

PIDM – 2023.2

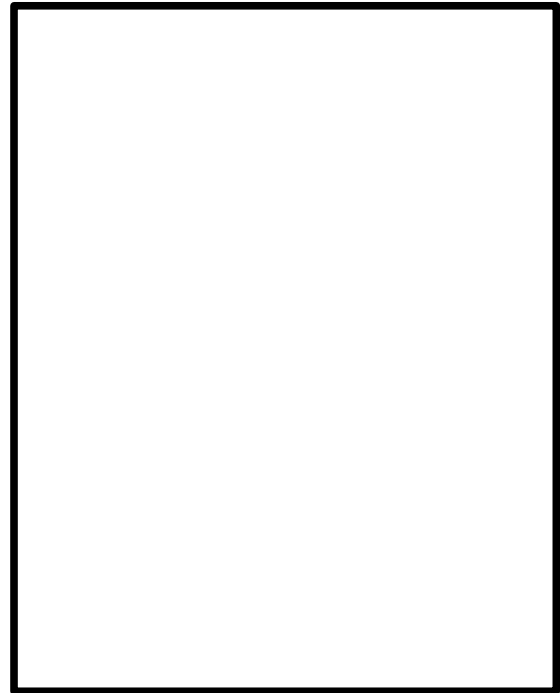
1) – Diga se a afirmação abaixo está correta ou errada e JUSTIFIQUE em ambos os casos:

“React Native utiliza componentes nativos em vez de componentes da Web como blocos de construção, existindo dois tipos de dados que controlam um componente: **state**, definido pelo pai e fixado durante todo o tempo de vida de um componente; e **props**, utilizado para os dados que irão mudar.”

2) – Quais as diferenças do uso de **props** e **state** em React? Explique com exemplos.

3) – Seja o código abaixo que representa o JSX de um componente React Native:

```
return (  
  <View>  
    <Text> Some text </Text>  
    <View  
      style={{flexDirection:"row"}}  
    >  
      <Text> Some more text </Text>  
      <Text> Some more text </Text>  
    </View>  
    <View  
      style={{flexDirection:"row",justifyContent:"center"}}  
    >  
      <TextInput  
        style={{  
          borderWidth: 1,  
          width:100  
        }}  
        defaultValue="You can type in me"  
      />  
      <TextInput  
        style={{  
          borderWidth: 1,  
          width:100  
        }}  
        defaultValue="You can type in me"  
      />  
    </View>  
  </View>  
)  
);
```



Desenhe, no retângulo ao lado, como irá se parecer essa tela com o máximo de aproximação do código.

4) – Seja o JSON abaixo:

```
1  const dados = [  
2    {  
3      title:"Terça",  
4      data:[  
5        {icone:"cart",nome:"Pinheiro...", hora:"16:41", valor:274.89},  
6        {icone:"tools",nome:"Netflix.Com", hora:"12:51", valor:74.89}  
7      ]  
8    },  
9    ,  
10   {  
11     title:"25 Set",  
12     data:[  
13       {icone:"medical-bag",nome:"Drogasil", hora:"16:41", valor:274.89},  
14       {icone:"tools",nome:"Netflix.Com", hora:"12:51", valor:74.89}  
15     ]  
16   },  
17   ,  
18   {  
19     title:"24 Set",  
20     data:[  
21       {icone:"cart",nome:"Pag*Bodegas", hora:"16:41", valor:274.89},  
22       {icone:"cart",nome:"Pag*Bodegas", hora:"12:51", valor:74.89}  
23     ]  
24   }  
25 ]
```

Implemente APENAS o código de uma SectionList para mostrar os dados do JSON em tela. Segue um exemplo:

```
<SectionList
  sections={//dados}
  keyExtractor={({item, index}) => //retorna um index}
  renderSectionHeader={
    ({ section }) => {
      return (//código JSX)
    }
  }
  renderItem={
    ({ item, section }) => {
      return (//código JSX)
    }
  }
/>
```

5 -) Seja o layout abaixo:

O diagrama mostra um formulário com os seguintes elementos:

- Um campo de texto rotulado "TÍTULO".
- Um campo de imagem rotulado "FOTO.PNG".
- Um campo de texto rotulado "NOME E SOBRENOME".
- Um campo de texto rotulado "IDADE".
- Um campo de texto rotulado "CURSO".
- Dois botões rotulados "VOLTAR" e "EDITAR".

Sejam TÍTULO, NOME E SOBRENOME, IDADE e CURSO textos; FOTO.PNG uma imagem e VOLTAR e EDITAR botões, implemente o código com FLEBOX e JSX para gerar um layout semelhante à figura.

6 -) Em React.Js, como são chamadas as entradas que são passadas na **criação** dos componentes React, usando uma convenção de nomenclatura semelhante aos atributos de tag HTML.

7 -) Qual a saída do código abaixo?

```
const make = 'Ford';
const model = 'Mustang';
const car = { make, model };
console.log(car);
```

8 -) Das afirmações abaixo, qual **NÃO** é uma regra para Hooks em React?

- a -) Hooks não podem ser condicionais
- b -) Hooks só podem ser chamados dentro de componentes do tipo Função
- c -) Hooks só podem ser chamados no “top-level” do componente do tipo
- d -) Função Hooks podem ser chamados dentro de um componente Classe ou Função

9 -) Qual, das formas abaixo, é o jeito **correto** de importar um componente React?

- a -) import React.Component from “react”
- b -) import {Component} from “react”
- c -) import [Component] from “react”
- d -) import Component from “react”

10 -) Como você pode combinar esses dois arrays usando o operador de “spread”? Marque a alternativa correta.

```
const array1 = [1, 2, 3];  
const array2 = [4, 5, 6];
```

- a -) `const combined = array1 + array2`
- b -) `const combined = ...array1 + ...array2`
- c -) `const combined = [array1,array2]`
- d -) `const combined = [...array1,...array2]`