

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

PUC Minas Virtual

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

Trabalho de Conclusão de Curso

Project Management Tool

Ian Conde Collares

Belo Horizonte
Outubro, 2021.

Trabalho de Conclusão de Curso

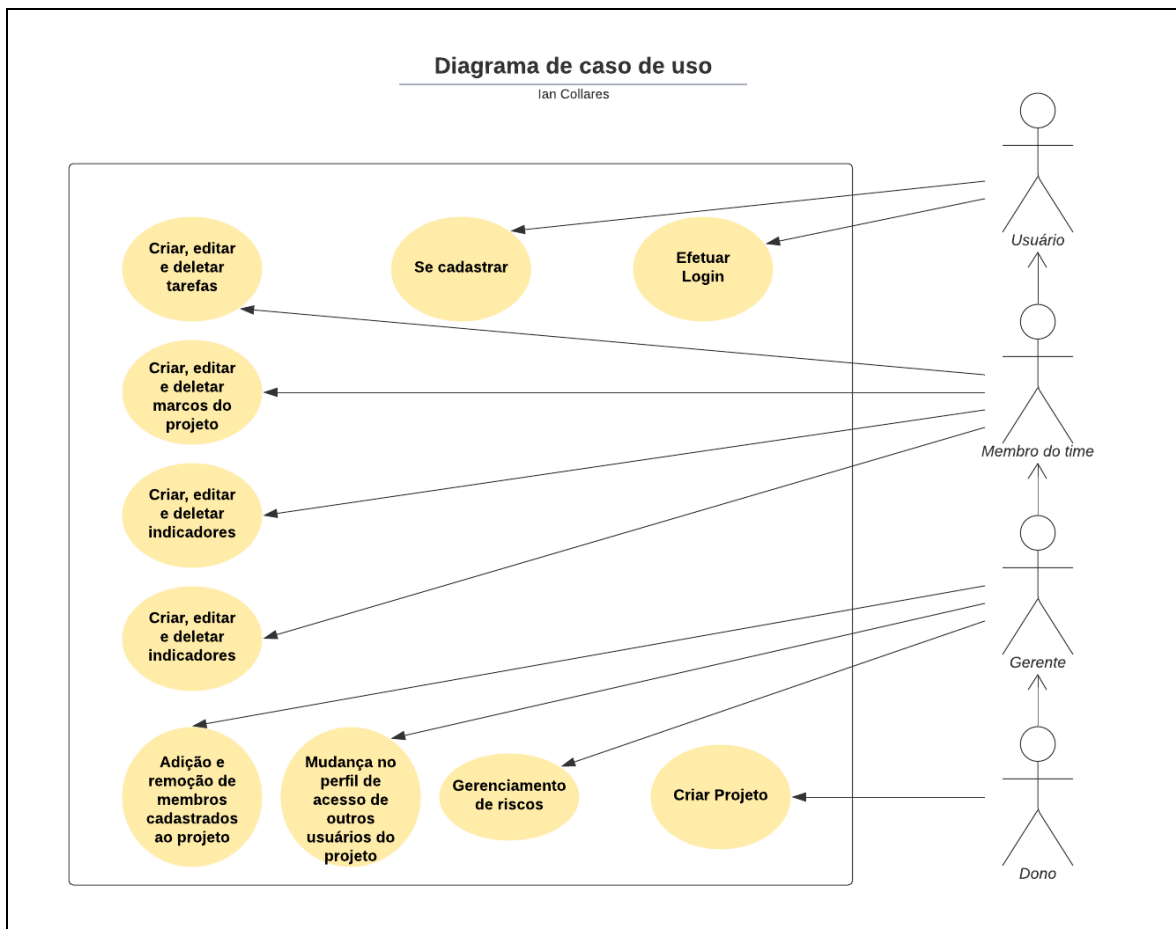
Sumário

Trabalho de Conclusão de Curso	2
1. Cronograma de trabalho	3
2. Diagrama de casos de uso	4
3. Requisitos não-funcionais	5
4. Protótipo navegável do sistema	6
5. Diagrama de classes de domínio	7
6. Modelo de componentes	8
6.1. Padrão arquitetural	8
6.2. Diagrama de componentes	9
6.3. Descrição dos componentes	10
7. Diagrama de implantação	12
8. Plano de Testes	13
9. Estimativa de pontos de função	15
10. Informações da implementação	15
11. Referências	15

