

Projeto DoaTech

Requisitos da disciplina Modelagem de Software e Arquitetura de Sistemas

INTEGRANTES DO PROJETO e RA'S

Kauã Daniel – 24026846

Caio Dantas – 24026710

Caio Gomes – 24026876

Saulo Santos – 24026911

São Paulo

2025

Sumário

1. Panorama da Solução Tecnológica	3
2. Estrutura em Camadas da Arquitetura	3
3. Comunicação entre Componentes	4
4. Escalabilidade e Organização Modular	4
5. Caminho de Navegação dos Usuários.....	5
6. Conjunto Tecnológico Recomendado	6
7. Casos de uso	7

1. Panorama da Solução Tecnológica

Este documento tem como propósito detalhar a composição da arquitetura do sistema criado para atender às demandas do Instituto Criativo. A plataforma, concebida em formato de painel interativo (dashboard), foi desenvolvida para abranger três perfis distintos de usuários sendo eles: Professores, Colaboradores e Administradores, cada um com atribuições e acessos específicos dentro do ambiente digital:

Professores: responsáveis por divulgar conteúdos institucionais, incluindo eventos e publicações;

Colaboradores: têm permissão para acessar relatórios e registrar comentários relevantes;

Administradores: encarregados da administração completa do sistema, com poder para criar, modificar e gerir relatórios e perfis de usuários.

2. Estrutura em Camadas da Arquitetura

A organização da arquitetura segue uma divisão lógica por responsabilidades técnicas, promovendo maior clareza e manutenção do sistema. As principais camadas são:

a) Interface do Usuário (Frontend)

Essa camada é a responsável por fornecer uma experiência visual e interativa ao usuário final. Desenvolvida com frameworks modernos como React, Vue.js ou Angular, e com auxílio de bibliotecas como Tailwind CSS e Axios, ela contempla formulários, autenticação, navegação entre telas, dashboards e controle de permissões conforme o perfil do usuário logado.

b) Camada de Processamento e Regras (Backend)

É nesta parte que acontecem as operações principais do sistema. Nela são aplicadas as lógicas de negócio, como autenticação, manipulação de dados, controle de permissões e gerenciamento de conteúdos (postagens, relatórios, comentários). Pode ser construída com tecnologias como Node.js (Express), Python (Django ou Flask), Ruby on Rails ou Java com Spring Boot.

c) Armazenamento de Dados (Banco de Dados)

Todos os dados essenciais — desde informações de login até registros de eventos e interações — são armazenados de forma segura nesta camada. Os sistemas recomendados incluem MySQL, PostgreSQL (ambos relacionais) ou MongoDB (orientado a documentos).

d) Integração e Comunicação (APIs)

Esta camada intermedia a comunicação entre a interface do usuário e o servidor. Utiliza RESTful APIs ou GraphQL para criar endpoints que permitam executar operações de leitura, criação, atualização e exclusão de dados (CRUD) de forma eficiente e segura.

e) Segurança e Proteção de Dados

Para garantir a integridade do sistema, são implementados mecanismos como **JWT** e **OAuth2** para autenticação, além de criptografia de senhas com **bcrypt**. Também são aplicadas proteções contra ameaças comuns, como **SQL Injection**, **XSS (Cross-site Scripting)** e **CSRF (Cross-site Request Forgery)**.

3. Comunicação entre Componentes

As diferentes camadas do sistema interagem por meio de requisições HTTP. A interface de usuário (frontend) envia solicitações para a camada de lógica (backend), que interpreta os pedidos, aplica as regras necessárias e se comunica com o banco de dados para ler ou gravar as informações solicitadas. A resposta é então enviada de volta ao frontend, completando o ciclo de interação.

4. Escalabilidade e Organização Modular

A plataforma foi desenvolvida com foco em crescimento progressivo e facilidade de manutenção. A escalabilidade pode ser realizada de duas formas:

Vertical, com o reforço da capacidade de um único servidor;

Horizontal, com a distribuição da aplicação em vários servidores ou serviços.

A modularidade, por sua vez, é garantida pela separação clara entre diferentes funcionalidades em módulos distintos, como:

Autenticação e controle de sessão;

Gerenciamento e publicação de eventos;

Moderação de comentários e análise de relatórios;

Administração geral e controle de acessos e permissões.

5. Caminho de Navegação dos Usuários

O uso do sistema segue um fluxo bem definido, que respeita as permissões de cada perfil de usuário:

1. O usuário acessa a página inicial e realiza a autenticação com login e senha;
2. O sistema valida as credenciais e identifica automaticamente o tipo de perfil (Professor, Colaborador ou Administrador);
3. Professores têm a liberdade de publicar conteúdos e acompanhar informações institucionais;
4. Colaboradores acessam relatórios disponíveis e podem registrar observações e comentários;
5. Administradores dispõem de funcionalidades completas para gerenciar relatórios, controlar os usuários cadastrados e definir os níveis de acesso.

6. Conjunto Tecnológico Recomendado

Para garantir o desempenho, segurança e manutenção eficiente do sistema, é indicada a utilização 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)

Segurança: JWT, OAuth2, bcrypt, conexão via HTTPS

APIs: RESTful ou GraphQL, conforme o nível de complexidade da comunicação entre componentes

7.Casos de uso





