

Material do Projeto da Disciplina Análise e Projeto de Software – Front-end React Native

Autor: davialmda

1 O que é o React Native?

O React Native permite desenvolver aplicativos mobiles usando **JavaScript** e a biblioteca **React**. Ele cria interfaces para Android e iOS usando componentes, estado e propriedades. O material apresenta um exemplo de **Tela de login** e os principais pilares para o front-end.

Código de Login em React Native

```
1 import React, { useState } from 'react';
2 import { View, TextInput, StyleSheet, Text, TouchableOpacity, Alert }
  from 'react-native';
3
4 export default function Login({ onNavigate }) {
5   const [email, setEmail] = useState('');
6   const [senha, setSenha] = useState('');
7
8   const fazerLogin = () => {
9     if (email === 'usuario@example.com' && senha === '123456') {
10       onNavigate('inicio');
11     } else {
12       Alert.alert('Erro', 'Credenciais inv lidas ');
13     }
14   };
15
16   return (
17     <View style={styles.container}>
18       <Text style={styles.title}>Login</Text>
19
20       <TextInput
21         style={styles.input}
22         placeholder="Digite seu e-mail"
23         value={email}
24         onChangeText={setEmail}
25         keyboardType="email-address"
26         placeholderTextColor="#f77ca9"
```

```

27     />
28     <TextInput
29       style={styles.input}
30       placeholder="Digite sua senha"
31       secureTextEntry
32       value={senha}
33       onChangeText={setSenha}
34       placeholderTextColor="#f77ca9"
35     />
36
37     <TouchableOpacity
38       style={styles.button}
39       onPress={fazerLogin}
40     >
41       <Text style={styles.buttonText}>Entrar</Text>
42     </TouchableOpacity>
43
44     <TouchableOpacity
45       style={styles.button}
46       onPress={() => onNavigate('home')}
47     >
48       <Text style={styles.buttonText}>Voltar</Text>
49     </TouchableOpacity>
50   </View>
51 );
52 }
53
54 const styles = StyleSheet.create({
55   container: {
56     flex: 1,
57     justifyContent: 'center',
58     alignItems: 'center',
59     backgroundColor: '#f7a8b8',
60     padding: 20,
61   },
62   title: {
63     fontSize: 32,
64     fontWeight: 'bold',
65     marginBottom: 40,
66     color: 'fff',
67   },
68   input: {
69     width: '80%',
70     padding: 15,
71     marginVertical: 10,
72     borderWidth: 1,
73     borderColor: '#f1c0d6',

```

```

74     borderRadius: 10,
75     fontSize: 16,
76     backgroundColor: '#fff',
77     color: '#333',
78     shadowColor: "#000",
79     shadowOffset: { width: 0, height: 4 },
80     shadowOpacity: 0.1,
81     shadowRadius: 4,
82   },
83   button: {
84     marginTop: 20,
85     width: '80%',
86     paddingVertical: 15,
87     borderRadius: 25,
88     backgroundColor: '#f77ca9',
89     alignItems: 'center',
90     justifyContent: 'center',
91     elevation: 3,
92   },
93   buttonText: {
94     fontSize: 18,
95     color: '#fff',
96     fontWeight: '600',
97   },
98 });

```

Explicações passo a passo

1. **useState:** Cria estado local para armazenar o email e a senha:

- `const [email, setEmail] = useState('')`
- `const [senha, setSenha] = useState('')`

2. **Função de login:** Verifica se as credenciais estão corretas e chama `onNavigate` para mudar de tela. Se forem inválidas, exibe um `Alert`.

3. **Inputs:** `TextInput` recebe `value` e `onChangeText` para sincronizar com o estado. `placeholder` e `placeholderTextColor` exibem dicas visuais ao usuário.

4. **Botões:** `TouchableOpacity` com `onPress` para executar funções (login ou voltar). O estilo torna os botões arredondados e coloridos, pensando no usuário.

5. **Estilos:** - `container`: deixa no centro o conteúdo e aplica fundo rosa pastel.
 - `input`: bordas arredondadas, sombra e fundo branco. - `button`: cor rosa , borda arredondada. - `title`: texto grande, negrito e branco.

Exercício para Contribuição – Sprint 1

User Story relacionada: US01 – Cadastro no aplicativo.

Objetivo: Criar uma tela de cadastro funcional em React Native, permitindo que o usuário coloque nome, email e senha.

Tarefas

1. Criar o arquivo **Cadastro.js**.
2. Adicionar campos de entrada para nome, email e senha.
3. Verificar se todos os campos foram preenchidos; exibir alerta caso, não tenha sido preenchido.
4. Salvar as informações em uma lista simulando um banco de dados.
5. Exibir mensagem de sucesso após o cadastro.
6. Comitar e subir para a branch correspondente no GitHub.
7. Atualizar status da US no Notion.

User Story relacionada: US02 – Login no aplicativo.

Objetivo: Criar uma tela de login funcional em React Native, aplicando conceitos de componentes, estado e eventos (Aprofundamento na **1.1 Documentação traduzida**).

Tarefas

1. Criar o arquivo **Login.js** no repositório front-end.
2. Adicionar campos de texto para email e senha.
3. Implementar verificação de campos preenchidos.
4. Implementar validação das credenciais de teste.
5. Exibir mensagens de sucesso ou erro usando **Alert**.
6. Comitar e subir a implementação na branch da US.
7. Atualizar status da US no Notion após a entrega.

1.1 Documentação traduzida:

<https://github.com/tavaresgerson/reactnatedocbr>

2 Referências

<https://reactnative.dev/docs/getting-started>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

<https://github.com/tavaresgerson/reactnativetodocbr>