# Teste para desenvolvedor de sistemas para Internet

# Introdução

Este teste é dividido em duas partes: A primeira é destinada a desenvolvedores de frontend conhecimento em ReactJS. A segunda, destinada a desenvolvedores de backend com Node JS e NestJS.

Você deve resolver as questões de acordo com a vaga em que estiver participando do processo seletivo, ou seja:

Desenvolvedor frontend: Questão 1

Desenvolvedor backend: Questão 2

Desenvolvedor fullstack: Ambas as questões

## Recomandações

Crie a sua solução em novo repositório público do github e envie o link para marcos.aquino@easytech.com.br

Para as questões de frontend utilize o backend disponível em:

https://github.com/marcosdxt/api-teste-analista

# Questão 1

Elaborar uma aplicação React para CRUD de usuários com nome, telefone e e-mail utilizando a API disponibilizada pela Easytech como repositório dos dados

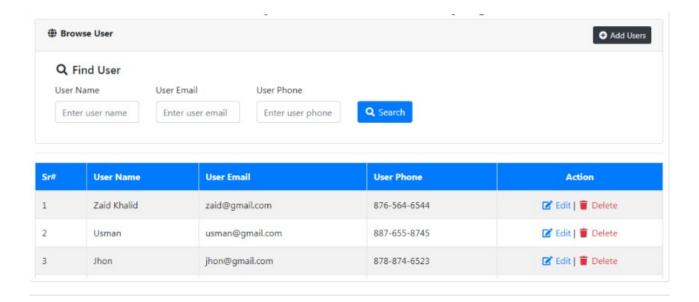
A aplicação deverá atender aos seguintes requisitos:

- Consumir a API REST disponibilizada pela Easytech;
- Permitir ao usuário cadastrar, editar, listar e remover os usuários;
- Permitir ao usuário visualizar os dados de forma tabular;
- Realizar busca de um usuário específico nos dados recuperados da API;
- Apresentar toast de confirmação a cada operação de inserção, atualização ou remoção executadas;
- Apresentar toast ao usuário em caso de API offline sem sair da página da aplicação;
- Utilizar o mecanismo de rotas para trazer qualquer variação da URL para a aplicação;
- Apresentar popup de confirmação para operação de remoção de usuário;

#### Orientações:

- Utilize a biblioteca de componentes da sua preferência
- Ao final, compartilhe o repositório github da sua solução com marcos.aquino@easytech.com.br

# Sugestão de layout:



#### Questão 2

Elaborar uma API REST para o CRUD de usuários utilizando Node ou NestJS, em que a entidade que representará o usuário deverá possuir os seguintes atributos:

- id
- nome
- telefone
- email
- data de criação
- senha

A API deverá atender aos seguintes requisitos:

- Autenticação por basic auth
- Os dados dos usuários deverão ser persistidos em banco de dados
- Implementar rotas para:
  - Cadastrar um novo usuário
  - Listar todos os usuários
  - Listar um usuário a partir do seu telefone, e-mail ou parte do nome
  - Editar um usuário
  - Excluir um usuário
  - Alterar a senha de um usuário
  - Login do usuário (rota em que é informado usuário e senha e retornado o objeto do usuário caso ele exista no banco)
- A senha do usuário deverá ser salva no banco em uma forma segura

## **Orientações**

- Utilize Node/Nest em conjunto com Express
- Utilize o banco de dados e ORM de sua preferência
- Utilize o banco de dados com o arquivo docker-compose disponível no repositório do frontend, ou então, utilize um provedor público de sua preferência
- Nesta soluções serão aceitos Bancos SQL ou noSQL
- Ao final, compartilhe o repositório github da sua solução com marcos.aquino@easytech.com.br