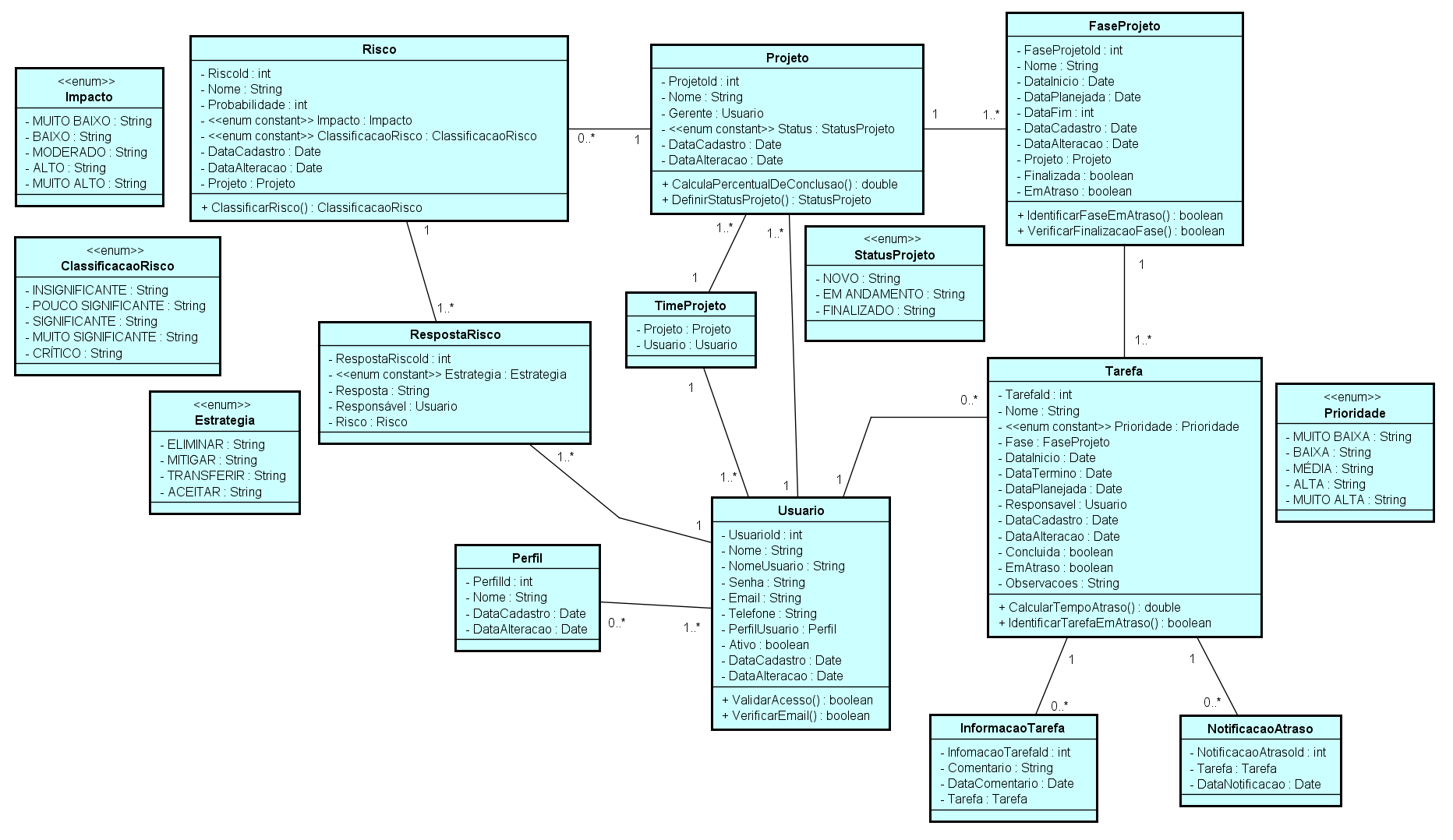
RNF8 - O projeto de software deve priorizar o baixo acoplamento e alta coesão, garantido uma melhor separação de responsabilidades.

