**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Trabalho de Conclusão de Curso

SisGePro – Sistema de Gerenciamento de Projetos

Matheus Santos de Oliveira

Belo Horizonte

Abril/2022.

# Trabalho de Conclusão de Curso

**Sumário**

[Trabalho de Conclusão de Curso 3](#_Toc101042540)

[1. Cronograma de trabalho 4](#_Toc101042541)

[2. Diagrama de casos de uso 5](#_Toc101042542)

[3. Requisitos não-funcionais 5](#_Toc101042543)

[4. Protótipo navegável do sistema 6](#_Toc101042544)

[5. Diagrama de classes de domínio 7](#_Toc101042545)

[6. Modelo de componentes 8](#_Toc101042546)

[6.1. Padrão arquitetural 8](#_Toc101042547)

[6.2. Diagrama de componentes 8](#_Toc101042548)

[6.3. Descrição dos componentes 8](#_Toc101042549)

[7. Diagrama de implantação 9](#_Toc101042550)

[8. Testes 11](#_Toc101042551)

[9. Estimativa de pontos de função 16](#_Toc101042552)

[10. Informações da implementação 17](#_Toc101042553)

[11. Referências 18](#_Toc101042554)

## Cronograma de trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 01/03/22 | 07/03/22 | 1. Desdobramento de requisitos | 1. Lista dos requisitos funcionais 2. Lista dos requisitos não-funcionais |
| 08/03/22 | 15/03/22 | 1. Desenho do diagrama de casos de uso | 1. Diagrama de casos de uso |
| 16/03/22 | 23/03/22 | 1. Rascunho do protótipo navegável do sistema | 1. Protótipo navegável do sistema |
| 24/03/22 | 30/03/22 | 1. Desenho do diagrama de classes de domínio | 1. Diagrama de classes de domínio |
| 01/04/22 | 05/04/22 | 1. Definição do modelo de componentes | 1. Diagrama de componentes |
| 06/04/22 | 08/04/22 | 1. Desenho do diagrama de implantação | 1. Diagrama de implantação |
| 09/04/22 | 12/04/22 | 1. Elaboração do plano de testes | 1. Plano de testes |
| 13/04/22 | 16/04/22 | 1. Cálculo dos pontos de função | 1. Planilha de pontos de função |

## Diagrama de casos de uso

|  |
| --- |
|  |

## Requisitos não-funcionais

REQ-NF-SGP-001: O sistema deve ser capaz de atender requisições do usuário em até 150ms.

REQ-NF-SGP-002: O sistema deve atender acesso simultâneos de, ao menos, 50 usuários diferentes.

REQ-NF-SGP-003: O sistema deve apresentar disponibilidade de, ao menos, 99% das 05h às 21h, entre segunda-feira e sexta-feira.

REQ-NF-SGP-004: O sistema deve apresentar disponibilidade de, ao menos, 80%, entre sábado e domingo.

REQ-NF-SGP-005: O sistema deve garantir a confidencialidade das informações através de criptografia dos dados de usuários.

REQ-NF-SGP-006: O sistema deve possuir implementação compatível com navegadores presentes a partir da versão 7.0 do sistema Android.

REQ-NF-SGP-007: O sistema deve possuir implementação compatível com navegadores presentes a partir da versão 10.0 do sistema iOS.

REQ-NF-SGP-008: O sistema deve ser compatível com o navegador Google Chrome a partir de sua versão 55.0.2883.103.

REQ-NF-SGP-009: O sistema deve ser compatível com o navegador Mozilla Firefox a partir de sua versão 50.1.0.

REQ-NF-SGP-010: O sistema deve ser capaz de ser completamente utilizado por um usuário após de treinamento de três horas.

## Protótipo navegável do sistema

O protótipo navegável pode ser acessado através do seguinte link:

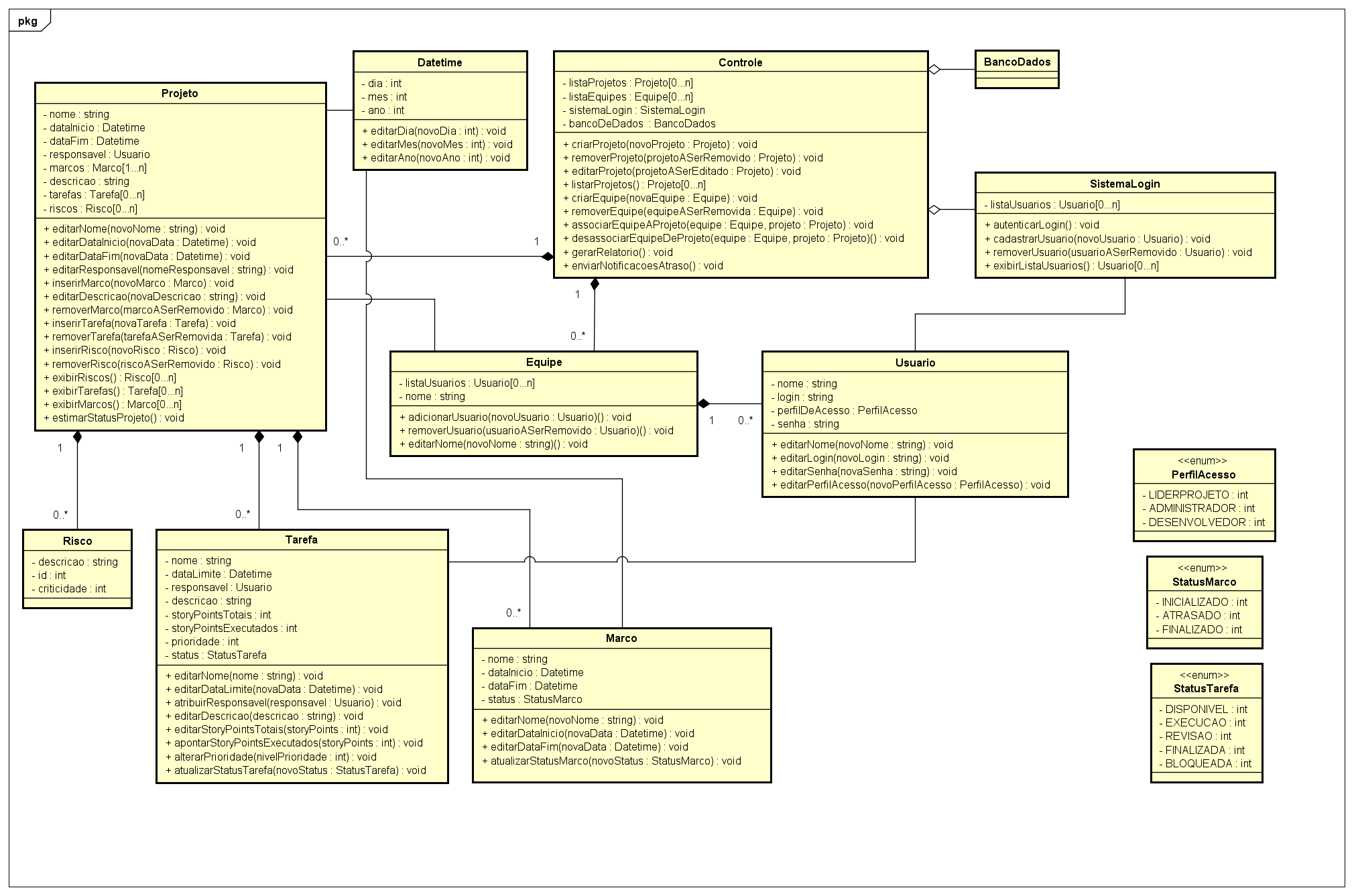
<https://www.figma.com/file/HGQLXTg2PLuMorpbGMM3HN/TCC1?node-id=0%3A1>

O vídeo de apresentação encontra-se no seguinte link:

Esse vídeo deve ter duração de, no máximo, 3 minutos. Utilize, preferencialmente, o formato AVI.

## Diagrama de classes de domínio

O arquivo original da imagem abaixo pode ser encontrado em:



## Modelo de componentes

### Padrão arquitetural

O padrão arquitetural utilizado no desenvolvimento da aplicação será o MVVM (Model, View, View-Model) por apresentar os seguintes benefícios:

* Desacoplamento entre a camada lógica e a camada de apresentação;
* Testabilidade;
* Facilidade de manutenção.

Serão utilizadas as seguintes tecnologias:

* Angular (Front-end)
* Bootstrap (Front-end)
* Node.JS (Back-end)
* Postman (Testes)
* PostgreSQL (Banco de dados)
* NGINX (Servidor web)
* OAuth2 (Autenticação)

### Diagrama de componentes

O arquivo original da imagem abaixo pode ser encontrado em:

Diagrama

Descrição gerada automaticamente

### Descrição dos componentes

|  |  |  |
| --- | --- | --- |
| **Número** | **Componente** | **Descrição** |
|  | Módulos de gerenciamento | São as páginas web desenvolvidas para visualização do sistema pelo usuário.  Correspondem ao *View*. |
|  | Framework Web | Framework responsável por acelerar o desenvolvimento dos módulos de gerenciamento e proporcionar sua comunicação com o servidor.  Corresponde ao *View-Model*.  Não necessita de aquisição, já que o Angular é uma plataforma de código aberto. |
|  | Servidor Javascript | Responsável pelas regras de negócio e pela manipulação dos dados da aplicação.  Corresponde ao *Model*.  Não necessita de aquisição, já que o NodeJS é um software de código aberto. |
|  | Servidor Web | Responsável por servir o sistema e prover respostas para requisições web.  Não necessita de aquisição, já que o NGINX é um software de código aberto. |
|  | Banco de dados | Responsável pela persistência dos dados da aplicação, através da manipulação das tabelas e registros do banco de dados.  Não necessita de aquisição, já que o PostgreSQL é um software de código aberto. |
|  | Módulo de autenticação | Responsável pela autenticação dos dados providos pelo usuário.  Não necessita de aquisição, já que o OAuth2 é um software de código aberto. |