1. Cronograma de trabalho

Datas		Atividade / Tarefa	Produto / Resultado
De	Até		
01 / 09 / 21	06 / 09 / 21	1. Construção dos casos de uso	Diagrama de casos de uso
06 / 09 / 21	07 / 09 / 21	2.Elicitação de requisitos não-funcionais	Requisitos não-funcionais
07 / 09 / 21	13 / 09 / 21	3.Prototipação	Protótipo navegável com 3 casos de uso principais
13 / 09 / 21	14 / 09 / 21	4.Vídeo do protótipo	Vídeo com 3 casos de uso principais disponível em repositório na internet
14 / 09 / 21	21 / 09 / 21	5. Produção de diagramas de classes de domínio	Diagramas de classes de domínio
21 / 09 / 21	23 / 09 / 21	6.Definição de padrão arquitetural	Padrão arquitetural
23 / 09 / 21	30 / 09 / 21	7.Produção do diagrama de componentes	Diagrama de Componentes
30 / 09 / 21	01 / 10 / 21	8.Descrição dos componentes	Componentes descritos
01 / 10 / 21	07 / 10 / 21	9.Produção do Diagrama de Implantação	Diagrama de Implantação
07 / 10 / 21	14 / 10 / 21	10.Estimativa de Pontos de Função	Planilha de Análise de Pontos de Função

2. Diagrama de casos de uso



Também disponível em: <https://github.com/Ian021/pucminas>

3. *Requisitos não-funcionais*

- O sistema Web deve ser responsivo, proporcionando a utilização de todas as suas funcionalidades em resoluções de mobile, tablet e desktop.
- O sistema deve rodar em celulares com sistema operacional Android e iOS.
- O sistema Web deve rodar nos cinco navegadores mais populares entre os usuários.
- O sistema deve ser facilmente escalável para suportar mais usuários conforme a demanda.
- O sistema deve ter disponibilidade de pelo menos 99%.
- O sistema deverá garantir a segurança dos dados dos usuários conforme a legislação vigente.
- O sistema deve seguir boas práticas de acessibilidade.
- O sistema deverá contar com um login seguro, utilizando hash nas senhas dos usuários, e/ou fazendo uso do serviço de empresas de referência.
- O sistema deve ter um controle de segurança baseado em perfis de acesso.

4. Protótipo navegável do sistema

Foi desenvolvido um protótipo de Aplicativo Mobile, levando em consideração a facilidade de gestão de um projeto à distância, somente com um celular. A aplicação foi desenhada de modo a ser intuitiva e auto explicativa, utilizando ícones de conhecimento geral, além de uma fácil navegação e visualização do estado do sistema.

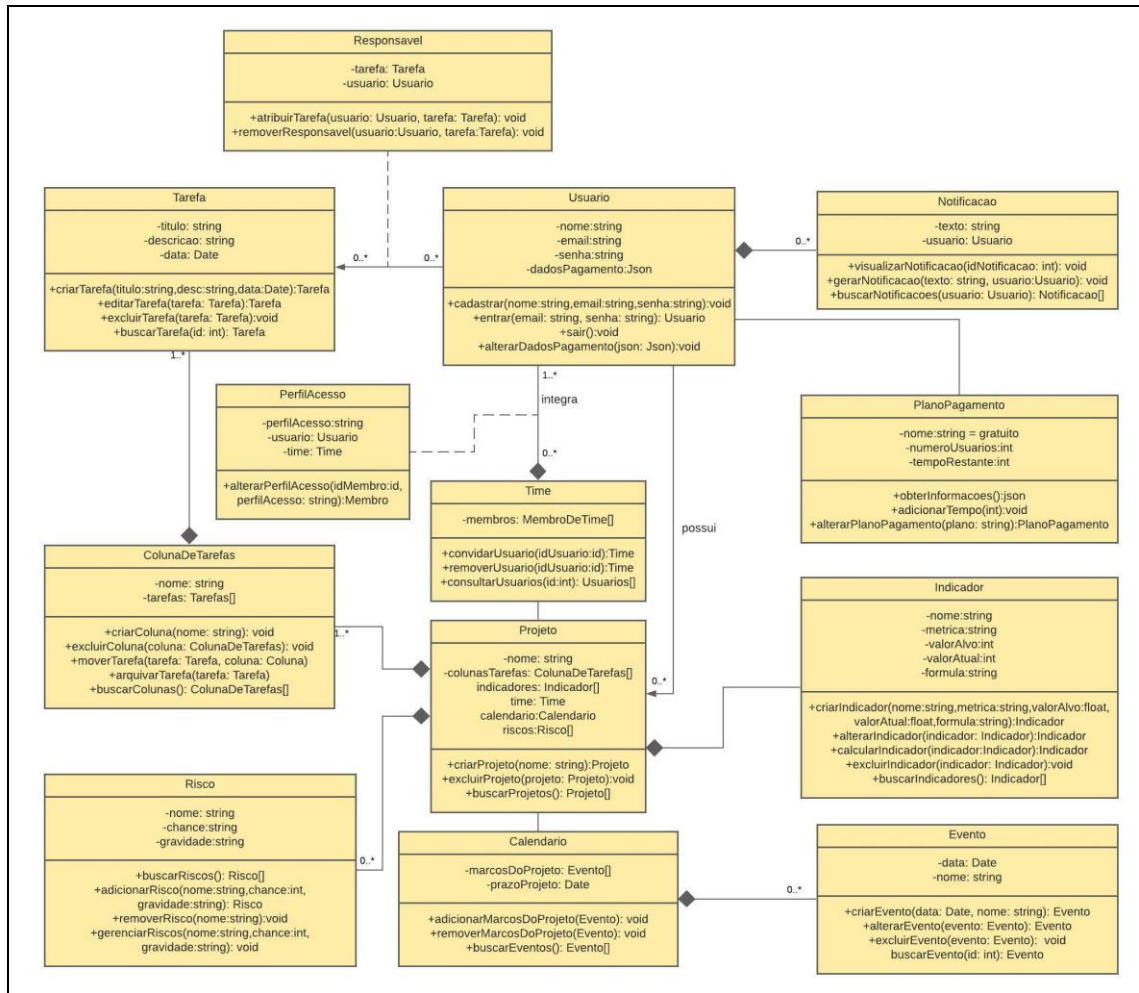
Protótipo navegável do sistema:

<https://www.figma.com/proto/038p1D0GFBZiBA2Pqf3qoj/Prototype?page-id=114%3A263&node-id=114%3A263&viewport=241%2C48%2C0.39&scaling=min-zoom&starting-point-node-id=114%3A264>

Repositório com vídeo dos três casos de uso principais:

<https://github.com/Ian021/pucminas>

5. Diagrama de classes de domínio



Também disponível em: <https://github.com/Ian021/pucminas>

6. Modelo de componentes

6.1. Padrão arquitetural

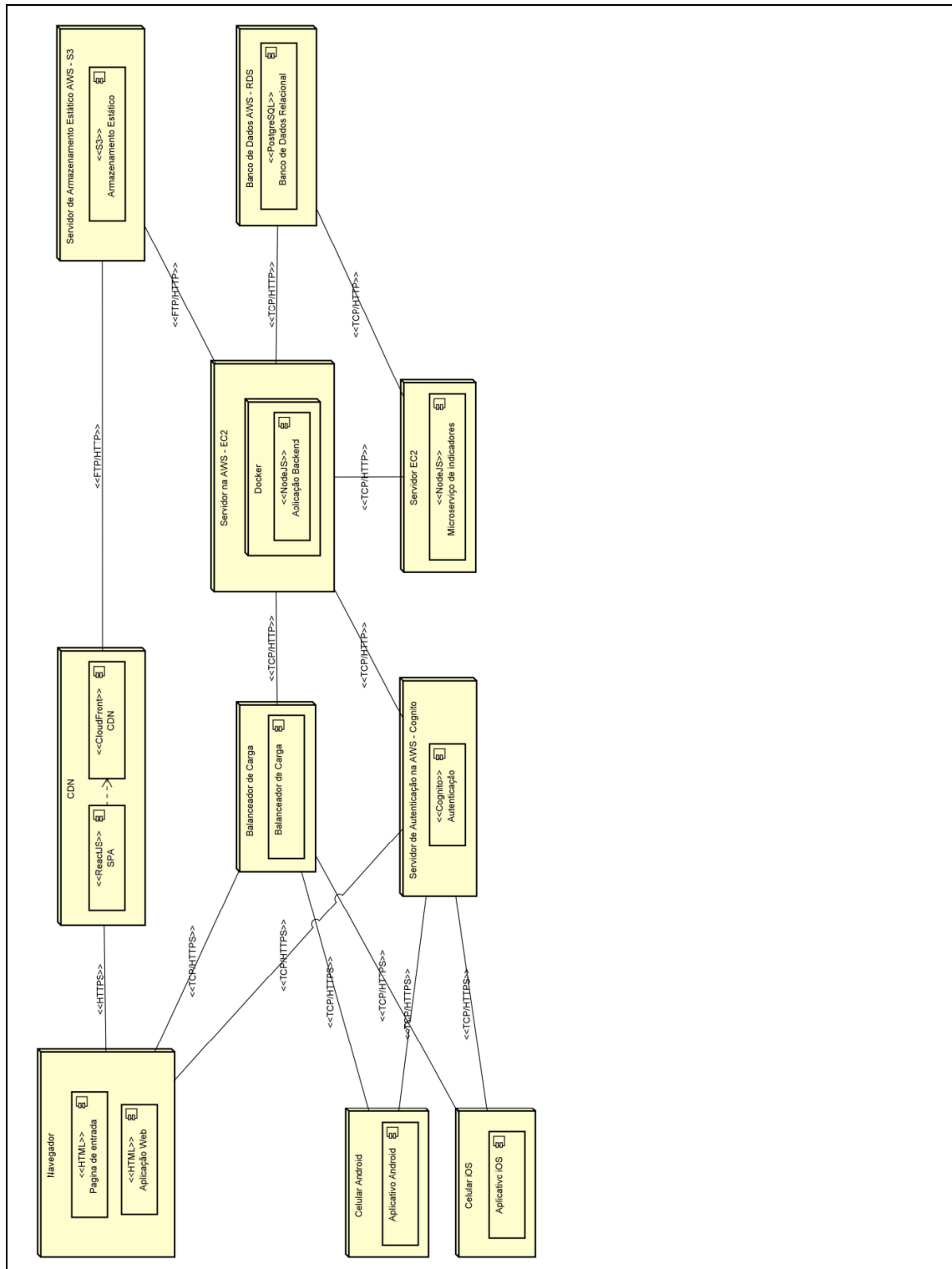
Foi escolhido o padrão MVVM para a aplicação, de modo que o tempo de resposta do sistema e o consumo de dados sejam baixos. A linguagem de programação predominante foi Javascript, utilizando o framework React para a aplicação web e React-Native para o aplicativo Mobile. Nos servidores foi utilizado o NodeJS, utilizando o framework Express. O sistema foi planejado para funcionar com a estrutura de nuvem da AWS, fazendo uso de diversos recursos para tornar o sistema seguro, escalável e altamente disponível.

