## CÓDIGO-FONTE

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

**App.js**

import React from 'react';

import { NavigationContainer } from '@react-navigation/native'; import { createStackNavigator } from '@react-navigation/stack';

import Login from './Páginas/Login'; // IMPORTANDO A TELA Login

import Cadastro from './Páginas/Cadastro'; // IMPORTANDO A TELA Cadastro import Quiz from './Páginas/Quiz'; // IMPORTANDO A TELA Quiz

const Stack = createStackNavigator(); const App = () => {

return (

<NavigationContainer>

<Stack.Navigator initialRouteName="Login">

<Stack.Screen name="Login" component={Login} />

<Stack.Screen name="Cadastro" component={Cadastro} />

<Stack.Screen name="Quiz" component={Quiz} />

</Stack.Navigator>

</NavigationContainer>

);

};

export default App;

Tela de login:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

import React, { useState } from 'react';

import { View, TextInput, Button, StyleSheet, Text, Image, TouchableOpacity} from 'react-native';

import { NavigationContainer } from '@react-navigation/native'; import { createStackNavigator } from '@react-navigation/stack';

const Stack = createStackNavigator(); const Login = ({ navigation }) => {

const [nomeUsuario, setNomeUsuario] = useState(''); const [senha, setSenha] = useState('');

const handleLogin = () => {

if (nomeUsuario === '123' && senha === '123') { navigation.navigate('Quiz'); // Navega para a Tela do Quiz

//LIMPANDO OS CAMPOS DE ENTRADAS DE DADOS

setUsername('');

setPassword('');

} else {

alert('Usuário e/ou Senha inválidos');

}

};

const handleCadastro = () => {

navigation.navigate('Cadastro'); // 'Cadastro' é o nome da tela de destino

};

const clickHandler = () => {

alert('Projeto desenvolvido por Thiago Valladão e Júlia Teles');

};

return (

//backgroundColor - COR DO FUNDO DA TELA

<View style={[styles.container, { backgroundColor: 'lightblue' }]}>

<Image style={styles.imagem} resizeMode="center"

source={require('./imagens/saudelogobg.jpg')}

/>

<Text style={[styles.label1, { marginBottom: -50, textAlign: 'center' }]}> Questionário de Rotina e Bem-Estar

</Text>

{/\* Input Container \*/}

<View style={styles.inputContainer1}>

<View style={styles.inputGroup}>

<Text style={styles.labelusuario}>Usuário</Text>

<TextInput

// AJUSTE DO TAMANHO DA LARGURA DO TEXTINPUT

style={[styles.input, { width: 200, textAlign: 'center' }]} value={nomeUsuario} onChangeText={setNomeUsuario}

/>

</View>

<View style={styles.inputGroup}>

<Text style={styles.label2}> Senha</Text>

<TextInput

// AJUSTE DO TAMANHO DA LARGURA DO TEXTINPUT

style={[styles.input, { width: 200, textAlign: 'center' }]} secureTextEntry

value={senha} onChangeText={setSenha}

/>

</View>

</View>

{/\* FLEXIBILIDADE DA POSIÇÃO DO BOTÃO \*/}

<View style={styles.ButtonContainer}>

<Button title="Login" onPress={handleLogin} />

<br></br>

</View>

<TouchableOpacity activeOpacity={0.7} onPress={clickHandler} style={styles.touchableOpacityStyle}>

<Image source={{ uri:

'https://developerplus.com.br/wp-content/uploads/2021/12/plus\_icon.png',

}}

style={styles.floatingButtonStyle}

/>

</TouchableOpacity>

</View>

);

};

const styles = StyleSheet.create({ container: {

flex: 1,

padding: 20,

},

label1: { color: 'blue', fontSize: 20,

fontFamily: 'Arial',

fontWeight: 'bold', marginBottom: 5,

},

labelusuario: { marginTop: 100, color: 'blue', fontSize: 18, fontFamily: 'Arial', fontWeight: 'bold', marginBottom: 5,

