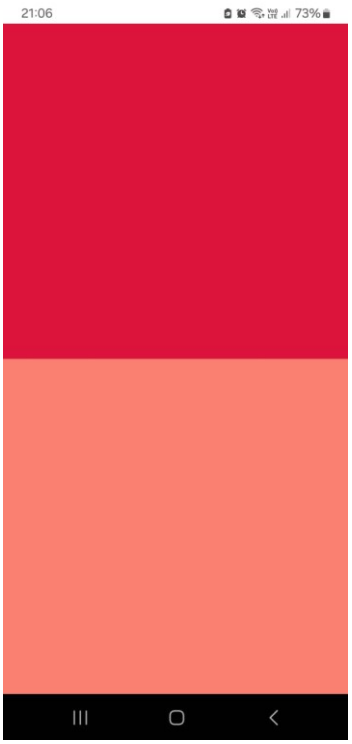
Lista de Exercícios – Expo / React Native

Professor: André Olímpio

Disciplina: Programação para Dispositivos Móveis I

Para cada exercício, adicione uma tela na pasta *screens* do projeto, no qual o componente *App* chama o *Home*.

Exercício 1 – Fazer um aplicativo RN com a tela dividida em duas partes na vertical.



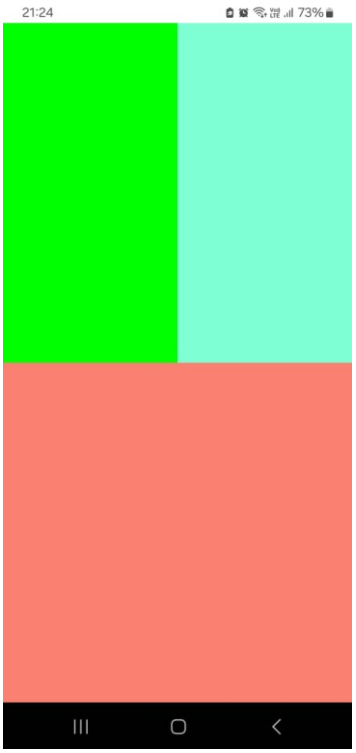
Requisitos:

- Crie um componente na pasta screens de nome *Um*;
- Altere o componente App para chamar o componente *Um*;
- Excluir da tela a status bar do dispositivo. Veja na figura ao lado que a *StatusBar* não faz parte da tela do aplicativo

Dicas:

- Crie um componente *container* pai utilizando *View*;
- Crie dois componentes filhos utilizando *View*;
- Utilize a propriedade *flexDirection* com valor *column* para tornar o eixo principal na coluna. Desta forma, os elementos filhos serão posicionados um abaixo do outro;
- Utilize a propriedade *flex* com valor 0.5 (50%) em cada componente filho;
- Utilize a propriedade *backgroundColor* com os valores *crimson* e *salmon*;
- Importe *Constants* do módulo *expo-contants* para ter acesso a informações constantes do dispositivo (https://docs.expo.dev/versions/latest/sdk/constants/);
- Utilize a propriedade *paddingTop* com valor *Constants.statusBarHeight* na *View* pai para deslocar o início da parte superior do aplicativo.

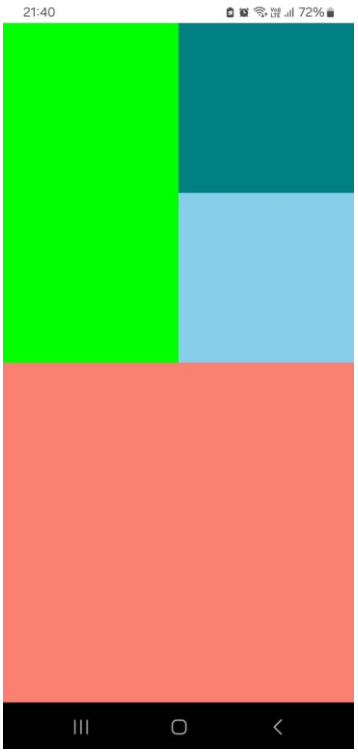
Exercício 2 – Alterar o aplicativo do Exercício 1 para dividir a tela assim como é mostrado ao lado.



Dicas:

- Adicione dois componentes *View* filhos do componente que possui a cor *crimson* do Exercício 1;
- Utilize a propriedade *flexDirection* com valor *row* no componente que possui a cor *crimson* para tornar o eixo principal na linha. Desta forma, os elementos filhos serão posicionados um à direita do outro;
- Utilize a propriedade *flex* com valor 0.5 (50%) em cada componente filho;
- Utilize as cores *lime* e *aquamarine*.

