**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Trabalho de Conclusão de Curso

Cronos GT (Gerenciador de Tarefas)

Vinícius Silva Avelino

Belo Horizonte

Fevereiro de 2022.

# Trabalho de Conclusão de Curso

**Sumário**

[Trabalho de Conclusão de Curso 3](#_Toc963)

[1. Cronograma de trabalho 4](#_Toc26614)

[2. Diagrama de casos de uso 5](#_Toc26071)

[3. Requisitos não-funcionais 5](#_Toc13629)

[4. Protótipo navegável do sistema 7](#_Toc21147)

[5. Diagrama de classes de domínio 8](#_Toc25532)

[6. Modelo de componentes 9](#_Toc4318)

[6.1. Padrão arquitetural 9](#_Toc21340)

[6.2. Diagrama de componentes 10](#_Toc11239)

[6.3. Descrição dos componentes 10](#_Toc2342)

[7. Diagrama de implantação 12](#_Toc10229)

[8. Plano de Testes 13](#_Toc9809)

[9. Estimativa de pontos de função 15](#_Toc24467)

## Cronograma de trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 13 / 12 / 21 | 14 / 12 / 21 | 1. Entendimento do escopo geral do sistema. | Definição inicial do projeto. |
| 14 / 12 / 21 | 15 /12 / 21 | 2. Compreensão e definição de caso de uso. | Concepção do Caso de Uso do sistema. |
| 20 / 12 / 21 | 21 / 12 / 21 | 3. Revisão e complemento do caso de uso. | Implementação de melhoria no caso de uso. |
| 21 /12 / 21 | 21 / 12 / 21 | 4. Definição dos requisitos não-funcionais. | Analise e definição dos requisitos não-funcionais. |
| 23 / 12 / 21 | 28 / 12 / 21 | 5. Concepção inicial do protótipo navegável. | Rascunho inicial do protótipo. |
| 29 / 12 / 21 | 10 / 01 / 22 | 6. Revisão do protótipo navegável e gravação do vídeo. | Finalização do protótipo navegável e gravação do vídeo. |
| 11 / 01 / 22 | 12 / 01 / 22 | 7. Concepção inicial da criação do Diagrama de Classes. | Base inicial do diagrama de classes do projeto. |
| 13 / 01 / 22 | 13 / 01 / 22 | 8. Revisão e finalização do Diagrama de Classes. | Finalização do diagrama de classes. |
| 13 / 01 / 22 | 14 / 01 / 22 | 9. Definição do padrão arquitetural da aplicação. | Definição inicial do padrão arquitetural para o projeto proposto. |
| 14 / 01 / 22 | 20/ 01 / 22 | 10. Definição e concepção inicial do diagrama de componentes. | Versão inicial do diagrama de componentes |
| 20 / 01 / 22 | 25 / 01 / 22 | 11. Refinamento do diagrama de Componentes. | Atualização no diagrama de componentes |
| 26 / 01 / 22 | 26 / 01 / 22 | 12. Concepção do diagrama de implantação. | Criação do Diagrama de implantação |
| 31 / 01 / 22 | 31 / 01 / 22 | 13. Definição dos casos de testes. | Definição dos casos de testes para implementação. |
| 31 / 01 / 22 | 01 / 02 / 22 | 14. Implantação da Analise de Pontos de Função | Definição da planilha de Analise de ponto de função. |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 15. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 16. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 17. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 18. |  |