},

label2: { color: 'blue', fontSize: 18,

fontFamily: 'Arial', fontWeight: 'bold', marginBottom: 5,

},

inputContainer1: { alignItems: 'center', marginBottom: 10,

},

inputGroup: { alignItems: 'center', marginBottom: 10,

},

input: { flex: 1,

padding: 10,

fontSize: 18, fontWeight: 'bold', borderWidth: 1, borderColor: '#ccc',

backgroundColor: 'white', textAlign: 'left', marginBottom: 10,

},

imagem: { alignSelf: 'stretch', height: 200,

size: 90,

marginTop: 5,

marginBottom: 10,

},

ButtonContainer: { flex: 1,

justifyContent: 'center', alignItems: 'center', marginTop: -50,

},

touchableOpacityStyle: { position: 'absolute', width: 50,

height: 50, alignItems: 'center',

justifyContent: 'center', right: 30,

bottom: 30,

},

floatingButtonStyle: { resizeMode: 'contain', width: 70,

height: 70,

},

});

export default Login;

## Telas do Quiz: perguntas

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamenteInterface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamenteInterface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

**QuestionScreen.js**

import React, { useState } from 'react';

import { View, Text, TouchableOpacity, StyleSheet, Button } from 'react-native'; import { questions } from './data';

const QuestionScreen = () => {

const [currentQuestionIndex, setCurrentQuestionIndex] = useState(0); const [score, setScore] = useState(0);

const [showResult, setShowResult] = useState(false);

const currentQuestion = questions[currentQuestionIndex]; const { question, options } = currentQuestion;

const handleAnswer = (points) => { setScore(score + points);

// Avance para a próxima pergunta ou mostre o resultado if (currentQuestionIndex < questions.length - 1) { setCurrentQuestionIndex(currentQuestionIndex + 1);

} else { setShowResult(true);

}

};

const Agradecimento = () =>{

return 'Obrigado por Utilizar Nosso App!'

}

const handleRestart = () => { setCurrentQuestionIndex(0); setScore(0); setShowResult(false);

};

// Função para determinar a mensagem com base na pontuação final

const getResultMessage = () => { if (score == 10){

return 'Sua pontuação foi mais baixa do que o ideal.\n Para uma rotina mais saudável, recomendamos beber mais água, praticar exercícios regularmente e adotar uma alimentação mais equilibrada.\n Pequenas mudanças podem fazer uma grande diferença!'

}

else if (score < 17.5) {

return 'Sua rotina tem espaço para melhorias. Recomendamos adicionar mais exercícios físicos à sua semana e aumentar o consumo de frutas e vegetais. Isso ajudará a melhorar seu bem-estar.';

} else {

return 'Parabéns! Sua rotina está indo muito bem.\n Continue mantendo seu estilo de vida saudável e ativo. Lembre-se de manter a consistência e se cuidar!'

}

};

if (showResult) { return (

<View style={styles.container}>

{/\* Exibe a pontuação final \*/}

<Text style={styles.resultText}>Resultado Final: {score} pontos</Text>

{/\* Exibe a mensagem com base na pontuação \*/}

<Text style={styles.messageText}>{getResultMessage()}</Text>

{/\* Exibe uma mensagem de agradecimento pelo uso do App \*/}

<Text style={styles.AgradecimentoText}>{Agradecimento()}</Text>

{/\* Botão para recomeçar o jogo \*/}

<Button title="Recomeçar" onPress={handleRestart} />

</View>

);

}

return (

<View style={styles.container}>

<View style={styles.questionContainer}>

{/\* Pergunta exibida dentro de um contêiner translúcido \*/}

<Text style={styles.questionText}>{question}</Text>

</View>

<View style={styles.optionsContainer}>

{options.map((option, index) => (

<TouchableOpacity key={index} onPress={() => handleAnswer(option.points)}>

{/\* Alternativas de resposta sem os valores \*/}

<Text style={styles.answerText}>{option.text}</Text>

</TouchableOpacity>

))}

</View>

{/\* Exibe a pontuação atual \*/}

<Text style={styles.scoreText}>Pontuação: {score} pontos</Text>

</View>

);

