

SEGUNDA AVALIAÇÃO PARCIAL – 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.).

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

- Marca do Processador (Intel ou AMD, por exemplo)
- Quantidade de Memória (Number)
- Clock do Processador (Number)
- Cache L1 (Number)

Na mesma tela, deve haver um botão “Adicionar” que quando tocado deverá armazenar os dados acima do **Computador** no **firebase.firestore()**. O botão deverá ser substituído por um “spinner” (MeuSpinner) enquanto a operação de adicionar o computador 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 Computadores salvos no **firebase.firestore()**. Mostre os dados dos computadores usando o **MeuLabelText** (Marca, Memória, Clock e Cache). Abaixo de CADA computador deverá haver um botão “Calcular Velocidade Máxima” que ao ser tocado deverá calcular a **velocidade ((memória * processador)/(cache+1))** do computador **cujo botão foi tocado**, mostrando-a 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 uso do **fetch** para ler TODOS os “users” da URL <https://jsonplaceholder.typicode.com/users>. Na mesma tela, exiba apenas as informações, para cada user:

- “email”
- “username”
- “address.city”
- “company.catchFrase”

