



Documento de Arquitetura de Software

Histórico de versões

Data	Versão	Descrição	Autor	Revisor	Aprovado por
21/06/2023	1.0	Criação da primeira versão do documento de visão	@Débora Cavalcante	@Socorro Vania	@Socorro Vania

1. Introdução

O presente documento tem como objetivo descrever o documento de arquitetura do projeto DEC. Este sistema tem como finalidade a criação de um software para controle e gerenciamento de comissões internas e externas para a reitoria da UFOPA.

1.1 Finalidade

Este documento oferece uma visão arquitetural geral do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do mesmo. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas no decorrer do processo de desenvolvimento.

1.2 Escopo

Este documento auxilia os envolvidos no projeto a captar aspectos arquiteturais do sistema que são necessários para o desenvolvimento de uma solução que atenda às necessidades dos usuários finais. Além de auxiliar no entendimento do sistema por novos membros da equipe.

1.3 Definições, Acrônimos e Abreviações

- Repository Pattern: O Repository Pattern é um padrão de projeto bastante utilizado quando precisamos separar a camada de acesso aos dados da camada de regras de negócios de uma aplicação (conhecida também como camada de domínio).

1.4 Visão Geral

São apresentados ainda neste documento diferentes visões arquiteturais de como o sistema deve se comportar em diferentes processos, como deve ser implantado e implementado e restrições de desempenho e qualidade.

2. Representação Arquitetural

- Visão de caso de uso:** Apresenta as funcionalidades arquiteturais importantes e os usuários do sistema.
- Visão lógica:** Descreve as classes e sua organização e apresenta o padrão de arquitetura que deverá ser utilizado para o desenvolvimento do sistema.
- Visão de processos:** Mostra o padrão de comportamento do sistema diante de diferentes ações do usuário.
- Visão de implantação:** Descreve a estrutura do ambiente onde o software será instalado.
- Visão de implementação:** Ilustra a distribuição do processamento em um conjunto de nós do sistema, incluindo a distribuição física dos processos e threads.

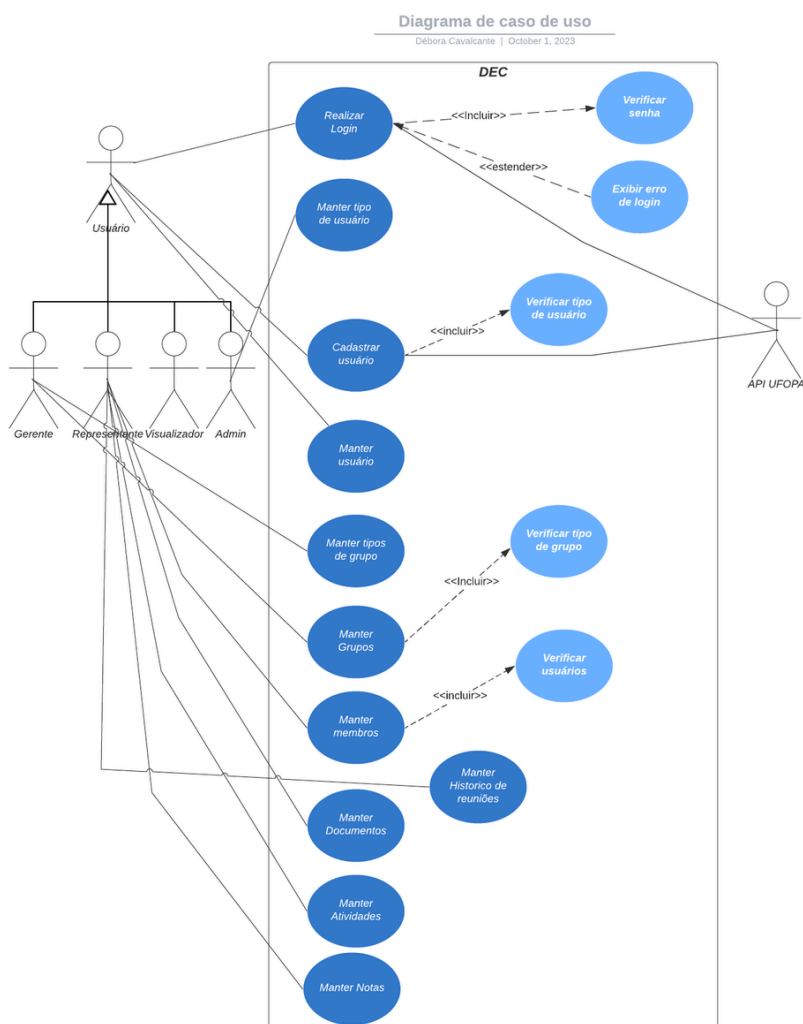
3. Metas e Restrições da Arquitetura

Existem algumas restrições de requisito e de sistema principais que têm uma relação significativa com a arquitetura, sendo elas:

- Utilização do paradigma Orientado a Objetos para o desenvolvimento;
- Repository Pattern;
- Linguagem Javascript;
- Linguagem de programação PHP;
- Biblioteca React e framework Laravel;
- O sistema em questão inicialmente funcionará apenas para web;
- Bancos de dados PostgreSQL.