};

const styles = StyleSheet.create({ container: {

flex: 1,

justifyContent: 'center', alignItems: 'center',

backgroundColor: 'rgba(255, 255, 255, 0.8)', // Cor de fundo da tela borderRadius: 10, // Borda arredondada

marginBottom: 20, border: '1px solid #000',

},

questionContainer: {

backgroundColor: 'rgba(210, 210, 250, 0.8)', // Fundo translúcido da pergunta padding: 20,

borderRadius: 10,

marginBottom: 20,

width: '80%',

border: '1px solid #000',

},

questionText: { fontSize: 24, textAlign: 'center',

},

optionsContainer: { width: '80%',

border: '1px solid #000', // Adiciona uma borda ao redor das alternativas padding: 10, // Espaçamento interno

borderRadius: 10, // Borda arredondada

backgroundColor: 'rgba(210, 210, 250, 0.8)', // Cor de fundo da tela

},

answerText: { fontSize: 18,

marginVertical: 5,

padding: 10, // Espaçamento interno borderRadius: 10, // Borda arredondada

backgroundColor: 'rgba(200, 200, 250, 0.8)', // Cor de fundo da tela

},

scoreText: { marginTop: 20,

fontSize: 24,

backgroundColor: 'rgba(210, 210, 250, 0.8)', // Cor de fundo da tela padding: 10, // Espaçamento interno

borderRadius: 10, // Borda arredondada

border: '1px solid #000', // Adiciona uma borda ao redor da Pontuação

},

resultText: {

fontSize: 24,

marginBottom: 20,

backgroundColor: 'rgba(210, 210, 250, 0.8)', // Fundo translúcido da pergunta padding: 20,

borderRadius: 10,

marginBottom: 20,

width: '80%',

border: '1px solid #000',

},

messageText: { fontSize: 18,

marginBottom: 20,

backgroundColor: 'rgba(210, 210, 250, 0.8)', // Fundo translúcido da pergunta padding: 20,

borderRadius: 10,

marginBottom: 20,

width: '80%',

border: '1px solid #000',

},

AgradecimentoText: { fontSize: 18,

marginBottom: 20,

},

});

export default QuestionScreen;

## data.js

// Banco de Questões export const questions = [

{

question: 'Pergunta 1:\nO quão regularmente você pratica atividade física?', options: [

{ text: 'a) Regularmente, pelo menos 3 vezes por semana.', points: 5 },

{ text: 'b) Ocasionalmente, cerca de 1-2 vezes por semana.', points: 3 },

{ text: 'c) Raramente ou nunca pratico atividade física.', points: 2 },

],

},

{

question: 'Pergunta 2:\nComo você descreveria sua ingestão diária de frutas e vegetais?',

options: [

{ text: 'a) Consigo comer uma variedade de frutas e vegetais todos os dias.', points: 5 },

{ text: 'b) Como algumas porções de frutas e vegetais ocasionalmente.', points:

3 },

{ text: 'c) Raramente consumo frutas e vegetais diariamente.', points: 2 },

],

},

{

question: 'Pergunta 3:\nQuantas horas em média você dorme por noite?', options: [

{ text: 'a) Entre 7 e 9 horas, regularmente.', points: 5 },

{ text: 'b) Entre 5 e 6 horas, ocasionalmente.', points: 3 },

{ text: 'c) Menos de 5 horas, com frequência.', points: 2 },

],

},

{

question: 'Pergunta 4:\nVocê costuma lidar bem com o estresse do dia a dia?', options: [

{ text: 'a) Sim, geralmente consigo gerenciar o estresse de forma saudável.', points: 5 },

{ text: 'b) Às vezes tenho dificuldade em lidar com o estresse.', points: 3 },

{ text: 'c) Tenho dificuldade constante em lidar com o estresse.', points: 2 },

],

},

{

question: 'Pergunta 5:\nComo você descreveria sua alimentação geral?', options: [

{ text: 'a) Minha alimentação é equilibrada e inclui uma variedade de grupos alimentares.', points: 5 },

{ text: 'b) Eu como muitos alimentos processados e não saudáveis, mas também como alimentos saudáveis.', points: 3 },

{ text: 'c) Minha alimentação é predominantemente composta por alimentos pouco saudáveis e processados.', points: 2 },

],

},

];

## Quiz.js

import React from 'react';

import { View, StyleSheet } from 'react-native'; import QuestionScreen from '../QuestionScreen';

import { NavigationContainer } from '@react-navigation/native'; import { createStackNavigator } from '@react-navigation/stack';

const Stack = createStackNavigator(); const Quiz = ({navigation}) => {

return (

<View style={styles.container}>

<QuestionScreen />

</View>

);

};

const styles = StyleSheet.create({ container: {

flex: 1,

justifyContent: 'center', alignItems: 'center',

},

});

export default Quiz;