## Diagrama de casos de uso

|  |
| --- |
|  |

## 

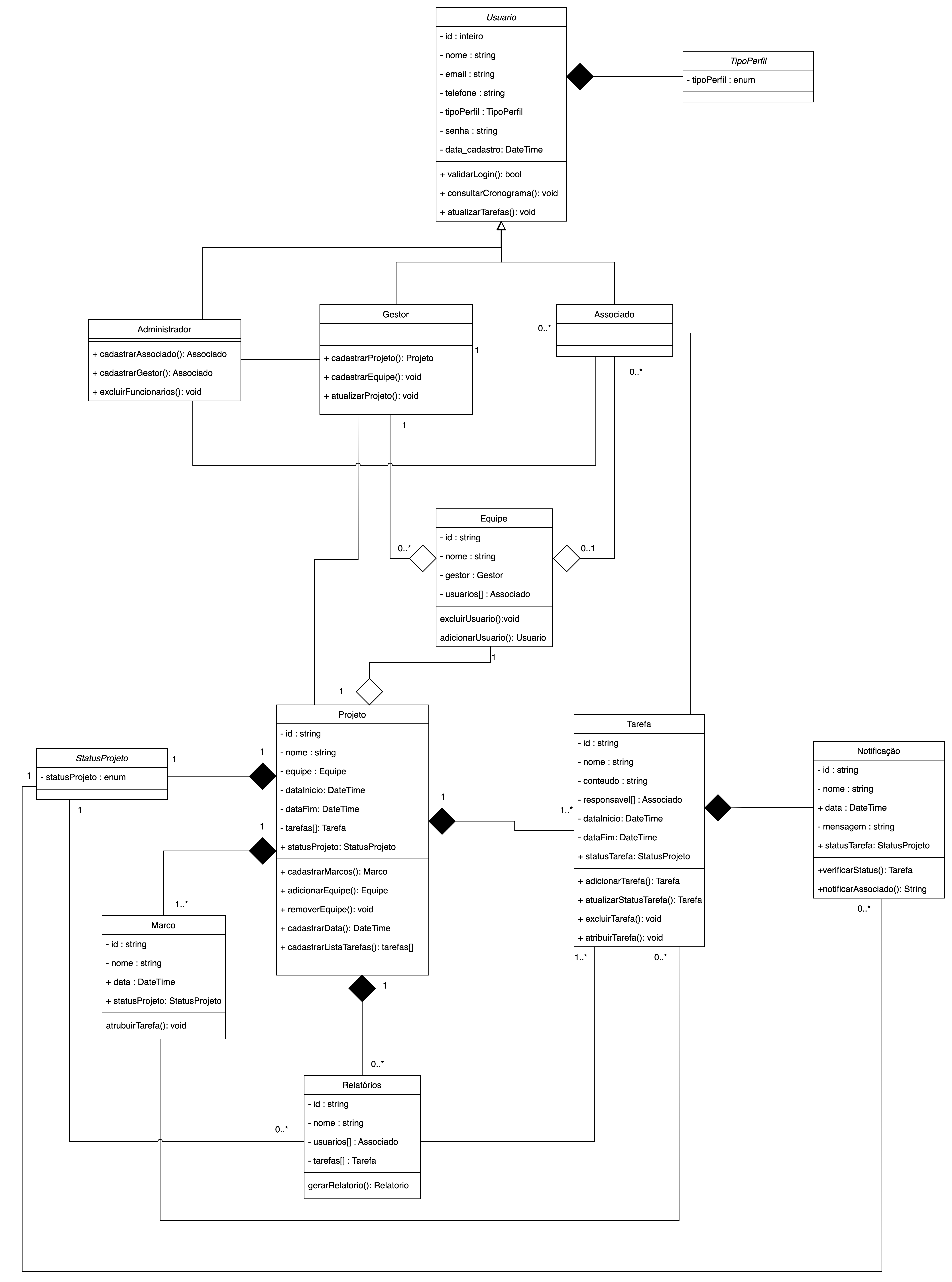
## Requisitos não-funcionais

1. O sistema deve poder ser acessado por meio da Web e em qualquer dispositivo móvel tais como tablets e smartphones;
2. O sistema deve ter um controle de segurança baseado em perfis de acesso;
3. Deve ser possível monitorar os principais marcos de entrega dos projetos, de forma individual e global;
4. A situação do projeto deve ser apresentada por meio de cores, de forma a facilitar a identificação de projetos que necessitam de maior atenção em um dado momento;
5. Deve ser possível visualizar o andamento das atividades por todos os envolvidos no projeto, com a utilização de cores para facilitar a identificação de tarefas atrasadas, com impedimentos ou associadas a maiores riscos;
6. Deve ser possível localizar facilmente as pessoas responsáveis pelo projeto.

## Protótipo navegável do sistema

[Link para vídeo do protótipo navegável.](https://github.com/Vinicius-Avelino/TCC-Engenharia-de-Software/tree/main/Video)

