Documentação Técnica do Projeto

Nome do projeto: **TechChallengeFase3**

Integrantes (Grupo 28)

Filipe Mulitemo – RM353810

Lucas de Moraes – RM353135

Marcelo Neves – RM355808

Song Shik – RM353956

Thiago Santos - RM355404

**Introdução**

Este projeto é parte do Tech Challenge Fase 3, desenvolvido pelo grupo 28.

**Objeto**

Disponibilizar uma plataforma onde professores possam realizar postagem de suas aulas e elas serem consumidas pelos seus alunos.

**Passos Iniciais**

1.Realizar o clone do repositório [*https://github.com/techchallengegroup28/TechChallengeFase3.git*](https://github.com/techchallengegroup28/TechChallengeFase3.git)

2.Abrir o projeto com sua IDE de desenvolvimento, navegar até o diretório TechChallengeFase3

3.No terminal de comando da sua IDE executar o comando ***npm install*** para baixar as dependências do projeto.

4.No terminal executar o comando ***docker compose up -d***este comando é responsável por iniciar os containers da aplicação web.

5.Abrir o navegado se sua escolha e acessar o endereço [http://localhost:3001](http://localhost:3001/api-docs/) será exibida a página inicial da aplicação contendo os posts.

6.Acessando a rota <http://localhost:3001/login> o usuário irá para a tela de login para realizar a autenticação do professor.

**Arquitetura do Projeto**

Desenvolvido na arquitetura node.js, React e NextJS, pois oferece várias vantagens, especialmente para aplicações que exigem alta escalabilidade e desempenho.

Abaixo vou descrever as principais bibliotecas utilizadas na construção da aplicação.

1. Framework principal
   * Next e React: Utilizado para criar a aplicação.
   * Node.js: Responsável por executar o código do lado do servidor.
2. Validação e gerenciamento de estados
   * Formik: Utilizado para gerenciamento de estados dos formulários.
   * Yup: Utilizado para validação dos dados informados nos formulários da aplicação.
3. Segurança
   * Js-Cookie: Utilizado para garantir a autenticação através de cookies.

A arquitetura do projeto utilizou o padrão gerado pelo Next.js, onde o sistema de roteamento é baseado em arquivos.

**Estrutura do Arquivos e Diretórios**

TechChallengeFase3/

├── Dockerfile # Arquivo de configuração do Docker

├── README.md # Documentação do projeto

├── .env # Variáveis de ambiente

├── .env.example # Variáveis de ambiente

├── .gitignore # Arquivos e diretórios a serem ignorados pelo Git

├── docker-compose.yml

├── next-env.d.ts

├── next.config.mjs

├── package-lock.json

├── package.json # Dependências e scripts do projeto

├── src

│   ├── app

│   │   ├── admin

│   │   │   ├── page.tsx # Página inicial do professor contendo todos os posts (administrador)

│   │   │   ├── post-criar

│   │   │   │   └── page.tsx # Página para o professor criar um post

│   │   │   └── post-editar

│   │   │   └── [id]

│   │   │   └── page.tsx # Página para o professor editar um post

│   │   ├── layout.tsx # Layout da aplicação

│   │   ├── login

│   │   │   └── page.tsx # Página de login para o professor (administrador)

│   │   ├── page.tsx # Página inicial da aplicação contendo os posts

│   │   └── post-detalhes

│   │   └── [id]

│   │   └── page.tsx # Página para ver um post específico

│   ├── components

│   │   ├── ListPostAdm.tsx # Componente para exibir posts na visão de professor (administrador)

│   │   ├── ListPostUser.tsx # Componente para exibir os posts que um usuário pode ver

│   │   ├── PostCard.tsx # Componente para renderizar os cards dos posts na página inicial

│   │   ├── bar.tsx # Componente para renderizar a barra de busca por post

│   │   ├── header.tsx # Componente para renderizar o header da aplicação

│   │   ├── longText.tsx # Lidar com a exibição de textos grandes (por exemplo: descrição)

│   │   ├── postDetails.tsx # Componente para lidar com os detalhes de um post

│   │   └── postForm.tsx # Componente para lidar com o formulário de um post

│   ├── interface

│   │   └── IPost.tsx # Interface de um post

│   ├── middleware.ts # Middleware para verificar se o usuário está autenticado

│   ├── services

│   │   └── posts.tsx # Service para fazer requisições para o back-end

│   ├── styles # Diretório com os arquivos CSS da aplicação e seus componentes

│   │   ├── bootstrap.css

│   │   ├── globals.css

│   │   └── modules

│   │   ├── barUser.module.css

│   │   ├── header.module.css

│   │   ├── helloWord.module.css

│   │   ├── listPostAdm.module.css

│   │   ├── listPostUser.module.css

│   │   ├── login.module.css

│   │   ├── longText.module.css

│   │   ├── pageDashAdmin.module.css

│   │   ├── postDetails.module.css

│   │   └── postForm.module.css

│   └── utils

│   └── appUtils.tsx # Funcionalidades utilizadas por várias partes da aplicação

└── tsconfig.json

**Páginas da aplicação**

* Página inicial
  + **Rota:** /
  + **Descrição:** Esta página mostra a listagem de posts existentes na aplicação para usuários não autenticados.

A screenshot of a computer

Description automatically generated

* Barra de busca na página inicial
  + **Rota:** /
  + **Descrição:** Esta funcionalidade busca algum termo digitado para encontrar um post específico.

A screenshot of a computer

Description automatically generated

* Página de detalhes de um post
  + **Rota:** /post-detalhes/{id}
  + **Descrição:** Esta página exibe as informações de um post específico.

A screenshot of a computer

Description automatically generated

* Página de login
  + **Rota:** /login
  + **Descrição:** Esta página exibe o formulário para o professor (administrador) realizar o login na aplicação.

A screenshot of a computer

Description automatically generated

* Página de admin
  + **Rota:** /admin
  + **Descrição:** Esta página exibe todos os posts da aplicação na visão de professor (administrador), contendo o botão de editar, excluir e criar um post.

A screenshot of a computer screen

Description automatically generated

* Página de criação de post
  + **Rota:** /admin/post-criar
  + **Descrição:** Esta é a página da aplicação onde o professor poderá criar um post.

A screenshot of a computer

Description automatically generated

* Página de edição de post
  + **Rota:** /admin/post-editar/{id}
  + **Descrição:** Esta é a página da aplicação onde o professor poderá editar um post.

A screenshot of a computer

Description automatically generated

**Processo de desenvolvimento do trabalho**

O desenvolvimento do projeto foi estruturado em diversas tarefas, que foram distribuídas entre os membros da equipe. Também estabelecemos prazos para a entrega e integração dos resultados.

Ao longo do desenvolvimento, realizamos reuniões periódicas para monitorar o progresso do projeto, o que nos permitiu compartilhar conhecimentos e experiências. Isso foi essencial para aplicarmos na prática o que aprendemos nas aulas sobre front-end com React, integrando com a aplicação feita na fase 2.

Um dos maiores desafios enfrentados pela equipe foi a responsividade e estilização da aplicação. Além disso, a adaptação e a integração com o back-end da aplicação, utilizando a arquitetura RESTful.

Outro desafio enfrentado pela equipe foi a manipulação das imagens.

Em resumo, a parte mais simples foi criar a estrutura do projeto, utilizando o NextJS que nos traz muitas facilidades e produtividade na elaboração do projeto.