PROJETO INOVAEDUCA

Requisitos da disciplina Modelagem de Software e Arquitetura de Sistemas

INTEGRANTES DO PROJETO e RA'S

Pedro Augusto da Silva Macedo	-	21010681
Luiz Eduardo Souza Rocha	-	24026664
Henrique Jorge Martins Figueiredo	-	24026809

São Paulo

2025

Sumário

1. Visão Geral do Sistema

O presente documento apresenta a estrutura de arquitetura do sistema desenvolvido para o Instituto Criativo. Trata-se de uma plataforma do tipo dashboard, com funcionalidades específicas destinadas a três perfis distintos de usuários: Professores, Colaboradores e Administradores. Cada perfil possui permissões e funcionalidades exclusivas, conforme segue:

- Professores: responsáveis pela publicação de postagens e eventos institucionais;
- Colaboradores: aptos a visualizar relatórios e inserir comentários pertinentes;
- Administradores: incumbidos de criar, editar e gerenciar relatórios e usuários.

2. Camadas da Arquitetura

A arquitetura do sistema é segmentada em múltiplas camadas, organizadas conforme suas responsabilidades e funcionalidades técnicas:

a) Camada de Apresentação (Frontend):

Responsável pela interface gráfica do usuário. Utiliza tecnologias como React, Vue.js ou Angular, com suporte de bibliotecas modernas como Tailwind CSS e Axios. Esta camada oferece formulários, login, painéis de visualização e controle de acesso conforme o papel do usuário.

b) Camada de Lógica de Negócios (Backend):

Executa as regras de negócio e manipulação dos dados. Pode ser construída com Node.js (Express), Python (Django/Flask), Ruby on Rails ou Java (Spring Boot). Gerencia autenticação, autorização, postagens, comentários e relatórios.

c) Camada de Persistência (Banco de Dados):

Armazena de forma segura todas as informações pertinentes ao sistema, como dados de usuários, eventos, relatórios e comentários. As tecnologias recomendadas são MySQL, PostgreSQL ou MongoDB.

d) Camada de APIs (Interface de Comunicação):

Responsável pela comunicação entre frontend e backend. Utiliza RESTful APIs ou GraphQL para expor endpoints que permitam operações CRUD seguras e eficientes.

e) Camada de Segurança:

Implementa mecanismos de autenticação (JWT, OAuth2), criptografia de senhas (bcrypt) e proteção contra vulnerabilidades como SQL Injection, XSS e CSRF.

3. Interação entre as Camadas

A interação entre as camadas ocorre por meio de requisições HTTP. O frontend consome dados do backend via APIs, que por sua vez processa as solicitações, aplica regras de negócio e interage com o banco de dados, garantindo a persistência e integridade das informações.

4. Considerações sobre Escalabilidade e Modularidade

A aplicação foi concebida com foco na escalabilidade e modularidade. A escalabilidade poderá ser implementada tanto de forma vertical (aumento dos recursos de um único servidor) quanto horizontal (distribuição do sistema em múltiplos servidores). A modularidade será promovida pela separação de funcionalidades em módulos independentes, como:

- Módulo de Autenticação;
- Módulo de Publicação de Eventos;
- Módulo de Comentários e Relatórios;
- Módulo de Administração e Controle de Permissões.

5. Fluxo de Acesso e Funcionalidades dos Usuários

O fluxo de uso do sistema segue a seguinte estrutura:

1. O usuário acessa a tela de login e informa suas credenciais;
2. O sistema valida os dados e identifica o perfil (Professor, Colaborador ou Administrador);
3. Professores podem criar postagens e visualizar conteúdos;
4. Colaboradores acessam os relatórios e inserem comentários;
5. Administradores têm acesso completo para gerenciamento de relatórios, usuários e permissões.

6. Tecnologias Recomendadas

Para o adequado funcionamento e manutenção do sistema, recomenda-se a adoção das seguintes tecnologias:

- Frontend: React.js, Tailwind CSS, Axios;
- Backend: Node.js com Express, ou Python com Django/Flask;
- Banco de Dados: PostgreSQL (relacional) ou MongoDB (NoSQL);
- Autenticação e Segurança: JWT, OAuth2, bcrypt e HTTPS;
- APIs: RESTful ou GraphQL, conforme a complexidade da integração entre camadas.