

TECH CHALLENGE 3

Documentação Técnica

Johnny da Silva Lima
RM: 361533

Este relatório compõe a entrega da documentação do desafio de programação - Tech Challenge 3.

São Paulo, 7 de outubro de 2025

Sumário

Lista de Figuras	2
1 Objetivo	3
2 Estrutura do Projeto	3
2.1 API	3
2.1.1 Models	3
2.1.2 Index.ts	3
2.1.3 Uploads	4
2.2 Client	5
2.2.1 Estrutura de Arquivos	5
3 Desafios encontrados	6
Referências	7

Lista de Figuras

1 Objetivo

Desenvolver uma interface gráfica para a aplicação de blogging utilizando React. A aplicação deve ser responsiva, acessível e fácil de usar, permitindo aos docentes e alunos(as) interagir com os diversos endpoints REST já implementados no back-end.

2 Estrutura do Projeto

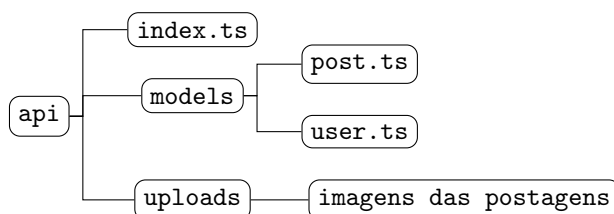
Para a elaboração do projeto, a estrutura foi dividida em duas partes:

- API: contendo o backend do projeto;
- CLIENT: contendo o frontend do projeto;

2.1 API

A seção de API utiliza um banco de dados em mongo db, através do mongoose , com chamadas REST para o endpoint localizado na porta 4000.

A seção de API contém a seguinte estrutura:



Nas próximas seções serão abordadas as funcionalidades de cada código.

2.1.1 Models

Post.ts: O código Post.ts é responsável por criar o esquema da entidade Post, que contém a seguinte estrutura:

- author: usuário que criou a postagem;
- title: título da postagem;
- summary: sumário da postagem;
- content: conteúdo principal da postagem;
- file: caminho do arquivo da imagem de capa;
- timestamps: data de criação de uma postagem;

User.ts: O código User.ts é responsável por criar o esquema da entidade User, que contém a seguinte estrutura:

- user: nome de usuário;
- password: senha do usuário;
- role: tipo de usuário, podendo ser 'user' (alunos) ou 'admin' (professor);

2.1.2 Index.ts

O arquivo **Index.ts** é o principal ponto de entrada do backend da aplicação. Ele configura os middlewares, realiza a conexão com o banco de dados MongoDB e define as rotas REST que compõem a API do sistema.

Importações e Configuração Inicial

O código começa com a importação das bibliotecas essenciais:

- `express` – Framework para criação do servidor HTTP.
- `cors` – Permite requisições do frontend hospedado em outra origem.
- `mongoose` – ODM para comunicação com o MongoDB.
- `bcryptjs` – Utilizado para criptografar senhas.
- `jsonwebtoken` – Gera e valida tokens JWT para autenticação.
- `cookie-parser` – Permite leitura e manipulação de cookies.
- `multer` – Middleware para upload de arquivos.
- `fs` – Módulo nativo para manipulação de arquivos.

Middlewares

São configurados os seguintes middlewares globais:

- `cors` com credenciais e origem definida.
- `express.json()` para interpretar o corpo das requisições em JSON.
- `cookieParser()` para leitura de cookies.
- `express.static()` para servir arquivos da pasta `uploads`.

Conexão com o Banco de Dados

A conexão com o MongoDB é realizada via `mongoose.connect()`, utilizando uma string de conexão com autenticação e cluster Atlas.

Rotas REST

O arquivo define as seguintes rotas:

- `POST /register` – Criação de novo usuário com senha criptografada.
- `POST /login` – Autenticação do usuário e geração de token JWT.
- `GET /profile` – Retorna os dados do usuário autenticado.
- `POST /logout` – Remove o token do cookie.
- `POST /post` – Criação de nova postagem com upload de imagem.
- `GET /posts` – Lista as últimas 20 postagens.
- `GET /posts/:id` – Retorna os dados de uma postagem específica.
- `PUT /posts/:id` – Atualiza uma postagem existente.
- `DELETE /posts/:id` – Remove uma postagem do banco de dados.

Execução

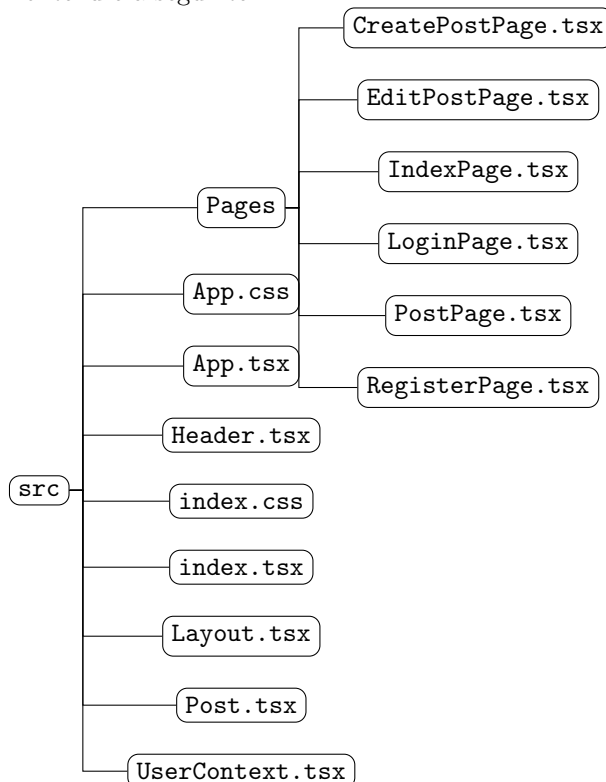
O servidor é iniciado na porta 4000 com o comando `'node index.ts'`

2.1.3 Uploads

A pasta `uploads` contém todos os arquivos de imagem que são enviados através das chamadas REST, os guardando e enviando conforme necessário.