## Protótipo navegável do sistema

O protótipo navegável do sistema bem e o seu vídeo de apresentação estão disponíveis no GitHub através do link:

<https://github.com/gdsantos86/TCC_Eng_Software_Puc/tree/main/TCC/Prot%C3%B3tipo>

## Diagrama de classes de domínio



## Modelo de componentes

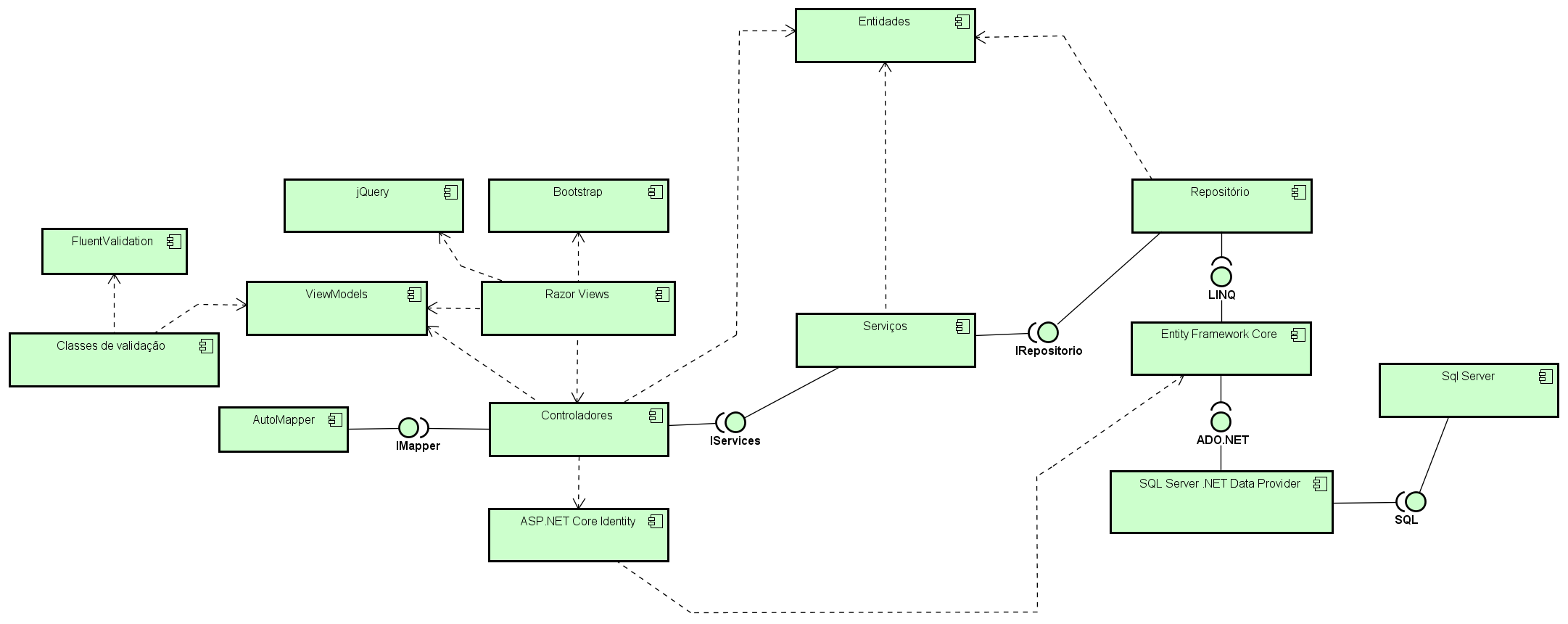
### Padrão arquitetural

O padrão arquitetural escolhido para o desenvolvimento do sistema foi o padrão em Camadas (*Layers*), que será utilizado em conjunto com o padrão MVC. As camadas serão divididas da seguinte forma: camada de apresentação, camada de negócios, camada de acesso a dados e camada de domínio/entidades. O padrão MCV será aplicado na camada de apresentação. Espera-se com a escolha desse tipo de arquitetura, alcançar os seguintes objetivos: realizar a separação de responsabilidades, facilitar manutenção do código, facilitar a realização de testes e o trabalho de desenvolvimento em equipe.

As tecnologias utilizadas na implementação na implementação da proposta serão as seguintes:

* .NET Core;
* Asp.NET Core MVC;
* Razor;
* C#;
* ASP.NET Identity;
* Entity Framework Core (abordagem Code-first);
* ADO.NET.