## Diagrama de implantação

Diagrama

Descrição gerada automaticamente

## Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  | Cadastrar projeto | Verificar se um novo projeto é cadastrado corretamente no sistema. | a) Realizar login no sistema com um usuário com permissão de administrador.  b) Clicar no botão “Projetos”.  c) Clicar no botão “Criar”.  d) Preencher os campos do pop-up exibido com as informações a seguir:  Nome: Projeto TCC  Data Inicial: 01/01/2022  Data Final: 01/01/2023  Responsável: Líder Projeto 1  Descrição: Projeto de exemplo.  e) Clicar no botão “Adicionar Projeto” | a) Projeto de nome “Projeto TCC” cadastro e exibido corretamente. |
|  | Cadastrar projeto | Verificar se os campos de preenchimento necessários para cadastrar um novo projeto são corretamente preenchidos. | a) Realizar login no sistema com um usuário com permissão de administrador.  b) Clicar no botão “Projetos”.  c) Clicar no botão “Criar”.  d) Preencher os campos do pop-up exibido com as informações a seguir:  Nome: Projeto TCC  Data Inicial: 01/01/2022  Data Final: 01/01/2023  Responsável: “não preencher”  Descrição: Projeto de exemplo.  e) Clicar no botão “Adicionar Projeto” | a) Um pop-up é exibido contendo a mensagem “Todos os campos devem ser preenchidos.”  b) O campo “Responsável” deve ser preenchido em cor vermelha para indicar o erro.  c) O projeto não é cadastrado. |
|  | Cadastrar tarefa | Verificar se uma nova tarefa é cadastrada corretamente em um projeto. | a) Realizar login no sistema com um usuário com permissão de administrador.  b) Clicar no botão “Projetos”.  c) Selecionar o “Projeto A” clicando no botão “Projeto A”.  d) Clicar no botão “Adicionar Tarefa”.  e) Preencher os campos do pop-up exibido com as informações a seguir:  Nome: Tarefa 14  Data Limite: 01/01/2023  Story Points: 5  Prioridade: Baixa  Responsável: Dev 2  Descrição: Tarefa de exemplo.  f) Clicar no botão “Criar Tarefa” | a) Tarefa de nome “Tarefa 14” criada e adicionada no card “To-do”. |
|  | Cadastrar tarefa | Verificar se os campos de preenchimento necessários para cadastrar uma nova tarefa são corretamente preenchidos. | a) Realizar login no sistema com um usuário com permissão de administrador.  b) Clicar no botão “Projetos”.  c) Selecionar o “Projeto A” clicando no botão “Projeto A”.  d) Clicar no botão “Adicionar Tarefa”.  e) Preencher os campos do pop-up exibido com as informações a seguir:  Nome: Tarefa 14  Data Limite: 01/01/2023  Story Points: “não preencher”  Prioridade: Baixa  Responsável: Dev 2  Descrição: Tarefa de exemplo.  f) Clicar no botão “Criar Tarefa” | a) Um pop-up é exibido contendo a mensagem “Todos os campos devem ser preenchidos.”  b) O campo “Story Points” deve ser preenchido em cor vermelha para indicar o erro.  c) A tarefa não é criada. |
|  | Gerar Relatório | Verificar se o relatório de um projeto é gerado corretamente. | a) Realizar login no sistema com um usuário com permissão de administrador.  b) Clicar no botão “Projetos”.  c) Selecionar o “Projeto A” clicando no botão “Projeto A”.  d) Clicar no botão “Gerar Relatório”. | a) A tela com o relatório do projeto é exibida. |
|  | Gerar Relatório | Verificar se o relatório de um projeto é exportado corretamente. | a) Realizar login no sistema com um usuário com permissão de administrador.  b) Clicar no botão “Projetos”.  c) Selecionar o “Projeto A” clicando no botão “Projeto A”.  d) Clicar no botão “Gerar Relatório”.  e) Clicar no botão “Exportar Relatório”.  f) Deixar o nome do arquivo vazio.  g) Clicar em “Salvar” | a) Um pop-up é exibido pedindo o preenchimento do nome do arquivo.  b) O relatório não é exportado. |

## Estimativa de pontos de função

<Indique o *link* do repositório (como o [GitHub](http://github.com), por exemplo) onde sua planilha de estimativa de pontos de função está disponível.>

## Informações da implementação

Não foi realizada implementação.

## Referências

Para a elaboração deste trabalho foi utilizado os vídeos e notas de aula das disciplinas oferecidas no curso de Pós-graduação em Engenharia de Software da PUC-MG.