2.2 Client

A seção de client utiliza react com react-router-dom para fazer o roteamento das rotas. A estrutura do frontend é a seguinte:



Nas próximas seções serão abordadas as funcionalidades de cada código.

2.2.1 Estrutura de Arquivos

- **App.css**
Define estilos globais da aplicação. Contém regras de layout, cores e espaçamentos aplicados ao componente principal **App.tsx**.
- **App.tsx**
Componente raiz da aplicação. Gerencia as rotas e renderiza o layout principal, incluindo cabeçalho e páginas.
- **Header.tsx**
Componente de cabeçalho fixo que aparece em todas as páginas. Contém links de navegação e lógica para exibir opções com base no estado de autenticação do usuário.
- **index.css**
Arquivo de estilos base que aplica configurações globais como reset de CSS e fontes padrão.
- **index.tsx**
Ponto de entrada da aplicação React. Renderiza o componente **App** dentro da **div** com **id="root"**.
- **Layout.tsx**
Componente que define a estrutura comum das páginas, incluindo o **Header** e o conteúdo principal. Facilita a reutilização de layout.
- **Post.tsx**
Componente que exibe uma postagem individual. Recebe dados via props e renderiza título, conteúdo e autor.
- **UserContext.tsx**
Implementa o contexto de usuário usando React Context API. Permite compartilhar informações de autenticação e dados do usuário entre componentes.

- **Pages/**

Pasta que contém as páginas principais da aplicação, cada uma representando uma rota.

- **CreatePostPage.tsx**

- Página para criação de novas postagens. Contém formulário com campos de título, conteúdo e imagem.

- **EditPostPage.tsx**

- Página para edição de postagens existentes. Carrega os dados da postagem e permite alterações.

- **IndexPage.tsx**

- Página inicial da aplicação. Lista todas as postagens disponíveis com links para leitura.

- **LoginPage.tsx**

- Página de login. Permite autenticação do usuário via formulário de e-mail e senha.

- **PostPage.tsx**

- Página de leitura de uma postagem específica. Exibe detalhes completos da postagem selecionada.

- **RegisterPage.tsx**

- Página de registro de novos usuários. Contém formulário para criação de conta.

3 Desafios encontrados

Os desafios encontrados durante a elaboração do tech challenge 3 foram, principalmente, em relação ao não funcionamento do tech challenge 2, que deveria ser a base para este trabalho. Com isso, o backend teve de ser refeito de maneira mais simples, o que atrasou o desenvolvimento do frontend.

Também foram encontrados dificuldades em configurar o dockerfile, o que levou a uma execução manual do projeto.

Referências

- [1] WatEletronics. What are the differences between microprocessor and microcontroller? <https://www.watelectronics.com/differences-between-microprocessor-and-microcontroller/>, Fevereiro 2020. Acessado em 15 de Novembro de 2022.
- [2] WatEletronics. Arduino technology architecture and its advantages. <https://www.watelectronics.com/arduino-technology-architecture-and-applications/>, Janeiro 2021. Acessado em 15 de Novembro de 2022.
- [3] Arduino. <https://www.arduino.cc/>.
- [4] Arduino Documentation. <https://docs.arduino.cc/>.
- [5] Arduino Stack Exchange. Is arduino a single board computer? <https://arduino.stackexchange.com/questions/58025/is-arduino-a-single-board-computer>, Novembro 2018. Acessado em 15 de Novembro de 2022.
- [6] Arduino, José Bagur, and Taddy Chung. Arduino memory guide. <https://docs.arduino.cc/learn/programming/memory-guide>, Novembro 2022. Acessado em 15 de Novembro de 2022.
- [7] Leo Louis. Working principle of arduino and using it as a tool for study and research. volume 1, 07 2018.
- [8] Teach Computer Science. Von-neumann vs harvard architecture. <https://teachcomputerscience.com/von-neumann-harvard-architecture/>. Acessado em 18 de Novembro de 2022.
- [9] Arduino. <https://en.wikipedia.org/wiki/Arduino>. Acessado em 18 de Dezembro de 2022.
- [10] Arduinome. <https://en.wikipedia.org/wiki/Arduinome>. Acessado em 18 de Dezembro de 2022.
- [11] ArduSat. <https://en.wikipedia.org/wiki/ArduSat>. Acessado em 18 de Dezembro de 2022.
- [12] Arduboy. <https://en.wikipedia.org/wiki/Arduboy>. Acessado em 18 de Dezembro de 2022.
- [13] ArduPilot. <https://en.wikipedia.org/wiki/ArduPilot>. Acessado em 18 de Dezembro de 2022.
- [14] ArduPilot Imagem. <https://www.americanas.com.br/produto/5430152717>. Acessado em 18 de Dezembro de 2022.
- [15] Arduinome Imagem. <https://www.flickr.com/photos/caseorganic/4506313870>. Acessado em 18 de Dezembro de 2022.
- [16] OBduino. <https://en.wikipedia.org/wiki/OBduino>. Acessado em 18 de Dezembro de 2022.
- [17] OBduino_ Imagem. <http://obduino.ca/>. Acessado em 18 de Dezembro de 2022.
- [18] Tutorials Point. Arduino - interrupts. https://www.tutorialspoint.com/arduino/arduino_interrupts.htm. Acessado em 19 de Dezembro de 2022.