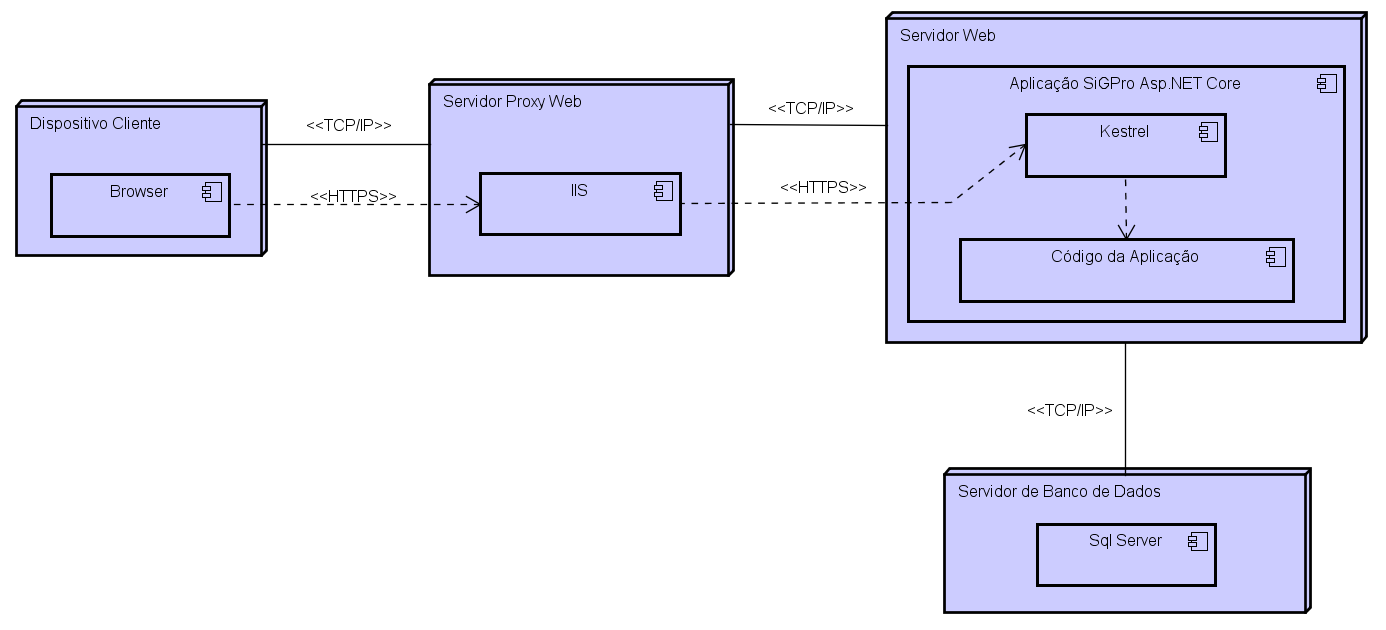
### Diagrama de componentes



### Descrição dos componentes

|  |  |  |
| --- | --- | --- |
| **Número** | **Componente** | **Descrição** |
| 1 | Bootstrap | Biblioteca gratuita utilizada na criação das páginas HTML visando melhorar a qualidade, organização, responsividade e portabilidade das telas. |
| 2 | JQuery | Componente *open-source* a ser utilizado principalmente para facilitar a manipulação de eventos nas views. |
| 3 | Razor Views | Páginas Web criadas como arquivos .cshtml que utilizam a sintaxe *Razor* para simplificar a codificação e torná-la mais legível. |
| 4 | ViewModels | Classes C# que serão criadas para agrupar informações de uma ou mais entidades de domínio, com a intenção de organizar e gerenciar os dados que serão exibidos nas *views*. |
| 5 | Classes de Validação | Conjunto de classes que serão desenvolvidas para realizar a validação das ViewModels (dados de entrada do sistema) por meio da FluentValidation. |
| 6 | FluentValidation | Biblioteca gratuita utilizada na criação de regras para validação de objetos. |
| 7 | AutoMapper | Biblioteca *open-source* que será utilizada para fazer o mapeamento entre as entidades de domínio e as viewmodels*.* |
| 8 | Controladores | Classes C# que serão desenvolvidas para conter a lógica de controle de fluxo da aplicação, sendo assim responsáveis por controlar a maneira como o usuário irá interagir com o sistema. |
| 9 | ASP.NET Core Identity | API que dá suporte a funcionalidade de login da aplicação. |
| 10 | Serviços | Conjunto de classes C# a serem desenvolvidas para implementação dos serviços da aplicação e das regras globais do sistema. Compõem a camada que se comunica com a interface do usuário e com o repositório. |
| 11 | Repositório | Conjunto de classes C# a serem desenvolvidas, que implementam o padrão de projeto *Repository*, com o propósito de realizar o isolamento entre as camadas de acesso a dados e as camadas de regras de negócio e de apresentação. |
| 12 | Entity Framework Core | Tecnologia de acesso a dados responsável pelo mapeamento objeto-relacional. Software livre. |
| 13 | SQL Server .NET Data Provider | Conjunto de classes disponibilizadas para acesso ao banco de dados SQL Server. |
| 14 | SQL Server | Banco de dados da aplicação. Será adquirido. |
| 15 | Entidades | Classes C# a serem desenvolvidas, que representarão o modelo de domínio da aplicação com suas regras de negócio. |