6.3. Descrição dos componentes

Número	Componente	Descrição
1	Página de entrada	Página onde há informações, vídeos promocionais, informações de contato e a tela de login. É executada no navegador do usuário.
2	Aplicação Web	Área do site acessada após o login na aplicação. É executada no navegador do usuário.
3	SPA	Componente desenvolvido utilizando ReactJS. Gera as Views no padrão MVVM.
4	State Manager	Componente desenvolvido utilizando Redux .Faz a gestão dos dados na aplicação web. Faz o papel do View-Model no padrão MVVM.
5	Estilo das Páginas	Componente adquirido para a padronização do layout e estilo. Possui componentes prontos para uso no frontend.
6	Aplicativo Android	Componente produzido a partir do componente <i>Aplicativo Híbrido</i> produzido com React-Native. Aplicativo mobile executado em dispositivos com o sistema operacional Android.
7	Aplicativo iOS	Similar ao componente <i>Aplicativo Android</i> . Aplicativo mobile executado em dispositivos com o sistema operacional iOS.
8	Aplicativo Híbrido	Produzido com a linguagem de programação Javascript, utilizando o React-Native para gerar as telas da aplicação mobile. Capaz de gerar o <i>Aplicativo Android</i> e <i>Aplicativo iOS</i> .
9	Autenticação	Componente de autenticação da AWS, Cognito. Utilizado para maior segurança do usuário e facilidade de manutenção.
10	Balanceador de Carga	Componente da AWS. Responsável por distribuir as requisições das Aplicações Web e Mobile entre servidores.
11	CDN	Componente da AWS, CloudFront. Utilizado para distribuir a Aplicação Web de forma rápida.
12	Proxy Reverso	Componente desenvolvido com a biblioteca Nginx. Responsável por receber as requisições e direcioná-las para suas devidas rotas dentro da