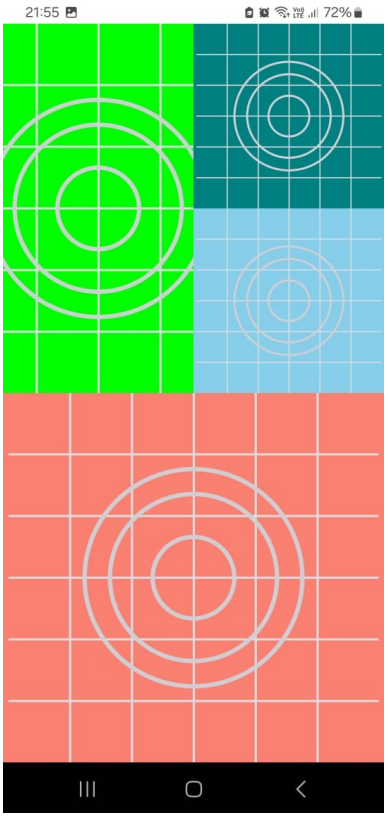
Exercício 3 – Alterar o aplicativo do Exercício 2 para dividir a tela assim como é mostrado ao lado.



Dica:

- Utilize as cores *teal* e *skyblue*.

Exercício 4 – Adicionar a imagem no aplicativo do Exercício 3.



Dica:

- Utilize o componente Image para exibir a imagem (https://reactnative.dev/docs/image);
- Use a instrução a seguir para importar o arquivo png na variável logo:

  import logo from "../../assets/adaptive-icon.png";
- Utilize a propriedade *alignSelf:center* para alinhar a imagem no centro;
- Utilize a propriedade *flex:1* para a imagem ocupar toda a área;
- Utilize a propriedade resizeMode com valor contain para adequar as dimensões da imagem (https://reactnative.dev/docs/image#resizemode).

Observação:

- A importação do arquivo PNG na variável logo pode causar erro.

  import logo from "../../assets/adaptive-icon.png";

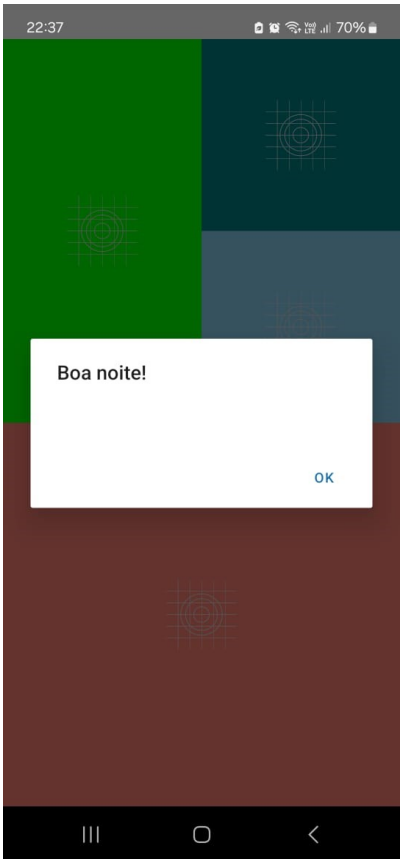
  O erro ocorre porque o TypeScript, utilizado em *arquivos .tsx*, não tem informações de tipo para módulos de imagem como PNGs, o que é necessário para que o *linting* e o sistema de tipos do TS reconheçam e processem corretamente esses arquivos.

Para resolver o erro crie um arquivo chamado *declarations.d.ts* (ou qualquer nome com a extensão *.d.ts*) na pasta *types*. Adicione o seguinte código no arquivo *declarations.d.ts* para declarar módulos de imagem PNG:

```
declare module "*.png" {
  const value: any;
  export default value;
}
```

Isso informa ao TS que qualquer importação de arquivos com a extensão .png deve ser tratada como um módulo que exporta qualquer valor.

