

Desafio do desenvolvedor mobile

O desafio consiste na criação de um simples aplicativo, seguindo os padrões que trabalhamos na empresa ou que vamos seguir nos projetos futuros.

Resumo da aplicação: O aplicativo consiste em mostrar uma lista de marcas de carro, podendo acessar uma marca específica mostrando assim os seus respectivos modelos.

Recursos obrigatórios: O aplicativo deverá apresentar as seguintes telas:

- SignIn (Login para acessar as outras telas)
- Home (Listagem das marcas de carros)
- Model (Listagem dos modelos de carros ao clicar na marca específica)

Bibliotecas/Tecnologias/Padrões de Projeto: O aplicativo deve ser criado utilizando os seguintes recursos obrigatoriamente:

- Expo
- Typescript
- Context API (Gerenciar dados de Login)
- Async Storage (Gerenciar dados de Login)
- NativeWind ou Styled Components
- React Navigation (Livre para escolher o tipo de navegação)
- Fetch Api ou Axios (Para consulta das API's)

Obs: você sai na frente se utilizar React Hook Form (Para os inputs de Login). Além disso, você é livre para adicionar bibliotecas que achar necessárias. O estilo do aplicativo fica por sua conta, seja criativo.

Fluxo do aplicativo: O usuário deverá fazer login e cair na tela “Home” listando todas as marcas de carros. Apresentar também nesta tela o nome do usuário(puxando do context) e um botão para sair do app. Ao clicar em uma marca, deverá ser encaminhado para a tela de “Model” listando todos os modelos de carros da marca escolhida.

Recursos de API:

Método para fazer login:

- **Url:** <https://test-api-y04b.onrender.com/signIn>
- **Método:** POST
- **Parâmetros requeridos no body:** user e password;
- **User:** teste / **Password:** 123

```
1 = {  
2   "user": "teste",  
3   "password": "123"  
4 }
```

Retorno para **sucesso**:

```
1 {
2   "error": false,
3   "user": {
4     "id": 1,
5     "name": "John Smith",
6     "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6Ikp
7   }
8 }
```

Retorno para **erro**:

```
1 {
2   "error": true,
3   "message": "Usuário e/ou senha inválido(s)."
```

- Em caso de **sucesso**, você deve salvar os dados retornados do usuário no Async Storage e dentro do seu Contexto e redirecionar o usuário para a tela de Home.
- Em caso de **erro**, você deve mostrar a mensagem de erro retornada da API.

Método para listar as marcas:

- **Url:** <https://parallelum.com.br/fipe/api/v1/carros/marcas>
- **Método:** GET
- Este método vai retornar código e nome;

Método para listar os modelos da marca escolhida:

- **Url:** <https://parallelum.com.br/fipe/api/v1/carros/marcas/{código da marca}/modelos>
- Ex: <https://parallelum.com.br/fipe/api/v1/carros/marcas/59/modelos>
- **Método:** GET
- Este método vai retornar código e nome;
-

Recomendação:

- Código em inglês.

Avaliação:

- Estrutura, organização e componentização.
- Boas práticas de programação.

Entrega:

- Você deve publicar o desafio em um repositório público do GitHub e enviar o link para o seguinte WhatsApp: 49 9 9838-1023.