## Diagrama de implantação



## Plano de Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  | Cadastrar Projeto | Validar o cadastro com sucesso de um novo projeto. | - Logar no sistema com um usuário previamente cadastrado.  - Clicar em “Projetos” no menu à esquerda, na tela principal.  - Após acessar a tela de listagem de projetos, clicar no botão “Inserir”;  - Na tela de cadastro de projetos, preencher o campo “Nome”, selecionar o “Gerente” e configurar o “Time” do projeto;  - Confirmar o cadastro do projeto clicando em “Inserir”. | - Mensagem de sucesso informando que o projeto foi cadastrado com sucesso;  - Os botões “Inserir Fases” e “Gerenciar Riscos” são habilitados para permitir o cadastro dos mesmos. |
|  | Cadastrar Projeto | Verificar que não é possível inserir um novo projeto no sistema, sem o preenchimento dos campos obrigatórios. | - Realizar o processo de cadastro informado no caso de teste 1, mas deixando de preencher o campo “Nome”.  - Clicar em “Inserir”. | - Mensagem de erro informando que existem campos obrigatórios que não foram preenchidos e que, portanto, o cadastro do projeto não foi efetivado.  - Os botões “Inserir Fases” e “Gerenciar Riscos” não devem ser habilitados. |
|  | Cadastrar Tarefa | Validar cadastro com sucesso de uma nova tarefa. | - Logar no sistema.  - Clicar em “Projetos” no menu à esquerda, na tela principal.  - Após acessar a tela de listagem de projetos, escolher um dos projetos da lista que não esteja com o status de “Finalizado” e clicar no botão “Fases”;  - Após acessar a tela de listagem de fases do projeto, escolher uma fase que não tenha sido finalizada e clicar no botão “Tarefas”;  - Após acessar a tela de listagem de tarefas, clicar no botão “Inserir”;  - Na tela de cadastro de tarefa, preencher/selecionar os campos “Nome”, “Responsável”, “Início” “Fim Planejado” e “Prioridade”;  - Confirmar o cadastro da tarefa clicando em “Inserir”. | - O sistema retorna para a listagem de tarefas e emite uma mensagem de sucesso informando que a tarefa foi cadastrada com êxito. |
|  | Cadastrar Tarefa | Verificar que não é possível inserir uma nova tarefa para um determinado projeto, sem o preenchimento dos campos obrigatórios. | - Realizar o processo de cadastro informado no caso de teste 3, mas deixando de preencher/selecionar um dos seguintes campos obrigatórios: “Nome”, “Responsável”, “Início” “Fim Planejado” e “Prioridade”;  - Clicar em “Inserir”. | - Mensagem de erro informando que existem campos obrigatórios que não foram preenchidos e que, portanto, o cadastro da tarefa não foi efetivado.  - O sistema permanece na tela de cadastro de tarefa. |
|  | Cadastrar Usuário | Validar o cadastro com sucesso de um novo usuário. | - Logar no sistema.  - Clicar em “Usuários” no menu à esquerda, na tela principal.  - Após acessar a tela de listagem de usuários, clicar no botão “Inserir”;  - Na tela de cadastro de usuários, preencher/selecionar os campos: Nome, Usuário, Senha, Confirmação de Senha, E-mail e Perfil;  - Confirmar o cadastro do usuário clicando em “Inserir”. | - O sistema retorna para a listagem de usuários e emite uma mensagem de sucesso informando que o usuário foi cadastrado com êxito. |
|  | Cadastrar Usuário | Verificar que não é possível inserir um novo usuário no sistema, sem o preenchimento dos campos obrigatórios. | - Realizar o processo de cadastro informado no caso de teste 5, mas deixando de preencher/selecionar um dos seguintes campos obrigatórios: Nome, Usuário, Senha, Confirmação de Senha, E-mail e Perfil;  - Clicar em “Inserir”. | - Mensagem de erro informando que existem campos obrigatórios que não foram preenchidos e que, portanto, o cadastro do usuário não foi efetivado.  - O sistema permanece na tela de cadastro de usuário. |

## Estimativa de pontos de função

A planilha de estimativa de pontos de função do sistema está disponível no GitHub através do link:

<https://github.com/gdsantos86/TCC_Eng_Software_Puc/tree/main/TCC/Estimativa%20de%20Pontos%20de%20Fun%C3%A7%C3%A3o>

## Referências

LARMAN, Craig. **Utilizando UML e Padrões: Uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo.** Tradução de Rosana T. Vaccare Braga *et al*. 3. ed. Porto Alegre: Bookman, 2007.

MENDES, Marco Aurélio de Souza. **Arquitetura de Sistemas Web: Princípios, Práticas e Tecnologias.** [*S. l.: s. n.*], 2020.

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira; ALBERT, Renato Machado. **Análise de Pontos de Função: Medição, Estimativas e Gerenciamento de Projetos de Software.** São Paulo: Saraiva Educação, 2018.