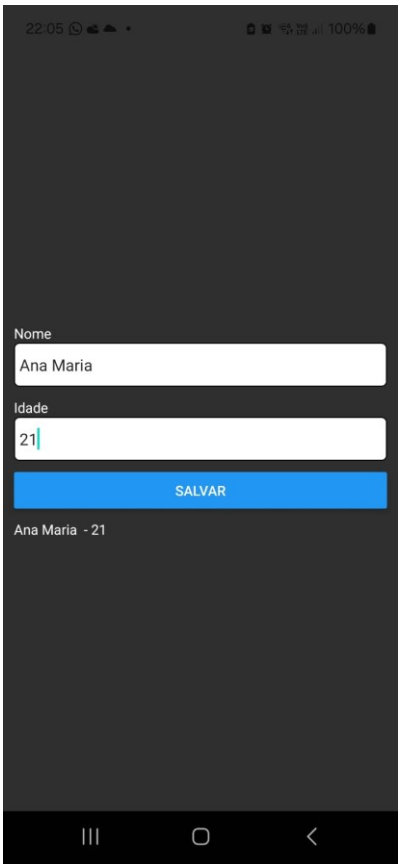
Exercício 5 – Transformar as imagens do Exercício 4 em botões clicáveis. Ao clicar no botão será exibida uma mensagem de alerta com o texto *"Boa noite!"*.



Dicas:

- Utilize o componente *TouchableOpacity* para criar uma área clicável (https://reactnative.dev/docs/touchableopacity);
- Coloque as dimensões das imagens em 64x64;
- Utilize a propriedade *justifyContent: "center"* e *alignItems: "center"*, nos componentes pais, para centralizar as imagens;
- Utilize o componente *Alert* para exibir a janela de alerta ao clicar no botão *TouchableOpacity* (https://reactnative.dev/docs/alert);
- Utilize o evento *onPress* do componente *TouchableOpacity* para exibir a janela de alerta.

Exercício 6 – Fazer um aplicativo com dois campos de entrada, assim como é mostrado na imagem ao lado.



Requisitos:

- O campo idade deverá habilitar o teclado numérico;
- Ao clicar no botão *Salvar* deverá ser exibido o nome e a idade abaixo do botão.

Dica:

- Utilize a propriedade *keyboardType="numeric"* no componente *TextInput* para habilitar o teclado numérico.

Exercício 7 – Fazer um aplicativo com os campos de entrada para e-mail e senha, assim como é mostrado na imagem ao lado.

Requisitos:

- Utilize *SafeAreaView* como componente base da estrutura da tela (https://reactnative.dev/docs/safeareaview);
- O campo *e-mail* deverá habilitar o teclado para e-mail;
- O campo *senha* deverá mascarar os caracteres;
- Ao clicar no botão *Logar* deverá ser exibido o e-mail e senha abaixo do botão.

Dica:

- Utilize as propriedades no campo de e-mail:

  autoCapitalize="none"

  autoComplete="email"

  autoCorrect={false}

  keyboardType="email-address"
- Utilize as propriedades no campo de senha:

  secureTextEntry={true}

  maxLength={8}
- Utilize o componente *TouchableOpacity* para criar os botões.

Exercício 8 – Alterar a tela do Exercício 7 para adicionar o campo de confirmação de senha, assim como é mostrado na imagem ao lado.

Requisito:

- Criar uma moldura em volta dos campos de entrada. Essa moldura deverá estar centralizada e ter largura máxima de 270.

Exercício 9 – Alterar a tela do Exercício 8 para adicionar o campo de escolha, assim como é mostrado na imagem ao lado.

Requisitos:

- O campo de escolha deverá ter as opções *"Administrador", "Gestor"* e *"Usuário"* com os valores "*admin", "manager"* e *"user",* respectivamente;
- O campo de escolha deverá ter como valor padrão *"manager".*

Dica:

- Sugere-se usar o componente *Picker* (https://www.npmjs.com/package/@react-native-picker/picker).

Exercício 10 – Alterar a tela do Exercício 9 para adicionar o campo de seleção, assim como é mostrado na imagem ao lado.

Requisitos:

- O *trackColor* deverá receber as cores { false: "#e77878", true: "#94df83" };
- O *thumbColor* deverá receber as cores de acordo com a propriedade de estar logado {logado ? "#47eb22" : "#ed1111"}.

Dica:

- Utilize o componente *Switch* (https://reactnative.dev/docs/switch).

Exercício 11 – Fazer uma tela de abertura do aplicativo, assim como é mostrado na imagem ao lado.

Requisitos:

- Exibir a imagem da Fatec com 140x140;
- Alinhar os botões em duas colunas.

Dica:

- Use o componente *Image* (https://reactnative.dev/docs/image).