4. Visão de Casos de Uso

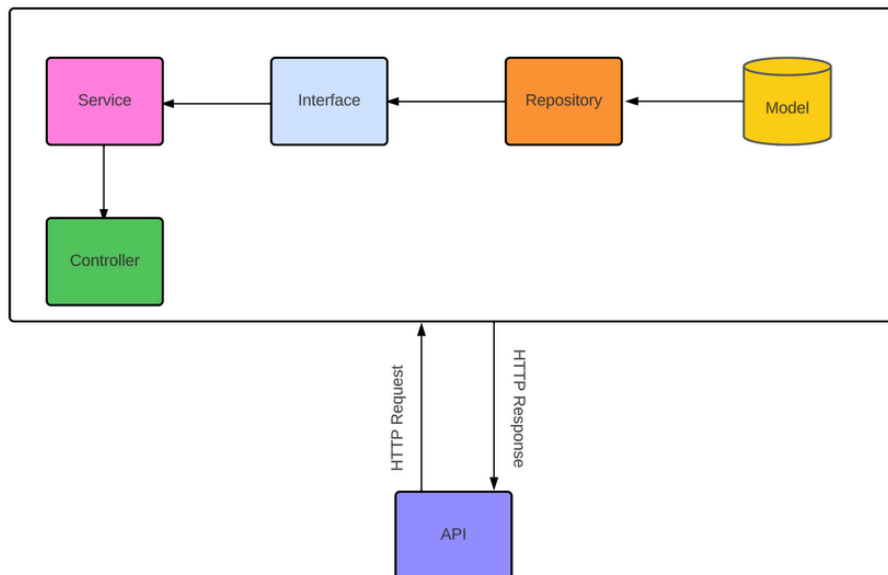
Os casos de uso do sistema DEC serão listados abaixo:



5. Visão Lógica [↗](#)

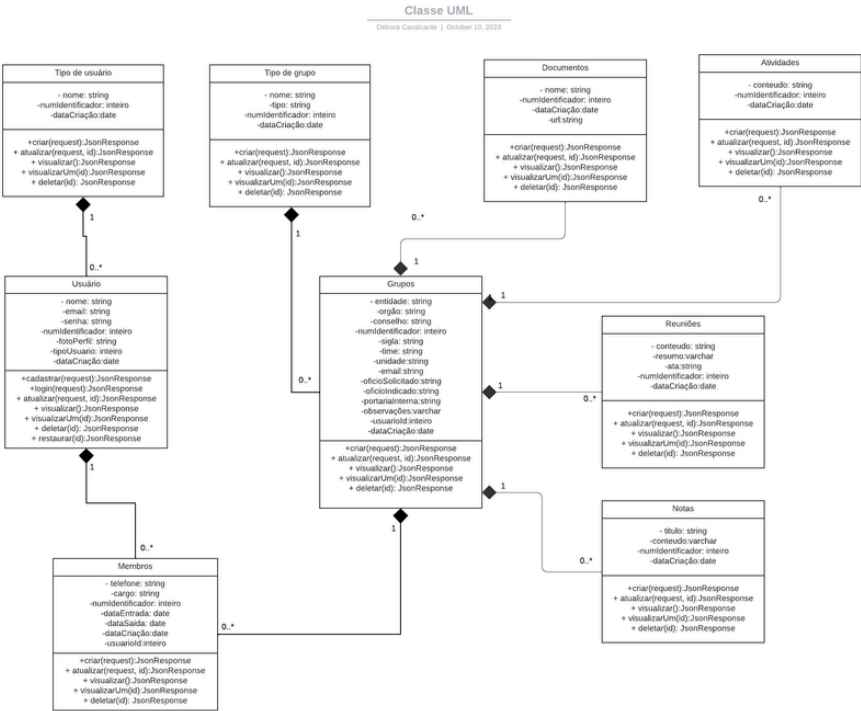
5.1 Visão Geral [↗](#)

A visão lógica define a estrutura da arquitetura. Abaixo será especificado o padrão utilizado para o desenvolvimento do sistema, no caso o Repository Pattern.



- **Model:** Representa a entidade principal com a qual estamos trabalhando.
- **Repository:** O repositório é responsável por interagir diretamente com o banco de dados para operações relacionadas ao modelo. Ele fornece uma camada de abstração sobre as operações de banco de dados.
- **Interface:** A Interface define os métodos que qualquer repositório da entidade deve implementar. Isso fornece um contrato para operações de leitura e escrita no banco de dados.
- **Service:** O serviço é responsável por conter a lógica de negócios. Ele utiliza o repositório para interagir com o banco de dados.
- **Controller:** O controlador é responsável por receber as solicitações do cliente, chamar os métodos apropriados do serviço e retornar as respostas.

5.2 Diagrama de classe



6. Visão de Implantação

O sistema será implantado quando for validado entre os stakeholders.

7. Visão da Implementação

O sistema será implementado no backend utilizando conceitos de Programação Orientada a Objetos através do framework Laravel, com Repository Pattern, Linguagem de Programação PHP e Bancos de Dados PostgreSQL, e no frontend será utilizado a biblioteca ReactJs, Linguagem de programação Javascript.

8. Tamanho e Desempenho

A aplicação deve rodar através de um site próprio na versão web funcionando apenas na rede da UFOPA.

9. Qualidade

O padrão de arquitetura adotado no projeto tem como finalidade garantir uma melhor organização do código-fonte, o que auxilia na manutenibilidade do software, bem como a portabilidade do mesmo.