

AVALIAÇÃO FINAL – PIDM – 2019.2

Nome:

Crie um projeto onde a primeira tela é um menu com botões para cada uma das cinco questões abaixo. Use, como nome dos botões, “Questão 1”, “Questão 2”, etc. Use, **obrigatoriamente**, a biblioteca **react-navigation-stack** e os nossos componentes vistos em sala de aula, na pasta **commons** (Cartao, CartaoItem, MeuBotao, etc.).

Nas questões que envolvem o **Firebase**, você deve criar um NOVO projeto no Firebase, coleções e diretórios. Não reuso os que já existem.

O que deve ser enviado: a pasta SRC, contendo apenas os fontes da prova e print-screens da tela do Firebase, provando que você criou um novo projeto e coleções.

1 -) Implemente uma tela onde serão pedidos os seguintes dados de um **Ator**:

- Nome
- Idade
- Talento (dança, canta, interpreta, etc...)
- Salário

Na mesma tela, deve haver um botão “Adicionar” que quando tocado deverá armazenar os dados acima do **Ator** no **firebase.firestore()**. O botão deverá ser substituído por um “spinner” (MeuSpinner) enquanto a operação de adicionar o ator no **firebase.firestore()** estiver em andamento. O botão voltará ao normal quando a operação remota finalizar. Indique, em um **alert**, uma mensagem de sucesso ou erro.

2 -) Implemente uma tela que lista todos os atores salvos no **firebase.firestore()**. Mostre os dados dos atores usando o **MeuLabelText** (Nome, Idade e Talento). Abaixo de CADA ator deverá haver um botão “Calcular Imposto” que ao ser tocado deverá calcular o imposto (**15% do Salário**) do ator **cujo botão foi tocado**, mostrando-o em um **alert**.

3 -) Implemente em uma tela um botão que irá fazer um upload de uma imagem qualquer no **firebase.storage()**, fazendo uso do componente **ImagePicker**. Ao mesmo tempo, salve as URLs da imagem no **firebase.database()**.

Ainda, **na mesma tela**, liste TODAS as imagens que estão salvas no **firebase.database()**, usando para isso um **FlatList**. Nessa questão, você ainda deve usar:

- Um spinner (**MeuSpinner**) para feedback visual no botão de upload e também enquanto a **FlatList** está sendo carregada.
- O componente **ProgressiveImage**, para carregar a imagem gradualmente.

4 -) Implemente uma tela para **EDITAR** os dados das entidades salvas pela Questão 1. Ao terminar a edição, sua aplicação deverá **REDIRECIONAR** (**this.props.navigation.navigate...**) para a tela de listagem implementada na Questão 2.

5 - Implemente uma tela que faz um fetch dos objetos em “<https://jsonplaceholder.typicode.com/users>” e exibe, em um **FlatList**, apenas os usuários (**name**) que iniciem com a “**D**”, “**C**” ou “**K**”. Exemplos: Clementina DuBuque, Kattie Turnpike, Dayna Park...