

1. Introdução

O objetivo deste teste é instigar os candidatos a resolver um desafio em tempo determinado, a fim de identificar qualidades e fraquezas técnicas para melhor direcionamento interno e onboarding de acordo com as práticas e tecnologias utilizadas na Brainny Smart Solutions.

2. Proposta

Um cliente (Fictício) da Brainny Smart Solutions realizou uma solicitação para o desenvolvimento de dois produtos. O primeiro consiste em um sistema de Registro de Pontos e o segundo uma Landing Page para a divulgação e comercialização deste mesmo sistema. A nossa tarefa será o desenvolvimento deste dois produtos.

2.1. Projeto do sistema

Esse projeto consiste na implementação de um sistema para o gerenciamento de **Registros de Ponto** dos colaboradores de uma empresa.

A necessidade é que tenhamos um ambiente WEB administrativo, para o gerenciamento do sistema, como um todo, e principalmente os registros de horário dos colaboradores e um ambiente WEB, no qual cada colaborador efetuará autenticação e realizará o registro do seu ponto diariamente.

O sistema deverá atender os seguintes Requisitos Funcionais e Não Funcionais:

Requisitos Funcionais

[RF01] - Gerenciar registros de horários;

[RF02] - Registrar horário de entrada e saída do colaborador;

[RF03] - O usuário Administrador poderá visualizar os registros dos colaboradores.

Requisitos Não Funcionais

[RNF01] - Somente os Colaboradores podem bater o ponto;

[RNF02] - Apenas o Administrador do sistema poderá visualizar a lista com os registros de todos os Colaboradores no dashboard do sistema;

[RNF03] - O sistema utilizará autenticação com JWT;

[RNF04] - Desenvolvimento de cliente WEB em React;

[RNF05] - Efetuar integração com API GraphQL fornecida utilizando Apollo Client.

Dicas para o projeto

Não é obrigatório seguir nenhum desses pontos, mas servirão para fácil adaptação ao cotidiano da Brainny.

- Utilizar uma biblioteca de UI ([Chakra Ui](#), [Material UI](#), [Reactstrap](#));
- Utilizar gitmoji para versionamento;
- Escrever código em inglês;
- Escrever código limpo e auto documentado;
- Aplicação de padrões de desenvolvimento.

Diferenciais de projeto

- Desenvolvimento com Typescript;
- Aplicação do padrão Atomic Design;
- Utilização de Codegen;
- Implementar arquitetura escalável com reutilização de componentes;
- Demonstrar boas práticas na utilização de react hooks;
- Implementação de Custom Hooks

2.2. Landing Page

Essa Landing Page tem como objetivo fazer a divulgação e a comercialização do produto, portanto, o seu design foi pensado da melhor forma possível para alcançar estes objetivos.

Levando estes aspectos em consideração, seja o mais cuidadoso possível na reprodução do design em código.

- Tenha atenção aos mínimos detalhes;
- Seja fiel ao design no desenvolvimento;
- Considere a possibilidade de acesso na landing page através de dispositivo mobile sem que haja quebras no design (design responsivo);

3. Documentação

3.1. Design

O design do projeto do sistema e da landing page encontra-se disponível em:

https://www.figma.com/file/YiK4fl7luc1IUUZMZ3bbog/Teste_Frontend_J%C3%BAnior_Brainny_Register?node-id=15%3A202

É essencial seguir o design de telas de forma criteriosa e a hierarquia de componentes apresentados no projeto.

3.2. Diagrama ER

Um ER é apresentado a seguir para servir de norteador do projeto.



3.3. API GraphQL

Será necessário a integração do sistema WEB com uma API do tipo GraphQL, que encontra-se disponível em:

<http://test.frontend.api.brainny.cc/graphql>

Os usuários do sistema já encontram-se cadastrados na API.

Na tabela abaixo temos as credenciais do Usuário Administrador (role admin), responsável por monitorar o sistema através da dashboard e os Usuários Colaboradores (role user) que efetuam o registro do ponto diariamente.

Login	Senha	Tipo
admin@brainny.cc	adminregister	Administrador
maria@brainny.cc	mariaregister	Colaborador
joao@brainny.cc	joaoregister	Colaborador
amanda@brainny.cc	amandaregister	Colaborador

3.4. Estrutura de páginas

Routes	Conteúdo
/	Landing Page
/login	Área de Login (administrador e colaborador)
/dashboard	Listagem de Pontos (administrador visualiza ponto de todos os colaboradores) Página deve ser acessível somente para administradores (role ADMIN)
/meus-registros	Listagem de Pontos (colaborador visualiza somente seus pontos) Colaborador bate Ponto (modal) Página deve ser acessível somente para colaboradores (role USER)

3.5. Documentação de Apoio

- [react.js] <https://reactjs.org/docs/getting-started.html>
- [apollo client] <https://www.apollographql.com/docs/react/>
- [apollo client react hooks] <https://www.apollographql.com/docs/react/api/react/hooks>
- [atomic design] <https://bradfrost.com/blog/post/atomic-web-design/>
- [chakra-ui] <https://chakra-ui.com/getting-started>
- [graphql codegen] <https://www.the-guild.dev/graphql/codegen/docs/getting-started>
- [gitmoji] <https://gitmoji.dev/>

4. Entrega

- O teste/projeto deverá ser entregue em repositório git com as devidas instruções necessárias para executar o projeto, juntamente com a descrição das tecnologias utilizadas. Tudo documentado no Readme do repositório.
- Juntamente deverá ser entregue um vídeo, apresentado pelo próprio candidato, demonstrando todo o fluxo e regras de negócio do sistema e landing page, exemplos de responsividade e os principais pontos de código que julgar importante destacar.
- O candidato deverá encaminhar um e-mail para **contato@brainny.cc** com o link do repositório, link do vídeo e descrever como foi a sua experiência e as dificuldades encontradas no processo de desenvolvimento desse projeto.

**CRIATIVIDADE E
INOVAÇÃO
É O QUE NOS MOVE.**