

Desafio Mobile Developer - Agriness

Introdução

Bem-vindo ao Desafio da squad Mobile da Agriness.

O objetivo deste desafio é entender como você aplicaria alguns conceitos técnicos que atualmente são utilizados por nós.

A seguir descrevemos alguns requisitos de um problema que você deve resolver utilizando as tecnologias da nossa stack.

Problema

Nosso cliente deseja administrar uma granja de animais e para isso ele precisa acessar a lista de animais da sua granja e o perfil de cada animal.

Nosso trabalho é construir uma API que nos forneça os dados dos animais. A API deve ser consumida pelo aplicativo mobile que será utilizado pelo dono da granja.

Desafio

Criar um aplicativo utilizando React Native, nós recomendamos que você use o Expo para facilitar o desenvolvimento do app.

Criar uma API preferencialmente utilizando Node.js com endpoints que permitam o acesso aos dados dos animais. Nós recomendamos que você use Node com typescript, mas não é obrigatório.

Em anexo a esse teste existe o arquivo animais.json com os dados dos animais para guiar a construção da api.

O aplicativo deve ter:

1. Tela de Listagem de animais
 - Uma lista de pelo menos 10 animais.
 - Deve ter paginação, scroll infinito ou alguma outra abordagem.
 - Escolha os campos que você ache interessante exibir para cada animal.
 - Deve permitir excluir um animal da lista.
 - Deve permitir buscar por nome ou localização do animal.
 - Pode usar uma biblioteca para filtrar ou js puro.
 - A busca deve ser offline.

2. Tela de Perfil do Animal

- Escolha os campos que você ache interessante exibir para cada animal.
 - Lembre-se de usar a criatividade para componentizar e estilizar essa tela, ela deve ficar simples e objetiva.
- Deve permitir alterar informações do animal
 - Campos sugeridos: Nome e status.
 - A edição pode ser feita em outra tela.

3. Tela de login (Opcional)

- Deve ter um campo de e-mail
 - Deve validar se o e-mail é válido.
- Campo de senha
 - Deve validar se a senha digitada é maior ou igual a 4.
- A tela de login é a única tela pública, as demais são privadas (necessário fazer login), sendo assim, lembre-se de fazer logout também.
- Após o login o app deve navegar até a tela de Listagem de animais.
- Se abrir o app e ele já estiver logado anteriormente, o app deve abrir na tela de Listagem de Animais.

Diferencial do app (opcional)

- Redux-saga
- Offline first
- Login

A api deve ter:

1. Endpoints que permitam a manipulação dos dados dos animais.
 - GET Listagem
 - GET by id
 - PUT
 - DELETE
2. Endpoint que permita o login na aplicação. (opcional)
3. Deve persistir os dados em um banco de dados (Relacional ou NoSQL de sua preferência)

Diferencial da api (opcional)

- RxJS
- Swagger
- Docker

Observações

1. Lembre-se que ao fazer as validações, o app deve fornecer um feedback visual ao usuário do que está acontecendo.
2. Não se esqueça de escrever testes automatizados nos dois projetos. Nós escrevemos testes, gostaríamos de saber se você escreve também.
3. Use os estilos e cores de sua preferência. Preferimos que não utilize libs de estilos, gostaríamos de ver suas habilidades criando componentes.
4. Fique a vontade para usar as bibliotecas que você achar interessante e os padrões de desenvolvimento que você acredita serem mais apropriados para ambos os projetos. Recomendamos que não faça uso de frameworks que fazem tudo "automaticamente", queremos ver sua forma de codificar.
5. Crie um arquivo readme em cada um dos projetos explicando passo a passo como rodar e testar eles e, se necessário, escreva detalhes das suas implementações e informações que você ache interessante compartilhar com a gente.
6. Entregue o projeto em um repositório Git (Github, BitBucket, Gitlab, etc)

Você tem liberdade na tomada de decisões que entender serem as melhores e também para implementar outras funcionalidades que considerar interessantes.

O desafio serve para mostrar as suas habilidades, fique a vontade para ir além... (vamos avaliar isso também!).

Estrutura dos dados dos animais (animais.json)

Código	Descrição
Id	Valor no formato UUID.
Nome	Nome do animal na granja.
Tipo do animal	Constante com os seguintes valores: POULTRY, SWINE
Status do animal	Possíveis valores: 0 - Ativo, 1 - Inativo, 2 - Vendido, 3 - Descartado.
Localização	Texto com a descrição da localização do animal na granja.
Data de nascimento	Data de nascimento do animal.
Entrada no plantel	Data da compra do animal.
Peso na compra	Valor em kg.
Raça	Identificador da raça do animal.

Código de rastreamento	Identificação para rastreamento do animal.
Fase de produção	Fase de produção ao qual o animal pertence. Exibir a descrição.
Tipo da granja	Tipo da granja ao qual o animal pertence. Exibir a descrição.

Avaliação

1. O desafio deve ser enviado por e-mail para a pessoa que estiver em contato com você com os links para os repositórios já no início do projeto para efetuarmos o acompanhamento e, também, avisado ao término do mesmo.
2. Iremos te avaliar pelas arquiteturas dos projetos, qualidade do código, entendimento das regras de negócio, atenção aos detalhes, dedicação e compromisso com o desafio e o quão preparado este estaria para ser rodado em produção.
3. Depois que corrigirmos o desafio, você receberá um e-mail de feedback em relação às próximas etapas do processo seletivo.
4. Boa sorte :)