## Diagrama de classes de domínio



## Modelo de componentes

### Padrão arquitetural

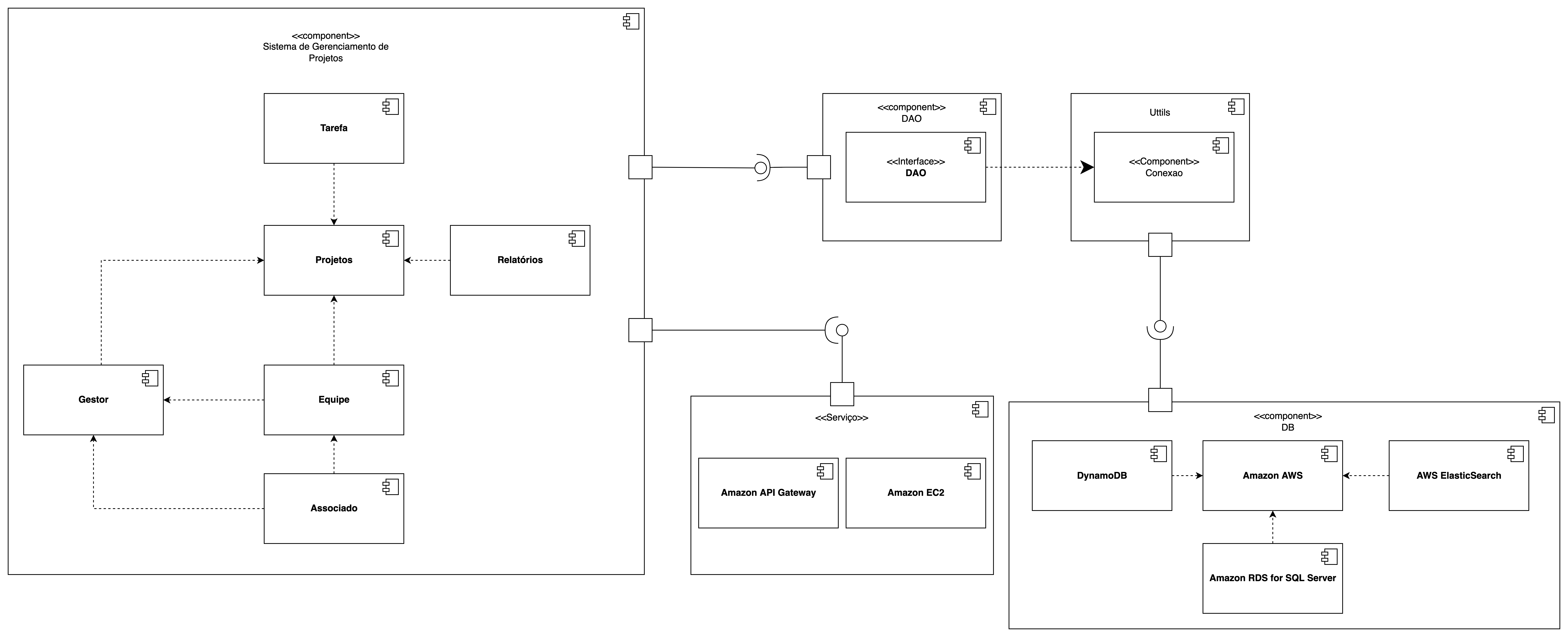
Será utilizado como padrão arquitetural o padrão de microsserviços visando o baixo acoplamento das funcionalidades desenvolvidas, possibilitando o foco na manutenibilidade do código fonte por se tratar de pequenos serviços isolados, sem causar muitos danos caso um deles venha a falhar.

Em conjunto serão utilizados os serviços da Amazon AWS com o ElasticSearch, garantindo a escalabilidade horizontal por meio de clusters sob a demanda da utilização da aplicação, a rapidez no armazenamento e busca de dados, garantindo análises rápidas e precisas e o Amazon API Gateway para criação, monitoração e proteção de APIs com o framework REST.

O sistema híbrido de banco de dados relacional e NoSQL serão utilizados para garantir o armazenamento de informações de forma estrutural em conjunto com a utilização de um banco chave valor para consultas eficazes.

As tecnologias utilizadas em conjunto para o desenvolvimento do projeto serão: Amazon AWS, Amazon API Gateway, SGDBR e NoSQL, ElasticSearch e Framework REST.