		aplicação Backend.
13	Aplicação Backend	Componente desenvolvido com NodeJS, utilizando o framework Express. Responsável por responder as requisições feitas pelo frontend. É onde estão presentes os Models no padrão MVVM.
14	Logs	Componente da AWS, CloudWatch. Utilizado para gravar e visualizar os logs de vários componentes da aplicação.
15	Armazenamento Estático	Componente da AWS, S3. Utilizado para salvar arquivos estáticos servidos para a aplicação backend, assim como armazenar o código pelo ReactJS, a serem servidos pelo CDN. Também é onde os aplicativos gerados pelo React-Native são salvos após gerados e testados, até serem manualmente colocados em produção na Play Store e na App Store.
16	Banco de Dados Relacional	Componente desenvolvido utilizando a estrutura da AWS, RDS, utilizando o PostgreSQL. Armazena informações dinâmicas para a aplicação.
17	Microserviço de Indicadores	Componente desenvolvido utilizando NodeJS e o framework Express. Utilizado para isolar os cálculos pesados da aplicação principal. Os cálculos são feitos no servidor e não no banco de dados para não congestionar o banco de dados.
18	Repositório de Código.	Componente da AWS, CodeCommit. Utilizado com o Git para salvar o código com versionamento na Nuvem e garantir que todos os desenvolvedores tem acesso à mesma versão de código.
19	Integração Contínua	Componente da AWS, CodeBuild. Testa e gera o código de produção da aplicação conforme atualizações são finalizadas pelos desenvolvedores.
20	Pipeline de Deploy	Componente da AWS, CodePipeline. Coordena e estabelece as regras da <i>Integração Contínua</i> a partir do <i>Repositório de Código</i> .
21	Testes Mobile	Componente da AWS, DeviceFarm. Utilizado para testar automaticamente os aplicativos iOS e Android gerados pelo <i>Aplicativo Híbrido</i> .

7. Diagrama de implantação



Também disponível em: <https://github.com/Ian021/pucminas>

8. Plano de Testes

Número	Caso de uso	Objetivo do caso de teste	Entradas	Resultados esperados
1	Criação de Projeto	Caso o usuário tente criar um projeto com o campo nome do projeto válido, verificar que o projeto foi criado com sucesso.	<ul style="list-style-type: none"> - Usuário: u1 - Nome do Projeto: Meu Projeto 	Projeto criado com sucesso. O usuário é direcionado para a página de configurações
2	Criação de Projeto	Caso o usuário tente criar um projeto com o campo nome do projeto vazio, retornar mensagem de erro informando que o campo nome o projeto não pode ser vazio e verificar que o projeto não é criado.	<ul style="list-style-type: none"> - Usuário: u1 - Nome do Projeto: <<String Vazia>> 	Projeto não é criado. Mensagem de erro informando que o campo nome do projeto não pode ser vazio
3	Adição de usuário ao projeto	Caso o usuário u1 , com perfil de acesso de Manager em um time t1 , tente adicionar o usuário u2 ao time t1 , verificar que o usuário u2 foi adicionado com sucesso ao time t1 .	<ul style="list-style-type: none"> - Usuário: u1 - Perfil de usuário de u1: Manager - Time: t1 - Segundo Usuário: u2 	Usuário é adicionado ao time do projeto com sucesso.
4	Adição de usuário ao projeto	Caso o usuário u1 , com perfil de acesso User em um time t1 , tente adicionar o usuário u2 ao time t1 , retornar uma mensagem de erro ao usuário u1 informando que ele não tem autorização para realizar esta ação e verificar que o usuário u2 não foi adicionado ao time t1 .	<ul style="list-style-type: none"> - Usuário: u1 - Perfil de usuário de u1: User - Time: t1 - Segundo Usuário: u2 	Usuário não é adicionado ao time Mensagem de erro indicando que o usuário não tem autorização para realizar esta ação
5	Criação de tarefas	Caso um usuário tente criar uma tarefa com seus campos devidamente	<ul style="list-style-type: none"> - Usuário: u1 - Título: Minha Tarefa 	Tarefa criada com sucesso com os valores definidos

		preenchidos, verificar que a tarefa é criada e que seus campos estão corretos.	<ul style="list-style-type: none"> - Descrição: Testar aplicação - Responsáveis: [u1] - Prazo: 15/10/21 	pelo usuário.
6	Criação de tarefas	Caso o usuário tente criar uma tarefa sem título, retornar uma mensagem de erro ao usuário informando que o campo título é obrigatório e verificar que a tarefa não é criada.	<ul style="list-style-type: none"> - Usuário: u1 - Título: <<String Vazia>> - Descrição: Testar aplicação - Responsáveis: [u1] - Prazo: 15/10/21 	<p>Tarefa não é criada.</p> <p>Mensagem de erro informando que o campo título não pode ser vazio</p>

9. Estimativa de pontos de função

A planilha de Análise de Pontos de Função encontra-se disponível em:

<https://github.com/Ian021/pucminas>

10. Informações da implementação

O software não foi desenvolvido por conta de sua complexidade e da falta de capacitação técnica em algumas áreas da aplicação.

11. Referências

Foi utilizado como referência somente o material em vídeo, imagem e texto disponibilizado pela PUC Minas em seu portal.