

Nome: Eduardo Katayama

Nome: Wilson M. U. Hiramatsu

Relatório Técnico

Requisitos

- Docker/Docker Compose

Instruções

- Baixar o projeto no GitHub (LINK ABAIXO)
- Acessar a pasta do projeto pelo terminal
- Rodar o comando do Docker > docker-compose up
- Acessar a <http://localhost:89>

Link github

<https://github.com/w4nh7u/tech-challenge-fase5>

- Este repositório contém todo o código-fonte dos serviços desenvolvidos.

Documentação das APIs

Endpoints para gerenciamento dos Locais

-- Auth

[POST] <http://localhost:3000/api/auth/new>

[POST] <http://localhost:3000/api/auth/login>

[GET] <http://localhost:3000/api/auth/logout>

-- Membros

[GET] <http://localhost:3000/api/members>

[POST] <http://localhost:3000/api/members/new>

[GET] <http://localhost:3000/api/members/member/:id>

[PUT] <http://localhost:3000/api/members/update/:id>

[DELETE] <http://localhost:3000/api/members/delete/:id>

-- Tasks para membros

[GET] <http://localhost:3000/api/member-task>

<http://localhost:3000/api/member-task/new>

[PUT] <http://localhost:3000/api/member-task/update/:id>

-- Tasks

[GET] <http://localhost:3000/api/tasks>

[POST] <http://localhost:3000/api/tasks/new>

[GET] <http://localhost:3000/api/tasks/task/:id>

[PUT] <http://localhost:3000/api/tasks/update/:id>

[DELETE] <http://localhost:3000/api/tasks/delete/:id>

Tecnologias e Ferramentas Utilizadas

- HTML: Utilizado para criar a estrutura básica da interface de usuário.
- JavaScript: Utilizado para a lógica de verificação de sobreposição de horários.
- CSS Utilizado para estilizar a interface de usuário, garantindo uma apresentação visual atraente e responsiva.
- Node.js: Utilizado para criar o backend e disponibilizar a API.
- Express.js: Framework para Node.js que simplifica o desenvolvimento de APIs REST.
- Git: Controle de versão utilizado para gerenciar o código-fonte.
- Firebase: Banco de dados relacional utilizado para armazenar informações de forma estruturada e eficiente.
- React: Biblioteca JavaScript utilizada para construir interfaces de usuário interativas e dinâmicas.
- Material UI: Biblioteca de componentes de interface de usuário para React, que oferece uma ampla gama de elementos estilizados e prontos para uso, facilitando o desenvolvimento de uma interface moderna e responsiva.
- Docker: Plataforma de containers utilizada para criar, implantar e executar aplicações de forma isolada e consistente, garantindo que funcionem da mesma maneira em diferentes ambientes.
- Dotenv: Utilizado para criar variáveis de ambiente.

Desafios Encontrados e Soluções Implementadas

Integração com firebase.

Solução: Versão do node.

Falta de documentação do Firebase para utilizar as versões corretas do Node.js.

Solução: Atualizar a versão do node da versão 12 para 17.3.