### Diagrama de componentes

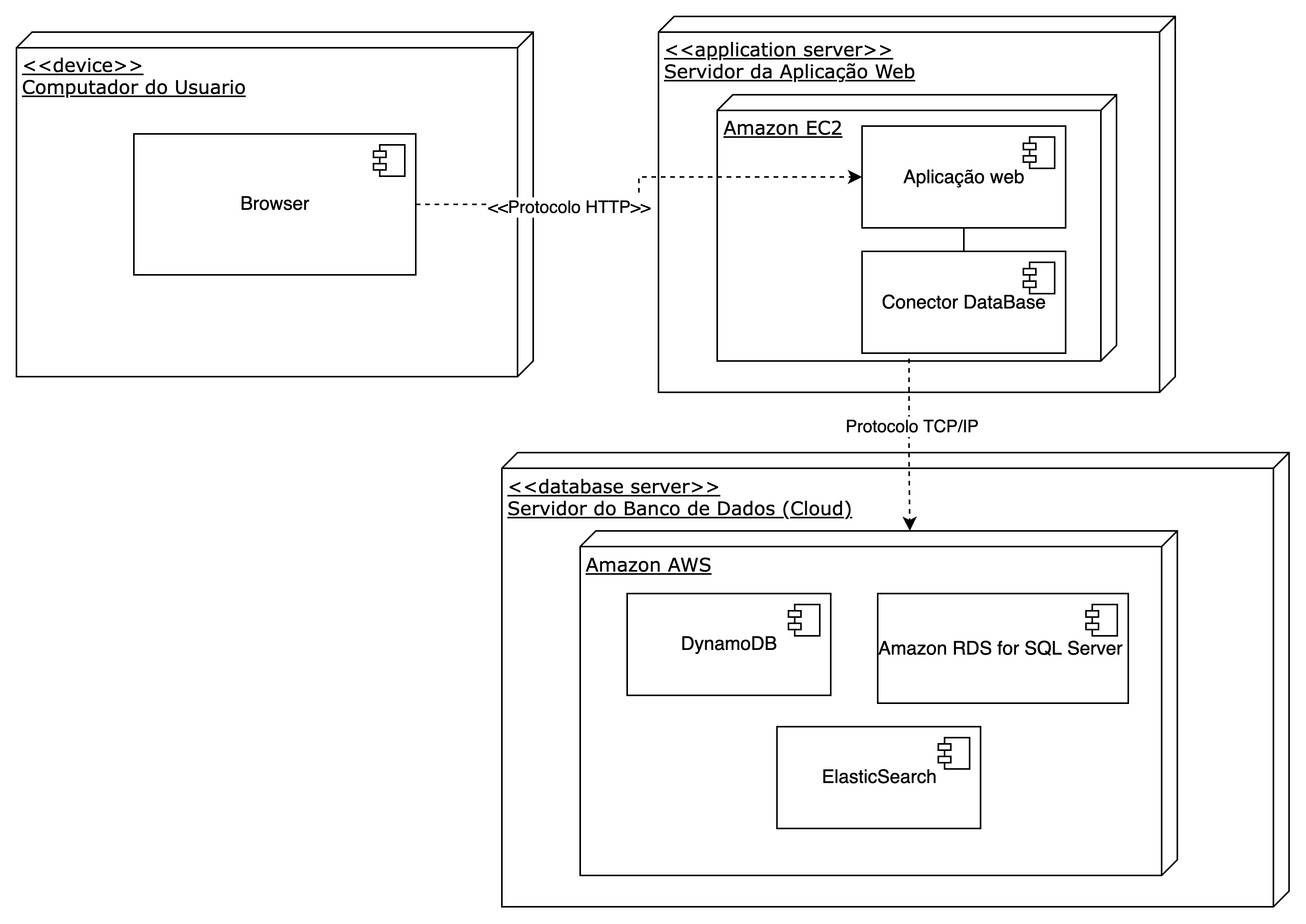


### Descrição dos componentes

|  |  |  |
| --- | --- | --- |
| **Número** | **Componente** | **Descrição** |
|  | Amazon API Gateway | Componente utilizado para a criação e gerenciamento de API REST no sistema de gerenciamento de projetos. |
|  | DynamoDB | Banco de dados do tipo Key Value utilizado para armazenamento de dados de associados, garantindo um login rápido e eficiente. |
|  | AWS ElasticSearch | Componente utilizado para o uso de relatórios no projeto e análises de negócio. |
|  | Amazon EC2 | Utilizado para a garantia de escalabilidade horizontal por meio de clusters sob demanda. |
|  | Amazon RDS for SQL Server | Componente utilizado para o armazenamento de dados estruturados no projeto. |
|  | DAO (Data Access Object) | Componente que gera persistência com o banco de dados. |
|  | Componente de conexão | Camada utilizada para realizar a conexão da aplicação com os bancos de dados. |

## 

## Diagrama de implantação



## Plano de Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  | Cadastrar projeto | Validação do cadastro do projeto no sistema. | * Entrar na aba de Projetos * Clicar em “+ Novo projeto” * Preencher o formulário com os dados corretos. * Clicar no botão verde no canto inferior direito. | * Quadro do projeto criado adicionado ao painel de projetos. |
|  | Priorizar tarefas | Validação da alteração de criticidade do projeto. | * Clicar no ícone de lápis no quadro “Lista de tarefas”. * Clicar em “Alterar criticidade”. * Clicar na bandeira de criticidade no pop-up “Criticidade”. | * Faixa de criticidade deve ser atribuída e exibida bi quadro da tarefa. |
|  | Atribuir tarefa ao membro | Validação de atribuição de tarefas a um usuário no sistema. | * Clicar no ícone de lápis no quadro “Lista de tarefas”. * Clicar em “Atribuir Membro” no pop-up “Propriedades da Tarefa”. * Selecionar o associado no pop-up “Quadro de membros”. | * O usuário será associado a tarefa selecionada. O ícone do associado será exibido no canto inferior direito da tarefa. |
|  | Gerar relatório | Validação da geração de relatórios no sistema. | * Clicar em “Dashboard” dentro do projeto. | * Os relatórios serão exibidos de acordo com os dados do projeto. |
|  |  |  |  |  |
|  |  |  |  |  |

## Estimativa de pontos de função

[Link para planilha de estimativa de pontos de função.](https://github.com/Vinicius-Avelino/TCC-Engenharia-de-Software)